GATE STROKING

Engineering and Research Center
Bureau of Reclamation

July 1979
Gate stroking consists of a series of continuous or discontinuous gate motions which produce a desired water surface profile in a canal. The first obvious application of gate stroking within the Bureau was on the Granite Reef Aqueduct, Central Arizona Project. The mathematical development of the technique is outlined and methods for treating the unique conditions found on Bureau aqueducts are presented. Sufficient computer documentation is provided to permit application of the program to other aqueducts. The technique can be applied either to the entire aqueduct or to the component reaches. If applied to the aqueduct, the output consists of both gate and pump schedules. When applied to reaches only, the set of gate schedules is produced.
ACKNOWLEDGEMENTS

Funds for this study were made available from the Central Arizona Project through coordination efforts by the E&R Center Automation Team. The careful review of this report and the continuous support by engineers in the Operations Division, Central Arizona Project is gratefully acknowledged.

The information contained in this report regarding commercial products or firms may not be used for advertising or promotional purposes and is not to be construed as an endorsement of any product or firm by the Bureau of Reclamation.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Summary and conclusions</td>
<td>1</td>
</tr>
<tr>
<td>Analytical development</td>
<td>2</td>
</tr>
<tr>
<td>Basic equations</td>
<td>2</td>
</tr>
<tr>
<td>Numerical method</td>
<td>5</td>
</tr>
<tr>
<td>Boundary conditions</td>
<td>5</td>
</tr>
<tr>
<td>Reach versus aqueduct computation</td>
<td>5</td>
</tr>
<tr>
<td>Changes in prism cross section</td>
<td>6</td>
</tr>
<tr>
<td>Turnouts</td>
<td>7</td>
</tr>
<tr>
<td>Gates</td>
<td>7</td>
</tr>
<tr>
<td>Siphons</td>
<td>8</td>
</tr>
<tr>
<td>Specifications constraints</td>
<td>9</td>
</tr>
<tr>
<td>Method of utilization</td>
<td>10</td>
</tr>
<tr>
<td>Validity of the program</td>
<td>11</td>
</tr>
<tr>
<td>Program description</td>
<td>11</td>
</tr>
<tr>
<td>Background</td>
<td>11</td>
</tr>
<tr>
<td>Problem formulation</td>
<td>11</td>
</tr>
<tr>
<td>Initial conditions</td>
<td>12</td>
</tr>
<tr>
<td>Boundary conditions</td>
<td>12</td>
</tr>
<tr>
<td>Depth schedules</td>
<td>12</td>
</tr>
<tr>
<td>Discharge schedules</td>
<td>12</td>
</tr>
<tr>
<td>Turnout schedules</td>
<td>12</td>
</tr>
<tr>
<td>Pumps</td>
<td>12</td>
</tr>
<tr>
<td>Gate positions</td>
<td>13</td>
</tr>
<tr>
<td>Inverted siphons</td>
<td>13</td>
</tr>
<tr>
<td>Boundaries where no control structure is present</td>
<td>13</td>
</tr>
<tr>
<td>Physical descriptors</td>
<td>13</td>
</tr>
<tr>
<td>Modular construction</td>
<td>13</td>
</tr>
<tr>
<td>Output</td>
<td>13</td>
</tr>
<tr>
<td>Development and structure of the program</td>
<td>13</td>
</tr>
<tr>
<td>Program details</td>
<td>14</td>
</tr>
<tr>
<td>Bibliography</td>
<td>15</td>
</tr>
</tbody>
</table>

### APPENDIX

Gate stroking computer program listing                           | 17   |
### FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Characteristic lines</td>
</tr>
<tr>
<td>2</td>
<td>Computational regions</td>
</tr>
<tr>
<td>3</td>
<td>Region 1</td>
</tr>
<tr>
<td>4</td>
<td>Region 2</td>
</tr>
<tr>
<td>5</td>
<td>Region 3</td>
</tr>
<tr>
<td>6</td>
<td>Interpolation scheme</td>
</tr>
<tr>
<td>7</td>
<td>Region 4</td>
</tr>
<tr>
<td>8</td>
<td>Region 5</td>
</tr>
<tr>
<td>9</td>
<td>Dead-band operation with a pump</td>
</tr>
<tr>
<td>10</td>
<td>Turnouts and change of cross-sectional area</td>
</tr>
<tr>
<td>11</td>
<td>Computational method across pool boundaries</td>
</tr>
<tr>
<td>12</td>
<td>Gate definition sketch</td>
</tr>
<tr>
<td>13</td>
<td>Coefficient of free and submerged Tainter-gate discharge for $r/a = 1.5$</td>
</tr>
<tr>
<td>14</td>
<td>Discharge coefficient as a function of submergence</td>
</tr>
<tr>
<td>15</td>
<td>Dead-band operation of gates</td>
</tr>
<tr>
<td>16</td>
<td>Typical siphon layout</td>
</tr>
<tr>
<td>17</td>
<td>Loss coefficients</td>
</tr>
</tbody>
</table>
INTRODUCTION

The change from one discharge to another in open-channel flow always creates a disturbance in the water-surface elevation. The magnitude of the disturbance is related to the manner in which the change is accomplished. In some cases, the disturbances persist for long periods with large amplitudes. In 1969, Wylie [1] developed a method to control the disturbances within certain limits. The objective of the method was to produce a predetermined variation in the water-surface elevation at one location in a canal pool by properly varying the boundary conditions at each end of the pool.

Since pool boundaries are usually gates, the technique of water-surface control is called "gate stroking." The term comes from a similar procedure used in closed conduits known as "valve stroking." Although the term "stroking" as used in this context cannot be found in the dictionary, its definition can be implied from the several meanings commonly listed. Thus, stroking is "any of a series of continuous or discontinuous efforts to do, produce, or accomplish something, especially a successful result." Based upon its usage and the general meaning of stroking, gate stroking can be defined as, "a continuous or series of discontinuous gate motions which produce a predetermined water-surface variation in a canal."

Since the original paper by Wylie, the number of gate stroking applications has been minimal. O'Loughlin [2] and Gientke [3] are two of the few who have referred to application of gate stroking for specific installations. Perhaps one of the reasons there are not more applications is that gate stroking requires a scheduled type of operation with centralized control. The number of systems meeting this requirement is rather limited.

The first obvious application of gate stroking within the Bureau was on the Granite Reef Aqueduct, Central Arizona Project. This aqueduct consists of several canal reaches separated by pumping stations. The aqueduct is intended to be controlled by a computer-assisted remote-control system. It anticipated that all delivery schedules can be reasonably estimated several days in advance of the actual need for the water.

In addition, the aqueduct is designed without any wasteways or reregulating reservoirs. When put into operation, the canal is to be operated at nearly the design capacity. Thus, there is very little margin for errors and the control of transients is a significant factor. These considerations coupled with a desire to minimize onpeak pumping costs led to the development of the gate stroking technique within the Bureau.

Wylie's original concept was followed very closely in the development which follows; however, provision had to be made to include structures like turnouts, siphons, and free flowing tunnels in the Bureau's computer program. In addition, a technique had to be developed in which the initial conditions were not always at steady state.

The purpose of this report is to outline the mathematical development; present the methods used in handling the unique conditions found on Bureau aqueducts; illustrate how the gate stroking is utilized; and provide sufficient documentation for the computer program, appendix A, so that it can be applied to other aqueducts.

SUMMARY AND CONCLUSIONS

Gate stroking is a series of continuous or discontinuous gate motions which produce a desired water-surface profile in a canal. The first obvious application of gate stroking within the Bureau was on the Granite Reef Aqueduct, Central Arizona Project.

The mathematical development of the technique is outlined and methods for treating the unique conditions found on Bureau aqueducts are presented. Sufficient computer documentation is provided to permit application of the program to other aqueducts.

The technique can be applied either to the entire aqueduct or to the component reaches. If applied to the aqueduct, the output consists of both gate and pump schedules. When applied to reaches only the set of gate schedules is produced.

The most nebulous relationship in the technique is the gate discharge equation. Additional research to better define the gate discharge coefficients is required.

* Numbers in brackets refer to literature cited in the bibliography.
ANALYTICAL DEVELOPMENT

Basic Equations

Any computational scheme which calculates the unsteady water-surface profiles can be used with the concept of gate stroking. The method chosen by Wylie is the method of characteristics. Two computational schemes are used with the method of characteristics. These are (a) the grid of characteristics and (b) the method of specified time increments. The grid of characteristics method was chosen for use in the computer program. This method is more accurate because interpolation at the interior of the grid is avoided. Since the computations for each segment are performed independently of the other segments, the grid does not have to intersect specified points on the boundary. Thus, the usual problem of joining computations at a boundary, when using the grid of characteristics, does not arise in the gate stroking application.

The equations of motion and continuity in an open channel can be expressed as four particular total differential equations. These are:

\[ \sqrt{\frac{gT}{A}} \frac{dy}{dt} + \frac{1}{A} \frac{dQ}{dt} + g (S-S_0) = 0 \]  
\[ \frac{dx}{dt} = \frac{Q}{A} + \sqrt{\frac{gA}{T}} \]  

\[ \sqrt{\frac{gT}{A}} \frac{dy}{dt} + \frac{1}{A} \frac{dQ}{dt} + g (S-S_0) = 0 \]  
\[ \frac{dx}{dt} = \frac{Q}{A} - \sqrt{\frac{gA}{T}} \]

where \( g \) = acceleration of gravity  
\( T \) = top width of water prism  
\( A \) = cross-sectional area of water prism  
\( y \) = flow depth  
\( t \) = time  
\( Q \) = discharge at section  
\( S \) = friction slope  
\( S_0 \) = bottom slope  
\( x \) = horizontal distance

The first two equations are valid along the positive characteristic \( C^+ \), figure 1. The last two are valid along the negative characteristic.

Using the notation of Wylie, the intersection point \( P \) presents a location at which the solution of the variables \( x, y, Q, \) and \( t \) is theoretically possible. If conditions are known simultaneously at either of the \( R \) and at either of the \( S \) points, then a numerical integration of the four characteristic equations will result in a solution of the variables at \( P \).

Several computational schemes are used in the computer program. Each of these is dependent upon which combination of points \( R \) and \( S \) is known. The schemes can be related to specific regions in the \( x-t \) plane, figure 2.
Each of the computational regions requires specific information from at least one of its borders. Each computational scheme then conveys information to the other borders. In general, borders occur along any time-distance line which borders a region, such as A-B, B-C, C-A, figure 3. Certain borders, however, have special significance. These occur at specific locations where depth variations are known or desired, and are known as boundaries. The canal segment between boundaries is called a "pool." The pool boundaries may or may not coincide with hydraulic barriers, such as gates. The canal segment between boundaries at which discharges are specified, is known as a "reach." Several pools may be contained within a reach.

The sequence of the computations for each of the regions which comprise a pool is described in the following paragraphs.

Region 1, figure 3.—The initial conditions, that is, depths and velocities along the entire length of the pool, must be defined at time equal zero. The variables at all points within the region can be determined by successively extending the grid forward in time. This region is also known as the domain of dependence since conditions within the region are uniquely determined by the conditions specified on one boundary. When the computations are completed, the value of the four variables will have been determined along the border A-B-C.

Region 2, figure 4.—In this region, conditions are known along the border A-B. At the point P, the value of the variable x is also known. However, to solve for the conditions at P using the positive characteristic, one additional variable must be specified. If the downstream boundary is a pump or a delivery point, the discharge will be specified. As will be seen later, if the boundary is a gate, the discharge will also be specified to maintain continuity. Therefore, along the boundary A-C, the discharge must be specified, whereas, time and depth will be determined. Interior points P, are computed from known values of R, and S, as was done in region 1. Upon completion of the computations, the values of all variables are known along the border B-C-A.

Region 3, figure 5.—This region is also a domain of dependence since the point P is determined from conditions given at the downstream boundary. Thus, along the boundary A-D, it is necessary to specify all four of the variables. To be compatible with the solution from region 2, the values of the four variables at point A must be identical with those at point C in region 2. The other boundary values along the line A-D are determined from the prescribed discharge and
water level schedules. When the computations are finished, the conditions are known along the borders A-B and C-D. To determine the conditions along B-C an interpolation is necessary. The interpolation scheme used involves a linear interpolation between points on the positive characteristics which cross the boundary (A to B, fig. 6). This is different than the scheme used by Wylie [1] who interpolated between the points of the diamond shape formed by the grid which crossed the boundary (C to D, fig. 6). Neither method seems to have any particular advantage over the other.

The computations are extended in time, as required, (region 3B), by the same computational scheme used in region 3, figure 2. Region 3B has no theoretical significance. It is merely a device for extending the computations forward to an arbitrary time t without requiring additional computer memory when t is large.

Region 4, figure 7.—According to Wylie [1], this region is known as the domain of influence since it is influenced by the values of the conditions at point A. The computations proceed from known values on the borders A-B and C-A, when the computations have been completed, conditions along the boundary C-B will have been determined.

If the discharge at the upstream end of the most upstream pool is specified, then the depth in that pool cannot be specified. In this case, a fifth computational procedure has been provided, figure 8.

Region 5, figure 8.—In this region, the discharge is specified along boundaries A-D and B-C. The computations proceed in a fashion similar to those in region 3. However, at the downstream boundary B-C, distance and discharge as a function of time are known. This computational procedure is required only when the upstream boundary discharge is scheduled.
Numerical Method

The four characteristic equations were integrated using the trapezoidal rule. The trapezoidal rule is given by

\[ \int_{x_0}^{x_1} Y \, dX = \frac{h}{2} (Y_0 + 2Y_1 + 2Y_2 + \ldots + 2Y_{n-1} + Y_n) \]

where \( h = X = X_1 - X_0 \)

If \( n = 1 \), then

\[ \int_{x_0}^{x_1} Y \, dX = \frac{h}{2} (Y_0 + Y_1) \]

Applying this rule to the characteristic equations gives

\[ \frac{g}{2} \left( \frac{1}{C_R} + \frac{1}{C_P} \right) (Y_p - Y_R) + (Y_p - Y_R) + \]

\[ \frac{g}{2} (S_R + S_p - 2S_0) (t_p - t_R) = 0 \]  

(5)

(6)

(7)

(8)

(9)

(10)

BOUNDARY CONDITIONS

Reach Versus Aqueduct Computation

The Granite Reef Aqueduct consists of several reaches separated by pumping stations. Each reach consists of several pools which are separated by control gates. Two methods of applying the gate stroking technique are possible. For one, the water level variations over an operating cycle within each reach are specified. In addition, the operation at each pumping station is specified. The other technique involves specifying water level changes at each pool in the entire aqueduct. Pumps are treated like gates having dead bands, figure 9. This type of operation could properly be called "pump stroking."

The difference between reach and aqueduct computations is mainly one of operating philosophy. If a pump and water level schedule are both specified, then the water level in at least one pool in a reach cannot be specified. That is, the

\[ C = \sqrt{\frac{gA}{T}} \]  

(11)

Figure 9.-Dead-band operation with a pump.
level in one pool must be allowed to seek its own level or "float." On the other hand, if only the water level variations and delivery schedules are specified, then the necessary pump schedules to achieve this can be uniquely determined. For the first case, the pump schedules are of primary importance and water level fluctuations in at least one pool must be tolerated. In the second case, water level fluctuations are controlled everywhere and the pumping schedule is the flexible component.

For a floating water level, the computations pass from a predictive to an analysis stage. The discharges at each end of the pool are specified and the water levels at the boundaries are calculated, figure 8. This computer program fails when there is a very small flow into the pool. As the water level drops, the wave celerity approaches zero. Since the reciprocal of the celerity is used in the method of characteristics, the range of permissible values for the variable may be exceeded in the computer with small flow depths.

Changes in Prism Cross Section

In the derivation of the characteristics method, it was assumed that the section was prismoidal in each pool. This assumption does not hold true if a change in cross-sectional shape occurs in the pool. For large changes in shape, the characteristic line which crosses the change can deviate greatly from a straight line (point A to point C, fig. 10A).

The exact solution of this case requires two additional equations for the characteristic which crosses the change in section. These are (1) an equation which defines the distance between the known point (A or B) and the change in section, and (2) an energy equation across the change in section. This requires that every step in the computational matrix of pools with changes in section be checked to determine whether a crossing has occurred. If one has occurred, then a set of six simultaneous equations must be solved using an iterative scheme. The iterations are necessary since the area, top width, and hydraulic radius at the unknown point C are functions of the unknown depth at that point.

If the cross-sectional changes are not too abrupt, the differences in the slopes of the characteristic at the change can be ignored. This approximation requires the solution of five simultaneous equations. Studies have not been performed to determine what constitutes a large change in section.

As an alternate approach, the pool can be divided into subpools. The computations in each subpool proceed as if each were a pool. The computations begin at the downstream subpool in which the depth must be assumed along the region 3 boundary, (fig. 10B). As a consequence, both the depth and discharge are determined just downstream of the change in cross section. If continuity of flow across the change is maintained, then application of the energy equation will yield the depths on the upstream side of the

A. Actual Characteristic Grid

B. Simplified Computational Scheme

Figure 10.-Turnouts and change of cross-sectional area.
transition. These depths can be compared with the depths resulting from the upstream subpool computations of region 2. If the differences between the two are within acceptable limits, the depth variation assumed at the downstream pool boundary is acceptable. If the two are outside of acceptable limits, the downstream pool boundary depths must be respecified. Generally, rapid changes in water depths do not occur; therefore, iteration has not proven to be necessary.

The largest errors with this procedure develop at the last computation in region 2 of the upstream subpool, (fig. 10B). The effect of errors along the timeline O-A can be minimized by a relatively simple procedure. Using continuity of flow at the intermediate boundary, the flow depths are determined in the upstream subpool region 2. Then, using the energy equation across the intermediate boundary with known values of flow and depth from region 4 (and in some cases, region 3) a slightly different upstream flow depth is determined at the intermediate boundary in region 2. This depth based on energy considerations is substituted for the depth based on continuity considerations. A linear interpolation is required at the transition, (fig. 11), since the grid points from the downstream subpool regions 3 and 4 do not match with those from the upstream subpool region 2.

With the present program it has been assumed that the flow passes from one cross section to the other with no loss of energy. However, the energy equation could include a transition loss if desired.

Turnouts

The concepts developed for changes in cross section also apply to turnouts. In this case, the continuity of flow also includes the proper accounting for the turnout discharge. Theory and experiment indicate that the turnout flow does not affect the energy head [4]. Therefore, energy is assumed constant across the turnout.

Gates

The general form of the equation for flow under a free flowing radial (Tainter) gate can be obtained from Bernoulli’s equation [5]. It is expressed as:

$$Q = C_c b B \sqrt{2g (\Delta y + V_f^2 / 2g)}$$  \hspace{1cm} (12)

where $C_c = \text{Contraction coefficient}$

$b = \text{gate opening}$

$B = \text{gate width}$

$\Delta y = \text{difference between upstream and downstream water depths}$

$V_f = \text{upstream mean velocity}$

$g = \text{acceleration of gravity}$

If the downstream depth exceeds the gate opening, the flow is said to be submerged. For this case, the equation must be modified to include losses that occur on the downstream side of the gate. It has been argued that the parameters $C_c$ and $V_f^2 / 2g$ are governed by several linear terms which are characterized by the flow pattern [6]. Thus, it is possible to express the discharge as:

$$Q = C_d b B \sqrt{2g y_f}$$  \hspace{1cm} (13)

where $C_d = \text{Discharge coefficient}$

$y_f = \text{Upstream depth}$

The discharge coefficient is a function of the submergence $y_s/b$, trunnion height $a/b$, and upstream depth $y_f/b$ ratios, (figs. 12 and 13).

The Corps of Engineers (7) uses the following form of the equation for submerged flow:

$$Q = C_s B V_f \sqrt{2g (\Delta y + V_f^2 / 2g)}$$  \hspace{1cm} (14)

where $C_s = \text{Submerged discharge coefficient}$.

With this equation, the discharge coefficient varies linearly on a log-log scale from a value of 0.04 at $y_s/b = 20$ to 0.35 at $y_s/b = 2.5$. The effect of the radius of curvature of the sector...
which forms the gate and the trunnion height has not been systematically investigated for this form of the equation.

The equation used in the computer program is based upon the equation from the Corps of Engineers, using the gate opening instead of the downstream depth.

\[ Q = C B b \sqrt{2g (\Delta y + \frac{V_i^2}{2g})} \]  

(15)

where \( C = \) discharge coefficient  
\( = C_s y_1/b \)

Since the spread in the data is so large (fig. 14) detailed model studies of the specific structures are recommended. With detailed studies, the simpler form of the discharge relationship given by equation (13) should be used.

In current practice, it is impractical for a gate to follow the timewise change in openings that are specified by the gate stroking technique. Instead, a dead-band type of operation is preferred. With this, a gate movement is initiated when the gate stroking predicts excursions which exceed the dead band. If this occurs, the time at which the gate motion should have started is calculated. This time is a function of the gate speed, (fig. 15). Future gate structure installations may include multi- and variable-speed gate motors. With these features the gates could follow the openings specified by the gate stroking technique more closely.

**Siphons**

Normally, siphons are located at the upstream end of a pool with a control gate located on the upstream end of the siphon, (fig. 16).
The passage of a pressure wave through the siphon actually requires a finite length of time. However, if the wave travel time is small relative to the time steps used in the open channel portion, the siphon can be approximated with a lumped parameter. That is, it is only necessary to treat the siphon as an isolated loss which occurs with no time delays. The lumped parameter approximation is valid when

\[
\frac{L}{a} < \Delta T
\]

where \(L\) = siphon length
\(a\) = wave velocity
\(\Delta T\) = computational time increment in open channel portion of conduit.

Normally, siphons are shorter than 2 km and wave speeds are of the order of 1000 m/s. These values indicate that the lumped parameter approximation is good for computational time increments exceeding 20 seconds in which \(\Delta T\) exceeds the \(L/a\) ratio by an order of magnitude.

Ignoring inertial effects, the equation of motion for flow in the siphon is given by

\[
\frac{V^2}{2g} + y_1 + z_1 = \frac{V^2}{2g} + y_2 + z_2 + KO^2
\]

where \(V\) = velocity
\(y\) = flow
\(z\) = distance from datum
\(g\) = local acceleration of gravity
\(K\) = loss coefficient

The subscripts 1 and 2 refer to locations in the canal immediately upstream and downstream of the siphon, respectively. The loss coefficient includes all singular and frictional effects. It can be approximated by

\[
K = \frac{1}{2ga^2} \left( \frac{HL}{D} + C_g + C_e \right)
\]

where \(A\) = cross-sectional area of siphon
\(f\) = Darcy-Weisbach friction factor
\(L\) = siphon length
\(D\) = equivalent hydraulic diameter of siphon = four times hydraulic radius
\(C_g\) = contraction coefficient of entrant flow
\(C_e\) = expansion coefficient of exit flow

The contraction and expansion coefficients can be accurately estimated from the ratio of the siphon area to the cross-sectional area of the respective water prisms using figure 17.

**Specifications Constraints**

With gate stroking, two boundary conditions are unknown (not specified) in each pool. These two unknowns are reduced to one, at boundaries that require a continuity of discharge. Based upon these two premises, it is possible to construct a series of general rules or constraints which define the computational procedures to be followed with gate stroking in canal reaches.
These are as follows:

1. In a reach there must exist at least one pool in which two conditions are specified (depth and discharge). This pool is defined as the pivot pool.

2. The computations start at a pivot pool and progress in both directions (upstream and downstream) away from the pivot pool. If the pivot pool is at the end of a reach, the computations progress in one direction only. The present program does not handle the condition of a pivot pool located other than at the end of a reach.

3. More than one pivot pool can exist in a reach; however, for the entire reach, exactly \( N+1 \) conditions must be specified, where \( N \) = number of pools.

4. A region 5 computation (fig. 8) must be performed when the specified condition is at the far end of the pool from the direction in which the computations are proceeding.

METHOD OF UTILIZATION

The Gate Stroking Model is intended to be interfaced with two other types of programs. The first of these is a scheduling program which determines the desirable water-surface elevations and pumping schedules for the daily operation of an aqueduct. Two types of scheduling programs have been developed for the Central Arizona Project. These are the Constant Volume Model and the Water Power Optimization Model. The Constant Volume Model assumes that some target storage value can be maintained in each aqueduct pool during both onpeak and offpeak periods. The Water Power Optimization Model on the other hand varies the storage value in each pool to optimize the onpeak and offpeak pumping schedules.

The second type of program with which the Gate Stroking Model was intended to interface is an analysis program. The analysis program actually simulates the canal flow. For that reason, it is called the Aqueduct Simulation Model.

Since the Aqueduct Simulation Model contains detailed flow data concerning the aqueduct at
any point in time, it is used to obtain initial conditions for the Gate Stroking Model. Then a schedule of operations is developed for a subsequent time interval based on either the Constant Volume or the Water Power Optimization Model. This schedule of operations consists of changes in deliveries, possible pump operations, and specified water level fluctuations within the pools. Using this information, the Gate Stroking Model determines the gate and, in some cases, the pump schedules for the time interval under consideration. These schedules include both the gate openings and timing for the operation of the gates and pumps. Basically, the changes in flow rates through the pumps and gates should be continuous functions. However, in current practice, all of these changes actually take place in discrete amounts at discrete times. Therefore, it is necessary to determine the effects of the discrete operations through application of the Aqueduct Simulation Model to the time interval being considered. At the end of the interval the entire process is repeated. The scheduled time interval is one day in the case of the Granite Reef Aqueduct.

VALIDITY OF THE PROGRAM

As a check on the validity of the gate stroking program, the program was restructured to act as an analysis-type program. Several comparisons were made between the Aqueduct Simulation Model and the restructured Gate Stroking Model. It was found that the restructured Gate Stroking Model, using the grid of characteristics, converges to the steady-state conditions determined by the standard step method. Conversely, the method of specified time increments, as used by Shand [8], allows the water surface to drift rather than reach steady state. Depth differences from several tenths to over 0.3 m were found in the various comparisons.

The technique for restructuring the Gate Stroking Model was to make an initial estimate of depths downstream of gates, calculate the state variables in the pool upstream of the gate using that estimate, calculate the state variables in the pool downstream of the gate using the upstream values just calculated, recalculate the upstream pool using the new downstream values, etc. This process was continued until the water surface along the length of the canal stayed within some convergence limits from one iteration to the next. The technique is stable and generally requires from two to five iterations for convergence, depending upon the length of time modeled and the number and magnitude of gate movements. Unfortunately, for a canal of any length and for time periods of the order of a day, computer resource use can be significant. Investigations are being continued in an effort to improve the convergence of the technique.

PROGRAM DESCRIPTION

Background

The program (listed in appendix) determines the gate stroking schedules for a series of pools in a canal reach. If pools are separated by pumps, the pump schedules may either be determined or specified. The program is of a general nature and may be used to generate gate and, if desired, pump schedules for virtually any canal. Modifications were made to the program for the purpose of expediting a series of operations studies on Granite Reef and Salt-Gila Aqueducts, and separate versions of the program were used for those studies. The program description below pertains to the general version of the program.

The purpose of the program is to determine gate schedules which will produce desired boundary conditions (that is, depths and discharges) in a canal. The differential equations describing the flow apply to free surface flow in a prismatic canal section. Following Wylie [1], the method of characteristics is used to solve the equations within the interior of a computational segment. Abbott [9] gives an excellent description of the significance of characteristic functions as they pertain to the equations of state of water conveyance systems.

Problem Formulation

The present gate stroking technique has been formulated such that:

1. The computations proceed from one pool to another in an upstream direction.
2. Initial conditions along the entire length of the aqueduct must be specified.
3. At the downstream end of the most downstream pool:
   a. The discharge must be specified for all time.
   b. The depth must be specified for all time greater than one wave travel time in the pool.

11
4. At intermediate pools, the depth must be specified for time greater than one wave travel time in the pool.

5. To obtain a gate schedule for all pools it is necessary to project the most downstream pool depth and discharge schedules at least one wave travel time for the reach into the future. If pump stroking is involved, the most downstream pool schedules must be projected at least one wave travel for the entire length of the aqueduct.

6. The discharge at all intermediate pools is a dependent variable; that is, the discharge cannot be specified.

Initial Conditions

The state of the system must be described completely at some specific time, usually at a zero time reference. The system may be in a steady-state condition (all time derivatives equal to zero) or in an unsteady, or transient, condition. The program will accept initial conditions in any of three formats, individually for each pool. The formats differ only in the number of points at which the initial conditions are specified. The state variables, \( x \) (position along the length of the pool), \( y \) (water depth), and \( Q \) (water discharge quantity), may be specified at zero time at up to 40 equally spaced locations along the length of the pool.

- If \( N = 1 \), the state variables must be specified at either end of the pool. A backwater calculation is then performed by the program to determine the state variables at 11 equally spaced points in the pool.
- If \( 2 \leq N \leq 10 \), linear interpolations of the state variables using the values at \( x \) and \( x_2 \) only are calculated at 11 equally spaced points in the pool.
- If \( 11 \leq N \leq 40 \), the values of the state variables provided by the user are accepted as the initial condition.

No checks are made at boundaries for consistency across boundaries.

Boundary Conditions

Depth schedules.—must be provided for the downstream boundaries of all pools. The program will accept schedules for a 48-hour period. If a depth schedule is not appropriate for a particular pool, that is, the most upstream pool in the reach or a pool with no control structure at its downstream end, the schedule will be ignored by the program. However, schedules must be provided for each and every pool. Depths are to be provided on the hour, and transitions from one depth to another are made linearly by the program. The specific format for input is described in subroutine READIT, which may be altered to suit the needs of the user.

Discharge schedules.—must be provided for the downstream boundaries of all pools. The discharge schedule at the upstream boundary of the most upstream pool must also be furnished to provide consistency with the input/output formats of the Constant Volume and Aqueduct Simulation Models. The program will ignore all but the most upstream and downstream schedules and will maintain discharge continuity throughout the length of the reach. Provision is made in the program to stroke the pumps at the upstream end of the reach or to stroke the gate which connects the reach to a reservoir. In these cases, the appropriate upstream discharge schedules will be ignored. Discharges are to be specified on the hour, and transitions from one discharge to another are made in one step at the time of change.

Turnout schedules.—must be provided for all turnouts. Turnouts are considered to be at the downstream ends of pools. They may occur separately or in conjunction with a control structure such as a check gate or siphon. The pool number, time of change of discharge (on the hour), and the change in discharge (positive discharge is outflow from the canal) are input for up to 48 discharge changes for each turnout. The initial turnout discharge is assumed zero, and the specified values are added to the current discharge at the specified times. No energy losses are considered at turnouts.

Pumps.—at the downstream end of the reach are treated as scheduled discharge devices. At the upstream end of the reach, pumps are treated as scheduled discharge devices if pump stroking is not employed.

With pump stroking, start and stop times for pumps within the reach are calculated using a dead band for discharge. That is, pumps will be started or shut down when the discharge change calculated by the program equals or exceeds the incremental discharge change specified for the pump in the physical descriptors for the pumps.
Gate positions.—are determined using the gate equation which is

\[ Q = C_d \cdot b \cdot B \cdot \sqrt{2g (y_1 - y_2) + V_i^2 / 2g} \]

where

- \( C_d \) = discharge coefficient
- \( b \) = gate opening
- \( B \) = gate width
- \( V_i \) = mean velocity in the rectangular section upstream of the gate
- \( y_1 \) = depth in the rectangular section upstream of the gate
- \( y_2 \) = depth in the rectangular section downstream of the gate

Gate movements are performed only when the calculated gate position has changed from the previous position by an amount equal to or exceeding the dead-band value for that gate. Gate structures are treated as having one gate. The width of the rectangular gate section is specified in the variable BOTGATE. Independent operation of multiple gates within a structure is not a feature of the program. This type of operation can be simulated, however, by modifying the discharge coefficient. The modified coefficient for single gate operation within a structure having multiple gates is given by

\[ C_{d_s} = \frac{C_d \cdot B_g}{B} \]

where \( B_g \) = bottom width of a single gate

Inverted siphons.—with or without control gates at the upstream end of the siphons are treated as zero length friction effects. The invert drop from the gate structure or siphon inlet to the siphon outlet and the friction loss factor are specified for each siphon.

Boundaries where no control structure is present.—(that is, changes in cross section, or turnouts) are treated as points at which no energy losses occur. Depths upstream of these boundaries are calculated using the energy equation.

Physical Descriptors

The constant descriptors of the reach, that is, stationing at pool ends, cross-section parameters, gate parameters, roughness, and siphon parameters, are set in DATA statements in the main program. Comment cards in the program describe the various parameters.

Modular Construction

The program has been constructed in a modular fashion. It is anticipated that expanded capabilities will be easily accomplished by this approach. For example, circular sections were added by defining a circular pool type and adding the appropriate code in the area, top width, and wetted perimeter function routines.

Output

The output file is formatted to be compatible for use as input to the Aqueduct Simulation Model.

Development and Structure of the Program

The gate stroking program was developed within rather severe time constraints to perform real-time operation studies for the Granite Reef and Salt-Gila Aqueducts. Previously published material dealt almost exclusively with techniques for the solution of the flow equations within a few computational segments. The equations of state and their solution for a system of pools present a challenge which is an order of magnitude greater than the problem of solving the open channel equations in a single segment. Some comments are appropriate concerning the general structure and the evolution of the present program.

The program is written in FORTRAN and is implemented on a CDC CYBER 74 computer system. Comment cards within the program explain the computational procedures. The assumption is made that anyone using the program will have a good understanding of free surface flow; an understanding of the various mathematical techniques encountered in the solution of the equations of state of the system; and a reasonable, but not expert, understanding of digital computing techniques. Within this framework, the program may be utilized to assist in the investigation of flow characteristics of virtually any canal configuration.

The program does not contain coding to accommodate all possible structures found in all canals. It also does not handle all possible canal cross sections. However, the modular structure of the program enables the user to add new structure types and pool types in a straightforward fashion. Likewise, from the multitude of possible operational philosophies for a canal, one was chosen as appropriate for the Granite Reef
Aqueduct, and that one was implemented in the
program. The philosophy chosen is that pump
schedules are predetermined at both ends of the
reach and the most upstream pool in the reach
will be uncontrolled with respect to depth
variations. The same computational techniques
can be used to “float” the downstream pool by
calculating in a downstream direction or to pivot
on an interior pool and calculate both upstream
and downstream of that pool. The modular
construction of the program allows the user to
implement these schemes in a relatively straight-
forward fashion. Provision was also made for
stoking the most upstream control structures.
The input and output for the program are
contained in the routines READIT and WRlTEIT
and may be modified easily to suit the user.

Various techniques were investigated for the
treatment of the system at control structures and
turnouts. The generalized program evolved out of
experience gained with these various tech-
niques. Some observations relative to the
evolution may be beneficial.

Since turnouts generally divert only a small
portion of the total discharge, the program used
for the Granite Reef study merely adjusted the
discharge within a computational segment when
a characteristic crossed the turnout location. This
treatment appears adequate when the turnout
diversion is small and when the turnout is not
near the control structures at the ends of the
pool. The advantage of this technique is that depth
continuity is maintained at a turnout location for
all time. If the turnout discharge were a large
percentage of the total discharge, a simultaneous
depth correction could be made at the turnout
using the energy equation. Difficulties in making
the proper discharge correction were en-
countered when the turnout fell between a
boundary and the grid point upstream or
downstream of the boundary. Since the method
of characteristics is stable for rather large time
increments and, consequently, the x spacing of
grid points can be rather large, the chance for
a turnout between a boundary and the next grid
point is fairly high. Thus, for the Salt-Gila study
and the general program, turnouts are treated
as boundaries. The disadvantage of this approach
is that depths at the turnout for one wave-travel
time in the upstream pool are not calculated
identically in the upstream and downstream
pools. Apparently, serious discrepancies do not
occur for normal operation schemes. The energy
equation is used to calculate a corrected depth
across the turnout so the effect of large diver-
sions is minimized.

As was discussed earlier, various equations have
been used to describe the state of a canal system
at gate structures. Mathematically, flows through
the gates can be very sensitive to small
differences in depths on the upstream and
downstream sides of the gates. Accordingly,
convergence criteria for the various iterative
schemes in the program should be examined
using the gate parameters particular to the
system under investigation. An adequate com-
promise between program run times and long-
term stability of the solution must be reached. For
self-consistency, identical methods of computing
the discharge through gates should be used in
both the gate stroking and the Aqueduct
Simulation Model programs. For this reason,
equations more closely approximating the
equations in Shand’s [8] program were used in
the Granite Reef and Salt-Gila versions of the
gate stroking program. Due to the finite number
of gate movements and different computational
techniques, differences will occur between the
predicted and “actual” water surfaces and
discharges, even with identical physical param-
eters and gate equations. These differences
should, however, be small. Large differences if
they occur, can be attributed to interpolation
errors in one or both of the programs.

Program Details

The equations upon which the coding is based
are described in the preceding sections. The
friction slope is computed using English cus-
tomy units. The conversion to SI metric units
can be accomplished by deleting the constant
1.49 in Subprogram S and by changing the value
of the gravitational constant GGraV to metric
units.

*The purposes of the subroutines and function
subprograms are as follows:
PROGRAM GSM - Main calling program. Canal
parameters are input through DATA state-
ments located in this program
SUBROUTINE READIT - Routine to read in dis-
charges, depths, turnout schedules, and
initial conditions
SUBROUTINE STROKER - Routine to solve gate
stroking problem for free surface flow
SUBROUTINE SOLVER - Routine to solve for X, t,
Q, and y at point P using values at point R on
positive characteristic and values at point S on
negative characteristic
FUNCTION C - Function to calculate wave celerity
FUNCTION R - Function to calculate hydraulic radius
FUNCTION A - function to calculate cross-sectional area in trapezoidal, horseshoe, or circular channels
FUNCTION TW - Function to calculate top width in trapezoidal, horseshoe, or circular channels
FUNCTION S - Function to calculate energy slope using Manning's equation
FUNCTION P - Function to calculate wetted perimeter in horseshoe, trapezoidal, or circular channels
SUBROUTINE BOUNDARY - Calculates \( t, \theta \), and \( y \) at an upstream boundary and \( t \) and \( y \) at a downstream boundary
FUNCTION HDN - Function to determine downstream boundary depth
FUNCTION QDN - Function to determine discharge boundary condition at downstream end of pools
FUNCTION QUPST - Function to specify discharge at upstream end of most upstream pool
SUBROUTINE TRNOUT - Routine to specify turnout discharge
SUBROUTINE GATEMO - Routine to calculate gate openings and pump schedules
SUBROUTINE GATEY - Routine to calculate depth in gate section from channel depth
FUNCTION CD - Routine to calculate discharge coefficient as a function of gate opening to upstream depth ratio
SUBROUTINE FOLLOWER - Routine to follow stroking motions to determine start times and final positions of gates and pump schedules
SUBROUTINE WRITEIT - Routine to merge turnout, gate, and pump schedules into input file for unsteady model

BIBLIOGRAPHY

APPENDIX

Gate Stroking Computer Program Listing
PROGRAM GSM 74/74 OPT=1

1 PROGRAM GSM(INPUT=64, OUTPUT=64, TAPE1=64, TAPE2=64, TAPE3=64, TAPE4=64, TAPE5=64, TAPE6=64, TAPE9=64)

C GENERALIZED GATE-STROKING PROGRAM

C THE GATE-STROKING TECHNIQUE IS TAKEN FROM BEN WYLIES
C PAPER CONTROL OF TRANSIENT FREE-SURFACE FLOW IN
C JAN., 1959, PROC OF ASCE, JOURNAL OF THE HYDRAULICS
C DIVISION

C THIS PROGRAM IS IN ENGLISH UNITS
C THE GRAVITATIONAL CONSTANT, GGRAV, AND THE FACTOR
C 1.49 IN MANNINGS N IN SUBPROGRAM S ARE THE ONLY
C CHANGES REQUIRED FOR METRIC CONVERSION
C SOME INTERMEDIATE OUTPUTS MAY BE AFFECTED BY A
C METRIC CONVERSION, AS FORMAT F10.3 IS GENERALLY
C USED, SO IT WOULD PAY TO CHECK AND SEE THAT NO
C SIGNIFICANCE WAS LOST IN Rounding

C THE PROGRAM IS SET UP TO INPUT PUMP SCHEDULES AND
C DEPTH SCHEDULES AT 0, 0700 AND 2300 HOURS.
C THE INPUT IS IN ROUTINE READIT, AND MAY HAVE TO BE
C CHANGED. THE OUTPUT IS IN ROUTINE WRITEIT, AND MAY
C HAVE TO BE CHANGED TO SUIT THE USERS REQUIREMENTS.

C TAPE1 IS INTERMEDIATE T,Q AND 9 FILE AT POOL BOUNDARIES
C TAPE2 IS INTERMEDIATE T AND GATE OPENING FILE
C TAPE3 IS INPUT FILE CONTAINING PUMP, DEPTH AND TURNOUT SCHEDULES
C TAPE4 IS SCRATCH FILE USED TO OVERLAY COMMON
C TAPE5 IS INTERMEDIATE START TIME AND GATE POSITION FILE
C TAPE6 IS OUTPUT FILE CONTAINING PROGRAM MESSAGES TO USER
C TAPE8 IS INPUT FILE CONTAINING INITIAL CONDITIONS
C TAPE9 IS OUTPUT FILE IN UNSTeadY MODEL FORMAT FOR INPUT
C TO UNSTeadY MODEL

COMMON /ALL/
C PHYSICAL PARAMETERS
. NMANN,50,GGRAV,PI,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
. MANN(50),BOTGRAO(50),SYFONDY(50),SYFNLOS(50),SPEED(51),PUMPQO
. (51),DEDBAND(51),NAGES,GATECO(11,51),DSINV(50),USINV(51),
C WORKING VARIABLES
. TMX,XEND,TING,NXINC,NPOOLS,IPool,TypPool,STABEG(50),
. STAEH(50),T1,STUDY,
C CANAL DESCRIPTION
. UPSTA(50),DWNSTA(50),NPOOLS,POOLTyp(50),STRTyp(51)
. TMXXNUMTRNS,USRES,OSRES
C REAL NMANN,MANN
C INTEGER TypPool,PoolTyp,TypType,STUDY
C SCRATCH AREA TO BE OVERLAID IN THE VARIOUS SUBRoutines
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
COMMON /SCRATCH/
. HDX(50),XSTART(40,50),YSTART(40,50),GSTART(40,50),QSTART(40,50),
. HSCHED(40,50),XINIT(40),YINIT(40),QINIT(40),H(40),
. DUMMY(7169)
C DIMENSION SCRATCH(15787)
EQUIVALENCE (SCRATCH(1),NDX(1))

60 C STUDY TELLS THE PROGRAM IF THE MOST UPSTREAM
C STRUCTURE, EITHER A PUMP OR A GATE, IS TO BE
C STROKED OR NOT
C STUDY=1 IS NO STROKING (Q IS SCHEDULED)
C STUDY=2 IS STROKING
DATA (STUDY=2)

C POOLS ARE NUMBERED FROM DOWNSTREAM TO UPSTREAM IN THE REACH
C ARRAYS ARE DIMENSIONED FOR 50 POOLS
C ARRAYS ARE DIMENSIONED FOR 40 GRID POINTS PER POOL

70 C SET MAX TIME FOR THE REACH
C MAX TIME MUST BE AT LEAST STUDY TIME (USUALLY 24 HOURS)
C PLUS WAVE TRAVEL TIME FOR THE REACH
C TIME IS IN SECONDS
DATA (TMAX=19600.)

C STATIONING AT UPSTREAM ENDS OF POOLS
C ARRAY IS UPST(POOL)
DATA (UPST=270494.4,234654.4,198105.6,162148.8,134270.4)

C STATIONING AT DOWNSTREAM ENDS OF POOLS
C ARRAY IS OWST(POOL)
DATA (OWST=304972.0,270494.4,234654.4,198105.6,162148.8)

C NUMBER OF POOLS IN THE REACH
DATA (NPOOLS=1)

C TYPES OF POOLS
C POOLTYP=0 THIS POOL NOT USED
C POOLTYP=1 TRAPEZOIDAL CHANNEL
C POOLTYP=2 HORSESHOE TUNNEL
C POOLTYP=3 CIRCULAR TUNNEL
C ARRAY IS POOLTYP(POOL)
DATA (POOLTYP=5,45=0)

C MANNINGS N
C ARRAY IS MANN(POOL)
DATA (MANN=5*0.016,45*0.0)

C BOTTOM SLOPE - 50
C ARRAY IS BOTGRAD(POOL)
DATA (BOTGRAD=5*0.00008,45*0.0)

C STRUCTURE TYPES AT UPSTREAM ENDS OF POOLS
C ARRAY IS STRTYPE(POOL)
C STRTYPE=1 MOST DOWNSTREAM STRUCTURE
C STRTYPE=0 POOL NOT USED
C STRTYPE=1 PUMP
C STRTYPE=2 NORMAL CHECK GATE
C STRTYPE=3 SYPHON WITH CHECK GATE
C STRTYPE=4 SYPHON WITH NO CONTROL STRUCTURE
C STRTYPE=5 NO CONTROL STRUCTURE
DATA (STRTYPE=3,2,3,2,45=0,2)

C TRAPEZOIDAL CHANNEL PROPERTIES - BOTTOM WIDTH AND SIDESLOPE
C SIDESLOPE=0.0 HOLDS A RECTANGULAR SECTION
C DATA (BOTTOM=5*24.0,45=0.0)
DATA (SIDSLOPE=5*1.5,45=0.0)
C CHECK GATE PROPERTIES
C POOL NUMBERING IS D/S TO U/S
C GATE IS AT U/S END OF POOL
C PARAMETERS FOR GATE AT D/S END OF REACH IN ARRAY(51)

C SET UPSTREAM AND DOWNSTREAM CONSTANT LEVEL
C RESERVOIR DEPTHS IF GATES AT U/S OR D/S ENDS OF REACH
DATA (USRES=16.0),(DSRES=14.0)
C CHECK GATE BOTTOM WIDTH
DATA (BOTGATE=5*36.0,45*0.0,36.0)
C GATE MOTOR SPEED
DATA (SPEED=5*0.75,45*0.0,0.75)
C GATE MOTION DEADBAND
DATA (DEDBAND=5*0.5,45*0.0,0.5)
C INVERT DROP FROM GATE SECTION TO CHANNEL ON D/S SIDE OF GATE
C IF SYFON, LUMP INVERT DROPS ON D/S SIDE IN SYFONDY BELOW
C IF GATE AT U/S END OF REACH, DEPTH IN USRES IS IN
C GATE SECTION - NO D/S INVERT DROP OR SYFON PARAMETERS ARE
C INPUT FOR THAT GATE
DATA (DSINVT=5*0.0,0.0,0.0,0.0,0.0)
C INVERT DROP FROM CHANNEL TO GATE SECTION ON U/S SIDE OF GATE
C IF GATE AT U/S END OF REACH, DEPTH IN USRES IS IN
C GATE SECTION - NO U/S INVERT DROP IS INPUT FOR THAT GATE
C U/S INVERT DROP IS ALLOWED FOR GATE AT D/S END
C OF REACH, IF ANY
DATA (USINVT=5*0.0,0.0)
C GATE COEFFICIENTS
C ARRAY IS GATECO(1-11,POOL)
C SEE ROUTINE GATEMO FOR THEIR USE IN GATE EQUATION
DATA (GATECO=
  .71,.688,.67,.655,.645,.644,.659,.693,
  .756,.862,1.019,
  .71,.688,.67,.655,.645,.644,.659,.693,
  .756,.862,1.019,
  .71,.688,.67,.655,.645,.644,.659,.693,
  .756,.862,1.019,
  .71,.688,.67,.655,.645,.644,.659,.693,
  .756,.862,1.019)
C PUMP PARAMETERS
C PUMPDQ(POOL) IS DISCHARGE INCREMENT FOR PUMPS AT U/S END OF POOL
C USED IN ROUTINE FOLLOWER TO DETERMINE PUMP SCHEDULES
C PUMPDQ(51) IS AT D/S END OF REACH
DATA (PUMPDQ=4*0.0,125.,45*0.0,125.)
C TUNNEL PROPERTIES
C RADIUS IS HEIGHT OF TUNNEL, INVERT TO CREST, FOR HORSESHOE TUNNEL
C RADIUS IS RADIUS OF TUNNEL FOR CIRCULAR TUNNEL
C ARRAY IS RADIUS(POOL)
DATA (RADIUS=50*0.0)
C SYFON PARAMETERS
C ARRAYS ARE SYFONDY(POOL) AND SYFNLOS(POOL)
C INVERT DROP IN SYPHON
   DATA (SYFONDY=3*0.0,3.24,0.0,45*0.0)
C SYPHON LOSSES
   DATA (SYFNLOS=3*0.0,3.21E-7,0.0,45*0.0)
C TURNOUTS
C TURNOUTS CAN BE AT D/S END OF ANY POOL
C NUMBER OF TURNOUTS IN THE REACH
   DATA (NUMTRNS=0)
C
   DATA (GGRAV=32.2),(PI=3.1415927)
C
REWIND 6
REWIND 1
REWIND 2
REWIND 3
REWIND 4
REWIND 5
REWIND 8
REWIND 9
C
C READ IN FLOWS, DEPTHS AND TURNOUT SCHEDULES
C AND INITIAL CONDITIONS
C
   CALL READIT
C
C SET TIME INCREMENT FOR CALCULATIONS AT BOUNDARIES (IN SECONDS)
C 6 MIN X 30 HOURS STAYS IN ARRAY LIMITS
C
   TINC=300.
C SET NUMBER OF POOLS IN THIS REACH
   NPOLS=NPOLLS
C SET BEGINNING AND ENDING STATIONING FOR POOLS
   DO 10 I=1,NPOLS
       STABEG(I)=UPSTA(I)
       STAEND(I)=DWNSTA(I)
   10 CONTINUE
C SET MAXIMUM CALCULATION TIME FOR THE REACH -
C MODEL TIME, NOT COMPUTE TIME
C
   TMAX=TMAXI
C SOLVE FOR EACH POOL
C CALCULATIONS PROCEED IN UPSTREAM DIRECTION
   DO 190 I=1,NPOLS
       IPOOL=I
C SET POOL LENGTH AND TYPE
       XEND=STAEND(IPOOL)-STABEG(IPOOL)
       TYPPOOL=TYP(POOL(IPOOL))
C SET INITIAL CONDITIONS FOR THIS POOL
C HSCHED IS D/S DEPTH SCHEDULE
C HSCHED WAS READ IN BY READIT
C SCHEDULE IS FOR 48 HOURS
   DO 20 J=1,48
       H(J)=HSCHED(J,IPOOL)
   20 CONTINUE
C SET NUMBER OF INCREMENTS ALONG LENGTH OF POOL
C CALCULATE BACKWATER CURVE IF ONLY ONE POINT IS GIVEN -
C THE NON-ZERO VALUE FOR Y MAY BE AT EITHER END OF POOL
C AND THE BACKWATER CURVE WILL BE CALCULATED TO THE OTHER END
C CALCULATE BACKWATER CURVE, IF DESIRED, FOR 11 POINTS IN POOL
IF(NXINC.EQ.2.AND.YSTART(1,IPOOL).EQ.0.0) CALL BAKWATR(1)
IF(NXINC.EQ.2.AND.YSTART(2,IPOOL).EQ.0.0) CALL BAKWATR(2)
C LINEAR WATER SURFACE IF LESS THAN 11 POINTS
C IF BAKWATR WAS CALLED, NXINC WAS SET TO 11
IF(NXINC.GE.11) GO TO 40
WRITE(6,9000)
DELX=XSTART(NXINC,IPOOL)-XSTART(1,IPOOL)
DELY=YSTART(NXINC,IPOOL)-YSTART(1,IPOOL)
DELO=OSTART(NXINC,IPOOL)-QSTART(1,IPOOL)
C LINEAR INTERPOLATION FROM ZERO TO XEND IN 10 EQUAL STEPS
DO 30 L=2,11
FL=L-1
XSTART(L,IPOOL)=XSTART(1,IPOOL)+FL*DELX/10.0
YSTART(L,IPOOL)=YSTART(1,IPOOL)+FL*DELY/10.0
QSTART(L,IPOOL)=QSTART(1,IPOOL)+FL*DELO/10.0
30 CONTINUE
NXINC=11
NDX(IPOOL)=11
40 CONTINUE
C CHECK THAT ARRAYS ARE NOT OVERRUN
WRITE(6,9010) NXINC
IF(NXINC.GT.40) STOP
C MEASURE X FROM ZERO AT UPSTREAM END OF POOL
DO 50 L=1,NXINC
XINIT(L)=XSTART(L,IPOOL)-STABEG(IPOOL)
YINIT(L)=YSTART(L,IPOOL)
QINIT(L)=QSTART(L,IPOOL)
50 CONTINUE
WRITE(6,9040) XINIT(1),XINIT(NXINC)
XINIT(1)=0.0
XINIT(NXINC)=XEND
C SAVE COMMON ON TAPE4
WRITE(4,9030) (SCRATCH(ISCR),ISCR=1,6450)
REWIND 4
C SOLVE THIS POOL
CALL STROKER
C RETRIEVE COMMON FOR NEXT POOL
READ(4,9030) (SCRATCH(ISCR),ISCR=1,6450)
REWIND 4
270 WRITE(6,9020) IPOOL
100 CONTINUE
REWIND 1
C CALCULATE GATE OPENINGS AND PUMP DISCHARGES FOR 24 HOUR PERIOD
CALL GATEMO
C CALCULATE GATE MOVEMENTS FROM GATE OPENINGS AND CALCULATE PUMP SCHEDULES
CALL FOLLOWER
C OUTPUT CALCULATED DATA FOR UNSTEADY MODEL INPUT
CALL WRITEIT
WRITE(6,9050)
REWIND 6
REWIND 9
STOP
C FORMATS
9000 FORMAT(1X,43HINTERPOLATING WATER SURFACE FROM END POINTS)
9010 FORMAT(1X,6HNXINC=,I10,19H MAX ALLOWED IS 40)
PROGRAM GSM

9020 FORMAT(1X,6HEND POOL,15/)
9030 FORMAT(5020)
9040 FORMAT(1X,6HINIT=,2F10.2)
9050 FORMAT(1X,18HNORMAL TERMINATION )

END
SUBROUTINE BAKDATR 74/74 OPT= 1 FTN 4.6+420 79/01/09. 14.41.14 PAGE 1

1 SUBROUTINE BAKDATR(I CHOOSE)
C ROUTINE TO CALCULATE INITIAL BACKWATER CURVES
C GIVEN DOWNSTREAM Y AND Q (ICHOOSE=1)
C OR UPSTREAM Y AND Q (ICHOOSE=2)
5 C SECOND ORDER RUNGA-KUTTA METHOD ON
C DY/DX=(SO-S)/(1-Q**2*l/A**3*G)
C DIMENSION X(40)
C COMMON /ALL/
C PHYSICAL PARAMETERS
10 . NMANN,50,GGRAY,P1,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51).
. MANN(50),BOTGRAD(50),SYFONDY(50),SYFNLOS(50),SPEED(51),PUMPQ
. (51),DEGAND(51),NGATES,GATECO(11,51),DSINVY(50),USINVY(51),
C WORKING VARIABLES
. TMAX,XEND,TINC,NINC,NPOOLS,IPool,TYPPOOL,STABEG(50),
. STAEND(50),T1,T1,STRTYPE.
C CANAL DESCRIPTION
. UPESTA(50),DWNSTA(50),NPOOLS,POOLTYP(50),STRTYPE(51),
. TMAX,NAMTRNS,USRES,OSRES
C
20 . REAL NMANN,MANN
. INTEGER TVPPOOL,POOLTYP,STRTYPE,STUDY
C SCRATCH AREA TO BE OVERLAID IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
25 . COMMON /SCRATCH/
. . NOX(50),ASTART(40,50),YSTART(40,50),QSTART(40,50),
. . HSCHL(49,50),GINIIT(40),GINIT(40),QINIT(40),H(40),
. . DUMMY(7169)
. REAL K
C WRITE(6,70)
C DIVIDE POOL INTO (N-1)*10 SEGMENTS
C STORE ONLY EVERY TENTH VALUE OF XN AND YN
C XEND=ASTART(2,IPool)-ASTART(1,IPool)
N=11
35 . S0=BOTGRAD(IPool)
. TYPPOOL=POOLTYP(IPool)
. NOX/IPool)=11
. NINC=11
. IF(ICHOOSE.EQ.1) GO TO 6
C CALCULATE IN DOWNSTREAM DIRECTION
C DO QSTART(1,IPool)
Y0=YSTART(1,IPool)
FN=N-1
40 . DX=XEND/(FN+10.)
X(K1)+0.0
L=0
YN=Y0
DO 2 I=2,N
L=L+1
50 . DO 1 J=1,10
FJ=J
XN=X(L)+FJ*DX
A1=A(YN)
TN=TW(TN)
1 IF(TN.LE.0.) GO TO 80
VC2=GGRAY*A1/TW
IF(VC2.LE.0.) GO TO 40
2 CONTINUE
5 IF(YN.LE.0.) GO TO 80
10 CONTINUE
20 CONTINUE
30 CONTINUE
V1=Q0/A1
IF(ABS(V1**2-VC2).LT.0.001) GO TO 50
R1=R(YN)
S1=S(Y1,R1)
F1=(S0-S1)/(1.-Q0**2*TW1/(A1**3*GGRAV))
Y2=YN+DX*F1
A2=A(Y2)

TW2=TW(Y2)
IF(TW2.LE.0.) GO TO 60
VC2=GGRAV+A2/TW2
IF(VC2.LE.0.) GO TO 40
V2=Q0/A2

IF(ABS(V2**2-VC2).LT.0.001) GO TO 50
R2=R(Y2)
S2=S(Y2,R2)
F2=(S0-S2)/(1.-Q0**2*TW2/(A2**3*GGRAV))
K=0.5*(F1+F2)

1 CONTINUE
X(L+1)=XN
XSTART(L+1,IPool)=XN+STABEG(IPool)
YSTART(L+1,IPool)=YN
QSTART(L+1,IPool)=Q0

2 CONTINUE
XSTART(1,IPool)=STABEG(IPool)
XSTART(N,IPool)=STAEN(IPool)
RETURN

C CALCULATE IN UPSTREAM DIRECTION

5 CONTINUE
Q0=QSTART(2,IPool)
YN=YSTART(2,IPool)
FN=1

DO 20 I=2,N
DX=XEN0/(FN*10.)
X(N)-XEN0
L=I-1
 DO 10 J=1,10
FJ=FN
XN=X(L)+FJ*DX
A1=A(YN)
TW1=TW(YN)
IF(TW1.LE.0.) GO TO 60
VC2=GGRAV*A1/TW1
IF(VC2.LE.0.) GO TO 40
V1=Q0/A1

IF(ABS(V1**2-VC2).LT.0.001) GO TO 50
R1=R(YN)
S1=S(Y1,R1)
F1=(S0-S1)/(1.-Q0**2*TW1/(A1**3*GGRAV))
Y2=YN+DX*F1
A2=A(Y2)

TW2=TW(Y2)
IF(TW2.LE.0.) GO TO 60
VC2=GGRAV+A2/TW2
IF(VC2.LE.0.) GO TO 40
SUBROUTINE BAKWATF 74/74 OPT=1 FTN 4.6+420 79/01/09. 14.41.14 PAGE 3

115  
V2=Q0/2  
IF(ABS(V2**2-VC2).LT.0.001) GO TO 50  
R2=R(V2)  
S2=S(V2,R2)  
F2=(50-52)/(1.-Q0**2+TW2/(A2**3*GGRAV))  
K=0.5*(F1+F2)  
YN=YN+DX*K  
CONTINUE

10   
X(L-1)=XN  
XSTART(L-1,IPOOL)=XN+STABEG(IPool)  
YSTART(L-1,IPOOL)=YN  
QSTART(L-1,IPOOL)=Q0  
CONTINUE

120   
XSTART(1,IPool)=STABEG(IPool)  
XSTART(N,IPool)=STAEND(IPool)  
YSTART(N,IPool)=YD  
QSTART(N,IPool)=Q0  
RETURN

CONTINUE

135   
WRITE(6,45) YN,XN  
FORMAT(1X,22HZERO OR NEG V++2 AT Y+,F10.3,  
3H X+,F10.3,12H IN BAKWATER )  
STOP

140   
WRITE(6,55) YN,XN  
FORMAT(1X,41HCALCULATING BACKWATER CURVE FOR THIS POOL  
STOP

50   
CONTINUE

55   
WRITE(6,65) YN,XN  
FORMAT(1X,32HZERO OR NEG TW IN BAKWATER AT Y+,F10.3,  
3H X+,F10.3)  
STOP

60   
CONTINUE

65   
WRITE(6,65) YN,XN  
FORMAT(1X,41HCALCULATING BACKWATER CURVE FOR THIS POOL  
STOP

70   
CONTINUE
SUBROUTINE READIT

C ROUTINE TO READ IN FLOWS AND DEPTHS AND TURNOUT SCHEDULES

C COMMON /ALL/
C PHYSICAL PARAMETERS
.FMANN, 50, GGRAV, PI, BOTTOM(50), SIDSDSLOP(50), RADIUS(50), BOTGATE(51),
.MANN(50), BOTGRAD(50), SYFUNDY(50), SYFNLOS(50), SPEED(51), PUMPQQ
.51, DEGAND(51), GATES, GATECO(11,51), OSINV(50), USINV(51),
C WORKING VARIABLES
.TMAX, XEND, TINC, NPOOLS, IPOOL, TYPOOL, STABEG(50),
.STAEND(50), Ti, Yi, STUDY,
C CANAL DESCRIPTION
.UPSTA(50), DNSTA(50), NPOOLS, POOLTY(50), STRTYPE(51),
C REAL NMANN, MANN
INTEGER TVPPDDL, PDOLTYP, STRTYP, STUDY

C SCRATCH AREA TO BE OVERLAI D IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
C COMMON /SCRATCH/
.NDX(50), XSTART(40,50), YSTART(40,50),
.HSCHED(48,50),
.XINIT(40), YINIT(40), QINIT(40), H(48),
.QD(48), QU(48), QT(48,50),
.DUMNV1(4673)

C TAPE 5 HAS UPSTREAM AND DOWNSTREAM Q SCHEDULES, DEPTH
C SCHEDULES AND TURNOUT SCHEDULES AT
C T=0, 0700 AND 2200
C SCHEDULES MUST BE SUPPLIED AT EVERY POOL BOUNDARY,
C EVEN WHEN THEY CANNOT BE USED, E.G., AT POOL BOUNDARIES
C WHERE NO CONTROL STRUCTURE IS PRESENT

C REWIND 3
C READ OVER 4 LINE HEADING
READ(3, 9000)
.MAX=NPOOLS
C READ AT HOURS 0, 7, 22
C READ U/S PUMP SCHEDULE
C INDEX IS HOUR+1
READ(3, 9030) QU(1), QU(8), QU(23)
BACKSPACE 3
C STORE IN D/S TO U/S ORDER
C TAPE 3 IS IN U/S TO D/S ORDER
INDEX=MAX
INDEX-INDEX-1
30 CONTINUE
C READ D/S PUMP SCHEDULE
C READ AT HOURS 0, 0700 AND 2200
C INDEX IS HOUR+1
READ(3, 9050) QD(1), QD(8), QD(23)
C FILL IN REMAINING HOURS OF DAY
SUBROUTINE READIT
74/74 OPT=1
DO 40 I=2,7
DO 40 J=1,MAX
QU(I)=QU(I-1)
QD(I)=QD(I-1)
HSCHED(I,J)=HSCHED(I-1,J)
CONTINUE
DO 50 J=1,MAX
QU(I)=QU(I-1)
QD(I)=QD(I-1)
HSCHED(I,J)=HSCHED(I-1,J)
CONTINUE
DO 60 J=1,MAX
HSCHED(24,J)=HSCHED(23,J)
CONTINUE
C CONTINUE SCHEDULES FOR SECOND 24 HOURS
DO 70 J=1,MAX
HSCHED(I,J)=HSCHED(I-1,J)
70 CONTINUE
C QT(HOUR,POOL) IS TURNOUT Q
C TURNOUT IS AT D/S END OF POOL
DO 80 I=1,50
DO 80 J=1,48
QT(J,I)=0.0
CONTINUE
C ARE THERE ANY TURNOUTS IN THIS REACH
C NUMTRNS IS SET IN A DATA STATEMENT IN MAIN PROGRAM
90 IF(NUMTRNS.EQ.0) GO TO 140
C READ TURNOUT SCHEDULES
C MAXTURN IS LAST INDEX IN THIS REACH
C READ OVER 3 LINE HEADING FOR SCHEDULE
READ(3,9060)
MAXTURN=NUMTRNS
DO 130 IND=1,MAXTURN
C READ NUMBER OF CHANGES AT THIS TURNOUT
READ(3,9090) LPOOL,NCHANGE
C READ TIME OF CHANGE AND DELTA Q
100 C CHANGES MAY ONLY BE MADE ON THE HOUR FROM 0000 TO 2300 HOURS
DO 120 I=1,NCHANGE
READ(3,9070) NTIME,DELO
C TIME INDEX IS HOUR+1
105 ITIME=NTIME+1
QT(ITIME,POOL)=QT(ITIME,POOL)+DELO
MIN=ITIME+1
IF(MIN.GT.24) GO TO 100
C SET Q AT TURNOUT FOR THE REMAINDER OF THE DAY
DO 90 J=MIN,24
QT(J,POOL)=QT(J-1,POOL)
CONTINUE
C CONTINUE SCHEDULE FOR SECOND 24 HOURS
DO 110 J=1,24
SUBROUTINE READIT

QT(1:24, LPOOL)=QT(24, LPOOL)

CONTINUE
CONTINUE
CONTINUE
CONTINUE

C TAPE 8 HAS INPUT FOR UNSTEADY MODEL
C READ INITIAL CONDITIONS FROM THIS FILE
C COPY TAPE 8 TO TAPE 0 FOR USE AS INPUT TO
C UNSTEADY MODEL WITH GATE SCHEDULES INSERTED
C AFTER THIS PROGRAM HAS CALCULATED THEM
C AND WITH TURNOUT SCHEDULES INSERTED
REWIND 0

C READ 3 TITLE CARDS AND STUDY START TIME CARD
DO 150 I=1,4
READ(6,9080)
WRITE(6,9080)
WRITE(6,9080)

CONTINUE

C READ INITIAL CONDITIONS
KPOOL=NPOOLS
C READING POOLS FROM UPSTREAM TO DOWNSTREAM
C STORING POOLS FROM D/S TO U/S
C X VALUES RUN FROM U/S TO D/S IN BOTH CASES
C MUST HAVE CARDS FOR U/S AND D/S ENDS OF ALL POOLS
C IF ONLY U/S AND D/S POINTS ARE GIVEN, THEN
C IF Y U/S = 0.0, CALCULATE BACKWATER CURVE FROM D/S END
C IF Y D/S = 0.0, CALCULATE BACKWATER CURVE FROM U/S END
C IF Y U/S AND Y D/S NE 0.0, LINEAR WATER SURFACE AND
C LINEAR Q CHANGE BETWEEN ENDS OF POOL
C
INDEX=1
READ(9,9110) YSTART(INDEX,KPOOL),QSTART(INDEX,KPOOL)
XSTART(INDEX,KPOOL)
WRITE(9,9110) YSTART(INDEX,KPOOL),QSTART(INDEX,KPOOL)
XSTART(INDEX,KPOOL)
C SEARCH FOR END OF POOL
XEND=DOWNSTA(KPOOL)
INDEX=INDEX+1
READ(9,9110) YSTART(INDEX,KPOOL),QSTART(INDEX,KPOOL)
XSTART(INDEX,KPOOL)
WRITE(9,9110) YSTART(INDEX,KPOOL),QSTART(INDEX,KPOOL)
XSTART(INDEX,KPOOL)
IF(XSTART(INDEX,KPOOL).EQ.XEND) GO TO 190
IF(XSTART(INDEX,KPOOL).LT.XEND) GO TO 180
WRITE(6,9120) XSTART(INDEX,KPOOL),XEND
STOP

CONTINUE

C STORE THE NUMBER OF INITIAL CONDITION POINTS
INDEX(KPOOL)=INDEX

C HAVE WE READ DOWN TO POOL 1, THE MOST D/S POOL IN THE REACH
IF(KPOOL.EQ.1) RETURN
KPOOL=KPOOL-1
C READ NEXT D/S POOL
GO TO 170
C
C FORMATS
9000 FORMAT(1X/1X)
9010 FORMAT(4.1S)
9030 FORMAT(17X,F11.2,2F16.2)
9040 FORMAT(1X/17X,F11.2,2F16.2/1X)
9050 FORMAT(17X,F11.2,2F16.2)
9060 FORMAT(1X/1X)
9070 FORMAT(15,3X,F10.2)
9080 FORMAT(40H)
9090 FORMAT(I10)
9100 FORMAT(10H ,I10,20H)
9110 FORMAT(F10.3,10H ,F10.3,30H)
9120 FORMAT(1X,13H FOUND STATION, F10.3, F10.2)
175 9030 FORMAT(17X,F11.2,2F16.2)
180 9040 FORMAT(1X/17X,F11.2,2F16.2/1X)
185 9110 FORMAT(F10.3,10H ,F10.3,30H)
189 9120 FORMAT(1X,13H FOUND STATION, F10.3, F10.2)
194 9050 FORMAT(17X,F11.2,2F16.2)
199 9060 FORMAT(1X/1X)
204 9070 FORMAT(15,3X,F10.2)
209 9080 FORMAT(40H)
214 9090 FORMAT(I10)
219 9100 FORMAT(10H ,I10,20H)
224 9110 FORMAT(F10.3,10H ,F10.3,30H)
229 9120 FORMAT(1X,13H FOUND STATION, F10.3, F10.2)
SUBROUTINE STROKER
C ROUTINE TO SOLVE GATE STROKING PROBLEM FOR OPEN
C CHANNEL FLOW AS PER WYLIE
C THIS PROGRAM IS SET UP FOR Q AND Y

COMMON /SOLV/ XP,YP,OP,XR,YP,TR,QR,XS,YS,TS,QS
COMMON /ALL/
C PHYSICAL PARAMETERS
NMANN,50,GOGRAV,PI,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
MANH(50),BOTRAD(50),SVFONDS(50),SFNYEDS(50),SPEED(51),PUMPPOQ(51),DEDBAND(51),GATECETO(11,51),DSINV(50),USINV(51).
C WORKING VARIABLES
TMAX,XEND,TINC, NXINC, NPOLS, IPOOL, TYPPOOL, STAEG(50),
STAENO(50), TI, TV, STUDY,
C CANAL DESCRIPTION
UPSTA(50), DWNSTA(50), NPOLPOLS, TYPOLYP(50), STRTYPE(51),
TMAFL, NUMTRNS, USRES, DSRES

REAL NMANN, MANN
INTEGER TYPPOOL, TYPPOO, STRTYP, STUOY

C SCRATCH AREA TO BE OVERLAID IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
COMMON /SCRATCH/ XCP1(40), YCP1(40), TCP1(40), QCP1(40),
XCM1(40), YCM1(40), TCM1(40), QCM1(40),
XCP2(40), YCP2(40), TCP2(40), QCP2(40),
XDN2(40), YDN2(40), TDN2(40), QDN2(40),
XCP3(40), YCP3(40), TCP3(40), QCP3(40),
XCM3(40), YCM3(40), TCM3(40), QCM3(40),
XDN3(40), YDN3(40), TDN3(40), QDN3(40),
XUP3(40), YUP3(40), TUP3(40), QUP3(40),
XCPB(40), YCPB(40), TCPB(40), QCPB(40),
XDNB(40), YDNB(40), TDNB(40), QDNB(40),
XUB(40), YUB(40), TUB(40), QUB(40),
XCP4(81), YCP4(81), TCP4(81), QCP4(81),
XCM4(81), YCM4(81), TCM4(81), QCM4(81),
XUP4(81), YUP4(81), TUP4(81), QUP4(81),
XCP5(81), YCP5(81), TCP5(81), QCP5(81),
DUMMY2(5394), XINIT(40), YINIT(40), QINIT(40), DUMMY2(2544),
IND(85), TIND(85), KOWN, TDW(500), TDWN(500), TDWN(500),
KUP, TUP(500), QUP(500), YUP(300),
KMAX, SAVETUP(500), SAVETYUP(500), SAVEEP(500), SAVEUP(500)
DIMENSION ISRONCE(5,2), X(N), Y(N), T(N), Q(N)
C ISEQUCE IS SEQUENCE TO CALCULATE SOLUTION REGIONS
C
C MAP OF SOLUTION REGIONS FOR ALL BUT MOST U/S POOL
C REGION 3B IS REPEATED AS NECESSARY TO GET TO TMAX
C Q AND Y ARE GIVEN AT D/S BOUNDARY AND Q AND Y ARE
C CALCULATED AT U/S BOUNDARY
C
MAP OF SOLUTION REGIONS FOR MOST U/S POOL
IF Q INTO POOL IS SCHEDULED (STUDY=1)
Q IS GIVEN AT BOTH BOUNDARIES AND Y IS CALCULATED
FOR BOTH BOUNDARIES
C ISEQNCE=0 EXITS THE COMPUTATION LOOP
C DATA (ISEQNCE=1,2,3,4,8,1,2,5,8,8)
C LAST=2 FOR MOST UPSTREAM POOL
C IF Q IS SCHEDULED FOR U/S END OF THAT POOL
C FLOAT DEPTH FOR MOST UPSTREAM POOL
C IF Q IS SCHEDULED
C IPool IS THE POOL BEING CALCULATED
C Npools IS THE NUMBER OF THE MOST U/S POOL
C IF (IPool.EQ.QPools.AND.Study.EQ.1) LAST=2
C Timinc IS THE TIME INCREMENT USED FOR CALCULATIONS
C IT MAY BE INCREASED IF 40 POINTS IN REGION 3 WONT GET
C THE SOLUTION TO X=0
C OR DECREASED IF POOL IS VERY SHORT
C Timinc=Timinc
C SET THE NUMBER OF COMPUTATION POINTS FOR EACH REGION
N1=NINic
N2=N1
N4=N1+N2-1
N5=N4
C KUp IS UPSTREAM INDEX FOR STORING T, Q AND Y
KUp=N4
Kdown IS DOWNSTREAM INDEX FOR STORING T, Q AND Y
Kdown=0
DO 800 IPass=1,5
ISEQ=ISEQNCE(IPass,LAST)
GO TO(100,200,300,400,500,900,900,900),1Seq
C X, Y, T AND Q ARE WORKING ARRAYS
C ...CPJ AND ...CMJ ARRAYS ARE X, Y, T AND Q VALUES
C ALONG + AND - CHARACTERISTICS IN REGION J
C
C REGION 1 - INITIAL CONDITIONS
100 CONTINUE
C Xinit, etc., ARE INITIAL CONDITIONS AT TIME = 0
DO 110 I=1,N1
X(I)=XInit(I)
Y(I)=VInit(I)
T(I)=0.
Q(I)=QInit(I)
110 CONTINUE
X(N1)=XEnd
XCP1(1)=X(1)
YCP1(1)=Y(1)
TCP1(1)=T(1)
QCP1(1)=Q(1)
XCM1(1)=X(N1)
YCM1(1)=Y(N1)
TCM1(1)=T(N1)
QCM1(1)=Q(N1)
M=N1-1
DO 150 I=2,N1
DO 150 J=1,M
C IP, IR AND IS ARE P, R AND S POINT INDICES IN X, Y, T
C AND Q WHERE P IS POINT TO BE SOLVED FOR,
SUBROUTINE STROKER  74/74  OPT=1

C R IS KNOWN POINT ON + CHARACTERISTIC AND
C S IS KNOWN POINT ON - CHARACTERISTIC
C P, R, S NOTATION FOLLOWS WYLIE

175  IP=J
      IR=J
      IS=J+1

C SET R AND S POINT VALUES
  XR=X(IR)
  YR=Y(IR)
  TR=T(IR)
  QR=Q(IR)
  XS=X(IS)
  YS=Y(IS)
  TS=T(IS)
  QS=Q(IS)

C SOLVE FOR POINT P
CALL SOLVER

C STORE NEW VALUES
  X(IP)=XP
  Y(IP)=YP
  T(IP)=TP
  Q(IP)=QP

C STORE END POINTS IN REGION 1 BOUNDING CHARACTERISTICS
  IF(J.EQ.1) GO TO 130
  120 IF(J.EQ.M) GO TO 140
  GO TO 150

C STORE IN + CHARACTERISTIC
  130 CONTINUE
    XCPl(I)=XP
    YCP(I)=YP
    TCP(I)=TP
    QCP(I)=QP

C LAST POINT IS ALSO ON - CHARACTERISTIC
C CHECK FOR THIS SITUATION
  GO TO 120

C STORE IN - CHARACTERISTIC
  140 CONTINUE
    XCMl(I)=XP
    YCMl(I)=YP
    TCMl(I)=TP
    QCMl(I)=QP
  150 CONTINUE
  M=M-1

  160 CONTINUE
WRITE(6,9000)ISEQ
GO TO 900

C REGION 2 - Q SPECIFIED D/S
  200 CONTINUE
    XDN2(I)=XCM1(I)
    YDN2(I)=YCM1(I)
    TBDN2(I)=TCM1(I)
    QDNN2(I)=QCM1(I)
  DO 230 I=2,M2
  225 X(I)=XCM1(I)
      Y(I)=YCM1(I)
      T(I)=TCM1(I)
      Q(I)=QCM1(I)
SUBROUTINE STROKER

M = 1
DO 210 J = 2, M
IR = J - 1
IS = J
IP = J
C SET R AND S POINT VALUES
XR = X(IR)
YS = Y(IR)
TR = T(IR)
QR = Q(IR)
C CHECK FOR BOUNDARY
IF (J .EQ. M) GO TO 220
XS = X(IS)
YS = Y(IS)
TS = T(IS)
QS = Q(IS)
C SOLVE FOR POINT P
CALL SOLVER
C STORE NEW VALUES
X(IP) = XP
Y(IP) = YP
Q(IP) = QP
2.0 CONTINUE
220 XP = XEND
C BNDRY(1) IS D/S BOUNDARY - Q SPECIFIED, C+ CHARACTERISTIC
CALL BNDRY(1)
C STORE BOUNDARY VALUES
X(IP) = XP
Y(IP) = YP
Q(IP) = QP
260 CONTINUE
C STORE IN D/S ARRAY
XMD2(I) = XP
YMD2(I) = YP
TMD2(I) = TP
QMD2(I) = QP
230 CONTINUE
C STORE IN REGION 2 BOUNDING CHARACTERISTIC
DO 240 I = 1, N2
XCP2(I) = X(I)
YCP2(I) = Y(I)
TCP2(I) = T(I)
QCP2(I) = Q(I)
240 CONTINUE
C STORE IN TDWN, QDWN, YDWN FOR GATE MOTION CALCULATIONS LATER
DO 250 I = 1, N2
TDWN(I) = TON2(I)
QDWN(I) = QDN2(I)
YDWN(I) = YDN2(I)
250 CONTINUE
C UPDATE D/S INDEX FOR SAVING T, Q AND Y
KMDN = N2
C STORE ENDING TIME AND DEPTH FOR INTERPOLATION
C IN POOL 2 AND FOLLOWING
T1 = TDN2(N2)
Y1 = YDN2(N2)
SUBROUTINE STROKER 74/74 OPT=1

WRITE(6,9000)ISEQ
GO TO 800

C REGION 3 - H AND Q SPECIFIED D/S
300 CONTINUE

C SET T PREVIOUS FOR STORING BOUNDARY VALUES ONLY
C EVERY TINC TIME ADVANCE
TUPPREV = 0.
TDNPREV = 0.

C COMPARE WAVE TRAVEL TIME WITH TIMINC TO SEE IF

C POOL IS TOO SHORT TO USE TIMINC
YCHECK = YDN2(N3)
CELERTY = C(YCHECK)
FN = N1 - 1
DXCHECK = XEND/FN
DX3 = CELERTY * TIMINC
IF (DX3/DXCHECK .GE. 2.) TIMINC = DXCHECK/CELERTY + 1
IF (DX3/DXCHECK .LT. 2.) WRITE(6,9040) TIMINC

C IF TIMINC TOO SHORT, TREAT AS ZERO LENGTH POOL
IF (TIMINC .LT. 0.6*TINC) GO TO 810

C NUMBER OF POINTS IN 3B REGION
N3B = 0

C NCROSS CHECKS THAT REGION 3 CROSSES X=0
NCROSS = 0

C MATCH WITH REGION 2 END POINT ON D/S BOUNDARY
XDN3(I) = XEND
TDN3(I) = TDN2(N2)
YDN3(I) = YDN2(N2)
QDN3(I) = QDN2(N2)
DO 304 I = 2, N0
310 N3 = I
XON3(I) = XEND
TDN3(I) = TDN3(I) + TIMINC
IF (TDN3(I) .GE. TMAX) GO TO 303
YDN3(I) = HDN(TDN3(I))
QDN3(I) = QDN(TDN3(I))
GO TO 304
302 TDN3(I) = TMAX
YDN3(I) = HDN(TMAX)
QDN3(I) = QDN(TMAX)
GO TO 306
304 CONTINUE
N3B = 1
306 CONTINUE
WRITE(6,9010) N3
IF (TDN3(I) .GE. TDN3(N3)) WRITE(6,9020) TDN3(I), TDN3(N3)
IF (YDN3(I) .GE. YDN3(N3)) STOP
330 XDN3(I) = XDN3(N3)
YCP3(I) = YDN3(N3)
TCP3(I) = TDN3(N3)
QCP3(I) = QDN3(N3)
335 XCM3(I) = XDN3(1)
YCM3(I) = YDN3(1)
TCM3(I) = TDN3(1)
QCM3(I) = QDN3(1)
340 DO 308 I = 1, N3
X(I) = XDN3(I)
Y(I) = YDN3(I)
SUBROUTINE STROKER 74/74 OPT=1

T(I)=TON3(I)
Q(I)=QON3(I)
308 CONTINUE
C X=UPSTREAM X=0 INTERSECT INDEX
K=1
M=N3-1
N=M
DO 322 I=1,M
C NALLX CHECKS THAT ALL XP ARE LESS THAN ZERO
NALLX=0
DO 320 J=1,N
IP=J
IS=J
IR=J+1
C SET R AND S POINT VALUES
XR=X(IR)
YR=Y(IR)
TR=T(IR)
QR=Q(IR)
XS=X(IS)
YS=Y(IS)
TS=T(IS)
QS=Q(IS)
305 C SOLVE FOR POINT P
CALL SOLVER
C SET NALLX IF ANY XP HAS NOT CROSSED X=0
IF(XF.GE.0.0) NALLX=1
C CHECK FOR CROSSING X=0
IF(XF.LT.0.0.AND.XS.GE.0.0) GO TO 318
310 CONTINUE
C STORE NEW VALUES
X(IP)=XP
Y(IP)=YP
T(IP)=TP
Q(IP)=QP
C CHECK FOR END POINTS
IF(J.EQ.1) GO TO 314
312 IF(J.EQ.N) GO TO 316
GO TO 320
C STORE IN C-
314 CONTINUE
XCMS(I+1)=XP
YCMS(I+1)=YP
TCMS(I+1)=TP
QCMS(I+1)=QP
C LAST POINT MAY ALSO BE ON + CHARACTERISTIC
C CHECK FOR THIS SITUATION
390 GO TO 312
C STORE IN C+
318 CONTINUE
XCP3(I+1)=XP
YCP3(I+1)=YP
TCP3(I+1)=TP
QCP3(I+1)=QP
GO TO 320
C INTERPOLATE UPSTREAM VALUES
C AT X=0
C SET NCROSS TO SHOW THAT X=0 WAS CROSSED
NCROSS=1
DELX=XP-XS
DX=AP
RATIO=DX/DELX
XUP3(K)=0.0
YUP3(K)=YP-RATIO*(YP-YS)
TUP3(K)=TP-RATIO*(TP-TS)
QUP3(K)=QP-RATIO*(QP-QS)
K=K+1
GO TO 310
C HAVE ALL POINTS CROSSED X=0 - IF SO STOP CALCULATING
IF(NALLX.EQ.0) GO TO 324
N=N-1
CONTINUE
C IF DID NOT CROSS X=0, STOP
IF(NCROSS.EQ.0) GO TO 346
C FIND X=0 INTERCEPT
DO 326 I=2,N3
IF(XCP3(I).GT.0.0) GO TO 326
DELX=XCP3(I)-XCP3(I-1)
DX=XCP3(I)
RATIO=DX/DELX
XUP3(K)=0.0
YUP3(K)=YCP3(I)-RATIO*(YCP3(I)-YCP3(I-1))
TUP3(K)=TCP3(I)-RATIO*(TCP3(I)-TCP3(I-1))
QUP3(K)=QCP3(I)-RATIO*(QCP3(I)-QCP3(I-1))
GO TO 328
326 CONTINUE
C SGRT UPSTREAM BOUNDARY ON T
DO 330 I=1,N
T(I)=TUP3(I)
TINOX(I)=1.
INDX(I)=0
ILOW=1
TLOW=T(I-1)
DO 332 J=1,K
IF(TINDX(I).LT.0.0) GO TO 332
IF(T(I).GE.TLOW) GO TO 332
ILOW=I
TLOW=T(I)
GO TO 336
332 CONTINUE
INDX(J)=ILOW
TINDX(ILOW)=-1.
DO 334 L=1,K
IF(TINDX(L).LT.0.0) GO TO 334
ILOW=I
TLOW=T(L)
GO TO 336
334 CONTINUE
336 CONTINUE
DO 338 I=1,K
SUBROUTINE STROKER 74/74 OPT=1

460 X(I)=XUP3(I)
Y(I)=YUP3(I)
T(I)=TUP3(I)
Q(I)=QUP3(I)

338 CONTINUE
DO 340 I=1,K
J=INDX(I)
XUP3(I)=X(J)
YUP3(I)=Y(J)
TUP3(I)=T(J)
QUP3(I)=Q(J)

340 CONTINUE
C STORE IN TDOW, QDOW, YDOW FOR GATE MOTION CALCULATIONS LATER
DO 342 I=1,N3
C UPDATE O/S INDEX TO SAVE T, Q AND Y
KDWN=KDWN+1
J=KDWN
TDOW(J)=TDN3(I)
QDOW(J)=QDN3(I)
YDOW(J)=YDN3(I)
IF(I.EQ.1) GO TO 342
C SAVE ONLY AFTER TINC TIME ADVANCE
IF(TDN3(I).GE.TDN3PREV+TINC) GO TO 341
KDNW=KDNW+1
GO TO 342
C UPDATE TDN3PREV (T PREVIOUS)
341 TDN3PREV=TDN3(I)-0.01
342 CONTINUE
C STORE IN TUP, QUP AND YUP FOR FUNCTIONS QDN AND HDN FOR NEXT POOL
DO 344 I=1,K
C UPDATE U/S INDEX TO SAVE T, Q AND Y
KUP=KUP+1
J=KUP
TUP(N)=TUP3(I)
QUP(N)=QUP3(I)
YUP(N)=YUP3(I)
IF(I.EQ.1) GO TO 344
C SAVE ONLY AFTER TINC TIME ADVANCE
IF(TUP3(I).GE.TUP3PREV+TINC) GO TO 343
KUP=KUP+1
GO TO 344
C UPDATE TUP3PREV (T PREVIOUS)
343 TUP3PREV=TUP3(I)-0.01
344 CONTINUE
C CHECK TO SEE IF 3B REGION MUST BE USED TO GET TO TMAX
IF(N3B.EQ.0) GO TO 800
C INITIALIZE FOR 3B REGION
505 XDN3B(1)=XEND
TDN3B(1)=TDN3(N3)
YDN3B(1)=YDN3(N3)
QDN3B(1)=QDN3(N3)
C START CALCULATIONS IN 3B REGION
GO TO 350
348 CONTINUE
WRITE(6,9030) IPOOL
C INCREASE TIME INCREMENT TO GET ACROSS POOL IN 40 POINTS
SUBROUTINE STROKER 74/74 OPT+1

TIMINC=TIMINC+TINC
IF(N3.LT.49) STOP
WRITE(6,9040) TIMINC
GO TO 300

C CALCULATE 3B REGION

350 CONTINUE

DO 354 I=2,40
N3B(I)=XDN3B(I)+XEND
TDN3B(I)=TDN3B(I-1)+TIMINC
IF(TDN3B(I).GE.TMAX) GO TO 352
YDN3B(I)=YDN(TDN3B(I))
QDN3B(I)=QDN(TDN3B(I))
GO TO 354

352 TDN3B(I)=TMAX
YDN3B(I)=YDN(TMAX)
QDN3B(I)=QDN(TMAX)
GO TO 356

354 CONTINUE

WRITE(6,9050) N3B
IF(TDN3B(I).GE.TDN3B(N3B)) WRITE(6,9060) TDN3B(I), TDN3B(N3B)
IF(TDN3B(I).GE.TDN3B(N3B)) STOP
XCP3B(I)=XDN3B(N3B)
YCP3B(I)=YDN3B(I)
TCP3B(I)=TDN3B(I)
QCP3B(I)=QDN3B(I)
DO 350 I=1,N3B
350 CONTINUE
C k=UPSTREAM X=0 INTERSECT INDEX

K=1
M=N3-1
N=N3B-1
DO 308 I=1,N
NALLX=0
L=N3B+1
DO 356 J=1,N
356 CONTINUE
C RETURN AND S POINT VALUES
XR=X(I)
YR=Y(I)
TR=Y(I)
QR=Q(I)
XS=X(J)
YS=Y(J)
TS=T(J)
QS=Q(J)
C SOLVE FOR POINT P
CALL SOLVER
SUBROUTINE STROKER 74/74OPT=1

C SET NALLX IF ANY XP HAS NOT CROSSED X=0
IF(XP.GE.0.0) NALLX=1
C
C CHECK FOR CROSSING X=0
IF(XP.LT.0.0.AND.XR.GE.0.0) GO TO 364
360 CONTINUE
C STORE NEW VALUES
X(IP)=XP
Y(IP)=YP
T(IP)=TP
C(IF(IP)=QP
C CHECK FOR END POINTS
IF(L.EQ.N3) GO TO 362
GO TO 366
C STORE IN CP
362 CONTINUE
XCPB(I+1)=XP
YCPB(I+1)=YP
TCPB(I+1)=TP
QCPB(I+1)=QP
GO TO 366
C INTERPOLATE UPSTREAM VALUES
C AT X=0
364 CONTINUE
DELX=XP-XR
DX=XP
RATIO=DX/DELX
XUPB(K)=0.0
YUPB(K)=YP-RATIO*(YP-YR)
TUPB(K)=TP-RATIO*(TP-TR)
QUPB(K)=QP-RATIO*(QP-QR)
K=K+1
GO TO 360
366 CONTINUE
C HAVE ALL POINTS CROSSED X=0 - IF SO, STOP CALCULATIONS
IF(NALLX.EQ.0) GO TO 370
IF(N.GE.N3) N=N-1
IF(I.GE.N3) GO TO 366
X(I)=XCPB(I+1)
Y(I)=YCPB(I+1)
T(I)=TCPB(I+1)
Q(I)=QCPB(I+1)
368 CONTINUE
370 CONTINUE
375 C SORT UPSTREAM BOUNDARY ON T
K=K-1
DO 372 I=1,K
T(I)=TUPB(I)
TINDX(I)=1.
372 INDEX(I)=
LOW=1
LOW=T(I)
DO 378 J=1,K
DO 374 I=1,K
IF(TINDX(I).LT.0.0) GO TO 374
IF(T(I).GE.TLOW) GO TO 374
LOW=I
TLOW=T(I)

CONTINUE

374

INDX(J)=ILOW
TINDEX(ILOW)=I.
DO 376 L=1,N
IF(TINDEX(L).LT.0.0) GO TO 376
ILOW=L
TLOW=T(L)
GO TO 376

376 CONTINUE

CONTINUE

378 CONTINUE

DO 380 I=1,N
J=INDX(I)
XUP3B(I)=X(J)
YUP3E(I)=Y(J)
TUP3B(I)=T(J)
QUP3B(I)=Q(J)

380 CONTINUE

C STORE IN TDWN,QDNW,YDWN FOR GATE MOTION CALCULATIONS LATER

DO 384 I=1,N
J=KDWN
TDWN(J)=TDN3B(I)
QDNW(J)=QDN3B(I)
YDNW(J)=YDN3B(I)

384 CONTINUE

KDOWN=KDOWN-1
GO TO 384

C SAVE ONLY AFTER TINC TIME ADVANCE

383 TON3B(I)=TON3B(I)+0.01

C STORE IN TUP, QUP AND YUP FOR FUNCTIONS QDN AND YDN FOR NEXT POOL

DO 386 I=1,N
J=KUP
TUP(N)=TUP3B(I)
QUP(N)=QUP3B(I)
YUP(N)=YUP3B(I)

386 CONTINUE

C SAVE ONLY AFTER TINC TIME ADVANCE

IF(TUP3B(I).GE.TUPPREV+TINC) GO TO 386

KUP=KUP-1
GO TO 386

C STORE TIME FOR MAX-FOR NEXT POOL

SAVTIME=TUP(KUP)

385 TUPPREV=TUP3B(I)-0.01

386 CONTINUE

C SAVE TIME FOR MAX-FOR NEXT POOL

SAVTIME=TUP(KUP)
C STORE IN C+3
DO 390 I=1,N3
XCP3(I)=XCP2(I)
YCP3(I)=YCP2(I)
TCP3(I)=TCP2(I)
QCP3(I)=QCP2(I)
390 CONTINUE
IF(TCP3(I).GE.TMAX) GO TO 800
C CLEAR FOR ANOTHER PASS THRU 3B BAND
DO 390 I=1,40
XUP3B(I)=0.
YUP3B(I)=0.
TUP3B(I)=0.
QUP3B(I)=0.
XDN3B(I)=0.
YDN3B(I)=0.
TDN3B(I)=0.
QDN3B(I)=0.
390 CONTINUE
XDN3B(I)=XEND
YDN3B(I)=YCP3B(I)
T DN3B(I)=TCP3B(I)
QDN3B(I)=QCP3B(I)
GO TO 390

C REGION 4 - NO BOUNDARY CONDITIONS
400 CONTINUE
C LOAD C+ FROM REGIONS 1 AND 2
M=N2
N=N4
J=N1
DO 410 I=1,M
J=J+1
XCP4(I)=XCP2(J)
YCP4(I)=YCP2(J)
TCP4(I)=TCP2(J)
QCP4(I)=QCP2(J)
410 CONTINUE

C INTERPOLATE C+ AT SAME X AS C
XCM4(I)=XCP4(I)
YCM4(I)=YCP4(I)
TCM4(I)=TCP4(I)
QCM4(I)=QCP4(I)
DO 440 I=2,N
DO 430 J=2,N3
IF(XCM3(J).GT.XCP4(I)) GO TO 430
DX=XCP4(I)-XCM3(J)
DELX=XCM3(J-1)-XCM3(J)
RATIO=DX/DELX
XCM4(I)=XCP4(I)
YCM4(I)=YCM3(J)+RATIO*(YCM3(J)-YCM3(J))
SUBROUTINE STROKER 74/74 OPT=1

TCM4(I)=TCM3(J)+RATIO*(TCM3(J-1)-TCM3(J))
QCM4(I)=QCM3(J)+RATIO*(QCM3(J-1)-QCM3(J))
GO TO 440

745 430 CONTINUE
750 440 CONTINUE
C K = INDEX FOR UPSTREAM BOUNDARY
K=1
DO 450 I=1,N4
X(I)=XCP4(I)
Y(I)=YCp4(I)
T(I)=TCP4(I)
Q(I)=QCP4(I)
450 CONTINUE
C SET X=0, T=0 UPSTREAM
XUP4(1)=0
YUP4(1)=TCP4(N4)
TUP4(1)=TCP4(N4)

760 DO 490 I=2,N4
DO 490 I=2,N4
X(I)=XCM4(I)
Y(I)=YCM4(I)
T(I)=TCP4(I)
Q(I)=QCM4(I)
490 CONTINUE

C SET R AND Q POINT VALUES
XR=X(IR)
YR=Y(IR)
TR=T(IR)
QR=Q(IR)
XS=X(IS)
YS=Y(IS)
TS=T(IS)
QS=Q(IS)

C SOLVE FOR POINT P
CALL SOLVER
C STORE NEW VALUES
X(IP)=XP
Y(IP)=YP
T(IP)=TP
Q(IP)=QP

785 C CHECK FOR X=0 CROSSING
IF(XP.LT.0.0.AND.XR.GE.0.0) GO TO 460
GO TO 470
C INTERPOLATE AT x=0 ON C+

790 K=K+1
DX=0.0-XP
DELR=XR-XP
RATIO=DX/DELR
XUP4(K)=0.0

795 YUP4(K)=YP+RATIO*(YR-YP)
TUP4(K)=TP+RATIO*(TR-TP)
QUP4(K)=QP+RATIO*(QR-QP)
470 CONTINUE
000 CONTINUE
WRITE(6,9000)ISEQ
C STORE IN TUP, QUP AND YUP FOR USE IN FUNCTIONS QDN AND HDN IN NEXT POOL
DO 490 I=1,K
TUP(I)=TUP4(I)
QUP(I)=QUP4(I)
YUP(I)=YUP4(I)
400 CONTINUE
GO TO 500
C REGION 5 - Q SPECIFIED ON BOTH BOUNDARIES
500 CONTINUE
C SET TPREV FOR STORING BOUNDARY VALUES ONLY EVERY
C TINC TIME ADVANCE
TPREV=-(TINC+0.01)
C LOAD C+ FROM REGIONS 1 AND 2
DO 510 I=1,N1
XCPS(I)=XCPl(I)
YCPS(I)=YCP1(I)
TCPS(I)=TCPl(I)
QCPS(I)=QCPl(I)
510 CONTINUE
M=M1
DO 520 I=1,N2
XCPS(I)=XCPS(I)
YCPS(I)=YCP2(I)
TCPS(I)=TCP2(I)
QCPS(I)=QCPS(I)
520 CONTINUE
M=M+1
DO 530 I=1,NB
X(I)=XCPS(I)
Y(I)=YCP5(I)
T(I)=TCP5(I)
O(I)=QCPS(I)
530 CONTINUE
C STORE END POINTS
540 CONTINUE
YUP(KUP)=Y(1)
TUP(KUP)=T(1)
QUP(KUP)=Q(1)
YDWN(KDWN)=Y(N5)
TDWN(KDWN)=T(N5)
QDWN(KDWN)=Q(N5)
C STOP AFTER TMAX
IF(T(N5)).GE.TMAX) GO TO 590
C STORE ONLY EVERY TIME T HAS ADVANCED BY TINC
C TO KEEP NUMBER OF POINTS IN YUP, YDWN, ETC.
C APPROXIMATELY THE SAME AS IN THE OTHER POOLS
DELTAT= T(1)-TPREV
IF(DELTAT.LT.TINC) GO TO 550
TPREV=T(1)
KUP=KUP+1
KDWN=KDWN+1
550 CONTINUE
DO 560 I=1,N5
560 CONTINUE
IR=I-1
SUBROUTINE STROKER 74/74 OPT=1

IS = IS + 1
IF (I.EQ.1) GO TO 560
IF (I.EQ.NS) GO TO 570
C SET R AND S POINT VALUES

650 XR = X(IR)
YR = Y(IR)
TR = T(IR)
QR = Q(IR)
XS = X(IS)
YS = Y(IS)
TS = T(IS)
QS = Q(IS)
C SOLVE FOR POINT P
CALL SOLVER

670 C STORE NEW VALUES
X(IP) = XP
Y(IP) = YP
T(IP) = TP
Q(IP) = OP
GO TO 580
C UPSTREAM BOUNDARY

580 XS = X(IS)
YS = Y(IS)
TS = T(IS)
QS = Q(IS)
XP = X(IS)
CALL BNDRY(0)
C

585 C STORE NEW VALUES
X(IP) = XP
Y(IP) = YP
T(IP) = TP
Q(IP) = OP
GO TO 580
C DOWNSTREAM BOUNDARY

590 XR = X(IR)
YR = Y(IR)
TR = T(IR)
QR = Q(IR)
XP = X(IR)
CALL BNDRY(1)
C

900 C STORE NEW VALUES
X(IP) = XP
Y(IP) = YP
T(IP) = TP
Q(IP) = OP
CONTINUE
GO TO 540
905 590 CONTINUE
WRITE(6,9000)ISEO
800 CONTINUE
GO TO 990
C TIME INCREMENT FOR CALCULATIONS TOO SHORT
C FOR REALISTIC RUN TIMES, SO TREAT AS ZERO
C LENGTH POOL
910 CONTINUE
C  CALCULATE D/S VALUES AT TINC INTERVALS FROM END OF REGION 2
C TO TIME TMAX
TP=TDN2(N2)-TINC
DO 820 I=N2,500
TP=TP+TINC
IF(TP.GT.TMAX) TP=TMAX
TDWN(I)=TP
ODWN(I)=ODN(TP)
YDWN(I)=YDN(TP)
KSAVE=I
IF(TP.GE.TMAX) GO TO 830
CONTINUE
C  SAVE SAME VALUES FOR U/S END OF POOL
CONTINUE
KMAX=KSAVE
DO 840 I=1,KSAVE
TUP(I)=TDWN(I)
OUP(I)=ODWN(I)
YUP(I)=YDWN(I)
CONTINUE
SAVE=TIME=TMAX
KUP=KMAX
KDOWN=KMAX
CONTINUE
C  SAVE MAX FOR NEXT POOL
TMAX=SAVE
C  SAVE KUP FOR USE IN FUNCTIONS QDN AND HDN IN NEXT POOL
KMAX=KUP
C  SAVE T, Q AND Y FOR MATCHING IN NEXT POOL
MAXP=KMAX
DO 1000 I=1,MAXP
SAVETUP(I)=TUP(I)
SAVEOUP(I)=OUP(I)
SAVEYUP(I)=YUP(I)
CONTINUE
C  
C  C  WRITE OUT T, Q AND Y AT BOTH ENDS OF POOL
C FOR USE IN GATE OPENING CALCULATIONS
C STRTYPE IS TYPE OF CONTROL STRUCTURE
C DEPTHS ARE IN THE CHANNEL, NOT IN THE GATE SECTION
WRITE(6,9100) IPOLK,KDOWN
WRITE(1,9070) IPOLK,KDOWN
WRITE(1,9080) (TDWN(I),I=1,KDOWN)
WRITE(1,9080) (ODWN(I),I=1,KDOWN)
WRITE(1,9080) (YDWN(I),I=1,KDOWN)
WRITE(6,9090) KDWN
WRITE(6,9100) KUP
WRITE(1,9110) IPOLK,KUP, STRTYPE(IPOLK)
WRITE(1,9080) (TUP(I),I=1,KUP)
WRITE(1,9080) (OUP(I),I=1,KUP)
WRITE(1,9080) (YUP(I),I=1,KUP)
RETURN
C FORMATS
9000 FORMAT(11H END REGION,15 )
9010 FORMAT(5H N3= ,I10)
9020 FORMAT(1X,4SHTRANSIENT TRAVEL TIME EXCEEDS TIME AVAILABLE /
SUBROUTINE STROKER

970      2BHTO TMAX. COMPUTATION HALTED.
975      /IX.6HT2MAX=,F10.3,10X,5HTMAX=,F10.3//)
980      43H AT UPSTREAM BOUNDARY IN REGION 3 FOR POOL ,15/)
980      FORMAT(1X,20H NEW TIME INCREMENT =,F10.1)
990      FORMAT(1X,31SH N38=,I10)
990      FORMAT(1X,17H STOP IN 38. TMIN=,F10.3,6H TMAX=,F10.3//)
990      FORMAT(1X,31SH TREATING AS ZERO LENGTH POOL )
END
SUBROUTINE SOLVER
C ROUTINE TO SOLVE FOR X, T, G AND Y AT POINT P
C USING VALUES AT POINT R ON + CHARACTERISTIC
C AND VALUES AT POINT S ON - CHARACTERISTIC
C METHOD AND NOTATION FOLLOW WYLIE
C
COMMON /SOLV/ XP,YP,TP,QP,QR,XR,TR,YS,TS,QS
C
COMMON /ALL/
C PHYSICAL PARAMETERS
. NMANN,50,GGRAV,P1,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
. MANN(50),BOTGROD(50),SYFONDY(50),SYFNLOS(50),SPEED(51),PUMPDQ
(51),DEDBAND(51),NGATE,GATECO(51),DSINV(50),USINV(51),
C WORKING VARIABLES
. TMAX,XEND,TINC,NXINC,NPOOLS,POOL,MYPPOOL,STABEG(50),
. STAEND(50),T1,Y1,STUDY,
C CANAL DESCRIPTION
. UPSTA(50),DOWNSTA(50),NOPOOLS,POOLTYPE(50),STRTYPE(51),
. TMAX,NUMTRANS,USRES,DSRES
C REAL NMANN,MANN
INTEGER TYPPOLL,POOLTYE,STRTYPE,STUDY
C
CR=CR(YR)
CS=CS(VS)
RR=RR(YR)
AR=AR(YR)
VR=VR/AR
SR=SR(VR,RR)
RS=RS(VS,RS)
VS=VS/QS/AS
SS=S(SVS,RS)
VP=VP(VR+VS/2.
VPPRCV=VP
YP=YP(VR+VS/2.
YPREV=YP
TP=TP(VR+CR+TR-(VS-CS)TS-XR+XS)/(VR+CR-VS+CS)
XR=XR(VR+CR)*(TP-TR)+XR
PS=BOTGROD(POOL)
AP=AP(YP)
AQ=VPPRCV AP
C IVALUE STOPS LOOPING IF SOLUTION DOES NOT CONVERGE
C ICOUNT=1
C
CONTINUE
RP=RP(YP)
AP=AP(YP)
VP=VP/AP
SP=SP(VP,RP)
CP=CP(VP)
TP=TP(2.*(XS-XR)+TR*(VP+CP+VR+CR)-TS*(VP-CP+VS-CS))/(VR+CR-(VS-CS)+2.*CP)
C4=GGRAV/2.*(1./CR+1./CP)
C2=VR+C4*CR-GGRAV/2.*(SR+SP-2.*SO)*(TP-TR)
C2=GGRAV/2.*(1./CR+1./CP)
C9=VS-C2+YS-GGRAV/2.*(SS+SP-2.*SO)*(TP-TR)
SUBROUTINE SOLVER    74/74    OPT=1

YP = \text{YPREV} = YP
YP = (C3 - C1) / (C2 + C4)

VPPREV = VP
VP = C3 - C4 * YP
AP = A(YP)
QP = VP * AP
ICOUNT = ICOUNT + 1

IF (ICOUNT > 50) WRITE(6,9000)
IF (ICOUNT > 50) STOP
IF (ABS(YPPREV - YP) > 0.001) GO TO 10
IF (ABS(VPPREV - VP) > 0.001) GO TO 10

XP = XR + ((VP + VR) / 2. + (CR + CP) / 2.) * (TP - TR)

C CHECK THAT FLOW IS NOT SUPERCRITICAL

CELRTY = C(YP)
AREA = A(YP)
IF (ABS(CELRTY * AREA / QP) > 1.) WRITE(6,9010) XP, YP, TP, QP
RETURN

C FORMATS

9000 FORMAT(1X,14HSTOP IN SOLVER)
9010 FORMAT(1X,22HSUPERCRITICAL FLOW AT
          /5X,3HX= ,F10.1/5X,3HY= ,F10.1/5X,
          /3HF= ,F10.1/5X,3HQ= ,F10.1/)
END
FUNCTION C

C Celerity

COMMON /ALL/

C Physical Parameters

NMANN, SO, GGRAV, PI, BOTTOM(SO), SIDSLOP(SO), RADIUS(SO), BOTGATE(SI),
MANN(SO), BOTGRAD(SO), SYFONDY(SO), SYFDLOS(SO), SPEED(SI), PUMPDQ
(SI), DEOBAND(SI), NGATES, GATECO(11, SI), OSINV(TO), USINV(SI);

C Working Variables

TMAX, XEND, TINC, NXINC, NPools, IPOOL, TYPPool, STABEG(SO),
STAEoD(SO), T1, Y1, STUDY,

C Canal Description

UPSTA(SO), DOWNSTA(SO), NPOOLS, POOLTP(SO), STRTYPE(SI),
TMAX, NUMTRNS, USRES, OSRES

C

REAL NMANN, MANN

INTEGER TYPPool, POOLTYP, STRTYPE, STUDY

C

C A=AREA
C TY=TOP WIDTH
C S=SQRT(GGRAV*A(V)/TW(Y))

RETURN
FUNCTION R 74/74 OPT-1

1
FUNCTION R(Y)
C HYDRAULIC RADIUS
C A-AREA
C P=WETTED PERIMETER

5
R=A(Y)/P(Y)
RETURN
END
FUNCTION A(Y)

C CROSS SECTIONAL AREA
C POOL IS TRAPEZOIDAL CHANNEL - TYPPOOL=1
C OR HORSESHOE TUNNEL - TYPPOOL=2
C OR CIRCULAR TUNNEL - TYPPOOL=3
C
C BOTTOM WIDTH AND SIDESLOPE OF TRAPEZOIDAL CHANNEL
C ARE SPECIFIED IN BOTTOM(POOL) AND SIDSLOP(POOL)
C
COMMON /ALL/
C PHYSICAL PARAMETERS

NMANN,50,GRAY,P1,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTTGAE(51),
MANN(50),BOTTGAD(50),SYNLOG(50),SPEED(51),PUMPDO
(51),DEBAND(51),NFACT,FAILO(11,51),DEINV(50),USINV(51),
D WORKING VARIABLES

UPSTA(50),DWHSTA(50),POOLS,PPOOLTY(TPY(50),STABEG(50),
STANA(50),11,Y1,STUDY,
C CANAL DESCRIPTION

.TPSTA(50),DOWNSTA(50),NPOOLS,POOLTY(50),STRTY(51),
C REAL NMANN,MANN
INTEGER TYPPOOL,POOLTY,STRTY,STUDY
C
IF(Y.LE.0.0) WRITE(6,9001) IPOOL,Y
IF(Y.LE.0.0) STOP
IF(TYPPOOL.EQ.1) GO TO 50
IF(TYPPOOL.EQ.2) GO TO 60
C IN HORSESHOE TUNNEL SECTION
C RAD IS INVERT TO CREST DISTANCE
RAD=RADIUS(POOL)
C CHECK FOR LOWEST ARC
Y1=RAD/4.*(3.-SORT(T))
C IF(Y.GT.Y1) GO TO 10
A=(Y-RAD)*SORT(RAD**2-(Y-RAD)**2)
+RAD**2*ASIN((Y-RAD)/RAD)+PI*RAD**2/2.
RETURN
C CHECK FOR AT SPRINGLINE
10 IF(ABS(Y-RAD/2.).GT.0.00001) GO TO 20
V3+RAD/2.
A=(Y1-RAD)*SORT(RAD**2-(Y1-RAD)**2)
+RAD**2*ASIN((Y1-RAD)/RAD)+PI*RAD**2/2.
-Y1-Y2)*SORT(RAD**2-(Y1-Y2)**2)
+RAD**2*ASIN((Y1-Y2)/RAD)
RETURN
C
C CHECK FOR BELOW SPRINGLINE
20 IF(Y.GT.RAD/2.) GO TO 30
V2+RAD/2.
A=(Y1-RAD)*SORT(RAD**2-(Y1-RAD)**2)
+RAD**2*ASIN((Y1-RAD)/RAD)+PI*RAD**2/2.
+(Y-Y2)*SORT(RAD**2-(Y-Y2)**2)
+RAD**2*ASIN((Y-Y2)/RAD)
-Y1-Y2)*SORT(RAD**2-(Y1-Y2)**2)
+RAD**2*ASIN((Y1-Y2)/RAD)-RAD*2/(Y-Y1)
RETURN
C CHECK FOR FILLED TUNNEL
30 IF(Y.GT.RAD) GO TO 40

60 Y2=RAD/2.
61 A=(Y1-RAD)*SQR2(RAD**2-(Y1-RAD)**2)
62 +RAD**2*ASIN((Y1-RAD)/RAD)+PI*RAD**2/2.
63 -(Y1-Y2)*SQR2((RAD**2-(Y1-Y2)**2)
64 -RAD**2*ASIN((Y1-Y2)/RAD)-RAD*(Y2-Y1)
65 +(Y-Y2)*SQR2((RAD/2.)*2-(Y-Y2)**2)
66 +(RAD/2.)*2*ASIN((Y-Y2)/(RAD/2.))
RETURN
40 WRITE(6,9000) Y,IPool
STOP

C TRAPEZOIDAL CHANNEL SECTION
50 CONTINUE

A=(BOTTOM(IPool)+SLOPE(IPool))*Y+Y
RETURN

C CIRCULAR TUNNEL SECTION
60 CONTINUE

RAD=RADIUS(IPool)
61 IF(ABS(Y-RAD).GT.0.00001) GO TO 70

C HALF-FULL
A=PI*RAD**2/2.
RETURN

70 IF(Y.GT.RAD) GO TO 60

C LOWER HALF OF SECTION

X=Y-RAD
THETA=ACOS(X/RAD)
A=THETA*RAD**2*X+RAD*SIN(THETA)
RETURN

C UPPER HALF OF SECTION
80 IF(Y.GT.RAD) GO TO 90

X=Y-RAD
THETA=ACOS(X/RAD)
A=(PI-THETA)*RAD**2+X-RAD*SIN(THETA)
RETURN

90 WRITE(6,9000) Y,IPool
STOP

C FORMAT
9000 FORMAT(1X,3HBDEPTH GREATER THAN TUNNEL HEIGHT AT
9001 SH Y=,F10.2,BH IN POOL ,15
9002 IF(13HBDEPTH IN POOL,15,2H=,F10.1/
9003 1X,20HSTOP IN AREA ROUTINE )
END
FUNCTION TW(Y)
C TOP WIDTH
C
COMMON /ALL/
C
C PHYSICAL PARAMETERS
NMANN,SO,GGRAV,PI,BOTLIES(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
MANN(50),BOTGRAD(50),SYFUNDY(50),SYFPAMS(50),SPEED(51),PUMPQQ,
(51),DEOBAND(51),NGATES,GATECQ(11,51),DSINVT(50),USINV(51),
C WORKING VARIABLES
2 XMAX,END,TINC,XNINC,PPOOLS,IPool,TPPOOL,STABEG(50),
STAEND(50),TI,Y1,STUDY,
C CANAL DESCRIPTION
UPSTA(50),WNSTA(50),NPOOLS,POOLTY(50),STRTYPE(51),
C
C REAL NMANN,MANN
INTEGER TYPPPOOL,POOLTYP,STRTYPE,STUDY

POOL IS TRAPEZOIDAL CHANNEL - TYPPPOOL=1
OR HORSESHOE TUNNEL - TYPPPOOL=2
OR CIRCULAR TUNNEL - TYPPPOOL=3

C ARE SPECIFIED IN BOTTOM AND SIDSLOP
IF(TYPPPOOL.EQ.1) GO TO 50
IF(TYPPPOOL.EQ.2) GO TO 60
C
C IN HORSESHOE TUNNEL SECTION
C RAD IS INVERT TO CREST DISTANCE
RAD=RADIUS(IPool)
A CHECK FOR LOWEST ARC
Y1=RAD/4.*((3.-SQRT(7.))
IF(Y.GT.Y1) GO TO 10
TM=2.*SQRT(RAD**2-(RAD-Y)**2)
RETURN
C CHECK FOR AT SPRINGLINE
10 IF(ABS(Y-RAD/2.).GT.0.00001) GO TO 20
THETA=ASIN((Y-Y2)/(RAD/2.))
RETURN
C CHECK FOR BELOW SPRINGLINE
20 IF(Y.GT.Y2) GO TO 30
THETA=ASIN((Y-Y2)/(RAD/2.))
RETURN
C CHECK FOR FILLED TUNNEL
30 IF(Y.GT.RAD) GO TO 40
THETA=ASIN((Y-Y2)/(RAD/2.))
RETURN
C TRAPEZOIDAL CHANNEL
C
C IN CIRCULAR TUNNEL

RETURN

C TRAPEZOIDAL CHANNEL
C
C IN CIRCULAR TUNNEL

RETURN

C TRAPEZOIDAL CHANNEL
C
C IN CIRCULAR TUNNEL

RETURN

C TRAPEZOIDAL CHANNEL
C
C IN CIRCULAR TUNNEL

RETURN

C TRAPEZOIDAL CHANNEL
C
C IN CIRCULAR TUNNEL

RETURN
CONTINUE
RAD=RAADIUS(IPOOL)
IF(ABS(Y-RAD).GT.0.00001) GO TO 70
C HALF-FULL
TW=2.*RAD
RETURN
IF(Y.GT.RAD) GO TO 80
C LOWER HALF OF SECTION
X=RAD-Y
THETA=ACOS(X/RAD)
TW=2.*RAD*SIN(THETA)
RETURN
C UPPER HALF OF SECTION
IF(Y.GT.2.*RAD) GO TO 90
X=Y-RAD
THETA=ACOS(X/RAD)
TW=2.*RAD*SIN(THETA)
RETURN
90 WRITE(6,9000) Y,IPool
STOP
C FORMAT
9000 FORMAT(1X,38HDEPTH GREATER THAN TUNNEL HEIGHT AT
3H Y=,F10.2,9H IN POOL ,15)
END
FUNCTION S(V,R)
C ENERGY GRADE
C
COMMON /ALL/
C PHYSICAL PARAMETERS
MANN,50,GGRAV,PI,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
MANN(50),BOTGRAD(50),SYFONDY(50),SYFNLOS(50),SPEED(51),PUMPDQ,
(51),DDBAND(51),NGATE,GATECO(11,51),DSSNW(50),USINVT(51),
C WORKING VARIABLES
TMAX,XEND,TINC,NCINC,NPOOLS,INPOOL,TYPPOL,STABEG(50).
STAEND(50),T1,Y1,STUDY,
C CANAL DESCRIPTION
UPSTA(50),DNSTA(50),NPOOLS,POOLTYP(50),STRTYPE(51),
MAXI,NUMTRNS,USRES,DSRES

REAL NMANN,MANN
INTEGER TYPPOOL,POOLTYP,STRTYPE,STUDY
C
C V IS VELOCITY
C R IS HYDRAULIC RADIUS
C GET MANNINGS N FOR THIS POOL
NMANN=MANN(IPOOL)
S=(V-NMANN/1.49)**2/R**(4./3.)
RETURN
END
FUNCTION P(Y)
C WETTED PERIMETER
C
C COMMON /ALL/
C PHYSICAL PARAMETERS
   NMANN,50,GOVAP,PI,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
   MANN(50),BOTGRAD(50),SYFONDY(50),SYFNLOS(50),SPEED(51),PUMPDD
   (51),DEBAND(51),NGATES,GATECO(11,51),DSINV(50),USINV(51).
C WORKING VARIABLES
   TMAX,XEND,INCI,NXINC,NPOOLS,IPPOOL,TYPPPOOL,STABEG(50),
   STAEND(50),YI,YI,STUDY,
C CANAL DESCRIPTION
   UPSIA(50),OWNSTA(50),NOPOLLS,POOLTY(50),STRTYPE(51),
   TMAXI,NUMTRANS,USRES,DSRES
C REAL NMANN,MANN
INTEGER TYPPOOL,POOLTY,STRTYPE,STUDY
C POOL IS TRAPEZOIDAL CHANNEL - TYPPPOOL=1
C OR HORSESHOE TUNNEL - TYPPPOOL=2
C OR CIRCULAR TUNNEL - TYPPPOOL=3
C
C BOTTOM WIDTH AND SLOPE OF TRAPEZOIDAL CHANNEL
C ARE SPECIFIED IN BOTTOM(POOL) AND SIDSLOP(POOL)
IF(TYPPPOOL.EQ.1) GO TO 10
IF(TYPPPOOL.EQ.2) GO TO 60
C IN TUNNEL SECTION
C RAD IS INVERT TO CREST DISTANCE
RAD=RADIUS(IPPOOL)
C CHECK FOR LOWEST ARC
   Y1=RAD/4.*(3.-SQRT(7.))
   IF(Y1.GT.Y) GO TO 10
   P=2.*RAD+ACOS((RAD-Y)/RAD)
   RETURN
   10 IF(ABS(Y-RAD/2.).GT.0.00001) GO TO 20
   Y2=RAD/2.
   P=2.*RAD+ACOS((RAD-Y)/RAD)+PI/2.-ACOS((Y-Y)/RAD)
   RETURN
C CHECK FOR BELOW SPRINGLINE
20 Y2=RAD/2.
   IF(Y.GT.Y2) GO TO 30
   P=2.*RAD+ACOS((RAD-Y)/RAD)+ACOS((Y2-Y)/RAD)
   -ACUS((Y2-Y)/RAD)
   RETURN
C CHECK FOR FILLED TUNNEL
30 IF(Y.GT.RAD) GO TO 40
   P=2.*RAD+ACOS((RAD-Y)/RAD)+PI/2.-ACOS((Y2-Y)/RAD))
   +RAD+ASIN((Y-Y2)/(RAD/2.))
   RETURN
40 WRITE(6,0000)Y,*POOL
   STOP
C TRAPEZOIDAL CHANNEL
50 CONTINUE
   P=BOTTOM(IPPOOL)+Y+2.*SQRT(SIDSLOP(IPPOOL)**2+1.)
   RETURN
C IN CIRCULAR TUNNEL
60 CONTINUE
FUNCTION P

NAD=RADIUS( IPOOL)

IF(ABS(Y-RAD).GT.0.00001) GO TO 70

60 C HALF-FULL
    P=PI*RAD
    RETURN

70 IF(Y.GT.RAD) GO TO 80

C LOWER HALF OF SECTION

65 X=RAD-Y
    THETA=ACOS(X/RAD)
    P=2.*RAD+THETA
    RETURN

C UPPER HALF OF SECTION

70 IF(Y.GT.Z.+RAO) GO TO 90

75 90 WRITE(0,9000) Y,POOL
    STOP

C FORMAT

9000 FORMAT(1X,3HDEPTH GREATER THAN TUNNEL HEIGHT AT
   . 3H Y=.F10.2,9H IN POOL ,IS)

END
SUBROUTINE BNDRY(IUPRDWN)
C UPSTREAM OR DOWNSTREAM BOUNDARY
C IUPRDWN ZERO FOR UPSTREAM BOUNDARY
C IUPRDWN ONE FOR DOWNSTREAM BOUNDARY
C E IS SPECIFIED IN FUNCTIONS GUPST AND QDN
C NOTATION FOLLOWS WYLIE
C
COMMON /SOLV/ XP,YP,TP,OP,XR,YR,TR,QR,XS,YS,TS,QS
C.
COMMON /ALL/
C PHYSICAL PARAMETERS
. . NMANN,SO,GGRAV,PI,90TOM(50),SIDSLDP(50),RADIUS(50),BOTGATE(51),
. . MANN(50),BOTGRAD(50),SYFONDY(50),SYFNLDS(50),SPEED(51),PUMPDQ
. . (51),DEDBAND(51),NGATES,GATECO(11,51),DSINTV(50),USINTV(51),
C WORKING VARIABLES
. . TMX,XEND,TINC,NXINC,NPOOL,IPPOOL,TYPPOOL,STABEG(50),
. . STAEND(50),T1,Y1,STUDY,
C CANAL DESCRIPTION
. . UPST(50),DNST(50),NPOOLP,POOLYP(50),STRTYP(51),
. . TMAXI,NUMTRANS,USRES,DSRES
C
REAL NMANN,MANN
INTEGER TVPPOOL,POOLTYP,STRTYP,STUDY
C
IF(IUPRDWN.EQ.0) GO TO 20
C DOWNSTREAM BOUNDARY
C INTERSECT C+ WITH BOUNDARY
30 CR=C(VR)
RR=R(VR)
AR=A(YR)
VR=QR/AR
SR=SlVR,RR)
V=VR
T=TR+(XP-XR)/(VR+CR)
TPREV=T
Y=VR
X=XEND
SO=BOTGRAD(IPPOOL)
AP=A(Y)
Q=QDN(T)
V=Q/AP
QPREV=Q
CONTINUE
40 YPREV=Y
RP=R(V)
SP=S(V,RP)
CP=C(Y)
Y=VR+(2.*(VR-V)/GGRAV-(SR+SP-2.*SO)*(T-TR))
50 .//(/1./CR+1./CP)
RP=R(V)
SP=S(V,RP)
CP=C(Y)
AP=A(Y)
V=VR-GGRAV/2.*(SR+SP-2.*SO)*(T-TR)
55 -GGRAV/2.*(1./CR+1./CP)+(Y-VR)
TPREV=T
SUBROUTINE BNDRY 74/74 OPT=1

60
T=2.*(XP-XR)/(V+VR+Ca+cP)+TR
QPREV=Q
Q=Q/ST
IF(ABS(QPREV-Q).GT.0.001) GO TO 10
IF(ABS(TPREV-T).GT.0.001) GO TO 10
IF(ABS(YPREV-Y).GT.0.001) GO TO 10
V=V
TP=T
YP=Y
QP=Q
C CORRECT FOR CHANGE IN CHANNEL CROSS SECTION AT NO GATE D/S BOUNDARY
IF(IPool.EQ.1) RETURN
C YBEFORE IS D/S Y CALCULATED FROM INITIAL CONDITIONS
C AND CONTINUITY OF Q
C YAFTER IS D/S Y CALCULATED FROM ENERGY BALANCE
C YBEFORE AND YAFTER SHOULD BE NEARLY EQUAL FOR
C A VALID SOLUTION ACROSS THE CHANGE IN SECTION
IF(STRTYPE(IPool-1).EQ.4) WRITE(6,9000) YP
IF(STRTYPE(IPool-1).EQ.4) WRITE(6,9010) YP
IF(STRTYPE(IPool-1).EQ.5) WRITE(6,9000) YP
IF(STRTYPE(IPool-1).EQ.5) WRITE(6,9010) YP
RETURN
C UPSTREAM BOUNDARY
C INTERSECT C- WITH BOUNDARY
20 CONTINUE
CS=C(YS)
RS=R(YS)
AS=A(YS)
VS=Q/AS
SS=SS(VS,RS)
V=VS
Y=YS
X=0.
S0=BOTGRAD(IPool)
AP=A(Y)
Q=Q/ST(T)
V=Q/AP
QPREV=Q
CONTINUE
YPREV=Y
RP=R(Y)
SP=SP(V,RP)
105 CP=C(Y)
Y=YS-2.*(VS-V)/GGRAV-(SS+SP-2.*S0)*(T-TS))/(1./CS+1./CP)
RP=R(Y)
SP=SP(V,RP)
CP=C(Y)
AP=A(Y)
V=VS-GGRAV/(SS+SP-2.*S0)*(T-TS)
+GGRAV/2.*(VS+2.*S0)*(T-TS)
TPREV=T
CONTINUE
SUBROUTINE BNDRY

115  
T=2.*(XP-XS)/(V+VS-CS-CP)+TS
QPREV=Q
Q=QUPST(T)
V=Q/AP
IF(ABS(QPREV-Q).GT.0.001) GO TO 30
120  
IF(ABS(TPREV-T).GT.0.001) GO TO 30
IF(ABS(YPREV-Y).GT.0.001) GO TO 30
VP=V
YP=Y
125  
QP=Q
RETURN
C FORMATS
9000 FORMAT(1X,9HY BEFORE=,F10.3)
9010 FORMAT(1X,9HY AFTER=,F10.3/)
130  
END
FUNCTION HDN(T)
C DOWNSTREAM BOUNDARY DEPTH
C T IS TIME IN SEC
C COMMON /ALL/
C PHYSICAL PARAMETERS
   NMANN, SO, GGRAV, PI, BOTTOM(50), SIDSLOP(50), RADUS(50), BOTGATE(51),
   MANN(50), BOTGRAD(50), BVFONDY(50), BVFMLOG(50), SPED(51), PUMPQQ
   (51), DEBAND(51), NGATES, GATECO(11, 51), DSINV(50), USINV(51),
C WORKING VARIABLES
   TMAX, XEND, TINC, NPOOLS, IPOOL, TYPPPOOL, STABEG(50),
   STAEND(50), T1, Y1, STUDY,
C CANAL DESCRIPTION
   UPOSTA(50), DWNSTA(50), NPOOLS, POOLTYP(50), STRTYPE(51),
C REAL NMANN, MANN
INTEGER TYPPPOOL, POOLTYP, STRTYPE, STUDY
C SCRATCH AREA TO BE OVERLAI IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
COMMON /SCRATCH/
   .DUM(2570)
   .H(48)
   .DUMMY2(5668)
   .NMAX
   .SAVETUP(500), SAVEDUP(500), SAVETUP(500)
C MAKE ENERGY BALANCE DEPTH CHANGE AT NO GATE CHANGE IN CROSS-SECTION
   IF(IPOOL.EQ.1) GO TO 10
   IF(STRTYPE(IPOOL-1).EQ.4) GO TO 40
   IF(STRTYPE(IPOOL-1).EQ.5) GO TO 40
10 CONTINUE
   TIMEHR=T/3600.
   ITIME=TIMEHR
   ITIME=ITIME+1
   IF(ITIME.LE.1) ITIME=1
   IF(ITIME.GT.48) ITIME=48
   IF(ITIME.GT.1) GO TO 20
C T1, Y1 IS LAST COMPUTATION POINT ON D/S BOUNDARY
C IN REGION 2
C MATCH WITH T1, Y1 IN 1ST HR
C THIS ASSUMES T1, LT. 1 HR
C CHECK WAVE TRAVEL TIME IF IN DOUBT
   IF(T1.GE.3600.) WRITE(6,9000) T1
   DELV=H(21)-V1
   DELT=3600.-T1
   I=ITIME-1
50 D=DELV/DELT
   HDN=Y1+D
   RETURN
C LINEARLY JOIN H(ITIME) AND H(ITIME+1)
20 IF(ITIME.EQ.48) GO TO 30
   DELT=3600.
   ITIME=1
   RETURN
   HDN=H(ITIME)+H(ITIME+1)
   D=DELH/DELT
   HDN=Y1+D
   RETURN
FUNCTION HDN  74/74  OPT=1

TREG=1
TSEG=TREG+3600.
DT=T-TREG
DY=DELY+DT/DELT
HDN=TIME*DY
RETURN

C HOLD DEPTH CONSTANT AFTER 47TH HOUR
30  HDN=TIME
RETURN

C MODIFIED NEWTON'S METHOD FOR ENERGY BALANCE
C YUP+VUP**2/2+G(YDOWN+YDOWN+2/2+G
40  CONTINUE

C SET DOWNSTREAM VALUES
C FIND DEPTH IN DOWNSTREAM SECTION
DO 50 I=2,KMAX
J=I
IF(T.LE.SAVE(T)) GO TO 60
50  CONTINUE
C T GREATER THAN TMAX - USE FINAL DEPTH
YD=SAVEUP(KMAX)
GO TO 70

C INTERPOLATE FOR Y
60  CONTINUE
DELY=SAVEUP(J)-SAVEUP(J-1)
DELT=SAVEUP(J)-SAVEUP(J-1)
DY=DELY+DY/DELT
YD=SAVEUP(J-1)+DY

C GET AREA AND Q IN DOWNSTREAM SECTION
70  CONTINUE
C QDN RETURNS Q UPSTREAM OF TURNOUT
QD=QD(T)
C GET TURNOUT Q
CALL TURNOUT(DELG,T)
QDS=QD-DELG
QUS=QD

C SET IPOOL FOR AREA ROUTINE
IPOOL=IPOOL-1
YP=POOL=POOL-TYP(POOL)
C IF SYPHON, CORRECT DEPTH FOR SYPHON LOSSES AND INVERT DROP
IF(STRYPE(IPOOL).NE.4) GO TO 75
C SYPHON EFFECTS ONLY - NO CHANGE IN CROSS-SECTION OF CHANNEL
C CHANGE IN CROSS-SECTION, IF ANY, IS TAKEN INTO ACCOUNT LATER
Y=YD
AB=ABELOW+YD
SYFON=SYFON+(IPOOL)
SYFON=SYFON+QDS*(2+GR(ABELOW+2)+YD+YSYPH)SYFON+QDS**2
RAD=ADIUS(IPOOL)
CONTINUE

IF(TYPPOOL.EQ.2.AND.Y.GE.RAD) Y=rad-0.1
IF(TYPPOOL.EQ.3.AND.Y.GE.RAD) Y=2*rad-0.1
IF(Y.LT.0.0) Y=0.1
F=F+QDS**2/(2.*GR(AB+A(Y+2)-C1)
FR=FR+QDS**2/(G(GR(AB+A(Y+3)-TW(Y)
YF=F/FPRIME
Y=F+FPRIME
FUNCTION HDN 74/74 OPT=1

115 IF(ABS(Y-YPREV).GT.0.001) GO TO 71
YD+T
C CALCULATE EFFECT DUE TO CHANGE IN CROSS-SECTION
75 CONTINUE
AD=A(YD)
120 C RESET IPOOL FOR U/S POOL
IPool=IPool+1
TYPool=TYPool(IPool)
YU=YO
125 ICOUNT=1
RAD=RADIUS(IPool)
80 CONTINUE
YPREV+YU
F=-YU+YD+(1./(2.*GRAV)+(QDS**2/AD**2-QUS**2/A(YU)**2))
Fprime=-1.+QUS**2*TW(YU)/(GRAV+A(YU)**3)
YU=F/Fprime
IF(YU.LT.0.1) YU=0.1
IF(TYPool.NE.2) GO TO 90
IF(YU.GE.RAD) YU=RAD-0.1
135 90 CONTINUE
IF(TYPool.NE.3) GO TO 100
IF(YU.GE.2.*RAD) YU=2.*RAD-0.1
100 CONTINUE
ICOUNT=ICOUNT+1
140 IF(ICOUNT.GT.20) WRITE(6,9010)
IF(ICOUNT.GT.20) STOP
IF(ABS(YU-YPREV).GT.0.001) GO TO 80
HDN=YU
RETURN
C FORMATS
9000 FORMAT(1X,3HT1=,F10.3,7H IN HDN )
9010 FORMAT(1X,11HSTOP IN HDN )
END
FUNCTION GDN 74/74 OPT=1

FUNCTION GDN(T)
C DOWNSTREAM FLOW AT EACH POOL
C SPECIFY Q AT POOL 1 IN Q0
C Q CONTINUITY IS MAINTAINED FOR ALL OTHER POOLS
C T IS TIME IN SEC
C
COMMON /ALL/
C PHYSICAL PARAMETERS
   MANN(50),GGRAV,P1,BOTTOM(50),SIDSDP(50),RADIUS(50),BOTGATE(51),
   MANN(50),BOTRAD(50),SYHFNDY(50),SYFNLOS(50),SPEED(51),PUMPDQ
   (51),DEBAND(51),NGATES,GATECD11,DSINV(50),USINV(51),
C WORKING VARIABLES
   TMAX,XEND,TINC,XINC,NPOOLS,IPPOOL,TPPOOL,STABEG(50),
   ETAEND(50),T1,Y1,STUDY,
C CANAL DESCRIPTION
   UPSTA(50),DNSTA(50),NPOOLS,POOLTY2(50),STRTYPE(51),
   TMXIII,MUNA3NS,USRES,DSRES
C
REAL MANN,MANN
INTEGER TYPPOOL,POOLTYP,STRTYPE,STUDY

C SCRATCH AREA TO BE OVERLAPPED IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
COMMON /SCRATCH/
   DUMMY1(8618),OD(46),DUMMY2(5620),KMAX,
   SAVETUP(500),SAVEUP(500),SAVEU9P(500)
C
C USE SAVETUP, SAVEUP FOR SECOND AND FOLLOWING POOLS
   IF(IPPOOL.EQ.2) GO TO 10
C Q STEPS AT EACH HOUR
   TIMEHR=T/3600.
   ITIME=ITIME+1
   IF(ITIME.LT.1) ITIME=1
   IF(ITIME.GT.48) ITIME=48
   QDN=QD(ITIME)
C CHECK FOR TURNOUT
   CALL TRNOUT(DELQ,T)
   QDN=QDN+DELQ
RETURN
C Q FOR SECOND AND FOLLOWING POOLS IN EACH REACH
C LINEAR INTERPOLATION OF UPSTREAM Q FROM PREVIOUS POOL
10 CONTINUE
   LIM=KMAX-1
   DO 20 I=1,LIM
      K=I
      IF(T.GE.SAVETUP(I).AND.T.LT.SAVETUP(I+1)) GO TO 30
20 CONTINUE
   IF(T.GE.SAVETUP(KMAX)) GO TO 30
C LET T RUN BEYOND TMAX FOR MOST UPSTREAM POOL FOR LAST POINT
   IF(T.GE.SAVETUP(K)) AND (IPPOOL.EQ.NPOOLS) GO TO 30
   WRITE(6,9000) T,IPPOOL
STOP
55 CONTINUE
   DT=T-SAVETUP(K)
FUNCTION QDN 74/74 OPT-1

DELT = SAVETUP(K+1) - SAVETUP(K)
RATIO = OT/DELT
QDN = SAVETUP(K) + RATIO * (SAVETUP(K+1) - SAVETUP(K))

C CHECK FOR TURNOUT
CALL TRNOUT(DELQ, T)
QDN = QDN + DELQ
RETURN

C FORMAT
9000 FORMAT(1X, 18HO(T) NOT FOUND AT , F10.3,
13H SEC FOR POOL, !R/) END
FUNCTION QUPST(T)
C UPSTREAM FLOW
C SPECIFY Q IN QU FOR MOST UPSTREAM POOL
C T IS TIME IN SEC
5
C
COMMON /ALL/
C PHYSICAL PARAMETERS
   NMANN,SO,SSG,GRAV,PI,BOTTOM(50),SIDSLOP(50),RADIUS(50),BOTGATE(51),
   MANN(50),BOTGATE(50),SYFONDY(50),SYFNLOS(50),SPEED(51),PUMPQ
   (51),REDGATE(51),NGATES,GATEG(11,51),DSINV(50),USINV(51),
C WORKING VARIABLES
   TMAX,END,TINC,MAINC,NPOOLS,POOLTYPE(50),STABEG(50),
   STAEND(50),TI,YI,STUDY,
C CANAL DESCRIPTION
   UPSTA(50),DNWSTA(50),NPOOLS,POOLTYPE(50),STRTYPE(51),
C
   REAL NMANN,MANN
   INTEGER TYPPOOL,POOLTYPE,STRTYPE,STUDY
20
C
C SCRAP AREA TO BE OVERLAID IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
COMMON /SCRATCH/
   DUMMY1(/5666),QU(40),DUMMY2(7073)
C
C Q STEPS AT EACH HOUR
   TIMEHR=T/3600.
30
   ITIME=TIMEHR
   ITIME=ITIME+1
   IF(ITIME.LT.1) ITIME=1
   IF(ITIME.GT.48) ITIME=48
   QUPST=QU(ITIME)
RETURN
35
END
SUBROUTINE TRNOUT(DEL0, T)
C ROUTINE TO GET TURNOUT Q FOR USE IN FUNCTION QDN
C
COMMON /ALL/
C PHYSICAL PARAMETERS
  NMANN, GGRAV, PI, BOTTOM, SIDSLOP, RADIUS, BOTGATE, MANN,
  BOTGRAD, SYFONNY, SYFNLOS, SPEED, PUMPQ
  (50), DEDBAND(51), NGATES, GATECO(11, 51), DSINV(50), USINV(51),
C WORKING VARIABLES
  TMAX, XEND, TINC, NXINC, NPOOLS, IPOOL, TYPPOOL, STDBEGIN, TEND,
  STAEND, T1, Y1, STUDY,
C CANAL DESCRIPTION
  USPAT(50), DNPSTA(50), NPOOLS, TYPPOOL(50), STRTYPE(50),
  TMXI, NUMTRANS, USRES, DSRES
C
REAL NMANN, MANN
INTEGER TYPPOOL, POOLTYP, STRTYPE, STUDY
C
COMMON /SCRATCH/
  DUMMY1(8714), QT(48, 50), DUMMY2(4673)
C
C INITIALIZE TURNOUT Q
  DEL0 = 0.0
C
C NUMTRANS IS NUMBER OF TURNOUTS IN THIS REACH
  IF(NUMTRANS.EQ.0) RETURN
  TIMEHR = T/3600.
  ITIME = TIMEHR
  ITIME = ITIME + 1
  IF(TIME.LT.1) ITIME = 1
  IF(TIME.GT.48) ITIME = 48
  DEL0 = QT(ITIME, IPOOL)
RETURN
END
SUBROUTINE GATEMO
C ROUTINE TO CALCULATE GATE OPENINGS
C AND PUMP SCHEDULES, IF ANY
C
C COMMON /ALL/
C PHYSICAL PARAMETERS
NMANN, GGRAV, PI, BOTTOM(50), SIDESLOP(50), RADIUS(60), BOTAITE(51),
MANN(50), BOTMAD(50), SYFONDY(50), SYFNLOS(50), SPEED(51), PUMP(50)
C WORKING VARIABLES
TMAX, XEND, TINC, NXINC, NPools, IPOOL, TPPOOL, STABEG(50),
STACND(50), TY, TSTUDY,
C CANAL DESCRIPTION
UPSTA(50), DWNSTA(50), NPools, POOLYP(50), STRTYPE(51),
C REAL NMANN, MANN
INTEGER TYPOOL, POOLYP, STRTYPE, STUDY
C
C SCRATCH AREA TO BE OVERLAID IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
COMMON /SCRATCH/
DUMMY(12000)
, TD(500), QO(500), YD(500)
, TV(500), QU(500), UY(500)
, DUMMY(787)
C
C WRITE(6, 9000)
C READ U/S T,Q,H OF FIRST STRUCTURE
C IF THIS IS A GATE, THE DEPTH D/S OF THE GATE MUST
C HAVE BEEN SPECIFIED IN DRES
C IF THIS IS A PUMP, SKIP CALCULATIONS AS Q WAS SCHEDULED
READ(1, 9010) LMAX
READ(1, 9020) (TU(I), I=1, LMAX)
READ(1, 9020) (QU(I), I=1, LMAX)
READ(1, 9020) (UY(I), I=1, LMAX)
C
C WRITE(6, 9030) IPOOL, KMAX, ISTRYP
C WRITE(6, 9030) JPOOL, LMAX
C
C CALCIULATE GATE OPENINGS FOR GATE OPENING INTO
C CONSTANT LEVEL RESERVOIR AT D/S END OF REACH
YDN=DRES
IPOOL=0
JPOOL=1
KMAX=0
ISTRYP=0
WRITE(6, 9030) IPOOL, KMAX, ISTRYP
C WRITE(6, 9030) JPOOL, LMAX
C
C CALCULATE GATE OPENINGS AT TIMES T+TINC
TMAX+TU(LMAX)
C LIMIT TO 24 HOURS
IF(TMAX.GT.86400.) TMAX=86400.
T=0.0
C CALCULATE "AT 5 MINUTE INTERVALS
TINC=300.
SUBROUTINE GATEMO 74/74 OPT=1

4 CONTINUE
C INTERPOLATE UPSTREAM
60 M=LMAX-1,
DO 7 I=1,M
K=I
IF(T.GE.TU(I).AND.T.LE.TU(I+1)) GO TO 8
7 CONTINUE
8 CONTINUE
DO T=TU(K),TU(K+1)-TU(K)
RATIO=DT/DELT
QU=QU(K)+RATIO*(QU(K+1)-QU(K))
70 C Q THRU GATE IS Q/U/S OF GATE LESS TURNOUT Q AT GATE
IPPOOL=1
CALL TRNOUT(DELT,I)
IPPOOL=0
QDN=QUP-DELT
YUP=QU(K)+RATIO*(YU(K+1)-YU(K))

75 C U/S SIDE OF GATE
C SET CHANGE IN INVERT ELEVATION COMING INTO GATE
EG=USINVT(S1)
SYPHON=0.0
YSYPHON=0.0
IPPOOL=IPPOOL+1
TYPOOL=POOLTYPE(IPPOOL)
IPUPDOWN=1
CALL GATEY(YUP,YC,EG,SYPHON,YSYPHON,YGATE,QDN,IPUPDOWN)
YUP=YGATE
IPPOOL=IPPOOL+1
C CHECK THAT U/S DEPTH IS GREATER THAN D/S DEPTH
IF(YUP.LE.YDN) GO TO 500

80 C
C CALCULATE GATE OPENING
G=YUP
IF(ABS(YUP-YDN).LT.0.01) GO TO 11
9 GPREV=G
G=2.0
IF(G.GT.0.0) G=0.0
IF(G.LT.-1.0) G=-1.0
C SET IPPOOL FOR COEFFICIENT ROUTINE
IPPOOL=1
COEFF=CD(G)
C RESET IPPOOL
IPPOOL=0
ADN=BOTGATE(S1)*YDN
C SET IPPOOL FOR AREA ROUTINE
IPPOOL=IPPOOL+1
TYPOOL=POOLTYPE(IPPOOL)
AUP=YUP*BOTGATE(S1)
C RESET IPPOOL FOR D/S POOL
IPPOOL=IPPOOL+1
G=QDN/(BOTGATE(S1)+COEFF*SQR(T.GE.GRAV*(YUP-YDN)+QDN+2/AUP+2))
IF(G.GT.YUP) G=YUP
G=0.5*(G+GPREV)
IF(ABS(G-GPREV).GT.0.001) GO TO 9
SUBROUTINE GATEMO 74/14 OPT=1

CONTINUE
WRITE(2,9040) T,G,YUP,YDN
T=T+TINC
IF(T.EE.TMAX) GO TO 4
C READ IN DOWNSTREAM AND UPSTREAM T,Q,H OF GATE
C INDEX IPOOL IS ON DOWNSTREAM SIDE OF GATE
C GATE TYPE IS DATA ON DOWNSTREAM SIDE OF GATE
C ISTRYP=0 POOL NOT USED
C =1 PUMPS
C =2 NORMAL GATE
C =3 SYMPHON DOWNSTREAM OF GATE
C =4 SYMPHON WITH NO GATE
C =5 NO GATE
CONTINUE
READ(1,9030) IPOOL,KMAX,ISTRYP
IF(EOF(1)) 400,20
CONTINUE
WRITE(6,9030) IPOOL,KMAX,ISTRYP
READ(1,9020) (TD(I),I=1,KMAX)
READ(1,9020) (QD(I),I=1,KMAX)
READ(1,9020) (YD(I),I=1,KMAX)
READ(1,9030) JPOOL,LMAX
IF(EOF(1)) 35,30
WRITE(6,9030) JPOOL,LMAX
READ(1,9020) (TU(I),I=1,LMAX)
READ(1,9020) (QU(I),I=1,LMAX)
READ(1,9020) (YU(I),I=1,LMAX)
GO TO 36
C FINISHED IF LAST STRUCTURE IS NOT STROKED
IF(STUDY.EQ.1) GO TO 400
CONTINUE
C SKIP NO GATE BOUNDARIES
C ISTRYP.EQ.4.OR.ISTRYP.EQ.5) GO TO 10
C CALCULATE GATE OPENINGS OR PUMP DISCHARGES AT TIMES T+TINC
TMAX=T+TMAX
C LIMIT TO 24 HOURS
IF(TMAX.GT.86400.) TMAX=86400.
T=0.0
C CALCULATE AT 5 MINUTE INTERVALS
TINC=300.
CONTINUE
C INTERPOLATE DOWNSTREAM
M=KMAX-1
DO 50 I=1,M
K=I
IF(T.GE.TD(I).AND.T.LT.TD(I+1)) GO TO 60
CONTINUE
50 CONTINUE
DU=T-TD(K)
DELT=TD(K+1)-TD(K)
RATIO=DU/DELT
QDN=QD(K)+RATIO*(QD(K+1)-QD(K))
YDN=YD(K)+RATIO*(YD(K+1)-YD(K))
C INTERPOLATE UPSTREAM
SUBROUTINE GATEMO

C SET U/S DEPTH JUST IN CASE THIS IS A GATE AT MOST U/S END OF REACH
YUP=USRES

175 IF(IPool.EQ.NPool) GO TO 85
M=LMAX+1
GO TO 70
K=1
IF(I.GE.TU(I).AND.T.LE.TU(I+1)) GO TO 80
CONTINUE
70 CONTINUE
80 CONTINUE
DT=T-TU(K)
DEL=DEL(TU(K+1)-TU(K))
RATIO=DT/DEL
YUP=YU(K)+RATIO*(YU(K+1)-YU(K))
CONTINUE
85 CONTINUE
C IS THIS A PUMP
IF(Istrtyp.EQ.1) GO TO 600
IF(Istrtyp.NE.2) GO TO 200
C CALCULATE Y IN NORMAL GATE STRUCTURE
C SET CHANNEL X-SECTION TYPE FOR AREA ROUTINE
TYPPOOL=POOLTYPM(IPOOL)
C IPOOL HAS BEEN READ IN ABOVE
C ZERU OUT SYPHON PARAMETERS
SYPHON=0.0
YSYPHON=0.0
C D/S SIDE OF GATE
C INVERT DROP TO TRAPEZOIDAL CHANNEL
C EC IS CHANNEL INVERT ELEVATION
EC=0.0
EG=DSINV(IPOOL)
C IUPRDN=0 FOR D/S SIDE OF GATE, =1 FOR U/S SIDE
CALL GATEY(QDN,YDN,EC,EG,SYPHON,YSYPHON,YGATE,QDN,IUPRDN)
YDN=YGATE
C U/S SIDE OF GATE
IF(IPool.EQ.NPool) GO TO 310
C SET CHANGE IN INVERT ELEVATION COMING INTO GATE
EC=0.0
EG=USINV(IPOOL)
IPool=IPool+1
TYPPOOL=POOLTYPM(IPOOL)
IUPRDN=1
CALL GATEY(QUP,YUP,EC,EG,SYPHON,YSYPHON,YGATE,QDN,IUPRDN)
YUP=YGATE
IPool=IPool+1
TYPPOOL=POOLTYPM(IPOOL)
GO TO 310
200 CONTINUE
205 C U/S SIDE OF GATE
210 C SET CHANNEL X-SECTION TYPE FOR AREA ROUTINE
TYPPOOL=POOLTYPM(IPOOL)
C D/S SIDE OF GATE
C SET SYPHON HEAD LOSS AND SYPHON DROP
SYPHON=SYPHLOD(IPOOL)
SUBROUTINE GATEMO  74/74  OPT=1  

YESYPHON-SYFPONDY(IPOOL)

230  C SET GATE AND CHANNEL ELEVATIONS

C INVERT DROP FROM GATE THRU SYPHON LUMPED IN SYFPONDY

EG=0.0
EC=0.0
IUPRDWN=0

235  CALL GATEY(QDN,YDN,EC,EG,YESYPHON,YGATE,QDN,IUPRDWN)

YDN=YGATE

C U/S SIDE OF GATE

IF(IPOL.EQ.NPOL) GO TO 310

C SET CHANGE IN INVERT ELEVATION GOING INTO GATE

240  SYFPONDY=0.0

YESYPHON=0.0

EC=0.0

EG=USIMVT(IPOOL)

IPOOL=IPOOL+1

245  TYPPOOL=POOLTYP(IPOOL)

IUPRDWN=1

CALL GATEY(QUP,YUP,EC,EG,YESYPHON,YGATE,QON,IUPRDWN)

YUP=YGATE

IPOOL=IPOOL-1

TPYPOOL=POOLTYP(IPOOL)

310  CONTINUE

C CHECK THAT U/S DEPTH IS GREATER THAN D/S DEPTH

IF(YUP.LT.YDN) GO TO 500

C

255  C CALCULATE GATE OPENING

C DEPTHS AND OS ARE INSIDE GATE SECTION

G=YUP

320  GPREV=G

IF(ABS(YUP-YDN).LT.0.01) GO TO 330

GOY=G/YUP

IF(GOY.GT.1.0) GOY=1.0

CDEFF=CD(GOY)

325  TYPPOOL=POOLTYP(IPOL)

ADN~YDN=BOTGATE(IPOL)

C GET AREA IN UPSTREAM SIDE OF GATE SECTION

260  AUP=YUP*90TGATE(IPOOL)

G=QDN/(90TGATE(IPOOL)*CDEFF+SQRT(2.*GGRAV*(YUP-YDN)

+QDN+*2/AUP**2))

270  IF(G.GT.YUP) G=YUP

G=0.5*(G+GPREV)

IF(ABS(G-GPREV).LT.0.001) GO TO 320

330  CONTINUE

WRITE(2,9040) T,G,YUP,YDN

T=T+TINC

400  REMIND 1

GO TO 10

500  CONTINUE

WRITE(5,9050) IPOOL,IPOOL,ISTRYP,T,YUP,YDN

STOP

505  C PUMP STROKING
600  CONTINUE
      WRITE(2,9040) T,QDN,YUP,YDN
      T=T+TINC
      IF(T.LE.TMAX) GO TO 40
   290  GO TO 10

C FORMATS
9000  FORMAT(1X,9HIN GATEMO/1X,
         . 30H POOL KMAX STR TYPE/)
9010  FORMAT(1X,10X,10I0)
9020  FORMAT(1X,10X,T10.3)
9030  FORMAT(1X,4I10)
9040  FORMAT(1X,4F10.3)
9050  FORMAT(1X,43H/S DEPTH LESS THAN D/S DEPTH BETWEEN POOLS,
         . 15,4H AND,15/IX,GATE TYPE,1S/IX,BHAT T=,F10.3
         . /1X,10H/S DEPTH=,F10.3/1X,10H/S DEPTH=,F10.3)
300  END
SUBROUTINE GATEY

C ROUTINE TO CALCULATE DEPTH IN GATE STRUCTURE FROM CHANNEL DEPTH

C COMMON /ALL/
C PHYSICAL PARAMETERS
   NMANN, MANN, GGRAV, PI, BOTTOM(50), SIDSLP(50), RADIUS(50), BOTGATE(51),
   MANN(20), BOTGRAD(20), SYFONDY(50), SYFNLOS(50), SPEED(51), PUMPQ
   (51), DEBBAND(51), NGATES, GATECO(11), DSINV(50), USINV(51),
C WORKING VARIABLES
   TMAX, XEND, TINC, NINC, NPOOLS, IPOOL, TYPPOOL, STABEG(50),
   STAEND(50), X1, Y1, STUDY,
C CANAL DESCRIPTION
   UPSL(50), DWNSTA(50), NPOOLS, POOLTY(50), STRTYP(51),
C REAL NMANN, MANN
   INTEGER TYPPOOL, POOLTYP, STRTYP, STUDY
C IUPRDWN=O FOR D/S SIDE OF GATE, =1 FOR U/S SIDE
IF(IUPRDWN.EQ.1) GO TO 20
C D/S SIDE OF GATE
Y=YCHANNL
ACHANNL=A(Y)
C1=QCHANNL+2/(2.*GGRAV*ACHANNL+1)+YCHANNL+ECHANNL-EGATE
   -YSYPHON*SYPHON*QCHANNL+2
C GATE BOTTOM WIDTH
B=BOTGATE(IPOOL)
C2=QCHANNL+2/(2.*GGRAV+2)
CONTINUE
C USE RECTANGULAR SECTION AT GATE
IF(Y.LE.0.) Y=0.1
F=Y+C2/Y+1-1
FPRI=1.-2.+C2/Y**3
YPREV=Y
Y=Y-F/FPRI
IF(ABS(Y-YPREV).GT.0.001) GO TO 10
YGATE=Y
RETURN
C U/S SIDE OF GATE
C IPOOL AND TYPPOOL WERE SET IN GATEMO FOR THE U/S POOL
CONTINUE
Y=YCHANNL
ACHANNL=A(Y)
C RESET TO D/S POOL FOR GATE PROPERTIES
IPOOL=IPOOL-1
IF(IPOOL.EQ.0) IPOOL=51
TYPPOOL=TYPPOOL(IPOOL)
C1=QCHANNL+2/(2.*GGRAV+ACHANNL+2)+YCHANNL+ECHANNL-EGATE
C GATE BOTTOM WIDTH
B=BOTGATE(IPOOL)
C2=QGATE**2/(2.*GGRAV+B+2)
SUBROUTINE GATEY 74/74 OPT=1

C RESET TO U/S POOL BEFORE RETURN TO GATE

IF(IPool.EQ.51) IPool=0
IPool=IPool+1
TYPPOOL=POOLTP(IPool)
CONTINUE

C USE RECTANGULAR SECTION AT GATE

IF(Y.LE.0.) Y=0.1
F=Y+C2/Y**2-C1
FPRIME=1.-2.*C2/Y**3
YPREV=Y
Y=Y-F/FPRIME
IF(ABS(Y-YPREV).GT.0.001) GO TO 30

YGATE=Y
RETURN
END
FUNCTION CD 74/74 OPT=1

1 FUNCTION CD(GOY)
   C ROUTINE TO CALCULATE GATE COEFFICIENT AS FUNCTION
   C OF GATE OPENING/YUPSTREAM
   C
   5 COMMON /ALL/
   C PHYSICAL PARAMETERS
   . NMANN, SO, GGRAV, PI, BOTTOM(SO), SIDSLOP(SO), RADIUS(SO), BOTAQTE(S1),
   . MANH(SO), BOTAQRA(SO), SYFONDY(SO), SYFNSLDS(SO), SPEED(S1), PUMPDO
   . (S1), DEDBAND(S1), NGATES, GATECO(11, S1), DSINV(SO), USINV(S1),
   10 C WORKING VARIABLES
   . TMAX, XEND, TINC, NINC, NPOLS, IPOLL, TYPPOOL, STABEG(S0),
   . NAMIN(SO), T1, T1, STOR.
   C CANAL DESCRIPTION
   . UPSTA(S0), DWINSTA(S0), NPOLS, POOLTYPE(S0), STRTYPE(S1),
   15 . TMAXI, NUNTRNS, USRES, USRES
   C
   REAL NMANN, MANN
   INTEGER TYPPOOL, POOLTYPE, STRTYPE, STUDY
   C
   C
   20 C FIT PARABOLA TO 3 POINTS NEAR GOY
   K=GOY+10.1.
   IF(K.LT.1) K=1
   IF(K.GT.9) K=9
   OFFSET=K
   25 X=GOY+10.-OFFSET
   T=IPOLL
   A=0.5*(GATECO(K, T)-2.*GATECO(K+1, T)+GATECO(K+2, T))
   B=0.5+(GATECO(K+2, T)-GATECO(K, T))
   C=GATECO(K+1, T)
   CD=A*X**2+B*X+C
   RETURN
   END
SUBROUTINE FLOWER 74/74 OPT=1

C ROUTINE TO FOLLOW GATE STROKING GATE MOTIONS TO DETERMINE
C START TIMES AND FINAL POSITIONS OF GATES
C ON TO TURN PUMPS ON AND OFF
C SPEED IS GATE SPEED IN FT/MIN
C DEADBAND IS DIFFERENCE ALLOWED IN GATE OPENING BETWEEN
C ACTUAL POSITION AND STROKING SOLUTION BEFORE MOVING GATE
C PUMPDO IS INCREMENTAL DISCHARGE FOR PUMPS
C PHYSICAL PARAMETERS
C WORKING VARIABLES
C GATE MOTIONS TO DETERMINE
GATE STROKING

DEDBAND IS DIFFERENCE ALLOWED IN GATE OPENING BETWEEN
ACTUAL POSITION AND STROKING SOLUTION BEFORE MOVING GATE
PUMPDO IS INCREMENTAL DISCHARGE FOR PUMPS

REAL NMANN,MANN
INTEGER TYPPOOL,POOLTYI,STRTYPE,STUDY

C SCRATCH AREA TO BE OVERLAID IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND

C EQUIVALENCE Q AND G SO PUMP STROKING VARIABLE NAMES MAKE
C SOME SENSE
C IF THIS IS A PUMP, GATEMO WROTE Q; IF IT IS A GATE, GATEMO
C WROTE GATE OPENING, G

DIMENSION Q(500),QLAST(500)
EQUIVALENCE (Q(I),G(I)),(QLAST(I),GPOSN(I),(PUMPQ,GATEP)
SUBROUTINE FOLLOWER 74/74  
MAX=I-1
C CALCULATE START TIMES
C INITIAL OPENING
C TIME IN MINUTES NOW
TMIN(I)=0.0
GPOSN(I)=G(I)
GATEOPN=GPOSN(I)
I=2
C FIND POOL D/S OF THIS GATE
IF(SRSTRYP(IPOOL).EQ.2) GO TO 45
IF(SRSTRYP(IPOOL).EQ.3) GO TO 45
C SKIP STROKING MOST D/S PUMP
IF(IPOOL.EQ.51) GO TO 44
IF(SRSTRYP(IPOOL).EQ.1) GO TO 80
CONTINUE
IF(IPOOL.EQ.51) IPOOL=0
IPOOL=IPOOL+1
GO TO 42
CONTINUE
DO 50 J=1.MAX
IF(AUGATE(GATEOPN-G(IJ)).LT.1E-6) GO TO 59
GATEDEL=ABS(GATEOPN-G(IJ))
OELT=OELY/SPEED(IPOOL)
TMIN(I)=T(IJ)/60.-DELT
GPOSN(I)=G(IJ)
GATEOPN=G(IJ)
x=1+1
CONTINUE
NOMOVE=I-1
WRITE(6,9020) NGATE,NOMOVE
FOR ROUTINE WRITE IT TO OUTPUT IN REVERSE ORDER
WRITE(5,9020) NGATE,NOMOVE
NGATES=NGATE
DO 60 I=1,NOMOVE
WRITE(5,9010) TMIN(I),GPOSN(I),SPEED(IPOOL)
CONTINUE
WRITE(6,9000) DEDEBAND(IPOOL)
NGATE=NGATE+1
IF(IPOOL.EQ.51) IPOOL=0
IPOOL=IPOOL+1
IF(IEOF.EQ.0) GO TO 10
REWIND 5
RETURN
10 IEOF=1
GO TO 40
C PUMP STROKING
CONTINUE
C SET INITIAL DISCHARGE TO N*PUMPQ
N=0
CONTINUE
FN=N
QTEST=PUMPQ-FN*PUMPQ(IPOOL)
IF(QTEST.LT.0.0) GO TO 90
N=N+1
GO TO 85
90 N=N-1
SUBROUTINE FOLOWER 74/74 OPT=1

115
FNPUMP=PUMPQ+PUMPDQ(IPPOOL)
QLAST(I)=PUMPQ
DO 110 J=2,MAX
IF(ABS(PUMPQ-Q(J))GE.PUMPDQ(IPPOOL)) GO TO 110
120 C T/CURN PUMP ON OR OFF
IF(PUMPQ.LT.Q(J)) GO TO 100
C T/CURN PUMP OFF
TMN(I)=T(J)/60.
95 CONTINUE
QLAST(I)=PUMPQ-PUMPDQ(IPPOOL)
PUMPQ=QLAST(I)
IF(ABS(PUMPQ-Q(J))GE.PUMPDQ(IPPOOL)) GO TO 95
I=I+1
GO TO 110
130 C T/CURN PUMP ON
100 CONTINUE
TMN(I)=T(J)/60.
105 CONTINUE
QLAST(I)=PUMPQ+PUMPDQ(IPPOOL)
PUMPQ=QLAST(I)
IF(ABS(PUMPQ-Q(J))GE.PUMPDQ(IPPOOL)) GO TO 105
I=I+1
110 CONTINUE
NOMOVE=I-1
WRITE(6,9050) NGATE,NOMOVE
WRITE(5,9050) NGATE,NOMOVE
NGATES=NGATE
DO 120 I=1,NOMOVE
WRITE(5,9040) TMN(I),QLAST(I)
120 CONTINUE
WRITE(6,9050) PUMPDQ(IPPOOL)
NGATE=NGATE+1
IPPOOL=IPPOOL+1
IF(I.EQ.0) GO TO 10
150 CONTINUE
REWIND 5
RETURN
C FORMATS
9000 FORMAT(1X,24HGATE MOVEMENT DEADBAND =,F10.3,3H FT)
9010 FORMAT(1X,3F10.3)
9020 FORMAT(1X,4HGATE,15,110,2X,9MOVEMENTS)
9030 FORMAT(1X,4HPUMP,15,110,2X,9CHANGES )
9040 FORMAT(1X,2F10.9,10X)
9050 FORMAT(1X,4HPUMP DELTA Q =,F10.3)
END
SUBROUTINE WRITE11

C ROUTINE TO MERGE TURNOUT SCHEDULES, GATE SCHEDULES
AND PUMP SCHEDULES INTO INPUT FILE FOR UNSTEADY MODEL
C FILE ON TAPE 6 IS POSITIONED AT END OF INITIAL CONDITIONS
C OUTPUT IS FROM UPSTREAM TO DOWNSTREAM

COMMON /ALL/
C PHYSICAL PARAMETERS
. NMANN, SO, GGRAV, PI, BOTTOM(SO), SIDELOP(SO), RADIUS(SO), BOTGATE(SI),
10 MANN(SO), BOTRAD(SO), SYFONDY(SO), SYFNLOS(SO), SPEED(SI), PUMPDOQ
(51), DEBAND(51), NGATES, GATECO(SI), DSNTV(SO), USINV(SI),
C WORKING VARIABLES
. TMAX, XEND, TINC, NXINC, NPPOOLS, IPOOL, TYPPOOL, STABEG(SO),
STAOEND(SO), TI, TV, STUDY,
15 C CANAL DESCRIPTION
. USTAD(SO), DWNSTA(SO), NPPOOLS, POOLTY(SO), STRTYPE(SI),
. TMAX, NUMTRNS, USRES, DSRES
C REAL NMANN, MANN
C INTEGER TYPPOOL, POOLTY, STRTYPE, STUDY
C C SCRATCH AREA TO BE OVERLaid IN THE VARIOUS SUBROUTINES
C TO KEEP MEMORY WITHIN SOME SORT OF REASONABLE BOUND
25 COMMON /SCRATCH/
. DUMMY1(SO), OUX(SO), OUX(SO), OUX(SO), DUMMY2(SO),
. TMQV(SO), GO(SO), SPEDE(SO), DUMMY3(2287)
C DIMENSION Q(SO)
30 C EQUIVALENCE GATE OPENING, GO, FOR GATES WITH Q FOR
C PUMPS SO VARIABLE NAMES WILL MAKE SOME SENSE.
C GO AND Q ARE IN THE SAME FIELD ON TAPE 5.
C C EQUIVALENCE (GO(1), Q(1))
35 C C START AT UPSTREAM END
NGATE = NGATES
JPOOL = 1
K = NPPOOLS
40 C WAS U/S STRUCTURE A GATE OR A PUMP
IF (STRTYPE(SI) .EQ. 1) GO TO 80
C GATE
C FIND GATE SCHEDULE ON TAPE 5
REوارD 5
45 10 READ(S, 9110) IGATE, IMOVE
DO 20 I=1, IMOVE
READ(S, 9120) TMOVE(I), GO(I), SPEDE(I)
20 CONTINUE
IF (IGATE .NE. NGATE) GO TO 10
50 C SET FOR NEXT D/S STRUCTURE
NGATE = NGATE + 1
WRITE(S, 9130) JPOOL, STRTYPE(SI), IMOVE
DO 30 I=1, IMOVE
WRITE(S, 9140) TMOVE(I), GO(I), SPEDE(I)
30 CONTINUE
GO TO 120
C PUMP
50 \text{CONTINUE}

C WAS PUMP SCHEDULED OR STROKED

IF \texttt{\text{STUDY.EQ.2}) \text{GO TO 60}

C SCHEDULED

C EXTRACT Q CHANGES FROM QU

\texttt{IMOVE=1}

\texttt{TMOVE(1)=0.0}

\texttt{Q(1)=Q(1)}

\texttt{DO 60 J=2,48}

\texttt{IF(Q(J).EQ.Q(J-1)) \text{GO TO 60}}

\texttt{IMOVE=IMOVE+1}

\texttt{TMOVE(\texttt{IMOVE})=(J-1)+60}

60 \text{CONTINUE}

\texttt{WRITE(9,9130) \texttt{JPOOL,STRTYPE(K),TMOVE}}

\texttt{DO 70 I=1,IMOVE}

\texttt{WRITE(9,9150) TMOVE(I),Q(I)}

70 \text{CONTINUE}

\texttt{GO TO 120}

C STROKED

80 \text{CONTINUE}

C FIND PUMP SCHEDULE ON TAPE 5

REWIND 5

90 READ(5,9110) IGATE,IMOVE

DO 100 I=1,IMOVE

READ(5,9120) TMOVE(I),O(I)

100 \text{CONTINUE}

IF(IGATE.NE.NGATE) \text{GO TO 90}

C SET FOR NEXT D/S STRUCTURE

\texttt{NGATE=NGATE-1}

\texttt{WRITE(9,9130) \texttt{JPOOL,STRTYPE(K),IMOVE}}

\texttt{DO 110 I=1,IMOVE}

\texttt{WRITE(9,9150) TMOVE(I),O(I)}

110 \text{CONTINUE}

C IS THERE A TURNOUT AT D/S END OF POOL

120 \text{CONTINUE}

\texttt{DO 130 I=1,48}

\texttt{IF(O(I,K).NE.0.0) \text{GO TO 140}}

130 \text{CONTINUE}

\texttt{GO TO 200}

C EXTRACT TURNOUT Q CHANGES FROM QT

140 \text{CONTINUE}

\texttt{IMOVE=1}

\texttt{TMOVE(1)=0.0}

\texttt{Q(1)=Q(1)}

\texttt{DO 150 J=2,48}

\texttt{IF(Q(J,K).EQ.QT(J-1,K)) \text{GO TO 150}}

\texttt{IMOVE=IMOVE+1}

\texttt{TMOVE(\texttt{IMOVE})=(J-1)+60}

\texttt{Q(\texttt{IMOVE})=Q(J,K)}

150 \text{CONTINUE}

\texttt{ITRNYPE=99}

\texttt{WRITE(9,9130) \texttt{JPOOL,ITRNYPE,IMOVE}}

\texttt{DO 160 I=1,IMOVE}

\texttt{WRITE(9,9150) TMOVE(I),Q(I)}

160 \text{CONTINUE}

C
SUBROUTINE WRITEIT 74/74 OPT=1

115 C INTERIOR BOUNDARIES

120 CONTINUE
125 C IS THERE A STRUCTURE AT U/S END OF THIS POOL
130 IF(STRTYPE(K).EQ.4) GO TO 300
135 IF(STRTYPE(K).EQ.5) GO TO 300
140 IF(STRTYPE(K).NE.1) GO TO 240

145 C PUMPS
150 CONTINUE
160 C EXTRACT CHANGES IN TURNOUT Q FROM QT
165 CONTINUE
170 CONTINUE

175 C PUMP SCHEDULE ON TAPE 5
180 READ(5,9110) IGATE,IMOVE
190 DO 220 I=1,IMOVE
200 READ(5,9120) TMOVE(I),Q(I)
210 CONTINUE

220 CONTINUE
230 IF(IGATE.NE.NGATE) GO TO 210

235 C SET FOR NEXT D/S STRUCTURE
240 NGATE=NGATE-1
250 WRITE(9,9130) JPOOL,STRTYPE(K),IMOVE
260 DO 270 I=1,IMOVE
270 WRITE(9,9140) TMOVE(I),Q(I),Q(1)
280 CONTINUE
290 GO TO 300

300 CONTINUE
310 IF(TOT(J,K).NE.0.0) GO TO 320
320 CONTINUE

325 C EXTRACT CHANGES IN TURNOUT Q FROM QT
330 CONTINUE

335 ITRTYPE=99
SUBROUTINE WRITEIT

WRITE(9,9130) JPOOL,ITRTYP,IMOVE
DO 340 I=1,IMOVE
    WRITE(9,9150) TMOVE(I),G(I)
CONTINUE
GO TO 200

C
C MOST D/S STRUCTURE
C
900 CONTINUE
C Q WAS SCHEDULED IN QD
C IS THIS A PUMP
    IF(STRTYPE(51).NE.1) GO TO 950
C PUMP
185 C EXTRACT CHANGES IN Q FROM QD
    IMOVE=1
    TMOVE(1)=0.0
    Q(1)=QD(1)
    DO 910 J=2,48
        IF(QD(J).EQ.QD(J-1)) GO TO 910
        TMOVE(IMOVE)=(J-1)*60
        Q(IMOVE)=QD(J)
        IMOVE=IMOVE+1
    CONTINUE
910 WRITE(9,9130) JPOOL,STRTYPE(51),IMOVE
    DO 920 I=1,IMOVE
        WRITE(9,9150) TMOVE(I),G(I)
    CONTINUE
920 CONTINUE
GO TO 1000

C GATE
950 CONTINUE
    READ(5,9110) IGATE,IMOVE
    WRITE(9,9130) JPOOL,STRTYPE(51),IMOVE
    DO 990 I=1,IMOVE
        READ(5,9120) TMOVE(I),GO(I),SPED(I)
    CONTINUE
990 CONTINUE
1000 CONTINUE
REWIND 5
210 REWIND 9
RETURN
9110 FORMAT(5X,15,I10)
9120 FORMAT(1X,3F10.3)
9130 FORMAT(3F10.3)
9140 FORMAT(3F10.3)
9150 FORMAT(2F10.3,10X)
END