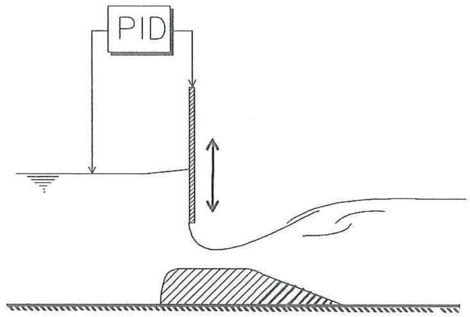
Appendix G MODIS User's guide

A model to study the hydraulic performance of controlled irrigation canals



Wytze Schuurmans

MODIS
User's Guide

Appendix G of
A model to study
the hydraulic performance
of controlled irrigation canals

Wytze Schuurmans

Preface

The MODIS model is a computer program package which can be used to study the hydraulic performance of controlled canals. MODIS is an acronym of MOdelling Drainage and Irrigation systems, and has been developed out of the existing program RUBICON developed for river applications.

The user's guide consist of three parts. The first four chapters provide general information about the program package and its use. The second part (chapter 5 to 10) deals with the input files of the program package. In an example all input files for a particular case are worked out. The third and last part of the user's guide (chapter 11 and 12) contains some detailed information about the mathematics underlying the program and the applied numerical methods of solution.

In addition to the user's guide, a software documentation of the program package is available for those who are going to make model modifications.

Wytze Schuurmans, August 1991

Table of contents

4 Input data files

5 Model definition

10 Example

11 Mathematical background

12 Numerical solution

1 Description of program modis

reprint of the paper "Description and evaluation of program MODIS" (schuurmans 1991)

Abstract MODIS is an implicit hydrodynamic modeling package that has been developed to investigate the hydraulic performance of dynamic controlled irrigation systems. The model's most apparent features are its accurate computation of a wide range of standard structures and its many operation possibilities. Furthermore, the model is able to compute performance indicators which allow for a fast and diagnostic interpretation of the model results. The user interface is not menu driven and the program is not public domain. The program is most suitable for experienced users and for large systems.

1.1 Introduction

The name MODIS is an acronym of "Modeling Drainage and Irrigation Systems", and has been developed at Delft University of Technology in 1990. The computational base program underlying MODIS is the river modeling package named Rubicon. The main reason for its development was the fact that no models tailored for controlled irrigation canals were to be found. The main limitations of existing programs are the lack of an accurate computation of standard irrigation structures and the limited operation possibilities of these structures. The "Task Group on Real Time Control of Urban Drainage Systems" also comes up with the conclusions that "*Models for the state of the system ... have been developed in a great number for static, non-controllable systems. However, hardly any model has been described allowing to simulate automatic regulators and external control input during the simulated process*". (Schilling 1987).

The same conclusion has been drawn two years later: *".. a key to efficient research on canal automation is the existence of an easy-to-use, accurate, and flexible unsteady flow canal hydraulic simulation program. This research project did not find such a program"*. (Burt, 1989)

In this paper a description of the program is presented following the "Canal model evaluation and comparison criteria" (Rogers et al 1991). Moreover, an application is presented to illustrate its use. The application deals with the modernization of a 110 km long irrigation canal in Jordan.

1.2 Technical Merit

Computational accuracy
The model solves the complete De Saint Venant equations. The applied numerical solution technique is based on finite differences by using the Preissmann implicit scheme. This implies that the numerical method is of second order accuracy in place and, usually, of first order in time (depending on the value of the time interpolation coefficient). The values of the non-linear terms are determined by interpolating between the values at the old and new time level, starting with the values at the old time level. The number of iterations can be specified by the user, but a value of two is recommended.

Numerical solution criteria
The numerical method is mass and momentum conservative if at least two iterations are used. By using an implicit scheme (the four point implicit Preissmann scheme), stability is guaranteed for any Courant number. The numerical solution converges to the real solution if the solution is stable, because the difference equations have proven to be consistent with the differential equations (Cunge et. al 1980).

Robustness

Robustness has been given priority to accuracy. If accuracy considerations are violated a warning message appears. For example, if input errors are encountered the model will continue reading the input data, and afterwards print the encountered errors in an echo file of the input file. In this way input errors are well traced and can be quickly corrected. To avoid program termination in case of dry bed flow, a Preissmann slot is automatically added to trapezoidal cross-sections. A special routine prevents the slot from falling dry by continuously checking if the water levels are lower than the bed levels. If so, the water depths at those locations are artificially increased to 0.01 m above the bottom level and a base flow of 0.001 m^3/s is generated. (A warning message is printed whenever this routine is activated).

Initial conditions

To solve the De Saint Venant equations initial and boundary conditions are needed. The initial condition requires water levels and discharges at every computational point at the beginning of the computation. The user has only to specify initial conditions at the branch ends, as the program interpolates the values for intermediate grid points located in that branch. It is also possible to use the outcome of a previous computation as an initial state for further computations. This feature can be used, for example, to use a pre-calculated steady state as an initial state.

Internal and external boundary condition analysis

The canal layout is modeled by using branches and nodes. Branches represent conveyance elements such as pools or reaches. Nodes are only used to link branches together and to indicate a branch end. In addition nodes can be used to model reservoirs whereby the storage area can be specified as a function of the water level.

External boundary conditions are imposed on nodes indicating branch ends. The user can choose between a water level, discharge, or stage-discharge relationship as boundary condition. The water level and discharge can be specified either as constants or as functions of time (stage hydrographs and discharge hydrographs).

Internal boundary conditions are needed to link branches, and there water level compatibility is assumed. Structures can be specified anywhere along a branch without having to specify boundary conditions. The boundary conditions are rewritten internally in the same linearized format as the mass and momentum equations, and thus fully incorporated in the implicit solution procedure. The boundary conditions can be specified as fixed values, as time series, or as a function of a user written fortran routine.

Special hydraulic conditions

The model has a special routine to avoid dry bed flow in order to keep the model running. However, no special equations are incorporated to calculate advances on a dry bed. Rapid flow changes and bore waves too are in principal not covered by the De Saint Venant equations, which are valid for gradually varied unsteady flow only. However, the error made is small and the model can handle rapid changes quite well as long as the Courant number is chosen sufficiently close to one and the time interpolation coefficient is somewhat greater than ½ (Contractor & Schuurmans 1991).

Supercritical flow cannot be handled. This is not because the De Saint Venant equations are not valid for supercritical flow, but it is due to the (double sweep) matrix solution algorithm. Hydraulic jumps can only be handled in the vicinity of structures, where the De Saint Venant equations have been replaced by structure equations. Reversal of flow directions will cause no problems. It is even possible to use different structure parameters for flow in a positive and in negative direction.

1.3 Modeling capabilities

System configuration

The system configuration is modeled by using branches and nodes. No restrictions are imposed on the branch lengths, and both branched and looped canal networks can be handled. The model generates a computational grid along every branch using a Δx increment, specified by the user. The user can add additional computational grid points to the canal system.

These user defined grid points are needed to specify the branch characteristics such as its profile and elevation. Every grid point has an elevation and a cross-section. The cross-section consists of a profile shape, Boussinesq coefficient(s) and resistance coefficient(s). All construction elements such as branches, nodes, grid points and cross-sections have user defined names instead of numbers.

Structures have to be located within a branch. As more than one structure can be placed within a branch, a branch can consist of various pools separated by structures. Moreover, it is possible to model composite structures by locating several structures at the same location.

Frictional resistance

The friction term of the De Saint Venant Equations is represented by Strickler/ Manning resistance formula. The resistance value can be varied in height and with the longitudinal distance.

Boundary Condition types

Structures are not treated as boundary conditions in the MODIS model, as they are placed within a branch. When a structure is encountered, the momentum equation of the De Saint Venant Equations is replaced by the structure equation which is rewritten in the same format as the momentum equation and thus fully incorporated in the implicit solution procedure.

The standard structure library incorporated in the MODIS model comprises: pumps, weirs, orifices, pipes, head loss structures, and Neyrtec baffle distributors. Furthermore, the user can add his own written Fortran defined structures, but this requires some knowledge of Fortran and the program. Structures can be placed in series and in parallel. The latter facility is used to define compound structures.

Special attention has been paid to an accurate computation of structure flow. As the upstream and downstream water level can fluctuate during a computational run, the flow condition can also fluctuate, e.g. from free to submerged flow, or from orifice flow to weir flow. The model continuously checks which flow condition is to be applied, taking the actual state of the system into account. Some default values for shifting from one flow condition to another are incorporated, but the user can also define these boundaries himself. Moreover, the value of various structure coefficients (e.g. the contraction coefficient or effective discharge coefficient) depends on the hydraulic conditions. The values of these coefficients can be specified by the user either as constants or as a function of hydraulic parameters.

Turnouts

Turnouts are treated in the same way as the structures described in the previous section. The user has to define a branch first, and then to locate a structure inside that branch. At the branch end a fixed water level could be specified as a boundary condition. Only if the outflow rate is predefined, lateral inflow/outflow facilities can be used which do not require an additional branch.

Operations duplication

The MODIS model is able to simulate all types of structure operation. Various structure parameters such as width, sill level, and gate opening height can be given a constant value or can be specified as functions. There can be time functions, but also functions of e.g. an upstream water level. Pumps are switched on if the actual water level exceeds a user defined level, and switched off if the actual water level drops below another lower user defined level. (These levels in turn can be specified as a function of time). Moreover, the capacity of the pump can be specified as a function of the head.

Automatic control

It is obvious that it is irrelevant for the behaviour of the system whether operation is carried out manually or automatically. Automatic control as a function of time and on/off control can therefore be modelled in the same way as described in 3.5. In MODIS it is also possible to simulate real time control (closed loop control) whereby control is based on the actual state of the system following a control algorithm.

Several types of control algorithms have been implemented in the MODIS model: a multiple speed controller, a Proportional Integral Differential (PID) controller, BIVAL-control, CARDD-control and

Linearized Quadratic Gaussian Control. For each controller control parameters have to be specified. These control parameters, such as gain factors, can be specified as constants, but also as a function of time or a hydraulic variable. Thus, the target levels of the controllers can be adjusted in time and the speed of gate movement can be specified as a function of the deviation from the set-point. Finally, different levels of control can be defined such as local and regional control.

As a result, all types of existing canal control systems, such as upstream control, downstream control, mixed control, BIVAL control, ELFLOW-control and CARDD-control can be modelled in the MODIS model. User defined control algorithms can be added to the model by using the Fortran function facility.

Miscellaneous limitations

Model limitations are mainly related to memory limitations. The user can tailor its model for a specific project by changing the maximum number of branches, nodes, structures, functions, controllers, et cetera. In order to do so, the user has to redefine some maximum parameters and to re-compile the complete model package. There are no limitations to physical dimensions. Both trapezoidal and irregular cross-sections can be modelled. The minimum computational time step is 1 second. The user can define the required format of the output data and thus presents the water levels as precise as he wants.

1.4 User considerations

User interface

The MODIS model is controlled from an operation template which is shown on the screen. By moving a highlighted bar through the template to a specific item and by pressing a help key, information about the selected item is provided. The program consists of several sub programs which have to be run in sequence. Every subprogram has its own input file and produces both an echo file with error messages, and an output file. Each input data file is first checked on possible errors before execution of the program. The error messages, divided into warnings, errors, and fatal errors, are printed in an echo file of the input file.

The input data (in metric units only) can be found in input files. Interactive data input is not possible. An input file comprises input tables supported by explanatory comment lines. The input data has to be specified in the input tables following a pre-described sequence, but no fixed column position is needed. Both names and numbers can be used to denote structures, branches, nodes, functions, controllers et cetera.

To facilitate easy data input, special information characters can be used to reduce the amount of input data. For example, the "=" symbol means that the value is equal to the value of the same column specified above. The "?" symbol stands for interpolation of data specified in the same column above and below. Furthermore, all editor features such as "find and replace", "copy", and "move" are available. Practice has shown that the use of input files instead of a menu driven input might be overwhelming for the first time user, whereas the more experienced user finds its way easily and quickly.

The final computational results, as functions of time or place, can be presented both graphically and in tables. Possible output parameters are: water levels, water depths, discharges, wetted cross-sectional area, flow width, storage width, Boussinesq coefficient, hydraulic radius, resistance, Froude number. The coloured graphs can be printed on various types of screens and on a wide variety of printers and plotters.

To facilitate an easy interpretation, it is possible to print only maximum, mean, and minimum values at certain locations during a specific periods of time. Furthermore, the percentage of time in which user defined minimum and maximum values have been exceeded can be printed.

To evaluate the water distribution in irrigation systems simulated by the MODIS model, operation performance indicators which are computed by the model can be used. They consist of a delivery performance ratio (DPR), which specifies to which extent the user intended distribution is satisfied, and an operation efficiency (e_0), which indicates how much water is lost due to inappropriate operation and leakage (Schuurmans & Maherani, 1991).

Documentation and support

The program package is supported by a user's manual, in which also examples are presented. Furthermore, an updated software documentation is available containing an alphabetic list of all parameters, all subroutines and Hipo diagrams of each subprogram. Purchasers of the program can get assistance both from Delft University of Technology and from Haskoning Consulting Engineers who developed the base river modeling package "Rubicon" out of which MODIS was developed.

Direct costs

The executable version of the program package costs about *ƒ* 25,000 and to obtain the source code an additional amount is needed, depending on the applications. The program can run on any IBM-compatible computer with 640 KB Ram memory, equipped with a mathematical co-processor and with a hard disk. The program performs most well on an AT-computer with a 80286 or higher processor. For the graphical output, the commercial "HALO" graphical package is needed.

Indirect costs

It takes one day to get familiar with the program and to know where to find what in the user's guide. To define a model of the irrigation system using the MODIS package requires another few days, depending on the size of the system. Apart from calibration (if needed) most of the time is involved in determining which runs have to be made and how to interpret the results. This requires a skilled watermanagement engineer for operation rather than a computer specialist. To find a sound solution several simulations are usually required. Interpretation of the model results is fastened with the help of the model interpretation parameters which also do have some diagnostic importance. The model execution time depends on the size of the model. Even for large irrigation systems with hundreds of grid points and more than a hundred structures it will take less than a second to proceed one computational step on a 386 machine. This implies that the total simulation time required to simulate a few days is a matter of minutes.

1.5 Summary and conclusions

A great number of flow models which calculate the unsteady flow phenomena in one dimensional canal systems do exist. Each program has its own characteristics and limitations. The MODIS model was developed especially to study the hydraulic performance of controlled irrigation canals. In that respect special attention has been paid to an accurate computation of structure flow and a wide range of operation concepts. Although the program is not menu driven, it has proven to be convenient for more experienced users and large canal systems. MODIS is a commercial program package and not public domain.

References

Burt, Charles (1989), "Canal Automation Providing On-Demand Water Deliveries for Efficient Irrigation", Department of Agricultural Engineering, California Polytechnic State University (Cal Poly), San Luis Obispo, CA 93407, U.S.A.

Cunge, J.A., F.M. Holly, A. Verwey (1980), "Practical Aspects of Computational River Hydraulics", Pitman Publishing Limited, London.

Contractor D.N., W. Schuurmans (1991), "Informed use and potential of canal models", ASCE National Conference on Irrigation and Drainage Engineering, Hawaii.

Haskoning, Jouzy & Partners (1991), "Rehabilitation and upgrading of the King Abdullah Canal, Hydraulic model for calculation and analysis of the water in unsteady state conditions", Nijmegen, The Netherlands.

Rogers D.C., J. Keith and W. Schuurmans (1991), "Canal model evaluation and comparison criteria", ASCE National Conference on Irrigation and Drainage Engineering, Hawaii.

Schilling W. (1987), "Real Time Control of Urban Drainage Systems, The state-of-the art", IAWPRC/IAHR Joint Committee.

Schuurmans W., M. Maherani (1991), "Operational performance of canal control systems", Water Resources Management, Kluwer Academic Press, Dordrecht, the Netherlands.

2 Getting started

2.1 Structure of the modis program package

The MODIS program package consists of five executable submodels which can be processed individually. The input data of each submodule are read from an user specified input file and possibly from internal output files, which have been produced by a previous submodule.
When a submodule is executed it produces two output files which can be looked up by the user. The first is a so-called "echo file" which contains a copy (echo) of the input file, to which possible error messages are added. The second output file is a normal output file and contains the processed data. The output files of some submodules are normally of no interest to the user. The most important output files, are the ones containing the results of the computations (Output tables and plots).

The main structure of MODIS, containing the five executable submodules is shown in Figure 2.1.1.

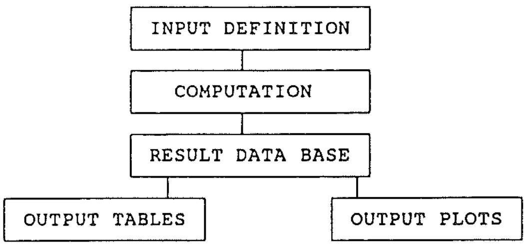


Figure 2.1.1 Main structure of program package MODIS

2.2 Installation of the program package

The MODIS program package consists of five executable submodules and one menu submodule. Each submodule should be located in a specific subdirectory under the main directory "MODIS" of which the names are given in Figure 2.2.1.

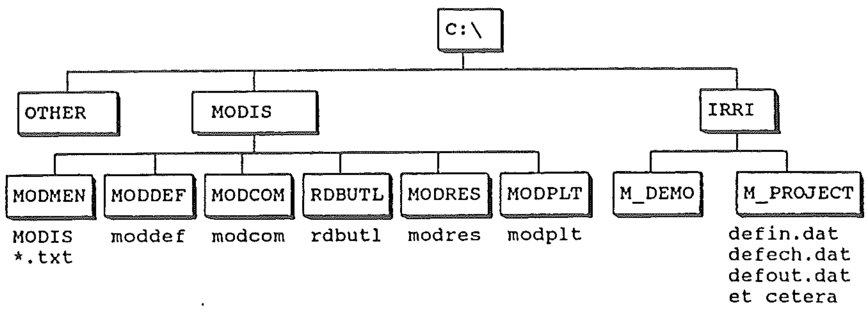


Figure 2.2.1 Location of submodules in subdirectories

For more detailed information about the structure and location of the system files, reference is made to chapter 13. With the help of the installation program located on diskette no. 1, the required directories are made automatically and the submodules are automatically copied to the right directories. In order to run the installation program, put diskette no. 1 in drive A and type the following MS-DOS command line (followed by <enter>):

```
A:\install
```

The install program will ask for the next floppies.

2.3 Starting the program

Making a default directory

The user of MODIS is advised to work in new directories for every new project, the so called "working directory", as to avoid overwriting of existing data. To make this "working directory", the user has to type the following command lines, each line terminated by pressing the <enter> key:

```
cd \
md IRRI
cd IRRI
```

Now the working directory \IRRI has been created and is consequently the default directory from which the program MODIS can be started. This implies that all input data files and all result files are written and read from this directory.

Starting the program

The program MODIS is now started by typing the following MS-DOS command line (terminated by <enter>):

C:\MODIS\MODMEN\MODIS.EXE

Tip 1 When a new working directory is made it is advised to copy already existing input files from an old working directory to this new working directory. In this way all standard input data of the input files do not have to be retyped and only data describing the water management system should be altered. In order to copy the demonstration input data files to the working directory, the following MS-DOS command line should be given before starting the program: COPY \IRRI\M_DEMO*.DAT

Tip 2 To facilitate the start of MODIS from every directory without having to type the whole path, the following statement should be included in the autoexec.bat file: PATH \MODIS\MODMEN

2.4 How to use the operation template ?

When the computer program is started, an introduction screen appears showing the name MODIS and the version of the program. When any key is hit the program skips to the operation template as shown in Fig. 2.4.1..

Using the arrow key, the user can move the highlighted bar through the template. To select a highlighted item, simply press the <enter> key. In addition to the selection of an item, various other functions can be performed from the operation template. An overview of the special keys and their function is presented in Table 2.4.1.. When pressing the <esc> key for example, a general help screen will be shown, which explains the function of special keys. To get help information about a highlighted item on the operation template (for example about the input of the model definition) simply press the <F1> function key.

To terminate the program package MODIS, move the highlighted bar on "END OF MODIS" (by using the arrow keys or by pressing the <end> key) followed by <enter>. To go temporarily to the MS-DOS operating system, press the <F2> function key. You can return to the operation template by typing EXIT, followed by <return>.

MODIS	OPERATION TEMPLATE	DATE : 03-08-89

SUBPROGRAM	INPUT	RUN	ECHO	OUTPUT
DEFINITION	I	R	E	O
COMPUTATION	I	R	E	O
RESULT DBASE	I	R	E	O
OUTPUT TABLES	I	R	E	O
OUTPUT PLOT	I	R	E	O
COMPLETE MODEL RUN				
END OF MODIS				

press <esc> for help

Figure 2.4.1 Operation template of MODIS

TABLE 2.4.1 Special keys of the operation template

KEY	FUNCTION
<enter>	Select the highlighted item.
<ESC>	General help information.
<F1>	Help information about the highlighted item.
<F2>	Go to DOS.
<F3>	Back-up of all input data files
<F4>	Copy a back-up input files to the working directory
<END>	Moves highlighted bar to "END OF MODIS".
<CURSOR UP>	Moves highlighted bar up.
<CURSOR DOWN>	Moves highlighted bar down.
<CURSOR LEFT>	Moves highlighted bar to the left.
<CURSOR RIGHT>	Moves highlighted bar to the right.

2.5 Running the model

As explained in paragraph 2.1 "Structure of the MODIS program package", the package consists of five executable submodules which have to be executed following a certain sequence.

Model definition

First the input file of the model definition has to be prepared for execution. In chapter five a detailed description of this input file is given. When the input file has been completed, the submodule MODEL DEFINITION is executed. This is achieved by steering the highlighted bar to row DEFINITION and column RUN and pressing <enter>. After execution a message will appear on the screen with the number of errors made. When errors have been found, the echo file of the model definition is checked in search of the detected errors. Of course the marked errors are not to be corrected in the echo file, but in the input file. When the detected errors have been corrected, the model definition is processed once more. This procedure must be repeated until there are no more errors to be found. When desired, the output file of the model definition can be looked up or printed to generate a good presentation of the input data.

Other submodules

This procedure of preparing the input file, running a submodule and correcting possible errors until no more errors are being found, should be followed for all the other submodules as well. The sequence in which the submodules should be executed has been listed beneath :

- Model Definition
- Model Computation
- Result Dbase
- Output Tables and/or Output Plots

3 Modelling the watermanagement system

3.1 Introduction

In this chapter the principles of how to model a watermanagement system in the MODIS computer package and of how to make simulations and evaluations, are explained. The MODIS model has a wide range of elements with which the user can easily model nearly any watermanagement system. The construction elements comprise: branches, nodes, grid points, cross-sections, regulators etcetera. The lay-out of the canal system is defined by using branches connected by nodes. It is possible to model both branched and looped networks. (It is not necessary to limit the branch length from a computational point of view, as the model will generate a computational grid over the branches afterwards). The canal cross-sections are attached to the branches by using (computational) grid points which are placed on the branches. Concerning the regulators, the user can describe nearly any structure as a wide range of standard structures is available in MODIS.

The operation elements comprise the various alternatives for operation of the regulators, varying from manual operation to automatic computerized operation. Furthermore the operation performance criteria which allow for a fast, diagnostic and comparative evaluation of operation alternatives are considered as "operation elements".

3.2 The canal system

Configuration

The canal configuration is schematized using branches and nodes. A branch is a conveyance element and has a certain length. A node is an element used to connect branches or to indicate a branch end. Both branched and looped canal networks can be handled by MODIS. The length of a branch is not limited and it is not required to use branches of the same length.

Boundary conditions

At every node an external or internal boundary condition has to be specified. For a node that connects branches, the internal boundary condition is usually the compatibility of the water levels in the connected branches. For a node at a branch end, the external boundary condition can be a specified water level, discharge or any relationship between the water level and discharge (for instance an end spillway).

Normally no water is stored at a node itself and all storage is located in the branches. In some special cases however, it is convenient to allow for storage at a node, e.g. to model a night reservoir. In that case the node has a storage area equal to A_S. The storage area A_S is assumed to equal zero by default.

Each branch and node must be given an unique name. Figure 3.2.1 shows an example of the use of branches and nodes in a modelled canal system.

Dimensions

The dimensions of the canal profile and the bed slope of the canals are given by "grid points". A grid point is a computational point located at a branch. Each grid point has a name, a reference level and a cross-section (including Manning's or Strickler roughness coefficient and the Boussinesq coefficient). Every branch must be bounded by grid points, one at each end. (In the nodes no water level or discharge is computed and therefore computational grid points are necessary). By defining these grid points in a branch, the branch will obtain a bed slope (implied by the reference levels of the grid points), a cross-sectional profile and a roughness.

Grid points

Grid points are required for numerical integration of the canal flow equations and the flow through the regulators. Grid points are used to discretize the branch. This is required for numerical integration of the partial differential equations describing the gradually varied unsteady flow. In MODIS a computational grid with several intermediate grid-points is automatically generated between the user defined grid points in a branch. The user just has to specify the maximum distance between the automatically generated grid points. In this way the user can easily vary the step size Δx and control the numerical errors. The cross-sectional data attached to the automatically generated grid points are obtained by interpolation from the adjacent user defined grid points.

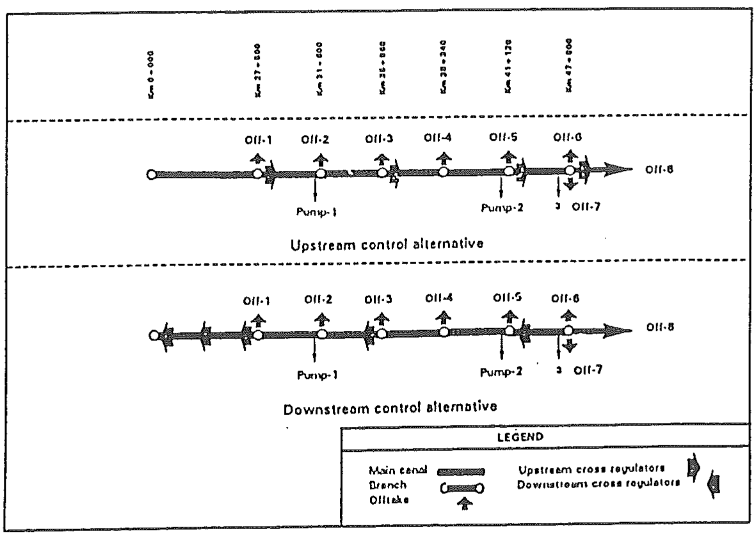


Figure 3.2.1 Example of a modelled canal system

Cross-sections

As mentioned before the cross-sectional data of a branch are specified via grid points. Each grid point refers to a cross-sectional name. This name is attached to a cross-sectional table. In each cross-sectional table, the canal profile (any shape is possible), the Boussinesq coefficient and the Manning or Strickler roughness coefficient are being specified.

3.3 Regulators

Regulators can be defined at any place along a branch. The MODIS model package offers a wide range of standard structures for both free and submerged flow conditions. The model automatically switches to the right flow condition and in case of orifice flow the model will also switch to free or submerged weir flow when the upstream water level drops too low.

List of structures

The structures included comprise:

- pumps
- overflow structures
- orifices
- pipes
- local head loss structures
- Baffle distributors as produced by Alsthom Fluides (Neyrtec)

Structures can be placed in series as well as parallel. The latter allows the modelling of composite structures such as the combination of an overflow weir and an orifice.

Flow direction

The structure parameters of all structures can be given for both positive and negative flow directions with respect to the canal axis. When structure parameters are given for just one flow direction, flow in the reverse direction is considered zero by default.

Structure parameters

Structure parameters such as discharge coefficients can be varied as function of numerous variables. For example, the effective discharge coefficient (c_e) of an overflow weir can be given as a function of the upstream water head. It so follows that for higher upstream heads the weir will switch from a broad crested to a sharp crested weir and vice versa.

Flow condition

The transition from free to submerged flow is treated in a different way with weir and orifice structures. With a weir a drowned flow reduction coefficient is used in combination with the normal free overflow stage discharge relation. The drowned flow reduction factor value can be specified by the user as a function of the ratio between the upstream and downstream water level. With orifice structures, the transition between different flow conditions is more complicated. Concerning free flow an other equation is used than for submerged flow and the orifice flow might turn into weir flow when the upstream head is sufficiently low. The model determines the applicable flow condition, taking the upstream head, the downstream head and the gate opening height into account. (For a more detailed description reference is made to paragraph 5.5, 11.2 and 12.4 of this user's guide).

Fortran structures

When a structure cannot be described by one of the standard structures, the user can define this structure as a Fortran structure. In that case the user has to write a (short) Fortran program in which the discharge is described as a function of the upstream and downstream water level. This routine should be linked to the package. Due to the applied flexibility in defining the standard structures parameters, the user is advised to seriously consider the necessity of a fortran structure.

3.4 Control of regulators

In the MODIS computer package special attention was paid to the control of regulators. As a result all types of operation alternatives can be simulated with the model. Both open and closed loop control can be handled.

Open loop control

In an open loop control system the operation is predefined as a function of time and the actual state of the system is not considered. With this facility (manual) time scheduled control can be simulated.

Closed loop control

In a closed loop control system, control is based on the actual state of the system. This implies that the actual state of the system, for example a water level upstream of a pumping station, is being read by the controller. The controller transforms the input value into an output signal (Fig. 3.4.1). In this way the pumping station can be switched on/off depending on the actual upstream water level. The controller reads the actual water level and then determines whether the pumping station should be switched on or off.

The output signal is pre-defined function of an other variable. In the example of the pump, the discharge is a pre-described function of the upstream water level, in which the water level is called the function variable. The following function variables are standard available in MODIS:

- upstream water level
- upstream water level head (= water level - sill level)
- Ratio of the upstream and downstream head

Advanced closed loop controllers

The controllers mentioned above, are probably the simplest type of closed loop controllers. In MODIS more sophisticated closed loop controllers have been implemented as well. With these sophisticated controllers the output value is computed by the controller and not by a pre-described function. In MODIS the following type of advanced closed loop controllers have been implemented:

- step controller with dead band
- PID-controller with speed limitation

With these advanced controllers it is possible for example, to simulate the behaviour of a system equipped with automatic upstream and downstream controlled water level regulators.

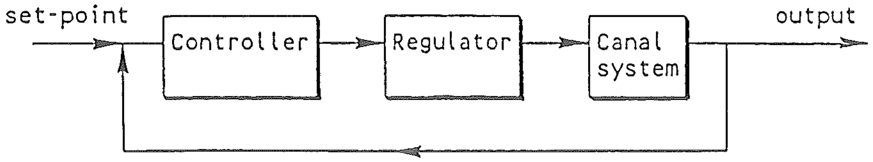


Figure 3.4.1 Principle of closed loop control

3.5 Operation performance criteria

To investigate to which extend the actual supply matches the intended supply and to compare operation alternatives, performance criteria have to be formulated. These so-called "operation performance criteria" comprises a delivery performance criterion and an operation efficiency. The first criterion specifies to which extend the intended supply is realized and the second criterion specifies with which efficiency the actual supply has been realized. These criteria are non statistical parameters and consequently easier interpretable than most statistical criteria.

Basically, the criteria make use of the actual volume of water delivered to an offtake, taking into account the flow rate and the moment in time. This implies that not only the volume of water delivered is considered to be important but also the flow rate with which this volume has been supplied and the moment in time of delivery. The user therefore has to specify the intended flow rate plus the tolerable variations in flow rate. Furthermore the period of time of delivery has to be specified. The model than computes in the specified period of time first the actual volume delivered Va and secondly the effective volume delivered Ve. The volume delivered is considered to be effective only when the flow rate is in between the tolerable range of flow rates.

Delivery performance

When the effective volume Ve is at least equal to the intended volume Vi, the delivery performance is perfect. For the offtake receives no deficit of water. This perfect performance results in a delivery performance ratio (DPR = Ve/Vi) of 100%.

Operation efficiency

However, there might still be losses of water. The losses consist of water which is supplied not within the specified range of flow rates and/or the volume supplied above the intended volume. The amount of losses are reflected in the operation efficiency e_0 (e_0 = Ve/Va). An operation efficiency of 100% indicates that no water is lost. However, it does not state anything about the delivery performance. The operation efficiency is part of the conveyance efficiency as defined by the ICID (Bos 1978).

3.6 Accuracy and stability

3.6.1 Numerical accuracy

The user of MODIS, and of any hydrodynamic flow model in general, should be well aware of the fact that the model can only provide an approximation of the exact solution of the de Saint Venant equations. This is an inevitable consequence of using discretized equations. (For a more detailed discussion of the mathematical backgrounds of MODIS, reference is made to chapter 11 and 12). Furthermore, the model results can only be as accurate as the (measured) input data describing the watermanagement system.

Accuracy of model computations

MODIS makes use of some computational parameters (Δx, Δt and θ), with which the user can influence the accuracy of the model computations. One must realize that the mesh size Δx and the time step Δt used to discretize the time space field, cannot be chosen at random. They must stand in reasonable relation to the dimensions (wave length) and time scale under investigation, as well as in a certain proportion to each other. Furthermore, the value of the time interpolation coefficient θ should be carefully selected. In general it can be stated that the simulation is most accurate when Δx and Δt are as small as possible, the Courant number is equal to unity and the time interpolation coefficient θ is equal to 0.5.

Mesh size Δx

The selected mesh size Δx is related to the problem under investigation. For an accurate representation of a backwater curve for example, several grid points should be located in that backwater curve. This usually results in a mesh size of order 100 m. For the representation of waves, the distance between two grid points should not exceed the wave length. Usually the wave length increases in time, due to diffusion. The small wave lengths usually increase in short time (order minutes) to an order of 100 m, resulting in a mesh size of order of 100 m. In many simulations one is not interested in an accurate simulation of the small waves, as the water volume involved is small as well. However, in case of closed loop control systems, an accurate representation of the small waves is often essential for a proper simulation of the controlled system.

Time step Δt

The chosen time step Δt should be related to the mesh size Δx. The numerical solution related to the wave celerity is most accurate when the so called Courant number equals unity [Abbott, 1979]. (Figure 3.5.1).

$$Cr = \{Q/A \pm \sqrt{(g.A/B)}\} \;.\; \Delta t/\Delta x = 1$$

In which,

Cr	=	Courant number	[-]
Q	=	discharge	[m^3/s]
A	=	cross sectional area	[m^2]

B = canal surface width [m]

g = acceleration of gravity [m/s^2]

The Courant number is not the only parameter that determines the time step. It should be noted that the accuracy of the regulators and controllers is also effected by the time step. The equation of the flow through the regulator is linearized and consequently only valid in a small range around the actual state. For an accurate computation of considerable variations, sufficient intermediate computations have to be made, resulting in small time steps. The same story holds true for the controller.

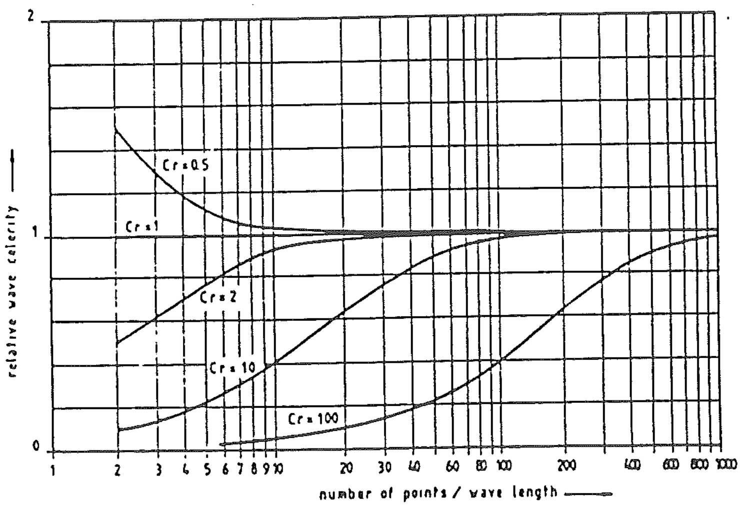


Figure 3.5.1 *Accuracy of model computation related to wave propagation celerity as a function of the number of grid points per wave length and the Courant number (Cr). The time interpolation coefficient (θ) is 0.5. Courant numbers greater than unity cause the computed wave to travel faster than in reality (Rubicon 1984).*

Time interpolation coefficient θ

The value of the time interpolation coefficient θ affects the wave deformation and should always be within the range 0.5 -1.0. A value for θ larger than 0.5 leads to damping of the wave, sometimes referred to as artificial or numerical diffusion. However, some artificial diffusion is desirable in order to damp small fluctuations which result from round of errors. Therefore usually a value of θ is selected of about 0.55 to 0.60. A value of 1.0 can be used when artificial numerical diffusion is desirable, for instance during a steady state run.

Concluding remarks

Accurate computations will lead to more computation time. Therefore always a compromise should be found between accuracy and computation speed. The user of MODIS should be well aware that the rules for accuracy of conventional flow models are not sufficient for MODIS in which closed loop controlled systems can be simulated. The proper time step is determined by the flow in the canals, the regulator and the controller. A practical and simple measure to evaluate the accuracy of the computations, is to repeat the computation with other (smaller) parameter values and compare the results. On a certain moment the results will not significantly change which indicates that it is of no use to further reduce the parameter values.

3.6.2 Numerical stability

Concerning the question of stability it should be noted that an implicit numerical scheme is used, which guarantees a numerical stable solution, when the time interpolation coefficient θ is larger than 0.5. (Stability has been defined in this user's guide as damped fluctuations). Although a stable numerical scheme is used, undesired fluctuations might occur. To damp these fluctuations, the time interpolation coefficient θ should be taken some closer to unity. However, it is well possible, especially due to the use of closed loop control, that physical instability occurs. In that case, the user should not try to damp these fluctuations but be aware of the physical origin.

Furthermore non-linear instability might occur. Non-linear instability results from linearizing the resistance coefficient and canal parameters such as the surface width and the cross-sectional area. In MODIS special care has been taken to avoid these non-linear stability problems, by iterating the computation (at least twice) and by evaluating the parameter values in between two time levels (see chapter 7).

The non-linear instability is in fact a result of an inaccurate computation and can be avoided by reducing the computational time step Δt. In practice instability mostly occur during a steady state run. Possible remedies in that case are:

- reducing the time step Δt
- a more accurate definition of the initial state.
- a time interpolation coefficient θ equal to 1.0
- A final solution is to simplify the canal configuration by closing some offtakes structures and opening them gradually during the run.

The eventually obtained steady state can be used as an initial state for further simulations.

3.7 Model calibration

3.7.1 Accuracy of input data

The accuracy of the model results is only as good as the input data. (This does not imply that inaccurate model simulations can be used when the accuracy of the input data is poor. It is just the opposite. When the input data is poor, accurate simulations should be made as not to increase uncertainty). The required accuracy of the input data is related to the desired accuracy of the model results.

Canal system

Normally the canal lay-out and canal length is known from maps. By

modelling the configuration, too detailed modelling should be avoided. Small inflow or offtakes can often be combined into one greater inflow or offtake point.
The canal profiles may alter in time and are frequently different from the designed profiles. The shape of the canal profile and the bottom levels are not so important because the model results are not very sensitive to those data. Consequently very detailed modelling of cross-sections is not required. Besides, possible errors can be compensated with the resistance coefficient. The flow width of the canal section should be measured more accurately because this determines the storage area of the canal and consequently effects the response time of the canal system considerably.

The resistance of the canal cannot be measured explicitly and should be determined in the calibration stage.

Regulators

The dimensions of the regulators are important as the flow is very sensitive for variations in dimensions. In practice, the actual state of many regulators is quite different from the original state and therefore it is worthwhile to check all regulators before modelling.

Boundary conditions

Finally the boundary conditions must be measured accurately. Particularly the accuracy of the inflow and the seepage (outflow) determine to a great extend the accuracy of the model results. Other boundary conditions can be taken so far downstream that they do not effect the model results.

3.7.2 Calibration

Calibration is a process of determining the best value of some (calibration) parameters by comparing real life measurement data with model simulation results. Model calibration is necessary only when an already existing water management system is modelled. In the design phase of new systems, no calibration is needed and consequently it is not always needed to use calibration data for MODIS. The main variable to be determined in the calibration phase is the resistance coefficient. This value may vary in time and space. Furthermore checks have to be made to confirm the magnitude of storage widths.

Verification

For calibration several real life measurement data sets are required, some of them are used to determine the value of the calibration coefficients and others are used to verify the calibrated parameters in order to raise the level of confidence in the simulation results.

Steady state

Normally calibration is first carried out in a steady state condition. This condition is achieved by keeping constant the inflow and gate openings. In order to compute the resistance parameter, it is advised to close as many gates as possible as not to disturb the computations by calibration errors of the regulators and non-uniform flow conditions (backwater curves). The same data sets can be used to estimate the amount of seepage losses (when the structures have been closed or calibrated accurately).

Simultaneously the regulators can be calibrated. However, it is most important to measure the right dimensions of the structures. The discharge coefficients can often be obtained from literature or a theoretical analyses. The error made is often small.

Unsteady state

In addition to the steady state calibration an unsteady state calibration can be carried out in order to check the measured input data and the previously computed resistance coefficient. In an unsteady calibration the discharge at the intake is increased and at several locations the flow rate and water level is measured at certain intervals of time. The propagation celerity of the disturbance is mainly determined by the storage width and the resistance in the canal. The measured data set can be used to verify these model parameters.

In addition the measured results can be used to raise the level of confidence in the model simulation results by comparing measured and simulated response time. However, calibration should in general not be used to verify the concept· of the model. Calibration is primarily required to asses certain parameter values such as the resistance coefficient.

4 Input data files

4.1 Structure of input data files

Input data is read by MODIS from input files. The input files which have to be defined by the user are similar to text files used in word processing. Each executable sub-program of the program package needs one user defined input file. All the user defined input files have a standard structure. An input file is built up of input segments, which in turn are built up of input blocks.

Input segments

Each input file is divided into input segments. An input segment is a group of input data of the same nature. For example, there is an input segment for the definition of branches and nodes, and another input segment has been reserved for the definition of structures. No special character has to be given to mark the beginning of an input segment. However, each input segment is closed by an input line containing the symbols "*END" in the first four columns of the data file.

Input blocks

Each input segment in turn consist of one or more input blocks. For example the input segment "structures" consist of a block for overflow structures, a block for underflow structures, et cetera. The beginning of an input block is marked by a so-called steer-key. (A steer-key is a word starting with the steer symbol "*" in the first column). For the input block overflow structures e.g., the steer key reads "*WEIR". The input block is closed by line containing the character "=" in the first column.

4.2 Data conventions

The input data of the user defined input file is format-free which implies that the location of the data along an input line is not important, and only the sequence should follow the guidelines. However, even format-free data processing has to follow some conventions. The general data conventions are:

- data items have to be given at all places where an input is expected;
- all input items are separated by at least one blank character;
- No data should be given in the first column. The characters given in the first column has a special meaning.

The symbols used in the first column are "+", "-", "*", and "=". The meaning of these symbols has been listed beneath:

+ comment line. Data specified on this line is not read by the program but copied to the echo file. The echo file is an echo of the input file in which error messages are include.
- comment line, Data specified on this line is also not read by the program but it is not copied to the echo file.
* steer key, used to mark the beginning of an input block or the end of an input segment (*END).
= used to mark the end of an input block;

If the first column is left blank, this indicates a normal data line. Data given on this line are read by the model

Note
Where in this description it has been specified that data are given on one line, all data should indeed be given on that line and not continued on another line. The field length of the lines is in principle limited to 80 characters. (However for some installations another field length is permitted). Comment lines can be included anywhere in the input. The use of "+" characters in the first column is especially useful as heading

for input data, whereas, the "-" characters can be used for deactivating data lines temporarily. An example of an input data file is given in Fig 4.2.1.

```
*ORIF
+
+ NAME    BRANCH    X-COORD    DIR   F     Ce    u    B    z       Y
+
  FS1      BP1         2998.0  PX    1.0   1.0  0.63 10   28.65    0.30
-  FS2      BP1         9998.0  =     =     =     =    =    27.95    0.55
  FS3      BP2        14002.0  =     =     =     =    =    27.55    LA$CNTR3
-  FS4      BP2        21998.0  =     =     =     =    6    26.75    0.90
  FS7      =          29298.0  =     =     =     =    =    23.10    0.53
  FS8      =          29598.0  =     =     =     =    =    21.85    0.53
  FS9      =          29998.0  =     =     =     =    =    20.55    0.55
  FS10     BP3        38002.0  =     =     =     =    5    18.40    LA$CNTR10
=
*WEIR
+ NAME    BRANCH      X-COORD    DIR     F       Ce      u    B      z
+
  WS3      BS3            5.0    PX    DT$FRED 1.0     1.5 12.0   TT$OFF1
  WS9      BS9            5.0    PX      =      =      1.5 12.0   TT$OFF2
=
*END    of ADDDEF
*AULC   AUTOMATIC LOCAL CONTROL
+
+ NAME   CONTROL  TYPE   Inival TARGET    Kp     Ki     Kd     SPEED  MIN   MAX
  CNTR3     PID     UC     0.23   30.15    -3.0   -1.8    0.5    0.5    0.0   2.0
  CNTR10    PID     UC     0.34   20.17    -3.0   -1.8    0.5    1.5    0.0   2.0
=
*END of automatic controllers
*TTIM          TABULATED FUNCTIONS OF TIME
+
  NAME               QINFLOW      OFF1      OFF2
+
  00:00:00:00         10.0        29.872    19.892
  00:07:59:00          =           =         =
  00:08:00:00         13.0        29.872    19.97
  05:00:00:00          =           =         =
+
=
```

Figure 4.2.1 Example of a part of an input data file

4.3 Data types

Data items can be of three different types:

- real value figures, characterized by a figure which may contain a point but not necessarily (e.g. 10.35 but also 10);
- integer value, characterized by a figure containing no point (e.g. 10, 1232);
- a character string, characterized by a mixture of alphabetical and numerical characters (e.g. STRUCT1, BRANCH2).

Character strings do not have to be enclosed in quotes ('). If these are specified, they form part of the character string. It is advised to work always with capital characters because the model distinguishes normal and capital characters which might lead to misunderstandings.

Names consist of any ANSI defined symbols and usually follow the convention that the length does not exceed 10 characters. Any characters following these 10 characters are truncated. The names should never contain blanks as this symbol has been reserved for separating input items. Instead of a blank the characters "-" may be used to form composite names. (So "BRANCH 2" is wrong but "BRANCH_2" is correct)

All input data have to be given in SI-units, except for the time which is given in a DHMS-notation (Days-Hours-Minutes-Seconds) as DDD:HH:MM:SS. (e.g. 6:10:28:15 stands for 6 days, 10 hours, 28 minutes and 15 seconds).

4.4 Special information characters

To facilitate easy input specification, the user can replace input data by so-called SIF-characters, Special Information Characters. A selection can be made out of the following SIF-characters:

- # substitute the default value or name specified in the program for this variable (can be used only in a limited number of places);
- = value or name is set equal to what has been given in the same data column on the previous line;
- < value or name is set equal to what has been given in the left hand neighbouring data column on the same line;
- > value or name is set equal to what has been given in the right hand neighbouring data column on the same line;
- ? interpolation of values or functions between data given above and below in the same data column.

In Fig. 4.2.1, these special information characters have been used too.

5 Model definition

5.1 Introduction

In the subsystem Model Definition (MODDEF) the configuration and the hydraulic characteristics of the open water system are defined. The input file of the model definition is usually called DEFIN.DAT. If a new input file is made it is advised to modify an existing input file instead of making a complete new input file. All standard data such as run control parameters, do not have to be typed again and only the data describing the irrigation system can be altered.

The input file of the model definition (usually called DEFIN.DAT) consists of seven input segments. As a special facility for slow computer systems each segment can be processed individually. However, to avoid inconsistency it is advised to reprocess the complete input file if changes are made.

In Table 5.1.1 all input segments and the steer-keys of their input blocks are presented. In the following of this chapter each input segment is discussed in detail.

TABLE 5.1.1 Description of the input segments of model definition

Segment number	Description	Input Blocks	Function
1	identification	*IDEN	run identification
		*SWIT	run switches
2	net definition	*NODE	definition of nodes
		*BRAN	definition of branches
3	grid definition	*CROS	cross-sectional data
		*TRAP	trapezoidal cross-sections
		*GRID	definition of grid points
4	structures	*LATQP	lateral point flows
		*WEIR	overflow structures
		*ORIF	underflow structures
		*PIPE	pipe opening structures
		*LOST	head loss structures
		*PUMP	pumps
		*NERP	neyrtec distributors
		*FDST	fortran defined structures
5	automatic control	*AULC	automatic local controllers
		*AURC	automatic remote controllers
6	functions	*TGFU	tabulated general functions
		*TTIM	tabulated time functions
		*FDFU	fortran defined functions
7	initial data	*INST	initial data of the system

5.2 Identification

The input segment IDENTIFICATION consist of two input blocks (see also Table 5.1.1):

*IDEN for giving the run identification parameters and
*SWIT for setting the run switches.

Normally the user can leave this input segment unchanged. Only if the name of the in and output files are changed input block *IDEN should be used.

5.2.1 Run Identification

The identification block starts after the steer-key *IDEN with 3 lines containing a standard code which is followed, after at least one blank, by a user-defined code or text. This code or text should not start before column 20.

This first part has the form:

RUN DESCRIPTION	users specified text (max 40 characters)
RUN CODE	users specified code (5 characters)
EXECUTED BY	users specified name (max 40 characters)

The codes in capitals have to be copied exactly.

The second part consists of information concerning file names. The model MODDEF must know the name of the input file and the names of the output files produced by MODDEF.
The input-, echo- and output files have to be specified in the job control for executing the job. The model data base and the result data base file names are needed to write the results in unformatted form to the right files. The names specified by the user here also should not be given before column 20.

This second part has now the form:

```
INPUT   file name         )
ECHO         file name    ) consistent with job control
OUTPUT       file name    )
MODELDB      file name    ) defined here
RESULDB      file name    )
```

The codes in capitals have to be copied again exactly. At the end of this input block a line is given with the character '=' in the first column. An example of the input block *IDEN is given in Fig. 5.2.1

```
*IDEN    RUN IDENTIFICATION
+            <............................>
         RUN-DESCRIPTION (40 CHARS:  MODIS RUN 1
+            <.......>
         RUNCODE                   :    TEST1
+            <................>
         EXECUTED BY             :   TU DELFT
+
+        DATASET (FILENAMES)
+
-        <..........>
         INPUT    DEFIN.DAT      (JCL.MODIFY)
         ECHO          DEFECH.DAT          "
         OUTPUT   DEFOUT.DAT           "
         MODELDB  USMDB.UNF
         SST-FILE USSST.UNF
=
```

*Fig. 5.2.1   Example of input block *IDEN (run identification)*

5.2.2 Switches

The run switches block begins with the steer-key *SWIT, followed by three data lines each containing 7 switches. The switch parameters activate or deactivate data transfer, blocks of switches, debug facilities, processing of input segments and the printing of their resulting output.

These switches have to be given in a specific sequence. To indicate the names, these appear in a list on a comment line. On the next input line the following characters can be given:

Y - activates the switch (abbreviation of YES);
N - deactivates the switch (abbreviation of NO);

On each line 7 switches are read. Some positions have no meaning yet as indicated by the heading 'FREE'. Here the default character "#" has to be given.

The switches with their meaning are given below in the sequence in which they appear in the input.

line 1

NEWMODDB	-	creates a new model data base
WRMODDB	-	writes processed results to the model data base
DEFALL	-	sets all the switches of line 2 to YES
OUTALL	-	sets all the switches of line 3 to YES
DEBUGOUT	-	switches debugging output on
FREE	-	reserved for extensions
FREE	-	" " "

line 2

DEFNET	-	switch for processing net definition in input
DEFGRD	-	" " " grid " "
DEFFUN	-	" " " function " "
DEFINI	-	" " " initial data " "

DEFLAT - " " " lateral flow " "
DEFSTRUC - " " " structure " " "
DEFACS - " " automatic control systems "

line 3

Similar as line 2 for producing outputs of the segments.

An example of the input segment *SWIT is given in Fig. 5.2.2

```
*SWIT- RUN SWITCHES     Y=YES        N=NO
+
+ NEWMODDB   WRMODDB    DEFALL    OUTFALL    DEBEGOUT    FREE
     Y          Y          N         N          N          #        #
+ DEFNET  DEFGRD   DEFFUN   DEFINI   DEFLAT   DEFSTRU  DEFACS
    Y        Y        Y        Y        Y        Y        Y
+ OUTNET  OUTGRD   OUTFUN   OUTINI   OUTFLAT  OUTSTRUC OUTACS
    Y        Y        Y        Y        Y        Y        Y
-
```

*Fig. 5.2.2 Example of input block * SWIT (run switches).*

Tip
If a steady state has been made and this steady state is used as the initial state (chapter 6), the model should not process the initial state of the Model Definition any more. To de-activate the processing of the initial state, the switches for DEFINI and OUTINI, should be set on N. However, if the canal configuration is changed, the switches should first be set on Y, till a new steady state has been made (see chapter 6).

5.3 Net definition

The input segment Net Definition consist of two input blocks:

*NODE for the definition of nodes, and

*BRAN for the definition of branches.

5.3.1 Nodes

The identification line for the node definition input block is:

```
* NODE        - DEFINITION OF NODES
+
+ NAME    TYPE         CONNECTING       BOUNDARY
+                      BRANCH(ES)       VALUE/REFERENCE
```

The data given in the lines starting with the character "+" in the first column are (optional) comment lines.

Data for each node are given on a new line. Internally the node names are transformed into numbers. The numbering follows from the sequence in which nodes are specified in the input. (This implies that the sequence also influences the band width of the coefficient matrix of the resulting node equations and hence the computer time required for solving this system of equations).

For each node the following data is required:

NAME	:	name of the node. Any name composed of ANSI characters is allowed. The length is limited to 10 characters.
TYPE	:	type of node condition (boundary condition); The type is given as a character string composed, in most cases, of two parts: an identification for the boundary type (node type) and the way in which data are given (data type). See Table 5.3.1 for possible node types.

CONNECTING BRANCHES	In principle any number of branches can be connected at a node. In the actual input processing, however, the number is limited to 7 (supposing that the array limit KNMAX has been set to 7 in the compilation). The names of these branches can be given in random order. A useful convention is to go around the node in clockwise direction. The length of the branch names again, are limited to 10 characters.
BOUNDARY VALUE	Where the data type has been specified as constant, the value can be substituted here directly either as an integer or as a real value. Where data are given in the form of a tabulated or Fortran function, reference is made here to the function name. The data should be given in the input part for the functions.

If all nodes have been specified the input list for nodes is closed with a line containing the "=" character in the first column.

TABLE 5.3.1 Type of node conditions

H	-	water level given;
Q	-	discharge given;
QH	-	discharge - stage relationship;
CF	-	critical outflow condition;
JW	-	junction with water level compatibility condition (no storage);
JS	-	junction as JW with storage (e.g. flood plain cell).

The value of each type of node condition can be given as:

C	-	constant value;
T	-	tabulated function;
F	-	Fortran-defined function;

From these options the following combinations are possible:

- HC - water level given as a constant value. The value is given with reference to the general model datum level;
- HT - water level given as a tabulated function of time;
- HF - water level given by any user-defined description in the form of a Fortran function;
- QC - discharge given as a constant value (e.g. Q = 0);
- QT - discharge given as a tabulated function of time;
- QF - discharge given by any user-defined description in the form of a Fortran function;
- QHT - discharge given as a tabulated function of the water level;
- QHF - discharge given as a function of the water level specified by the user in the form of a Fortran function.
- CF - critical outflow from the channel at the node. The parameters follow from the channel cross-section at the node;
- JW - water level compatibility at the node. Where energy level compatibility is preferred the boussinesq coefficient β can be set equal to zero.
- JSC - storage node with water level compatibility and storage defined as a constant value;
- JST - storage node with storage given as a tabulated function of the water level;
- JSF - storage node with storage defined through a Fortran function.

Note:
For the node types CF and JW no additional data are required and the space under "boundary value/reference" has to be left blank.

An example of the input block *NODE is given in Fig. 5.3.1

An example of the input block *bran is given in Fig. 5.3.2

```
*BRAN     - DEFINITION OF BRANCHES
+
+NAME     BEGIN-NODE     END-NODE     X-BEGIN   X-END    DX-MAX
+
BR1       NUP            NDIV               0   20000    2000
BR2       NDIV           NFP1           20000   42000    1000
BR3       NFP1           NFP3           42000   52800    2000
BR4       NFP3           NRES           52800   73000       -
BRES      NRES           NDOWN          73000   73010       -
BLB1      NDIV           NLB                0   12600       -
BLB2      NLB            NRES           12600   56300       -
BRB       NLB            NRES               -   58200       -
BFP1      NFP1           NFP2               0      20       -
BFP2      NFP2           NFP4               -       -       -
BFP3      NFP3           NFP4               -       -       -
BFP4      NFP4           NFP5               -       -       -
BFP5      NFP5           NFES               -       -       -
-
```

*Fig. 5.3.2 Example of input block *BRAN (branch definition).*

5.4 Grid definition

The input segment GRID DEFINITION consist of three input blocks:

*CROS - definition of cross sections
*TRAP - definition of trapezoidal cross sections
*GRID - definition of a computational grid

The first two input blocks have the same function and are used for the generation of cross-sectional data whereby the bed resistance and canal profile are defined. Input block *CROS can be used for irregularly shaped cross-sections, whereas input block *TRAP can be used for trapezoidal cross sections. The program transforms the trapezoidal cross-sections into tables, in the same format as the irregularly shaped cross-section tables. However in the computation, the cross-sectional tables are not used as the model computes the cross-sectional data. In this way interpolation errors are avoided. The reasons for generating the cross-sectional tables is that the tables are used for system generated cross-sections.

In case of non-uniform canal, the model generates cross-sections in intermediate points if the "?" character is used for the cross-section name (see *GRID). The system generated cross-sections are treated in the same way as the cross-sections defined in *CROSS.

The input block *GRID is used to define a computational grid over the branches.

5.4.1 General cross-sections

Cross-sections can be specified in any sequence. For each cross-section an
input identification line is required of the form:

```
*CROSS  -  Cross-sectional parameters
+
 NAME          users specified name
 REFERENCE     users specified reference level
```

Where after the key-word NAME and at least one blank the name of the cross section is substituted, following the standard name conventions. The key-word NAME may start in any column, except the first one of the input line.

On the next input line the reference level of the given cross section data is given, after the key-word REFERENCE and at least one blank. This level is the reference level of the cross section.

Note: this reference level is not necessarily the level above the general model datum level as a correction is still possible under the input for the grid specification. To avoid possible errors it is advised to put the reference level on zero as much as possible.

After the name and the reference level the cross-sectional parameters are given as tabulated functions of the water level. Data for each level are given on a new input line in the sequence as shown in the optional heading:

```
+ LEVEL  AREA   BS   BF   BOUSSINESQ   RESISTANCE   RADIUS
```

On each subsequent line 7 input data have to be given as a function of water level (all in metric units).

LEVEL	:	water level above reference
AREA	:	cross-sectional area conform to the specified water level
BS	:	storage width conform to the specified water level
BF	:	flow width

BOUSSINESQ: Boussinesq coefficient, normally a value of 1 or slightly higher is found

RESISTANCE: resistance coefficient in the form of a Manning or Strickler coefficient.

RADIUS : hydraulic radius.

Data for at least two levels have to be specified, while the model interpolates between two levels. The input lines have to show water levels in increasing order. If the distance between two water levels is too high, the interpolation might become inaccurate. As usual, the input block for information on a specific cross-section is closed again by giving the "=" character in the first column of a separate last line.

Note:

- The use of SIF-characters is especially convenient for this input item, e.g. by setting a constant Manning coefficient for all levels, storage width equal to flow width etc.
- Special care should be taken that the canals do not run dry. If the water depth in the canals becomes zero (or negative), the momentum equation can not be solved anymore and an error message will occur. Improvements can be made by extending the canal cross-section with a slot in the bottom. This so called Preissmann slot has a small width and a high resistance coefficient and may keep the program running. The model prevents (to some extend) the slot from running dry. (In input block *TRAP, a Preissmann slot is automatically generated at the bottom of each cross-section.

An example of the input block *CROS is given in Fig. 5.4.1

```
*CROSS          - CROSS-SECTIONAL PARAMETERS
+
 NAME           RIVER
 REFERENCE      0.00
+
+LEVEL    AREA      BS      BF       BOUSSINESQ        RESISTANCE   RADIUS
+
  0.000     0.      100.     <           1.05      .030            0.00
  1.000   101.      102.     <           =         =               1.00
  2.000   205.      106.     <           =         =               1.00
  5.000   315.      110.     <           =         .025            5.00
  5.200   335.      130.     110.        =          =              5.20
=
```

*Fig. 5.4.1 Example of input block *CROS (general cross-sectional data)*

5.4.2 Trapezoidal cross-sections

If trapezoidal cross-sections are used, these cross-sections can be defined far more easily in this input block. The input identification line is of the form:

```
*TRAP  Trapezoidal cross-sections
+
+ Name  Reference  Boussinesq  Resistance  Bottom Width  Side slope  Hmax
+
```

The following data is required for each input line:

NAME :	Name of cross section, following the standard name conventions. Later in the input block *GRID this name is coupled to a branch.
REFERENCE:	Reference level of the given cross section data. Normally a reference level of zero can be used as later in the input block *GRID, the reference above (or below) project datum can be given.
BOUSSINESQ:	Boussinesq coefficient. For trapezoidal cross-sections normally a value of 1.0 or 1.05 can be used.
RESISTANCE:	Manning or Strickler resistance coefficient can be used. (The model automatically transforms the strickler coefficient into a Manning coefficient if the value of the resistance parameter exceeds 1).
BOTTOM :	The bottom width of the cross-section
SIDE slope:	The value of m has to be given whereby the side slope is defined as 1 vertical : m horizontal.
HMAX :	The maximum water depth. Only needed for the cross-sectional tables generated by the model. The model automatically increases the maximum water depth (maximum + 1.00 m) to avoid problems if the water level exceeds the maximum depth.

As usual, the input block is closed by an input line with the symbol "=" in the first column.

Notes

The program automatically generates a so called Preissmann slot at the bottom of each trapezoidal cross-section. This slot does not contribute to the canal flow but prevents the canal falling dry and consequently avoids to some extend termination of the program.

The maximum water depth (HMAX) should not be taken too high, as the model generates a cross-sectional table with a limiting number of rows. In between those rows, data is linear interpolated. If the maximum water depth is too high, the interpolation errors might become too high. However this consideration is only valid in case system generated cross-sections are applied in non-uniform canals. For trapezoidal cross-section data is not read out of the generated tables but computed out of the specified data. This has been done to avoid the linear interpolation errors

An example of the input block *TRAP is given in Fig. 5.4.2.

```
*TRAP  Trapezoidal cross-sections
+
+ Name Reference  Boussinesq  Resistance  Bottom Width  Side slope  Hmax
+
  CROSS1   0.00        1.0        .35          2.0           2.0      0.00
  MAINC1    -           -          -          10.0           2.5      5.50
  LATC02   8.00        1.05       40.          7.0           2.1      5.00
-
```

*Fig. 5.4.2 Example of input block *TRAP (trapezoidal cross-sections)*

5.4.3 Grid - definition

For the numerical integration of the de Saint Venant Equations, a computational grid is required along each branch. This grid is automatically generated by the system following the directives set under this input item and the maximum value for Δx specified in the branch input (see paragraph 5.3.2). The system calculates the number of grid points which have to be generated between the user-defined grid points in order to satisfy the condition on the maximum step size Δx.
The grid points are also used to link the hydraulic canal data to the branches by specifying for each grid point the branch and its location in that branch and the name of the cross-section table.

For each branch at least two grid points have to be defined by the user: one at the beginning and another at the end of the branch. The definition of additional grid points in between these boundary points is needed only if structures are located inside a branch. In that case upstream and downstream of that structure additional grid points need to be defined. Of course the boundary grid points can also be used as grid points upstream and downstream of a structure. However this is only allowed as long as the distance between the structure and the boundary points is limited, or if the dynamic behaviour of the branch can be neglected.

The identification line for the grid definition reads as:

```
*GRID         -     DEFINITION OF GRID POINTS
+
+ GRID-POINT  CROSS-SECTION  BRANCH  X-COORDINATE  REFERENCE
+  NAME         NAME          NAME                   LEVEL
```

On each subsequent line information about a user-defined grid point is given in the sequence:

GRID POINT: name of the grid point given by a character string following the standard convention for system element names

CROSS-SECTION:	name of a cross-section which has been specified as described in par. 5.4.1 & 5.4.2;
BRANCH :	name of the branch in which the grid point is located
X-COORDINATE:	x-coordinate in the branch for this grid point;
REFERENCE :	reference level, giving the position of the reference level of the cross-section data with respect to the general model datum level.

The input of lines does not require a special sequence, neither for giving the information per branch, nor for following the x-direction in the branch. It is, however, advisable to group the data per branch and separate these groups of lines by blank comment lines.

Note:
The interpolation SIF-character "?" (= interpolation of row data) can conveniently be used in the column for cross-section names to interpolate the cross-section tables at intermediate grid points. This avoids the manual calculation of such cross-sectional data where an additional user-defined grid point is introduced merely for reasons of, say, output of results.
In addition the SIF-character "?" can conveniently be used in the column for reference level, if the canal has a constant slope. In Fig. 5.4.3 an example of the input block *GRID has been given.

```
*GRID      -        DEFINITION OF GRIDPOINTS
+ GRIDPOINT   CROSS-SEC     BRANCH   X-COORDINATE  REF LEVEL
  G11         CROSS1        B1         0        10.00
  G12         ?             =        500        9.90
  G13         ?             =        510        9.80
  G14         CROSS2        =       1500        9.75
  G21         CROSS3        B2         0        ?
  G22         =             =       1250        950
=
*END OF GRDDEF
```

*Fig. 5.4.3   Example of input block *GRID (grid definition)*

5.5 Structures

5.5.1 Introduction

If structures are placed in the canal system or if lateral in or outflow occur, the input segment STRUCTURES should be used. Structures and lateral flow can be placed anywhere in a branch and should be located in between two grid points. If structures are placed in parallel, all the parallel located structures should be placed in between the same grid points. If at the other hand structures are placed in series, they should be separated by grid points. The input segment structures consists of several input blocks:

*LATQP	-	later point flows
*WEIR	-	overflow structures
*ORIF	-	underflow structures
*PIPE	-	pipe structures
*PUMP	-	pump structures
*LOST	-	local loss structures
*NERP	-	Neyrtec module distributors
*FDST	-	fortran defined structures

Each input block will be discussed subsequently. If an input block is not used, the complete block including its keyword can be skipped.

The structure parameters (such as sill level or gate opening height) can be given either as a constant value or as a function of an other parameter. If the latter is used, this function can be given in tabulated form, as a fortran defined function or as a function of an automatic controller.

If a constant value is used, this value can be given. In the other cases a name is given which refers to a table, a fortran function or an automatic controller. Automatic controllers can only be used for the weir and orifice type of structures. The reference convention consist of four parts (Table 5.5.1):

1) A one character code for the type of variable of which the parameter is dependent of (e.g T = time, H = upstream water level).
2) A one character code for the function type, T for tabulated functions and F for fortran defined functions.
3) After the two character codes selected from the list, the symbol "$" must be given.
4) Finally the name of the function must be given. The name is limited up to 7 characters.

TABLE 5.5.1 **Function references for standard structures except pumps and lateral inflow**

Part 1	**:**	**code description**
	T	- Time
	H	- Water level
	Z	- Crest level
	A	- Cross sectional Area of opening
	U	- Upstream head over the structure
	D	- Ratio downstream/upstream head
	O	- Opening height of the gate
	Y	- Ratio gate opening height/upstream head
	L	- Local Automatic control
	R	- Regional Automatic control
	G	- Global Automatic control (not yet included)
PART 2	**:**	**Code Description**
	T	- Tabulated form
	F	- Fortran function
	A	- Automatic control
PART 3	**:**	**$ (dollar sign)**
PART 4	**:**	**Users defined name, maximum 7 characters**

Remark
- head is defined as water level minus crest level of sill.
- Automatic control is only possible in combination with Local, Regional or Global automatic control, e.g. "LA" or "RA".

5.5.2 Lateral discharges

The identification for this input item is given by:

```
*LATQP        -    LATERAL POINT FLOWS AND PUMPS
+
+ NAME     BRANCH   X-COORDINATE    DATA TYPE  VALUE/REFERENCE
```

For each lateral flow source the information is given on a separate line under the heading in the sequence:

SOURCE NAME	:	point source name of maximum 10 characters;
BRANCH NAME	:	branch name
X-COORDINATE	:	x-coordinate in the branch;
DATA TYPE	:	C for constant value;
		F for a fortran defined function
		TT for a tabulated function of time
		HT for a tabulated function of upstream water level
		AT for a tabulated function of cross-sectional area
		PT for a tabulated function of wetted perimeter
VALUE	:	Real value of discharge (m3/s) if data type is C (constant) and reference name if data type is a (T) tabulated or (F) fortran defined structure

Note

Remark that for lateral flow other function parameters are defined than for structures as specified in Table 5.5.1. The lateral flows can be specified in random sequence. However, it is advised also here to group them following a well organized pattern as to keep an overview of all data. An example of the input block *LATQP is given in Fig. 5.5.1.

```
*LATQP- LATERAL POINT FLOWS
+ NAME   BRANCH-NAME   X-COORD   TYPE   REFERENCE/VALUE
+
 LL1     BLB2          15800.     C      5.80
 LL2     -             48500.     HT     HTAB
 LR1     BRB           16200.     TT     LATQRBI
-LR1     BRB           16200.     C      5.20
 LR2     -             43400.     PT     TAB1
-
```

*Fig. 5.5.1 Example of input block *LATQP (lateral point flows)*

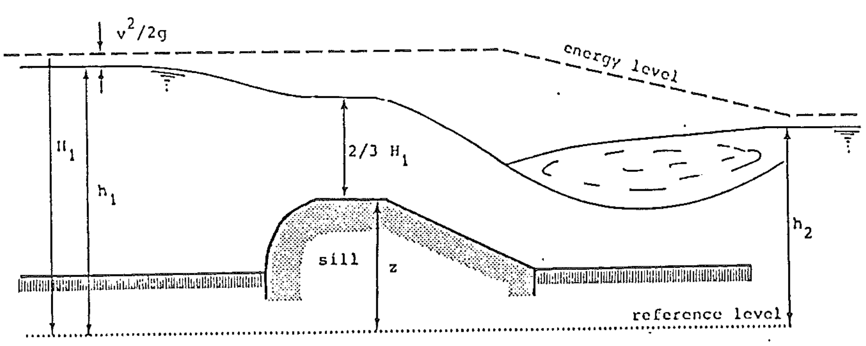
Tip

With lateral flow it is possible to simulate canal seepage. In that case flow is subtracted (negative discharge) from the canal. The amount of subtracted flow can be given as a (tabulated) function of the wetted perimeter of the canal (DATA TYPE is PT).

5.5.3 Overflow structures

Introduction

Overflow structures are structures whereby the upper water surface is not touched by the structure. Both free flow and submerged flow conditions can occur and the model switches automatically to the right flow condition. Furthermore both broad crested and sharp crested weirs can be modelled. In principle, overflow structures with any control section can be modelled, as the exponent u in the head discharge equation (Eq. 5.1) is specified by the user. As a special facility the structure can act as an automatic water level control structure. Structure parameters can be given both for a positive and a negative flow direction. If no negative flow direction has been specified the discharge in the negative direction is assumed to be zero. The applied equation for overflow structures reads (Bos, 1976):



$$Q = f\, c_e\, \frac{2}{3} \sqrt{\left(\frac{2}{3} g\right)}\, b\, (h-z)^u \qquad 5.1$$

Where

Q	= discharge	m^3/s
c_e	= effective discharge coefficient	-
f	= drowned flow reduction factor	-
h	= upstream water level head	m
b	= width of the weir	m
z	= crest level of weir	m
u	= exponent, normally 1.5	-
g	= acceleration due to gravity	m/s^2

For more detailed information about overflow structures, reference is made to par 11.2.2.

Input data

Under the heading:

```
*WEIR - overflow structure
+
+ Name   <. .Location. .>     <...    ..parameters..      ...>
+ Name   Branch   X-coor  Dir    f       Ce      u       B      z
```

The following data have to be given in sequence on each input line:

NAME	:	Name of the structure (maximum 10 characters). For each structure a new name has to be specified.
BRANCH	:	Branch name in which the structure is located.
X-COOR	:	Position along the x-axis in the branch.
DIR	:	Flow direction for which the structure parameters are used. This is specified by a string of two characters: PX : for flow in positive x-direction through the structure. NX : for flow in negative x-direction through the structure.
F	:	Drowned flow reduction factor. If the flow through the structure is always modular (free flow), a value of 1.0 can be given.
CE	:	Effective discharge coefficient.
U	:	The power used in the weir flow equation. Normally a value of 1.5 is used. For non rectangular control sections other values may occur.
B	:	Width of control section of the weir.
Z	:	Sill level of the weir.

If all structures have been defined, the input segment is closed by a line with the symbol "=" in the first column.

Notes

- A smooth transition between free and submerged flow can be achieved by specifying the value of the drowned flow reduction factor f as a function of the ratio between the downstream and upstream head e.g. DT$FRED (see Table 5.5.1 and Fig 11.2.2).

(h2-z)/(h1-z)	FRED
0.00	1.00
0.67	1.00
0.80	0.75
0.90	0.55
1.00	0.00

- The same procedure can be adapted for specifying the effective discharge coefficient Ce. By giving the value of Ce as a function of the upstream head (e.g. UT$CE), the transition from a sharp to a broad crested weir can be simulated.
- Automatic control of an overflow structure by means of controllers is possible. Either the parameter B or Z can be automatically adjusted in order to keep a constant water level somewhere in the canal system. In that case reference is made to input segment automatic control, using the standard function reference (e.g. LA$CONTR or RA$CONTR, paragraph 5.6).
- Sometimes it is needed to use the energy head H, instead of the water level h in Eq. 5.1. This can be achieved by specifying an additional grid point close to the structure with a Boussinesq coefficient β of the upstream cross-section equal to zero. In that way the velocity head is transformed into a static head [Rubicon, 1984].

An example of the input segment *WEIR is given in Fig. 5.5.2

```
*WEIR - overflow structure
+ Name  <. .Location. .>      <...      ..parameters..        ...>
+ NAME  BRANCH  X-COOR  DIR   F         Ce      u      B      Z
  W1    B2      950     PX    DT$FRED   UT$CE   1.5    2      4.0
  W1    -       -       NX    -         -       -      -      -
  REG   B8      375     PX    1         0.9     1.5    6      LA$CONTR1
-
```

*Fig. 5.5.2   Example of input segment *WEIR (overflow structures)*

Notes:

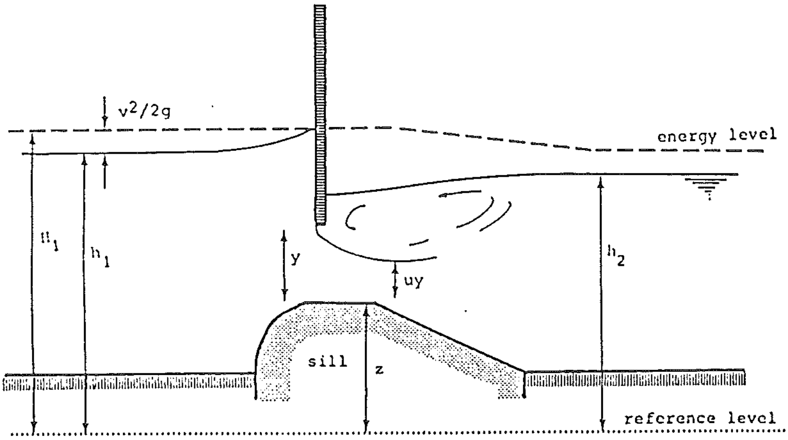
- The structure W1 can flow in two directions. The sill height of the structure REG is automatically adjusted (LA$CONTR). All functions and controllers have to be specified later.

5.5.4 Orifice structures

Introduction

Underflow structures are structures whereby the upper flow surface is touched by the structure gate. However, if the upstream head becomes too low the orifice structure will automatically act as a (free or submerged) weir.

Under the heading "*ORIF" underflow structures are grouped. The input segment for underflow structures is very much the same as for overflow structures. The equation for both free and submerged, orifice flow with a rectangular control section and variable gate opening reads:



$$Q = c_e \mu b y \sqrt{2 g \Delta h}$$ 5.2

Where

Q	=	discharge	m^3/s
c_e	=	effective discharge coefficient	-
μ	=	opening contraction coefficient (0.63)	-
b	=	width of control section	m
y	=	opening height of gate	m
dh	=	head difference.	m
g	=	acceleration of gravity	m/s^2

If the downstream water level does not effect the discharge (modular flow) the head difference (dh) becomes:

$$\Delta h = h_1 - (z + \mu y) \qquad 5.3$$

Where

z	=	sill level	m
μ	=	opening contraction coefficient	-
y	=	gate opening height.	m

If the flow is submerged, the head difference (dh) is equal to:

$$\Delta h = h_1 - h_2 \qquad 5.4$$

Where h_1 and h_2 are the upstream and downstream water levels respectively. The user can specify if the flow becomes submerged. This is done by using a flow factor f, which can be specified as a function of upstream and downstream water level head and gate opening height. (Beware that this factor f is not the same as the drowned flow reduction factor of the weir formula presented in Equation 5.1).

Input

Under the heading:

```
*ORIF - underflow structure
+ Name  <. .Location. .>      <...      ..parameters..       ...>
+ NAME  BRANCH   X-COOR  DIR    F       Ce      U      B     Z     Y
```

The following data has to given in sequence in each input line:

NAME : Name of the structure (maximum 10 characters). For each structure an unique name must be given.

BRANCH : Branch name in which the structure is located.

X-COOR : Position along the x-axis in the branch.

DIRECTION : Flow direction for which the structure parameters are used. The flow direction can be either:

- PX : for flow in positive x-direction through the structure.
- NX : for flow in negative x-direction through the

structure.

F	:	Drowned flow coefficient which value is in between 0 and 1. If the flow through the structure is always submerged f should be equal to one. If at the other hand there is always free flow, f should be equal to zero.
Ce	:	Effective discharge coefficient.
U	:	Opening contraction coefficient (≈ 0.63).
B	:	Width of control section.
Z	:	Sill level.
Y	:	Opening height of the gate

If all structures have been defined, this input segment is closed by a line with the symbol "=" in the first column.

Notes

Beware that the F-coefficient for orifice structures is different from the drowned flow reduction coefficient used in overflow structures ! The computer model checks which type of flow is applicable by comparing the actual ratio of the gate opening height and the downstream head with the value of f. If the actual ratio is higher than the value of f, modular flow is assumed (see paragraph 11.2.3). The factor f can be given as a constant value or as a function of the ratio gate opening height and upstream water level head YT$FCOEF. For example:

h2-z/y	FCOEF
0.000	0.000
0.001	0.002
0.100	0.227
0.125	0.256
0.143	0.270
0.166	0.300
0.200	0.333
0.250	0.377
0.333	0.456
0.500	0.588
0.667	0.770
1.000	1.000

If the downstream water level is below the gate opening height and the upstream head becomes less than 1.5 times the gate opening height, the structure will act as a (free or submerged) weir with a c_e coefficient of 1.0.

The contraction coefficient can be given as a constant or as a function of the gate opening height and upstream head e.g. YT$CNTRAC.

The orifice can act as an automatic controlled water level regulator by making reference to the automatic controllers instead of giving a value for the gate opening height.

An example of the input segment *ORIF is given in Fig. 5.5.3

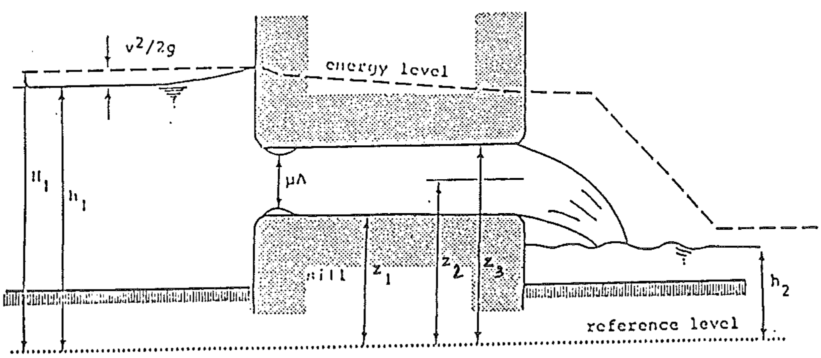
```
*ORIF - underflow structure
+
+ Name  <. .Location. .>      <...        ..parameters..         ...>
+ NAME  BRANCH  X-COOR  DIR   F       CE    U       B     Z      Y
+
 GATE1    B2     950    PX  YT$FCOEF 0.8   0.63    2.0  11.20  RA$CNTR
  -       -       -     NX   0.0      0.0   -       -     -       -
 ORIF     B8     375    PX   1.0      0.9   0.70    6    10.20  TT$OP1
-
```

*Fig. 5.5.3 Example of input segment *ORIF (orifice structures)*

5.5.5 Pipe structures

Introduction

Pipe structures can be used to model flow through pipe offtakes and horizontal culverts. The characteristics of pipe flow are complicated. Several flow conditions may occur, whereby the pipe can be completely or partly filled. In principle, hydraulic long culverts can be modelled only. In case of hydraulic short culverts, whereby the culvert is partly filled and acts as an orifice, the user is advised to model the culvert as an orifice structure. [Ven te Chow, 1959]. If the pipe is completely filled, submerged flow, free flow and aerated free flow can occur. The pipe will act as a weir if the downstream level is below the top of the pipe opening and if the upstream water level is below a critical level. This critical level can be specified by the user and normally varies between 1.5 and 1.2 times the pipe diameter.



The general equation for full pipe flow reads:

$$Q = \frac{1}{\sqrt{c_t}} A \sqrt{2 g \Delta h} \qquad 5.5$$

Where

Q	=	discharge	m^3/s
c_t	=	total loss coefficient, consisting of entrance, friction and exit losses	-
A	=	cross-sectional area of gate opening	m^2
dh	=	head difference .	m

The head difference (dh) depends on the flow condition. Three flow conditions are distinguished:

aerated flow

If the downstream water level is below the bottom level of the opening, the flow is fully aerated and dh becomes:

$$\Delta h = h_1 - \frac{1}{2}(h_T - h_{bottom}) \quad 5.6$$

Where h_1 is the upstream water level, h_{top} is the top of opening and h_{bottom} is the bottom level of the pipe.

modular flow

If the downstream water is above the bottom level and below the top of opening dh becomes:

$$\Delta h = h_1 - h_T \quad 115.7$$

submerged flow

If the downstream water level is above the top level of opening, the flow is submerged and dh becomes:

$$\Delta h = h_1 - h_2 \quad 5.8$$

Where h_2 is the downstream water level.

Partly filled pipe flow

If the downstream water level is below the top of opening of the pipe, the pipe can be partly filled, depending on the upstream water level. If the upstream water level is above a critical level, the pipe is assumed to be completely filled. The critical level is equal to :

$$h_{critical} = h_{bottom} + f\,(h_T - h_{bottom}) \quad 5.9$$

Whereby the factor f is specified by the user. Normally f varies between 1.5 and 1.2, the latter for longer pipe lengths.

If the upstream water level is below the critical level the pipe will be partly filled and the weir flow equation is applied. The flow can be either free or submerged depending on the ratio of the upstream and downstream head. If the downstream head is less than 2/3 of the upstream head, free weir flow is assumed:

$$Q = 1.7 \frac{A}{h_t - h_{bottom}} (h_1 - h_{bottom})^{1.5} \qquad 5.10$$

If the downstream head is greater than 2/3 of the upstream head, drowned weir flow will be assumed:

$$Q = \frac{A}{h_t - h_{bottom}} (h_2 - h_{bottom}) \sqrt{2 g \Delta h} \qquad 5.11$$

The loss coefficient

The loss coefficient c_e consist of :

-entrance loss coefficient

$$c_{entrance} = (\frac{1}{\mu} - 1)^2 \qquad 5.12$$

-resistance loss coefficient

$$c_{resistance} = \frac{2 g L}{k^2 R^{4/3}} \qquad 5.13$$

-exit loss coefficient (1.0)

$$c_{exit} \approx 1.0 \qquad 5.14$$

Where, R = hydraulic radius of pipe (= A/πD), D = diameter of pipe, k = Strickler resistance coefficient and L = length of pipe.

Input

Under the heading:

```
*PIPE - pipe structure
+
+       <. .LOCATION. .>      <...      ..PARAMETERS..        ...>
+ NAME  BRANCH   X-COOR  DIR    F       Ct      A     Hbottom  Htop
```

The following data has to given:

NAME : Name of the structure (maximum 10 characters).

BRANCH : Branch name in which the structure is located.

X-COOR : Position along the x-axis in the branch.

DIRECTION : Flow direction for which the structure parameters are used. The flow direction can be either:

- PX : for flow in positive x-direction through the structure.
- NX : for flow in negative x-direction through the structure.

F : F-coefficient specifies the critical level below which weir flow will occur. The value of f is normally in between 1.0 and 1.5.

Ct : Total loss coefficient.

A : Cross-sectional area of pipe opening.

Hbottom : Bottom level of opening.

Htop : Top level of opening

When all structures have been defined, this input segment is closed by a line with the symbol "=" in the first column. An example of the input segment *PIPE is given in Fig. 5.5.4.

```
*PIPE - pipe structure
+ Name   <. .Location. .>      <...        ..parameters..        ...>
+ NAME    BRANCH  X-COOR  DIR  F       Ct   A      Hbottom    Htop
  PIPE1    B2      950     PX   1.3     2.5  2.0    10.30      10.60
  OUTLET   B8      375     PX   1.5     1.9  0.70    9.30       9.70
=
```

*Fig. 5.5.4   Example of input segment *PIPE*

5.5.6 Pump structures

Introduction

Pump structures are non gravity flow structures which implies that the flow through the pump structure is not influenced by the upstream and downstream water levels, but the discharge is pre-described by the user.

Operation of the pump is controlled by specifying a start and stop water level. Pump operation is started if the actual water level exceeds the start level and pump operation is stopped if the actual water level drops below the stop level. The stop water level should always be below the start water level in order to avoid too frequent on and off switching.

If a pump station consisting of several pumps with different start and stop water levels need to be modelled, the user is advised to model several pumps with different names but on the same location, each one with its own start and stop levels.

Input

Under the heading

```
*PUMP - pump structures
+
+ NAME   BRANCH   X-COOR   DIR   F   CAPACITY   H-START   H-STOP
```

The following data has to be given:

NAME	:	Name of the pump structure
BRANCH	:	Branch name in which the structure is located
X-COOR	:	Position along the x-axis
DIR	:	Water level which controls pump operation: PX - Upstream water level NX - Downstream water level
F	:	Reduction factor of pump capacity, can be given e.g. as a tabulated function of the head.
CAPACITY	:	Capacity or discharge of the pump if the pump is in operation

H-START : Water level above which pump is turned on

H-STOP : Water level below which pump is turned off

If all pump structures have been defined, this input segment is closed by a line with the symbol "=" in the first column.

Example

An example of this input block is given in Fig. 5.5.5

```
*PUMP - pump structures
+
+ NAME   BRANCH     X-COOR   DIR   F         CAPACITY  H-START    H-STOP
  PUMP1   BRANCH3   2000     PX    HT$HEAD   2.50      10.45      10.35
  PUMP2   BRANCH3   2000     PX    HT$HEAD   1.50      10.55      10.45
=
```

*Fig. 5.5.5 Example of input segment *PUMP*

5.5.7 Local loss structures

Introduction

Local loss structures are frequently found along irrigation canals. Example of local loss structures are bridges and culverts but also abrupt profile changes. The input segment for local loss structures is very much the same as the definition of other type of structures.

The head loss due to abrupt profile changes can be computed using the formulae:

$$\Delta h = c \frac{(v_1 - v_2)^2}{2g} \qquad 5.15$$

while the head loss of a loss structures can be computed using the formulae:

$$\Delta h = c \frac{v_1^2}{2g} \qquad 5.16$$

Where

dh	=	head loss	m
c	=	loss coefficient	
v_1	=	flow velocity at the vena contracta	m/s
v_2	=	flow velocity downstream	m/s

Input

Under the heading:

```
*LOST - local loss structures
+
+ Name  <. .Location. .>              <..parameters..>
+ Name  Branch   X-coor   Type   Dir   C      A
```

The following data has to be given:

NAME	:	Name of the structure (maximum 10 characters).
BRANCH	:	Branch name in which the structure is located.
X-COOR	:	Position along the x-axis in the branch.

TYPE : The type of losses, which can be either:

- CL - for Carnot type loss description which is valid for profile changes.
- HL - standard head loss description which is valid for bridges and culverts.

DIR : Flow direction for which the structure parameters are used. The flow direction can be both :

- PX - for flow in positive x-direction through the structure.
- NX - for flow in negative x-direction through the structure.

C : Loss coefficient which indicates which amount of the velocity head is considered as a head loss.

A : Cross-sectional area of structure. If a Carnot type of headloss has been selected, this parameter can be left blank, as the head loss is computed out of the canal cross-sections.

If all local loss structures have been defined, this input segment is closed by a line with the symbol "=" in the first column.

An example of the input segment *LOST is given in Fig. 5.5.6

```
*LOST - local loss structures
+ Name  <. .Location. .>          <. .parameters. .>
+ Name  Branch  X-coor  Type Dir    C         A
  BRIDGE1  B2     950   HL    PX   0.50     15.0
  CANAL    B7     500   CL    PX   0.5
=
```

*Fig. 5.5.6 Example of input segment *LOST*

5.5.8 Neyrtec distributors

Introduction

Neyrtec distributors are special offtake structures, which are able to maintain a nearly constant discharge while the upstream water level varies within a certain range. The Neyrtec distributors act as a combination of weir and orifice structure with a highly variable contraction coefficient. In the computer model standard tables have been included in which the discharge as a function of the upstream water level has been tabulated.

Input

Under the heading:

```
*NERP
+ Name     Branch     X   Type   Xdir    B       Z
+
```

The following data has to be given:

NAME	:	Name of the structure (maximum 10 characters).
BRANCH	:	Branch name in which the structure is located.
X-COOR	:	Position along the x-axis in the branch.
TYPE	:	One of the types of Table 5.5-2 can be selected. (The applied notation is similar to the notation of Neyrtec, which produces the modules).
Xdir	:	PX for positive flow direction. NX for negative flow direction.
B	:	The effective width of the module. The effective width is somewhat less than the total width. Example, a real length of 0.32 m (Type :X1) has an effective length of 0.30 m as the nominal discharge is 0.30 m * 100 l/s/m = 30 l/s.
Z	:	Sill level of the module. In Table 5.5-2 the design heads of the modules are given. Ideally the sill level should be equal to the designed water level minus the design head.

TABLE 5.5.2 Data of neyrtec distributors

Type	Capacity*)	Head**)	Discharges***)				
-	l/s/m	m	min	step	max1	step	max2
X1	100	0.170	0.030	0.030	0.150		
X2	100	0.175	0.030	0.030	0.150		
XX1	200	0.270	0.030	0.030	0.240	0.060	0.480
XX2	200	0.280	0.030	0.030	0.150	0.060	0.480
L1	500	0.500	0.500	0.050	1.500		
L2	500	0.510	0.500	0.050	1.500		
C1	1000	0.790	1.000	0.100	5.000		
C2	1000	0.810	1.000	0.100	5.000		
CC1	2000	1.260	-	-	-		
CC2	2000	1.290	-	-	-		

*) In the second column the capacity of the module in litre per second per meter length is given.

**) In the third column the design head above the sill level is given.

***) In the following three columns the minimum discharge capacity, the increments in discharge capacity and the maximum discharge capacity is given respectively. Only for the XX1 and XX2 gates, the capacity can be further increased to 0.48 in steps of 0.06 m3/s.

An example of the input segment *NERP is given in Fig. 5.5.7

```
*NERP     - Neyrtec Modules
+ Name     Branch     X    Type    Xdir     B           Z
  MOD1     BR1       5.0   XX1     PX      0.90        12.50
  MOD2       BR4      5.0   L1       PX      TT$LMOD2     11.80
-
```

*Fig. 5.5.7 Example of input segment *NERP*

Note Operation of the neyrtec distributors can be simulated by specifying the width of the sill as a tabulated function of time for example.

5.5.9 Fortran defined function

Introduction

If a structure can not be described by one of the standard structures, this structure can still be included in the model, namely as a Fortran defined structure. Before using this facility however, the user is advised to convince himself that it is impossible to make use of one of the standard structure, as nearly all structures can be rewritten as one of the given standard structures. A fortran defined structure is somewhat more complicated than the other structures, as the user has to write a fortran subroutine and link this subroutine to the model. If a fortran defined structure is used the following data have to be given in the model definition.

Input

Under the heading:

```
*FDST
+ NAME        BRANCH    X
```

the user gives on each input line the following data :

NAME	:	name of fortran structure (max 10 characters)
BRANCH	:	the branch in which the structure is positioned
X-COOR	:	Position along the x-axis in the branch.

The input block is closed by line containing the symbol "=" in the first column.

Writing a fortran function

To include a fortran defined function the following procedure has to be followed.

1) Give the required input data as discussed above.
2) Write the fortran function in FORTRAN 77 computer language.
3) Change the source code of Fortran subroutine FSTRUC, located in file MODCOMXF.FOR.
4) Compile both the newly written fortran function and FSTRUC with

a professional Fortran Compiler.

5) Link the whole subsystem MODCOM again.

ad 1) The name given in the input block refers to a Fortran-defined function which has to be written by the user and then must be compiled and linked to the subsystem MODCOM.

ad 2) The fortran program should start with the heading

FUNCTION CULVERT (I,H1,H2)
where the variables represent:
I : structure number
H1 : water level at the upstream grid point.
H2 : water level at the downstream grid point.
Then the body of the program can be written. An example of such a program is given in Fig. 5.5.7

ad 3) Read MODCOMXF.FOR in an editor. In the subprogram FSTRUC it is already indicated where to change the source code. The only thing the user has to do is to change the name of CULVERT in to the new name.

ad 4) Compiling the programs is quit simple. If one is using RMFORT-compiler the commando should read:
RMFORT MODCOMXF /Z <ENTER>
If the Microsoft fortran compiler is used the commando should read:
FL /c MODCOMXF.FOR followed by pressing <ENTER>
If no error messages are given one can continue, otherwise the program(s) should be made error free.

ad 5) Finally the complete program can be linked again: COMLNK <enter> Now the new written program is included in MODCOM.EXE

```
FUNCTION CULVERT (I,H1,H2)
INTEGER I
REAL H1, H2
IF (H2 .GT. H1 ) THEN
     CULVERT = 0.0
ELSE IF (H1 .LE. Z ) THEN
     CULVERT = 0.0
ELSE
     CULVERT = 1.7 * (H1 - Z)**1.66
END IF
RETURN
END
```

Fig. 5.5.9 Example of a fortran structure program

5.6 Automatic control systems

5.6.1 Introduction

In the computer model various automatic controllers are incorporated which can be used to control weir or orifice structures. One can select between three levels of control, namely: local, regional and global control. Furthermore either step- or PID-controllers can be used.
The output signal of a PID-controller reads:

$$u_t = z_0 + K_p e_t + K_i \sum e_t + K_d (e_t - e_{t-1}) \qquad 5.17$$

The output signal of a step controller is computed by:

$$u_t = u_t + v \Delta t \qquad 5.18$$

For regional control, the output signal can also be computed according to CARDD-control or BIVAL-controller. The latter is very much the same as a step controller, whereby the target level is varied according to the following formula:

$$Y_{target} = Y_{target,Q_{max}} + D \frac{Q_{max} - Q_{pivot}}{Q_{max}} \qquad 5.19$$

Where

ut =	output signal
y =	target level
D =	dead band
z0 =	constant (parm2)
Kp =	proportional gain factor
Ki =	integral gain factor
Kd =	differential gain factor
v =	speed of gate movement (meters/minute)
et =	deviation from set-point (= target level - actual level)
Q_{max}=	Maximum discharge at pivot point

For more information about the controllers and the determination of value of the control parameters, please refer to par. 11.3.4. With the help of the automatic control facility various control systems can be implemented in MODIS such as automatic upstream and downstream control, Bival control, constant volume control et cetera. The controller computes every simulation step an output signal based on actual water level measurements. This output signal is used to determine for example a new sill level or a gate opening height in order to maintain a constant level somewhere in the system.

To use the automatic control facility the user has to specify by the definition of the structures, a name for the controller instead of a structure parameter. (Maximum total length of the reference is 10 characters e.g. LA$CONTR1 whereby "LA" indicates Local Automatic control, "$" is a separation sign and CONTR1 is a users specified name). The name "CONTR1" refers to the automatic control input segment which will be discussed here.

The input segment automatic control consists of three input blocks

*AULC	-	automatic local control
*AURC	-	automatic regional control
*AUGC	-	automatic global control

5.6.2 Automatic local control

Automatic local control implies that the water level, either upstream or downstream of the structure is used to control the structure. Under the heading:

```
*AULC - automatic local control
+ NAME   CONTROL  TYPE   PARM 1  PARM 2   PARM 3   PARM 4 ....  PARM 7
```

The following data have to be given:

NAME : Name of the controller (maximum 10 characters).

CONTROL : Control mode which can be either:

STEP Step controller

	PID	Proportional Integral Differential controller
	AVIO	AVIO downstream control gate manufactured by Alsthom Fluides
TYPE :	Type of control which can be either;	
	UC	Upstream control.
	DC	Downstream control (a must for AVIO-control).
	MC	Mixed control (only possible in combination with step control)

If CONTROL is equal to "STEP" the following data are required:

PARM 1 : The initial value of the parameter which is automatically controlled

PARM 2 : The target level to be maintained.

PARM 3 : The allowable variation in meters from the target level.

PARM 4 : The speed of variation in meter per minute (m/m) !.

PARM 5 : The minimum value of the controlled parameter.

PARM 6 : The maximum value of the controlled parameter.

If CONTROL is equal to "PID" the following data is required:

PARM 1 : Initial value of output parameter.

PARM 2 : Target level or setpoint.

PARM 3 : Kp Proportional gain factor.

PARM 4 : Ki Integral gain factor.

PARM 5 : Kd Differential gain factor.

PARM 6 : The maximum speed of variation in meter per minute (m/min).

PARM 7 : The minimum value of the controlled parameter.

PARM 8 : The maximum value of the controlled parameter.

If CONTROL is equal to "AVIO" the following data is required:

PARM 1 : Initial value of output parameter.

PARM 2 : Target level or setpoint.
PARM 3 : Xradial, radius of the gate in cm
PARM 4 : Decrement of the controller in cm
PARM 5 : damp, cross-sectional area of floater opening in cm2
PARM 7 : The minimum value of the controlled parameter.
PARM 8 : The maximum value of the controlled parameter.

If TYPE is equal to "MC" (only possible in combination with step-controller) the following data is required:

PARM 1 : The initial value of the parameter which is automatically controlled
PARM 2 : The lower upstream target level.
PARM 3 : The upper upstream target level.
PARM 4 : The downstream target level.
PARM 5 : The allowable variation in meters from the downstream target level.
PARM 6 : The speed of variation in meter per minute (m/min) !.
PARM 7 : The minimum value of the controlled parameter.
PARM 8 : The maximum value of the controlled parameter.

All parameters (except the initial value of a STEP controller), can be given as a constant value or as a function of other variables using the standard reference convention already discussed in paragraph 5.5.1 (Table 5.6.1). The reference convention consists of four parts.

1) A one character code for the type of the independent variable. H, T, U, et cetera. (Table 5.6.1).
2) A one character code for the function type. The following function types are available: T - Tabulated form
   F - Fortran defined function
3) After the two character codes selected from the list, the symbol "$" must be given.
4) Finally the name of the function must be given. The name is limited up to 7 characters.

TABLE 5.6.1 Function references for automatic control parameters

PART 1	:	Code Description
		T - Time
		H - Water level
		Z - Crest level
		A - Cross sectional Area of opening
		U - Upstream head over the structure
		D - Deviation from set-point (allows for multiple speed)
		O - Opening height of the gate
		Y - Ratio gate opening height/upstream head
PART 2	:	Code Description
		T - Tabulated form
		F - Fortran function
		A - Automatic control
PART 3	:	$ (dollar sign)
PART 4	:	Users defined name, maximum 7 characters

Remark : head is defined as water level minus crest level of sill.

If all control data have been specified this input block is closed by a line with the symbol "=" in the first column. The input segment is closed by a line containing "*END" in the first four columns.

An example of the input segment *AULC is given in Fig. 5.6.1

```
*AULC     AUTOMATIC LOCAL CONTROL
+
+NAME   CONTROL  TYPE  INI VAL  TARG L  DEAD BAND   SPEED   MIN   MAX
 CONTR2  STEP     UC    1.10     10.30   0.02        0.01    0.00  1.50
+
+NAME   CONTROL  TYPE  inival  TARGET L  Kp    Ki    Kd  SPEED  MIN  MAX
 CONTR1  PID      UC    1.20    10.30     3.5   0.5   1.5 0.10   0.0  1.50
+NAME CONTR  TYPE INI   TARG   x (cm)   decr (cm)  damp (cm2) min   max
 CNTR1  AVIO DC   0.28  10.70   28.      1.4        15.       0.0   0.25
-
*END
```

*Fig. 5.6.1   Example of input segment *AULC (automatic local control)*

5.6.3 Automatic regional control

Automatic regional control implies that two (or three) water levels located along the canal, either upstream or downstream of the structure, are used to control a structure. Under the heading:

*AURC - automatic regional control
+NAME CONTROL TYPE PARM 1 PARM 2 PARM 3 PARM 4 .... PARM 7

The following data have to be given:

NAME	:	Name of the controller (maximum 10 characters).
GRD1	:	Name of the first grid point.
GRD2	:	Name of the second grid point.
GRD3	:	Name of the third grid point (Only for CARDD-control).
TYPE	:	Type of controller which can be either: PID Proportional Integral Differential controller BIVAL BIVAL control system
INITVAL	:	Initial value of parameter which is automatically controlled.
TARGETL	:	Target level to be maintained

If TYPE is equal to "BIVAL" the following additional data will be required:

ALPHA	:	value of alpha which determined the location of the pivot point. X(p) = X(grd1) + (1-alpha) (X(grd2)-X(grd1))
QMAX	:	Maximum discharge of the canal reach at pivot point
DEADB	:	Dead band of the step controller. Note that the allowable variation is equal to TARGETL ± 1/2 DEADB !
DECR	:	Decrement of the water level from the target level at the pivot point for zero discharge.
SPEED	:	Speed of control parameter in meter per minute m/m.
MIN	:	Minimum value of control parameter.
MAX	:	Maximum value of control parameter.

If CONTROL is equal to "PID" the following data will be required:

ALPHA	:	value of alpha which determined the location of the pivot point. X(p) = X(grd1) + (1-alpha) (X(grd2)-X(grd1))
Kp	:	Kp Proportional gain factor.
Ki	:	Ki Integral gain factor.
Kd	:	Kd Differential gain factor.
SPEED	:	The maximum variation in meter per minute m/m.
MIN	:	The minimum value of the controlled parameter.
MAX	:	The maximum value of the controlled parameter.

If CONTROL is equal to "CARDD" the following data, apart from the additional grid point, will be required:

INTERVAL	:	The interval (sample period) between which new gate settings are calculated.
SPEED	:	The maximum variation in meter per minute (m/m).
MIN	:	The minimum value of the controlled parameter.
MAX	:	The maximum value of the controlled parameter.

Furthermore, 11 constants have to be specified which are fixed for the total CARDD-system. These constants are defined in a new input block named *CARDD. After this steer-key, the value of the 11 parameters are given in sequence on one input line, see example below.

```
*CARDD
+ C1    C2  C3      C4    C5   C6   C7   C8   C9    C10   C11
  200   60  0.0015  0.01  0.15 1.60 0.03 4.50 0.075 0.005 0.5
=
```

All parameters (except the initial value of a STEP controller), can be given as a constant value or as a function of other variables using the standard reference convention already discussed in paragraph 5.6.1.

If all control data have been specified this input block is closed by a line with the symbol "=" in the first column.

An example of the input segment *AURC is given in Fig. 5.6.2

```
*AURC     AUTOMATIC REGIONAL CONTROL
+NAME  GRD1 GRD2 CONTR INIV TARGETL ALPHA  Kp    Ki  Kd   SPEED MIN  MAX
+
 CNTR1 G12  G21  PID   0.40 111.580  0.67  -1.5 -0.1 -0.0 1.    0.0 1.30
+
+NAME  GRD1 GRD2 CONTR INIV TARGETL ALPHA QMAX DEADB DECR SPEED MIN MAX
+
 CNTR2 G22  G31  BIVAL 0.4  111.580 0.67  15.0 0.04  0.0  0.05  0.0 1.30
-
*END OF ACSDEF
```

*Fig. 5.6.2 Example of input segment *AURC (automatic regional control)*

5.6.4 Automatic global control

Automatic global control implies that any number of water levels scattered along the canal are used to control a structure. The input data required is more than for local or regional control, and for the assessment of various values, additional special calculation programs are required. The main purpose of modelling global control in MODIS is to test the applicability of global control for non-linear conditions.

Under the heading:

*AUGC - automatic global control
+NAME CONTROL INIVAL INTERVAL TARGET SPEED MIN MAX

The following data is specified.

NAME	:	Name of the controller (maximum 10 characters).
CONTROL	:	Control mode which can be : LQG Linear Quadratic Gaussian control
INITVAL	:	Initial value of parameter which is automatically controlled.
TARGETL	:	Target level to be maintained
INTERVAL	:	The interval (sample period) between which new gate settings are calculated.
SPEED	:	The maximum variation in meter per minute (m/m).
MIN	:	The minimum value of the controlled parameter.
MAX	:	The maximum value of the controlled parameter.

The input block is closed by a line with the symbol "=" in the first column.

Thereafter the output variables are specified in an input block named

*GLGR - Global grid points
+ GRD1 GRD2 GRD3 GRD4 GRD5 GRD6 GRD7 GRD8

The grid point names can be specified in any order and do not have to be placed on a single line. In fact, per line not more than 10 grid point names can be given.

Finally, a number of matrices are defined headed by a characteristic steer-key.

The following matrices are required for LQG (Linear Quadratic Gaussian) Control, in which the system description reads:

Open loop system:

$$x(kh+h) = \Phi\, x(kh) + \Gamma u(kh) + v(kh)$$
$$y(k) = C\, x(kh) + e(kh)$$

Observer:

$$\hat{x}(kh+h) = \Phi\, \hat{x}(kh) + \Gamma u(kh) + Ks[y(k) - C\, \hat{x}(kh)]$$

Disturbance model:

$$z(kh+h) = W\, z(kh) + H[y(kh) - y_r(kh)]$$

and a feedback law:

$$u(kh+h) = -K_f\, \hat{x}(kh)$$

in which,

- x = state variable
- $\hat{x}$ = estimator state variable
- h = the sample period and
- y_r = the reference (setpoint)
- k = time level
- u = input variable
- v = state disturbance
- e = measurement noise
- y = output
- z = state variable of disturbance model

The matrices to be defined are:

*PHIM - Φ

*GAMM - Γ

*CMAT - C
*KFMA - Kf
*KSMA - Ks
*WMAT - W
*HMAT - H
*KDMA - Kd (Extra matrix)

If one matrix is not used, it can simply be skipped. The data in the matrixes are just figures separated by one or more blank spaces. There is no limitation of the number of matrix element as long as the array dimensions (specified in \modis\include\matrix.add) are not exceeded.

The input segment is closed by a line containing "*END" in the first four columns.

An example of the input segment *AURC is given in Fig. 5.6.4

5.7 Functions

If no functions are used, this part can be skipped. The input segment Functions consist of three input blocks

*TTIM - tabulated functions of time
*TGFU - tabulated general function
*FDFU - fortran defined functions

The first input block (*TTIM) is reserved for variables of which the values have been defined previously to be a function of time. The second and third input block is for all other functions. Preferable tabulated functions should be used, as for fortran functions, the function has to be written, compiled and linked to the program package, which takes some more time.

5.7.1 Tabulated functions of time

Under the heading

*TTIM - TABULATED FUNCTIONS OF TIME

Following the identification line the input continues with a line of the form:

NAME FUN1 FUN2 FUN3 FUN4 FUN5

The first name should always read "NAME". The names FUN1, FUN2, etc are user defined names. On one line a maximum of 10 functions can be given with values for the independent variable in the first data column and values for dependent variables in successive data columns. The input identification lines for functions can be repeated any number of times in any order to add more functions to the input file. Obviously this freedom holds as long as no array limits for storing the data are exceeded. The first input string (under the header "NAME") is reserved for the function variable time specified in either dhms or second notation.

As a next input item the values are given in table form with on each line the value of the independent variable, followed by the function value(s). The values of the independent variable have to be given in increasing order.

If all control data have been specified this input block is closed by a line with the symbol "=" in the first column.

An example of the input block *TTIM is given in Fig. 5.7.1

```
*TTIM          -   TABULATED FUNCTIONS OF TIME
 NAME            QUPSTREAM      LARQR     B1DIV
 +
  0:00:00:00      70.           5.20      25.50
  0:12:00:00      98.           5.30      25.50
  01:00:00:00    175.           2.40      22.00
  01:12:00:00    324.            -        21.00
  02:00:00:00    341.            -        21.00
  02:12:00:00    270.            -        21.50
  03:00:00:00    212.            -        21.50
  03:12:00:00    164.            -        22.00
  04:00:00:00    120.            -        25.00
 =
```

*Fig. 5.7.1 Example of input block *TTFU (tabulated functions of time)*

As a next input item the values are given in table form with on each line the value of the independent variable, followed by the function value(s). The values of the independent variable have to be given in increasing order.

If all control data have been specified this input block is closed by a line with the symbol "=" in the first column.

An example of the input block *TTIM is given in Fig. 5.7.1

```
*TTIM         -     TABULATED FUNCTIONS OF TIME
 NAME           QUPSTREAM      LARQR     B1DIV
+
 0:00:00:00      70.           5.20      25.50
 0:12:00:00      98.           5.30      25.50
 01:00:00:00    175.           2.40      22.00
 01:12:00:00    324.            =        21.00
 02:00:00:00    341.            =        21.00
 02:12:00:00    270.            =        21.50
 03:00:00:00    212.            =        21.50
 03:12:00:00    164.            =        22.00
 04:00:00:00    120.            =        25.00
=
```

Fig. 5.7.1 *Example of input block *TTFU (tabulated functions of time)*

An example of the input block *TGFU is given in Fig. 5.7.2

```
*TGFU- GENERAL TABULATED FUNCTIONS
+
     NAME          CELL2     AREARES
+      H           AREA      AREA
       8.00        0.0E7     5.E7
      10.00         =         =
      12.00         =         =
      14.00         =        1.E8
=
```

Fig. 5.7.2 *Example of input block *TGFU (Tabulated general functions)*

5.7.3 Fortran defined functions

Instead of a tabulated function, mathematical functions can sometimes be used, for example to simulate the water level in a tidal river with a sinusoidal water level fluctuation. In order to do so, the user can write his own function as a Fortran function and compile and link this function.

Under the header

```
*FDFU  - Fortran defined functions
+
+   NAME
```

The name of the function is specified.

Where
NAME : name of fortran structure (max 10 characters)

If all names have been specified this input block is closed by a line with the symbol "=" in the first column.

The user now has to write a subprogram in Fortran for the function. This is done by reading the file MODCOMXF.FOR in an editor and changing the subprogram FFUN. The new function routine e.g SIN is written in this same file. Then the program is saved, compiled and linked to modcom.exe. (See also paragraph 5.5.9, Fortran structures).

5.8 Initial state

Introduction

The last input segment of the input file for subsystem MODRES is the initial state. For the numerical integration of the equations describing the unsteady flow in an open canal system, an initial state is needed in every grid point. However the user does not need to give initial data in each grid point.

For each branch initial data (water level and discharge) have to be given at least at the end grid points. At grid points between the given points the system will generate values by linear interpolation or calculate them from a given function of x. The system will first set values at the points specified by the user and then fill the gaps by linear interpolation or calculation from the specified functions. This implies that input is not bound to a certain sequence. However, it is advised to group the input per branch and separate these groups by blank comment lines. If the initial state specified by the user is very different from a realistic initial state, the program might face so called non-linear instability problems. These instabilities can be avoided by giving a more accurate definition of the initial state. A simple initial state is to define horizontal water levels with zero discharges.

Input

The input identification line for this item reads as:

```
*INST        DEFINITION OF INITIAL STATE
+
+ BRANCH    GRID POINT    DEFINITION OF H            DEFINITION OF Q
+                         TYPE  VALUE/REFERENCE   TYPE    VALUE/REFERENCE
```

Data are given in the form of data columns. Each input line specifies the initial state at a user-defined grid point.

The data columns are given in the sequence:

NAME : branch name;
GRID POINT: name of the user-defined grid point;
TYPE : data type for H, following the convention:
C for a constant;
T for a tabulated function of X;
F for a Fortran function defined by the user;
VALUE/REF: value of h at the grid point or the function name;
TYPE : data type for Q, following the same convention as given for h;
VALUE/REF : value of Q at the grid point or the function name.

As usual the list of input for initial data is closed again by a line with the "=" symbol in the first column.

As this block forms at the same time a complete segment this line is followed by an *END steer-key line.

An example of input block *INST is given in Fig. 5.8.1

```
*INST          - DEFINITION OF INITIAL STATE
+BRANCH  GRID-POINT    TYPE    VALUE/REF.    TYPE    VALUE/REF.
+                               H             Q
 BR1     GUP           C       26.58         C       70.0
 =       GDIV1         =       22.58         =       =
 BR2     GDIV2         =       20.58         =       34.0
 =       GDIV3         =       15.18         =       =
 =       GPF11         =       15.18         =       0.
 BR3     GPF12         =       =             =       =
  =      GPF31         =       =             =       =
=
*END     - OF INIDEF
```

*Fig. 5.8.1 Example of input segment *INST (initial state)*

6 Model computation

6.1 Introduction

The real computations are performed by the computational subsystem called MODCOM (MODel COMputation). This system reads the model definition, which has been processed by MODDEF, from the model data base, and starts the simulation. The data to control the simulation process have to be supplied on an input file named COMIN.DAT. This input file consists of one segment which is made up of nine input blocks. These input blocks are:

*IDEN - run identification;
*SWIT - run switches;
*PARM - numerical and general physical parameters;
*TIME - time control parameters;
*RSST - system state result data file;
*RTIF - result time function file;
*ROUT - route definitions through net
*BASF - result base functions;
*RPLF - result place functions.

Each input block is described in detail in the following of this chapter.

6.2 Run control parameters

6.2.1 Run identification

The first input block of the input file for the model computation is used to identify the input and output files. Normally the user can leave this input block unchanged.

The identification block starts with the steer-key line:

```
*IDEN -   RUN IDENTIFICATION
+
+    DATASET (FILENAMES)
+
-   <.......>
     INPUT      file name \
     ECHO       file name )-consistent with job control
     OUTPUT     file name /
     MODDB      file name \
     SST FILE   file name )-defined here
     PLF FILE   file name )
     RTF FILE   file name /
```

The codes in capitals have to be copied exactly.

The file names are given by the user and should not start before column 20. Normally standard names are used (see Fig. 6.2.1).

At the end of this input block a line is given with the character '=' in the first column.

```
*IDEN -   RUN IDENTIFICATION
+
+    DATASET (FILENAMES)
+
-   <.......>
     INPUT      COMIN.DAT
     ECHO       COMECH.DAT
     OUTPUT     COMOUT.DAT
     MODDB      USMDB.UNF
     SST FILE   USSST.UNF
     PLF FILE   USPLF.UNF
     RTF FILE   USRTF.UNF
-
```

*Fig. 6.2.1 Example of input block *IDEN (run identification)*

6.2.2 Switches

The switch parameters activate or deactivate data transfer, creation of new files, debug facilities and the printing of their resulting output. These switches have to be given in a specific sequence to indicate the names (these appear in a list on a comment line). On the input line below the comment line, the following characters can be given:

Y	to activate the switch (abbreviation of YES);
N	to deactivate the switch (abbreviation of NO).
#	to set the default

On each line 7 switches are read. Some positions have no meaning yet, as indicated by the heading 'FREE'. Here the default character '#' has to be given.

The switches with their meaning are given below in the sequence in which they appear in the input.

line 1

TEST	some general test output
TESTC	test output computational algorithm
TESTS	test output structures
TESTO	to switch out the computational routine COMSYS and activate dummy computational routine DUMCOM
DEBUG	log file for result data base actions
LOCONTR	writing the computed local control output signals to output file (comout.dat).
RECONTR	writing the computed regional control output signals to output file (comout.dat), but only if LOCONTR = N.

Remark on LOCONTR and RECONTR:
Only one of the switches LOCONTR or RECONTR should be 'Y', and the other switch should be 'N'.

line 2

PLFRDB	activates or deactivates generation of result place functions
TIFRDB	activates or deactivates generation of result time functions
SSTRDB	activates or deactivates generation of system states
STEADY	with this switch the system state file is continuously re-positioned to contain only the last transferred state
3*FREE	three positions reserved for extensions (substitute three times # separated by one or more blanks)

line 3

NEWRPF	with this switch possible existing contents on the file specified in *IDEN is overwritten. Deactivation of this switch leads to extension of the existing file with newly computed data.
NEWRTF	similar definition for the result time function file.
5*FREE	five positions reserved for extensions. (Substitute for each the default character #).

An example of possible switch settings is given in Fig. 6.2.2.

```
*SWIT    RUN-SWITCHES
+meaning of characters :  Y=YES  N=NO  #=DEFAULT  SETTING
+TEST    TESTC    TESTS    TESTO    DEBUG    LOCONTR    RECONTR
 Y       Y        Y        N        Y        Y          N
+PLFRDB  TIFRDB   SSTRDB   STEADY-STATE      <FREE-SWITCH >
 Y       Y        Y         N                #  #  #
+NEWRPF  NEWRTF                              <FREE-SWITCH >
 Y       Y                                   #  #  #  #  #
-
```

*Fig. 6.2.2 Example of possible switch settings for *swit*

6.3 Computational parameters

The computational parameters are given in the input block *PARM identified by:

```
*PARM -   NUMERICAL AND PHYSICAL PARAMETERS
+ PSI         THETA        ITERM      G

+ EPSOFL      EPSUFL       EPSHFL       EPSDH
```

Were:

PSI	is the weighting coefficient for the discretizations in x. Usually a value of 0.5 is given;
THETA	is the weighting coefficient for the discretizations in time. For accurate computations a factor of 0.55 is given. For steady state computations it is advantageous to take a factor close to 1, in order to damp undesired osculations and to converge quickly to the steady state.
ITERM	is the number of iteration steps within one time step. As a rule it does not make sense to take a value different from 2.
G	is the acceleration due to gravity. A value of 9.82 usually is accurate enough.

The second line of computational parameters concern some numerical test values used during the evaluation of structure functions. (When this second input line is skipped the default values given between brackets are used).

EPSOFL	Test value for dh whether linearization is required for overflow structures. Usual value is 0.01
EPSUFL	Test value for linearization for underflow structures. Usual value is 0.01
EPSHFL	Test value for linearization for head loss structures. Usual value is 0.01
EPSDH	Increment value for water level used during numerical differentiation of structures. Usual value is 0.01

The input block is closed by a line containing the symbol "=" in the first column. An example of input block *PARM is given in Fig. 6.3.1

```
*PARM -   NUMERICAL AND PHYSICAL PARAMETERS
+
+ PSI          THETA     ITERM     G
  0.5          0.55      2         9.82
+ EPSOFL       EPSUFL    EPSHFL    EPSDH
  0.01         0.01      0.01      0.01
=
```

*Fig. 6.3.1   Example of input block *PARM (computational parameters)*

6.4 Time control parameters

In the input block for time control parameters values have to be given to control the start and the end time for the computations. Besides the time step and what system state is used to (re)start the system file have to be specified. Data are given under the header:

```
*TIME -  TIME CONTROL PARAMETERS
+
+ BEGIN-TIME    END-TIME     TIME-STEP     SST-TIME
```

Where:

BEGIN TIME is the starting time of the simulation. Like many other time values time is given as either in seconds or a time string in dhms - notation i.e. D:H:M:S, with :

D the number of days (1,2 or 3 digits);
H the number of hours (2 digits);
M the number of minutes (2 digits);
S the number of seconds (2 digits).

END TIME is the end time of the simulation period given in time string format as described above or in seconds. Simulation is terminated as soon as this moment is reached.

TIME STEP specifies the time step to be used by the system. The following options are available:

- a constant time step given as a time string in dhms notation;
- the time step in seconds (upto two digits behind the point).
- a time step given as a tabulated function of time. The specification reads T$<name>. The table is supposed to be specified in input segment FUNCTION of subsystem MODDEF.
- a time step given by a Fortran defined function of time. In this case the specification reads F$<name>. The function <name> has to be called and defined in the interface routine FFUN.

SST-TIME specifies a string informing the system how to read the initial state from the system state file. Possible options are:

INIDEF the system will read the system state from the SST-file, which was defined in INIDEF (input segment of MODDEF);

PERM the system will try to read an initial state for the SST file which was previously written by system MODCOM during a steady state run with the steady state switch on Y(es). Beware that the system generated steady state will be destroyed if MODDEF is run with the INIDEF-switch on Y. So to avoid this overwritting of the system generated steady state, the INIDEF-switch should be set on N, after a steady state has been made.

BEGIN the system will read the system state from the SST-file with the last time level <= BEGIN-TIME.

Some additional explanation has to be given on the time step definition in general. During input processing the system generates moments in time where during the simulation the process is forced to arrive at a computational time level. This is used for instance, to generate result place functions at a specific moment in time. Due to this algorithm the time step really used can be somewhat smaller then the one specified by the user. The time step actually used is shown in the system log on the echo file.

An example of input block *TIME is depicted in Fig. 6.4.1

```
*TIME      TIME CONTROL PARAMETERS
+BEGIN-TIME   END-TIME      TIME-STEP     SST-USAGE
 00:00:00:00  01:03:00:00   T$DELTA-T     INIDEF
-
```

*Fig. 6.4.1 Example of input block *TIME (run time specification)*

6.5 Output to the system state file

Time intervals have to be specified in order to write results of the computations to the System State Result Data file (SST-file). The system state result data file contains a complete description specified in h and Q of the system at a certain moment in time. This system state can be used later to restart a computation. The parameters to control the system state file actions have to be given in the input block headed by the identification line:

```
*RSST     SYSTEM STATE RESULT DATA FILE
+
+         BEGIN          END  INTERVAL
```

Where

BEGIN — a time value after which during the simulation it is checked for every time step whether the system has to be stored or not. Possible specifications are:

- B — Checking starts immediately at the beginning of the simulation. This is also the default built-in value;
- <time> — a time string in dhms notation, or in seconds.

END — a time value after which checking can be stopped and no data is stored in the file. Possible specifications are:

- E — Checking continues until the end of the simulation. This is also the default built-in value;
- <time> — a time string in dhms notation or in seconds.

INTERVAL — For this time interval the following options are defined:

- N — The corresponding file facility is not used. In fact it inactivates the input block;
- <no.>* — a file action is performed with an interval of <no.> time steps (e.g. 10*);
- <time> — a time string in dhms-notation, giving a constant interval. It is also possible to specify this as C$<time> time is given in seconds;

T$<name> a tabulated function with name <name> and time as independent variable. Function values are expected in the input for the subsystem MODDEF;

F$<name> a Fortran defined function with name <name> and based on time as independent variable;

\# use the default value, set to 1* (each time step);

Each input block is closed as usual by a line containing the symbol "=" in the first column. An example of the input blocks *RSST is given in Fig. 6.5.1.

```
*RSST     SYSTEM STATE RESULT DATA FILE
+
+BEGIN    END            INTERVAL
 B        00:12:00:00    20*
=
```

*Fig. 6.5.1 Example of input blocks *RSST*

6.6 Output as a function of time

In this input block it is defined how certain flow and geometrical variables computed by the model (normally h and Q) are written to a file as a function of time at selected user-defined grid-points.

The input block is headed by :

```
*RTIF   -      SPECIFICATION RESULT TIME FUNCTION FILE
+ BEGIN        END            INTERVAL
+ TITLE        GRID-POINT          VAR-SPEC     FUNCTION-TEXT
```

Where

BEGIN — a time value after which during the simulation it is checked for every time step whether data has to be stored or not. Possible specifications are:

- B — Checking starts immediately at the beginning of the simulation. This is also the default built-in value;
- <time> — a time string in dhms notation, or in seconds.

END — a time value after which checking can be stopped and no data is stored on the file. Possible specifications are:

- E — Checking continues until the end of the simulation. This is also the default built-in value;
- <time> — a time string in dhms notation, or in seconds.

INTERVAL — For this time interval the following options are defined:

- N — The corresponding file facility is not used. In fact it inactivates the input block;
- <no.>* — a file action is performed with an interval of <no.> time steps (e.g. 10*);
- <time> — a time string in dhms-notation, giving a constant interval. It is also possible to specify this as C$<time>: time in seconds;
- T$<name>: — a tabulated function with name <name> and time as independent variable. Function

values are expected in the input for the subsystem MODDEF;

F$<name>: a Fortran defined function with name <name> and based on time as independent variable;

: use the default value, set to 1* (= each time step);

The rest of all *RTIF date lines contain data items which define one result function per line. The specification of functions is given on successive input lines where the following input is expected:

TITLE	a function title of maximum 10 characters. This function title will be used later for all output presentations.
GRID POINT	the name of the user-defined grid point where the result function should be retrieved.
VAR-SPEC	a specification of the variable according to the syntax described on the next page. The most common definitions are HN and QN which retrieve the water level and discharge respectively at the end of the time step.
FUNCT.TEXT	a free text of maximum 40 characters. This can be used to give a brief description of the function. The text will be reproduced in the function summary later on in the output.

The input block is closed by a line containing the symbol "=" in the first column.

Variable specification

The results which the user wants produced on output in the form of time or place functions can be of very different nature. Therefore, the option has been built-in to address many variables through a name string given in the input. Standard strings are:

HO	water level at the beginning of the actual time step;
HN	water level at the end of the actual time step;
HM	the average of HO and H*; with H* the water level computed in the one but last iteration step;
QO, QN, QM	similar definitions for the discharge;

A	the wetted area of the cross-section;
BF	the flow width of the cross-section;
BS	the storage width;
BSQ	the Boussinesq value used;
DM	the water depth
Z	the bottom level
RA	the hydraulic radius;
RC	the resistance coefficient;
FRICM	the total resistance;
FROUDS	the resulting Froude number.

All the water level dependent coefficients have been computed just before the last iteration step, based on HM. Besides the standard result values, it is also possible to compute any result values by specification through the general function system. As independent variable used in the function call the grid point number j is used. The options are:

T$<name>	referring to a tabulated function <name>.
F$<name>	referring to a Fortran defined function <name>.

The last option can be quite useful in special cases.

Note
A function named TIME containing the time base is generated automatically when result time functions are transformed into GRF format by subsystem RDBUTL.

An example of the input block *RTIF is given in Fig. 6.6.1

```
*RTIF     CONTROL PARAMETERS FOR RESULT-TIME FUNCTION FILE
+
+BEGIN          END  INTERVAL
  B             E         #
+CODE           GRIDPOINT       FUN.SPEC   FUNCTION TEXT (40 CHARACTERS)
 H-G11          G11             HN         head of the canal
 H-G12          G12             HN
 H-G21          G21             HN
 H-G22          G22             HN
 H-G31          G31             HN
 H-G32          G32             HN
-
```

*Fig. 6.6.1   Example of input block *RTIF (result time function file)*

6.7 Output as a function of place

Instead of output results as a function of time, it is also possible to define output as a function of place. If this is not desired, this paragraph can be skipped. (Remember to close the input segment with *END)

If output as a function of place is desired three input blocks have to be used in sequence.

- At first it is needed to define so called "routes". A route is a path defined by a sequence of nodes along which output can be produced.
- Thereafter "base result" functions need to be specified which define an axis along a route, e.g. x-coordinated, names of grid points.
- Finally the variables which are given along the routes as a function of the specified axis, are defined.

6.7.1 Definition of routes

Under the heading:

```
*ROUT
+
+    ROUTE     NODES
+ ......     ......   .......     ........     ........
```

The following data are required:

ROUTE — user defined name of a route, following the standard name convention.

NODES — name of node, already defined in the input file of the model definition (see previous chapter)

The input block is closed by a line containing the symbol "=" in the first column. An example of input block *ROUT is depicted in Fig. 6.7.1.

```
*ROUT
+
+    ROUTE     NODES
+ ......  ......    .......        .........     .........
     ROUT1     N1        N2           N4
     ROUT2     N3        NS4          NS5           NS7
=
```

Fig. 6.7.1 *Example of input block *ROUT (definition of routes)*

6.7.2 Base function

Base functions are defined as functions invariable with time which can be used as abcs in the output presentation. Examples are x-coordinates, grid point numbers, and grid point names in the form of a sequence of character strings along a branch. The base functions are not automatically available on the result data base, so they have to be specified in this input file. As a result they are written to the RPF-file at the beginning of the simulation. The specifications have to be given in the input block which is headed by the identification line:

```
*BASF   -     SPECIFICATION OF BASE RESULT FUNCTIONS
+
+ TITLE     ROUTE       VAR-SPEC     FUNCTION-TEXT
```

Whereafter the following input data are expected:

TITLE — a general result function title of up to 10 characters to be used during output production.

ROUTE — the route name where the base function has to be taken.

VAR-SPEC — a string which defines the function type and values to be retrieved. Possible options are:

- GRIDNO — an integer function is created containing all internal grid numbers of the branch;
- NAMES — a character function is created containing all

grid point names of the branch;

X a real function is created containing all grid point coordinates;

T$<name> a reference to a tabulated function.

F$<name> a Fortran defined function reference. (In the fortran function, the grid point numbers J are used as the independent variables, see paragraph User-defined Fortran Routines.)

FUNCT-TEXT a free descriptive text of maximum 40 characters.

The input block is closed by a line containing the symbol "=" in the first column. An example of input block *BASF is depicted in Fig. 6.7.2.

```
*BASF  CONTROL PARAMETERS BASE-RESULT FUNCTIONS
+    CODE            ROUTE    FUN.SPEC FUNCTION TEXT (40 CHARACTERS)
+
     N-ROUTE1   R$ROUT1  NAMES        ROUTE1: N3-1, N3, N4, N4-1
     N-ROUTE2   R$ROUT2    =          ROUTE2: N3-1, N3, N2, N2-1
     X-ROUTE1   R$ROUT1    X          ROUTE1: N3-1, N3, N4, N4-1
     X-ROUTE2   R$ROUT2    =          ROUTE2: N3-1, N3, N2, N2-1
=
```

*Fig. 6.7.2 Example of input block *BASF (Base result functions)*

6.7.3 Result Place function

The result place function specifications have to be given in a *RPLF block. These functions are defined as a series of model result values at a certain moment in time at all grid points along a branch.

The block heading has to be given as:

```
*RPLF -    SPECIFICATION OF RESULT PLACE FUNCTIONS
+  RESET-FUNCTION
+  TITLE    BRANCH    VAR-SPEC    TIME    FUNCTION-TEXT
```

The first input is related to possible actions on the (possible) existing result place function file. Where the restart facility is used it may be necessary to remove part of the existing result place function files and overwrite them with newly computed functions. If the item on the first input line is set to <name> the system will remove all the functions from the specified function name onward before writing new results to this file.

The following input data should be given on the subsequent line:

RESET-FUNCTION:	<name>	name of the function from where on all functions have to be removed from the file.
	N	if this is not necessary or if the file is still empty;

and where on subsequent lines the following information is given:

TITLE	a function title of up to 10 characters;
BRANCH	the name of the branch where the result place function is generated;
VAR-SPEC	specification of a string indicating the variable name as described in the previous paragraph e.g. HN to retrieve the water levels at the end of the time step;
TIME	an indication of the time where the result place function should be produced. Options are: B at the beginning of the computation; E at the end of the computations. <time> at the time specified by the time string in the usual dhms-notation.
FUNCTION-TEXT	a text of up to 40 characters which is used as a comment presented on output of the functions.

Beware that if the time is specified in dhms notation this, will effect the selection of time steps during the computations, as the system will arrange a computational time level at exactly this time. In the computation of the required time step the previous defined time step will never be exceeded, however.

The list of functions is terminated with the usual line indicating the end of the block ('=' character in first column). As this input block is the last input block of the input segment the segment is closed by a line containing the string *END in the first columns. An example of the input block RPLF is given in Fig. 6.7.3.

```
*RPLF     CONTROL  PARAMETERS  RESULT  PLACE  FUNCTION
+
+    RESET   FUNCTION
       N
+    CODE          BRANCH    FUN.SPEC TIME MOMENT    FREE-TEXT
+
     H-RT1-1   R$ROUT1     HN          000:04:00:00  H-ROUT1: 04:00:00
     H-RT2-1   R$ROUT2     =                =        H-ROUT2: 04:00:00
     Q-RT1-1   R$ROUT1     QN               =        Q-ROUT1: 04:00:00
     Q-RT2-1   R$ROUT2     =                =        Q-ROUT2: 04:00:00
     H-RT1-2   R$ROUT1     HN          000:04:30:00  H-ROUT1: 04:30:00
     H-RT2-2   R$ROUT2     =                =        H-ROUT2: 04:30:00
     Q-RT1-2   R$ROUT1     QN               =        Q-ROUT1: 04:30:00
     Q-RT1-2   R$ROUT2     =                =        Q-ROUT2: 04:30:00
     H-RT1-3   R$ROUT1     HN          000:05:00:00  H-ROUT1: 05:00:00
     Q-RT1-3   R$ROUT1     QN               =        Q-ROUT1: 05:00:00
     Q-RT2-3   R$ROUT2     =                =        Q-ROUT2: 05:00:00
     H-RT1-4   R$ROUT1     HN          000:05:30:00  H-ROUT1: 05:30:00
     H-RT2-4   R$ROUT2     =                =        H-ROUT2: 05:30:00
     Q-RT1-4   R$ROUT1     QN               =        Q-ROUT1: 05:30:00
     Q-RT2-4   R$ROUT2     =                =        Q-ROUT2: 05:30:00
     H-RT1-5   R$ROUT1     HN          000:11:00:00  H-ROUT1: 11:00:00
     H-RT2-5   R$ROUT2     =                =        H-ROUT2: 11:00:00
     Q-RT1-5   R$ROUT1     QN               =        Q-ROUT1: 11:00:00
     Q-RT2-5   R$ROUT2     =                =        Q-ROUT2: 11:00:00
=
*END
```

*Fig. 6.7.3   Example of input block *RPLF (result place function)*

7 Result data base

7.1 Introduction

In a Result Data Base, the results of the computations performed by subsystem MODEL COMPUTATION are stored. The results can be presented in different ways such as tables, graphs and performance parameters (see following chapters). However, the result time functions in the result data base are not in a format directly accessible by the model results subprograms (MODRES and MODPLT). The reason is that during the computations the result time functions are stored sequentially so that for each time step the time value is followed by results at different grid points. Before these data can be further processed the sequence has to be transformed, to give values at the different points as time sequences. This transformation into a so called GRF-file format is performed by the subsystem Result Data Base Utility (RDBUTL).

Another function of the RDBUTL subsystem is to investigate the contents of a result data base. The type of information given depends on the file which is investigated.

The input to the subsystem RDBUTL is given in an input file named UTLIN.DAT and has been organised in one segment of three blocks:

- *FILS giving information on file names and setting switches for selecting the actions with the sybsystem;
- *TRAFO transforming result time functions to the format required for use in MODRES;
- *CONT giving information on the contents of the result data base files (SST-file, RTF-file and the five GRF-files).

In the following of this chapter each input block will be discussed in detail.

Tip In most cases the content of the input file UTLIN.DAT is not altered. In that case one can copy an already existing input file named UTLIN.DAT from a previous project to its working directory and run RESULT DBASE.

7.2 Identification of files

The first input block is identified by the line:

```
*FILS -    SPECIFICATION OF FILES USED
```

On successive lines names have to be given for the files used in this subsystem. The form is:

```
-    .........^col no  2    ^col no  20
     INPUT                file name )
     ECHO                 file name )   consistent with job control
     OUTPUT               file name )

     RDB-SST              file name )
     RDB-RTF              file name )
     GR-FILE1             file name )
     GR-FILE2             file name )   defined here
     GR-FILE3             file name )
     GR-FILE4             file name )
```

Remark :

- The codes shown in capitals have to be copied exactly.
- The file names specified by the user should not start before column 20.

The input, echo and output file, again, are fixed in the system and can only be modified by system dependent job control. The other files are assigned dynamically by the program. All files have to exist already, if used, except the GRF-file where the transformed result time functions are stored. If a file is not used by the system, one can specify N for the file names. No assignment will be performed then.

The last data line, of this input block is headed by the comment line:

```
+    TRAFO ACTION        CONT ACTION
```

In the block, the characters Y and N below the keywords, indicate whether the following actions have to be performed or not:

TRAFO ACTION transformation of result time functions to GRF format;

CONT ACTION investigate the contents of the result data base.

If one of these switches has been set to N (No) a possible input for this specific action, given under the following input blocks (*TRAFO and or *CONT) will be neglected. As usual the block is closed with a line containing '=' as a first character. An example of input block *FILS is depicted in Fig. 7.2.1.

```
*FILS  SPECIFICATION OF USED DATASETS
+
     INPUT        UTLIN.DAT
     ECHO         UTLECH.DAT
     OUTPUT       UTLOUT.DAT
     RDB-SST      USSST.UNF
     RDB-RTF      USRTF.UNF
+
+  RDB-FILES IN GENERAL RESULTFUNCTION FORMAT
     GR-FILE1     USPLF.UNF
     GR-FILE2       N
     GR-FILE3       N
     GR-FILE4       N
     GR-FILE5     USTIF.UNF
+
+  SELECT SWITCHES FOR UTL-ACTIONS
+     TRAFO-ACTION         CONT-ACTION
          Y                    Y
+
=
```

*Fig. 7.2.1 Example of input block *FILS (Identification of files)*

7.3 Transformation of time function

The second input block is headed by the identification line:

```
*TRAFO    -    SPECIFICATION OF RESULT DATA BASE TRANSFORMATION
+  T-BEGIN      T-END       (OF RTF-FILE)
```

Where:

T-BEGIN specifies the starting time for the transformation of result time functions. Options are:

- -DHMS time string in dhms-notation;
- -B to indicate that transformation starts at the beginning of the functions.

T-END specifies the end time for the transformation of result time functions with the options:

- -DHMS time string in dhms notation;
- -E to indicate the end of the functions.

Remark : normally the characters B and E are used.

On the next line, headed by the comment:

```
+     FUNCTIONS TO BE TRANSFORMED
```

a list of function names is given.
Each name is given on a separate line. Also one line with code ALFUN can be given to indicate that the complete RTF-file has to be transformed.

The input block is closed by a line containing the termination symbol '=' as a first character. An example of this input block is given in Fig. 7.3.1.

```
*TRAFO    SPECIFICATION OF TIME RANGE FOR  RESULTDATABASE TRANSFORMATION
+           T-BEGIN         T-END
              B              E
+
+     FUNCTIONS TO BE TRANSFORMED    (ALFUN :  for all functions)
+
       ALFUN
-
```

Fig. 7.3.1 *Example of input block *TRAFO (Transormation of place functions)*

7.4 Investigation of content

The third input block, contains switches in the usual way, for investigating the contents of the SST and RTF file and the five GRF files. The input block is identified by:

```
*CONT -  INVESTIGATE CONTENTS OF RESULT DATA BASE
+    TIME FUNCTIONS
+    SST  RTF
```

Where for each file the character Y (yes) or N (no) can be given indicating wheter or not to check the contents of the specific file.

The input block is followed by the following comment lines:

```
+    PLACE-FUNCTIONS
+    GRF-1     GRF-2     GRF-3     GRF-4     GRF-5
```

Again, for each GRF-# file the character Y (yes) or N (no) can be given indicating wheter or not to check the content of the specific file. The input segment is closed by a line containing the character string *END in the first columns. An example of the input block *CONT is shown in Fig. 7.4.1

```
*CONT          INVESTIGATE CONTENTS OF RESULT-DATABASE
+    TIME FUNCTIONS
+    SST  RTF
      Y    Y
+    PLACE-FUNCTIONS
+    GRF-1     GRF-2     GRF-3     GRF-4     GRF-5
      Y         N         N         N         Y
=
*END
```

*Fig. 7.4.1 Example of input block *CONT (content check)*

8 Model results, tables

8.1 Introduction

A selective access to the computed results can be made with the help of subsystem MODRES. In this way only the results which are of direct interest to the user are printed or shown on the screen. A first selection is made by the user in the subsystem MODCOM. There it is specified which computed results and other data are to be written to the result data base, either in the form of functions of time or as functions of place. In this subsystem (MODRES) a further selection is made and the form is specified in which the functions are to be produced on print output.

The user can select for each set of functions the page format by giving the number of characters per line and the number of lines per page. In this way the output can be tailored to any output medium.

For each function some general information can be given, such as variable units, etc. as well as the format in which the data are to be printed. Functions are printed in vertical columns of user-specified width. The number of functions that can be printed in parallel is calculated from the page width and the number of characters needed for each individual function. For each set of functions printed access can be made to 5 different GRF-files (general result files, which are the files with functions arranged in the format required for subsystem MODRES or MODPLT).

This enables the user to compare in one printout, for example, measured results with results from computations. One function can be printed several times. This is particularly useful when in the first column, for example, time, x, grid point numbers or grid point names are displayed, against which the result function values are set in the subsequent columns.

Except for time, these values are not automatically available on the GRF-files. They have to be produced in MODCOM in the same way as, for example, water levels and discharges are produced (see par. 8.9).

Some typical forms of output that can be designed with MODRES are:

- printing the results at different grid points as a function of time;
- comparing at one grid point the measured results with the results of different calibration runs;
- printing water levels at different points along a branch against the grid point names;
- producing for each branch a table of grid point names with x-values and grid point numbers.
- Producing for each grid point operation performance parameters. These parameters are the delivery performance ratio and the operation efficiency. The delivery performance ratio expresses to which extend the actual water delivery has met the intended water delivery during a certain period of time. The operation efficiency specifies which part of the actual supply has been given effectively.

The same kind of functions can be represented on graphical output by means of sub-system MODPLT which produces graphs and plots.

Place functions are produced per route. In the output it is possible to concatenate 5 different functions. This enables the user to print results of up to five routes in sequence in one table.

The input of subsystem MODRES, is given in the input file named RESIN.DAT. This file has been organized in one segment of three blocks:

*PRI - giving names of files to be used and paper format specification;

*BLK - giving the specifications of each set of functions to be printed.

*OPP - giving the specifications of grid points of which Operation Performance Parameters have to be determined.

The blocks *BLK and *OPP can be repeated an unlimited number of times or simply skipped if not used.

8.2 Identification

The first input block is headed by the identification line:

*PRI - FILES AND PAGE FORMAT SPECIFICATION

The first part of this block consists of a specification of file names in the following form:

INPUT	file name)	
ECHO	file name)	consistent with job control
OUTPUT	file name)	
GR-FILE1	file name)	
GR-FILE2	file name)	
GR-FILE3	file name)	defined here
GR-FILE4	file name)	
GR-FILE5	file name)	

Remark:

- The codes shown in capitals again, have to be copied exactly. For the file names the same applies as in the file specification for the previous subsystems. The character N is given again when the specified file is not used.
- The codes shown in capitals should be located within column 2 -20, while the user defined filenames are located after column 20.

The page format specification is given under the comment heading:

+ CHARACTERS/LINE LINES/PAGE

Where:

CHARACTERS/LINE the maximum number of characters that can be printed on a line by the program. Usually it is 80 for screen display and 132 for printers. The system checks for each set of functions the required number of characters by adding up, for each function, the maximum of printing format and function title. When the calculated value exceeds the one specified, an error message will be produced;

LINES/PAGE the maximum number of lines per page,usually 60 or 64. The system will automatically skip to the next page as soon as this number of lines has been printed.

The last data of this input block are given under the comment line

\+ SUMMARY OUTPUT ONLY ? (Y/N)

where either Y(es) or N(o) can be given.

When Y is specified, only summary output is given, consisting of a title page and a function summary of all the functions of the set, containing the following characteristics per function:

- function column number;
- function title;
- function text of 40 characters;
- the specified unit (e.g. m3/s);
- the maximum and minimum values of the function;
- the number of elements in the function;

When N is given also the function tables are printed, where each function specified is presented in a column, headed by the function title.

The input for block *PRI is terminated again by giving the character '=' in the first column of a closing line.
An example of input block *PRI is given in Fig. 8.2.1.

```
*PRI   SPECIFICATION OF FILES AND PAPER-FORMAT
+      USED DATA SETS
       INPUT          RESIN.DAT
       ECHO           RESECH.DAT
       OUTPUT         RESOUT.DAT
       GR-FILE1       USPLF.UNF
       GR-FILE2          N
       GR-FILE3          N
       GR-FILE4          N
       GR-FILE5       USTIF.UNF
+
+     CHARACTERS/LINE   LINES/PAGE
          132               60
+
+     SUMMARY OUTPUT ONLY (Y/N)
                N
+
-
```

*Fig. 8.2.1 Example of input block *PRI (Identification of files and print format)*

8.3 Specification of tabulated output

Input is continued by giving one or more specifications for sets of functions to be printed, each preceded by the identification line:

*BLK - SPECIFICATION OF AN OUTPUT SET
+
+ BLOCK-TITLE

a string of 40 characters can be given describing a general characteristic of the set of functions. This title will be reproduced on the output file.

Function specifications are given following the comment line:

+ COLNO GR-FILE FORMAT UNITS F F F F F

Where:

COLNO — The sequential column number where the results will be printed. Usually they are in sequence of the data lines, though this is not necessary.

GRF-FILE — The GRF file number where the function should be read from. The number can vary from 1 to 5, assuming that the corresponding file has been assigned in the *PRI-block.

FORMAT — The format to be used during printing of the result values. This is given in standard FORTRAN notation. The format depends on the function type. Usually the default value can be specified. The system then assumes as formats A, I4 or F8.2 respectively for the variable types character, integer and real.

UNITS — A character string can be given here representing the units of the values, e.g. m^3/s.

F — Up to 5 function titles can be given for each column. During printing they will be concatenated and presented as one single function column. In this way it is possible to specify several place functions in different branches as a route through the channel network.

If during processing the *BLK-input block, a fatal error is encountered output production stops, also for the block following that specific input block. Each block is terminated again with a closing line with the "=" symbol in the first column. An example of input block *BLK is given in Fig. 8.3.1.

```
*BLK    SPECIFICATION OF PRINT OUTPUT SET
+
+       BLOCK TITLE
        WATER LEVELS
+
+
+COLNO GR-FILE FORMAT  UNITS    FUNCTION  FUNCTION  FUNCTION  FUNCTION
   1      5      #    D:H:M:S   TIME
   2      =      #    ...M..    H-G10
   3      =      #      =       H-G11
   4      =      #      =       H-G12
   5      =      #      =       H-G13
   6      =      #      =       H-G14
-   7      =      #      =       H-G15
-   8      =      #      =       H-G16
-   9      =      #      =       H-G17
-  10      =      #      =       H-G18
=
```

*Fig. 8.3.1 Example of input block *BLK*

8.4 Operation performance parameters

Input can be continued by giving one or more specifications of the grid points of which Operation Performance Parameters have to be calculated. Not only the operation performance parameters at the individual gridpoint can be calculated, but also operation performance parameters of the total system can be calculated.

■ Single offtakes

For the evaluation of the simulated water distribution to an individual offtake, two operation performance parameters are defined. The first one is called the Delivery Performance Ratio (DPR) and specifies the extent to which an offtake receives its intended supply. The second one is called the operation efficiency (e_o) and specifies the amount of water lost by inappropriate allocation of the water to an offtake. In formulae these parameters read:

$$DPR = \frac{V_e}{V_i} * 100\% \qquad 8.1$$

$$e_o = \frac{V_e}{V_a} * 100\% \qquad 8.2$$

Where:

DPR = Delivery performance ratio (-)

e_o = Operation efficiency (-)

V_e = Volume effectively delivered (m^3)

V_i = Volume intended to be delivered (m^3)

V_a = Volume actually delivered (m^3)

As can be read from the presented formulae, three volumes of water are distinguished: An intended volume to be supplied, an actual volume supplied, and an effective volume supplied.

■ Complete system

To evaluate the operational performance of a canal system including all its offtakes, overall performance parameters are needed. The overall Delivery Performance Ratio is calculated by a weighed average of the DPR of the individual offtakes and reads:

$$DPR_{overall} = \frac{\sum_{n=1}^{n=p} V_{e,n}}{\sum_{n=1}^{n=p} V_{i,n}} = \frac{\sum_{n=1}^{n=p} DPR_n \; V_{i,n}}{\sum_{n=1}^{n=p} V_{i,n}} \qquad 8.3$$

Where $DPR_{overall}$ is the overall Delivery Performance Ratio, $V_{e,n}$ is the effective volume received by offtake n (m^3), $V_{i,n}$ is the intended volume to be received by offtake n (m^3), and p is the number of offtakes involved.

The overall operation efficiency incorporates the operational losses of the individual offtakes, the losses due to filling up the canal to its operational level, and the leakage losses and or spill losses if these occur. In formula it reads,

$$e_{0,overall} = \frac{\sum_{n=1}^{n=p} V_{e,n}}{V_{a,intake}} \qquad 8.4$$

Where $e_{0,overall}$ is the overall operation efficiency, $V_{a,intake}$ is the actual intake volume at the head of the main canal (m^3), $V_{e,n}$ is the effective volume received by offtake n (m^3), and p is the number of offtakes involved.

The overall delivery performance ratio becomes equal to the overall operation efficiency if the sum of intended volumes is equal to the volume taken in. In that case the overall performance of the canal system can be characterized by one single figure.

□ Seepage losses

Although seepage losses will always occur in this study no special attention has been paid to it. Nonetheless it is very well possible to incorporate seepage losses in the model by extracting certain amounts of water at certain locations. The losses due to seepage can be accounted for by the overall operation efficiency only.

For a more detailed discussion of the operation performance parameters, reference is made to chapter 11.

Input

The identification line is read:

*OPP
+ Block Title

A name of the output block can be specified up to 40 characters long. This name is copied to the header of the output file.

Function specifications are given following the comment line:

+ LINNO GR-file Name Tbegin Tend Qtarget Qmax Qmin

Where

LINNO	Line number usually starting at 1.
GR-FILE	The GR-file number where the function should be read from. Usually number 5 is chosen, as in this file water levels and discharges are stored.
NAME	Name of grid point of which the OPP must be calculated. The specified name of the grid point must already be defined in COMIN.DAT, otherwise no data have been stored of that grid point.
TBEGIN	Starting time of determination of the OPP. TBEGIN must be given as a time string in d:h:m:s notation. When a "B" is given the calculation of the OPP starts with the first time specified in subsystem MODCOM. When a "=" character is given the Btime is set equal to Btime of the above input line.
TEND	End time is the time specifying the end of the calculation of the OPP. Similar with TBEGIN, TEND must be given either in d:h:m:s notation or with the character E. When "E" is given TEND is equal to the last time of the GR-File. When a "=" character is given Tend is set equal to Tend of the previous input line.
QTARGET	Qtarget is the target or intended value of the discharge. The model compares the actual delivered discharge with the

intended discharge to calculate the operation performance parameters.

Qtmax — Maximum acceptable flow rate. Qtmax can be given either as a figure or as a percentage of the offset of the target discharge. In the latter case the percentage sign "%" must be placed directly after the digits. So, if 25% is specified, the maximum allowable discharge is equal to: Qtarget + 25% of Qtarget. If the actual discharge exceeds the maximum discharge the difference between the actual and maximum acceptable discharge is considered as a loss.

Qtmin — Minimum acceptable discharge. The same notations can be used as for Qtmax, either a percentage or a number. So, if 25% is specified, the minimum allowable discharge is equal to: Qtarget - 25% of Qtarget. If the actual delivered discharge drops below the minimum discharge the actual delivered discharge will be considered to be a discharge loss.

Total performance

It is also possible to evaluate the total performance of the canal system. In that case, the individual offtakes have to be specified first in the input block OPP. The last line of the input block *OPP should then read:

```
+ LINNO   GR-file   Name   Tbegin   Tend   Total
```

Where LINNO, GR-file, Name, Tbegin and Tend remain the same.
Under Total the word TOTAL should be given.
The Name to be specified is the name of the grid-point where the discharge is taken in. This grid-point is used to calculate the $V_{a,intake}$ of Eq. (8.4). All the proceeding points are incorporated in the computation of the overall performance.

The input block is closed as usual by a line containing the "=" character in the first column.

An example of the input block "*OPP" is given in Fig. 8.4.1.

```
*OPP
+ Block Title
  Operation Performance Ratio's
+ LINNO   GR-file   Name    Tbegin  Tend          Qtarget   Qtmax   Qtmin
 1        5         Q-G11   B       00:10:00:00   1.5       2.0     40%
 2        -         Q-G12   B       -             1.5       25%     -
 3        -         Q-G15   B       -             1.5       1.95    0.75
 4        -         Q-G10   B       -             TOTAL
-
*END
```

*Fig. 8.4.1 Example of input format of block *OPP (operation performance parameters)*

8.5 Maximun mean and minimum values

Input can be continued by giving one or more specifications of the grid points at which other operational performance parameters have to be calculated than the DPR and eo. These other parameters were originally used to evaluate water levels rather than discharges but discharges can be evaluated as well. The parameters which are calculated are the maximum, minimum and mean values during a certain period of time. Furthermore, the percentage of time during which a minimum level has been exceeded, and the percentage of time during which a maximum level has been exceeded can be presented.

Input

The identification line is read:

*MMM
+ Block Title

A name of the output block can be specified up to 40 characters long. This name is copied to the header of the output file.

Function specifications are given following the comment line:

+ LINNO GR-file Name Tbegin Tend Htarget Hmax Hmin

Where

LINNO	Line number usually starting at 1.
GR-FILE	The GR-file number where the function should be read from. Usually number 5 is chosen, as in this file water levels and discharges are stored.
NAME	Name of grid point of which the OPP must be calculated. The specified name of the grid point must already be defined in COMIN.DAT, otherwise no data have been stored of that grid point.
TBEGIN	Starting time of determination of the OPP. TBEGIN must be

	given as a time string in d:h:m:s notation. When a "B" is given the calculation of the OPP starts with the first time specified in subsystem MODCOM. When a "=" character is given the Btime is set equal to Btime of the above input line.
TEND	End time is the time specifying the end of the calculation of the OPP. Similar with TBEGIN, TEND must be given either in d:h:m:s notation or with the character E. When "E" is given TEND is equal to the last time of the GR-File. When a "=" character is given Tend is set equal to Tend of the previous input line.
HTARGET	Htarget is the target or intended value of the discharge. The model only copies this target level to the output file in order to facilitate easy comparison with the mean value.
Htmax	Maximum acceptable water level. The total time during which the actual level exceeds the maximum level is monitored. In the output file this time divided by the specified period of time is presented as a percentage
Htmin	Minimum acceptable water level. If the actual water level drops below the minimum water level, the time of exceedings is monitored. In the output file, again this time is presented as a percentage of the total time considered.

The input block is closed as usual by a line containing the "=" character in the first column. An example of the input block "*MMM" is given in Fig. 8.5.1.

```
*MMM
+ Block Title
  Maximum Minimum and mean values
+ LINNO  GR-file Name   Tbegin  Tend         Htarget  Hmax   Hmin
 1       5       H-G11   B      00:10:00:00  15.30    15.00  15.60
 2       -       H-G12   B         -         14.25    13.95  15.00
 3       -       H-G15   B         -         13.46    13.23  13.75
-
*END
```

*Fig. 8.5.1 Example of input format of block *MMM (maximum minimum and mean values).*

The complete input list for sub-system MODRES is terminated by an empty line containing "*END" in the first columns.

9 Model results, plots

9.1 Introduction

It is possible to present the results of a simulation in a graph or plot. This graphical subsystem is called MODPLT (MODel PLoTs). The input of subsystem MODPLT is specified in input file which is named PLTIN.DAT. The input file has been organized in one segment of five blocks:

*PLOT specifies the file names and plotter device to be used;
*SIZE specifies the paper size or format of the paper;
*TEXT specifies some general text to be plotted on the drawing;
*AXES specifies the axes to be drawn, and the scales to be used;
*LINS specifies the functions to be represented by a line and the line code to be used;

The blocks have to be repeated in the sequence as above for each new drawing. This can be done an unlimited number of times.

9.2 Identification

The first input block is headed by the identification line:

```
*PLOT  - FILES AND OUTPUT SPECIFICATION
```

The first part of this block consists of a specification of file names in the following form:

```
     INPUT              file name \
     ECHO               file name  >  consistent with job control
     OUTPUT             file name /

     GR-FILE1           file name \
     GR-FILE2           file name  |
     GR-FILE3           file name  >  defined here
     GR-FILE4           file name  |
     GR-FILE5           file name /

-    ^col no >= 2       ^col no >= 20
```

The codes shown in capitals have to be copied again exactly. For the file names the same applies as in the file specification for the previous sub-systems. The character N is given again when the specified file is not used.

Thereafter, the plotter device to be used is given under the comment heading:

```
+      PLOTTER DEVICE
```

At present the available plotter devices are listed:

- CALCOMP
- HP7475A
- HERCULES
- CGA
- VGA
- EGA

The last four devices are producing a colour graph on your screen. Of course the plotter device should be chosen in accordance with your graphical cart e.g Hercules, VGA.

Finally, the text font have to be specified, under the heading:

+ TEXT FONT

The two possible text fonts are:

- DRAFT
- FINAL

The input block is closed again by giving the character '=' in the first column of a closing line.

An example of the input block *PLOT is given in Fig. 9.2.1

```
*PLOT  -  SPECIFICATION OF FILES AND PAPER-FORMAT
+      USED DATA SETS
       INPUT         PLTIN.DAT
       ECHO          PLTECH.DAT
       OUTPUT        PLTOUT.DAT
       GR-FILE1      USPLF.UNF
       GR-FILE2         N
       GR-FILE3         N
       GR-FILE4         N
       GR-FILE5      USTIF.UNF
+     PLOTTER DEVICE
-        CALCOMP
       VGA
-        EGA
+
+ TEXT FONT
  FINAL
-
```

*Fig. 9.2.1   Example of input block *PLOT*

9.3 Specification of plot size

Input has to be continued by the actual specification of a drawing. The first input block of such a specification is *SIZE with identification line:

```
*SIZE -   SPECIFICATION OF PLOT SIZE
```

The first input line is preceded by the comment line:

```
+     PLOT SIZE          WIDTH       HEIGHT
```

Where:

PLOT SIZE: specifies the paper format. Usually a standard size according to the A-norm will be used (e.g. A4 or A3). One can also select a number of times a certain standard format. For instance 3A4 specifies a paper format with the height of an A4 sheet and the width of 3 times an A4 sheet. A non-standard format can be specified by U. In this case the next data items on the line will be read, to specify the paper size. The default value is: A4.

WIDTH: specifies the width of the paper in centimetres in case of a non-standard paper size. The default value is: 21.0 (A4 width).

HEIGHT: specifies the height of the paper in centimetres in case of a non-standard paper size. The default value is 29.7. (A4 height).

The input block is terminated again by giving the character '=' in the first column of a closing line.

An example of the input block *SIZE is given in Fig. 9.3.1

```
*SIZE -   PLOT SIZE
+
+   PLOT SIZE          WIDTH   HEIGHT
+
       A3                #        #
-      A4                #        #
-      U               29.7     21.0
-
```

*Fig. 9.3.1   Example of input block *PLOT*

9.4 Specification of plot text

Input is continued by input block *TEXT to specify some general text to be plotted. The input starts with the identification line

```
*TEXT   -     PLOT TEXT
+  HEIGHT     XCO    YCO        TEXT
```

Where:

HEIGHT	Specifies the height of the characters when plotted. There is a default for each string number.
XCO	Specifies the x-coordinate of the text when plotted. There is a default value for each string number.
YCO	Specifies the y-coordinate of the text when plotted. There is a default value for each string number.
TEXT	Specifies the text to be plotted. At maximum 10 text strings can be specified to be plotted on the drawing.

If for all position values (XCO and YCO) default values are used the 10 text strings are positioned as in Figure 9.4.1. The sequence number is based on the sequence in the input form. One can change this sequence however by using the specification POS<no> in stead of (XCO,YCO). If the position is specified for instance by POS4 the default position values are used for string 4, independent of the sequence number in the input form.

The default values for HEIGHT for the various strings are:

string no	1	2	3	4	5	6	7	8	9	10
height in cm	0.4	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.5	0.4

The input block *TEXT is terminated again by giving the character '=' in the first column of a closing line.

An example of the input block *TEXT is given in Fig. 9.4.1

```
*TEXT  -  PLOT TEXT
+
+    HEIGHT        XCO   YCO           TEXT
+
     #              POS1               MODPLT
     #              POS2               F19 Control of W.M.S.
     #              POS3               Main canal performance
-     #              POS4               DISCHARGE OBSERVATIONS
     #              POS4               WATER LEVELS
     0.4            POS5               KP = 10, KD = 3, S = 0.002
     0.2            POS6               T.U. DELFT
     #              POS7
     #              POS8
     #              POS9
-     #              POS9
     #              POS10              MODIS
=
```

*Fig. 9.4.1   Example of input block *PLOT*

9.5 Definition of axes

The next input block of the form is used to define the axes and the scale used in the drawing. The block is started with the identification line:

```
*AXES  -     SPECIFICATION OF AXES
+
+ NR.  START  END  LEN  GRD  PLACE  TEXT
```

Where:

NR	Specifies the axis number. This number is used later on in input block *LINS to specify what axis and scale have to be used to draw a certain function. Number 1 is used to define the X-axis.
START	Defines the off set value at the beginning of the axes.
END	Defines the end value at the end of the axes.
LEN	Defines the length of the axes on the paper in cm. Thus the scale is defined by (END-START)/LEN.
GRD	Specifies whether a grid has to be plotted or not. If 'Y', usually a vertical grid is drawn. If it is used for the X-axes a horizontal grid is drawn. Default value is 'N'.
PLACE	Defines the place where the axes have to be plotted. Possible values are 'L' or 'R'. For the X-axis this specification does not have a meaning, but a value has to be given (e.g. XX). If 'L' is specified, the axis is positioned at the left end of the X-axis. If 'R' is specified the axis is positioned at the right end. It is possible to specify 'L' or 'R' more then once. In this case the axes are drawn next to each other. Default value is 'L'
TEXT	Defines the text to be plotted. The default text is blank.

As usual the input block is terminated by a line with a '=' as the control character.

An example of the input block *SIZE is given in Fig. 9.5.1

```
*AXES    -    SPECIFICATION OF AXES
+             AXIS NR 1 IS THE X-AXIS
+
+ NR.     START       END            LEN GRD PLACE TEXT
+
  1     00:00:00:00 00:09:00:00      #    N    -    SIMULATION TIME
- 2       0.0         25.0           #    N    L    DISCHARGE IN M3/S
  2     109.0        111.0           #    N    L     WATER LEVEL IN M
-
```

*Fig 9.5.1 Example of input block *PLOT*

9.6 Function specification

Input is continued by the last input block needed to define one drawing. This input block starts with the identification line:

```
*LINS       -  SPECIFICATION OF FUNCTIONS TO BE DRAWN
+
+ NUMBER   GR    CODE  UNIT  FUNCTIONS ( MAX. 5) +  AXIS   FILE
```

Where:

NO. AXIS	Is the axis number from input block *AXES to be used when drawing the function
GR-FILE	Is the GR-file number corresponding to input block *PLOT from which the function has to be retrieved.
CODE	Defines the line code to be used to represent the funct-ion. Possible values are $1 to $24 corresponding to 24 different line fonts. Besides that, the user can define its own line font by specifying two arbitrary characters. This will generate a line font of a continuous line, con-taining those two characters periodically.
UNIT	Defines a unit string used in the plot legend
FUNCTION	Here the function codes have to be given to be plotted. More than one function can be given, if it is done for the whole block. Then the functions are concatenated.

The input block is closed by a line containing the symbol "=" in the first column. The complete input list for sub-system MODPLT is terminated again by the line containing the character string "*END" in the first columns.

An example of the input block *LINS is given in Fig. 9.6.1

```
*LINS    -    SPECIFICATION OF LINES IN DRAWING
+
+ NUMBER  GR  CODE  UNIT  FUNCTIONS (MAX. 5)
+  AXIS  FILE
   1       5     XX  D:M:H:Y   TIME
   2       =     11  ..M...   H-GM02.1
   2       =     12  ..M...   H-GM03.1
   2       =     13  ..M...   H-GM04.1
=
*END OF PLTDEF
```

*Fig 9.6.1   Example of input block *PLOT*

10 Example

10.1 Description of the system

To illustrate the use of the MODIS model an example will be given. First, a description of the system will be presented. In subsequent paragraphs the input files of the MODIS model will be shown provided with explanatory comment.

■ Description of the system

A 60 km long primary irrigation canal receives its water from the New Delft river. (Fig. 10.1.1). The canal has a capacity of about 40 m^3/s and a Strickler resistance coefficient of 40 $m^{1/3}/s$. The longitudinal section of the primary canal has been depicted in Fig. 10.1.2. The required irrigation water is extracted from the river by using a pumping station with a capacity 40 m^3/s. In total 13 secondary canals receive water from the primary canal.

In the main canal the water level is controlled with cross regulators located downstream of each offtake, and the inflow to the secondary canals is controlled by offtake structures. No seepage losses are assumed to occur in the primary canal.

In this example only two offtakes and two water level regulators are operated: all other offtake structures are assumed to be closed. The water level regulators of the primary canal system are flat sliding gates. The offtake structures are moveable weirs (e.g. Romijn weirs).

The intended water distribution is given in Table 10.1. After 8 o' clock the inflow discharge is increased from 10 m^3/s to 13 m^3/s. The problem is how and when to reset the offtakes in order to realize the new intended water distribution as presented in Table 10.1. Furthermore, the question is raised : which gain parameters are needed to maintain the target water levels at the upstream control water level regulators?

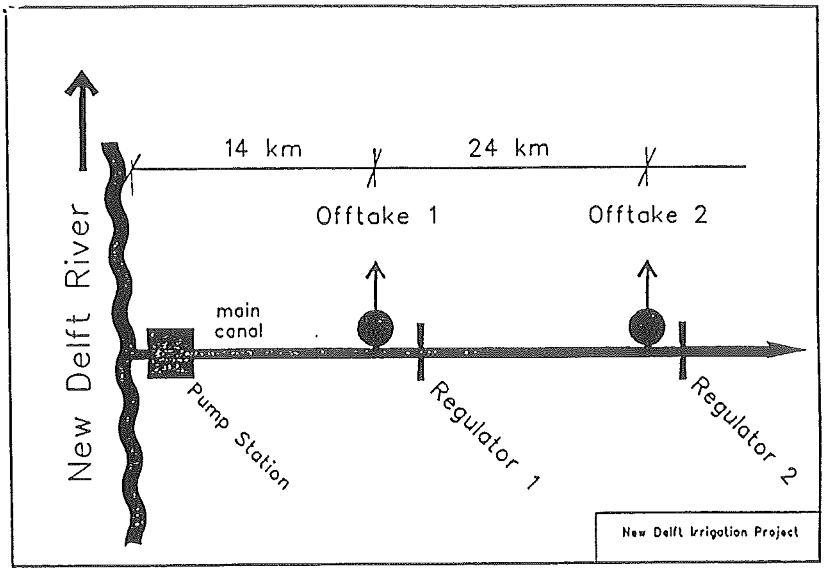


Fig. 10.1.1 New Delft irrigation project

To answer the questions stated above, the irrigation system is modelled and simulated. The results are evaluated using the operation performance indicators incorporated in the MODIS model.

Table 10.1 Intended discharges

Location	INTENDED DISCHARGES in m^3/s	
	Before 8 a.m.	After 8 a.m.
Q_{pump}	10.0	13.0
$Q_{offtake1}$	3.0	3.0
$Q_{offtake2}$	3.0	2.0
$Q_{ongoing}$	4.0	8.0

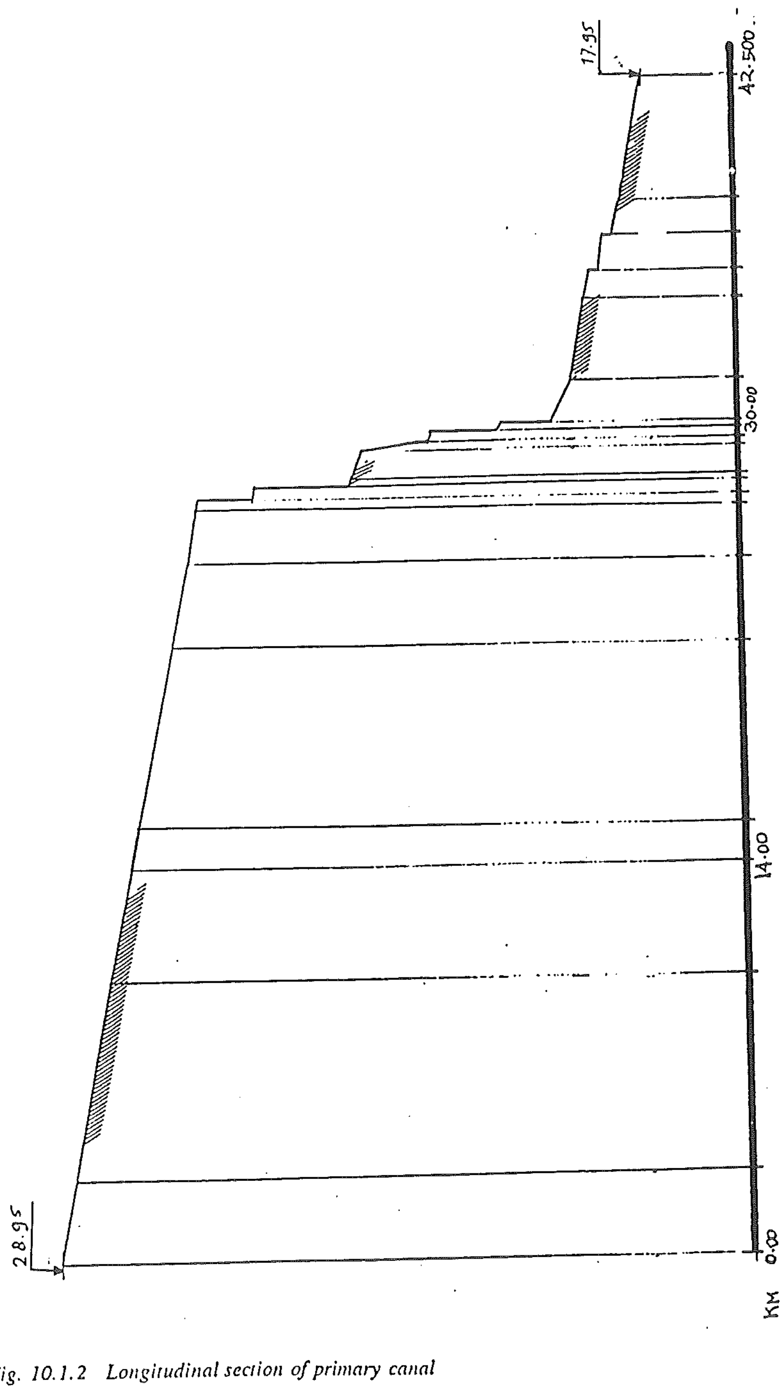


Fig. 10.1.2 Longitudinal section of primary canal

10.2 Model Definition

The input file DEFIN.DAT of the model is shown on the following pages. Some additional comment on the input file will be listed. The remarks are grouped by the input segments. (Also refer chapter 5).

IDEN The data specified in this input segment can usually be copied from another existing input file. With respect to the switches it should be noted that DEFINI which is now on Y(es), have to be put on "N" if one wants to start the computation from a previous computed steady state. The first time a computation is made, this switches should be put on Y.

GRDDEF The schematisation of the system is shown in Fig. 10.2.1. The applied boundary conditions are:
QT (Discharge tabulated as a function of time) with the Table name QINFLOW for the first node.
JW (Junction of water levels) for the internal nodes
HC (Water level constant) for the boundary notes. The water levels are specified lower than the sill levels of the structures so that free flow through the structures occurs.
Six trapezoidal cross-sections have been defined.
Gridpoints have been defined at the beginning and end of each branch, and/or upstream and downstream of each structure. The reference level refers to the bottom level of the canal.

ADDDEF 11 orifice have been defined. they are functioning as water level regulators. The F-coefficient of each orifice equals one which implies that submerged flow is assumed if the downstream level exceeds the gate opening height. Only the PX positive direction has been defined. Reversal flow will be equal to zero.
Two regulators have been automated. For those regulators the gate opening height has been given as a function of LA (Local Automatic) control, instead of a fixed value. The name of the controllers that are used are CNTR3 and CNTR10.
The offtake structures are weirs whereby the sill level is varied as a Tabulated function of Time (TT). The drowned flow reduction coefficient is a Tabulated function of the ratio

downstream and upstream head (TD).

ACSDEF Two local controllers have been used. The control parameters are specified as fixed values in this example.

FUNDEF Two types of function have been used: Tabulated function of time and general tabulated functions.

INIDEF The initial state. For every branch the initial state has to be defined at least the boundary gridpoints of that branch.

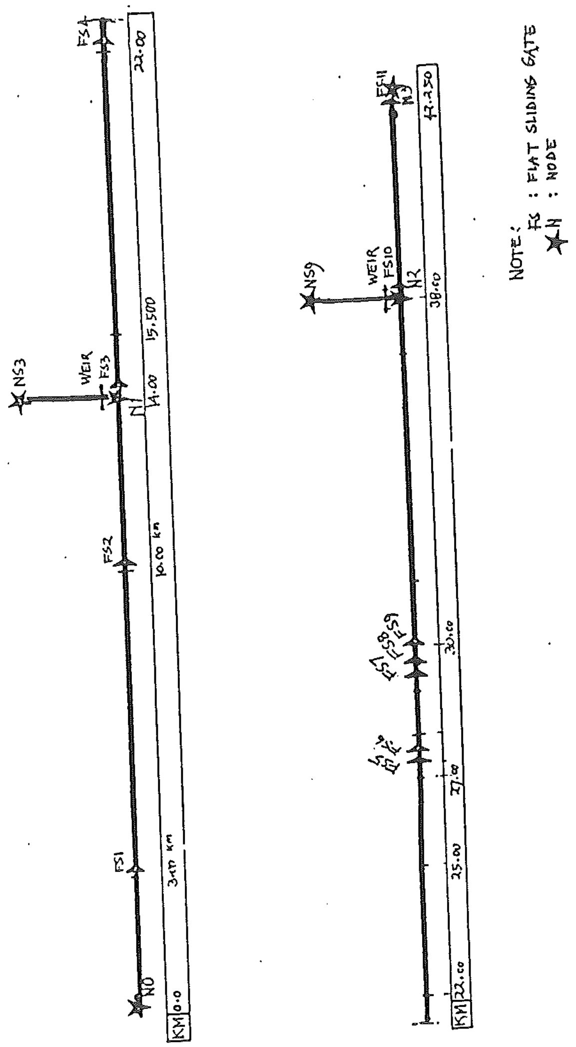


Fig. 10.2.1 Modelled Irrigation system

Input file DEFIN.DAT

```
*IDEN          RUN-IDENTIFICATION
+                                   <...................................>
     RUN-DESCRIPTION (40 CHARS.): New Delft irrigation project
+                                   <...>
     RUNCODE                      : March 1991
+                                   <............>
     EXECUTED  BY                 : TU-DELFT
+
+    DATASET (FILENAMES)
+
-                <.........>
     INPUT       DEFIN.DAT    (JCL-MODIFY)
     ECHO        DEFECH.DAT       ''
     OUTPUT      DEFOUT.DAT       ''
     MODELDB     USMDB.UNF
     SST-FILE    USSST.UNF
=
*SWIT     RUN-SWITCHES
+                      Y=YES N=NO #=DEFAULT SETTING
+
+  NEWMODDB   WRMODDB    DEFALL     OUTALL     DEBUGOUT   FREE       FREE
      Y          Y          N          N          N          #          #
+  DEFNET     DEFGRD     DEFFUN     DEFINI     DEFLAT     DEFSTRUC   DEFACS
      Y          Y          Y          Y          Y          Y          Y
+  OUTNET     OUTGRD     OUTFUN     OUTINI     OUTLAT     OUTSTRUC   FREE
      Y          Y          Y          Y          Y          Y          #
=
*END      of  IDNDEF
*NODE     Definitions of nodes
+
+NAME  TYPE   CONNECTED BRANCHES                     Boundary Condition
+                                                                        value / reference
  NO     QT    BP1                                   QINFLOW
  N1     JW    BP1    BS3    BP2
  NS3    HC    BS3                                   28.0
  N2     JW    BP2    BS9    BP3
  NS9    HC    BS9                                   19.0
  N3     HC    BP3                                   18.40
+
=
```

```
*BRAN     Definition of branches
+NAME          BEGIN-NODE  END-NODE  X-BEGIN      X-END       DX-MAX
+
   BP1          N0          N1           0.0      14000.0      1000.0
   BS3          N1          NS3          0.0        100.0       100.0
   BP2          N1          N2       14000.0      38000.0      1000.0
   BS9          N2          NS9          0.0        100.0       100.0
   BP3          N2          N3       38000.0      42500.0      1000.0
+
=
*END
*TRAP  Trapezoidal cross-sections
+ NAME     REFNCE    BUESNQ    Res      BOTWIDTH  SIDESLOPE    Hmax
+                              coeff
  C0         0.0       1.0      40        12.0       2.0        3.98
  C1         0.0       1.0      40        10.8       2.0        =
  C2         0.0       1.0      40        10.4       2.0        =
  C3         0.0       1.0      40         9.6       2.0        =
  C4         0.0       1.0      40         7.8       2.0        =
  C5         0.0       1.0      40         7.2       2.0        =
=
*GRID     DEFINITION OF GRIDPOINTS
+ GRIDPOINT  CROSS-SEC    BRANCHE   X-COORDINATE        REF. LEVEL
+    name       name        name    in the branch       SI unit
  G0            C0          BP1          0.0            28.95
  G1            =           =         2995.0            28.65
  G1R           C1          =         3000.0            =
  G2            =           =         9995.0            27.95
  G2R           C2          =        10000.0            =
  G3L           =           =        14000.0            27.55
  G3            C3          BP2      14000.0            =
  G3R           C3          BP2      14005.0            =
  G4            C3          BP2      21995.0            26.75
  G4R           C4          =        22000.0            =
  G5            =           =        27295.0            26.22
  G5R           =           =        27300.0            25.15
  G6            =           =        27595.0            25.12
  G6R           =           =        27600.0            23.40
  G7            =           =        29295.0            23.10
  G7R           =           =        29300.0            21.90
  G8            =           =        29595.0            21.85
  G8R           =           =        29600.0            20.65
  G9            =           =        29995.0            20.55
  G9R           =           =        30000.0            19.64
  G10L          =           =        38000.0            18.40
  G10           C4          BP3      38000.0            18.40
  G10R          =           BP3      38005.0             =
  G11           =           BP3      42495.0            17.95
  G12           =           =        42500.0            16.95
```

```
  GS3           C5          BS3            0.0           27.55
  GS3E          =           =            100.0           27.50
  GS9           C5          BS9            0.0           18.40
  GS9E          =           =            100.0           18.35
=
*END OF GRDDEF
*ORIF
+
+ NAME    BRANCH    X-COORD    DIR   F     Ce    u   B    z        Y
+
  FS1     BP1        2998.0   PX    1.0   1.0  0.63 10   28.65    0.30
  FS2     BP1        9998.0   =     =     =    =    =    27.95    0.55
  FS3     BP2       14002.0   =     =     =    =    =    27.55    LA$CNTR3
  FS4     BP2       21998.0   =     =     =    =    6    26.75    0.90
  FS5     BP2       27298.0   =     =     =    =    =    26.22    0.50
  FS6     BP2       27598.0   =     =     =    =    =    25.12    0.53
  FS7      =        29298.0   =     =     =    =    =    23.10    0.53
  FS8      =        29598.0   =     =     =    =    =    21.85    0.53
  FS9      =        29998.0   =     =     =    =    =    20.55    0.55
  FS10    BP3       38002.0   =     =     =    =    5    18.40    LA$CNTR10
  FS11    BP3       42498.0   =     =     =    =    =    17.95    0.50
=
*WEIR
+ NAME    BRANCH      X-COORD    DIR     F      Ce     u    B      z
+
  WS3     BS3             5.0    PX    DT$FRED 1.0    1.5 12.0  TT$OFF1
  WS9     BS9             5.0    PX      =      =     1.5 12.0  TT$OFF2
=
*END    of ADDDEF
*AULC   AUTOMATIC LOCAL CONTROL
+
+ NAME   CONTROL  TYPE   Inival TARGET    Kp     Ki     Kd     SPEED  MIN    MAX
  CNTR3    PID     UC    0.23   30.15    -3.0   -1.8   0.5     0.5   0.0    2.0
  CNTR10   PID     UC    0.34   20.17    -3.0   -1.8   0.5     1.5   0.0    2.0
=
*END of automatic controllers
*TTIM          TABULATED FUNCTIONS OF TIME
+
  NAME              QINFLOW      OFF1      OFF2
+
  00:00:00:00        10.0        29.872    19.892
  00:07:59:00         =           =         =
  00:08:00:00        13.0        29.872    19.97
  05:00:00:00         =           =         =
+
=
```

```
*TGFU
+
  NAME              FRED
+(h2-z)/(h1-z) Drowned flow
+                reduction factor
  0.000        1.00
  0.667        1.00
  0.800        0.75
  0.900        0.55
  1.000        0.00
  10.0         0.00
=
*END    of FUNDEF
*INST   Definition of initial state of canal system
+
+ BRANCH       Grid Point    TYPE  VAL/REF       TYPE  VAL/REF
+                                   H                    Q
  BP1          G0            C     31.93         C     0.0
   =           G1            =      =            =      =
   =           G1R           =     30.67         =      =
   =           G2            =      =            =      =
   =           G2R           =     30.15         =      =
   =           G3L           =      =            =      =
  BS3          GS3           =      =            =      =
   =           GS3E          =      =            =      =
  BP2          G3            =      =            =      =
   =           G3R           =     29.55         =      =
   =           G4            =      =            =      =
   =           G4R           =     28.18         =      =
   =           G5            =      =            =      =
   =           G5R           =     27.08         =      =
   =           G6            =      =            =      =
   =           G6R           =     25.06         =      =
   =           G7            =      =            =      =
   =           G7R           =     23.81         =      =
   =           G8            =      =            =      =
   =           G8R           =     22.51         =      =
   =           G9            =      =            =      =
   =           G9R           =     20.19         =      =
   =           G10L          =      =            =      =
  BP3          G10           =      =            =      =
  BP3          G10R          =      =            =      =
   =           G12           =      =            =      =
  BS9          GS9           =      =          =      =
   =           GS9E          =     19.40         =      =
 +
 =
 *END OF INIDEF
```

10.3 Model computation

The input file COMIN.DAT of the subprogram MODCOM is shown on the following pages. For more information about the input data reference is made to chapter 6.

Only one remark is made about the input block *TIME. In the first computation, the SST (System State Time) file should be on INIDEF which implies that the initial state defined in DEFIN.DAT is read and used as the initial state.
If the Steady state switch of input block *SWIT has been put on Y(es) a new initial state has been generated by the model and the SST could be set on PERM (permanent state) for the next computations. To save the initial state generated by the model the STEADY STATE switch should be put on N(o) and in the input file DEFIN.DAT the DEFINI switch should be put on N(o). If this is not done the model will continually re-write a new initial state or overwrite the model generated initial by the user defined initial state.

Input file COMIN.DAT

```
*IDEN   RUN IDENTIFICATION
-                <...>
     RUNCODE     NEW DELFT
+
-                <..........>
     INPUT       COMIN.DAT    (FIXED)
     ECHO        COMECH.DAT     ,,
     OUTPUT      COMOUT.DAT     ,,
     MODDB       USMDB.UNF
     SST-FILE    USSST.UNF
     PLF-FILE    USPLF.UNF
     RTF-FILE    USRTF.UNF
=
```

```
*SWIT     RUN-SWITCHES
+                        Y=YES N=NO #=DEFAULT SETTING
+  TEST        TESTC      TESTS    TESTO    DEBUG    <FREE-SWITCHES>
    N           N          N        N         N          #     #
+  PLFRDB      TIFRDB    SSTRDB   STEADY-STATE      <FREE-SWITCHES>
    Y           Y          Y        Y                  #  #  #
+  NEWRPF      NEWRTF                                <FREE-SWITCHES>
     Y           Y                                     # # # # #
=
*PARM    COMPUTATIONAL PARAMETERS
+     PSI        THETA       ITERM        G
     0.5         0.55         2          9.82
+     EPSOFL     EPSUFL      EPSHFL     EPSDH
     0.01        0.01        0.01       0.01
=
*TIME          TIME CONTROL PARAMETERS
+     BEGIN-TIME  END-TIME      TIME-STEP    SST-USAGE
   00:00:00:00  01:00:00:00   00:00:20:00  INIDEF
-     00:00:00:00  02:00:00:00   00:00:10:00  PERM
=
*RSST  SYSTEM STATE RESULT DATA FILE
+
+   BEGIN       END      INTERVAL
    B           E        00:08:00:00
=
*RTIF  CONTROL PARAMETRS FOR RESULT-TIME FUNCTION FILE
+
+   BEGIN       END        INTERVAL
      B          E           1*
+ CODE     GRIDPOINT    FUN.SPEC        FUNCTION TEXT ( 40 CHARACTERS)
+
   H-G0       G0           HN            Water level at intake
   Q-INFLOW   G0           QN            Discharge at intake
   H-G1       G1           HN            G1
   H-G2       G2           HN            G2
   H-CHECK1   G3L          HN            Water level upstream of Regulator 1
   Q-G3L      G3L          QN            Discharge upstream of Regulator 1
   Q-G3R      G3R          QN            Q  AFTER OFFTAKE S3
   H-G3R      G3R          HN            WL AFTER x = 14000 m
   H-G4       G4           HN            G4
   H-G5       G5           HN            G5
   H-G6       G6           HN            G6
   H-G7       G7           HN            G7
   H-G8       G8           HN            G8
   H-G9       G9           HN            G9
   H-CHECK2   G10L         HN            Water level upstream of Regulator 2
   Q-ONGOING  G10R         QN            Discharge downstream of Regulator 2
   Q-G10L     G10L         QN            Discharge upstream of Regulator 2
   H-G10      G10          HN            Water level upstream of Regulator 2
```

```
  H-G10R    G10R        HN          Water level downstream of Regulator 2
  H-G11     G11         HN          G11
  H-G12     G12         HN          WL on x = 42500 m
  Q-G12     G12         QN          Q  on x = 42500 m
  Q-OFF1    GS3         QN          Discharge through Offtake 1
  Q-OFF2    GS9         QN          Discharge through Offtake 2
  H-GS9     GS9         HN          WL S9
=
*ROUT
+
+ ROUTE   NODES
  ROUT1   N0   N1    N2   N3
=
*BASF
+ TITLE    ROUTE     VAR SPEC  FUNCTION TEXT
  N-ROUTE  R$ROUT1   GRIDNO    GRID NUMBERS
  X-ROUTE  R$ROUT1   X         X-COORDINATES
=
*RPLF
+
+ RESET
  N
+ TITLE    BRANCH    FUNC.SPEC    MOMENT IN TIME     TEXT
  H-ROUT1  R$ROUT1   HN           00:01:00:00        WL AT 1 HOUR
  H-ROUT2  R$ROUT1   HN           00:09:00:00        WL AT 9 HOUR
  H-ROUT3  R$ROUT1   HN           00:13:00:00        WL AT 13 HOUR
  H-ROUT4  R$ROUT1   HN           00:17:00:00        WL AT 17 HOUR
  H-ROUT5  R$ROUT1   HN           00:21:00:00        WL AT 21 HOUR
=
*END
```

10.4 Result Data Base

The input data file UTLIN.DAT of the subprogram RDBUTL is shown below. For more information reference is made to chapter 7.

Input file UTLIN.DAT

```
*FILS  SPECIFICATION OF USED DATA SETS
+
     INPUT       UTLIN.DAT
     ECHO        UTLECH.DAT
     OUTPUT      UTLOUT.DAT
     RDB-SST     USSST.UNF
     RDB-RTF     USRTF.UNF
+
+  RDB-FILES IN GENERAL RESULT FUNCTION FORMAT
     GR-FILE1    USPLF.UNF
     GR-FILE2      N
     GR-FILE3      N
     GR-FILE4      N
     GR-FILE5    USTIF.UNF
+
+  SELECT SWITCHES FOR UTL-ACTIONS
+     TRAFO-ACTION         CONT-ACTION
          Y                    Y
+
=
*TRAFO    SPECIFICATION OF TIME RANGE FOR  RESULT DATA BASE TRANSFORMATION
+         T-BEGIN         T-END
              B             E
+
+    FUNCTIONS TO BE TRANSFORMED   (OR    ALFUN   FOR ALL FUNCTIONS)
+
            ALFUN
=
+
+
*CONT      INVESTIGATE CONTENTS OF RESULT-DATABASE
+ TIME FUNCTIONS
+ SST       RTF
 *Y         Y
+ PLACE-FUNCTIONS
+          GRF-1    GRF-2    GRF-3    GRF-4    GRF-5
             Y        N        N        N        Y
=
*END
```

10.5 Model Results Tables

The results of the simulations can be presented in various ways. Here the input file RESIN.DAT has been presented in which each method of presentation is used. In practice, not all of them are used simultaneously.

Input file RESIN.DAT

```
*PRI   SPECIFICATION OF FILES AND PAPER-FORMAT
+     USED DATA SETS
      INPUT        RESIN.DAT
      ECHO         RESECH.DAT
      OUTPUT       RESOUT.DAT
      GR-FILE1     USPLF.UNF
      GR-FILE2        N
      GR-FILE3        N
      GR-FILE4        N
      GR-FILE5     USTIF.UNF
+
+    CHARACTERS/LINE   LINES/PAGE
         132              60
+
+    SUMMARY OUTPUT ONLY (Y/N)
              N
+
=
*BLK    SPECIFICATION OF PRINT OUTPUT SET
+
+    BLOCK TITLE
     Q Offtakes & Regulators
+
+COLNO GR-FILE FORMAT  UNITS    FUNCTION  FUNCTION  FUNCTION  FUNCTION
    1     5      #     D:H:M:S     TIME
    2     =      #     ...M3/S     Q-INFLOW
    3     =      #     ...meter    H-CHECK1
    4     =      #     ...M3/S     Q-OFF1
    5     =      #     ...meter    H-CHECK2
-    6     =      #     ...M3/S     H-G10R
    6     =      #     ...M3/S     Q-OFF2
    7     =      #     ...M3/S     Q-ONGOING
=
```

```
*BLK    SPECIFICATION OF PRINT OUTPUT SET
+
+BLOCK TITLE
 ROUT1
+COLNO GR-FILE FORMAT  UNITS    FUNCTION  FUNCTION  FUNCTION  FUNCTION
    1     1      #      ....M...     X-ROUTE
    2     =      #      ...M...      H-ROUT1
    3     =      #      ...M...      H-ROUT2
    4     =      #      ...M...      H-ROUT3
    5     =      #      ...M...      H-ROUT4
    6     =      #      ...M...      H-ROUT5
=
*OPP
+BLOCK TITLE
 Operation performance indicators
+LINNO GR-F NAME      Tbegin       Tend         Qtarget  Qtmax  Qtmin
  1     5    Q-OFF1    00:08:00:00  02:00:00:00  3.0      15%    15%
  2     =    Q-OFF2    =            =            2.0      =      =
  3     =    Q-ONGOING =            =            8.0      =      =
  4     =    Q-INFLOW  =            =            TOTAL
=
*MMM
+
+Block Title
 Maximum, minimum and mean values
+LINNO GR-F NAME      Tbegin       Tend         Htarget  Hmax   Hmin
+
  1     5    H-CHECK1 00:00:00:00  02:00:00:00  30.15    30.35  30.14
  2     =    H-CHECK2 =            =            20.17    20.37  20.16
=
*END
```

After running the subprogram MODRES, the output file RESOUT.DAT will look as follows.

Output file RESOUT.DAT

```
05 APR 91 09:09:35              - 1-


 TITLE :                                                   USED DATASETS

                                                             5 :  45

 FUNCTION SUMMARY
 ----------------

 COL NAME       FUNCTION DESCRIPTION                    TYP MIN.VALUE MAX.VALUE MEAN VAL. LEN
 ------------------------------------------------------------------------------------------
   1 TIME       TIME FUNCTION                           C        .00       .00       .00  72
   2 Q-INFLOW   Discharge at intake                     R      10.00     13.00     12.04  72
   3 H-CHECK1   Water level upstream of Regulator 1     R      30.13     30.19     30.15  72
   4 Q-OFF1     Discharge through Offtake 1             R       -.23      3.64      2.97  72
   5 H-CHECK2   Water level upstream of Regulator 2     R      20.09     20.25     20.17  72
   6 Q-OFF2     Discharge through Offtake 2             R       1.57      4.37      2.22  72
   7 Q-ONGOING  Discharge downstream of Regulator 2     R        .00      7.87      6.34  72

05 APR 91 09:09:35              - 2-

 Q Offtakes & Regulators


 TIME          Q-INFLOW  H-CHECK1  Q-OFF1    H-CHECK2  Q-OFF2    Q-ONGOING
 D:H:M:S       ...M3/S   ...meter  ...M3/S   ...meter  ...M3/S   ...M3/S
 -------------------------------------------------------------------------
 000:00:20:00     10.00     30.14      -.23     20.09      2.15       .01
 000:00:40:00     10.00     30.19      3.64     20.12      2.25       .00
 000:01:00:00     10.00     30.18      3.47     20.09      1.75       .77
 000:01:20:00     10.00     30.13      2.63     20.13      2.40       .00
 000:01:40:00     10.00     30.16      3.15     20.22      3.80       .45
 000:02:00:00     10.00     30.14      2.86     20.25      4.37      3.91
 000:02:20:00     10.00     30.14      2.78     20.21      3.69      6.09
 000:02:40:00     10.00     30.16      3.19     20.18      3.15      5.87
 000:03:00:00     10.00     30.14      2.81     20.21      3.70      5.41
 000:03:20:00     10.00     30.15      3.00     20.20      3.43      6.45
 000:03:40:00     10.00     30.15      3.00     20.18      3.09      6.45
 000:04:00:00     10.00     30.14      2.89     20.20      3.52      6.22
 000:04:20:00     10.00     30.15      3.04     20.17      3.07      6.82
 000:04:40:00     10.00     30.15      2.95     20.18      3.24      6.50
```

000:05:00:00	10.00	30.15	2.99	20.18	3.20	6.74
000:05:20:00	10.00	30.15	2.97	20.17	3.03	6.77
000:05:40:00	10.00	30.15	3.02	20.19	3.26	6.67
000:06:00:00	10.00	30.15	2.93	20.17	2.95	6.91
000:06:20:00	10.00	30.15	3.05	20.18	3.23	6.69
000:06:40:00	10.00	30.15	2.94	20.17	2.95	6.93
000:07:00:00	10.00	30.15	3.02	20.18	3.17	6.75
000:07:20:00	10.00	30.15	2.98	20.17	2.98	6.92
000:07:40:00	10.00	30.15	2.99	20.18	3.11	6.79
000:08:00:00	13.00	30.15	3.00	20.20	2.24	7.09
000:08:20:00	13.00	30.15	2.99	20.20	2.21	7.66
000:08:40:00	13.00	30.15	3.00	20.19	2.10	7.77
000:09:00:00	13.00	30.15	3.00	20.19	2.11	7.77
000:09:20:00	13.00	30.15	3.00	20.19	2.05	7.84
000:09:40:00	13.00	30.15	3.01	20.19	2.04	7.85
000:10:00:00	13.00	30.15	3.01	20.18	2.01	7.87
000:10:20:00	13.00	30.15	3.01	20.18	2.00	7.87
000:10:40:00	13.00	30.15	3.02	20.18	1.97	7.87
000:11:00:00	13.00	30.15	3.02	20.18	1.96	7.87
000:11:20:00	13.00	30.15	3.02	20.18	1.94	7.86
000:11:40:00	13.00	30.15	3.02	20.18	1.93	7.85
000:12:00:00	13.00	30.15	3.02	20.18	1.92	7.84
000:12:20:00	13.00	30.15	3.02	20.18	1.91	7.83
000:12:40:00	13.00	30.15	3.02	20.17	1.90	7.82
000:13:00:00	13.00	30.15	3.02	20.17	1.89	7.81
000:13:20:00	13.00	30.15	3.02	20.17	1.87	7.78
000:13:40:00	13.00	30.15	3.02	20.17	1.84	7.74
000:14:00:00	13.00	30.15	3.02	20.17	1.80	7.68
000:14:20:00	13.00	30.15	3.02	20.16	1.75	7.60
000:14:40:00	13.00	30.15	3.02	20.16	1.70	7.49
000:15:00:00	13.00	30.15	3.02	20.16	1.67	7.37
000:15:20:00	13.00	30.15	3.02	20.16	1.63	7.26
000:15:40:00	13.00	30.15	3.02	20.15	1.60	7.14
000:16:00:00	13.00	30.15	3.02	20.15	1.58	7.02
000:16:20:00	13.00	30.15	3.02	20.15	1.57	6.90
000:16:40:00	13.00	30.15	3.02	20.15	1.57	6.79
000:17:00:00	13.00	30.15	3.02	20.15	1.57	6.68
000:17:20:00	13.00	30.15	3.02	20.15	1.58	6.58

05 APR 91 09:09:35 - 3-

Q Offtakes & Regulators

TIME D:H:M:S	Q-INFLOW ...M3/S	H-CHECK1 ...meter	Q-OFF1 ...M3/S	H-CHECK2 ...meter	Q-OFF2 ...M3/S	Q-ONGOING ...M3/S
000:17:40:00	13.00	30.15	3.02	20.15	1.59	6.49
000:18:00:00	13.00	30.15	3.02	20.15	1.62	6.42
000:18:20:00	13.00	30.15	3.02	20.16	1.63	6.36
000:18:40:00	13.00	30.15	3.02	20.16	1.64	6.29
000:19:00:00	13.00	30.15	3.02	20.16	1.65	6.21
000:19:20:00	13.00	30.15	3.02	20.16	1.69	6.16
000:19:40:00	13.00	30.15	3.02	20.16	1.71	6.14
000:20:00:00	13.00	30.15	3.02	20.16	1.72	6.12
000:20:20:00	13.00	30.15	3.02	20.16	1.72	6.09
000:20:40:00	13.00	30.15	3.02	20.16	1.73	6.04
000:21:00:00	13.00	30.15	3.02	20.16	1.76	6.00
000:21:20:00	13.00	30.15	3.02	20.16	1.76	6.00
000:21:40:00	13.00	30.15	3.02	20.17	1.77	5.98
000:22:00:00	13.00	30.15	3.02	20.17	1.79	5.96
000:22:20:00	13.00	30.15	3.02	20.17	1.79	5.99
000:22:40:00	13.00	30.15	3.02	20.17	1.81	5.97
000:23:00:00	13.00	30.15	3.02	20.17	1.81	5.99
000:23:20:00	13.00	30.15	3.02	20.17	1.81	5.99
000:23:40:00	13.00	30.15	3.02	20.17	1.83	6.00
001:00:00:00	13.00	30.15	3.02	20.17	1.83	6.03

05 APR 91 09:09:36 - 4-

TITLE : USED DATASETS

1 : 41

FUNCTION SUMMARY

COL	NAME	FUNCTION DESCRIPTION		TYP	MIN.VALUE	MAX.VALUE	MEAN VAL.	LEN
1	X-ROUTE	ATES		R	.00	42500.00	22553.83	60
2	H-ROUT1	00:00	WL AT 1 HOUR	R	18.40	31.98	26.33	60
3	H-ROUT2	00:00	WL AT 9 HOUR	R	18.40	32.30	26.25	60
4	H-ROUT3	00:00	WL AT 13 HOUR	R	18.40	32.69	26.21	60
5	H-ROUT4	00:00	WL AT 17 HOUR	R	18.40	32.93	26.18	60
6	H-ROUT5	00:00	WL AT 21 HOUR	R	18.40	33.10	26.19	60

05 APR 91 09:09:36 - 5-

ROUT1

X-ROUTE ....M...	H-ROUT1 ...M...	H-ROUT2 ...M...	H-ROUT3 ...M...	H-ROUT4 ...M...	H-ROUT5 ...M...
.00	31.98	32.30	32.69	32.93	33.10
998.33	31.97	32.29	32.68	32.93	33.10
1996.67	31.97	32.28	32.68	32.93	33.10
2995.00	31.96	32.28	32.67	32.92	33.09
3000.00	30.76	30.78	30.87	30.94	31.00
3999.29	30.72	30.75	30.83	30.91	30.96
4998.57	30.70	30.72	30.80	30.88	30.94
5997.86	30.68	30.70	30.78	30.85	30.91
6997.14	30.66	30.68	30.75	30.83	30.89
7996.43	30.65	30.66	30.74	30.81	30.87
8995.71	30.64	30.64	30.72	30.79	30.85
9995.00	30.62	30.63	30.70	30.78	30.84
10000.00	30.26	30.23	30.24	30.26	30.26
11000.00	30.22	30.21	30.22	30.22	30.23
12000.00	30.22	30.19	30.19	30.20	30.20
13000.00	30.21	30.17	30.17	30.17	30.17
14000.00	30.18	30.15	30.15	30.15	30.15
14000.00	30.18	30.15	30.15	30.15	30.15
14005.00	29.56	29.06	29.04	29.07	29.13
15003.75	29.50	29.00	28.96	28.99	29.04
16002.50	29.46	28.95	28.89	28.91	28.96
17001.25	29.43	28.90	28.82	28.83	28.88
18000.00	29.39	28.85	28.75	28.76	28.81
18998.75	29.36	28.80	28.69	28.69	28.74
19997.50	29.33	28.76	28.64	28.63	28.68
20996.25	29.32	28.72	28.59	28.58	28.63
21995.00	29.28	28.68	28.54	28.53	28.58
22000.00	28.67	28.40	28.28	28.26	28.29
22882.50	28.53	28.31	28.18	28.16	28.18
23765.00	28.47	28.22	28.07	28.04	28.06
24647.50	28.34	28.13	27.96	27.92	27.94
25530.00	28.19	28.02	27.83	27.78	27.80
26412.50	28.05	27.92	27.68	27.61	27.64
27295.00	27.92	27.79	27.49	27.39	27.42
27300.00	26.67	26.65	26.40	26.32	26.34
27595.00	26.60	26.59	26.31	26.22	26.24
27600.00	25.13	25.07	24.77	24.68	24.69
28447.50	25.04	24.97	24.59	24.48	24.48
29295.00	24.97	24.88	24.39	24.21	24.21
29300.00	23.64	23.67	23.24	23.06	23.06
29595.00	23.60	23.64	23.17	22.96	22.96
29600.00	22.41	22.43	22.25	21.87	21.77
29995.00	22.34	22.39	22.20	21.74	21.61

30000.00	21.31	21.26	21.22	21.11	21.08
31000.00	20.88	21.11	21.08	20.96	20.93
32000.00	21.03	20.96	20.94	20.82	20.79
33000.00	20.49	20.82	20.79	20.68	20.66
34000.00	20.25	20.68	20.66	20.56	20.54
35000.00	20.15	20.54	20.53	20.44	20.42
36000.00	20.11	20.42	20.40	20.33	20.32
37000.00	20.12	20.30	20.28	20.23	20.24
38000.00	20.09	20.19	20.17	20.15	20.16

05 APR 91 09:09:36 - 6-

ROUT1

X-ROUTE ....M...	H-ROUT1 ...M...	H-ROUT2 ...M...	H-ROUT3 ...M...	H-ROUT4 ...M...	H-ROUT5 ...M...
38000.00	20.09	20.19	20.17	20.15	20.16
38005.00	19.90	19.89	19.95	19.88	19.75
38903.00	19.89	19.77	19.86	19.79	19.66
39801.00	19.88	19.65	19.75	19.70	19.55
40699.00	19.85	19.52	19.65	19.60	19.43
41597.00	19.81	19.38	19.55	19.50	19.30
42495.00	19.75	19.22	19.44	19.39	19.14
42500.00	18.40	18.40	18.40	18.40	18.40

05 APR 91 09:09:37 - 7-

Operation performance indicators

Operation Performance Parameters

Point	DPR %	Eo %	Qtarg	Qmean	Qmax	Qmin	Tbegin	Tend
Q-OFF1	100.	99.	3.00	3.02	3.02	2.99	00:08:00:00	02:00:00:00
Q-OFF2	65.	72.	2.00	1.80	2.24	1.57	00:08:00:00	02:00:00:00
Q-ONGOIN	52.	59.	8.00	6.96	7.87	5.96	00:08:00:00	02:00:00:00

Overall Performance
Total DPR : 65.%
Total Eo : 65.%

No.	Minimum	Maximum	Mean	Target	Tmax%	Tmin%
1	30.13	30.19	30.15	30.15	.00	2.78
2	20.09	20.25	20.17	20.17	.00	25.00

10.6 Model results, Plots

The input file PLTIN.DAT of subprogram MODPLT has been depicted below. Both longitudinal sections can be plotted, and, after removing the "-" characters in the first column, water level and discharge variation as a function of time can be plotted.

Input file PLTIN.DAT

```
*PLOT  -  SPECIFICATION OF FILES AND PAPER-FORMAT
+      USED DATASETS
       INPUT        PLTIN.DAT
       ECHO         PLTECH.DAT
       OUTPUT       PLTOUT.DAT
       GR-FILE1     USPLF.UNF
       GR-FILE2        N
       GR-FILE3        N
       GR-FILE4        N
       GR-FILE5     USTIF.UNF
+
+PLOTTER DEVICE
  VGA
-   CGA
+
+TEXT FONT
-  DRAFT
 FINAL
=
*SIZE -   PLOT SIZE
+
+   PLOT SIZE          WIDTH    HEIGHT
+
        A3               #        #
=
```

```
*TEXT  -  PLOT TEXT
+
+    HEIGTH        XCO   YCO           TEXT
+
     #              POS1               M O D I S
     #              POS2               New Delft Irrigation
     #              POS3               CANAL PERFORMANCE
     #              POS4               Discharge
     0.4            POS5               Civil Engineering Dept
     0.2            POS6               T.U. DELFT
     #              POS7
     #              POS8
     #              POS9
     #              POS10
=
*AXES   -   SPECIFICATION OF AXES
+           AXIS NR 1 IS THE X-AXIS
+
+ NR.    START      END         LEN  GRD PLACE TEXT
+
- 1  00:00:00:00  01:00:00:00    #   N   -     Time
  1    0.0          42500        #   N   -     Distance
  2    0.0           13.1        #   N   L     Discharge
  3   18.0           33.0        #   N   R     Water level
=
*LINS   -   SPECIFICATION OF LINES IN DRAWING
+
+ NUMBER  GR  CODE  UNIT  FUNCTIONS (MAX. 5)
+  AXIS  FILE
-   1      5   XX  D:M:H:Y   TIME
    1      1   XX  ..M....   X-ROUTE
    3      =   $1  ..M....   H-ROUT1
    3      =   $1  ..M....   H-ROUT2
-   2      5   $1  ..M3/S.    Q-OFF1
-   2      =   $2  ..M3/S.    Q-OFF2
-   2      =   13  ..M3/S.    Q-ONGOING
-   2      =   14  ..M3/S.    Q-INFLOW
-   3      =   15  ..M...     H-CHECK2
=
*END OF PLTDEF
```

10.7 Concluding remarks

The input files presented in this chapter can also be found on a diskette labelled "DEMO.MOD". After a back up of this diskette has been made, one can play with the model, and see how the system reacts on variations in operation strategy, gain factors of the automatic upstream controllers, and/or other types of offtake structures.

11 Mathematical background

11.1 Canal flow

11.1.1 de Saint Venant equations

In MODIS program package, the unsteady flow in open canal systems is computed on the basis of two partial differential equations of first order. These "de Saint Venant equations" read:

$$\frac{\partial Q}{\partial x} + b_s \frac{\partial h}{\partial t} = q \qquad 11.1$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(\beta \frac{Q^2}{A}) + gA\frac{\partial h}{\partial x} + \frac{gn^2 Q|Q|}{AR^{4/3}} = 0 \qquad 11.2$$

With:

A	= cross sectional area	m^2
Q	= discharge	m^3/s
R	= hydraulic radius	m
b_s	= cross sectional storage width	m
h	= water level	m
n	= Mannings resistance coefficient	$s/m^{1/3}$
q	= lateral inflow	m^2/s
ß	= Boussinesq coefficient	-
g	= acceleration due to gravity	m/s^2
x	= distance along canal axis	m
t	= time	s

Equation (11.1) represents continuity of volume which equals continuity of mass when the density of water is assumed to be constant (fluid is well mixed). The second equation (11.2) represents continuity of momentum, which can be interpreted as dynamic equilibrium of forces including acceleration forces.

11.1.2 Assumptions underlying the de saint venant equations

The assumptions underlying the De Saint Venant Equations can be found by reviewing the derivation of the equations. I based on the law of mass and momentum conservation, is presented. The main simplifications underlying the De Saint Venant Equations are listed:

1) The velocities perpendicular to the direction of flow are negligible (and thus the accelerations also) as compared to the velocity in the direction of flow. This implies that the slope of the water level perpendicular to the direction of flow is assumed to be horizontal.
2) A negligible curvature of the water surface implies a hydrostatic pressure distribution perpendicular to the flow direction (or bed slope).
3) Friction losses in unsteady flow are not significantly different from those in steady flow and therefore the same resistance formulae may be applied.
4) Density of the fluid is assumed to be constant.
5) The derivatives of the coefficient of Boussinesq ß, expressing the non-uniformity of the velocity distribution across the wetted area, are assumed to be constant.
6) The bed slope s is assumed to be small. So that $\cos(s) = 1$ and $\sin(s) = s$
7) Lateral inflow is assumed to be perpendicular to the canal axis and therefore not contributing to the momentum equation
8) Wind forces are not incorporated. To incorporate wind forces the term $b.w^2.\gamma.(\rho - \varphi)$ should be added to the left hand side of equation (11.2). (w = wind velocity, b = cross-sectional flow width, γ = wind conversion coefficient, ρ = wind direction in degrees, φ = direction of canal axis in degrees, measured clockwise from the north.

11.2 Structures

11.2.1 Introduction

A vast number of standard structures are included in MODIS, each with its own characteristic stage-discharge curve. The stage discharge curve of a structure specifies the discharge through a structure as a function of the upstream and downstream water levels. It is more correct to use the energy levels instead of the water levels but, the difference between the energy head and water level head is usually small as a result of the low velocities in canals. Besides water levels are more easy to measure. As a result water levels are used and a correction coefficient is introduced to correct the (small) error of not taking the energy levels.

The location of the water levels are important for the stage discharge curves, as the water levels vary especially in the neighbourhood of structure. This local variation close to the structures is a result of the acceleration and retardation of the flow.

In MODIS computer package, the water levels are taken at some distance from the structure, although the computational grid points might be located close to the structure.

11.2.2 Overflow structures

■ Introduction

Overflow structures (or weir structures) are characterized by flow passing over the structure whereby the upper water surface is not touched by an obstacle such as a gate. Weirs are frequently found in irrigation schemes and are used as discharge measurements structures, discharge regulating and water level control structures. Examples of weir type structures are: division boxes, Romijn weirs and Parshall flumes.
Two type of flow conditions may occur, free flow and submerged flow (also referred to as modular and drowned flow respectively). In the latter case (submerged flow), the flow is reduced by the downstream water level.

In Fig. 11.2.1 the principle of weir flow is presented.

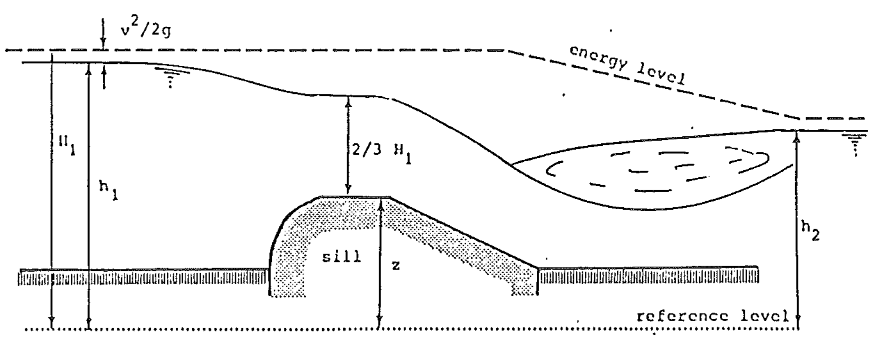


Fig. 11.2.1 Weir flow

■ Equation

The stage discharge relationship for both free and submerged flow over a weir with a rectangular control section, can be derived using an energy or impulse balance. [Ackers et al, 1978]. In literature many notations are found for the stage discharge curves of weir flow in which different coefficients are used. For watermanagement practices a simple formulae is suitable as an accuracy of 5% is often sufficient. In MODIS the

notation applied by Bos [Bos, 1976] is used as this notation is generally accepted and frequently used in irrigation practices.
The stage discharge relationship reads:

$$Q = f\, c_e \frac{2}{3} \sqrt{(\frac{2}{3} g)}\, b\, (h-z)^u \qquad 11.3$$

Where,

Q	=	discharge	m^3/s
c_e	=	effective discharge coefficient	-
f	=	drowned flow reduction factor	-
h	=	upstream water level head	m
b	=	width of the weir	m
z	=	crest level of weir	m
u	=	exponent, normally 1.5	-
g	=	acceleration due to gravity	m/s^2

In Equation (11-3) the upstream water level h is used instead of the energy level head H. The error introduced by neglecting the velocity head in the approach canal, is corrected by the factor c_v which in turn is incorporated in the overall correction factor c_e. In the above notation only dimensionless coefficients have been used.

The c_e-coefficient and the factor f are generally determined empirically and depend on factors such as: the rate of submergence, the shape of the sill crest and the ratio h/d, where h is the upstream head and d is the length of the sill crest in the direction of flow.

The value of the power coefficient u depends on the shape of the control section. For rectangular control sections u is about 1.5 while for triangular control sections u = 2.5.

■ **Functions**

The c_e and f coefficients can be given as a (tabulated) function. For example, the correction coefficient c_e can be given as a function of the upstream water level head and the drowned flow reduction factor f can be given as a function of (h2-z)/(h1-z). (Fig. 11.2.2).

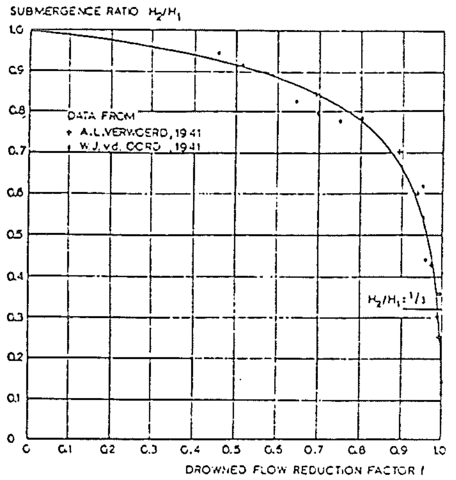


Fig. 11.2.2 *Value of drowned flow reduction factor (f) as a function of the submergence ratio (h2-z)/(h1-z).*

11.2.3 Orifice Structures

■ Introduction

Orifice flow is characterized by flow passing under gates whereby the upper water surface touches the gate and resistance is negligible. Orifice structures are frequently found in irrigation schemes and are used as discharge regulating and measurements structures and as water level control structures.

The flow through an orifice can be either free or submerged (modular or drowned). In the latter case (submerged flow) the downstream water level effects the discharge through the orifice opening and the velocity is less than the critical velocity. In Fig. 11.2.3 The characteristic of orifice flow is depicted.

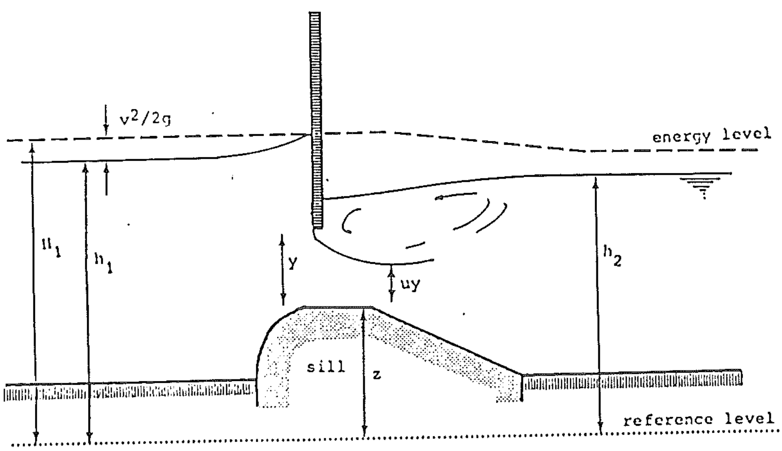


Fig. 11.2.3 Orifice flow

■ Equations

The equation for both free and submerged orifice flow with a rectangular control section reads:

$$Q = c_e \mu b y \sqrt{2 g \Delta h} \qquad 11.4$$

Where,

Q	=	discharge	m^3/s
c_e	=	effective discharge coefficient	-
μ	=	opening contraction coefficient (0.63)	-
b	=	width of control section	m
y	=	opening height of gate	m
dh	=	head difference.	m
g	=	acceleration of gravity	m/s^2

If the downstream water level does not effect the discharge (modular flow) the head difference (dh) becomes:

$$\Delta h = h_1 - (z + \mu y) \qquad .5$$

Where,

z	=	sill level	m
μ	=	opening contraction coefficient	-
y	=	gate opening height.	m

If the flow is submerged, the head difference (dh) is equal to:

$$\Delta h = h_1 - h_2 \qquad .6$$

Where h_1 and h_2 are the upstream and downstream water levels respectively.

■ Free and submerged flow

The boundary between free and submerged orifice flow can in some cases be determined theoretically but is usually determined empirically. This boundary is a function of the contraction coefficient μ and the ratio's $(h_1\text{-}z)/y$ and $(h_2\text{-}z)/y$.

In the computer model MODIS, a factor f is introduced to check which flow condition is applicable for a given upstream and downstream water level and gate opening height. The model computes at every time step the actual value of $y/(h_2\text{-}z)$.
When the actual value of $y/(h_2\text{-}z)$ is lower than the specified value of f, the flow is assumed to be submerged. When the flow is always free f

should be taken equal to zero. When on the other hand the flow is always submerged f should be taken equal to unity.

■ Weir flow

When the downstream water level is below the gate opening height and the upstream head becomes less than 1.5 times the gate opening height, the structure will act as a weir with a c_e coefficient of 1.0. Submerged weir flow is assumed when the downstream head is more than 2/3 of the upstream head. In all other cases, free weir flow is assumed. The equation for submerged weir flow reads:

$$Q = b\,(h_2 - z)\,\sqrt{2\,g\,(h_1 - h_2)} \qquad 11.7$$

■ Functions

The submerged factor f can be given as a function of the ratio $y/(h_1\text{-}z)$. (Fig. 11.2.4). The effective discharge coefficient c_e is usually taken as a constant but can be specified also as a function of the ratio $y/(h_1\text{-}z)$.

The contraction coefficient μ depends on the ratio of the gate opening height and the upstream water level head $y/(h_1\text{-}z)$ and can be specified by the user either as a constant or as a function of $(y/h_1\text{-}z)$ (Table 11.2.1)

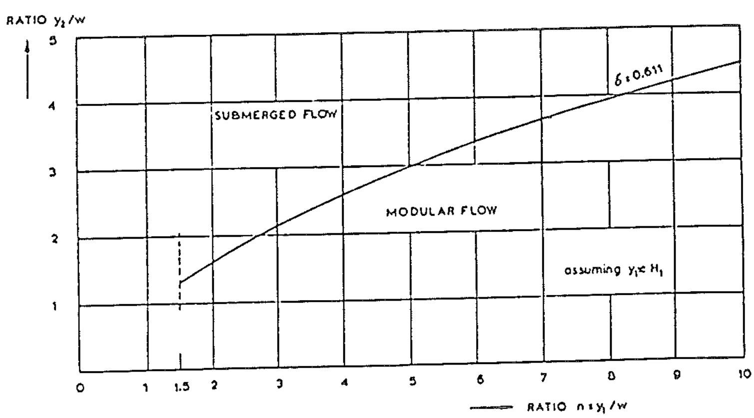


Fig. 11.2.4 Value of submerged factor f for orifice flow with a contraction coefficient μ of 0.611. [Bos, 1976]

Table 11.2.1 Value of contraction coefficient μ as a function of the upstream water level head and gate opening height (Brouwer, 1986).

y/(hl-z)	0.1	0.2	0.3	0.4	0.5	0.6
μ	0.61	0.62	0.63	0.65	0.68	0.72

11.2.4 Pipe

Introduction

Pipe flow structures with a variable cross-sectional opening area are standard included in MODIS. These structures are in principle very much the same as the orifice flow structures. Instead of a gate opening height and bottom width of control section, the wet cross-sectional area A is given. Example of this type of orifice structures are gates with a radial gate opening and culverts.

For pipe flow structures three flow conditions are distinguished; aerated free flow, normal free flow and submerged flow. The difference between aerated and normal free flow is, that in the first case the outgoing flow jet is fully aerated while in the second case (normal free flow) only the upper surface of the outgoing flow is aerated. As a result no hydrostatic pressure can be assumed for fully aerated flow, resulting in a somewhat different stage discharge curve.

The characteristics of the three flow conditions are presented in Fig. 11.2.5 a, b and c.

Equation

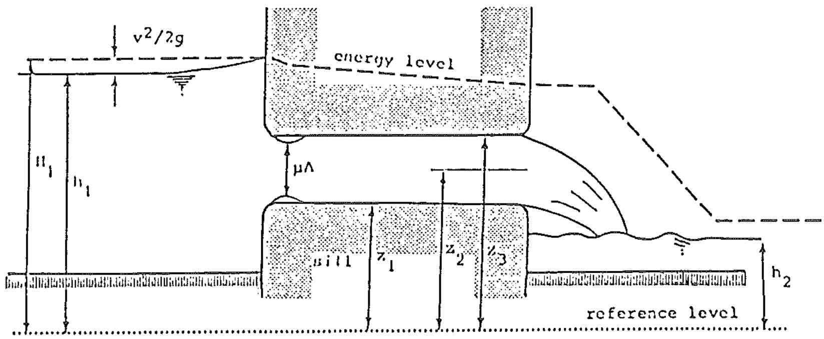
The basic equation for pipe flow reads:

$$Q = \frac{1}{\sqrt{c_t}} A \sqrt{2 g \Delta h} \qquad 11.8$$

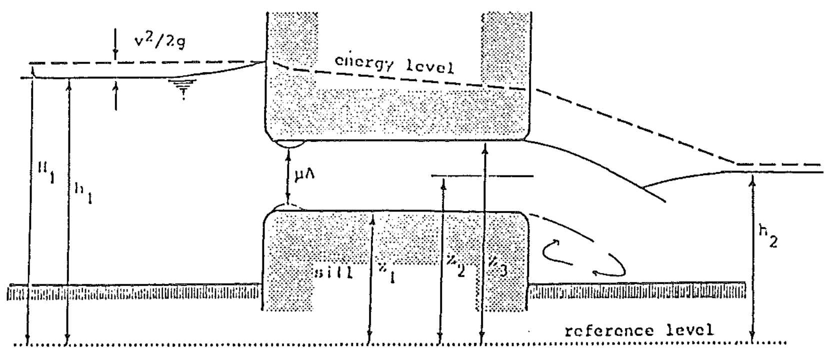
Where,

Q	=	discharge	m^3/s
c_t	=	total loss coefficient, consisting of entrance, friction and exit losses	-
A	=	cross-sectional area of gate opening	m^2
dh	=	head difference .	m

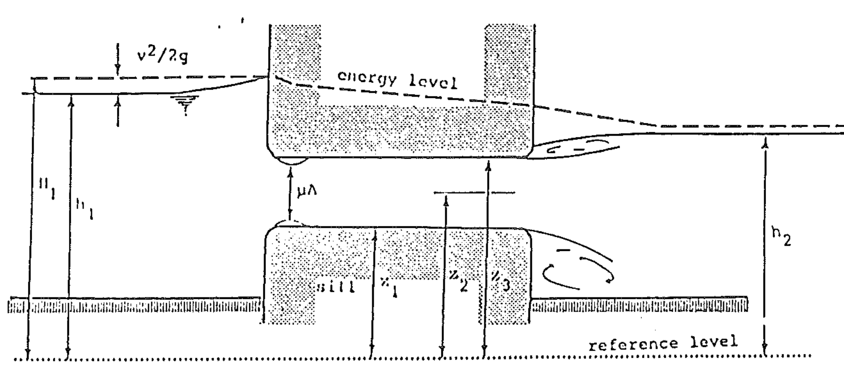
The head difference (dh) depends on the flow condition. Three flow conditions are distinguished:



a) free aerated flow



b) free flow



c) submerged flow

Fig. 11.2.5 Different flow conditions for pipe flow

■ Aerated flow

If the downstream water level is below the bottom level of the opening, the flow is fully aerated and dh becomes:

$$\Delta h = h_1 - \frac{1}{2}(h_t - h_{bottom}) \qquad 11.9$$

Where h_1 is the upstream water level, h_t is the top of opening and h_{bottom} is the bottom level of the pipe.

■ Modular flow

If the downstream water is above the bottom level of opening and below the top of opening dh becomes:

$$\Delta h = h_1 - h_T \qquad 11.10$$

■ Submerged flow

If the downstream water level is above the top level of opening, the flow is submerged and dh becomes:

$$\Delta h = h_1 - h_2 \qquad 11.11$$

Where h_2 is the downstream water level.

■ Partly filled pipe flow

When the downstream water level is below the top of opening of the pipe, the pipe can be partly filled, depending on the upstream water level. When the upstream water level is above a critical level, the pipe is assumed to be completely filled. The critical level is equal to :

$$h_{critical} = h_{bottom} + f(h_T - h_{bottom}) \qquad 11.12$$

Whereby the factor f is specified by the user. Normally f varies between 1.5 and 1.2, the latter for longer pipe lengths.

If the upstream water level is below the critical level the pipe is partly filled and the weir flow equation is applied. The flow can be either free or submerged depending on the ratio of the upstream and downstream head. When the downstream head is less than 2/3 of the upstream head, free weir flow is assumed:

$$Q = 1.7 \frac{A}{h_t - h_{bottom}} (h_1 - h_{bottom})^{1.5} \qquad 11.13$$

If the downstream head is greater than the upstream head, drowned weir flow is assumed:

$$Q = \frac{A}{h_t - h_{bottom}} (h_2 - h_{bottom}) \sqrt{2 g \Delta h} \qquad 11.14$$

■ **The loss coefficient**

The loss coefficient c_e consist of :

-entrance loss coefficient:

$$c_{entrance} = (\frac{1}{\mu} - 1)^2 \qquad 11.15$$

-resistance loss coefficient:

$$c_{resistance} = \frac{2 g L}{k^2 R^{4/3}} \qquad 11.16$$

-exit loss coefficient (1.0):

$$c_{exit} \approx 1.0 \qquad 11.17$$

Where, D = diameter of pipe, k = Strickler resistance coefficient and L = length of pipe.

■ **Functions**

The user can specify the effective discharge coefficient c_e and the contraction coefficient μ either as constants or as a function e.g. of the upstream water level. The physical properties of orifice structure the the sill level z_1, and the level of top of opening z_2 must be specified as constants.

11.2.5 Headloss structures

In canal systems all kind of head loss structures, such as bridges and inverted syphons, are frequently found. These structures create head loss due to the retardation of the accelerated flow and possibly due to the resistance of the flow passing through these structures. In MODIS these types of structures can be modelled as head loss structures.

The head loss due to abrupt profile changes can be computed using the formulae:

$$\Delta h = c \frac{v_1^2}{2g} \qquad 11.18$$

Where Q is the discharge and A is the (contracted) wet cross-sectional area. The velocity downstream v_2 is assumed to be negligible as compared to the velocity v_1. The user MODIS can specify the coefficient c and the wet cross sectional area A either as constants or as a function of other parameters.

In addition to head as a result of structures head loss might also occur as a result of a sudden increase of the cross-sectional area. In that case the head is computed as:

$$\Delta h = c \frac{(v_1 - v_2)^2}{2g} \qquad 11.19$$

Where,

dh	=	head loss	m
c	=	loss coefficient	
v_1	=	flow velocity at the vena contracta	m/s
v_2	=	flow velocity downstream	m/s

The flow velocity v_1 is the velocity at the vena contracta and equal to $v_1 = Q/A$. If the term $(v_1 - v_2)$ becomes negative, the head loss is set to zero, as no head loss occur when the flow is accelerating.

A

ll coefficients c and A can be given as constants but also as a function of time or any other function reference parameter.

11.2.6 Neyrtec distributors

■ **Introduction**

The French company ALSTHOM has developed a line of simple equipments for automatic water level and discharge control. Especially the Neyrtec (former Neyrpic) modules or baffle distributors are popular in irrigation practices. These modules maintain a nearly constant discharge irrespective of substantial upstream and downstream water level fluctuations within a specific range. The stage discharge relationships of all the Neyrtec modules have been incorporated as standard structures in MODIS. The characteristic of a Neyrtec module is given in Fig. 11.2.6.

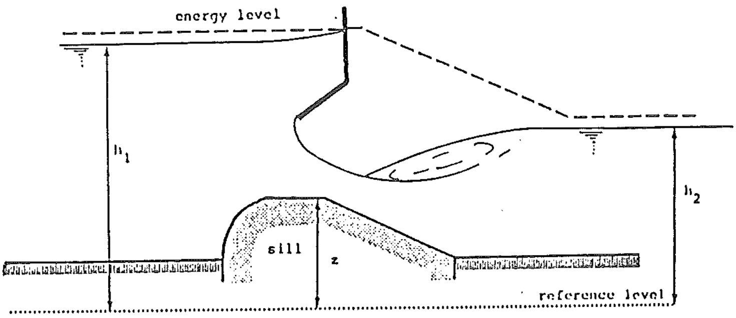


Fig. 11.2.6 Neyrtec distributor

■ **Equation**

The stage discharge relationship of the Neyrtec modules are usually given in graphs and tables. In the model MODIS, these tables are also used. For a certain water level, the corresponding discharge is calculated by means of interpolation. When the water level drops below the sill, the discharge is set to zero. When the water level exceeds the maximum allowable water level, a maximum discharge is set. In reality too high water levels will cause overtopping of the modules.

The user has to specify the sill level of the module and the length of the module. Instead of using the real length it is advised to use the effective length of the modules. Example, a real length of 0.32 m (Type :X1) has an effective length of 0.30 m as the nominal discharge is 0.30 m * 100 l/s/m = 30 l/s.

In Table 5.5.2 the different modules and their capacities have been listed. The Roman numbers (X, L et cetera) indicate the capacity in litre per second per decimeter length. The Arabic numbers (1 or 2) indicate the amount of Baffles. With two baffles the allowable range of the water level variation is higher.

■ Functions

Both structure parameters of the neyrtec distributors can be given either as a constant values or as a function of other variables such as time. By varying the width of the distributor, opening and closing of modules can be simulated. (see next paragraph)

11.3 Structure operation

11.3.1 Introduction

To facilitate operation of structures, some structure parameters such as opening height, sill level and width of control section should be variable during a simulation run. In model MODIS two types of operation are offered;

- Operation as a function of time.
- Operation as a function of a dependent variable such as the water level.

The first one is an open loop control system and the latter is a closed loop control system.

■ Open loop control
In an open loop control system the operation is predefined as a function of time and the actual state of the system is not considered. With this facility time scheduled control can be simulated such as in manual controlled systems.

■ Closed loop control
In a closed loop control system, control is based on the actual state of the system. This implies that the actual state of the system, for example a water is being read by the controller. The controller transforms the input signal into an output signal. The transformation function can be a fixed relationship between input and output signal. This is called simple closed loop control. The transformation function can also be a more complicated function such as a PID-controller. In that case one speaks of advanced closed loop control.

11.3.2 Open loop control

Open loop control of an irrigation system can be simulated by specifying the value of a structure parameter as a function of time. The model checks during simulation for each new time level the value of the structure parameters and adjust them.

11.3.3 Closed loop control

▪ Simple closed loop control

Simple closed loop control is simulated by specifying the value of a structure parameter as a function of a model variable. This can be a tabulated function or a fortran defined function. Examples of model variables are upstream water level, upstream head, downstream head. (Table 5.5.1)

▪ Advanced closed loop control

Two types of advanced closed loop controllers are at present implemented in MODIS. These are:

- a so called step-controller and
- a PID-controller.

11.3.4 Step controller

In the step controller the control variable (water level) is kept within a specified allowable range of variation. Inside this range no action is taken. When the control variable (e.g. water level) is outside this range the structure parameter (e.g. sill level) is adjusted. The rate of adjustment is constant and specified by a velocity.

The output signal of a step controller is computed by:

$$u_t = u_t + v\Delta t \qquad 11.20$$

where,

ut = output signal
z0 = constant (parm2)
v = speed of gate movement
et = deviation from set-point (= h - target level)

The set point can be variable with Q. In BIVAL control the formula reads:

$$h_{target} = h_0 + D \frac{Q_{max} - Q_{pivot}}{Q_{max}} \qquad 11.21$$

Where D is a constant and Q_{pivot} is the discharge in the pivot point.

11.3.5 PID control

The Proportional Integral Differential (PID)-controller is more advanced control mechanism. The control mechanism consists of three parts as its name indicates. Each mechanism will be explained subsequently.

■ Proportional control.

A simple control system is proportional control also referred to as P-control (more properly called proportional-positioned control). Under proportional control, the water surface varies from one end of the proportional band for no flow to the other edge of the band for maximum flow. The position of the regulator is defined by multiplying the deviation of the control variable by a gain factor K_p. This implies that the gate opening is proportional to the deviation of the water level from the set-point. The user has to specify this gain factor (K_p). In addition the set point has to be defined and the initial gate position.
The output signal is computed by:

$$u = K_p e(t) \qquad 11.22$$

Where,

u_t = controller output (signal for regulator)
e_t = deviation from set point (target value - actual value)
K_p = proportional gain factor

Proportional control are commonly found by automatic hydraulic regulators. The gate opening is proportional to the deviation from the set-point. The accuracy of a P-controller is small as deviations from the setpoint can not be avoided. For every new water level a new equilibrium position is set. A small proportional gain factor makes the controller slow, while a too high factor causes overcompensation of the regulator leading to instabilities. Proportional control is generally unacceptable for controlling gravity deliveries because the proportional band required to achieve stable flow is too large. (Zimbelman,1983)

- **Integral controller**

Sometimes a more thoughtful control is necessary, as if the controller has a sort of memory of deviations in the past. The integral controller bases its output signal not only on the present deviation, but also on the previous deviations by taken the sum of all deviations up to the present time. In this way, extraordinary deviations are damped, but, on the other hand, very small deviations can be undervalued. For example, if the actual water level has been above the target level for a considerable period of time, the sum of deviations is large positive. If the target level is almost reached, the integral controlled gate does not stabilize, but will still close relatively quick, because of the large integral. In this case, to reach the target level, the water level needs to drop below the target level to reduce the integral. This process will always cause an oscillation around the target level. If the K_I-factor is not too large, the oscillation can damp out. Characteristic for the integral controller is that it always forces the process back to its set-point.

The output signal of the integral controller reads:

$$u(t) = K_I \cdot \sum_{t=0}^{t} e(t) \qquad 11.23$$

Where K_I is the integral gain factor.

In MODIS, computer package, the sum of the deviations is not altered if the gate is in its maximum or minimum position in order to avoid too large values of the sum, which will result in a too slow response of the integral controller.

Differential controller

In case of sudden variations the PI-controlled process (a controller with a proportional and a integral factor) does not satisfy. This can be explained as follows: in the ideal situation, the regulator would immediately respond on the variation. As stated before, only the proportional factor K_P gives an output signal based on the present deviation. The integral factor K_I has the feature of damping extraordinary deviations according to history, which is now the case. The K_P-factor should be rather large to be able to respond on the sudden change. This would affect the performance of the system over a longer time, and instability might occur. That is why the K_P-factor cannot chosen as large as needed at this specific situation.
There is the need for a factor which comes out at the specific case of a sudden large variation, and which effect is negligible in more stable circumstances. The differential controller satisfies this demand with the following output signal:

$$u(t) = K_D \cdot (e(t) - e(t-1)) \qquad 11.24$$

The K_D-factor is only applied if the expected changes in water level need to be corrected very quickly. However, if the sign of the K_D-factor is chosen opposite to the sign of the K_P-factor, it serves as a retardation of the gate movement. This can be useful if very large sudden deviations are not desired to be followed immediately by a considerable change of the gate setting.

PID-controller

The PID (Proportional, Integral, Differential) controller combines three types of control techniques. If the deviation on time *t* is expressed by *e(t)*, the output signal is computed as:

$$u(t) = K_P \cdot e(t) + K_I \cdot \sum_{t=0}^{t} e(t) + K_D \cdot (e(t) - e(t-1))$$ 11.25

The determination of the best values of the three parameters of the PID controller is a process on itself, which asks for a more systematical approach. To obtain the best values, it should possible to use a program with control parameter adjustment rules, in which simplified Barré de Saint-Venant equations are used. It is also possible to use some rules derived by Ziegler and Nichols (Cool et al 1985). (Fig. 4.3.3). Since such a program is not yet available, a different approach is necessary not to get lost in too much trial-and-error. A proper understanding of the influence of each separate factor is needed to be able to gain good values.

The K_P-factor is the proportional factor. This means that the measured deviation is proportionally compensated for. Most of the cases, the K_P-factor is greater than unity, which means the gate will open more than the sensed deviation from the target level. This factor plays an important part in the gate control, since it gives a reaction to the gate as soon as a deviation is measured.

■ Summarized:

K_P: for a immediate, proportional deviation related change of the gate; usually the largest factor. The target level will never be obtained, but the water level oscillates around it.

K_I: for a memory-based control of the gate, to damp out sudden variations and thus obtain more stability and a water level equal to the target level. This factor is commonly small.

K_D: to have a quick response on sudden changes, or opposite a tamed response on sudden changes, dependant on the sign of the factor. The factor is chosen relative to the expected changes, but normally smaller than K_P.

An optimal controller has the following properties;
- quick response
- sufficient damping to avoid instabilities
- small static deviation

In fact these requirements are contradictory. Therefore, it is needed to find a compromise between the mention requirements.

Ziegler Nichols adjustment rules

- all factors are set to zero, except the K_P-factor. This factor is chosen minimal, but so that a stable oscillation is just acquired, This corresponds with the graph below. The used K_P-factor is called K_U, the ultimate gain.
- the period of the obtained oscillation is called the ultimate period T_U, and can be measured from the graph.

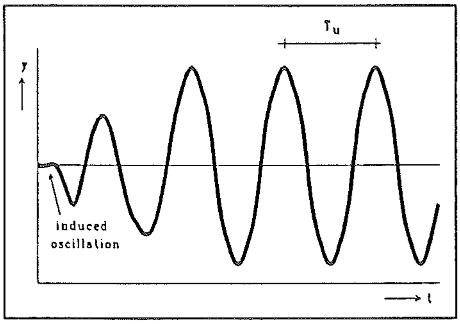


Fig. 4.3.3 Ziegler-Nichols rules

the values of the gain factors can then be derived as:

$$K_P = 0.59 \cdot K_U$$

$$K_I \approx 1.18 \cdot \frac{K_U}{T_U} \cdot \Delta t$$

$$K_D \approx \frac{0.074 \cdot T_U \cdot K_U}{\Delta t}$$

The values thus obtained can serve as a first estimation of the input for the controller, and might need to be adjusted for the specific situation.

11.3.6 Local and regional control

In general, three levels of control are distinguished; local, regional and global control. In MODIS local and regional control have been implemented. When a regulator is local controlled, control consist of one control loop, whereby the sensor is placed in the direct neighbourhood of the regulator. Regional control is an intermediate level between local and global control and is applied when measurements other than at the regulator site, are required.

■ **Local**

Standard local automatic control systems included in MODIS are upstream control, downstream control and mixed control. Automatic water level control is only possible for the weir and orifice types of structures.
In case of local upstream control, the water level in the grid point just upstream of the regulator is used as the input variable of the controller. In case of downstream control, the water level, downstream of the regulator is used. For mixed control both levels are used.

■ **Regional control**

When regional control is applied two water level sensors can be placed anywhere at a user defined grid-point. The input of the controller is weighed average of these two water levels. The weighing coefficients are α for the first gridpoint and $1-\alpha$ for the second gridpoint.

11.4 Evaluation of simulation results

11.4.1 Introduction

To compare and evaluate operation alternatives on their effectiveness, delivery performance parameters are needed which characterize the quality of water delivery. Often statistical parameters are calculated to determine the mean delivery and the variation in flow rate in time.

In MODIS non statistical parameters have been used, as those parameters are in general difficult to interpret. Instead a delivery performance parameter and a operation efficiency have been introduced. The definition of both parameters will be treated in the following. Both parameters have a value between 0% and 100%. A perfect performance is obtained when both parameters equal 100%.

The parameters are calculated on basis of the actual water supply and the intended water supply. The intended water supply is specified by the user in terms of begin and end time of supply, flow rate and lower and upper limit of flow rate. (Fig. 11.4.1).

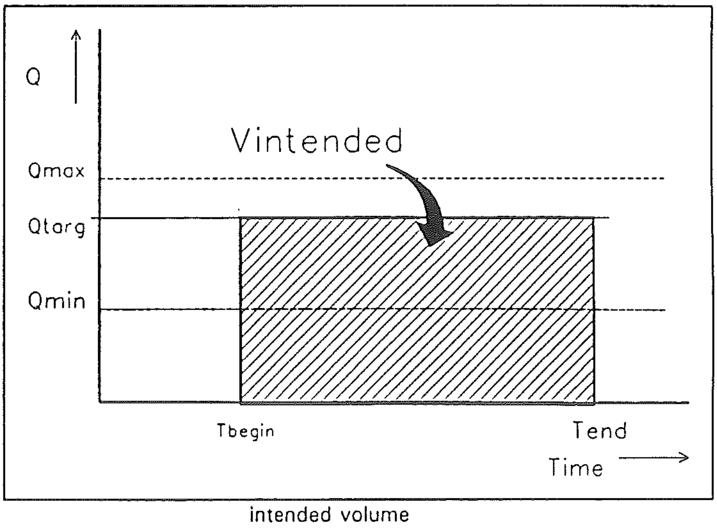
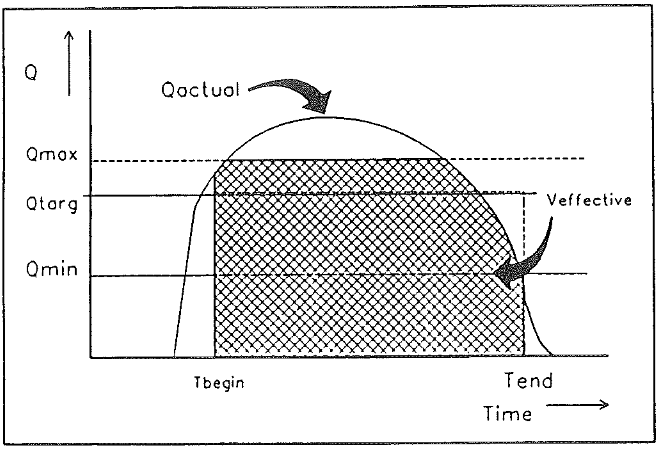


Fig.11.4.1 Definition of intended volume

By defining the intended flow rate and duration of supply, the total intended volume to be supplied, hereafter called the intended volume (Vi), is fixed. The lower and upper limit of acceptable flow rates allows for some variation in flow rates.

The actual supplied volume received by an offtake (Va) is computed by integrating the actual flow rate with respect to the specified time interval. However, the actual flow rate received by an offtake during the specified time interval might exceed the allowable range of flow rates. Therefore, the actual flow rate is considered to be an effective flow rate when the actual flow rate is within the specified boundaries. When the actual flow rate exceeds the upper boundary, the effective flow rate is taken at the upper boundary. A too high flow rate for example, might lead to spillage whereas a too small flow rate might not be handled at all.

The effective supplied volume (Ve) can now be calculated by integrating the effective flow rate with respect to the specified time interval. When the effective supplied volume exceeds the intended volume, the effective volume is taken equal to the intended volume as the additional supplied volume can not be considered to be effective, although the flow rates are within the specified boundaries. (Fig. 11.4.2)



effective volume

Fig. 11.4.2 Definition of effective volume

Now three volumes have been computed, the intended volume, the actual volume and the effective volume. With these three volumes two operation performance parameters are computed.

11.4.2 Delivery performance ratio

The delivery performance ratio (DPR) specifies to what extent the offtake has received its intended supply. The DPR is defined as the ratio of the effective volume (Ve) and the intended volume (Vi), in formulae:

$$DPR = \frac{V_e}{V_i} 100\% \qquad .29$$

11.4.3 Operation efficiency

The second parameter is the operation efficiency (e_o) which specifies with which efficiency the irrigation water is distributed. The operation efficiency is defined as the ratio of the effective supplied volume and the actual supplied volume, in formulae

$$e_0 = \frac{V_e}{V_a} 100\% \qquad .30$$

Where,

DPR	=	Delivery performance ratio
e_0	=	Operation efficiency
V_a	=	Volume actual delivered
V_e	=	Volume effectively delivered
V_i	=	Volume intended to be delivered

An operation efficiency of 0% indicates that all water supplied is lost, whereas an operation efficiency of 100% indicates that all water supplied is effectively supplied.

11.4.4 Overall performance

The overall Delivery Performance Ratio is calculated by a weighed average of the DPR of the individual offtakes and reads:

$$DPR_{overall} = \frac{\sum_{n=1}^{n=p} V_{e,n}}{\sum_{n=1}^{n=p} V_{i,n}} = \frac{\sum_{n=1}^{n=p} DPR_{,n}\, V_{i,n}}{\sum_{n=1}^{n=p} V_{i,n}} \qquad 11.31$$

Where $DPR_{overall}$ is the overall Delivery Performance Ratio, $V_{e,n}$ is the effective volume received by offtake n (m^3), $V_{i,n}$ is the intended volume to be received by offtake n (m^3), and p is the number of offtakes involved.

The overall operation efficiency incorporates the operational losses of the individual offtakes, the losses due to filling up the canal to its operational level, and the leakage losses and or spill losses if these occur. In formula it reads,

$$e_{0,overall} = \frac{\sum_{n=1}^{n=p} V_{e,n}}{V_{a,intake}} \qquad 11.32$$

Where $e_{0,overall}$ is the overall operation efficiency, $V_{a,intake}$ is the actual intake volume at the head of the main canal (m^3), $V_{e,n}$ is the effective volume received by offtake n (m^3), and p is the number of offtakes involved.

The overall delivery performance ratio becomes equal to the overall operation efficiency if the sum of intended volumes is equal to the volume taken in. In that case the overall performance of the canal system can be characterized by one single figure.

■ Interpretation of parameters

An operation efficiency of 100% does not indicate that the offtake received enough water. Therefore both parameters should be used in combination. In that way the parameters becomes diagnostic parameters as well. When for example both parameters are less than 100%, the water manager knows that there is still room for improvements, without increasing the actual water supply.

■ **General remarks**

In literature other performance parameters are found. Examples are the Operation Performance Ratio (OPR) defined by [Lenton, 1982] and used by [IIMI, 1987], [Makin,1986] and [Francis & Elawad,1989]. The Operation performance Ratio is defined as the ratio between the actual flow rate and the intended flow rate. The drawbacks of the OPR is that only the flow rate is considered and not the moment of delivery. Besides, no explicit relation is made with the irrigation efficiency.

12 Numerical solution

12.1 Introduction

The unsteady flow in open canals is described by the de Saint Venant Equations. Both the equation of continuity and the equation of momentum of the de Saint Venant equations are partial differential equations of the first order. To solve this set of two partial differential equations, the partial differentials are replaced by finite difference equations which can be solved numerically. In MODIS the transformation from differentials to finite differences is performed with a four point implicit Preissmann scheme. (Cunge, Holly, Verwey, 1979)

In the transformed equations two unknowns can be distilled for every computational grid point, namely ΔQ and Δh representing the increment during one time step Δt. The new values for Q and h at the new time level become equal to the values at the previous time level plus the increments.

When a structure is located inside a branch, the original momentum equation of that branch is replaced by the stage discharge relationship of that structure. To do so, the stage discharge relationship is rewritten into the same form as the original momentum equation. In this way the computation remains implicit and no special arrangements have to be made in the solution procedure.

In this chapter the discretization of the canal flow equations, the boundary conditions and the structure equations will be explained in detail. Furthermore the solution procedure of the unknowns is treated.

12.2 Discretizing the canal flow equations

12.2.1 General

For the solution of the system of flow equations, Verwey's variant of the Preissmann scheme has been chosen. (Cunge, Holy, Verwey, 1979). The Preissmann scheme is a four point implicit scheme, which implies that four variables can be distinguished namely ΔQ and Δh in two adjacent grid points (Fig. 12.2.1).

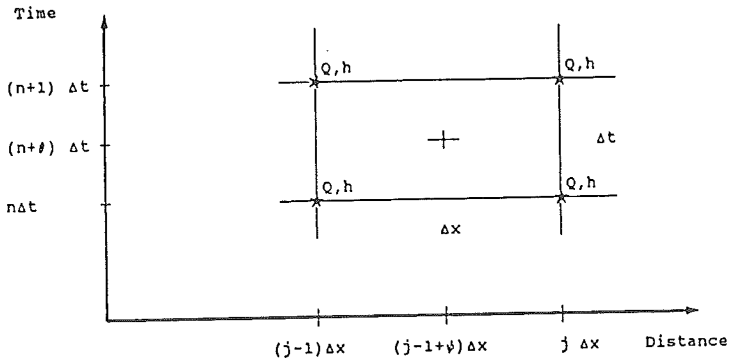


Fig. 12.2.1 Preissmann scheme

The derivatives with respect to x and t are discretized on the grid shown in Fig. (12.2.1) as follows:

$$\frac{\partial Q}{\partial x} = \frac{Q_j^n - Q_{j-1}^n}{\Delta x} + \theta \frac{(\Delta Q_j - \Delta Q_{j-1})}{\Delta x} \qquad 12.1$$

$$\frac{\partial h}{\partial t} = (1-\psi)\frac{\Delta h_{j-1}}{\Delta t} + \psi \frac{\Delta h}{\Delta t} \qquad 12.2$$

Where,

$$\Delta Q_j = Q_j^{n+1} - Q_j^n \qquad 12.3$$

$$\Delta h_j = h_j^{n+1} - h_j^n \qquad 12.4$$

n = superscript for time level

j = subscript for space location

θ = a weighting coefficient for distributing the space derivatives over the two time levels nΔt and (n+1)Δt and

ψ = a weighting coefficient defined similarly for distributing terms in space.

12.2.2 Continuity equation

The full discretized form of the continuity equation (Eq 10.1) now reads:

$$\frac{Q_j - Q_{j-1}}{\Delta x} + \theta \frac{\Delta Q_j - \Delta Q_{j-1}}{\Delta x} + (1-\psi)\, b_{s(j-1)}^{n+1/2} \frac{\Delta h_{j-1}}{\Delta t} + \psi\, b_{s(j)}^{n+1/2} \frac{\Delta h_j}{\Delta t} = q_{j-1/2}^{n+1/2} \qquad 12.5$$

Remarks

- For the coefficient b_s the best known approximation for its variation in time will be taken at the point (n+½)Δt. Evaluating the value of the coefficients at time level (n+1/2), is the main principle of Verwey's variant of the Preissmann scheme.
- The lateral flow (inflow or outflow) is specified between two grid points and therefore will be taken at point (j-½)Δx, (n+½)Δt.

Equation (12.5) can be written in a form like:

$$A1_j\, \Delta h_{j-1} + B1_j\, \Delta Q_{j-1} + C1_j\, \Delta h_j + D1_j\, \Delta Q_j = E1_j \qquad 12.6$$

by multiplication of all terms by Δx. The ABCDE coefficients become:

$$A1_j = (1-\psi)\, b_{s(j-1)}^{n+1/2} \frac{\Delta x}{\Delta t} \qquad 12.7$$

$$B1_j = -\theta \qquad 12.8$$

$$C1_j = \psi\, b_{s\,j}^{n+1/2} \frac{\Delta x}{\Delta t} \qquad 12.9$$

$$D1_j = \theta \qquad 12.10$$

$$E1_j = Q_{j-1}^{n} - Q_j^{n} + q_{j-1/2}^{n+1/2} \qquad 12.11$$

12.2.3 Momentum equation

The momentum equation (Eq 10.2) is discretized following similar principles. In the momentum equation however non-linear terms appear which have to be linearized.

In the original model Rubicon, the convective acceleration term is approximated in an explicit form. In MODIS, an implicit notation has been used distributed over the grid point j-1 and j.

The friction term is distributed over the grid points j-1 and j with weighting coefficients of $(1 - \psi)$ and ψ respectively and the product $|Q|.Q$ is written as a product of the discharge at time levels $n\Delta t$ and $(n+1)\Delta t$ respectively. It can be proven that this is the most accurate schematization of the resistance term.

The full finite difference scheme for the momentum equation is then read:

$$(1-\psi)\ \frac{\Delta Q_{j-1}}{\Delta t} + \psi\ \frac{\Delta Q_j}{\Delta t} \qquad +$$

$$\frac{\left[\left(\frac{\beta}{A}\right)^{n+1/2} Q^n\,(Q^n+\Delta Q)\right]_j - \left[\left(\frac{\beta}{A}\right)^{n+1/2} Q^n\,(Q^n+\Delta Q)\right]_{j-1}}{\Delta x} \qquad +$$

$$g\,A^{n+1/2}_{j-1/2}\left[\frac{h_j - h_{j-1}}{\Delta x} + \theta\ \frac{\Delta h_j - \Delta h_{j-1}}{\Delta x}\right] \qquad + \tag{12.12}$$

$$(1-\psi)\ \frac{g\,n^2}{(A\,R^{4/3})^{n+1/2}_{j-1}}\ |Q^n_{j-1}|(Q^n_{j-1} + \Delta Q_{j-1}) \qquad +$$

$$\psi\ \frac{g\,n^2}{(A\,R^{4/3})^{n+1/2}_{j}}\ |Q^n_j|(Q^n_j + \Delta Q_j) = 0$$

Multiplying all terms by Δx, the coefficients in the difference scheme can be written in the form:

$$A2_j\ \Delta h_{j-1} + B2_j\ \Delta\ Q_{j-1} + C2_j\ \Delta h_j + D2_j\ \Delta Q_j = E2_j \tag{12.13}$$

Where the ABCDE coefficients read:

$$A2_j = -\theta\,g\,A^{n+1/2}_{j-1/2} \tag{12.14}$$

$$B2_j = (1-\psi)\ \frac{\Delta x}{\Delta t} + (1-\psi)\ \frac{g\,n^2 |Q^n_{j-1}|\Delta x}{(A\,R^{4/3})^{n+1/2}_{j-1}} - \left(\frac{\beta}{A}\right)^{n+1/2}_{j-1} Q^n_{j-1} \tag{12.15}$$

$$C2_j = \theta\,g\,A^{n+1/2}_{j-1/2} \tag{12.16}$$

$$D2_j = \psi\ \frac{\Delta x}{\Delta t} + \psi\ \frac{g\,n^2 |Q^n_j|\Delta x}{(A\,R^{4/3})^{n+1/2}_{j}} + \left(\frac{\beta}{A}\right)^{n+1/2}_{j} Q^n_j \tag{12.17}$$

$$\begin{aligned} E2_j = &\left[\left(\frac{\beta}{A}\right)^{n+1/2} Q^n\,Q^n\right]_{j-1} - \left[\left(\frac{\beta}{A}\right)^{n+1/2} Q^n\,Q^n\right]_j - \\ &g\,A^{n+1/2}_{j-1/2}(h_j - h_{j-1}) - \\ &(1-\psi)\ \frac{g\,n^2\,\Delta x}{(A\,R^{4/3})^{n+1/2}_{j-1}}\ |Q^n_{j-1}|(Q^n_{j-1}) - \\ &\psi\ \frac{g\,n^2\,\Delta x}{(A\,R^{4/3})^{n+1/2}_{j}}\ |Q^n_j|(Q^n_j) \end{aligned} \tag{12.18}$$

Remark

For higher Froude numbers the convective acceleration term may lead to instabilities. Under these conditions it is better to use a type of diffusive wave approximation. This is achieved by giving $\beta=0$ in the input, thus neglecting the convective acceleration term.

12.3 Boundary conditions

For each canal section (that is a canal enclosed between two computational grid points) Equations (12.6) and (12.13) can be set up. Suppose a branch is discretized into N grid point and in every grid point we define two unknowns (namely ΔQ and Δh), then in total there will be 2N unknowns. However, there are only (N-1) canal sections and consequently we can only define 2(N-1) Equations.
To obtain a solution of the set of equations two additional equations are required, these additional equations are found considering the boundary conditions. A boundary condition specifies a relationship between a flow and a geometrical variable at the boundary. Examples of boundary conditions are:

- Water level given as a function of time.
- Discharge given as a function of time.
- Relationship between water level and discharge (e.g critical outflow).
- Water level compatibility (internal node)
- Given storage area in a node combined with water level compatibility

The boundary condition can also be expressed in ΔQ and Δh. A boundary condition expressed in this form reads:

$$\alpha \Delta h + \beta \Delta Q = \gamma \qquad 12.19$$

where α, β and γ are coefficients of which the values depend on the type of boundary condition. In the following the α β γ coefficients are given for the boundary conditions which have been implemented in MODIS.

12.3.1 Water level given.

In this case the α β γ coefficients become,

$$\alpha = 1 \qquad 12.20$$

$$\beta = 0 \tag{12.21}$$

$$\gamma = h^{n+1} - h^{n} \tag{12.22}$$

12.3.2 Discharge given:

In this case the α β γ coefficients become,

$$\alpha = 0 \tag{12.23}$$

$$\beta = 1 \tag{12.24}$$

$$\gamma = Q^{n+1} - Q^{n} \tag{12.25}$$

12.3.3 Q-h relationship given

The discharge Q is a function of the water level h (Q= f(h)).
In order to transform the Q-h relationship into the standard form with ΔQ and Δh as variables a function F is introduced. This function F is defined by,

$$F(Q,h) = Q(h) - Q \tag{12.26}$$

Where,

Q(h) = the discharge corresponding to water level h according to the boundary condition.

Q = Discharge computed by the model.

Ideally, the function F should always be zero. However, due to the numerical computation, some deviation from zero can be found. The variation of F(Q,h) during a time step Δt can be expressed as:

$$\Delta F = \frac{\delta F}{\delta h}\Delta h + \frac{\delta F}{\delta Q}\Delta Q \tag{12.27}$$

Where,

$$\Delta F = F^{n+1} - F^{n} \qquad 12.28$$

If we assume that F^{n+1} is equal to zero (which is not necessarily true but that is what it should be), then ΔF becomes equal to:

$$\Delta F = - F^{n} \qquad 12.29$$

The derivative of F with respect to h is computed at the time level n+½, in order to make the most accurate discretization (of second order). The order of accuracy can be computed by developing the function $F^{(n+1)\Delta t}$ and $F^{n\Delta t}$ in Taylor series using $F^{(n+½)\Delta t}$ as a reference point. Now the α β γ coefficients can be determined:

$$\alpha = \left(\frac{\Delta Q(h)}{\Delta h}\right)^{n+1/2} \qquad 12.30$$

$$\beta = -1 \qquad 12.31$$

$$\gamma = -F^{n} \qquad 12.32$$

12.3.4 Critical outflow

Critical outflow is in fact a special type of Q-h relationship described by:

$$Q(h) = \sqrt{\frac{gA^3}{T}} \qquad 12.33$$

or

$$Q^2(h) = \frac{gA^3}{T} \qquad 12.34$$

To rewrite this relationship in the required format, a similar procedure is followed and a function F is introduced. The function F reads:

$$F(Q,h) = Q^2(h) - Q^2 \tag{12.35}$$

and the α β γ coefficients become

$$\alpha = \left[\frac{gA^2}{T} \left(\frac{\Delta A}{\Delta h} - \frac{A}{T}\frac{\Delta T}{\Delta h} \right) \right]^{n+1/2} \tag{12.36}$$

$$\beta = -2 \left[Q \right]^{n+1/2} \tag{12.37}$$

$$\gamma = -F^n \tag{12.38}$$

12.3.5 Nodal point with storage

The relationship between Q and h when there is storage at a node reads:

$$\sum_{k=1}^{k_m} Q_k = A_s \frac{dh}{dt} \tag{12.39}$$

Where k_m is the amount of branches connected to the storage node and A_s is the storage area of the node. To obtain the standard "boundary condition" format Eq. 12.39 is discretized as:

$$\sum_{k=1}^{k_m} \left(Q_k + \frac{1}{2}\Delta Q_k\right) = \frac{A_s}{\Delta t} \Delta h \tag{12.40}$$

For an individual branch k connected to the storage node, the discretized equation can be written as:

$$Q_k + \frac{1}{2}\Delta Q_k = \frac{A_s}{\Delta t\, k_m} \Delta h \tag{12.41}$$

In the numerical computation a switch s_k has been introduced (which is multiplied by Q) to take account of the sign conventions used.
The value of s_k is -1 when the x-axis in channel k points toward the node and +1 when it is directed away from the node. A_s is the total storage area at the node. Applying the boundary format, the boundary condition leads to the following α β γ coefficient expressions,

$$\alpha = \frac{A_s^{n+1/2}}{k_m \Delta t} \qquad 12.42$$

$$\beta = \frac{1}{2} s_k \qquad 12.44$$

$$\gamma = -s_k\, Q_k^n \qquad 12.43$$

12.3.6 Water level compatibility

The case of water level compatibility is a special case of the previous boundary condition, whereby the storage area A_s is equal to zero. In this case the factor ½ in the coefficient ß is set to one. The α β γ coefficients read:

$$\alpha = 0 \qquad 12.45$$

$$\beta = s_k \qquad 12.46$$

$$\gamma = -s_k\, Q_k^n \qquad 12.47$$

12.4 Computation of a branch with structures

12.4.1 General

When a structure is placed in a branch, the flow in that branch is determined by the characteristics of the structure whereas the influence of the original momentum equation of the de Saint Venant equations on the flow can be neglected. Therefore the original momentum equation is replaced by the stage discharge relationship of the structure. The equation of continuity remains indifferent. To fit the stage discharge curve of the structure in the overall matrix structure (see paragraph 12.5), the stage discharge relationship is rewritten in the same form as the momentum equation,

$$A2_j\, \Delta h_{j-1} + B2_j\, \Delta Q_{j-1} + C2_j\, \Delta h_j + D2_j\, \Delta Q_j = E2_j \qquad 12.48$$

12.4.2 Determination of the ABCDE coefficients.

The stage discharge curve of a structure reads:

$$Q_{j-1} = f(h_{j-1}, h_j) \qquad 12.49$$

This equation states that the flow through a structure is a function of the upstream and downstream water level. We are searching for an equation expressed in Δh and ΔQ which gives a relationship between the variation in discharge and water levels. Therefore we introduce a function F (see also paragraph 12.3) which is defined as:

$$F(Q,h) = Q(h_{j-1}, h_j) - Q_{j-1} \qquad 12.50$$

Where,

$Q(h_{j-1}, h_j)$ = the discharge computed by the water levels and Q-h relationship

Q_{j-1} = the discharge computed by the model.

Ideally, the function F should be zero. However, due to the numerical computation, some deviation from zero can be found. The variation of F(Q,h) over a time step Δt can be expressed as:

$$\Delta F = \frac{\delta F}{\delta h_{j-1}} \Delta h_{j-1} + \frac{\delta F}{\delta h_j} \Delta h_j + \frac{\delta F}{\delta Q_{j-1}} \Delta Q_{j-1} \qquad 12.51$$

Where:

$$\Delta F = F^{n+1} - F^n \qquad 12.52$$

If we assume that F^{n+1} is equal to zero (which is not necessarily true, but that is what it should be), then ΔF becomes equal to:

$$\Delta F = - F^n \qquad 12.53$$

Hence, the ABCDE coefficients read:

$$A2 = \left(\frac{\Delta F}{\Delta h_{j-1}}\right)^{n+1/2} \qquad 12.54$$

$$B2 = -1 \qquad 12.55$$

$$A2 = \left(\frac{\Delta F}{\Delta h_j}\right)^{n+1/2} \qquad 12.56$$

$$D2 = 0 \qquad 12.57$$

$$E2 = -F^n \qquad 12.58$$

The derivatives of F with respect to the water levels are evaluated at the time level $n+½$, in order to make the most accurate discretization (of second order), which is of the same accuracy as the branch flow discretization.
The order of accuracy can be determined by developing the function $F^{(n+1)}$ and F^n in Taylor series using $F^{(n+½)}$ as a reference point.

The derivatives of F with respect to the upstream and downstream water levels are computed numerically in the model by changing the water levels 0.01 m (Δh = 0.01 m) and computing the resulting variation in F.

The advantages of a numerical computation of the derivatives are that:
- discharge coefficients which are a function of the water levels are automatically included. In this way submerged flow is computed in the model as the drowned flow reduction factor f can be given as a

function of the water levels.

- Non standard structures such as Fortran defined structures or the Neyrtec distributors for example, can easily be incorporated in the program.

The disadvantage of the numerical computation of the derivatives is that the computation time increases as compared to analytical expressions of derivatives.

When several structures are placed in parallel, then for each structure the coefficients A,B, C and E are computed and added up to obtain new coefficients A,B,C and E.

12.5 Solution procedure

When all coefficients ABCDE and α, β and γ of the boundary conditions have been determined, a system of algebraic equations in terms of unknown flow and geometrical variables at the time level $(n+1)\Delta t$ has to be solved. For the computation of the ABCDE coefficients the old values of the flow and geometrical variables are used. Since an implicit scheme has been used, all equations are coupled. Two problems occur, the first problem is related to the determination of the coefficients ABCDE and α, β and γ and the second problem concerns the question how to solve the unknowns.

12.5.1 Determination of the coefficients

The coefficients are evaluated by taking the average value of the coefficients at the old time level and the new time level. However in a first iteration the values at the new time level are still unknown and therefore the coefficients are first evaluated at the old time level only. With these coefficients the unknown flow and geometrical variables are computed. With the known Δh and ΔQ in every grid point, the new water levels and discharges can be computed in each grid point.

With these new values the coefficients are re-computed in a second iteration. The coefficients in the momentum equation are re-computed using the mean value of the flow and geometrical variables at the new time level and at the old time level, thus at $(n+\frac{1}{2})$.

In the original model Rubicon, the computation of the ABCDE values in case of structures was different. When a structure was placed in a canal section, the ABCDE- coefficients were re-computed by taking the average of the old ABCDE and the new ABCDE-values. The new values were computed, using the water levels at the new time level. However, in MODIS, the ABCDE-coefficients are computed with average values of the flow and geometrical variables.

The coefficients in the boundary conditions are re-computed using the average value of the flow and geometrical variables.

When all coefficients have been re-computed, the equations are solved again (second iteration). Practice has proved that after this second iteration, an accurate solution is obtained and no more iterations are required. (Cunge, Holly, Verwey, 1979). If a more accurate solution is desired it is more efficient to alter the time and or space steps, instead of increasing the number of iterations.

12.5.2 Solution of unknowns

For each branch the system of algebraic equations can be written into a matrix notation. This matrix has a banded structure with a band width of four.

$$\begin{pmatrix} A1_1 & B1_1 & C1_1 & D1_1 & & & & & & \\ A2_1 & B2_1 & C2_1 & D2_1 & & & & & & \\ & & A1_2 & B1_2 & C1_2 & D1_2 & & & & \\ & & A2_2 & B2_2 & C2_2 & D2_2 & & & & \\ & & & & & & A1_{jj} & B1_{jj} & C1_{jj} & D1_{jj} \\ & & & & & & A2_{jj} & B2_{jj} & C2_{jj} & D2_{jj} \end{pmatrix} X \begin{pmatrix} \Delta h_0 \\ \Delta Q_0 \\ \Delta h_1 \\ \Delta Q_1 \\ \Delta h_2 \\ \Delta Q_2 \\ \Delta h_{jj} \\ \Delta Q_{jj} \end{pmatrix} = \begin{pmatrix} E1_0 \\ E2_0 \\ E1_1 \\ E2_1 \\ E1_{jj} \\ E2_{jj} \end{pmatrix} \qquad 12.59$$

Where, the subscript 0 refers to the first grid point inside a branch and the subscript jj refers to the last grid point inside that branch. The boundary conditions have not been included in this matrix notation.

The matrix can be rewritten into a format whereby every unknown is expressed as a function of the water levels at the branch ends. The advantage of this notation is that all unknowns can be solved directly, when the water levels at the branch ends are known. The value of the water levels at the branch ends, are determined in a later stage with the help of the boundary conditions.

The required format of the matrix can be obtained efficiently using a double sweep algorithm [Vreugdenhil, 1985]. However, to apply this double sweep algorithm, the matrix should have a band width of three instead of

four. (The band width is determined by the number of unknowns in one row).
By sweeping the matrix, a band width of three can be obtained resulting in a matrix like,

$$\begin{pmatrix} a1_1 & b1_1 & c1_1 & 0 & & & & & & \\ 0 & b2_1 & c2_1 & d2_1 & & & & & & \\ & & a1_2 & b1_2 & c1_2 & 0 & & & & \\ & & 0 & b2_2 & c2_2 & d2_2 & & & & \\ & & & & a1_{jj} & b1_{jj} & c1_{jj} & 0 \\ & & & & 0 & b2_{jj} & c2_{jj} & d2_{jj} \end{pmatrix} X \begin{pmatrix} \Delta h_0 \\ \Delta Q_0 \\ \Delta h_1 \\ \Delta Q_1 \\ \Delta h_2 \\ \Delta Q_2 \\ \Delta h_{jj} \\ \Delta Q_{jj} \end{pmatrix} = \begin{pmatrix} e1_0 \\ e2_0 \\ e1_1 \\ e2_1 \\ e1_{jj} \\ e2_{jj} \end{pmatrix} \quad 12.60$$

Note: The coefficients abcde are new coefficients and have other values than the ABCDE coefficients of the first matrix. The same remark holds true for the matrixes used in the remainder of this paragraph.

Now the double sweep algorithm can be used. In the first forward sweep, starting at the top of the matrix, the coefficients below the main diagonal are eliminated and the values at the main diagonal are set to 1. The resulting matrix yields,

$$\begin{pmatrix} g_1 & 1 & c1_1 & 0 & & & & \\ g_2 & 0 & 1 & d2_1 & & & & \\ g_3 & 0 & 0 & 1 & c1_2 & 0 & & \\ g_4 & 0 & 0 & 0 & 1 & d2_2 & & \\ g_{n-1} & 0 & 0 & 0 & 0 & 1 & c1_{jj} & 0 \\ g_n & 0 & 0 & 0 & 0 & 0 & 1 & d2_{jj} \end{pmatrix} X \begin{pmatrix} \Delta h_0 \\ \Delta Q_0 \\ \Delta h_1 \\ \Delta Q_1 \\ \Delta h_2 \\ \Delta Q_2 \\ \Delta h_{jj} \\ \Delta Q_{jj} \end{pmatrix} = \begin{pmatrix} e1_0 \\ e2_0 \\ e1_1 \\ e2_1 \\ e1_{jj} \\ e2_{jj} \end{pmatrix} \quad 12.61$$

In the return sweep, the coefficients above the diagonal are eliminated, starting from the bottom. The resulting matrix then looks like,

$$\begin{pmatrix} g_1 & 1 & 0 & 0 & 0 & 0 & f_n & 0 \\ g_2 & 0 & 1 & 0 & 0 & 0 & f_{n-1} & 0 \\ g_3 & 0 & 0 & 1 & 0 & 0 & f_4 & 0 \\ g_4 & 0 & 0 & 0 & 1 & 0 & f_3 & 0 \\ g_{n-1} & 0 & 0 & 0 & 0 & 1 & f_2 & 0 \\ g_n & 0 & 0 & 0 & 0 & 0 & f_1 & 1 \end{pmatrix} X \begin{pmatrix} \Delta h_0 \\ \Delta Q_0 \\ \Delta h_1 \\ \Delta Q_1 \\ \Delta h_2 \\ \Delta Q_2 \\ \Delta h_{jj} \\ \Delta Q_{jj} \end{pmatrix} = \begin{pmatrix} e1_0 \\ e2_0 \\ e1_1 \\ e2_1 \\ e1_{jj} \\ e2_{jj} \end{pmatrix} \qquad 12.62$$

Physically the double sweep procedure can be interpreted as a procedure in which information given at one boundary is transferred to the other boundary and vice versa. As long as the flow is subcritical the direction of the sweeps with respect to the direction of flow is of no importance. The supercritical flow situation however is more complex and requires more care (Abbott, 1979).

12.5.3 Boundary conditions

The water levels at the branch ends are computed using the boundary conditions. A new matrix is set up with N rows, where N is the number of nodes. The boundary condition is expressed as:

$$\alpha \,\Delta h + \beta \,\Delta Q = \gamma \qquad 12.63$$

The ΔQ can be eliminated by substituting for ΔQ the expression, obtained in the double sweep procedure:

$$g_1 \,\Delta h_0 + \Delta Q_0 + f_n \,\Delta h_{jj} = e1_0 \qquad 12.64$$

The boundary condition expressed in water levels at both branch ends then reads:

$$(\alpha - \beta g_1)\,\Delta h_0 - \beta f_n \,\Delta h_{jj} = \gamma - e1_0 \qquad 12.65$$

For every node, a set of "boundary" equations can be defined.
For a node connected to more than one branch, a so called internal node, all boundary conditions are combined into one equation. Thereby the coefficients related to the boundary water level close to the node, are summed up in to a new coefficient. The resulting set of N equations can be written in a matrix notation whereby the band width is equal to the maximum number of branches connected to one node.

This matrix will not have a banded structure and is solved in Modis, with a Gauss elimination procedure.

When the boundary water levels are known, all water levels inside the branches and all discharges can be computed straightforward.

12.6 Operation of structures

In paragraph 12.4 the numerical computation procedure of a structure equation has been discussed. When the structure is operated, some additional remarks have to be made.

In MODIS two types of automatic control of structures are possible. In the first case, the structure parameter which is controlled, is adjusted according to a predefined function. This function can be a function of time, but also a function of a hydraulic parameter, such as the upstream head for example.

In the MODIS program this type of automatic control has been implemented as follows.

First the structure parameters are determined. When they are a function of another parameter, the value of the parameter is read from that function table or fortran function. This implies that in the first iteration step the old values at time level n are used.
For the determination of the derivatives of the structure equations, the same procedure is followed. In the first iteration step the old values are used whereas in the next iteration(s), the mean water levels are used.

In the second case, whereby the structure parameter is computed by the model (advanced closed loop control), a slightly different procedure is followed. In the first iteration step, the value of the automatically controlled structure parameter is determined by the model. This value of the structure parameter is not changed anymore during subsequent iterations or during the computation of the structure derivatives. This has been done to avoid frequent fluctuations of the structure parameters.

Appendix I References

Abbott 1979	Abbott, M.B.,"Computational Hydraulics" ,Elements of the Theory of Free Surface Flow, Pitman, London, 1979.
Bos 1976	Bos, M.G.," Discharge measurements structures", Laboratory for Hydraulics and catchment Hydrology, report no 4, 1976.
Brouwer 1990	Brouwer, Robert, Paul Ankum, Fons Nelen en Wytze Schuurmans, "Control of watermanagement systems" lecture notes, Delft University of Technology, Faculty of Civil Engineering, Department of watermanagement, 1990.
Chow 1959	Chow Ven Te, " Open-channel hydraulics" , Mc Graw Hill Book company, inc, 1959.
Cunge 1980	Cunge, J.A., F.M. Holly, A Verwey, "Practical Aspects of Computational River Hydraulics", Pitman Publishing Limited, London, 1980.
Francis 1989	Francis, Marcus R., Elawad Omer M.A., "Diagnostic investigations and rehabilitation of canals in the Geriza Irrigation Scheme, Sudan", Proceedings of Asian Regional Symposium on the modernization and rehabilitation of irrigation and drainage schemes, The Philippines, 1989.
Lenton 1982	Lenton Roberto L., "Management tools for improving irrigation performance" Lecture notes presented at the special course on "Water Management in Irrigation Systems", Central Water Commission.
Makin, 1987	Makin, I.W., "Warabandi in Punjab India", Report OD/96 of Hydraulics Research Wallingford in collaboration with Irrigation and Power Research Institute Amritsar, Punjab, India, 1987.
Preissmann 1961	Preissmann, A, "Propagation des intumescences dans les caneaux et rivières", Proceedings of the First Congress of the French Association for Computation, Grenoble, French, 1961.

Rubicon 1984 Haskoning/ D.E.S. , " Users Guide Hydrodynamic Modelling System Rubicon" , 1984.

Schuurmans 1991 Schuurmans Wytze, "Description and Evaluation of Program MODIS", Proceedings of the National Conference on Irrigation and Drainage Engineering, ASCE, Honolulu, Hawaii.

Appendix II List of symbols

A	=	area of cross section
a	=	hydraulic exponent of Bakhmeteff
A_s	=	storage surface area
b	=	canal bed width
b_s	=	storage width of canal
b	=	hydraulic exponent of Bakhmeteff
C	=	Chezy resistance coefficient
c	=	critical velocity
Cr	=	Courant number
D	=	diffusion coefficient
e	=	2.71828
exp	=	exponential function
Fr	=	Froude number (= v/c)
g	=	gravitational acceleration
H	=	mean water depth (= A/B)
h	=	water level
I	=	function build up of accumulative distribution functions
K	=	discharge coefficient
k	=	Strickler resistance coefficient (=1/n)
M	=	bed resistance term
m	=	side slope (m hor : 1 ver)
n	=	Manning resistance coefficient (=1/k)
n	=	ratio of H and H_0
P	=	cumulative distribution function
P	=	wetted perimeter of cross-section
p	=	hydraulic exponent of Bakhmeteff
Q	=	flow rate
Q	=	complementary cumulative probability function
q	=	flow rate per unit width
q	=	lateral inflow per unit length
q	=	flow rate of offtake

R	=	hydraulic radius
r	=	hydraulic exponent of Bakhmeteff
S	=	storage volume
s	=	canal bed slope
s	=	variable
T	=	response time
T	=	width of canal at water surface level
t	=	time
τ	=	dimensionless time variable
t	=	time
t'	=	time on new reference system
u	=	exponent
u	=	fluid velocity
u	=	celerity of a bore
v	=	mean fluid velocity (=Q/A)
v_1	=	velocity of lateral inflow
W	=	gravity force minus resistance
x	=	space coordinate in flow direction
x'	=	space coordinate in flow direction of new reference system
y	=	water depth according to parabola profile
Z	=	normal or Gaussian probability function
z	=	bottom level
z	=	sill level of structures
α	=	coefficient in canal routing approximation
α	=	interpolation coefficient between two succesive space steps
ß	=	Boussinesq coeffient for velocity distribution
π	=	3.14159
φ	=	celerity of diffusion or celerity of moving reference system
λ	=	coefficient in the diffusion coefficient
λ	=	dimensionless resistance factor
μ	=	variable of cumulated distribution function
ρ	=	density of water
τ	=	shear stress
τ	=	dimensionless time variable
ξ	=	dimensionless space variable
η	=	coefficient

θ	=	interpolation coefficient between two successive time levels
θ	=	angle of lateral inflow
δ	=	partial differential
d	=	normal differential
Δ	=	small difference
υ	=	time integration variable
κ	=	coefficient of diffusion celerity

Subscripts:

x	=	derivative with respect to x
t	=	derivative with respect to t
in	=	inflow
out	=	outflow
0	=	original value
1	=	new value
m	=	mean value

Appendix III User's comment