**US Army Corps
of Engineers**
Waterways Experiment
Station

CE-QUAL-RIV1: A Dynamic, One-Dimensional (Longitudinal) Water Quality Model for Streams

User's Manual

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

CE-QUAL-RIV1: A Dynamic, One-Dimensional (Longitudinal) Water Quality Model for Streams

User's Manual

by Environmental Laboratory

U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report

Approved for public release; distribution is unlimited

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under CWIS Work Unit 31595

**US Army Corps
of Engineers**
Waterways Experiment
Station

N

INFORMATION
TECHNOLOGY
LABORATORY

GEOTECHNICAL
LABORATORY

HEADQUARTERS
BUILDING

MAIN
ENTRANCE

HYDRAULICS
LABORATORY

COASTAL ENGINEERING
RESEARCH CENTER

ENVIRONMENTAL
LABORATORY

STRUCTURES
LABORATORY

FOR INFORMATION CONTACT :
PUBLIC AFFAIRS OFFICE
U. S. ARMY ENGINEER
WATERWAYS EXPERIMENT STATION
3909 HALLS FERRY ROAD
VICKSBURG, MISSISSIPPI 29180-6199
PHONE : (601)634-2502

SCALE

AREA OF RESERVATION = 2.7 sq km

PREFACE

This report serves as a user's manual for the dynamic riverine water quality model, CE-QUAL-RIV1. The development of this model was sponsored by Headquarters, U.S. Army Corps of Engineers (HQUSACE), as a part of the Environmental and Water Quality Operational Studies, Work Unit IC.3 (CWIS Work Unit 31595), entitled "Improve and Verify Riverine Water Quality and Ecological Predictive Techniques," and is assigned to the U.S. Army Engineer Waterways Experiment Station (WES) under the purview of the Environmental Laboratory (EL). The HQUSACE Technical Monitors were Mr. Earl Eiker, Dr. John Bushman, and Mr. James Gottesman.

The first draft of the manual, dated September 1982, was prepared by Drs. Keith W. Bedford, Robert M. Sykes, and Charles Libicki of Ohio State University under Contract No. DACW39-82-3548. The original version of the model code was also developed by Ohio State University. This present version of the user's manual is a result of revisions to the 1990 user's manual and reflects model modifications made after 1990. Revisions to the manual were made by Dr. D. M. Griffin, Louisiana Tech University, and Dr. James Martin and Mr. Tim Wool, AScI Corporation. Revisions were also made by Dr. Mark Dortch and Ms. Toni Schneider of the Water Quality and Contaminant Modeling Branch (WQCMB), Environmental Processes and Effects Division (EPED), (EL). The revisions to the model code were made by Ms. Schneider, Dr. Griffin, Dr. Martin, and Mr. Wool.

This work was conducted under the general supervision of Dr. John Harrison, Chief, EL, and Mr. Donald Robey, Chief, EPED, and under the direct supervision of Dr. Dortch, Chief, WQCMB. This report is published under the Corps Numerical Model Maintenance Program.

This user's manual was reviewed by Dr. Patrick Deliman and Ms. Dorothy Tillman of WQCMB. Dr. Robert W. Whalin was Director of WES during publication of this manual. COL Bruce K. Howard, EN, was Commander.

This report should be cited as follows:

Environmental Laboratory. 1995. "CE-QUAL-RIV1: A Dynamic, One-Dimensional (Longitudinal) Water Quality Model for Streams: User's Manual," Instruction Report EL-95-2, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CONVERSION FACTORS, NON-SI TO SI (METRIC)
UNITS OF MEASUREMENT

Non-SI units of measurement used in this report can be converted to SI (metric) units as follows:

Multiply	By	To Obtain
cubic feet	0.02831685	cubic meters
feet	0.3048	meters
inches	25.4	millimeters
miles (U.S. statute)	1.609347	kilometers
square feet	0.09290304	square meters

CE-QUAL-RIV1: A DYNAMIC, ONE-DIMENSIONAL (LONGITUDINAL) WATER QUALITY MODEL FOR STREAMS
USER'S MANUAL

PART I: INTRODUCTION

Background

1. The U.S. Army Corps of Engineers (CE) has major responsibility for regulation of the Nation's streams, rivers, and waterways. This activity often requires resolving issues and concerns with regard to the water quality of these regulated systems. Water quality simulation models can be powerful tools for studying these issues. Of course, to be useful, the water quality model must be properly suited for the problem at hand.

2. Regulated stream systems may include complicating physical features, such as multiple run-of-the-river dams, locks and dams, and reregulation dams. In addition, highly unsteady flows may exist or may be expected as in the planning for peaking hydropower releases. Numerous water quality models are in existence, but most were developed for steady flow conditions and are not appropriate when time-varying flows are to be considered.

3. The model presented herein was originally developed at Ohio State University at the request of the U.S. Environmental Protection Agency (EPA) for the purpose of predicting water quality associated with storm water run-off. Researchers at the U.S. Army Engineer Waterways Experiment Station (WES) were attracted to the model because it is fully dynamic for determining flow and water quality and it has several desirable numerical features, such as a two-point fourth-order scheme for accurately predicting advective water quality concentrations.

4. The WES contracted Ohio State University to modify the model code to handle control structures. This modification, along with the unsteady flow feature, gave the model the versatility needed for simulating CE-regulated stream/waterway projects. Subsequently, the updated version was tested at WES, and additional modifications and corrections were made, resulting in the model presented herein, CE-QUAL-RIV1.

Objective

5. The purposes of this user's manual are to (a) document the mathematical and numerical formulations of CE-QUAL-RIV1 and (b) provide the guidance necessary for its implementation. The first five parts deal with documentation, and the last two parts with operation. Because the codes are separate, some parts are associated only with the hydraulic code, RIV1H; others deal only with the water quality code, RIV1Q. The output from RIV1H is used to drive RIV1Q. However, a variety of hydraulic codes have also been used to drive RIV1Q.

Model Selection Criteria

6. The first criterion for selection of CE-QUAL-RIV1 for an application is whether the issues can be resolved with a one-dimensional (1-D) (cross-sectionally averaged) model. Most riverine water quality issues can be resolved with a 1-D model; that is, in riverine systems, lateral and vertical gradients in water quality constituent concentrations are generally insignificant and unimportant relative to longitudinal gradients. This assumption implicitly means that vertical temperature, density, and chemical stratifications (which can play a dominant role in the water quality of lakes and reservoirs) are nonexistent or negligible for practical purposes. Thus, although this model can be used for run-of-the-river reservoirs, locks and dams, and reregulation pools, the user must first be sure that vertical stratification does not exist or is so minor as to not affect water quality conditions. In addition, where point source discharges are concerned the model should not be applied "near field" where mixing may not have occurred sufficiently to satisfy the 1-D assumption.

7. The second criterion for selection of CE-QUAL-RIV1 concerns the nature of the flow and the issues; although CE-QUAL-RIV1 was developed for water quality simulations of riverine systems with highly unsteady flow, it can be used for steady flow conditions. However, it may be easier and more economical to use another more simplistic formulation, such as the EPA QUAL2E model (Brown and Barnwell 1987), which assumes steady flow. If the issues demand high resolution and the flows change substantially over a period of hours or days, then a fully dynamic model, such as CE-QUAL-RIV1 should be used.

8. Specific guidance has not been developed to determine at what condition a dynamic flow model should be used. The development of specific guidance would depend on the issues and the required resolution. For example, if daily average (or longer time averages, perhaps even steady-state) predictions are sufficient, then more simplistic modeling approaches may suffice, even for unsteady flow projects. However, if diel fluctuations (and peaks and troughs) are important and various interest groups are sensitive to such fluctuations, then a dynamic model would be necessary for an unsteady flow project. Transient flow conditions can produce substantially greater fluctuations in diel temperature and dissolved oxygen (DO) values than the natural diel effects (Matter et al. 1983).

9. Even when a study is highly sensitive and high resolution is required with diel fluctuations, a time-varying water quality model that assumes steady flow (such as QUAL2E) may suffice if the flows are relatively constant or change slowly during the simulation with respect to the travel time of the system. However, if the flows change substantially during a day and from day to day, then a dynamic flow model (such as CE-QUAL-RIV1) should be considered. The study of riverine water quality resulting from the releases from peaking hydropower dams is a good example of the use of CE-QUAL-RIV1.

10. It should be noted that RIV1H uses the fully dynamic flow equations; thus it has wide-ranging capabilities with good resolution. This advantage must be weighed against possible disadvantages such as unfamiliarity of the user with the code. RIV1Q can be driven by any hydraulic or hydrologic routing model, no matter how simple, as long as the proper linkages are made.

PART II: THE GOVERNING EQUATIONS

General Considerations

11. The transport of momentum and water quality constituents during unsteady flows can be marked by sharp gradients in flow, elevation, and water quality concentrations. These gradients can be propagated by the flow wave through regions of highly variable cross sections intermittently joined by major inflow tributaries, with the magnitudes of concentrations often being augmented by both point and nonpoint source contributions.

12. Since the speed of flow waves is often quite high, water quality concentrations can be dominated by advective transport rather than biogeochemical interactions and diffusion. Therefore, the water quality model must be dynamic and have the following minimum attributes. The model must:

> <u>a</u>. Account for time-varying flow, elevation, and water quality constituent changes resulting from highly unsteady flows.
>
> <u>b</u>. Include the direct explicit interaction of flow and elevation on the constituent distributions.
>
> <u>c</u>. Be applicable for a river channel of arbitrary cross section and specified bottom slope.
>
> <u>d</u>. Allow for a number of water quality constituents and the proper mathematical specification of their mathematical interrelationship.
>
> <u>e</u>. Account for the effects of lateral inputs of water and associated pollutant concentrations.
>
> <u>f</u>. Allow simulation of multiple hydraulic control structures.

13. The assumptions that the above attributes require for the development of the basic governing transport equations are listed below. For a river that is much longer than wide or deep, the following assumptions apply:

> <u>a</u>. Hydrostatic pressure exists.
>
> <u>b</u>. Lateral and vertical gradients are small and neglected; thus the equations are cross-sectionally averaged for flow and constituent variables (1-D assumption).
>
> <u>c</u>. All cross sections and bottom configurations are known.
>
> <u>d</u>. All lateral point and nonpoint source flows and input concentrations are known.

14. When solved, the hydraulic transport equations permit the calculation of downstream histories of flow and water surface elevation. These equations are well known, and their derivations are quite routine. There are two different approaches to the derivation of flow wave equations. The first

starts with the basic three-dimensional (3-D) equations of continuity and
Navier Stokes (Bird, Stewart, and Lightfoot 1964) and by cross-sectional
averaging reduces the four equations to a coupled pair of dynamic cross-
sectionally averaged equations for longitudinal space and time patterns of
flow and elevation. This procedure, however, always results in the necessity
of specifying a very ambiguous eddy viscosity to account for the ever-present
correlations between fluctuating components resulting from the averaging.
Longitudinal eddy viscosities are very small, particularly during elevated
flows (Fischer et al. 1979), and are quite frequently used improperly to tune
the correct answer into existence rather than to represent actual problem
physics. A second approach, which assumes inviscid flow, is the control
volume method (Liggett 1975) used to derive the equations herein.

15. The notation and coordinate system are defined as in Figure 1. It
is assumed that the coordinate system is placed in the river bottom with the
bed slope relative to a gravity-based coordinate system being such that
$\tan \theta \approx \theta$, and, therefore, θ is very small. It is further assumed that x
is directed positive downstream and that because the bed slope is very small,
the water elevation h(x,t) , directed parallel to the gravitational direc-
tion, is perpendicular to x. A(x,t) is the cross-sectional area of flow,
and B(x,t) is the top width of flow. If the relationship among area of
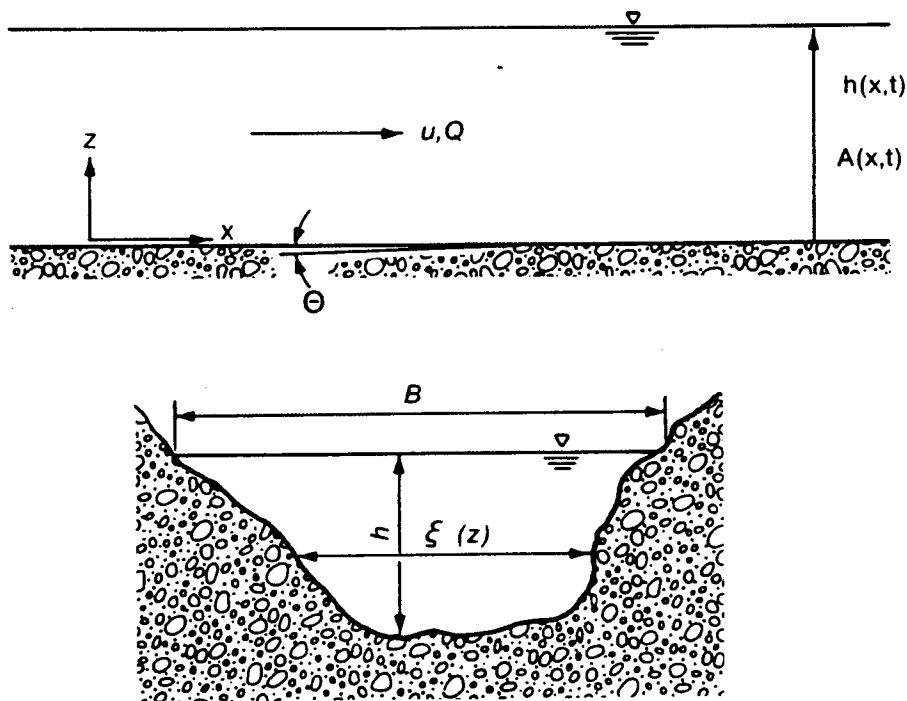
Figure 1. Coordinate system and notation

9

flow, top width, and flow depth is represented by the shape factor $\xi(z)$, A and B can be related to $h(x,t)$.

16. Assume as in Figure 2 that a discrete length Δx of river channel is isolated. If the flow is from left to right, then unit normals n_o and n_i are defined as being positive away from each face through which flow is entering and exiting. Total mass, momentum, and species mass will be conserved within this control volume. Streeter and Wylie (1979) present the general form of the control volume conservation law in a continuum with mass concentration b as

$$\frac{d\beta}{dt} = \iiint_{cv} \frac{\partial b}{\partial t} dV + \iint_{cs} b\left(\bar{v}.\hat{n}\right) dA \tag{1}$$

Equation 1 states that the total time rate of change of mass in the control volume, β , ($\beta = b\bar{v}$) equals the time rate of accumulation of b within the control volume (cv) plus the net rate at which b is being carried into the control volume through the control surface (cs). The cv conservations of water mass, momentum, and species mass are, respectively, written for an arbitrary cv in a 3-D flow field $\bar{v}$ as

$$\text{Mass:} \quad \iiint_{cv} \frac{\partial \rho}{\partial t} dV + \iint_{cs} \left(\rho\bar{v} \cdot \hat{n}\right) dA = 0 \tag{2}$$

$$\text{Momentum:} \quad \iiint_{cv} \frac{\partial \bar{v}\rho}{\partial t} dV + \iint_{cs} \bar{v}\left(\rho\bar{v} \cdot \hat{n}\right) dA = \bar{F} \tag{3}$$

$$\text{Mass species:} \quad \iiint_{cv} \frac{\partial \alpha}{\partial t} dV + \iint_{cs} \alpha\left(\bar{v} \cdot \hat{n}\right) dA = S* \tag{4}$$

$+ N_i$

$+ N_o$

$u(x,t), \ Q(x,t)$

Δx

$+X \longrightarrow$

$A(x,t)$

q

$h - \dfrac{\partial h}{\partial x} \dfrac{\Delta x}{2}, \ A - \dfrac{\partial A}{\partial x} \dfrac{\Delta x}{2}$

A,h

$h + \dfrac{\partial h}{\partial x} \dfrac{\Delta x}{2}, \ A + \dfrac{\partial A}{\partial x} \dfrac{\Delta x}{2}$

$u - \dfrac{\partial u}{\partial x} \dfrac{\Delta x}{2}, \ Q - \dfrac{\partial Q}{\partial x} \dfrac{\Delta x}{2}$

$u + \dfrac{\partial u}{\partial x} \dfrac{\Delta x}{2}, \ Q + \dfrac{\partial Q}{\partial x} \dfrac{\Delta x}{2}$

Δx
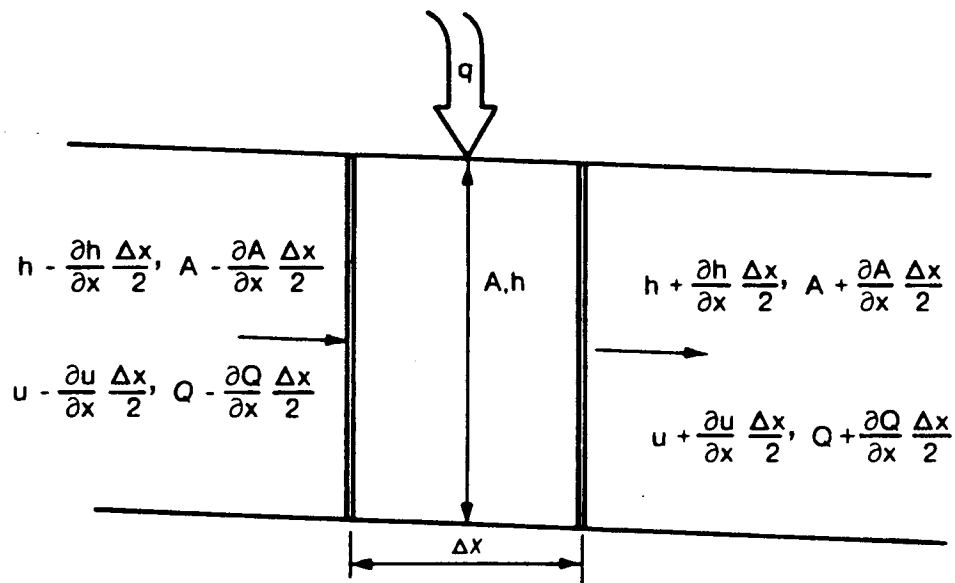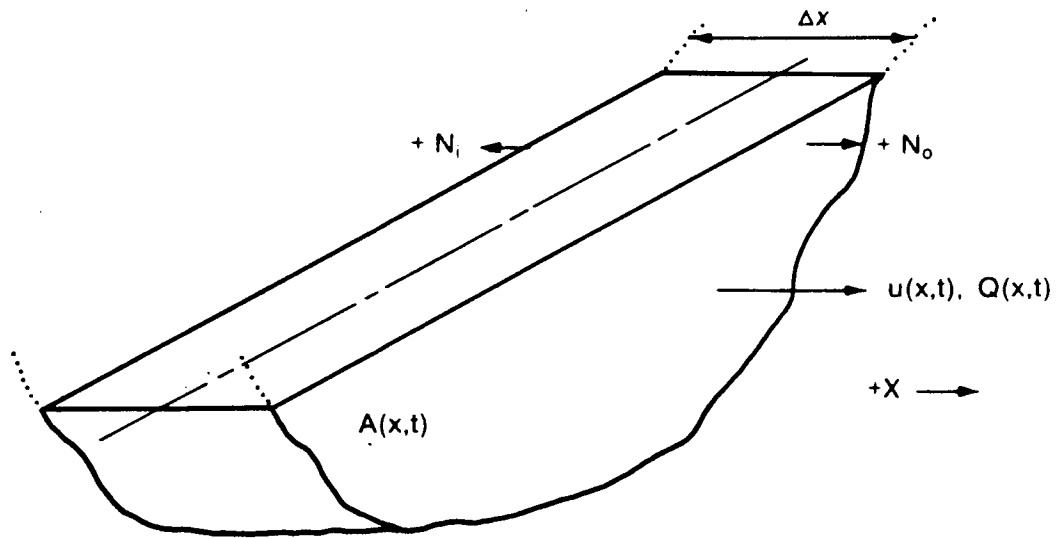
Figure 2. Control volume definitions

where

ρ - fluid density, M/L^3

t - time

dV - differential volume, L^3

$\bar{v}$ - velocity vector $(u\hat{i} + v\hat{j} + w\hat{k})$, L/t

dA - differential area, L^2

$\bar{F}$ - vector sum of the real applied external forces on the cv, ML/t^2

α - species mass concentration, M/L^3

$S*$ - net source term for biochemical changes in α, M/t

The variables M, L, and t are general units of mass, length, and time, respectively.

Derivation of Equations

17. As in Liggett (1975), a much simpler derivation is obtained for the river problem by assuming that the average velocity U (or average species concentration α) is known at a cross section. A Taylor series expansion across inlet and outlet permits much simpler expressions to be identified.

Conservation of mass

18. The conservation of water mass equation shows that the time rate of change of storage equals net mass efflux through the cv surface. By a Taylor series expansion, therefore

$$\frac{\partial(\rho A \Delta x)}{\partial t} = \rho\left[\left(U - \frac{\partial U}{\partial x}\frac{\Delta x}{2}\right)\left(A - \frac{\partial A}{\partial x}\frac{\Delta x}{2}\right) - \left(U + \frac{\partial U}{\partial x}\frac{\Delta x}{2}\right)\left(A + \frac{\partial A}{\partial x}\frac{\Delta x}{2}\right)\right] \tag{5}$$

For an incompressible fluid such as water, ρ - constant; multiplying out and dividing by $\rho\Delta x$ gives for small Δx

$$\frac{\partial A}{\partial t} + \frac{\partial(UA)}{\partial x} = 0 \text{ or } \frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{6}$$

where Q is the volumetric flow rate.

Momentum conservation

19. The momentum equation is a bit more difficult. It states that the net force acting on the cv equals the time rate of change of momentum in the cv plus the net rate of efflux of momentum through the cv. Therefore

$$\frac{\partial (Q\Delta x)\rho}{\partial t} + \rho \left\{ [U(UA)] + \frac{\partial}{\partial x} [U(UA)] \frac{\Delta x}{2} \right\}$$

(7)

$$- \rho \left\{ [U(UA)] - \frac{\partial}{\partial x} \left[U(UA) \frac{\Delta x}{2} \right] \right\} = \overline{F}$$

The force vector $\overline{F}$ requires further expansion into three subcategories: gravity, shear, and pressure forces.

20. **Gravity force, f_g.** The total gravity force is nothing more than the component of the weight of water in the cv ($\rho gA\Delta x$) directed in the x-direction or $\rho gA\Delta x \sin \theta$. Therefore

$$f_g = \rho gA\Delta x S_o$$

(8)

where $S_o = \sin \theta \approx \theta$ is the slope.

21. **Shear force, f_τ.** The channel sides and bottoms deplete momentum by the action of bottom friction or shear. This depletion is made mathematically analogous to the gravity slope term by setting

$$f_\tau = \rho gA\Delta x S_f$$

(9)

where S_f is the friction slope. Several forms for the friction slope exist, but either Chezy or Manning forms predominate, i.e.

$$S_f = \frac{U^2 n^2}{\left(1.486 \ R^{2/3} \right)^2}$$

(10)

13

where

n = Manning's friction factor

R = hydraulic radius which is approximately equal to A/B , where B is the top width

Therefore,

$$f_\tau = \rho g A \Delta x \frac{U^2 n^2}{\left(1.486 \ R^{2/3}\right)^2}$$

or

$$f_\tau = \rho g A \Delta x \frac{n^2 |Q| Q}{2.2 \ A^2 \ R^{4/3}} \tag{11}$$

where the absolute value has been retained to ensure that no matter which way the wave propagates, shear always dissipates momentum.

22. <u>Pressure force, f_p.</u> The total pressure force on the face of the cv is the integral of the irregular trace of the cv, i.e.

$$f_p = \int_0^h \rho g (h - z) \xi(z) dz \tag{12}$$

where $\xi(z)$ is the channel width at height z above the bottom. A Taylor series expansion gives the net pressure in the downstream direction

$$f_p = - \frac{\partial}{\partial x} \int_0^h \rho g (h - z) \xi(z) dz \Delta x$$

$$= - \rho g \int_0^h \frac{\partial}{\partial x} [(h - z) \xi(z)] dz \Delta x \tag{13}$$

and by chain rule differentiation

$$f_p = - \rho g \left[\frac{\partial h}{\partial x} \int_0^h \xi(z)dz + \int_0^h (h - z) \frac{\partial \xi(z)}{\partial x} dz \right] \Delta x \tag{14}$$

The first term represents the pressure force at that particular cross section. The second term represents the net pressure force caused by rapid area changes over the length Δx . If the channels are considered prismatic and regular, then the last term has little significance. Therefore

$$f_p = - \rho g A \frac{\partial h}{\partial x} \Delta x \tag{15}$$

The final equation for momentum is then

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} (QU) + gA \frac{\partial h}{\partial x} = gA(S_o - S_f) \tag{16}$$

Modifications to momentum and continuity

23. _Lateral and tributary inflow._ Runoff from lands adjacent to the channel or tributary inflow can cause increased levels of total mass and momentum in the river. If q is the flow per unit channel length entering the river with velocity U_q , then Equations 6 and 16 become, respectively,

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q \tag{17}$$

and

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} (UQ) + gA \frac{\partial h}{\partial x} = gA(S_o - S_f) + qU_q \tag{18}$$

However, the model code does not include the last term in Equation 18 because it is relatively insignificant compared with other terms of the momentum equation.

24. **Floodplain storage.** As sometimes occurs, excess quantities of non-moving water are often stored in the floodplain. Since the water is not moving, the momentum equation remains unaffected by this feature. The continuity equation must, however, account for the excess mass. If, as in Figure 3, A_o is defined as the cross-sectional area of the floodplain waters, then Equation 17 becomes

$$\frac{\partial (A + A_o)}{\partial t} + \frac{\partial Q}{\partial x} = q \tag{19}$$

$$A_o = A_{01} + A_{02}$$
$$B_o = B_{01} + B_{02}$$
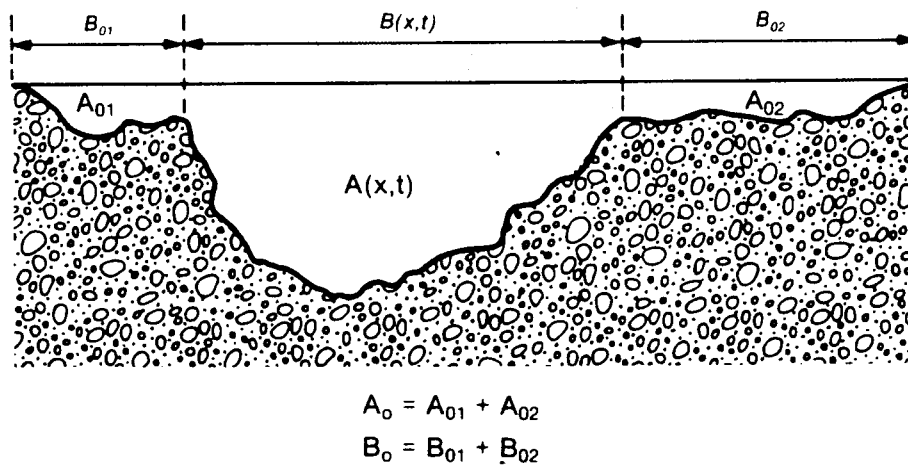
Figure 3. Floodplain geometry and notation

However, the effect of floodplain storage is not presently included in the code.

25. **Channel constrictions.** Often very intense channel constrictions, due to bridges for example, occur over channel lengths that are far too small to economically resolve in the model. The subgrid scale effect of such constrictions is a momentum loss and backwater effect. This effect is accounted for in the right side of the momentum equation by subtracting the force term, $\rho g \Delta h_E$ where h_E is a head loss and K_E is a coefficient to be selected and optimized.

$$h_E = \frac{K_E}{2g} \left(\frac{Q}{A} \right)^2 \qquad (20)$$

The default value is zero for no constriction loss. A value for K_E as high
.as 0.5 may be appropriate for an abrupt constriction.

26. _Momentum correction factor._ When the velocity across the channel
is substantially nonuniform through the model reach, it may be necessary to
use a momentum correction factor ß in the momentum equation. The momentum
correction factor ß multiplies the second term on the left side of Equa-
tion 18; this correction permits the use of the average velocity, U , in the
solution whereas the velocity distribution at each cross section may be quite
different from U . For example, for laminar flow in a straight round tube,
ß is 4/3. It equals 1.0 for uniform flow and cannot be less than 1.0. For
RIV1H, a constant value is used throughout the modeled reach. The default
value of 1.0 is recommended for rivers and streams.

27. _Tributary networks._ The momentum and continuity equations above
must be applied to each and every tributary entering the main stem. At each
junction, the water surfaces in each branch must be equal.

Mass species equation

28. Again, by using the Taylor series approach, the advection and
diffusion of a biogeochemically reactive substance of concentration α is

$$\frac{\partial(\alpha A)}{\partial t} + \frac{\partial(Q\alpha)}{\partial x} = \frac{\partial}{\partial x} \left(DA \frac{\partial \alpha}{\partial x} \right) + \gamma q + S* \qquad (21)$$

where

    D = turbulent dispersion coefficient

    γ = concentration of species type α entering the tributary from
       lateral flow q

    $S*$ = source/sink term in units of (M/Lt)

Equation 21 must be written for every transported species.

Initial conditions

29. For continuity, the only initial conditions to be specified at time $t = t_o$ for all the nodes i for $i = 1$ to N are

$$A(t = t_o, x) = A_i(x) \tag{22}$$

or since there is a direct correspondence between A and h, it is permissible to use

$$h(t = t_o, x) = h_i(x) \tag{23}$$

Either Equation 22 or 23 must be used, but not both.

30. For the momentum equation either

$$Q(t = t_o, x) = Q_i(x) \tag{24}$$

or

$$U(t = t_o, x) = U_i(x) \tag{25}$$

is permissible.

31. For the species transport equation, the following is used:

$$\alpha(t = t_o, x) = \alpha_i(x) \tag{26}$$

Boundary conditions for
continuity and momentum

32. For the upstream boundary at $x = 0$ and the downstream boundary at $x = L$, the following sets of boundary conditions may be used for the combined set of continuity and momentum equations. Please note that two first-order

equations require only a total of two boundary conditions. Therefore, only one set may be used for each simulation.

$$h(t,x = 0) = h_u(t) \quad \text{and} \quad h(t,x = L) = h_d(t) \tag{27}$$

$$Q(t,x = 0) = Q_u(t) \quad \text{and} \quad Q(t,x = L) = Q_d(t) \tag{28}$$

$$h(t,x = 0) = h_u(t) \quad \text{and} \quad Q(t,x = L) = Q_d(t) \tag{29}$$

$$Q(t,x = 0) = Q_u(t) \quad \text{and} \quad h(t,x = L) = h_d(t) \tag{30}$$

$$h(t,x = 0) = h_u(t) \quad \text{and} \quad f_1(Q,h) = f_{1d}(t) \tag{31}$$

$$Q(t,x = 0) = Q_u(t) \quad \text{and} \quad f_2(Q,h) = f_{2d}(t) \tag{32}$$

where f_1 and f_2 are rating curves relating h and Q for the downstream d boundary conditions. It is permissible to properly substitute A and U for Q in these expressions.

33. It should be noted that the upstream boundary condition for a tributary can be selected by specifying either the elevation or the flow. At the downstream tributary boundary, i.e. the confluence with the main stem, only the elevation is allowed as a "boundary" condition to ensure that continuity is preserved at the junction.

Boundary conditions
for species transport

34. The boundary conditions for species transport are

$$\alpha(t,x = 0) = \alpha_u(t)$$
$$\tag{33}$$
$$\alpha(t,x = L) = \alpha_d(t)$$

However, the condition $\alpha_d(t)$ is presently not specified as input data for riverine simulations. This must be included as input if the model is modified for inflow at the downstream end, such as occurs with estuarine boundary conditions.

Equation Summary

35. The following equations govern the unsteady, 1-D (longitudinal) hydrodynamics and transport:

Continuity

$$\frac{\partial (A + A_o)}{\partial t} + \frac{\partial Q}{\partial x} = q \tag{34}$$

Momentum

$$\frac{\partial Q}{\partial t} + \frac{\partial (UQ)}{\partial x} + gA\,\frac{\partial h}{\partial x} = gA\left(S_o - S_f - \frac{h_E}{\Delta x}\right) + qU_q \tag{35}$$

Equations 34 and 35 are commonly referred to as the St. Venant equations.

Constituent transport

$$\frac{\partial (\alpha A)}{\partial t} + \frac{\partial (Q\alpha)}{\partial x} = \frac{\partial}{\partial x}\left(DA\,\frac{\partial \alpha}{\partial x}\right) + \gamma q + S* \tag{36}$$

Equation 36 is often referred to as the "advection-dispersion equation with source/sink terms." The initial conditions are Equations 22 (or 23), 24 (or 25), and 26 for the continuity, momentum, and mass species equations, respectively. Boundary conditions are one set selected from Equations 27 through 32 for the momentum and continuity equation and Equation 33 for the mass species equation.

PART III: THE NUMERICAL SOLUTION FOR FLOW AND ELEVATION

Rationale

36. As noted, flood and peaking hydropower waves and associated transport are marked by rapidly varying flows, elevations, and concentrations. As derived, the continuity and momentum equations, from which elevations and flows are calculated, are hyperbolic. The transport equation is dominated by advection; thus it has hyperbolic features. All of these equations, then, are very difficult to solve numerically because the smoothing, stabilizing effects of dispersion are eliminated or reduced. Inspection of the governing equations reveals that they are coupled, unsteady, and nonlinear, but because the contaminant concentrations do not affect the flow field, it is possible to uncouple the solution of the continuity and momentum equations from the solution of the species transport equation. Once solved, the complete time histories of flow and elevation can be stored and used as input information for the transport calculations. Because the governing equations are hyperbolic, the solution procedure for the flows and elevations is quite different from the transport solution; therefore, this part presents the solution procedure for the continuity and momentum equations. Solution of the transport equations is covered in Part IV.

Numerical Approximations

37. Three numerical procedures are useful for solving hyperbolic equations: the finite element method, the method of characteristics, and implicit, finite difference methods. The method of characteristics is quite accurate but can be difficult to program by anyone but a specialist. Reviews of this method are found in Liggett and Cunge (1975) and Abbott (1979). Implicit, finite difference methods are simpler to program because they are much more direct numerical approximation techniques to partial derivatives. These methods also possess favorable stability behavior even in applications with variable space and time-steps. Explicit finite difference methods are generally unstable and are not considered.

38. There are many implicit procedures, but the method to be used here is the four-point implicit method first used by Preissmann (1961) with subsequent applications by, among others, Amein and Fang (1970) and Amein and Chu

(1975). This formulation is currently being used by Fread (1973, 1978) in the National Weather Service Dambreak Model (Fread 1978). The method is weighted implicit at each time level, is unconditionally stable for $0.5 < \theta \leq 1.0$, and permits relatively unequal space and time-steps. The scheme has second-order accuracy when $\theta = 0.5$ and first-order accuracy when $\theta = 1.0$. It is fully nonlinear but yet is a compact scheme requiring just two points at each time level for second-order spatial accuracy.

39. The river system is discretized (Figure 4) by a network of time and space nodes separated by time and space increments Δx_i, Δt_j. If β denotes the point about which the governing equation is to be discretized, then the values of the variables at the four points surrounding β are used to form the appropriate derivatives and weighted averages. For a general variable ω, then

$$\omega(\beta) \approx \theta \left(\frac{\omega_i^{j+1} + \omega_{i+1}^{j+1}}{2} \right) + (1 - \theta) \left(\frac{\omega_i^j + \omega_{i+1}^j}{2} \right) \tag{37}$$

$$\frac{\partial \omega(\beta)}{\partial x} \approx \theta \left(\frac{\omega_{i+1}^{j+1} - \omega_i^{j+1}}{\Delta x_i} \right) + (1 - \theta) \left(\frac{\omega_{i+1}^j - \omega_i^j}{\Delta x_i} \right) \tag{38}$$

$$\frac{\partial \omega(\beta)}{\partial t} \approx \left(\frac{\omega_i^{j+1} + \omega_{i+1}^{j+1}}{2\Delta t_j} \right) - \left(\frac{\omega_i^j + \omega_{i+1}^j}{2\Delta t_j} \right) \tag{39}$$

Application to Governing Equations

The continuity equation

40. From Equation 34, the continuity equation is

$$\frac{\partial (A + A_o)}{\partial t} + \frac{\partial Q}{\partial x} - q = 0 \tag{40}$$

Using the definitions in Equations 37, 38, and 39, the discretized form of this equation is
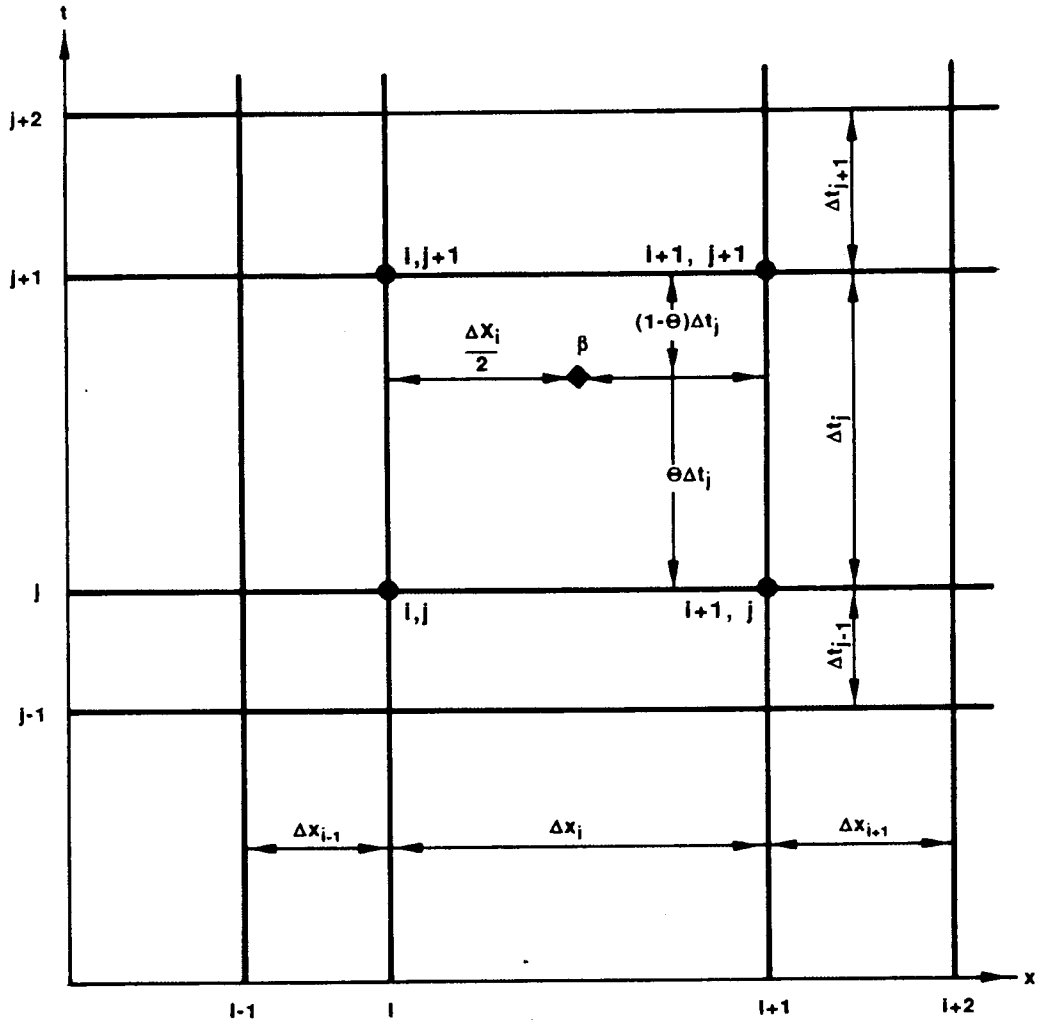
Figure 4. Numerical grid

$$\frac{1}{2\Delta t_j}\left[\left(A + A_o\right)_i^{j+1} + \left(A + A_o\right)_{i+1}^{j+1} - \left(A + A_o\right)_i^{j} - \left(A + A_o\right)_{i+1}^{j}\right]$$

$$+ \theta\left[\frac{1}{\Delta x_i}\left(Q_{i+1}^{j+1} - Q_i^{j+1}\right)\right] - \theta\,\frac{\left(q_{i+1}^{j+1} + q_i^{j+1}\right)}{2}$$

(41)

$$+ (1 - \theta)\left[\frac{1}{\Delta x_i}\left(Q_{i+1}^{j} - Q_i^{j}\right)\right] - (1 - \theta)\,\frac{\left(q_{i+1}^{j} + q_i^{j}\right)}{2}$$

$$= F_i\,(Q_{i+1},\ A_{i+1},\ Q_i, A_i) = 0$$

23

The momentum equation

41. From Equation 35 and using the definitions for S_E and S_f from Equations 20 and 11, the numerical discretization of the momentum equation is

$$\frac{1}{2\Delta t_j}\left(Q_i^{j+1} + Q_{i+1}^{j+1} - Q_i^j - Q_{i+1}^j\right)$$

$$+ \theta\left\{\frac{1}{\Delta x_i}\left[\left(\frac{Q^2}{A}\right)_{i+1}^{j+1} - \left(\frac{Q^2}{A}\right)_i^{j+1}\right]\right\} + \theta g\left(\frac{A_{i+1}^{j+1} + A_i^{j+1}}{2}\right)\left[\frac{1}{\Delta x_i}\left(h_{i+1}^{j+1} - h_i^{j+1}\right)\right]$$

$$+ \frac{g\theta}{2(2.2)}\left[\left(\frac{n_{i+1}^2|Q_{i+1}|Q_{i+1}}{A_{i+1}R_{i+1}^{4/3}}\right)^{j+1} + \left(\frac{n_i^2|Q_i|Q_i}{A_iR_i^{4/3}}\right)^{j+1}\right]$$

$$- \theta g\left(\frac{A_{i+1}^{j+1} + A_i^{j+1}}{2}\right)\left(\frac{So_{i+1} + So_i}{2}\right)$$

$$+ \theta\left(\frac{A_{i+1}^{j+1} + A_i^{j+1}}{8}\right)\left\{\frac{K_E}{\Delta x_i}\left[\left(\frac{Q^2}{A^2}\right)_{i+1}^{j+1} + \left(\frac{Q^2}{A^2}\right)_i^{j+1}\right]\right\}$$

$$- \theta\left(\frac{q_{i+1}^{j+1} + q_i^{j+1}}{2}\right)\left(\frac{U_{q_{i+1}}^{j+1} + U_{q_i}^{j+1}}{2}\right)$$

$$\qquad(42)$$

$$+ (1 - \theta)\left\{\frac{1}{\Delta x_i}\left[\left(\frac{Q^2}{A}\right)_{i+1}^j - \left(\frac{Q^2}{A}\right)_i^j\right]\right\} + g(1 - \theta)\left(\frac{A_{i+1}^j + A_i^j}{2}\right)\left[\frac{1}{\Delta x_i}\left(h_{i+1}^j - h_i^j\right)\right]$$

$$+ \frac{g(1 - \theta)}{2(2.2)}\left[\left(\frac{n_{i+1}^2|Q_{i+1}|Q_{i+1}}{A_{i+1}R_{i+1}^{4/3}}\right)^j + \left(\frac{n_i^2|Q_i|Q_i}{A_iR_i^{4/3}}\right)^j\right]$$

$$- g(1 - \theta)\left(\frac{A_{i+1}^j + A_i^j}{2}\right)\left(\frac{So_{i+1} + So_i}{2}\right)$$

$$+ (1 - \theta)\left(\frac{A_{i+1}^j + A_i^j}{8}\right)\left\{\frac{K_E}{\Delta x_i}\left[\left(\frac{Q^2}{A^2}\right)_{i+1}^j + \left(\frac{Q^2}{A^2}\right)_i^j\right]\right\}$$

$$- (1 - \theta)\left(\frac{q_{i+1}^j + q_i^j}{2}\right)\left(\frac{U_{q_{i+1}}^j + U_{q_i}^j}{2}\right)$$

$$= G_i\left(Q_{i+1}, A_{i+1}, Q_i, A_i\right) = 0$$

<u>The boundary conditions</u>

42. The boundary conditions must also be "discretized." This is accomplished as follows. From the list of permissible pairs of boundary conditions, one set is selected, say Equation 27. Then the discretized form of the upstream boundary in terms of A becomes

$$F_o = A_1^{j+1} - A_u\left(t^{j+1}\right) = 0 \qquad (43)$$

while the downstream boundary condition at node N becomes

$$F_N = A_N^{j+1} - A_d\left(t^{j+1}\right) = 0 \qquad (44)$$

If flows or discharges are to be specified, then from Equation 28 the upstream boundary condition becomes

$$F_o = Q_1^{j+1} - Q_u\left(t^{j+1}\right) = 0 \qquad (45)$$

and the downstream condition is

$$F_N = Q_N^{j+1} - Q_d\left(t^{j+1}\right) = 0 \qquad (46)$$

<u>Equation Assembly</u>

43. Equations F_i (Equation 41) and G_i (Equation 42) are applied at every node i on both the main stem and any included tributaries. The system of algebraic equations results in 2N-2 equations for 2N unknowns. The two boundary conditions are sufficient to completely close the problem. If $G_o(Q_1,A_1)$ and $G_N(Q_N,A_N)$ are the boundary conditions written in the form of Equations 43-46, then the resulting system of 2N nonlinear equations is formally written as

25

$$G_0(Q_1, A_1) = 0$$

$$F_1(Q_2, A_2, Q_1, A_1) = 0$$

$$G_1(Q_2, A_2, Q_1, A_1) = 0$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$F_i(Q_{i+1}, A_{i+1}, Q_i, A_i) = 0$$

$$\tag{47}$$

$$G_i(Q_{i+1}, A_{i+1}, Q_i, A_i) = 0$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$F_{N-1}(Q_N, A_N, Q_{N-1}, A_{N-1}) = 0$$

$$G_{N-1}(Q_N, A_N, Q_{N-1}, A_{N-1}) = 0$$

$$G_N(Q_N, A_N) = 0$$

44. The general solution of these nonlinear equations can proceed in two ways. First the nonlinear terms may be linearized by using information from the jth time-step. Time marching proceeds very easily by this procedure. However, under the potentially rapid and sharp gradients being reproduced here, a full nonlinear solution by Newton-Raphson iteration procedures is recommended and implemented.

Newton-Raphson Solution for Flow and Elevation

45. The Newton-Raphson method for nonlinear systems is based upon estimating the residuals from each algebraic equation introduced by assuming initial answers for Q_i and A_i . The iterative reduction of the residuals by successive Taylor series refinement of the estimates of Q_i and A_i completes the procedure. Details of Newton's method can be found in Burden and Faires (1989). The generalized Newton-Raphson procedure must be performed at

each time-step. This method has been used with excellent success by the previously cited authors.

46. However, unlike the procedures by these authors, the formulation herein employs a direct simultaneous solution at each iteration for all the flows and elevations in both main stem and tributaries. Previous methods employed an iterative method of estimating such variables. Therefore, for each Newton-Raphson iteration, another sequence of interior iterations was performed. The formulation herein requires only one five-band matrix solution at each Newton-Raphson iteration. The Newton-Raphson concept is presented herein, and details of the matrix packing and solution procedure are presented in the program operation portion of the manual (Part VI).

47. The residuals, $R_{1,i}^k$ and $R_{2,i}^k$, from the kth estimate of Q_i and A_i are found for Equations F_i and G_i as

$$G_0\left(Q_1^k, A_1^k\right) = R_{2,0}^k$$

$$F_1\left(Q_2^k, A_2^k, Q_1^k, A_1^k\right) = R_{1,1}^k$$

$$G_1\left(Q_2^k, A_2^k, Q_1^k, A_1^k\right) = R_{2,1}^k$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$F_i\left(Q_{i+1}^k, A_{i+1}^k, Q_i^k, A_i^k\right) = R_{1,i}^k$$

$$G_i\left(Q_{i+1}^k, A_{i+1}^k, Q_i^k, A_i^k\right) = R_{2,i}^k \qquad (48)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$F_{N-1}\left(Q_N^k, A_N^k, Q_{N-1}^k, A_{N-1}^k\right) = R_{1,N-1}^k$$

$$G_{N-1}\left(Q_N^k, A_N^k, Q_{N-1}^k, A_{N-1}^k\right) = R_{2,N-1}^k$$

$$G_N\left(Q_N^k, A_N^k\right) = R_{2,N}^k$$

48. A generalized Taylor series for a function T that is a function of four dependent variables, S_1, S_2, S_3, and S_4 is written as

$$T^{k+1} = T^k + \frac{\partial T}{\partial S_1} dS_1 + \frac{\partial T}{\partial S_2} dS_2 + \frac{\partial T}{\partial S_3} dS_3 + \frac{\partial T}{\partial S_4} dS_4 \tag{49}$$

This general form is now used to relate the residuals to the gradients as follows. Keep in mind that we want to drive the residuals to zero; thus we want F_i and G_i to be zero. If T^{k+1} represents the exact solution for F_i or G_i, then T^{k+1} is zero. This means that the unknowns are dS_1, dS_2, dS_3, and dS_4. Let T^k be the R_{ji}^k residual. Further, assume that S_1, S_2, S_3, and S_4 become the values Q_{i+1}, A_{i+1}, Q_i, and A_i. In the Newton-Raphson iteration, the gradients are known from the kth estimate of Q_i and A_i, and the relationship between the gradients and the residuals becomes

$$\frac{\partial G_0}{\partial A_1} dA_1 + \frac{\partial G_0}{\partial Q_1} dQ_1 = R_{2,0}^k$$

$$\frac{\partial F_1}{\partial Q_2} dQ_2 + \frac{\partial F_1}{\partial A_2} dA_2 + \frac{\partial F_1}{\partial Q_1} dQ_1 + \frac{\partial F_1}{\partial A_1} dA_1 = R_{1,1}^k$$

$$\frac{\partial G_1}{\partial Q_2} dQ_2 + \frac{\partial G_1}{\partial A_2} dA_2 + \frac{\partial G_1}{\partial Q_1} dQ_1 + \frac{\partial G_1}{\partial A_1} dA_1 = R_{2,1}^k$$

. . . .

. . . .

$$\frac{\partial F_i}{\partial Q_{i+1}} dQ_{i+1} + \frac{\partial F_i}{\partial A_{i+1}} dA_{i+1} + \frac{\partial F_i}{\partial Q_i} dQ_i + \frac{\partial F_i}{\partial A_i} dA_i = R_{1,i}^k$$

$$\frac{\partial G_i}{\partial Q_{i+1}} dQ_{i+1} + \frac{\partial G_i}{\partial A_{i+1}} dA_{i+1} + \frac{\partial G_i}{\partial Q_i} dQ_i + \frac{\partial G_i}{\partial A_i} dA_i = R_{2,i}^k \tag{50}$$

. . . .

$$\frac{\partial F_{N-1}}{\partial Q_N} dQ_N + \frac{\partial F_{N-1}}{\partial A_N} dA_N + \frac{\partial F_{N-1}}{\partial Q_{N-1}} dQ_{N-1} + \frac{\partial F_{N-1}}{\partial A_{N-1}} dA_{N-1} = R_{1,N-1}^k$$

$$\frac{\partial G_{N-1}}{\partial Q_N} dQ_N + \frac{\partial G_{N-1}}{\partial A_N} dA_N + \frac{\partial G_{N-1}}{\partial Q_{N-1}} dQ_{N-1} + \frac{\partial G_{N-1}}{\partial A_{N-1}} dA_{N-1} = R_{2,N-1}^k$$

$$\frac{\partial G_N}{\partial Q_N} dQ_N + \frac{\partial G_N}{\partial A_N} dA_N = R_{2,N}^k$$

where the minus sign (from moving T^k to the other side of the equation) has been included in the residuals; thus the residuals are the negative of Equations 41 and 42. Note that all derivatives are taken with respect to Q_{i+1}^{j+1}, A_{i+1}^{j+1}, Q_i^{j+1}, A_i^{j+1}, etc. Additionally, for all Q_i and A_i evaluated at time $j+1$

$$dQ_1 = Q_1^{k+1} - Q_1^k$$

$$dA_1 = A_1^{k+1} - A_1^k$$

. . . .

. . . .

$$dQ_i = Q_i^{k+1} - Q_i^k$$

$$(51)$$

$$dA_i = A_i^{k+1} - A_i^k$$

. . . .

. . . .

$$dQ_N = Q_N^{k+1} - Q_N^k$$

$$dA_N = A_N^{k+1} - A_N^k$$

49. The derivatives of F_i and G_i with respect to A_{i+1}^{j+1}, Q_{i+1}^{j+1}, A_i^{j+1}, and Q_i^{j+1} are found from Equations 41 and 42 as follows:

$$\frac{\partial F_i}{\partial A_i^{j+1}} = \frac{1}{2\Delta t_j} \qquad (52)$$

$$\frac{\partial F_i}{\partial Q_i^{j+1}} = \frac{\theta}{\Delta x_i} \qquad (53)$$

29

$$\frac{\partial F_i}{\partial A_{i+1}^{j+1}} = \frac{1}{2\Delta t_j} \tag{54}$$

$$\frac{\partial F_i}{\partial Q_{i+1}^{j+1}} = \frac{\theta}{\Delta x_i} \tag{55}$$

$$\frac{\partial G_i}{\partial Q_i^{j+1}} = \frac{1}{2\Delta t_j} + \theta \left[\frac{-2}{\Delta x_i} \frac{Q_i^{j+1}}{A_i^{j+1}} + \frac{g}{2(2.2)} \frac{n_i^2 |Q_i^{j+1}|}{A_i^{j+1}\left(R_i^{j+1}\right)^{4/3}} \right.$$

$$\left. + 2 \left(\frac{A_{i+1}^{j+1} + A_i^{j+1}}{8\Delta x_i} \right) K_E \frac{Q_i^{j+1}}{\left(A_i^{j+1}\right)^2} \right] \tag{56}$$

$$\frac{\partial G_i}{\partial Q_{i+1}^{j+1}} = \frac{1}{2\Delta t_j} + \theta \left[\frac{2}{\Delta x_i} \frac{Q_{i+1}^{j+1}}{A_{i+1}^{j+1}} + \frac{g}{2(2.2)} \frac{n_{i+1}^2 |Q_{i+1}^{j+1}|}{A_{i+1}^{j+1}\left(R_{i+1}^{j+1}\right)^{4/3}} \right.$$

$$\left. + 2 \left(\frac{A_{i+1}^{j+1} + A_i^{j+1}}{8\Delta x_i} \right) K_E \frac{Q_{i+1}^{j+1}}{\left(A_{i+1}^{j+1}\right)^2} \right] \tag{57}$$

$$\frac{\partial G_i}{\partial A_{i+1}^{j+1}} = \theta \left\{ \frac{-1}{\Delta x_i} \left(\frac{Q^2}{A^2} \right)_{i+1}^{j+1} + \frac{g}{2\Delta x_i} \left[\left(h_{i+1}^{j+1} - h_i^{j+1} \right) + \frac{\left(A_{i+1}^{j+1} + A_1^{j+1} \right)}{B_{i+1}^{j+1}} \right] \right.$$

$$+ \frac{gn_{i+1}^2}{6(2.2)} \frac{|Q_{i+1}^{j+1}|Q_{i+1}^{j+1}}{A_{i+1}^{j+1}\left(R_{i+1}^{j+1}\right)^{4/3}} \left[\frac{-7}{A_{i+1}^{j+1}} + \frac{4\left.\frac{dB}{dh}\right|_{i+1}^{j+1}}{\left(B_{i+1}^{j+1}\right)^2} + \frac{6\left.\frac{\partial n}{\partial h}\right|_{i+1}^{j+1}}{n_{i+1} B_{i+1}^{j+1}} \right] \tag{58}$$

$$\left. - \frac{g}{2} \left(\frac{S_{o_{i+1}} + S_{o_i}}{2} \right) + \frac{k_E}{8\Delta x_i} \left[\left(\frac{Q^2}{A^2} \right)_i^{j+1} - \left(\frac{Q^2}{A^2} \right)_{i+1}^{j+1} \left(1 + \frac{2A_i^{j+1}}{A_{i+1}^{j+1}} \right) \right] \right\}$$

$$\frac{\partial G_i}{\partial A^{j+1}} = \theta \left\{ \frac{1}{\Delta x_i} \left(\frac{Q^2}{A^2}\right)^{j+1}_1 + \frac{g}{2\Delta x_i} \left[\left(h^{j+1}_{i+1} - h^{j+1}_1\right) - \frac{\left(A^{j+1}_{i+1} + A^{j+1}_i\right)}{B^{j+1}_i}\right] \right.$$

$$+ \frac{g n_i^2}{6(2.2)} \frac{\left|Q^{j+1}_i\right| Q^{j+1}_i}{A^{j+1}_i \left(R^{j+1}_i\right)^{4/3}} \left[\frac{-7}{A^{j+1}_i} + \frac{4 \left.\frac{dB}{dh}\right|^{j+1}_i}{\left(B^{j+1}_i\right)^2} + \frac{6 \left.\frac{\partial n}{\partial h}\right|_i}{n_i B^{j+1}_i}\right] \qquad (59)$$

$$\left. - \frac{g}{2}\left(\frac{S_{o_{i+1}} + S_{o_i}}{2}\right) + \frac{K_E}{8\Delta x_i}\left[\left(\frac{Q^2}{A^2}\right)^{j+1}_{i+1} - \left(\frac{Q^2}{A^2}\right)^{j+1}_i \left(1 + \frac{2A^{j+1}_{i+1}}{A^{j+1}_i}\right)\right] \right\}$$

50. The following subsidiary manipulations have been used. First, area and depth gradients are related by the equation $\partial A/\partial z = B\ \partial h/\partial x$ and second, the hydraulic radius is approximately equal to the hydraulic depth or $R = A/B$. The explicit relationship between A and h is permitted if the general empirical form for most channel cross sections is used, i.e.

$$A = a_{oh} + a_1 h^{a_2} \qquad (60)$$

By appropriate selection of the coefficients, most regularly formed channel cross sections can be modeled by this function. A library of such shapes was created and will be described in detail in Part VI. It is also noted that Equations 52-59 have been multiplied by $2\Delta t_j$ in the code.

<u>Calculation Procedure</u>

51. The nonlinear solution procedure is used to calculate the new flows Q^{j+1}_i and areas A^{j+1}_i as follows:

> <u>Step 1</u>. Assume that Q^j_i and A^j_i are known either from initial conditions or from the completion of the previous Newton-Raphson solution.

Step 2. By insertion of Q_i^j and A_i^j into the equations for F and G, form the k - 1 residuals R_{ji}^1, by assuming that k - 1 estimate for $Q_i^{j+1,k}$ and $A_i^{j+1,k}$ is the jth time-step value for Q and A or the initial condition (Step 1). For k > 1, the previous k^{th} estimates of Q_i^{j+1} and A_i^{j+1} are used.

Step 3. After forming the residuals, the gradients are formed from Equations 52-59, again by using the k - 1 estimates for Q and A. For subsequent iterations, the previous k^{th} estimates are used.

$$
\begin{bmatrix}
\frac{\partial G_0}{\partial A_1} & \frac{\partial G_0}{\partial Q_1} & & & & & & & & & & \\
\frac{\partial F_1}{\partial A_1} & \frac{\partial F_1}{\partial Q_1} & \frac{\partial F_1}{\partial A_2} & \frac{\partial F_1}{\partial Q_2} & & & & & & & & \\
\frac{\partial G_1}{\partial A_1} & \frac{\partial G_1}{\partial Q_1} & \frac{\partial G_1}{\partial A_2} & \frac{\partial G_1}{\partial Q_2} & & & & & & & & \\
& & & \cdots\cdots\cdots & & & & & & & & \\
& & & & \frac{\partial F_i}{\partial A_i} & \frac{\partial F_i}{\partial Q_i} & \frac{\partial F_i}{\partial A_{i+1}} & \frac{\partial F_i}{\partial Q_{i+1}} & & & & \\
& & & & \frac{\partial G_i}{\partial A_i} & \frac{\partial G_i}{\partial Q_i} & \frac{\partial G_i}{\partial A_{i+1}} & \frac{\partial G_i}{\partial Q_{i+1}} & & & & \\
& & & & & & & \cdots\cdots\cdots & & & & \\
& & & & & & & & \frac{\partial F_{N-1}}{\partial A_{N-1}} & \frac{\partial F_{N-1}}{\partial Q_{N-1}} & \frac{\partial F_{N-1}}{\partial A_N} & \frac{\partial F_{N-1}}{\partial Q_N} \\
& & & & & & & & \frac{\partial G_{N-1}}{\partial A_{N-1}} & \frac{\partial G_{N-1}}{\partial Q_{N-1}} & \frac{\partial G_{N-1}}{\partial A_N} & \frac{\partial G_{N-1}}{\partial Q_N} \\
& & & & & & & & & & \frac{\partial G_N}{\partial A_N} & \frac{\partial G_N}{\partial Q_N}
\end{bmatrix}
\begin{bmatrix}
dA_1^k \\ dQ_1^k \\ dA_2^k \\ dQ_2^k \\ \cdots \\ dA_i^k \\ dQ_i^k \\ dA_{i+1}^k \\ dQ_{i+1}^k \\ \cdots \\ dA_{N-1}^k \\ dQ_{N-1}^k \\ dA_N^k \\ dQ_N^k
\end{bmatrix}
=
\begin{bmatrix}
R_{2,0}^k \\ R_{1,1}^k \\ R_{2,1}^k \\ R_{1,2}^k \\ \cdots \\ R_{1,i}^k \\ R_{2,i}^k \\ R_{1,i+1}^k \\ R_{2,i+1}^k \\ \cdots \\ R_{1,N-1}^k \\ R_{2,N-1}^k \\ R_{2,N}^k
\end{bmatrix}
\tag{61}
$$

Step 4.   From Equation 50 and Steps 1 through 3, a coefficient
matrix is formed and assembled as in Equation 61; a system
of linear simultaneous equations $[M]^k (D)^k = (R)^k$ is
solved for $(D)^k$.

Step 5.   When solved for, the departure vector $(D)^k$ is added to
the old estimates of Q and A , to give a new estimate
for Q and A , i.e.

$$Q_i^{j+1,k+1} = Q_i^{j+1,k} + dQ_i^k \qquad (62)$$

and

$$A_i^{j+1,k+1} = A_i^{j+1,k} + dA_i^k \qquad (63)$$

Step 6.   A check is made to see how close $Q_i^{j+1,k+1}$ and $A_i^{j+1,k+1}$ are
to $Q_i^{j+1,k}$ and $A_i^{j+1,k}$ and if the largest difference is
less than some specified tolerance, the iteration stops and
the new values of A and Q for the j+1 time-steps are
at hand. If the tolerance is exceeded, return to Step 2
and using $Q_i^{j+1,k+1}$ and $A_i^{j+1,k+1}$, repeat Steps 2 through 6.

52. The programming of this routine is discussed in the program structure
section, Part VI.

PART IV: THE NUMERICAL SOLUTION FOR CONSTITUENT TRANSPORT

Rationale

53. By inspection of the governing equations for flow and elevation (Equations 34 and 35) and constituent concentration (Equation 36), it is apparent that the flow field is not affected by constituent concentration. A complete prediction of Q and A can be made without one's solving for the constituents. This is convenient since the constituent equation can be solved separately, thereby providing economy. Since any number of species could conceivably be solved for, the numerical solution must be as quick as possible. This suggests the use of explicit time-marching procedures. However, simple explicit (and also simple implicit methods) time marching for the advection problem is a very severe test for which, unfortunately, the simple methods fail to provide the desired accuracy. It should be noted that pure advection is the single most difficult test for a numerical method since the initial concentration distribution imposed on the problem must be numerically advected or transported without loss of mass, shape, and peak value, or distortion of the statistics of the distribution including mean, variance, skew, and kurtosis.

54. A powerful and accurate explicit method based upon compact, but fourth-order accurate, numerical expressions is used to solve Equation 36 for advection. An implicit fractional step method is subsequently used for the dispersion term.

The Governing Equation

55. The general form of the equation to be solved is, for a mass concentration α

$$\frac{\partial(A\alpha)}{\partial t} + \frac{\partial(UA\alpha)}{\partial x} = \frac{\partial}{\partial x}\left(DA\,\frac{\partial\alpha}{\partial x}\right) + q\gamma + S* \tag{64}$$

where

> D = dispersion coefficient
>
> γ = concentration of the runoff input to the channel by distributed flow q
>
> $S*$ = source/sink term which accounts for changes in α due solely to biological and chemical reasons

The variable $S*$ is decomposed into a source/sink term that is a function of the present concentration of α and a function that is not; therefore

$$\frac{\partial(A\,\alpha)}{\partial t} + \frac{\partial(Q\,\alpha)}{\partial x} = \frac{\partial}{\partial x}\left(DA\frac{\partial\alpha}{\partial x}\right) + q\gamma + AC_1\alpha + AC_2 \tag{65}$$

where C_1 has units of $(1/t)$ and C_2 has units of (M/L^3t) . This is the form of the equation to be solved, and it requires the initial and boundary conditions as stipulated in Equations 26 and 33, respectively.

56. The governing equation after chain rule differentiation is reassembled in the form

$$\frac{\partial\alpha}{\partial t} + U\frac{\partial\alpha}{\partial x} + \frac{\alpha\,\partial U}{\partial x} - \left(\frac{D}{A}\frac{\partial A}{\partial x} + \frac{\partial D}{\partial x}\right)\frac{\partial\alpha}{\partial x} = D\frac{\partial^2\alpha}{\partial x^2} - \mathcal{C}_1\alpha + \mathcal{C}_2 \tag{66}$$

where

$$\mathcal{C}_1 = \left(\frac{1}{A}\frac{\partial A}{\partial t} + \frac{U}{A}\frac{\partial A}{\partial x} - C_1\right) \tag{67}$$

and

$$\mathcal{C}_2 = \left(C_2 + \frac{q\gamma}{A}\right) \tag{68}$$

Equation 66 can be rewritten

$$\frac{\partial \alpha}{\partial t} + \bar{u} \; \frac{\partial \alpha}{\partial x} = D \; \frac{\partial^2 \alpha}{\partial x^2} - \mathcal{C}_1 \alpha + \mathcal{C}_2 - \alpha \frac{\partial U}{\partial x} \tag{69}$$

where

$$\bar{u} = U - \left(\frac{\partial D}{\partial x} + \frac{D}{A} \frac{\partial A}{\partial x} \right) = U - DDA \tag{70}$$

and

$$DDA = \frac{\partial D}{\partial x} + \frac{D}{A} \left(\frac{\partial A}{\partial x} \right) \tag{71}$$

From continuity (Equation 34)

$$\frac{\partial A}{\partial t} + A \; \frac{\partial U}{\partial x} + U \; \frac{\partial A}{\partial x} = q \tag{72}$$

Using Equation 72, the second and fourth terms on the right side of Equation 69 can be combined as

$$-\alpha \left(\mathcal{C}_1 + \frac{\partial U}{\partial x} \right) = -\alpha \left(\frac{q}{A} - C_1 \right) \tag{73}$$

and $\mathcal{C}_1$ is redefined as

$$\mathcal{C}_1 = \frac{q}{A} - C_1 \tag{74}$$

Thus Equation 69 becomes

$$\frac{\partial \alpha}{\partial t} + \bar{u}\,\frac{\partial \alpha}{\partial x} = D\,\frac{\partial^2 \alpha}{\partial x^2} - \mathfrak{c}_1 \alpha + \mathfrak{c}_2 \qquad (75)$$

57. The $\mathfrak{c}_1$ and $\mathfrak{c}_2$ terms of Equation 75 are written in expanded form so that they can be followed more easily in the code; thus

$$\frac{\partial \alpha}{\partial t} + \bar{u}\,\frac{\partial \alpha}{\partial x} = D\,\frac{\partial^2 \alpha}{\partial x^2} + \frac{q}{A}(\gamma - \alpha) - K_s \alpha + \text{SINKS} \qquad (76)$$

where

$K_s = -C_1 =$ biochemical uptake or decay rates (+) and growth rates (-)

SINKS $= C_2 =$ biochemical sources (+) and sinks (-)

58. The left side of Equation 76 is solved for the new time level value of α (α_{i+1}^{j+1}) with the fourth-order explicit scheme. Using this new value, α_{i+1}^{j+1} is incremented due to the effects of the second, third, and fourth terms on the right side of Equation 76. The SINKS and K_s values are interpolated values between nodes (see Equation 90). Finally, the solution is completed by adding the effect of diffusion (first term on right side of Equation 76), which is computed implicitly.

Fourth-Order Explicit Scheme

59. The compact, fourth-order accurate scheme presented by Holly and Preissmann (1978) is a satisfactory scheme for 1-D advection calculations and is used for all mass transport calculations.

Polynomial assumption

60. It is assumed that the variation of any quantity between two adjacent spatial nodes is not linear but is depicted by a cubic polynomial such that for a variable Y

$$Y(\xi) = A\xi^3 + B\xi^2 + D\xi + E \qquad (77)$$

where

$$\xi = \frac{u*\tau}{x_{i+1} - x_i} \tag{78}$$

where

u* = average characteristic velocity

τ = time-step $t_{i+1} - t_i$

The coefficients for the polynomial are evaluated from the conditions that

$$Y(1) = \alpha_i^j \ ; \ Y(0) = \alpha_{i+1}^j \ ; \ \dot{Y}(1) = \alpha x_i^j \ ; \ \dot{Y}(0) = \alpha x_{i+1}^j$$

$$\alpha x = \frac{\partial \alpha}{\partial x} \tag{79}$$

$$\dot{Y}(\xi) = \frac{dY}{dx}\bigg|_\xi$$

After some algebra

$$Y(\xi) = A_1 \alpha_i^j + A_2 \alpha_{i+1}^j + A_3 \alpha x_i^j + A_4 \alpha x_{i+1}^j \tag{80}$$

where

$$A_1 = \xi^2(3 - 2\xi) \tag{81}$$

$$A_2 = 1 - A_1 \tag{82}$$

$$A_3 = \xi^2(1 - \xi)\left(x_{i+1} - x_i\right) \tag{83}$$

$$A_4 = -\xi(1 - \xi)^2\left(x_{i+1} - x_i\right) \tag{84}$$

Note that since the polynomial is parameterized by first derivatives, an equation for the first derivatives is also necessary. A polynomial for the first derivatives is formed from

$$\dot{Y}(\xi) = b_1 \alpha_i^j + b_2 \alpha_{i+1}^j + b_3 \alpha x_i^j + b_4 \alpha x_{i+1}^j \qquad (85)$$

where

$$b_1 = 6\xi(\xi - 1)(x_{i+1} - x_i)^{-1} \qquad (86)$$

$$b_2 = -b_1 \qquad (87)$$

$$b_3 = \xi(3\xi - 2) \qquad (88)$$

$$b_4 = (\xi - 1)(3\xi - 1) \qquad (89)$$

Both $Y(\xi)$ and $\dot{Y}(\xi)$ will be used to determine α_{i+1}^{j+1} and αx_{i+1}^{j+1}, respectively, resulting from pure advective transport (the left side of Equation 76).

Solution procedure for α_{i+1}^{j+1}

61. Any variable or coefficient can be interpolated to obtain the average value between nodes as

$$K* = \frac{\left[K_{i+1}^{j+1} + K_{i+1}^j (1 - \xi) + K_i^j \xi \right]}{2} \qquad (90)$$

where

$$\xi = \frac{u*\tau}{x_{i+1} - x_i}$$

39

and u* is the average characteristic velocity between nodes. Likewise, u*
can be found by placing a linear interpolation between $\bar{u}$ and $\bar{u}_{i+1}$, based on
u* , or from Equation 90

$$u* = \frac{\left[\bar{u}_{i+1}^{j+1} + \bar{u}_{i+1}^{j} - \left(\frac{u*\tau}{x_{i+1} - x_i}\right)\left(\bar{u}_{i+1}^{j} - \bar{u}_i^{j}\right)\right]}{2} \tag{91}$$

solving for u* and substituting Equation 70 for $\bar{u}$

$$u* = \frac{U_{i+1}^{j+1} + U_{i+1}^{j} - DDA_{i+1}^{j+1} - DDA_{i+1}^{j}}{2 + \frac{\tau}{x_{i+1} - x_i}\left(U_{i+1}^{j} - U_i^{j}\right)} \tag{92}$$

The term $\frac{\partial DDA}{\partial x}$ becomes zero since D and A are allowed only linear
variations between nodes.

62. The value of α_{i+1}^{j+1} due to advection, α_{i+1}^{**} , can now be determined
from

$$\alpha_{i+1}^{**} = Y(\xi) \tag{93}$$

where $Y(\xi)$ is evaluated by Equations 80-84. The values for αx_i^j and αx_{i+1}^j
in Equation 80 must be determined from

$$\alpha x_{i+1}^{j} = \dot{Y}(\xi) \tag{94}$$

for Equations 85-89. The decay, sources/sinks, and lateral inflow terms are
next added to α_{i+1}^{**} such that

$$\alpha_{i+1}^{j+1} = \alpha_{i+1}^{**} \left(1 - \tau K_s\right) + \tau \left[SINKS + \frac{q}{A}\left(\gamma - \alpha_{i+1}^{**}\right)\right] \tag{95}$$

Solution procedure for αx_{i+1}^{j+1}

63. A spatial derivative of the transport equation must be developed to update αx_{i+1}^{j+1} . This development is done by taking the derivative of Equation 76 with respect to x ,

$$\frac{\partial \alpha'}{\partial t} + \bar{u}\frac{\partial^2 \alpha'}{\partial x} = D\frac{\partial^2 \alpha'}{\partial x^2} - \bar{u}'\alpha' + \left(\frac{q}{A}\right)(\gamma - \alpha) - \frac{q}{A}\alpha' - k_s\alpha' + SINKS' \tag{96}$$

where

$$\bar{u} = \bar{u} - D'$$

and the prime denotes $\frac{\partial}{\partial x}$; thus $\alpha' = \frac{\partial \alpha}{\partial x} = \alpha x = \alpha_x$. Now $\bar{u} = U'$ since D and A are allowed only linear variations between nodes. If D' is small and D at a node is constant or changes slowly over time, the characteristic velocity for the spatial gradient is approximated by $u** = u^* - D'$. With the value $u**$, $\xi*$ can be determined from

$$\xi* = \frac{\tau u**}{x_{i+1} - x_i} \tag{97}$$

64. Now $\dot{Y}(\xi^*)$ can be evaluated through Equations 85-89. This evaluation yields $\alpha x_{i+1}^{**} = \dot{Y}(\xi^*)$, which is the solution to the left side of Equation 96. All but the first term on the right side of Equation 96 are added to α_{i+1}^{**} such that

$$\alpha x_{i+1}^{j+1} = \alpha x_{i+1}^{\bullet\bullet} \left[1 - \frac{\tau \left(U_{i+1}^j - U_i^j \right)}{x_{i+1} - x_1} - \tau \left(K_s + \frac{q}{A} \right) \right]$$

$$+ \tau \left[\left(\frac{q}{A} \right)' (\gamma - \alpha) - \alpha K_s' + SINKS' \right]$$

(98)

The final update for αx_{i+1}^{j+1} is completed when the implicit solution due to the first term on the right side of Equation 96 (diffusion) is added.

65. The derivatives for coefficients used in Equations 96-98 are found by

$$\frac{\partial K^j}{\partial x} = \frac{K_{i+1}^j - K_i^j}{x_{i+1} - x_i}$$

(99)

and

$$\frac{\partial^2 K^j}{\partial x^2} = \frac{\left(\left. \frac{\partial K^j}{\partial x} \right|_{i+1} - \left. \frac{\partial K^j}{\partial x} \right|_i \right)}{x_{i+1} - x_i}$$

(100)

The steps outlined in this section are repeated for each spatial node before moving on to the next time line.

Implicit Diffusion

66. The constituent transport (Equation 76) and the spatial derivative transport (Equation 96) equations are now ready to have the effects of diffusion added by

$$\alpha_i^{j+1} = \alpha_i^j + \tau \, D \, \frac{\partial^2 \alpha}{\partial x^2}$$

<div align="right">(101)</div>

$$\alpha x_i^{j+1} = \alpha x_i^j + \tau \, D \, \frac{\partial^2 \alpha'}{\partial x^2}$$

where now the j time-level is actually at the new time-level following the advection and kinetic reactions but just prior to diffusion. The diffusion terms are approximated by difference equations, approximately centered in space and time. That is, new time information ($j+1$) is weighted by a factor $\theta = 0.55$ to enhance stability. The second derivative is replaced by the difference operator D_{xx} ; thus

$$\frac{\partial^2 \alpha}{\partial x^2} = \theta D_{xx}\left(\alpha^{j+1}\right) + \left(1 - \theta\right) D_{xx}\left(a^j\right)$$

<div align="right">(102)</div>

67. In space, centering would be exact for a regular grid. However, with an irregular grid, a quadratic interpolation is used by taking a Taylor series such that

$$D_{xx}(\alpha) = 2\left[\frac{\alpha_{i+1} - \alpha_i}{\Delta x_i\left(\Delta x_{i-1} + \Delta x_i\right)} + \frac{\alpha_{i-1} - \alpha_i}{\Delta x_{i-1}\left(\Delta x_{i-1} + \Delta x_i\right)}\right]$$

<div align="right">(103)</div>

where

$$\Delta x_i = x_{i+1} - x_i$$

Applying Equation 103 to the difference operator (Equation 102) which is subsequently applied to Equation 101 results in Equation 104 for α_i^{j+1},

$$\alpha_i^{j+1} = \alpha_i^j + 2\tau D\theta \left[\frac{\alpha_{i+1}^{j+1} - \alpha_i^{j+1}}{\Delta x_i (\Delta x_{i-1} + \Delta x_i)} + \frac{\alpha_{i-1}^{j+1} - \alpha_i^{j+1}}{\Delta x_{i-1} (\Delta x_{i-1} + \Delta x_i)} \right]$$

(104)

$$+ \, 2\tau D(1-\theta) \left[\frac{\alpha_{i+1}^j - \alpha_i^j}{\Delta x_i (\Delta x_{i-1} + \Delta x_i)} + \frac{\alpha_{i-1}^j - \alpha_i^j}{\Delta x_{i-1} (\Delta x_{i-1} + \Delta x_i)} \right]$$

An equation similar to Equation 104 is developed for αx_i^{j+1}.

68. When Equation 104 is applied to every node for i going from 2 to N-1, N-2 equations for N unknowns are formed. The system is completed by the upstream and downstream boundary conditions. The upstream boundary condition is given explicitly in the data (or inferred for the spatial derivative of diffusion). The downstream boundary condition is simply that the last node is not affected by diffusion.

69. The system of equations is assembled in tridiagonal form with all new values (j+1) on the left side and all old (j) values on the right. The implicit solution is accomplished with the Thomas Algorithm (subroutine TRIDAG). Subroutine TRIDAG is also used to calculate a cubic spline through the initial data.

Stability Requirements

70. The solution of the constituent transport equation has a Courant number restriction for numerical stability. This is due to the explicit solution scheme for advection. The Courant number, defined as

$$CN = \frac{u\tau}{\Delta x}$$

(105)

must be less than 1.0 to preserve stability. This requirement is not considered too restrictive, although it must be kept in mind during an application.

71. This version of the RIV1Q model is coded to expect flows in the downstream direction only. Erroneous numerical solutions can be experienced if flows in the upstream direction are experienced. Upstream flows do not

44

usually occur in nontidal streams. However, highly unsteady flows may reflect off downstream control structures resulting temporarily in upstream "reverse flows." RIV1H can yield reverse flows that are realistic, but reverse flows that persist very long will yield unrealistic results from RIV1Q.

72. In a model application, it was possible to damp out reverse flows that were reflected off a dam downstream of a peaking hydropower dam by providing a minimum flow during nongeneration periods. This minimum flow was equivalent to the dam leakage. RIV1Q is being modified to allow reversing flows.

PART V: SCHEMATIC MODEL OF RIVER WATER QUALITY

73.   In the following paragraphs, the mathematical formulations of the sources, sinks, and reaction kinetics for various water quality components are presented. The most significant of these are temperature and DO. However, other variables are also included because of their effect on DO. These variables include carbonaceous biochemical oxygen demand (CBOD) as well as nitrogen and phosphorous compounds. The growth and decay of algae and macrophytes are also modeled to complete the DO balance. Algae is modeled as a state variable (subject to advective and dispersive transport) but macrophytes are not. Nitrite plus nitrate-nitrogen is included to complete the nitrogen cycle. Organic and phosphate-phosphorus are modeled to complete the phosphorous cycle. Coliform bacteria are included as a state variable because of their importance as an indicator of stream pollution. The coliform bacteria variable can be used to model other nonconservative variables or a conservative variable (by setting the die-off rate to zero). Dissolved iron and manganese are included because of their impact on DO in streams below dams with anoxic hypolimnetic releases. Thus, 12 modeled variables are included; those to be modeled are selected by the user. Transport of each state variable is accomplished using Equation 76. Figure 5 shows schematically how the variables interact. The constituents as they appear in the code are as follows: temperature, $C(1,I)$; CBOD, $C(2,I)$; organic nitrogen, $C(3,I)$; ammonia nitrogen, $C(4,I)$; nitrate nitrogen, $C(5,I)$; organic phosphorus, $C(6,I)$; phosphate phosphorus, $C(7,I)$; dissolved manganese, $C(8,I)$; dissolved iron, $C(9,I)$; dissolved oxygen, $C(10,I)$; coliform bacteria, $C(11,I)$; and algae, $C(12,I)$.

74.   All of the state variables may be simulated or "bypassed." If the user elects to bypass a particular state variable (using the bypass options as described in Part VII, then the concentrations will be held to those specified in the initial conditions for the duration of the simulation.

Temperature

75.   It may be necessary to account for the effects of temperature changes along a channel in order to account for temperature gradients created by cultural inputs such as power plant effluents or natural processes. In addition to its intrinsic value, temperature information is required in order to correct kinetic rate coefficients. Stream temperatures may be read in
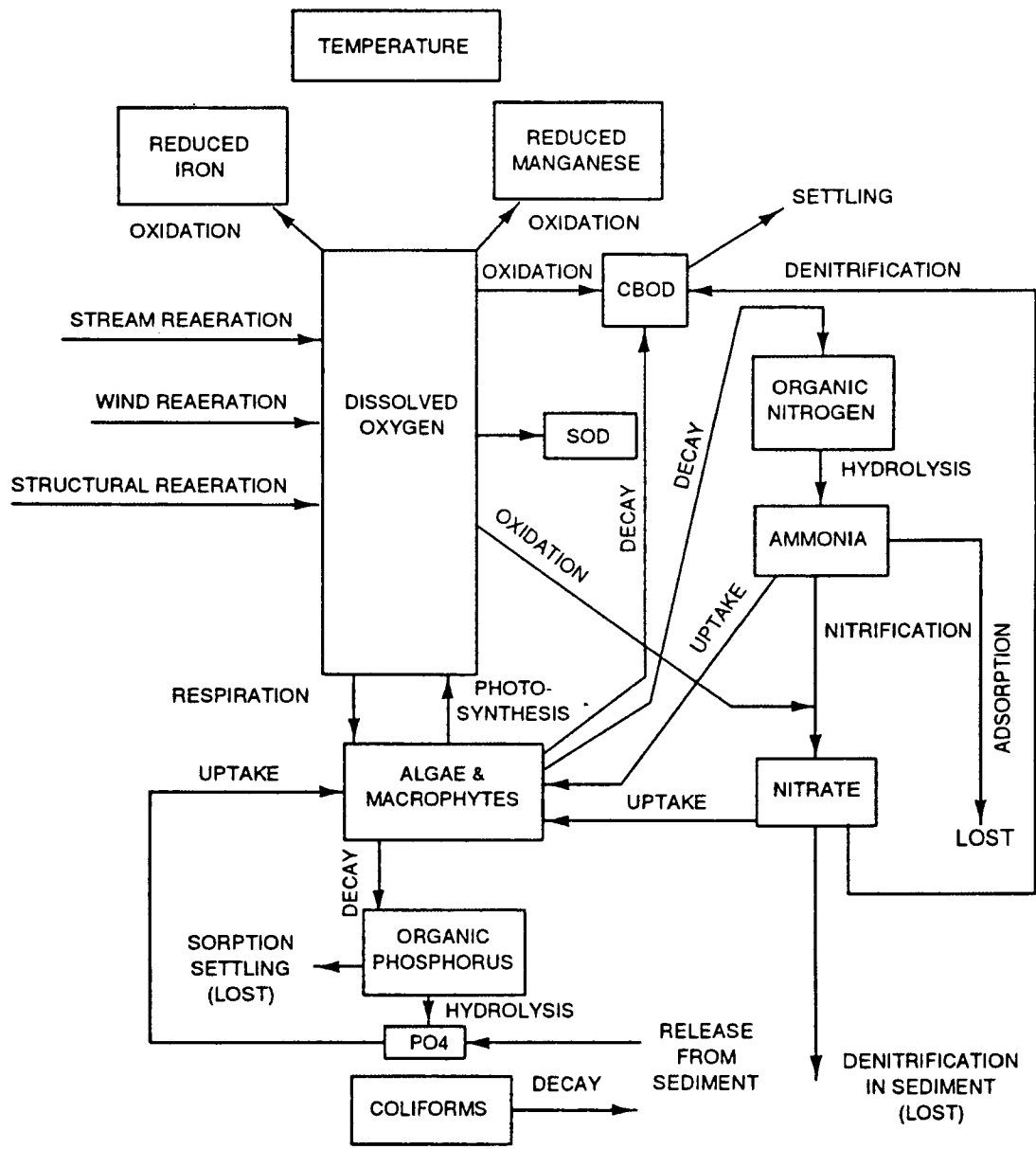
Figure 5. CE-QUAL-RIV1 water quality compartmental diagram

directly or may be calculated/predicted from the solution of an appropriate thermal energy balance.

76. The transport equation (Equation 76) developed in Part IV, is used to solve for each water quality variable, α ; thus for temperature, $\alpha = T$. The specification of the source/sink term and the decay term of Equation 76 must account for all the mechanisms other than advection, diffusion, and lateral inflows, which cause net transfers of energy to the control volume.

Heat balance approach

77. In general, the heat balance approach used here is the same as the one used in the QUAL2E model (1987). The external sources and sinks (excluding lateral inflows) for heat are described by

$$H_N = H_S + H_L - H_E - H_B \pm H_C \tag{106}$$

where

H_N = net heat transfer, $\dfrac{\text{heat energy}}{\text{surface area} * \text{time}}$

H_S = net short-wave radiation

H_L = net long-wave radiation

H_E = heat loss because of evaporation

H_B = heat loss because of back radiation of the water

H_C = heat transferred by conduction at the water surface and the bottom

78. There are two options for computing H_N . In the first option, each term in Equation 106 is computed (i.e., subroutine HEATFLUX) and added to form H_N , which is converted to a rate of temperature change by

$$\dot{T} = \frac{H_N}{\rho C_p H} \cdot \text{conv} \tag{107}$$

where

$\dot{T}$ = rate of temperature change, $\dfrac{\text{degrees}}{\text{time}}$

ρ = specific mass of water, $\dfrac{\text{mass}}{\text{volume}}$

C_p = specific heat of water, $\dfrac{\text{heat energy}}{\text{mass} * \text{degree}}$

H = hydraulic depth, $\dfrac{A}{B}$, length

conv = conversion factor from English to metric units (HEATFLUX performs computations in English units)

48

The variable ΔT is used for the SINKS term in Equation 76, and the K_S term is zero. The algorithms in subroutine HEATFLUX are based on those used in QUAL-II (Roesner, Giguere, and Evenson 1981) to compute the terms in Equation 106. The computations within HEATFLUX depend on: water temperature, T_S ; time of year and day, site latitude, longitude, and elevation; and local meteorological data. The previously computed value (previous time-step) is used for T_S . The meteorological data, obtained from the National Oceanic and Atmospheric Administration (NOAA), consist of dry and wet bulb temperatures (°C), wind speed, cloud cover, and barometric pressure. Generally, meteorological data are collected and reported in 1- or 3-hr intervals. This method is referred to as a direct energy balance because independent determination of the full terms on the right side of Equation 106 is made within the simulation, in contrast to the alternative heat exchange method.

Equilibrium temperature approach

79. The alternative method uses the well-known equilibrium temperature approach, as developed by Edinger, Duttweiler, and Geyer (1968) and Edinger, Brady, and Geyer (1974), to account for the effects of surface heating and radiation. The equilibrium temperature approach is based on the concept that heat exchange does not occur when the water temperature, T_S , equals the equilibrium temperature, T_E ; thus, $H_N = 0$. With $T_S = T_E$ and $H_N = 0$, the terms on the right side of Equation 106 can be expressed in terms of T_E , and T_E can be solved iteratively. Net heat transfer, H_N , is computed from

$$H_N = K_E \left(T_E - T_S \right) \tag{108}$$

where K_E is the heat exchange coefficient (heat energy/surface area/time/ degree). Using a computer program with the same meteorological data used for the direct energy balance, T_E and K_E are computed independent of the simulation. This program can be obtained from WES or the Hydrologic Engineering Center (HEC). A description of the T_E and K_S computation is given in Edinger, Brady, and Geyer (1974).

80. If the equilibrium temperature approach is followed, the model uses a single, constant value for T_E and K_E . This is equivalent to modeling stream temperature under constant meteorological conditions. The computations of T_E and K_E in the WES and HEC program are based on daily average values for meteorological data; thus daily average values of T_E and K_E are

produced. Daily average values for T_E and K_E are adequate for reservoir thermal simulations but may not be adequate for riverine applications where diel fluctuations are important. In these cases, the direct energy balance method should be used.

81. By inspection of Equation 108, the K_S coefficient of Equation 76 is K_E (KTS in the code) and the SINKS term is $K_E T_E$ (KTS*TEQ in code). Thus, KTS is converted to units of per time (per day) by dividing by $(\rho c_p H)$ with the proper conversion units.

82. Either heat exchange method can be augmented with bottom heat transfer by adding the term K_B (TSINK - T_S) where TSINK is the bottom/ ground temperature in degrees and K_B is the bottom heat transfer coefficient in units of per day. In this case, K_B and TSINK must be specified, the K_S term of Equation 76 is incremented by K_B (KTB in the code), and the SINKS term of Equation 76 is incremented by K_B*TSINK .

83. The latest release of this model includes mechanisms to allow for algae self-shading as this effect can be significant for small streams. In future releases, consideration should be given to allow the solar radiation that passes through the water column (for clear, shallow streams) to be absorbed by the bottom and reradiated as heat, thus providing bottom warming later. At present, all solar radiation is absorbed by the water column.

Carbonaceous Biochemical Oxygen Demand

84. The CBOD represents (as chemically equivalent oxygen) the ultimate amount of biodegradable organic matter present, excluding organic nitrogen. Model output is given in terms of first-stage biochemical oxygen demand (BOD) (i.e., CBOD); input to the model must also be in terms of CBOD's. If BOD_5 data are available, these must be converted to CBOD using the classical BOD equation.

$$CBOD = \frac{BOD_5}{1 - e^{(-K \cdot 5)}} \qquad (109)$$

where

   CBOD = ultimate first-stage (carbonaceous) BOD, g/m^3
   BOD_5 = 5-day (carbonaceous) BOD, g/m^3
   K = bottle BOD decay rate, day^{-1}

The BOD5 should be determined using nitrification-inhibited samples to avoid double counting of the nitrogenous BOD. If values of the reaction rate constant K are not available for the wastewater at hand, they may be approximated using Figure 6.

NOTE: K_1 (BASE E) = 2.3 K_1 (BASE 10)
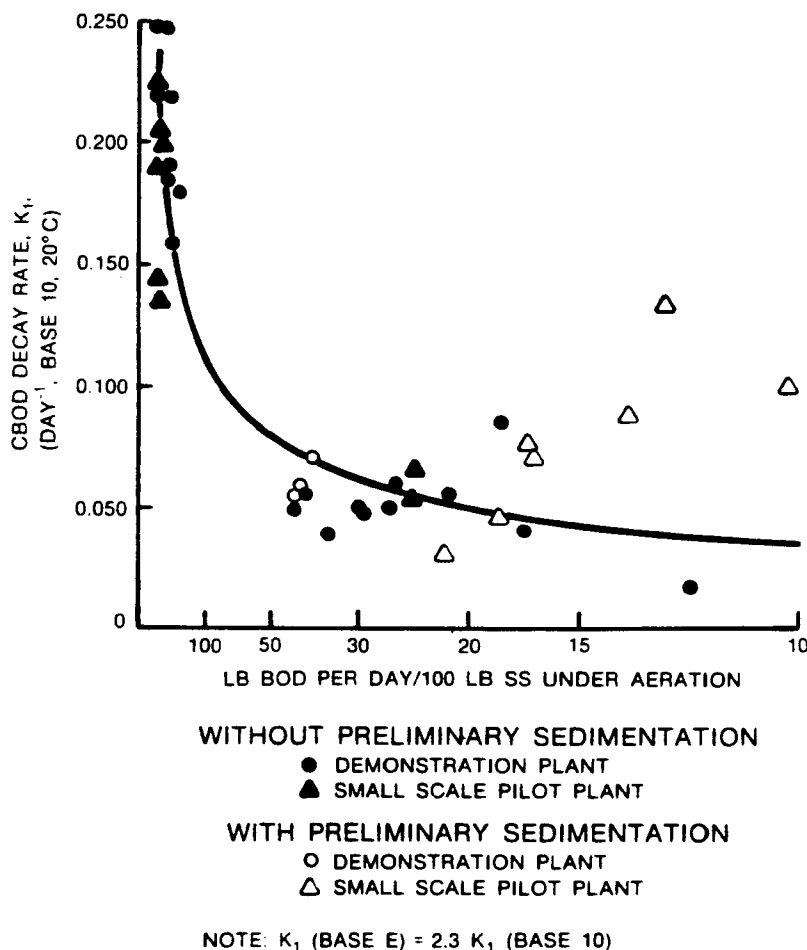
Figure 6. Variation of the bottle K_1 with the degree of biological wastewater treatment (after Schroepfer, Robins, and Susag 1960) (Copyright 1960 Water Pollution Control Federation (WPCF), reproduced by permission of WPCF)

85. CBOD is a result of biochemical oxidation of organics by heterotrophic bacteria using either oxygen or nitrate as the terminal electron acceptor. The latter process is called denitrification. In this model, it is assumed that oxygen is the terminal electron acceptor at high DO's and nitrate at low DO's. Thus, the basic reaction for CBOD is

$$\begin{pmatrix} \text{Rate of CBOD loss due to} \\ \text{oxidation and settling} \\ \text{g/m}^3\text{/day} \end{pmatrix} = - \left(K1 + KDN + \frac{CBODSR}{H} \right) * CBOD \qquad (110)$$

where

   $K1$ = temperature corrected rate coefficient for aerobic oxidation of CBOD, day^{-1}

   KDN = temperature corrected rate coefficient for nitrate reduction and anaerobic CBOD oxidation, day^{-1}

   $CBOD$ = concentration of CBOD, g O_2/m^3, from previous time-step, $C(2,I)$

   $CBODSR$ = rate coefficient for CBOD removal by settling, m/day

   H = depth, m

In the present model, a fraction (FCBOD) of decaying algae and macrophytes is assumed to contribute to CBOD, the remainder (1-FCBOD), being oxidized immediately for energy.

$$\begin{pmatrix} \text{CBOD gain from} \\ \text{algae/macrophyte decay} \\ \text{g-BOD/m}^3\text{/day} \end{pmatrix} = OPDECY * \left(\frac{(FCBOD*DO) + KOCB1}{DO + KOCB} \right) * \\ (ALGADK + MDEATH) \qquad (111)$$

where

   $OPDECY$ = oxygen-to-biomass ratio for oxygen production by algae and macrophytes when ammonia is nitrogen source, suggested value, 1.59

   $FCBOD$ = fraction of algal and macrophyte mortality contributing to CBOD

   DO = average dissolved oxygen along the characteristic, g/m^3, DOX in the code

   $KOCB1$ = Monod half velocity constant for oxygen-limited aerobic systems, g O_2/m^3

   $ALGADK$ = algal death rate, $g/(m^3day)$

   $MDEATH$ = macrophyte death rate, $g/(m^3day)$

It should be realized that K_S is (K1 + KDN + CBODSR) , SINKS is the CBOD contribution from algal/macrophyte decay, and α in Equation 76 is CBOD by inspection of Equation 110.

   86. The rate coefficients K1 and KDN are presumed to be functions of the ambient water temperature, the local DO, and the local nitrate concentration. The temperature correction is taken from Streeter and Phelps (1925), as is customary, although arguable. The coefficient K1 is assumed to increase with DO according to a simple Monod function (cf. Hoover and Porges

1952), and the coefficient KDN is assumed to decrease according to an analogous formula. The coefficient KDN is also assumed to depend on nitrate in the same way that K1 depends on oxygen.

$$K1 = AK1 * \left[TBIOS^{(TEMP-20.)} \right] * \left(\frac{DO}{DO + KOCB1} \right) \qquad (112)$$

$$KDN = ADN * \left[TBIOS^{(TEMP-20.)} \right] * \left(\frac{KOCBDN}{DO + KOCBDN} \right) * \left(\frac{NO_3-N}{NO_3-N + KNCBDN} \right) \qquad (113)$$

where

AK1, ADN = uncorrected rate coefficients for oxidation and denitrification of CBOD, respectively, day^{-1}

TBIOS = temperature coefficient for biological processes, unitless

TEMP = T_S = the ambient stream temperature, °C

DO = local stream oxygen concentration, g O_2/m^3

KOCB1 = Monod half-velocity constant for oxygen-limited aerobic systems, g O_2/m^3

KOCBDN = denitrification inhibition half-velocity constant, g O_2/m^3

NO_3-N = local nitrate-nitrogen concentration, g N/m^3

KNCBDN = Monod half-velocity constant for nitrate-limited denitrification, g N/m^3

CBODSR is not corrected for environmental conditions. In addition, no provision for fermentation has been provided, so in the absence of oxygen and nitrate, there is no CBOD removal except through settling.

87. The net rate of accumulation of CBOD may be written as

$$\begin{pmatrix} \text{Net rate of} \\ \text{accumulation} \\ \text{of CBOD} \\ g/m^3/day \end{pmatrix} = \begin{matrix} - \text{(AEROBIC OXIDATION OF CBOD)} - \text{(DENITRIFICATION)} \\ \\ - \text{(SETTLING)} + \text{(ALGAE/MACROPHYTE DECAY)} \end{matrix} \qquad (114)$$

and the final equation is

$$= \left(K_1 + KDN + \frac{CBODSR}{H} \right) * CBOD + OPDECY * \frac{(KOCB1 + DOX*FCBOD)}{(DOX + KOCB1)} * (ALGADK + MDEATH) \qquad (115)$$

88. The forms of nitrogen (N) recognized by the model are organic nitrogen, ammonium, and nitrate. Nitrite is not considered because the over-all rate of nitrification is ammonia-limited (Parker et al. 1975), and stream surveys do not exhibit significant increases in nitrite in nitrifying reaches (Garland 1978, Miller and Jennings 1979).

Organic nitrogen

89. Organic nitrogen is a constituent of the organic matter although its oxygen demand is not a part of the CBOD test result. In this model organic-N is produced from the decay of algae and macrophytes and lost due to hydrolysis and settling. For pure compounds, it is well known that their constituent nitrogen is released as ammonia during the exertion of the CBOD. However, in this model algae and macrophytes are assumed to release 100% organic nitrogen upon decay. This is done primarily to be consistent with other models (QUAL2E). Hydrolysis to ammonia is computed using rate coefficients from Equations 112 and 113.

$$\begin{pmatrix} \text{Rate of loss of organic nitrogen} \\ \text{due to hydrolysis to ammonia} \\ \text{and settling} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = \left(K1N + KDN + \frac{KNSET}{H} \right) * ORGAN \quad (116)$$

where

KIN = ACK * TBIOS/(1. + KOCB1(IBRANCH)/DOS

KDN = temperature corrected rate coefficient for nitrate reduction and anaerobic CBOD oxidation, day^{-1}

KNSET = rate coefficient for removal of org-N by settling, m/day

H = stream depth, m

ORGAN = concentration of org-N, C(3,I), g-N/m^3

where

KIN = temperature and DO corrected rate coefficient for organic-N, day^{-1}

ACK = rate coefficient for organic-N hydrolysis to NH$^+_4$, day^{-1}

IBRANCH = segment (branch) number

90. The rate of organic-N increase resulting from algal and macrophyte decay is taken to be proportional to the rate of decay of algae and macrophytes. The nitrogen content of algae is variously reported as 7 to

10 percent by weight (Foree and McCarty 1968, MacKenthum and Ingram 1967), whereas the nitrogen content of macrophytes is only 2 to 4 percent by weight (MacKenthum and Ingram 1967, Gerloff 1969). Assuming algae are dominant and a nitrogen content of 7.5 percent (Foree and McCarty 1968)

$$\begin{pmatrix} \text{Rate of increase of organic-N} \\ \text{due to algal/macrophyte decay} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = +\text{ANCONT} * (\text{ALGADK} + \text{MDEATH}) \qquad (117)$$

where

ANCONT = nitrogen-to-biomass ratio in algae and macrophytes, g/g

ALGADK = rate of algal decay, g/m^3/day

MDEATH = rate of macrophyte decay, g/m^3/day

91. It may be noted that all the algal/macrophyte N is assumed to be released upon decay of the cell (cf. Foree and McCarty 1968, Jewell and McCarty 1968). The algal/macrophyte decay rate itself is considered in greater detail in the following paragraphs. Although Equations 116 and 117 are conceptually appealing, it may be questioned whether the measured organic nitrogen in streams undergoes any net removal. For example, Ruane and Krenkel (1977) have published data for the Holston River that exhibit little change in organic nitrogen even in rapidly nitrifying reaches. Therefore, in some field applications organic nitrogen may be regarded as inert and deleted from the model. The net rate of change of the organic nitrogen concentration can be stated as:

$$\begin{pmatrix} \text{Net rate of accumulation} \\ \text{of organic nitrogen} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = - \begin{pmatrix} \text{loss due to} \\ \text{hydrolysis} \\ \text{and settling} \end{pmatrix} + \begin{pmatrix} \text{gain due to} \\ \text{plant decay} \end{pmatrix} \qquad (118)$$

and stated in equation form as

$$\begin{pmatrix} \text{Net rate of accumulation} \\ \text{of organic nitrogen} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = \begin{aligned} & - \left(\text{K1N} + \text{KDN} + \frac{\text{KNSET}}{\text{H}} \right) * \text{ORGAN} \\ & + [\text{ANCONT} * (\text{ALGADK} + \text{MDEATH})] \end{aligned} \qquad (119)$$

55

Ammonium

92. Ammonium increases because of the hydrolysis of organic nitrogen (if it occurs). It decreases because of conversion of ammonium to nitrate (nitrification), algal and macrophyte uptake, and sorption.

93. The increase in ammonium because of organic nitrogen hydrolysis is represented by Equation 116, with the minus sign changed to a plus because the process is a source of ammonia.

94. The rate of ammonium decrease because of nitrification is assumed to be first order

$$\left(\begin{array}{c} \text{Rate of nitrification} \\ \text{of ammonium-N to nitrate} \\ \text{g-N/m}^3\text{/day} \end{array} \right) = -KN*NH_4^+-N \qquad (120)$$

where

$\quad$ KN = nitrification rate coefficient, day^{-1}

$\quad$ NH_4^+-N = ammonia-N concentration, g-N/m^3

The rate coefficient KN is assumed to depend on oxygen and temperature in a way similar to K1

$$KN = AKN*\left[TAMMON^{(TEMP-20)} \right] * \left(\frac{DO}{DO + KON} \right) \qquad (121)$$

where

$\quad$ AKN = uncorrected rate coefficient for nitrification, day^{-1}

$\quad$ TAMMON = temperature coefficient for ammonium oxidation (suggested value is 1.1)

$\quad$ KON = Monod half-velocity constant for oxygen limitation of nitrification, g O_2/m^3

where

$$KN = \frac{AKN}{1 + \dfrac{KON*(IBRANCH)}{DOX}}$$

The form of the oxygen limitation factor is suggested by the data summarized by Parker et al. (1975). A correction for pH has not been incorporated

because of the width of the optimum range (about pH 7 to 9 according to Parker et al. (1975)).

95. The rate of uptake of ammonium nitrogen caused by algal and macrophyte growth is entirely analogous to Equation 122. However, the model assumes that the plants can use both nitrate and ammonium. The total nitrogen consumption rate is partitioned between these two forms in proportion to their relative concentrations:

$$\begin{pmatrix} \text{Rate of ammonium-N} \\ \text{decrease due to} \\ \text{algal/macrophyte uptake} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = -ANCONT * \left(\frac{NH_4^+\text{-N}}{NH_4^+\text{-N} + NO_3\text{-N}} \right) * (ALGRO + MGRATE) \quad (122)$$

where

$ANCONT$ = nitrogen-to-biomass ratio in algae and macrophytes, 0.075 suggested value

$NH_4^+\text{-N}$ = concentration of ammonium nitrogen, g-N/m^3, C(4,I)

$NO_3\text{-N}$ = concentration of nitrate nitrogen, g-N/m^3, C(5,I)

$ALGRO$ = growth rate of algae, g/m^3/day

$MGRATE$ = growth rate of macrophytes, g/m^3/day

The partitioning function was introduced primarily to avoid double counting of nitrogen uptake by plants. More sophisticated selectivity factors (e.g., O'Connor, Thomann, and DiToro 1973; Baca and Arnett 1976; DiToro et al. 1977) were not used, because there does not appear to be any empirical data or theory that warrants the use of such factors in models that lump all the photosynthetic flora together. This is not meant to deny that nutrient selection occurs in some pure cultures.

96. Finally, a term for physicochemical sorption of ammonium by the sediments is included. The need for such a process is shown by the data of Ruane and Krenkel (1977), Donigan and Crawford (1979), and Miller and Jennings (1979). The first two reports recorded losses in total inorganic nitrogen along well-aerated nitrifying reaches of about one-third to two-thirds of the total input. Ruane and Krenkel (1977) also produced plant biomass data that suggested nitrogen uptake by plants was relatively small and inadequate to account for the observed losses. The diel-average oxygen levels in all three studies ranged from 2 to 8 mg/ℓ in the reaches studied, so denitrification may not have been substantial either. Moreover, sediment analyses reported by

57

Donigan and Crawford (1979) show that in at least two small streams most of the ammonium in the water-sediment system is in the sediments.

97. The representation adopted for ammonium sorption by sediment is a simple first-order decay, like that for CBOD

$$\begin{pmatrix} \text{Rate of sorption of} \\ \text{ammonium-N by sediments} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = -KNX*NH_4^+\text{-N} \tag{123}$$

where KNX is the sorption rate coefficient, day^{-1}. The rate coefficient KNX is presumed to be a function of temperature.

$$KNX = (AKNX) * \left[TPHYS^{(TEMP-20.)} \right] \tag{124}$$

where

AKNX — uncorrected rate coefficient for sorption of ammonia by sediment, day^{-1}

TPHYS = rate coefficient for temperature correction

98. The ammonium-N reactions may be combined and stated as

$$\begin{pmatrix} \text{Net rate of} \\ \text{accumulation} \\ \text{of ammonium} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = + \begin{pmatrix} \text{hydrolysis} \\ \text{of org-N} \end{pmatrix} - \begin{pmatrix} \text{plant} \\ \text{uptake} \end{pmatrix} - (\text{nitrification}) - \begin{pmatrix} \text{sorption} \\ \text{onto} \\ \text{sediment} \end{pmatrix} \tag{125}$$

and written in equation form as

$$\begin{pmatrix} \text{Net rate of} \\ \text{accumulation} \\ \text{of ammonium-N} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = (K1N + KDN)*ORGAN + ANCONT \tag{126}$$

$$*ALGADK - \left[ANCONT* \frac{NH_4N}{NH_4^+\text{-N} + NO_3^-\text{-N}} \right] * (ALGRO + MGRATE) - (KN + KNX)*NH_4^+\text{-N}$$

In Equation 126, the coefficient of the last term is K_S and the remaining terms make up SINKS of Equation 76.

Nitrate

99. Nitrate is formed by nitrification and removed by denitrification and plant uptake. Denitrification can take place in the water column under low DO conditions and in the bottom sediments.

100. The accumulation of nitrate-N is stated as

$$
\begin{pmatrix} \text{Net rate of} \\ \text{accumulation} \\ \text{of nitrate-N} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = -\begin{pmatrix} \text{water column} \\ \text{denitrification} \end{pmatrix} + (\text{nitrification}) \\ -\begin{pmatrix} \text{plant} \\ \text{uptake} \end{pmatrix} - \begin{pmatrix} \text{sediment} \\ \text{denitrification} \end{pmatrix}
\tag{127}
$$

This can be written in equation form as

$$
\begin{pmatrix} \text{Net rate of accumulation} \\ \text{of nitrate-N} \\ \text{g-N/m}^3\text{/day} \end{pmatrix} = -(\text{ONEQUI} * \text{KDN} * \text{CBOD}) + (\text{KN} * \text{NH}_4\text{N}) \\ - \text{ANCONT} * (\text{ALGRO} + \text{MGRATE}) * \left(\frac{\text{NO}_3^- \text{-N}}{\text{NH}_4^+ \text{-N} + \text{NO}_3 \text{-N}} \right) \\ - (\text{KDSED} * \text{NO}_3^- \text{-N})
\tag{128}
$$

All terms in Equation 128 are in the SINKS term of Equation 76 and K_S is KDSED, where KDSED is the sediment denitrification rate.

101. The coefficient ONEQUI (suggested value 0.35) is a conversion factor for oxygen to nitrogen equivalents and is evaluated as follows. The half cell reaction for nitrate reduction is

$$
6\text{H}^+ + \text{NO}_3^- + 5\text{e}^- = \frac{1}{2}\text{N}_2 + 3\text{H}_2\text{O}
\tag{129}
$$
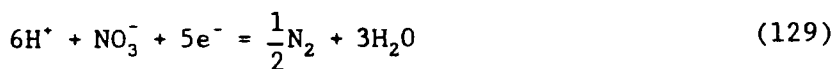
Therefore, the equivalent weight of nitrate nitrogen is 2.8 g. The equivalent weight of oxygen is 8.0 g, and the mass of nitrogen equivalent to 1 g of oxygen is 0.35 g.

102. Wheatland, Barrett, and Bruce (1959) presented some interesting data regarding nitrification and denitrification in the Thames Estuary.

59

First, denitrification occurred only in reaches with a DO below about 1 mg/ℓ, a finding they supported with laboratory data. More importantly, however, they showed that in the Thames the total inorganic nitrogen concentration declined from about 9 to about 3 mg/ℓ in a 15-mile reach nearly devoid of both oxygen and nitrate. While this decline might be interpreted as algal uptake, the depth and turbidity of the Thames argue against it, and the observed nitrogen losses might be due to simultaneous nitrification and denitrification at very low DO.

103. More recently, the studies of Williams and Lewis (1986) support the idea that nitrification and denitrification occur simultaneously because of processes within the sediments. Additionally, Hill and Sanmugadas (1985) showed that denitrification rates were significantly correlated to stream-sediment characteristics; this correlation helps to explain why recent studies have revealed considerable nitrate loss even during transport in well-oxygenated streams. Thus, denitrification via the water-sediment interface is considered in this model.

Algae and Macrophytes

104. Algae is modeled as a state variable while macrophytes are not. The growth rates of both are limited by light intensity. Furthermore, macrophytes are assumed to be benthal. This last assumption is defensible for small streams overhung by forest canopy and for turbid streams, both of which are common.

Algal growth

105. Algae are treated as a state variable; their growth rate is affected by light intensity, self-shading, and nutrients (N and P). The effect of light intensity on the algal growth rate is obtained by substituting Beer's law into a Monod-type relationship and integrating over the depth of the channel cross section. The effects of nitrogen and phosphorus on growth are accounted for using Monod-type expressions, similar to those in QUAL2E. This results in a multiplicative expression for growth also similar to one of the QUAL2E growth options. The resulting growth rate in reach I is then

$$ \text{ALGRO} = \text{C}(12,\text{I}) * \text{ALG1} * \left(\frac{1}{\text{KEXT} * \text{H}}\right) * \ln\left(\frac{\text{KLITE} + \text{SWALG}}{\text{KLITE} + \text{SWALG} * \text{EXP}(-\text{KEXT} * \text{H})}\right) * \text{FN} * \text{FP} \qquad (130) $$

or

$$ \text{ALGRO} = \text{C}(12,\text{I}) * \text{ALG1} * \text{FL} * \text{FN} * \text{FP} \qquad (131) $$

where

ALGRO = algal growth rate, corrected for light, temperature, and nutrient availability, g-biomass/m^3 day^{-1}

C(12,I) = algal concentration at node I at time t

ALG1 (KALGRO) = maximum specific algal growth rate, day^{-1}. Input as KALGRO, renamed ALG1 in subroutine SEG

KEXT = light extinction coefficient, m^{-1}, corrected for algal self-shading

H = hydraulic depth, A/B, m

KLITE = half-velocity constant for light intensity, watt/m^2

SWALG = short-wave radiation intensity at the water surface, watt/m^2

FN = nitrogen growth adjustment factor

FP = phosphorus growth adjustment factor

FL = light growth adjustment factor

106. The light extinction coefficient KEXT is coupled to algal density using the nonlinear equation

$$ \text{KEXT}(\lambda) = \lambda_0 + \lambda_1 \, [\alpha_0 * \text{C}(12,\text{I})] = \lambda_2 \, [\alpha_0 * \text{C}(12,\text{I})]^{2/3} \qquad (132) $$

where

λ = KEXT = corrected extinction coefficient

λ_0 = nonalgal portion of light extinction coefficient - user specified (LAMBD0)

λ_1 = linear algal self-shading - user specified (LAMBD1) value previously used, 0.0088 m^{-1} (μg-Chl$\underline{a}$/L)$^{-1}$ (Riley equation in Bowie et al. 1985)

λ_2 = nonlinear algal self-shading - user specified (LAMBDA2) value previously used, 0.054 m^{-1} (μg-Chl$\underline{a}$/L)$^{-2/3}$ (Riley equation in Bowie et al. 1985)

α_0 = algae chlorophyll conversion factor - user specified (ALPHA0)

C(12,I) = algal biomass concentration at node I

The same modification of KEXT is also used to modulate the growth rate of macrophytes.

Effect of nutrient
availability on algal growth

107. Algal growth is modulated by nutrient (N and P) availability using a Monod-like expression identical to that used by QUAL2E. The following equations are used

$$FN = \frac{NPOOL}{(NPOOL + KNPOOL)}$$

$$FP = \frac{PO_4}{(PO_4 + KPO_4)}$$

$$(133)$$

where

FN = nitrogen limitation factor

FP = phosphate limitation factor

$NPOOL$ = (nitrate+ammonia) concentration, g/m^3

$KNPOOL$ = half-velocity constant relating inorganic nitrogen to algal growth, g/m^3

PO_4 = phosphate concentration, g/m^3

KPO_4 = half-velocity constant relating phosphate concentration to algal growth rate, g/m^3

Algal decay

108. Algal respiration and death (i.e., decay) occur continuously and can be described as follows

$$\begin{pmatrix} \text{Corrected} \\ \text{algal} \\ \text{decay rate} \\ \text{(ALGADK)} \end{pmatrix} = \begin{pmatrix} \text{algal} \\ \text{concentration} \end{pmatrix} * \begin{pmatrix} \text{specific algal} \\ \text{decay rate} \\ \text{(ALGO)} \end{pmatrix} * \begin{pmatrix} \text{DO/decay} \\ \text{correction} \end{pmatrix} \quad (134)$$

In equation form, decay becomes

$$ALGADK = C(12,I) * ALGO * \left(\frac{DO}{DO + KOALDK} \right) \quad (135)$$

where

$ALGADK$ = algal decay rate, $g/(m^3 day)$

$ALGO$ = maximum specific algal decay rate, originally entered as KALGDK, renamed ALGO in subroutine SEG, day^{-1}

$C(12,I)$ = algae concentration at node i, g/m^3

$KOALDK$ = DO half-velocity constant for algal decay, g/m^3

Decay is coupled to DO levels by a Monod-type relationship because decay is assumed to slow down at low oxygen concentrations.

Macrophyte growth

109. The macrophyte growth rate, MGRATE $(g/m^3/day)$, is taken to be the product of the benthal macrophyte density, the local light intensity (corrected for attenuation en route), and a reaction rate coefficient. To further simplify the analysis, the stream cross section is approximated as a rectangle, as shown in Figure 7. Thus, the plant growth is distributed along the channel bottom and sides.
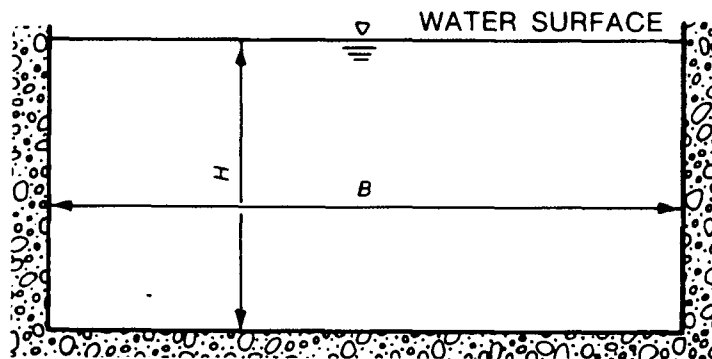
Figure 7. Channel geometry assumed for calculation of algal/macrophyte growth/decay

$$\begin{pmatrix} \text{Plant growth per} \\ \text{unit length of channel} \\ \text{(MGRATE)} \\ g/m^3/d \end{pmatrix} = \begin{pmatrix} \text{bottom} \\ \text{growth} \end{pmatrix} + \begin{pmatrix} \text{side} \\ \text{growth} \\ 2 \text{ sides} \end{pmatrix} \qquad (136)$$

or

$$MGRATE = \frac{B*MACRO*SWALG*e^{(-KEXT*H)}}{A}$$

$$+ \left(\frac{2}{KEXT*A}\right) * MACRO * SWALG * \left[1. - e^{(-KEXT*H)}\right] \qquad (137)$$

where

B = stream top width, m

MACRO $(g/watt*day)$ = MACLITE $(m^2/watt/day)$ * MACROB (g/m^2)

63

MACLITE = macrophyte growth rate coefficient, m^2/watt/day

MACROB = macrophyte surface density, g/m^2

SWALG = light intensity (net short-wave radiation) at the water surface, watt/m^2

KEXT = light extinction coefficient for the particular reach, m^{-1}

H = hydraulic depth, m

A = channel cross-sectional area, m^2

SWALG must appear explicitly in the equation because it is recomputed at each time-step. The first term on the right-hand side of Equation 137 includes the light intensity on the bottom (SWALG*EXP(-KEXT*H)). The second term is the summation of the plant growth occurring at all depths on both (vertical) sides:

$$\left(\frac{MGRATE_{sides}}{\left(\dfrac{g}{(m^3*day)}\right)}\right) = \frac{2.*MACRO*SWALG*\displaystyle\int_{0}^{H} EXP[-KEXT(H-Z)]*dZ}{A} \tag{138}$$

where

MACRO = MACLITE*MACROB

To account for diel effects, the surface light intensity (net short wave radiation) is computed as a function of the time of day. This is accomplished in subroutine HEATFLUX if the direct heat balance option is used; otherwise, the following relation is used

$$SWALG = HNEFSW*SIN\left[\frac{PI*(CLOCK-DAWN)}{LAMBDA}\right] \tag{139}$$

where

HNEFSW = maximum surface light intensity at local noon, watts/m^2

PI = 3.14159...

CLOCK = actual time of day, 24-hr clock

DAWN = time of local dawn, 24-hr clock

LAMBDA = elapsed time between local dawn and local sunset, hr

110. In one instance, numerical problems presented by the expression

$$\frac{2}{KEXT} * (1 - e^{(-KEXT*H)}) + B * e^{(-KEXT*H)} \qquad (140)$$

used in determining available photosynthetic energy (Equation 137) must be handled. First, the number of calls to the external function EXP (exponentiation) can be halved by rewriting the above expression as the algebraically equivalent

$$\frac{2}{KEXT} + \left(B - \frac{2}{KEXT}\right) * e^{(-KEXT*H)} \qquad (141)$$

The above expression, however, becomes numerically unstable as KEXT approaches zero, although mathematically it approaches the value

$$2 * H + B \qquad (142)$$

This problem is handled by taking a Taylor's series expansion of EXP(-KEXT*H) in the expression 2./KEXT*[1.-EXP(-KEXT*H)] about KEXT = 0 carried to three terms,

$$\frac{2}{KEXT} * [1. - (1. - KEXT*H + \frac{(KEXT*H)^2}{2} = \ldots .)] \qquad (143)$$

This simplifies to

$$H * (2. - KEXT*H) \qquad (144)$$

Equation 140 is then expressed as

$$H * (2. - KEXT*H) + B*e^{(-KEXT \cdot H)} \qquad\qquad (145)$$

for KEXT less than 0.01.

    111. Therefore, for KEXT less than 0.01

$$MGRATE_{KEXT \leq 0.01} = MACRO*SWALG*\frac{H*(2. - KEXT*H) + B(I)*e^{-KEXT \cdot H}}{A(I)}$$
$$\left(\frac{g}{m^3 * day}\right) \qquad\qquad (146)$$

For KEXT $\geq$ 0.01

$$MGRATE_{KEXT \geq 0.01} = MACRO*SWALG*\frac{\dfrac{2}{KEXT} + \left[B(I) - \dfrac{2}{KEXT}\right]*e^{-KEXT \cdot H}}{A(I)}$$
$$\left(\frac{g}{m^3 * day}\right) \qquad\qquad (147)$$

    112. As Beck and Young (1975) noted, streams subjected to wastewater discharges are likely to have high nutrient concentrations and are also likely to be turbid. Both these conditions are reversed in lakes, so that the lake plankton are often, if not usually, nutrient- rather than light-limited. Thus, lake models almost always include nutrients in the formula for the algal growth rate. Of course, macrophytes obtain their nutrients from the sediments (Bole and Allan 1978, Carignan and Kalff 1980), so stream nutrient levels are irrelevant to macrophyte growth.

Macrophyte decay

    113. Algae and macrophytes are presumed to respire and die (i.e. decay) continuously. The decay rate is controlled, in part by the DO level since decay slows at low DO levels. Because of the lack of reasonably precise data, no temperature correction is attempted. The decay rate is represented as

$$MDEATH = \frac{MACDKY*MACROB*\left(\dfrac{DO}{DO+KOALDK}\right)*(B+2*H)}{A}$$
$$\left(\frac{g}{m^3 * day}\right) \qquad\qquad (148)$$

66

where

MDEATH = macrophyte decay rate, $g/m^3/d$

MACDKY = macrophyte specific decay rate, day^{-1}

MACROB = macrophyte density, g/m^2

KOALDK = Monod half-velocity constant for oxygen limitation of macrophyte decay, $g\ O_2/m^3$

Oxygen

114. The sources of oxygen (O_2) are reaeration and photosynthesis; the sinks are the exertion of CBOD, nitrification, plant respiration, and the oxidation of reduced iron and manganese. The present model uses equations developed for algal synthesis and decay for both algae and macrophytes. The stoichiometry of these processes is based on the data of Foree and McCarty (1968) and depends on whether ammonium or nitrate is involved. The stoichiometry is obtained from the following equations

$$132\ CO_2 + 16\ NH_4^+ + H_3PO_4 + \frac{177}{2}\ H_2O = C_{132}H_{229}O_{58}N_{16}P + \frac{597}{4}\ O_2 + 16H^+ \qquad (149)$$

$$132\ CO_2 + 16\ HNO_3 + H_3PO_4 + \frac{209}{2}\ H_2O = C_{132}H_{228}O_{58}N_{16}P + \frac{725}{4}\ O_2 \qquad (150)$$
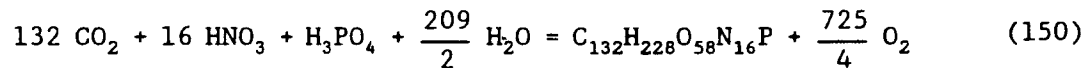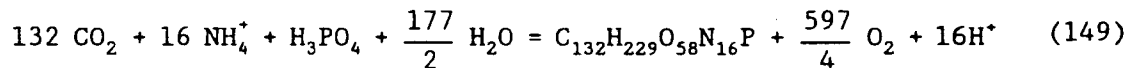
The oxygen production indicated by Equation 149 when ammonium is the nitrogen source is 1.59 g O_2/g algae, and from Equation 150 is 1.94 g O_2/g algae when nitrate is the nitrogen source. Therefore, the rate of oxygen production by photosynthesis depends upon the relative proportions of ammonia and nitrate consumed:

$$\begin{pmatrix} \text{Rate of oxygen} \\ \text{production by} \\ \text{photosynthesis} \\ g\ O_2/m^3/day \end{pmatrix} = \left[+ \text{OPDECY} + \text{ONEQUI} * \left(\frac{NO_3^--N}{(NO_3^--N + NH_3-N)} \right) \right] * (\text{ALGRO} + \text{MGRATE}) \qquad (151)$$

67

where recommended values are

OPDECY = 1.59

ONEQUI - 0.35

115. The rate of oxygen consumption resulting from algal decay does not involve nitrification, so it is the reverse of Equation 149 after accounting for the fraction that feeds directly into CBOD:

$$\left(\begin{array}{c}\text{Rate of oxygen consumption} \\ \text{by plant decay} \\ \text{g } O_2/m^3/day\end{array}\right) = OPDECY*(ALGADK + MDEATH) \qquad (152)$$

116. The rate of oxygen consumption resulting from CBOD decay is equal to the rate of CBOD decay, because of the definition of CBOD. This rate is given in Equation 110 (without CBODSR and KDN), and K1 is computed from Equation 112.

117. The rate of oxygen consumption resulting from ammonium oxidation is based on Equations 120 and 121. Since the units of the SINK term are g-N/m^3/day, a conversion factor is required. This is computed from Equation 153:

$$NH_4^+ + 2 \ O_2 = H\dot{N}O_3 + H_2O + H^+ \qquad (153)$$

The oxygen consumption here is ONITRI g O_2/g N (ONITRI = 4.57). Some authors prefer a value of 4.33 g O_2/g N, arguing that some of the ammonium consumed by the nitrifiers goes into cell synthesis (Garland 1978). However, the nitrifiers are themselves subject to predation, which returns the incorporated ammonia to the stream for further nitrification. The actual amount of ammonia removed or nitrified is unknown, but the difference between 4.57 and 4.33 is insignificant when one considers the usual accuracy of field data.

118. The rate of oxygen uptake resulting from stream reaeration can be formulated as:

$$\left(\begin{array}{c} \text{Increase in DO} \\ \text{due to reaeration} \\ \text{g } O_2/m^3/\text{day} \end{array} \right) = K2*(DOSAT-DO) \qquad (154)$$

where

K2 — reaeration rate coefficient, day^{-1}

DOSAT — local solubility of oxygen, g O_2/m^3

DO — local oxygen concentration, g O_2/m^3

Reaerations equations for K2 may be stated in a general form as

$$K2 = \frac{AG*(U^{E1})}{(H^{E2})} * \left[TPHYS^{(TEMP-20.)} \right] \qquad (155)$$

where

AG, E1, E2 — empirical coefficients

TPHYS = temperature correction coefficient for reaeration

In using Equation 155, it must be noted that AG is sometimes given for 25 °C. For those who prefer the Tsivoglou-Wallace (1972) equation in its published form, provision has been made in the program to permit its use as an option where:

$$K2 = \frac{TSIV*[E(I-1)-E(I)]}{(DX/U)} * TPHYS^{(TEMP-20)} \qquad (156)$$

and where

TSIV — empirical coefficient, m^{-1} (the value is entered in units of ft^{-1} and converted, suggested value is 0.054)

E(I-1), E(I) — water surface elevations at the upstream and downstream ends of the reach under consideration, m

DX — length of the reach, m

U — reach stream velocity, m/day

In using Equation 156, it must be noted that TSIV is sometimes given for 25 °C. As discussed in Part V, the model can also account for structural and wind-driven reaeration.

119. The solubility of oxygen in water is calculated using the following formula (Elmore and Hayes 1960):

$$DOSAT = 14.652 + [-0.41022 + (0.007991 - 0.000077774*TEMP)*TEMP] \quad (157)$$

120. The oxidation of reduced iron (FE) and manganese (Mn) is considered to occur as a first-order reaction. Accounting for stoichiometric conversions, the oxygen used during oxidation of reduced iron and manganese is

$$\begin{pmatrix} \text{Rate of oxygen used} \\ \text{for oxidation of Fe} \\ \text{and Mn, g } O_2/m^3/day \end{pmatrix} = -OFEDEC*KFEDK*FE - OMNDEC*KMNDK*MN \quad (158)$$

where

OFEDEC = oxygen-to-iron ratio for iron oxidation

KFEDK = oxidation rate for iron, day^{-1}

FE = concentration of reduced iron, g/m^3

OMNDEC = oxygen-to-manganese ratio for manganese oxidation

KMNDK = oxidation rate for manganese, day^{-1}

MN = concentration of reduced manganese, g/m^3

121. A DO loss for sediment oxygen demand (SOD) is also provided as a zero-order sink term. SOD is input by the user as KSOD, $g/m^2/day$.

122. The complete balance of DO reactions is

$$\begin{pmatrix} \text{Net rate of accumulation} \\ \text{of dissolved oxygen,} \\ g/m^3/d \end{pmatrix} = \begin{array}{l} \text{(Reaeration) - (CBOD oxidation)} \\ \\ \text{- (nitirfication)} + \begin{pmatrix} \text{DO production} \\ \text{from algae/macrophytes} \end{pmatrix} \quad (159) \\ \\ - \begin{pmatrix} \text{DO used in algal/} \\ \text{macrophyte respiration} \end{pmatrix} - \text{(Fe oxidation)} \\ \\ \text{(Mn oxidation) - (SOD)} \end{array}$$

This is stated in equation form as

$$\begin{pmatrix} \text{Rate of accumulation} \\ \text{of dissolved oxygen} \\ \text{g } O_2/m^3/\text{day} \end{pmatrix} = K2*(DOSAT - DO) - K1*CBOD - ONITRI*KN*NH_4N$$

$$+ OPDECY + \left[ONEQUI * \left(\frac{NO_3^--N}{(NO_3^--N + NH_4^+-N)} \right) \right] *(ALGRO + MGRATE) \qquad (160)$$

$$- OPDECY*(1 - FCBOD)*(ALGADK + MDEATH) - OFEDEC*KFEDK*FE$$

$$- OMNDEC$$

Phosphorus Interactions

123. Hydrolysis of organic-P is the source of phosphate while sinks are sorption to the sediments (Taylor and Kunishi 1971) and plant uptake of phosphate. These last two mechanisms appear to be competitive, and phosphate sorbed to sediments may be unavailable to plants (Fitzgerald 1970). Additionally, organic P can be lost through settling and gained through plant decay.

124. The physicochemical sorption occurs only on aerobic sediments, and phosphate is released from anaerobic conditions (Gummerman 1970, Olsen 1964). This latter possibility is ignored in the present model under the assumption that anaerobic conditions are uncommon in streams. As much as half the phosphate present in streams subjected to detergent phosphate discharges may be polymerized as either tripolyphosphate or pyrophosphate (Engelbrecht and Morgan 1959). These different forms are believed to be sorbed and consumed at the same rates, so they have not been distinguished in the present model. In this model macrophytes are assumed to take up phosphate from the water column. In reality, most macrophytes take nutrients from sediments. However, epiphytes attached to macrophytes can strip nutrients from the water column.

*KMNDK*MN - KSOD

125. The phosphate sorption process is represented as a simple first-order decay:

$$\begin{pmatrix} \text{Rate of loss of} \\ \text{phosphate through} \\ \text{sediment sorption} \\ \text{g-P/m}^3/\text{day} \end{pmatrix} = -KPO4DK*PO4 \qquad (161)$$

71

where

KPO4DK = first-order reaction rate coefficient for sorption, day^{-1}

PO4 = local phosphate concentration, g P/m^3

The rate coefficient is corrected for the appropriate temperature from the following expression.

$$KPO4DK = APO4*\left[TPHYS^{(TEMP-20.)}\right] \tag{162}$$

where

APO4 = uncorrected rate coefficient for phosphate sorption, day^{-1}

126. The algal/macrophyte sources and sinks of P are evaluated using Equations 149 and 150. These equations indicate that algae are about 1.0 percent P by weight. Therefore, the plant source and sink terms for P are:

$$\begin{pmatrix} \text{Release of org-P by} \\ \text{algal/macrophyte decay} \\ g\text{-}P/m^3/day \end{pmatrix} = APCONT*(ALGADK + MDEATH) \tag{163}$$

where

APCONT = phosphorus-to-biomass ratio in algae and macrophytes, approximately 0.01

The loss of organic P due to hydrolysis and settling is stated as

$$\begin{pmatrix} \text{Loss of org-P due to} \\ \text{hydrolysis to phosphate} \\ \text{and settling} \\ g\text{-}P/m^3/day \end{pmatrix} = (ORG\text{-}P) * \left(KPDK + \frac{KPSET}{H}\right) \tag{164}$$

where

ORG-P = organic-P concentration, C(6,I), g/m^3

KPDK = org-P hydrolysis rate, day^{-1}

KPSET = org-P settling rate, m/day

$$
\begin{pmatrix}
\text{Uptake of phosphate-P by} \\
\text{algal/macrophyte growth} \\
\text{g-P/m}^3\text{/day}
\end{pmatrix} = \text{APCONT}*(\text{ALGRO} + \text{MGRATE}) \tag{165}
$$

The net rate of accumulation of organic-P is stated as

$$
\begin{pmatrix}
\text{Net rate of accumulation} \\
\text{of org-P} \\
\text{g/m}^3\text{/day}
\end{pmatrix} = -\begin{pmatrix}
\text{hydrolysis to phosphate} \\
\text{and settling} \\
\text{g/m}^3\text{/day}
\end{pmatrix} + (\text{plant decay}) \tag{166}
$$

or

$$
\begin{pmatrix}
\text{Net rate of accumulation of} \\
\text{org-P per unit volume,} \\
\text{g P/m}^3\text{/day}
\end{pmatrix} = \begin{array}{l} - \text{ORG-P}*(\text{KPDK} + \text{KPSET}) \\ \\ + 0.01*(\text{MDEATH} + \text{ALGADK}) \end{array} \tag{167}
$$

The net rate of accumulation of phosphate-P is stated as

$$
\begin{pmatrix}
\text{Net rate of accumulation} \\
\text{of phosphate-P} \\
\text{g-P/m}^3\text{/day}
\end{pmatrix} = \begin{pmatrix}
\text{hydrolysis of} \\
\text{.organic-P}
\end{pmatrix} - \begin{pmatrix}
\text{sediment} \\
\text{sorption}
\end{pmatrix} \\ - \begin{pmatrix}
\text{uptake by} \\
\text{algae/macrophytes}
\end{pmatrix} \tag{168}
$$

or

$$
\begin{pmatrix}
\text{Net rate of accumulation} \\
\text{of phosphate-P} \\
\text{g-P/m}^3\text{/day}
\end{pmatrix} = \begin{array}{l} + \text{ORG-P}*\text{KPDK} - \text{KPO4DK}*\text{PO}_4 \\ \\ - \text{APCONT}*(\text{ALGRO} + \text{MGRATE}) \end{array} \tag{169}
$$

Iron and Manganese

127. As modeled in CE-QUAL-RIV1, iron and manganese have no autochthonous sources. They can enter the system only in reduced form from upstream boundaries, tributaries, and lateral inflows. Oxygen depletion resulting from

73

the oxidation of these reduced metals may have an adverse impact on water quality downstream from the inflow source. Since the presence of reduced metals probably accompanies releases of waters already somewhat depleted of DO, metal oxidation may exacerbate existing conditions.

128. Oxidation of reduced iron or manganese (loss from the system) is treated simply as a first-order process.

$$
\begin{pmatrix} \text{Loss of Mn} \\ \text{due to oxidation} \\ \text{g/m}^3\text{/day} \end{pmatrix} = -KMNDK*MN
$$

$$
\begin{pmatrix} \text{Loss of Fe} \\ \text{due to oxidation} \\ \text{g/m}^3\text{/day} \end{pmatrix} = -KFEDK*FE
$$

(170)

Oxidation does not occur if DO is less than the user-specified value of OXIDAT. Use of Equation 170 may require field data to estimate oxidation (i.e., decay) rates.

129. Temperature does not affect the oxidation rates in the model. Such rate corrections are unnecessary since the current model formulation does not account for other, possibly more important effects such as pH and auto-catalysis. Future code improvements may allow implementation of a more mechanistic approach, accounting for variations in process rates, such as those presented by Dortch et al. (1992).

Coliform Bacteria and Other Conservative and Nonconservative Constituents

130. Fecal coliform bacteria, which are a preferred indicator for total coliform bacteria, enter the system only through inflows that represent agricultural or urban flows. Fecal coliforms do not reproduce in natural aquatic environments, and their populations decay exponentially; thus

$$
\begin{pmatrix} \text{Loss of fecal coliforms,} \\ \text{colonies/100 ml/day} \end{pmatrix} = -KCOLIDK*COLI*TBIOS^{(TEMP-20.)}
$$

(171)

where

> KCOLIDK = rate coefficient for fecal coliform die-off, day^{-1}
>
> COLI = fecal coliform count, colonies/100 ml

131. The coliform bacteria variable, C(11,I) , can be used for modeling other miscellaneous nonconservative or conservative constituents since it does not interact with other water quality variables. Equation 171 does not contribute to the SINKS term of the transport equation (Equation 76), and KCOLIDK is the K_s term of Equation 76. Therefore, the first-order loss of other nonconservative constituents could be represented by Equation 171 and the variable C(11,I) . Likewise, a conservative constituent could be modeled by setting KCOLIDK = 0.0 .

132.   As has been noted earlier, the model equations for constituent
transport are dependent upon hydraulic variables, but not vice versa. Because
of this uncoupling, two entirely separate computer programs can be written.
The first program, called RIV1H, solves for the time and space distribution of
flow, cross-section area, top width, and depth. When completed, these calcu-
lations are stored for use in RIV1Q, the water quality model. RIV1Q is struc-
tured so that for any one set of hydraulic data, a variety of water quality
simulations can be performed. This part describes in detail the structure and
operation of RIV1H.

Overview

133.   The hydrodynamic program contains numerous subroutines in addition
to the main program. Also, a separate file linked to the main program and
subroutines, using an INCLUDE statement, allows flexible dimensioning through
use of a PARAMETER statement. Furthermore, the INCLUDE file declares the
units used for the various input and output files in the program and contains
COMMON blocks for some key variables.

134.   CE-QUAL-RIV1 expects the user to create an execution control file
(RIV1H.CTL) prior to running RIV1H. This control file provides the names of
the input and auxiliary files that are required to successfully run the model.
The control file must be in the same directory as the RIV1H executable file
and as such consist of standard ASCII text. The syntax of the control file is
simple. The descriptions on the left are used as place holders so that the
user knows which line of the control file is used for which input file. The
descriptions can be 15 characters long with the name of the input file start-
ing in column 16. The first line of RIV1H.CTL is the model input data file,
the second line the lateral inflow file, and the third the cross-section
lookup table file name. If there is no cross-section lookup table file, then
insert a blank line for record three of the control file. The order in which
this information is entered cannot be changed. A listing of RIV1H.CTL is
given below.

```
INPUT FILE      SAMPLEH.INP
LATERAL INFLOW  SAMPLEH.LAQ
XSECT TABLR     XSECT.XSF
```

135. Cross-sectional shapes may be described in RIV1H by using either geometric equations or tabular, surveyed cross sections. If the option for surveyed cross sections is selected in the main input file, then the filename is given in the control file. Subroutines AFROMH, AFROMH2, HFROMA, READXS, SHELL, and XSECT are then called to compute cross-section flow properties from surveyed cross-section input; the purpose of each is described later in this section.

136. Nonpoint source flows may be either constant or time-varying in RIV1H. If the user elects to provide time-varying lateral inflows, then the name of the lateral inflow file must be in the control file. The time-varying lateral input file will be read, and flows updated, in the subroutine TIME_VARYING_DATA.

137. The main program then sorts through the tributary network, calling subroutine BUBBLE, and determines what types of boundary conditions are to be found. It sets up the main time march in which values for all nodes are solved simultaneously while time-steps are handled successively. At each time-step, boundary conditions are assigned (in subroutine TIME_VARYING_DATA), and subroutine CALC is called for each segment in a downstream order (from lowest order streams to highest order). Within each time-step the governing equations are solved iteratively; the first iteration is completed by calling subroutine NEW for each segment in an upstream order. Subsequent iterations begin with calls to ITER, an entry point within CALC, and again are completed by calls to NEW. The iteration continues until values of the controlling variables, flow and area, have converged to within tolerance or up to a limit of 50 iterations. Subroutine CALC contains the governing equations — conservation of mass and momentum. Using the residuals and partial derivatives from these equations, subroutine CALC sets up a multidimensional Newton-Raphson iteration in matrix format. The matrix equation is reduced to two vectors and solved completely by subroutine MAT5. Subroutine NEW updates the values of flow, area, and other hydrodynamic variables. In the case of tributaries, it also completes the solution of the matrix equation.

Structure of RIV1H

138. This section is a guided tour; that is, topics are discussed in the order they appear in the listing of the code, which is found in Appendix A. Certain topics, such as the tributary structure, boundary conditions,

cross-section formulas, and input and output are mentioned briefly in this section but are covered much more fully in their individual sections. The meaning and use of all the variable names are given in Appendix C, but it should be noted that in order to conserve storage, several arrays are reused in different ways in different sections of the program. The flowchart for RIV1H appears in Figure 8.

Main program

139. The main program opens the control file and determines the names of the various input files. The main program then proceeds to read the title card and then the grid card (see Input and Output) in which the user indicates the number of nodes, beginning and ending dates for the simulation, and segments in the system. For the purposes of this program, a segment is defined as a stretch of stream, each boundary of which is either a system boundary, a receiving stream, or a control structure (see Figure 9).

140. The beginning and ending dates, as with all time-varying data, are provided by year, month, day, and hour (or fraction of an hour) and are converted to internal Julian day units by the subroutine JULIAN. Output is reconverted to the year, month, day, hour format by the subroutine GREGORIAN. Next, the MAIN program reads the constants GR, THETA, TOLER, and BETA. The default value of acceleration caused by gravity, GR, is set at 32.174 ft/sec^2 under the assumption that the units used in modeling are the customary English units. If the user wishes to use SI units, then a value of 9.80 m/sec^2 should be input (see Input and Output). In this case, Manning's coefficients should be adjusted by a factor of 1.49 to account for the SI form of the Manning's equation. However, the water quality program does not accept input data with SI units, although it converts to SI units.

141. The weighting factor, THETA, is discussed in Part III. A default value of 0.55 is cited in the literature as optimal for model accuracy; however, a higher value (i.e., 0.6 to 0.75) is often used to enhance stability. The relative error tolerance, TOLER, is assigned a default value of 0.001. Iteration ceases when all residuals (i.e., differences in successive iterations) for flow and area are less than TOLER times the root mean square of all flows or areas in the system. Experience suggests a larger value (i.e., 0.1) can reduce run time without substantially sacrificing accuracy. The momentum correction factor, BETA, is given the default value of 1.0, that is, no correction.

78
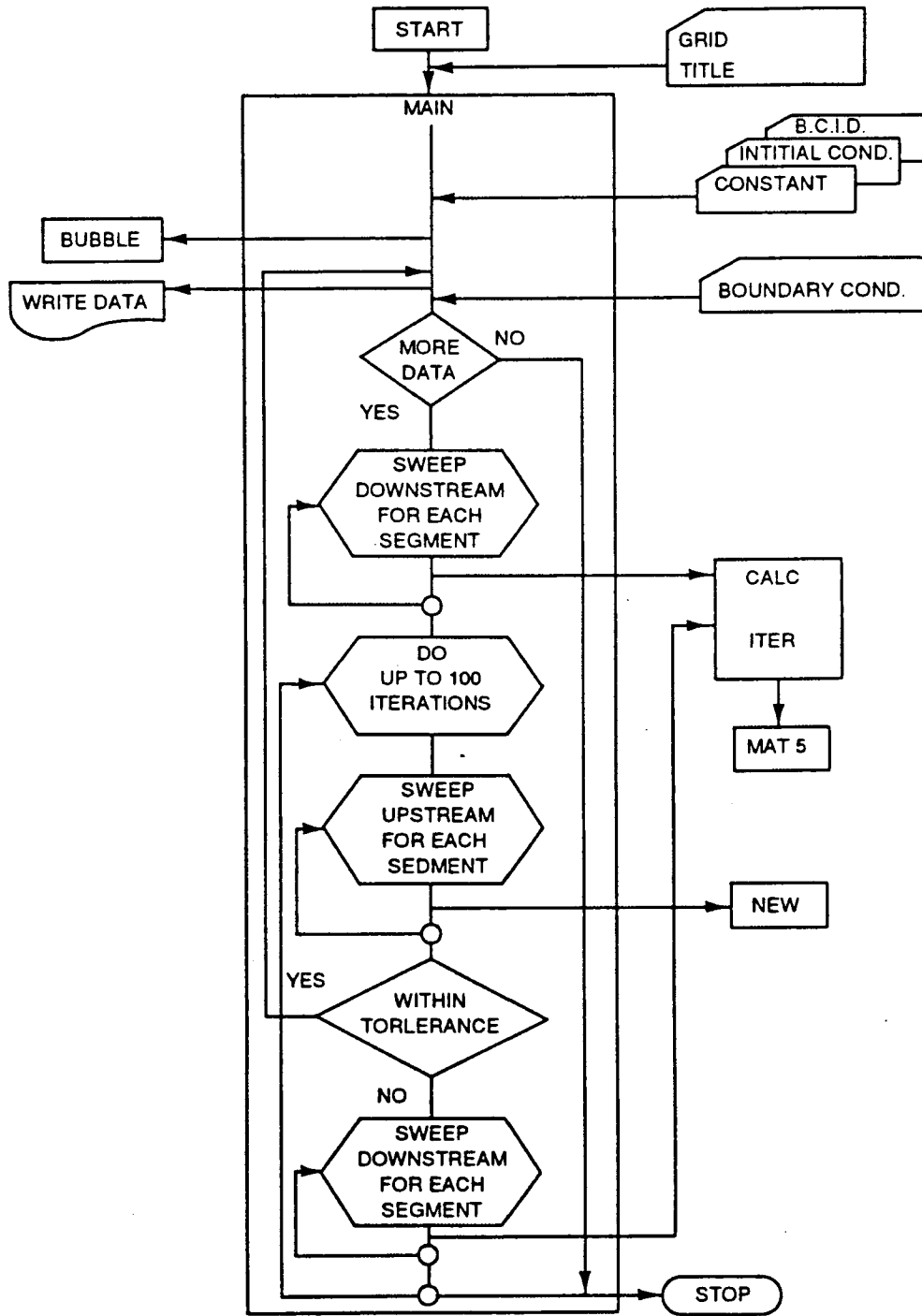
Figure 8. RIVlH organizational flowchart

SEGMENT NUMBER AND ARRAY POSITION	NNODE	FEEDS	JNODE	COSP	ORDER	IT0	IT1	IT2
1	18	3	10	60	7	2	0	0
2	23	3	17	60	3	3	0	0
3	26	7	0		6	0	2	4
4	26	3	21	60	1	4	6	6
5	13	4	17	30	2	6	0	0
6	25	7	8	60	4	1	5	5
7	16	0	0		8	0	1	1
8	9	6	10	120	5	5	0	0
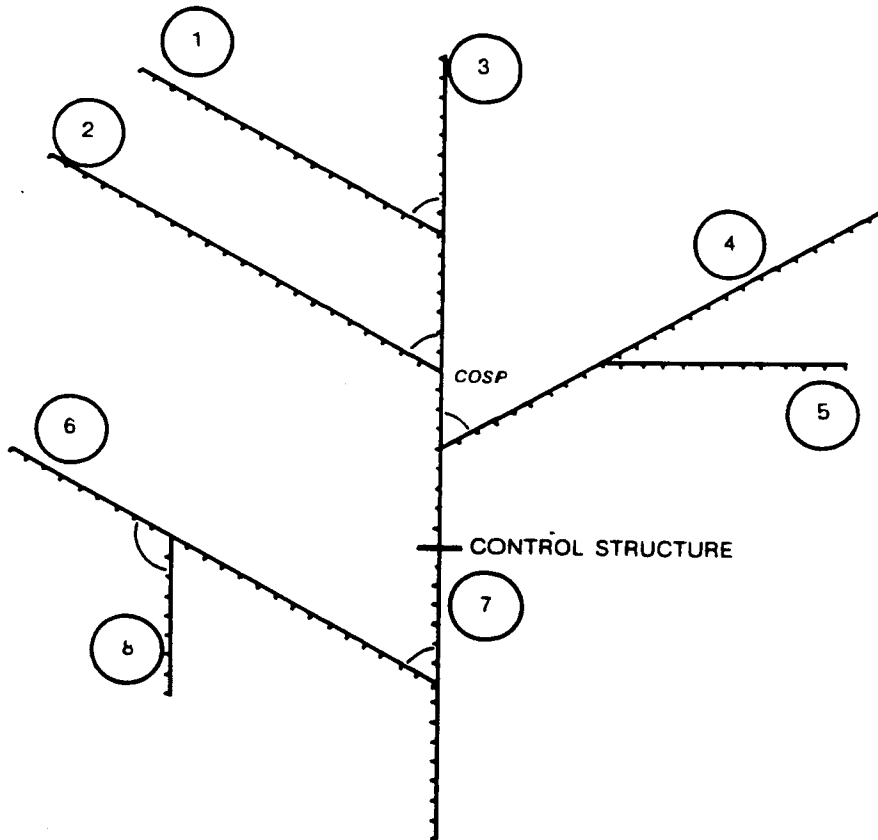
Figure 9. Sample network

142. The program then proceeds to inputting print intervals to output files. For each segment, the segment card is read and parsed; then initial conditions cards are read, one for each node. Following these is the boundary conditions identification card, in which the user informs the program which boundary conditions are being supplied in the input stream. Time variable

information is then specified, including the model time-steps and names of files containing the time-varying boundary condition data.

143. The user refers to each segment by its identification number, but the program refers to each segment by its order of appearance in the input. All references by identification number are therefore converted to references by array index, using the array ID as a dictionary. In the same way, it is convenient for the user to know to which segment each boundary condition belongs, and the program stores this information in the array IBC. For the program, however, it is more convenient to know at which array location a particular boundary condition is to be found. The appropriate indices are determined and stored in the arrays JBCU and JBCD, for upstream and downstream, respectively. Next, the number of boundary condition updates, the update intervals, and the boundary conditions are read.

144. As mentioned previously, the iteration process consists of one downstream and one upstream sweep through the system. The next block of the program determines the proper ordering of the segments for the two sweeps and establishes the packing of arrays through which tributaries and receiving streams will pass information. The water quality model requires that data for all tributaries to any given segment be packed in a downstream order. This sorting is done by subroutine BUBBLE. Also in the program section, boundary conditions that are generated within the stream network, as opposed to being given in the input, are reset in the arrays JBCU and JBCD.

145. For the user, the controlling variables are flow and stage, but the program works predominantly with flow and area. The next section calculates area and channel top-width from the initial stage and the cross section given for each (see Cross-Section Formulas).

146. River miles are calculated for the system nodes as follows. Proceeding upstream through the network, if a segment is the mainstream of the system, the river mile of the last node is set to RMILEO (which is obtained from the input). If it is a tributary, the river mile of the last node is zero. If it terminates at a control structure, then it is considered to be part of the same stream as the segment downstream, and the river mile of the last node is equal to that of the latter's first node. Once the river mile of the most downstream node is established, the river miles of the rest of the nodes of the segment are calculated by successive addition of distance increments, indexing upstream.

147. The acceleration caused by gravity is divided by two in most places that it appears in the governing equations, so to streamline the program, this division is carried out once and for all before the time march begins.

148. A purely relative tolerance test risks a zero divide error in the case of a flow reversal where at some instant flow may go to zero. For this reason and others, the difference between successive estimates of the dependent variables flow and area is compared not with the values themselves but with the root-mean-square values for all nodes, one for flow and the other for area, times TOLER.

149. The input title, constants, and initial conditions are copied onto output. Grid parameters, network organization variables, distance increments, lateral inflows, and river miles are written onto disk storage for use by the water quality model.

150. The time march begins by initializing the variables for tolerance testing. Data from the previous time-step are then written (in the case of the first time-step, this refers to initial conditions) to the output file and to a file for graphical post-processing. At the top of the page of the output file comes the program title. The run title, supplied by the user, is written below. Next the elapsed time, which has units of seconds for computation, is written out in year, month, day, and hours; next to this appear the time-step number, the segment identification number, and the segment name. Headings are written for river mile, flow, area, width, stage, and water surface elevation. Data appear in columns below the headings. Information is also provided whenever time-varying data (time-steps, boundary conditions, or lateral inflows) are updated, including the date, value, and next update interval. Flow, area, width, lateral inflow, and water surface elevation are written onto disk storage. After incrementing the elapsed time, the calculation for the next time-step proceeds.

151. One complete iteration of the Newton-Raphson method consists of a downstream sweep through the system, an upstream sweep, and a convergence test. For the first iteration within a time-step, the downstream sweep consists of updating all time-varying boundary conditions and calling subroutine CALC for each segment. Subsequent iterations require updating only those boundary conditions that are internally generated (i.e., rating curves and water surface elevation at tributary junctions). In addition, certain

calculations within CALC are bypassed by entering the subroutine at entry ITER. The upstream sweep in both cases calls subroutine NEW for each segment. Fifty iterations are permitted. If results have not converged to within tolerance by then, the program stops, writes an error message, and puts out a condition code of 7. The condition code (which has a value of zero if instead the run is successful) can serve as a signal to other job steps not to run.

Subroutine CALC

152. Subroutine CALC contains the governing equations of the hydrodynamic model--conservation of mass and conservation of momentum. It sets up these equations as the sum of terms having to do with the previous time-step and terms having to do with the current time-step. Terms having to do with the previous time-step need be calculated only once per time-step. Therefore, all iterations past the first enter subroutine CALC at entry ITER, after the calculation of previous time-step terms. The terms are lumped into the array XC for the continuity or conservation of mass equation and XM for the conservation of momentum equation. The final entries of the arrays XC and XM are reserved for the contribution of the segment to the continuity and momentum equations, respectively, of its receiving stream (see Tributaries).

153. To enhance execution, repeatedly used array quantities are assigned to scalar variables; for instance, when dealing with the reach between nodes I and I+1 , the value of Q(I) is assigned to Q0 , the value of Q(I+1) is assigned to Q1 , and DX takes on the value DX1(I) .

154. The governing equations are written in the program as they appear in Part III with two exceptions: the off-channel storage area AO is ignored and the slope SO is combined with the pressure gradient term and appears in the equation as

$$DE = EL(I+1) - EL(I) \qquad (172)$$

where EL is the water surface elevation.

155. The upstream boundary condition (see Boundary Conditions) can be either a flow or a stage. If a stage is used, it must be converted to the cross-sectional area by the appropriate formula (see Cross-Section Formulas). The set of boundary conditions employed is indicated by the value of LIB (see Boundary Conditions). The partial derivative array AA and the residual

vector R are packed accordingly. Downstream boundary conditions can be flow, stage, or a rating curve and are handled in the same way.

156. The governing equations are given next, but a little preparatory work is necessary first. In calculating the partial derivatives of the governing equations with respect to A_i and A_{i+1} (cross-sectional areas of nodes I and I+1 , respectively), values for dB_i/dA_i and dB_{i+1}/dA_{i+1} are required. Applying the chain rule:

$$\frac{dB}{dA} = \frac{dB}{dH} \cdot \frac{dH}{dA} = \frac{dB}{dH} \cdot \frac{1}{B} \tag{173}$$

It remains to calculate dB_i/dH_i and dB_{i+1}/dH_{i+1} . Now as calculation proceeds from node to node, dB_{i+1}/dH_{i+1} is updated as the variable DBDH . It takes on its new value in between the calculation of the partial of the momentum equation with respect to A_i and the partial with respect to A_{i+1} . This leaves the values of dA_1/dH_1 , which must then be calculated, outside of the loop.

157. Proceeding in order then, the reach specific quantities are set. The residual of the continuity equation is calculated. The residual of the momentum is calculated. Calling the continuity equation F and the momentum equation G , $\partial F/\partial Q_i$, $\partial F/\partial A_i$, $\partial F/\partial Q_{i+1}$, $\partial F/\partial A_{i+1}$, $\partial G/\partial Q_i$, $\partial G/\partial A_i$, and $\partial G/\partial Q_{i+1}$ are calculated. A new value is calculated for DBDH to permit calculation of $\partial G/\partial A_{i+1}$. The matrix of partial derivatives AA is packed appropriately (see Boundary Conditions).

158. The residuals and partial derivatives are adjusted to account for the effects of any tributaries entering this segment. Next, allowance is made for the fact that the matrix solver requires nonzero entries in the upper left-hand and lower right-hand corners of the coefficient matrix AA. Under certain types of boundary conditions, i.e. those involving stage at the upstream end or flow at the downstream end, this assumption would be invalid. To compensate, the program interchanges columns in the coefficient matrix prior to calling the matrix solver, subroutine MAT5, and reverses the interchange in the solution vector upon return.

159. Finally, before returning to the main program, if this segment is a tributary, it loads data into the tributary information passing array T.

<u>Subroutine MAT5</u>

160. Subroutine MAT5 contains an algorithm for solving the five-banded coefficient matrix AA constructed in subroutine CALC. It takes advantage of several facts about this matrix to increase efficiency. The matrix's structure is shown in Equation 190. The value of unity in the upper left-hand corner is implied, and the extreme diagonals are only half-populated. By pairwise row operations, these extreme diagonals are eliminated, leaving a matrix of tridiagonal structure. This is readily solved by a recursive Gaussian elimination (Carnahan, Luther, and Wilkes 1969) known as the Thomas algorithm. First, the lower codiagonal is eliminated proceeding from the top of the matrix down. Proceeding from the bottom up, the upper codiagonal is eliminated, and each row is normalized to leave the identity matrix on the left and the solution vector on the right. In the case of a tributary, a companion vector C is generated in this last step (see Tributaries). Normally, the lower right-hand element is assumed to be unity, and the element to its left is assumed to be zero. These assumptions would not be the case for a segment whose downstream boundary is a rating curve. Such a segment must have its last row handled in a special way, which produces modified values for the solution vector (but not the companion vector C as such a segment cannot be a tributary).

<u>Subroutine NEW</u>

161. The output for subroutine MAT5 is vector R, which now contains adjustments to be made in flow and area to arrive at the new estimates. First, however, the vector R may have to be adjusted to account for effects of the receiving stream if this segment is a tributary. Once this is done, alternate values of R are added alternatively to flow and area. From the new area estimates, new stage and width estimates can be made. Now the cross-section formulas explicitly yield area and width, given stage. There is no explicit formulation, however, that will yield width and stage given area. The answer is, therefore, arrived at using another Newton iterative procedure. This is of the form

$$H_{new} = H_{old} - \frac{f(H_{old}) - A}{f'(H_{old})}$$ (174)

where

H_new - new estimate of stage

H_{new} = new estimate of stage

H_{old} = previous estimate of stage

$f(H)$ = cross-section formula for area as a function of stage

A = new cross-sectional area

$f'(H)$ = derivative of the cross-section formula for area as a function of stage with respect to stage

The variable $f'(H)$ is equivalent to the cross-section formula for width as a function of stage, and so it is called BO in the program, that is, a provisional estimate of width just as HO is a provisional estimate of stage.

162. The test of satisfactory convergence is the relative difference between successive estimates, where the tolerance is fixed at 0.01. Failure to converge in 10 iterations causes the program to terminate, print an error message, and put out a condition code of 7.

Subroutine BUBBLE

163. As mentioned previously, the water quality model requires the tributary-data-passing arrays JT and T to be packed such that all tributaries entering a given segment can be accessed in downstream order. The subroutine scans the list of segments to see if any has more than one tributary. If one is found, pointers to the tributaries and entries in the array JT are sorted in a downstream order according to the BUBBLE algorithm (Knuth 1973).

Cross-Section Formulas

164. It is possible to have a program of this sort work with explicit cross-section representations; that is, the coordinates measured in the field could be used directly to relate area, width, and depth. As short-hand cross-section descriptions, however, RIV1H has two stage-area and stage-width equations (stage is height of the water surface above the channel bottom): a hybrid power function and an ellipsoid function.

165. The hybrid power function (standard formula) is

$$A = C_1 H + C_2 H^{C_3} \tag{175}$$

Consequently, since B is equal to dA/dH

$$B = C_1 + C_2 C_3 H^{C_3 - 1} \qquad\qquad (176)$$

where

    A — area

    H — stage

    B — channel top width

These formulas can describe a variety of standard shapes. With $C_2 = 0$, they describe a rectangle of width C_1. With $C_1 = 0$ and $C_3 = 2$, they describe a triangle with a height-to-width ratio of $\dfrac{1}{2C_2}$. With $C_3 = 2$ and $C_2 = \dfrac{1}{2}\left[\dfrac{1}{Y} + \dfrac{1}{Z}\right]$, they describe a trapezoid of bottom-width C_1 and side-slopes of Y and Z. With $C = 0$, $C_2 = \dfrac{4}{3}\alpha^{1/2}$, and $C_3 = \dfrac{3}{2}$, they describe a parabolic cross section of the form

$$H = \frac{1}{\alpha}\left(\frac{B}{2}\right)^2 \qquad\qquad (177)$$

where α is the distance from the vertex to the focus. Other shapes can be approximated by curve fitting.

    166. In cases where an ellipse would give a better fit (for instance, flow in a partially full conduit), the user has the option of so describing such cross sections. The geometric descriptions at each node are independent; for instance, ellipses may be interspersed with standard descriptions. In designating the ellipse, C_1 is half the vertical axis length, C_2 is half the horizontal axis length, and C_3 is set to -1 to indicate to the program that an ellipsoid description is intended. If $C_1 = C_2$, of course, the cross section is circular. The formulas are

$$B = 2\,\frac{C_2}{C_1}\sqrt{2C_1 H - 2} \qquad\qquad (178)$$

and

$$A = C_1 C_2 \arccos\left(1 - H/C_1\right) - B\left(C_1 - H\right)/2.\tag{179}$$

All formulas are summarized in Figure 10.

CROSS-SECTIONS FALL INTO TWO BASIC TYPES:

1. THE STANDARD FORMULA:

$$A = C_1 H + C_2 H^{C_3} , \quad B = C_1 + C_2 C_3 H^{C_3 - 1}$$

IN THIS CATEGORY FALLS

RECTANGULAR,
$C_1 = W$
$C_1 = O$
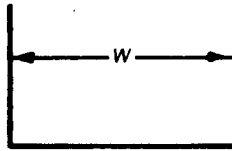
TRIANGULAR,
$C_1 = O$
$C_2 = \dfrac{1}{2}\left(\dfrac{1}{z} + \dfrac{1}{y}\right)$
$C_3 = 2$
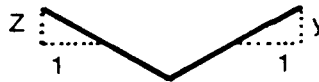
TRAPEZOIDAL,
$C_1 = B_O$
$C_2 = \dfrac{1}{2}\left(\dfrac{1}{z} + \dfrac{1}{y}\right)$
$C_3 = 2$
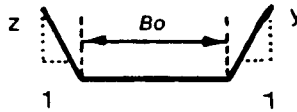
AND PARABOLIC,
$C_1 = O$
$C_2 = \dfrac{4}{3}\,a^{1/2}$
$C_3 = \dfrac{3}{2}$

$$H = \frac{1}{a}\left(\frac{B}{2}\right)^2$$
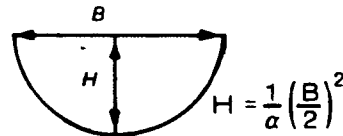
OTHER SHAPES CAN BE APPROXIMATED BY CURVE-FITTING

2. ELLIPSOID

$$B = 2C_2/C_1 \sqrt{2C_1 H - H^2}$$

$$A = C_1 C_2 ARCCOS(1 - H/C_1) - B(C_1 - H)/2$$
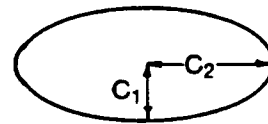
Figure 10. Cross-section types

167. In the course of the program, several manipulations of the cross-section formulas are required. These formulas (Equations 178 and 179) are found in the main program in the initial calculations of areas and widths, again in subroutine CALC in performing the same calculations for upstream and downstream boundary conditions, and in subroutine NEW where the stages are

88

updated. Later in subroutine CALC, when calculating the partial derivatives of the momentum equation with respect to area, a value is required for dB/dA . This is computed using Equation 176; it is equal to $\frac{1}{B}\frac{dB}{dH}$. The derivative $\frac{dB}{dH}$ is called DBDH in the program and for the standard formula is equal to

$$C_2C_3 \left(C_3 - 1 \right) H^{(C_3-2)} \tag{180}$$

and for the ellipsoid formula

$$4\left(\frac{C_2}{C_1}\right)^2 \left(\frac{C_1 - H}{B}\right) \tag{181}$$

168. In most computer systems, algorithms for taking integral powers are much less involved than those for taking nonintegral powers. RIV1H takes advantage of this by determining for each cross section whether the exponent C3 is integral. If it is, the integer array I3 receives the value of C3 , and C3 is set to zero, which serves as a signal to the rest of the program to use I3 in cross-section calculations rather than C3 .

Surveyed Cross Sections

169. RIV1 can now develop cross-sections shapes from survey data in two ways. The simplest method develops a relationship for flow, area, and depth from a set of X,Y pairs. A second method allows for development of the necessary cross-section relationships by "blending" information from the upstream and downstream cross sections (parent cross sections). In cases where only one parent cross section exists, the computed cross section is an exact copy of the parent. If there are no parent cross sections, an error results.

170. RIV1 reads in the cross-section coordinate list and uses it to generate a table of area versus height and width. The algorithm used allows cross sections to be bumpy but they cannot fold in on themselves or have over-hangs. Before proceeding, the program inspects the data to make sure this is the case. Once each section of the table is completed, the routines match the

cross-section ID with the corresponding node. Determining H from A, or vice versa, then consists of table lookup and interpolation.

171. Each surveyed cross section maintains a pointer (IXSA) in the table. Each blended cross section maintains two pointers, one upstream (IXSA) and one downstream (IXSB). As the water level rises or falls in each cross section, these pointers indicate the last table entry used for interpolation. Each section of the table is bounded above and below by zero so that the routine can tell if the river has run dry or overflowed its banks.

172. To determine A from H for a surveyed cross section, the user must first find the appropriate position in the lookup table. Then calculate the width B by linear interpolation, and use this B, together with the tabulated A, to compute the area of the cross section corresponding to H. To determine H given A again, the user begins by finding the correct position in the table. Then solve the relation

$$A = A_o + B_o \cdot (H-H_o) + \frac{1}{2} \cdot \frac{dB}{dH} \cdot (H-H_o)^2 \qquad (182)$$

where the subscript o refers to the table value for H. To determine H from A for blended cross sections, perform the above routine for each parent and average the results.

173. Determining A from H for a blended cross section is the most complicated and cumbersome task but it only has to be done at model start-up. Once the model is in operation, it only calculates H from A since A and Q are the primary variables. An exception to this is boundary conditions, but a boundary condition can never be a blended cross section. The process begins as before by (provisionally) locating the correct table position for each parent and solving the set of equations for H. The subscripts a and b refer to the upstream and downstream parent, respectively, and ξ is the weighting factor for the downstream parent. The resulting H may lie outside the range of the table entry, and if this is the case, the table pointer would be adjusted and the process repeated.

$$A = A_{o.} + \left(B_{oa} + \frac{dB}{dH_{oa}}\right) \cdot \frac{H_a - H_{oa}}{2} \cdot (H_a - H_{oa})$$

$$A = A_{ob} + \left(B_{ob} + \frac{dB}{dH_{ob}}\right) \cdot (H_b - H_{ob}) \qquad (183)$$

$$H = (1-\xi) \cdot H_a + \xi \cdot H_b$$

Subroutines

174. The following subroutines are used for calculating the flow properties of surveyed and blended cross sections:

READXS   Reads in cross-section data from a specified file and matches cross-section ID codes with the proper node designation.

SHELL   A sorting algorithm, used by the scan line conversion routine, adapted from The C Programming Language by Kernighan and Ritchie.

XSECT   The scan line algorithm which converts cross-section coordinates into a table of area A and width B versus height H.

AFROMH   Calculates A given H for a surveyed cross section.

HFROMH   Calculates H given A for a surveyed cross section. For a blended cross section, routine is called once for each parent and the results are averaged.

AFROMH2   Calculates A given H for a blended cross section.

Boundary Conditions

175. For the system of equations to be solved, a pair of boundary conditions, one upstream and one downstream, must be known for all time-steps. Either flow or stage is given for the upstream condition, and either flow,

stage, or a rating curve can be given for the downstream condition. The six alternative sets are given in Table 1 along with values of the residual matrix (see Equation 190). The rating curve must be of the form

$$H = COEF \ Q^{EXPO} \tag{184}$$

where

H = depth of flow, ft

Q = discharge, ft^3/sec

Then COEF and EXPO are input in the "segment" card (see Input and Output). If the rating curve is instead of the form $Q = aH^b$, then simply set EXPO = 1/b and COEF = $a^{(-1/b)}$.

Table 1

Library (LIB) of Boundary Conditions

LIB	Condition	Condition	R_n	a_1	a_2	a_m	a_{m-1}
1	H	H	0	0	1	0	1
2	H	Q	0	0	1	1	0
3	H	RC	RC	0	1	DBCQ	DBCA
4	Q	H	0	1	0	0	1
5	Q	Q	0	1	0	1	0
6	Q	RC	RC	1	0	DBCQ	DBCA

Note: H = stage
        Q = flow
       RC = Equation 185
     DBCQ = Equation 186
     DBCA = Equation 187
R_n, a_1, a_2, a_m, a_{m-1} are stated in Equation 190

176. One purpose of the boundary conditions is to supply values for use in the next time-step. A boundary condition of flow can be used directly. A boundary condition of stage can be used after the area and width are calculated from it. The major purpose of the boundary conditions is to properly constrain the system of equations. This purpose is accomplished by treating the equations that assign values to the boundary conditions--or alternatively the rating curve equation--in the same way as the governing equations. The residual is placed in the residual matrix R and the partial derivatives are

placed in the partial derivative matrix AA . For instance, the assignment of the upstream boundary condition to H is H = BCU or A = f(BCU) where f is the stage-area function and BCU is the value of the upstream boundary condition. Isolating all known quantities on the left, we have

$$A - f(BCU) = 0 \tag{185}$$

The residual is equal to zero. The partial derivative with respect to A is 1. The partial derivative with respect to Q is 0. The same reasoning applies to the downstream condition and to flow Q as a boundary condition.

177. A slightly different situation is presented by having a rating curve as the downstream condition. Here the equation is

$$H = COEF*Q^{EXPO} \tag{186}$$

Now this is not an assignment statement, but a constraint on the behavior of H and Q . Since the solution scheme involves iterative approximation, Equation 186, when coupled with the governing equations, is not solved exactly. It leaves a nonzero residual.

$$H - COEF*Q^{EXPO} = R \tag{187}$$

This residual finds its place in the residual matrix R. The partial derivatives of the above equation are taken with respect to Q and A . These are called in the program, respectively, DBCQ and DBCA,

$$\frac{\partial(-R)}{\partial Q} = EXPO*COEF*Q^{EXPO-1} \tag{188}$$

$$\frac{\partial(-R)}{\partial A} = -\frac{dH}{dA} = -\frac{1}{B} \tag{189}$$

93

The matrix becomes

$$
\begin{bmatrix}
a_1 & a_2 & & & & & \\
DFQ & DFA & DFQ1 & DFA1 & & & \\
DGQ & DGA & DGQ1 & DGA1 & 0 & & \\
& 0 & DFQ & DFA & DFQ1 & DFA1 & \\
& & DGQ & DGA & DGQ1 & DGA1 & 0 \\
& & & 0 & DFQ & DFA & DFQ1 & DFA1 \\
& & & & DGQ & DGA & DGQ1 & DGA1 \\
& & & & & & a_{m-1} & a_m
\end{bmatrix}
\cdot
\begin{bmatrix}
\Delta Q_1 \\
\Delta A_1 \\
\Delta Q_2 \\
\Delta A_2 \\
\Delta Q_3 \\
\Delta A_3 \\
\Delta Q_4 \\
\Delta A_4
\end{bmatrix}
=
\begin{bmatrix}
0 \\
R1_1 \\
R2_1 \\
R1_2 \\
R2_2 \\
R1_3 \\
R2_3 \\
R_n
\end{bmatrix}
\qquad (190)
$$

with a_1 , a_2 , a_{m-1} , a_m , and R_n taking on values which depend on the value of LIB (see Table 1).

Tributaries

178. The major concerns in designing a program to handle tributaries are that: (a) the relevant physics be modeled with fidelity; (b) the program should be capable of handling any arbitrary tributary network; (c) it should be efficient in input and output, computer storage, and execution; and (d) it should be amenable to possible future change.

179. In this program, a stream network is envisioned as a group of discrete segments. A segment is defined as a stretch of stream whose boundaries are a system boundary, a receiving stream, or a control structure. No control structures are permitted within a segment. Based on user-supplied information about each segment, the program constructs the network, determines an upstream ordering of the segments, decides what types of boundary conditions apply to each segment boundary, and determines at which array locations each of these boundary conditions values is to be found. An upstream ordering assures that when the turn comes for any given segment to be processed, the segment it

94

feeds into has already been processed. Conversely, a downstream ordering assures that when the turn comes for any given segment to be processed, all segments flowing into it have already been processed.

180. The first iteration of the Newton-Raphson method at each time-step requires new boundary conditions to be assigned at all time-varying boundaries. Subsequent iterations require new values only for boundary conditions that are generated within the systems, i.e., tributary junctions and flow through control structures. The program then assembles and solves the Newton-Raphson matrix for the entire system in what might be regarded as one step. In fact, however, the matrix is assembled and reduced segment by segment as the algorithm proceeds downstream. When the final segment is reached, the process is reversed, and the reduced, partitioned matrix is solved completely, segment by segment in an upstream order.

181. The detailed features of this algorithm can be illustrated by following the solution of a simple tributary system from the inside of the solution scheme out. The tributary J joins with the mainstream M at node K . In the remainder of this section, the portions of the governing equations relevant to tributary junctions are reviewed and placed within the matrix structure, and the solution of the matrix is delineated. Next, the way the main program controls the flow of information between subroutines and coordinates the matrix assembly and solution is described. Finally, the method of constructing the network and assigning priority and boundary conditions is discussed.

The coefficient matrix

182. The governing equations are discussed in Parts II and III but are reviewed here as the structure of the matrix solution is intimately tied to the structure of the equations. The Newton-Raphson method assembles a set of linear equations of the form

$$A \cdot x = R \tag{191}$$

where

$\quad$ A = coefficient matrix with elements of the form $\partial R_i / \partial x_j$

$\quad$ x = vector of increments in the controlling variables (Q and A)

$\quad$ R = vector of the negative residuals from the governing equations (continuity and momentum)

In the case of a single segment, the five-banded coefficient matrix is assembled as in the form of Equation 192, with three fully populated bands and two extreme bands half-populated.

$$
\begin{bmatrix}
1 & & & & & & & & \\
\partial F1/\partial Q1 & \partial F1/\partial A1 & \partial F1/\partial Q2 & \partial F1/\partial A2 & & & & & \\
\partial G1/\partial Q1 & \partial G1/\partial A1 & \partial G1/\partial Q2 & \partial G1/\partial A2 & 0 & & & & \\
& 0 & \partial F2/\partial Q2 & \partial F2/\partial A2 & \partial F2/\partial Q3 & \partial F2/\partial A3 & & & \\
& & \partial G2/\partial Q2 & \partial G2/\partial A2 & \partial G2/\partial Q3 & \partial G2/\partial A3 & 0 & & \\
& & & 0 & \partial F3/\partial Q3 & \partial F3/\partial A3 & \partial F3/\partial Q4 & \partial F3/\partial A4 & \\
& & & & \partial G3/\partial Q3 & \partial G3/\partial A3 & \partial G3/\partial Q4 & \partial G3/\partial A4 & \\
& & & & & & & 1 &
\end{bmatrix}
\cdot
\begin{bmatrix}
\Delta Q_1 \\ \Delta A_1 \\ \Delta Q_2 \\ \Delta A_2 \\ \Delta Q_3 \\ \Delta A_3 \\ \Delta Q_4 \\ \Delta A_4
\end{bmatrix}
=
\begin{bmatrix}
0 \\ -F_1 \\ -G_1 \\ -F_2 \\ -G_2 \\ -F_3 \\ -G_3 \\ 0
\end{bmatrix}
\qquad (192)
$$

183. In the case of a single tributary, two of these five-banded structures are placed head-to-toe and nine off-diagonal members are added to handle junction conditions, as in Equation 194.

184. The single off-diagonal element in the first quadrant $(-B_J/B_K)$ represents the condition that the water surface elevation at the mouth of the tributary is equal to that at the junction point on the mainstream, i.e. $E_J = E_K$. Taking differentials, $dE_J = dE_K$, However, the water surface elevation is the sum of the bed elevation and the depth of flow, and the bed elevation is taken to be a constant at any node, so $dH_J = dH_K$. To fit into the Newton-Raphson scheme, H must be replaced by the controlling variable A , or

$$
dH_J = dA_J \cdot \frac{dH_J}{dA_J} = dA_J \frac{1}{B_J} = dA_K \frac{1}{B_K} \qquad (193)
$$

As it appears in the matrix

$$
\begin{bmatrix}
1 \\
a_{11} & a_{12} & a_{13} & a_{14} \\
a_{21} & a_{22} & a_{23} & a_{24} & 0 \\
 & 0 & a_{31} & a_{32} & a_{33} & a_{34} \\
 & & a_{41} & a_{42} & a_{43} & a_{44} & 0 \\
 & & & 0 & a_{51} & a_{52} & a_{53} & a_{54} \\
 & & & & a_{61} & a_{62} & a_{63} & a_{64} \\
 & & & & & & 1 & & & & & -\dfrac{B_J}{B_K} \\
 & & & & & & & 1 \\
 & & & & & & a_{11} & a_{12} & a_{13} & a_{14} \\
 & & & & & & a_{21} & a_{22} & a_{23} & a_{24} & 0 \\
 & \dfrac{\partial F_{k-1}}{\partial Q_J} & \dfrac{\partial F_{k-1}}{\partial A_J} & & 0 & a_{31} & a_{32} & a_{33} & a_{34} \\
 & \dfrac{\partial G_{k-1}}{\partial Q_J} & \dfrac{\partial G_{k-1}}{\partial A_J} & & & a_{41} & a_{42} & a_{43} & a_{44} & 0 \\
 & \dfrac{\partial F_{k}}{\partial Q_J} & \dfrac{\partial F_{k}}{\partial A_J} & & & 0 & a_{51} & a_{52} & a_{53} & a_{54} \\
 & \dfrac{\partial G_{k}}{\partial Q_J} & \dfrac{\partial G_{k}}{\partial A_J} & & & & a_{61} & a_{62} & a_{63} & a_{64} & 0 \\
 & & & & & & & 0 & a_{71} & a_{72} & a_{73} & a_{74} \\
 & & & & & & & & a_{81} & a_{82} & a_{83} & a_{84} \\
 & & & & & & & & & & & 1
\end{bmatrix}
\begin{bmatrix} X \end{bmatrix}
=
\begin{bmatrix}
0 \\
R_1 \\
R_2 \\
R_3 \\
R_4 \\
R_5 \\
R_6 \\
0 \\
0 \\
R_1 \\
R_2 \\
R_3 \\
R_4 \\
R_5 \\
R_6 \\
R_7 \\
R_8 \\
0
\end{bmatrix}
\qquad (194)
$$

$$
dA_J - dA_K \frac{B_J}{B_K} = 0
$$

$$
\left[1 - \frac{B_J}{B_K} \right]
\begin{bmatrix}
\Delta A_J \\
\\
\Delta A_K
\end{bmatrix}
= 0
\qquad (195)
$$

185. The eight elements in third quadrant are $\dfrac{\partial F_{K-1}}{\partial Q_J}$, $\dfrac{\partial F_{K-1}}{\partial A_J}$, $\dfrac{\partial G_{K-1}}{\partial Q_J}$, $\dfrac{\partial G_{K-1}}{\partial A_J}$, $\dfrac{\partial F_K}{\partial Q_J}$, $\dfrac{\partial F_K}{\partial A_J}$, $\dfrac{\partial G_K}{\partial Q_J}$, $\dfrac{\partial G_K}{\partial A_J}$, where F and G are the residuals from the continuity equations, respectively, for the reaches above and below the junction point of the mainstream. These residuals must be adjusted for the presence of the tributary:

$$R_{2K-1} = -F_K + \frac{dt}{dX_K}\left[\theta Q_J + (1 - \theta)\, Q_J^j\right] = -F_K + \frac{F_J}{dX_K}$$

$$\text{(196)}$$

$$R_{2K} = -G_K + \frac{dt}{dX_K}\left[\theta\, \frac{Q_J^2}{A_J} + (1 - \theta)\, \frac{Q_J^{j2}}{A_J}\right]\cos\phi = -G_K + \frac{G_J}{dX_K}$$

where

F, G — lumping of terms

j — previous time-step

ϕ — junction angle

The derivatives are

$$\frac{\partial F_K}{\partial Q_J} = -\frac{dt}{dX_K}\,\theta$$

$$\frac{\partial F_K}{\partial A_J} = 0$$

$$\text{(197)}$$

$$\frac{\partial G_K}{\partial Q_J} = \frac{dt}{dX_K}\,\theta\left(\frac{Q_J}{A_J}\right)^2 \cos\phi$$

$$\frac{\partial G_K}{\partial A_J} = \frac{dt}{dX_K}\,\theta\left(\frac{Q_J}{A_J}\right)^2 \cos\phi$$

186. Conceptually, the solution begins with reducing the second quadrant to the identity matrix by Gauss elimination. First, the extreme diagonals are eliminated by pair-wise operations on the rows. Next, the lower codiagonal is eliminated by standard tridiagonal matrix methods. Finally, in eliminating

the upper codiagonal, the single first-quadrant element, which has to this point been untouched, generates a column of numbers above it (Equation 198). All elements of this column are proportional to the single original element in that column, $-B_J/B_K$. Because of this fact, that element can be given the value of $-B_J$ provisionally (as the tributary does not "know" the value of B_K), and the other elements can be multiplied by the factor $1/B_K$ later. The last and next-to-last rows in the tributary portion of the matrix can now be brought down to eliminate the eight (six nonzero) third-quadrant elements. This elimination changes the value of the four residuals around the junction node and the four elements of the coefficient matrix below the elements C_i . Following the numbering in Equations 194 and 198

$$
\begin{bmatrix}
1 \\
& 1 & & & & & & & & & & & & C_1 \\
& & 1 & & & & & & & & & & & C_2 \\
& & & 1 & & & & & & & & & & C_3 \\
& & & & 1 & & & & & & & & & C_4 \\
& & & & & 1 & & & & & & & & C_5 \\
& & & & & & 1 & & & & & & & C_6 \\
& & & & & & & 1 & & & & & & C_7 \\
& & & & & & & & 1 \\
& & & & & & a_{11} & a_{12} & a_{13} & a_{14} \\
& & & & & & a_{21} & a_{22} & a_{23} & a_{24} \\
& \frac{\partial F_{K-1}}{\partial Q_J} & 0 & & & & a_{31} & a_{32} & a_{33} & a_{34} \\
& \frac{\partial G_{K-1}}{\partial Q_J} & \frac{\partial G_{K-1}}{\partial A_J} & & & & a_{41} & a_{42} & a_{43} & a_{44} \\
& \frac{\partial F_K}{\partial Q_J} & 0 & & & 0 & a_{51} & a_{52} & a_{53} & a_{54} \\
& \frac{\partial G_K}{\partial Q} & \frac{\partial G_K}{\partial A_J} & & & & a_{61} & a_{62} & a_{63} & a_{64} \\
& & & & & & a_{71} & a_{72} & a_{73} & a_{74} \\
& & & & & & a_{81} & a_{82} & a_{83} & a_{84} \\
& & & & & & & & & & & & & & 1
\end{bmatrix}
\begin{bmatrix} \\ \\ \\ \\ \\ X \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}
=
\begin{bmatrix}
0 \\ R_1^* \\ R_2^* \\ R_3^* \\ R_4^* \\ R_5^* \\ R_6^* \\ 0 \\ 0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \\ R_8 \\ 0
\end{bmatrix}
\tag{198}
$$

$$a_{34} = \frac{\partial F_{K-1}}{\partial A_K} + \frac{dt}{dX_{K-1}} \; \theta \; \frac{C_6}{B_K}$$

$$a_{44} = \frac{\partial G_{K-1}}{\partial A_K} + \frac{2dt}{dX_{K-1}} \cos \phi \; \theta \; \frac{Q_J}{A_J} \frac{C_6}{B_K} + \frac{dt}{dX_{K-1}} \; \theta \left(\frac{Q_J}{A_J} \right)^2 \frac{B_J}{B_K} \cos \phi$$

$$a_{52} = \frac{\partial F_K}{\partial A_k} + \frac{dt\theta}{dX_K} \frac{C_6}{B_K}$$

$$a_{62} = \frac{\partial G_K}{\partial A_K} + \frac{2dt}{dX_K} \; \theta \; \frac{Q_J}{A_J} \frac{C_6}{B_K} \cos \phi + \frac{dt}{dK_K} \; \theta \left(\frac{Q_J}{A_J} \right)^2 \frac{B_J}{B_K} \cos \phi$$

$$R_3 = -F_{K-1} + \frac{F_J}{dX_{K-1}} + \frac{dt\theta}{dX_{K-1}} R_6^{\cdot} \qquad (199)$$

$$R_4 = -G_{K-1} + \frac{G_J}{dX_{K-1}} + \frac{2dt}{dX_{K-1}} \frac{Q_J}{A_J} \cos \phi \; R_6^{\cdot}$$

$$R_5 = -F_K + \frac{F_J}{dX_K} + \frac{dt\theta}{dX} R_6^{\cdot}$$

$$R_6 = -G_k + \frac{G_J}{dX_K} \frac{2dt}{dX_K} \; \theta \; \frac{Q_J}{A_J} \cos \phi \; R_6^{\cdot}$$

Terms that do not have to do with the main stem can be collected.

$$T_1 = dt \; \theta \; C_6$$

$$T_2 = F_J + dt \; \theta \; R_6^{\bullet}$$

$$T_3 = 2dt \; \theta \frac{Q_J}{A_J} C_6 + dt \; \theta \left(\frac{Q_J}{A_J}\right)^2 B_J \cos \phi \qquad (200)$$

$$T_4 = G_J + 2dt \; \theta \; \frac{Q_J}{A_J} R_6^{\bullet}$$

to leave

$$a_{39} = \frac{\partial F_{K-1}}{\partial A_K} + \frac{T_1}{dX_{K-1}B_K}$$

$$A_{44} = \frac{\partial G_{K-1}}{\partial A_K} + \frac{T_3}{dX_{K-1}B_K}$$

$$a_{52} = \frac{\partial F_K}{\partial A_K} + \frac{T_1}{dX_K B_K}$$

$$a_{62} = \frac{\partial G_K}{\partial A_K} + \frac{T_3}{dX_K B_K} \qquad (201)$$

$$R_3 = -F_{K-1} + \frac{T_2}{dX_{K-1}}$$

$$R_4 = -G_{K-1} + \frac{T_4}{dX_{K-1}}$$

$$R_5 = -F_K + \frac{T_2}{dX_K}$$

$$R_6 = -G_{K-1} + \frac{T_4}{dX_K}$$

187.   In the notation of the program, if the junction occurs at node   i
and   IR — 2(i-1)   and   IA — 5*IR

101

$$AA(IA-6) = a_{34}$$

$$AA(IA-2) = a_{44}$$

$$AA(IA+2) = a_{52}$$

$$AA(IA+6) = a_{62}$$

$$R(IR) = R_3$$ (202)

$$R(IR+1) = R_4$$

$$R(IR+2) = R_5$$

$$R(IR+3) = R_6$$

The terms T1 through T4 are calculated in subroutine CALC when it solves the tributary portion of the matrix and are passed back to that subroutine when it is called to work on the main stem. The adjustment in the main stem portion of the matrix and residuals vector is made, and the main stem's system of equations can be solved completely by Gauss elimination. This leaves all main stem variables solved, but the tributary system as yet undetermined. Once the coefficient matrix A is known, however, the elements C_i can be eliminated from the tributary matrix and R_i^{**} can be calculated from:

$$R_i^{**} = R_i^* - C_i \frac{\Delta A_K}{B_K}$$ (203)

Then R_i^{**} is used to make adjustments in the controlling variables Q and A in the normal way.

Flow of information

188. The preceding scheme can be generalized to handle any arbitrary system. The tributary passes to the receiving stream the values T_1 through T_4 and also K , the junction node. The receiving stream passes back to the tributary the value $\Delta A_K/B_K$, placing it into T_1. In the general case, T is an array of dimension (4,NS) where NS is the number of segments in the network. Other arrays involved in this information-passing process are JT(NS), which contains the junction nodes ITO(NS), IT1(NS), and IT2(NS). The array

ITO(L) tells where in the array T segment L should place its tributary information (ITO(L) = 0 if L is not a tributary). The array IT1(L) gives the index within the array T of information from the first tributary entering segment L, and IT2 gives the index of the information from the last tributary. If no tributaries enter segment L, IT1 and IT2 are set to zero. See Figure 13 (page 112) for an example.

189. The main program works downstream through the network, and for each segment L, it calls subroutine CALC with the arguments T, JT, ITO(L), IT1(L), and IT2(L). In subroutine CALC, segment L picks up tributary information from T between the indices IT1 and IT2. The matrix solver is called and upon return, segment L loads its own information into T at index ITO. The solution is completed by sweeping upstream through the system; for each segment, subroutine NEW is called with the same arguments. This time the segment picks up information from T at index ITO and distributes information to its tributaries at indices IT1 through IT2.

190. One sweep down and one sweep up determine the system. No segment-wise iteration is necessary. Other advantages of this procedure are that (a) it allows backwaters up tributaries; (b) the coefficient matrix is assembled and solved in pieces as required; and (c) since a banded-style solution is done, the number of operations performed is of order N , where N is the total number of nodes in the system. In addition, since the scheme is generalized, the number of "bookkeeping" operations is of order M , where M is the number of segments in the system.

Network structure

191. The user provides in the input NS the total number of segments, and then for each segment L gives (among other data) FEEDS(L) and JNODE(L); that is, which stream, if any, this segment flows into and the node on the receiving stream at which the junction occurs. If the two segments are parts of the same stream, but separated by a control structure, the input field of JNODE is left blank and so is taken to be zero.

192. From these data, the program infers the network structure, establishes an upstream ordering of the segments, determines where boundary conditions that are passed from segment to segment are to be found, and establishes the packing of the tributary-information-passing arrays JT and T. The latter is done by fixing the values of ITO, IT1, and IT2 for each segment.

193. The program begins with the assumption that the system has a single terminus. Only one segment, the last segment of the main stem, does not

flow into another segment. The index of this stream is the first entry in the array ORDER. Next the array FEEDS is scanned to find streams that feed into this first stream, and they are successively entered into ORDER. Next, the program looks at the second entry of the array ORDER and scans FEEDS for streams that flow into it. The process continues until the array ORDER is completely filled. Whenever a match is found between a tributary J and its receiving stream M, the program discriminates between JNODE equal to zero and JNODE greater than zero. In the former case, the two form a single stream separated by a control structure. The segment M therefore will find its upstream boundary condition at the last node of segment J. The latter is the case of a true tributary. Segment J will find its downstream boundary condition at the junction node on M. Also a new slot, IT, is called for the tributary-information arrays JT and T. The tributary J will deposit its information in slot IT. The receiving stream M will withdraw information from slot IT. Using the definitions of IT0, IT1, and IT2 given above, IT0(J) is set to IT, IT1(M) is set to IT if it has not already been set, and IT2(M) is set to IT.

Boundary conditions
in the tributary structure

194. For each segment, the user specifies what types of boundary conditions will apply upstream and downstream. Flow or stage (designated Q or H) may be given upstream, and flow, stage, or a rating curve (designated Q, H , or R) may be given downstream. The default at both ends is stage. Based on this input, the program assigns a value of one through six to LIB for that segment (Table 1).

195. The network structure may, however, override this assignment. A segment downstream of a control structure must have an upstream boundary condition of flow. A segment that is a true tributary must have a downstream boundary condition of stage.

196. Once the types of boundary conditions are established, the question is from where are the values of the boundary conditions to come. The answer is given in the arrays JBCU and JBCD for upstream and downstream, respectively. The default is that the boundary condition is time-invariant and is given on the "segment" card of the input deck (see Input and Output formats). In this case JBCU or JBCD is given a value of zero. The second possibility is that they are supplied in the "boundary conditions" cards. The user indicates to the program which boundary conditions will appear there by

104

means of the "boundary conditions identification" card. The value of JBCU or JBCD is set to the position of that piece of data on the boundary conditions card. Finally, as indicated above, the boundary condition may come from within the system; for example, the downstream boundary condition for segment L might come from node J. In this case JBCD(L) is set to -J.

197. When it is time to disburse boundary conditions, again three cases arise. If JBCU or JBCD is equal to zero, the boundary condition is time-invariant, and no change is made. If JBCU or JBCD is positive, then the boundary condition is obtained from that position in the input stream. If JBCU or JBCD is negative, the boundary condition is obtained from the node number -JBCU or -JBCD. As mentioned previously, an internally generated upstream boundary condition must be a flow. An internally generated downstream boundary condition must be a stage. Stage at the junction point on the receiving stream, however, is not equal to stage at the mouth of the tributary--it must first be corrected for the difference in bed elevation between the two.

Input and Output Formats

198. The input to the hydrodynamic model is provided in up to four files, depending upon the options selected. As stated previously, RIV1H expects an execution control file to provide the names of the input and auxiliary files that are required to run the program. The main input file contains information on the model configuration, such as specification of constants, initial conditions, and boundary specifications. Time-varying boundary conditions, lateral inflows, and spatially varying surveyed cross sections are specified in separate files. The structure of these files is described below. DOS extensions to the filenames are enforced. The main input file should have the extension *.INP, the time-varying lateral inflow file the extension *.LAQ, the boundary condition file *.BCF, and the surveyed cross-section data file the extension *.XSF. Similarly, output files will have the same name as the input file, but with the extension *.OUT for the main output file, *.HYD for the interface file for the water quality model, *.ERR for the error file, or *.HDF for the dump file containing data for graphical post-processing.

Main input

199. Input to RIV1H consists of eight types of cards: (a) a title card, (b) a grid card, (c) a constant card, (d) segment cards, (e) initial conditions cards, (f) a boundary conditions identification card, (g) time-step

105

cards, and (h) boundary filename specification cards. The details of the
input cards are discussed in the following paragraphs, and examples are given
in Figure 11. Output consists of the input data followed by the values of the

```
SAMPLE RIV1H DATA SET - USER'S MANUAL
 40, 1983,6,18,0.0, 1983,6,20,17.0,3
 1.0      32.17  000.00  0.67     0.100   1    0
    2
    6.0  0.0   6.0    1000.
 1 MAIN STEM ABOVE REREG DAM                      16   2   Q          Q  400.00
 4014.  500.   10.0    0.0 495.0 690.0  7.00    2.0  .050   0.0 0.625 0.025
 5248.  500.   10.2    0.0 494.0 153.0  3.40    2.0  .050   0.0 0.600 0.010
 3983.  500.   10.8    0.0 492.0  80.0  5.70    2.0  .050   0.0 0.000 0.000
 4334.  623.   10.2    0.0 485.0 185.0  2.10    2.0  .050   0.0 0.000 0.000
 4428.  750.   10.4    0.0 484.0 240.0  2.10    2.0  .050   0.0 0.000 0.000
 4428.  750.   10.4    0.0 479.0 300.0  1.85    2.0  .050   0.0 0.000 0.000
 4933.  750.   10.8    0.0 474.0 355.0  1.60    2.0  .050   0.0 0.000 0.000
 3667.  750.   10.9    0.0 472.0 175.0  3.20    2.0  .050   0.0 0.000 0.000
 4200.  750.   10.7    0.0 472.0 225.0  0.70    2.0  .050   0.0 0.000 0.000
 4200.  750.   10.7    0.0 470.0 224.0  0.80    2.0  .050   0.0 0.000 0.000
 4200.  750.   10.7    0.0 468.0 223.0  0.90    2.0  .050   0.0 0.000 0.000
 4200.  750.   10.5    0.0 466.0 221.0  1.00    2.0  .050   0.0 0.500 0.010
 4664.  750.   10.5    0.0 463.5 220.0  1.10    2.0  .050   0.0 0.000 0.000
 4664.  750.   10.4    0.0 460.5 215.0  1.80    2.0  .050   0.0 0.000 0.000
 4664.  750.   10.5    0.0 457.5 210.0  2.50    2.0  .050   0.0 0.000 0.000
 4352.  750.   10.4    0.0 455.0 205.0  3.20    2.0  .050   0.0 0.000 0.000
 2 MAIN STEM BELOW REREG DAM                      18       Q 400.    H 15.00
 4352.  750.   10.2    0.0 455.0 203.0  2.80    2.0  .050   0.0 0.550 0.011
 4352.  750.   10.9    0.0 455.0 200.0  2.40    2.0  .050   0.0 0.000 0.000
 4352.  750.   10.5    0.0 455.0 198.0  2.00    2.0  .050   0.0 0.000 0.000
 3733.  750.   10.8    0.0 455.5 195.0  1.60    2.0  .050   0.0 0.000 0.000
 3733.  750.   10.8    0.0 453.5 200.0  1.30    2.0  .050   0.0 0.000 0.000
 3733.  750.   10.6    0.0 451.5 205.0  1.00    2.0  .050   0.0 0.000 0.000
 4498.  750.   10.2    0.0 449.0 210.0  0.60    2.0  .050   0.0 0.000 0.000
 4498.  750.   10.3    0.0 445.0 225.0  0.90    2.0  .050   0.0 0.000 0.000
 4498.  750.   10.4    0.0 442.0 245.0  1.30    2.0  .050   0.0 0.000 0.000
 4498.  750.   10.9    0.0 439.0 260.0  1.60    2.0  .050   0.0 0.000 0.000
 5772.  750.   10.4    0.0 435.5 280.0  2.00    2.0  .050   0.0 0.000 0.000
 5772.  750.   10.8    0.0 434.5 300.0  2.00    2.0  .050   0.0 0.000 0.000
 5772.  750.   10.5    0.0 433.5 325.0  2.00    2.0  .050   0.0 0.000 0.000
 5772.  750.   10.3    0.0 432.5 345.0  2.00    2.0  .050   0.0 0.000 0.000
 5140.  750.   10.2    0.0 432.5 370.0  1.60    2.0  .050   0.0 0.000 0.000
 5140.  750.   10.1    0.0 428.5 610.0  1.75    2.0  .050   0.0 0.000 0.000
 5140.  750.   15.0    0.0 424.5 850.0  1.90    2.0  .050   0.0 0.000 0.000
 4450.  750.   15.0    0.0 420.01100.0  2.10    2.0  .050   0.0 0.000 0.000
 3 TRIBUTARY UPSTREAM OF REREG DAM                 6   1   4Q 250.   H        45.0
 2564.  250.   10.5    0.0 488.5 177.0  3.00    2.0  .040   0.0 0.500 0.010
 2970.  250.   10.2    0.0 488.0 140.0  1.90    2.0  .040   0.0 0.000 0.000
 2970.  250.   10.5    0.0 484.8 143.0  2.10    2.0  .040   0.0 0.000 0.000
 2793.  250.   10.7    0.0 481.5 147.0  2.30    2.0  .040   0.0 0.000 0.000
 1977.  250.   10.7    0.0 482.5 212.0  1.80    2.0  .050   0.0 0.000 0.000
 100.0  250.   10.7    0.0 482.5 250.0  2.40    2.0  .050   0.0 0.000 0.000
+1/
        2
 1983     6     1      0.0      450.
 1983     6    30      0.0      450.
SAMPLEH.BCF
```

Figure 11.   Sample RIV1H *.INP input data file

hydrodynamic variables for every node of each segment (one segment per page) for each time-step print interval. In addition to printed output, there is output onto disk storage for later use by RIV1Q or plotting programs.

200. <u>Title card.</u> The user has the first 80 columns of the title card to write any appropriate title. It will be printed, verbatim, at the top of every page of the output file.

201. <u>Grid card.</u> The grid card defines the total number of nodes, start and end times for the simulation, and segments in the run. The format of the start and end times are year, month, day, hour, where the year, month, and day are integers and the hour is real (decimal fraction). The number of nodes and segments are integers. The ten numbers can appear anywhere on the card and can be separated by a comma or blanks or both, but they must appear in the following order: nodes, start year, start month, start day, start hour, end year, end month, end day, end hour, and segments. Note also that the node numbers correspond to grid lines, not intervals. For instance, if the study area extends from river mile 1.0 to river mile 0.0 with 0.1-mile reaches, the number of nodes is 11, not 10.

202. <u>Constant card.</u> There are seven variables that may be assigned values on the constant card: BETA, GR, RMILE0, THETA, TOLER, IQL, and IXS. The variable RMILE0 is the river mile of the first node. The variable THETA is the weighting factor (see The Governing Equation, Part IV). Historically, models of this type have used a value of 0.55 or greater for stability reasons. This is the default RIV1H uses.

203. With some applications, a value of THETA higher than 0.55 (between 0.55 and 1.0) may be desirable to reduce parasitic oscillations (Liggett and Cunge 1975). Parasitic oscillations, which are actually numerical dispersion or phase errors associated with short waves, can cause modeling problems when the parasitic oscillation is on the same order of magnitude as the depth (a negative depth can result). Values of THETA between 0.6 and 0.75 have been used to reduce these oscillations without significant loss of accuracy. However, accuracy does decrease as THETA is increased because of increased numerical dampening. The requirement for numerical stability is $0.5 < \theta < 1.0$.

204. The variable TOLER is the maximum acceptable relative tolerance used by the program to decide whether another iteration is necessary. The default value is 0.001; that is, iteration will stop when the difference between current and previous estimates of flow and area for every node is less than 0.001 times the root mean square of all flows and areas, respectively,

throughout the system. If the user desires greater accuracy and has a stable system, a lower value of TOLER may be desired. Conversely, if the user requires less accuracy and swifter computation time, then a higher value may be desired, such as 0.1. This value has yielded acceptable accuracy. It should be pointed out that TOLER is a control on precision from one time-step to the next--not from the beginning of the run to the end.

205. The variable GR is acceleration caused by gravity. Its default value is 32.174 ft/sec^2. If the user wishes to make a run in SI units, then GR should be set at 9.80 m/sec^2. Several things to keep in mind in this case are that the river miles in the output are meaningless, the Manning's coefficient must be multiplied by 1.49, and the output is not usable in RIV1Q. The variable BETA is the momentum correction factor, with a default value of 1.0. The variable IQL specifies the option for lateral inflows. If IQL is greater than or equal to one, then time-varying lateral inflows will be expected, and the lateral inflow file name is given in the second line of RIV1H.CTL. The variable IXS specifies the option for cross sections. If IXS is greater than or equal to one, then the name of the file containing the surveyed cross-sectional data should be listed on the third line of RIV1H.CTL.

206. The form of the constant card is similar to the grid card. Variables are read in free format and must be in the following order: BETA, GR, RMILEO, THETA, TOLER, IQL, and IXS. The variables BETA, GR, RMILEO, THETA, and TOLER are real numbers, while IQL and IXS are integers. The variables can be separated by spaces or a comma (see Figure 11).

207. <u>Print update card.</u> The print update card identifies the number of print interval updates that will be read from the following card.

208. <u>Print interval card.</u> On the print interval card, the user specifies pairs of data containing the print interval in hours and the Julian day for the print interval update. The print intervals allow the user to print more frequently during critical periods of the simulation. The format of the update cards is n pairs of print intervals and Julian days (both real), where n is the number of print interval updates specified on the print update card.

209. <u>Segment card.</u> On the segment card, the user specifies segment-specific data: ID, the identification number; SNAME, the name; NNODE, the number of nodes in this segment; FEEDS, which segment it flows into, if any; JNODE, the number of the junction node on the receiving stream; BTU, the type of boundary condition imposed at the upstream end; BCU, a value for that boundary condition if it is time-invariant; BTD, the type of boundary

condition imposed on the downstream end; BCD, a value for that boundary condition if it is time-invariant; and COSP, the junction angle in degrees if the segment is a tributary. The format is

$$I2,A40,3I3,2(A1,F8.0),F8.0.$$

The choice of identification number and the ordering of the segment cards are arbitrary. The only constraint is that identification numbers used here must be consistent with those used in the water quality model and they must be positive. The segment name can be up to 40 characters long. It has no use in the model other than labeling output. NNODE counts the total number of nodes in the segment, which is one more than the number of reaches. FEEDS refers to the receiving stream by its identification number. JNODE refers to the junction node on the receiving stream, starting with the first node of the stream equal to one. Referring to Figure 9, JNODE for Segment 1 would be 10. When two segments are actually part of the same stream but separated by a control structure, JNODE is left blank (control structures are not permitted within a segment, only at the boundaries). The boundary type H signifies stage, depth above bottom (and is the default), Q signifies flow, and R signifies a rating curve. A rating curve is not a permissible upstream boundary condition. The value of the boundary condition is in feet for stage and cubic feet per second for flow. COSP is the angle formed by the tributary junction, in degrees. See Figure 9 for the sense of this angle. In the case of a rating curve of the form

$$H = COEF*Q^{EXPO} \tag{204}$$

the exponent EXPO is given in place of BCD and COEF in place of COSP. There is no conflict as a tributary cannot be terminated by a rating curve. If, instead, the rating curve is of the form $Q = aH^b$, then set $EXPO = 1/b$ and $COEF = a^{(-1/b)}$.

210. _Initial conditions card._ Each segment card is followed by a set of initial conditions cards, one per node. Twelve pieces of data can appear on each initial conditions card. Each number can occupy a field of six. Any right-justified number without a decimal point will be taken as having a decimal point at the end (e.g., 25 will be read as 25.0). Any value not specified will be taken as zero. These pieces of data are, in order: length of the

reach downstream from the node in feet (DX1); flow in cubic feet per second (Q); stage-height of water above the channel bottom in feet (H); constant (time-invariant) lateral inflow in cubic feet per second per foot (QL) (flow divided by DX1); streambed elevation relative to some datum in feet (Z, initially EL is used and moved into Z array), (C1), (C2), and (C3) constants in the cross-section formula (see Cross-Section Formulas, above); Manning's coefficient for the reach downstream of the node (CN1); constriction energy loss coefficient for the reach downstream of the node (KE1); intercept of the linear equation relating Manning's n to depth of flow in the stream (AX); and the slope (DNDH) of the linear equation relating Manning's n to depth. If the user elects to use surveyed cross sections (IXS $\geq$ 1), then in place of the constant C1, the user specifies an alphanumeric code (up to six characters) identifying the cross-sectional data. For blended cross sections, the field for C1 is left blank. The term KE1 should be used only when there is a rapid constriction in the channel and only after it has been determined that refining the grid work in the area of the constriction is infeasible. Values for KE1 range from 0.0 to 0.5, with the higher value associated with abrupt constriction.

211. Manning's n at a cross section can vary with flow conditions in many streams. High values may be more representative for shallow-depth conditions whereas lower values may be appropriate for deeper flow. For time-varying flow conditions, a variable Manning's coefficient may have to be adjusted as a function of depth during the simulation. Typically, shoal areas exhibit a variable n that can have a significant effect on computed stage; at low stage, n is usually larger than at high stage. If the user wishes to allow for variable Manning's coefficients, the values of AX and DNDH in the input data, which appear on the initial conditions card, should be set to the appropriate values. If the user chooses not to vary Manning's n, these values should be set to 0. The equation relating Manning's n to depth is XMAN = AX - DNDH*H where H is the value for depth at a particular node. During initiation, XMAN is set to the original value (CN1) at that node. As the program executes, XMAN is adjusted with relation to depth over time. If at some point during the execution, the value of XMAN becomes less than 0.01, XMAN is reset to 0.01. If this event occurs, a message is printed to the diagnostic file, FILE7. The first location in the code where the adjusted Manning's values occur is several lines after statement 301 while the second location is several lines after statement 170.

212. <u>Boundary conditions identification card.</u> On the boundary conditions identification card, the user indicates which time-varying boundary conditions will be provided in external boundary conditions files (for flow Q and head H boundary conditions only). As an example, consider a system with two segments separated by a control structure where segment 1 is the reach above the structure and segment 2 is the reach below. The upstream and downstream boundary conditions will be specified for segment 1, whereas the downstream boundary condition will be specified for segment 2. The upstream boundary condition for segment 2 (below the control structure) is determined internally by the program. The card would appear as

<div align="center">+1,-1,-2/</div>

with the numbers being the ID numbers, the positive sign standing for upstream, and the negative for downstream. The list must be terminated by a slash.

213. <u>Time-step update card.</u> On the time-step update card, the user indicates the total number (NUM) of time-step updates. A minimum of two update cards is required (even if the time-step does not change) or the program will abort (see bottom of Figure 11). The format is I10. In the following cards, a total of NUM time-step cards must be provided.

214. <u>Time-step update interval card.</u> On the time-step update interval card, the user indicates the year, month, day, and hour of the update and the model time-step (seconds) which will apply for the period between that period and the next update interval. The format of the update card is 3I5, 2F10.0. There must be NUM update intervals specified.

215. <u>Boundary condition cards.</u> On these cards, the user specifies the file names containing the time-varying boundary condition data. The number of files specified and their order should correspond to the boundary conditions identification card. The format is A15 for each card.

<u>Boundary conditions file(s)</u>

216. Time-varying boundary conditions are specified in external files, whose names are specified in the main input data set. An example of the boundary condition file (with DOS extension *.BCF) is given in Figure 12. The first line of these files contains the segment number for which the boundary is provided, the number of boundary condition updates (NUMBC), and the

<div align="center">111</div>

```
                1                24              0
          1983    6   18       0.00        500.0
          1983    6   18      12.01        500.0
          1983    6   18      15.01        500.0
          1983    6   18      18.01        500.0
          1983    6   18      20.51        600.0
          1983    6   18      23.01        600.0
          1983    6   19       1.01        600.0
          1983    6   19       3.51        600.0
          1983    6   19       6.01        700.0
          1983    6   19       8.51        700.0
          1983    6   19      11.01        800.0
          1983    6   19      13.51        800.0
          1983    6   19      16.01        700.0
          1983    6   19      18.51        700.0
          1983    6   19      19.76        700.0
          1983    6   19      22.26        700.0
          1983    6   20       0.76        600.0
          1983    6   20       2.01        500.0
          1983    6   20       4.51        400.0
          1983    6   20       7.01        400.0
          1983    6   20       9.51        400.0
          1983    6   20      12.01        400.0
          1983    6   20      14.51        400.0
          1983    6   20      17.01        400.0
```

Figure 12.   Sample RIV1H *.BCF input data file

variable INTBC. If the variable INTBC is greater than or equal to one, the boundary conditions will be linearly interpolated between update intervals. Otherwise, they will be held constant between updates (a step function). The format of the first line of the file is 3I10.

217. Following the initial line of the boundary condition file are NUMBC lines of update intervals. Each line of update will contain the year, month, day, and hour for the update and the corresponding boundary condition (flow or head). The format for these lines is 3I5, 2F10.0. The update times should bracket the period of model simulation.

Lateral inflow file

218. The lateral inflow file contains the time-varying lateral inflows. If the user elects to input time-varying lateral inflows ($IQL \geq 1$), the file name (with DOS extension *.LAQ) should be given in the second record of RIV1H.CTL. An example of the lateral inflow file is given in Figure 13.

```
           6           5           0
    25
    26
    27
    28
    29
  1983    6    18       0.0       0.01      0.01      0.01      0.01      0.01
  1994    6    20       0.0       0.01      0.01      0.01      0.01      0.01
  1994    6    20       2.0       0.01      0.01      0.01      0.01      0.01
  1994   ·6    20       6.0       0.02      0.01      0.02      0.01      0.01
  1994    6    20      12.0      .0.01      0.01      0.01      0.01      0.01
  1994   12    21      17.1       0.01      0.01      0.01      0.01      0.01
```

Figure 13.   Sample RIV1H *.LAQ input data file

219.   The first line of the lateral inflow file contains the specifica-
tions for the number of updates (NUMLAT_UD), the number of nodes which will
receive lateral inflows (NUML), and the variable LATOPT.   If the variable
LATOPT is greater than or equal to one, the lateral inflows will be linearly
interpolated between update intervals.   Otherwise, they will be held constant
between updates (a step function).   The format for this line is 3I5.

220.   Following the initial line of the input will be NUML lines on
which the node numbers that will receive the lateral inflows are specified.
The node numbers correspond to the order specified in the main input.   The
format is I5, so there will be one node specification per line of input.

221.   Following the specification of node numbers, there will be
NUMLAT_UD lines of input specifying the date (year, month, day, hour) followed
by the lateral inflow (flow rate per unit length) for each of the NUML nodes.
The format of these lines of input is 3I5, 150F10.0.

Surveyed cross-section file

222.   The type of cross section employed at any particular node is indi-
cated on the initial conditions card for that node.   For surveyed cross sec-
tions this is done by placing an alphanumeric code (up to six characters) in
the field normally reserved for the constant C_1.   For blended cross sections
leave the field blank.

223.   The cross-section data themselves are given in a separate file
(Figure 14) whose name (with DOS extension *.XSF) is given in the third record
of the control file.   The option for using surveyed cross sections (IXS $\geq$ 1)
is specified on the constant card in the main input file.   The cross-section
file has the following format.   Each cross section has a six-character ID code

113

```
A23     Five miles beyond the I-70 overpass
        0.0       0.0
        2.0       3.0
        3.0       6.0
        6.0       5.0
        7.0       3.0
        9.0       6.0
       12.0       9.0
       16.0      10.0
       17.0       8.0
       18.0       7.0
       20.0       7.0
       22.0       6.0
       24.0       3.0
       25.0       1.0
A95     Ten miles beyond the I-70 overpass
        0.0       0.0
        4.0       3.1
        8.0       4.0
       12.0       7.0
       15.0      11.0
       18.0       8.0
       22.0       5.0
       25.0       1.0
```

Figure 14.   Sample RIV1H *.XSF input data file

followed by an optional description. On succeeding records are the X and Y coordinates, in 2F10.0 format, with the origin for X and Y beginning at the top of the left streambank, where X increases to the right and Y increases downward when looking downstream. No other delimiters or separators are required. Units are in feet. As with the algebraic method, the channel bed elevation is understood to refer to the absolute lowest point in the surveyed cross section.

Output

224. Output from RIV1H consists of several files. One of the files, which will have the DOS extension *.OUT, consists of the input data and the simulation results at specified print intervals. The simulation results contain the information for each segment at each time-step print interval, one segment per page. At the top of the page appears the program title. The next line prints the run title as supplied by the user. On the following lines are the time-step number and the elapsed time in the simulation, in year, month,

day, hour, followed by the segment ID number and name. Below that are column
headings and data for river mile, flow (cubic feet per second), area (feet),
width (feet), stage (feet), and water surface elevation (feet) (Figure 15).
The output file also is updated whenever there is a change in the time-varying
(time-step, boundary condition, or lateral inflow data), with the update time,
new value, and next update.

225. Another output file is in standard FORTRAN unformatted records
used as input to RIV1Q and is written to a file with the name of the main
input file but with the DOS extension *.HYD. The records are organized as
follows:

Record 1   MNODE,STARTIME,ENDTIME,SYEAR,NS
Record 2   JT,ITO,ITI,IT2,NNODE,NODE1,ID,JBCU
Record 3   DX1,QL,RMILE
Record 4   DT,Q,A,B,EL

where all variables are arrays except for MNODE, STARTTIME, ENDTIME, SYEAR,
NS, and DT. Record 4 is contained in the array HYDRO (with exceptions of DT)
and is written every time-step, as this information is required to drive RIV1Q
and plot hydrodynamic output. All succeeding records are identical to
record 4 but for succeeding time-steps. A diagnostics output file (file 7) is
generated under certain conditions during the execution of RIV1H. If the
Courant number becomes less than 1.0, if Manning's n becomes less than or
equal to 0.01, or if the depth goes below 0.0, then a message is written to
file 7 telling the user at which node this event occurred. If the program has
trouble converging, a diagnostic message is written to file 7, allowing the
user to examine the node at which nonconvergence is occurring.

Special considerations

226. Upland streams characterized by riffles and pools can be difficult
to model with hydraulic routing models, such as RIV1H. Generally, two types
of problems may be encountered: start-up of the model and phase errors
associated with unsteady flow waves. Streams with riffle sections accentuate
these problems.

227. If inaccurate initial depths are selected for the initial flows,
the model will have trouble getting started. If good initial estimates of
depths are not available, it is best to run the model for a steady-state flow
condition using deeper than expected conditions. With a constant inflow rate
and either a constant head or rating curve downstream, the water surface in
the segment will eventually drop to a steady-state water surface profile

NUMBER OF NODES = 40   NUMBER OF SEGMENTS = 3

STARTIME: YEAR -1983,MONTH- 6,DAY 18,HOUR = 0.0000 JULIAN DAY = 169.000
END TIME: YEAR -1983,MONTH- 6,DAY 20,HOUR = 17.0000 JULIAN DAY = 171.708

```
BETA   =   1.00
GR     =  32.17
IPRINT =   1
RMILE0 =   0.00
THETA  =   0.67
TOLER  =   0.1000
```

NUMBER OF PRINT UPDATES   2
PRINT INTERVAL =   6.000   PRINT TIME   0.000
PRINT INTERVAL =   6.000   PRINT TIME   1000.000

1 MAIN STEM ABOVE REREG DAM
```
                                      16  2  OQ    0.Q    400.    0.
4014.000  500.000  10.000  0.000  495.000  690.0  7.000   0.  0.050  0.000  0.625  0.025
5248.000  500.000  10.200  0.000  494.000  153.0  3.400  2.000  0.050  0.000  0.600  0.010
3983.000  500.000  10.800  0.000  492.000   80.0  5.700  2.000  0.050  0.000  0.000  0.000
4334.000  623.000  10.200  0.000  485.000  185.0  2.100  2.000  0.050  0.000  0.000  0.000
4428.000  750.000  10.400  0.000  484.000  240.0  2.100  2.000  0.050  0.000  0.000  0.000
4428.000  750.000  10.400  0.000  479.000  300.0  1.850  2.000  0.050  0.000  0.000  0.000
4933.000  750.000  10.800  0.000  474.000  355.0  1.600  2.000  0.050  0.000  0.000  0.000
3667.000  750.000  10.900  0.000  472.000  175.0  3.200  2.000  0.050  0.000  0.000  0.000
4200.000  750.000  10.700  0.000  472.000  225.0  0.700  2.000  0.050  0.000  0.000  0.000
4200.000  750.000  10.700  0.000  470.000  224.0  0.800  2.000  0.050  0.000  0.000  0.010
4200.000  750.000  10.500  0.000  468.000  223.0  0.900  2.000  0.050  0.000  0.500  0.000
4664.000  750.000  10.500  0.000  466.000  221.0  1.000  2.000  0.050  0.000  0.000  0.000
4664.000  750.000  10.400  0.000  463.500  220.0  1.100  2.000  0.050  0.000  0.000  0.000
4352.000  750.000  10.500  0.000  460.500  215.0  1.800  2.000  0.050  0.000  0.000  0.000
          750.000  10.400  0.000  457.500  210.0  2.500  2.000  0.050  0.000  0.000  0.000
                                   455.000  205.0  3.200  2.000  0.050  0.000  0.000  0.000
```

2 MAIN STEM BELOW REREG DAM
```
                                      18  0  OQ   400.H    15.    0.
4352.000  750.000  10.200  0.000  455.000  203.0  2.800   0.  0.050  0.000  0.550  0.011
4352.000  750.000  10.900  0.000  455.000  200.0  2.400  2.000  0.050  0.000  0.000  0.000
4352.000  750.000  10.500  0.000  455.000  198.0  2.000  2.000  0.050  0.000  0.000  0.000
3733.000  750.000  10.800  0.000  453.500  195.0  1.600  2.000  0.050  0.000  0.000  0.000
3733.000  750.000  10.800  0.000  453.500  200.0  1.300  2.000  0.050  0.000  0.000  0.000
3733.000  750.000  10.600  0.000  451.500  205.0  1.000  2.000  0.050  0.000  0.000  0.000
4498.000  750.000  10.200  0.000  449.000  210.0  1.000  2.000  0.050  0.000  0.000  0.000
4498.000  750.000  10.300  0.000  445.000  225.0  0.600  2.000  0.050  0.000  0.000  0.000
4498.000  750.000  10.400  0.000  442.000  245.0  0.900  2.000  0.050  0.000  0.000  0.000
4498.000  750.000  10.900  0.000  439.000  260.0  1.300  2.000  0.050  0.000  0.000  0.000
5772.000  750.000  10.400  0.000  435.500  280.0  1.600  2.000  0.050  0.000  0.000  0.000
5772.000  750.000  10.800  0.000  434.500  300.0  2.000  2.000  0.050  0.000  0.000  0.000
5772.000  750.000  10.500  0.000  433.500  325.0  2.000  2.000  0.050  0.000  0.000  0.000
5140.000  750.000  10.300  0.000  432.500  345.0  2.000  2.000  0.050  0.000  0.000  0.000
5140.000  750.000  10.200  0.000  432.500  370.0  1.600  2.000  0.050  0.000  0.000  0.000
5140.000  750.000  10.100  0.000  428.500  610.0  1.750  2.000  0.050  0.000  0.000  0.000
4450.000  750.000  15.000  0.000  424.500  850.0  1.900  2.000  0.050  0.000  0.000  0.000
          750.000  15.000  0.000  420.000 1100.0  2.100  2.000  0.050  0.000  0.000  0.000
```

Figure 15. Sample RIV1H *.OUT output data file (Sheet 1 of 14)

```
3 TRIBUTARY UPSTREAM OF REREG DAM
                                          6  1    4Q   250.H     0.        45.
2564.000  250.000  10.500  0.000  488.500  177.0  3.000  0.040  0.000  0.500  0.010
2970.000  250.000  10.200  0.000  488.000  140.0  1.900  0.040  0.000  0.000  0.000
2970.000  250.000  10.500  0.000  484.800  143.0  2.100  0.040  0.000  0.000  0.000
2793.000  250.000  10.700  0.000  481.500  147.0  2.300  0.040  0.000  0.000  0.000
1977.000  250.000  10.700  0.000  482.500  212.0  1.800  0.050  0.000  0.000  0.000
 100.000  250.000  10.700  0.000  482.500  250.0  2.400  0.050  0.000  0.000  0.000

*************** TIME STEPS *******************
  Number of Time/Date Pairs =     2
Year  1983 Month  6 Day  1 Hour  0.0000 Julian day   152.0000   Time Step (sec) =   450.0000
Year  1983 Month  6 Day 30 Hour  0.0000 Julian day   181.0000   Time Step (sec) =   450.0000

*** Boundary Conditions Specified for Segment ***    1
Will be Read From File =  SAMPLEH.BCF
Interpolation Option:   STEP FUNCTION

Time Varying Lateral Inflows Specified at          x-sections
  25    26    27    28    29
Interpolation Option:   STEP
1

SAMPLE RIV1H DATA SET - USER'S MANUAL
TIME STEP (sec) =    450.00
```

```
OUTPUT FOR SEGMENT NO.   1
Time: Year =1994,Month= June,Day  18,Hour =  0.0000 Step=  0 Segment=  1 MAIN STEM ABOVE REREG DAM
Upstream Boundary  0.000  Downstream Boundary = 400.000
```

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	500.000	7600.000	830.000	10.000	505.000	0.050
2	27.00	500.000	1914.336	222.360	10.200	504.200	0.050
3	26.01	500.000	1528.848	203.120	10.800	502.800	0.050
4	25.25	623.000	2105.484	227.840	10.400	495.200	0.050
5	24.43	750.000	2723.136	283.680	10.400	494.400	0.050
6	23.59	750.000	3320.096	338.480	10.800	489.400	0.050
7	22.76	750.000	4020.624	389.560	10.800	484.800	0.050
8	21.82	750.000	2287.692	244.760	10.900	482.900	0.050
9	21.13	750.000	2487.643	239.980	10.700	482.700	0.050
10	20.33	750.000	2488.392	241.120	10.700	480.700	0.050
11	19.54	750.000	2489.141	242.260	10.700	478.700	0.050
12	18.74	750.000	2430.750	242.000	10.500	476.500	0.050
13	17.94	750.000	2431.275	243.100	10.500	474.000	0.050
14	17.06	750.000	2430.688	252.440	10.500	470.900	0.050
15	16.18	750.000	2480.625	262.500	10.500	468.000	0.050
16	15.29	750.000	2478.112	271.560	10.400	465.400	0.050

```
OUTPUT FOR SEGMENT NO.   2
Time: Year =1994,Month= June,Day  18,Hour =  0.0000 Step=  0 Segment=  2 MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000  Downstream Boundary = 15.000
```

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	750.000	2361.912	260.120	10.200	465.200	0.050
2	14.47	750.000	2465.144	252.320	10.900	465.900	0.050
3	13.65	750.000	2299.500	240.000	10.500	465.500	0.050

Figure 15. (Sheet 2 of 14)

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
4	12.82	750.000	2292.624	229.560	10.800	466.300	0.050
5	12.11	750.000	2311.632	228.080	10.800	464.300	0.050
6	11.41	750.000	2285.360	226.200	10.600	462.100	0.050
7	10.70	750.000	2204.424	222.240	10.300	459.200	0.050
8	9.85	750.000	2412.981	243.540	10.400	455.300	0.050
9	9.00	750.000	2688.608	272.040	10.400	452.400	0.050
10	8.15	750.000	3024.096	294.880	10.400	449.900	0.050
11	7.29	750.000	3128.320	321.600	10.800	445.900	0.050
12	6.20	750.000	3473.280	343.280	10.500	445.300	0.050
13	5.11	750.000	3633.000	367.000	10.300	444.000	0.050
14	4.01	750.000	3765.680	386.200	10.200	442.800	0.050
15	2.92	750.000	3940.464	402.640	10.100	442.700	0.050
16	1.95	750.000	6339.518	645.350	10.000	438.600	0.050
17	0.97	750.000	13177.500	907.000	15.000	439.500	0.050
18	0.00	750.000	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1994,Month= June,Day 18,Hour = 0.0000 Step= 0 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM
Upstream Boundary 250.000 Downstream Boundary = 0.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	2189.250	240.000	10.500	499.000	0.040
2	2.03	250.000	1625.676	178.760	10.200	498.200	0.040
3	1.47	250.000	1733.025	187.100	10.500	495.300	0.040
4	0.90	250.000	1836.227	196.220	10.700	492.200	0.040
5	0.37	250.000	2474.482	250.520	10.700	493.200	0.050
6	0.00	250.000	2949.776	301.360	10.700	493.200	0.050

SAMPLE RIV1B DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1994,Month= June,Day 18,Hour = 5.9941 Step= 48 Segment= 1 MAIN STEM ABOVE REREG DAM
Upstream Boundary 500.000 Downstream Boundary = 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	500.000	6367.830	808.961	8.496	503.496	0.413
2	27.00	565.073	1550.958	210.959	8.523	502.523	0.515
3	26.01	586.084	174.449	101.875	1.918	493.918	0.050
4	25.25	732.701	853.515	203.454	4.394	489.394	0.050
5	24.43	891.114	439.885	247.580	1.804	485.804	0.050
6	23.59	974.095	1107.662	313.362	3.612	482.612	0.050
7	22.76	1124.173	3095.652	381.886	8.402	482.402	0.050
8	21.82	1267.734	2150.025	241.134	10.333	482.333	0.050
9	21.13	1340.368	2384.559	239.378	10.270	482.270	0.050
10	20.33	1413.427	2857.186	243.556	12.222	482.222	0.050
11	19.54	1478.980	3346.439	248.551	14.193	482.193	0.050
12	18.74	1467.357	3710.010	252.351	15.676	481.676	0.050
13	17.94	1311.427	4244.576	258.988	17.723	481.223	0.343
14	17.06	1044.485	5227.752	289.592	20.721	481.221	0.050
15	16.18	742.667	6388.022	328.602	23.721	481.221	0.050
16	15.29	400.000	7575.371	372.811	26.221	481.221	0.050

Figure 15. (Sheet 3 of 14)

118

OUTPUT FOR SEGMENT NO. 2
Time: Year =1994,Month= June,Day 18,Hour = 5.9941 Step= 48 Segment= 2   MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000   Downstream Boundary = 15.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1396.964	238.443	6.329	461.329	0.480
2	14.47	428.346	847.719	219.404	4.042	459.042	0.050
3	13.65	469.167	776.776	213.117	3.779	458.779	0.050
4	12.82	509.183	444.605	202.166	2.239	457.739	0.050
5	12.11	547.996	470.211	206.024	2.316	455.816	0.050
6	11.41	594.116	490.175	209.729	2.364	453.864	0.050
7	10.70	643.322	425.069	212.417	2.013	451.013	0.050
8	9.85	718.248	536.035	229.250	2.360	447.360	0.050
9	9.00	836.667	612.756	251.422	2.469	444.469	0.050
10	8.15	1034.480	752.284	269.097	2.844	441.844	0.050
11	7.29	1320.339	1531.906	301.084	5.273	440.773	0.050
12	6.20	1743.746	1761.524	322.632	5.658	440.158	0.050
13	5.11	2181.443	2002.968	348.783	5.945	439.445	0.050
14	4.01	2590.439	2195.529	369.585	6.145	438.645	0.050
15	2.92	2877.362	1426.153	382.137	3.792	436.292	0.050
16	1.95	2985.809	4101.262	633.092	6.598	435.098	0.050
17	0.97	3008.472	9143.251	889.937	10.510	435.010	0.050
18	0.00	3011.542	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1994,Month= June,Day 18,Hour = 5.9941 Step= 48 Segment= 3   TRIBUTARY UPSTREAM OF REREG DAM
Upstream Boundary 250.000   Downstream Boundary = 6.894

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	809.918	202.603	4.267	492.767	0.457
2	2.03	251.915	247.840	146.573	1.730	489.730	0.040
3	1.47	256.843	705.921	162.415	4.623	489.423	0.040
4	0.90	263.233	1305.901	183.366	7.906	489.406	0.040
5	0.37	270.516	1548.261	236.835	6.899	489.399	0.050
6	0.00	276.853	1837.674	283.090	6.894	489.394	0.050

SAMPLE RIV1H DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1994,Month= June,Day 18,Hour = 11.9883 Step= 96 Segment= 1   MAIN STEM ABOVE REREG DAM
Upstream Boundary 500.000   Downstream Boundary = 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	500.000	6095.411	804.225	8.159	503.159	0.421
2	27.00	513.416	1483.602	208.774	8.202	502.202	0.518
3	26.01	517.714	158.250	100.042	1.758	493.758	0.050
4	25.25	650.657	827.398	202.916	4.266	489.266	0.050
5	24.43	783.626	398.695	246.879	1.638	485.638	0.050
6	23.59	766.306	1150.885	313.873	3.750	482.750	0.050
7	22.76	724.697	3186.137	382.644	8.639	482.639	0.050
8	21.82	681.052	2218.626	242.947	10.617	482.617	0.050
9	21.13	655.754	2463.796	239.841	10.601	482.601	0.050
10	20.33	626.553	2946.963	244.144	12.590	482.590	0.050
11	19.54	596.667	3443.751	249.252	14.584	482.584	0.050

Figure 15. (Sheet 4 of 14)

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
12	18.74	565.776	3924.089	254.042	16.521	482.521	0.335
13	17.94	533.467	4568.453	261.727	18.967	482.467	0.050
14	17.06	494.462	5591.175	294.077	21.966	482.466	0.050
15	16.18	450.239	6800.919	334.827	24.965	482.465	0.050
16	15.29	400.000	8044.222	380.777	27.465	482.465	0.050

OUTPUT FOR SEGMENT NO. 2 MAIN STEM BELOW REREG DAM
Time: Year =1994,Month= 6 ,Day 18,Hour = 11.9883 Step= 96 Segment= 2
Upstream Boundary 400.000 Downstream Boundary = 15.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1373.252	237.885	6.230	461.230	0.481
2	14.47	401.509	783.551	217.996	3.749	458.749	0.050
3	13.65	403.771	712.033	211.897	3.474	457.474	0.050
4	12.82	406.025	382.423	201.178	1.931	457.431	0.050
5	12.11	408.206	388.425	204.988	1.918	455.418	0.050
6	11.41	411.366	387.389	208.745	1.873	453.373	0.050
7	10.70	414.706	332.979	211.895	1.578	450.578	0.050
8	9.85	420.681	363.858	227.893	1.607	446.607	0.050
9	9.00	449.877	472.606	249.966	1.910	443.910	0.050
10	8.15	504.980	388.055	264.734	1.479	440.479	0.050
11	7.29	583.991	853.977	291.944	2.986	438.486	0.050
12	6.20	720.620	997.991	313.024	3.256	437.756	0.050
13	5.11	876.820	1178.795	339.199	3.550	437.050	0.050
14	4.01	1012.046	1383.073	360.682	3.920	436.420	0.050
15	2.92	1083.676	1065.929	379.107	2.846	435.346	0.050
16	1.95	1101.682	4047.381	632.795	6.513	435.013	0.050
17	0.97	1103.287	9135.556	889.904	10.501	435.001	0.050
18	0.00	1103.496	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3 TRIBUTARY UPSTREAM OF REREG DAM
Time: Year =1994,Month= 6 ,Day 18,Hour = 11.9883 Step= 96 Segment= 3
Upstream Boundary 250.000 Downstream Boundary = 6.766

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	808.539	202.562	4.260	492.760	0.457
2	2.03	250.616	235.612	146.255	1.646	489.646	0.040
3	1.47	252.481	685.197	161.878	4.495	489.295	0.040
4	0.90	255.043	1282.354	182.774	7.770	489.277	0.040
5	0.37	257.974	1517.831	236.373	6.770	489.270	0.050
6	0.00	260.575	1801.329	282.475	6.766	489.266	0.050

SAMPLE RIVIH DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1 MAIN STEM ABOVE REREG DAM
Time: Year =1983,Month= 6 ,Day 18,Hour = 17.9824 Step= 144 Segment= 1
Upstream Boundary 500.000 Downstream Boundary = 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	500.000	6037.277	803.222	8.086	503.086	0.423
2	27.00	503.754	1468.937	208.299	8.132	502.132	0.519
3	26.01	505.057	155.205	99.700	1.727	493.727	0.050
4	25.25	639.965	802.729	202.401	4.144	489.144	0.050
5	24.43	772.217	404.849	246.983	1.663	485.663	0.050

Figure 15. (Sheet 5 of 14)

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
6	23.59	751.890	1325.733	315.927	4.305	483.305	0.050
7	22.76	711.044	3414.179	384.546	9.233	483.233	0.050
8	21.82	668.152	2365.335	246.782	11.216	483.216	0.050
9	21.13	643.514	2608.519	240.684	11.203	483.203	0.050
10	20.33	615.484	3094.799	245.111	13.194	483.194	0.050
11	19.54	586.849	3594.939	250.341	15.190	483.190	0.050
12	18.74	557.346	4081.038	255.275	17.137	483.137	0.329
13	17.94	526.621	4732.629	263.104	19.593	483.093	0.050
14	17.06	489.555	5775.900	296.330	22.592	483.092	0.050
15	16.18	447.673	7011.418	337.956	25.591	483.091	0.050
16	15.29	400.000	8283.770	384.782	28.091	483.091	0.050

OUTPUT FOR SEGMENT NO.
Time: Year =1983, Month= June, Day 18, Hour = 17.9824 Step= 144 Segment= 2 MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000 Downstream Boundary = 15.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.989	237.855	6.224	461.224	0.482
2	14.47	400.101	779.946	217.916	3.733	458.733	0.050
3	13.65	400.210	708.361	211.827	3.457	458.457	0.050
4	12.82	400.294	378.861	201.121	1.913	457.413	0.050
5	12.11	400.375	383.884	204.930	1.896	455.396	0.050
6	11.41	400.568	380.167	208.676	1.838	453.338	0.050
7	10.70	400.598	327.746	211.865	1.554	450.554	0.050
8	9.85	400.899	350.684	227.788	1.549	443.549	0.050
9	9.00	423.870	464.967	249.886	1.879	443.879	0.050
10	8.15	469.089	364.300	264.446	1.389	440.389	0.050
11	7.29	518.309	747.455	290.481	2.620	438.120	0.050
12	6.20	587.264	838.185	310.975	2.744	437.244	0.050
13	5.11	662.030	978.382	336.826	2.957	436.457	0.050
14	4.01	710.826	1184.402	358.469	3.367	435.867	0.050
15	2.92	722.919	1002.589	378.572	2.679	435.179	0.050
16	1.95	726.330	4042.922	632.772	6.506	435.006	0.050
17	0.97	726.462	9135.090	889.903	10.501	435.001	0.050
18	0.00	726.015	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO.
Time: Year =1983, Month= June, Day 18, Hour = 17.9824 Step= 144 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM
Upstream Boundary 250.000 Downstream Boundary = 6.643

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	807.858	202.542	4.257	492.757	0.457
2	2.03	250.709	225.421	145.990	1.576	489.576	0.040
3	1.47	252.746	665.898	161.376	4.375	489.175	0.040
4	0.90	255.456	1260.307	182.217	7.656	489.156	0.040
5	0.37	258.671	1489.218	235.935	6.649	489.149	0.050
6	0.00	262.421	1766.984	281.887	6.644	489.144	0.050

SAMPLE RIVIH DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

Figure 15. (Sheet 6 of 14)

OUTPUT FOR SEGMENT NO. 1
Time: Year =1983,Month= June,Day 18,Hour = 23.9766 Step= 192 Segment= 1 MAIN STEM ABOVE REREG DAM
Upstream Boundary 600.000 Downstream Boundary = 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	600.000	6469.535	810.722	8.622	503.622	0.409
2	27.00	581.684	1566.004	211.444	8.594	502.594	0.514
3	26.01	575.631	172.609	101.670	1.900	493.900	0.050
4	25.25	703.882	791.563	202.169	4.089	489.089	0.050
5	24.43	826.821	443.757	247.643	1.820	485.820	0.050
6	23.59	799.323	1515.276	318.139	4.903	483.903	0.050
7	22.76	751.630	3650.855	386.510	9.847	483.847	0.050
8	21.82	702.514	2518.396	250.720	11.831	483.831	0.050
9	21.13	674.461	2757.133	241.547	11.819	483.819	0.050
10	20.33	642.481	3246.320	246.098	13.811	483.811	0.050
11	19.54	609.652	3749.782	251.452	15.807	483.807	0.322
12	18.74	576.092	4240.143	256.518	17.759	483.759	0.050
13	17.94	541.615	4897.647	264.480	20.218	483.718	0.050
14	17.06	500.301	5961.975	298.582	23.217	483.717	0.050
15	16.18	453.434	7223.820	341.084	26.217	483.717	0.050
16	15.29	400.000	8525.738	388.786	28.717	483.717	0.050

OUTPUT FOR SEGMENT NO. 2
Time: Year =1983,Month= June,Day 18,Hour = 23.9766 Step= 192 Segment= 2 MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000 Downstream Boundary = 15.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.916	237.854	6.224	461.224	0.482
2	14.47	399.998	779.744	217.912	3.732	458.732	0.050
3	13.65	399.996	708.153	211.824	3.456	458.456	0.050
4	12.82	400.019	378.666	201.118	1.912	457.412	0.050
5	12.11	400.028	383.637	204.927	1.895	455.395	0.050
6	11.41	400.039	379.765	208.672	1.836	453.336	0.050
7	10.70	400.049	327.480	211.863	1.553	450.553	0.050
8	9.85	400.072	349.962	227.783	1.546	446.546	0.050
9	9.00	422.568	464.580	249.882	1.878	443.878	0.050
10	8.15	467.596	362.999	264.430	1.384	440.384	0.050
11	7.29	512.930	735.426	290.316	2.579	438.079	0.050
12	6.20	572.032	815.301	310.681	2.670	437.170	0.050
13	5.11	631.954	946.030	336.442	2.861	436.361	0.050
14	4.01	663.504	1152.399	358.112	3.278	435.778	0.050
15	2.92	665.184	993.450	378.495	2.655	435.155	0.050
16	1.95	665.462	4042.318	632.769	6.505	435.005	0.050
17	0.97	665.491	9134.990	889.902	10.501	435.001	0.050
18	0.00	665.408	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1983,Month= June,Day 18,Hour = 23.9766 Step= 192 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM
Upstream Boundary 250.000 Downstream Boundary = 6.588

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	807.602	202.535	4.256	492.756	0.457
2	2.03	250.330	220.808	145.870	1.545	489.545	0.040
3	1.47	250.952	656.996	161.144	4.320	489.120	0.040

Figure 15. (Sheet 7 of 14)

4	0.90	251.768	1250.209	181.962	7.601	489.101	0.040
5	0.37	253.114	1476.156	235.735	6.594	489.094	0.050
6	0.00	254.946	1751.432	281.622	6.589	489.089	0.050

SAMPLE RIV1H DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1983,Month= June,Day 19,Hour = 5.9707 Step= 240 Segment= 1 MAIN STEM ABOVE REREG DAM
Upstream Boundary 700.000 Downstream Boundary = 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	700.000	6815.166	816.670	9.047	504.047	0.399
2	27.00	648.451	1642.557	213.892	8.954	502.954	0.510
3	26.01	631.260	186.857	103.254	2.039	494.039	0.050
4	25.25	755.860	763.517	201.585	3.950	488.950	0.050
5	24.43	874.420	497.337	248.550	2.036	486.036	0.050
6	23.59	842.047	1737.078	320.708	5.597	484.597	0.050
7	22.76	789.151	3925.346	388.776	10.555	484.555	0.050
8	21.82	734.220	2698.007	255.263	12.541	484.541	0.050
9	21.13	702.608	2929.256	242.543	12.530	484.530	0.050
10	20.33	667.364	3421.936	247.237	14.523	484.523	0.050
11	19.54	631.512	3929.367	252.734	16.519	484.519	0.050
12	18.74	594.818	4424.748	257.953	18.477	484.477	0.315
13	17.94	557.026	5089.187	266.068	20.940	484.440	0.050
14	17.06	511.348	6178.531	301.182	23.939	484.439	0.050
15	16.18	459.336	7471.442	344.695	26.939	484.439	0.050
16	15.29	400.000	8808.180	393.408	29.439	484.439	0.050

OUTPUT FOR SEGMENT NO. 2
Time: Year =1983,Month= June,Day 19,Hour = 5.9707 Step= 240 Segment= 2 MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000 Downstream Boundary = 15.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.913	237.854	6.224	461.224	0.482
2	14.47	400.009	779.732	217.912	3.732	458.732	0.050
3	13.65	399.997	708.141	211.823	3.456	458.456	0.050
4	12.82	400.005	378.655	201.118	1.912	457.412	0.050
5	12.11	399.996	383.619	204.926	1.895	455.395	0.050
6	11.41	400.004	379.741	208.672	1.836	453.336	0.050
7	10.70	400.007	327.465	211.863	1.552	450.552	0.050
8	9.85	400.001	349.918	227.782	1.546	446.546	0.050
9	9.00	422.496	464.559	249.881	1.877	443.877	0.050
10	8.15	467.476	362.913	264.429	1.384	440.384	0.050
11	7.29	512.507	734.119	290.298	2.575	438.075	0.050
12	6.20	570.365	812.575	310.645	2.661	437.161	0.050
13	5.11	628.355	941.955	336.394	2.848	436.348	0.050
14	4.01	657.504	1148.341	358.067	3.267	435.767	0.050
15	2.92	657.749	992.317	378.485	2.652	435.152	0.050
16	1.95	657.786	4042.252	632.768	6.505	435.005	0.050
17	0.97	657.739	9135.000	889.902	10.501	435.001	0.050
18	0.00	657.773	16972.500	1163.000	15.000	435.000	0.050

Figure 15. (Sheet 8 of 14)

OUTPUT FOR SEGMENT NO. 3
Time: Year -1983,Month- June,Day 19,Hour = 5.9707 Step= 240 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM
Upstream Boundary 250.000 Downstream Boundary = 6.449

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	807.410	202.529	4.255	492.755	0.457
2	2.03	250.230	210.520	145.602	1.474	489.474	0.040
3	1.47	250.614	635.053	160.571	4.184	488.984	0.040
4	0.90	250.985	1225.086	181.326	7.463	488.963	0.040
5	0.37	251.668	1443.487	235.236	6.455	488.955	0.050
6	0.00	252.700	1712.349	280.955	6.450	488.950	0.050

SAMPLE RIVIH DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year -1983,Month- June,Day 19,Hour = 11.9648 Step= 288 Segment= 1 MAIN STEM ABOVE REREG DAM
Upstream Boundary 800.000 Downstream Boundary - 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	800.000	7339.452	825.579	9.685	504.685	0.383
2	27.00	752.053	1766.164	217.777	9.527	503.527	0.505
3	26.01	736.288	214.553	106.257	2.304	494.304	0.050
4	25.25	858.924	755.038	201.418	3.908	488.908	0.050
5	24.43	971.479	594.169	250.183	2.424	486.424	0.050
6	23.59	930.504	1995.246	323.675	6.398	485.398	0.050
7	22.76	867.218	4240.813	391.365	11.364	485.364	0.050
8	21.82	801.073	2906.700	260.444	13.351	485.351	0.050
9	21.13	762.894	3126.124	243.676	13.340	485.340	0.050
10	20.33	720.346	3622.710	248.533	15.333	485.333	0.050
11	19.54	677.018	4134.665	254.192	17.329	485.329	0.050
12	18.74	632.879	4635.123	259.579	19.290	485.290	0.307
13	17.94	587.676	5306.887	267.862	21.756	485.256	0.050
14	17.06	533.200	6425.340	304.117	24.755	485.255	0.050
15	16.18	471.103	7754.207	348.772	27.754	485.254	0.050
16	15.29	400.000	9131.150	398.627	30.254	485.254	0.050

OUTPUT FOR SEGMENT NO. 2
Time: Year -1983,Month- June,Day 19,Hour = 11.9648 Step= 288 Segment= 2 MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000 Downstream Boundary - 15.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.914	237.854	6.224	461.224	0.482
2	14.47	400.003	779.730	217.911	3.732	458.732	0.050
3	13.65	400.000	708.140	211.823	3.456	458.456	0.050
4	12.82	399.997	378.654	201.118	1.912	457.412	0.050
5	12.11	399.997	383.620	204.926	1.895	455.395	0.050
6	11.41	400.004	379.739	208.672	1.836	453.336	0.050
7	10.70	399.998	327.463	211.863	1.552	450.552	0.050
8	9.85	400.002	349.913	227.782	1.546	446.546	0.050
9	9.00	422.491	464.561	249.881	1.877	443.877	0.050
10	8.15	467.468	362.908	264.429	1.384	440.384	0.050
11	7.29	512.449	733.983	290.296	2.574	438.074	0.050
12	6.20	570.200	812.260	310.641	2.660	437.160	0.050
13	.5.11	627.944	941.491	336.388	2.847	436.347	0.050

Figure 15. (Sheet 9 of 14)

124

Figure 15. (Sheet 10 of 14)

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
14	4.01	656.849	1147.866	358.061	3.265	435.765	0.050
15	2.92	656.821	992.193	378.484	2.651	435.151	0.050
16	1.95	656.876	4042.244	632.768	6.505	435.005	0.050
17	0.97	656.934	9135.009	889.902	10.501	435.001	0.050
18	0.00	656.839	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1983,Month=   June,Day 19,Hour = 11.9648 Step= 288 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM
Upstream Boundary 250.000   Downstream Boundary = 6.409

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	807.375	202.528	4.255	492.755	0.457
2	2.03	249.873	207.499	145.523	1.453	489.453	0.040
3	1.47	249.328	628.468	160.401	4.143	488.943	0.040
4	0.90	248.660	1217.544	181.138	7.421	488.921	0.040
5	0.37	248.383	1433.654	235.090	6.413	488.913	0.050
6	0.00	249.002	1700.534	280.763	6.408	488.908	0.050

SAMPLE RIVIH DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1983,Month=   June,Day 19,Hour = 17.9590 Step= 336 Segment= 1 MAIN STEM ABOVE REREG DAM
Upstream Boundary 700.000   Downstream Boundary = 400.000

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	700.000	7145.934	822.373	9.450	504.450	0.389
2	27.00	724.430	1724.123	216.495	9.333	503.333	0.507
3	26.01	731.539	212.938	106.110	2.289	494.289	0.050
4	25.25	873.461	724.108	200.788	3.754	488.754	0.050
5	24.43	1002.207	718.881	252.225	2.921	486.921	0.050
6	23.59	958.728	2294.042	327.002	7.317	486.317	0.050
7	22.76	891.659	4605.500	394.275	12.292	486.292	0.050
8	21.82	823.353	3151.946	266.281	14.282	486.282	0.050
9	21.13	784.018	3354.093	244.956	14.273	486.273	0.050
10	20.33	740.303	3855.647	249.998	16.268	486.268	0.050
11	19.54	695.447	4373.167	255.841	18.264	486.264	0.050
12	18.74	649.284	4880.416	261.424	20.231	486.231	0.298
13	17.94	601.329	5561.520	269.903	22.703	486.203	0.050
14	17.06	543.007	6714.978	307.457	25.702	486.202	0.050
15	16.18	476.301	8086.791	353.411	28.702	486.202	0.050
16	15.29	400.000	9511.580	404.565	31.201	486.201	0.050

OUTPUT FOR SEGMENT NO. 2
Time: Year =1983,Month=   June,Day 19,Hour = 17.9590 Step= 336 Segment= 2 MAIN STEM BELOW REREG DAM
Upstream Boundary 400.000   Downstream Boundary = 15.000

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.912	237.854	6.224	461.224	0.482
2	14.47	400.004	779.731	217.911	3.732	458.732	0.050
3	13.65	400.021	708.140	211.823	3.456	458.456	0.050
4	12.82	400.010	378.655	201.118	1.912	457.412	0.050
5	12.11	399.995	383.619	204.926	1.895	455.395	0.050
6	11.41	400.000	379.740	208.672	1.836	453.336	0.050

Figure 15. (Sheet 10 of 14)

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
7	10.70	399.979	327.463	211.863	1.552	450.552	0.050
8	9.85	399.999	349.913	227.782	1.546	446.546	0.050
9	9.00	422.490	464.560	249.881	1.877	443.877	0.050
10	8.15	467.454	362.907	264.429	1.384	440.384	0.050
11	7.29	512.443	733.961	290.296	2.574	438.074	0.050
12	6.20	570.171	812.233	310.641	2.660	437.160	0.050
13	5.11	627.921	941.436	336.387	2.847	436.347	0.050
14	4.01	656.734	1147.819	358.061	3.265	435.765	0.050
15	2.92	656.790	992.175	378.484	2.651	435.151	0.050
16	1.95	656.812	4042.252	632.768	6.505	435.005	0.050
17	0.97	656.728	9135.003	889.902	10.501	435.001	0.050
18	0.00	656.770	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1983,Month= June,Day 19,Hour = 17.9590 Step= 3
Upstream Boundary 250.000 Downstream Boundary = 6.259

Segment= 336 3 TRIBUTARY UPSTREAM OF REREG DAM

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	807.908	202.543	4.257	492.757	0.457
2	2.03	250.911	198.446	145.295	1.391	489.391	0.040
3	1.47	255.134	604.868	159.805	3.995	488.795	0.040
4	0.90	261.608	1190.172	180.467	7.270	488.770	0.050
5	0.37	268.209	1397.745	234.557	6.260	488.760	0.050
6	0.00	273.579	1657.404	280.044	6.254	488.754	0.050

SAMPLE RIV1H DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1983,Month= June,Day 19,Hour = 23.9531 Step= 384
Upstream Boundary 600.000 Downstream Boundary = 400.000

Segment= 1 1 MAIN STEM ABOVE REREG DAM

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	600.000	6868.996	817.586	9.113	504.113	0.397
2	27.00	665.664	1662.788	214.535	9.048	503.048	0.510
3	26.01	687.491	203.514	105.094	2.199	494.199	0.050
4	25.25	819.666	704.682	200.358	3.657	488.657	0.050
5	24.43	933.983	855.473	254.528	3.460	487.460	0.050
6	23.59	892.630	2566.728	330.141	8.147	487.147	0.050
7	22.76	833.393	4937.453	397.019	13.131	487.131	0.050
8	21.82	772.614	3378.518	271.790	15.124	487.124	0.050
9	21.13	737.586	3561.413	246.164	15.117	487.117	0.050
10	20.33	698.930	4067.653	251.381	17.113	487.113	0.050
11	19.54	659.438	4590.384	257.399	19.111	487.111	0.050
12	18.74	618.885	5104.674	263.172	21.086	487.086	0.289
13	17.94	576.883	5795.089	271.842	23.565	487.065	0.050
14	17.06	525.858	6981.499	310.631	26.564	487.064	0.050
15	16.18	467.250	8393.503	357.820	29.564	487.064	0.050
16	15.29	400.000	9862.969	410.208	32.064	487.064	0.050

Figure 15. (Sheet 11 of 14)

126

OUTPUT FOR SEGMENT NO. 2
Time: Year =1983,Month= June,Day 19,Hour = 23.9531 Step= 384 Segment= 2 MAIN STEM BELOW REREG DAM

River Mile	Flow (cfs)	Upstream Boundary 400.000 / Downstream Boundary = 15.000 Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.913	237.854	6.224	461.224	0.482
2	14.47	399.991	779.732	217.911	3.732	458.732	0.050
3	13.65	400.007	708.140	211.823	3.456	458.456	0.050
4	12.82	400.001	378.654	201.118	1.912	457.412	0.050
5	12.11	399.995	383.619	204.926	1.895	455.395	0.050
6	11.41	399.998	379.741	208.672	1.836	453.336	0.050
7	10.70	399.996	327.460	211.863	1.552	450.552	0.050
8	9.85	399.998	349.915	227.782	1.546	446.546	0.050
9	9.00	422.491	464.558	249.881	1.877	443.877	0.050
10	8.15	467.475	362.909	264.429	1.384	440.384	0.050
11	7.29	512.446	733.959	290.296	2.574	438.074	0.050
12	6.20	570.178	812.226	310.641	2.660	437.160	0.050
13	5.11	627.879	941.433	336.387	2.847	436.347	0.050
14	4.01	656.759	1147.808	358.061	3.265	435.765	0.050
15	2.92	656.754	992.176	378.484	2.651	435.151	0.050
16	1.95	656.586	4042.261	632.768	6.505	435.005	0.050
17	0.97	656.738	9134.981	889.902	10.501	435.001	0.050
18	0.00	657.011	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1983,Month= June,Day 19,Hour = 23.9531 Step= 384 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM

River Mile	Flow (cfs)	Upstream Boundary 250.000 / Downstream Boundary = 6.157 Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	808.196	202.552	4.259	492.759	0.457
2	2.03	250.046	192.419	145.129	1.350	489.350	0.040
3	1.47	250.476	589.533	159.376	3.899	488.699	0.040
4	0.90	251.604	1172.712	179.993	7.173	488.673	0.040
5	0.37	253.550	1375.033	234.187	6.163	488.663	0.050
6	0.00	255.060	1630.302	279.552	6.157	488.657	0.050

SAMPLE RIV1H DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1983,Month= June,Day 20,Hour = 5.9473 Step= 432 Segment= 1 MAIN STEM ABOVE REREG DAM

River Mile	Flow (cfs)	Upstream Boundary 400.000 / Downstream Boundary = 400.000 Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	400.000	5836.009	799.679	7.835	502.835	0.429
2	27.00	472.031	1427.272	206.926	7.931	501.931	0.521
3	26.01	494.914	156.457	99.829	1.740	493.740	0.050
4	25.25	627.828	684.413	199.940	3.556	488.556	0.050
5	24.43	747.702	978.861	256.560	3.943	487.943	0.050
6	23.59	720.941	2785.600	332.586	8.807	487.807	0.050
7	22.76	683.012	5203.451	399.158	13.799	487.799	0.050
8	21.82	643.970	3562.510	276.089	15.795	487.795	0.050

Figure 15. (Sheet 12 of 14)

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
9	21.13	621.568	3727.709	247.108	15.792	487.792	0.050
10	20.33	596.831	4237.978	252.463	17.789	487.789	0.050
11	19.54	571.210	4765.084	258.618	19.788	487.788	0.282
12	18.74	544.770	5285.540	264.543	21.772	487.772	0.050
13	17.94	517.201	5983.929	273.367	24.258	487.758	0.050
14	17.06	483.541	7197.600	313.126	27.257	487.757	0.050
15	16.18	444.759	8642.676	361.285	30.257	487.757	0.050
16	15.29	400.000	10148.783	414.644	32.757	487.757	0.050

OUTPUT FOR SEGMENT NO. 2
Time: Year =1983,Month= 432 Segment= 2 MAIN STEM BELOW REREG DAM
June,Day 20,Hour = 5.9473 Step= 15.000
Upstream Boundary 400.000 Downstream Boundary 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	15.29	400.000	1371.913	237.854	6.224	461.224	0.482
2	14.47	399.990	779.731	217.911	3.732	458.732	0.050
3	13.65	400.013	708.141	211.823	3.456	458.456	0.050
4	12.82	400.000	378.653	201.118	1.912	457.412	0.050
5	12.11	400.002	383.621	204.926	1.895	455.395	0.050
6	11.41	400.000	379.741	208.672	1.836	453.336	0.050
7	10.70	399.995	327.460	211.863	1.552	450.552	0.050
8	9.85	400.004	349.915	227.782	1.546	446.546	0.050
9	9.00	422.491	464.561	249.881	1.877	443.877	0.050
10	8.15	467.465	362.910	264.429	1.384	440.384	0.050
11	7.29	512.452	733.959	290.296	2.574	438.074	0.050
12	6.20	570.172	812.226	310.641	2.660	437.160	0.050
13	5.11	627.873	941.432	336.387	2.847	436.347	0.050
14	4.01	656.771	1147.812	358.061	3.265	435.765	0.050
15	2.92	656.734	992.169	378.484	2.651	435.151	0.050
16	1.95	656.672	4042.250	632.768	6.505	435.005	0.050
17	0.97	656.735	9134.993	889.902	10.501	435.001	0.050
18	0.00	656.810	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3
Time: Year =1983,Month= 432 Segment= 3 TRIBUTARY UPSTREAM OF REREG DAM
June,Day 20,Hour = 5.9473 Step= 6.057
Upstream Boundary 250.000 Downstream Boundary

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	2.51	250.000	808.799	202.570	4.262	492.762	0.457
2	2.03	250.256	187.212	144.993	1.314	489.314	0.040
3	1.47	251.520	573.789	158.963	3.800	488.600	0.040
4	0.90	253.787	1154.546	179.532	7.072	488.572	0.040
5	0.37	256.213	1351.328	233.827	6.062	488.562	0.040
6	0.00	257.451	1602.012	279.075	6.056	488.556	0.050

SAMPLE RIV1H DATA SET - USER'S MANUAL
TIME STEP (sec) = 450.00

OUTPUT FOR SEGMENT NO. 1
Time: Year =1983,Month= 480 Segment= 1 MAIN STEM ABOVE REREG DAM
June,Day 20,Hour = 11.9414 Step= 400.000
Upstream Boundary 400.000 Downstream Boundary 400.000

	River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
1	27.76	400.000	5508.740	793.955	7.424	502.424	0.439
2	27.00	418.068	1349.118	204.349	7.551	501.551	0.524
3	26.01	423.428	135.791	97.457	1.530	493.530	0.050

Figure 15. (Sheet 13 of 14)

128

Rows 4–16 (continuation of Segment 1 output):

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
25.25	537.259	705.410	200.371	3.661	488.661	0.050
24.43	640.968	1068.626	258.023	4.291	488.291	0.050
23.59	619.277	2921.043	334.089	9.213	488.213	0.050
22.76	592.766	5366.903	400.466	14.208	488.208	0.050
21.82	565.479	3676.221	278.712	16.205	488.205	0.050
21.13	549.715	3829.322	247.683	16.202	488.202	0.050
20.33	532.653	4341.933	253.121	18.201	488.201	0.050
19.54	515.204	4871.656	259.359	20.199	488.199	0.278
18.74	497.200	5395.590	265.374	22.187	488.187	0.050
17.94	478.518	6098.571	274.288	24.676	488.176	0.050
17.06	455.889	7329.021	314.633	27.676	488.176	0.050
16.18	429.964	8794.365	363.378	30.676	488.176	0.050
15.29	400.000	10322.950	417.323	33.176	488.176	0.050

OUTPUT FOR SEGMENT NO. 2   MAIN STEM BELOW REREG DAM
Time: Year –1983, Month= June, Day 20, Hour = 11.9414 Step= 480 Segment=
Upstream Boundary 400.000 Downstream Boundary = 15.000

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
15.29	400.000	1371.910	237.854	6.224	461.224	0.482
14.47	399.990	779.733	217.912	3.732	458.732	0.050
13.65	399.995	708.140	211.823	3.456	458.456	0.050
12.82	400.005	378.655	201.118	1.912	457.412	0.050
12.11	399.998	383.617	204.926	1.895	455.395	0.050
11.41	399.996	379.741	208.672	1.836	453.336	0.050
10.70	399.974	327.458	211.863	1.552	450.552	0.050
9.85	399.980	349.912	227.782	1.546	446.546	0.050
9.00	422.509	464.565	249.881	1.877	443.877	0.050
8.15	467.534	362.913	264.429	1.384	440.384	0.050
7.29	512.407	733.958	290.296	2.574	438.074	0.050
6.20	570.161	812.221	310.641	2.660	437.160	0.050
5.11	627.885	941.435	336.387	2.847	436.347	0.050
4.01	656.724	1147.809	358.061	3.265	435.765	0.050
2.92	656.785	992.166	378.484	2.651	435.151	0.050
1.95	656.575	4042.257	632.768	6.505	435.005	0.050
0.97	656.689	9134.989	889.902	10.501	435.001	0.050
0.00	656.948	16972.500	1163.000	15.000	435.000	0.050

OUTPUT FOR SEGMENT NO. 3   TRIBUTARY UPSTREAM OF REREG DAM
Time: Year –1983, Month= June, Day 20, Hour = 11.9414 Step= 480 Segment=
Upstream Boundary 250.000 Downstream Boundary = 6.160

River Mile	Flow (cfs)	Area (sq ft)	Width (Feet)	Depth (Feet)	WSEL (Feet)	Mannings n
2.51	250.000	808.104	202.549	4.258	492.758	0.457
2.03	249.427	192.062	145.119	1.347	489.347	0.040
1.47	246.437	589.467	159.374	3.899	488.699	0.040
0.90	241.424	1172.918	179.998	7.174	488.674	0.040
0.37	235.652	1375.666	234.196	6.166	488.666	0.050
0.00	231.608	1631.325	279.567	6.161	488.661	0.050

Figure 15. (Sheet 14 of 14)

corresponding to that flow, channel geometry, and bed roughness. If problems are still encountered starting the model, the use of small time-steps, at least initially, has helped. After a steady-state flow and water surface profile have been achieved, it is best to use these results to start subsequent simulations.

228. Phase errors (see Constant Card, above), which may accompany unsteady flow waves, can result in negative depths (usually in or near shallow riffle sections). If this happens, a fatal execution error occurs, and the program is stopped. Several things can be done to eliminate this problem. First, make sure that the cross-sectional area is adequately described, especially at low stage. Next, check the value of THETA (see Constant Card, above); higher value (e.g. 0.75) may help. If problems are still encountered, it may be necessary to change the time-step such that the surface wave Courant number is closer to 1.0. As values of this number decrease below or increase above 1.0, phase errors become worse (Lyn and Goodwin 1987, Liggett and Cunge 1975). The surface Courant number is defined as

$$C_N = \frac{U_s \Delta t}{\Delta x} \qquad (205)$$

where

U_s = speed of the surface wave = $\sqrt{gd}$

d = water depth

Δt = time-step size

Δx = spatial step size

The same result can be achieved by adjustment of the spatial steps. If these measures are not successful, ramping of the inflow hydrograph to yield a gentler, longer wave will usually reduce phase errors since the unsteady flow waves are more spread out. As an example, hydropower turbines can come up to full power in about 5 min; peaking hydropower releases can result in highly unsteady flow waves in the receiving stream. It may be necessary to spread out the generation start-up over 15 to 30 min. Judicious smoothing of the inflow hydrograph can be accomplished without jeopardizing study objectives.

130

229. The water quality program contains numerous subroutines in addition to the main program. There are also two files linked to the main program and subroutines using INCLUDE statements. The primary INCLUDE file contains a PARTITION statement, which allows flexible redimensioning of the model. The main program sets up the flexible dimensioning of arrays, as is done in the hydrodynamic program, and handles all input and output and the organization of the tributary structure.

230. CE-QUAL-RIV1 expects the user to create an execution control file (RIV1Q.CTL) prior to running RIV1Q. The control file must be in the same directory as the RIV1Q executable and as such consist of standard ASCII text. This file follows the same syntax as the control file for RIV1H discussed in Part VI. The first line is the name of the input file, the second line is the hydrodynamic file from a previous RIV1H simulation, the third line is the lateral flow file, and the fourth is the meteorological data file. The order in which this information is entered cannot be changed. A listing of RIV1Q.CTL is given below.

```
INPUT FILE      SAMPLEQ.INP
HYDRO FILE      SAMPLEH.HYD
LATERAL FILE    SAMPLEH.LAC
MET DATA FILE   SAMPLEQ.MET
```

231. As noted in Part IV, RIV1Q achieves its fourth-order accuracy by advecting derivatives as well as concentrations. Initial concentrations are given by the input data, but initial derivatives must be estimated from these data. The main program calls subroutine SPLINE to pass a cubic spline through the data points, from which this estimate is obtained. The algorithm for constructing a cubic spline produces a tridiagonal matrix, which is solved by subroutine TRIDAG. At each step of the time march, the main program proceeds through the tributary network in a downstream order (see Tributaries, Part VI), assigning appropriate boundary conditions for each segment and calling subroutine SEG to complete the solution.

232. The major loops of the program are, in order of nesting, the time, segment, node, and constituent. The program handles up to a maximum of 12 constituents: temperature (TEMP), CBOD, organic nitrogen (ORGAN), ammonia nitrogen (NH3-N), nitrate nitrogen (NO3-N), organic phosphate, dissolved phosphate (PO4), dissolved iron, dissolved manganese, dissolved oxygen (DO),

coliform bacteria, and algae. In addition, the effects of macrophytes are simulated, although macrophytes is not a state variable. Minor loops of the program cycle through segments, nodes, or constituents as appropriate. A flowchart is shown in Figure 16.
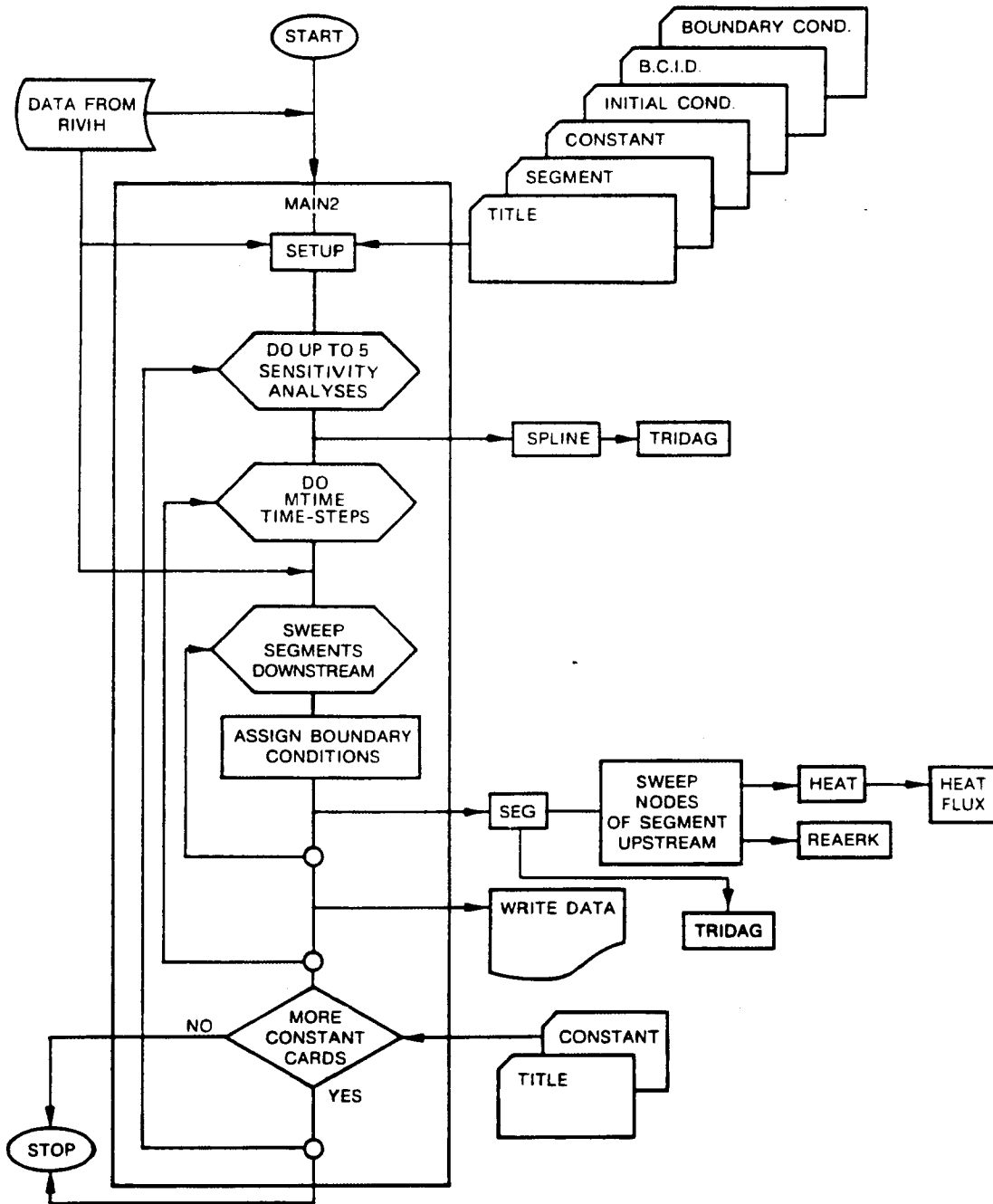
Figure 16. RIV1Q organizational flowchart

233. Structured as a guided tour, this section is best read concurrently with the listing of the code in Appendix B.

The main program

234. The main program opens the control file (RIV1Q.CTL) and determines the names of the various input files. The main program in RIV1Q serves to apportion storage in the same way as it does in RIV1H. The major difference is that whereas the grid information for the hydrodynamic program comes from the input deck, the water quality model obtains its grid information from the hydrodynamic model. Variables that determine the total array storage required by the program are: IND1, the total number of computational nodes; IBRAN, the total number of branches; IUPDATE, the number of time-varying updates; and ISYS, the number of water quality constituents. These variables are defined in the PARAMETER statement of the main INCLUDE file. The main program performs the bulk of the work for the rest of the program.

235. Because of all the K-rates used, all variables whose names begin with the letter "K" are implicitly declared real. The array CP, an array of flags, indicates whether each of the 12 constituents is to be simulated or held constant at its initial value. The DATA statements assign default values to parameters when reasonable, zero otherwise.

236. The first item in the main input stream is the title card. Then, the program reads a series of global constants, applicable everywhere in the model domain, followed by input which determines the frequency of printing, and then bypass options. The bypass options, stored in the CP array, determine whether a particular constituent is to be simulated or held constant at its initial conditions (0 simulated, 1 not simulated). Then for each segment, the program reads the segment card, the constant cards, and initial conditions cards. The computational nodes are stored in a 1-D array, even though they are handled segment by segment. The node number is initialized at 1, and the segment loop is entered. The segment card is read, and then the constant card. On the constant card, the user indicates the name and value of any parameter whose default value is to be overridden. Most of these parameters can be stream-specific, but need not be.

237. Following the constant card are the initial conditions cards, one for each node in the segment. The initial concentrations are stored in the array INIT.

238. Not every segment receives boundary conditions from the input data stream; specifically, those segments below control structures receive their boundary conditions from the segment immediately above. Those segments that will receive their boundary conditions from the input data stream are indicated on the boundary conditions identification card in the order that the respective boundary conditions cards will appear. The filenames containing the names of the files containing the time-varying boundary condition data are then specified in input. The program then opens and reads these files, in subroutine TIME_VARYING_DATA.

239. The program refers to segments according to the order they appear in the input deck, but the user refers to each segment by its ID number. The logic must make the appropriate translation wherever such references occur. Similarly, it is convenient for the user to give the ID number of each of the segments for which boundary conditions are given; this information is stored in the array IBC. For the program, however, it is more convenient to know where the boundary conditions for each segment are to be found. This cross-reference type of information is given in the array JBC.

240. Distance increments (DX1), constant lateral inflows (QL), and river miles (RMILE) are time-invariant and so are read in from the hydro-dynamic model outside of the time-march loop. Flows, time-varying lateral inflows, areas, widths, and water surface elevations are read in from the hydrodynamic linkage file within the time-march loop.

241. The starting and ending dates for the simulation are specified in the input. The format for these dates is year, month, day, and hour. These dates are converted internally to Julian dates by the subroutine JULIAN_DAY and reconverted for output by the subroutine GREGORIAN_DAY. A "seed year" is also provided in the hydrodynamic linkage file for consistent computation of Julian dates between the water quality and hydrodynamic program. The date of the program execution is called so that it can be printed on the output.

242. All time-varying data are read in the subroutine TIME_VARYING_DATA. These may include meteorological data, time-varying lateral inflow concentrations, and boundary concentrations. The flows corresponding to the lateral inflow and boundary concentrations are taken from the hydrodynamic linkage files. The user must ensure that the data are consistent.

243. The running parameters of photoperiod and elapsed time are initialized. If the simulation begins during daylight hours, the sine of the incident light is calculated for use in algal productivity calculations.

244. As mentioned previously, the fourth-order method requires initial derivatives as well as concentrations, although the user would have no a priori knowledge about them. Subroutine SPLINE is called to estimate derivatives at each point based on a cubic spline through the data. The spline is constructed one segment at a time.

245. Hydrodynamic data for the initial conditions are read and converted from customary English to SI units. The program calls subroutine SEG to set initial rate constants. Since actual transport and decay processes are not calculated at this point (this is indicated to the subroutine by setting DT = 0.), boundary conditions are not passed, and the ordering of the segments is unimportant.

246. The main time-march loop begins with setting MBC, the index to the boundary conditions array, for this time-step. The program reads hydrodynamic data and sets the hydrodynamic array pointers. The solution algorithm in this fourth-order scheme is explicit, and so only one time level of information is required at any step.

247. To ensure proper passing of boundary conditions, subroutine SEG is called for each segment, working through the network in a downstream order. Boundary conditions are required for every segment, either from the input stream (JBC > 0) or passed down from another segment (JBC < 0). (JBC = 0 can happen only if boundary conditions are specified incorrectly. In this case, the program aborts.) In the first case, the program determines where in BOUND the boundary conditions for that segment are found and calls subroutine SEG with BOUND at that index. In the second case, the appropriate location in the tributary information transfer array T is given immediately by -JBC(L). The entry in the array T that stands for DO, i.e. T(10,L2), is adjusted for reaeration through control structures according to the energy dissipation model (Wilhelms and Smith 1981). The program then calls subroutine SEG with boundary conditions given in T.

248. After processing all of the segments, the program prints out the data for all the segments, one to a page, in the order that the segments are given in the main input file.

135

Subroutine SEG

249. Subroutine SEG performs the main decay rate and transport and decay calculations for each segment. In its argument list are the distance increments (RLENGTH), hydrodynamic data from the previous time-step (FLOWOLD, AREAOLD, WIDTHOLD, ELEVOLD), hydrodynamic data from the current time-step (FLOW, AREA, WIDTH, ELEV), constant (QLC) and time-varying (QLT) lateral inflows, constituent concentrations (C), spatial derivatives of the concentrations (DC), decay rates (K), source/sink terms (SINK), boundary conditions (BOUND), presence/absence flags (CP), constant lateral inflow concentrations (CLC), time-varying lateral inflow concentrations (CLT), the sine of the incident light (SINI), the number of nodes (NNODE), the index of the first tributary to that segment (IT0), the index of the last tributary (IT1), the index where this segment is to place its data (IT2), the tributary information transfer arrays (JT,T), the dipsersion coefficient (DISP), dispersion correction term (F), number of segments (NS), and flag for wind-driven reaeration (QWIND or QDUM depending on location).

250. First, a derivative at the first node is estimated with a cubic interpolation based on the concentration at the boundary at the new time-step and at the first and second nodes at the old time-step, and the derivative at the second node. Markers to define the downstream extent of algal nutrient depletion are initialized.

251. In this explicit method, only two nodes from the old time-step are required in the solution of any node at the current time-step (Figure 17); that is, α_i^{j+1} is calculated from α_i^j and α_{i-1}^j. If calculation were to proceed in a downstream fashion, α_{i-1}^j would have to be saved so it would not be overwritten in the calculation of α_{i-1}^{j+1}. Proceeding upstream, however, removes this difficulty. Entries at i-1 are old information by default, and α_i^{j+1} simply replaces α_i^j as it is calculated. For this reason, the node loop marches backwards through the indices.

252. Once inside the node march loop, velocities and the rate constant adjustments for flow, QX, are calculated. The succeeding hydrodynamic manipulations are useful only in the transport calculations and so are bypassed for initial and boundary conditions. For interior grid points, these statements prepare the groundwork for the solution of the final transport Equations 92, 95, and 98. The distance increment DX is assigned to $X_{i+1} - X_i$ (the node to be calculated is termed node I in the program, as opposed to node i+1 in
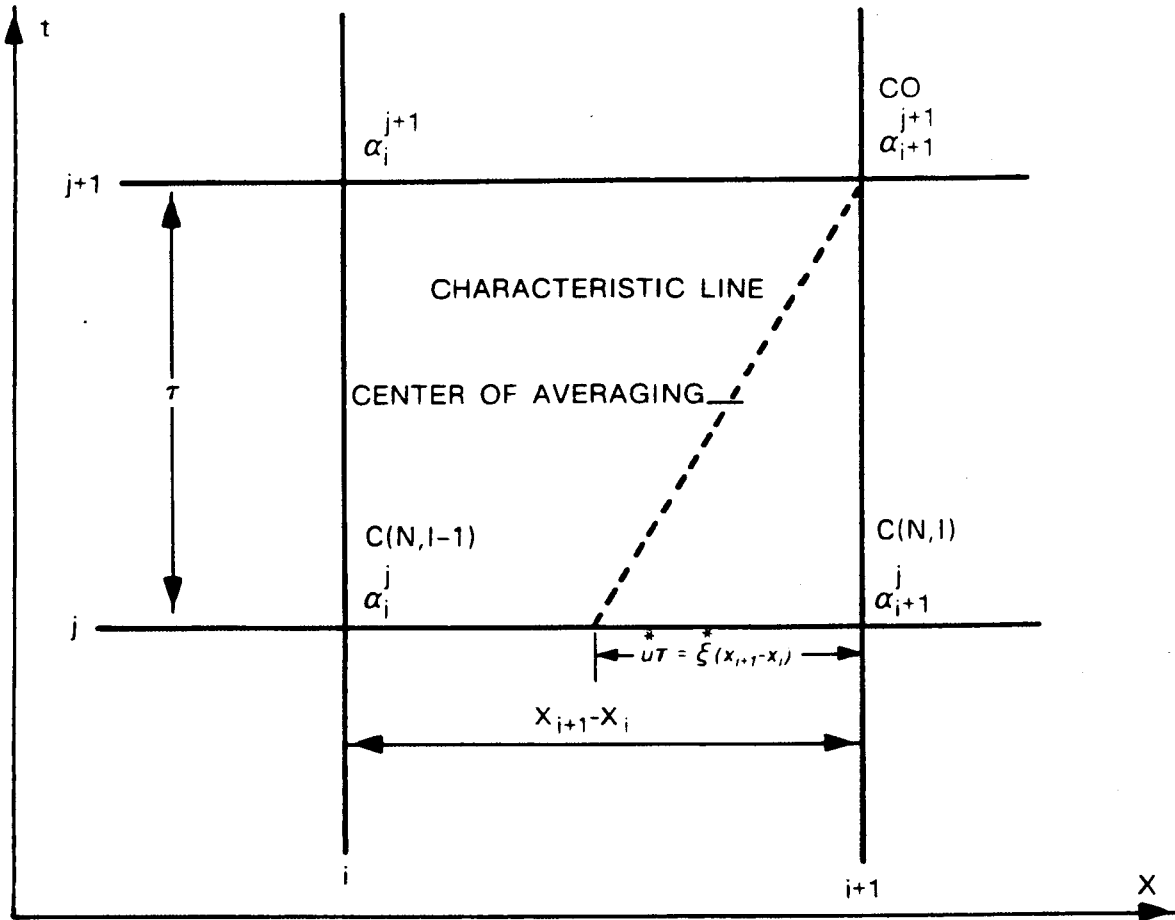
Figure 17. Compact fourth-order numerical gridwork

Part IV). For any parameter a , $\partial a/\partial x$ is calculated as $(a^j_{i+1} - a^j_i)/(X_{i+1} - X_i)$, $\partial a/\partial t$ is calculated as $(a^{j+1}_i - a^j_i)/\tau$, and the estimated average as

$$\overset{*}{a} = \frac{\left[a^{j+1}_{i+1} + a^j_{i+1}(1 - \xi) + a^j_i\xi\right]}{2} \tag{206}$$

The exception to this rule is u*, whose calculation must precede that of ξ and is given by Equation 92. The variable ξ is calculated by

$$\xi = \frac{u*\tau}{X_{i+1} - X_i} \tag{207}$$

137

Its complement $(1 - \xi)$ is also assigned a variable. In the calculation of Equation 98, the following is the same for all constituents and so is assigned to the variable GO :

$$1 - \frac{r\left(u_{i+1} - u_i\right)}{X_{i+1} - X_i} \tag{208}$$

The coefficients A_1 through A_4 and b_1 through b_4 are calculated according to Equations 81 through 84 and 86 through 89, respectively, based on the estimated value of ξ and its complement.

253. With all the background to the transport equations prepared, attention shifts to the decay rates and source/sink terms. Decay rates, source/sink terms, and final concentrations and derivatives are calculated within a loop that cycles through each of the 12 modeled constituents. Decay rates and source/sink terms are particular to each constituent, and so each is calculated in a separate block. Once decay rates and source/sink terms for a particular constituent are calculated, the final averaging, concentration, and derivative equations can be solved in a stereotyped way, and so the loop reassembles. (The reason that decay rates and source/sink terms are within the loop at all is the interdependence of constituents; for instance, the sink term for DO contains the final concentration of CBOD. If all decay rates and source/sink terms were calculated outside the loop, this information would not be available.) Finally the effect of diffusion is calculated implicitly and added to the constituent concentration.

254. The equations that appear in the listing are equivalent to those given in Parts IV and V with some minor modifications. In one instance, numerical problems presented by the expression

$$2./KEXT*(1-EXP(-KEXT*H))+B*EXP(-KEXT*H) \tag{209}$$

used in determining available photosynthetic energy (Equations 146 and 147) must be handled. First, the number of calls to the external function EXP (exponentiation) can be cut in half by rewriting the above expression as the algebraically equivalent

$$2./KEXT + (B-2./KEXT)*EXP(-KEXT*H) \qquad (210)$$

The above expression, however, becomes numerically unstable as KEXT approaches zero, although mathematically it approaches the value

$$2.*H + B \qquad (211)$$

This problem is handled by taking a Taylor's series expansion of EXP(-KEXT*H) in the expression 2./KEXT*(1.-EXP(-KEXT*H)) about KEXT = 0 carried to three terms,

$$2./KEXT*(1. - (1. - KEXT*H + ((KEXT*H)**2)/2 + ...)) $$

This simplifies to

$$H*(2. - KEXT*H) \qquad (212)$$

The entire expression is then

$$H*(2. - KEXT*H) + B*EXP(-KEXT*H) \qquad (213)$$

for KEXT less than 0.01.

255. After calculating decay rates and source/sink terms, the loop reassembles to calculate averages and spatial derivatives of these quantities. Finally, the concentrations and their spatial derivatives are calculated.

256. Lateral inflow is modeled by simple mass balance. Subtracting the continuity equation (Equation 34) times the concentration, α, from the pollutant transport equation (Equation 36), yields (neglecting off-channel storage, diffusion, decay, and source/sink terms)

$$\frac{\partial \alpha}{\partial T} + u \frac{\partial \alpha}{\partial X} = \frac{q}{A}(\gamma - \alpha) \tag{214}$$

where q is the lateral inflow (both constant and time-varying, L^2T^{-1}) and γ is the concentration in the lateral inflow. The quantity q/A is denoted as QLC or QLT and $\gamma - \alpha$ is replaced by CLC or CLT. The right-hand side of the transport equation is therefore augmented by the term

$$(CLT*QLT + CLC*QLC) \tag{215}$$

The derivative transport equation receives this term after product rule differentiation

$$(-DC*QLC + CLC*DQLC) \tag{216}$$

for constant, or simularly time-varing, lateral inflows, where DC and DQLC are, respectively, αx and $\partial(q/A)/\partial x$.

257. Once the calculations for all the constituents are complete, the frame of interest shifts by one node, and iteration continues.

258. At the completion of the node march, the boundary conditions are assigned to the first node, and the segment loads its own information into T.

Subroutine SPLINE

259. As stated earlier, this solution scheme requires spatial derivatives for the concentrations of all modeled constituents at initial and boundary conditions. Because the user would have no a priori information as to the values of the derivatives, the program estimates them based on a polynomial interpolation of the initial and boundary data themselves. The scheme's being fourth order suggests that interpolation should likewise be fourth order, that is, a cubic spline. Such a spline is completely defined by the following considerations: the curve should pass through every data point; it should be fourth order; although the equation may vary from one interval to the next, the curve and its first and second derivatives should be continuous at every point; and the second derivative at the two boundaries should be zero (that

140

is, there should be no bending of the curve at the end points, commonly known as a "relaxed" or "natural" condition).

260. A complete development of the equations is given by Rogers and Adams (1976). Briefly, the cubic equation for each interval is defined by the values and derivatives at each end of the interval, just as is done in the development of the fourth-order scheme in Part IV. Requiring that the second derivative be continuous at every point implies that the cubic equations for any two adjacent intervals yield equal second derivatives at their point of juncture. This leads to a system of n - 2 linear equations, where n is the number of nodes, that is, one equation for each interior node. The system is completed by the equation that sets the second derivative to zero at the first node and the one that does the same for the last node. This produces a system of n linear equations in n unknowns, where the coefficient matrix is of tridiagonal form filled with nonzero members along the main diagonal, one row above the main diagonal and one below. Tridiagonal matrices are amenable to much faster solution than general square matrices--the solution scheme employed here is subroutine TRIDAG, taken with slight modification, from Carnahan, Luther, and Wilkes (1969).

261. The mathematical formalism proceeds as follows. Recall Equation 80 in expanded form:

$$Y(\xi) = \xi^2(3 - 2\xi)\alpha_i + \left[1 - \xi^2(3 - 2\xi)\right]\alpha_{i+1}$$

$$+ \xi^2(1 - \xi)\left(x_{i+1} - x_i\right)\alpha x_i - \xi(1 - \xi)^2\left(x_{i+1} - x_i\right)\alpha x_{i+1}$$

(217)

The expression for the second derivative is then

$$\ddot{Y}(\xi) = (6 - 12\xi)\alpha_i + (-6 + 12\xi)\alpha_{i+1}$$

$$+ (2 - 6\xi)\left(x_{i+1} - x_i\right)\alpha x_i + (4 - 6\xi)\left(x_{i+1} - x_i\right)\alpha x_{i+1}$$

(218)

Approaching node i+1 from the left, $\xi = 0$, and approaching it from the right $\xi = 1$. The requirement that the second derivative be continuous translates mathematically to

$$\frac{d^2y}{dx^2}\bigg|_{x_{i+1}^-} = \frac{d^2y}{dx^2}\bigg|_{x_{i+1}^+} \tag{219}$$

Now

$$\frac{d^2y}{dx^2}\bigg|_{x_{i+1}^-} = \ddot{Y}(\xi)\bigg|_{\xi=0} \cdot \frac{1}{\left(x_i - x_{i+1}\right)^2} \tag{220}$$

and

$$\frac{d^2y}{dx^2}\bigg|_{x_{i+1}^+} = \ddot{Y}(\xi)\bigg|_{\xi=1} \cdot \frac{1}{\left(x_{i+1} - x_{i+2}\right)^2} \tag{221}$$

Equating the two

$$\frac{1}{\left(x_i - x_{i+1}\right)^2} \ddot{Y}(\xi)\bigg|_{\xi=0} = \frac{1}{\left(x_{i+1} - x_{i+2}\right)^2} \ddot{Y}(\xi)\bigg|_{\xi=1} \tag{222}$$

which can also be written

$$\frac{x_{i+2} - x_{i+1}}{x_{i+1} - x_i} \ddot{Y}(\xi)\bigg|_{\xi=0} = \frac{x_{i+1} - x_i}{x_{i+2} - x_{i+1}} \ddot{Y}(\xi)\bigg|_{\xi=1} \tag{223}$$

In expanded form

$$\frac{x_{i+2} - x_{i+1}}{x_{i+1} - x_i} \left[6\alpha_i - 6\alpha_{i+1} + 2(x_{i+1} - x_i)\alpha x_i + 4(x_{i+1} - x_i)\alpha x_{i+1} \right]$$

$$= \frac{x_{i+1} \, x_i}{x_{i+2} - x_i} \left[-6\alpha_{i+1} + 6\alpha_{i+2} - 4(x_{i+2} - x_{i+1})\alpha x_{i+1} - 2(x_{i+2} - x_{i+1})\alpha x_{i+2} \right] \tag{224}$$

Rearranging so that all the derivatives are on the left-hand side and everything else on the right-hand side and dividing by two yields

$$(x_{i+2} - x_{i+1})\alpha x_i + 2\left[(x_{i+2} - x_{i+1}) + (x_{i+1} - x_i) \right] \alpha_{i+1} + (x_{i+1} - x_i)\alpha x_{i+2}$$

$$= 3 \left[\frac{x_{i+2} - x_{i+1}}{x_{i+1} - x_i} (\alpha_{i+1} - \alpha_i) + \frac{x_{i+1} - x_i}{x_{i+2} - x_{i+1}} (\alpha_{i+2} - \alpha_{i+1}) \right] \tag{225}$$

The second derivative being zero at the first node requires (setting $\xi = 1$) that

$$-6\alpha_1 + 6\alpha_2 - 4(x_2 - x_1)\alpha x_1 - 2(x_2 - x_1)\alpha x_2 = 0 \tag{226}$$

or

$$\alpha x_1 + \frac{1}{2}\alpha x_2 = \frac{3}{2}\frac{1}{(x_2 - x_1)}(\alpha_2 - \alpha_1) \tag{227}$$

Similarly for the last node

$$\frac{1}{2}\alpha x_{n-1} + \alpha x_n = \frac{3}{2}\frac{1}{(x_n - x_{n-1})}(\alpha_n - \alpha_{n-1}) \tag{228}$$

262. The entire system can be written in matrix form and given to subroutine TRIDAG to solve.

$$
\begin{bmatrix}
1 & \dfrac{1}{2} & & \\
(x_{i+2} - x_{i+1}) & 2[(x_{i+2} - x_{i+1}) + (x_{i+1} - x_i)] & (x_{i+1} - x_i) \\
& & \dfrac{1}{2} & 1
\end{bmatrix} \cdot
\begin{bmatrix}
\alpha x_i \\
\alpha x_{i+1} \\
\alpha x_n
\end{bmatrix}
$$

$$
= \begin{bmatrix}
\dfrac{3}{2} \dfrac{1}{(x_2 - x_1)} \left(\alpha_2 - \alpha_1\right) \\[2mm]
3\left[\dfrac{x_{i+2} - x_{i+1}}{x_{i+1} - x_i} \left(\alpha_{i+1} - \alpha_i\right) + \dfrac{x_{i+1} - x_i}{x_{i+2} - x_{i+1}} \left(\alpha_{i+2} - \alpha_{i+1}\right) \right] \\[2mm]
\dfrac{3}{2} \dfrac{1}{\left(x_n - x_{n_1}\right)} \left(\alpha_n - \alpha_{n-1}\right)
\end{bmatrix}
\tag{229}
$$

Tributaries

263. Tributaries in the constituent transport equations are considered to be point additions of materials. The concentration at the junction point is a function of the concentration and flow of the tributary and the receiving stream just upstream of the junction. For a junction at node i :

$$
\alpha_i = \frac{\left(\alpha\!\emptyset_i \cdot Q\!\emptyset_i + \alpha_T \cdot Q_T\right)}{\left(Q\!\emptyset_i + Q_T\right)}
\tag{230}
$$

where

α - concentration

$\emptyset$ - receiving stream just upstream of the junction

Q - flow

T - tributaries

RIV1Q requires two reaches to achieve full dilution of the tributary flow; therefore Q_{i+1} replaces the sum of $Q\emptyset$ and Q_T in Equation 230,

144

$$\alpha_i = \frac{\left[\alpha \varnothing_i \left(Q_{i+1} - Q_T\right) + \alpha_T \cdot Q_T\right]}{Q_{i+1}}$$

$$= \alpha \varnothing_i \left(1 - \frac{Q_T}{Q_{i+1}}\right) + \alpha_T \left(\frac{Q_T}{Q_{i+1}}\right)$$

(231)

The quantity Q_T/Q_{i+1} can be represented by the dilution ratio D :

$$\alpha_i = \alpha \varnothing_i \left(1 - D\right) + \alpha_T D$$

(232)

264. To preserve fourth-order correctness through a junction, derivatives must be advected as well. The derivatives are diluted in the same way as concentrations, with the addition that spatial derivatives on the tributary must be adjusted for the difference between the velocity on the tributary and on the receiving stream:

$$\alpha x_i = \frac{\left[\alpha x \varnothing_i \left(Q_{i+1} - Q_T\right) + \alpha x_T Q_T \frac{u_T}{u_i}\right]}{Q_{i+1}}$$

(233)

$$= \alpha x \varnothing_i \left(1 - \frac{Q_T}{Q_{i+1}}\right) + \left(\alpha x_T \frac{u_T}{u_i}\right) \frac{Q_T}{Q_{i+1}}$$

$$= \alpha x \varnothing_i \left(1 - D\right) + \alpha x_T DD$$

where DD represents the quantity $D \cdot u_T/u_i$.

265. Tributary influx is modeled here as a discontinuity, and so the values immediately upstream of the junction point must be stored. The first 20 members of the T-array contain concentrations and their derivatives at the mouth. Location 21 contains tributary flow, and location 22 contains tributary velocity. Locations 23 through 32 contain concentrations on the main stem just upstream of the junction point, and locations 33 through 42 contain the derivatives there. In the solution procedure for the receiving stream,

each node is checked to see if it is a junction point. If it is not, the solution proceeds normally. If it is, the array pointers shift to the appropriate positions in the T-array and $\alpha\phi$ and $\alpha x\phi$ are used in place of α_i and αx_i in performing advection and first-order decay. The calculated values become $\alpha\phi$ and $\alpha x\phi$ for the new time-step. Finally, α_i and αx_i are calculated according to Equations 232 and 233.

Input and Output Formats

266. The input to the water quality model is provided in up to four files, depending upon the options selected, in addition to the hydrodynamic linkage file. The main input file contains information on the model configuration, such as specification of constants, initial conditions, and boundary specifications. Time-varying boundary conditions, meteorological conditions, and lateral inflows are specified in separate files. The structure of these files are described in the following sections. DOS extensions to the file-names are enforced. The main input file should have the extension *.INP, the time-varying lateral inflow file the extension *.LAC, the boundary condition file *.BCF, and the meteorological file the extension *.MET. The hydrodynamic linkage file must have the extension *.HYD. Similarly, output files will have the same name of the input file, but with the extension *.OUT for the main output file, *.HYD for the interface file for the water quality model, *.ERR for the error file, or *.EDF for the dump file containing data for graphical post-processing.

Main input

267. The main input deck for RIV1Q consists of card (or card-image) input from the user. Card input is composed of nine types of cards: title, global constants, print updates, bypass options, segment, constant, initial conditions, boundary conditions identification, and boundary file specification. The title card supplies the title with which the output of each run is labeled. The global constant cards assign constants that do not vary spatially, the print update cards specify print frequencies, while the bypass card specifies whether a particular constituent is to be simulated or held constant. The segment card gives data pertinent to each segment, such as name and identification number. On the constant card, rate coefficients and the other parameters are given values by name. The initial conditions cards assign concentrations to each of the modeled constituents at every node for

the first time-step. The segments for which boundary conditions are given are listed on the boundary conditions identification card. If the user does not want to simulate a particular constituent (such as reduced iron), then the bypass option for that constituent should be set to 1. This will cause the concentration for that constituent to remain constant at its initial value. However, if that initial concentration is nonzero and decay rates are specified, then the impact of that decay rate on other constituents may still be realized. For example, if BOD is bypassed, its initial condition is nonzero, and a decay rate is specified, it will remove DO at a constant rate (KC).

268. The organization of the main input file is shown in Figure 18. The title card comes first, followed by global constant cards, print interval cards, a bypass card, a segment card, constant cards, and initial conditions cards for each segment. Boundary conditions are introduced by a boundary conditions identification card and followed by the name of the files containing the time-varying data. This is the complete file for making a single run. Depending on the options selected for lateral inflows and temperature simulation, other input files may be necessary. These files are discussed later.

269. <u>Title card.</u> The user has the full 80 columns of the title card to write any title of his choosing. It will be printed, verbatim near the top of every page of the output file.

270. <u>Global constant card</u>. These cards specify values of model constants which do not vary spatially. The constants are identified in Table 2. The order in which they must appear on the global constant card is shown in the example input file (Figure 18). The format for these cards is 5(8X,F8.0), or 5(8X,I8). Space for identifying names is provided in input. However, the names are only for convenience and are not used by the model. Rather, the specific constants are identified by their location in input.

271. <u>Print update card</u>. The print update card identifies the number of print interval updates that will be read from the following card(s). The format of the card is 8X,I8.

272. <u>Print interval card(s)</u>. The print interval cards control the times at which model results are printed to the output file. On this card, the user provides a print interval (hours) and time (Julian day), up to which that print interval applies. The print interval will be constant between the times specified. The format for this card is 4(8X,F8.0), so that two pairs of data can be input per line.

```
SAMPLE RIV1Q DATA SET - USER'S MANUAL
    START     18.     END     20.0
    TBIOS    1.047  TAMMON      1.1  TPHYI    1.024
   APCONT     0.01  ANCONT    0.075
   ONEQUI     0.35  ONITRI     4.56  OPDECY   1.59   OFEDEC   0.14  OMNDEC    0.1
     DAWN      6.5  SUNSET     20.5
   LATOPT        1
  NOPRINT        2
 INTERVAL     12.0    TIME    0.0INTERVAL    12.0    TIME   1000.
   NOPLOT        2
 INTERVAL      1.0    TIME    0.0INTERVAL     1.0    TIME   1000.
     0      0      0       0     0       0      0      0     0      0     0      0
 1 MAIN STEM ABOVE REREG DAM          2   0.045
      ADN      0.1      AG    12.81     AKN    0.3    AKNX    0.0     AK1    0.2
      ATB      0.0     ATS      0.0    APO4    0.0      BK  -0.49   CSINK     0.
       E1      0.5      E2      1.5   KALGDK   0.1   KALGRO   .02
   KNCBDN      0.1   KOALDK     0.5   KOCBDN   0.5    KOCB1    0.5
      KON      0.5    TEMP     23.0    TSINK  10.0
     TSIV      0.0  KCOLIDK    1.4    KMNDK    0.5    KFEDK    0.5  OXIDAT     1.
   HNEFSW      0.0   KNPOOL     0.1    KPO4X    0.1    KDNO2    0.0     ACK     0.
  LAMBDA0      1.0  LAMBDA1     0.0  LAMBDA2   0.0   ALPHA0    0.0    XONS     0.
     ABSR      0.0   CBODSR     0.0    FCBOD    0.0     KPDK    0.0   KPSET     0.
   BENPO4      0.0     SOD      0.0   MACROB    0.0   MACGRO    0.0  MACDKY     0.
    KLITE     21.0    ITEM       0
   QWINDO  .FALSE.
    23.4    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    22.8    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.60    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    22.8    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.70    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    24.3    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    24.4    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    24.5    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    24.7    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.1    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.4    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.7    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    26.0    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    26.0    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.7    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
    25.3    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.40    0.00    0.00
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 250.
```

Figure 18. Sample RIV1Q *.INP input data file (Sheet 1 of 3)

```
2 MAIN STEM BELOW REREG DAM
   ADN      0.1       AG   12.81      AKN      0.3     AKNX      0.0       AK1      0.2
   ATB      0.0      ATS     0.0     APO4      0.0       BK    -0.49     CSINK      0.
    E1      0.5       E2     1.5   KALGDK      0.1   KALGRO      .02
KNCBDN      0.1   KOALDK     0.5   KOCBDN      0.5    KOCB1      0.5
   KON      0.5     TEMP    23.0    TSINK     10.0
  TSIV      0.0  KCOLIDK     1.4    KMNDK      0.5    KFEDK      0.5    OXIDAT      1.
 HNEFSW     0.0   KNPOOL     0.1    KPO4X      0.1    KDNO2      0.0       ACK      0.
LAMBDA0     1.0  LAMBDA1     0.0  LAMBDA2      0.0   ALPHA0      0.0      XONS      0.
  ABSR      0.0   CBODSR     0.0     FCBOD      0.0     KPDK      0.0     KPSET      0.
 BENPO4     0.0      SOD     0.0   MACROB      0.0   MACGRO      0.0    MACDKY      0.
 KLITE     21.0     ITEM       0
 QWINDO .FALSE.
   24.8    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   24.3    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.60    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.8    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.60    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.7    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.60    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.8    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   24.0    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   25.0    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.40    0.00    0.00
    0.0     0.0     0.0     0.0     0.0    .0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   25.6    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   26.0    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   25.4    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.30    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   24.2    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.4    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.50    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.7    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.40    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.20    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.10    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.10    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
   23.9    0.01    0.01    0.10    0.01    0.00    0.02    0.00    0.00    8.10    0.00    0.00
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
250.
```

Figure 18.   (Sheet 2 of 3)

```
3 MAIN STEM TRIBUTARY ABOVE REREG DAM
    ADN    0.1       AG   12.81      AKN    0.3    AKNX    0.0      AK1    0.2
    ATB    0.0      ATS    0.0      APO4    0.0      BK   -0.49   CSINK    0.
     E1    0.5       E2    1.5    KALGDK    0.1   KALGRO    .02
 KNCBDN    0.1   KOALDK    0.5    KOCBDN    0.5    KOCB1    0.5
    KON    0.5     TEMP   23.0     TSINK   10.0
   TSIV    0.0  KCOLIDK    1.4     KMNDK    0.5    KFEDK    0.5   OXIDAT    1.
 HNEFSW    0.0   KNPOOL    0.1     KPO4X    0.1    KDNO2    0.0      ACK    0.
LAMBDA0    1.0  LAMBDA1    0.0   LAMBDA2    0.0   ALPHAO    0.0     XONS    0.
   ABSR    0.0   CBODSR    0.0      FCBOD    0.0     KPDK    0.0    KPSET    0.
 BENPO4    0.0      SOD    0.0    MACROB    0.0   MACGRO    0.0   MACDKY    0.
  KLITE   21.0     ITEM      0
 QWINDO .FALSE.
   25.0    0.01    0.01   0.10    0.01   0.00   0.02   0.00   0.00   7.20   0.00   0.00
    0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 250.
   25.1    0.01    0.01   0.10    0.01   0.00   0.02   0.00   0.00   7.40   0.00   0.00
    0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 250.
   25.4    0.01    0.01   0.10    0.01   0.00   0.02   0.00   0.00   7.50   0.00   0.00
    0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 250.
   26.2    0.01    0.01   0.10    0.01   0.00   0.02   0.00   0.00   7.50   0.00   0.00
    0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 250.
   26.3    0.01    0.01   0.10    0.01   0.00   0.02   0.00   0.00   7.60   0.00   0.00
    0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 250.
   26.2    0.01    0.01   0.10    0.01   0.00   0.02   0.00   0.00   7.60   0.00   0.00
    0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 250.
+1,+3/
SAMPLEQ1.BCF
SAMPLEQ2.BCF
```

Figure 18.   (Sheet 3 of 3)

273.   Plot update card. The plot interval card identifies the number of plot interval updates that will be read from the following card(s). The format of the card is 8X,I8.

274.   Plot interval card(s). The plot interval cards control the times at which model results are printed to files for graphic post-processing. On this card, the user provides a plot interval (hours) and time (Julian day), up to which that plot interval applies. The plot interval will be constant between the times specified. The format for this card is 4(8X,F8.0), so that two pairs of data can be input per line.

275.   System bypass card. This card contains the bypass options for each of the 12 water quality constituents. The values of the option may either be 0, indicating simulated, or 1, indicating that the constituent should be bypassed. If a constituent is bypassed, its concentrations remain constant over the period of simulation and equal to the values specified in the initial conditions. The format of this card is 12I5.

150

Table 2

Parameters That Can Appear in the Global Constants Cards

Name	Unit	Explanation*
START	Day	Start time for simulation, input for convenience but overriden by hydrodynamic file start time (R)
END	Day	End time for simulation, input for convenience but overridden by hydrodynamic file end time (R)
TBIOS	None	Temperature coefficient for biological processes (R; default value = 1.047)
TAMMON	None	Temperature coefficeint for nitrification (R; default value = 1.1)
TPHYI	None	Temperature coefficient for physical processes (R; default value = 1.024)
APCONT	None	Phosphorus-to-biomass ratio in algae and macrophytes (R; default value = 0.01)
ANCONT	None	Nitrogen-to-biomass ratio in algae and macrophytes (R; default value = 0.075)
ONEQUI	None	Incremental increase in oxygen-to-algal biomass ratio for oxygen production by algae and macrophytes when nitrate is used as a nitrogen source (R; default value = 0.35)
ONITRI	None	Oxygen-to-nitrogen ratio for ammonia oxidation (R; default= 4.56)
OPDECY	None	Oxygen-to-biomass ratio for oxygen production by algae and macrophytes when ammonia is the nitrogen source (R; default value = 1.59)
OFEDEC	None	Oxygen-to-iron ratio for iron oxidation (R; default value = 0.14)
OMNDEC	None	Oxygen-to-manganese ratio for oxidation (R; default value = 0.15)
DAWN	Hours	Time of sunrise (R)
SUNSET	Hours	Time of sunset (R)
LATOPT	None	Time-varying lateral flow option. If LATOPT≥1, then the name of the lateral inflow file is given in the third record of the RIV1Q control file (RIV1Q.CTL) (I)

* R = real variable; I = integer.

276. **Segment card.** The format of the segment card is I2,10A4,I2,F10.0. The items specified are ID, the identification number for this segment, SNAME, its name, IDAM, and DAMK. The two latter variables concern reaeration that occurs through the control structure, if any, at the bottom of this segment. If this segment does not end with a control structure or if reaeration is not

modeled, these two fields are left blank. If IDAM is 0 or blank, structural reaeration is not modeled. If IDAM is given a value of one or greater, structural reaeration is modeled with the energy dissipation model (Wilhelms and Smith 1981),

$$r = e^{-C_T \Delta H} \qquad (234)$$

where

r — deficit ratio (final DO deficit/initial DO deficit)

C_T — escape coefficient

ΔH — difference in water surface elevation across the control structure

The value for the escape coefficient at 20 °C is input by DAMK with units of ft^{-1}; the program corrects for temperature by

$$C_T = DAMK*1.022^{(TEMP-20)} \qquad (235)$$

A value of 0.045 ft^{-1} for DAMK was recommended (Wilhelms and Smith 1981) for gated-conduit outlet works and low head weirs and gated spillways (H < 25 ft) with free hydraulic jumps. The value for this coefficient may be significantly different for other types of outlet structures or for conditions not conducive to reaeration, such as submerged hydraulic jumps that can be experienced with many low-sill gated structures. The program is not set up to allow structural reaeration at headwater nodes.

277. <u>Segment specific constant cards.</u> On these cards, the user specifies values by name for parameters and coefficients used in the model. Constant cards must be given for each segment, even if the constants are the same for every segment. Their names and uses are given in Table 3.

278. The segment specific constants must appear in the order specified on the lines illustrated in the example input data set (Figure 18). The format for these cards is 5(8X,F8.0), or 5(8X,I8). Space for identifying names is provided in input. However, the names are only for convenience and are not used by the model. Rather, the specific constants are identified by their location in input.

Table 3

Parameters That Can Appear on the Constant Card

Name	Units	Default	Explanation
ABSR	$g\ m^{-2}$		Benthal source rate for ammonia
ACK	day^{-1}		Decay rate for organic nitrogen
ADN	day^{-1}	0.1	Rate coefficient for denitrification
AG	$day^{-1}fps^{-E1}/ft^{-E2}$	12.81	Rate coefficient for stream reaeration rate in the form $$K_2 = \frac{AG\ U^{E1}}{H^{E2}}$$
AKN	day^{-1}	0.3	Rate coefficient for nitrification
AKNX	day^{-1}	0.0	Rate coefficient for ammonia adsorption
AK1	day^{-1}	0.15	Rate coefficient for CBOD decay and organic nitrogen decay
ALPHA0			Conversion factor from algae to chlorophyll
APO4	day^{-1}	0.0	Rate coefficient for phosphate adsorption
ATB	day^{-1}	0.0	Rate coefficient for bottom heat exchange

(Sheet 1 of 5)

Table 3 (Continued)

Name	Units	Default	Explanation
ATS	$\dfrac{W}{m^2\ ^\circ C}$	0.0	Rate coefficient for surface heat exchange in constant equilibrium temperature method*
BENPO4	$g\ m^{-2}$		Benthic source term for dissolved phosphorus
CBODSR	$m\ day^{-1}$		Settling rate for CBOD
E1	Unitless	0.50	Exponent of velocity in stream reaeration rate formulation in the form
			$$K_2 = \frac{AG\ U^{E1}}{H^{E2}}$$
			Default value is for the O'Connor, Dobbins (1958) formulation.
E2	Unitless	1.50	Exponent of depth in stream reaeration rate formulation in the form
			$$K_2 = \frac{AG\ U^{E1}}{H^{E2}}$$
			Default value is for the O'Connor, Dobbins (1958) formulation.
FCBOD	Unitless	0 to 1	Fraction of algal and macrophyte decay which goes to CBOD

(Sheet 2 of 5)

* $1\ \dfrac{W}{m^2\ ^\circ C} = 4.23\ \dfrac{BTU}{ft^2 day\ ^\circ F}$

Table 3 (Continued)

Name	Units	Default	Explanation
HNEFSW	W/m^2	0.0	Surface light intensity at local noon. A value is required if modeling photosynthesis but not using direct energy budget that computes HNEFSW.
ITEM		0	To designate type of heat exchange solution. For ITEM=1, the constant equilibrium temperature solution is used (should also specify ATS and TEMP). For ITEM=0, the direct energy balance is used for temperature solution, and meteorological data must be furnished. For any other value of ITEM other than 0 or 1, heat exchange is not modeled.
KALGDK	day^{-1}	0.0	Algal decay rate
KALGRO	$m^2Watts^{-1}day^{-1}$	0.0	Algal growth rate
KCOLIDK	day^{-1}	0.0	Rate coefficient for coliform bacteria mortality
KDSED	day^{-1}		Sediment denitrification rate
KFEDK	day^{-1}	0.0	Rate coefficient for iron oxidation
KLITE			Light intensity at which photosynthesis rate reduced by 1/2
KMNDK	day^{-1}	0.5	Rate coefficient for manganese oxidation

Table 3 (Continued)

Name	Units	Default	Explanation
KNCBDN	mg/l	0.1	Nitrate half-saturation constant for denitrification
KNPOOL	mg/l		Total nitrogen concentration at which algal growth rate is reduced by 1/2
KNSET	m day^{-1}		Settling rate for organic nitrogen
KOALDK	mg/l	0.5	DO half-saturation constant for algal decay
KOCBDN	mg/l	0.5	DO half-saturation for denitrification
KOCB1	mg/l	0.5	DO half-saturation constant for CBOD decay (Hoover and Porges 1952)
KON	mg/l	0.5	DO half-saturation for nitrification
KPDK	day^{-1}		Decay rate of organic phosphorus
KPO4X	mg/l	.	Phosphorus concentration at which algal growth rate is reduced by 1/2
KPSET			Settling rate coefficient for organic phosphorus
LAMBDAO			Nonalgal portion of light extinction coefficient
LAMBDA1	$\dfrac{1/m}{\text{ug biomass/l}}$		Linear algal self-shading coefficient
LAMBDA2	$\dfrac{1/m}{(\text{ug biomass/l})^{2/3}}$		Nonlinear algal self-shading coefficient
MACDKY	day^{-1}		Specific macrophyte decay rate

(Sheet 4 of 5)

Table 3 (Concluded)

Name	Units	Default	Explanation
MACROB	$g\ m^{-2}$		Macrophyte density on channel surfaces
MACGRO	day^{-1}		Macrophyte growth rate
QWINDO		FALSE	A logical variable to invoke wind-driven reaeration. To turn on wind reaeration for a segment, set QWINDO = TRUE.
SOD	$g\ m^{-2}$		Sediment oxygen demand
TEMP	°C	23.0	If temperature is modeled with the constant equilibrium temperature approach, then TEMP is used to input the equilibrium temperature for each stream segment.
TSINK	°C	10.0	Source/sink term for bottom heat exchange
TSIV	ft^{-1}	0.0	Coefficient in the Tsivoglou-Wallace (1972) reaeration formula (a value of zero serves to indicate that the equation $$K_2 = \frac{AG\ U^{E1}}{H^{E2}}$$ will be used rather than this). If the Tsivoglou-Wallace (1972) formula is used, a value of 0.054 is suggested for TSIV.

279. <u>Initial conditions cards.</u> As stated in Part IV, necessary and sufficient initial conditions are the concentrations of all the parameters to be modeled. Accordingly, each initial conditions card is divided into 12 areas, each eight columns wide, which contain the values of temperature, CBOD, organic nitrogen, ammonia nitrogen, nitrate plus nitrate nitrogen, organic phosphate, dissolved phosphate, reduced manganese, reduced iron, dissolved oxygen, coliform bacteria, and algae, respectively. Columns corresponding to constituents that are not modeled are set to zero. The format for these cards is 12F8.0.

280. <u>Constant lateral inflow cards</u>. The values for the concentrations of the constant lateral inflows are placed on the line immediately below the initial conditions, in the same order and using the same format. These concentrations must correspond to the conditions for the constant lateral flows specified in the hydrodynamic input. If the user elects to simulate the impacts of time-varying lateral flows, the name of the file containing the time-varying data is listed in the third record of the RIV1Q control file.

281. <u>Dispersion coefficient cards</u>. These cards assign the horizontal dispersion coefficient. The format for the dispersion coefficient is F10.0.

282. The user can simulate an arbitrary conservative or nonconservative constituent with the coliform bacteria variable by setting KCOLIDK - 0.0 or to a specified first-order kinetic rate, respectively.

283. In most applications, the initial concentrations of water quality constituents at each node will not be known. This problem can be overcome by several means. If water quality data are available at several stations along the stream, values could be interpolated for the nodes. If such data are not available for the conditions being simulated, then a simulation can be made with estimated initial conditions and time-invariant boundary conditions that are the same as the first set of time-variant boundary conditions. The resulting steady-state nodal concentrations are then used as the initial conditions for subsequent simulations. However, these steady-state initial conditions still may not be representative. If the nodal concentrations change significantly following the first flush (system travel time), then the initial conditions were probably not representative, and the results during the first flush should not be used for interpretation.

284. It is possible to improve the initial condition estimates to provide more meaningful results during first flush if conditions (flows,

loadings, etc.) during or at the end of the simulation (but after first flush) are similar to the starting conditions. A dynamic simulation is made with estimated initial conditions; the simulated nodal concentrations, at the time that conditions (flows, loadings, etc.) are similar to initial conditions, are selected for initial conditions in subsequent simulations. For example, suppose that the simulation starts on Monday and ends on Sunday for a stream below a peaking hydropower dam where peaking operations occur on weekdays and steady low flows are maintained on weekends. Then the simulated nodal concentrations on Sunday should provide reasonable estimates for the initial Monday conditions (provided sufficient time elapsed for first flush and other factors are similar at the beginning and end of the simulation).

285. <u>Boundary conditions identification card.</u> The transport equations require boundary conditions for the upstream end of every segment. However, in the case where the segments are actually part of the same stream but are separated by a control structure, two segments are modeled but the boundary conditions for the lower segment are determined by the upper segment and do not appear in the input. The type of boundary conditions required, therefore, is uniquely determined once the network structure is fixed. The user indicates to the program on this card the segments (ID) whose boundary conditions will be given. The numbers can appear anywhere on the card and can be separated by a comma or blanks or both, but they must appear in order, with the list terminated by a slash (/).

286. <u>Boundary condition cards.</u> On these cards, the user specifies the filename containing the time-varying boundary condition data. The number of files specified, and their order, should correspond to the boundary conditions identification card. The format is A15.

<u>Boundary conditions file(s)</u>

287. Time-varying boundary conditions are specified in external files whose names are specified in the main input data set. An example of one boundary condition file is given in Figure 19. The files are read by subroutine TIME_VARYING_DATA. On the initial call to this subroutine, the files are opened, the specification card read, and the initial update time and boundary conditions read. On subsequent calls, if the simulation time equals or exceeds the next update time, the boundary conditions are updated. For simulation times between the update intervals, the boundary conditions may be linearly interpolated or held constant, at the option of the user. Upon reading the files, the filenames and data are written to the output file.

```
1983  1  18  6  0.0   0  12.   1.   0.5   0.20  0.1  0.00  0.05  0.  0.  2.5  0.  0.
1983  6  18     19.4     12.   1.   0.5   0.20  0.1  0.00  0.05  0.  0.  2.5  0.  0.
1983  6  19     2.9      15.   1.   0.5   0.20  0.1  0.00  0.05  0.  0.  4.0  0.  0.
1983  6  20     2.1      12.   1.   0.5   0.20  0.1  0.00  0.05  0.  0.  2.5  0.  0.
1983  6  21     17.1     12.   1.   0.5   0.20  0.1  0.00  0.05  0.  0.  4.0  0.  0.
                0.0      TEMP CBODNS ORGAN NH3N NO3N ORG-P PO4   MN  FE  DO   CLFM ALGAE

1983  3  0   2  0.0   0  20.   1.   0.1   0.01  0.1  0.00  0.01  0.  0.  8.  0.  0.
1983  1  12     0.0      20.   1.   0.1   0.01  0.1  0.00  0.01  0.  0.  8.  0.  0.
                         TEMP CBODNS ORGAN NH3N NO3N ORG-P PO4   MN  FE  DO   CLFM ALGAE
```

Figure 19.   Sample RIV1Q *.BCF input data file: The two files listed above are actually two separate
*.BCF files that RIV1Q will need. They have been combined for this figure.

160

288. <u>Specification card</u>. The first line of these files contains the segment number for which the boundary is provided, the number of boundary condition updates (NUMBC), and the variable INTBC. If the variable INTBC is greater than or equal to one, the boundary conditions will be linearly interpolated between update intervals. Otherwise, they will be held constant between updates (a step function). The format of the first line of the file is 3I10.

289. <u>Boundary update cards</u>. Following the initial line of the boundary condition file are NUMBC lines of update intervals. Each line of update will contain the year, month, day, and hour for the update and the corresponding boundary condition concentrations (mg/l) for each of the 12 water quality constituents, in the following order: temperature, CBOD, organic nitrogen, ammonia nitrogen, nitrate plus nitrate nitrogen, organic phosphate, dissolved phosphate, reduced manganese, reduced iron, dissolved oxygen, coliform bacteria, and algae. The format for each update is 3I5, 15F10.0. The update times should bracket the period of model simulation.

<u>Lateral inflow file</u>

290. The lateral inflow file contains the time-varying lateral inflow concentrations. If the user elects to input time-varying lateral inflows (LATOPT≥1), the file name (with DOS extension *.LAC) is given in the control file. An example of the lateral inflow file is given in Figure 20. The file is read by subroutine TIME_VARYING_DATA. On the initial call to this subroutine, the file is opened, the specification card read, and the initial update time and lateral inflow concentrations read. On subsequent calls, if the simulation time equals or exceeds the next update time, the boundary conditions are updated. For simulation times between the update intervals, the lateral inflow concentrations may be linearly interpolated or held constant, at the option of the user. Upon reading the files, the data and filename are written to the output

291. <u>Specification card</u>. The first line of the lateral inflow file contains the specifications for the number of updates (NUMLAT_UD), the number of nodes which will receive lateral inflows (NUML), variable LATOPT, and the number of water quality constituents (NUMSYS). If the variable LATOPT is greater than or equal to one, the lateral inflows will be linearly interpolated between update intervals. Otherwise, they will be held constant between updates (a step function). The format for this line is 4I10.

161

```
            2          5          1          4
25
26
27
28
29
 1
 2
 3
 4
   TEMP                EXAMPLE INPUT DATA SET FOR CONSTANT
  CBODNS               LATERAL INFLOW CONCENTRATIONS
  ORGAN
   NH3N
 1983     1     0      0.0
                      29.0       30.0       29.0       29.0       29.0
                       1.0       11.5        1.1        1.1        1.1
                       0.8        2.0        0.6        0.1        0.1
                       0.1        1.2        0.1        0.0        0.0
 1983    10     0      0.0
                      29.0       30.0       29.0       29.0       29.0
                       1.0       11.5        1.1        1.1        1.1
                       0.8        2.0        0.6        0.1        0.1
                       0.1        1.2        0.1        0.0        0.0
```

Figure 20. Sample RIV1Q *.LAC input data file

292. 'Node specification cards. Following the initial line of the input will be NUML lines on which the node numbers that will receive the lateral inflows are specified. The node numbers correspond to the order specified in the main input. The format is I5, so there will be one node specification per line of input.

293. System specification cards. Following the node specification, there will be NUMSYS lines of input on which the user specifies the water quality constituents which will have time-varying lateral inflow concentrations specified. There may be up to 12 concentrations updated, and the number system corresponds to the number of the state variables (1 - temperature, 2 - CBOD, 3 - organic nitrogen, 4 - ammonia nitrogen, 5 - nitrate plus nitrate nitrogen, 6 - organic phosphate, 7 - dissolved phosphate, 8 - reduced manganese, 9 - reduced iron, 10 - dissolved oxygen, 11 - coliform bacteria, and 12 - algae). The format of these cards is I5.

294. System label cards. There will be NUMSYS system label cards on which the user may specify the names of the water quality constituents in the

162

order in which they occur in the system specification card. This card is only for the convenience of the user, as the names are not used in the program.

295. <u>Update cards</u>. This card group consists of NUMSYS+1 lines of input and the group will be repeated NUMLAT_UD times. The first line of the input card contains the year, month, day, and hour of the update, with the format I5,I5,I5,F10.0. Following the update time will be NUMSYS lines of input containing the concentrations of the water quality constituents in the order in which they were named in the system specification card. There will be NUML columns of data on each line, corresponding to, and in the order of, the nodes specified on the node specification card. The format for each of the NUMSYS lines of input is 15X, 150F10.0. Therefore, following the update time will be NUMSYS lines of input, each line containing NUML data points.

<u>Meteorological input</u>

296. Meteorological data, required for heat-exchange computations when ITEM is equal to zero, are provided in a separate input file. If the user elects to compute heat exchange using the full heat balance method (ITEMP=0), the name of the meteorological input file (with DOS extension *.MET) is listed in the fourth record of the control file. An example of the *.MET file is given in Figure 21. The file will be opened and read in the subroutine TIME_VARYING_DATA.

297. <u>Card 1.</u> The user here specifies the dust-attenuation coefficient (DUC). Water Resources Engineers (1967) gives a range of 0.0 to 0.13 for several locations, although results are not very sensitive to changes in this range. The format for this card is F10.0.

298. <u>Card 2.</u> The user here specifies the local latitude (LATUDC) and the local longitude (LONTUC) of the project, and the longitude (LSMC) of the standard meridian east of the project (standard meridians are in increments of 15 deg, e.g., 75, 90, 105, etc.). Each value on card 2 is in degrees. The format for this card is 3F10.0.

299. <u>Card 3.</u> Here the user specifies the interpolation option (INTMET) and the number of meteorological updates (NUMT). If INTMET ≥ 1, then the values for meteorological variables will be linearly interpolated between update intervals. Otherwise, they will be held constant (a step function). The format for this card is 2I10.

300. <u>Update cards.</u> Here there will be NUMT lines of input providing the update times and meteorological data. The user here specifies, in the

163

```
0.12
37.0        37.0         74.0
 0          22
1983    6    18      0.00     0.80     8.0     82.00    61.00    28.81
1983    6    18      2.75     0.80     4.0     82.00    61.00    28.81
1983    6    18      6.01     0.70     5.0     81.00    59.00    28.80
1983    6    18      9.01     0.30     5.0     81.00    60.00    28.79
1983    6    18     12.01     0.20     3.0     78.00    58.00    28.79
1983    6    18     15.01     0.20     3.0     74.00    56.00    28.80
1983    6    18     18.01     0.50     3.0     71.00    55.00    28.82
1983    6    18     21.01     0.50     3.0     65.00    54.00    28.84
1983    6    19      0.01     0.50     0.0     61.00    52.00    28.84
1983    6    19      3.01     0.50     4.0     58.00    53.00    28.85
1983    6    19      6.01     0.50     1.0     58.00    53.00    28.85
1983    6    19      9.01     0.50     0.0     54.00    50.00    28.86
1983    6    19     12.01     0.50     0.0     55.00    50.00    28.86
1983    6    19     15.01     0.50     0.0     51.00    49.00    28.86
1983    6    19     18.01     0.50     0.0     51.00    48.00    28.87
1983    6    19     21.01     0.50     4.0     53.00    49.00    28.89
1983    6    20      0.01     0.50     3.0     62.00    54.00    28.89
1983    6    20      3.01     0.60     4.0     66.00    56.00    28.90
1983    6    20      6.01     0.50     3.0     70.00    56.00    28.90
1983    6    20      9.01     0.20     3.0     73.00    59.00    28.90
1983    6    20     12.01     0.50     4.0     77.00    60.00    28.88
1983    6    20     15.01     0.50     2.0     80.00    62.00    28.87
1983    6    20     17.01     0.50     4.0     83.00    64.00    28.85
```

Figure 21.   Sample RIV1Q *.MET input data file

following order, the year, month, day, and hour of the update, the cloud cover (between 0.0 and 1.0), wind speed (miles per hour), dry bulb temperature (degrees Fahrenheit), wet bulb temperature (degrees Fahrenheit), and the atmospheric pressure (conventional inch of mercury). The format of these cards is I5,I5,I5,6F10.0.

Output

301.   The output from RIV1Q (Figure 22) consists of data for each segment at each print interval. At the top of the page appears the program title followed by the run date. The next line prints the run title supplied by the user. After that comes the elapsed time in year, month, day and hours, and the segment ID number and name. Only those parameters modeled appear in the output. Headings, units, and values appear in column format. The values of CBOD and DO are printed with one decimal place (corresponding to analytical accuracy), the remaining variables with two decimal places.

302. Graphics output from RIV1Q consists of binary data for each segment and is available with the option IPLOT. This file will have the DOS extension *.GRF, with the same name as the input file. An ASCII text file with the extension *.EDF will also be available for graphical post-processing.

```
          ************************************************
          *            C E - Q U A L - R I V 1           *
          *        W A T E R   Q U A L I T Y   M O D E L *
          ************************************************

SAMPLE RIV1Q DATA SET - USER'S MANUAL
=====================================

Start Time For Simulation,(hh.mm), Start        =  18.00
End Time For The Simulation (HH.MM), End         =  20.00
Lateral Inflow Option
(0= Constant, 1 = Time Variable)                 =   1
Temperature Coeff. For Biological Processes      =   1.0470
Temperature Coeff. For NH3 Reactivity            =   1.1000
Temperature Coeff. For Physical Processes        =   1.0240

Algal Phosp. Content (Fraction By Weight)        =   0.0100
Algal Nitrogen Content (Fraction By Weight)      =   0.0750
Temperature Coeff. For Physical Processes        =   0.3500
Oxygen/Nitrogen Equivalence (gO2/gN)
Oxygen/Nitrogen Ratio For Nitrification
(gO2/gN)                                         =   4.5600
Oxygen Consumption/Plant Decay (gO2/g Plant)     =   1.5900
Oxygen Consumption/Iron Oxidized (gO2/g Fe)      =   0.1400
Oxygen Consumption/Manganese Oxidized
(gO2/g Mn)                                       =   0.1000

Time of Sunrise, (HH.MM), Dawn                   =   6.50
Time of Sunset, (HH.MM), Sunset                  =  20.50


                   Print Intervals
                   ===============
Interval (Hours)   12.0000 from Time (Days) =       0.0000
Interval (Hours)   12.0000 from Time (Days) =    1000.0000


                   Plot Intervals
                   ==============
Interval (Hours)    1.0000 from Time (Days) =       0.0000
Interval (Hours)    1.0000 from Time (Days) =    1000.0000


              System Bypass Options
              =====================

              (0 = Simulate, 1 = Bypass)

Temp  BOD  OrgN  NH3  NO3  OrgP  PO4  Mn  Fe  DO  CLFM  Algae
 0     0    0     0    0    0     0    0   0   0    0      0

++++++++++++++++++++++++++++ END GLOBAL CONSTANTS ++++++++++++++++++++++++++++
```

Figure 22. Sample RIV1Q *.OUT output data file (Sheet 1 of 14)

```
==============================================
                Segment Parameters
==============================================

For Segment #  1    Desc:    MAIN STEM ABOVE REREG DAM
                 IDAMO = 2      DAMKO =       0.

   Constant Data for Segment No.  1

   NOTE: These Data may be Corrected for Temp., D.O., etc. During
         Program Execution


Stream Segment Temperature, Deg. C,                              TEMP = 23.00
Rate Coefficient for Bottom Heat Exchange (1/DAY)                 ATB =  0.00
Rate Coefficient for Surface Heat Exchange, W/M**2 DEG. C, ATS =  0.00
Source/Sink Term for Bottom Heat Exchange, DEG. C,              TSINK = 10.00
Variable Designating type of Heat Exchange Solution,            ITEM =  0
1--Constant Temp. Equilibrium 0--Full Heat Balance


Nitrate Conc. at Which Denitrification Rate is 1/2
Maximum, mg/l,                                                   NCBDN =  0.10
D.O. Conc. at Which Algal Decay Rate is 1/2 Maximum
mg/l,                                                           KOALDK =  0.50
D.O. Conc. at Which the Rate of Denitrification is
Reduced by 1/2, MG/L,                                           KOCBDN =  0.50
D.O. Conc. at Which CBOD Decay rate is 1/2 Maximum
Rate, mg/l                                                       KOCB1 =  0.50
D.O. Conc. at Which Nitrification Rate is 1/2
Maximum, mg/l                                                     KON =  0.50


Rate Coefficient for Stream Reaeration Rate                       AG = 12.81
Exponent of Velocity in Stream Reaeration Rate, Formulation  E1 =  0.50
Depth Exponent in Stream Reaeration Formulation,                 E2 =  1.50
Coefficient in the Tsivoglou-Wallace Reaeration Equation
1/ft,                                                            TSIV =  0.00
Specific Rate Coefficient, Uncorrected, for
Denitrification 1/day                                            ADN =  0.10
Specific Rate Coefficient, Uncorrected, for
Nitrification 1/day                                              AKN =  0.30
Specific Rate Coefficient, Uncorrected, for
Ammonia Adsorption 1/day                                        AKNX =  0.00
Specific Rate Coefficient, Uncorrected, for
Phosphate Adsorption, 1/day                                     APO4 =  0.00
```

Figure 22. (Sheet 2 of 14)

167

Specific Rate Coefficient For Iron Oxidation, 1/day,   KFEDK = 0.50
Rate Coefficient For Coliform Mortality, 1/day,   KCOLIDK = 1.40
Sediment Oxygen Demand, gr./m**2   SOD = 0.00
Total Nitrogen Conc. at Which Algal Growth
Rate is Reduced by 1/2, mg/1   KNPOOL = 0.10
Phosphorous Conc. at which Algal Growth Rate Reduced by
1/2, mg/1   KPO4X = 0.10
Sediment Denitrification Rate   KDSED = 0.00

Macrophyte Density on Channel Surfaces, g/m**2,   MACROB = 0.00
Specific Macrophyte Growth Rate, 1/day   MACGRO = 0.00
Specific Macrophyte Decay Rate, 1/day   MACDKY = 0.00

Algal Growth Rate, mg/1*day   KALGRO = 0.02
Algal Decay Rate, mg/1*day   KALGDK = 0.10

D.O. Conc. Below Which Oxidation of
Iron and Manganese Do Not Occur, mg/1   OXIDAT = 1.00
Specific Rate Coefficient Manganese Oxidation, 1/day,   KMNDK = 0.50

Empirical Coefficient Reflecting Thickness ofBoundary Layer BK = -0.49
Rate Coefficient Carbonaceous Oxygen Demand, 1/day   AK1 = 0.20
Light Intensity at Which Photosynthesis Rate Reduced by
1/2,   KLITE = 21.00
Rate Coefficient for Org-N Decay To NH3   ACK = 0.00
Surface Light Intensity at Local Noon,   HNEFSW = 0.00
Is Wind Driven Reaeration Used for Segment ?, T or F,QWIND(0) = F

Non-algal Portion of Light Extinction Coefficient   LAMBDA0 = 1.00
Linear Algal Self Shading Coefficient   LAMBDA1 = 0.00
Nonlinear Algal Self Shading Coefficient   LAMBDA2 = 0.00
Conversion Factor From Algae To Chlorophyll,   ALPHA0 = 0.00

Settling Rate for Organic Nitrogen,   KNSET = 0.00
Benthal Source Rate for Ammonia, g/m**2   ABSR = 0.00
Settling Rate for CBOD, m/day   CBODSR = 0.00
Fraction of Algal and Macrophyte Decay
Which Goes to CBOD,   FCBOD = 0.00

Decay Rate For Organic - P,   KPDK = 0.00
Settling Rate Coefficient For Org-P,   KPSET = 0.00
Benthic Source Term for Dissolved P,   BENPO4 = 0.00

For Segment # 2   Desc: MAIN STEM BELOW REREG DAM
                    IDAM0 = 0   DAMK0 = 0.

Constant Data for Segment No. 2

NOTE: These Data may be Corrected for Temp., D.O., etc. During
      Program Execution

Figure 22.   (Sheet 3 of 14)

168

Stream Segment Temperature, Deg. C, TEMP = 23.00
Rate Coefficient for Bottom Heat Exchange (1/DAY) ATB = 0.00
Rate Coefficient for Surface Heat Exchange, W/M**2 DEG. C, ATS = 0.00
Source/Sink Term for Bottom Heat Exchange, DEG. C, TSINK = 10.00
Variable Designating type of Heat Exchange Solution, ITEM = 0
1--Constant Temp. Equilibrium 0--Full Heat Balance

Nitrate Conc. at Which Denitrification Rate is 1/2 Maximum, mg/l, NCBDN = 0.10
D.O. Conc. at Which Algal Decay Rate is 1/2 Maximum mg/l, KOALDK = 0.50
D.O. Conc. at Which the Rate of Denitrification is Reduced by 1/2, MG/L, KOCBDN = 0.50
D.O. Conc. at Which CBOD Decay rate is 1/2 Maximum Rate, mg/l KOCB1 = 0.50
D.O. Conc. at Which Nitrification Rate is 1/2 Maximum, mg/l KON = 0.50

Rate Coefficient for Stream Reaeration Rate AG = 12.81
Exponent of Velocity in Stream Reaeration Rate, Formulation E1 = 0.50
Depth Exponent in Stream Reaeration Formulation, E2 = 1.50
Coefficient in the Tsivoglou-Wallace Reaeration Equation 1/ft, TSIV = 0.00
Specific Rate Coefficient, Uncorrected, for Denitrification 1/day ADN = 0.10
Specific Rate Coefficient, Uncorrected, for Nitrification 1/day AKN = 0.30
Specific Rate Coefficient, Uncorrected, for Ammonia Adsorption 1/day AKNX = 0.00
Specific Rate Coefficient, Uncorrected, for Phosphate Adsorption, 1/day APO4 = 0.00

Specific Rate Coefficient For Iron Oxidation, 1/day, KFEDK = 0.50
Rate Coefficient For Coliform Mortality, 1/day, KCOLIDK = 1.40
Sediment Oxygen Demand, gr./m**2 SOD = 0.00
Total Nitrogen Conc. at Which Algal Growth Rate is Reduced by 1/2, mg/l KNPOOL = 0.10
Phosphorous Conc. at which Algal Growth Rate Reduced by 1/2, mg/l KPO4X = 0.10
Sediment Denitrification Rate KDSED = 0.00

Macrophyte Density on Channel Surfaces, g/m**2, MACROB = 0.00
Specific Macrophyte Growth Rate, 1/day MACGRO = 0.00
Specific Macrophyte Decay Rate, 1/day MACDKY = 0.00

Algal Growth Rate, mg/l*day KALGRO = 0.02
Algal Decay Rate, mg/l*day KALGDK = 0.10

D.O. Conc. Below Which Oxidation of Iron and Manganese Do Not Occur, mg/l OXIDAT = 1.00
Specific Rate Coefficient Manganese Oxidation, 1/day, KMNDK = 0.50

Figure 22. (Sheet 4 of 14)

Empirical Coefficient Reflecting Thickness of Boundary Layer BK = -0.49
Rate Coefficient Carbonaceous Oxygen Demand, 1/day   AK1 = 0.20
Light Intensity at Which Photosynthesis Rate Reduced by
1/2,   KLITE = 21.00
Rate Coefficient for Org-N Decay To NH3   ACK = 0.00
Surface Light Intensity at Local Noon,   HNEFSW = 0.00
Is Wind Driven Reaeration Used for Segment ?, T or F, QWIND(0) = F

Non-algal Portion of Light Extinction Coefficient   LAMBDA0 = 1.00
Linear Algal Self Shading Coefficient   LAMBDA1 = 0.00
Nonlinear Algal Self Shading Coefficient   LAMBDA2 = 0.00
Conversion Factor From Algae To Chlorophyll,   ALPHA0 = 0.00

Settling Rate for Organic Nitrogen,   KNSET = 0.00
Benthal Source Rate for Ammonia, g/m**2   ABSR = 0.00
Settling Rate for CBOD, m/day   CBODSR = 0.00
Fraction of Algal and Macrophyte Decay
Which Goes to CBOD,   FCBOD = 0.00

Decay Rate For Organic - P,   KPDK = 0.00
Settling Rate Coefficient For Org-P,   KPSET = 0.00
Benthic Source Term for Dissolved P,   BENPO4 = 0.00

For Segment # 3   Desc: MAIN STEM TRIBUTARY ABOVE REREG DAM
              IDAMO = 0   DAMKO =   0.

    Constant Data for Segment No.   3

    NOTE: These Data may be Corrected for Temp., D.O., etc. During
          Program Execution

Stream Segment Temperature, Deg. C,   TEMP = 23.00
Rate Coefficient for Bottom Heat Exchange (1/DAY)   ATB = 0.00
Rate Coefficient for Surface Heat Exchange, W/M**2 DEG. C,   ATS = 0.00
Source/Sink Term for Bottom Heat Exchange, DEG. C,   TSINK = 10.00
Variable Designating type of Heat Exchange Solution,   ITEM = 0
1--Constant Temp. Equilibrium 0--Full Heat Balance

Nitrate Conc. at Which Denitrification Rate is 1/2
Maximum, mg/l,   NCBDN = 0.10
D.O. Conc. at Which Algal Decay Rate is 1/2 Maximum
mg/l,   KOALDK = 0.50
D.O. Conc. at Which the Rate of Denitrification is
Reduced by 1/2, MG/L,   KOCBDN = 0.50

Figure 22.   (Sheet 5 of 14)

D.O. Conc. at Which CBOD Decay rate is 1/2 Maximum Rate, mg/l   KOCB1 = 0.50
D.O. Conc. at Which Nitrification Rate is 1/2 Maximum, mg/l   KON = 0.50

Rate Coefficient for Stream Reaeration Rate   AG = 12.81
Exponent of Velocity in Stream Reaeration Rate, Formulation   E1 = 0.50
Depth Exponent in Stream Reaeration Formulation,   E2 = 1.50
Coefficient in the Tsivoglou-Wallace Reaeration Equation 1/ft,   TSIV = 0.00
Specific Rate Coefficient, Uncorrected, for Denitrification 1/day   ADN = 0.10
Specific Rate Coefficient, Uncorrected, for Nitrification 1/day   AKN = 0.30
Specific Rate Coefficient, Uncorrected, for Ammonia Adsorption 1/day   AKNX = 0.00
Specific Rate Coefficient, Uncorrected, for Phosphate Adsorption, 1/day   APO4 = 0.00

Specific Rate Coefficient For Iron Oxidation, 1/day,   KFEDK = 0.50
Rate Coefficient For Coliform Mortality, 1/day,   KCOLIDK = 1.40
Sediment Oxygen Demand, gr./m**2   SOD = 0.00
Total Nitrogen Conc. at Which Algal Growth Rate is Reduced by 1/2, mg/l   KNPOOL = 0.10
Phosphorous Conc. at which Algal Growth Rate Reduced by 1/2, mg/l   KPO4X = 0.10
Sediment Denitrification Rate   KDSED = 0.00

Macrophyte Density on Channel Surfaces, g/m**2,   MACROB = 0.00
Specific Macrophyte Growth Rate, 1/day   MACGRO = 0.00
Specific Macrophyte Decay Rate, 1/day   MACDKY = 0.00

Algal Growth Rate, mg/l*day   KALGRO = 0.02
Algal Decay Rate, mg/l*day   KALGDK = 0.10

D.O. Conc. Below Which Oxidation of Iron and Manganese Do Not Occur, mg/l   OXIDAT = 1.00
Specific Rate Coefficient Manganese Oxidation, 1/day,   KMNDK = 0.50

Empirical Coefficient Reflecting Thickness of Boundary Layer   BK = -0.49
Rate Coefficient Carbonaceous Oxygen Demand, 1/day   AK1 = 0.20
Light Intensity at Which Photosynthesis Rate Reduced by 1/2,   KLITE = 21.00
Rate Coefficient for Org-N Decay To NH3   ACK = 0.00
Surface Light Intensity at Local Noon,   HNEFSW = 0.00
Is Wind Driven Reaeration Used for Segment 7, T or F,QWIND(0) =   F

Non-algal Portion of Light Extinction Coefficient   LAMBDA0 = 1.00
Linear Algal Self Shading Coefficient   LAMBDA1 = 0.00
Nonlinear Algal Self Shading Coefficient   LAMBDA2 = 0.00
Conversion Factor From Algae To Chlorophyll,   ALPHA0 = 0.00

Figure 22.   (Sheet 6 of 14)

171

Settling Rate for Organic Nitrogen,
Benthal Source Rate for Ammonia, g/m**2
Settling Rate for CBOD, m/day
Fraction of Algal and Macrophyte Decay
Which Goes to CBOD,

KNSET = 0.00
ABSR = 0.00
CBODSR = 0.00

FCBOD = 0.00

Decay Rate For Organic - P,
Settling Rate Coefficient For Org-P,
Benthic Source Term for Dissolved P,

KPDK = 0.00
KPSET = 0.00
BENPO4 = 0.00

SAMPLE RIV1Q DATA SET - USER'S MANUAL
Initial Conditions
===================

	TEMP	CBODNS	ORGAN	NH3N	NO3N	ORG-P	PO4	MN	FE	DO	CLFM	ALGAE
1	23.40	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
2	22.80	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.60	0.00	0.00
3	22.80	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.70	0.00	0.00
4	24.30	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
5	24.40	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.30	0.00	0.00
6	24.50	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.30	0.00	0.00
7	24.70	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.30	0.00	0.00
8	25.10	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
9	25.40	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
10	25.70	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
11	25.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
12	26.00	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.30	0.00	0.00
13	26.00	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.40	0.00	0.00
14	25.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
15	25.70	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
16	25.30	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.60	0.00	0.00
17	24.80	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.60	0.00	0.00
18	24.30	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
19	23.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.60	0.00	0.00
20	23.80	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.60	0.00	0.00
21	23.70	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
22	23.80	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
23	24.00	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.40	0.00	0.00
24	25.00	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.30	0.00	0.00
25	25.60	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.30	0.00	0.00
26	26.00	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
27	25.40	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.50	0.00	0.00
28	24.20	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.40	0.00	0.00
29	23.40	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
30	23.70	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.10	0.00	0.00
31	23.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.10	0.00	0.00
32	23.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.20	0.00	0.00
33	23.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.10	0.00	0.00
34	23.90	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.10	0.00	0.00
35	25.00	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.20	0.00	0.00
36	25.10	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.40	0.00	0.00

Figure 22. (Sheet 7 of 14)

	TEMP	CBODNS	ORGAN	NH3N	NO3N	ORG-P	PO4	MN	FE	DO	CLFM	ALGAE
37	25.40	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.50	0.00	0.00
38	26.20	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.50	0.00	0.00
39	26.30	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.60	0.00	0.00
40	26.20	0.01	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.60	0.00	0.00

Lateral Inflow Concentrations
==============================

	TEMP	CBODNS	ORGAN	NH3N	NO3N	ORG-P	PO4	MN	FE	DO	CLFM	ALGAE
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

**** BOUNDARY CONDITIONS SPECIFIED FOR SEGMENT****   1
WILL BE READ FROM FILE = SAMPLEQ1.BCF
INTERPOLATION OPTION   STEP FUNCTION

Figure 22.   (Sheet 8 of 14)

Figure 22. (Sheet 9 of 14)

```
**** BOUNDARY CONDITIONS SPECIFIED FOR SEGMENT****  3
     WILL BE READ FROM FILE =  SAMPLEQ2.BCF
     INTERPOLATION OPTION    STEP FUNCTION


**************METEOROLOGICAL DATA**************
DUC= 0.12 LATUDC= 37.00 LONTUC= 37.00 LSMC= 74.00
     INTERPOLATION OPTION    STEP FUNCTION


**TIME VARYING LATERAL INFLOWS SPECIFIED AT        X-SECTIONS
   25   26   27   28   29
FOR WATER QUALITY CONSTITUENTS:
   1   TEMP
   2   CBODNS
   3   ORGAN
   4   NH3N
     INTERPOLATION OPTION    LINEAR INT


CE-QUAL-RIV1 WATER QUALITY MODEL

     SAMPLE RIV1Q DATA SET - USER'S MANUAL
CLOUD COVER  0.800 WIND SPEED=  4.00 DRY BULB   0.00 SEGMENT  1: MAIN STEM ABOVE REREG DAM
     82.00 WET BULB TEMP =   61.00 ATMOSPHERIC PRESSURE =   28.81
```

YEAR 1983 MONTH   June   DAY 18 HOUR 18

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
1	27.76	23.40	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.5	0.00	0.00
2	27.00	22.80	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.6	0.00	0.00
3	26.01	22.80	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.7	0.00	0.00
4	25.25	24.30	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
5	24.43	24.40	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
6	23.59	24.50	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
7	22.76	24.70	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
8	21.82	25.10	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
9	21.13	25.40	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
10	20.33	25.70	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
11	19.54	25.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
12	18.74	26.00	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
13	17.94	26.00	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
14	17.06	25.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
15	16.18	25.70	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
16	15.29	25.30	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.4	0.00	0.00

```
                        0.00 SEGMENT  2: MAIN STEM BELOW REREG DAM
```

YEAR 1983 MONTH   June   DAY 18 HOUR 18

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
17	15.29	24.80	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.5	0.00	0.00
18	14.47	24.30	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.5	0.00	0.00
19	13.65	23.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.6	0.00	0.00
20	12.82	23.80	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.6	0.00	0.00
21	12.11	23.70	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.5	0.00	0.00
22	11.41	23.80	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.5	0.00	0.00
23	10.70	24.00	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.4	0.00	0.00
24	9.85	25.00	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00			

Figure 22. (Sheet 9 of 14)

174

YEAR 1983 MONTH June DAY 18 HOUR

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
25 9.00	25.60	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
26 8.15	26.00	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.3	0.00	0.00
27 7.29	25.40	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.5	0.00	0.00
28 6.20	24.20	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.4	0.00	0.00
29 5.11	23.40	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
30 4.01	23.70	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.2	0.00	0.00
31 2.92	23.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.1	0.00	0.00
32 1.95	23.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.1	0.00	0.00
33 0.97	23.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.1	0.00	0.00
34 0.00	23.90	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	8.1	0.00	0.00

YEAR 1983 MONTH June DAY 18 HOUR 0.00 SEGMENT 3: MAIN STEM TRIBUTARY ABOVE REREG DAM

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
35 2.51	25.00	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.2	0.00	0.00
36 2.03	25.10	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.4	0.00	0.00
37 1.47	25.40	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.5	0.00	0.00
38 0.90	26.20	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.5	0.00	0.00
39 0.37	26.30	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.6	0.00	0.00
40 0.00	26.20	0.0	0.01	0.10	0.01	0.00	0.02	0.00	0.00	7.6	0.00	0.00

CE-QUAL-RIV1 WATER QUALITY MODEL

SAMPLE RIV1Q DATA SET - USER'S MANUAL
CLOUD COVER 0.200 WIND SPEED= 3.00 DRY BULB 78.00 WET BULB TEMP = 58.00 ATMOSPHERIC PRESSURE = 28.79

YEAR 1983 MONTH June DAY 18 HOUR 11.99 SEGMENT 1: MAIN STEM ABOVE REREG DAM

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
1 27.76	12.00	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	2.5	0.00	0.00
2 27.00	12.43	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	2.7	0.00	0.00
3 26.01	13.03	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	3.6	0.00	0.00
4 25.25	17.38	0.9	0.36	0.16	0.10	0.00	0.04	0.00	0.00	5.4	0.00	0.00
5 24.43	17.96	0.9	0.35	0.16	0.10	0.00	0.04	0.00	0.00	5.9	0.00	0.00
6 23.59	18.35	0.8	0.36	0.16	0.10	0.00	0.04	0.00	0.00	6.5	0.00	0.00
7 22.76	17.24	0.8	0.36	0.16	0.10	0.00	0.04	0.00	0.00	6.5	0.00	0.00
8 21.82	21.98	0.4	0.17	0.10	0.06	0.00	0.03	0.00	0.00	7.7	0.00	0.00
9 21.13	24.03	0.2	0.09	0.08	0.04	0.00	0.02	0.00	0.00	8.1	0.00	0.00
10 20.33	23.39	0.1	0.07	0.09	0.04	0.00	0.02	0.00	0.00	7.9	0.00	0.00
11 19.54	23.42	0.1	0.03	0.08	0.03	0.00	0.02	0.00	0.00	8.1	0.00	0.00
12 18.74	24.00	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
13 17.94	24.47	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
14 17.06	25.00	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.1	0.00	0.00
15 16.18	25.46	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.1	0.00	0.00
16 15.29	25.58	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.1	0.00	0.00

YEAR 1983 MONTH June DAY 18 HOUR 11.99 SEGMENT 2: MAIN STEM BELOW REREG DAM

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
17 15.29	25.58	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.1	0.00	0.00
18 14.47	25.85	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.1	0.00	0.00
19 13.65	25.99	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
20 12.82	26.03	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
21 12.11	26.06	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
22 11.41	25.99	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
23 10.70	25.90	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.2	0.00	0.00
24 9.85	25.76	0.1	0.05	0.08	0.03	0.00	0.02	0.00	0.00	7.9	0.00	0.00
25 9.00	25.63	0.8	0.19	0.15	0.02	0.00	0.02	0.00	0.00	7.4	0.00	0.00
26 8.15	25.82	1.0	0.24	0.17	0.03	0.00	0.02	0.00	0.00	7.2	0.00	0.00
27 7.29	25.67	0.8	0.20	0.14	0.03	0.00	0.02	0.00	0.00	7.0	0.00	0.00
28 6.20	25.05	0.6	0.15	0.11	0.03	0.00	0.02	0.00	0.00	6.9	0.00	0.00
29 5.11	24.85	0.6	0.15	0.11	0.03	0.00	0.02	0.00	0.00	6.9	0.00	0.00

Figure 22. (Sheet 10 of 14)

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
30	4.01	24.41	0.5	0.11	0.10	0.03	0.00	0.02	0.00	0.00	7.2	0.00	0.00
31	2.92	24.34	0.3	0.09	0.10	0.03	0.00	0.02	0.00	0.00	7.6	0.00	0.00
32	1.95	23.86	0.2	0.06	0.09	0.03	0.00	0.02	0.00	0.00	7.9	0.00	0.00
33	0.97	24.41	0.1	0.03	0.08	0.03	0.00	0.02	0.00	0.00	8.0	0.00	0.00
34	0.00	23.72	0.0	0.01	0.08	0.03	0.00	0.02	0.00	0.00	8.3	0.00	0.00

YEAR 1983 MONTH June DAY 18 HOUR 11.99 SEGMENT 3: MAIN STEM TRIBUTARY ABOVE REREG DAM

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
35	2.51	20.00	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.0	0.00	0.00
36	2.03	20.45	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00
37	1.47	21.01	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.2	0.00	0.00
38	0.90	21.21	0.9	0.10	0.02	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00
39	0.37	23.38	0.8	0.08	0.06	0.10	0.00	0.02	0.00	0.00	7.6	0.00	0.00
40	0.00	25.23	0.7	0.07	0.08	0.10	0.00	0.02	0.00	0.00	7.3	0.00	0.00

CE-QUAL-RIV1 WATER QUALITY MODEL

SAMPLE RIV1Q DATA SET - USER'S MANUAL
CLOUD COVER 0.500 WIND SPEED= 0.00 DRY BULB 61.00 WET BULB TEMP = 52.00 ATMOSPHERIC PRESSURE = 28.84

YEAR 1983 MONTH June DAY 18 HOUR 23.98 SEGMENT 1: MAIN STEM ABOVE REREG DAM

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
1	27.76	15.00	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	4.0	0.00	0.00
2	27.00	14.76	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	4.0	0.00	0.00
3	26.01	14.35	1.0	0.50	0.14	0.10	0.00	0.04	0.00	0.00	4.5	0.00	0.00
4	25.25	16.97	0.9	0.37	0.14	0.10	0.00	0.04	0.00	0.00	6.1	0.00	0.00
5	24.43	15.44	0.9	0.37	0.14	0.10	0.00	0.04	0.00	0.00	6.1	0.00	0.00
6	23.59	13.10	0.9	0.36	0.13	0.10	0.00	0.04	0.00	0.00	6.9	0.00	0.00
7	22.76	18.07	0.8	0.36	0.14	0.11	0.00	0.04	0.00	0.00	6.7	0.00	0.00
8	21.82	19.70	0.7	0.34	0.14	0.12	0.00	0.04	0.00	0.00	6.8	0.00	0.00
9	21.13	19.78	0.5	0.25	0.11	0.12	0.00	0.03	0.00	0.00	7.2	0.00	0.00
10	20.33	21.13	0.2	0.13	0.08	0.10	0.00	0.02	0.00	0.00	7.7	0.00	0.00
11	19.54	23.27	0.1	0.06	0.07	0.07	0.00	0.02	0.00	0.00	8.1	0.00	0.00
12	18.74	24.08	0.0	0.03	0.07	0.05	0.00	0.02	0.00	0.00	8.1	0.00	0.00
13	17.94	24.46	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.1	0.00	0.00
14	17.06	25.00	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.1	0.00	0.00
15	16.18	25.50	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.1	0.00	0.00
16	15.29	25.78	0.0	0.02	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00

YEAR 1983 MONTH June DAY 18 HOUR 23.98 SEGMENT 2: MAIN STEM BELOW REREG DAM

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
17	15.29	25.78	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
18	14.47	25.63	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
19	13.65	25.39	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
20	12.82	25.24	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
21	12.11	25.16	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
22	11.41	25.17	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
23	10.70	25.22	0.0	0.01	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
24	9.85	25.36	0.0	0.06	0.06	0.05	0.00	0.02	0.00	0.00	8.0	0.00	0.00
25	9.00	25.90	0.1	0.06	0.07	0.04	0.00	0.02	0.00	0.00	7.5	0.00	0.00
26	8.15	26.55	0.8	0.21	0.14	0.04	0.00	0.02	0.00	0.00	7.0	0.00	0.00
27	7.29	27.17	1.1	0.27	0.16	0.04	0.00	0.02	0.00	0.00	6.6	0.00	0.00
28	6.20	28.15	1.1	0.25	0.14	0.05	0.00	0.01	0.00	0.00	6.1	0.00	0.00
29	5.11	28.71	1.0	0.24	0.12	0.05	0.00	0.01	0.00	0.00	5.9	0.00	0.00
30	4.01	28.37	1.0	0.24	0.11	0.05	0.00	0.01	0.00	0.00	6.1	0.00	0.00
31	2.92	28.02	0.9	0.20	0.10	0.05	0.00	0.01	0.00	0.00	6.5	0.00	0.00
32	1.95	27.84	0.8	0.07	0.10	0.05	0.00	0.02	0.00	0.00	6.8	0.00	0.00
33	0.97	25.55	0.3	0.07	0.07	0.05	0.00	0.02	0.00	0.00	7.7	0.00	0.00
34	0.00	24.73	0.0	0.02	0.07	0.05	0.00	0.02	0.00	0.00	8.1	0.00	0.00

Figure 22. (Sheet 11 of 14)

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
35	2.51	20.00	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.0	0.00	0.00
36	2.03	19.89	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00
37	1.47	19.64	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.2	0.00	0.00
38	0.90	20.42	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.2	0.00	0.00
39	0.37	21.78	0.9	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00
40	0.00	22.58	0.9	0.10	0.02	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00

CE-QUAL-RIV1 WATER QUALITY MODEL

SAMPLE RIV1Q DATA SET - USER'S MANUAL
CLOUD COVER 0.500 WIND SPEED= 0.00 DRY BULB 55.00 WET BULB TEMP = 50.00 ATMOSPHERIC PRESSURE = 28.86

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
1	27.76	12.00	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	2.5	0.00	0.00
2	27.00	12.24	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	2.6	0.00	0.00
3	26.01	12.56	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	3.3	0.00	0.00
4	25.25	14.64	1.0	0.40	0.15	0.10	0.00	0.04	0.00	0.00	5.5	0.00	0.00
5	24.43	14.90	1.0	0.39	0.14	0.10	0.00	0.04	0.00	0.00	6.0	0.00	0.00
6	23.59	14.46	0.9	0.38	0.13	0.11	0.00	0.04	0.00	0.00	7.1	0.00	0.00
7	22.76	16.79	0.9	0.37	0.13	0.12	0.00	0.04	0.00	0.00	7.0	0.00	0.00
8	21.82	16.16	0.9	0.37	0.13	0.12	0.00	0.04	0.00	0.00	6.9	0.00	0.00
9	21.13	16.65	0.9	0.36	0.12	0.13	0.00	0.04	0.00	0.00	6.9	0.00	0.00
10	20.33	18.48	0.8	0.35	0.11	0.12	0.00	0.04	0.00	0.00	7.4	0.00	0.00
11	19.54	20.10	0.6	0.30	0.09	0.10	0.00	0.03	0.00	0.00	7.7	0.00	0.00
12	18.74	21.79	0.4	0.10	0.07	0.08	0.00	0.02	0.00	0.00	7.7	0.00	0.00
13	17.94	23.29	0.2	0.04	0.06	0.06	0.00	0.02	0.00	0.00	8.0	0.00	0.00
14	17.06	24.26	0.1	0.02	0.06	0.06	0.00	0.02	0.00	0.00	8.0	0.00	0.00
15	16.18	24.89	0.0	0.02	0.05	0.06	0.00	0.02	0.00	0.00	8.0	0.00	0.00
16	15.29	25.25	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.0	0.00	0.00

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
17	15.29	25.25	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.0	0.00	0.00
18	14.47	25.56	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.0	0.00	0.00
19	13.65	25.74	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.1	0.00	0.00
20	12.82	25.81	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.1	0.00	0.00
21	12.11	25.86	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.1	0.00	0.00
22	11.41	25.79	0.0	0.01	0.05	0.06	0.00	0.02	0.00	0.00	8.2	0.00	0.00
23	10.70	25.69	0.0	0.01	0.06	0.06	0.00	0.02	0.00	0.00	7.8	0.00	0.00
24	9.85	25.50	0.1	0.06	0.06	0.06	0.00	0.02	0.00	0.00	7.3	0.00	0.00
25	9.00	25.30	0.8	0.21	0.13	0.05	0.00	0.02	0.00	0.00	7.1	0.00	0.00
26	8.15	25.44	1.1	0.27	0.15	0.05	0.00	0.02	0.00	0.00	6.7	0.00	0.00
27	7.29	25.33	1.1	0.27	0.14	0.05	0.00	0.01	0.00	0.00	6.5	0.00	0.00
28	6.20	24.87	1.1	0.26	0.12	0.05	0.00	0.01	0.00	0.00	6.6	0.00	0.00
29	5.11	24.82	1.0	0.25	0.11	0.05	0.00	0.01	0.00	0.00	6.8	0.00	0.00
30	4.01	24.45	1.1	0.25	0.11	0.06	0.00	0.01	0.00	0.00	6.8	0.00	0.00
31	2.92	24.76	1.1	0.28	0.10	0.07	0.00	0.01	0.00	0.00	7.2	0.00	0.00
32	1.95	25.44	0.5	0.15	0.07	0.07	0.00	0.02	0.00	0.00	7.9	0.00	0.00
33	0.97	25.67	0.1	0.03	0.05	0.06	0.00	0.02	0.00	0.00	7.9	0.00	0.00
34	0.00	24.59											

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
35	2.51	20.00	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.0	0.00	0.00
36	2.03	20.34	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00

Figure 22. (Sheet 12 of 14)

37	1.47	20.80	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.2	0.00	0.00
38	0.90	20.22	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.00	0.00
39	0.37	19.94	0.9	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.00	0.00
40	0.00	19.96	0.9	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.00	0.00

CE-QUAL-RIV1 WATER QUALITY MODEL

SAMPLE RIV1Q DATA SET - USER'S MANUAL

CLOUD COVER 0.500 WIND SPEED= 3.00 DRY BULB 23.95   62.00 WET BULB TEMP = 54.00 ATMOSPHERIC PRESSURE = 28.89

YEAR 1983 MONTH June DAY 19 HOUR   SEGMENT 1: MAIN STEM ABOVE REREG DAM

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
1	27.76	12.00	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	2.5	0.000	0.00
2	27.00	11.98	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	2.6	0.000	0.00
3	26.01	12.01	1.0	0.50	0.14	0.10	0.00	0.04	0.00	0.00	3.4	0.000	0.00
4	25.25	14.56	1.0	0.39	0.14	0.10	0.00	0.04	0.00	0.00	5.1	0.000	0.00
5	24.43	14.51	1.0	0.39	0.14	0.10	0.00	0.04	0.00	0.00	5.5	0.000	0.00
6	23.59	14.69	0.9	0.39	0.14	0.10	0.00	0.04	0.00	0.00	5.9	0.000	0.00
7	22.76	16.10	0.9	0.39	0.13	0.11	0.00	0.04	0.00	0.00	6.0	0.000	0.00
8	21.82	16.68	0.9	0.39	0.12	0.12	0.00	0.04	0.00	0.00	6.3	0.000	0.00
9	21.13	17.00	0.9	0.38	0.12	0.12	0.00	0.04	0.00	0.00	6.7	0.000	0.00
10	20.33	17.45	0.8	0.37	0.11	0.13	0.00	0.04	0.00	0.00	7.0	0.000	0.00
11	19.54	18.26	0.7	0.35	0.10	0.13	0.00	0.04	0.00	0.00	6.9	0.000	0.00
12	18.74	19.89	0.6	0.30	0.08	0.13	0.00	0.04	0.00	0.00	7.0	0.000	0.00
13	17.94	21.77	0.4	0.21	0.06	0.11	0.00	0.03	0.00	0.00	7.3	0.000	0.00
14	17.06	23.54	0.2	0.10	0.04	0.09	0.00	0.02	0.00	0.00	7.7	0.000	0.00
15	16.18	24.62	0.1	0.04	0.04	0.08	0.00	0.02	0.00	0.00	7.9	0.000	0.00
16	15.29	25.12	0.0	0.02	0.04	0.07	0.00	0.02	0.00	0.00	8.0	0.000	0.00

YEAR 1983 MONTH June DAY 19 HOUR   SEGMENT 2: MAIN STEM BELOW REREG DAM

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
17	15.29	25.12	0.0	0.02	0.04	0.07	0.00	0.02	0.00	0.00	8.0	0.000	0.00
18	14.47	24.91	0.0	0.02	0.04	0.07	0.00	0.02	0.00	0.00	8.0	0.000	0.00
19	13.65	24.55	0.0	0.01	0.04	0.07	0.00	0.02	0.00	0.00	8.1	0.000	0.00
20	12.82	24.30	0.0	0.01	0.04	0.07	0.00	0.02	0.00	0.00	8.1	0.000	0.00
21	12.11	24.14	0.0	0.01	0.04	0.07	0.00	0.02	0.00	0.00	8.1	0.000	0.00
22	11.41	24.07	0.0	0.01	0.04	0.07	0.00	0.02	0.00	0.00	8.1	0.000	0.00
23	10.70	24.13	0.1	0.01	0.04	0.07	0.00	0.02	0.00	0.00	7.6	0.000	0.00
24	9.85	24.66	0.8	0.06	0.12	0.06	0.00	0.02	0.00	0.00	7.1	0.000	0.00
25	9.00	25.34	1.1	0.21	0.14	0.06	0.00	0.02	0.00	0.00	6.7	0.000	0.00
26	8.15	26.01	1.1	0.27	0.12	0.06	0.00	0.02	0.00	0.00	6.2	0.000	0.00
27	7.29	27.13	1.1	0.27	0.11	0.06	0.00	0.01	0.00	0.00	6.1	0.000	0.00
28	6.20	27.81	1.0	0.26	0.11	0.06	0.00	0.01	0.00	0.00	6.5	0.000	0.00
29	5.11	27.51	1.0	0.25	0.09	0.07	0.00	0.01	0.00	0.00	6.1	0.000	0.00
30	4.01	27.10	1.0	0.25	0.10	0.06	0.00	0.01	0.00	0.00	6.5	0.000	0.00
31	2.92	26.68	1.0	0.25	0.07	0.07	0.00	0.01	0.00	0.00	7.0	0.000	0.00
32	1.95	26.03	0.7	0.19	0.07	0.06	0.00	0.01	0.00	0.00	7.1	0.000	0.00
33	0.97	25.24	0.2	0.06	0.05	0.08	0.00	0.02	0.00	0.00	7.7	0.000	0.00
34	0.00												

YEAR 1983 MONTH June DAY 19 HOUR   SEGMENT 3: MAIN STEM TRIBUTARY ABOVE REREG DAM

	RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
35	2.51	20.00	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.0	0.000	0.00
36	2.03	19.85	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.1	0.000	0.00
37	1.47	19.53	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.000	0.00
38	0.90	19.92	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.000	0.00
39	0.37	20.91	0.9	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.000	0.00
40	0.00	21.37	0.9	0.10	0.01	0.10	0.00	0.01	0.00	0.00		0.000	0.00

Figure 22. (Sheet 13 of 14)

CE-QUAL-RIV1 WATER QUALITY MODEL

SAMPLE RIV1Q DATA SET - USER'S MANUAL
CLOUD COVER   0.500 WIND SPEED=   4.00 DRY BULB   77.00 WET BULB TEMP =   60.00 ATMOSPHERIC PRESSURE =   28.88

YEAR 1983 MONTH   June   DAY 20   HOUR   11.94 SEGMENT 1: MAIN STEM ABOVE REREG DAM

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
27.76	12.00	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	4.0	0.00	0.00
27.00	12.44	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	4.2	0.00	0.00
26.01	13.03	1.0	0.50	0.20	0.10	0.00	0.05	0.00	0.00	5.1	0.00	0.00
25.25	15.84	1.0	0.35	0.13	0.10	0.00	0.04	0.00	0.00	6.7	0.00	0.00
24.43	15.97	0.9	0.35	0.12	0.10	0.00	0.04	0.00	0.00	6.9	0.00	0.00
23.59	15.44	0.9	0.36	0.13	0.11	0.00	0.04	0.00	0.00	6.5	0.00	0.00
22.76	15.07	0.9	0.38	0.13	0.11	0.00	0.04	0.00	0.00	6.0	0.00	0.00
21.82	15.68	0.8	0.39	0.13	0.12	0.00	0.04	0.00	0.00	6.2	0.00	0.00
21.13	16.27	0.8	0.39	0.12	0.12	0.00	0.04	0.00	0.00	6.4	0.00	0.00
20.33	16.88	0.7	0.38	0.11	0.13	0.00	0.04	0.00	0.00	6.7	0.00	0.00
19.54	17.54	0.7	0.37	0.10	0.14	0.00	0.04	0.00	0.00	6.9	0.00	0.00
18.74	18.57	0.6	0.33	0.08	0.14	0.00	0.04	0.00	0.00	7.0	0.00	0.00
17.94	20.15	0.5	0.27	0.06	0.13	0.00	0.03	0.00	0.00	7.4	0.00	0.00
17.06	22.16	0.3	0.17	0.04	0.12	0.00	0.02	0.00	0.00	7.7	0.00	0.00
16.18	23.70	0.1	0.09	0.04	0.10	0.00	0.02	0.00	0.00	7.7	0.00	0.00
15.29	24.45	0.1	0.04	0.04	0.09	0.00	0.02	0.00	0.00	7.8	0.00	0.00

YEAR 1983 MONTH   June   DAY 20   HOUR   11.94 SEGMENT 2: MAIN STEM BELOW REREG DAM

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
15.29	24.45	0.1	0.04	0.04	0.09	0.00	0.02	0.00	0.00	7.8	0.00	0.00
14.47	24.83	0.0	0.03	0.04	0.08	0.00	0.02	0.00	0.00	7.9	0.00	0.00
13.65	25.06	0.0	0.03	0.03	0.08	0.00	0.02	0.00	0.00	8.0	0.00	0.00
12.82	25.17	0.0	0.03	0.03	0.08	0.00	0.02	0.00	0.00	8.1	0.00	0.00
12.11	25.23	0.0	0.03	0.03	0.08	0.00	0.02	0.00	0.00	8.1	0.00	0.00
11.41	25.20	0.0	0.02	0.03	0.08	0.00	0.02	0.00	0.00	8.2	0.00	0.00
10.70	25.13	0.0	0.07	0.03	0.08	0.00	0.02	0.00	0.00	8.2	0.00	0.00
9.85	25.01	0.1	0.07	0.04	0.07	0.00	0.02	0.00	0.00	7.4	0.00	0.00
9.00	24.92	0.9	0.21	0.12	0.07	0.00	0.02	0.00	0.00	7.1	0.00	0.00
8.15	25.15	1.1	0.28	0.14	0.06	0.00	0.02	0.00	0.00	6.8	0.00	0.00
7.29	25.11	1.1	0.27	0.12	0.06	0.00	0.01	0.00	0.00	6.6	0.00	0.00
6.20	24.69	1.1	0.26	0.11	0.06	0.00	0.01	0.00	0.00	6.6	0.00	0.00
5.11	24.60	1.0	0.25	0.10	0.06	0.00	0.01	0.00	0.00	6.9	0.00	0.00
4.01	24.13	1.0	0.25	0.09	0.06	0.00	0.01	0.00	0.00	6.8	0.00	0.00
2.92	24.31	1.0	0.25	0.09	0.08	0.00	0.01	0.00	0.00	7.0	0.00	0.00
1.95	24.84	0.7	0.20	0.06	0.09	0.00	0.01	0.00	0.00	7.4	0.00	0.00
0.97	25.46	0.2	0.20	0.06	0.09	0.00	0.01	0.00	0.00		0.00	0.00
0.00	25.06		0.09	0.04	0.09	0.00	0.02	0.00	0.00		0.00	0.00

YEAR 1983 MONTH   June   DAY 20   HOUR   11.94 SEGMENT 3: MAIN STEM TRIBUTARY ABOVE REREG DAM

RIVER MILE	TEMP (DEG. C)	CBODNS (MG/L)	ORGAN (MG/L)	NH3N (MG/L)	NO3N (MG/L)	ORG-P (MG/L)	PO4 (MG/L)	MN (MG/L)	FE (MG/L)	DO (MG/L)	CLFM COL-DL	ALGAE (MG/L)
2.51	20.00	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.0	0.00	0.00
2.03	20.38	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.1	0.00	0.00
1.47	20.95	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.00	0.00
0.90	20.46	1.0	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.00	0.00
0.37	20.07	0.9	0.10	0.01	0.10	0.00	0.01	0.00	0.00	8.3	0.00	0.00
0.00	19.98										0.00	0.00

Figure 22.   (Sheet 14 of 14)

179

REFERENCES

Abbott, M. B. 1979. <u>Computational Hydraulics.</u> Pitman Publishing Limited, London.

Amein, M., and Chu, H. L. 1975. "Implicit Numerical Modeling of Unsteady Flows," <u>Journal, Hydraulics Division, American Society of Civil Engineers,</u> Vol 101, No. HY6, pp 717-731.

Amein, M., and Fang, C. S. 1970. "Implicit Flood Routing in Natural Channels," <u>Journal, Hydraulics Division, American Society of Civil Engineers,</u> Vol 96, No. HY12, pp 2481-2499.

Baca, R. G., and Arnett, R. C. 1976. "A Limnological Model for Eutrophic Lakes and Impoundments," Report No. 2311102062, Battelle Pacific Northwest Lab, Richland, WA.

Beck, M. B., and Young, P. C. 1975. "A Dynamic Model for BOD-DO Relationships in a Non-Tidal Stream," <u>Water Research,</u> Vol 9, pp 769-776.

Bird, R. B., Stewart, W. E., and Lightfoot, E. N. 1964. <u>Transport Phenomena,</u> John Wiley and Sons, New York.

Bole, J. B., and Allan, J. R. 1978. "Uptake of Phosphorus from Sediment by Aquatic Plants, *Myriophyllum spicatum* and *Hydrilla verticillata*," <u>Water Research,</u> Vol 12, pp 353-358.

Bowie, G. L., Mills, W. B., Porcella, D. B., Campbell, C. L., Pagenkopt, J. R., Rupp, G. L., Johnson, K. M., Chan, P. W. H., and Gherini, S. A. 1985. <u>Rates, Constants, and Kinetics Formulations in Surface Water Quality Modeling,</u> 2nd ed., EPA/600/3-85/040, U.S. Environmental Protection Agency, Athens, GA.

Brown, L. C., and Barnwell, T. O. 1987. "The Enhanced Stream Water Quality Models QUAL2E and QUAL2E-UNCAS: Documentation and User Manual," EPA/600/3-87/007, U.S. Environmental Protection Agency, Environmental Research Laboratory, Athens, GA.

Burden, R. L., and Faires, J. D. 1989. <u>Numerical Analysis.</u> 4th ed., PWS-Kent Pub. Co., Boston, MA.

Carignan, R., and Kalff, J. 1980. "Phosphorus Sources for Aquatic Weeds: Water or Sediments?" <u>Science,</u> Vol 207, pp 987-989.

Carnahan, B., Luther, H. A., and Wilkes, J. O. 1969. <u>Applied Numerical Methods,</u> John Wiley and Sons, New York.

Curtis, E. J. C., Durrant, K., and Harman, M. M. I. 1975. "Nitrification in Rivers in the Trent Basin," <u>Water Research,</u> Vol 9, pp 225-268.

DiToro, D. M., Thomann, R. V., O'Connor, D. J., and Mancini, J. L. 1977. "Estuarine Phytoplankton Biomass Models--Verification Analyses and Preliminary Applications," <u>The Sea,</u> Vol 6, E. D. Goldberg et al., ed., John Wiley and Sons, New York, pp 969-1020.

Donigan, A. S., and Crawford, N. H. 1979. "Water Quality Model for Agricultural Runoff," <u>Modeling of Rivers,</u> H. W. Shen, ed., John Wiley and Sons, New York, pp 12-1 - 12-82.

Dortch, M. S., Tillman, D. H., and Bunch, B. W. 1992. <u>Modeling Water Quality of Reservoir Tailwaters.</u> TR-W-92-1, Environmental Laboratory, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Edinger, J. E., Brady, D. K., and Geyer, J. C. 1974. "Heat Exchange and Transport in the Environment," Report No. 14, EPRI Publication No. 74-049-00-3, prepared for the Electric Power Research Institute, Palo Alto, CA.

Elmore, H. L., and Hayes, T. W. 1960. "Solubility of Atmospheric Oxygen in Water," Journal, Sanitation Engineering Division, American Society of Civil Engineers, Vol 86, No. SA4, pp 41-53.

Engelbrecht, R. S., and Morgan, J. J. 1959. "Studies on the Occurrence and Degradation of Condensed Phosphate in Surface Waters," Sewage and Industrial Wastes, Vol 31, pp 458-478.

Fischer, H. B., List, E. J., Koh, R. C., Imberger, J., and Brooks, N. H. 1979, Mixing in Inland and Coastal Waters, Academic Press, New York.

Fitzgerald, G. P. 1970. "Aerobic Lake Muds for the Removal of Phosphorus from Lake Waters," Limnology and Oceanography, Vol 15, pp 550-555.

Foree, E. G., and McCarty, P. L. 1968. The Decomposition of Algae in Anaerobic Waters, Technical Report No. 95, Department of Civil Engineering, Stanford University, Stanford, CA.

Fread, D. L. 1973. "Technique for Implicit Dynamic Routing in Rivers with Tributaries," Water Resources Research, Vol 9, No. 4, pp 918-925.

_____. 1978. DAMBRK: The NWS Dam-Break Flood Forecasting Model, Office of Hydrology, National Weather Service, Silver Spring, MD.

Garland, J. H. N. 1978. "Nitrification in the River Trent," Mathematical Models in Water Pollution Control, A. James, ed., John Wiley and Sons, New York.

Gerloff, G. C. 1969. "Evaluating Nutrient Supplies for the Growth of Aquatic Plants in Natural Waters," Eutrophication: Causes, Consequences, Correctives, National Academy of Sciences, Washington, DC, pp 537-555.

Gummerman, R. C. 1970. "Aqueous Phosphate and Lake Sediment Interaction," Proceedings of the 13th Conference on the Great Lakes Reservoir, Vol 13, pp 675-683.

Hill, A. R., and Sanmugadas, K. 1985. "Denitrification Rates in Relation to Stream Sediment Characteristics," Water Research, Vol 19, No. 12, pp 1579-1586.

Holly, F. M., and Preissman, A. 1978. "Accurate Calculation of Transport in Two Dimensions," Journal, Hydraulics Division, American Society of Civil Engineers, Vol 103, No. HY11, pp 1259-1277.

Hoover, S. R., and Porges, N. 1952. "Assimilation of Dairy Wastes by Activated Sludge--II. The Equation of Synthesis and Rate of Oxygen Utilization," Sewage and Industrial Wastes, Vol 24, pp 306-312.

Jewell, W. J., and McCarty, P. O. 1968. Aerobic Decomposition of Algae and Nutrient Regeneration, Technical Report No. 91, Department of Civil Engineering, Stanford University, Stanford, CA.

Kernighan, B. W., and Ritchie, D. M. 1988. The C Programming Language, 2nd ed., Prentiss Hall, Inglewood Cliffs, NJ.

Knuth, D. E. 1973. "Sorting and Searching," The Art of Computer Programming, Vol 3, Addison-Wesley.

Liggett, J. A. 1975. "Basic Equations of Unsteady Flow," Chapter 2, <u>Unsteady Flow in Open Channels,</u> K. Mahmood and V. Yevjevitch, eds., Water Resources Publication, Fort Collins, CO.

Liggett, J. A., and Cunge, J. A. 1975. "Numerical Methods of Solution of the Unsteady Flow Equations," Chapter 4, <u>Unsteady Flow in Open Channels,</u> K. Mahmood and V. Yevjevitch, eds., Water Resources Publication, Fort Collins, CO.

Lyn, D. A., and Goodwin, P. 1987. "Stability of a General Preissman Scheme," <u>Journal, Hydraulic Engineering, American Society of Civil Engineers,</u> Vol 113, No. 1, pp 16-28.

MacKenthum, K. M., and Ingram, W. M. 1967. <u>Biological Associated Problems in Freshwater Environments,</u> Federal Water Pollution Control Administration, U.S. Department of the Interior, Washington, DC.

Matter, W., Hudson, P., Nestler, J., and Saul, G. 1983. "Movement, Transport, and Scour of Particulate Organic Matter and Aquatic Invertebrates Downstream from a Peaking Hydropower Project," Technical Report E-83-12, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Miller, J. E., and Jennings, M. E. 1979. "Modeling Nitrogen, Oxygen, Chattahoochee River, Ga.," <u>Journal, Environmental Engineering Division, American Society of Civil Engineers,</u> Vol 105, pp 641-653.

O'Connor, D. J., and Dobbins, W. E. 1958. "The Mechanism of Reaeration in Natural Streams," <u>Journal, Sanitary Engineering Division, American Society of Civil Engineers,</u> Vol 82, pp 1-30.

O'Connor, D. J., Thomann, R. V., and DiToro, D. M. 1973. "Dynamic Water Quality Forecasting and Management," EPA-660/3-73-009, Office of Research and Development, U.S. Environmental Protection Agency, Washington, DC.

Olsen, S. 1964. "Phosphate Equilibrium between Reduced Sediments and Water," <u>International Association of Theoretical Applied Limnology,</u> Vol 15, pp 333-341.

Parker, D. S., Stone, R. W., Stenquist, R. J., and Culp, G. 1975. <u>Process Design Manual for Nitrogen Control,</u> Office of Technology Transfer, U.S. Environmental Protection Agency, Washington, DC.

Preissmann, A. 1961. "Propagation des Intumescences dans les Canaux et Rivieres," <u>First Congress of the Calculation Society of France,</u> Grenoble, pp 433-442.

Roesner, L. A., Giguere, P. R., and Evenson, D. E. 1981. "Computer Program Documentation for the Stream Quality Model QUAL-II," EPA 600/9-81-014, Water Resources Engineers, Inc., Walnut Creek, CA.

Rogers, D., and Adams, J. 1976. <u>Mathematical Elements for Computer Graphics,</u> McGraw-Hill, New York.

Ruane, R. J., and Krenkel, P. A. 1977. "Nitrification and Other Factors Affecting Nitrogen in the Holston River," <u>Progress in Water Technology,</u> Vol 8, No. 4-5, pp 209-224.

Schroepfer, G. J., Robins, M. L., and Susag, R. H. 1960. "A Reappraisal of Deoxygenation Rates of Raw Sewage Effluents and Receiving Waters," <u>Journal of Water Pollution Control Federation,</u> Vol 32, pp 1212-1231.

Streeter, H. W., and Phelps, E. B. 1925. "A Study of the Pollution and Natural Purification of the Ohio River--III. Factors Concerned in the Phenomena of Oxidation and Reaeration," Bulletin No. 146, U.S. Public Health Service, Cincinnati, OH.

Streeter, V., and Wylie, E. B. 1979. Fluid Mechanics, McGraw-Hill, New York.

Taylor, A. W., and Kunishi, H. M. 1971. "Phosphate Equilibria on Stream Sediment and Soil in a Watershed Draining and Agricultural Region," Journal of Agriculture Food Chemistry, Vol 19, pp 827-831.

Tsivoglou, E. C., and Wallace, J. R. 1972. Characterization of Stream Reaeration Capacity, EPA-R3-72-012, Ecology Research Service, Office of Research and Monitor, U.S. Environmental Protection Agency, Washington, DC.

Water Resources Engineers, Inc. 1967. "Prediction of Thermal Energy Distribution in Streams and Reservoirs," prepared for the California Department of Fish and Game.

Wheatland, A. B., Barrett, M. J., and Bruce, A. M. 1959. "Some Observations on Denitrification in Rivers and Estuaries," Journal and Proceedings of the Institute of Sewage Purification, pp 149-162.

Wilhelms, S. C., and Smith, D. R. 1981. "Reaeration Through Gated-Conduit Outlet Works," Technical Report E-81-5, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Williams, R. E., and Lewis, M. S. 1986. "Stream Model of Benthic Nitrification-Denitrification," Journal, Environmental Engineering, American Society of Civil Engineers, Vol 112, No. 2, pp 367-386.

Wilson, G. T., and Macleod, N. 1974. "A Critical Appraisal of Empirical Equations and Models for the Prediction of the Coefficient of Reaeration of Deoxygenated Water," Water Research, Vol 8, pp 341-366.

BIBLIOGRAPHY

Banks, R. B. 1975. "Some Features of Wind Action on Shallow Lakes," Journal, Environmental Engineering, American Society of Civil Engineers, Division No. EE5, pp 813-827.

Bansal, M. K. 1976. "Nitrification in Natural Streams," Journal of Water Pollution Control Federation, Vol 48, pp 2380-2393.

Bennet, J. P., and Rathbun, R. E. 1972. "Reaeration in Open Channel Flow," U.S. Geological Society Professional Paper.

Black, W. M., and Phelps, E. B. 1911. "Report Concerning the Location of Sewer Outlets and the Discharge of Sewage into New York Harbor," Contributions from the Sanitary Research Laboratory and Sewage Experiment Station, Vol VII, Massachusetts Institute of Technology, Cambridge, MA.

Bowles, D. S., Fread, D., and Grenney, W. 1977. "Coupled Dynamic Streamflow-Temperature Model," Journal, Environmental Engineering Division, American Society of Civil Engineers, Vol 103, No. HY5.

Brown, L. C. 1974. "Statistical Evaluation of Reaeration Prediction Equations," Journal, Environmental Engineering Division, American Society of Civil Engineers, Vol 100, pp 1051-1068.

Camp, T. R., and Meserve, R. L. 1974. Water and Its Impurities, 2d ed., Dowden, Hutchinson & Ross, Inc., Stroudsburg, PA.

Churchill, M. A., Elmore, H. L., and Buckingham, R. A. 1962. "The Prediction of Stream Reaeration Rates," International Journal of Air and Water Pollution, Vol 6, pp 467-504.

Covar, A. P. 1976. "Selecting the Proper Reaeration Coefficient for Use in Water Quality Models," Proceedings of the Conference on Environmental Modelling and Simulation, U.S. Environmental Protection Agency, Cincinnati, OH, pp 340-343.

Edinger, J. E., Duttweiler, D., and Geyer, J. 1968. "The Response of Water Temperatures to Meteorological Conditions," Water Resources Research, Vol 4, No. 5.

Elmore, H. C., and West, W. F. 1961. "Effect of Water Temperature on Stream Reaeration," Journal, Sanitation Engineering Division, American Society of Civil Engineers, Vol 87, No. SA6, pp 59-71.

Ford, D. E., and Johnson, L. S. 1986. "An Assessment Reservoir Mixing Processes," Technical Report E-86-7, prepared by Ford, Thornton, Norton and Associates, Ltd., Little Rock, AR, for the U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Fread, D. L., and Smith, C. F. 1978. "Calibration Technique for 1-D Unsteady Flow Models," Journal, Hydraulics Division, American Society of Civil Engineers, Vol 104, No. HY7, pp 1027-1044.

Frost, W. H., and Streeter, H. W. 1924. Public Health Bulletin, No. 143, U.S. Public Health Service, Washington, DC.

Garland, J. H. N., and Rolley, H. L. J. 1977. "Studies of River Water Quality in the Lancashire Tame," Water Pollution Control, Vol 76, pp 301-326.

Haslam, S. M. 1978. River Plants, Cambridge University Press, Cambridge, MA.

Holler, A. G. 1970. "Reaeration of Discharge Through Hydraulic Structures," U.S. Army Engineer Division, Ohio River, Cincinnati, OH.

Holly, F. M., and Cunge, J. 1979. "Coupled Dynamic Streamflow - Temperature Models Discussion," Journal, Hydraulics Division, American Society of Civil Engineers, Vol 104, pp 316-318.

Hynes, H. B. N. 1970. The Ecology of Running Waters, University of Toronto Press, Toronto.

International Business Machines Corp. 1968. "IBM System/360 Scientific Subroutine Package," 360A-CM-03X Version III Programmer's Manual.

_____. 1974. "IBM System/360 and System/370 FORTRAN IV Language," CG28-6515-10.

Kittrell, F. W., and Furfari, S. A. 1963. "Observations of Coliform Bacteria in Streams," Journal of Water Pollution Control Federation, Vol 35, pp 1361-1385.

Krombach, H., and Barthel, J. 1963. "Investigation of a Small Watercourse Accidentally Polluted by Phenol Compounds," International Journal of Air and Water Pollution, Vol 7, pp 39-46.

Leopold, L. B., Wolman, M. G., and Miller, J. P. 1964. Fluvial Processes in Geomorphology, W. H. Freeman and Co., San Francisco, CA.

Liggett, J. A., and Woolhiser, D. A. 1967. "Difference Solutions of the Shallow Water Equation," Journal, Engineering Mechanics Division, American Society of Civil Engineers, Vol 93, No. EM2, pp 39-71.

Novotny, V. 1969. "Boundary Layer Effect on the Course of the Self-Purification of Small Streams," Advances in Water Pollution Research, S. H. Jenkins, ed., Pergamon Press, Oxford, pp 39-55.

O'Connor, D. J. 1983. "Wind Effects on Gas-Liquid Transfer Coefficients," Journal, Environmental Engineering Division, American Society of Civil Engineers, Vol 109, pp 731-752.

Owens, M., Edwards, R. W., and Gibbs, J. W. 1964. "Some Reaeration Studies in Streams," International Journal of Air and Water Pollution, Vol 8, pp 469-486.

Strand, S. 1975. Statistics of the BOD Parameters, M.S. thesis, Ohio State University, Columbus, OH.

Theriault, E. J. 1927. The Oxygen Demand of Polluted Waters, Bulletin No. 173, U.S. Public Health Service, Washington, DC.

U.S. Environmental Protection Agency. 1987. The Enhanced Stream Water Quality Models QUAL2E and QUAL2E-UNCAS, Documentation and User Manual, EPA/600/3-87/007, Athens, GA.

Velz, C. J. 1939. "Deoxygenation and Reoxygenation," Transactions of the American Society of Civil Engineers, Vol 104, pp 560-578.

_____. 1947. "Factors Influencing Self-Purification and Their Relation to Pollution Abatement," Sewage Works Journal, Vol 19, pp 629-644.

_____. 1970. Applied Stream Sanitation, John Wiley and Sons, New York.

Whitton, B. A., ed. 1975. River Ecology, University California Press, Berkeley, CA.

Wright, R. M., and McDonnell, A. J. 1979. "In-Stream Deoxygenation Rate Prediction," _Journal, Environmental Engineering Division, American Society of Civil Engineers,_ Vol 105, pp 323-335.

Yalin, M. S. 1977. _Mechanics of Sediment Transport,_ 2d ed., Pergamon Press, Oxford.

Yen, B. C. 1979. "Unsteady Flow Mathematical Modeling Techniques," _Modeling of Rivers,_ H. W. Shen, ed. John Wiley and Sons, New York.

Zison, S. W., Mills, W. B., Deimer, D., and Chen, C. W. 1978. _Rates, Constants, and Kinetics Formulations in Surface Water Quality Modeling,_ EPA-600/3-78-105, Environmental Research Laboratory, U.S. Environmental Protection Agency, Athens, GA.

APPENDIX A: LIST OF THE HYDRODYNAMIC CODE RIV1H

```
C  **************************************************************
C                   CE-QUAL-RIV1, VERSION 2                     *
C                   RIV1H CODE - HYDRODYNAMIC MODEL             *
C                         JULY, 1995                            *
C                                                               *
C      CE-QUAL-RIV1H was originally developed by Ohio State     *
C           University, Bedford, Sykes, and Libicki             *
C                                                               *
C                   and later modified by                       *
C                  M. Dortch and T. Schneider                   *
C       Water Quality and Contaminant Modeling Branch           *
C                  Environmental Laboratory                     *
C               USAE Waterways Experiment Station               *
C                 Vicksburg, Mississippi, and                   *
C                     D. M. Griffin, Jr.                        *
C                  Louisiana Tech University                    *
C                     Ruston, Louisiana                         *
C                                                               *
C       This version has been further modified by:              *
C                                                               *
C                  Tim Wool and James Martin                    *
C                     AScI Corporation                          *
C                     Athens, Georgia                           *
C                                                               *
C                                                               *
C  **************************************************************
         PROGRAM RIV1H

         REAL KE1
         INTEGER ORDER,FEEDS

         INCLUDE 'RIV1H.CMN'

         CHARACTER*6 XSCODE(IND1)

         CHARACTER MONTH*9
         INTEGER SYEAR,SMONTH,SDAY,EYEAR,EMONTH,EDAY

         CHARACTER*30 RIVPATH

         CHARACTER*12 FILES(30), MESSFIL

         LOGICAL CTRL

         COMMON/ABLOCK/ MTIME, NS, TITLE, MNODE

         COMMON/XSDATA/ IXS, HXS(MAXXS), AXS(MAXXS), BXS(MAXXS)

         DIMENSION DX1(IND1), Q(IND1), A(IND1), B(IND1), I3(IND1),
        .         C1(IND1), C2(IND1), C3(IND1), H(IND1), EL(IND1),
        .         QL(IND1), QLC(IND1), QLT(IND1), CN1(IND1),
        .         KE1(IND1), R(IND3), AA(IND2), XC(IND1), XM(IND1),
        .         RMILE(IND1), C(IND3), LIB(IBRAN), ID(IBRAN),
        .         NODE1(IBRAN), NNODE(IBRAN), FEEDS(IBRAN),
        .         ORDER(IBRAN), JNODE(IBRAN), JBCU(IBRAN), JBCD(IBRAN),
        .         BCU(IBRAN), BCD(IBRAN), JT(IBRAN), COSP(IBRAN),
        .         IT0(IBRAN), IT1(IBRAN), IT2(IBRAN), T(4,IBRAN),
        .         IBC(IBRAN+1), BC(IBRAN+1), Z(IND1), XMAN(IND1),
        .         DNDH(IND1), AX(IND1), IXSA(IND1),IXSB(IND1),DBDH(IND1)

         DIMENSION TPRNT(50), TPR(50)
```

A2

```
C  ************************************************************************
C                  200 update intervals, 12 boundary conditions
C  ************************************************************************

       CHARACTER*1 BTU,BTD
       CHARACTER*40 SNAME(IBRAN)
       CHARACTER*80 TITLE
       DATA GR/32.174/,THETA/0.55/,TOLER/0.001/,BETA/1.0/,IPRINT/1/
       IFIRST = 0
C
       OPEN(UNIT=1,FILE='RIV1H.CTL',IOSTAT=ISTAT,STATUS='OLD')
       IF (ISTAT .NE. 0)THEN
          WRITE(6,6000)
 6000     FORMAT(' RIV1H Control File Not Found ')
          STOP
        ENDIF
C
       READ (1,5000)INFIL
       READ (1,5000)LATFIL
       READ (1,5000)XSFILE
 5000 FORMAT(15X,A12)

       DO 101 I = 1, 12
          IF (INFIL (I:I) .EQ. '.') GO TO 102

             OUTFIL (I:I) = INFIL (I:I)
             HYDFIL (I:I) = INFIL (I:I)
             DMPFIL (I:I) = INFIL (I:I)
             ERRFIL (I:I) = INFIL (I:I)
          GO TO 101
 102      CONTINUE
             OUTFIL (I:I + 3) = '.OUT'
             HYDFIL (I:I + 3) = '.HYD'
             DMPFIL (I:I + 3) = '.HDF'
             ERRFIL (I:I + 3) = '.ERR'
          GO TO 103
 101   CONTINUE
 103   CONTINUE

C  ************************************************************************
C         Model results are written to File 6 and File 8.
C         Diagnostics are written to File 7.
C  ************************************************************************

       OPEN (UNIT=INPUT, FILE= INFIL  ,STATUS='OLD')
       OPEN (UNIT=OUT, FILE= OUTFIL ,STATUS='UNKNOWN')
       OPEN (UNIT=HYD, FILE= HYDFIL ,FORM='UNFORMATTED',STATUS='UNKNOWN')
       OPEN (UNIT=RIVDMP,FILE= DMPFIL ,STATUS='UNKNOWN')
       OPEN (UNIT=NERRFIL, FILE= ERRFIL ,STATUS='UNKNOWN')

C  ************************************************************************
C         2)  Read Input File
C  ************************************************************************
C     A)  Read All Global Information
C  ************************************************************************

C
       READ(INPUT,10)TITLE
  10 FORMAT(A80)
       WRITE(OUT,*)TITLE
       READ(INPUT,*)MNODE,SYEAR,SMONTH,SDAY,SHR,
      .              EYEAR,EMONTH,EDAY,EHOUR,NS
```

A3

```
      CALL JULIAN_DATE(STARTTIME,ENDTIME,SYEAR,SMONTH,SDAY,SHR,
     .                  EYEAR,EMONTH,EDAY,EHOUR)
      TNXTVD= STARTTIME

      WRITE(OUT,300)MNODE,NS
      WRITE(OUT,310) SYEAR,SMONTH,SDAY,SHR,STARTTIME
      WRITE(OUT,320) EYEAR,EMONTH,EDAY,EHOUR,ENDTIME

C
C     MNODE = IND1 = Number of nodes I the system
C     MTIME = Number of timesteps
C     NS = Number of river segments
C **********************************************************************

      READ(INPUT,*)BETA,GR,RMILE0,THETA,TOLER,IQL,IXS

      WRITE(OUT,330)BETA,GR,IPRINT,RMILE0,THETA,TOLER

C     Read Print Interval Information
C **********************************************************************

      READ(INPUT,*) NP
      WRITE(OUT,340) NP
      READ(INPUT,*) (TPRNT(I),TPR(I),I=1,NP)
      WRITE(OUT,350) (TPRNT(I),TPR(I),I=1,NP)
C
C     Initialize Counters for Printing
C **********************************************************************

      TDUM = 0.
      IPRINT = 1
      TPRINT = TPRNT(IPRINT)

C     Initialize Lateral Flows
C **********************************************************************

      DO I=1,MNODE
         QL(I)  = 0.
         QLC(I) = 0.
         QLT(I) = 0.
      END DO

C     Determine Name and Open Lateral Inflow File
C **********************************************************************

      IF(IQL.GE.1)OPEN (UNIT=LAT,FILE= LATFIL ,STATUS='OLD')


C     Determine Name and Open X-Section File
C **********************************************************************

      IF (IXS.GE.1)THEN
C
C     C)  If X-Section Data to be Read From a File (IXS >=1),
C         Determine Files and Open
C **********************************************************************
C
         OPEN (UNIT=FXSEC,FILE= XSFILE ,STATUS='OLD')
      ENDIF

      M2 = 0
```

```
C       D)  Within the "DO 1" loop below, the segment and initial condition
C           cards are read for each segment, boundary conditions are established
C           for each segment, and each segment is checked to see if it is a
C           tributary.
C  *************************************************************************
C

        DO 1 L=1,NS
          NODE1(L) = M2+1

C         1) Read the segment card
C  *************************************************************************
C

          READ(INPUT,2)ID(L),SNAME(L),NNODE(L),FEEDS(L),JNODE(L),
         .              BTU,BCU(L),BTD,BCD(L),COSP(L)
     2    FORMAT(I2,A40,3I3,2(A1,F8.0),F8.0)

          WRITE(OUT,360)ID(L),SNAME(L),NNODE(L),FEEDS(L),JNODE(L),
         .              BTU,BCU(L),BTD,BCD(L),COSP(L)

C         NNODE(L) = Number of Nodes in Segment Number 1
C  *************************************************************************

C         2) Parse the segment card. Establish type of boundary conditions.
C  *************************************************************************

          LIB(L)=1
          IF(BTU.EQ.'Q')LIB(L) = 4
          IF(BTD.EQ.'Q')LIB(L) = LIB(L)+1
          IF(BTD.EQ.'R')LIB(L) = LIB(L)+2

C         3)  Check to see if this segment is a tributary
C  *************************************************************************

          IF(JNODE(L).EQ.0)GO TO 4
          COSP(L) = COS(0.0174533*COSP(L))
          IF(LIB(L).LE.3)GO TO 5
          LIB(L) = 4
          GO TO 4
     5    LIB(L) = 1
     4    CONTINUE

          M1 = NODE1(L)
          M2 = M1+NNODE(L)-1

C         4)  Read initial conditions cards - ml and m2 correspond to the node
C             numbers of the first and last nodes in each segment.  Nodes are
C             read consecutively, the first node being 1 and incremented by 1
C             until the last node is read. Nodes are read one segment at a
C time.
C             Ml and M2 are updated each time a new segment is read by the
C             "do 1" loop above
C  *************************************************************************

          DO 7 I=M1,M2
            READ(INPUT,20)DX1(I),Q(I),H(I),QLC(I),EL(I),XSCODE(I),C2(I),
           .              C3(I),CN1(I),KE1(I),AX(I),DNDH(I)

            WRITE(OUT,370)DX1(I),Q(I),H(I),QLC(I),EL(I),XSCODE(I),C2(I),
           .              C3(I),CN1(I),KE1(I),AX(I),DNDH(I)

     7    CONTINUE
```

A5

```
      1 CONTINUE

C     E)  Read the boundary conditions id card and get initial time varying
data
C
*********************************************************************

      DO 26 L=1,NS+1
   26    IBC(L) = 0

      READ(INPUT,*)IBC

C     F) Unscramble references
C ****************************************************************

      DO 31 L=1,NS
         JBCU(L) = 0
         JBCD(L) = 0
         DO 31 M=1,NS
            IF(FEEDS(L).EQ.ID(M))FEEDS(L) = M
   31    CONTINUE

         NBC = 0
         DO 29 L=1,NS+1
            IF(IBC(L).EQ.0)GO TO 24
            NBC =NBC+1
            DO 25 M=1,NS
               IF(IBC(L).EQ.ID(M))IBC(L) = M
               IF(IBC(L).EQ.-ID(M))IBC(L) = -M
   25       CONTINUE
   29    CONTINUE

   24 CONTINUE


C     G) Construct cross-reference boundary conditions directory
C ****************************************************************

      IF(NBC.LE.0)GO TO 33

      DO 32 L=1,NBC
         IF(IBC(L))28,32,27

   28    JBCD(-IBC(L)) = L
         GO TO 32

   27    JBCU(IBC(L)) = L
   32 CONTINUE

   33 CONTINUE

C     H) Establish an "upstream" ordering of the segments
C ****************************************************************

      DO 42 I=1,NS
         ORDER(I) = 0
         IT1(I) = 0
         IT0(I) = 0

C ****************************************************************
         IT2(I) = 0
         IF(FEEDS(I).NE.0)GO TO 42
```

```
          ORDER(1) = I
    42 CONTINUE

       L = 1
       IT = 0

       DO 44 I=1,NS
          M = ORDER(I)
          DO 44 J=1,NS
             IF(FEEDS(J).NE.M)GO TO 44
             L = L+1
             ORDER(L) = J
             IF(JNODE(J).GT.0)GO TO 39
             JBCU(M) = -(NODE1(J)+NNODE(J)-1)
             IF(LIB(M).LE.3)LIB(M) = LIB(M)+3
             GO TO 38

    39       IT = IT+1
             IF(IT1(M).EQ.0)IT1(M) = IT
             JT(IT) = JNODE(J)
             IT2(M) = IT
             IT0(J) = IT
             JBCD(J) = -(NODE1(M)+JNODE(J)-1)

    38       IF(L.EQ.NS)GO TO 43
    44    CONTINUE

    43 CONTINUE

       IF(NS.GT.2)CALL BUBBLE(IT0,IT1,IT2,JT,NS)

C     I) Read in the surveyed cross-section data, if present
C     ****************************************************************

       CALL READXS(XSCODE, IXSA, IXSB, MNODE)
C     ****************************************************************

C     IXSA and IXSB are pointers into a cross-section table.
C     If they are both positive, then C1 serves as the linear
C     interpolation factor between the two values pointed to.
C     (remember, for this scheme to work, every section of the
C     cross-section table must be bounded above and below by
C     hxs = 0, i.e., the first x-y pair for a x-section must be
C     0,0). If only IXSA is positive, then that node is associated
C     with a single surveyed cross-section and no interpolation is
C     required.
C     ****************************************************************

C     The "DO 67" loop carried out for each segment independently
C     first sweep downstream
C     ****************************************************************

       DO 67 L = 1, NS
          M1 = NODE1(L)
          M2 = M1+NNODE(L)-1
          IXS = 0
          DO 68 I = M1, M2
             IF (IXSA(I) .GT. 0) THEN
                 D = DX1(I)
                 IXS = IXSA(I)
             ELSE
                 C1(I) = D
```

```
                  D = D + DX1(I)
               ENDIF

               IXSA(I) = IXS
68          CONTINUE

C ***** Then sweep upstream
C *********************************************************************

            IXS = 0
            DO 69 I = M2, M1, -1
               IF (IXSB(I) .GT. 0) THEN
                  D = 0.
                  IXS = IXSB(I)
               ELSE IF (IXS .GT. 0) THEN
                  D = D + DX1(I)
                  C1(I) = C1(I)/(C1(I) + D)
               ENDIF
               IXSB(I) = IXS
69          CONTINUE
67       CONTINUE

***** ............... END "DO 67" LOOP ...............

***** Transform initial stages into initial cross-sectional areas
***** And channel top-widths.  C3<0 is the signal that an ellipsoid
***** Cross-section is indicated.
C *********************************************************************
C *********************************************************************

C          Manning's n, originally input as CN1(I) is redefined as
C          XMAN(I).  CN1(I) is then redefined to be "CN1(I)**2/2.2*gr" for
C          later use.  Z(I) is set equal to the bottom elevation,
C          originally input as EL(I), EL(I) is then redefined as the
C          W.S.E. = H(I) + EL(I), where H(I) is the computed water depth.
C *********************************************************************
         DO 70 I=1,MNODE
            XMAN(I)=CN1(I)                          .
            CN1(I) = CN1(I)**2/2.2*GR
            Z(I) = EL(I)
            EL(I) = H(I)+EL(I)

            IF ((IXSA(I) .EQ. 0 .AND. IXSB(I) .EQ. 0) .OR.
     .          (IXSA(I) .NE. IXSB(I) .AND.
     .              XSCODE(I) .NE. '      ')) THEN
               READ(XSCODE(I),'(F6.0)') C1(I)
               IXSA(I) = 0
               IXSB(I) = 0
            ELSE IF (IXSA(I) .EQ. 0) THEN
               IXSA(I) = IXSB(I)
               IXSB(I) = 0
            ELSEIF (IXSA(I) .EQ. IXSB(I)) THEN
               IXSB(I) = 0
            ENDIF

            IF (IXSA(I) .GT. 0) THEN
               IF (IXSB(I) .GT. 0) THEN
                  CALL AFROMH2(H(I),A(I),B(I),DBDH(I),IXSA(I),
     .                         IXSB(I),C1(I))
               ELSE
                  CALL AFROMH(H(I), A(I), B(I), DBDH(I), IXSA(I))
```

A8

```fortran
              ENDIF
           ELSE

C       **************************** N O T E **********************************
C       * Transform initial stages into initial cross-sectional areas and
C       * channel top-widths.  C3<0 is the signal that an ellipsoid cross-
C       * section is indicated.
C       *********************************************************************

              IF(C3(I).LT.0.0) GO TO 60
              IF(AINT(C3(I)).EQ.C3(I))GO TO 65
              I3(I) = 0
              A(I)=C1(I)*H(I)+C2(I)*H(I)**C3(I)
              B(I)=C1(I)+C2(I)*C3(I)*H(I)**(C3(I)-1.)
              DBDH(I)=C2(I)*C3(I)*(C3(I)-1.)*H(I)**(C3(I)-2.)
              GO TO 70

65            I3(I) = C3(I)
              A(I)=C1(I)*H(I)+C2(I)*H(I)**I3(I)
              B(I)=C1(I)+C2(I)*C3(I)*H(I)**(I3(I)-1)
              DBDH(I)=C2(I)*C3(I)*(C3(I)-1.)*H(I)**(I3(I)-2.)
              GO TO 70

60            I3(I) = -1
              B(I)=2.*C2(I)/C1(I)*SQRT((2.*C1(I)-H(I))*H(I))
              A(I)=C1(I)*C2(I)*ACOS(1.-H(I)/C1(I))-B(I)*(C1(I)-H(I))/2.
              DBDH(I)=4.*(C2(I)/C1(I))**2*(C1(I)-H(I))/B(I)
           ENDIF

     70 CONTINUE

C      Calculate river mile for each point in the network
C ******************************************************************************

       DO 49 LL=1,NS
          L = ORDER(LL)
          M1 = NODE1(L)+1
          M2 = NNODE(L)+M1-2
          IF(LL.GT.1)GO TO 51
          RMILE(M2) = RMILE0
          GO TO 52

     51   IF(JNODE(L).GT.0)GO TO 53
          LJ = FEEDS(L)
          MJ = NODE1(LJ)
          RMILE(M2) = RMILE(MJ)
          GO TO 52

     53   RMILE(M2) = 0.

     52   CONTINUE
          DO 54 II=M1,M2
             I = M1+M2-II
             RMILE(I-1) = RMILE(I)+DX1(I-1)/5280.
     54   CONTINUE
     49 CONTINUE

C      Write out initial data for use in RIV1Q Program
C ******************************************************************************

       WRITE(HYD)MNODE,STARTTIME,ENDTIME,SYEAR,NS
       DO 1000 I =1,NS
```

A9

```fortran
            WRITE(HYD)JT(I),IT0(I),IT1(I),IT2(I),ORDER(I),NNODE(I),
     .            NODE1(I),ID(I),JBCU(I)
 1000 CONTINUE

      DO 1010 I =1,MNODE
          WRITE(HYD)DX1(I),QLC(I),RMILE(I)
 1010 CONTINUE

      G2 = GR/2.

      ITIME = 0
      ELAPSE = STARTTIME

      CALL GREGORIAN_DATE(ELAPSE,SYEAR,MONTH,IDAY,HOUR)

      CALL TIME_VARYING_DATA (ELAPSE,SYEAR,TNXTVD,NBC,DT,BC,IQL,QLT)

      DO 1031 I=1,MNODE
          IF(IQL.GE.1)THEN
              QL(I) = QLC(I)+QLT(I)
          ELSE
              QL(I) = QLC(I)
          END IF
 1031 CONTINUE


C    3)  BEGIN TIME MARCH LOOP
C        IT ENDS WITH GO TO 80 STATEMENT AT END OF LOOP
C *************************************************************************

   80 CONTINUE

C *************************************************************************
C        The tolerance test within a timestep is carried out by
C        first computing the "Euclidian Norm" for the flow and
C        area vectors for the entire system. This value is multiplied
C        by the "tol" input parameter and divided by the number of
C        nodes in the system.  The resulting parameters are called
C        rmsq and rmsa.  The largest departure for every q and a in
C        the system, within a timestep, must be less than the value
C        of the corresponding tolerance parameter.  If not, the code
C        cycles thru another iteration.
C *************************************************************************

C    A)   Calculate root mean square Q & H for tolerance test
C *************************************************************************
C
      RMSQ = 0.
      RMSA = 0.

      DO 66 I=1,MNODE
          RMSQ = RMSQ+Q(I)*Q(I)
          RMSA = RMSA+A(I)*A(I)
   66 CONTINUE

      RMSQ = TOLER*SQRT(RMSQ)/FLOAT(mnode)
      RMSA = TOLER*SQRT(RMSA)/FLOAT(mnode)


C    B)  Write out the previous timestep's data
C *************************************************************************
```

```
        IF(TDUM.GE.TPRINT.OR.TDUM.EQ.0.) THEN
            WRITE(OUT,110)TITLE
            WRITE(OUT,125)DT
            DO 81 L=1,NS
                WRITE(OUT,115)L
                WRITE(OUT,120)SYEAR,MONTH,IDAY,HOUR,ITIME,ID(L),SNAME(L)
                WRITE(OUT,126)BCU(L), BCD(L)
                WRITE(OUT,130)
                M = NODE1(L)-1
                M2 = NNODE(L)
                 WRITE(OUT,140)(I,RMILE(I+M),Q(I+M),A(I+M),B(I+M),H(I+M),
                                EL(I+M),XMAN(I+M),I=1,M2)
   81       CONTINUE

            CALL RIVDU(MNODE,DX1(1),RMILE(1),Q(1),QLC(1),QLT(1)
                     ,A(1),B(1),EL(1),XMAN(1))

            TDUM = 0.
        END IF

        TDUM = TDUM + DT/3600. +0.00001

C       C) Write to output file for RIV1Q Program
C ******************************************************************

        WRITE(HYD)DT

        DO 1020 I = 1, MNODE
            WRITE(HYD)Q(I),QLT(I),A(I),B(I),EL(I)
 1020   CONTINUE

C       D)  Check Courant No.  If greater than 1.0, print to error file.
C           Courant No. Greater than 1.0 will cause the RIV1Q code to
C           become unstable
C ******************************************************************

        MNM1 = MNODE - 1

        DO 683 IK = 2, MNM1
            COURANT = ( Q(IK) * DT ) / ( DX1(IK) * A(IK) )
            IF ( COURANT .GT. .99 ) WRITE(NERRFIL,684) IK, ELAPSE, COURANT
  683   CONTINUE

  684   FORMAT (10X, 'Node    ', I3, 10X, 'Elapse = ', F10.6,
       . 5X, 'Courant No = ', F6.2 )

        ITIME=ITIME+1

C       E)  Update times and stop if time is greater than the endtime
C ******************************************************************

        ELAPSE=ELAPSE+DT/86400.
        IF(ELAPSE.GT.TPR(IPRINT).AND.NP.GT.1)THEN
            IPRINT=IPRINT+1
            TPRINT=TPRNT(IPRINT)
        END IF

        IF(ELAPSE.GE.ENDTIME) STOP

        CALL GREGORIAN_DATE(ELAPSE,SYEAR,MONTH,IDAY,HOUR)

C        F)  Update time varying data
```

All

```fortran
C     ********************************************************************
      IF (ELAPSE.GE.TNXTVD)
     .     CALL TIME_VARYING_DATA (ELAPSE,SYEAR,TNXTVD,NBC,DT,BC,IQL,QLT)

      DO 1030 I=1,MNODE
         IF(IQL.GE.1)THEN
            QL(I) = QLC(I)+QLT(I)
         ELSE
            QL(I) = QLC(I)
         END IF
1030  CONTINUE

C     ********************************************************************
C     Variable Manning's n Values:
C     Manning's n is allowed to vary in accordance with a linear
C     relationship between n and the DEPTH H in the X-section.
C     AX(I) is the value of n when the DEPTH is 0.0.  The rate of
C     change of n with DEPTH, DNDH, is the coefficient of H.  The
C     user is alerted if n drops below zero and n is reset to 0.01.
C     WARNING! The basic equations have been modified to allow only for
C     linear variation of n with depth.  If the user contemplates other
C     relationships, the equations must be modified accordingly.
C     ********************************************************************

      DO 165 I=1,MNODE
         IF(AX(I).NE.0.0.AND.DNDH(I).NE.0.0)THEN
            XMAN(I)=AX(I) - DNDH(I)*H(I)
            IF(XMAN(I).LE.0.01)THEN
               WRITE(NERRFIL,177)I
177            FORMAT('Value for XMAN at Node I =',I9,
     .                ' is < 0.01. it has been RESET to 0.01')
               XMAN(I)=0.01
            END IF
            CN1(I)=XMAN(I)**2*GR/2.2
         END IF
165   CONTINUE

C     G)  CALLING SUBROUTINES
C     ********************************************************************

      DO 400 LL=1,NS
         L = ORDER(NS-LL+1)
         M = NODE1(L)
         M1 = NNODE(L)
         M2 = 2*M-1
         IF(JBCU(L))401,402,403

401      JJ = -JBCU(L)
         BCU(L) = Q(JJ)
         GO TO 402

403      JJ = JBCU(L)
         BCU(L) = BC(JJ)

402      IF(JBCD(L))404,405,406

404      JJ = -JBCD(L)
         BCD(L) = EL(JJ)-EL(M+M1-1)+H(M+M1-1)
         GO TO 405

406      JJ = JBCD(L)
```

A12

```fortran
            BCD(L) = BC(JJ)
  405       CALL CALC(THETA,DT,DX1(M),Q(M),A(M),B(M),I3(M),C1(M),C2(M),
     .            C3(M),H(M),EL(M),QL(M),G2,CN1(M),KE1(M),XC(M),XM(M),
     .            C(M2),R(M2),AA,LIB(L),BCU(L),BCD(L),COSP(L),BETA,
     .            IT0(L),IT1(L),IT2(L),JT,T,M1,10*(M1-1),2*M1,NS,
     .            ID(L),XMAN(M),DNDH(M),IXSA(M),IXSB(M),DBDH(M))

  400 CONTINUE

C
C     The "DO 180" loop iterates to find a set of qs and as which will
C     satisfy the basic equations within the tolerance specified by the
C     tolerance parameter
C **********************************************************************
C

      DO 180 K=1, 50

C     Sweep upstream
C **********************************************************************

          DO 407 LL=1,NS
            L = ORDER(LL)
            M = NODE1(L)
            M1 = NNODE(L)
            M2 = 2*M-1
            CALL NEW(R(M2),Q(M),A(M),B(M),H(M),EL(M),I3(M),C1(M),C2(M),
     .          C3(M),C(M2),IT0(L),IT1(L),IT2(L),JT,T,M1,2*M1,NS,
     .          Z(M),IXSA(M),IXSB(M),DBDH(M))
  407     CONTINUE

C     H)  Check each of the computed departures for Q and A. If any is large  r
C         than the computed tolerance for each, go for another iteration,
C         a Maximum of 50.
C **********************************************************************

          DO 160 I=1,MNODE
            IF(ABS(R(2*I-1)).GT.RMSQ.OR.ABS(R(2*I)).GT.RMSA)GO TO 170
  160     CONTINUE

C
C     At this point, flows and water depths for a single timestep have
C     been computed.  Computations for the next timestep are begun at
C     statement 80 above, where the time march begins.
C **********************************************************************
C

          GO TO 80

  170     CONTINUE

C     To examine node at which non-convergence is occurring, the
C     following line has been added to the code as a diagnostic.
C **********************************************************************

          IF(K.GE.25)THEN
            WRITE(NERRFIL,*) 'Z(I)= ',Z(I),' ELAPSE= ',ELAPSE,
     .                  ' TOLER TST'
            WRITE(NERRFIL,*) 'ABS(R(2*I-1))= ',ABS(R(2*I-1)),
     .                  ' RMSQ= ',RMSQ
            WRITE(NERRFIL,*) 'ABS(R(2*I)= ',ABS(R(2*I)),' RMSA= ',RMSA
```

A13

```
      END IF
C
C       This loop is identical to the "DO 165" loop above.  It recomputes
C       Manning's n and CN1(I) using new H(I) values, within a timestep.
C  *****************************************************************
C
        DO 166 I=1,MNODE
          IF(AX(I).NE.0.0.AND.DNDH(I).NE.0.0)THEN
            XMAN(I)=AX(I) - DNDH(I)*H(I)
            IF(XMAN(I).LE.0.01)THEN
              WRITE(NERRFIL,177)I
              XMAN(I)=0.01
            END IF
            CN1(I)=XMAN(I)**2*GR/2.2
          END IF
 166      CONTINUE
C
C       The "DO 408" loop resets boundary conditions, recomputes Manning's n
C       for corrected flows and solves the system of equations for
C       new departures.
C  *****************************************************************
C
        DO 408 LL=1,NS
          L = ORDER(NS-LL+1)
          M = NODE1(L)
          M1 = NNODE(L)
          M2 = 2*M-1
          IF(JBCU(L).GE.0)GO TO 409
          JJ = -JBCU(L)
          BCU(L) = Q(JJ)

 409      IF(JBCD(L).GE.0)GO TO 410
          JJ = -JBCD(L)
          BCD(L) = EL(JJ)-EL(M+M1-1)+H(M+M1-1)
 410      CALL ITER(THETA,DT,DX1(M),Q(M),A(M),B(M),I3(M),C1(M),C2(M),
     .             C3(M),H(M),EL(M),QL(M),G2,CN1(M),KE1(M),XC(M),XM(M),
     .             C(M2),R(M2),AA,LIB(L),BCU(L),BCD(L),COSP(L),BETA,
     .             ITO(L),IT1(L),IT2(L),JT,T,M1,10*(M1-1),2*M1,NS,ID(L),
     .             XMAN(M),DNDH(M),IXSA(M),IXSB(M),DBDH(M))
 408      CONTINUE

 180 CONTINUE

     WRITE(OUT,190)
     STOP 'Failed to Converge'
 200 CONTINUE
     STOP
  20 FORMAT(5F6.0,A6,6F6.0)
 110 FORMAT(1H1,//,1X,A80)
 115 FORMAT(/,'  OUTPUT FOR SEGMENT NO.   ',I5)
 120 FORMAT(1X,'Time: Year =',I4,',Month= ',A9,',Day',I4,',Hour =',
     .      F8.4,'  Step= ',I5,'  Segment=',I3,'  ',A40)
 125 FORMAT(1X, 'TIME STEP (sec) =  ',F8.2)
 126 FORMAT(  ' Upstream Boundary ',F8.3,
     .        '  Downstream Boundary =',F8.3)
 130 FORMAT(1x,T7,'River',7X,'  Flow ',7X,'  Area  ',7X,'  Width ',
     .      7X,'  Depth ',7X,'   WSEL',10X,'Mannings n',/,
     .      '  ',T8,'Mile',7X,'  (cfs) ',7X,' (sq ft)',7X,' (Feet) ',
     .      7X,' (Feet) ',7X,' (Feet) ')
```

```fortran
  140 FORMAT(1x,I2,F9.2,6F15.3)
  190 FORMAT('The Iteration has FAILED to CONVERGE in 50 steps. Run',
     .     ' Aborted')
  300 FORMAT(1X,'NUMBER OF NODES = ',I5,' NUMBER OF SEGMENTS = ',I5)
  310 FORMAT(/,' STARTIME: YEAR =',I4,',MONTH= ',I4,',DAY',I4,',HOUR =',
     .     F8.4, ' JULIAN DAY =   ',F8.3)
  320 FORMAT(' END TIME: YEAR =',I4,',MONTH= ',I4,',DAY',I4,',HOUR =',
     .     F8.4, ' JULIAN DAY =   ',F8.3)
  330 FORMAT(/,
     .' BETA   = ',F8.2,/,
     .' GR     = ',F8.2,/,
     .' IPRINT = ',I8,/,
     .' RMILE0 = ',F8.2,/,
     .' THETA  = ',F8.2,/,
     .' TOLER  = ',F8.4)
  340 FORMAT(/,' NUMBER OF PRINT UPDATES',I5)
  350 FORMAT(' PRINT INTERVAL = ',F8.3,'  PRINT TIME ',F8.3)
  360 FORMAT(/,I2,A40,3I3,2(A1,F8.0),F8.0)
  370 FORMAT(1X,5F10.3,A6,6F10.3)
C
      END
C ********************************************************************
C                  S U B R O U T I N E   A F R O M H
C ********************************************************************

      SUBROUTINE AFROMH(H, A, B, DBDH, IPC)

C
C     Given the area and width vs. depth table, and the current depth,
C     Calculate the area, channel top width, and derivative of the top width
C     Using linear table search (linear is not inefficient because you keep
C     An updatable pointer into the table) and linear interpolation.
C     The limit switches on the table search (top and bottom) are hxs = 0
C ********************************************************************
C
      PARAMETER(MAXXS=1000)

      COMMON/XSDATA/IXS,HXS(MAXXS),AXS(MAXXS),BXS(MAXXS)

      IF (H .LE. 0.) STOP 'AFROMH: Zero or Negative Depth of Flow'

      DO 10 I = IPC, MAXXS
         IF (HXS(I+1) .LE. 0)
     .      STOP 'AFROMH: River has Overflowed its Banks!'
   10    IF (H .LE. HXS(I+1)) GOTO 11

   11    IPC = I

      DO 20 I = IPC, 1, -1
   20    IF (H .GT. HXS(I)) GOTO 21

   21 IPC = I
      DBDH = (BXS(I+1) - BXS(I))/(HXS(I+1) - HXS(I))
      DH = H - HXS(I)
      B = BXS(I) + DBDH*DH
      A = AXS(I) + (BXS(I) + DBDH*DH/2.)*DH
      RETURN
      END
C ********************************************************************
C                  S U B R O U T I N E   A F R O M H 2
C ********************************************************************
      SUBROUTINE AFROMH2(H, A, B, DBDH, IA, IB, E)
```

A15

```
C
C       Given the area and width vs. depth table, and the current depth,
C       calculate the area, channel top width, and derivative of the top width
C       for "blended" cross-sections using linear table search
C       (linear is not inefficient because you keep an updatable pointer
C       into the table) and inverse linear interpolation. the limit switches
C       on the table search (top and bottom) are hxs = 0
C  ****************************************************************************

        PARAMETER(MAXXS=1000)


        COMMON/XSDATA/IXS,HXS(MAXXS),AXS(MAXXS),BXS(MAXXS)

        LOGICAL RECALC

        IF (H .LE. 0) STOP 'River has DRIED up.'

        EC = 1. - E

10      IF (HXS(IA+1) .LE. 0)STOP 'River has OVERFLOWED its banks'
        IF (HXS(IA+1) .GT. H) GOTO 11
        IA = IA + 1
        GOTO 10

11      IF (HXS(IB+1) .LE. 0)STOP 'River has OVERFLOWED its banks'
        IF (HXS(IB+1) .GT. H) GOTO 12
        IB = IB + 1
        GOTO 11

C
C       At this point, both heights are below the target.  Now use the formula
C       to get the exact height and area.  If either of the heights puts you
C       outside of the interval, then adjust the interval.
C  ****************************************************************************

12      CONTINUE
        BOA = BXS(IA)
        BOB = BXS(IB)
        HOA = HXS(IA)
        HOB = HXS(IB)
        DBDHA = (BXS(IA+1) - BOA)/(HXS(IA+1) - HOA)
        DBDHB = (BXS(IB+1) - BOB)/(HXS(IB+1) - HOB)
        HR   = HOB - H/E
        ER   = EC/E
        AQ2 = DBDHA - ER*ER*DBDHB
        BQ = BOA - DBDHA*HOA + ER*(BOB - DBDHB*HR)
        CQM2 = -2.*(AXS(IA) - AXS(IB) + HOA*(-BOA +DBDHA*HOA/2)
     .         -HR*(-BOB + DBDHB*HR/2))
        Q = SQRT(BQ*BQ + CQM2*AQ2)

        IF (SIGN(BQ, Q) .EQ. BQ) THEN
            HA = CQM2/(BQ + Q)
            HB = (H - EC*HA)/E
            IF (HA .GT. 0 .AND. HB .GT. 0) GOTO 13
            HA = (-BQ - Q)/AQ2
            HB = (H - EC*HA)/E
        ELSE
            HA = CQM2/(BQ - Q)
            HB = (H - EC*HA)/E
            IF (HA .GT. 0 .AND. HB .GT. 0) GOTO 13
            HA = (-BQ + Q)/AQ2
```

A16

```
          HB = (H - EC*HA)/E
      ENDIF

13    RECALC = .FALSE.

      IF (HA .GT. HXS(IA+1)) THEN
          RECALC = .TRUE.
14        IA = IA + 1
          IF (HXS(IA+1) .LE. 0)STOP 'River has OVERFLOWED its banks'
          IF (HA .GT. HXS(IA+1)) GOTO 14
      ELSEIF (HA .LT. HXS(IA)) THEN
          RECALC = .TRUE.
15        IA = IA - 1
          IF (HA .LT. HXS(IA)) GOTO 15
      ENDIF

      IF (HB .GT. HXS(IB+1)) THEN
          RECALC = .TRUE.
16        IB = IB + 1
          IF (HXS(IB+1) .LE. 0)STOP 'River has OVERFLOWED its banks'
          IF (HB .GT. HXS(IB+1)) GOTO 16
      ELSEIF (HB .LT. HXS(IB)) THEN
          RECALC = .TRUE.
17        IB = IB - 1
          IF (HB .LT. HXS(IB)) GO TO 17
      ENDIF

      IF (RECALC) GOTO 12
      DBDH = EC*DBDHA + E*DBDHB
      B = EC*(BOA + DBDHA*(HA - HOA)) + E*(BOB + DBDHB*(-ER*HA - HR))
      A = AXS(IA) + (BOA + DBDHA*(HA - HOA)/2)*(HA - HOA)

      RETURN
      END
C *********************************************************************
c                  S U B R O U T I N E    B U B B L E
C *********************************************************************
      SUBROUTINE BUBBLE(IT0,IT1,IT2,JT,NS)

      DIMENSION IT0(NS),IT1(NS),IT2(NS),JT(NS)

      DO 1 L=1,NS
          IF(IT2(L).LE.IT1(L))GO TO 1
          LAST = IT2(L)
      2   LIMIT = LAST-1
          LAST = IT1(L)
          M1 = LAST
          DO 3 I=M1,LIMIT
              IF(JT(I).LE.JT(I+1))GO TO 3
              ITEMP = JT(I)
              JT(I) = JT(I+1)
              JT(I+1) = ITEMP
              LAST = I
              DO 5 M=1,NS
                  IF(IT0(M).NE.I)GO TO 6
                  IT0(M) = I+1
                  GO TO 5
      6           IF(IT0(M).EQ.I+1)IT0(M) = I
      5       CONTINUE
      3   CONTINUE
          IF(LAST.GT.M1)GO TO 2
      1 CONTINUE
```

```
      RETURN
      END
C ***********************************************************************
c                    S U B R O U T I N E   C A L C
C ***********************************************************************
C
C
C        PURPOSE
C             to calculate the residuals from the governing equations
C             to calculate the partial derivatives of the governing equation
C             to set up a vector of the residuals
C             to set up a matrix of the partial derivatives
C             to solve this system of equations for the differences between
C              the values of q and a in two consecutive iterations
C        DESCRIPTION OF PARAMETERS
C             XC      - the j terms from the continuity equation
C             XM      - the j terms from the momentum equation
C             R1      - the residuals from the continuity equation
C             R2      - the residuals from the momentum equation
C             R       - a collection of all of the residuals in the proper
C                       order for the vector
C             F       - the continuity equation
C             G       - the momentum equation
C             DFQ     - partial differential of f with respect to q(i)
C             DFA     - partial differential of f with respect to a(i)
C             DFQ1    - partial differential of f with respect to q(i+1)
C             DFA1    - partial differential of f with respect to a(i+1)
C             DGQ     - partial differential of g with respect to q(i)
C             DGA     - partial differential of g with respect to a(i)
C             DGQ1    - partial differential of g with respect to q(i+1)
C             DGA1    - partial differential of g with respect to a(i+1)
C             AA      - a collection of all of the partial derivatives in
C                       the proper order for the banded matrix
C             DNDH    - partial of n with respect to h
C ***********************************************************************
      SUBROUTINE CALC(THETA,DT,DX1,Q,A,B,I3,C1,C2,C3,H,EL,QL,G2,CN1,KE1,
     .XC,XM,C,R,AA,LIB,BCU,BCD,COSP,BETA,IT0,IT1,IT2,JT,T,IND1,IND2,
     .IND3,NS,ID,XMAN,DNDH,IXSA,IXSB,DBDH)

      REAL KE1
      LOGICAL FLIP1,FLIP2

      DIMENSION Q(IND1),A(IND1),B(IND1),H(IND1),C1(IND1),C2(IND1),
     .C3(IND1),EL(IND1),QL(IND1),CN1(IND1),KE1(IND1),R(IND3),AA(IND2),
     .XC(IND1),XM(IND1),DX1(IND1),C(IND3),I3(IND1),JT(NS),T(4,NS),
     .ID(NS),XMAN(IND1),DNDH(IND1),IXSA(IND1),IXSB(IND1),DBDH(IND1)

C
C        The previous timestep components of the governing equations
C ***********************************************************************
      MP=IND1-1
      Q0 = Q(1)
      A0 = A(1)
      P0 = (B(1)/A0)**(4./3.)*ABS(Q0)/A0
      D1 = (1.-THETA)*DT

      DO 10 I=1,MP
         D2 = D1/DX1(I)*2.
         Q1 = Q(I+1)
         A1 = A(I+1)
```

```
            P1 = (B(I+1)/A1)**(4./3.)*ABS(Q1)/A1
            DE = EL(I+1)-EL(I)
            XC(I)= -A0-A1+D2*(Q1-Q0)-D1*(QL(I)+QL(I+1))
            XM(I)= -Q0-Q1+D2*(BETA*(Q1*Q1/A1-Q0*Q0/A0)+G2*(A1+A0)*DE)
     .              +D1*(CN1(I)*P0*Q0 + (CN1(I+1)*P1*Q1))
            IF(KE1(I).GT.0)XM(I)=XM(I)+D2/8.*(A0+A1)*KE1(I)*(Q1*Q1/(A1*A1)
     .                                +Q0*Q0/(A0*A0))
            P0 = P1
            Q0 = Q1
            A0 = A1
   10 CONTINUE

      XC(IND1) = D1*Q1
      XM(IND1) = D1*Q1*Q1/A1*COSP

C
C      This is the entry point for all succeeding iterations of the
C      Newton-Raphson procedure
C ********************************************************************

      ENTRY ITER(THETA,DT,DX1,Q,A,B,I3,C1,C2,C3,H,EL,QL,G2,CN1,KE1,XC,
     .XM,C,R,AA,LIB,BCU,BCD,COSP,BETA,IT0,IT1,IT2,JT,T,IND1,IND2,IND3,
     .NS,ID,XMAN,DNDH,IXSA,IXSB,DBDH)

C      Upstream boundary conditions
C ********************************************************************

      R(1) = 0.
      IF(LIB.GT.3)GO TO 30
      EL(1) = EL(1)-H(1)+BCU
      H(1) = BCU

      IF (IXSA(1) .GT. 0) THEN
         CALL AFROMH(H(1), A(1), B(1), DBDH(1), IXSA(1))
      ELSE
         IF(I3(1))20,14,15

   14    A(1)=C1(1)*H(1)+C2(1)*H(1)**C3(1)
         B(1)=C1(1)+C2(1)*C3(1)*H(1)**(C3(1)-1.)
         DBDH(1)=C2(1)*C3(1)*(C3(1)-1.)*H(1)**(C3(1)-2.)
         GO TO  40

   15    A(1)=C1(1)*H(1)+C2(1)*H(1)**I3(1)
         B(1)=C1(1)+C2(1)*C3(1)*H(1)**(I3(1)-1)
         DBDH(1)=C2(1)*C3(1)*(C3(1)-1.)*H(1)**(I3(1)-2)
         GO TO  40

   20    B(1)=2.*C2(1)/C1(1)*SQRT((2.*C1(1)-H(1))*H(1))
         A(1)=C1(1)*C2(1)*ACOS(1.-H(1)/C1(1))-B(1)*(C1(1)-H(1))/2.
         DBDH(1)=4.*(C2(1)/C1(1))**2*(C1(1)-H(1))/B(1)
         GO TO  40

      ENDIF

      GO TO  40

   30 Q(1)=BCU
   40 CONTINUE

C
C      Downstream boundary conditions
C ********************************************************************
```

A19

```fortran
      GO TO(43,60,70,43,60,70),LIB

   43 EL(IND1) = EL(IND1)-H(IND1)+BCD
      H(IND1)=BCD
      AA(IND2) = 0.
      R(IND3) = 0.

      IF (IXSA(IND1) .GT. 0) THEN
         CALL AFROMH(H(IND1), A(IND1), B(IND1), DBDH(IND1), IXSA(IND1))
      ELSE
         IF(I3(IND1))50,44,45

   44    A(IND1)=C1(IND1)*H(IND1)+C2(IND1)*H(IND1)**C3(IND1)
         B(IND1)=C1(IND1)+C2(IND1)*C3(IND1)*H(IND1)**(C3(IND1)-1.)
         DBDH(IND1)=C2(IND1)*C3(IND1)*(C3(IND1)-1.)*H(IND1)**
      .              (C3(IND1)-2.)
         GO TO  51

   45    A(IND1)=C1(IND1)*H(IND1)+C2(IND1)*H(IND1)**I3(IND1)
         B(IND1)=C1(IND1)+C2(IND1)*C3(IND1)*H(IND1)**(I3(IND1)-1)
         DBDH(IND1)=C2(IND1)*C3(IND1)*(C3(IND1)-1.)*H(IND1)**(I3(IND1)
      .              -2)
         GO TO  51

   50    B(IND1)=2.*C2(IND1)/C1(IND1)*H(IND1)*SQRT(2.*C1(IND1)/H(IND1)
      .           -1.)
         A(IND1)=C1(IND1)*C2(IND1)*ACOS(1.-H(IND1)/C1(IND1))
      .           -B(IND1)*(C1(IND1)-H(IND1))/2.
         DBDH(IND1)=4.*(C2(IND1)/C1(IND1))**2*(C1(IND1)-H(IND1))/B(IND1)
      ENDIF

   51 C(IND3) = -B(IND1)

      GO TO  80

   60 Q(IND1)=BCD
      R(IND3)=0.
      AA(IND2) = 0.
      GO TO  80

   70 EXPO = BCD
      COEF = COSP
      R(IND3)=-(COEF*Q(IND1)**EXPO-H(IND1))
      DBCQ=COEF*EXPO*Q(IND1)**(EXPO-1.)
      DBCA=-1./B(IND1)
      AA(IND2-1)=DBCQ
      AA(IND2)=DBCA

   80 CONTINUE

  100    DNDH2 = DNDH(1)

C
C     The governing equations
C ****************************************************************************

      MP = IND1-1
      Q0 = Q(1)
      A0 = A(1)
      P0 = (B(1)/A0)**(4./3.)*ABS(Q0)/A0
      IR = 2
      IA = 1
```

```
        D1 = THETA*DT
        DO 130 I=1,MP
          D2 = D1/DX1(I)*2.
          Q1 = Q(I+1)
          A1 = A(I+1)
          P1 = (B(I+1)/A1)**(4./3.)*ABS(Q1)/A1
          CN = CN1(I)*D1
          CN2 = CN1(I+1)*D1
          DE = EL(I+1)-EL(I)
          R(IR)=-(A0+A1+D2*(Q1-Q0)-D1*(QL(I)+QL(I+1))+XC(I))
          R(IR+1)=-(Q0+Q1+D2*(BETA*(Q1*Q1/A1-Q0*Q0/A0)+G2*(A0+A1)*DE)
     .          +(CN*Q0*P0+CN2*Q1*P1)+XM(I))
          DFQ = -D2
          DFA = 1.
          DFQ1 = D2
          DFA1 = 1.
          DGQ = 1.-2.*D2*BETA*Q0/A0 + CN*P0
          DGA = D2*(BETA*Q0*Q0/(A0*A0)+G2*(DE-(A1+A0)/B(I)))
     .          +CN*Q0*P0/3.*(-7./A0+4.*DBDH(I)/(B(I)*B(I)+6./
     .          (XMAN(I)*B(I))*DNDH2))
          DGQ1 = 1.+2.*D2*BETA*Q1/A1+ CN2*P1
          DNDH2=DNDH(I+1)
          DGA1 = D2*(-BETA*Q1*Q1/(A1*A1)+G2*(DE+(A0+A1)/B(I+1)))
     .          +CN2*Q1*P1/3.*(-7./A1+4.*DBDH(I+1)/(B(I+1)*B(I+1)+
     .          6./(XMAN(I+1)*B(I+1))*DNDH2))
          IF(DGA1.EQ.0.0) DGA1 = 1.0E-5
          IF(KE1(I).LE.0.)GO TO 125
          R(IR+1)=R(IR+1)-D2/8.*(A0+A1)*(Q1*Q1/(A1*A1)+Q0*Q0/(A0*A0))
     .          *KE1(I)
          DGQ=DGQ+KE1(I)*D2/4.*(A0+A1)*Q0/(A0*A0)
          DGA=DGA+D2/8.*KE1(I)*(Q1*Q1/(A1*A1)-Q0*Q0/(A0*A0)
     .          *(1.+2.*A1/A0))
          DGQ1=DGQ1+KE1(I)*D2/4.*(A0+A1)*Q1/(A1*A1)
          DGA1=DGA1+D2/8.*KE1(I)*(Q0*Q0/(A0*A0)-Q1*Q1/(A1*A1)
     .          *(1.+2.*A0/A1))

125       AA(IA) = DFQ
          AA(IA+1) = DFA
          AA(IA+2) = DFQ1
          AA(IA+3) = DFA1
          AA(IA+4) = DGQ
          AA(IA+5) = DGA
          AA(IA+6) = DGQ1
          AA(IA+7) = DGA1
          IR = IR+2
          IA = IA+10
          Q0 = Q1
          A0 = A1
          P0 = P1
130     CONTINUE

C
C     Account for tribs and flip columns if necessary
C ********************************************************************

        IF(IT1.EQ.0)GO TO 71

        DO 72 L=IT1,IT2
          I = JT(L)
          DX = DX1(I-1)
          IR = 2*(I-1)
          IA = 5*IR
```

A21

```
            BO = B(I)
            AA(IA-6) = AA(IA-6)+T(1,L)/(DX*BO)
            AA(IA-2) = AA(IA-2)+T(3,L)/(DX*BO)
            R(IR) = R(IR)+T(2,L)/DX
            R(IR+1) = R(IR+1)+T(4,L)/DX
            DX = DX1(I)
            AA(IA+2) = AA(IA+2)+T(1,L)/(DX*BO)
            AA(IA+6) = AA(IA+6)+T(3,L)/(DX*BO)
            R(IR+2) = R(IR+2)+T(2,L)/DX
            R(IR+3) = R(IR+3)+T(4,L)/DX
   72 CONTINUE

   71 FLIP1 = LIB.LE.3
            FLIP2 = LIB.EQ.2.OR.LIB.EQ.5
            IF(.NOT.FLIP1)GO TO 73
            SAVE = AA(1)
            AA(1) = AA(2)
            AA(2) = SAVE
            SAVE = AA(5)
            AA(5) = AA(6)
            AA(6) = SAVE

   73 IF(.NOT.FLIP2) GO TO 74
            SAVE = AA(IND2-2)
            AA(IND2-2) = AA(IND2-3)
            AA(IND2-3) = SAVE
            SAVE = AA(IND2-6)
            AA(IND2-6) = AA(IND2-7)
            AA(IND2-7) = SAVE
   74 CONTINUE

            CALL MAT5(AA,C(2),R(2),IND2,IND3-1)

C
C     Undo flipping and place data into the T-array
C ****************************************************************

            IF(.NOT.FLIP1)GO TO 75

            R(1) = R(2)
            R(2) = 0.

   75 IF(.NOT.FLIP2)GO TO 76
            R(IND3) = R(IND3-1)
            R(IND3-1) = 0.
            GO TO 77

   76 IF(IT0.EQ.0)GO TO 77
            T(1,IT0) = D1*C(IND3-1)
            T(2,IT0) = D1*(R(IND3-1)+Q1)+XC(IND1)
            T(3,IT0) = D1*COSP*Q1/A1*(2.*C(IND3-1)+Q1*B(IND1)/A1)
            T(4,IT0) = D1*COSP*Q1/A1*(2.*R(IND3-1)+Q1)+XM(IND1)

   77 CONTINUE
            RETURN
            END
C ********************************************************************
c                   S U B R O U T I N E   H F R O M A
C ********************************************************************
            SUBROUTINE HFROMA(H, A, B, DBDH, IPC)

C
```

```
C     Given the area and width vs. depth table, and the current area,
C     calculate the depth, channel top width, and derivative of the top width
C     using linear table search (linear is not inefficient because you keep
C     an updatable pointer into the table) and inverse linear interpolation.
C     The limit switches on the table search (top and bottom) are hxs = 0.
C ****************************************************************************

      PARAMETER(MAXXS=1000)

      COMMON/XSDATA/IXS,HXS(MAXXS),AXS(MAXXS),BXS(MAXXS)

      IF (A .LE. 0) STOP 'River has DRIED up.'

      DO 10 I = IPC, MAXXS
         IF (HXS(I+1) .LE. 0) STOP 'River has OVERFLOWED its banks.'
10       IF (A .LE. AXS(I+1)) GOTO 11

11    IPC = I

      DO 20 I = IPC, 1, -1
20       IF (A .GT. AXS(I)) GOTO 21

21    IPC = I
      DBDH = (BXS(I+1) - BXS(I))/(HXS(I+1) - HXS(I))
      DA = A - AXS(I)
      B0 = BXS(I)
      DH = 2.*DA/(B0 + SQRT(B0*B0 + 2.*DBDH*DA))
      B = DBDH*DH + B0
      H = HXS(I) + DH
      RETURN
      END
C ****************************************************************************
c                    S U B R O U T I N E   M A T 5
C ****************************************************************************

      SUBROUTINE MAT5(AA,C,R,M0,M)

      DIMENSION AA(M0),C(M),R(M)
      M1=M-1

C
C     Eliminate extreme diagonals
C ****************************************************************************

      J=1

      DO 1 I=1,M1,2
         D = -AA(J+3)/AA(J+7)
         AA(J) = AA(J)+AA(J+4)*D
         AA(J+1) = AA(J+1)+AA(J+5)*D
         AA(J+2) = AA(J+2)+AA(J+6)*D
         R(I) = R(I) + R(I+1)*D
         D = -AA(J+4)/AA(J)
         AA(J+5) = AA(J+5) + AA(J+1)*D
         AA(J+6) = AA(J+6) + AA(J+2)*D
         R(I+1) = R(I+1) + R(I)*D
         J =J+10
    1 CONTINUE

C
C     Eliminate lower codiagonal
C ****************************************************************************
```

```
        J = 6

        DO 2 I=2,M1
            D = -AA(J)/AA(J-4)
            AA(J+1) = AA(J+1) + AA(J-3)*D
            R(I) = R(I) + R(I-1)*D
            J = J+5
    2 CONTINUE

C
C     In the case of a non-tributary with a rating curve downstream,
C     the bottom row of the matrix needs work.
C ****************************************************************

        IF(AA(M0).EQ.0)GO TO 5
        D = -AA(M0-1)/AA(M0-3)
        R(M) = (R(M)+R(M1)*D)/(AA(M0)+AA(M0-2)*D)

C
C     Eliminate upper codiagonal and normalize matrix
C ****************************************************************

    5 J = 5*M1 - 2

        DO 3 II=1,M1
            I = M1-II+1
            C(I) = C(I+1)* (-AA(J)/AA(J-1))
            R(I) = (R(I)-AA(J)*R(I+1))/AA(J-1)
            J = J-5
    3 CONTINUE
        RETURN
        END
C ****************************************************************
c                 S U B R O U T I N E   N E W
C ****************************************************************
C
        SUBROUTINE NEW(R,Q,A,B,H,EL,I3,C1,C2,C3,C,IT0,IT1,IT2,JT,T,M1,M2,
       .NS,Z,IXSA,IXSB,DBDH)

        INCLUDE 'RIV1H.CMN'

        DIMENSION R(M2),A(M1),Q(M1),B(M1),C1(M1),C2(M1),
       .C3(M1),H(M1),EL(M1),I3(M1),C(M2),JT(NS),T(4,NS),Z(M1),
       .IXSA(M1),IXSB(M1),DBDH(M1)

        IF(IT0.NE.0)D = T(1,IT0)

        DO 60 I=1,M1
            II = 2*I
            IF(IT0.EQ.0)GO TO 1
            R(II-1) = R(II-1)-C(II-1)*D
            R(II) = R(II)-C(II)*D
    1       Q(I)=Q(I)+R(II-1)
            A(I)=A(I)+R(II)
            H1 = H(I)

C
C     Diagnostic check to see if depth goes to zero
C ****************************************************************

        IF (IXSA(I) .GT. 0) THEN
            CALL HFROMA(H(I), A(I), B(I), DBDH(I), IXSA(I))
```

```fortran
              IF (IXSB(I) .GT. 0) THEN
                 CALL HFROMA(HB, A(I), BB, DBDHB, IXSB(I))
                 C1B = C1(I)
                 C1A = 1. - C1B
                 H(I) = C1A*H(I) + C1B*HB
                 B(I) = C1A*B(I) + C1B*BB
                 DBDH(I) = C1A*DBDH(I) + C1B*DBDHB
              ENDIF
           ELSE
              DO 30 K=1,10
                 H0=H1
                 IF(H0.LE.0)WRITE(NERRFIL,819)
819              FORMAT(2X,'ALERT - Computed Depth less than zero')
                 IF(H0.LE.0) WRITE(NERRFIL,818) Z(I),H0,K
818              FORMAT(10X,'Node Elev.=',F6.2,5X,'H0=',F8.2, 5X,'K=',I4)
                 IF(I3(I))10,4,5
4                B0=C1(I)+C2(I)*C3(I)*H0**(C3(I)-1.)
                 H1=H0-(C1(I)*H0+C2(I)*H0**C3(I)-A(I))/B0
                 GO TO 20
5                B0=C1(I)+C2(I)*C3(I)*H0**(I3(I)-1)
                 H1=H0-(C1(I)*H0+C2(I)*H0**I3(I)-A(I))/B0
                 GO TO 20
10               B0=2.*C2(I)/C1(I)*SQRT((2.*C1(I)-H0)*H0)
                 H1=H0-(C1(I)*C2(I)*ACOS(1.-H0/C1(I))-B0*(C1(I)-H0)/2.
               -A(I))/B0
20               IF(ABS((H1-H0)/H0).LE.0.01) GO TO 50
30            CONTINUE

              WRITE(OUT,40)
              STOP 7

50            B(I)=B0
              H(I) = H1
              IF(I3(I))110,104,105
104           DBDH(I)=C2(I)*C3(I)*(C3(I)-1.)*H1**(C3(I)-2.)
              GO TO 120
105           DBDH(I)=C2(I)*C3(I)*(C3(I)-1.)*H1**(I3(I)-2)
              GO TO 120
110           DBDH(I)=4.*(C2(I)/C1(I))**2*(C1(I)-H1)/B0
           ENDIF
120        EL(I) = H(I)+Z(I)
60      CONTINUE

C
C     Pass data to tribs
C **********************************************************************

      IF(IT1.EQ.0)GO TO 70
      DO 80 L=IT1,IT2
         K = JT(L)
         T(1,L) = R(2*K)/B(K)
80    CONTINUE

70    RETURN
40    FORMAT('The Stage-Area equation has FAILED to CONVERGE in',
     .       ' 10 iterations.  RUN ABORTED.')
      END
C **********************************************************************
c                 S U B R O U T I N E   R E A D X S
C **********************************************************************
      SUBROUTINE READXS(XSCODE, IXSA, IXSB, MNODE)
```

```
          INCLUDE 'RIV1H.CMN'

          DIMENSION IXSA(MNODE),IXSB(MNODE),X(MAXXS),Y(MAXXS),IA(MAXXS)

          CHARACTER*6 XSCODE(MNODE)
          CHARACTER*80 CARD
          CHARACTER C*1

          LOGICAL*1 EOF

          COMMON/XSDATA/IXS,HXS(MAXXS),AXS(MAXXS),BXS(MAXXS)
C     ***********************************************************************
C          If no cross-section filename was given, just return.
C     ***********************************************************************
          IF (IXS.LT.1) RETURN

          EOF = .FALSE.
          READ(FXSEC,'(A)',END=11)CARD
          N = 0
          NP = 1
10        CONTINUE

          DO 21 J = 1, MNODE
             IF (XSCODE(J) .EQ. CARD(1:6)) GO TO 22
21        CONTINUE

          WRITE (OUT,'('' CODE '',A,'' NOT FOUND!'')') CARD(1:6)
          STOP
22        IXSA(J) = NP
          IXSB(J) = NP

          DO 20 I = 1, MAXXS
             READ(FXSEC, '(A)', END = 11) CARD
             C = CARD(1:1)
             IF (EOF .OR. (C .GE. 'A' .AND. C .LE. 'Z') .OR.
     .          (C .GE. 'a' .AND. C .LE. 'z')) GOTO 12
             N = N + 1
             READ(CARD, '(2F10.0)') X(N), Y(N)
20        CONTINUE

          GOTO 12

11        EOF = .TRUE.

12        CONTINUE

          N1 = N
          CALL XSECT(X, Y, IA, HXS(NP), BXS(NP), AXS(NP), N1)
          N = NP
          NP = NP + N1
          HXS(NP) = 0.
          N = 0
          IF (.NOT. EOF) GOTO 10
          CLOSE(10)
          RETURN
13        STOP 'Cross-Section File is Empty'
          END
C
C     ***********************************************************************
C                         S U B R O U T I N E   S H E L L
```

A26

```
C ********************************************************************
      SUBROUTINE SHELL(R, IA, N)
C
      DIMENSION R(N), IA(N)

      DO 5 I = 1,N
5     IA(I) = I

      IG = N
10    IG = IG/2
      IF (IG .LE. 0) GOTO 999

      DO 20 I = IG+1, N
         J = I - IG
30       IF (J .LT. 1 .OR. R(IA(J)) .LE. R(IA(J+IG))) GOTO 20
         IT = IA(J)
         IA(J) = IA(J+IG)
         IA(J+IG) = IT
         J = J - IG
         GO TO 30
20    CONTINUE
      GOTO 10
999   RETURN
      END
C ********************************************************************
c                    S U B R O U T I N E   X S E C T
C ********************************************************************
      SUBROUTINE XSECT(X, Y, IA, H, B, A, N)
      INCLUDE 'RIV1H.CMN'
      DIMENSION X(N), Y(N), IA(N), H(N), B(N), A(N)
      DIMENSION LP(NLIST)

C ********************************************************************
C                    Check if X is monotonic
C ********************************************************************

      DO 5 I = 2, N
         IF (X(I) .LE. X(I-1)) STOP
5     CONTINUE

      CALL SHELL(Y, IA, N)

C ********************************************************************
C        Now the Y's are sorted by the index IA.
C        Start from the stream bottom and go up.
C ********************************************************************

      YBOT = Y(IA(N))
      H(1) = 0.
      B(1) = 0.
      A(1) = 0.
      NL = 0

      DO 10 I = 1, N
         IC = N - I + 1
         IS = IA(IC)

C ********************************************************************
C        Calculate the width and area at this point.
C ********************************************************************

         H(I) = YBOT - Y(IS)
```

```fortran
                B(I) = 0.
                A(I) = 0.
                IF (NL .GT. 0) THEN
                    DO 20 IL = 1, NL, 2
                        L0 = LP(IL)
                        L1 = LP(IL+1)
                        IF (Y(IS) .EQ. Y(L0)) THEN
                            X0 = X(L0)
                        ELSE
                            X0 = X(L0) + (X(L0+1) - X(L0))*(Y(IS) - Y(L0))/
                                (Y(L0+1)- Y(L0))
                        ENDIF
                        IF (Y(IS) .EQ. Y(L1+1)) THEN
                            X1 = X(L1+1)
                        ELSE
                            X1 = X(L1) + (X(L1+1) - X(L1))*(Y(IS) - Y(L1))/
                                (Y(L1+1)- Y(L1))
                        ENDIF
20                      B(I) = B(I) + X1 - X0
                    A(I) = A(I-1) + (B(I) + B(I-1))*(H(I) - H(I-1))/2.
                ENDIF
C     ************************************************************************

C         If you've hit the bank, you're done.

C     ************************************************************************
                IF (IS .LE. 1 .OR. IS .GE. N) GOTO 999
C     ************************************************************************

C         Now find out if the new point is a v or a ^ or a \ or a /
C         so that you can adjust the intersections list.
C     ************************************************************************

                IF (Y(IS-1) .LT. Y(IS) .AND. Y(IS) .LT. Y(IS+1)) THEN
                    DO 30 IL = 1, NL, 2
30                      IF (LP(IL) .EQ. IS) GO TO 31
31                  LP(IL) = IS - 1
                ELSE IF (Y(IS-1) .GE. Y(IS) .AND. Y(IS) .GE. Y(IS+1)) THEN
                    DO 40 IL = 2, NL, 2
40                      IF (LP(IL) .EQ. IS - 1) GO TO 41
41                  LP(IL) = IS
                ELSE IF (Y(IS-1) .LT. Y(IS) .AND. Y(IS) .GE. Y(IS+1)) THEN
                    IF (NL .EQ. 0) THEN
                        IL = 0
                    ELSE
                        DO 50 IL = NL, 1, -1
                            IF (LP(IL) .LT. IS) GOTO 51
                            LP(IL + 2) = LP(IL)
50                      CONTINUE
                    ENDIF
51                  LP(IL + 2) = IS
                    LP(IL + 1) = IS - 1
                    NL = NL + 2
                ELSE
                    DO 60 IL = 1, NL
60                      IF (LP(IL) .EQ. IS-1) GOTO 61
61                  NL = NL - 2
                    IF (IL .LE. NL) THEN
                        DO 62 JL = IL, NL
62                          LP(JL) = LP(JL+2)
                    ENDIF
                ENDIF
```

```
10      CONTINUE
999     N = I
        RETURN
        END

C ****************************************************************
C           S U B R O U T I N E   G R E G O R I A N   D A T E
C ****************************************************************
C
        SUBROUTINE GREGORIAN_DATE(ELAPSE,YEAR,MONTH,GDAY,HOUR)


***** Variable declarations

        LOGICAL LEAP_YEAR
        CHARACTER MONTH*9
        INTEGER YEAR, GDAY

C ****************************************************************
C         Determine if new year (regular or leap) and increment year
C ****************************************************************

        JDAYG=ELAPSE - 1
        HOUR = (ELAPSE-FLOAT(JDAYG) - 1)*24.
        LEAP_YEAR        = MOD(YEAR,4).EQ.0

        IF (.NOT.LEAP_YEAR.AND.JDAYG.EQ.365) THEN
           JDAYG     = JDAYG-365
           YEAR      = YEAR+1
           LEAP_YEAR = MOD(YEAR,4).EQ.0
        ELSE IF (JDAYG.EQ.366) THEN
           JDAYG     = JDAYG-366
           YEAR      = YEAR+1
           LEAP_YEAR = .FALSE.
        END IF

C ****************************************************************
C       Determine month and day of year
C ****************************************************************

        IF (LEAP_YEAR) THEN
           IF (JDAYG.GE.0.AND.JDAYG.LT.31) THEN
              GDAY  = JDAYG+1
              DAYM  = 31.0
              MONTH = '  January'
           ELSE IF (JDAYG.GE.31.AND.JDAYG.LT.60) THEN
              GDAY  = JDAYG-30
              DAYM  = 29.0
              MONTH = ' February'
           ELSE IF (JDAYG.GE.60.AND.JDAYG.LT.91) THEN
              GDAY  = JDAYG-59
              DAYM  = 31.0
              MONTH = '    March'
           ELSE IF (JDAYG.GE.91.AND.JDAYG.LT.121) THEN
              GDAY  = JDAYG-90
              DAYM  = 30.0
              MONTH = '    April'
           ELSE IF (JDAYG.GE.121.AND.JDAYG.LT.152) THEN
              GDAY  = JDAYG-120
              DAYM  = 31.0
              MONTH = '      May'
           ELSE IF (JDAYG.GE.152.AND.JDAYG.LT.182) THEN
```

```
                  GDAY  = JDAYG-151
                  DAYM  = 30.0
                  MONTH = '      June'
            ELSE IF (JDAYG.GE.182.AND.JDAYG.LT.213) THEN
                  GDAY  = JDAYG-181
                  DAYM  = 31.0
                  MONTH = '      July'
            ELSE IF (JDAYG.GE.213.AND.JDAYG.LT.244) THEN
                  GDAY  = JDAYG-212
                  DAYM  = 31.0
                  MONTH = '     August'
            ELSE IF (JDAYG.GE.244.AND.JDAYG.LT.274) THEN
                  GDAY  = JDAYG-243
                  DAYM  = 30.0
                  MONTH = 'September'
            ELSE IF (JDAYG.GE.274.AND.JDAYG.LT.305) THEN
                  GDAY  = JDAYG-273
                  DAYM  = 31.0
                  MONTH = '   October'
            ELSE IF (JDAYG.GE.305.AND.JDAYG.LT.335) THEN
                  GDAY  = JDAYG-304
                  DAYM  = 30.0
                  MONTH = ' November'
            ELSE IF (JDAYG.GE.335.AND.JDAYG.LT.366) THEN
                  GDAY  = JDAYG-334
                  DAYM  = 31.0
                  MONTH = ' December'
            END IF
      ELSE
            IF (JDAYG.GE.0.AND.JDAYG.LT.31) THEN
                  GDAY  = JDAYG+1
                  DAYM  = 31.0
                  MONTH = '   January'
            ELSE IF (JDAYG.GE.31.AND.JDAYG.LT.59) THEN
                  GDAY  = JDAYG-30
                  DAYM  = 29.0
                  MONTH = ' February'
            ELSE IF (JDAYG.GE.59.AND.JDAYG.LT.90) THEN
                  GDAY  = JDAYG-58
                  DAYM  = 31.0
                  MONTH = '     March'
            ELSE IF (JDAYG.GE.90.AND.JDAYG.LT.120) THEN
                  GDAY  = JDAYG-89
                  DAYM  = 30.0
                  MONTH = '     April'
            ELSE IF (JDAYG.GE.120.AND.JDAYG.LT.151) THEN
                  GDAY  = JDAYG-119
                  DAYM  = 31.0
                  MONTH = '       May'
            ELSE IF (JDAYG.GE.151.AND.JDAYG.LT.181) THEN
                  GDAY  = JDAYG-150
                  DAYM  = 30.0
                  MONTH = '      June'
            ELSE IF (JDAYG.GE.181.AND.JDAYG.LT.212) THEN
                  GDAY  = JDAYG-180
                  DAYM  = 31.0
                  MONTH = '      July'
            ELSE IF (JDAYG.GE.212.AND.JDAYG.LT.243) THEN
                  GDAY  = JDAYG-211
                  DAYM  = 31.0
                  MONTH = '     August'
            ELSE IF (JDAYG.GE.243.AND.JDAYG.LT.273) THEN
```

```
                GDAY  = JDAYG-242
                DAYM  = 30.0
                MONTH = 'September'
            ELSE IF (JDAYG.GE.273.AND.JDAYG.LT.304) THEN
                GDAY  = JDAYG-272
                DAYM  = 31.0
                MONTH = '  October'
            ELSE IF (JDAYG.GE.304.AND.JDAYG.LT.334) THEN
                GDAY  = JDAYG-303
                DAYM  = 30.0
                MONTH = ' November'
            ELSE IF (JDAYG.GE.334.AND.JDAYG.LT.365) THEN
                GDAY  = JDAYG-333
                DAYM  = 31.0
                MONTH = ' December'
            END IF
        END IF
        RETURN
        END


C *************************************************************************
C              S U B R O U T I N E   J U L I A N  D A T E
C *************************************************************************

        SUBROUTINE JULIAN DATE(STADY,ENDY,SYEAR,SMONTH,SDAY,SHOUR,
     .        EYEAR,EMONTH,EDAY,EHOUR)

***** Variable declarations

        DIMENSION DAYM(12)
        INTEGER   SYEAR,SMONTH,SDAY,EYEAR,EMONTH,EDAY

C
C     Specify days for each month of the year
C *************************************************************************

        DAYM(1)   = 31.0
        DAYM(2)   = 29.0
        DAYM(3)   = 31.0
        DAYM(4)   = 30.0
        DAYM(5)   = 31.0
        DAYM(6)   = 30.0
        DAYM(7)   = 31.0
        DAYM(8)   = 31.0
        DAYM(9)   = 30.0
        DAYM(10)  = 31.0
        DAYM(11)  = 30.0
        DAYM(12)  = 31.0


C     Determine Julian day for start time
C *************************************************************************

        STADY = 0.
        DAYM(2) = 28.

        IF(MOD(SYEAR,4).EQ.0)DAYM(2) = 29.0

        IF(SMONTH.GT.1) THEN
            DO 100 I=1,SMONTH-1
                STADY = STADY + DAYM(I)
100         CONTINUE
```

A31

```
          END IF

          STADY = STADY + SDAY + SHOUR/24.

C
C         Determine Julian day for end time
C  *************************************************************************

          ENDY = 0.

          IF(EYEAR.GT.SYEAR) THEN
              DO 200 I=SYEAR,EYEAR-1
                  IF(MOD(I,4).EQ.0)THEN
                      ENDY= ENDY + 366.
                  ELSE
                      ENDY = ENDY + 365.
                  END IF
200           CONTINUE
          END IF

          DAYM(2) = 28.
          IF(MOD(EYEAR,4).EQ.0)DAYM(2)=29.

          IF(EMONTH.GT.1) THEN
              DO 300 I=1,EMONTH-1
                  ENDY = ENDY + DAYM(I)
300           CONTINUE
          END IF

          ENDY = ENDY +EDAY + EHOUR/24.

          RETURN
          END

C  *************************************************************************
C                    S U B R O U T I N E   R I V D U
C  *************************************************************************
          SUBROUTINE RIVDU(MNODE,DX,RMILE,Q,QLC,QLT,A,B,EL,XMAN)
C
          SAVE NFIRST

          INCLUDE 'RIV1H.CMN'
          CHARACTER*30 DNAME (18)

          DIMENSION DX(MNODE),RMILE(MNODE),Q(MNODE),QLC(MNODE),
     +        QLT(MNODE),A(MNODE),B(MNODE),EL(MNODE),XMAN(MNODE)

          DATA DNAME/
     +          'Delta-x                ' , 'River Mile           ',
     +          'Flow (cfs)             ' , 'Constant Lat. Flow   ',
     +          'Time Var. Lat Flow     ' , 'Area                 ',
     +          'Width                  ' , 'Water Surf. Elev     ',
     +          'Mannings n             ' , 'Dummy                ',
     +          'Dummy                  ' , 'Dummy                ',
     +          'Dummy                  ' , 'Dummy                ',
     +          'Dummy                  ' , 'Dummy                ',
     +          'Dummy                  ' , 'Dummy                '/

          DUMMY = 0.
          RDUMMY = 0.

          IF (NFIRST .NE. 1) THEN
```

```
           NCHEM = 1
           WRITE (RIVDMP, 6000) NCHEM
 6000      FORMAT (I5)
C
           WRITE (RIVDMP, 6010) RDUMMY,RDUMMY
 6010      FORMAT(A40,E10.2,F10.2)
           TIME = 0.
           TFIN = 0.
C
           WRITE (RIVDMP, 6020) TIME, TFIN, MNODE
 6020      FORMAT(1X,F10.2,F10.2,/,I5)
C
           WRITE (RIVDMP, 6030) (DNAME (JK), JK = 1, 6 + (12*NCHEM))
 6030      FORMAT(A30)
           NFIRST = 1
       END IF
C
C
       DO 100 I=1,MNODE
           SYSDUMP(1,I)=DX(I)
           SYSDUMP(2,I)=RMILE(I)
           SYSDUMP(3,I)=Q(I)
           SYSDUMP(4,I)=QLC(I)
           SYSDUMP(5,I)=QLT(I)
           SYSDUMP(6,I)=A(I)
           SYSDUMP(7,I)=B(I)
           SYSDUMP(8,I)=EL(I)
           SYSDUMP(9,I)=XMAN(I)
           SYSDUMP(10,I)=DUMMY
           SYSDUMP(11,I)=DUMMY
           SYSDUMP(12,I)=DUMMY
           SYSDUMP(13,I)=DUMMY
           SYSDUMP(14,I)=DUMMY
           SYSDUMP(15,I)=DUMMY
           SYSDUMP(16,I)=DUMMY
           SYSDUMP(17,I)=DUMMY
           WRITE(RIVDMP,6040)    I,ELAPSE,DX(I),RMILE(I),Q(I),QLC(I),
     +     QLT(I),A(I),B(I),EL(I),XMAN(I),DUMMY,DUMMY,DUMMY,
     +     DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY

 6040      FORMAT(1X,I5,F10.2,3X,/,6(E11.3),/,6(E11.3))
 100   CONTINUE

       INUM=MNODE
C
       RETURN
       END
C ****************************************************************
C             F U N C T I O N     V A L N E W
C
C        THIS FUNCTION PERFORMS LINEAR INTERPOLATION
C ****************************************************************

       FUNCTION VALNEW(DAY,T1,T2,V1,V2)
C
       RATIO  = (T2-DAY)/(T2-T1)
       VAL    = (1.0-RATIO)*V2+RATIO*V1
       VALNEW = VAL
       RETURN
       END
C ****************************************************************
C                 S U B R O U T I N E  R I V 1 H _ I N T
```

```
C ***********************************************************************
      SUBROUTINE RIV1H_INIT
      INCLUDE 'RIV1H.CMN'
      ISEGOUT(1)=1
      ISEGOUT(2)=2
      ISEGOUT(3)=3
      ISEGOUT(4)=4
      ISEGOUT(5)=5
      ISEGOUT(6)=6
      IDISPLAY(1)=1
      IDISPLAY(2)=2
      IDISPLAY(3)=3
      IDISPLAY(4)=4
      IDISPLAY(5)=5
      IDISPLAY(6)=6
      IDISPLAY(7)=7
      IDISPLAY(8)=8
      IDISPLAY(9)=9
      IDISPLAY(10)=10
      LABELS(1)  = 'Delta-x'
      LABELS(2)  = 'River Mile'
      LABELS(3)  = 'Flow (cfs)'
      LABELS(4)  = 'QLC(cfs/f)'
      LABELS(5)  = 'QLD(cfs/f)'
      LABELS(6)  = 'Area (sft)'
      LABELS(7)  = 'Width (ft)'
      LABELS(8)  = 'WSElev(ft)'
      LABELS(9)  = 'Mannings n'
      LABELS(10)= 'Dummy'
      LABELS(11)= 'Dummy'
      LABELS(12)= 'Dummy'
      LABELS(13)= 'Dummy'
      LABELS(14)= 'Dummy'
      LABELS(15)= 'Dummy'
      LABELS(16)= 'Dummy'
      LABELS(17)= 'Dummy'
      RETURN
      END

C ***********************************************************************
C      S U B R O U T I N E   T I M E   V A R Y I N G   D A T A
C ***********************************************************************

      SUBROUTINE TIME_VARYING_DATA(JDAY,IYR,TNXTVD,NBC,DLT,
     .                             DUMMY,IQL,QLT)

      INCLUDE 'RIV1H.CMN'

      REAL JDAY

      LOGICAL   INT_BC(IBRAN)
      LOGICAL   INT_LAT

      CHARACTER*15 INTOPT, BCFNAME(IBRAN)

C      Dimensions for boundary conditions
C ***********************************************************************

      DIMENSION DUMMY(IBRAN),DUMMY1(IBRAN),DUMMY2(IBRAN),
     .       TNXTBC(IBRAN),TNXTBC2(IBRAN),NUMBC(IBRAN),INBC(IBRAN)
```

```
C     Dimensions for lateral inflows
C     ***************************************************************

      DIMENSION QLT(IND1), QLT2(IND1), QLT1(IND1),NPS_REA(IND1)

C     Dimensions for time steps
C     ***************************************************************

      DIMENSION TDT(200),DTV(200)

      SAVE

C
C     I.  Read all time-varying data on initial call
C     ***************************************************************

      IF(IFIRST.NE.1)THEN

         IDUM = 0
         DUM = 0.

C     Determine initial year of simulation
C     ***************************************************************

         INITYR = IYR

C     1st - Read timesteps
C     ***************************************************************

         READ(INPUT,160)NUMDT
         WRITE(OUT,260)NUMDT
         DO I=1,NUMDT
            READ(INPUT,170)IYR,IMO,IDY,THR,DTV(I)
            CALL JULIAN_DATE(DUM,TDT(I),INITYR,IDUM,IDUM,DUM,
     *                       IYR,IMO,IDY,THR)
            WRITE(OUT,270)IYR,IMO,IDY,THR,TDT(I),DTV(I)
         END DO

C     2nd - Open file for boundary conditions
C     ***************************************************************

         IBF = 30
         DO 100 I=1,NBC
         READ(INPUT,165)BCFNAME(I)
         OPEN(UNIT=IBF,FILE=BCFNAME(I),STATUS='OLD')
         READ(IBF,180)INBC(I),NUMBC(I),INTBC
         WRITE(OUT,280)INBC(I)
         WRITE(OUT,290)BCFNAME(I)

         IF(INTBC.GE.1)THEN
            INT_BC(I)=.TRUE.
            INTOPT ='LINEAR INTERPOL'
         ELSE
            INTOPT ='STEP FUNCTION'
         END IF

         WRITE(OUT,300)INTOPT
         TNXTBC(I) = 0.
         IBF = IBF + 1
100      CONTINUE
```

```
C      3rd - Read time varying lateral inflow file
C ***********************************************************************

       IF(IQL.GE.1) THEN
          READ(LAT,180)NUMLAT_UD,NUML,LATOPT
          IF(LATOPT .GE. 1)THEN
             INT_LAT = .TRUE.
             INTOPT ='LINEAR'
          ELSE
             INTOPT ='STEP'
          ENDIF

          READ(LAT,190)(NPS_REA(I), I = 1, NUML)
          WRITE(OUT,310)
          WRITE(OUT,320)(NPS_REA(I), I = 1, NUML)
          WRITE(OUT,300)INTOPT
       END IF

       IDT = 1
       IFIRST = 1
       END IF

C
C      II.  Assign and/or update all time-varying data all calls
C ***********************************************************************

C      1st - Assign time steps
C ***********************************************************************

       IF (JDAY.GE.TNXTDT) THEN
          DO WHILE (JDAY.GE.TNXTDT)
             DLT = DTV(IDT)
             IDT = IDT + 1
             TNXTDT = TDT(IDT)
          END DO
C         WRITE(OUT,330)JDAY,DLT,TNXTDT
       END IF

       TNXTVD = MIN(TNXTVD,TNXTDT)

C      2nd -  Read boundary data
C ***********************************************************************

        IBF = 30

        DO 600 I=1,NBC
          IF (JDAY.GE.TNXTBC(I)) THEN
             DO WHILE(JDAY.GE.TNXTBC(I))
                TNXTBC2(I) =TNXTBC(I)
                DUMMY2(I) = DUMMY1(I)
                READ(IBF,170)IYR,IMO,IDY,THR, DUMMY1(I)
                CALL JULIAN_DATE(DUM,TNXTBC(I),INITYR,IDUM,IDUM,DUM,
                                 IYR,IMO,IDY,THR)
             END DO
             TNXTVD = MIN(TNXTVD,TNXTBC(I))
             DUMMY(I)=DUMMY1(I)
C            WRITE(OUT,340)  INBC(I),IYR,IMO,IDY,THR,JDAY,
C                            DUMMY1(I),TNXTBC(I)
          END IF
          IF(INT_BC(I)) DUMMY(I) = VALNEW(JDAY,TNXTBC(I),TNXTBC2(I),
                        DUMMY1(I),DUMMY2(I))
          IBF = IBF + 1
```

```
600   CONTINUE

C     3rd - Assign lateral inflow data
C ********************************************************************
C
      IF(IQL.GE.1) THEN

         IF (JDAY.GE.TNXTLT) THEN
            DO WHILE (JDAY.GE.TNXTLT)
               TNXTLT2 = TNXTLT
               DO I=1,NUML
                  QLT2(NPS_REA(I))=QLT1(NPS_REA(I))
               END DO
               READ(LAT,200)IYR,IMO,IDY,THR,
     .                      (QLT1(NPS_REA(IK)),IK=1,NUML)
               CALL JULIAN_DATE(DUM, TNXTLT,INITYR,IDUM,IDUM,DUM,
     .         IYR,IMO,IDY,THR)
            END DO
            TNXTVD = MIN(TNXTVD,TNXTLT)

            DO IK=1,NUML
               QLT(NPS_REA(IK))=QLT1(NPS_REA(IK))
            END DO

C            WRITE(OUT,350) IYR,IMO,IDY,THR,JDAY, TNXTLT
C            WRITE(OUT,360)(NPS_REA(I),I=1,NUML)
C            WRITE(OUT,370)(QLT(NPS_REA(IK)),IK=1,NUML)
         END IF

         IF(INT_LAT) THEN
            DO J=1,NUML
               QLT(NPS_REA(J)) = VALNEW(JDAY,TNXTLT,TNXTLT2,
     .                           QLT1(NPS_REA(J)),QLT2(NPS_REA(J)))
            END DO
         END IF
      END IF
      RETURN
C     INPUT FORMATS
C ********************************************************************

  160 FORMAT(I10)
  165 FORMAT(A15)
  170 FORMAT(I5,I5,I5,2F10.4)
  180 FORMAT(8I10)
  190 FORMAT(I5)
  200 FORMAT(I5,I5,I5,150(F10.4))

C     OUTPUT FORMATS
C ********************************************************************

  260 FORMAT(/,'***************** TIME STEPS *****************',/,
     .' Number of Time/Date Pairs = ',I10)
  270 FORMAT('Year ',I6,' Month ',I3,' Day ',I3,' Hour',F8.4,
     .' Julian day ',F10.4,'  Time Step (sec) = ',F10.4)
  280 FORMAT(/,'*** Boundary Conditions Specified for Segment ***',I4)
  290 FORMAT('   Will be Read From File = ',A15)
  300 FORMAT('     Interpolation Option:    ',A15)
  310 FORMAT(/,' Time Varying Lateral Inflows Specified at
     .x-sections')
  320 FORMAT(10(1X,I5))
  330 FORMAT(/,'*******Time Step Updated at Julian Date',F10.4,
     .' to ',F10.4,' (sec) ',
```

A37

```
      .'Next Update on Julian Day ',F10.4)
  340 FORMAT(/,'******* Boundary Condition for Segment ',I5,
      .' Updated at: Year ',I6,' Month ',I3,' Day ',I3,' Hour',F8.4,
      .' Julian Date',F10.4,' TO  ',F10.4,' Next Update on Julian Day '
      .,F10.4)
  350 FORMAT(/,'*******Lateral Inflows Updated on: Year ',I6,' Month '
      .,I3,' Day ',I3,' Hour ',F8.4,' Julian Date',F10.4,
      .' Next Update on Julian Day ',F10.4, ' New Values are:')
  360 FORMAT(' ***Cross-Section:',150I10)
  370 FORMAT(' ***Flow (cfs/f) :',150F10.4)
      END
```

APPENDIX B: LIST OF THE WATER QUALITY CODE RIV1Q

```
C   ***********************************************************************
C                         CE-QUAL-RIV1, VERSION 2                         *
C                     RIV1Q CODE - WATER QUALITY MODEL                    *
C                             JULY, 1995                                  *
C                                                                         *
C          CE-QUAL-RIV1Q was originally developed by Ohio State           *
C               University, Bedford, Sykes, and Libicki                   *
C                                                                         *
C                         and later modified by                          *
C                      M. Dortch and T. Schneider                        *
C             Water Quality and Contaminant Modeling Branch              *
C                       Environmental Laboratory                         *
C                    USAE Waterways Experiment Station                    *
C                     Vicksburg, Mississippi, and                        *
C                          D. M. Griffin, Jr.                            *
C                       Louisiana Tech University                        *
C                          Ruston, Louisiana                             *
C                                                                         *
C             This version has been further modified by:                 *
C                                                                         *
C                     Tim Wool and James Martin                          *
C                        AScI Corporation                                *
C                        Athens, Georgia                                 *
C                                                                         *
C   ***********************************************************************

        BLOCK DATA

C       This subprogram initializes those variables used in common block
C       statements
C   ***********************************************************************

        INCLUDE 'RIV1Q.CMN'
        INCLUDE 'TRANSP.CMN'

        DATA CP /12*.FALSE./
        DATA DARK /.FALSE./
        DATA PI/3.14159/
        DATA TBIOS,TAMMON,TPHYSI/1.0,1.0,1.0/
        DATA APCONT,ANCONT/0.01,0.075/
        DATA ONEQUI,ONITRI,OPDECY,OFEDEC,OMNDEC/0.35,4.57,1.59,0.14,0.15/
        DATA DEPTH0,RHO0/5.0,1.0/
        END

C       End of Block Data Statement
C
C   ***********************************************************************
C                              PROGRAM RIV1-Q
C   ***********************************************************************

        PROGRAM RIV1Q

        INCLUDE 'RIV1Q.CMN'
        INCLUDE 'TRANSP.CMN'

        DIMENSION IBYPASS(ISYS)
        DIMENSION TPRNT(IUPDATE), TPR(IUPDATE)
        DIMENSION TPLT(IUPDATE), TPL(IUPDATE)
        CHARACTER MONTH*9,FILES(30)*12, RIVPATH*30, MESSFIL*12
        LOGICAL CTRL
        INTEGER YEAR,SYEAR,SMONTH,SDAY,EYEAR,EMONTH,EDAY
        COMMON/ABLOCK/  NS
```

```
          CHARACTER*8  BLABEL(12),BUNIT(12)/12*' (MG/L) '/,
      .               ALABEL(12)/'  TEMP  ',' CBODNS ','  ORGAN ',
      .               '  NH3N  ',' NO3N  ','  ORG-P  ','   PO4   ',
      .               '   MN   ',' FE   ','   DO   ','  CLFM ',
      .               '  ALGAE  '/,AUNIT(2)/'(DEG. C)',' COL-DL '/,
      .               FMT1/', F12.1'/, FMT2/', F12.2'/,
      .     FMT(15)/'(1X,I2,F','8.2     ',12*'       ',')'         '/

          CHARACTER*9 ADATE
          CHARACTER*80 TITLE

          OPEN (UNIT = 1, STATUS = 'OLD', ACCESS = 'SEQUENTIAL',
      1        FILE = 'RIV1Q.CTL',IOSTAT=ISTAT)

          IF (ISTAT .NE. 0)THEN
             WRITE(6,6000)
 6000        FORMAT(' RIV1Q Control File Not Found ')
             STOP
          ELSE
             READ(1,8000)INFIL
             READ(1,8000)HYDIN
             READ(1,8000)LATIN
             READ(1,8000)METIN
 8000        FORMAT(15X,A12)
             CLOSE(UNIT=1)
          ENDIF


 C    F)  Open files
 C **************************************************************************

          DO 1010 I = 1, 12
             IF (INFIL (I:I) .EQ. '.') GO TO 1020
                OUTFIL (I:I) = INFIL (I:I)
                GRPFIL (I:I) = INFIL (I:I)
                DMPFIL (I:I) = INFIL (I:I)
             GO TO 1010
 1020        CONTINUE
                OUTFIL (I:I + 3) = '.OUT'
                DMPFIL (I:I + 3) = '.EDF'
                GRPFIL (I:I + 3) = '.GRF'
             GO TO 1030
 1010 CONTINUE
 1030 CONTINUE

          OPEN (UNIT=INPUT, FILE= INFIL, STATUS='OLD')
          OPEN (UNIT=OUT, FILE= OUTFIL, STATUS='UNKNOWN')
          OPEN (UNIT=HYD,FILE=HYDIN,FORM='UNFORMATTED',STATUS='OLD')
          OPEN (UNIT=GRAPH,FILE=GRPFIL,FORM='UNFORMATTED',STATUS='UNKNOWN')
          OPEN (UNIT=RIVDMP,FILE=DMPFIL, STATUS='UNKNOWN')

 C **************************************************************************
 C     NOTE: IND1  = Total Number of Nodes in System
 C           MTIME = Total number of timesteps in the simulation
 C           NS    = Number of segments in the system
 C **************************************************************************

 C    2)  Read hydrodynamic linkage and input file
 C **************************************************************************


 C    a) Read simulation times and controls from hydrodynamic linkage file
```

B3

```
C  ************************************************************************

      READ (HYD) MNODE,STARTTIME,ENDTIME,SYEAR,NS

      INITYR = SYEAR

      DO 1000 I =1,NS
         READ(HYD) JT(I),ITO(I),IT1(I),IT2(I),ORDER(I),NNODE(I),
     1            NODE1(I),ID(I),JBC(I)
 1000 CONTINUE


C     B)  Read and write global constants (not segment specific)
C         from input file
C  ************************************************************************
      READ (INPUT,10)    TITLE
      READ (INPUT,1111)  START, END
      READ (INPUT,1111)  TBIOS,TAMMON,TPHYSI
      READ (INPUT,1111)  APCONT,ANCONT
      READ (INPUT,1111)  ONEQUI,ONITRI,OPDECY,OFEDEC,OMNDEC
      READ (INPUT,1111)  DAWN, SUNSET
      READ (INPUT,1112)  ICL
      READ (INPUT,1112)  NPRINT
      READ (INPUT,1116)  (TPRNT(I),TPR(I),I=1,NPRINT)
      READ (INPUT,1112)  NPLOT
      READ (INPUT,1116)  (TPLT(I),TPL(I),I=1,NPLOT)
      READ (INPUT,1115)  (IBYPASS(I),I=1,ISYS)
C
      WRITE (OUT,1190)
      WRITE (OUT,1191)
      WRITE (OUT,1192)
      WRITE (OUT,1190)
      WRITE (OUT,200)   TITLE
      WRITE (OUT,1193)
      WRITE (OUT,1201) START, END, ICL
      WRITE (OUT,1202) TBIOS,TAMMON,TPHYSI
      WRITE (OUT,1203) APCONT,ANCONT
      WRITE (OUT,1204) ONEQUI,ONITRI,OPDECY,OFEDEC,OMNDEC
      WRITE (OUT,1205) DAWN, SUNSET
      WRITE (OUT,1206)

      DO I=1,NPRINT
         WRITE(OUT,1208) TPRNT(I),TPR(I)
      END DO

      WRITE (OUT,1207)

      DO I=1,NPLOT
         WRITE(OUT,1208) TPLT(I),TPL(I)
      END DO

      WRITE (OUT,1209) (IBYPASS(I),I=1,12)
      WRITE (OUT,1210)

C     Initialize QWIND to false
C  ************************************************************************

      DO I=1,NS
         QWIND(I)=.FALSE.
      END DO
```

```fortran
C     Initialize counters for printing
C     ********************************************************************

      TEST PRINT = 0.
      IPRINT     = 1
      TPRINT     = TPRNT(IPRINT)
      IFIRST     = 0

C     Initialize counters for ploting                    .
C     ********************************************************************

      TEST PLOT = 0.
      IPLOT     = 1
      TPLOT     = TPLT(IPLOT)


C     C) Determine name and open lateral inflow file (if used)
C     ********************************************************************

      IF(ICL.GE.1)THEN
          OPEN (UNIT=LAT, FILE= LATIN , STATUS='OLD')
      ENDIF

C     D) Read reach specific constants, initial conditions
C     ********************************************************************

C     NOTE: The "DO 2" loop below reads the stream card and constant
C     card for each segment.  Then, the initital values for each
C     contaminant, the constant lateral inflow concentrations, and
C     the value of the dispersion coefficient area read in for each
C     node in each segment (segment).
C     ********************************************************************
      WRITE(OUT,1193)
      WRITE(OUT,1320)
      WRITE(OUT,1193)

      DO 2 IB = 1,NS
          IBC(IB) = 0

C     1) Read stream card
C     ********************************************************************

          READ(INPUT,1250) IDLL, SNAME0, IDAM0, DAMK0
          WRITE(OUT,1251) IDLL, SNAME0, IDAM0, DAMK0
 1250     FORMAT(I2,10A4,I2,F10.0)
 1251     FORMAT(//,5x,' For Segment # ',I2,5x' Desc:    ',10A4,/,
     .    21x,'  IDAM0 = ',I2,5x,'  DAMK0 = ',F10.0,/,5x,70('_'))

          DO 100 L0 = 1,NS
              L = L0
              IF (ID(L0).EQ.IDLL) GO TO 101
  100     CONTINUE

          WRITE(OUT,102) IDLL
  102     FORMAT(' ID#',I3,' NOT FOUND IN OUTPUT FROM RIV1H.')

          STOP

  101     ID0(IB) = L
          IDAM(L) = IDAM0
          IF (IDAM0.GT.0) DAMK0 = DAMK0/0.3048
          DAMK(L) = DAMK0
```

B5

```
C  *******************************************************************
C        NOTE: The constant damk0 above is used as input to Wilhelm's
C              structural reaeration equation, REF: Wilhelm, S.C. and
C              Smith, D.R. 1981. Reaeration Through Gated-Conduit Outlet
C              Works. Technical Rept. E-81-5. US Army Engineer Waterways
C              Experiment Station, Vicksburg, Mississippi.
C  *******************************************************************

           DO 103 JJJ = 1,10
    103    SNAME(JJJ,L) = SNAME0(JJJ)

C        2)  Read constant card (namelist variables)
C  *******************************************************************

           QWINDO=.FALSE.

C        3)  Read model input constants
C  *******************************************************************

           READ(INPUT,1111)  ADN(IB),AG(IB),AKN(IB),AKNX(IB),AK1(IB)
           READ(INPUT,1111)  ATB(IB),ATS(IB),APO4(IB),BK(IB),CSINK(IB)
           READ(INPUT,1111)  E1(IB),E2(IB),KALGDK(IB), KALGRO(IB)
           READ(INPUT,1111)  KNCBDN(IB),KOALDK(IB),KOCBDN(IB),KOCB1(IB)
           READ(INPUT,1111)  KON(IB),TDUM(IB),TSINK(IB)
           READ(INPUT,1111)  TSIV(IB),KCOLIDK(IB),KMNDK(IB),KFEDK(IB),
        .                    OXIDAT(IB)
           READ(INPUT,1111)  HNEFSW(IB),KNPOOL(IB),KPO4X(IB),KDSED(IB),
      1                      ACK(IB)
           READ(INPUT,1111)  LAMBDA0(IB),LAMBDA1(IB),LAMBDA2(IB),
      1                      ALPHA0(IB), KNSET(IB)
           READ(INPUT,1111)  ABSR(IB),CBODSR(IB),FCBOD(IB),KPDK(IB),
        .                    KPSET(IB)
           READ(INPUT,1111)  BENPO4(IB),SOD(IB),MACROB(IB),MACGRO(IB),
        .                    MACDKY(IB)
           READ(INPUT,1113)  KLITE(IB), ITEM(IB)
           READ(INPUT,1114)  QWINDO


    1111   FORMAT(5(8X,F8.0))
    1112   FORMAT(5(8X,I8))
    1113   FORMAT(8X,F8.0,8X,I8)
    1114   FORMAT(8X,L8)
    1115   FORMAT(12I5)
    1116   FORMAT(4(8X,F8.0))
C
           IF(QWINDO) QWIND(IB)=.TRUE.
           IF (IB.EQ.1) GO TO 4
           M1 = L
           M2 = L
           GO TO 5
      4    M1 = 1
           M2 = NS
      5    CONTINUE

           WRITE(OUT,1321) IB
           WRITE(OUT,1324) TDUM(IB), ATB(IB), ATS(IB), TSINK(IB),ITEM(IB)
           WRITE(OUT,1325) KNCBDN(IB), KOALDK(IB), KOCBDN(IB),KOCB1(IB),
        .                  KON(IB)
           WRITE(OUT,1328) AG(IB),E1(IB),E2(IB),TSIV(IB),ADN(IB),AKN(IB),
        .                  AKNX(IB), APO4(IB)
           WRITE(OUT,1330) KFEDK(IB), KCOLIDK(IB), SOD(IB), KNPOOL(IB),
        .                  KPO4X(IB), KDSED(IB)
```

```
            WRITE(OUT,1331) MACROB(IB),MACGRO(IB),MACDKY(IB),KALGRO(IB),
           1                KALGDK(IB),OXIDAT(IB),KMNDK(IB)
            WRITE(OUT,1332) BK(IB), AK1(IB), KLITE(IB), ACK(IB)
            WRITE(OUT,1333) HNEFSW(IB), M, QWINDO
            WRITE(OUT,1334) LAMBDA0(IB), LAMBDA1(IB), LAMBDA2(IB),
           1                ALPHA0(IB)
            WRITE(OUT,1335) KNSET(IB), ABSR(IB), CBODSR(IB), FCBOD(IB)
            WRITE(OUT,1336) KPDK(IB), KPSET(IB), BENPO4(IB)


C        4) Read initial conditions cards
C  ****************************************************************************

            M1 = NODE1(L)
            M2 = M1 + NNODE(L)-1

            DO 9 I = M1,M2
               READ(INPUT,7) (INIT(N,I),N=1,12)
               READ(INPUT,7) (CLC(N,I),N=1,12)
           7   FORMAT(12F7.0)

C        5) Read in dispersion coefficient in ft2/sec for each reach
C  ****************************************************************************

               READ(INPUT,11) DISP(I)
               DISP(I) = DISP(I)*8026.8
          11   FORMAT(F10.0)
           9   CONTINUE

         2 CONTINUE

C            End of "DO 2" LOOP - All initial data read in
C  ****************************************************************************

C     E)  Determine name and open meteorological input file (if used)
C  ****************************************************************************

         IF (ITEM(1).EQ.0)THEN
            OPEN (UNIT=MET, FILE= METIN , STATUS='OLD')
         ENDIF


C     F)  Read boundary conditions ID card - locations where b.c. are in effect
C  ****************************************************************************


         READ(INPUT,*) IBC
         NBC = 0

C  ****************************************************************************
C      NOTE:NBC is number of locations for boundary condition changes
C  ****************************************************************************

         DO 38 L = 1,NS
            IF(IBC(L).EQ.0)GO TO 39
            NBC = NBC + 1
        38 CONTINUE
        39 CONTINUE

C
C     F)  Convert time from HH.MM to days
C  ****************************************************************************
```

```
      START    = STARTTIME
      SUNSET   = SUNSET/14.4-AINT(SUNSET)/36.
      DAWN     = DAWN/14.4-AINT(DAWN)/36.
      LAMBDA   = SUNSET-DAWN
      ELAPSE   = STARTTIME
      TNXTVD   = STARTTIME
      CLOCK    = START
      JDY0     = STARTTIME
      HSTART   = (START-AINT(START)) * 24.

C     Unscramble references
C     *************************************************************************

      DO 31 L = 1,NS
      DO 31 M = 1,NS
         IF (IBC(L).EQ.ID(M)) IBC(L) = M
   31 CONTINUE

C     Construct cross-reference boundary conditions directory
C     *************************************************************************

      DO 32 L = 1,NBC
         IBCL     = IBC(L)
         IF (JBC(IBCL).GE.0) GO TO 33
         WRITE (OUT,34) ID(IBCL)
   34    FORMAT(' SEGMENT #',I3,' May not receive input boundary',
     *          ' CONDITIONS.')

         STOP

   33    JBC(IBCL) = L
   32 CONTINUE

C     Read in distance increments and lateral inflows from the
C     hydrodynamic model.  Convert english to metric units
C     *************************************************************************

      DO 47 I = 1,MNODE
         READ (HYD)  RLENGTH(I), QLC(I), RMILE(I)
         RLENGTH(I)  = RLENGTH(I)*0.3048
         QLC(I)      = QLC(I)*8026.85
   47 CONTINUE


      DO 9111 I=1,MNODE
         FLOWOLD(I) = 0.
         AREAOLD(I) = 0.
         WIDTHOLD(I)= 0.
         ELEVOLD(I) = 0.
 9111 CONTINUE

      WRITE(OUT,10)   TITLE
      WRITE(OUT,135) ALABEL
      WRITE(OUT,140) (I,(INIT(N,I),N=1,12),I=1,MNODE)
      WRITE(OUT,136) ALABEL
      WRITE(OUT,140) (I,(CLC(N,I),N=1,12),I=1,MNODE)


C     *************************************************************************
C               DEFINITION OF CONSTITUTENTS
C
C                 CONC(1)  = TEMPERATURE
```

```
C                      CONC(2)   = CBOD
C                      CONC(3)   = ORG-N
C                      CONC(4)   = NH3-N
C                      CONC(5)   = NO3-N
C                      CONC(6)   = ORG-P
C                      CONC(7)   = ORTHO-P
C                      CONC(8)   = DISSOLVED MN
C                      CONC(9)   = DISSOLVED FE
C                      CONC(10)  = DO
C                      CONC(11)  = COLIFORMS
C                      CONC(12)  = ALGAE
C     ***************************************************************************

C     Assign initial conditions CONC(N,I)=INIT(N,I) and test for presence
C     ***************************************************************************

      DO 51 N = 1,12
      DO 51 I = 1,MNODE
         CONC(N,I)   = INIT(N,I)
         IF(IBYPASS(N).EQ.0)CP(N)=.TRUE.
   51 CONTINUE
C
C     ALGAE and MACROPHYTES can function as sources of ORG-N, ORG-P,
C     and DO variable "ALGAEB" replaced by CONC(12,I)
C
C
C     The code segment down to "630 CONTINUE" assigns an output format to
C     each state variable
C     ***************************************************************************
C
      MM = 0
      DO 630 N = 1,12
         IF(.NOT.CP(N)) GO TO 630
         MM = MM + 1
         MC(MM) = N
         IF (N.NE.2.AND.N.NE.10) FMT(MM + 2) = FMT2
         IF (N.EQ.10.OR.N.EQ.2)  FMT(MM + 2) = FMT1
         BLABEL(MM) = ALABEL(N)
         IF (N .EQ. 1) BUNIT(MM) = AUNIT(1)
         IF (N .EQ.11) BUNIT(MM) = AUNIT(2)
  630 CONTINUE

      MAX = MM

C     C A L L     S U B R O U T I N E    'SPLINE'
C     ***************************************************************************

      DO 55 L = 1,NS
         M1 = NODE1(L)
         M2 = NNODE(L)
         CALL SPLINE(CONC(1,M1),DC(1,M1),CP,RLENGTH(M1),F,M2)
   55 CONTINUE


C
C    3)          T I M E   S T E P   C O M P U T A T I O N
C       Do time march while elapsed time is less than the endtime
C     ***************************************************************************

      JTIME = 1

      DO WHILE (ELAPSE.LE.ENDTIME)
```

```
C    A)  Read hydrodynamics, alternating between HYDRO1 and HYDRO2
C  ********************************************************************

        READ(HYD,END=9999,ERR=9999)DT

        DO 1021 I = 1, MNODE
            FLOWOLD(I)=FLOW(I)
            AREAOLD(I)=AREA(I)
            WIDTHOLD(I)=WIDTH(I)
            ELEVOLD(I)=ELEV(I)
            READ(HYD)FLOW(I),QLT(I),AREA(I),WIDTH(I),ELEV(I)
            RJUNK(I)=ELEV(I)
 1021   CONTINUE

        L1 = 0
        DT = DT/86400.
        M1 = L1 + 1
        M2 = L1 + MNODE

C  ********************************************************************
C        NOTE:  2446.6 m**3 = 1 cfs*day (a flow of 1cfs for 1 day)
C  ********************************************************************

        DO 113 I = 1,MNODE
            FLOW(I)   = FLOW(I)*2446.59
            AREA(I)   = AREA(I)*0.0929034
            WIDTH(I)  = WIDTH(I)*0.3048
            ELEV(I)   = ELEV(I)*0.3048
            RJUNK(I)  = RJUNK(I)*0.3048
            QLT(I)    = QLT(I)*8026.85
 113    CONTINUE

C    B)  Set  running parameters
C  ********************************************************************

        IF(ELAPSE.GT.TPR(IPRINT).AND.NPRINT.GT.1)THEN
            IPRINT=IPRINT+1
            TPRINT=TPRNT(IPRINT)
        END IF

        IF(ELAPSE.GT.TPL(IPLOT).AND.NPLOT.GT.1)THEN
            IPLOT=IPLOT+1
            TPLOT=TPLT(IPRINT)
        END IF

        IF (JTIME.EQ.1) DT = 0.
        ELAPSE = ELAPSE + DT
        CLOCK  = AMOD(CLOCK + DT,1.)
        DARK   = CLOCK.GE.SUNSET.OR.CLOCK.LE.DAWN
        IF (DARK) SINI = 0.
        IF (.NOT.DARK) SINI=SIN(PI*(CLOCK-DAWN)/LAMBDA)
        DELTAT = DT * 24.

        IF (JTIME .EQ. 1) STB = HSTART
        STE = STB + DELTAT

        IF (JTIME .EQ. 1) THEN
            TOFDAY = HSTART
        ELSE
            TOFDAY = TOFDAY  + DELTAT
        END IF
```

```
C     C)  Update boundaries and date
C     ********************************************************************

          CALL GREGORIAN_DATE(ELAPSE,SYEAR,MONTH,IDAY,HOUR)
          IF (ELAPSE.GE.TNXTVD)CALL TIME_VARYING_DATA (ELAPSE,TNXTVD,
     1                                                 NBC,BOUND,NS)
          TEST_PLOT = TEST_PLOT + DT*24. +0.00001
          TEST_PRINT = TEST_PRINT + DT*24. +0.00001

C     D)  Perform segment calculations
C     ********************************************************************
C         NOTE: "DO 505" loop calls SUBROUTINE SEG and processes
C               segments in a downstream direction. This loop is
C               executed during each time step.
C     ********************************************************************
          DO 505 LL = 1,NS
              LMN     = LL
              L       = ORDER(NS-LL + 1)
              IBRANCH = L
              ITOSTAT = ITO(L)
              IT1STAT = IT1(L)
              IT2STAT = IT2(L)
              M       = NODE1(L)
              M1      = L1 + M
              TEMP    = TDUM(L)

C
C
C         C A L L    S U B R O U T I N E    'SEG'
C     ********************************************************************
              IF (JBC(L)) 600,601,602
  602         JJ = L

              QDUM = QWIND(L)
              CALL SEG(RLENGTH(M),FLOWOLD(M),AREAOLD(M),WIDTHOLD(M),
     1                 ELEVOLD(M),FLOW(M),AREA(M),WIDTH(M),ELEV(M),
     2                 QLC(M),QLT(M),CONC(1,M),DC(1,M),K(1,M),SINK(1,M),
     3                 BOUND(1,JJ),CP,CLC(1,M),CLT(1,M),SINI,NNODE(L),
     4                 ITO(L),IT1(L),IT2(L),JT,T,DISP(M),F,NS,QDUM)

              GO TO 505

  601         WRITE(OUT,603) ID(L)
  603         FORMAT(' MISSING BOUNDARY CONDITION FOR SEGMENT',I3,
     *               '.  RUN ABORTED.')

              STOP

C         1) Pass boundary conditions through control structures in
C            other cases
C     ********************************************************************

  600         JJ = -JBC(L)

C         2) Reaeration over dams.  This section has been modified
C            greatly to accommodate EDM reaeration.
C     ********************************************************************

              IF (CP(1)) TEMP = CONC(1,M)
              DOSAVE = CONC(10,JJ)
              IF (IDAM(L)-1) 507,508,508
```

```
C          3)  EDM reaeration below
C  ***************************************************************

     508       CONTINUE
               CTR    = DAMK(L) * 1.022 ** (TEMP-20.)
               DELTAH = ABS( ELev( L1 + JJ) - ELev(M1))
               R      = EXP (-CTR * DELTAH)
               DOSAT  = 14.652+(-0.41022+(0.007991-0.000077774*TEMP)*TEMP)
     1                  *TEMP
               CONC(10,JJ) = DOSAT*(1.-R) + DOSAVE*R

     507       QDUM = QWIND(L)

C
C             C A L L   S U B R O U T I N E   'SEG'
C  ***************************************************************

               CALL SEG(RLENGTH(M),FLOWOLD(M),AREAOLD(M),WIDTHOLD(M),
     1                  ELEVOLD(M),FLOW(M),AREA(M),WIDTH(M),ELEV(M),
     2                  QLC(M),QLT(M),CONC(1,M),DC(1,M),K(1,M),SINK(1,M),
     3                  CONC(1,JJ),CP,CLC(1,M),CLT(1,M),SINI,NNODE(L),
     4                  ITO(L),IT1(L),IT2(L),JT,T,DISP(M),F,NS,QDUM)

               CONC(10,JJ) = DOSAVE

     505      CONTINUE


C        "DO 505" LOOP COMPLETE - CALLS TO SEG Complete for this time step
C  ***************************************************************


C    E)  Write out updated information
C  ***************************************************************

C
C          Write out segment data to graphics files
C  ***************************************************************
C
               IF(TEST_PLOT.GE.TPLOT.OR.JTIME.EQ.1) THEN
                  IF(JTIME.EQ.1)WRITE(GRAPH) (RMILE(IN), IN=1,MNODE)
                  CALL RIVDU(JTIME)
                  WRITE(GRAPH) ELAPSE
                  WRITE(GRAPH) ((CONC( NN, IN) , NN=1,12), IN=1,MNODE)
                  TEST_PLOT = 0.
               END IF



C          Write out segment data to output file
C  ***************************************************************

               IF(TEST_PRINT.GE.TPRINT.OR.JTIME.EQ.1) THEN
                  WRITE(OUT,190)
                  WRITE(OUT,200) TITLE
                  WRITE(OUT,211) CLOUD0,WIND0,DRY0,WET0,ATM0
               DO 500 L = 1, NS
                  M1 = NODE1(L)-1
                  M2 = NNODE(L)
                  WRITE(OUT,210) SYEAR,MONTH,IDAY,HOUR,ID(L),
     1                           (SNAME(JJ,L),JJ=1,10)
                  WRITE(OUT,260) (BLABEL(MM),MM=1,MAX)
```

B12

```
                  WRITE(OUT,270) (BUNIT(MM),MM=1,MAX)
                  WRITE(OUT,FMT) (I,RMILE(I),(CONC(MC(MM),I),MM=1,MAX),
         1                        I=M1+1,M1+M2)
    500           CONTINUE
                  TEST_PRINT = 0.
             END IF

             STB = STE
             IF (STB .GE. 24.   ) STB = STB-24.
             JDYO = JDYO
             IF (TOFDAY .GE. 24.) JDYO = JDYO + 1
             IF (TOFDAY .GE. 24.) TOFDAY = TOFDAY-24.
             IF (JDYO .GT. 365  ) JDYO = JDYO-365

             JTIME = JTIME + 1
          END DO
C ***************************************************************************
C                        END OF TIME LOOP
C     THE END DO COMPLETES THE MAIN TIME MARCH, CONTINUE UNTIL
C     THE ELAPSED TIME IS <= THE SPECIFIED END TIME
C
C ***************************************************************************
 9999 CONTINUE
      STOP
C
C       Output format section
C ***************************************************************************
   10 FORMAT(A80)
  135 FORMAT(5x,'Initial Conditions',/,5x,18('='),//,5X,12(A8),/)
  136 FORMAT(//,5x,'Lateral Inflow Concentrations'/,5X,29('='),//,5x,
     .12(A8),/)
  140 FORMAT(I3,12F8.2)
  190 FORMAT(1H1,//,1x,'CE-QUAL-RIV1 WATER QUALITY MODEL ')
  200 FORMAT(/,5x,A80)
  210 FORMAT(/,'YEAR ',I4,' MONTH ',A9,'  DAY  ',I2,
     *   '  HOUR ',F8.2,' SEGMENT',I3,' ',10A4,
     *       T61,':')
  211 FORMAT(' CLOUD COVER ', F8.3,' WIND SPEED= ',F8.2,' DRY BULB ',
     .  F8.2,' WET BULB TEMP = ', F8.2, ' ATMOSPHERIC PRESSURE = ',F8.2)
  260 FORMAT(/,T7,'RIVER',6X,11(A8,4X),A8)
  270 FORMAT(' ',T8,'MILE',6X,11(A8,4X),A8)


C
C     FORMAT STATEMENTS for global constants
C ***************************************************************************

 1190 FORMAT(13x,'*********************************************')
 1191 FORMAT(13x,'*          C E - Q U A L - R I V 1          *')
 1192 FORMAT(13x,'*       W A T E R  Q U A L I T Y  M O D E L  *')
 1200 FORMAT(13x,'*********************************************',/)
 1193 format(5x,70('='))
 1201 FORMAT('   Start Time For Simulation,(hh.mm), Start   = ',F5.2,
     ./,   '    End Time For The Simulation (HH.MM), End   = ',F5.2,
     ./,   '    Lateral Inflow Option ',
     ./,   '    (0= Constant, 1 = Time Variable)           = ',I5)

 1202 FORMAT('   Temperature Coeff. For Biological Processes = ',F8.4,
     ./,   '    Temperature Coeff. For NH3 Reactivity       = ',F8.4,
     ./,   '    Temperature Coeff. For Physical Processes   = ',F8.4,
     ./)

 1203 FORMAT('   Algal Phosp. Content (Fraction By Weight)   = ',F8.4,
```

B13

```
      ./,        '   Algal Nitrogen Content (Fraction By Weight)    = ',F8.4,
      ./,        '   Temperature Coeff. For Physical Processes      = ',F8.4,
      ./)

 1204 FORMAT('   Oxygen/Nitrogen Equivalence (gO2/gN)            = ',F8.4,
      ./,        '   Oxygen/Nitrogen Ratio For Nitrification',
      ./,        '   (gO2/gN)                                       = ',F8.4,
      ./,        '   Oxygen Consumption/Plant Decay (gO2/g Plant)   = ',F8.4,
      ./,        '   Oxygen Consumption/Iron Oxidized (gO2/g Fe)    = ',F8.4,
      ./,        '   Oxygen Consumption/Manganese Oxidized'
      ./,   '  (gO2/g Mn)                                      = ',F8.4,/)

 1205 FORMAT('   Time of Sunrise, (HH.MM), Dawn                  = ',F5.2,
      ./,        '   Time of Sunset, (HH.MM), Sunset                = ',F5.2,
      ./)

 1206 FORMAT(///,21x,'Print Intervals',/,21x,15('='))
 1207 FORMAT(///,22x,'Plot Intervals',/,22x,14('='))
 1208 FORMAT('   Interval (Hours)',f10.4,' from Time (Days) = ',F10.4)

 1209 format(///,21x,'System Bypass Options',/,21x,20('='),//,
      .18x,' (0 = Simulate, 1 = Bypass) ',//,'  Temp  BOD  OrgN NH3   NO3
      . OrgP  PO4   Mn    Fe    DO   CLFM Algae',/,12(2x,I3),////)


 1210 FORMAT(30('+'),' END GLOBAL CONSTANTS ',30('+'),//)


C      FORMAT STATEMENTS for segment parameters
C ****************************************************************************
 1320 FORMAT(31x,'Segment Parameters')
 1321 FORMAT(//,10x,'Constant Data for Segment No. ',I2,//,10x,
      . 'NOTE:',' These Data may be Corrected for Temp., D.O., etc. Durin
      .g',/,16x,'Program Execution'//)

 1324 FORMAT('   Stream Segment Temperature, Deg. C,',23x,'TEMP = ',F5.2,
      ./,        '   Rate Coefficient for Bottom Heat Exchange (1/DAY)',
      .9x,' ATB = ',F5.2,
      ./,        '   Rate Coefficient for Surface Heat Exchange, W/M**2 DEG.
      .C,',' ATS = ',F5.2,
      ./,        '   Source/Sink Term for Bottom Heat Exchange, DEG. C,',
      .7x,'TSINK = ',F5.2,
      ./,        '   Variable Designating type of Heat Exchange Solution,'
      .,5x,' ITEM = ',I2,
      ./,' 1--Constant Temp. Equilibrium  0--Full Heat Balance ',///)

 1325 FORMAT('   Nitrate Conc. at Which Denitrification Rate is 1/2',
      ./,        '   Maximum, mg/l,',43x,'NCBDN = ',F5.2,
      ./,        '   D.O. Conc. at Which Algal Decay Rate is 1/2 Maximum',
      ./,        '   mg/l,',51x,'KOALDK = ',F5.2,
      ./,        '   D.O. Conc. at Which the Rate of Denitrification is',
      ./,        '   Reduced by 1/2, MG/L,',35x,'KOCBDN =',F5.2,
      ./,        '   D.O. Conc. at Which CBOD Decay rate is 1/2 Maximum',
      ./,        '   Rate, mg/l',47x,'KOCB1 = ',F5.2,
      ./,        '   D.O. Conc. at Which Nitrification Rate is 1/2',
      ./,        '   Maximum, mg/l',46x,'KON = ',F5.2,///)

 1328 FORMAT('   Rate Coefficient for Stream Reaeration Rate',17x,
      .'AG = ',F5.2,/,
      .'   Exponent of Velocity in Stream Reaeration Rate,',
      .'  Formulation',1x,'E1 = ',F5.2,/,
      .'   Depth Exponent in Stream Reaeration Formulation,',
```

```
      .'              E2 = ',F5.2,/,
      .'    Coefficient in the Tsivoglou-Wallace Reaeration',
      .' Equation',/,'    1/ft,',53x,'TSIV = ',F5.2,/,
      .'    Specific Rate Coefficient, Uncorrected, for',
      ./'  Denitrification 1/day',37x,' ADN = ',F5.2,/,
      .'    Specific Rate Coefficient, Uncorrected, for',/,
      .'  Nitrification 1/day',39x,' AKN = ',F5.2,/,
      .'    Specific Rate Coefficient, Uncorrected, for',/,
      .'  Ammonia Adsorption 1/day',34x,'AKNX = ',F5.2,/,
      .'    Specific Rate Coefficient, Uncorrected, for',/,
      .'  Phosphate Adsorption, 1/day',31x,'APO4 = ',F5.2,///)

 1330 FORMAT('  Specific Rate Coefficient For Iron Oxidation, 1/day,',
      .4x,' KFEDK = ',F5.2,/,
      .'  Rate Coefficient For Coliform Mortality, 1/day,',
      .7x,' KCOLIDK = ',F5.2,/,
      .'    Sediment Oxygen Demand, gr./m**2',27x,'SOD = ',F5.2,/,
      .'    Total Nitrogen Conc. at Which Algal Growth',/,
      .'    Rate is Reduced by 1/2, mg/1',28x,'KNPOOL = ',F5.2,/,
      .'    Phosphorous Conc. at which Algal Growth Rate Reduced by',/,
      .'    1/2, mg/1',48x,'KPO4X = ',F5.2,/
      .'    Sediment Denitrification Rate',28x,'KDSED = ', F5.2,/)

 1331 FORMAT('  Macrophyte Density on Channel Surfaces, g/m**2,',
      .'            MACROB = ',F5.2,/,
      .'    Specific Macrophyte Growth Rate, 1/day',18X,'MACGRO = ',F5.2,/,
      .'    Specific Macrophyte Decay Rate, 1/day',19X,'MACDKY = ',F5.2,//,
      .'    Algal Growth Rate, mg/1*day',29X,'KALGRO = ',F5.2,/,
      .'    Algal Decay Rate, mg/1*day',30X,'KALGDK = ',F5.2,//,
      .'    D.O. Conc. Below Which Oxidation of',/,
      .'    Iron and Manganese Do Not Occur, mg/1',19X,'OXIDAT = ',F5.2,/,
      .'    Specific Rate Coefficient Manganese Oxidation, 1/day, ',
      .'    KMNDK = ',F5.2,/)

 1332 FORMAT('  Empirical Coefficient Reflecting Thickness of',
      .'Boundary Layer BK = ',F5.2,/,
      .'    Rate Coefficient Carbonaceous Oxygen Demand, 1/day',9X,
      .'AK1 = ',F5.2,/,
      .'    Light Intensity at Which Photosynthesis Rate Reduced by'/,
      .'    1/2,',52X,' KLITE = ',F5.2,/
      .'    Rate Coefficient for Org-N Decay To NH3',18X,'  ACK = ',F5.2)
c#
c#    Problem with QWIND (xx)
c#
 1333 FORMAT('  Surface Light Intensity at Local Noon,
      .HNEFSW = ',F8.2,/,
      .'  Is Wind Driven Reaeration Used for Segment ?,',
      .' T or F,','QWIND(',I2,') = ',L2,/)

 1334 FORMAT('  Non-algal Portion of Light Extinction Coefficient',5x,
      .' LAMBDA0 = ',F5.2,/,
      .'    Linear Algal Self Shading Coefficient',18x,'LAMBDA1 = ',F5.2,/,
      .'    Nonlinear Algal Self Shading Coefficient',15x,'LAMBDA2 = ',
      .F5.2,/,
      .'    Conversion Factor From Algae To Chlorophyll,',11x,
      .' ALPHA0 = ',F5.2,//)

 1335 FORMAT('  Settling Rate for Organic Nitrogen,',22x,' KNSET = ',
      .1F5.2,/,
      .'    Benthal Source Rate for Ammonia, g/m**2',19x,'ABSR = ',F5.2,/,
      .'    Settling Rate for CBOD, m/day',27x,'CBODSR = ',F5.2,/,
      .'    Fraction of Algal and Macrophyte Decay',/,
```

```
           .' Which Goes to CBOD, ',37x,'FCBOD = ',F5.2//)

    1336 FORMAT(' Decay Rate For Organic - P,',30x,' KPDK = ', F5.2,/,
           .' Settling Rate Coefficient For Org-P,',20x,' KPSET = ',F5.2,/,
           .' Benthic Source Term for Dissolved P,',19x,' BENPO4 = ',F5.2,//)

           STOP
           END

C *****************************************************************************
C                     S U B R O U T I N E   S E G
C *****************************************************************************

           SUBROUTINE SEG(DX1,QO,AO,BO,ELO,Q,A,B,EL,QLC,QLT,C,DC,K,SINK,BC,
          .               CP,CLC,CLT,SINI,MNODE,ITO,IT1,IT2,JT,T,DISP,F,NS,
          .               QWIND)

           INCLUDE 'RIV1Q.CMN'

           DIMENSION DX1(MNODE),QO(MNODE),AO(MNODE),BO(MNODE),ELO(MNODE),
          *          Q(MNODE),A(MNODE),B(MNODE),EL(MNODE),QLC(MNODE),
          .          QLT(MNODE),C(12,MNODE),DC(12,MNODE),K(12,MNODE),
          .          SINK(12,MNODE),BC(12),CP(12),JT(NS),T(50,1),
          .          CLC(12,MNODE),CLT(12,MNODE),DISP(MNODE),F(MNODE,6)
C *****************************************************************************

           LOGICAL      JUNCT
           DATA         THETA/0.55/

           RTHETA = (THETA - 1.)/THETA
           L    = IT2 + 1
           IT   = MAXO(1,IT1)
           IF (DT.EQ.0.) GO TO 17
           ULL = QO(MNODE)/AO(MNODE)

C       Calculate the derivative at the first node
C *****************************************************************************

           DO   = DT/2.*(QO(1)/AO(1) + Q(1)/A(1))
           DEN  = 1.5*DX1(1) + 2.*DO
           B1   = 1.5*DX1(1)/(DO*DEN)
           B2   = 3.*DO/(DX1(1)*DEN)
           B3   = DO/DEN

C       Relfect boundary conditions at node 1 when there is no inflow
C       at the upstream boundary. (1 cms or 86400 cubic meters/day is
C       considered no flow)
C *****************************************************************************

           DO 63 N = 1,12
              IF(CP(N)) THEN
                 IF(Q(1).LT.86400.) THEN
                    BC(N)=C(N,2)
                 END IF
                 DC(N,1) = B1*(C(N,1)-BC(N)) + B2*(C(N,2)-C(N,1))-B3*DC(N,2)
              END IF
        63 CONTINUE
           GO TO 14
C
C       Place initial concentrations into tributary data transfer
C       array (t-array)
C *****************************************************************************

                                   B16
```

```
   17 IF (IT1.EQ.0) GO TO 14

      DO 15 LL = IT1,IT2
         I = JT(LL) - 1

         DO 15 N  = 1,12
            IF (.NOT. CP(N) ) GO TO 15
            T(N + 26,LL) = C(N,I)
            T(N + 38,LL) = DC(N,I)
   15 CONTINUE
C
C     Initialize nutrient depletion parameters
C **********************************************************************
   14 CONTINUE

C .................. N O D E    C O M P U T A T I O N ..................
C
C     "DO 470" - NODE MARCH.  Certain program blocks are bypassed for
C     DT=0 (INITIAL CONDITIONS) and I=1 (BOUNDARY CONDITIONS).
C     loop marches through nodes in an upstream direction and
C     cycles thru every node in the system once during each time step.
C **********************************************************************
C
      DO 470 II = 1,MNODE
         LII = II
         I   = MNODE-II + 1
         UUR = Q(I)/A(I)
         UUL = UUR
         IF(I.NE.1) UUL = Q(I-1)/A(I-1)
C
C     Check to see if we have an initial or boundary condition
C **********************************************************************

         IF (DT.EQ.0..OR.I.EQ.1) GO TO 150

C     Check to see if node is at a junction, if not GO TO STATEMENT 20
C **********************************************************************

         IF (L.LE.IT) GO TO 20
         IF (JT(L-1).NE.I) GO TO 20
C
C     Calculate dilution ratios for tributary junctions
C **********************************************************************
C
         L = L-1
         JUNCT = .TRUE.
         D = T(25,L)/Q(I + 1)
         DD = T(26,L)*D/UUR
         GO TO 13
   20    JUNCT = .FALSE.
   13    ULL   = Q0(I-1)/A0(I-1)

         DX  = DX1(I-1)
         DUDX= (ULR-ULL)/DX
         DADX= (A0(I) - A0(I-1))/DX
         DADX2=(A(I) - A(I-1))/DX
         DDISP= (DISP(I) - DISP(I-1))/DX
         DDA1 = DISP(I)/A0(I)*DADX + DDISP
         DDA2 = DISP(I)/A(I)*DADX2 + DDISP
C **********************************************************************
C         NOTE: Above will not work for homogeneous channel or
C         and uniform dispersion.  Also, RDDA not used so commented
```

B17

```
C           out. "US" is the characteristic velocity of the constituent
C           concentration and "E" is the resulting courant number.
C           These parameters are computed once for each node during each
C           time step.
C     *********************************************************************

            US   = (UUR + ULR-DDA1-DDA2)/(2. + DT*DUDX)
            E    = US*DT/DX
            EC   = 1.-E
            AS   = (A(I) + A0(I-1)*E + A0(I)*EC)/2.
            QLA  = (QLC(I)*(1. + EC) + QLC(I-1)*E)/(2.*AS)
            DQLA= ((QLC(I)-QLC(I-1))/DX-QLA * DADX)/AS
            QLB  = (QLT(I)*(1. + EC) + QLT(I-1)*E)/(2.*AS)
            DQLB= ((QLT(I)-QLT(I-1))/DX-QLB * DADX)/AS
            GO   = 1.-DT*DUDX

C     "A1" thru "A4" are the coefficients for the polynomial  estimator
C        of the concentration at node I and time step J
C     *********************************************************************

            A1   = E*E*(3.-2.*E)
            A2   = 1.-A1
            A3   = E*E*EC*DX
            A4   = -E*EC*EC*DX

C     "US" is redefined as characteristic velocity of the spatial
C     derivative of the constituent concentration and "ED" is the
C     resulting Courant number.  They are computed once for  each node
C     during each time step.
C     *********************************************************************

            US   = US - DDISP
            ED   = US * DT / DX
            EDC = 1. - ED

C     "B1 thru B4" are the coefficients of the polynomial estimator
C        of the spatial derivative at node I and time step J.
C     *********************************************************************

            B1   = 6.*ED*(-EDC)/DX
            B2   = -B1
            B3   = ED*(ED-EDC-EDC)
            B4   = EDC*(EDC-ED-ED)

C     Correct light extinction coefficient, kext, for algal  self-shading
C     ref: qual2e manual affects both algae and macrophytes
C     *********************************************************************

            KEXT = LAMBDA0(IBRANCH) + LAMBDA1(IBRANCH)*ALPHA0(IBRANCH)*
          *          C(12,I) + LAMBDA2(IBRANCH)*(ALPHA0(IBRANCH)*C(12,I))
          *          **2/3

C
C     Average value of D.O., NO3, NPOOL, and PO4
C     *********************************************************************
C
            DOX  = C(10,I-1)*E + C(10,I)*EC
            IF (DOX.LT.0.0) DOX = 0.0
            NO3NX = C(5,I-1)*E + C(5,I)*EC

C     NPOOL is the sum of NH3 and NO3
C     *********************************************************************
```

B18

```
          NPOOL = C(4,I-1)*E + C(4,I)*EC + NO3NX

C      Ortho-P only
C ****************************************************************************

          PO4X  = C(7,I-1)*E + C(7,I)*EC

C      Compute nutrient limitation factors to be used in computing
C      algal growth rate for non-boundary and non-initial conditions

C      Nitrogen limitation factor in algae growth
C ****************************************************************************

          FN  = NPOOL/(NPOOL + KNPOOL(IBRANCH))

C      PO4 limitation factor in algae growth
C ****************************************************************************

          FP  = PO4X/(PO4X + KPO4X(IBRANCH))

          GO TO 140

C      Value of D.O., NO3, NPOOL, and PO4 used for initial and
C      boundary conditions
C ****************************************************************************

  150     DOX  = C(10,I)
          NO3NX= C(5,I)

C      NPOOL is sum of NH3 + NO3
C ****************************************************************************

          NPOOL= NO3NX + C(4,I)
          PO4X = C(7,I)
C
C    Correct light extinction coefficient, KEXT, for algal self-shading
C    affects both algae and macrophytes
C ****************************************************************************

          KEXT = LAMBDA0(IBRANCH) + LAMBDA1(IBRANCH)*ALPHA0(IBRANCH)*
      *          C(12,I) + LAMBDA2(IBRANCH)*(ALPHA0(IBRANCH)*C(12,I))
      *          **2/3

C      Nitrogen limitation factor in algae growth
C ****************************************************************************

          FN  = NPOOL/(NPOOL + KNPOOL(IBRANCH))

C      PO4 limitation factor in algae growth
C ****************************************************************************

          FP  = PO4X/(PO4X + KPO4X(IBRANCH))
C
C      Compute hydraulic depth for current node
C ****************************************************************************

  140     H  = A(I)/B(I)

C      Statements below shut down algal growth in darkness, SINI.LE.0.0
C      or lack of nutrients, (PO4X OR NPOOL .LE. 0.0)
C ****************************************************************************
```

```
            IF (SINI.LE.0.) ALGRO = 0.0

            IF (PO4X.GT.0.0) GO TO 310
            ALGRO = 0.0

     310    IF (NPOOL.GT.0.0) GO TO 315

            ALGRO = 0.0

C     Temp set equal to C(1,I) for use in temperature correction
C     equations below
C     **************************************************************************

     315    IF (.NOT.CP(1))GO TO 319

            TEMP = C(1,I)

C     The following three statements are temperature corrections for
C     various types of reactions
C     **************************************************************************

     319    TBIO  = TBIOS**(TEMP-20.)
            TNH3  = TAMMON**(TEMP-20.)
            TPHYS = TPHYSI**(TEMP-20.)

C     DOSAT is the saturation value of D.O. as a function of water temp
C     **************************************************************************

            DOSAT=14.652+(-0.41022+(0.007991-0.000077774*TEMP)*TEMP)*TEMP

C     Algae are light limited only in this version
C     **************************************************************************

            IF (SINI.LE.0.) GO TO 330

C         If you want to include phosphate and nitrogen limitation to
C         algal growth, comment out the above statement and uncomment
C         the one below
C     **************************************************************************
C
C 320    IF (SINI.LE.0..OR.PO4X.LE.0.0.OR.NPOOL.LE.0.0) GO TO 330

C     **************************************************************************
C     Light limitation under equilibrium temperature approach.
C     NOTE:  DAWN, SUNSET, HNEFSW must be specified in constant list
C     for this case.
C     SWALG is short wave light intensity at surface.
C     **************************************************************************

            IF(ITEM(IBRANCH).EQ.1)THEN
                SWALG= HNEFSW(IBRANCH)*SINI
            ELSE
                SWALG= HNEFSW(IBRANCH)
            END IF

C     Algal growth rate (ALGRO) defined in terms of algal concentration
C     computed in the previous time step, (C(11,I), corrected for light and
C     nutrient availability
C     **************************************************************************

            ALGRO = C(12,I)*KALGRO(IBRANCH)*(1/(KEXT*H))*
          .         LOG((KLITE(IBRANCH)+SWALG)/(KLITE(IBRANCH)+SWALG*
```

```
                    EXP(-KEXT*H)))*FN*FP

C      Macrophyte growth rate, MGRATE, defined in terms of reach
C      dependent parameter, MACO, corrected for light but not nutrients,
C      see manual
C ********************************************************************

            IF(KEXT.GE.0.01)MGRATE = MACROB(IBRANCH)*MACGRO(IBRANCH)
       .                             *SWALG*(2./KEXT + (B(I)-2./KEXT)*
       .                             EXP(-KEXT*H))/A(I)

            IF(KEXT.LT.0.01)MGRATE = MACROB(IBRANCH)*MACGRO(IBRANCH)*
       .                             SWALG*(H*(2.-KEXT*H) + B(I)*
       .                             EXP(-KEXT*H))/A(I)

C      Define P as the fraction of NO3 in npool (NPOOL = NH3 + NO3)
C ********************************************************************

  330       IF (NPOOL .GT. 0.) P = NO3NX/NPOOL

C      When D.O. drops to zero algal decay, MAC. DECAY, aerobic
C      hydrolysis of ORG-N, and nitrification cease.  Therefore ALGADK,
C      MDEATH, K1, K1N, and KN set to zero at 380.
C ********************************************************************

            IF (DOX.LE.1.E-06) GO TO 380

C      If D.O. not zero go ahead and compute algal death rate (ALGADK),
C      K1, and KN, and reach dependent macrophyte death rate (MDEATH).
C      Correct algal death rate (ALGADK) for D.O. availability
C ********************************************************************

            ALGADK = C(12,I)*KALGDK(IBRANCH)*(DOX/(DOX+KOALDK(IBRANCH)))

C      Correct macrophyte death rate (MDEATH) for D.O. availability
C ********************************************************************

            MDEATH = MACROB(IBRANCH)*MACDKY(IBRANCH)*(B(I) + H + H)/((1. +
       .             KOALDK(IBRANCH)/DOX)*A(I))

C      Correct rate coefficient for CBOD decay (AK1) for temperature and
C      D.O. availability
C ********************************************************************

            K1 = AK1(IBRANCH)*TBIO/(1.+ KOCB1(IBRANCH)/DOX)

C      Correct rate coefficient for ORG-N (ACK) for temperature and D.O.
C      availability
C ********************************************************************

            K1N = ACK(IBRANCH)*TBIO/(1+ KOCB1(IBRANCH)/DOX)

C      Correct rate coefficient for nitrification (AKN), temperature, and
C      D.O. Availability
C ********************************************************************

            KN = AKN(IBRANCH)*TNH3/(1.+ KON(IBRANCH)/DOX)

C      Correct sediment oxygen demand for temperature and channel depth
C ********************************************************************

            KSOD     = (SOD(IBRANCH)*TBIO)/H
```

B21

```
          KBENPO4  = (BENPO4(IBRANCH)*TBIO)/H

C     Reaction constants for iron and manganese not corrected for
C     temperature see RIV1-Q documentation
C     ************************************************************************

          KMNX   = KMNDK(IBRANCH)
          KFEX   = KFEDK(IBRANCH)

          GO TO 390

C     If D.O. is zero, set ALGADK, MDEATH, K1, K1N, and KN to zero.
C     ************************************************************************

   380    ALGADK = 0.0
          MDEATH = 0.0
          K1     = 0.0
          K1N    = 0.0
          KN     = 0.0

          IF(NO3NX.LT.1.0E-06.AND.KNCBDN(IBRANCH).LE.1.0E-06)THEN
            WRITE(OUT,389)
   389      FORMAT(1X,'ERROR: NO3X & KNCBDN BOTH 0.0, KDN IS UNDEFINED')
            STOP
          ENDIF

C     Correct rate coefficient for denitrification (ADN and KDSED),
C     TEMP, D.O. inhibition, and Nitrate availability
C     ************************************************************************

   390    KDN = ADN(IBRANCH)*TBIO*KOCBDN(IBRANCH)/(DOX +
         .        KOCBDN(IBRANCH))*NO3NX/(NO3NX + KNCBDN(IBRANCH))
          KDSED(IBRANCH)= KDSED(IBRANCH)*TBIO*KOCBDN(IBRANCH)/(DOX +
         .                KOCBDN(IBRANCH))*NO3NX/(NO3NX + KNCBDN(IBRANCH))

C     Correct rate coefficient for NH3 adsorption for temperature
C     ************************************************************************

          KNX = AKNX(IBRANCH)*TPHYS

C     When the D.O., (DOX), level drops below a specified value (OXIDAT)
C     iron and manganese oxidation stops, KMNX and KFEX set to 0.0.
C     ************************************************************************

          IF (DOX .LT. OXIDAT(IBRANCH)) KMNX = 0.
          IF (DOX .LT. OXIDAT(IBRANCH)) KFEX = 0.
C
C     This code segment (DOWN TO 391) computes reaeration coefficients
C     ************************************************************************

          AGC = AG(IBRANCH)*3.79726E-05**E1(IBRANCH)*0.3048**E2(IBRANCH)
          K2 = AGC*ABS(UUR)**E1(ibrANCH)/H**E2(IBRANCH)*TPHYS

          IF (I.EQ.1 .OR. DT.EQ.0.) GO TO 391

C     Compute Tsivoglou reaeration coefficient, K2, if TSIV .gt. 0.
C     otherwise use exponential reaeration coefficient, K2, computed
C     above
C     ************************************************************************

          IF (TSIV(IBRANCH).GT.0.) THEN
            DELTAH = EL(I) - EL(I-1)
```

```
             K2 = TSIV(IBRANCH)/0.3049 * TPHYS * ABS(DELTAH)*0.5*
                  ABS( UUR + UUL)/DX
           END IF
     391   CONTINUE
C
C   ************************************************************************
C                      End reaeration routines
C
C      Correct PO4 adsorption rate coefficient for temperature
C   ************************************************************************
C
           KPO4DK = APO4(IBRANCH)*TPHYS
C
C      Use equilibrium temperature approach if ITEM=1 (in constant list)
C      conversion factor changes units from w/m2-c to per day (0.02064=
C      86400/(4184*1000)
C   ************************************************************************
C
           IF(ITEM(IBRANCH).EQ.1)THEN
             KTS=ATS(IBRANCH)*0.02064/H
             KTB=ATB(IBRANCH)*TPHYS
           END IF

C
C      "DO 460" Loop computes decay constants as well as kinetic constants
C      for sources or sinks for a given node.  Constituent concentrations
C      corrected for decay, sources or sinks are then computed
C   ************************************************************************

           DO 460 N = 1,12

             IF (.NOT.CP(N)) GO TO 460

             IF (DT.EQ.0..OR.I.EQ.1) GO TO 450

             KS    = K(N,I-1)*E + K(N,I)*EC
             DK    = (K(N,I)-K(N,I-1))/DX
             SINKS = SINK(N,I-1)*E + SINK(N,I)*EC
             DSINK = (SINK(N,I)-SINK(N,I-1))/DX

     450     GO TO (1,2,3,4,5,6,7,8,9,10,11,12),N
C
C   ************************************************************************
C                          TEMPERATURE
C
C      Decay rate and source/sink term for temperature, K(1,I),
C      initially set to zero
C   ************************************************************************

     1       K(1,I)    = 0.
             SINK(1,I) = 0.
C
C      Equilibrium temperature simulation used if ITEM=1
C      Full heat balance simulation used if ITEM=0
C      If ITEM not equal to 0 or 1 heat exchange does not occur
C   ************************************************************************

             IF (ITEM(IBRANCH).NE.1 .AND.ITEM(IBRANCH).NE.0)GO TO 480

C      Equilibrium approach for computing temperature sink
C   ************************************************************************
```

B23

```
              IF(ITEM(IBRANCH).EQ.1)THEN
                 K(1,I)    =   KTS + KTB
                 SINK(1,I)=   KTS * TEMP + KTB * TSINK(IBRANCH)
                 GO TO 480
              END IF
C
              DEPTHO = H * 3.281
              ELEVO =EL(I)*3.281
C
C     Full heat balance approach (ITEM=0; default value) requires calling
C     SUBROUTINES HEAT and HTFLUX
C ********************************************************************************
              IF(ITEM(IBRANCH).EQ.0.)THEN
                 CALL HEAT
                 IF (JTIME .NE. 1) THEN
                    K(1,I) = KTB
                    SINK(1,I) = DTEM/DT + KTB * TSINK(IBRANCH)
                 END IF
                 GO TO 480
              END IF
C
C ********************************************************************************
C               End of Temperature Computation Segment
C
C                              CBOD
C ********************************************************************************
C
C     K(2,I) accounts for CBOD exertion, denitrification and CBOD
C     settling SINK(2,I) accounts for ALGAE and MACS. Converted to CBOD
C     upon death
C ********************************************************************************
C
      2          K(2,I) = K1 + KDN + CBODSR(IBRANCH)
                 SINK(2,I) = OPDECY * ((FCBOD(IBRANCH) * DOX +
                 .            KOCB1(IBRANCH))/(DOX + KOCB1(IBRANCH)))*
                 .            (ALGADK+MDEATH)

                 GO TO 480
C
C ********************************************************************************
C                        ORGANIC NITROGEN
C ********************************************************************************
C
C     K(3,I) accounts for decay of ORG-N to NH3 (K1N) and loss from
C     denitrification (KDN)
C     SINK(3,I) accounts for Organic-N produced by algal and macrophyte
C     decay model assumes ALGAE and macs initially produce 100% ORG-N
C ********************************************************************************
C
      3          K(3,I) = K1N + KDN + KNSET(IBRANCH)/H
                 SINK(3,I) = ANCONT*(ALGADK+MDEATH)
                 CP(4) = .TRUE.
                 GO TO 480
C
C ********************************************************************************
C                        AMMONIA NITROGEN
C ********************************************************************************
C
C     K(4,I) represents NH3 loss due to adsorption onto sediment and
C     nitrification.
C     SINK(4,I) represents net NH3 loss by algal and macrophyte uptake
```

```
C     and NH3 production by hydrolysis of ORG-N.
C     ****************************************************************************
C
4             K(4,I) = KN + KNX
              SINK(4,I) = (K1N + KDN)*C(3,I)-ANCONT*((1.-P)*
     .                    (ALGRO+MGRATE))-(ALGADK+MDEATH)
              IF(AKN(IBRANCH).GT.0.0)CP(5) = .TRUE.
              GO TO 480
C
C     ****************************************************************************
C                          NITRATE NITROGEN
C          (NITRITE NOT EXPLICITLY INCLUDED IN DECAY MECHANISM)
C     ****************************************************************************
C
C     K(5,I) represents NO3 loss via denitrification in the sediment
C     SINK(5,I) represents nitrification in the water column,
C     NO3 loss by denitrification in the water column, and plant uptake
C     of nitrate
C     ****************************************************************************
C
5             K(5,I) =  KDSED(IBRANCH)
              SINK(5,I) = KN*C(4,I)-ANCONT*P*(ALGRO+MGRATE)-ONEQUI*
     .                    KDN*C(2,I)
              GO TO 480
C     ****************************************************************************
C                          ORGANIC PHOSPHATE
C     ****************************************************************************
C
C     K(6,I) represents loss of ORG-P by decay to PO4 and settling of
C     ORG-P SINK(6,I) represents gain of ORG-P by algal and mac decay.
C     algae and mac decay to 100% ORG-P
C     ****************************************************************************
C
6             K(6,I) = KPDK(IBRANCH) + KPSET(IBRANCH)
              SINK(6,I) = APCONT*(ALGADK+MDEATH)
              GO TO 480
C
C     ****************************************************************************
C                          DISSOLVED PHOSPHATE
C     ****************************************************************************
C
C     SINK(7,I) represents uptake of P by algal and macrophyte growth
C     and PO4 release from sediments
C     ****************************************************************************
C
7             K(7,I) = 0.0
              SINK(7,I) = KBENPO4 - APCONT*(ALGRO+MGRATE)
              GO TO 480
C
C     ****************************************************************************
C                          DISSOLVED MANGANESE
C     ****************************************************************************
C
C     K(8,I) represents mn loss by oxidation.  Reaction not corrected
C     for temp.
C     ****************************************************************************
C
8             IF (C(10,I) .GT. OXIDAT(IBRANCH)) THEN
                    K(8,I) = KMNDK(IBRANCH)
              ELSE
                    K(8,I) = 0.0
              END IF
```

```
              SINK(8,I) = 0.0

              GO TO 480
C     *****************************************************************
C                          DISSOLVED IRON
C     *****************************************************************
C
C     K(9,I) represents fe loss by oxidation.  Reaction rate constant
C     not corrected for temperature.
C     *****************************************************************
C
      9       IF (C(10,I) .GT. OXIDAT(IBRANCH)) THEN
                      K(9,I) = KFEDK(IBRANCH)
              ELSE
                      K(9,I) = 0.0
              END IF

              SINK(9,I) = 0.0

              GO TO 480
C
C     *****************************************************************
C                          DISSOLVED OXYGEN
C     *****************************************************************
C
C     K(10,I) represents loss of DOX to the atmosphere by reaeration.
C     SINK(10,I) represents gain of DOX by photosynthesis and  reaeration:
C     loss of DO from fraction of algae and macrophytes oxidized
C     immediately (1-FCBOD(IBRANCH))
C     *****************************************************************

      10      K(10,I) = K2
              SINK(10,I) = (ONEQUI*P + OPDECY)*(ALGRO+MGRATE)-OPDECY*
              .            (DOX/(KOALDK(IBRANCH)+DOX))*(1-FCBOD(IBRANCH))*
              .            (ALGADK+MDEATH)+K2*DOSAT - ONITRI*KN*C(4,I)-K1*
              .            C(2,I)-OMNDEC*KMNX*C(8,I)-OFEDEC*KFEX*C(9,I)
              .            - KSOD
              GO TO 480
C     *****************************************************************
C                          COLIFORM BACTERIA
C     *****************************************************************
C
C     K(11,I) represents loss of fecal coliforms due to die-off,
C     KCOLIDK, corrected for temp
C     *****************************************************************

      11      K(11,I) = KCOLIDK(IBRANCH) * TBIO
              SINK(11,I) = 0.0

              GO TO 480
C     *****************************************************************
C              ALGAL AND MACROPHYTE PRODUCTION AND DECAY
C     *****************************************************************
C
C
C     K(12,I) represents combined effect of growth and decay on algae,
C     corrected for D.O., light and nutrient availability
C     *****************************************************************


      12      K(12,I) = (KALGRO(IBRANCH)*(1/(KEXT*H))*LOG((KLITE(IBRANCH)
              .            +SWALG)/(KLITE(IBRANCH)+SWALG*EXP(-KEXT*H))))*FN*
```

```
                     FP-(KALGDK(IBRANCH)*DOX/(DOX+KOALDK(IBRANCH)))
              SINK(12,I) = 0.0

              GO TO 480
C  **************************************************************************
C                       END CONSTITUENT REACTION ROUTINES
C  **************************************************************************

C     Statements which compute average values of reaction rates and sinks
C     across a time step for each constituent
C  **************************************************************************
C

  480         IF (DT.EQ.0..OR.I.EQ.1 ) GO TO 460
              KS    = (K(N,I) + KS)/2.
              SINKS = (SINK(N,I) + SINKS)/2.


C
C     Final calculation of the constituent concentration and spatial
C     derivative for all nodes.  Statement below checks to see if
C     node is a junction point.  IF SO GO TO 38.
C  **************************************************************************
C
              IF (JUNCT) GO TO 38

              CO  = A1*C(N,I-1) + A2*C(N,I) + A3*DC(N,I-1) + A4*DC(N,I)
              CCL = CLC(N,I)-CO
              CCM = CLT(N,I)-CO
              IF(QLA.LE.0.) CCL =0.
              IF(QLB.LE.0.) CCM =0.
              DC(N,I)=(B1*C(N,I-1)+B2*C(N,I)+B3*DC(N,I-1)+B4*DC(N,I))
        .             *(G0-DT*(KS+QLA+QLB))+DT*(DSINK-C0*DK+CCL*DQLA
        .             +CCM*DQLB)

C     C(N,I) is the concentration of constituent N at (non-junction)
C     node I, time increment J, corrected for sources, sinks and lateral
C     inflow
C  **************************************************************************

              C(N,I) = CO*(1-DT*KS) + (SINKS + CCL*QLA+CCM*QLB)*DT

              GO TO 465
C
C     C(N,I) is the concentration of constituent N for (junction) nodes
C     corrected for sources, sinks and lateral inflow
C  **************************************************************************
C
  38          COO  = T(N + 26,L)
              DC00 = T(N + 38,L)
              CO   = A1*C(N,I-1) + A2*C00 + A3*DC(N,I-1) + A4*DC00
              CCL  = CLC(N,I)-CO
              CCM  = CLT(N,I)-CO
              IF(QLA.LE.0.) CCL = 0.
              IF(QLB.LE.0.) CCM = 0.
              DC00 = (B1*C(N,I-1)+B2*C00+B3*DC(N,I-1)+B4*DC00)*(G0-DT*(KS
        .            + QLA+QLB))+DT*(DSINK-C0*DK + CCL*DQLA+CCM*DQLB)
              C00  = CO*(1-DT*KS) + (SINKS + CCL*QLA+CCM*QLB)*DT
              T(N + 26,L) = C00
              T(N + 38,L) = DC00
              C(N,I)  = C00*(1.-D) + T(N,L)*D
              DC(N,I) = DC00*(1.-D) + T(N + 12,L)*DD


                                  B27
```

```
      465       CONTINUE

C   ***************************************************************************
C         Call to wind-driven reaeration SUBROUTINE REAERK - correct D.O.
C         Concentration to account for wind driven reaeration
C
C         IF ITEM=1, TAIR=DRYO and WIN=WINDO*0.447 with the actual values
C         put in for DRYO and WINDO by user.
C         IF ITEM=0, TAIR=DRYO and WIN=WINDO*0.447, value for tair and
C         win are obtained from met data during heat exchange calculation.
C   ***************************************************************************
                IF (N.EQ.10 .AND. QWIND)  THEN
                    TAIR=DRYO
                    WIN=WINDO*0.447
                    IF(ITEM(IBRANCH).EQ.1.AND.TAIR.EQ.0.AND.WIN.EQ.0)THEN
                        WRITE(OUT,9089)
 9089 FORMAT(' USER MUST PROVIDE A CONSTANT VALUE FOR DRY BULB TEMP',/ ,
     .  ' AND WIND SPEED (TAIR AND WIN) SINCE ITEM=1 AND QWIND=TRUE.',/ ,
     .  ' DO THIS BY REPLACING DRYO AND WINDO WITH NUMERICAL VALUES',/,
     .  ' IN THE STATEMENTS DIRECTLY ABOVE LINE 9089 IN THE SOURCE',/,
     .  ' CODE.  IF NOT DONE DRYO AND WINDO ARE SET TO 0 AND THIS ',/,
     .  ' ERROR RESULTS.')
                        STOP
                    END IF

                    CALL REAERK(WIN,TAIR,TEMP,RK)
                    DTS = DT * 86400.
                    RK = RK/86400.
                    C(10,I) = C(10,I) + RK * (DOSAT-C(10,I)) * DTS / H
                END IF

C   ***************************************************************************
C         Negative hedge for all wq constituents except temperature.
C         That is, no concentrations are allowed to go below zero.
C   ***************************************************************************
                IF (N.GT.1) THEN
                    IF (C(N,I).LT.0.) C(N,I) = 0.
                END IF

      460       CONTINUE

C         END "DO 460" Concentration-computation loop for a given node and
C         time step
C   ***************************************************************************

              ULR=ULL

      470 CONTINUE

C         END "DO 470" Node computation loop, put values in proper arrays
C         and DO diffusion calculations
C   ***************************************************************************

          IF (ITO .LE. 0 ) GO TO 560
C
C         Values at the downstream terminus placed in the t-array
C   ***************************************************************************
C
          T(25,ITO) = Q(MNODE)
          T(26,ITO) = Q(MNODE)/A(MNODE)

          DO 16 N = 1,12
```

```fortran
      IF (.NOT.CP(N)) GO TO 16
      T(N,ITO) = C(N,MNODE)
      T(N + 12,ITO) = DC(N,MNODE)
  16 CONTINUE
 560  CONTINUE


C     Assign boundary conditions to the first node
C *****************************************************************************
C
      IF (DT.LE.0.)THEN
         DO 1112 I=1,MNODE
         Q0(I) = Q(I)
         A0(I) = A(I)
         B0(I)= B(I)
         EL0(I) = EL(I)
 1112    CONTINUE
         RETURN
      ENDIF

      DO 403 N = 1,12
         IF (CP(N)) C(N,1) = BC(N)
  403 CONTINUE
C
C     At this point, the computed constituent concentrations are
C     corrected for dispersion using an implicit numerical scheme.
C     Resulting set of equations solved using subroutine tridag.
C *****************************************************************************

      F(1,2)      = 1.
      F(1,3)      = 0.
      F(MNODE,1)  = 0.
      F(MNODE,2)  = 1.
      MN1         = MNODE - 1

      DO 500 I = 2,MN1
         F0 = 2.*DISP(I)*THETA*DT/(DX1(I-1) + DX1(I))
         F(I,1) = -F0/DX1(I-1)
         F(I,3) = -F0/DX1(I)
         F(I,2) = 1. - F(I,1) - F(I,3)
 500     CONTINUE
C
C     Now the right-hand side for each modeled constituent
C *****************************************************************************
C
      DO 510 N = 1,12
         IF(.NOT.CP(N))GO TO 510
         F(1,4) = C(N,1)
         F(MNODE,4) = C(N,MNODE)

         DO 520 I = 2,MN1
            F(I,4) = RTHETA*(F(I,1)*C(N,I-1) + F(I,3)*C(N,I+1) +
     *               (F(I,2)-1.)*C(N,I)) + C(N,I)
 520        CONTINUE

         CALL TRIDAG(F(1,1),F(1,2),F(1,3),F(1,4),F(1,5),F(1,6),
     *               C,MNODE,N)

         F(1,4) = DC(N,1)
         F(MNODE,4) = DC(N,MNODE)

         DO 540 I = 2,MN1
```

```
              F(I,4) = RTHETA*(F(I,1)*DC(N,I-1) + F(I,3)*DC(N,I+1) +
         *              (F(I,2)-1.)*DC(N,I)) + DC(N,I)
540       CONTINUE

          CALL TRIDAG (F(1,1),F(1,2),F(1,3),F(1,4),F(1,5 ),F(1,6),
         *              DC,MNODE,N)
510   CONTINUE
C
C     Dispersion for Tributary Junctions
C
C     Tributary influx is modeled as a discontinuity and so the values
C     immediately upstream of the junction point must be stored.  The
C     first 24 members of the T-array contain concentrations and their
C     derivatives at the mouth of the trib, location 25 contains trib
C     flow, and location 26 contains the trib velocity.  Locations 27-38
C     contain concentrations in the mainstem just upstream of the
C     junction point. Locations 39-50 contain the concentration
C     derivatives there.
C ******************************************************************************

      IF (IT1.LE.0)RETURN

      DO 550 LL = IT1,IT2
         I = JT(LL)
         F0 = 2.*DISP(I)*DT/(DX1(I-1) + DX1(I))
         F1 = F0/DX1(I-1)
         F3 = F0/DX1(I)
         F2 = 1. + F1 + F3

         DO 550 N=1,12
            IF (.NOT.CP(N)) GO TO 550
            T(N + 26,LL) = (T(N+26,LL)+F1*C(N,I-1)+F3*C (N,I+1))/F2
            T(N + 38,LL) = (T(N+38,LL)+F1*DC(N,I-1)+F3*DC(N,I+1))/F2
550   CONTINUE

      RETURN
      END
      SUBROUTINE SPLINE (C,DC,CP,DX,F,MNODE)
C ******************************************************************************
C                 S U B R O U T I N E    S P L I N E
C
C     SPLINE: A program to fit a cubic spline to the data, with
C     continuous zeroeth, first and second derivatives.   the output
C     is the value of derivative at each point.
C ******************************************************************************

      DIMENSION C(12,MNODE),DC(12,MNODE),CP(12),DX(MNODE),F(MNODE,6)

      LOGICAL CP

      F(1,1)    = 0.
      F(1,2)    = 1.
      F(1,3)    = 0.5
      F(MNODE,1) = 0.5
      F(MNODE,2) = 1.
      F(MNODE,3) = 0.
      M2 = MNODE-1

      DO 2 I = 2,M2
         F(I,1) = DX(I)
         F(I,2) = 2.*(DX(I-1) + DX(I))
         F(I,3) = DX(I-1)
```

```fortran
    2 CONTINUE

      DO 4 N = 1,12
          IF (.NOT.CP(N)) GO TO 4
          F(1,4) = 1.5/DX(1)*(C(N,2)-C(N,1))
          F(MNODE,4) = 1.5/DX(M2)*(C(N,MNODE)-C(N,M2))

          DO 3 I = 2, M2
              F(I,4) = 3.*(DX(I-1)/DX(I)*(C(N,I + 1)-C(N,I))+ DX(I)/
                       DX(I-1)*(C(N,I)-C(N,I-1)))
    3     CONTINUE
C
C         CALL SUBROUTINE TRIDAG
C ***************************************************************************
C
          CALL TRIDAG (F(1,1),F(1,2),F(1,3),F(1,4),F(1,5),F(1,6),
     *                 DC,MNODE,N)
    4 CONTINUE

      RETURN
      END

C ***************************************************************************
C               S U B R O U T I N E    T R I D A G
C ***************************************************************************
      SUBROUTINE TRIDAG (A,B,C,D,BETA,GAMMA,V,M1,N)

      DIMENSION A(M1),B(M1),C(M1),D(M1),BETA(M1),GAMMA(M1),V(12,M1)
C
C ***************************************************************************
C         NOTE that the MATRICES F(1,1) through F(1,6) are passed to
C         TRIDAG where they are received as 1-D vectors with M1
C         positions.  What has actually occurred is a matrix F(I,N)
C         has been received as a 1-D vector containing
C         MN1 values corresponding to the right most subscript
C         where:
C              A(M1) = F(I,1), I VARIES FROM 1 TO MN1
C              B(M1) = F(I,2), I VARIES FROM 1 TO MN1
C              C(M1) = F(I,3), I VARIES FROM 1 TO MN1
C              D(M1) = F(I,4), I VARIES FROM 1 TO MN1
C              BETA(M1) = F(I,5),  "    "    "    "    "    "
C              GAMMA(M1) = F(I,6), "    "    "    "    "    "
C ***************************************************************************
C
      BETA(1) = B(1)
      GAMMA(1)= D(1)/BETA(1)

      DO 1 I = 2, M1
          BETA(I) = B(I)-A(I)*C(I-1)/BETA(I-1)
    1 GAMMA(I)= (D(I)-A(I)*GAMMA(I-1))/BETA(I)

      V(N,M1) = GAMMA(M1)
      M2 = M1-1

      DO 2 K = 1,M2
          I = M1-K
    2 V(N,I) = GAMMA(I)-C(I)*V(N,I + 1)/BETA(I)

      RETURN
      END
C
C ***************************************************************************
```

```
C              S U B R O U T I N E      H E A T
C
C         This SUBROUTINE is adapted from "QUAL-II" Stream
C         Quality Model, written by Larry A. Roesner, Paul
C         R. Giguere and Donald E. Evenson in Water Resources
C         Engineers, Inc.  710 South Broadway, Walnut Creek,
C         California 94596
C ****************************************************************************

       SUBROUTINE HEAT

       INCLUDE 'RIV1Q.CMN'
C ****************************************************************************
C                       DEFAULT VALUE
C ****************************************************************************
C
C
C         IRFLAG : 0, Read only once
C                  1, No longer read data
C         JDYO   : The day of year to start simulation
C         LMN    : Counter for segment number
C         LII    : Counter for node number
C         LATUDC : Local latitude, degree
C         LONTUC : Local longitude, degree
C         LSMC   : Longitude of standard meridian, degree
C         DUC    : Dust attenuation coefficient
C         NUM    : Number of meteorological updates
C         NUP(I) : largest iteration at which data I applies
C         DUMMY  : Array containing all meteorological updates
C         CLOUDO : Cloud cover (fraction)
C         WINDO  : Wind speed (mile/hr)
C         DRYO   : Dry bulb temperature (f)
C         WETO   : Wet bulb temperature (f)
C         ATMO   : Atmospheric pressure (inch/hg)
C
C         The following data are only read once
C         IRFLAG will equal 0 unless specified otherwise
C         as is done elsewhere
C ****************************************************************************
C
C
          FWST = TEMP


C     Call to SUBROUTINE HEATFLUX
C ****************************************************************************
C
       CALL HEATFLUX (JDYO,              DUC,              LSMC,
      *               LLMC,              LATUDC,           LONTUC,
      *               TEMP,              WINDO,            CLOUDO,
      *               ATMO,              DEPTHO,           RHOO,
      *               WETO,              DRYO,             HFLUX )

C
C     Conversion heat flux from 'BTU' to KCAL, then to degree  (f),
C     then to (c).  Thus, DTEM is in Deg C.
C     NOTE: Here and in CONS7
C           CP = 1.0 CAL/G-C     RHO = 1 G/CM^3
C           BTU * 0.252 = KCAL
C           28.33 = (100CM)**3/(1000 CAL/KCAL* 35.31 FT**3/M**3)
C ****************************************************************************
C
       DTEM  = (HFLUX * 0.252) /CONS7
```

```
      FWST = FWST + DTEM
      TEMP = FWST

      RETURN
      END
C
C *****************************************************************
C          S U B R O U T I N E     H E A T F L U X
C *****************************************************************
C
C *****************************************************************
C      This is general subroutine to calculate the heat.  The
C      following data have to be supplied, either by reading from
C      own, or from other program.
C
C                JDY      : Day of year starting simulation
C                LSM      : Longitude of standard
C                           meridian,   degree
C                LATUD    : Local latitude, degree
C                LONTUD   : Local logitude, degree
C                           -1 for west,  + 1 for east
C                CLOUD    : Cloudiness, 0 - 0.99
C                DU       : Dust attenuation, value
C                           from 0 - 0.13
C                WETBLB   : Wet bulb temp, f
C                DRYBLB   : Dry bulb temp, f
C                WINSPD   : Wind speed, mile/hour
C                ELEV     : Water surface elev,ft
C                TEM      : Temperature,  c
C                RHO=RHOO : Density, gram per cubic
C                           millimeter
C                HNEF     : Heat flux, value returned
C                           as BTU/FT**2
C *****************************************************************

      SUBROUTINE HEATFLUX (JDY,DU,LSM,LLM,LATUD,LONTUD,TEM,WINSPD,
     .                     CLOUD,ATMPR,DEPTH,RHO,WETBLB,DRYBLB,HNEF)

      INCLUDE 'RIV1Q.CMN'


      REAL          LONTUD,          LATUD,          LSM,
     *              LLM,             DEWPON
C
C     Constant
C *****************************************************************
C
      DATA          HCAP/1.0/,       RWSAR/0.03/,
     *              HSC/438./        SPWEVA/62.22/

      ELEVO = 0.
      ELEVP = EXP(-ELEVO /2532.)
      CONS1 = 2.*PI/365.
      CONS2 = PI*LATUD  /180.
      CONS3 = 180./PI
      CONS4 = 23.45*PI/180.
      CONS5 = PI/12.
      CONS6 = 12./PI
      CONS7 = RHO*28.336*HCAP*DEPTH
      CONST3 = 6.8E-04
      CONST4 = 2.7E-04
```

```
C       Stefan-Boltzman Constant
C ******************************************************************

        STBOLC = 1.73E-09
C
C                   C O M P U T A T I O N
C
C             R A D I A T I O N     E X C H A N G E     H E A T
C
C         (A) Net short-wave solar radiation (HNSWR)
C
C         A-I.  Radiation flux
C       Convert temperature from deg. C to deg. F
C ******************************************************************

        WST  = (TEM * 9.)/5. + 32.
C
C       For the same day, following calculation is same
C ******************************************************************

        IF ( (JTIME .EQ. 1 .AND. LMN .EQ. 1 .AND. LII .EQ. 1) .OR.
     *       (JDY    .NE. JDYO .AND. LMN .EQ. 1 .AND. LII .EQ. 1) ) THEN
C
C       Declination
C ******************************************************************
C
          DESUN  = CONS4*COS(CONS1*(173.-JDY))
          DECLIN = ABS(DESUN)

C       Earth-sun distance
C ******************************************************************

          REARTH = 1.0 + 0.017*COS(CONS1*(186.-JDY))

          ETIME  = 0.000121-0.12319*SIN(CONS1*(JDY-1.)-0.0714)-
     *             0.16549*SIN(2.*CONS1*(JDY-1.) + 0.3088)
C
C       Difference between standard and local civic time
C ******************************************************************
C
          DSTLT = (LSM - LONTUD)/15.
C
C       Standard time of sunrise and sunset
C ******************************************************************
C
          ACS = TAN(CONS2)*TAN(DECLIN)
C
          IF (ACS .NE. 0.) THEN

            X1  = SQRT(1.-ACS*ACS)
            X1  = X1/ACS
            ACS = ATAN(X1)

            IF (DESUN .GT. 0.) ACS = PI-ACS
          ELSE
            ACS = PI/2.0
          END IF
C
          TIMRIS = 12.-(CONS6*ACS) + DSTLT
          TIMSET = 24.-TIMRIS  + 2.*DSTLT
C
        END IF
```

```
C
C         Saturated vapor pressure of the air, SVPA
C     ***********************************************************************
C
          SVPA  = 0.1001*EXP(0.03*WST )-0.0837
C
C     The following are the same for each node but can change at each
C     time step.  LMN and LII equal 1 at each new time step.
C     ***********************************************************************
C
          IF ( LMN .NE. 1 .OR. LII .NE. 1 ) GO TO 480

C
C     If you want to read in dew point temperatures instead of wet
C     bulb, comment out all statements between the XXXXXXX Section
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
C         Saturated vapor pressure,  SVAP
C     ***********************************************************************
C
           SVAP  = 0.1001*EXP(0.03*WETBLB )-0.0837
C
C     Water vapor pressure (WVAP)
C     ***********************************************************************
C
          WVAP  = SVAP-0.000367*ATMPR*(DRYBLB-WETBLB )*(1.0+(WETBLB-32.)/
        .          1571.)
C
C     Dew point
C     ***********************************************************************
C
          DEWPON  = LOG((WVAP + 0.0837)/0.1001)/0.03
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

C     Hour angle
C     ***********************************************************************
C
          IF (TIMSET   .LE. STB .OR.   TIMRIS    .GE. STE)THEN

              GO TO 35

          ELSE IF (TIMRIS   .GT. STB .AND. TIMRIS   .LT. STE)THEN

              DIUEXP  = 1.
              ANGRIS  = TIMRIS -DSTLT  + ETIME-12.
              ANGSET  = STE     -DSTLT  + ETIME-12.

          ELSE IF (TIMSET    .LT. STE .AND. TIMSET .GT. STB)THEN

              DIUEXP  = 1.
              ANGRIS  = STB-DSTLT   + ETIME-12.
              ANGSET  = TIMSET  -DSTLT   + ETIME-12.

          ELSE

              DIUEXP   = 1.
              ANGRIS   = STB-  DSTLT   +  ETIME-12.
              ANGSET   = STE-  DSTLT   +  ETIME-12.
C
          END IF
C
C     Calculate hour angle,T,(HOURAN)
```

```
C  **************************************************************************
C
       HOURAN = (ANGRIS    + ANGSET   )/2.
C
C      Short-wave solar radiation flux
C  **************************************************************************
C
       HRF   = HSC/(REARTH**2.)*(SIN(CONS2)*SIN(DESUN)*(ANGSET-ANGRIS)
      .        +CONS6*COS(CONS2)*COS(DESUN)*(SIN(CONS5*ANGSET)-SIN(CONS5
      .        *ANGRIS)))*DIUEXP

C
C               A-II.   Atmospheric Transmission term, AT(ATT)
C
C      Sun altitude in radians
C  **************************************************************************
C
       ALPH  = SIN(CONS2)*SIN(DESUN)+COS(CONS2)*COS(DESUN)*COS(CONS5*
      .        HOURAN  )

       IF (ABS(ALPH ) .NE. 1.0) THEN

          Y1    = SQRT(1.-ALPH **2)
          Y1    = ALPH /Y1
          ALPH  = ATAN(Y1)

       ELSE IF (ALPH   .NE. -1.0) THEN

          ALPH  = PI/2.0

       ELSE

          ALPH = -PI/2.0

       END IF

       IF (ALPH  .LT. 0.01) GO TO 35
C
C      Optical air mass
C  **************************************************************************
C
       OAM = ELEVP/(SIN(ALPH )+0.15*((180.*ALPH/PI+3.885)**(-1.253)))
C
C      Mean daily precipitate water content
C  **************************************************************************
C
       PWC   = 0.00614*EXP(0.0489*DEWPON )
C
C      Mean atmospheric coefficient, A'(AC)
C  **************************************************************************
C
       AC    = EXP(-(0.465 + 0.0408*PWC )*(0.129 + 0.171*EXP(-0.880*
      *          OAM ))*OAM )
C
C      Mean atmospheric transmission coefficient
C  **************************************************************************
C
       ATC   = EXP(-(0.465+0.0408*PWC)*(0.179+0.421*EXP(-0.721*OAM))*
      .          OAM)
C
C      Determine the value of AR (A), and BR (B), from the value of cloud
C      for the function of reflection  coefficient ,RS.
```

B36

```
C  ***********************************************************************
C
      CNL = CLOUD *10 + 1.0
      NLL = CNL
      GO TO (50,51,51,51,51,51,52,52,52,52,53),NLL
   50 AR = 1.18
      BR = -0.77
      GO TO 54
   51 AR = 2.20
      BR = -0.97
      GO TO 54
   52 AR = 0.95
      BR = -0.75
      GO TO 54
   53 AR = 0.35
      BR = -0.45
   54 CONTINUE
C
C                         Reflection Coefficient
C
C     Solar altitude in degree
C  ***********************************************************************
C
      RS  = AR*(CONS3*ALPH )**BR
      IF(RS.GT.1.0)RS=1.0
C
C     Dampening effect on the solar radiation flux gives satisfactory
C     results except for heavy overcast condition,i.e. when cloud
C     approaches 1.0
C  ***********************************************************************
C
      IF (CLOUD  .LE. 0.9) THEN
         CS = 1.0-0.65*CLOUD **2.
      ELSE
         CS = 0.5
      END IF
C
C     Calculate atmospheric transmission term
C  ***********************************************************************
C
      ATT = (ATC+0.5*(1.-AC-DU))/(1.-0.5*RS *(1.-AC + DU))
C
C     Net short-wave solar radiation btu/ft**2
C  ***********************************************************************
C
      HNSWR = HRF*ATT*(1.-RS)*CS

      GO TO 479

   35 DIUEXP = 0.
      HNSWR  = 0.

  479 CONTINUE
C
C     (B) Calculate long-wave atmospheric radiation,HAN (HNLWR)
C  ***********************************************************************
C
      HNLWR = (2.89E-06)*STBOLC*((DRYBLB+460.)**6)*(1.0+0.17*
     .        (CLOUD**2.))*(1.-RWSAR)

C
C     Compute insolation for algae
```

```fortran
C  ********************************************************************
C
       IF ( DELTAT .LE. 0. ) GO TO 481
       HNEFSW(IBRANCH) = HNSWR / DELTAT

       GO TO 482
  481  CONTINUE
       HNEFSW(IBRANCH) = 0.
  482  CONTINUE
C
C      Convert to w/m**2
C  ********************************************************************
C
       HNEFSW(IBRANCH) = HNEFSW(IBRANCH) * 3.155
  480  CONTINUE
C
C      (C) Calculate water surface back radiation, (HOLWBR)
C  ********************************************************************
C
       HOLWBR = 0.97*STBOLC*((WST  + 460.)**4)
C
C                    E V A P O R A T I O N      H E A T
C
C      (D) Calculate evaporation  (HEVAP)
C
C      D-I.  Calculate evaporate rate, E, (EVARAT)
C  ********************************************************************
C
       EVARAT = (CONST3 + CONST4*WINSPD )*(SVPA  -WVAP )
       IF (EVARAT  .LT. 0.) EVARAT =0.
C
C      D-II. Calculate sensible heat loss(HSENH)
C  ********************************************************************
C
       HSENH = 0.
C
C      D-III. Calculate latent heat of evaporization, HL,(HLEVA)
C  ********************************************************************
C
       HLEVA = 1084.-0.5*WST
C
C      Heat loss by evaporation HE (HEVAP)
C
       HEVAP  = SPWEVA*HLEVA *EVARAT  + HSENH
C
C                    C O N D U C T I O N     H E A T
C
C      (E) Calculate conduction heat, HC, (HCOND)
C  ********************************************************************
C
       HCOND = SPWEVA*HLEVA*(CONST3+CONST4*WINSPD)*(0.01*ATMPR /29.92)*
      .        (DRYBLB -WST)
C
C       Finally, calculate net energy flux passing
C       the air-water interface, HN(HNEF),BTU/FT**2,DELTAT=HOURS
C
       HNEF = HNSWR+(HNLWR-HOLWBR+HCOND-HEVAP)*DELTAT

C
C      Return value here
C  ********************************************************************
C
```

```
      RETURN
      END

      FUNCTION TAN (X)
      TAN=SIN(X/57.3)/COS(X/57.3)
      RETURN
      END

      FUNCTION ASIN(X)
      ASIN=X + (X**3./6.) + (3.*X**5./40.) + (15.*X**7./336.)
      RETURN
      END

      FUNCTION ACOS(X)
      ACOS=ASIN(SQRT(1.-X**2.))
      RETURN
      END
C
C
C ********************************************************************
C
C            S U B R O U T I N E    R E A E R K
C
C     SUBROUTINE REAERK calculates reaeration coefficient (RK) (M/DAY)
C     Given wind speed (WS) (M/S) and temperature (T) (C) using the
C     method presented in JOUR OF ENV ENG,VOL. 109, NO.3,PP.731-752,
C     June 1983,Author: D.J.O'CONNOR, Title: "Wind Effects on Gas-
C     liquid Transfer Coefficients
C     Parameters used in this subroutine were those given for
C     intermediate scale correlation (TABLE 2), I.E., Large Laboratory
C     Systems and Moderately sized lakes.
C ********************************************************************
      SUBROUTINE REAERK(WS,TA,TW,RK)

      INCLUDE 'RIV1Q.CMN'

      REAL*4 KA,LAM,KA3
C
C     Calculate diffusivity of oxygen in water (DIFF) (CM**2/SEC),
C     viscosity of water (VW) (CM**2/SEC),viscosity of air (VA)
C     (CM**2/SEC),density of water (PW) (G/CM**3), density of air
C     (PA) (G/CM**3)
C
      DIFF=4.58E-07*TW+1.2E-05
      TA = (TA-32.)*5./9.
      VW=0.0164-.00024514*TW
      VA=.133+.0009*TA
      PA=.00129-.0000040*TA
      PW=1.00
      WS=WS*100.
      RK=1.
      N=0
C
C        Use Newton-Raphson method to calculate the square
C        root of the drag coefficient.  Parameters used in
C        the model include transitional shear velocity -
C        UT(M/SEC); Critical shear velocity- UC(CM/SEC);
C        Von Karman's constant-(KA); Equilibrium roughness-ZE(CM);
C        1/LAM is a Reynolds Number; GAM is a non-dimensional
C        coefficient dependent on the water body size.  LAM,
C        GAM, UT, MC, and ZE are also dependent on water body size.
C
```

```
C
      UT=10.0
      UC=11.0
      KA=0.4
      KA3=KA**.3333
      ZE=0.25
      LAM=3.0
      GAM=6.5
      WH=1000.

C     Make initial guess for square root of the drag coefficient
C ****************************************************************************
      SRCD=0.04
   10 N=N+1

C     Calculate value of FUNCTION(F2) and derivative of function(FP)
C ****************************************************************************
      EF=EXP(-SRCD*WS/UT)
      F1=LOG((WH/ZE)+(WH*LAM/VA)*SRCD*WS*EF)
      F2=F1-KA/SRCD
      FP1=1./((WH/ZE)+(LAM*WH/VA)*SRCD*WS*EF)
      FP2=((WH*LAM)/(VA*UT))*SRCD*WS**2*EF
      FP3=(WH*LAM/VA)*WS*EF
      FP4=FP1*(FP2+FP3)+(KA/(SRCD**2))

C     Calculate new value of square root of drag coefficient. Compare
C     to previous value. If not acceptable return to Newton-Raphson
C     algorithm.
C ****************************************************************************

      SRCD2=SRCD-F2/FP4
      ERR=ABS(SRCD-SRCD2)
      IF(ERR.GT.0.0005.AND.N.LT.8)THEN
         SRCD=SRCD2
         GO TO 10
      END IF
      IF(ERR.GT.0.005.AND.N.GE.8) GO TO 90
      CD=SRCD**2
      US=SRCD*WS
      Z0=1./((1./ZE)+LAM*US*EXP(-US/UT)/VA)
      WS=WS/100.
      IF(WS.LT.6.0)GO TO 60
      IF(WS.GE.6.0.AND.WS.LE.20.0)GO TO 80
      IF(WS.GT.20.0)GO TO 70

C     Calculate rk for windspeeds less than 6.0 m/sec based on
C     equation 23b in citation
C ****************************************************************************

   60 RK1=(DIFF/VW)**0.666667*SRCD*(PA/PW)**0.5
      RK=RK1*KA3*WS/GAM
      RK=RK*3600.*24.
      GO TO 85

C     Calculate RK for windspeeds greater than 20 m/s based on
C     equation 25b in citation
C ****************************************************************************

   70 RK=(DIFF*PA*VA*US/(0.1*PW*VW))**0.5
      RK=RK*3600.*24./100.
      GO TO 85
```

```
C      Calculate RK for windspeeds between 6 and 20 m/s based on
C      equations 26a and 27 in citation
C *****************************************************************************

   80  GAMU=GAM*US*EXP(-US/UC+1.)/UC
       RK1=(DIFF/VW)**.6667*KA3*(PA/PW)**0.5*US/GAMU
       RK2=(DIFF*US*PA*VA/(KA*ZO*PW*VW))**0.5
       RK3=(1./RK1)+(1./RK2)
       RK=1./RK3
       RK=RK*3600.*24./100.
       GO TO 85
   90  WRITE(OUT,102)
  102  FORMAT(5X,'SOLUTION DID NOT CONVERGE')
   85  CONTINUE
       RETURN
       END
C
C
C *****************************************************************************
C                  S U B R O U T I N E   R I V D U
C *****************************************************************************
C
       SUBROUTINE RIVDU(IFIRST)
C
       INCLUDE 'RIV1Q.CMN'
       INCLUDE 'TRANSP.CMN'

       COMMON/ABLOCK/ NS

C
       CHARACTER*30 DNAME (18)
       DATA DNAME/
      +         'Distance (mi)          ' , 'Stage (ft)            ',
      +         'Velocity (ft/sec)      ' , 'Flow (cfs)            ',
      +         'Hydraulic Depth (ft)'   , 'Temperature           ',
      +         'CBODNS                 ' , 'Organic N             ',
      +         'NH3-N                  ' , 'NO3-N                 ',
      +         'ORGANIC-P              ' , 'PO4                   ',
      +         'MN                     ' , 'FE                    ',
      +         'DO                     ' , 'COLIFORM BACT         ',
      +         'ALGAE                  ' , 'DUMMY                 '/


       RDUMMY = 0.

       IF (IFIRST .EQ. 1) THEN
          NCHEM = 1
          WRITE (RIVDMP, 6000) NCHEM
 6000     FORMAT (I5)
C
          WRITE (RIVDMP, 6010) RDUMMY,RDUMMY
 6010     FORMAT(A40,E10.2,F10.2)
          TIME = 0.
          TFIN = ENDTIME
          WRITE (RIVDMP, 6020) TIME, TFIN, MNODE
 6020     FORMAT(1X,F10.2,F10.2,/,I5)
          WRITE (RIVDMP, 6030) (DNAME (JK), JK = 1, 6 + (12*NCHEM))
 6030     FORMAT(A30)
          IFIRST = 1
       END IF
C
       DO 100 I=1,MNODE
```

```fortran
      SYSDUMP(1,I) = RMILE(I)
      SYSDUMP(2,I) = ELEV(I)
      SYSDUMP(3,I) = ((FLOW(I)/86400.)/AREA(I))
      SYSDUMP(4,I) = FLOW(I)/86400.
      SYSDUMP(5,I) = AREA(I)/WIDTH(I)
      DO 1 IK = 1,12
          SYSDUMP(5+IK,I) = CONC(IK,I)
    1 CONTINUE
      WRITE(RIVDMP,6040)I,ELAPSE,RMILE(I),ELEV(I),(FLOW(I)/86400)/
     .              AREA(I),FLOW(I)/86400.,AREA(I)/WIDTH(I),
     .              (CONC(NN,I),NN=1,12),RDUMMY

 6040    FORMAT(1X,I5,F10.2,3X,/,6(E11.3),/,6(E11.3))
 100  CONTINUE
C

      RETURN
      END

C     S U B R O U T I N E    G R E G O R I A N    D A T E

      SUBROUTINE GREGORIAN_DATE(ELAPSE,YEAR,MONTH,GDAY,HOUR)

C     Variable declarations
C *************************************************************************

      LOGICAL LEAP_YEAR
      CHARACTER MONTH*9
      INTEGER YEAR, GDAY

C     Determine if new year (regular or leap) and increment year
C *************************************************************************

      JDAYG=ELAPSE - 1
      HOUR = (ELAPSE-FLOAT(JDAYG) - 1)*24.
      LEAP_YEAR       = MOD(YEAR,4).EQ.0

      IF (.NOT.LEAP_YEAR.AND.JDAYG.EQ.365) THEN
          JDAYG      = JDAYG-365
          YEAR       = YEAR+1
          LEAP_YEAR = MOD(YEAR,4).EQ.0
      ELSE IF (JDAYG.EQ.366) THEN
          JDAYG      = JDAYG-366
          YEAR       = YEAR+1
          LEAP_YEAR = .FALSE.
      END IF

C     Determine month and day of year
C *************************************************************************

      IF (LEAP_YEAR) THEN
          IF (JDAYG.GE.0.AND.JDAYG.LT.31) THEN
              GDAY  = JDAYG+1
              DAYM  = 31.0
              MONTH = '  January'
          ELSE IF (JDAYG.GE.31.AND.JDAYG.LT.60) THEN
              GDAY  = JDAYG-30
              DAYM  = 29.0
              MONTH = ' February'
          ELSE IF (JDAYG.GE.60.AND.JDAYG.LT.91) THEN
              GDAY  = JDAYG-59
              DAYM  = 31.0
              MONTH = '    March'
```

```
      ELSE IF (JDAYG.GE.91.AND.JDAYG.LT.121) THEN
         GDAY  = JDAYG-90
         DAYM  = 30.0
         MONTH = '     April'
      ELSE IF (JDAYG.GE.121.AND.JDAYG.LT.152) THEN
         GDAY  = JDAYG-120
         DAYM  = 31.0
         MONTH = '      May'
      ELSE IF (JDAYG.GE.152.AND.JDAYG.LT.182) THEN
         GDAY  = JDAYG-151
         DAYM  = 30.0
         MONTH = '     June'
      ELSE IF (JDAYG.GE.182.AND.JDAYG.LT.213) THEN
         GDAY  = JDAYG-181
         DAYM  = 31.0
         MONTH = '     July'
      ELSE IF (JDAYG.GE.213.AND.JDAYG.LT.244) THEN
         GDAY  = JDAYG-212
         DAYM  = 31.0
         MONTH = '   August'
      ELSE IF (JDAYG.GE.244.AND.JDAYG.LT.274) THEN
         GDAY  = JDAYG-243
         DAYM  = 30.0
         MONTH = 'September'
      ELSE IF (JDAYG.GE.274.AND.JDAYG.LT.305) THEN
         GDAY  = JDAYG-273
         DAYM  = 31.0
         MONTH = '  October'
      ELSE IF (JDAYG.GE.305.AND.JDAYG.LT.335) THEN
         GDAY  = JDAYG-304
         DAYM  = 30.0
         MONTH = ' November'
      ELSE IF (JDAYG.GE.335.AND.JDAYG.LT.366) THEN
         GDAY  = JDAYG-334
         DAYM  = 31.0
         MONTH = ' December'
      END IF
   ELSE
      IF (JDAYG.GE.0.AND.JDAYG.LT.31) THEN
         GDAY  = JDAYG+1
         DAYM  = 31.0
         MONTH = '  January'
      ELSE IF (JDAYG.GE.31.AND.JDAYG.LT.59) THEN
         GDAY  = JDAYG-30
         DAYM  = 29.0
         MONTH = ' February'
      ELSE IF (JDAYG.GE.59.AND.JDAYG.LT.90) THEN
         GDAY  = JDAYG-58
         DAYM  = 31.0
         MONTH = '    March'
      ELSE IF (JDAYG.GE.90.AND.JDAYG.LT.120) THEN
         GDAY  = JDAYG-89
         DAYM  = 30.0
         MONTH = '    April'
      ELSE IF (JDAYG.GE.120.AND.JDAYG.LT.151) THEN
         GDAY  = JDAYG-119
         DAYM  = 31.0
         MONTH = '      May'
      ELSE IF (JDAYG.GE.151.AND.JDAYG.LT.181)THEN
         GDAY  = JDAYG-150
         DAYM  = 30.0
         MONTH = '     June'
```

```fortran
      ELSE IF (JDAYG.GE.181.AND.JDAYG.LT.212) THEN
          GDAY  = JDAYG-180
          DAYM  = 31.0
          MONTH = '     July'
      ELSE IF (JDAYG.GE.212.AND.JDAYG.LT.243) THEN
          GDAY  = JDAYG-211
          DAYM  = 31.0
          MONTH = '   August'
      ELSE IF (JDAYG.GE.243.AND.JDAYG.LT.273) THEN
          GDAY  = JDAYG-242
          DAYM  = 30.0
          MONTH = 'September'
      ELSE IF (JDAYG.GE.273.AND.JDAYG.LT.304) THEN
          GDAY  = JDAYG-272
          DAYM  = 31.0
          MONTH = '  October'
      ELSE IF (JDAYG.GE.304.AND.JDAYG.LT.334) THEN
          GDAY  = JDAYG-303
          DAYM  = 30.0
          MONTH = ' November'
      ELSE IF (JDAYG.GE.334.AND.JDAYG.LT.365) THEN
          GDAY  = JDAYG-333
          DAYM  = 31.0
          MONTH = ' December'
      END IF
      END IF
      RETURN
      END

C ***********************************************************************
C               S U B R O U T I N E   J U L I A N   D A T E
C ***********************************************************************

      SUBROUTINE JULIAN_DATE(STADY,ENDY,SYEAR,SMONTH,SDAY,SHOUR,
     .       EYEAR,EMONTH,EDAY,EHOUR)

C      Variable declarations
C ***********************************************************************

      DIMENSION DAYM(12)
      INTEGER  SYEAR,SMONTH,SDAY,EYEAR,EMONTH,EDAY

C      Specify days for each month of the year
C ***********************************************************************

      DAYM(1)  = 31.0
      DAYM(2)  = 29.0
      DAYM(3)  = 31.0
      DAYM(4)  = 30.0
      DAYM(5)  = 31.0
      DAYM(6)  = 30.0
      DAYM(7)  = 31.0
      DAYM(8)  = 31.0
      DAYM(9)  = 30.0
      DAYM(10) = 31.0
      DAYM(11) = 30.0
      DAYM(12) = 31.0


C      Determine Julian day for start time
C ***********************************************************************
      STADY = 0.
```

```fortran
      DAYM(2) = 28.
      IF(MOD(SYEAR,4).EQ.0)DAYM(2) = 29.0
      IF(SMONTH.GT.1) THEN
          DO 100 I=1,SMONTH-1
              STADY = STADY + DAYM(I)
100       CONTINUE
      END IF

      STADY = STADY + SDAY + SHOUR/24.

C     Determine Julian day for end time
C     ****************************************************************************

      ENDY = 0.
      IF(EYEAR.GT.SYEAR) THEN
          DO 200 I=SYEAR,EYEAR-1
              IF(MOD(I,4).EQ.0)THEN
                  ENDY= ENDY + 366.
              ELSE
                  ENDY = ENDY + 365.
              END IF
200       CONTINUE
      END IF

      DAYM(2) = 28.

      IF(MOD(EYEAR,4).EQ.0)DAYM(2)=29.
      IF(EMONTH.GT.1) THEN
          DO 300 I=1,EMONTH-1
              ENDY = ENDY + DAYM(I)
300       CONTINUE
      END IF
      ENDY = ENDY +EDAY + EHOUR/24.

      RETURN
      END

      FUNCTION VALNEW(DAY,T1,T2,V1,V2)
      RATIO = (T2-DAY)/(T2-T1)
      VAL = (1.0-RATIO)*V2+RATIO*V1
      VALNEW = VAL
      RETURN
      END

C     ****************************************************************************
C            S U B R O U T I N E    T I M E    V A R Y I N G    D A T A
C     ****************************************************************************

      SUBROUTINE TIME_VARYING_DATA (JDAY,TNXTVD,NBC,DUMMY,NS)

      INCLUDE 'RIV1Q.CMN'
      INCLUDE 'TRANSP.CMN'

      REAL JDAY


      LOGICAL  INT_LAT, INT_BC(IBRAN), INT_MET
      CHARACTER*15 NPSNAME(ISYS)
      CHARACTER*15 INTOPT, BCFNAME(IBRAN)

C     Dimension variables for boundary conditions
C     ****************************************************************************
```

```fortran
      DIMENSION   TNXTBC(IBRAN), TNXTBC2(IBRAN)
      DIMENSION   DUMMY(ISYS,IBRAN),DUMMY1(ISYS,IBRAN), DUMMY2(ISYS)
      DIMENSION   INBC(IBRAN),NUMBC(IBRAN)
C     Dimension variables for lateral inflows
C     ***********************************************************************

      DIMENSION CLT2(ISYS,IND1),CLT1(ISYS,IND1), NPS_REA(IND1),
     .          NPS_SYS(ISYS)

      SAVE

C
C     On first call to SUBROUTINE:  Initialize all variables and
C     open files
C     ***********************************************************************

C     Initialize logical variables
C     ***********************************************************************

      IF(JTIME.EQ.1)THEN
          DUM  = 0.
          IDUM = 0
          INT_LAT = .FALSE.
          DO J=1,NBC
              INT_BC(J)= .FALSE.
          END DO

          INT_MET = .FALSE.

C     1st - Set up boundary conditions
C     ***********************************************************************

          IBF = 30

          DO I=1,NBC
              READ(INPUT,1011)BCFNAME(I)
              OPEN(UNIT=IBF,FILE=BCFNAME(I),STATUS='OLD',IOSTAT=ISTAT)
              IF (ISTAT .NE. 0)THEN
                  WRITE(6,6000)
6000              FORMAT('Time Variable Boundary File Missing!')
                  STOP
              ENDIF
              READ(IBF,1010)INBC(I),NUMBC(I),INTBC
              WRITE(OUT,2000)INBC(I)
              WRITE(OUT,2010)BCFNAME(I)
              IF(INTBC.GE.1)THEN
                  INT_BC(I)=.TRUE.
                  INTOPT ='LINEAR INTERPOL'
              ELSE
                  INTOPT ='STEP FUNCTION'
              END IF
              WRITE(OUT,2045)INTOPT
              TNXTBC(I) = 0.
              IBF = IBF + 1
          END DO

C     2nd - Read meteorlogical data
C     ***********************************************************************

          IF(ITEM(1).EQ.0) THEN
              READ(MET, 1015)DUC
```

B46

```
            READ(MET, 1015)LATUDC, LONTUC, LSMC
            READ(MET,1010)INTMET,NUMT
            IF(INTMET.GE.1)THEN
                INT MET= .TRUE.
                INTOPT ='LINEAR INTERPOL'
            ELSE
                INTOPT ='STEP FUNCTION'
            END IF

            WRITE(OUT,2020) DUC,LATUDC,LONTUC,LSMC
            WRITE(OUT,2045)INTOPT
        END IF

        TNCTMT = 0.

C     3rd - Read time varying lateral concentrations (IF ICL *=1)
C     ***********************************************************************

        IF(ICL.GE.1) THEN
            READ(LAT,1010)NUMLAT UD,NUML,LATOPT,NUMSYS
            IF(LATOPT .GE. 1)THEN
                INT LAT=.TRUE.
                INTOPT ='LINEAR INT'
            ELSE
                INTOPT ='STEP FUNC'
            END IF

            READ(LAT,1030)(NPS_REA(I), I = 1, NUML)
            WRITE(OUT,2030)
            WRITE(OUT,2040)(NPS REA(I), I = 1, NUML)
            READ(LAT,1030)(NPS_SYS(I), I = 1, NUMSYS)
            WRITE(OUT,2050)
            DO I= 1, NUMSYS
                READ(LAT,1040)NPSNAME(I)
                WRITE(OUT,2060)NPS_SYS(I),NPSNAME(I)
            END DO
            WRITE(OUT,2045)INTOPT
        END IF


    END IF

C
C     On first and all subsequent calls, update time varying data
C     ***********************************************************************
C

C     Assign boundary data
C     ***********************************************************************

        IBF = 30

        DO 600 I=1,NBC

            IF (JDAY.GE.TNXTBC(I)) THEN
                DO WHILE (JDAY.GE.TNXTBC(I))
                    TNXTBC2(I)  =TNXTBC(I)
                    DO J=1,ISYS
                        DUMMY2(J) = DUMMY1(J,INBC(I))
                    END DO
                    READ(IBF,1020)IYR,IMO,IDY,THR,(DUMMY1(J,INBC(I)),
                                  J=1,ISYS)
```

B47

```
                CALL JULIAN_DATE(DUM,TNXTBC(I),INITYR,IDUM,IDUM,DUM,
                            IYR,IMO,IDY,THR)
            END DO

            DO J=1,ISYS
                DUMMY(J,INBC(I))= DUMMY1(J,INBC(I))
            END DO

C           WRITE(OUT,2070) INBC(I),IYR,IMO,IDY,THR,JDAY,TNXTBC(I)
C           WRITE(OUT,2080)(J,DUMMY(J,INBC(I)),J=1,ISYS)
            TNXTVD = MIN(TNXTVD,TNXTBC(I))

        END IF

        IF(INT_BC(I)) THEN
            DO J=1,ISYS
                DUMMY(J,INBC(I))=VALNEW(JDAY,TNXTBC(I),TNXTBC2(I),
                            DUMMY1(J,INBC(I)),DUMMY2(J))
            END DO
        END IF

        IBF = IBF + 1

600     CONTINUE

C
C
C       Assign meteorological data
C       ********************************************************************
C
        IF (JDAY.GE.TNXTMT.AND.ITEM(1).EQ.0) THEN
            DO WHILE (JDAY.GE.TNXTMT)
                TNXTMT2 = TNXTMT
                CLOUD2  = CLOUD1
                WIND2   = WIND1
                DRY2    = DRY1
                WET2    = WET1
                ATM2    = ATM1
                READ(MET,1020) IYR,IMO,IDY,THR,CLOUD1,WIND1,DRY1,WET1,ATM1
                CALL JULIAN_DATE(DUM,TNXTMT,INITYR,IDUM,IDUM,DUM,
                            IYR,IMO,IDY,THR)

            END DO

            CLOUD0  = CLOUD1
            WIND0   = WIND1
            DRY0    = DRY1
            WET0    = WET1
            ATM0    = ATM1
C           WRITE(OUT,2090) IYR,IMO,IDY,THR,JDAY,TNXTMT
C           WRITE(OUT,3000) CLOUD0,WIND0,DRY0,WET0,ATM0
            TNXTVD = MIN(TNXTVD,TNXTMT)
            FWST = TEMP
        END IF

        IF(INT_MET.AND.ITEM(1).EQ.0) THEN
            CLOUD0 = VALNEW(JDAY,TNXTMT,TNXTMT2,CLOUD1,CLOUD2)
            WIND0  = VALNEW(JDAY,TNXTMT,TNXTMT2,WIND1,WIND2)
            DRY0   = VALNEW(JDAY,TNXTMT,TNXTMT2,DRY1,  DRY2)
            WET0   = VALNEW(JDAY,TNXTMT,TNXTMT2,WET1,  WET2)
            ATM0   = VALNEW(JDAY,TNXTMT,TNXTMT2,ATM1,  ATM2)
        END IF
```

B48

```
C      Assign lateral inflow data
C      **********************************************************************
C
       IF (JDAY.GE.TNXTLT.AND.ICL.GE.1) THEN
          DO WHILE (JDAY.GE.TNXTLT)
             TNXTLT2 = TNXTLT
             DO II = 1, NUMSYS
                DO IK = 1, NUML
                   CLT2(NPS_SYS(II),NPS_REA(IK)) =
                        CLT1(NPS_SYS(II),NPS_REA(IK))
                END DO
             END DO
             READ(LAT,1020)  IYR,IMO,IDY,THR
             CALL JULIAN_DATE(DUM,TNXTLT,INITYR,IDUM,IDUM,DUM,
                              IYR,IMO,IDY,THR)
             DO II = 1, NUMSYS
                READ(LAT,1050)(CLT1(NPS_SYS(II),NPS_REA(IK)),IK=1,NUML)
             END DO
          END DO

C         WRITE(OUT,3010) IYR,IMO,IDY,THR,JDAY,TNXTLT
C         WRITE(OUT,3020)(NPS_REA(IK),IK=1,NUML)

          DO II=1,NUMSYS
             DO IK = 1, NUML
                CLT(NPS_SYS(II),NPS_REA(IK)) =CLT1(NPS_SYS(II),
                                               NPS_REA(IK))
             END DO
C            WRITE(OUT,3030)NPS_SYS(II),(CLT(NPS_SYS(II),NPS_REA(IK)),
C            .               IK=1,NUML)
          END DO
          TNXTVD = MIN(TNXTVD,TNXTLT)
       END IF

       IF(INT_LAT.AND.ICL.GE.1) THEN
          DO I = 1, NUMSYS
             DO J=1,NUML
                CLT(NPS_SYS(I),NPS_REA(J)) =VALNEW(JDAY,TNXTLT,TNXTLT2,
                                            CLT1(NPS_SYS(I),NPS_REA(J)),
                                            CLT2(NPS_SYS(I),NPS_REA(J)))
             END DO
          END DO
       END IF


C      FORMAT STATEMENTS
C         1) Input formats
C      **********************************************************************

 1010 FORMAT(8I10)
 1011 FORMAT(A12)
 1015 FORMAT(15F10.4)
 1020 FORMAT(I5,I5,I5,15F10.4)
 1030 FORMAT(I5)
 1040 FORMAT(A15)
 1050 FORMAT(15X,150(F10.0))

C      2) Output formats
C      **********************************************************************

 2000 FORMAT(/,'**** BOUNDARY CONDITIONS SPECIFIED FOR SEGMENT****',I4)
 2010 FORMAT('   WILL BE READ FROM FILE =  ',A15)
```

```
 2020 FORMAT(/,'**************METEOROLOGICAL DATA***************',/,
     .'DUC= ',F5.2,1X,'LATUDC=',F6.2,1X,'LONTUC=',F6.2,1X,'LSMC=',F6.2)
 2030 FORMAT(/,'**TIME VARYING LATERAL INFLOWS SPECIFIED AT
     .X-SECTIONS')
 2040 FORMAT(10(1X,I5))
 2045 FORMAT('     INTERPOLATION OPTION    ',a15,//)
 2050 FORMAT(' FOR WATER QUALITY CONSTITUENTS:')
 2060 FORMAT(I5,A15)
C
 2070 FORMAT(/,'     **BOUNDARY CONDITION FOR SEGMENT ',I5,
     .' UPDATED AT: YEAR ',I6,' MONTH ',I3,' DAY ',I3,' HOUR',F8.4,
     . ' JULIAN DATE',F10.4,', NEXT UPDATE ON JULIAN DAY '
     . ,F10.4,/)
 2080 FORMAT('SYSTEM = ',I5,' NEW VALUE = ',F10.4)
 2090 FORMAT(/,'     **METEOROLOGICAL CONDITIONS UPDATED AT: YEAR ',
     . I6,' MONTH ',I3,' DAY ',I3,' HOUR',F8.4,
     . ' JULIAN DATE',F10.4,', NEXT UPDATE ON JULIAN DAY '
     . ,F10.4,/)
 3000 FORMAT(' CLOUD COVER ', F8.3,' WIND SPEED= ',F8.2,' DRY BULB ',
     . F8.2,' WET BULB TEMP = ', F8.2, ' ATMOSPHERIC PRESSURE = ',F8.2)
 3010 FORMAT(/,'     **LATERAL INFLOW CONDITIONS UPDATED AT: YEAR ',
     . I6,' MONTH ',I3,' DAY ',I3,' HOUR',F8.4,
     . ' JULIAN DATE',F10.4,', NEXT UPDATE ON JULIAN DAY ',F10.4,/)
 3020 FORMAT(20X,' REACH =   ',150I10)
 3030 FORMAT('CONST. NO.',I5,' NEW VALUE =    ',150F10.4)
      RETURN
      END
```

APPENDIX C: FORTRAN VARIABLES

A	Cross-section area (L^2)
AA	The coefficient matrix
ACLOCK	The 24-hr clock time (T)
ADN	An empirical coefficient relating to the reactivity and diffusivity of nitrate in the benthal boundary layer ($T^{-BK}L^{-3BK}$)
AG	An empirical coefficient in the general equation for K2 ($T^{E1-2}L^{E2-E1}$)
AKC	An empirical coefficient relating the reactivity and diffusivity of the free variable in the benthal boundary layer ($T^{BK-1}L^{-3BK}$)
AKN	An empirical coefficient relating the reactivity and diffusivity of the ammonia variable in the benthal boundary layer ($T^{BK-1}L^{-3BK}$)
AKNX	An empirical coefficient relating the absorptivity and diffusivity of ammonia in the benthal boundary layer ($T^{BK-1}L^{-3BK}$)
AK1	An empirical coefficient relating the reactivity and diffusivity of carbonaceous biochemical oxygen demand ($T^{BK-1}L^{-3BK}$)
ALABEL	The names of each of the eight modeled species
ALGADK	The rate of algal decay ($MT^{-1}L^{-3}$)
ALGAEB	The benthal concentration of algae (ML^{-2})
ALGRO	The algal growth rate coefficient ($ML^{-3}T^{-1}$)
ALGO	Lumped algal decay rate coefficient ($ML^{-2}T^{-1}$)
ALG1	Lumped algal growth rate coefficient ($ML^{-2}T^{-1}$)
ANCONT	Nitrogen-to-biomass ratio in algae and macrophytes
APCONT	Phosphorus-to-biomass ratio in algae and macrophytes
APO4	An empirical coefficient relating the absorptivity and diffusivity of phosphate in the benthal boundary layer ($T^{BK-1}L^{-3BK}$)
AS	Cross-sectional area averaged over the characteristic line (L^2)
ATB	The rate coefficient for bottom heat loss
ATS	The rate coefficient for surface heat loss
AUNIT	The units of each of the eight modeled species
AO	RIV1H: A(I), RIV1Q: Area at the previous time-step
A1	RIV1H: A(I+1), RIV1Q: Cubic interpolation coefficient for concentrations
A2	Cubic interpolation coefficient for concentration
A3	Cubic interpolation coefficient for concentration
A4	Cubic interpolation coefficient for concentration
B	Channel top-width (L). TRIDAG: The main diagonal
BC	Boundary conditions
BCD	Downstream boundary conditions

Note: Appendix C may not contain all of the FORTRAN variables.

BCU	Upstream boundary conditions
BETA	Momentum correction factor. TRIDAG: Workspace array
BK	An empirical coefficient reflecting the thickness of the benthal boundary layer
BLABEL	Array of labels, packed for output
BOUND	Boundary conditions storage array
BTD	Boundary condition type downstream
BTU	Boundary condition type upstream
BUNIT	Array of units, packed for output
BO	RIV1H: Provisional estimate of B. RIV1Q: B at the previous time-step (L)
B1	Cubic interpolation coefficient for spatial derivative of the concentration
B2	Cubic interpolation coefficient for spatial derivative of the concentration
B3	Cubic interpolation coefficient for spatial derivative of the concentration
B4	Cubic interpolation coefficient for spatial derivative of the concentration
C	RIV1H: The vector C_i of the matrix solution procedure RIV1Q: The concentrations of each of the eight modeled species (ML^{-3}). TRIDAG: The upper codiagonal
CBOD	Ultimate first-stage (carbonaceous) BOD, g/m^3
CBODL	The concentration of carbonaceous biochemical oxygen demand in the lateral inflow (ML^{-3})
CBODSR	Rate coefficient for CBOD removal by settling, day^{-1}
CCL	The difference between concentration in the lateral inflow and the stream (ML^{-3})
CL	The concentrations of each of the eight modeled species in the lateral inflow (ML^{-3})
CLABEL	The label assigned to the free variable
CLOCK	The clock time in fraction of a day (T)
CN	Lumped scalar equivalent of CN1(I)
CN1	Modified Manning's coefficient
COEF	Coefficient in the rating curve
COL	The concentration of the free variable in the lateral inflow (ML^{-3})
·CONST	The name of the namelist
CONVRT	The conversion factor for customary to SI units
COSP	The cosine of the junction angle

CP	The presence/absence of each of the eight modeled species
CSINK	The sink term for the free variable $(ML^{-3}T^{-1})$
CUNIT	The units of the free variable
CO	The scalar equivalent of C(I) (ML^{-3})
COO	The concentration just upstream of the junction node (ML^{-3})
C1	Coefficient in the stage-area formula
C2	Coefficient in the stage-area formula
C3	Coefficient in the stage-area formula
D	RIV1H: A/B. MAT5: Gauss elimination factor
	RIV1Q: Tributary dilution ratio
DADX	Spatial derivative of area (L)
DAMK	Empirical coefficient in the formula for reaeration through control structures
DAMKO	Scalar equivalent of DAMK
DARK	Logical flag for whether the current time-step is outside of the daylight hours
DATE	The date of the program's execution
DAWN	Time of sunrise (T)
DBCA	Derivative of the rating curve with respect to A
DBCQ	Derivative of the rating curve with respect to Q
DBDH	Derivative of B with respect to H
DC	Spatial derivative of the concentration (M^3L^{-4})
DCOO	Spatial derivative of the concentration just upstream of the junction node (M^3L^{-4})
DD	Dilution ratio of the tributary's spatial derivative
DE	Elevation head loss over a reach (L)
DEN	Lumped denominator
DFA	Derivative of the residual of the continuity equation with respect to A(I)
DFA1	Derivative of the residual of the continuity equation with respect to A(I+1)
DFQ	Derivative of the residual of the continuity equation with respect to Q(I)
DFQ1	Derivative of the residual of the continuity equation with respect to Q(I+1)
DGA	Derivative of the residual of the momentum equation with respect to A(I)
DGA1	Derivative of the residual of the momentum equation with respect to A(I+1)

DGQ	Derivative of the residual of the momentum equation with respect to Q(I)
DGQ1	Derivative of the residual of the momentum equation with respect to Q(I+1)
DJ	The main storage array
DK	The spatial derivative of the decay rates $(T^{-1}L^{-1})$
DOL	Dissolved oxygen concentration in the lateral inflow (ML^{-3})
DOSAT	Dissolved oxygen saturation value (ML^{-3})
DOSAVE	Dissolved oxygen concentration just upstream of a control structure (ML^{-3})
DOX	Provisional estimate of dissolved oxygen concentration (ML^{-3})
DQLA	Spatial derivative of the lateral inflow per unit area $(T^{-1}L^{-1})$
DSINK	Spatial derivative of the source/sink terms $(T^{-1}L^{-1})$
DT	Time increment (T)
DUDX	Spatial derivative of velocity (T^{-1})
DX	Reach length (L)
DX1	Reach length (L)
D0	A distance equivalent of the time increment, used for estimating a derivative at the boundary (L)
D1	RIV1H: THETA*DT(T), RIVIQ: DX(1) (L)
D2	DWPMN: 2*D1/DX1(I) (TL^{-1}), RIV1Q: DX*(2) (L)
E	Fraction of the reach length above the node at which the characteristic line originated
EC	The complement of E; i.e., 1-E
ELO	Water surface elevation (L)
ELAPSE	Total elapsed simulation time (T)
ELO	Water surface elevation at the previous time-step (L)
EXPO	Exponent in the rating curve formula
E1	An empirical coefficient for velocity in the general equation for k2
E2	An empirical coefficient for mean depth in the general equation for k2
F	A workspace array
FEEDS	FEEDS(I) is the identification number of the segment that segment I feeds into
FLIP1	A logical flag which indicates that the first two columns of the coefficient matrix have been pivoted
FLIP2	A logical flag which indicates that the last two columns of the coefficient matrix have been pivoted
FLOW	A scalar assigned the literal value 'Q' for scanning the input deck
FMT	The output format array

FMT1	The format specification F14.1
FMT2	The format specification F14.2
GAMMA	TRIDAG: A workspace array
GR	Acceleration due to gravity (LT^{-2})
GO	A correction factor for the spatial derivative due to accelerating flow.
G2	GR/2 (LT^{-2})
H	RIV1H: depth of flow above the channel bottom (L), RIV1Q: average depth of flow (L)
HYDRO	The output array for hydrodynamic data
HYDRO1	The input array for hydrodynamic data at odd time-steps
HYDRO2	The input array for hydrodynamic data at even time-steps
HO	RIV1H: a provisional estimate of H (L), RIV1Q: H at the previous time-step (L)
H1	A provisional estimate of H (L)
I	An index (usually for nodes)
IA	An index for the coefficient matrix
IBC	An index for boundary conditions
IBCL	IBC(L)
IBRANCH	Segment branch number
ICYCLE	An index for sensitivity analysis
ID	Segment identification number
IDAM	Control structure type
IDAMO	A scalar equivalent of IDAM
IDAY	The day number of the simulation
IDLL	ID(LL)
IDO	A scalar equivalent of ID
IEHOUR	The hours component of the 24-hr clock time
IEMIN	The minutes component of the 24-hr clock time
II	An index
IND1	The total number of nodes in the system
IND2	RIV1H: The array space needed for AA, RIV1Q: The number of time intervals
IND3	RIV1H: The array space needed for C and R; RIV1Q: The total possible number of boundary condition values
IND4	The array space required for HYDRO, HYDRO1, and HYDRO2
IND5	The array space required for each component of HYDRO1 and HYDRO2
INIT	The array of initial conditions (ML^{-3})

IPRINT	The print interval
IR	An index for the array R
ISURMX	The maximum light intensity at the water surface (MT^{-3})
IT	Indexes tributaries
ITIME	Time-step number
ITO	ITO(L) is the index of the T-array at which segment L deposits its data
IT1	IT1(L) is the index of the T-array at which the first tributary to segment L deposits its data
IT2	IT2(L) is the index of the T-array at which the last tributary to segment L deposits its data
I3	The integer equivalent of C3
J	An index (usually for time)
JBC	Segment L will find its boundary conditions at index JBC(L)
JBCD	Segment L will find its downstream boundary conditions at index JBCD(L)
JBCU	Segment L will find its upstream boundary conditions at index JBCU(L)
JJ	An index
JNODE	Junction node
JT	An array of junction nodes, packed in the same manner as T
JUNCT	A flag for whether the current node is a junction node
K	RIV1H: An index; RIV1Q: Decay rates for each of the eight modeled species (T^{-1})
KALGDK	The specific algal decay rate (T^{-1})
KALGRO	The specific algal growth rate (T^2M^{-1})
KC	The decay rate for the free variable (T^{-1})
KDN	The denitrification rate (T^{-1})
KEXT	The light extinction coefficient (L^{-1})
KE1	Channel constriction energy loss coefficient
KLITE	Half-velocity constant for light intensity, watt/m^2
KN	The rate of ammonia decay due to nitrification (T^{-1})
KNCBDN	The Monod half-velocity constant for nitrate- inhibited denitrification (ML^{-3})
KNPOOL	Half-velocity constant relating inorganic nitrogen to algal growth as per QUAL2E
KNX	The rate of ammonia decay due to sediment sorption (T^{-1})
KOALDK	The Monod half-velocity constant for oxygen limitation of algal decay (ML^{-3})
KOCBDN	The Monod half-velocity constant for oxygen inhibition of denitrification (ML^{-3})

KOCB1   The Monod half-velocity constant for oxygen limitation of
           carbonaceous biochemical oxygen demand decay (ML^{-3})

KON   The Monod half-velocity constant for oxygen limitation of
           denitrification (ML^{-3})

KPO4   Half-velocity constant relating phosphate concentration to algal
           growth rate

KPO4DK   The phosphate decay rate (T^{-1})

KS   The average K-rate across the characteristic line (T^{-1})

KTB   The bottom-water heat exchange rate (T^{-1})

KTS   The air-water heat exchange rate (T^{-1})

K1   The CBOD decay rate (T^{-1})

K2   The reaeration rate (T^{-1})

L   An index for segments

LAMBDA   The duration of daylight (T)

LAST   The index of the last item interchanged

LIB   The set of boundary conditions for each segment

LIMIT   The index of the last item to be inspected

LJ   An index

LL   An index

LO   An index

L1   An index

M   An index

MACROB   Macrophyte density on the bottom, g/m^2

MAX   The number of species actually modeled out of a total of eight
           possible

MBC   Used in indexing boundary conditions

MBOUND   The number of boundary conditions

MC   The index of species

MDEATH   Macrophyte death rate, g/m^2 day^{-1}

MGRATE   Macrophyte growth rate, g/m^2 day^{-1}

MJ   An index

MM   An index of labels and species

MNODE   The number of nodes

MP   The number of reaches

MTIME   The number of time-steps

MO   An index

M1   The lower loop bound

M2   The upper loop bound

N	An index of species
NBC	The number of boundary conditions at each time-step
NDEPLO	The first node at which the nitrogen pool is depleted
NDEPL1	The last node at which the nitrogen pool is depleted
NH3NL	The concentration of ammonia in the lateral inflow (ML^{-3})
NODE1	The index of the first node in each segment
NOPO40	The first node at which the phosphate is depleted
NOPO41	The last node at which the phosphate is depleted
NO3NL	The concentration of nitrate in the lateral inflow (ML^{-3})
NO3NX	A provisional estimate of the nitrate concentration (ML^{-3})
NPOOL	The total inorganic nitrogen concentration--the sum of nitrate and ammonia (ML^{-3})
NS	The number of segments in the system
N1 through NS38	Locations within the main storage array
OFEDEC	Oxygen-to-iron ratio for iron oxidation
OMNDEC	Oxygen-to-manganese ratio for oxidation
ONEQUI	Incremental increase in oxygen-to-algal biomass ratio for oxygen production by algae and macrophytes when nitrate is used as a nitrogen source
ONITRI	Oxygen-to-nitrogen ratio for ammonia oxidation
OPDECY	Oxygen-to-biomass ratio for oxygen production by algae and macrophytes when ammonia is the nitrogen source
ORDER	A collection of segment numbers in an upstream order
ORGANL	The concentration of organic nitrogen in the lateral inflow
P	The fraction of the nitrogen pool composed of nitrate
PARM	A collection of parameters
PI	π, 3.14159
PO4L	The concentration of phosphate in the lateral inflow (ML^{-3})
PO4X	A provisional estimate of phosphate concentration (ML^{-3})
P0	A lumped friction loss term for node $I(L^{-1/3}/T^{-1})$
P1	A lumped friction loss term for node $I+1(L^{-1/3}/T^{-1})$
Q	The stream flow (L^3T^{-1})
QL	Lateral inflow per unit of stream length (L^2T^{-1})
QLA	Lateral inflow per unit of stream volume (T^{-1})
QX	Q $(L^{3BK}T^{-BK})$
Q0	RIV1H: Q(I); RIV1Q: Q at the previous time-step
Q1	Q(I+1)

R ratio	RIV1H: The residuals from the governing equations; RIV1Q: The dissolved oxygen deficit
RCURVE	This is assigned a literal value of "R" and is used for scanning the input
RMILE	River mile (L)
RMILEO	The river mile of the downstream terminus of the system (L)
RMSA	The root mean square of the initial cross-sectional areas at every node times the tolerance
RMSQ	The root mean square of the initial flow at every node times the tolerance
SAVE	A temporary storage location used in column pivoting
SINI	The sine of the incident light angle
SINK	The source/sink term for each of the eight modeled species ($ML^{-3}T^{-1}$)
SINKS	The average source/sink value across the characteristic line ($ML^{-3}T^{-1}$)
SNAME	The segment name
SNAMEO	The segment name
START	The start time of the simulation (T)
SUNSET	The time of sunset (T)
T	The tributary data transfer array
TAMMON	Temperature coefficient for ammonia oxidation
TBIO	The temperature correction factor for biochemical processes
TEMP	The temperature (°C)
TEMPL	The temperature of the lateral inflow (°C)
TEQ	The equilibrium temperature (°C)
THETA	The weighting factor
TIME	The time of the program's execution
TITLE	The title of the program run
TNH3	The temperature correction factor for nitrification
TOLER	The relative tolerance criterion for the Newton-Raphson procedure
TPHYS	The temperature correction factor for physical processes
TRIDAG	The lower codiagonal
TSINK	The source/sink term for temperature (°C T^{-1})
TSIV	An empirical coefficient in the Tsivoglou-Wallace reaeration equation (L^{-1})
ULL	The velocity at node I-1 of the previous time-step (LT^{-1})
ULR	The velocity at node I of the previous time-step (LT^{-1})
US	The average velocity across the characteristic line (LT^{-1})
UUR	The velocity at node I of the current time-step (LT^{-1})

V	TRIDAG: The solution vector
XC	The previous time-step components of the continuity equation (L^2)
XM	The previous time-step components of the momentum equation (L^3T^{-1})
Z	The channel bed elevation (L)

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of Information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1995	Final report

4. TITLE AND SUBTITLE
CE-QUAL-RIV1: A Dynamic, One-Dimensional (Longitudinal) Water Quality Model for Streams, User's Manual

5. FUNDING NUMBERS
CWIS 31595

6. AUTHOR(S)

Environmental Laboratory

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
USAEWES
Environmental Laboratory
3609 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER
Instruction Report
EL-95-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Corps of Engineers
Washington, DC 20314-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A dynamic, one-dimensional (longitudinal), water quality model for unsteady flows in rivers and streams, CE-QUAL-RIV1, is presented.

CE-QUAL-RIV1 is developed in two parts, hydrodynamic and water quality. Output from the hydrodynamic solution is used to drive the water quality model.

The hydrodynamic code uses a four-point implicit Newton-Raphson procedure to solve the nonlinear St. Venant equation. Numerical accuracy for the advection of sharp gradients is preserved in the water quality code through the use of the explicit two-point, fourth-order accurate, Holly-Preissmann scheme.

Water quality constituents include temperature, carbonaceous biochemical oxygen demand, organic nitrogen, ammonia nitrogen, nitrate nitrogen, orthophosphate phosphorus, dissolved manganese, dissolved iron, dissolved oxygen, coliform bacteria, and algae. The effects of macrophytes are also included.

(Continued)

14. SUBJECT TERMS
Dynamic
One-dimensional
Streams
Water quality model
Riverine

15. NUMBER OF PAGES
290

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED		

13. (Concluded).

The model allows simulation of branched river systems with multiple hydraulic control structures, such as run-of-the river dams, waterway locks and dams, and reregulation dams. The model was developed to simulate the transient water quality conditions associated with highly unsteady flows that can occur on regulated and unregulated streams.