APPENDIX 1. DESCRIPTION OF THE MODELING SYSTEM FOR WATER MANAGEMENT PROBLEMS INCLUDING QUALITY AND ENERGY

Contents

Introduction

The program system is designed by specialists of EPIC in cooperation with specialists of BVO Syrdarya and the technical group from the Central Asian countries.

The program system is intended for automated creating a file of the GAMS model (file with the **.gms** extension) that can be compiled by the GAMS system. The GAMS solver (a package of GAMS programs that does mathematical calculations) can carry out optimization calculations for the model, and then display calculation results to the user in a format convenient for the user.

The description is arranged as follows:

- In the first chapter, we give the description of a test problem, for which a file of the GAMS model (file with the **.gms** extension) will be created and an optimization calculation for operation of a river network will be carried out with the help of the GAMS technology.

- After that, we present a GAMS file automatically created for our test problem using the program system (file with the **.gms** extension). We also specify the modules of the program system that prepare information (form special files) for each block of this file (file with the **.gms** extension).
- Further, we consider operation of each module incorporated in the program system and describe the structure of all output files forming in the process of their execution. The description of modules is given in the sequence they should be executed when a GAMS file is created (file with the **.gms** extension).

1. Example Problem from the Syrdarya Basin

As a test problem, we are offered to calculate optimal releases from reservoirs for a reach of Syrdarya including:

- 2 reservoirs;
- Inflow to each of reservoirs;
- Tributary inflow downstream from the reservoir;
- 4 water diversions;
- 7 points of control over water consumption in the river;
- 7 points of the confluence of tributaries and the river or points of water diversion on the river;
- 1 point of calculation of channel losses;
- 1 point of calculation of flow lag time;
- 1 point of the river mouth.

Besides, in the test problem we intended to take into account reservoir and channel losses.
The scheme of the calculated reach is shown on Figure 1.

Below we enumerate common names of every entity of the river reach, for which we carried out test calculation. These common names will be further named external names of entities:

1. Inflow to Andijan Reservoir
2. Losses from Andijan Reservoir
3. Andijan Reservoir
4. The point of control over releases from Andijan Reservoir
5. The point of water diversion into the feeding canal
6. Feeding canal
7. Karabagish measurement station
8. The imputed point of the confluence of the river and tributary inflow on the Andijan – Uchtepa reach
9. Tributary inflow on the Andijan – Uchtepa reach
10. The point of water diversion on the Andijan – Uchtepa reach
11. Water diversion on the Andijan – Uchtepa reach
12. Uchtepa measurement station
13. Inflow to Toktogul Reservoir

14. Losses from Toktogul Reservoir
15. Toktogul Reservoir
16. The point of control over releases from Toktogul Reservoir
17. The imputed point of confluence of the river and tributary inflow on the reach Toktogul – Uchkurgan water structure
18. Tributary inflow on the reach Toktogul – the Uchkurgan water structure
19. The point of water diversion 1 on the reach Uchkurgan water structure– Uchkurgan measurement station
20. Water diversion 1 on the reach Uchkurgan water structure– Uchkurgan measurement station
21. The imputed point of channel losses
22. Channel losses
23. Uchkurgan measurement station
24. The point of water diversion 2 on the reach Uchkurgan water structure– Uchkurgan measurement station
25. Water diversion 2 on the reach Uchkurgan water structure– Uchkurgan measurement station
26. Uchkurgan water structure
27. The point of confluence
28. Control point
29. The point of calculation of channel losses
30. The point of calculation of lag time
31. River mouth

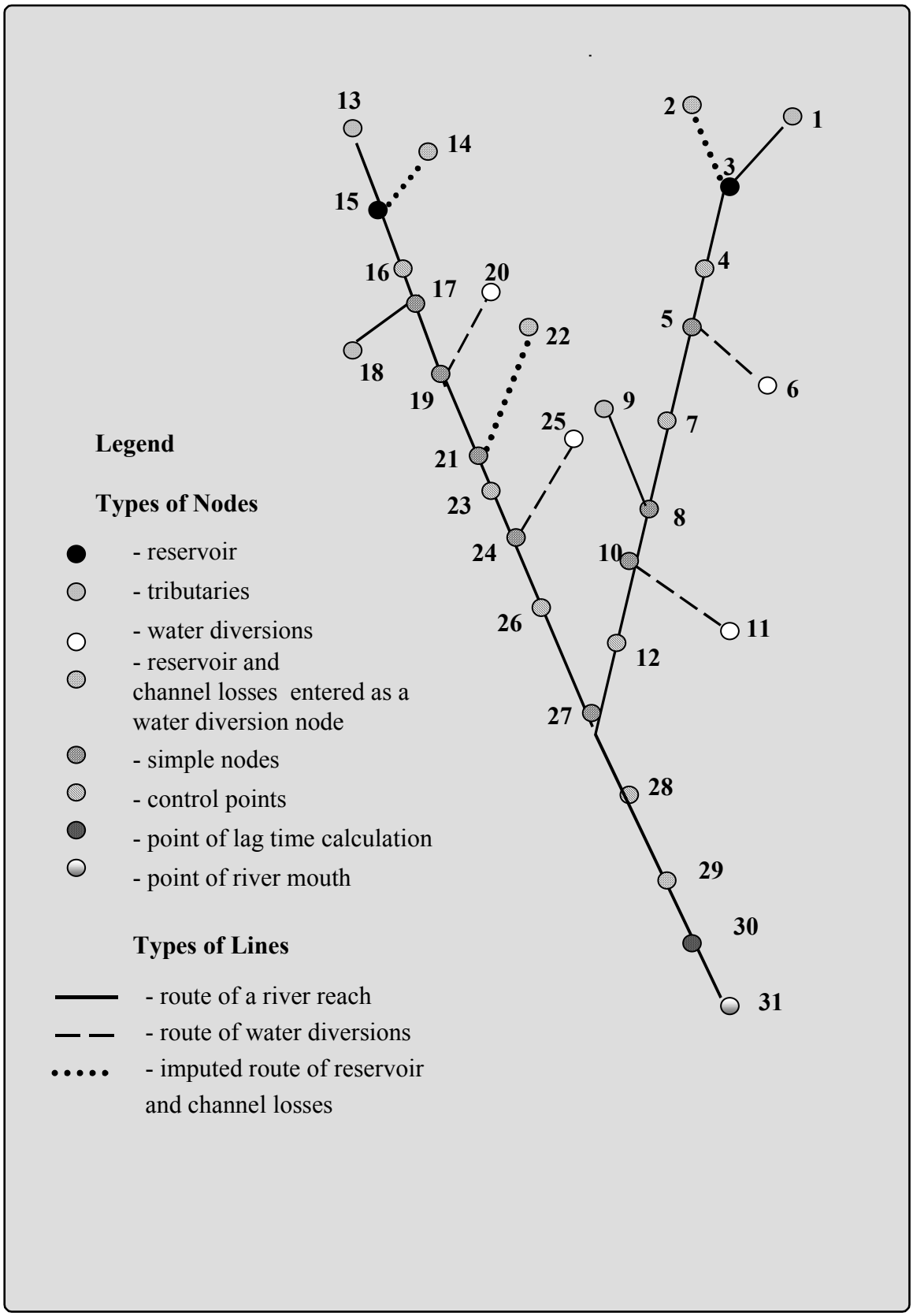


Figure 1. The Schematic of a Test Reach of the Syr Darya River.

List of Reference Data in the Test Problem

Entities	Reference Data
Reservoirs	• Initial volume for each reservoir • Water levels and respective reservoir volumes and water tables (curves of volumes and water tables) • Constraints (upper and lower) of storage for each reservoir • Releases from reservoirs in each time step (not necessary) • Final volume of each reservoir (not necessary) • In this problem, the issue of salinity is not considered. It is necessary, however, to introduce initial water salinity of the reservoir, if you take into account salinity
Tributaries	• Consumption (flow) for each time step on each tributary • In this problem, the issue of salinity is not considered. It is necessary, however, to introduce water salinity, if you take into account salinity
Water Diversions	• Water requirements (consumption or flow) for each time step
Evaporation	• In our test problem evaporation is assigned for the entire calculation scheme and can be edited in the model itself for each entity apart
River Mouth	• The test problem provides for neither a requirement nor a constraint on maximum and minimum (sanitary release) water flows

In this tutorial, we show all steps of creating a GAMS model (file with the **.gms** extension). Therefore, in the test problem we considered two options of assigning releases from the reservoir:

- Releases from Toktogul Reservoir have fixed values;
- Releases from Andijan Reservoir do not have fixed values;

Therefore, our test problem can be considered as both an **optimization and simulation** problem. If we remove fixed values of water releases from reservoirs and constraints on water flows in the river mouth, then the problem becomes an **optimization** problem.

If we assign fixed values of water releases from all reservoirs and introduce fixed values of water flows in the river mouth, then the problem becomes a **balance** problem.

Below, we give numerical values for all initial parameters.

Inflow Values (cu m/s)

External Names of Entities (Nodes)	Time Steps					
	M1	M2	M3	M4	M5	M6
Inflow to Andijan Reservoir	164	457	648	452	196	69
Tributary Inflow on the Kara Darya River	140	177	207	159	114	97
Inflow to Toktogul Reservoir	272	508	1224	1312	760	475
Tributary Inflow on the Toktogul – Uchkurgan Water Structure Reach	228	298	190	96	188	177

Water Diversions from the River (cu m/s)

External Names of Entities (Nodes)	Time Steps					
	M1	**M2**	**M3**	**M4**	**M5**	**M6**
Losses from Andijan Reservoir	10	10	10	10	10	10
Water Diversion to the Feeding Canal	43	86	119	214	188	76
Water Diversion on the Andijan – Uchtepa Reach	74	73	112	142	122	44
Losses from Toktogul Reservoir	10	10	10	10	10	10
Water Diversion on the Uchkurgan Water Structure – Uchkurgan Measurement Station Reach	178	177	192	256	259	197
Channel Losses	1	1	1	1	1	1
Water Diversion 2 on the Uchkurgan Water Structure – Uchkurgan Measurement Station Reach	20	30	20	30	30	25

Fixed Values of Water Releases from Toktogul Reservoir (cu m/s)

External Names of Entities (Nodes)	Time Steps					
	M1	**M2**	**M3**	**M4**	**M5**	**M6**
Releases from Toktogul Reservoir	223	165	147	288	332	244

Fixed Values of Storage of Reservoirs

External Names of Entities	Time Steps							
	Initial Storage	M1	M2	M3	M4	M5	M6	Final Storage
Andijan Reservoir	633	-	-	-	-	-	-	1097
Toktogul Reservoir	7247	-	-	-	-	-	-	10728

Conversion Coefficients for the Monthly Time Step

Parameters	Time Steps					
	M1	M2	M3	M4	M5	M6
Coefficients to Convert Water Consumption (cu m/s) into Monthly Water Flow (million cu m/s)	2,59	2,68	2,59	2,68	2,68	2,59

Measured Water Levels and Respective Reservoir Volumes and Water Tables

No. of Measurement	Andijan Reservoir		Toktogul Reservoir	
	Level	Volume	Level	Volume
1	**800**	**0**	**759**	**0**
2	**854**	**100**	**765**	**65,4**
3	**855**	**200**	**770**	**108**
4	**860**	**300**	**800**	**2065**
5	**867**	**400**	**820**	**3000**
6	**871**	**500**	**825**	**3647**
7	**876**	**600**	**830**	**4348**
8	**879**	**700**	**835**	**5104**
9	**885**	**800**	**850**	**7717**
10	**893**	**1000**	**855**	**8690**
11	**898**	**1200**	**860**	**9706**
12	**902**	**1300**	**865**	**10764**
13	**904**	**1400**	**870**	**11862**
14	**905**	**1450**	**875**	**13003**
15	**907**	**1470**	**878**	**13708**
16	**909**	**1500**	**880**	**14188**
17	**910**	**1600**	**883**	**14924**
18	**911**	**1700**	**890**	**16716**
19	**912**	**1800**	**895**	**18061**
20	**914**	**2000**	**900**	**19458**
Note	**b = 3.88** **a = $2.7*10^{-6}$** **h = 77.92**		**b = 3.37** **a = $13.32*10^{-6}$** **h = 122.38**	

2. Interface

The main control program of the program system is the **menu_r.exe** interface program.
The working window of the interface is shown on Figure 2.

The interface is made on the following principle: on the left there are buttons of executable modules located in the sequence these modules are run to create a GAMS model, carry out calculation, and obtain results in Russian. That is if you run all executing modules sequentially, you will get a calculation result in Russian.

Using this package, you can carry out optimization calculations with regard to salinity. For that, you have the special **Salinity Consideration Block**. Considering salinity, all operations on making a calculation diagram, entering data, assigning constraints and priorities are performed in the main block. Making a GAMS model (file with the **.gms** extension), calculating, and translating into Russian, you should also use this **Salinity Consideration Block** to make a model for optimal management of the water and salt flow regime in a river basin.

Using this package, you can carry out optimization calculations taking into account production, consumption, and transmission of power. For that you may apply a special version of interface, in which we replaced the **Salinity Consideration Block** with the **Energy Indicators Block**.

When we take into consideration the energy factor, all operations associated with constructing a calculation scheme, entering data, assigning constraints and priorities are performed in the main block. Making a GAMS model (file with the **.gms** extension), calculating, and translating into Russian, you should also use this **Energy Indicators Block** to make a model for optimal management of the water and salt flow regime in a river basin.

To start any module you need to move the mouse pointer onto the button and press the left mouse button.

The button of each executable module (***.exe**) is connected to a block with buttons to view output files obtained as a result of executing the module.

On the lower part of the interface, there is the **EPIC & BVO Syrdarya** button. If you point to this button and press the left mouse button, you will see a window with a list of implementers of this program. To return to the interface press the **Esc** key on the keyboard.

In order to familiarize himself with the interface structure, the user may see it on Figure 3.

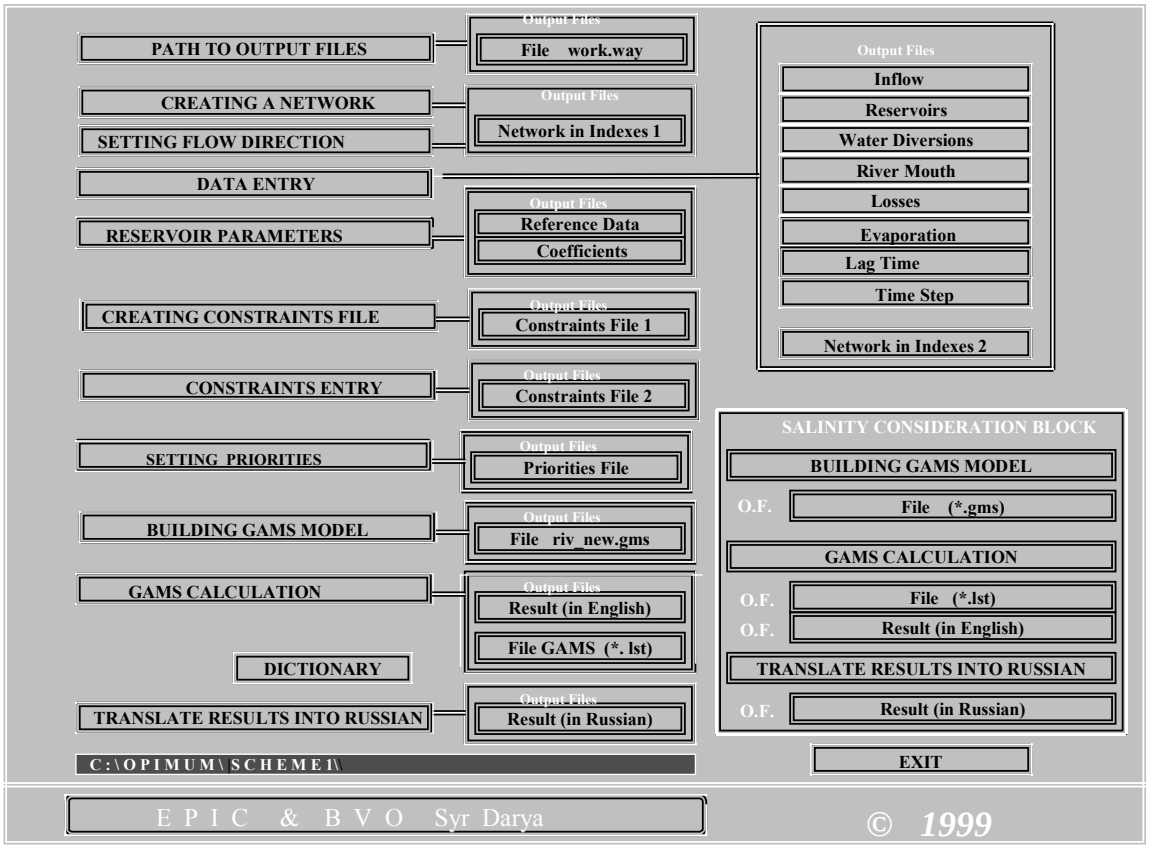


Figure 2. The Image of the Interface on the Screen

3. GAMS Model File

We created the GAMS model (file with the **.gms** extension) given below applying a special program, which uses the results of executing all program modules of the interface. The description of these modules will follow. We made the model using the data of our test problem.

Framed are individual blocks of the GAMS model (file with the **.gms** extension), and before each block we specified the data of what program module the selected block used.

In the block we used data from executing the **CREATING A NETWORK** module. This module sets a general number of nodes, node types, and a number of nodes of each type. In addition, it sets connections between nodes in arcs regardless of direction.

```
SETS nt
/K1,K2,K3,K4,K5,K6,K7,K8,I1,I2,I3,I4,V1,V2,U1,U2,U3,U4,U5,U6,U7,R1,C1,
C2,C3,C4,C5,C6,C7,P1,T1,total /;
SETS n1(nt)
/K1,K2,K3,K4,K5,K6,K7,K8,I1,I2,I3,I4,V1,V2,U1,U2,U3,U4,U5,U6,U7,R1,C1,
C2,C3,C4,C5,C6,C7,P1,T1,total /;
```

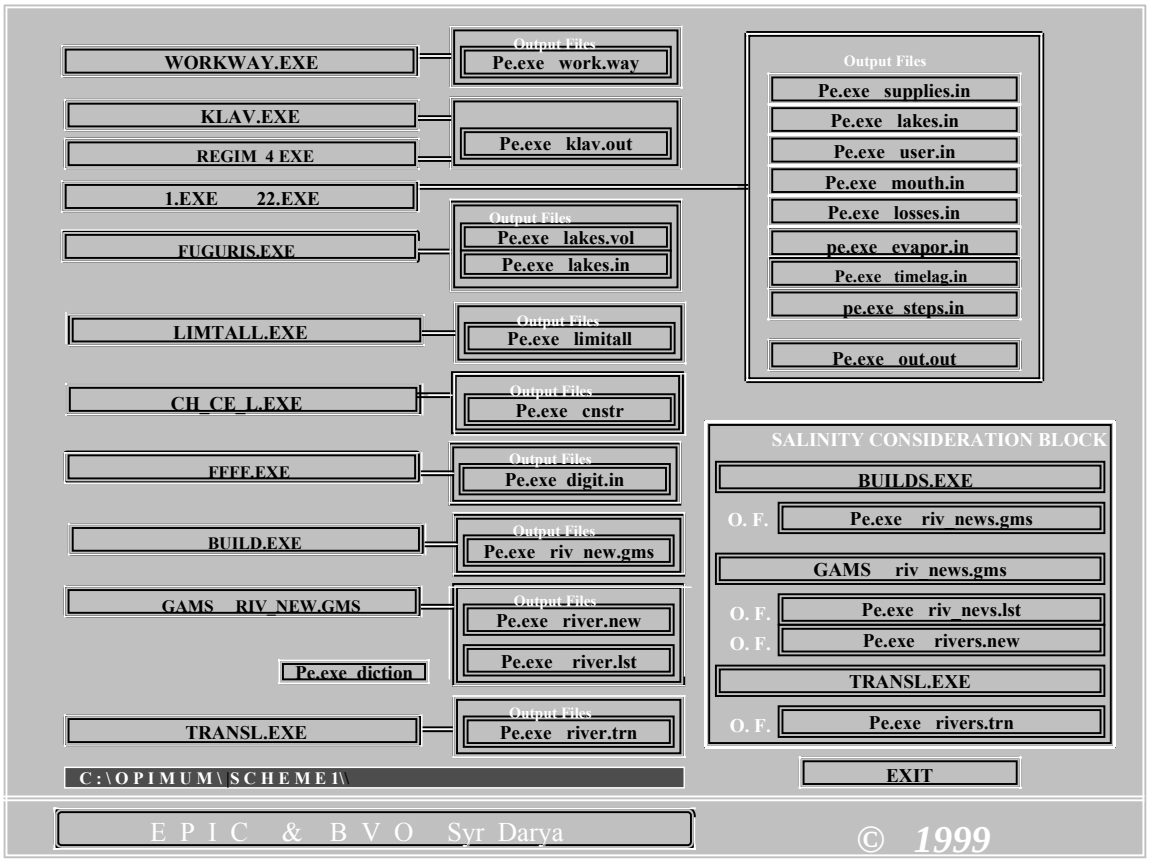


Figure 3. The Image of the Interface on the Screen

The block is created directly by the **BUILDING GAMS MODEL** module based on the **klav.out** file.

```
SETS i(n1) supplies          /    I1,I2,I3,I4          /;
SETS r(n1) users                        /  U1,U2,U3,U4,U5,U6,U7 /;
SETS rl(n1) mouth                       /  R1 /;
SETS   k(n1)   simple+control                                                    /
K1,K2,K3,K4,K5,K6,K7,K8,C1,C2,C3,C4,C5,C6,C7 /;
SETS c(n1) control and supplies         /I1,I2,I3,I4,C1,C2,C3,C4,C5,C6,C7  /;
SETS T(n1) lag_time                     /  T1 /;
SETS P(n1) loses                        /  P1/;
SETS V(n1) lakes                        /  V1,V2 /;
PARAMETERS  V_BEG(N1)                   /  V1     663.00, V2     7247.00 /;

SETS duga
      /
U1_V1,      V1_I1,      C1_V1,      K1_C1,      U2_K1,      C2_K1,      K2_C2,
K2_I2,      K3_K2,      U3_K3,      C3_K3,      K8_C3,      V2_I3,      U4_V2,
C4_V2,      K4_C4,      K4_I4,      K5_K4,      U5_K5,      K6_K5,      U6_K6,
C5_K6,      K7_C5,      U7_K7,      C6_K7,      K8_C6,      C7_K8,      P1_C7,
```

```
T1_P1,        R1_T1,        outside                                                    /;

Set  conec_du(duga,N1)                                                                 /
U1_V1.U1,     V1_I1.V1,     C1_V1.C1,     K1_C1.K1,     U2_K1.U2,
C2_K1.C2,     K2_C2.K2,     K2_I2.K2,     K3_K2.K3,     U3_K3.U3,
C3_K3.C3,     K8_C3.K8,     V2_I3.V2,     U4_V2.U4,     C4_V2.C4,
K4_C4.K4,     K4_I4.K4,     K5_K4.K5,     U5_K5.U5,     K6_K5.K6,
U6_K6.U6,     C5_K6.C5,     K7_C5.K7,     U7_K7.U7,     C6_K7.C6,
K8_C6.K8,     C7_K8.C7,     P1_C7.P1,     T1_P1.T1,     R1_T1.R1      /;
Set  conec_ud(N1,duga)                                                                 /
V1.U1_V1,     I1.V1_I1,     V1.C1_V1,     C1.K1_C1,     K1.U2_K1,
K1.C2_K1,     C2.K2_C2,     I2.K2_I2,     K2.K3_K2,     K3.U3_K3,
K3.C3_K3,     C3.K8_C3,     I3.V2_I3,     V2.U4_V2,     V2.C4_V2,
C4.K4_C4,     I4.K4_I4,     K4.K5_K4,     K5.U5_K5,     K5.K6_K5,
K6.U6_K6,     K6.C5_K6,     C5.K7_C5,     K7.U7_K7,     K7.C6_K7,
C6.K8_C6,     K8.C7_K8,     C7.P1_C7,     P1.T1_P1,     T1.R1_T1      /;
set DUGA1(DUGA);
LOOP(N1,DUGA1(duga)$(conec_DU(duga,N1) and C(n1))=yes;);
Set  conec_udp1(N1,duga) /total.outside/ ;
Set  conec_UDP2(duga,N1) /outside.TOTAL/ ;
Set  conec_UDP3(DUGA)  ;
Set  conec_UDP(N1,DUGA)  ;
conec_udp1(N1,duga)=yes$(conec_ud(N1,duga) and v(n1));
conec_udp2(duga,N1)=yes$(conec_du(duga,N1) and c(n1));
conec_udp(N1,duga)=yes$(conec_UDP2(duga,N1) OR conec_udp1(N1,duga)) ;
PARAMETER FFF(N1,DUGA),FUF(DUGA);
FFF(N1,DUGA)=0;
FFF(N1,DUGA)=1$conec_udp(N1,duga);
FUF(DUGA)=SUM(N1,FFF(N1,DUGA));
LOOP(DUGA, IF (FUF(DUGA)>1, conec_UDP3(DUGA)=YES););
parameter p1,p2,p3,p4,P5;

*************************************************************************
******
$INCLUDE 'C:\GUIDE\DIGITS.IN';
*************************************************************************
**********
SETS s(n1)        /  K1,K2,K3,K4,K5,K6,K7,K8,
                     C1, C2,C3,C4,C5,C6,C7,P1,T1 /;
SETS mt        /M1,    M2,    M3,    M4,    M5,    M6,total     /;
SETS m(mt)     /M1,    M2,    M3,    M4,    M5,    M6           /;
```

In the block we used data from executing the **SETTING FLOW DIRECTION** module and the **DATA ENTRY** module. The first module sets connections between nodes (or direction of water

flow in the calculation scheme); the second module defines the ordinal number of node of each type with the whole set of the data incident to the node.

In this block we distinguished the **INCLUDE** operator. This operator defines the place to connect the file containing priorities of subtasks included in the criteria and objective function. We describe this file as we describe the operation of the optimization task priorities module.

```
PARAMETERS   LAGTIME(n1)      /  T1   1.0000 /;
PARAMETERS   LLxKP(n1)        /  P1   1.0000 /;
PARAMETERS   en_up(n1)  /  V1 1000.0000,     V2 10000.0000   /;
PARAMETERS   en_lo(n1)  /  V1      0.0000,     V2       0.0000   /;
PARAMETERS   H_0(n1)    /  V1    799.0000,     V2    757.9200   /;
PARAMETERS   kpda(n1)   /  V1      0.0000,     V2       0.0000   /;
PARAMETERS   kpdb(n1)   /  V1      0.0000,     V2       0.0000   /;
PARAMETERS   kpdc(n1)   /  V1      0.8000,     V2       0.8500   /;
PARAMETERS   k_nba(n1)  /  V1      0.0000,     V2       0.0000   /;
PARAMETERS   k_nbb(n1)  /  V1      0.0000,     V2       0.0000   /;
PARAMETERS   k_nbc(n1)  /  V1   765.4900,     V2    735.0000   /;
PARAMETERS   a11(n1)      / V1   3.2300,           V2   2.1900 /;
PARAMETERS   b11(n1)      / V1   0.0001,           V2   0.3800 /;
PARAMETERS aa(n1),bb(n1);
  bb(n1)$(a11(n1)):=((a11(n1)-1)/a11(n1))$(a11(n1));
  aa(n1)$(a11(n1)):=(a11(n1)*b11(n1)/(b11(n1)**bb(n1)))$(a11(n1));
PARAMETERS cc(n1), dd(n1)  ;
 cc(n1)$(a11(n1)):=(1/b11(n1))**(1/a11(n1) );
 dd(n1)$(a11(n1)):=1/a11(n1);
PARAMETERS   hh(n1)    / V1 799.0000,V2 757.9200 /;
PARAMETERS   kpd(n1)   / V1   1.0000,V2   1.0000 /;
PARAMETERS   ee(m)     / m1  0.0400,m2  0.0400,m3  0.0400,m4  0.0400,m5  0.0400,
m6   0.0400 /;
PARAMETERS  time(m)    / m1   2.5900,m2 2.6800,m3  2.5900,m4  2.6800,m5  2.6800,
m6   2.5900 /;
PARAMETERS   ret(n1)       / U1   0.0,U2   0.0,U3   0.0,U4   0.0,U5   0.0,U6   0.0,U7
0.0 /;
PARAMETERS   reb(n1)       / U1   0.0,U2   0.0,U3   0.0,U4   0.0,U5   0.0,U6   0.0,U7
0.0 /;
scalar cek /10/;
table supl(n1,m)
          m1           m2          m3          m4          m5
i1      164.000    457.000    648.000     452.000    196.000
i2      362.600    474.360    536.130     426.120    305.520
i3      704.480   1361.440   3170.160    3516.160   2036.800
i4      590.520    798.640    492.100     257.280    503.840
    +      m6
i1        69.000
i2       251.230
```

```
i3      1230.250
i4       458.430   ;
table reqr(n1,m)
              m1          m2       m3          m4       m5          m6
u1         25.903     26.803    25.903     26.803    26.803     25.903
u2        111.370    230.480   308.210    573.520   503.840    196.840
u3        191.660    195.640   290.080    380.560   326.960    113.960
u4         25.900     26.800    25.900     26.800    26.800     25.900
u5        461.020    474.360   497.280    686.080   694.120    510.230
u6          2.590      2.680     2.590      2.680     2.680      2.590
u7         51.800     80.400    51.800     80.400    80.400     64.750   ;
table reqm(n1,m)
           m1        m2        m3        m4        m5
r1        20.000    20.000    20.000    20.000    20.000
    +      m6
r1        20.000 ;
PARAMETER BACK(N1,M);  BACK(N1,M)=1;
LOOP(N1,LOOP(M,IF (reqm(n1,m)<0,BACK(N1,M)=-1);););
REQM(N1,M)=ABS(REQM(N1,M));
    reqm(n1,m) = reqm(n1,m)*1000;
    reqr(n1,m) = reqr(n1,m)+reqm(n1,m);
$INCLUDE 'C:\MOE\MODEL\MODEL_WR\GUIDE\equation';
```

The block is created directly by the **BUILDING GAMS MODEL** module.

In the block we used data from executing the **SETTING FLOW DIRECTION** module. The module sets links between nodes (or direction of water flow in the calculation scheme).
Data from executing modules **DATA ENTRY** and **PARAMETERS OF RESERVOIR** that set numerical values for initial information used in calculation are used in the block.

In this block we distinguished the **INCLUDE** operator. This operator defines the place to connect the file containing the most stable part of the model – a list of equations, variables, and sequence of data output. On the next page, you can see the composition of this model block singled out in the separate file **EQUATION**.

This dissection of the general model body into parts allows the user to study and modify the calculation model block for various river systems. The fact is that this file receives an update only at will of the user. It means that as the user updates the model, the modifications he entered before will not be lost, unless the user wishes it.

```
POSITIVE VARIABLES
 flow(duga,m), tranzit(N1,m), power(N1,m), vol(N1,m);
 VARIABLES
 knot(N1,m),  obj;
 EQUATION
 first(N1,m),  second(N1,m),  third_1(N1,m),   third_2(N1,m),   power_prod(n1,m),
 forth(N1,m),    fifth(N1,m),     sixth(N1,m),  addition(N1,m),   ben;
*      MODEL DESCRIPTION BEGIN  *
fifth(N1,m)$p(N1)..
sum(duga$conec_ud(N1,duga),
flow(duga,m))=e=sum(duga$conec_du(duga,N1),flow(duga,m))*
                                                (1-llxkp(N1)*ee(m));
sixth(N1,m)$t(N1)..
 sum(duga$conec_ud(N1,duga),        flow(duga,m))=e=        sum(duga$conec_du(duga,N1),
flow(duga,m--1))*
 lagtime(N1)+ sum(duga$conec_du(duga,N1), flow(duga,m))*(1-lagtime(N1));
 power_prod(n1,m)$V(N1)..
 power(N1,m)  =e=time(m)/3600*  9.81*  (kpd(n1)*0.5*cc(N1))*  ((vol(N1,m)+vol(N1,m-
1)$(ORD(M) gt 1)+
                    v_BEG(N1)$(ORD(M) EQ 1))**dd(N1)+hh(n1) - (k_nba(n1)*

sum(duga$conec_udp3(duga),flow(duga,m))*sum(duga$conec_udp3(duga),flow(duga,m))/
                    time(m)/time(m)+k_nbb(n1)*
sum(duga$conec_udp3(duga),flow(duga,m))/time(m)
                    +k_nbc(n1)))* sum(duga$conec_udp3(duga),flow(duga,m));
forth(N1,m)$v(N1)..
 sum(duga$conec_du(duga,N1),         flow(duga,m))-         sum(duga$conec_ud(N1,duga),
flow(duga,m))=e=
 vol(N1,m)-vol(N1,m-1)$(ORD(M) gt 1)- v_BEG(N1)$(ORD(M) EQ 1)-
 (ee(m)*0.5*aa(N1))*  ((vol(N1,m)+vol(N1,m-1)$(ORD(M) gt 1)+  v_BEG(N1)$(ORD(M)
EQ 1))**bb(N1));
first(N1,m)$k(N1)..
 sum(duga$conec_ud(N1,duga),        flow(duga,m))=e=        sum(duga$conec_du(duga,N1),
flow(duga,m));
second(N1,m)$i(N1)..
sum(duga$conec_ud(N1,duga),                flow(duga,m))=e=                supl(N1,m)+
sum(duga$conec_du(duga,N1), flow(duga,m));
third_1(N1,m)$r(N1)..
sum(duga$conec_ud(N1,duga),                                flow(duga,m))=e=tranzit(N1,m)+
(sum(duga$conec_du(duga,N1),
                                                flow(duga,m))-
tranzit(N1,m))*ret(N1)+reb(n1);
third_2(N1,m)$r(N1)..
     sum(duga$conec_du(duga,N1),flow(duga,m))-tranzit(N1,m)=l=reqr(N1,m);
addition(N1,m)$r(N1)..
     sum(duga$conec_du(duga,N1),      flow(duga,m))=g=tranzit(N1,m);
```

```
**********************************
ben.. obj=e=-100*sum(m,sum(N1$k(N1), knot(N1,m)*knot(N1,m)))- sum(N1$r(N1),
sum(m,(
 (1+reqr(N1,m)-(sum(duga$conec_du(duga,N1), flow(duga,m)))-tranzit(N1,m))
/(1+reqr(N1,m))*
 (1+reqr(N1,m)-(sum(duga$conec_du(duga,N1), flow(duga,m)))-tranzit(N1,m))
/(1+reqr(N1,m))*
 (1+reqr(N1,m)-(sum(duga$conec_du(duga,N1), flow(duga,m)))-tranzit(N1,m))
/(1+reqr(N1,m))*
 (1+reqr(N1,m)-(sum(duga$conec_du(duga,N1), flow(duga,m)))-tranzit(N1,m))
/(1+reqr(N1,m)) )))
/card(r)/card(m)*P1 + sum(N1$rl(N1), sum(m, (BACK(N1,M)*
sum(duga$conec_du(duga,N1), flow(duga,m)) ) /(0.01+reqr(N1,m))))/card(rl)/card(m)*P2
+
  SUM(M$(ORD(M)=CARD(M)),sum(N1$v(N1),
(vol(N1,m)/(1+vol.up(N1,m)))))/card(v)* P3 -P4 *
sum(m,(sum(n1$v(n1),((power(N1,m)/1000)*(power(N1,m)/1000) ) ) ) ) -
sum(m,sum(duga,( 4*
(flow(duga,m)-flow(duga,m--1))*(flow(duga,m)-flow(duga,m--1))/
(flow(duga,m)+flow(duga,m--1)+0.001)/ (flow(duga,m)+flow(duga,m--1)+0.001) )
))/card(N1)/card(N1)/card(m)*P5 ;
vol.up(N1,m) =12000;
$INCLUDE 'C:\GUIDE\cnstr';
MODEL EPIC /ALL/;
solve EPIC using DNLP maximizing OBJ;
```

The block is created directly by the **BUILDING GAMS MODEL** module using data of the modules **CREATING CONSTRAINTS FILE** and **CONSTRAINTS ENTRY.**

The results of executing modules **CREATING CONSTRAINTS FILE** and **CONSTRAINTS ENTRY** are recorded in the **File of Constraints 2** module, which is in the above described block under the **INCLUDE cnstr** operator**.** We describe the composition and structure of this file further in the section named **CREATING CONSTRAINTS FILE** by the **r_ch_liw.exe** program.

```
PARAMETERS rol,t1,t2,t3,t4,t5,nnn,leveln(n1,m) ;
nnn= sum(N1$r(N1),sum(m$(reqr(N1,m)>1),((sum(duga$conec_du(duga,N1),1)))));
leveln(n1,m)$v(n1) = (k_nba(n1)*sum(duga$conec_udp3(duga),flow.l(duga,m))*
sum(duga$conec_udp3(duga),flow.l(duga,m))/time(m)/time(m)+
 k_nbb(n1)*sum(duga$conec_udp3(duga),flow.l(duga,m))/time(m)+k_nbc(n1));
t1= sum(m,sum(N1$k(N1), knot.l(N1,m)*knot.l(N1,m)));
t2= sum(N1$r(N1),sum(m$(reqr(N1,m)>1),((sum(duga$conec_du(duga,N1),
      flow.l(duga,m)))-tranzit.l(N1,m)) /(0.0001+reqr(N1,m))))/nnn;
t3= sum(N1$rl(N1),sum(m, (BACK(N1,M)*sum(duga$conec_du(duga,N1),
flow.l(duga,m)) )
```

```
/(0.01+reqr(N1,m))))/card(rl)/card(m);
t4=                                    SUM(M$(ORD(M)=CARD(M)),sum(N1$v(N1),
(vol.l(N1,m)/(1+vol.up(N1,m)))))/card(v);
t5= sum(m,sum(duga,( 4*(flow.l(duga,m)-flow.l(duga,m--1))*
(flow.l(duga,m)-flow.l(duga,m--1))/(flow.l(duga,m)+flow.l(duga,m--1)+0.001)/
(flow.l(duga,m)+flow.l(duga,m--1)+0.001))  ))/card(N1)/card(N1)/card(m);
 rol=sum(m,sum(N1$k(N1), knot.l(N1,m)*knot.l(N1,m)));
```

The last block of the base model of optimal water management describes the organization and procedure a decision is delivered to the output file. The operator defining this file's location in a working directory under the fixed name **river.new** incorporates the name of this file.
The block is created directly by the **BUILDING GAMS MODEL** module.

```
file res /C:/GUIDE\river.new/;                                          put res;
 PUT obj.l:20:10;PUT  < obj function     ;PUT rol:20:10;PUT  < virtual water    ;PUT
nnn:20:10;
 PUT  < real user*month ; put /;
 PUT card(duga):10:0     ,  <number lines          ,t2:9:4, <2 task user      ;  put /;
 PUT card(r)     :10:0     ,  <number users        ,t3:9:4, <3 task mouth    ;  put /;
 PUT card(rl)    :10:0     ,  <number mouth        ,t4:9:4, <4 task filling    ;  put /;
 PUT card(v)     :10:0     ,  <number lakes        ,t5:9:4, <5 task stabl.     ;  put /;
      put _____ ;  put /;
loop(m,put  time-step =   ; put m.tl:4; put      connections   LINES  ;          put /;
     put _____ ;    put /;
loop(N1$conec_ud(N1,duga),put  ,N1.TL:4);
put ==>  ,loop(N1$conec_du(duga,N1),put  ,N1.TL:4); put       ,duga.TL:8,   , Put /; );
put _____ ;       put /; );
put _____ ;       put /;
put                 USERS  ;put /;
put _____ ;  put /;
loop(m,put  All-IN-flow  percentage  requirement  Transit  Return,
put   USER      time = ,m.tl:8,                                          put /;
loop(N1$r(N1),put ((sum(duga$conec_du(duga,N1), flow.l(duga,m)))):13:4,
put ((sum(duga$conec_du(duga,N1), flow.l(duga,m))-tranzit.l(N1,m))/
    (0.01+reqr(N1,m))):13:4,   put (reqr(N1,m)):14:4,  put (tranzit.l(N1,m)):10:4,
put ((sum(duga$conec_du(duga,N1),flow.l(duga,m))-tranzit.l(N1,m))*ret(N1)+reB(N1)):9:2,
put   N1= ,N1.TL:8,  !,                                                  put /;  );
put _____ ;  put /;); put_____ ;  put /;
put                 MOUTH   ; put /;PUT _____ ;  put /;
loop(m,put  All_IN_flow  percentage  requirement  Transit, put      USER       time =
,m.tl:8,                                  put /; loop(N1$rl(N1),
put ((sum(duga$conec_du(duga,N1), flow.l(duga,m)))):13:4,
put ((sum(duga$conec_du(duga,N1), flow.l(duga,m))-tranzit.l(N1,m))/
  (0.01+reqr(N1,m))):13:4,  put (reqr(N1,m)):14:4,put (tranzit.l(N1,m)):10:4,
    put (BACK(N1,m)):3:0, put   N1= ,N1.TL:8,   !,put /;  );
```

```
put _____ ;put /;);
      PUT ==========;      put /;
      PUT    WATER BALANCE for all Reservoirs    ;    put /;
      PUT ==========;      put /;
loop(N1$(v(N1)), PUT ==========;      put /;
put v_BEG(N1):10:2;put  => start volume;PUT  WATER BALANCE FOR ,N1.tl, ! ;
put /; PUT ==========;    put /;
PUT   in-flow  volume  out-flow  difference ;PUT inten.evap.  av-surf.  evap.  LEVELV;
PUT   LEVELN   EL.mln.kVtH  time  !;
 put /; PUT ==========;    put /;
loop(m,put  sum(duga$conec_du(duga,N1), flow.l(duga,m)): 9:2,
        vol.l(N1,m): 9:2,  sum(duga$conec_ud(N1,duga), flow.l(duga,m)):10:2,
  (-sum(duga$conec_du(duga,N1),   flow.l(duga,m))+           sum(duga$conec_ud(N1,duga),
flow.l(duga,m))):10:2,
    ee(m):11:4, ( 0.5*aa(N1) *  ((vol.l(N1,m)+vol.l(N1,m-1)$(ORD(M) gt 1)+
 v_BEG(N1)$(ORD(M) EQ 1))**bb(N1))):10:2, (ee(m)*0.5*aa(N1)*
 ((vol.l(N1,m)+vol.l(N1,m-1)$(ORD(M)      gt      1)+          v_BEG(N1)$(ORD(M)      EQ
1))**bb(N1))):9:2,
 ((vol.l(N1,m)+vol.l(N1,m-1)$(ORD(M)     gt     1)+           v_BEG(N1)$(ORD(M)     EQ
1))**dd(N1)+hh(n1)):9:2,
  leveln(n1,m):9:2,  power.l(n1,m):9:2;PUT    time_step => ,PUT m.TL:6;      put /; ); );
PUT
==========;put /;
PUT    TABLE OF COEFFICIENTS FOR RESERVOIRES;          put /;
PUT  S = a * W^b                H = c * W^d +h ; put /;PUT  S - area;   W - volume;       H -
Level;    h - level-of 0;  ;  put /;PUT               n - efficiency;   t - hours-in interval = ,cek:7;
put /;
PUT
==========;put
/;put /;
loop(N1$(v(N1)),put    a = ,aa(N1):8:2, b = ,bb(N1):8:2,     n = ,kpd(N1):8:2;put  RES-RE
,N1.TL:8;
 put  /;  );                PUT  ==========;  put
/;loop(N1$(v(N1)),
put   c = ,cc(N1):8:2, d = ,dd(N1):8:2,    h = , hh(N1):8:2;put  RES-RE ,N1.TL:8;
put /; );PUT ==========;put /;
```

4. Path to Output Files Module

We recommend you to start making a network and doing calculations from this module. The button of this module is located in the upper left corner of the interface's working window. In the lower left corner of the main menu, there is a black window, which shows location of source and resulting data files. To define location the program system uses the DOS system of symbols.

You start this module by pressing the left mouse button after placing the cursor on the **PATH TO OUTPUT FILES** button. The complex control program transfers control to the individual executable **R_WRK_W.EXE** module. The image consisting of two text columns (Figure 4) should appear on the screen. In the first column, each line incorporates eight symbols and defines the name of one subdirectory located inside the directory that contains the program system. In case if the number of subdirectories exceeds the number of lines on the screen, the program system starts scrolling in the **CRT** mode using pointer control keys. The very last line in this column has a fixed name, **NEW**. Only when the pointer is in this line, the user has access to Latin letters **KEYBOARD**. When the user has changed the word **NEW** into any other word, a new subdirectory named with this new word appears in a working subdirectory. If the user is on some line and presses **ENTER**, then the directory named thereby becomes the subdirectory of current optimization calculations. Not to forget the essence of an optimization task located in some subdirectory, the user may insert an explaining comment in the right column. The program system saves a comment for each directory on the hard disc in the **model.cfg** text file and records the name of a current working directory in the **work.way** text file.

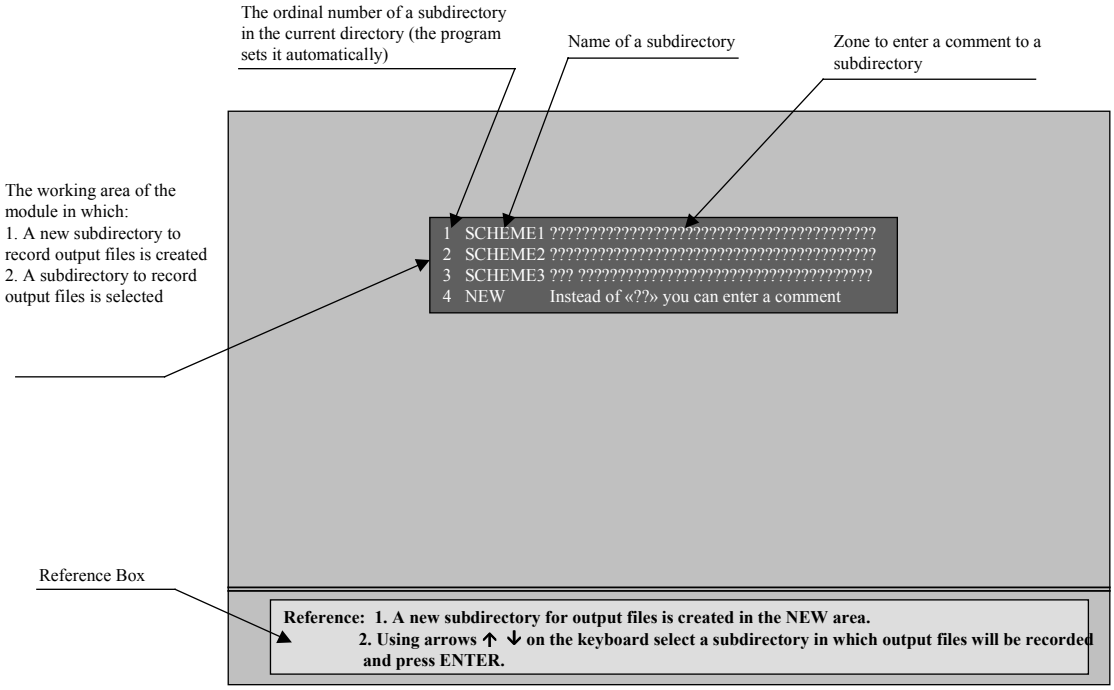


Figure 4. The Image of the Working Window on the Screen Running the SELECTION OF SUBDIRECTORY Module

The user can gain access to each of these files **work.way** and **model.cfg** using any text editor. The user can access the first file through a special button in the main menu. Access is useful if the user wants to place the calculation directory outside the directory that contains executable files. In this case, the user should write his new path using the mode of editing the **work.way** file. The program system updates the **work.way** file constantly. This file incorporates comments that elucidate the meaning of the first file line. A special algorithm checks the **model.cfg** file to be consistent with the actual structure of the directory. If a striking discrepancy

is found, the program system automatically updates this file. This special algorithm aims to keep as many previous comments to existing and recognized directories as possible.

In the **R_WRK_W.EXE** module, you can perform the following operations:

- To create a new subdirectory, in which output files of all executable modules will be written, in the current directory.
- To insert a comment into the subdirectory or modify the text of an already existing comment.
- To choose an already existing subdirectory in the current directory. In this subdirectory (name it working directory) all output files of executable modules will be written.

Creating Subdirectories

If you want to create a new subdirectory, then you should move down to the line with the **NEW** subdirectory name using arrows on the keyboard (↑↓) and instead of **NEW** introduce a new name of the subdirectory (name's length should not exceed seven characters). After the introduction of a new name, press the **Enter** key on the keyboard.

If you press **Enter** while **NEW** is highlighted (i.e. you have not changed the name of the **NEW** file), then in a comment line you will see a message: **You did not assign a name to the new subdirectory.**

Inserting Comments

You can insert a comment for any subdirectory in the comment line (for instance, a more complete name of a calculated river reach). By default, the text of the comment looks as follows **(?…?)**. The comment length cannot exceed 40 characters. The comment text is inserted through the keyboard. To insert a comment:

- Select the subdirectory you need using arrows on the keyboard (↑↓).
- Then, using the keyboard arrows (↑↓), point to the first character of your comment.
- Insert the text of your comment
- After you have completed the insertion of the comment text, press **Enter** on the keyboard or leave the comment line using arrows ↑↓.

If you want to change comment text (for example correct spelling mistakes), then you should completely reinsert the entire comment text. Using any editor, you can also open the **model.cfg** file and make necessary changes in the file itself.

Selecting Working Subdirectory

You select your working subdirectory as follows. Specify your working subdirectory using keyboard arrows (the color of the name of this working subdirectory will change from yellow to white) and press the **Enter** key on the keyboard. With that, you, having defined the path to your working directory, exit the **PATH TO OUTPUT FILES** module**.**

The path to the working subdirectory will be recorded in a separate file (**work.way**), which you can view, if you press the **File work.way** button of the interface. You can change the path directly in this file or assign an external directory or even a directory on another hard disk as your working directory.

5. Structure of the Work.Way File

Consider the structure of the **work.way** file using an example.

Assume that you are in the **OPTIMUM** directory and you want to write output files in the **Scheme 1** subdirectory, and this subdirectory has the following comment text: **Calculated Reach 1 of the Syrdarya River.** Then, the **work.way** file (it always contains three lines) will have the appearance shown in the table below.

No. of File Line	Contents of File Line	Note
Line 1	C:\OPTIMUM\ SCHEME1\	Path to Working Subdirectory
Line 2	You will work in this subdirectory	Reference Text
Line 3	Calculated Reach 1 of the Syrdarya River	Comment to the Subdirectory Selected **PATH TO OUTPUT FILES** Module.

Only the first line of the **work.way** file is important for the program system.

6. Network Creation Module

The **CREATING A NETWORK (R_KLV.EXE)** module serves to create a calculation scheme of a river network. The maximum number of nodes in the calculation scheme allowable by the program system is 400. An attempt to exceed this number leads to the activation of the memory protection algorithm. This algorithm overwrites the 400th line with the 401st line, automatically deleting the 400th line.

The **CREATING A NETWORK** module is operable only if a corresponding version of a graphics driver in PASCAL, **egavga.bgi**. is available.

When the **CREATING A NETWORK** module is executed, a new calculation scheme is created or the existing calculation scheme of the river network is modified. Each node of the calculation scheme is assigned with:

- **External name** (usually, it is presented by names of water diversions, reservoirs or names of measurement stations conventional in literature);

- **National Identity** (since the calculation scheme is created for the Syrdarya River basin, a calculated node can belong to one of the five countries: Tajikistan, Kyrgyzstan, Uzbekistan, or Kazakhstan). These data are essential for tasks of the complex interstate management and will be used in further activities.

The results of executing the **CREATING A NETWORK** module are recorded in the **klav.out** file. The user can locate the path to this file using data in the **work.way** file. The way to start the **CREATING A NETWORK** module is as follows. Point to the **CREATING A NETWORK** interface button and press the left mouse button. An image appears on the screen (Figure 5). This image includes a working area (big black square with a calculation scheme). Above and on the left there are buttons and dialog boxes.

On Figure 5, we show a calculation scheme for our test problem, made in the **CREATING A NETWORK** module.

You build a calculation scheme of a river network on the working area. In the general case, the calculation scheme of the model is a system of calculated arcs (further on we will say simply **arcs**) connected in accordance with relations that exist between these arcs under actual circumstances. Each arc includes two nodes (the first and the second) and a straight line connecting these two nodes. Under actual circumstances, the node is an entity, and the straight line connecting two nodes means that a certain relationship between these two entities exists. If some node (entity) has connections with several other nodes (entities), then this node belongs to several arcs. On a calculation scheme, several lines connecting this node with corresponding nodes will go off it.

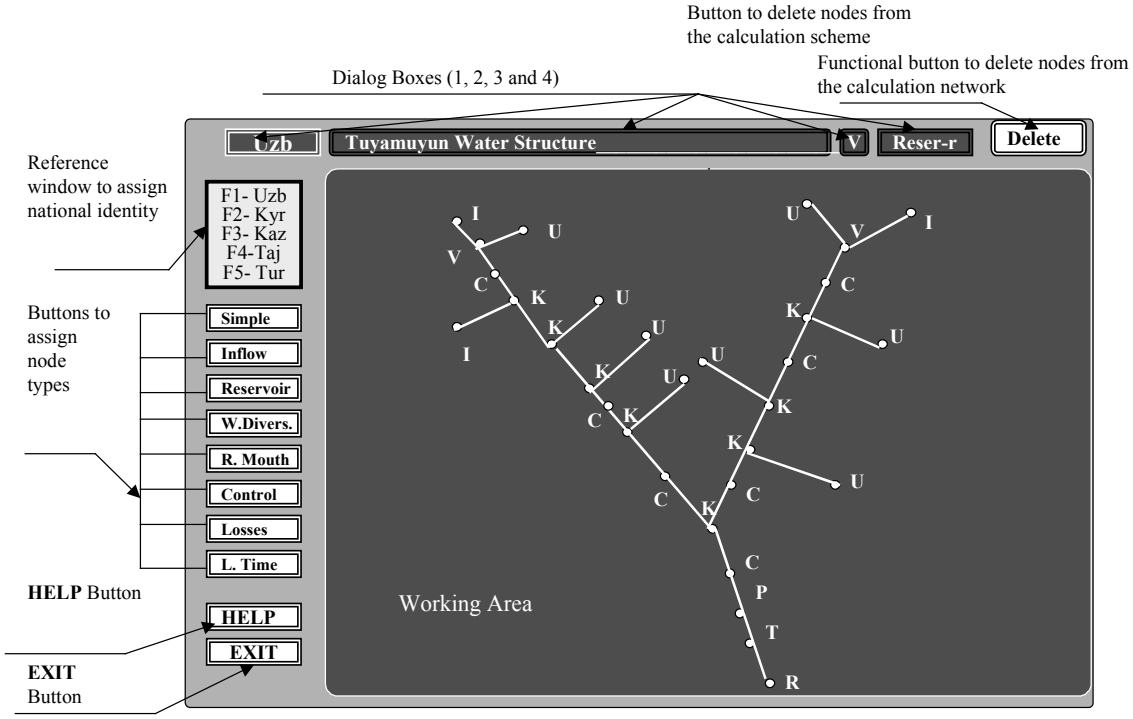


Figure 5. The Image of the Working Area on the Screen Running the CREATING A NETWORK Module.

In the **CREATING A NETWORK** module, you perform the following operations:

1. Creating a system of nodes and arcs of the calculation scheme;
2. Changing the internal name (type) and national identity of the node;
3. Changing the external name of the node;
4. Deleting the node and its arcs from the calculation scheme;
5. Getting help;
6. Exiting the module.

Consider each operation performed in module **CREATING A NETWORK** in details.

Creating a System of Arcs of the Calculation Scheme

To create the first node of an arc move the mouse pointer to the place of the working area you need (it may be any point on the working area or any already existing node) and press the left mouse button. A sound signal is a message to the user that the first node of the arc being created has been already made and is present in the scheme. No signal informs the user that the node being created has not been linked with the already existing calculation scheme.

After that, (to create the second node of an arc) having released the left mouse button, point to another point of the working area (it may be also any point on the black area or any already existing node). At this time, a red line should appear on the screen. One end of this line is in the point of node 1 of the arc; the second end is at the position of the mouse pointer. Press the right mouse button. A new arc should appear on the screen: two nodes and a line connecting these two nodes, all of the white color. Letter **K** of the green color will appear near each newly created node. It is caused by the fact that by default each node is assigned with type **simple node,** with internal name (index) **K** and external name **???????????** which may be later changed to necessary ones (see below).

Building a scheme observes a rule embedded in the program by a special algorithm: if in an arc being newly created you set the second node very close to the first one (closer than 0.5 cm on the screen), then the program will treat both the first and the second nodes as one node. Such an arc will not be included in the scheme. On the screen, if you call the module again, you will not see this supershort arc.

We advise you not to set a new node on the lines that form arcs (on white lines) because a difference between computer and visual treating of the calculation scheme may occur. It is easier to check the existence of such imaginary links in the SETTING FLOW DIRECTION module by slight deformation of the created network.

By default a node is assigned with:

- Node type – simple, and a corresponding internal name (index) for a simple node – **(K)**
- External name (**???????????????**)

- National identity – numerical index **1**

To view features of nodes there are dialog boxes (1, 2, 3, and 4) located above the working area (Figure 5).

Dialog box 1 displays national identity of the node.
Dialog box 2 displays external name of the node.
Dialog box 3 displays internal name (or index that specifies the type) of the node.
Dialog box 4 displays type of the node.

To change features of a node by default, use buttons on the left from the working area. To select a button to define a node type you use the mouse or control keys of the keyboard.

Changing Internal Name (Node Type)

You change the internal type of the node using the buttons for assigning node types. These buttons are located on the left side of the working area (Figure 5). The label on an active button that defines node types is black (for example, on Figure 5 the active button is the button named **Simple**). Other buttons determining the node type (**Inflow, Reservoir, Water Diversion, River Mouth, Control, Losses, and Lag Time)** are not active and have the gray color.

To change the internal name (type) of the node, you should:

- Point to the node you need (the color of the node should change and become red)

- Not moving the mouse pointer from the node (the color of the node constantly remains red), you select the node type you need (the label of the button with the name of a necessary node type should become black, that is you activate the button of selecting a node type) by pressing arrows on the keyboard (↑ or ↓). You can activate the button of selecting a node type using the mouse.

- Press the **Tab** key on the keyboard

The letter that defines the node (index) will change in accordance with Table 1 given below. Corresponding changes will also take place in dialog boxes 3 and 4.

Below is given the table of types of nodes and corresponding symbols in internal names (or indexes). Full internal names (letter index + ordinal number of the node of a specific type in the calculation scheme) will be assigned to nodes in the **DATA ENTRY** module.

Types of Nodes	**Internal Name (or Index)**
Simple	**K**
Inflow	**I**
Reservoir	**V**
Water Diversion	**U**
River Mouth	**R**

Control Point	**C**
Losses	**P**
Lag Time	**T**

Changing National Identity of the Node

You change national identity of a node using functional keys **F1, F2, F3, F4,** or **F5** on the keyboard. On the left side of the working area there is an inquiry window (Figure 5) enabling you to select a functional key correctly.

Below is given a table showing national identity of a node, the corresponding functional key on the keyboard, and the index set in this package of programs. At the user's desire the sequence of indexes can be changed, since the specific name of a country is for referential use and is employed only in one example in appendixes.

National Identity of Node	**Functional Key on the Keyboard**	**Internal Index**
Uzbekistan	**F1**	**1**
Kyrgyzstan	**F2**	**2**
Kazakhstan	**F3**	**3**
Tajikistan	**F4**	**4**
Turkmenistan	**F5**	**5**

To change national identity of a node you should:

- Point to a node (color of this node should change for red)
- Not moving the mouse pointer from the node (color of the node should be constantly red), press one of functional keys on the keyboard (**F1, F2, F3, F4,** or **F5**). You will see corresponding changes in the dialog box.

Attention! Since by default the program system assigns the **K**-simple type to each node, we advise you to review carefully labels of nodes checking if one node belongs to two different types. You can see this discrepancy in the form of simultaneous displaying of the same symbols in one place. You have to reassign types of such controversial nodes.

Changing External Name of a Node

To change the external name of a node you should:

- Point to a node (color of this node should change for red), and a prior assigned external name of the node should appear in dialog box 2 instead of the line "++++++". If that external name was not assigned, then by default it will be the **????????** line;
- Enter a new external name of the node from the keyboard. At the moment you are defining an external name, the mouse becomes deactivated, and the dialog box becomes blue.

- After you entered a new external name of the node, press the **Enter** functional key on the keyboard. The program system automatically cancels the entry mode, as symbols used to assign the external name are over. Dialog box 2 changes its color from blue to black when you cancel entering the external name. The mouse becomes active again.

You should remember that you cannot correct the text of the external name directly in the module. If you have found a spelling mistake in the external name of a node, then you should retype the entire text of the external name. You can introduce changes in the external name of the node in the Network in Indexes 1 output file (see below). Therewith, you should take heed of not breaking the structure of the Network in Indexes 1 file because in this case programs will not work.

The maximum allowable length of the external name is 44 characters.

Deleting Arcs from the Calculation Scheme

To delete nodes and respective arcs from the calculation scheme there is the **DELETE** button located in the upper right corner of the screen (Figure 5).

To delete arcs from the calculation scheme you should:

- Point to a node (color of this node should change for red) and press the left mouse button.
- Having released the left mouse button, point to the **DELETE** button and press the right mouse button.

When you apply the **DELETE** mode, you also delete all nodes forming arcs with the given node provided they do not belong to other arcs.

Getting Help

To get help there is special button **HELP** in the left lower corner of the working window (Figure 5). To enter the Help mode you should:

- Point to the **HELP** button
- Press the right mouse button. Information about all features of this module will appear on the screen.

To exit the Help mode you should press any key on the keyboard.

Quitting the Module

To quit the **CREATING A NETWORK** module there is the **EXIT** special button in the left lower corner of the working window (Figure 5). To quit the **CREATING A NETWORK** module you should:

- Point to the **EXIT** button

- Press the right mouse button

An image appears on the screen (Figure 6).

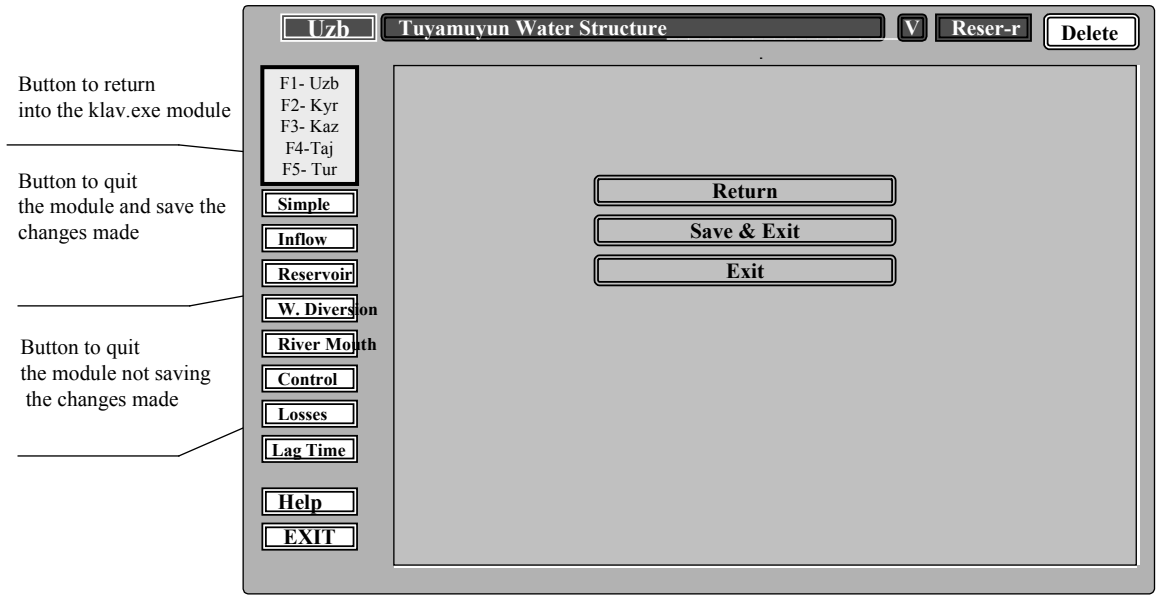


Figure 6. The Image of the Working Window on the Screen As You Press the EXIT Button in the CREATING A NETWORK Module.

Point to the button you need (either **SAVE and EXIT** or **EXIT,** or **RETURN**) and press the left mouse button. If you chose the **SAVE and EXIT** button, then after pressing the left mouse button you will see the following message box:

You decided to save a new calculation scheme.

If you added or deleted some nodes, then you should create all output files of the DATA ENTRY module again.

If you want to save a newly created network, press Y

If you do not want to save a newly created network, press any key

As you run the **saving** mode, the program system renames all files with data on the calculation scheme in the working subdirectory. These files form as you run next modules. Names of these files will remain, but these files will have the ***.bak** extension. Only the newly formed **Network in Indexes 1** file will exist in the working subdirectory.

In that situation files named **copy_1.new**, **copy_1.bak, copy_2.new, copy_2.bak, copy_3.new** and **copy_3.bak** will be automatically created. These files are of particular importance for restoration of data previously entered. We will tell you about that below.

The special block for checking the correctness of scheme building will check a newly created scheme and will delete impossible combinations of arcs and nodes from it. Therefore, after you created the scheme, we advise to restart the **CREATING A NETWORK** module and visually assess the admissibility of the changes the program made in the calculation scheme.

Attention! *This package of programs provides for the calculation of power generation by reservoirs when they release water through turbines. Power generation is calculated only on arcs, where the first node is a reservoir, and the second node is a control point. Therefore, avoid setting many control points on arcs going out of the reservoir because it may lead to problems in calculating power generation. If there are no those arcs where the first node is the reservoir and the second node is the control point, power generation is not calculated.*

In addition, in the process of saving the user gains access to the summary of initial testing of the system's correctness, which specifies whether optimization is possible in principle. For example if there is no water source in the scheme, then there is actually nothing to manage and optimal water allocation does not make sense. The same occurs if there are no consumption nodes in the scheme because there is nobody to supply him water. If there is no river mouth node, the user will receive a warning about the possibility of emergency in the event of water excess.

If the user wants to quit the **CREATING A NETWORK** module at once, then he should press the **Esc** functional key on the keyboard twice. The user will quit the **CREATING A NETWORK** module **without** saving the changes made.

Execution of the **CREATING A NETWORK** module results in formation of the **Network in Indexes 1-klav.out** file. This file serves the actual basis for all further operations of the program system, and therefore we should spend some time to describe its structure.

Structure of an Output File of the CREATING A NETWORK Module – the Network in Indexes 1 – klav.out File

Consider the structure of an output file created by the **CREATING A NETWORK** module **for** our test problem – the **Network in Indexes 1** file.

To open the **Network in Indexes 1** file, perhaps for editing, you should:

- Point to the **Network in Indexes 1** button
- Press the left mouse button

An output file after executing the **CREATING A NETWORK** module for our calculation scheme looks as follows:

A reference table, which specifies always occupies the first ten lines of the output file, where digital codes of node types (first column of the table) and national identity of the node (second column of the table) are specified. This table looks as is shown below.

The following five lines of the output file are names of areas of the main table, in which all data on the calculation scheme of the model are given.

The first table area (first column) is named **N**. This area shows the ordinal number an arc appears on the scheme in the creation process as you run **CREATING A NETWORK** the module.

The general name **"CONNECTIONS"** unites the following four areas (columns)**.** Each area (column) has a subname (**x, y, x, y**). Each pair of figures (**x** and **y**) is the coordinates of nodes (in pixels) in the arc on the working area of module **CREATING A NETWORK.** The first pair of coordinates **x** and **y** is coordinates of the node in the arc under the condition that the line in the arc connecting this pair of nodes is directed from the first node to the second. For instance, in arc **1** water flow is directed from the first node with coordinates (463, 76) to the second node in the arc with coordinates (435, 74). Each area (column) is provided with 4 positions in the line because three positions are enough to describe coordinates under the standard screen resolution of 640*480. There should be not less than one position of spaces.

The general name **"TYPES"** unites the next pair of areas (columns)**,** and each area (column) has a subname (**1** and **2**). These areas show numerical codes of node types in accordance with the reference table provided at the beginning of the output file. In the area (column) subnamed **1,** there are numerical codes of a node type for the first node in the arc. In the area (column) subnamed **2,** there are numerical codes of a node type for the second node in the arc. As for coordinates of nodes, that node is considered the first in the arc from which a connection is directed to another node. For example, flow is directed from the first node to the second. Each numerical code of a node type (area) is provided with 2 positions in the line. There should be not less than one position of spaces between areas.

The general name "**STATES**" unites the next pair of areas (columns) and each area (column) has a subname (**1** and **2**). These areas show numerical codes of national identity of the node in accordance with the reference table at the beginning of the output file. In the area (column) subnamed **1,** there are numerical codes of national identity for the first node in the arc. In the area (column) subnamed **2,** there are numerical codes of national identity for the second node in the arc. As for coordinates of nodes, that node is considered the first in the arc from which a connection is directed to another node. For example, flow is directed from the first node to the second. Each numerical code of a node type (area) is provided with one position in the line. There should be not less than 1 position of spaces between areas.

Further, there is a space and symbol ' => 'indicating the end of the description of the calculation scheme in numerical symbols.

Next, there are external names of nodes (first the external name for the first node, then for the second). Each external name of the node is provided with 44 positions in the line. The |

symbol separates the external name of the first node from the external name of the second node. If an external node name has a length of less than 44 characters, then you should continue it by the _ symbols up to the length of 44 characters.

The program system adds these symbols automatically, when you finish entering an external name in the **CREATING A NETWORK** module. The external name of the second node should end with the ' <= ' symbol indicating the line end. Do not delete the _ symbol from the name because later it may cause problems to identify external names. The appearance of the automatically created **Network in Indexes 1 – klav.out** file is shown below.

7. Flow Direction Module

The **SETTING FLOW DIRECTION-R_REGIM.EXE** module serves to assign and change direction in arcs.

The **SETTING FLOW DIRECTION** module is operable only if a version of a graphics driver in PASCAL, **egavga.bgi** is available in the same directory.

The file of data entry for the **SETTING FLOW DIRECTION** module is the **Network in Indexes 1** file. The adjusted **Network in Indexes 1** file is also an output file after you have operated the **SETTING FLOW DIRECTION** module. This module neither uses any nor creates other files.

The way to start the **SETTING FLOW DIRECTION** module is as follows. Point to the **SETTING FLOW DIRECTION** button in the main menu. Press the left mouse button. The following image (Figure 7) appears on the screen. It includes the working area (big black square), two buttons (**HELP** and **EXIT**) on the left of the area, and a dialog box above of the area.

If you point to a node, then in the dialog box, you will see features of this node: external node name, node type, and national identity of the node.

On the working area (Figure 7), you will see a calculation scheme of the model created as the program system executes the **SETTING FLOW DIRECTION** module. Along the red lines connecting nodes in arcs, you will see moving white points. Direction of movement of white points indicates directions of connections among nodes. If white points move too fast, then point to any node, and the points will move slower. We provided this opportunity because different types of computers have different speeds. The user is thereby able to build both complex and simple schemes.

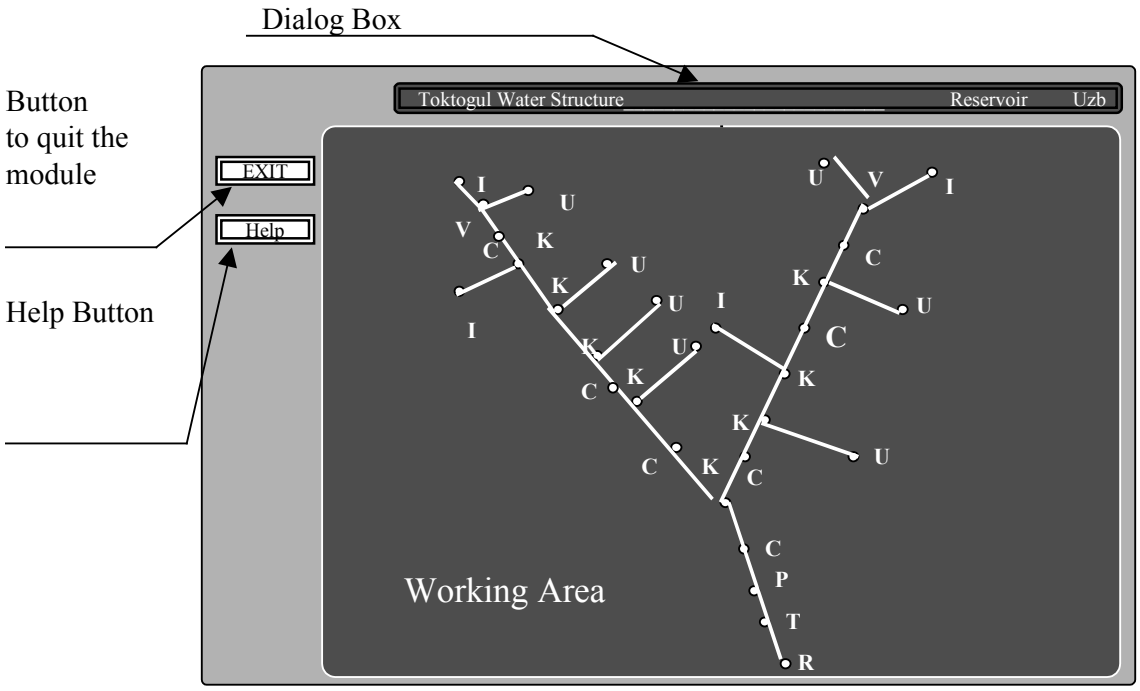


Figure 7. Image of the Working Window As You Run the SETTING FLOW DIRECTION Module.

The **SETTING FLOW DIRECTION** module enables performing two operations with the calculation scheme created as the program system executes the **CREATING A NETWORK** module**:**

- Changing configuration of a calculation scheme with modifying structural connections in it.
- Changing direction of connections among nodes.
- Make a primary analysis of the scheme's correctness.

Changing Configuration of a Calculation Scheme

To change configuration of a calculation scheme you should:

1. Point to the node you need
2. Press the left mouse button (at that you will hear a sound)
3. Pressing the left mouse button drag the node to the position you need
4. Release the mouse button at the moment when the node will be in the position you need

Changing Flow Direction of Links among Nodes

To change flow direction of connections among nodes you should:
1. Point to the middle of a line that connects nodes (at that you should hear a sound signal)
2. Press the left mouse button

A prior existed direction of connection between nodes in the arc will change for the opposite.

Getting Help

To get Help you should:

1. Point to the **HELP** button in the left upper corner of the working window
2. Press the left mouse button

Quitting Help

To quit Help you should press the left mouse button

Quitting the Module

To quit the **SETTING FLOW DIRECTION** module you should
1. Point to the **EXIT** button in the left upper corner of the working window (Figure 7)
2. Press the left mouse button

An image should appear on the screen (Figure 8). Point to one of three buttons and press the left mouse button.

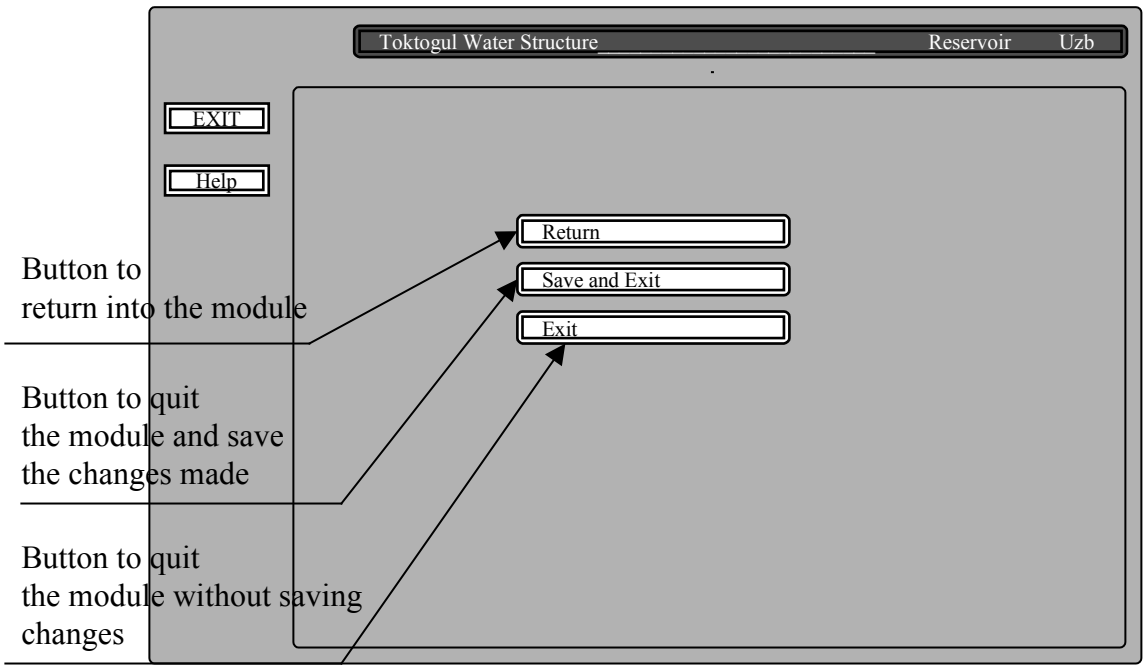


Figure 8. The Image of the Working Window on the Screen As You Press the EXIT Button in the SETTING FLOW DIRECTION Module

If you choose the **RETURN** button, you return to the working window of the **SETTING FLOW DIRECTION** module (Figure 7).

If you have chosen the **EXIT** button**,** then the **Network in Indexes 1** module will remain unchanged. If you have chosen the **SAVE & EXIT** button, then all changes made in the configuration of the calculation scheme and changes of directions of connections among nodes will be recorded in the **Network in Indexes 1** file. However, in some cases, **SAVE & EXIT** does not work, and the user will return to the editing of the scheme. The matter is that the algorithm that makes a primary check of the calculation scheme's correctness will start automatically, being latent for the user. Question marks on the scheme designate dubious nodes. For example, the program system does not accept the node indicated as a source with only one incoming arc. Also, the program system does not accept a river mouth node having at least one outgoing arc. Possible ways out of this situation are:

1. adjust direction of arcs
2. exit this module without saving and reassign node types in the previous module.

8. Data Entry Module

The destination of the **DATA ENTRY** module is to enter data (numerical values) for all nodes (entities) of a calculation scheme of a river network. In terms of design, this module consists of two executable programs **R_1.EXE** and **R_22.EXE**, executed sequentially.

The **DATA ENTRY** module can work correctly only if the previous modules (**CREATING A NETWORK** and **SETTING FLOW DIRECTION**) have completed their operation because it uses information of the **Network in Indexes 1-klav.out** output file. In the process of executing the **DATA ENTRY** module, the **Network in Indexes 1** output file remains unchanged.

The **DATA ENTRY** module is operable only if a graphics driver in PASCAL, **egavga.bgi** is available.

These modules **R_1.EXE** and **R_22.EXE** start working, when the user presses the left mouse button pointing to the **DATA ENTRY** button. The first module operates automatically without the user's participation. Its major objective is to form the out.out file. Basing on the data of the **klav.out** file, the module automatically assigns internal names to all nodes incorporated in the irrigation river network by adding an ordinal number of this node type to the letter that characterizes the node type in the **klav.out** file. The program system adjusts internal names assigned by this module in accordance with external names defined by the user. A table in the out.out text file incorporates external and internal names with their correspondences.

Structure of the Output File *Network in Indexes 2*

The **Network in Indexes 2-out.out** output file contains full information on the created calculated scheme of the river network.

The first line of the Network in Indexes 2 file contains statistical parameters of the calculated river network. Digits in the first line show the total number of nodes in the calculated network and the number of nodes of each type:

- The first digit shows the number of simple nodes in the calculated network;
- The second – the number of tributaries;
- The third – the number reservoirs;
- The fourth – the number of water diversions;
- The fifth – the number of nodes of the river mouth
- The sixth – the number of control points;
- The seventh – the number of nodes of channel losses calculation;
- The eighth – the number of nodes of lag time calculation;
- The ninth – the total number of nodes in the calculated network.

Next lines of the file from the second to N+1 (where N is the number of nodes in the calculation scheme) contain information on each node in the calculated network:

- The first digit in the line – consecutive numbering of nodes.
- The second and third digits in the line – coordinates of a node (x, y) on the screen.
- The fourth digit in the line – numerical index of a node type (see the chapter Module klav.exe).
- The fifth digit in the line – numerical index of national identity (See Module **CREATING A NETWORK**).
- The sixth digit in the line – ordinal number of a nodes for the nodes of one type. This ordinal number for nodes of each type is set automatically, as the program system executes the **R_1.EXE** module.
- Then, the letter index of the node type with the ordinal number of the node.
- Then, division symbol ¦ .
- The external name of the node follows the ¦ division symbol.

Between each digit, there should be at least one space. All information is grouped in columns.

The **Network in Indexes 2** file first describes simple nodes of the calculated network, then tributaries, reservoirs, water diversions, the river mouth, control points, nodes of channel losses calculation, and at last nodes of lag time calculation.

Below we give the full text of the **Network in Indexes 2** file formed for our problem.

No. of File Lines	Data Given in Each File Line
Line 1	8 4 2 7 1 7 1 1 31
Line 2	1 420 148 1 1 1 K1 ¦Point of Water Diversion to the Feeding Canal_____
Line 3	2 385 197 1 1 2 K2 ¦Imputed Point of Confluence of the River on Kara Darya____
Line 4	3 370 226 1 1 3 K3 ¦Point of Water Diversion on the Andijan-Uchtepa Reach____
Line 5	4 194 148 1 1 4 K4 ¦Imputed Point of Confluence with Tributary Inflow___
Line 6	5 212 172 1 1 5 K5 ¦Point of Water Diversion 1 on the Toktogul –Uchtepa Reach_
Line 7	6 235 196 1 1 6 K6 ¦Imputed Point of Channel Losses on a River Reach_____
Line 8	7 281 249 1 1 7 K7 ¦Point of Water Diversion 1 on a Reach to the Uchkurgan Measurement Station_____
Line 9	8 326 289 1 1 8 K8 ¦Point of Confluence (Uchkurgan Water Structure+Uchtepa Measurement Station)_____
Line 10	9 430 63 2 1 1 I1 ¦Inflow to Andijan Reservoir_____
Line 11	10 364 147 2 1 2 I2 ¦Tributary Inflow on Kara Darya_____
Line 12	11 128 70 2 1 3 I3 ¦Inflow to Toktogul Reservoir_____
Line 13	12 145 152 2 1 4 I4 ¦Tributary Inflow on the Toktogul-Uchkurgan Reach
Line 14	13 463 96 3 1 1 V1 ¦Andijan Reservoir_____
Line 15	14 150 96 3 1 2 V2 ¦Toktogul Reservoir_____
Line 16	15 495 74 4 1 1 U1 ¦Losses from Andijan Reservoir_____
Line 17	16 464 143 4 1 2 U2 ¦Feeding Canal_____
Line 18	17 425 216 4 1 3 U3 ¦Water Diversion on the Andijan-Uchtepa Reach_____
Line 19	18 182 68 4 1 4 U4 ¦Losses from Toktogul Reservoir_____
Line 20	19 237 132 4 1 5 U5 ¦Water Diversion on the Toktogul-Uchtepa Reach_____
Line 21	20 264 158 4 1 6 U6 ¦Channel Losses on a River Reach_____
Line 22	21 316 201 4 1 7 U7 ¦Water Diversion 2_____
Line 23	22 323 399 5 1 1 R1 ¦River Mouth_____
Line 24	23 441 123 6 1 1 C1 ¦Point of Control over Releases from Andijan Reservoir_____
Line 25	24 406 174 6 1 2 C2 ¦Karabagish Measurement Station_____
Line 26-	25 351 257 6 1 3 C3 ¦Uchtepa Measurement Station_____
Line 27	26 172 127 6 1 4 C4 ¦Point of Control over Releases from Toktogul Reservoir___
Line 28	27 263 224 6 1 5 C5 ¦Uchkurgan Measurement Station_____
Line 29	28 305 267 6 1 6 C6 ¦Uchkurgan Water Structure_____

Line 30	29 336 318 6 1 7 C7 ¦Point of Control over Confluence_____
Line 31	30 336 339 7 1 1 P1 ¦Point of Channel Losses Calculation_____
Line 32	31 333 369 8 1 1 T1 ¦Point of Lag Time Calculation_____

The user can access this file, perhaps for editing, through the main menu by pressing the left mouse button and using the Network in Indexes 2 – out.out button.

The **DATA ENTRY** module (part **R_22.EXE**) creates a group of files that contain information about the data entered. For the data entered, a separate file is created for each node type. If you did not enter numerical values for some node, then these values are adopted by default.

The table shows filenames that form depending on a node type as data are being entered (values for a given node type adopted by default).

- **Internal name** (which is an index characterizing the node type and its ordinal number in the scheme);

File Title	Filename	Data Entered by Default
Inflows	supplies.in	0.00
Reservoirs	lakes.in	0.00
Water Diversions	user.in	0.003
River Mouths	mouth.in	0.00
Losses	losses.in	0.00
Lag Time	lagtime.in	0.00
Time Step	steps.in	4 – month index
Evaporation	evapor.in	0.00
Network in Indexes 2	out.out	Table of Correspondences

The **DATA ENTRY** module performs the following operations:

- Assigns a number of time steps;
- Assigns a type of time steps;
- Assigns a unit to measure the data being entered;
- Enters necessary coefficients to take into account the variation of the number of days in months and ten-day periods during the year;
- Enters necessary coefficients to calculate evaporation from reservoirs
- Enters necessary data for the **Inflow** nodes;
- Enters necessary data for the **Reservoir** nodes;
- Enters necessary data for the **Water Diversions** nodes;
- Enters necessary data for the **River Mouth** nodes;
- Enters necessary data to calculate channel losses;
- Enters necessary data to calculate lag time.

The way to start the **DATA ENTRY** module is as follows. In the interface, point to the **DATA ENTRY** button and press the left mouse button. An image (Figure 9) will appear on the screen. This image consists of a working area with a calculation scheme of a river network (big black square with the scheme), a toolbar with buttons and dialog boxes above the area, an area for the data entry tables (See below) on the left, and a Help text for urgent quitting the module (exit without saving the changes made) below the area.

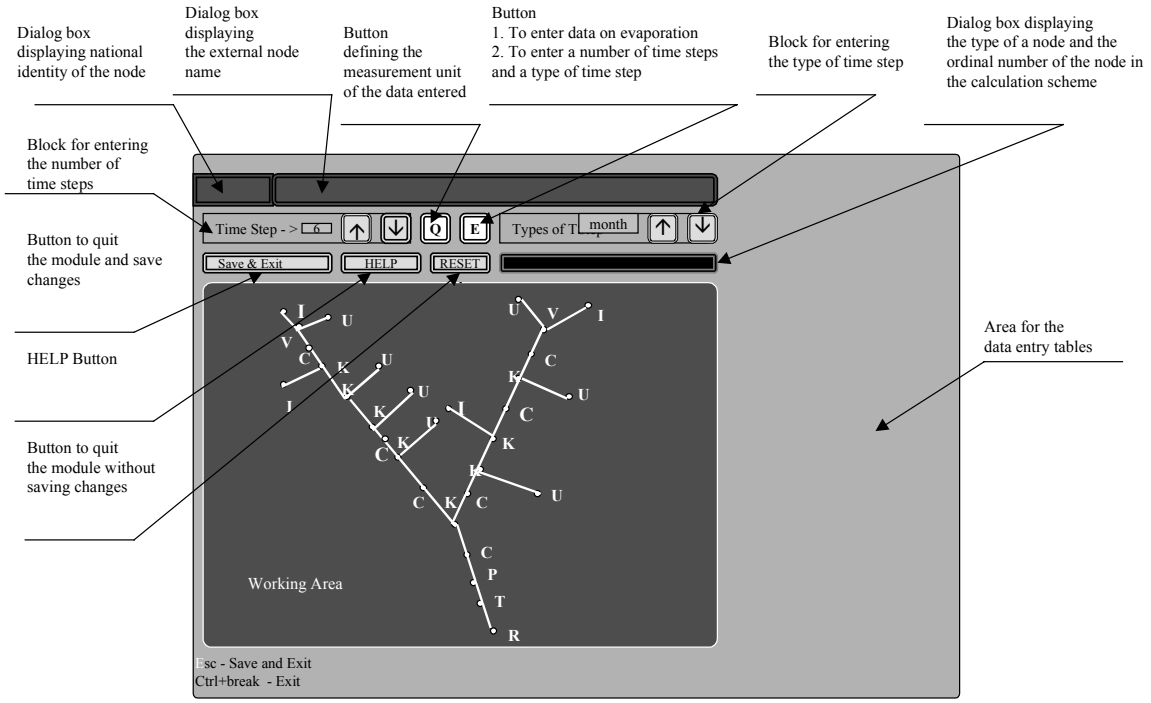


Figure 9. Image of the Working Window on the Screen as You Enter the DATA ENTRY Module.

If you point to any node, then you will see features of this node in dialog boxes.

Two dialog boxes of the upper bar, which are located in the upper line of the screen (dialog boxes 1 and 2) (Figure 9) display:

- Dialog box 1 – national identity of the node
- Dialog box 2 – external name of the node.

Dialog box 3 located in the third line on the right from the toolbar (Figure 9) displays the type of a node and its ordinal number on a calculated river network.

The toolbar (second line from above) includes two blocks:

- Buttons to enter the number of time steps.
- Button defining a unit to measure the data entered and being entered.
- Buttons to enter the type of time steps.

Assigning Number of Time Steps

To set a number of time steps you have the area for entering the number and type of time steps. For this purpose, there is the special **E** button located in the upper part of the screen, in the second line, and the block for entering a number of time steps. The block incorporates:

- Expression **Time Step** is a comment to this functional block.
- Dialog box displays the number of the time steps set.
- Two buttons. The button with the up arrow is intended to increase the number of time steps. The button with the down arrow is intended to decrease the number of time steps.

The maximum number of time steps is **36**.

When the **E** button is pressed, the scheme on the working area is not active; therefore, you cannot enter data for scheme nodes. By default, the block of entering the number of time steps is disengaged, and if you press buttons with arrows (↓↑), then they will not change the number of time steps. To activate the block of entering the number of time steps you should click the left mouse button on the **E** button. The block of entering the number of time steps is active and you can enter a number of time steps pressing (↓↑) buttons on the screen. To increase the number of time steps, point to the button with the up arrow (Figure 10). Click the left mouse button. The number of time steps will increase by a unit. If you point to the button with the down arrow and click the left mouse button, then the number of time steps will decrease by a unit. Change the number of time steps until you get the number you need using these buttons.

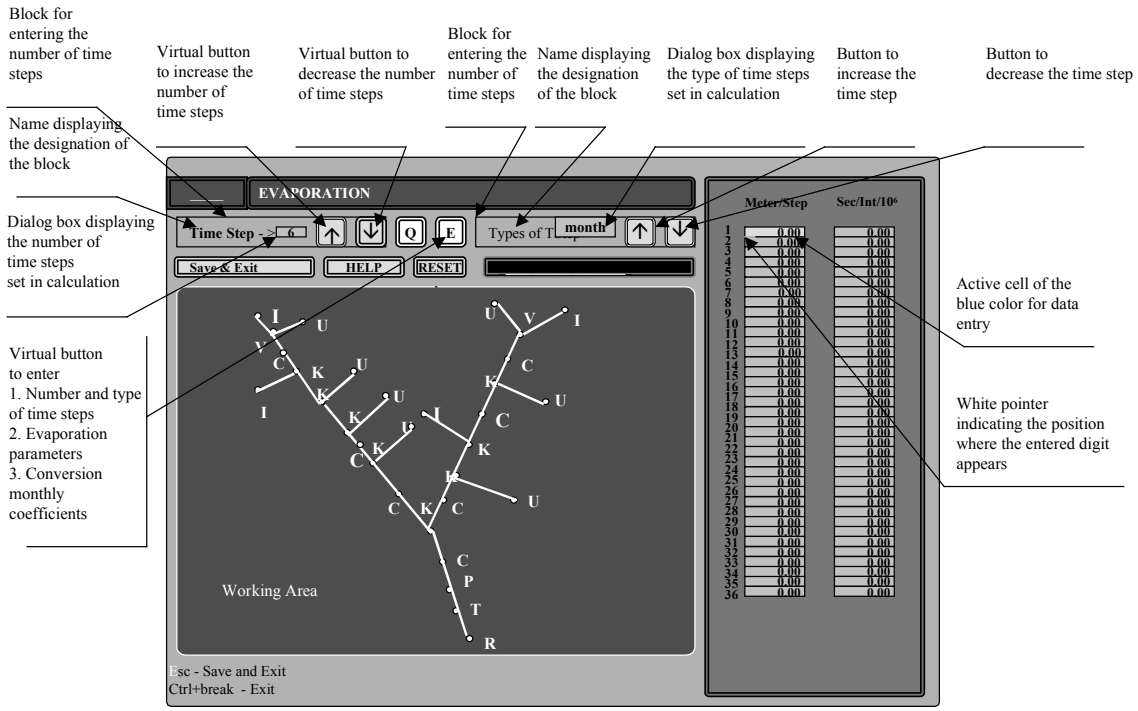


Figure 10. The Image of the Working Window on the Screen in the DATA ENTRY Module As You Enter Evaporation Parameters, Conversion Monthly Coefficients, Type of Time Step, and Number of Time Steps.

After you have completed entering a number of time steps, click the left mouse button on the **E** button to disengage the block of entering time steps, and activate the calculation scheme on the working area. At this moment, the set number of time steps is recorded in the **Time steps – step.in** file. The user can access this file, perhaps for editing, through the control menu. The program system provides all records in this file with comments.

Assigning Type of Time Steps

To set a type of time steps you have the area for entering a type of time steps, the same **E** button located in the upper part of the screen, in the second line, and the block of entering the type of time steps. The block consists of

- Word **typesINT** is a comment that characterizes the toolbar area intended to define a type of a time step.
- Dialog box located after the **typesINT** word displays a type of the set time step.
- Two buttons. The button with the up arrow is intended to increase time steps. The button with the down arrow is intended to decrease time steps.

There are 4 types of time steps: **month, ten-day period, day, and hour**.

By default, the block of entering time steps is disengaged, and if you press buttons with arrows (↑↓), they will not change the type of time steps.

To activate the block of entering the type of time steps you should:

Press the **E** button using the mouse driver. The block of entering a type of a time step becomes active and allows the user to enter a type of a time step using the (↓↑) buttons on the screen.

When the **E** button is pressed, the scheme on the working area is not active; therefore, you cannot enter data for scheme nodes.

To obtain a longer type of time steps (for example we had the **ten-day** time step, point to the button with the up arrow (Figure 10). Click the left mouse button. The type of time steps becomes **month** (you can see it in the dialog box). If you point to the button with the down arrow and click the left mouse button, then the type of time steps becomes **day**. Change the type of time steps using these buttons until you get the type you need.

After you completed entering a type of time steps, click the left mouse button on the **E** button to disengage the block of entering time steps and to activate a calculation scheme on the working area. At this moment, the set number of time steps is recorded in the **Time Steps** file. The user has access to this text file and may correct it without opening this executable module.

When the user presses the **E** button, two columns of data entry (Figure 10) should appear in the data entry area. The user can define two groups of parameters described in details in next two items of the description.

Entry of Conversion Coefficients to Consider Variation of the Number of Days in Months and Ten-Day Periods during a Year

To enter conversion coefficients you have the special **E** button located in the upper part of the screen in the second line, and a table of data located in the area for data tables.

To activate the table of conversion coefficients data you should click the left mouse button on the **E** button, in consequence of that two data columns should appear in the data entry area (Figure 10). In the first column, you enter data on evaporation from the surface of reservoirs, in meters for a time step. In the second column, you enter conversion coefficients to convert water consumption into flow in million cu m for a time step. If you ran the program for the first time, then conversion coefficients adopted by default for this type of time step are written in the working area.

If the white pointer indicating the position where the entered digit appears is located in the wrong cell, you can move it to the cell you need using command keys on the keyboard (↑↓→ and ←). The pointer marks the entry position, which changes as the user presses command keys and the **Ctrl** key simultaneously. The user completes entering a digit by pressing **ENTER** or leaving the area of the entered digit. If the user cannot leave it, it means that the program system cannot interpret the digits entered and therefore the user has to correct these digits. Until the entry completion, no other operations in the module are possible. Direct entry of conversion coefficients using different digits allows the user to expand types of time steps (for instance, pentads, three-day periods, etc).

After the completion of entering conversion coefficients point to the **E** button and release it. At that moment, the coefficients entered are recorded in the special **Losses evapor.in** output file. See the structure of the **Losses** output file below.

Entering Evaporation to Calculate Losses

You enter coefficients to calculate losses from reservoirs in the same way as you enter conversion coefficients to consider variation of the number of days in ten-day periods and months during a year, you only enter data in the column named **Meter/Step**.

After you have completed entering evaporation, deactivate the mode as follows. After you have entered conversion coefficients point to the **E** button and release it. At that moment, the coefficients entered are recorded in special output file **Losses**. See the structure of the **Losses** output file below.

When you press the E button, you activate at once the block of entering the number of time steps, the block of entering the type of time step and tables of entering conversion coefficients and evaporation. Therefore, pressing the E button you can perform operations related to this button:

- Assign a number of time steps
- Assign a type of time steps

- Enter necessary coefficients to consider variation of days in ten-day periods and months during a year
- Enter evaporation values to calculate evaporation

Assigning Units to Measure the Data Entered

To set measurement units of calculation parameters for nodes that constitute an irrigation river network, you have a button located between the block of entering the number of time steps and the block of entering the type of time step. The label on this button may be either the **Q** letter or the **W** letter or other symbol. This button defines the unit to measure the data entered.

- Letter **Q** on the button means that data are entered in cu m/s
- Letter **W** on the button means that data are entered in million cu m
- Other symbol on the button means the situation when this button defines nothing and affects nothing.

By default, when you start the module, the button for assigning measurement units is in the **W** mode. If for example we have the **Q** letter on the button and want to change it to the **W** letter, then we should point to the button and click the left mouse button. The button is not pressed, but the label on it changes. After the completion of entering units for measuring the data entered, point to any node on your calculation scheme located on the working area. At this moment, a measurement unit assigned for the given node is recorded in the output file that is yet located in the random access memory and corresponds to the node type.

Entering Parameters for Nodes in Arcs

To enter parameters in a node of any type you should activate corresponding tables of data entry. This mode is achievable only when the **E** button is pulled. Point to the node on the calculation scheme. At that individual messages on the screen change in accordance with the type of the node:

- A green cursor should appear near the node;
- In the national identity dialog box you will see an abbreviated country name (for our test problem **Uzb**);
- In the external name dialog box you will see an external name assigned to the node;
- In the node type dialog box you will see a type and an index ordinal number of the node;
- In the area of data entry tables corresponding tables of data entry will appear (for the **Inflow** node Figure 11, for the **Reservoir** node Figure 12, for the **Water Diversion** node Figure 13, for the **River Mouth** node Figure 14, for the **Lag Time** node Figure 15, for the **Channel Losses** node Figure 16, for the **Control and Downtime** node Figure 17).

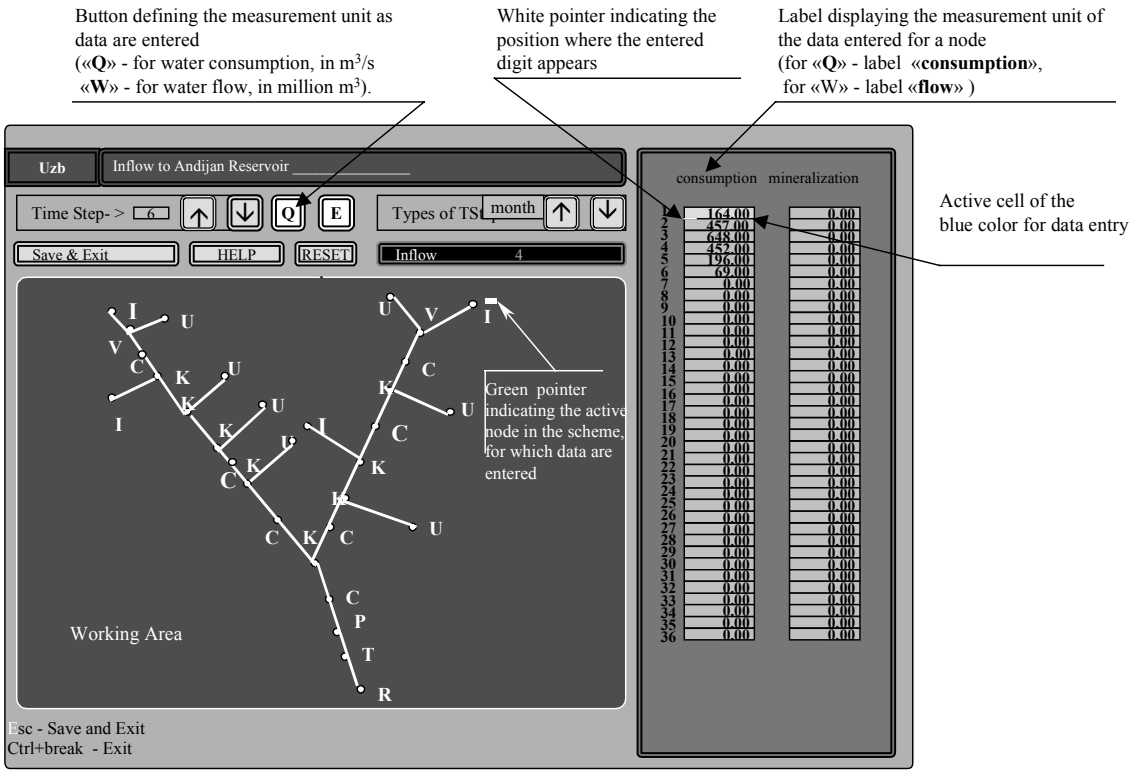


Figure 11. Working Window for the DATA ENTRY Module for the Inflow Node.

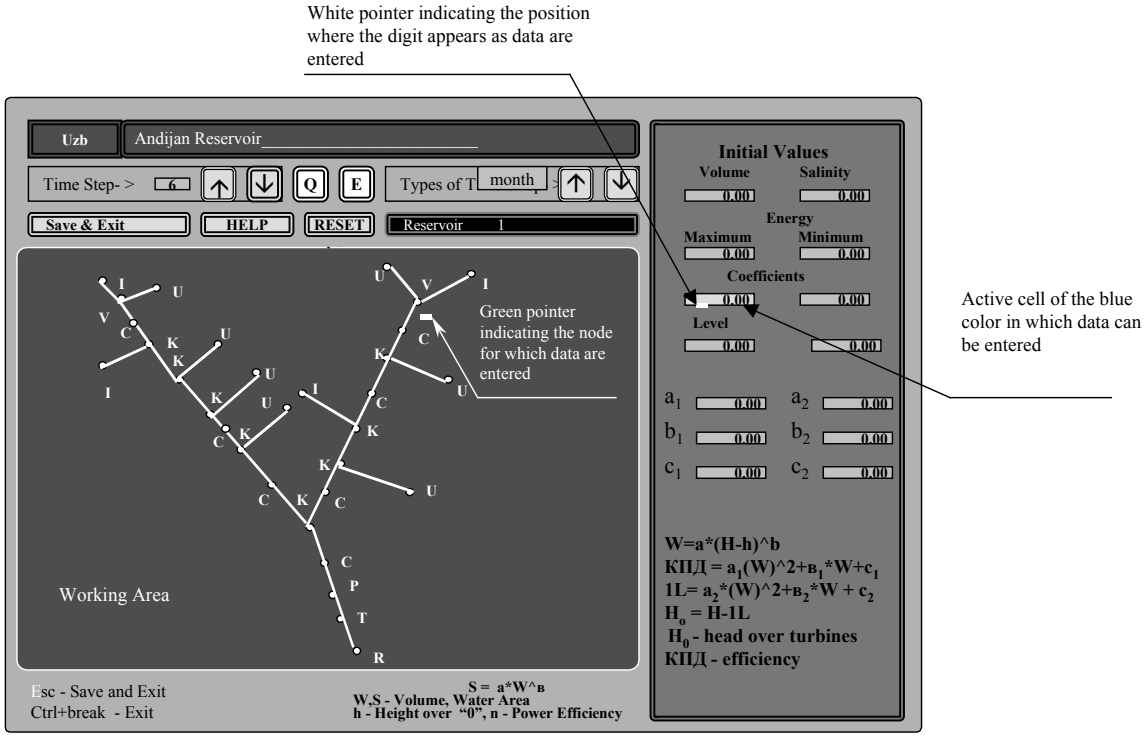


Figure 12. Working Window in the DATA ENTRY Module for the Lake Node

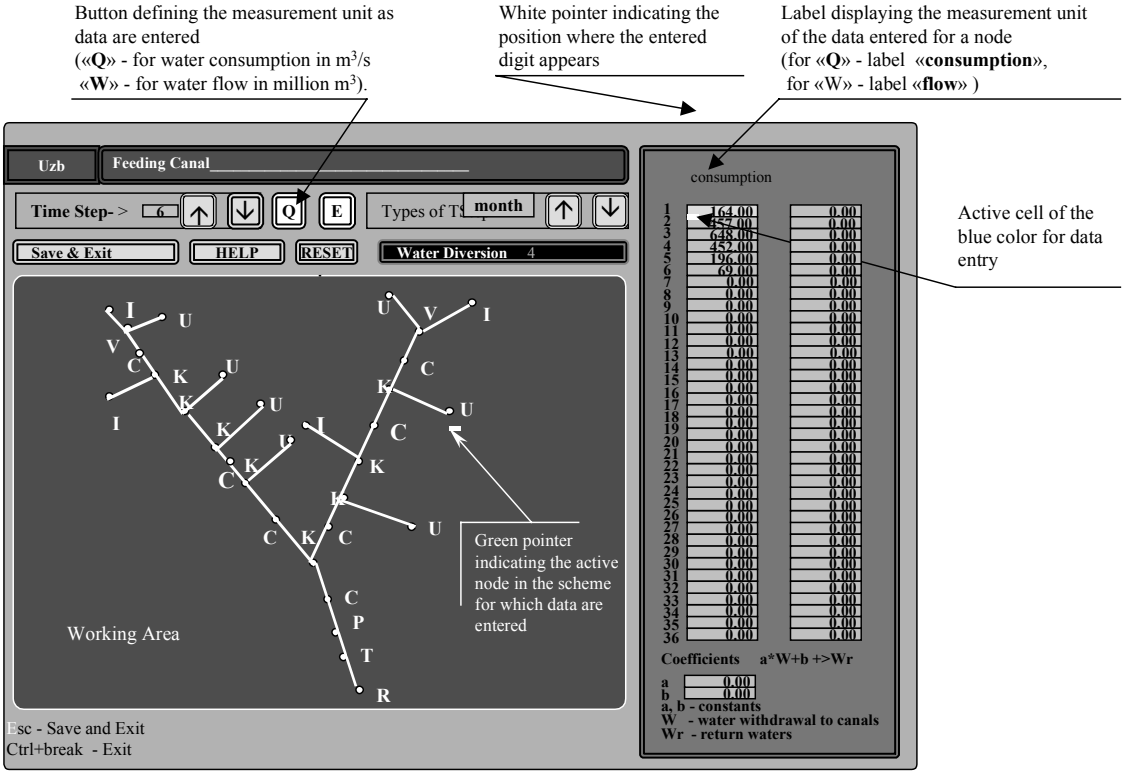


Figure 13. Working Window for the DATA ENTRY Module for a Water Diversion Node

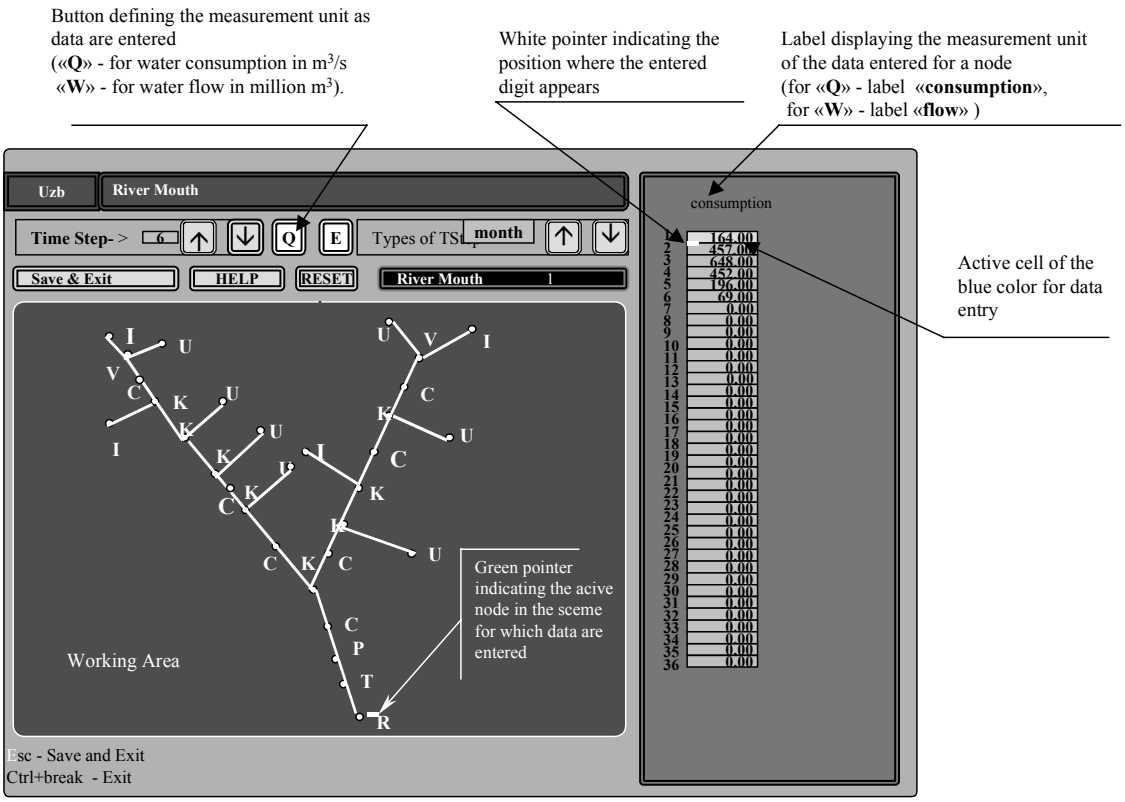


Figure 14. Working Window for the DATA ENTRY Module for the River Mouth Node

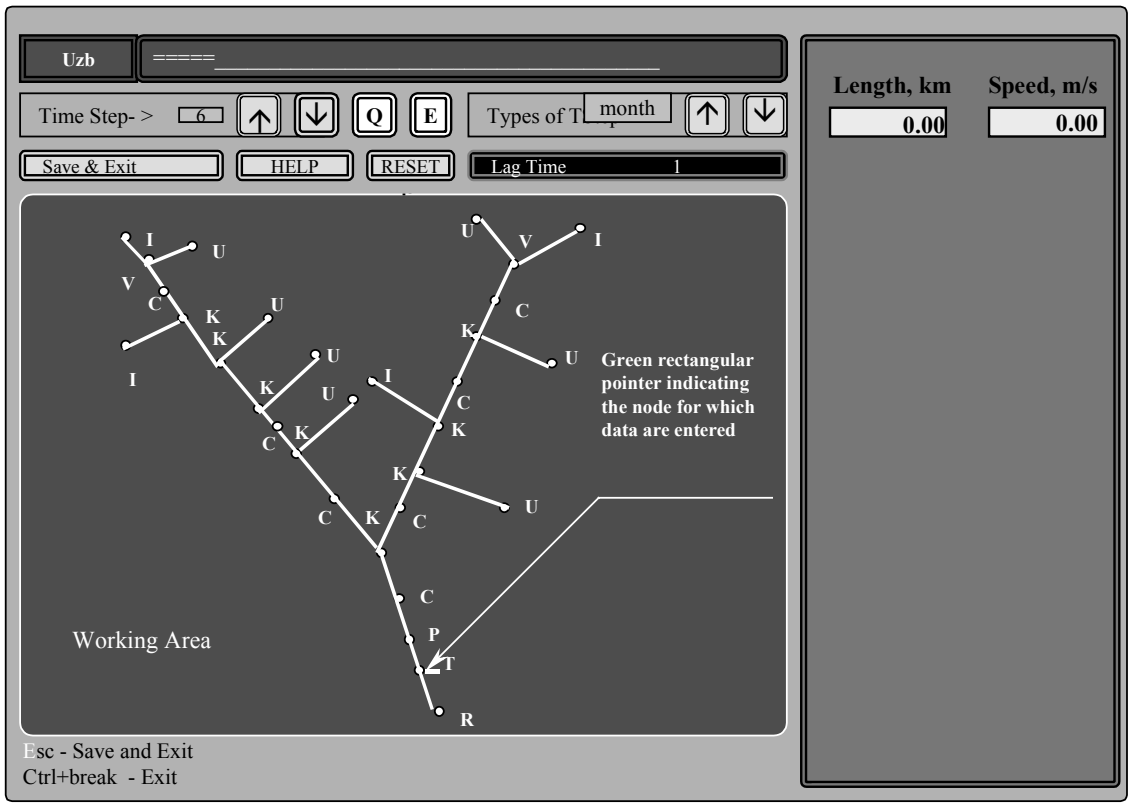


Figure 15. DATA ENTRY Module for a Simple Node and the Lag Time Node

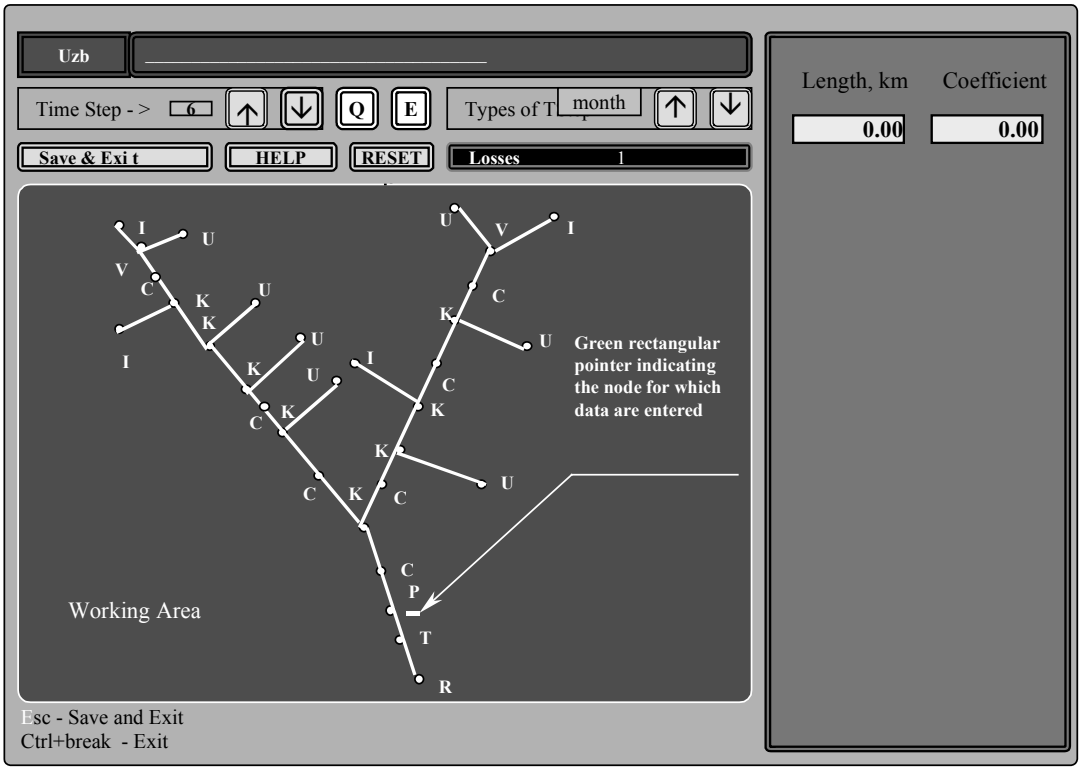


Figure 16. DATA ENTRY Module for the Losses Node

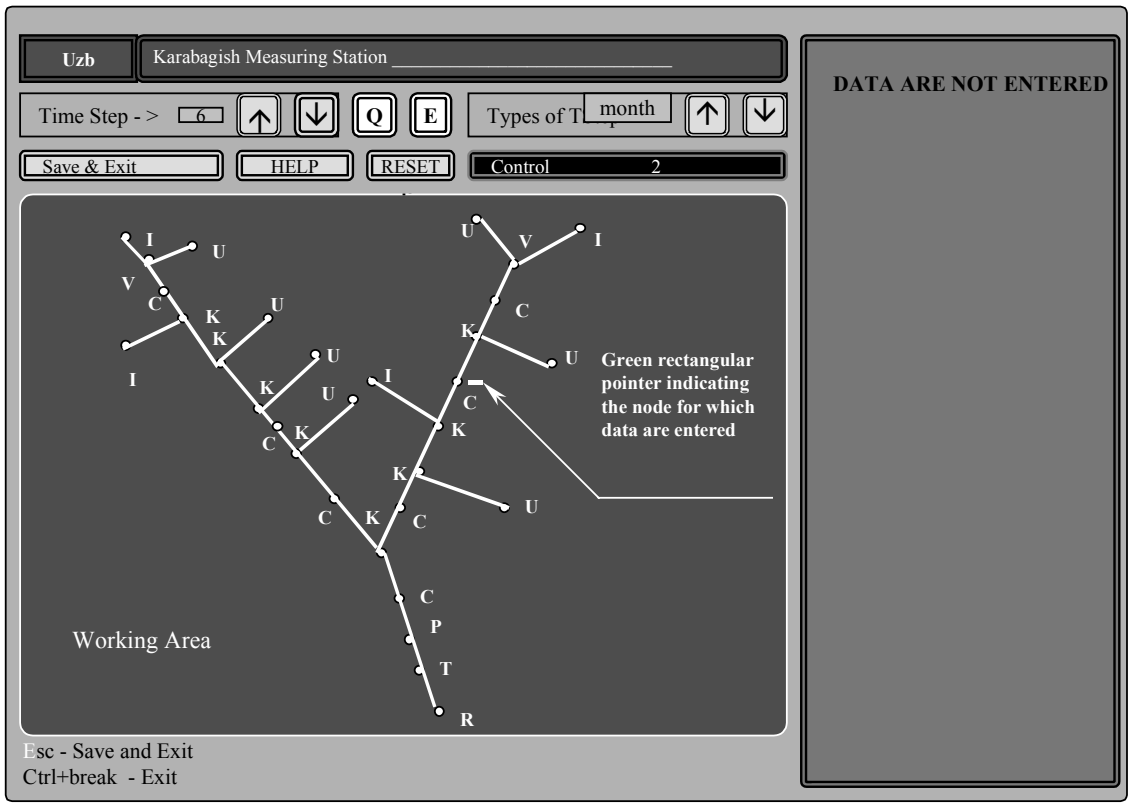


Figure 17. DATA ENTRY Module for the Simple Node and the Control Node.

You enter data in tables. If the white pointer indicating the position where the entered digit will appear is located in a wrong table or cell, then you can move it to the cell you need using command keys on the keyboard (↑↓→ and ←).

The pointer marks the entry position, which changes as the user presses command keys and the **Ctrl** key simultaneously. Until entry completion, all other operation modes are dead. The user completes entering a digit by pressing **ENTER** or leaving the area of the entered digit. If the user cannot go to another entry cell, it means that the program system cannot interpret the digits entered and therefore the user has to correct these digits.

When you enter data for nodes **Inflow**, **Water Diversion**, and **River Mouth**, you can enter data in various measurement units. You can assign water flow for a time step in million cu m (on the button defining the measurement unit of the data entered should be the **W** letter). You can enter data as water consumption in cu m/s (on the button defining the measurement unit of the data entered should be the **Q** letter). You can assign measurement units of the data entered both before and after data entry. In this case, entry dimension will belong to that node which dialog boxes of the module interface display.

It is desirable to begin with entering data for the nodes where data are water consumption, and then enter data for the nodes where data are water flow for a time step. In this case, it is not necessary to constantly toggle entry dimensions by the **W/Q** button. In the area for digits, entry the user is able to enter ten digits and a dot separating a fractional part of a number. However,

for **river mouth** nodes, the user can use the minus sign. A special algorithm checks the presence of minuses in data for nodes of this type and makes the model interpret this time step as the time step during which water in this river mouth is not desirable.
This feature allows the user to identify points of unproductive and undesirable water losses caused only by technical characteristics of management.

See the structure of output files below.

Changing the measurement unit of the data entered in the program has an automated change of table headings. When you changed the **W** letter to the **Q** letter on the button, then the label above the table that specifies the data measurement unit change where it makes sense and is necessary.

The Structure of the Output File *Time Steps*

The **Time Steps-step.in** output file contains information on the number of time steps adopted in calculation, and the type of time step. Data are given in numerical indexes.

Below we give the full text of the **TIME STEPS-step.in** file for our test problem.

No. of Line	**Data Given in Each File Line**
First File Line	¦ number of time steps (1->36)¦ ¦ hour-1,day -2,ten-day period -3,month -4 (1->4)¦
Second File Line	6 4

The first file line is a comment line. In it, we describe the meaning of numerical indexes adopted in this package of programs. These numerical indexes are located and used in the second line of the **Time Steps** file.

The first digit of the second line of the file displays the number of time steps.
The second digit of the second line is the numerical index of the type of time step

1 – for time step **hour**
2 – for time step **day**
3 – for time step **ten-day period**
4 – for time step **month**

The Structure of the Output File *Evaporation*

The **Evaporation-evapor.in** file consists of five lines. The first three lines are the comment lines. The second two lines contain entered values of evaporation and conversion coefficients equal to the number of seconds in the time step divided by a million.

No. of File Line	**Data Given in Each File Line**	**Note**

1	First three lines – comment lines. Lines 4 and 5 – values of coefficients.	Comment
2	First line of digits – evaporation in meters during the time step	Comment
3	Second line of digits – value of conversion coefficients	Comment
4	1.00 1.00 1.00 … 1.00 1.00 1.00	Evaporation for 36 Time Steps
5	2.592 2.680 2.592 … 2.592 2.592 2.592	Conversion Coefficients for 36 Time Steps

The package of programs works so that only the first **N** coefficients in each line of the **Evaporation** file are active (**N** – number of time steps). However, the file should have 36 values in each line for operability of programs.

The Structure of the Output File for the Inflow Node

The **Inflow-supplies.in** output file has the following structure: digit blocks of 38 lines each follow the first file line, which is a comment about the file.

In the first file line, the first digit is the number of nodes of a given type and so of blocks in the calculation scheme. Further, the comment follows.
The number of blocks equals the first digit in the first file line.

In the first line of the block (each block consists of a two-columns table and 38 lines):

- Digit means the ordinal number of the **Inflow** node in the calculation scheme and correspondingly the number of the block;
- Next two digits are coordinates of a given node in the calculation scheme;

These coordinates are determining for restoration modes and for saving data from prior corrected schemes (See the data restoration mode).

- Then, the => symbol should follow;
- Next the external node name is given.

In the second line of the block

- The first digit is the numerical index of measurement units of the data entered;
- The second digit means the numerical index of the type of the time step;
- Then the reference text on the indexes adopted follows.

The digital area of each block consists of two columns:

First column – data on consumption (or flow) of water depending on the first digit in the second line of each block;
Second column – data on salinity in g/l.

For our test problem, the **Inflow** output file has the following structure.

No. of Block	No. of Block Line	*Data Given in Each File Line*
		4 Data on Water Inflow and Salinity
Block 1	1	1 430 63 => Inflow to Andijan Reservoir___
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	164 0.00
	4	457 0.00
	5	648 0.00
	6	452 0.00
	7	196 0.00
	8	69 0.00
	9	0.00 0.00
	…….	……..
	38	0.00 0.00
Block 2	1	2 364 147 => Tributary Inflow on Kara Darya_____
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	140 0.00
	4	177 0.00
	5	207 0.00
	6	153 0.00
	7	114 0.00
	8	97 0.00
	9	0.00 0.00
	……	……..
	38	0.00 0.00
Block 3	1	3 128 70 => Inflow to Toktogul Reservoir
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	172 0.00
	4	508 0.00
	5	1224 0.00
	6	1312 0.00
	7	760 0.00
	8	475 0.00
	9	0.00 0.00
	…….	…….
	38	0.00 0.00
Block 4	1	4 145 152 => Tributary Inflow on the Toktogul-Uchkurgan Reach
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	288 0.00
	4	298 0.00

	5	190 0.00
	6	96 0.00
	7	188 0.00
	8	177 0.00
	9	0.00 0.00
	…….	…….
	38	0.00 0.00

The package of programs works so that only the first **N+2** lines of each block of the **Inflow** output file are active (**N** is the number of set time steps). The below lines are not recognized by the program, but they should be in the file for correct operation of the program.

The Structure of the Output File *Reservoir*

The **Reservoir-lakes.in** output file has the following structure: blocks of 44 lines each follow the first file line, which is the full filename.

In the first file line, the first digit is the total number of nodes-blocks of the **Reservoir** type in the calculation scheme. Next, the comment follows. Each block correlated to one of reservoirs has a uniform structure.

In the first block line:

- Digit shows an ordinal number of a given **Reservoir** node in the calculation scheme;
- Next two digits are coordinates of a given node in the calculation scheme;

These coordinates are determining for restoration modes and for saving data from prior corrected schemes (See the data restoration mode).

- Then the => symbol follows;
- Then the external name of the **Reservoir** node is given.

In the second line of the block, we give the description of what digits in block lines from 3 to 9 stand for, in particular:

- **Line 3** – initial volume and salinity
- **Line 4 –** upper available power generation and lower power generation
- **Line 5 –** morphological coefficients **a** and **b** functionally connecting water volume and water level through a power function
- **Line 6 –** difference in meters between the position of zero of the power function that describes the reservoir's morphology and zero elevation, regarding which the calculation is carried out (approximately a little less than the 1 meter elevation of the lowest point of the reservoir bottom). Second digit in this line can be 1 or 0; it shows whether power generation is calculated.
- **Line 7 –** coefficients **a1** and **a2**;
- **Line 8 –** coefficients **b1** and **b2**;

– **Line 9 –** coefficients **c1** and **c2**;

These coefficients are regression coefficients that define the following relationships:

Efficiency = **a1***(H-h_0)*(H-h_0) + **b1***(H-h_0) + **c1**
h_0= **a2***(Q*Q) + **b2***Q + **c2**

where

Efficiency = efficiency of turbines;
h_0 = downstream level of the hydroelectric power station;
H = upstream level of the hydroelectric power station;
Q = water flow through turbines to the downstream.

All specified formulas automatically appear in the form of a comment under the data entry table. For our test problem, the **Reservoir** output file has the following structure.

No. of Block	**No. of Block Line**	***Data Given in Each File Line***
		2 information on reservoir, volume, morphology, and power generation
Block 1	1	1 463 96 => Andijan Reservoir
	2	Initial volume and salinity/ upper and lower power generation / coefficients **a** and **b**/ h_0 and efficiency of turbines
	3	633 0. 00
	4	0.00 0. 00
	5	3.2317348879 0.0004219444
	6	799.00 1. 00
	7	0.00 0.00
	8	0.00 0. 00
	9	0.80 765.49
Block 2	1	2 150 96 => Toktogul Reservoir
	2	Initial volume and salinity/ upper and lower power generation / coefficients **a** and **b**/ h_0 and efficiency of turbines
	3	7247 0.00
	4	10000.00 0. 00
	5	2.1910453827 0.3757732247
	6	757.92 1
	7	0.00 0.00
	8	0.00 0.00
	9	0.85 735. 00

The package of programs works so that all lines of each block in the Reservoir-lakes.in output file are active.

The Structure of the Output File *Water Diversion*

The **Water Diversion** output file has the following structure: blocks of 41 lines each follow the first file line that is the full filename.

In the first file line, the first digit – the number of the **Water Diversion** nodes (and so of data blocks) in the calculation scheme. Then the comment follows.

In the first block line:
- digit means the ordinal number of a given node type
- then certainly follows the => symbol
- then the external node name is given

In the second block line:
- the first digit is the numerical index of measurement units of entered data
- the second digit is the numerical index of the type of the time step
- then the reference text on adopted numerical indexes follows

From 3rd to 38th block lines
- the first digit is the ordinal number of the time step
- the second digit is water consumption (flow) declared as a water claim
- the third digit is salinity of return waters

39th block line
- comment = reset ratios

40th block line
- the first digit is the ordinal number
- the second digit is the value of the **a** coefficient in the formula for calculating return waters
- the third digit means nothing, but it should be in the file for correct operation of the program. It can be of any value. We assigned 777 to it.

41st block line
- the first digit is the ordinal number
- the second digit is the value of the **b** coefficient in the formula for calculating return waters
- the third digit means nothing, but it should be in the file for correct operation of the program. It can be of any value. We assigned 777 to it.

For our test problem, the **Water Diversion** output file has the following structure.

No. of Block	**No. of Block Line**	***Data Given in Each File Line***
		7 Data on water diversions, water withdrawal and return flow
Block 1	1	1=> Losses from Andijan Reservoir

	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 10.0 5
	4	2 10.0 5
	5	3 10.0 5
	6	4 10.0 5
	7	5 10.0 5
	8	6 10.0 5
	9	7 0.001 5
	…….	……..
	38	36 0.001 5
	39	Reset Ratios
	40	37 0,00 777
	41	38 0,00 777
Block 2	1	2=> Feeding Canal_____
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 43 5
	4	2 286 5
	5	3 119 5
	6	4 214 5
	7	5 188 5
	8	6 76 5
	…….	……..
	38	38 0.001 5
	39	Reset Ratios
	40	37 0.00 777
	41	38 0.00 777
Block 3	1	3=> Water Diversion on the Andijan-Uchtepa reach
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 74 5
	4	2 73 5
	5	3 112 5
	6	4 142 5
	7	5 122 5
	8	6 44 5
	…….	…….
	38	36 0.001 5
	39	Reset Ratios
	40	37 0.00 777
	41	38 0.00 777
Block 4	1	4=> Losses from Toktogul Reservoir

	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 10 5
	4	2 10 5
	5	3 10 5
	6	4 10 5
	7	5 10 5
	8	6 10 5
	…….	…….
	38	36 0.001 5
	39	Reset Ratios
	40	37 0.00 777
	41	38 0.00 777
Block 5	1	5=> Water diversion on the Toktogul-Uchtepa reach
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 178 5
	4	2 177 5
	5	3 192 5
	6	4 256 5
	7	5 259 5
	8	6 197 5
	…….	…….
	38	36 0.001 5
	39	Reset Ratios
	40	37 0,00 777
	41	38 0,00 777
Block 6	1	6=> Channel Losses
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 1 5
	4	2 1 5
	5	3 1 5
	6	4 1 5
	7	5 1 5
	8	6 1 5
	…….	…….
	38	36 0.001 5
	39	Reset Ratios
	40	37 0,00 777
	41	38 0,00 777
Block 7	1	7=> Water diversion 2 on the Toktogul-Uchtepa reach

	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 20 5
	4	2 30 5
	5	3 20 5
	6	4 30 5
	7	5 30 5
	8	6 25 5
	…….	……
	38	36 0.001 5
	39	Reset Ratios
	40	37 0,00 777
	41	38 0,00 777

The package of programs works so that only the first **N+2** lines of each block of the **Water Diversion** output file and the lines from 40 to 41 of each block are active (**N** is the number of set time steps). The program will not recognize lines from N+2 to 39 (i.e. it is not important what numerical values are entered in these lines), but these lines should be in the file for correct operation of the program.

The Structure of the Output File *River Mouth*

The **River Mouth** output file has the following structure: blocks of 38 lines each follow the first file line that is the full filename.

In the first file line, the first digit – the number of the **River Mouth** nodes in the calculation scheme. Then the comment and data on the parameters entered into tables follow.

In the first block line:
- digit means the ordinal number of the internal name of a given node
- then the => symbol should follow
- then the external node name is given

In the second block line:
- the first digit is the numerical index of measurement units of entered data
- the second digit is the numerical index of the type of the time step
- then the reference text on adopted numerical indexes follows

From 3rd to 38th block lines
- the first digit is the ordinal number of the time step
- the second digit is the entered value of water release requirements in the river mouth

If requirements have negative values, the model will automatically convert them into positive values. However, the model's objective will be to bar water flow during a given time step.

In our test problem we did not provide any requirements or constraints on consumption of water flowing to the river mouth, therefore this file was formed with assigned zero values. However, you should take into account that in the future while creating a GAMS model and running the **BUILDING GAMS MODEL** module water requirements for the river mouth will be assigned equal to 10,000 million cu m.

No. of Block	**No. of Block Line**	***Data Given in Each File Line***
		1 Data on the River Mouth in the System
Block 1	1	1=> River Mouth
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month /length, speed
	3	1 0.00
	4	2 0.00
	…….	……..
	38	36 0.00

The package of programs works so that only the first **N+2** lines of each block of the **River Mouth** output file are active (**N** is the number of set time steps). The program will not recognize the block lines below (i.e. it is not important what numerical values are entered in these lines), but these lines should be in the file for correct operation of the program.

The Structure of the Output File *Losses*

The **Losses-losses.in** output file contains parameters necessary to calculate channel losses along the river length.

The **Losses** output file has the following structure: blocks of 3 lines each follow the first file line that is the full filename and a comment.

In the first file line, the first digit – the number of **Losses** nodes in the calculation scheme. Then a comment about data and parameters, which are entered in block tables, follow.

In the first block line:
- digit means the ordinal number in the internal node name
- then the => symbol should follow
- then the external node name is given

In the second block line:
- the first digit is the numerical index of measurement units of entered data (in this case it does not influence calculation)
- the second digit is the numerical index of the type of the time step
- then the reference text on adopted numerical indexes follows

In the third block line

- the first digit is the ordinal number of the time step
- the second digit is the entered value of length of a river reach in km, for which channel losses are calculated
- the third digit is an empirical coefficient of linear relationship between the channel width and water consumption for a calculated river reach

In the fourth block line
- the first, second, and third digits are a spare place

The package of programs reads only the first two block lines. In other lines (from the 3rd to the 4th), entered parameters can have any numerical values. It is only important to retain the file structure.

In our test problem, we do not calculate channel losses, but enter them as water withdrawal. Therefore, by default all values equal zero. The **Losses** file for our problem looks as follows.

No. of Block	**No. of Block Line**	***Data Given in Each File Line***
		1 Calculation of losses (+) or infiltration (-) [L, k] Losses = L*K*E(i)
Block 1	1	1 336 339 => Point of calculation of channel losses
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month
	3	1 0.00 0.00
	4	2 0.00 0.00

The package of programs works so that only the line for the first time step in each block of the **Losses** output file is active. The program will not recognize the block line below (i.e. it is not important what numerical values are entered in this line), but this line should be in the file for correct operation of the program.

The Structure of the Output File *Lag Time*

The **Lag Time-lagtime.in** output file contains parameters necessary to take into account the lag time of flowing from one node to the next one with the stream. We recommend to consider lag time only for the **day** and **hour** time steps.

The **Lag Time** output file has the following structure: blocks of 3 lines each follow the first file line that is the full filename.

In the first file line, the first digit – the number of the **Lag Time** nodes in the calculation scheme. Then data on parameters, which are entered in tables, follow.

In the first block line:
- digit means the ordinal number in the internal node name
- then the => symbol should follow

– then the external node name is given

In the second block line:
– the first digit is the numerical index of measurement units of entered data
– the second digit is the numerical index of the type of the time step
– then the reference text on adopted numerical indexes follows

In the third line:
– the first digit is the ordinal number of the time step
– the second digit is the entered value of length of a river reach in km, for which time is calculated
– the third digit is average speed of flow on a calculated river reach in m/s

In our test problem, the type of the time step is **month**. Therefore, we do not calculate the lag time. The **Lag Time** file for our problem looks as follows.

No. of Block	**No. of Block Line**	***Data Given in Each File Line***
		1 Calculation of losses (+) or infiltration (-) [L, k] Losses = L*K*E(i)
Block 1	1	1 333 369 => Point of calculation of channel losses
	2	1 4 (flow -1) (consumption 1) salinity; 1-hour, 2-day, 3-ten days, 4-month /length, speed
	3	1 3000.00 0.25
	4	2 0.00 0.00

The package of programs works so that only the line for the first time step in each block of the **Lag Time.lagtime.in** output file is active. The program will not recognize the block lines below (i.e. it is not important what numerical values are entered in these lines), but these lines should be in the file for correct operation of the program.

9. Reservoir Parameters Module

The **RESERVOIR PARAMETERS-rfigures.exe** module serves to enter data for each reservoir in the calculation scheme of a river network. It also serves to calculate empirical coefficients you need to do calculations in the GAMS model. In the module, you enter water levels of the reservoir and respective reservoir volumes in million cu m, and water tables in sq. km. The execution of the **RESERVOIR PARAMETERS** module results in recording empirical coefficients **a, b** and $\mathbf{h_0}$ in the **Lakes-lakes.in** output file. These coefficients connect the storage of reservoir with its water level

$$W = a^* (H - h_0)^* b$$

where W – storage of a reservoir;

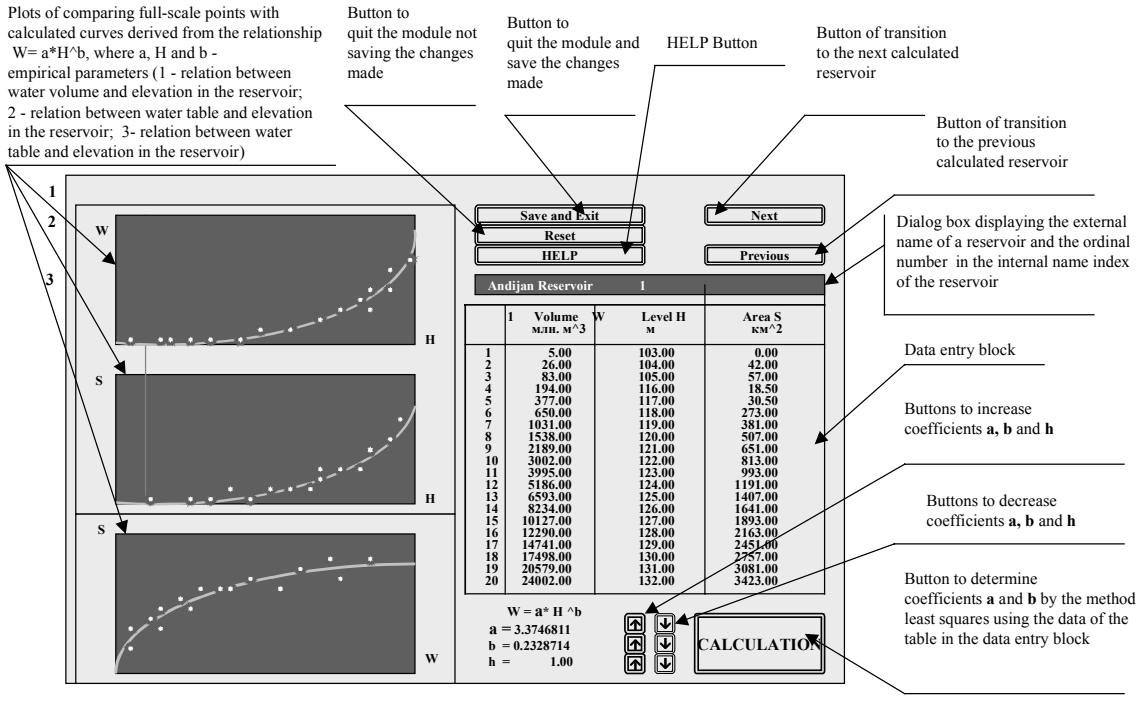
H – absolute water level of a reservoir;
Ho – absolute zero storage of a reservoir;
a, b – empirical coefficients;

A special embedded algorithm calculates empirical coefficients by the method of least squares.

To update empirical coefficients you have to run the **RESERVOIR PARAMETERS** module each time when you have changed the river reaches that incorporate reservoirs in the calculation scheme, or if you have added or deleted the **Lakes** nodes in the calculation scheme. You have to do it because when you enter such changes, the ordinal number in the internal node name as well as connections of the **Lakes** node with other nodes may change.

The **RESERVOIR PARAMETERS** module has a constraint on the number of data samples incorporated in the statistical calculation for obtaining empirical coefficients. They should be exactly 20. Therefore, if you need enter more data in a statistical calculation, then you can use other program packages for statistical calculations and enter obtained coefficients in the **Lakes** output file, directly in the **DATA ENTRY** module. In this case, the **RESERVOIR PARAMETERS** module does not start, and this is its **optionality**.

You start the **RESERVOIR PARAMETERS** module from the interface by pressing the corresponding button using the left mouse button. As you activate the module, an image appears on the screen (Figure 18).



1	Volume млн. м^3	W	Level H м	Area S км^2
1	5.00		103.00	0.00
2	26.00		104.00	42.00
3	83.00		105.00	57.00
4	194.00		116.00	18.50
5	377.00		117.00	30.50
6	650.00		118.00	273.00
7	1031.00		119.00	381.00
8	1538.00		120.00	507.00
9	2189.00		121.00	651.00
10	3002.00		122.00	813.00
11	3995.00		123.00	993.00
12	5186.00		124.00	1191.00
13	6593.00		125.00	1407.00
14	8234.00		126.00	1641.00
15	10127.00		127.00	1893.00
16	12290.00		128.00	2163.00
17	14741.00		129.00	2451.00
18	17498.00		130.00	2757.00
19	20579.00		131.00	3081.00
20	24002.00		132.00	3423.00

Figure 18. The Image of the Working Window on the Screen as You Enter the RESERVOIR PARAMETERS Module

You operate the **RESERVOIR PARAMETERS** module in accordance with designations of buttons. Pressing the left mouse button activates all buttons.

Entering Data

As you run the **RESERVOIR PARAMETERS** module for the first time, the **Reference Data-lakes.vol** output file is formed. In this file, reference data on water levels and respective water volumes and water tables for each reservoir are assigned by default. After that, you should make corresponding changes in your data on reservoir parameters. You can do it either directly in the **RESERVOIR PARAMETERS** module or make changes in the **Reference Data** output file (see the structure of the output file below).

You go from reservoir to reservoir using buttons **Next** and **Previous**. When you go to another reservoir, changes in the data entry block and calculation results for a current calculated reservoir are saved.

In the dialog box located above the data entry block, you should always see the external name of the **Lakes** node and its numerical index in the internal name. Numerical indexes are assigned to **Lakes** nodes in accordance with the **Network in Indexes 2-out.out** output file.

The Structure of the Output File *Reference Data*

The **Reference Data-lakes.vol** output file is formed for an updated calculation scheme of a river network as you run the **RESERVOIR PARAMETERS** module for the first time.
The **Reference Data-lakes.vol** output file consists of blocks, 28 lines each for each **Lakes** node. This file has as many blocks as there are **Lakes** nodes in the calculation scheme.
As the program system forms the Reference Data file, data on water levels and respective water volumes and water tables for each reservoir are entered by default.

In the first block line

- The first digit – numerical index in the internal name in accordance with the **Network in Indexes 2** output file; the second and third digits are coordinates (x, y) of the node on the screen in accordance with the **Network in Indexes 2** output file.
- Obligatory symbol =>
- External node name

In the second block line

- Value of empirical coefficient **a** (In the first run it is entered by default. After calculation of empirical coefficients (see below), its value is changed to the calculated value.)
- Value of empirical coefficient **b** (In the first run it is entered by default. After calculation of empirical coefficients (see below), its value is changed to the calculated value.)
- Relative (or absolute) zero elevation, in meters h_0
- Obligatory symbol =>
- Reference on designation of the first three digits in the line

Third, fourth, and fifth lines – supplementary lines. Here may be any text. By default a table heading with column names is given.

From the sixth to the twenty-fifth line – values of reference data (three columns) and values for calculated curves (next three columns). When you enter data, the last three columns are not necessary.

The first column is the ordinal number.
The second column is measured values of the reservoir volume.
The third column – measured values of the reservoir level.
The fourth column is reference values of the reservoir water table. The fourth column is not necessary because it is a derivative from the second and third columns. A special algorithm always adjusts it in accordance with the second and third columns. In numerical differentiation the Newton algorithm is used.
The fifth column is calculated values of the reservoir volume in relationship W=f(L).
The sixth column is calculated values of the reservoir water table in relationship S=f(L).
The seventh column is calculated values of the reservoir water table in relationship S=f(W).

W – reservoir volume (million cu m),
L – reservoir water level (m),
S – water table (cu km)

The fifth, sixth, and seventh columns are reference columns, and their existence or even the condition of records in them in no way affect operability of the program system. The user, however, can always discern a difference between actual measurements and those morphological figures that will be used in the model's calculations.

From the twenty-sixth to the twenty-eighth line – restrictive lines. By default, this is the frame end and symbols -. These lines can include any symbols, but they should not be empty.

Below is presented a data file to our test problem put through the **RESERVOIR PARAMETERS** module.

1 Andijan Reservoir
3.2317348879 0.0004219444 799.00 <= a, b, h_0

No.	Volume	Level	Area	Calculated Volume	Calculated Area	Calculated Area
	W	L	S	W (L)	S (L)	S (W)
1	0.000	800.000	0.000	0.00	0.00	0.00
2	100.640	854.000	3.727	177.68	2.31	7.05
3	200.640	855.000	3.636	188.34	10.65	11.35
4	300.270	860.000	33.210	248.29	11.98	15.00
5	400.180	867.000	16.652	352.73	14.90	18.29
6	500.730	871.000	18.282	424.29	17.88	21.35
7	600.640	876.000	22.202	527.10	20.55	24.21

8	700.270	879.000	24.907	596.40	23.10	26.92
9	800.330	885.000	22.236	753.43	26.16	29.52
10	1000.000	893.000	28.524	1004.34	31.34	34.43
11	1211.110	898.000	32.478	1187.46	36.61	39.29
12	1300.000	902.000	19.753	1349.62	40.53	41.26
13	1400.110	904.000	33.370	1436.16	43.27	43.43
14	1450.000	905.000	33.260	1480.84	44.67	44.50
15	1470.000	907.000	13.333	1573.05	46.10	44.92
16	1500.000	909.000	5.000	1669.15	48.05	45.55
17	1600.330	910.000	66.887	1718.69	49.54	47.63
18	1700.670	911.000	100.340	1769.23	50.54	49.68
19	1800.000	912.000	99.330	1820.79	51.56	51.66
20	2000.000	914.000	133.333	1927.01	53.11	55.56

2 Toktogul Reservoir

2.1910453827 0.3757732247 757.92 <= a, b, h_0

No.	Volume W	Level L	Area S	Calculated Volume W (L)	Calculated Area S (L)	Calculated Area S (W)
1	0.000	759.000	0.000	0.44	0.90	0.00
2	25.400	765.000	8.467	27.34	4.39	8.13
3	108.000	770.000	15.018	88.20	12.14	17.87
4	1065.000	800.000	54.686	1359.11	41.86	61.99
5	3000.000	820.000	77.400	3186.40	91.24	108.84
6	3947.000	825.000	75.760	3775.85	117.88	126.35
7	4448.000	830.000	100.200	4420.05	128.83	134.83
8	5204.000	835.000	151.200	5119.75	139.93	146.84
9	7017.000	850.000	181.300	7558.89	162.56	172.74
10	8990.000	855.000	197.300	8487.43	185.70	197.65
11	9806.000	860.000	163.200	9474.73	197.46	207.21
12	10764.000	865.000	191.600	10521.35	209.32	217.98
13	11862.000	870.000	219.600	11627.85	221.29	229.80
14	13003.000	875.000	228.200	12794.73	233.37	241.56
15	13708.000	878.000	176.250	13524.06	243.11	248.59
16	14188.000	880.000	192.000	14022.52	249.23	253.29
17	14924.000	883.000	294.400	14788.62	255.37	260.35
18	16716.000	890.000	358.400	16662.72	267.72	276.90
19	18061.000	895.000	224.167	18076.07	282.67	288.80
20	19458.000	900.000	279.400	19552.18	295.22	300.74

Figures in the fourth column that correspond to areas of open water surface may shock the user. Their pulsation relates to the inaccuracy of measuring water volumes of reservoirs and respective water levels, which did not show up in a sharp manner. The user should take the fourth column

only as an index of accuracy and correctness of the second and third table columns, which determine calculations on reservoir morphology.

The user can correct data directly in the **RESERVOIR PARAMETERS** module. A special cursor moving between the first and the second graphs corresponds to the position of the cursor in the data entry table. If you changed reservoir parameters (water level, respective volume and water table of the reservoir) directly in the module, then all changes entered are automatically saved in the **Reference Data** file when you go to another reservoir. When you quit the module, the program system saves the changes you made.

Calculating Empirical Coefficients

After you have completed entering actual data for each reservoir in the data entry block directly in the module (or in the **Reference Data** file), you can calculate empirical coefficients.

If you click the left mouse button on the **GAMS Calculation** button, then you will obtain values of empirical coefficients **a** and **b** calculated through the method of least squares for the entire data selection. During this operation a disaster dump caused by the inaccuracy of reference data may occur. For instance, if you move from top downward, the data in the second and third columns should increase. Data irregularity can also cause a program failure. In this case, we recommend you to open the module again and repeat all operations not making blunders as you enter data. If a insoluble problem arises, delete the lakes.vol file using an external editor and repeat your calculation.

On the graphs in the left half of the screen, you will see reference data compared with a calculated curve. The upper graph presents the results of comparing the reservoir volume with the reservoir water level to determine their relationship. The middle graph presents the results of comparing the reservoir water table with the reservoir water level to determine their relationship. The lower graph presents the results of comparing the reservoir water table with the reservoir volume to determine their relationship.

The module provides an opportunity to change values of empirical coefficients **a**, **b**, and $\mathbf{h_0}$ decreasing or increasing them by special buttons. At that on comparison graphs, you will observe how a calculated exponential curve built according to new values of empirical coefficients changes its form. This opportunity allows the user to select coefficients **a**, **b**, and $\mathbf{h_0}$ in such a way that discrepancies between reference data and an empirical curve would be minimal in a needed range of water levels in the reservoir.

After you have completed calculating empirical coefficients for one of reservoirs, you need to click the left mouse button on either button **Next** or **Previous**. Obtained values of coefficients **a**, **b**, and $\mathbf{h_0}$ will be recorded in the **Reference Data** file**.**

Quitting the Module

The user can quit the module with saving the changes made for **reservoir** nodes or without saving them. To quit the module without saving you may press **Esc** on the keyboard or click the left mouse button on the **Cancel** button.

To quit the module with saving changes you need to click the left mouse button on the Save **button**. An image (Figure 19) will appear on the screen. Follow the instructions given on the screen. If you press the **Include in Calculation** button, all newly obtained calculation coefficients will be recorded in the **Lakes-lakes.in** output file and the **Reference Data-lakes.vol** file. If you press the **Ignore** button, all newly obtained calculation coefficients will be saved only in the **Reference Data-lakes.vol** file. This provides the optionality of the program because eventually only those coefficients that are in the lakes.in file are important for subsequent calculations. These coefficients can get in this file not only from this program system but also from using any other way not disturbing the operability of the entire program system.

By default (i.e. if you did not run the **RESERVOIR PARAMETERS** module nor enter your data directly in the **Lakes** file), empirical coefficients **a** and **b** are accepted equal to zero, what may later lead to emergency completion of the **GAMS** model's operation. These coefficients should be positive and, as the practice showed, be less than 5.

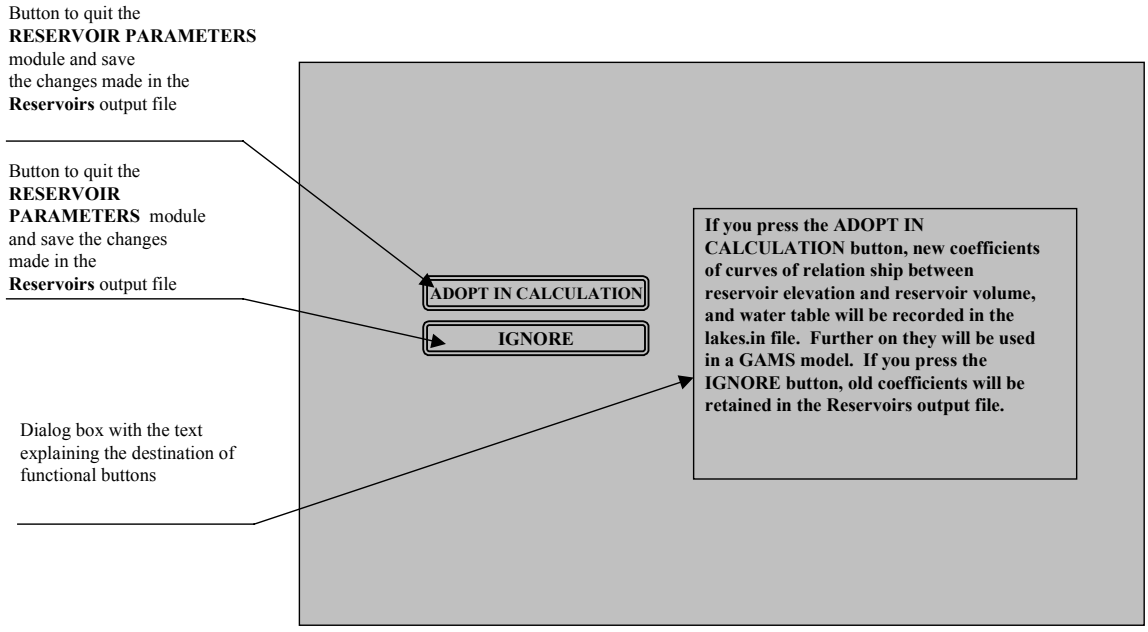


Figure 19. The Image of the Working Window on the Screen As You Press the Save and Exit Button in the RESERVOIR PARAMETERS Module.

Getting Help

To get Help in the **RESERVOIR PARAMETERS** module, you should click the left mouse button on the **Help** button. A reference text divided in three segments for convenient perception will appear on the left half of the screen. To quit the **Help** mode you need to press Esc on the keyboard.

10. Constraints File Module

Description of the Module

The **CREATING CONSTRAINTS FILE** module serves to create the **limitall** output file in which all variables on all links and entities are included. This file is based on information from the **klav.out** and **out.out** files. From the first file, the program system determines all arcs along which water flow is possible, and from the second file, it determines all reservoirs included in the calculation scheme. Before the module has created this internal database to the calculation scheme, it repeats a query about the need in this database and warns you about consequences.

Despite the fact that the system, which restores and saves prior entered data can restore an updated database, it can do it only if there is a saved copy of a previous database (See the **Data Saving and Restoration System** described below). Therefore, we recommend you to use the **CREATING CONSTRAINTS DATABASE** mode only once for each newly made calculation scheme because this is the most guaranteed way not to lose the constraints entered in the next module.

Description of the Constraints Database

The described module actually creates a table to place there all possible constraints peculiar to a given calculation scheme, but does not connect it to the **GAMS** model.

The upper line of the constraints database-limitall file is actually a copy of data about the number of different node types, except that the last but one digit shows not the total number of nodes, but the number of paired associations taken from the klav.out file. The last digit shows the number of time steps engaged.

Further, in the file two groups of blocks follow:

The first group of blocks describes constraints imposed on reservoir volumes incorporated in the calculation scheme. The number of these blocks equals the number of reservoirs. A block begins with a digital line that characterizes the digital code of a node and its coordinate location. The next line contains the internal node name. The following line contains the external node name, what makes this file easy for correction by an external editor.

The first, second and third lines in the block describing reservoirs are repeated. The table heading, in which explanations for the below digits are provided, occupies the next place.
The first column is the time step (all 36 possible time steps are defined).

The second, third and fourth columns of digits are minimum, maximum and fixed values of the reservoir volume.

The fifth, sixth, and seventh columns of digits may contain either 0 or 1.

The digit 0 means that a given constraint is not entered in a calculation model.
The digit 1 means that a specified constraint is incorporated in calculations.
The fifth column determines whether the second column participates or not.
The sixth column determines whether the third column participates or not.
The seventh column determines whether the fourth column participates or not.

Below is given a fragment of a block of this group.

```
   8   4   2   7   1   7   1   1  30   6
   3   1   1 463  96
 V1
 Andijan Reservoir_____________________
   3   1   1 463  96
 V1
 Andijan Reservoir_____________________
_______________volumes for lakes_____________________________
_____________________________________________________________
```

Time Step	Lower	Upper	Fixed	use not use	1 0
M1	150	1900	1000	1	1 0
M2	150	1900	1000	1	1 0
M3	150	1900	1000	1	1 0
....	.....	.....	.....	....	....

After the group of reservoir blocks goes a group of blocks that describes all paired associations described in the model.

The first group of blocks differs from the second only in a slightly changed form of the table heading and in the fact that lines 4, 5, 6 are not copies of lines 1, 2, 3 as it is described above. In this case lines 1, 2, 3 describe the initial node of a paired association, and lines 4, 5, 6 describe the final node of the association.

Below is given a fragment of a block of this group.

```
3   1   1      463       96
 V2
 Toktogul Reservoir__________
  6   1   1      441       123
 C4
  Point of control of releases from Toktogul Reservoir_______________
_______water flow on the arc_______________ between first and second nodes
_____________________________________________________________
```

Time Step	Lower	Upper	Fixed	use not use	1 0

M1	0	1000	577.57	0	0	1
M2	0	1000	442.20	0	0	1
M3	0	1000	380.93	0	0	1
M4	0	1000	771.84	0	0	1
M5	0	1000	889.76	0	0	1
M6	0	1000	631.96	0	0	1

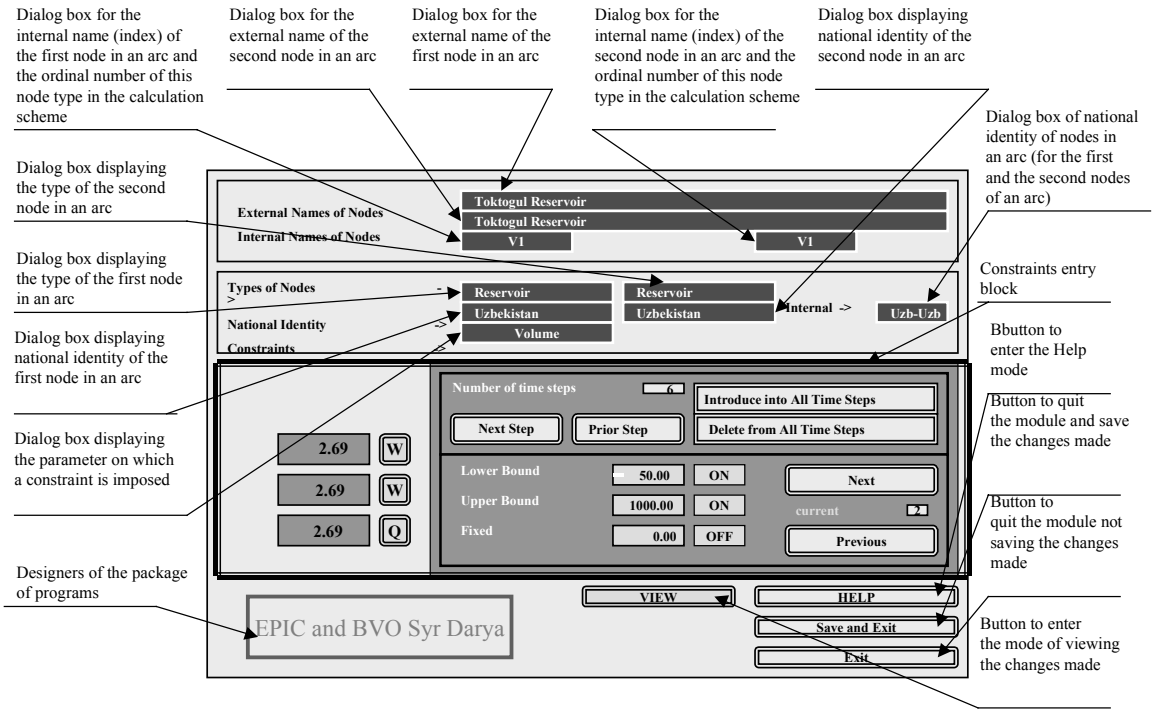


Figure 20. The Image of the Working Window on the Screen As You Enter the CONSTRAINTS ENTRY Module

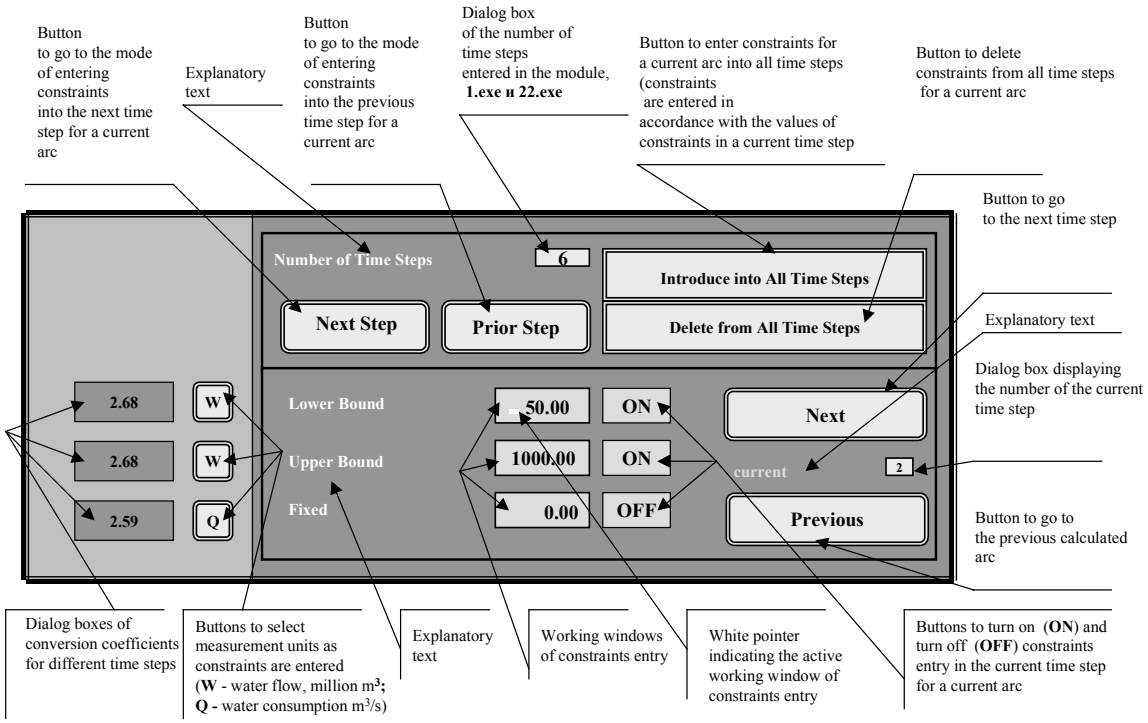


Figure 21. The Constraints Entry Block in the Working Window of the CONSTRAINTS ENTRY Module

Editing Constraints Database

Since the internal database file has a form of a table with needed explanations, you can edit this file using any external editor. In addition, you can edit this file by an internal editor, which starts if the you point at the **Constraints File 1-limitall** button and then press the left mouse button (see Figures 20 and 21).

11. Constraints Entry Module

Description of the Module

The **CONSTRAINTS ENTRY** module serves to change values of constraints in the **Constraints File 1-limitall** file entered by default in executing the **CREATING CONSTRAINTS FILE** module and creating the new **Constraints File 2** file which is directly used in the file of the GAMS model (in the file with the **.gms** extension).

The **CONSTRAINTS ENTRY** module is operable only if a corresponding version of the graphics driver in PASCAL, **egavga.bgi** is available in the same directory.

You run the **CONSTRAINTS ENTRY** module from the interface by pressing the corresponding button in the menu using the left mouse button. An image of Figure 21 should appear on the

screen (the constraints entry block is shown more detailed on Figure 22, which results when you press "**VIEW**" in the button on Figure 21.).

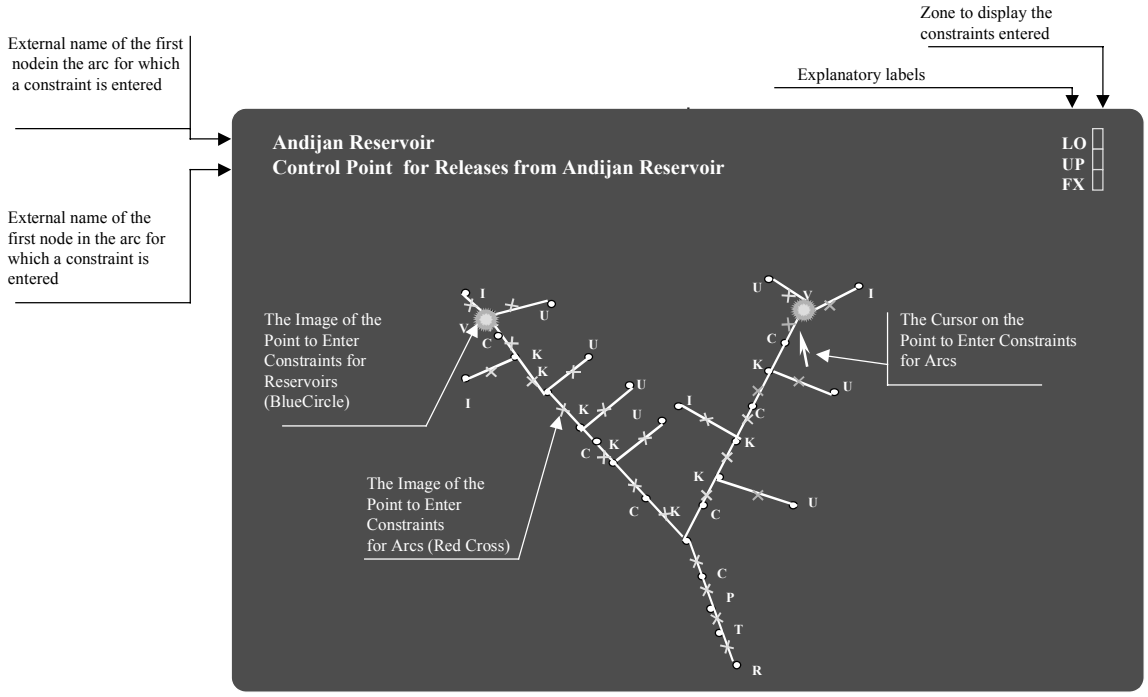


Figure 22. The Image of the Working Window on the Screen As You Press the View Button in the CONSTRAINTS ENTRY Module

You work in the **CONSTRAINTS ENTRY** module in accordance with the designation of buttons. Pressing the left mouse button activates all virtual buttons.

Labels in dialog boxes enable you to identify external names of nodes (or arcs) and numbers of time steps, for which you enter constraints.

To go from calculated arc to calculated arc you need to click the left mouse button on the **Next** or **Previous** button. In so doing, you actually move by blocks of the **limitall** database described above still being in the line of a specific time step.

To go from time step to time step you need to click the left mouse button on the Next Time Step or the **Prior Time Step** button. In so doing, you actually move by lines in a block of the **limitall** database described above.

Dialog boxes of conversion coefficients show values of the coefficient to recalculate water consumption into water flow for a time step, adjusted for variation of number of days in months and ten-day periods. Values of these coefficients were entered in the **DATA ENTRY** block (see the section above).

Entering Constraints

In water management problems, by default all variables have a lower boundary (lower limit) of change of the variable, which equals to zero.

Program system has an opportunity to change upper and lower boundaries of the range of the variables assigned by default or fix a value of a variable (for all or specific time steps).

You can enter constraints in different measurement units. In order to assign measurement units for entered constraints you have buttons with letter **Q** (or **W**). As in the **DATA ENTRY** module, the **Q** letter means that data for water consumption are in cu m/s. The **W** letter means that data for water flow are in million cu m for a time step.

In order to enter new values of constraints you need to:

To choose **Q** on the measurement units button;
To enter a new value for boundaries of the variable range in the working window for constraints entry;

Move the cursor from the constraints entry window (you will see that a value in the variable entry window changes, since water consumption in cu m/s to water flow in million cu m.

To enter new constraints for variables in water flow, you need to:
To choose **W** on the measurement units button;
To enter a new value for boundaries of the variable range in the working window for constraints entry.

If you want to enter the same constraints for some node for all time steps, then you need to enter constraints for any time step and after that press the **Enter in All Time Steps** button. The condition of buttons and information about constraints will automatically extended to all time steps. If you want to delete constraints for some node for all time steps, then you need to press the **Delete from All Time Steps** button. The condition of all buttons **ON-OFF** will change to **OFF** for all time steps.

If you want to delete a constraint entered for a current variable from some time step, you should press the left mouse button pointing on the **ON** window to change the label to **OFF**. Therewith, in the working window for constraints entry you will see the values of constraints entered by you, but ignored by the program. In this case, the program system adopts upper and lower boundaries by default equal to 1000 and zero.

NAVIGATOR Mode

The program incorporates a special algorithm for convenient viewing of the calculation scheme with constraints entered. To activate it, press the **NAVIGATOR** button ("**VIEW**" button on Figure 21) using the left mouse button.

In the **NAVIGATOR** mode the calculation scheme has the following designations:

- If neither constraint is adopted in the model for the **Lake** node, then this node is drawn as a blue circle.
- If at least one constraint is adopted for the Lake node (i.e. the entered constraint is active, that is the **ON/OFF** button is **ON**), then this node is drawn as a bright blue circle white banded.
- If neither constraint for arcs is adopted in the model, then crosses on the lines that link nodes are orange.
- If at least one constraint for an arc is adopted in the model (i.e. the entered constraint is active, that is the **ON/OFF** button is **ON**), then a cross on the line that links nodes is white.

If you move the mouse pointer to a cross on an arc or to a Lake node, then in the upper left corner you will see external node names in this arc (flow goes from the first node to the second). If at least one constraint entered for the node under consideration is active, then in the upper right corner in the area to display entered constraints bright yellow strips. Strips in the areas to display entered constraints are located sequentially in accordance with an ordinal number of a time step. The ordinal number of time steps increases from left to right. If an entered constraint is not active for some time step, then the respective area to display entered time steps remains black. Visual assessment of this window allows the user to see the quantity and type of constraints adopted in calculation at once.

The **NAVIGATOR** mode allows the user to go to a needed **Lake** node or a calculated arc quickly to enter constraints for it. For this purpose the user needs to:

- To start the **NAVIGATOR** mode,
- To point at a needed node or cross on a needed calculated arc,
- To press the right mouse button.

You go back to the **CONSTRAINTS ENTRY** Module in the place you need.

As you enter the **NAVIGATOR** mode, a yellow square marks your initial position in the constraints database as applied to the calculation scheme of links of a river system.

Quitting the Module

The user can quit the **CONSTRAINTS ENTRY** module with saving changes made to constraints and without saving them. To quit the module and save the changes made you need to click the left mouse button on the **Exit & Save** button. As a result, corresponding changes will be entered in the **Constraints File 1** file. The **Constraints File 2-cnstr** file will be formed anew. To quit the module without saving changes you can click the left mouse button on the **Exit** button or press **Esc** on the keyboard.

The Structure of the Output File *Constraints File 2*

Values of constraints are recorded in the **Constraints File 2** in accordance with the relevant GAMS rules.

If all constraints entered are not active, then this file has only one line looking as follows *_____. The * symbol has the first position. If there are active entered constraints, then the program system first records all constraints for **Lake** nodes, then goes the line looking *_____, and after that the program system records all constraints for calculated arcs. Constraints for each node or arc and for each time step are recorded in a separate line.

Below is shown how the above-presented part of **Constraints File 1-limitall** is written in **Constraints File 2-cnstr.**

```
  vol.lo('V1','m1')   =     150.00;
  vol.up('V1','m1')   =    1900.00;
  vol.lo('V1','m2')   =     150.00;
  vol.up('V1','m2')   =    1900.00;
  vol.lo('V1','m3')   =     150.00;
  vol.up('V1','m3')   =    1900.00;
  vol.lo('V1','m4')   =     150.00;
  vol.up('V1','m4')   =    1900.00;
  vol.lo('V1','m5')   =     150.00;
  vol.up('V1','m5')   =    1900.00;
  vol.lo('V1','m6')   =     150.00;
  vol.up('V1','m6')   =    1900.00;
*
*_____
flow.fx('C4_V2','m1') =   577.570;
flow.fx('C4_V2','m2') =   442.200;
flow.fx('C4_V2','m3') =   380.930;
flow.fx('C4_V2','m4') =   771.840;
flow.fx('C4_V2','m5') =   889.760;
flow.fx('C4_V2','m6') =   631.960;
```

Getting Help

To get information about a module you need to click the left mouse button on the **HELP** button. A reference text will appear on the screen. To quit the **Help** mode you need to press **Esc** on the keyboard.

Constraints File 2 Output File

The **Constraints File 2-cnstr** output file is a part of the **GAMS** model. The user can edit it using an internal embedded editor, which he can activate through the main control menu. The only condition of this file's correctness is that this file exists and is not empty. Therefore, this file always contains a comment beginning with the asterisk. However, this is a requirement of the GAMS compiler, which the user should know.

12. Setting Priorities Module

Description of the Module

The **SETTING PRIORITIES-FFFF.EXE** module serves to form the **Priorities File-digits.in** file with coefficients used in the objective function to set priorities in allocating water among entities in the calculation scheme. The **SETTING PRIORITIES** module is operable only if a corresponding version of the graphics file in PASCAL, **egavga.bgi**. is available. You run the **SETTING PRIORITIES** module from the interface by pressing the corresponding button using the left mouse button. At that, the picture of Figure 23 should appear on the screen. You operate the **SETTING PRIORITIES** module in accordance with comments on the screen.

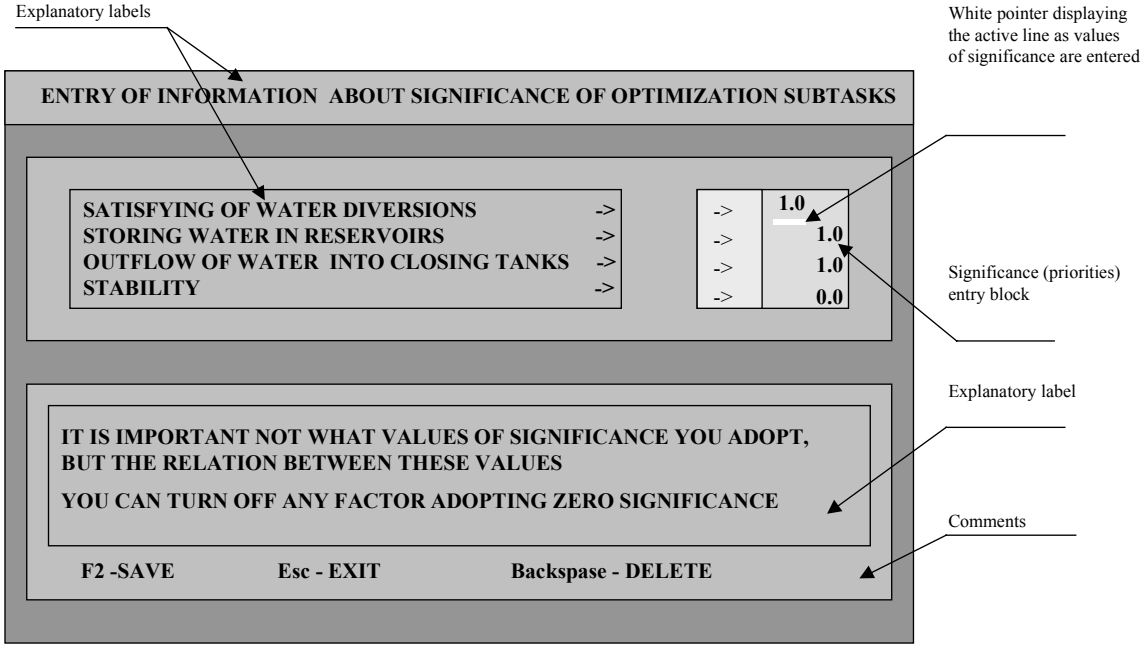


Figure 23. The Image of the Working Window on the Screen As You Start the SETTING PRIORITIES Module.

The Structure of the Output File *Priorities File*

The **Priorities File** output file has a very simple structure. It consists of two lines, where the first line is a comment text about coefficients. This line necessarily begins with the * symbol.

The second line contains numerical values of entered coefficients located in accordance with names of coefficients in the first line. The second line also necessarily begins with the * symbol. Let us remind you that the GAMS compiler recognizes the line that begins with * as a comment, and therefore an opportunity to connect the **Priorities** output file to the model as an **INCLUDE** file arises.

From the third to the seventh lines are entered coefficients, which are recorded in the form of an equation ending in the ; symbol.

Below is shown **Priorities File** for our test problem.

```
P1=    100.00000    ;
P2=      1.00000    ;
P3=      1.00000    ;
P4=      0.00100    ;
P5=     10.00000    ;
```

13. Building the GAMS Model Module

Composition and Structure of the GAMS Model

Constructively, this model incorporates four interacting and isolated files that contain the text of the model of optimal water management. The model is written under the rules of the GAMS compiler.

The first part defines all parameters and initial data that determine a specific calculation problem.

The second part is actually the universal part of the model including equations and the procedure of delivering a calculated result to an individual file.

The third part is a file with information about priorities in the optimization objective.
The fourth part is a file of the constraints imposed on variables, which are defined in the solution process.

To solve a raised problem each of these parts should be available. You may run the built model in the directory where it was built. It loses its operability if it is written in a different directory because paths to the model's components cease conforming to a new location of the model. To restore the model's operability it is necessary to find path specifications in the model and change them according to the new location (or completely delete them). Here we mean the following paths:

- The path to the **equations** block specified in the **riv_new.gms** file;
- The path to the **digits.in** file, with the specification of priorities in the objective function.
- The path to the **cnstr** file that contains constraints;
- The path to the **river.new** file where results are located.

Operation of the Module Providing the Automated GAMS Model Building

To run this module the user should point at the **BUILDING GAMS MODEL** button and press the left mouse button. The **build.exe** module will become active.

This program has many various testings to identify the availability and methods of locating reference data in the GAMS model's text. For instance, if the calculation scheme does not incorporate such nodes as reservoirs, naturally it does not include all parameters bound to given calculated entities. Moreover, the description of the model text should not include both equations and the node type **Lakes.**

There are many such testings as the model is constructed, and the user can see results of these testings on the screen and trace the entire sequence of the model creation. Most testings are concerned with the specifics of the **GAMS** language, and therefore situations, where the user has to correct an error while he operates the mode of editing the constructed model, may occur.

The only question that the user will be asked is whether it is necessary to update the equation block as the model is created. If the user does not want to update the equation block, then a previously created block remains in the model. Such independent creation of blocks allows the user to conduct research of the created model changing and modernizing the equation block for his specific tasks. At the beginning of this tutorial, we provide the text of the **GAMS** model automatically created by the model construction module.

Optimization Calculations on the GAMS Model

To start the GAMS compiler you should point at the **GAMS CALCULATION** button and press the left mouse button. The **GAMS riv_new.gms** command line is formed and executed. Messages of the **GAMS** compiler will appear on the screen and the user should trace the appearance of the label **OPTIMAL SOLUTION FOUND**. Only after this label has appeared, the user can undertake actions associated with translation and formation of output calculation forms. Otherwise, these forms may either not work or contain wrong data. The appearance of the **OPTIMAL SOLUTION FOUND** label means that water balance for the entire river system created is kept precisely and an optimal solution is found in full accordance with the set system of priorities.

Let us stop on the alarm messages of the **GAMS** compiler that arise more often.

"INFEASIBLE SOLUTION" – more often this message appears in problem, where mutually exclusive constraints on decision variables are recorded. For example, if the user fixes a release from a reservoir equal to 100 cu m/s and specifies that flow to a downstream node should not be more than 99 cu m/s, he will sure receive this message. The only means against this message is to turn off all constraints on water flows simultaneously and then sequentially turn them on recalculating the model.

"BADLY BOUND" – this message appears resulted from uncoordinated entry of an initial water volume in the reservoir and possible upper and lower bounds of its change. For example, initial storage of a reservoir is 100 million cu m, the lower storage bound is 200 million cu m, and the upper storage bound is 1000 million cu m. This situation will assure the appearance of the message.

"TOO MANY ITERATIONS" – this message can easily do with replicating of the line that contains the work **SOLVE** several times. Generally, the last recipe works against some other alarm messages of the **GAMS** compiler.

Access to the Main Output File of the GAMS Compiler

In the process of the **GAMS** compiler's operating, the **riv_new.lst** file forms automatically. This file has all information about calculation process, including that about errors of testing the program body. All compilation errors are marked with four asterisks going in sequence.

The user can see the contents of this file if he presses the left mouse button at the moment when the mouse pointer is at the **riv_new.lst** main output file of the menu.

Having detected and identified an error, the user can correct directly in the model text or through making corrections using previous modules described above.

The user can access the model's file by pointing at the **GAMS MODEL TEXT** button and pressing the left mouse button.

Access to the Main File of the GAMS Model Text

The **GAMS MODEL TEXT** button consists of two parts. Positioning the mouse pointer on the left part of the button, the user gains access to reference data characterizing a specific problem. Pointing at the right part of the button, the user gains access to the editing of the calculation part of the model and the information on the procedure of delivering results to the **river.new** file.

Access to the File of Calculation Results of the GAMS Model

Pointing at the **MODEL CALCULATION RESULTS** button and pressing the left mouse button, the user gains access to the file of calculation results of the model to assess the solution obtained.

Despite our attempt to present results in tabular forms convenient for perception, you can learn how to assess results quickly only after some practice. This primarily relates to the fact that information about obtained solutions is displayed only for internal names used in the **GAMS** model. In addition, our desire to create an instrument to solve similar problems in random river systems impeded the formation of convenient output tables at this stage of calculation. Thus, we could not attach our actions to stable forms of output. We put the solution of this problem on a group of the above-described modules that arrange output forms.

Nevertheless, we want to explain and show the essence of some output blocks, which surely are in the result file.

The output file begins with the following data block, in which comments in italics explain the meaning of displayed digits.

30 <number lines
number of arcs forming a network equals 30

7 <number users
7 users in the river system

1 <number mouth
River system has one river mouth

2 <number lakes
River system has two reservoirs

0.9381 <2 task user
average water availability for users 93%

0.0162 <3 task mouth
water flow in the river mouth is 1.6% of the possible

0.7037 <4 task filling
water bodies are 70 % filled at the end of a time step

0.0176 <5 task stabl.
higher sustainability of solution (the lesser the less sustainable)

For each time step and connection forming arcs (lines), data about water flow in million cu m are displayed.

time-step = M1 connections LINES

24.5107	V1	==>	U1	U1_V1
164.0000	I1	==>	V1	V1_I1
96.1868	V1	==>	C1	C1_V1
96.1868	C1	==>	K1	K1_C1
96.1868	K1	==>	U2	U2_K1
0.0000	K1	==>	C2	C2_K1
0.0000	C2	==>	K2	K2_C2

....
....
....

For each user (N1= U1, etc) and each time step (time = M1) the following data are displayed: All-IN-flow –total water withdrawal; percentage - fulfillment of requirements in percent; requirements; transit; return.

USERS

All-IN-flow	percentage	requirement	Transit	Return	USER	time = M1
24.5107	0.9459	25.9030	0.0000	0.00	N1= U1	!
96.1868	0.8636	111.3700	0.0000	0.00	N1= U2	!

For each river mouth (N1= R1, etc) and each time step (time = M1) the following data are displayed: All-IN-flow –total water withdrawal; percentage - fulfillment of requirements in percent; requirements; transit; return.

MOUTH

All_IN_flow	percentage	requirement	Transit	USER	time = M1
251.0288	0.0126	20000.0000	0.0000	1 N1= R1	!

Tables of calculated water balance are drawn up for all reservoirs. For example here 663.00 is the initial volume of the reservoir VI. **In-flow** is inflow, **volume** is volume at the end of a time step, **out-flow** is release to the downstream, **difference** is regulation, in.evap is evaporation, **av-surf** is average surface for a time step, **evap**. is total evaporation losses, **LEVELV** is the upstream water level, **LEVELN** is the downstream water level, **EL.mln.kVtH** is power generation in million kWh, **time** is the identifier of the time step.

WATER BALANCE for all Reservoirs

663.00 => start volume WATER BALANCE FOR V1 !

in-flow	volume	out-flow	difference	in.evap.	av-surf.	evap.	LEVELV	LEVELN	EL.mln.kVtH	time !
164.00	706.85	120.70	-43.30	0.0400	13.66	0.55	808.36	765.49	775.73	time_step => M1
457.00	950.22	214.26	-242.74	0.0400	15.57	0.62	808.92	765.49	1068.05	time_step => M2
648.00	1327.90	271.09	-376.91	0.0400	19.40	0.78	809.95	765.49	1213.14	time_step => M3
452.00	1325.43	455.34	3.34	0.0400	21.56	0.86	810.48	765.49	2360.96	time_step => M4
196.00	1113.10	409.14	213.14	0.0400	20.34	0.81	810.19	765.49	1649.74	time_step => M5
69.0	995.01	187.83	118.83	0.0400	18.39	0.74	809.69	765.49	981.22	time_step => M6

Further goes the block that contains comments, equations describing morphology, and coefficients in these equations intrinsic to each calculated reservoir.

TABLE OF COEFFICIENTS FOR RESERVOIRES
S = a * W^b H = c * W^d +h
S - area; W - volume ; H - Level; h - level-of 0;
n - efficiency ; t - hours-in interval = 10.00

a =	0.19	b =	0.69	n =	1.00	RES-RE V1
a =	1.41	b =	0.54	n =	1.00	RES-RE V2

c =	17.31	d =	0.31	h =	799.00	RES-RE V1
c =	1.56	d =	0.46	h =	757.92	RES-RE V2

If the user wishes next modules to be able to read and locate results correctly in the tables more understandable and acceptable for analysis, then he should not change the structure of the **river.new** output file.

Translation of Results into Other Languages

The **GAMS** compiler works only with letters of the Latin alphabet, and that causes problems of readability of output forms. In addition, internal names given by the program system to nodes of the calculated scheme impede the reading and interpretation of results. To solve this problem, we created the module of the translator of results.

The translator of results is the **TRANSL.EXE** module that processes the main files of results **river.new** and **rivers.new**. Unlike other modules, this module is operable only if it is started through creating a command line containing three sequential parameters. For example:

TRANSL.EXE c:/model/ c:/model/test 1 for **river.new**

TRANSL.EXE c:/model/ c:/model/test 2 for **rivers.new**

The first parameter shows the location of the internal complex dictionary, the second parameter shows locations of files for translation. English terms are replaced with their analogs from the internal dictionary.

System for Arrangement of Output Forms

This system incorporates three modules **REACHES.EXE**, **FORM.EXE,** and **MAIN.EXE.** All these modules are activated through buttons of the main control menu, which are located a bit above the viewing window of the working directory's location (lower part of the menu).

Calculated Reaches Forming Module

The module of forming calculated reaches **REACHES.EXE** requires the **egavga.bgi** graphics driver. As input data, this module uses files **klav.out** and **out.out**. Based on data from these files, a created river system (Figure 24) s displayed on the screen. Buttons on the right display four features of the lines along which water can flow.

The beginning of the reach **UP**; The end of the reach **DOWN**;
Inflow to the reach **IN**; Outflow from the reach **OUT**

The user activates a feature by pointing at the corresponding button and pressing the left mouse button. The feature activated is displayed white.

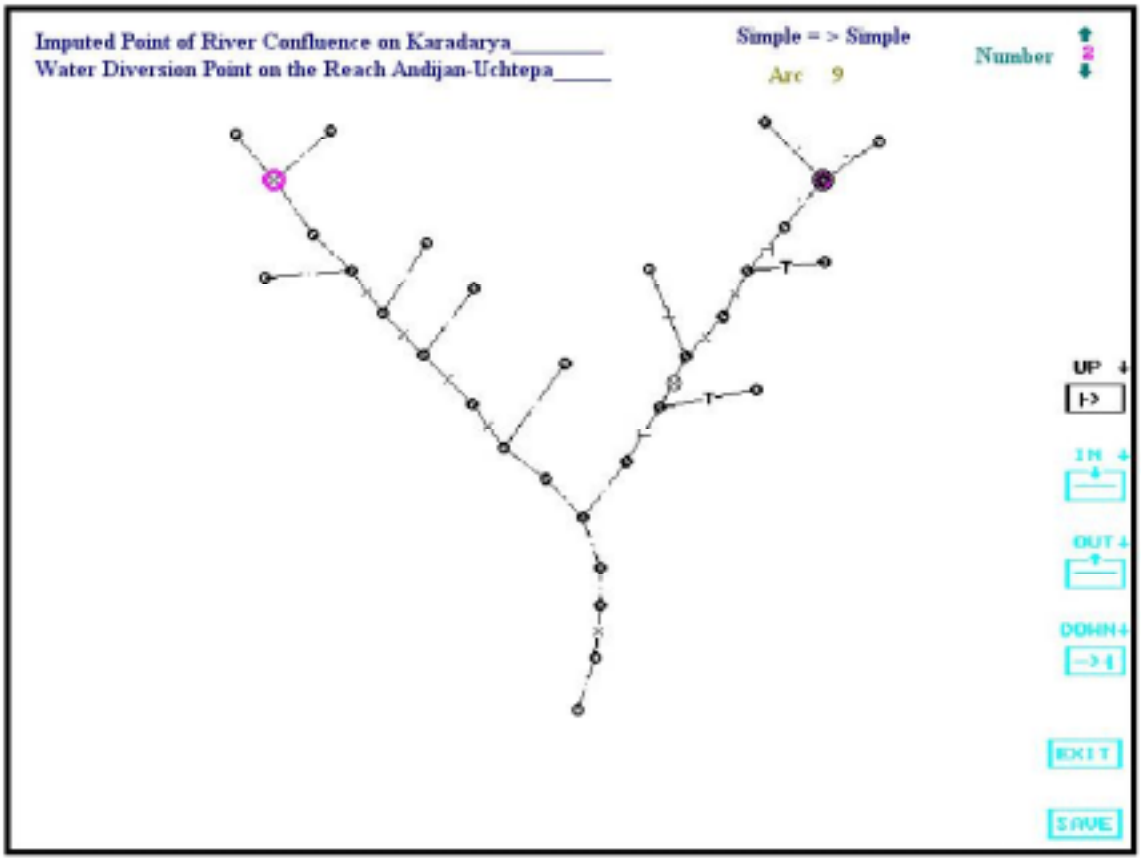


Figure 24. Working Window on the Screen as You Press the View Button in the CONSTRAINTS ENTRY Module

The user arranges the composition of an output form under the following rules:
If the **UP** button is active, the data from the arc specified by the user will be in the line of the table named **Inflow to the Calculated Reach**. If the **DOWN** button is active, the data from the arc specified by the user will be in the line of the table named **Outflow from the Calculated Reach**. If the **IN** button is active, the data from the arc specified by the user will be in the line of the table named **Inflow to the Calculated Reach**. If the **OUT** button is active, the data from the arc specified by the user will be in the line of the table named **Water Withdrawal and Losses on the Calculated Reach**.

All information will be in cu m/s, and resultant values will be in million cu m. To make analysis simpler, all data in both inflow and outflow parts will be summed. However, if the user, while arranging an output form of the table, activates the point that simulates a reservoir, then the form of the table will change. It will be filled with data about water volumes in the reservoir at the beginning and at the end of a time step.

A button located in the right upper part of the screen sets the procedure of displaying forms on calculated reaches and reservoirs. The user has to the formation of 20 sequential reporting forms. The output file of this module is the **lines** file. A fragment of this file is given below.

```
0000000  1  2  1  1 463  96 495  74 479  85 ¦Andijan Reservoir ¦ Losses from Andijan
                                                                    Reservoir________¦
0000000     1    0    2    1  430    63  463    96  447    80 ¦Inflow to Andijan Reservoir_________¦ Andijan
                                                                    Reservoir_____________________¦
0000000  1  3  3  1 463  96 441 123 452 110 ¦ Andijan Reservoir _____________________¦ Point of Control of
                                                                    Releases from Andijan Reservoir
0000000  2  0  4  0 441 123 420 148 431 136 ¦Point of Control of Releases from Andijan Reservoir_______¦Point
                                                                          of Water Diversion of
                                                                       Feeding Canal__________¦
0000000  2  2  5  0 420 148 464 143 442 146 ¦Point of Water Diversion of Feeding Canal__________¦Feeding
                                                                    Canal_____________________________¦
0000000  0  0  6  0 420 148 406 174 413 161 ¦Point of Water Diversion of Feeding Canal__________¦Karabagish
                                                          Measurement Station_____________________________¦
0000000  0  0  7  0 406 174 385 197 396 186 ¦Karabagish Measurement Station_____________________¦ Imputed Point
                                                                    of River Confluence on Kara Darya____¦
0000000  2  1  8  2 364 147 385 197 375 172 ¦Tributary Inflow on Kara Darya_____________________¦Imputed Point of
River Confluence on Kara Darya____¦
```

Let us note that the created file precisely corresponds to the order of data layout in the main **river.new** output file. The first column of digits serves to place data from the **river.new** file as a comment for possible manual adjustment of the **lines** file. The second column of digits characterizes the procedure of displaying a reporting form. The third column of digits characterizes the condition of buttons **UP, IN, OUT,** and **DOWN** at the moment when a reach is set by codes **0, 1, 2,** and **3** respectively.

The third column of digits is the ordinal number of the location of this information in the **river.new** output file. The fourth column of digits can contain either 0 or the ordinal number of a reservoir in the **out.out** file's list, thereby characterizing the table type (river reach or reservoir). The next four columns of digits are coordinates of arcs and serve as reference information. Further, there are external names of nodes characterizing a given calculated arc. Here the user should pay attention on the following feature. External names of nodes characterizing an arc do not always fit to form reporting forms, making them ill-looking and unintelligible. If the user changes the name of the first node (to the one that is more consistent with the output form), then the new name will get in the output reporting form. The arcs that characterize inflow to the reach and outflow from the reach are the exception. In the first case, the external node name goes to the table heading, and in the second case, the name is completely ignored.

If the user changes the sequence of lines in this file, then the sequence of lines in output forms also changes. Sorting and moving lines is doable only within each reporting form. Below is given an fragment of the reporting form arranged in one of calculated variants of the forecast for 2000 in BVO Syrdarya.

```
1          Toktogul Reservoir_________________
-------------------------------------------------------------------------------------------------------------------------
balance sheet items                                  1        2        3        4        5        6  SUM
-------------------------------------------------------------------------------------------------------------------------
VOLUME at the beginning of the period      : million cu m:  10903.0 11102.4  12260.2  14415.1 15570.1  16111.5
Inflow to reservoir_____________     : cu m/s  :     308.0    663.0   1043.0    911.0    628.0    347.0   10299.2
Precipitation _____________________  : cu m/s  :       0.0      0.0      0.0      0.0      0.0      0.0       0.0
TOTAL    INFLOW                  : cu m/s  :     308.0    663.0   1043.0    911.0    628.0    347.0  10299.2
Evaporation_____________________     : cu m/s  :       1.0      1.0      1.0      4.0      6.0      7.0      52.8
Releases to the downstream._____________     : m3/c  :     230.0    230.0    210.0    476.0    420.0    190.0    4649.4
```

TOTAL OUTFLOW : cu m/s :	231.0	231.0	211.0	480.0	426.0	197.0	4702.2
VOLUME at the end of period : cu m/s :	11102.4	12260.2	14415.1	15570.1	16111.5	16500.0	
STORAGE/DEPLETION(+/-): million cu m :	-1994	-1157.8	-2154.9	-1155.1	-541.4	-388.5	-5597.0
ELEVATION AT THE END OF THE PERIOD: m :							

2 REACH: TOKTOGUL – Uch Kurgan Hydroengineering Complex

	1	2	3	4	5	6	SUM
FLOW IN THE UPSTREAM SECTION : cu m/s :	230.00	230.00	210.00	476.00	420.00	190.00	4649.4
Tributary inflow : cu m/s :	84.00	121.00	107.00	71.00	45.00	33.00	1215.3
Channel regulation : cu m/s :	0	0	0	0	0	0	0
Total outflow : cu m/s :	187.68	209.00	256.66	309.86	251.35	117.58	3519.6
OUTFLOW TO THE DOWNSTREAM SECTION : cu m/s :	126.32	142.00	60.34	237.15	213.65	105.42	2345.2

Modules Arranging a Short Reporting Form on the Operation of Reservoirs and End Sections of the River System

These modules **FORM.EXE** and **MAIN.EXE** are operable if the **river.new** calculation result is available. They arrange output forms approved in the ICWC that contain information about operation modes of reservoirs and water inflow in river mouths (Aral, Arnasai). To obtain these reporting forms, the user should simply run these modules from the main menu and gain results. The user can read a result still being in the main menu. A fragment of this reporting form automatically arranged is given below.

Toktogul Reservoir_____

TIME STEPS : :	1	2	3	4	5	6	SUM
VOLUME: BEGINNING OF THE PERIOD :million cu m:	10903.00	11102.43	12260.19	14415.07	15570.14	16111.50	10903.00
INFLOW TO RESERVOIR : cu m/s :	308.00	663.00	1043.00	911.00	628.00	347.00	
INFLOW TO RESERVOIR :million cu m:	797.72	1776.84	2701.37	2441.48	1683.04	898.73	10299.18
RELEASE FROM RESERVOIR : cu m/s :	231.00	231.00	211.00	480.00	426.00	197.00	
RELEASE FROM RESERVOIR :million cu m:	598.29	619.08	546.49	1286.41	1141.68	510.23	4702.18
VOLUME: END OF THE PERIOD :million cu m:	11102.43	12260.19	14415.07	15570.14	16111.50	16500.00	16500.00

14. Saving and Restoring Previously Entered Data

The aim of this system is to restore previously entered data in nodes of the system that remained unchanged. A node, and so the data adopted for it as reference data are considered unchanged if

1) The node did not change its type;
2) The node's location did not change.

This system includes three modules:

REPLACE_1.EXE, REPLACE_2.EXE, and REPLACE_3.EXE.

Individual modules of this system operate automatically; they become active without the user's participation. Their activation is based on the availability and condition of special files, which store data about the history of effects on the calculation scheme located in the working directory.

As these modules become active, the algorithm treats the causal effect recorded in a group of files **copy_1.new, copy_1.bak, copy_2.new, copy_2.bak, copy_3.new, copy_3.bak.**

At the moment of the first creation of a calculation scheme, these files do not exist. The logical block of each of three executable modules having detected the absence of these files, deactivates the **SYSTEM FOR SAVING AND RESTORING PREVIOUSLY ENTERED DATA.**

However, if the already created scheme is changed by adding or changing the number or types of nodes, all previously created files containing data are renamed in such a way that their ***.in** extension transforms into the ***.bak** extension. It takes place if the network creation module ends its operation with saving, but all data files have been already created for a previous scheme and are present in the working directory. At that three files **copy_1.bak, copy_2.bak, copy_3.bak** are created. In all the three files, the only word **Yes** is recorded.

For the first file, it means that the system made copies of the previously existed files:
supplies..in user.in mouth.in lagtime.in lakes.in losses.in under the names **supplies.bak, user.bak, mouth.bak, lagtime.bak, lakes.bak, losses.bak.**

For the second file, it means that the system made a copy of the previously existed file **lakes.vol** under the name **lakes.bak.**

For the third file, it means that the system made a copy of the previously existed file **limitall** under the name **limitall.bak**.

When the user activates the **DATA ENTRY** mode and then quits it saving the changes made (provided that the **copy_1.bak** file containing the **YES** word is in the working directory), then the **copy_1.new** file containing the **YES** word will be created (see Figure 25). This word **YES** means a new group of files **supplies.in, user.in, mouth.in, lagtime.in, lakes.in,** and **losses.in** has been created. Simultaneous existence of files **copy_1.new** and **copy_1.bak** and the word **YES** activates running the **REPLACE_1.EXE** module. This module will show itself in the following way. The user will be asked a question whether the system should try to carry digital data from files of same names and files previously existed to newly created files or not. The third variant of the answer supposes disconnecting this module **REPLACE_1.EXE** from the data restoration system.

The action of this module while restoring is in the following: if a newly created file and a prior existed file have nodes with same coordinates, then previously entered numerical data are placed on corresponding places in the newly created file in the block that describes this node of the network.

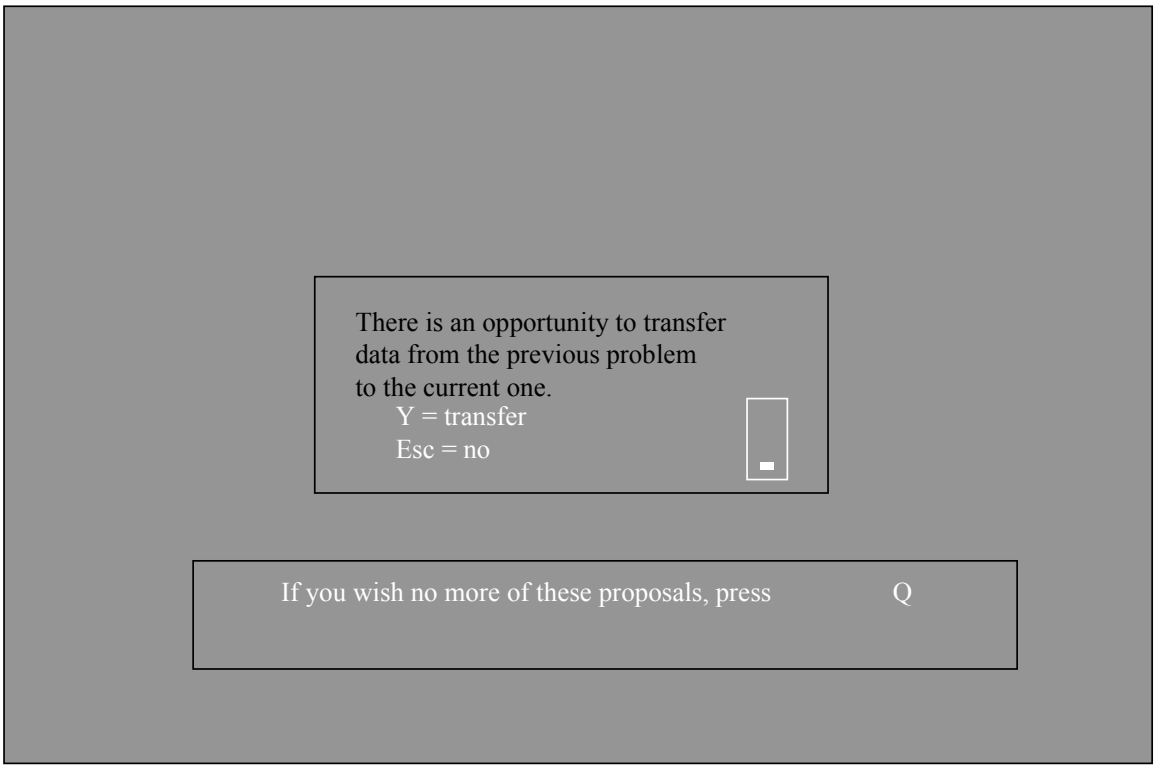


Figure 25. The Image on the Screen as You Quit the DATA ENTRY Module

Disconnecting this module from the data restoration system results from the replacement of the **YES** word in the **copy_1.bak** file with the **NO** word and a corresponding comment about the way of repeated connection of this module into the data restoration system.

When the user activates the **RESERVOIR PARAMETERS** mode and quits saving the changes made, then the **copy_2.new** file containing the word **YES** will be generated. This word **YES** means the system has created the **lakes.vol** file. Simultaneous existence of files **copy_2.new** and **copy_2.bak** and words **YES** activates running the **REPLACE_2.EXE** module. This module will show itself in asking the user whether or not the program system should attempt to transfer digital data from the prior file of the same name **lakes.bak** to the newly created file **lakes.vol**. The third answer option supposes disconnecting the **REPLACE_2.EXE** module from the data restoration system.

In restoring a module, this module operates as follows. If in the newly created **lakes.vol** and the prior existed **lakes.bak** files a node with the same coordinates is present, then the previously entered digital data on the reservoir morphology are put on corresponding places in the newly created file in the block describing a given node in the network.

Disconnecting this module from the data restoration system occurs resulted from the substitution of the word **NO** with a relevant comment for the word **YES** in the **copy_2.bak** file. The comment concerns the repeated connection of this module to the data restoration system.

When the user activates the **CREATING CONSTRAINTS FILE** module and quits it with saving, then the program system will create the **copy_3.new** file containing the word **YES**. This word **YES** means that the **limitall** file has been created. Simultaneous existence of files **copy_3.new** and **copy_3.bak** and words **YES** activates the execution of the **REPLACE_3.EXE**. This module will show itself in asking the user whether or not the program system should attempt to transfer digital data from the prior file of the same name **limitall.bak** to the newly created file **limitall**. The third answer option supposes disconnecting the **REPLACE_3.EXE** module from the data restoration system.

In restoring a module, this module operates as follows. If in the newly created **limitall** and the prior existed **limitall.bak** files a pair of nodes with the same coordinates is present, then the previously entered digital data on the reservoir morphology are put on corresponding places in the newly created file in the block describing this pair of nodes in the network.

Disconnecting this module from the data restoration system occurs resulted from the substitution of the word **NO** with a relevant comment for the word **YES** in the **copy_3.bak** file. The comment concerns the repeated connection of this module to the data restoration system.

The operability of the **SYSTEM FOR SAVING AND RESTORING PREVIOUSLY ENTERED DATA** will be restored automatically as you activate the network creation mode and then quit the system in the saving mode (in total) or by correcting files **copy_1.bak, copy_2.bak, copy_3.bak** using an external editor (by parts).

This system is not a necessary part of the program system , but, if you change and supplement it, this system considerably facilitates you study processes in a river system.

Note that the examination of identity of nodes is based on features of identity of types and coordinates of nodes in a prior existed system and a newly created one. Therefore, the user can easily withdraw any node from the restoration mode by using the **SETTING FLOW DIRECTIONS** module. The user slightly changes the node's coordinates on the scheme by dragging the node across the working field on the screen.

15. Optional Blocks of the Program System

The primary purpose of the program system is to solve problems of optimal water management in a river basin. The interface, however, includes a place designated to place an optional block, which enables the user to solve additional problems basing on the base water problem described above.

These optional capacities may be realized in two ways:

1. The user changes and corrects the block of model equations, which exists separately from the block made on specific data and a specific river system. For instance, the user may add equations providing certain relationships between groups of water diversions or may change the reservoir morphology function from power to any other, or …

2. The user may utilize optional blocks created by specialists who worked on this program system (salinity block and energy block).

These blocks have their own optional modules providing automated calculation process. You can connect this modules to a water problem through the following actions:

1. Add a group of executable block modules to the program system.
2. Change or use operation modules, in which these optional systems have been already incorporated.

We present the descriptions of optional modules to a water problem in next additional chapters.

16. Water Quality Block (Salinity)

Essentials

This block is a complex consisting of program modules **CREATING CONSTRAINTS DATABASE, CONSTRAINTS ENTRY,** and **BUILDING GAMS MODEL.**

This block is operable, if you have managed to do and conduct calculations for the base task *Optimal Water Management*. Otherwise, any mistakes in forming a river scheme may prevent you from obtaining a good and right solution in case you connect the water quality consideration block. Let us remind, this block is organically connected with the base model made in BVO Syrdarya. Let us note that the program system has the menu that provides solution of the water and energy problem, but we will present it below.

Creating Constraints Data Base Module

You activate the **Creating Constraints Database-r_limtls.exe** module by placing the mouse cursor on a corresponding button of the virtual menu and at the same time pressing the left mouse button. On the screen, the user is offered an option of decisions.

ATTENTION!

If you press **ENTER**, the program will write a new **limitls** file. All data, which were in the previous **limitls** file will be lost. Use this program only if you have changed a river network or a number of time steps. If you want to quit the module without any actions, press **ESC**

Make your choice:
ESC – to quit
ENTER – to execute

The structure of this file precisely coincide with the structure of the **limitall** file described above except that this file is not connected and not controlled by the protection and saving of previously entered data. Usually, constraints on the salinity factor are entered few, and creating systems to save these data overloads the program system.

If you enter all constraints in the **limitall** file in million cu m, then all constraints in the **limitals** file are in gram/liter. The structure and meaning of all digital columns are identical in both files, and, therefore, to save room we omit the redescription of the **limitals** file. We provided the **limitals** file with comments about all data containing in this file.

Through a special button of the main menu, the user has access to the **limitals** file for possible editing and viewing constraints in the **WATER QUALITY** subblock (salinity factor).

Constraints Entry Module

The **CONSTRAINTS ENTRY-r_ch_lis.exe** module starts operating, if you place the cursor on the **Constraints Entry** button of the virtual menu and press the left mouse button. The user sees a virtual menu, which enables him to enter constraints:

1) Fixed value
2) Minimum limit
3) Maximum limit

This module is provided with almost all capabilities incorporated and described in the module of entering water constraints that are described in the base water management complex described above. The appearance of the menu, dialog boxes, and control buttons – everything almost coincides in appearance and location with the constraints entry module in the base complex *Optimal Water Management* described above. Navigator operates likewise and under the same principle. However, we changed the palette of schemes and drawings for the user not to confuse these complexes while operating. Also, we changed some lines commenting the data.

The difference is that there are no some capabilities unnecessary for the salinity factor. No conversion of entered constraints from one dimension to another is available, i.e. buttons to convert water consumption to flow are absent (so do not operate). It is clear that you do not need them because all constraints are in grams/liter for any reservoir or arc.

A data output file is the **cnstr_s** file. The structure of this file identically coincides with the structure of the **cnstr** file that defines water constraints (described above).

The user, through a special button of the main menu, has access to the **cnstr_s** file to edit and view constraints in the **WATER QUALITY** subblock (salinity factor). Having certain practice, the user may enter constraints directly in this file even without using the modules described above.

Building GAMS Model Module (with Regard to Water Quality)

You activate the **BUILDING GAMS MODEL-builds.exe** module if you place the cursor on the **BUILDING GAMS MODEL** button of the virtual menu and press the left mouse button.

The user can in no way influence the process of building a water quality consideration model. The user sees on the screen running comments concerning model building. The only thing the user can do is to abort model building by pressing the keys **CTRL + BREAK**. The result of this module's operation, if not aborted, is the **riv_news.gms** file.

Through a special button of the main menu, the user has access to the **riv_news.gms** file for editing and viewing. Along with its own file of constraints **cnstr_s**, the model incorporates the files **cnstr** and **digits.in** created in forming an optimal water management model. That is why the condition of preliminary forming and calculating the model *Optimal Water Management in a River Basin* is of importance.

This model is bigger and more complicated than the *Optimal Water Management in a River Basin* model. However, the principles of building its equations are similar and based on the following statement.

Salt is the same conservative component as water with the difference that the evaporation factor cannot alter the amount of the salt dissolved and the salt transported.

Studying the *Optimal Water Management* model and the *Optimal Water Management with Regard to Water Salinity* model at the same time, the user may notice that all equations of the first model are transferred practically unchanged to the second model. These equations are supplemented with duplicates describing salt movement in water flows, salt accumulation in reservoirs, salt inflow to water users, and salt outflow to mouths of irrigation and river systems. The duplicates of equations describing the salinity factor lack evaporation components.

Supplemented are only two equations, which enabled us to go from calculation variables in thousand tons of salt to salinity in gram/liter. These equations allow us to establish a connection with the above-described file of constraints on water salinity. The first of the below given equations enables us to calculate water salinity in any reservoir of our calculated scheme through dividing the amount of salt by the amount of water in any time step.

salin_v(n1,m).. vol_s(n1,m) =e= s_vol(N1,m) / (vol(N1,m) + 0.001);

where:

M	=	set of time steps
N1	=	subset of reservoirs in set **N1** of all nodes of a calculation scheme
salin_v(n1,m)..	=	equation name
vol_s(n1,m)	=	water salinity in a reservoir (g/l)
s_vol(N1,m)	=	amount of salt in a reservoir (thousand tons)
vol(N1,m)	=	amount of water in a reservoir (million cu m)

salin_f(duga,m).. flow_s(duga,m) =e= s_flow(duga,m) / (flow(duga,m) + 0.001);

where:

duga	=	subset of all calculated arcs of the calculated scheme
salin_f(duga,m)	=	equation name
flow_s(duga,m)	=	salinity of the water moving along an arc (g/l)
s_flow(duga,m)	=	amount of salt moving along an arc (thousand t)
flow(duga,m)	=	amount of water moving along an arc (million cu m)

Also, you can easily notice a difference in the equations that describe an offtake of water flow from the main channel (be it a water diversion or a lateral canal).

If for the water factor, we just subtract diverted water from the total water inflow to the point of flow ramification, then for the salinity factor we calculate diverted salt as a share of the total amount of salt that came into the point of ramification. This share of salt equals a share of water ramification from the total water flow.

Conducting Calculations Module (with Regard to Water Quality)

You activate the **CONDUCTING CALCULATIONS** mode by placing the cursor on the **GAMS CALCULATION** button and pressing the left mouse button. Thereby, you form a command line connecting the **GAMS** compiler to a create complex model *Optimal Water Management with Regard to Water Quality*. While running this mode, the main listing of results **riv_news.lst** and the main output file **rivers.new** located in a working directory are formed. The first of the above mentioned files is necessary to assess a solution found and to control conducted calculations, to seek and identify errors. The second file in compressed tabular forms presents an identified solution (in English and using internal names of the model). Structurally, it slightly differs from the **river.new** file described in the basic part of the program system. This file is added a series of columns informing about the data on water quantity along with the data on salt flow and water salinity. Comments included in this file univocally define each of derived calculation figures.

Apart from the tables containing inwards and outwards of water balance in calculated water bodies, we added the tables containing salt balance for each calculated water body.

The user gains access to these files, perhaps for correction, through the buttons of the water salinity consideration block.

Translating Calculations Results Module (with Regard to Water Quality)

This module is described in the basic part of the program system, and in this case, it is used in this subblock a second time. Reorientation of this module to new files is carried out in a command line, as you run this module (see the basic part of the program system). The file of results to which the user gains access through a button in the menu has the name **rivers.trn.**

Results of Testing the Water Quality Block (salinity)

Initiators and actual developers of this block were specialists of the Coordination Group and the Technical Group from the Republic of Kazakhstan. Therefore, most tests of this program system were conducted within a task where the downstream of Chardara Reservoir was the main control section.

In the process of preparing test calculations by Kazakh specialists, they collected and summarized the material concerning salt flow of the Syrdarya River (over 10 sections) under a monthly pattern for almost twenty years.

The specialists conducted five test calculations.

1) calculation without salinity constraints produced results identical to the results obtained in the base model (without regard to water salinity)
2) calculations with salinity constraints were entered for the downstream of Chardara Reservoir
   a. Water flowing to the downstream should have salinity of less than 1.5 g/l. The result was achieved by changing operation modes of Kairakkum and Charvak reservoirs without infringed interests of irrigation users in the Fergana Valley.
   b. Water flowing to the downstream should have salinity of less than 1.2 g/l. The result was achieved by changing operation modes of Kairakkum and Charvak reservoirs with infringed interests of irrigation users in the Fergana Valley by 10 % on average.
   c. Water flowing to the downstream should have salinity of less than 1.0 g/l. The result was achieved by changing operation modes of Kairakkum and Charvak reservoirs with infringed interests of irrigation users in the Fergana Valley by 25 % on average.

You may familiarize yourself with calculation results more particularly in the seminar materials of the Technical Group from the Republic of Kazakhstan.

17. Power Generation, Distribution and Consumption Block

Essentials

This block is a complex including program modules **DETERMINING POWER PRODUCTION COSTS, ENERGY CONSTRAINTS ENTRY,** and **BUILDING GAMS MODEL.** This block is a supplementary block to the base model of BVO Syrdarya *Optimal Water Management*. You may obtain a part of the files needed for operability of the latter module of this subblock **BUILDING GAMS MODEL** only resulted from the use of the base model. All buttons to work with this block are located in the right lower corner of the main control menu. On Figure 26, you can see a version of the main menu, when the energy block is connected. The difference from the water quality block is small and is in a changed comment to this subblock and a label on one of buttons.

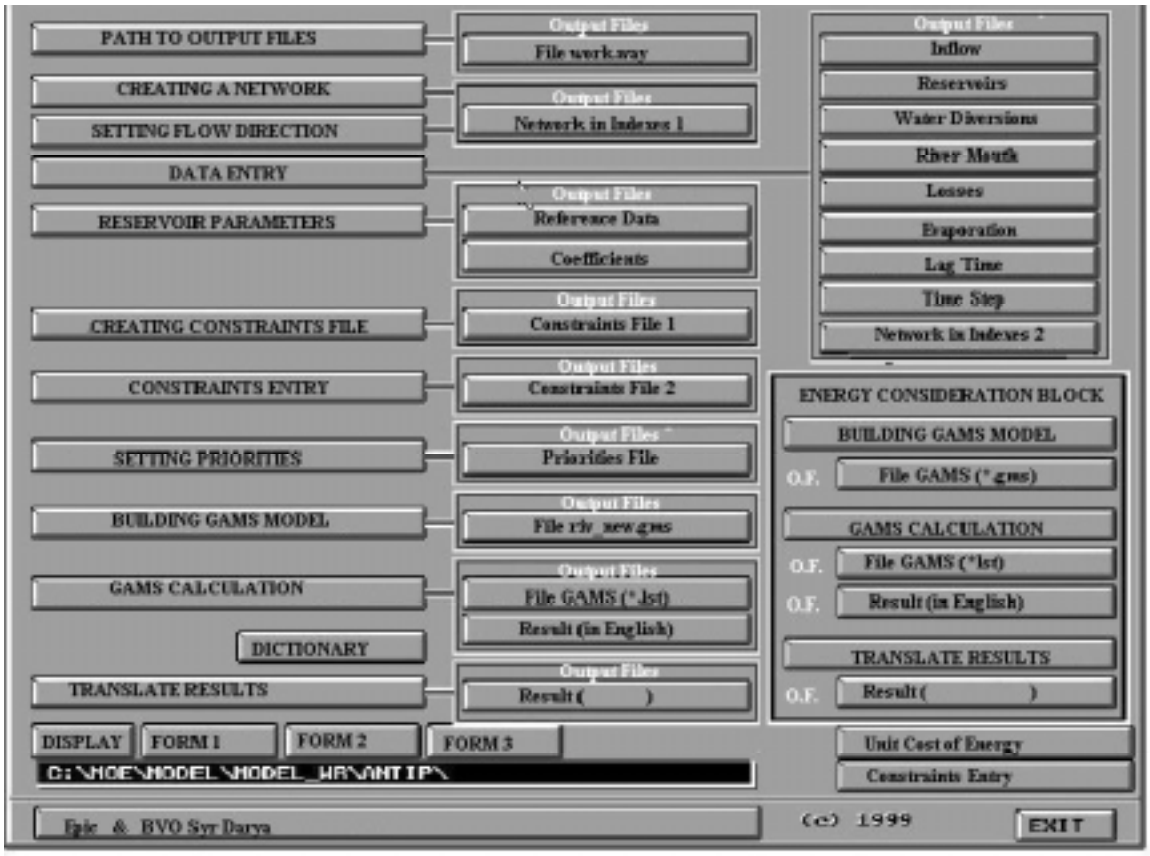


Figure 26. Energy Block Menu.

Determining Power Production Costs Module

You activate the **DETERMINING POWER PRODUCTION COSTS-energy.exe** module by placing the cursor on the **Power Unit Cost** button of the menu and pressing the left mouse button at the same time.

On the screen, the user can see a panel to enter cost indicators

Production Cost of Minor Thermoelectric Power Stations	
Production Cost of Major Thermoelectric Power Stations	
Production Cost of Hydroelectric Power Stations	
Consumers' Damage from Power Undersupply	
Unit Cost of Power Supplied through External Systems	

Make Your Choice:
Esc Exit
F2 Save & Exit

The primary task of the **energy.exe** module is building and then editing the **money.in** file. This file is built in such a way that on one hand, it is a table containing cost indicators of power production, and on the other hand, it may be incorporated in the text of a model created under the rules of the **GAMS** compiler.

When you run the **energy.exe** module for the first time, it fills all cells of the table of cost indicators by default. For default, it uses the unit costs of power, which were in the Kyrgyz Republic at the beginning of the year 2000.

The user may change these indicators in any way he wishes. Overstating power production costs for some type of stations, the user can withdraw this type from calculation.

Below is presented a version of the **money.in** file.

```
*  small_heat   big_heat  hydro_staion      UDC       USERS
*    3.70000  92.00000    1.30000    200.00000    100.00000
P1=      3.70     ;
P2=     92.00     ;
P3=      1.30     ;
P4=    200.00     ;
P5=     100.00     ;
```

This file is a component of the energy consideration complex, therefore its presence in a working directory is required. In other words, running this executable module at least once is a sine qua non condition for the energy consideration complex to operate.

In correcting this file by an external editor the user should not delete asterisks in the first position of the first two lines because under the GAMS compiler's rules they are treated as a comment.

The structure of the next five lines should not change. The number of spaces is insignificant.

The amount of damage to users resulted from undersupplied power should be higher than the power production cost of any station. Otherwise, unreasonable disconnection of power producing facilities may considerably infringe users.

Constraints Entry Module

The **CONSTRAINTS ENTRY-r_ch_lie.exe** module starts operating as you place the cursor on the **CONSTRAINTS ENTRY** button of the menu and press the left mouse button.

On Figure 27, we present the menu to operate this module. Before imposing constraints on specific entities of the power system, it is useful to run the **Navigator** mode (see Figure 28) and study the structure of interrelationships between entities of the power system. The structure of

entities is universal for this type of problems and is shown on Figure 28. Three producing complexes (minor thermoelectric power stations, major thermoelectric power stations, and hydroelectric power stations) supply power to the node of power collection. The node of power collection supplies power to the node of power distribution.

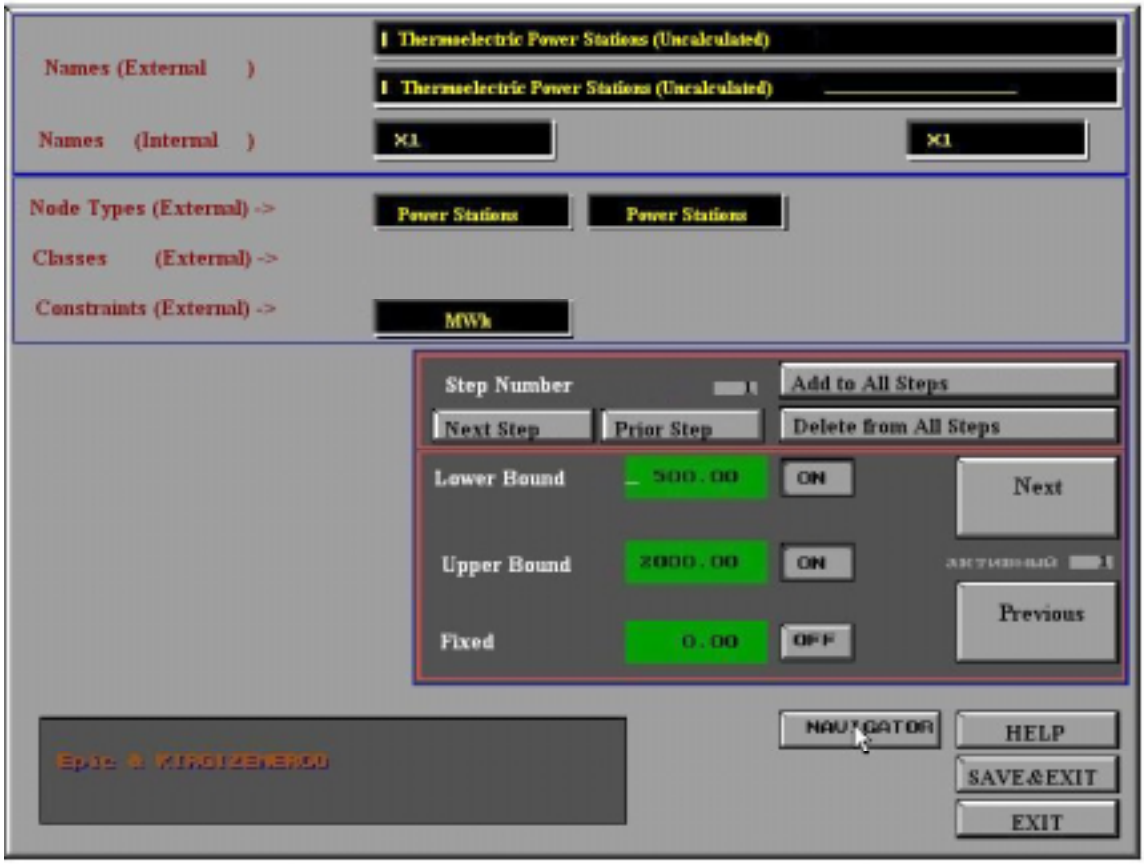


Figure 27. Energy Constraint Constructor.

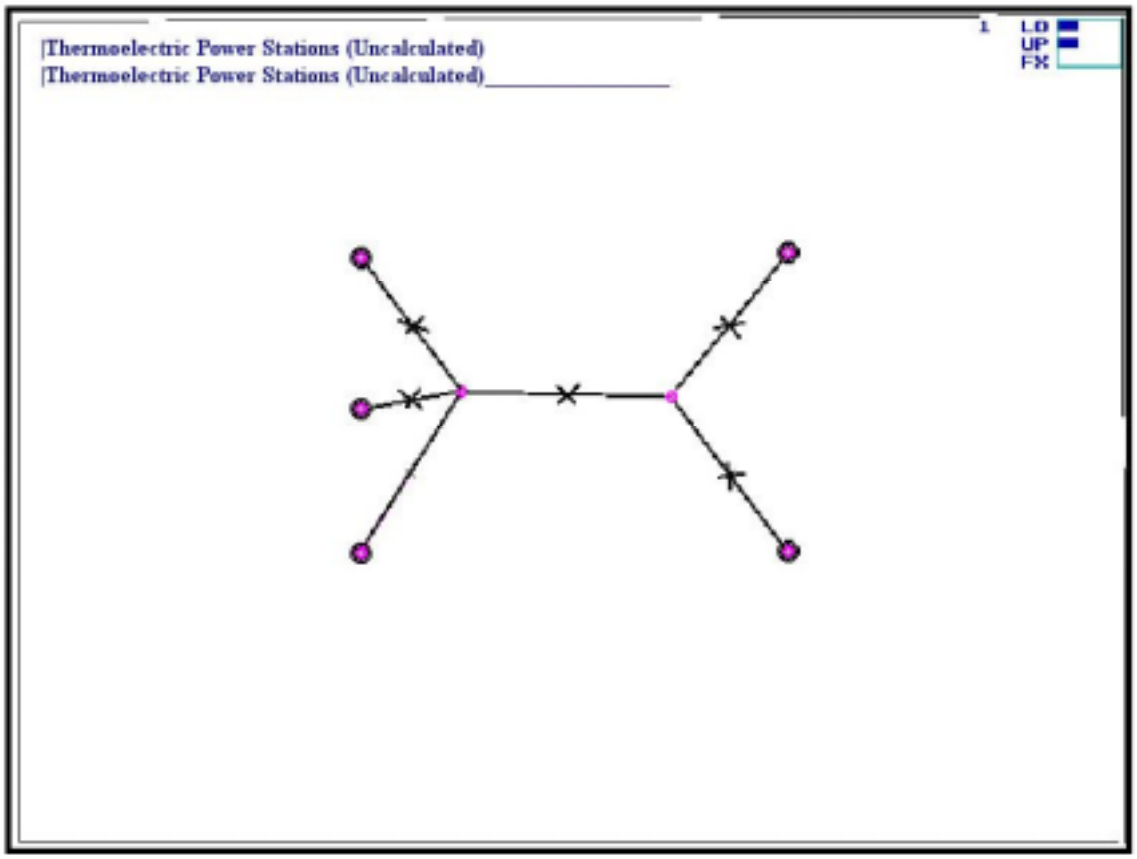


Figure 28. Navigator Mode for Energy Block.

The node of power distribution divides power between an aggregated user and an external power system. In addition to the 7 nodes, directions of the arcs connecting these nodes are described above. It is clear that the only arc along which power flow may be negative is the arc connecting the node of power distribution and the node representing an external power system in the aggregate. If positive, power moves towards the external system; if negative, power moves in from the external power system.

The user has access to five nodes of the power system to enter constraints on power production and consumption and six arcs to enter constraints on power transfers. Placing the cursor on one of the five nodes or one of the six arcs (zones marked with crosses) and pressing the right mouse button, the user returns to the main menu of constraints entry adjusted for the entity specified by **Navigator**.

The user can see a menu that enables him to enter constraints:

1) fixed value
2) minimum bound
3) maximum bound

This module has almost all capacities, which are incorporated and described in the module of water constraints entry in the base water management complex described above. The menu's appearance, dialog boxes, and controls – everything almost completely coincides by appearance and location with the constraints entry module in the base complex *Optimal Water Management* described above. Some data-commenting lines are changed. Dialog boxes show names of aggregated nodes of the power system, which are external for the GAMS model.

Differences are in the missing of several capacities to recalculate from one dimension to another the constraints entered but not needed for power consideration. In other words, buttons to convert water consumption to flow are missing (so do not operate). It is clear that you do not them because all constraints are in MWh for any entity of an aggregated power scheme or a power transmission line.

The data output file is the **cnstr_e** file. The structure of this file identically coincides with the above-described structure of the **cnstr** file defining water constraints.

Out of the constraints entered, the complex of models will use for calculation only those, for which buttons **ON/OFF** will be initiated. All other constraints will be incorporated into the internal database of constraints, but the model will not use them. The procedure of working with buttons and constraints entry boxes is described in the guide for the base model ***Optimal Water Management in a River Basin*.**

Below is given an example of a constraints file incoming to a complex **GAMS** model.

```
e.lo('H1','m1')  =    60.00; e.up('H1','m1')  =   345.00; e.lo('H1','m2')  =   125.00;
e.up('H1','m2')  =   345.00; e.lo('H1','m3')  =   125.00; e.up('H1','m3')  =   345.00;
e.lo('H1','m4')  =   130.00; e.up('H1','m4')  =   345.00; e.lo('H1','m5')  =   130.00;
e.up('H1','m5')  =   345.00; e.lo('H1','m6')  =   125.00; e.up('H1','m6')  =   345.00;
e.lo('H1','m7')  =    60.00; e.up('H1','m7')  =   100.00; e.lo('H1','m8')  =    15.00;
e.up('H1','m8')  =    45.00; e.lo('H1','m9')  =    15.00; e.up('H1','m9')  =    45.00;
e.lo('H1','m10') =    15.00; e.up('H1','m10') =    45.00; e.lo('H1','m11') =    15.00;
e.up('H1','m11') =    45.00; e.lo('H1','m12') =    15.00; e.up('H1','m12') =    45.00;
*
```

The most important distinctive of this executable module is that to ensure its operability you need not specially prepare an internal database to locate constraints. The internal database of constraints **limitale** is formed independently when you run this module for the first time or is read from a working directory if it has been created before. Such a degree of automation is achieved because the appearance and structure of the aggregated power system do not change from task to task. Therefore, you may carry the internal database of constraints **limitale** from one directory to another, and that will not affect the operability of the **CONSTRAINTS ENTRY** module.

Structurally, the file of the internal database of constraints **limitale** completely coincides with the structure of the files **limitall** and **limitals** described in the tutorial to the base model ***Optimal Water Management in a River Basin***.

A sine qua non condition of operability of the ***Optimal Water Management in a River Basin*** complex model with regard to energy is running this module at least once. The complex model will seek the **cnstr_e** file, and in case it is missing the model will display an error message. An empty **cnstr_e** file surely contains a comment line, what is provided through an algorithm while this file is generated.

Building GAMS Model Module (with Regard to Energy)

You activate the **BUILDING GAMS MODEL-builde.exe** module if you place the cursor on the **BUILDING GAMS MODEL** button of the virtual menu and press the left mouse button.

The user can in no way influence the process of building an energy consideration model. The user sees on the screen running comments concerning model building. The only thing the user can do is to abort model building by pressing the keys **CTRL + BREAK**. The result of this module's operation is the **riv_newe.gms** file. The most stable part of the model in the **model_e** file is connected to this file. This division enables the user to raise the model's capacity and modernize its versions.

The question of rewriting the last file **model_e** envisages two answers: recreate or leave the old version. Running the **BUILDING GAMS MODEL-builde.exe** module for the first time, you have to create this subblock of the model because the **GAMS** compiler needs it to operate.

Through a special button of the main menu, the user has access to the **riv_newe.gms** file for editing and viewing (left part of the button for the **riv_newe.gms** file and right part of the button for the **model_e** file) Along with its own file of constraints **cnstr_s, money.in**, the model incorporates the files **cnstr** and **digits.in** created in forming an optimal water management model. That is why the condition of preliminary forming and calculating the model *Optimal Water Management in a River Basin* is of importance.

This model is bigger and more complicated than the model *Optimal Water Management in a River Basin*. However, the principles of building its equations are similar and based on the theory disclosed thoroughly in a formalized description of this complex task.

The analysis of the created model shows that its water part is actually identical to the water part of the base model of BVO Syrdarya. Several variables and a group of equations are added, however. This block is a creatively reconceived and considerably reinforced in scientific terms version of a training problem stated in the GAMS tutorial developed by the EPIC Program.

Into the power consideration subblock, we added a group of variables and equations.

Defined variables: **iii(j,j1,m)** = power flow in MW from node **j** to node **j1** for each time **m**.
E(j,m) = production or consumption of power in node **j** at time **m**.

We defined sets of nodes, names of which dialog boxes of the constraints entry module described above display.

me_G(j) = set of hydroelectric power stations
me_H(j) = set of calculated thermoelectric power stations
mf(j) = set of non-calculated power stations
ma(j) = set of consumption nodes (including external power systems)
coni(j,j1,m).. = name of the equation applied to all pairs of nodes in all time intervals
iii(j,j1,m) =e= -iii(j1,j,m); = power flow gone out from node **j** to node **j1** equals power flow come into node **j1** from node **j**

---

conue_G(j,m)$me_G(j).. = name of the equation applied to all nodes of hydroelectric power stations in all time intervals
sum(j1$ener(j,j1),iii(j,j1,m))=e= e(j,m); = sum of incoming and outgoing power flows equals power production at hydroelectric power stations

---

conue_H(j,m)$me_H(j).. = name of the equation applied to all nodes of calculated thermoelectric power stations in all time intervals
sum(j1$ener(j,j1),iii(j,j1,m)) =e= e(j,m); = sum of incoming and outgoing power flows equals power production at thermoelectric power stations

---

conuf(j,m)$mf(j).. = name of the equation applied to all nodes of non-calculated thermoelectric power stations in all time intervals
sum(j1$ener(j,j1),iii(j,j1,m)) =e= 0; = sum of incoming and outgoing power flows equals power production at non-calculated thermoelectric power stations

---

conua(j,m)$ma(j).. = name of the equation applied to all nodes of power consumption in all time intervals
sum(j1$ener(j,j1),iii(j,j1,m)) =e= -potr_f(j,m); = sum of incoming and outgoing power flows equals power consumption in aggregated nodes of power consumption

---

The equation providing connection of the base model of Syrdarya with the power consideration subblock

water(j,m)$me_G(j).. = name of the equation applied to all nodes identified as hydroelectric power stations
sum(n1$V(n1),power(n1,m)) =e= e(j,m); = sum of power generations at a subset of hydroelectric power station nodes from the basic task of BVO Syrdarya equals the hydropower production in the power consideration subblock developed in Kyrgyzenergo.

We added two components into the fair objective function:

First component is the difference between the power consumption demand made by a user and the power amount actually supplied to the user.

1000* SUM(j$ma(j),SUM(m,((potr(j,m)-potr_f(j,m))* (potr(j,m)-potr_f(j,m)))))+

Second component is the cost of total power production with regard to its incoming or receiving through an external power system.

SUM(j$me_H(j),(price(j)*SUM(m,e(j,m))))

Conducting Calculations Module (with Regard to Energy)

You activate the **CONDUCTING CALCULATIONS** mode by placing the cursor on the **GAMS CALCULATION** button and pressing the left mouse button. Thereby, you form a command line connecting the **GAMS** compiler to the created complex model *Optimal Water Management with Regard to Energy*. While running this mode, the main listing of results **riv_newe.lst** and the main output file **rivere.new** located in a working directory are formed.

The first of the above mentioned files is necessary to assess a solution found, to control conducted calculations, and to seek and identify errors.

The second file in compressed tabular forms presents an identified solution (in English and using internal names of the model). Structurally, it does not differ from the **river.new** file described in the basic part of the program system. However at the end of the file the user can see supplementary information in a tabular form and with needed comments. This information concerns optimal operation of hydroelectric power stations, thermoelectric power stations with regard to all constraints on water and power transfers between all nodes of an aggregated power network. The user gains access to these files, perhaps for correction, through the buttons of the water salinity consideration block.

Translating Calculation Results Module (with Regard to Energy)

This module is described in the basic part of the program system, and in this case it is used in this subblock a second time. Reorientation of this module to new files is carried out in a command line, as you run this module (see the basic part of the program system). The file of results to which the user gains access through a button in the menu has the name **rivere.trn.**

Results of Testing the Energy Block

Initiators and actual developers of this block were specialists of the Coordination Group and the Technical Group from the Kyrgyz Republic. Therefore, most tests of this program system were conducted within the problem statement where the downstream of the Uchkurgan

hydroengineering complex was the main control section and all five reservoirs of the Naryn cascade were considered simultaneously.

The specialists conducted a test calculation based on actual constraints and concluded interstate agreements concerning guaranteed water releases and assigned power transfers to the Power Grid of Central Asia.

A fragment of a result on power transfers from the test calculation file for one time step says that

	X1	A1	H1	O1	F1	F2	G1	
X1	0.00	0.00	0.00	0.00	-20.00	0.00	0.00	non-calculated stations
A1	0.00	0.00	0.00	0.00	0.00	533.57	0.00	consumers
H1	0.00	0.00	0.00	0.00	-15.00	0.00	0.00	calculated stations
O1	0.00	0.00	0.00	0.00	0.00	-482.34	0.00	external power system
F1	20.00	0.00	15.00	0.00	0.00	-51.23	16.23	node of power collection
F2	0.00	-533.57	0.00	482.34	51.23	0.00	0.00	node of distribution
G1	0.00	0.00	0.00	0.00	-16.23	0.00	0.00	hydroelectric power stations

For example, hydroelectric power stations put 16.23 MWh into the node of power collection for a time step. You can understand the rest of the figures by analogy.

You may familiarize yourself with calculation results more particularly in the seminar materials of the Technical Group from the Kyrgyz Republic.