TBC
Transport, Biochemistry and Chemistry

User's guide for TBC - Version 2.01

An Efficient Simulator for
Three-Dimensional Saturated Groundwater Flow
+ Multi-Species Transport
+ Reactions
in Porous Formations

Dirk Schäfer
Wolfgang Schäfer[*]
René Therrien

[*] Dr.-Ing. Wolfgang Schäfer
Grundwassermodellierung
Odenwaldstr. 6
D-69168 Wiesloch

September 2003

Introduction

TBC 2.01 is a further development of the numerical model FLOWBIO developed by René Therrien and Wolfgang Schäfer, Kassel University, March 1993. TBC is able to simulate three-dimensional saturated groundwater flow as well as transport and (bio)chemical reactions of any number of model species. The flow and transport equations are discretised by using the finite difference method, resulting in a seven-point template, or by using the Garlerkin finite element technique, resulting in a 27-point discretization molecule. The use of the seven-point template considerably reduces memory and CPU time required for a solution. The final system of equations is solved by a very fast and efficient preconditioned ORTHOMIN solver.

The biochemical reactions are described by Monod-type kinetics and the equation system is solved by a special DGEAR routine. Chemical reactions can either be defined as a set of fast equilibrium reactions or as kinetic chemical reactions. The equation system is solved by a Newton-Raphson solver.

The objective of this guide is to describe the requirements for the TBC input file which should be named <name>.in . The manual only describes the formal creation of the input file. It is not suited to explain the physical, chemical or microbiological background of the model. For a meaningful application of TBC the user should have basic knowledge of hydraulics, microbial ecology, equilibrium chemistry and kinetic chemistry. Some standard text books on these issues are Morel (1983), Stumm and Morgan (1996), Schlegel (1985), Kinzelbach (1992), and Chapelle (1993). A more detailed description of TBC can be found in Schäfer et al. (1998a and 1998b).

Data Input

In this chapter the creation of the general input data file for the numerical simulation is described in detail. First, some general information about format and nature of the input data is given.

A title name for the simulation needs to be chosen before starting the simulation. This name, designated as the problem prefix, determines the input and output filenames. The **general input file name** consists of the problem prefix and the extension **".in"**. The **general output file name** consists of the problem prefix which the letter **"o"** and the extension **".out"** are appended to. (For example, if the problem prefix is "test", the user has to create the general input file "test.in". The general output file generated by the program is "testo.out".)

Some simulations require more than one input file and result in the generation of more than one output file. As a general rule, the names of most input files needed for a specific simulation consist of the problem prefix plus a given extension and all output filenames are composed of the problem prefix which the letter "o" and the given extension are appended to. (For example, the given extension of an input file containing initial Dirichlet boundary conditions always has to be ".dbc"). Input files for hydraulic conductivities and initial concentrations have no predefined names. The model finds the path of the general input file in a file called **tbc.fil**, which has to be prepared in the subdirectory of the program. The first line of tbc.fil contains the complete pathname, the second line the problem prefix.

tbc.fil:

<pathname>	/user/example/problem/
<problem prefix>	test

The general input data file is divided into several groups and each group is identified by a single line of format A80. These identification lines make it easier to examine and modify the data groups. All other input is free-format except for alphanumeric or logical variables (which are .true., .false., T or F). Furthermore the standard FORTRAN norm for naming variables is applied, where i.e. INTEGERS are represented by variables whose names begin with the letters I to N, and REAL variables start with the remaining letters (A to H and O to Z).

In this manual groups and data items are listed and described according to order of appearance in the general input file. First, the general form and an example are given

<general form>   example,

then all parameters are described in detail.

The following conventions are used throughout the description:

The name of FORTRAN variables representing data items are written in bold font. There are three different classes of variables:

- **DATAITEM1** If the variable name is written in capital letters, the variable **must** be present in the general input file.
- **dataitem2** If the variable name is written in small letters, its appearance in the general input file is optional. Whether or not they are present depends on the specific problem that is considered.

- **DATAITEM3 (maxsize1, maxsize2, ..., maxsizeN)** Array variables can consist of one or several dimensions. Variable names are followed by brackets '()'. The number(s) inside the brackets define(s) the maximum size of the array. The maximum size of data items represented by empty brackets is equal to the last listed maximum size. The allowed maximum array sizes are specified as PARAMETERS in the file **TBC.PRM**. The program checks the array sizes and issues an error message in the general output file if an array size exceeds the parameter size listed in TBC.PRM. In order to increase parameter sizes of one or more arrays, they must be changed in TBC.PRM and the program must be compiled and linked again.

Units are not predefined in the program. The user must decide for units for length (L), time (T) and mass (M) of the various input variables and consistently use those chosen units. For example, if one wants to specify the dimensions of a domain in metres (m) and the time in seconds (s), all measurements of length and time will have to be in metres and seconds, respectively. The hydraulic conductivity should therefore be specified in m/s, the pumping rate in m^3/s etc.. The program does not perform any checks to ensure unit consistency. If chemistry is considered in the simulation, mass has to be defined in units of moles.

Often the domain of computation can be divided into several zones with different properties. The assignment of properties to the zones is accomplished by delimiting these zones through coordinates in the x-, y- and z-directions. In this case the input data set of some data items is also characterized by these zones. The variables used are "xfrom", "xto", "yfrom", "yto", "zfrom", "zto" and they define a regular volume. The coordinates must correspond to nodal coordinates. Otherwise the program issues an error stating that it could not define the zone in question. Most distributed variables can alternatively be provided in a node by node or an element by element mode.

General input file

As previously mentioned, the general input file is divided into several groups. The first twelve groups contain data defining the geometry of the model domain and defining the hydraulics of the system. Data for transport simulation are specified in groups 13 to 20. Biochemical data are declared in group 21. The input for equilibrium and kinetic chemistry is done in group 22 and 23. Each group consists of the following:

LINE 1: Group title (FORMAT A80)

FOLLOWING LINES: Group data

The group titles separate the groups and help reading the input file. Each group title has to start with an asterisk ("*"). The program checks the first letter of the group title, if the first letter is no "*", an error message is displayed and the program stops. This is designated to help finding errors in the input file.

In the following table all groups and titles are listed:

DATA GROUP 1 Title group 7
DATA GROUP 2 Simulation control parameters 7
DATA GROUP 3 Output control parameters 9
DATA GROUP 4 Grid data 10
DATA GROUP 5 Physical flow parameters 14
DATA GROUP 6 Initial piezometric head data 17
DATA GROUP 7 First-type boundary conditions for flow 19
DATA GROUP 8 Second-type boundary conditions for flow 22
DATA GROUP 9 Source and sink data for the fluid 25
DATA GROUP 10 Observation well data 27
DATA GROUP 11 Time specifications 28
DATA GROUP 12 Orthomin solver data for flow 31
DATA GROUP 13 Control parameters for transport simulation 33
DATA GROUP 14 Output control parameters for transport 33
DATA GROUP 15 Transport parameters 34
DATA GROUP 16 Initial condition data for transport 36
DATA GROUP 17 First-type boundary conditions for transport 40
DATA GROUP 18 Third-type boundary conditions for transport 45
DATA GROUP 19 Concentration at injection wells 52
DATA GROUP 20 Orthomin solver and time data for transport 54
DATA GROUP 21 Biochemical reactions 58

DATA GROUP 22 Equilibrium Chemistry 62
DATA GROUP 23 Kinetic chemistry........ 68

The following is a description of the input data for the different groups. For reasons of brevity, the group title is not mentioned but it is understood that it must be the first line of the group.

DATA GROUP 1 Title group

```
<TITLE>             TBC-Test
```

1) **TITLE** A title that describes the problem in 80 characters or less.

DATA GROUP 2 Simulation control parameters

```
<FINITE_DIFF>                 .true.
<TIME_STEP_CONTROL>           .false.
<TRANSPORT>                   .true.
<SOLVE>                       .true.
<GEN_GRID>                    .true.
<NX>                          31
<NY>                          21
<NZ>                          2
<NLN>                         8
```

1) logical switch **FINITE_DIFF** *finite difference / finite element*

TRUE (default): The finite difference method is used to discretize flow and transport equations. Note that the variables maxnja and maxnja2, specified by the PARAMETER statement in the file tbc.prm, control the size of the global matrix to be solved. The size of the matrix is proportional to the number of nodal connections in the grid. Because of the smaller number of nodal connections a finite difference representation will require less memory than a finite element representation.

FALSE: A finite element formulation is used for the flow and transport equations.

2) logical switch **TIME_STEP_CONTROL** *Automatic time step control*

TRUE: Automatic time step control is performed for transient flow and for transport, if simulated. Here time steps do not need to be specified prior to the simulation. The user only has to declare times (in group 11 or 20) at which a solution output is desired (target times).

FALSE (default): Time-steps for transient flow and transport, if simulated, have to be defined by the user.

3) logical switch **TRANSPORT** *Simulation of reactive transport*

TRUE: (default): Based on the current flow field, a transport solution is computed for each time step. For example: for steady-state flow conditions, the transport solution is computed for the desired output time(s) after computation of the steady-state hydraulic head distribution.

FALSE: Only flow is simulated.

4) logical switch **SOLVE** *Solution of the problem*

TRUE (default): The complete solution procedure is carried out normally.

FALSE: The program is stopped prior to the start of the time-step procedure but after assigning of arrays and grid parameters as well as after checking of array sizes. This can be useful for very large problems, where it is desirable to make sure that all the input is correct, before actually carrying out the simulation.

5) logical switch **GEN_GRID** *Automatic grid generation*

TRUE: A regular grid is automatically generated according to specifications provided by the user (group 4).

FALSE: The grid is read from a data file which needs to be specified in group 4.

6) **NX** Number of nodes in x-direction

7) **NY** Number of nodes in y-direction

8) **NZ** Number of nodes in z-direction

9) **NLN** Number of local nodes for each 3D element. The type of three-dimensional elements that will be used depends on the value of nln. It must be 8 (default) for block elements or 6 if triangular prisms are used. Mixing the elements is not possible.

DATA GROUP 3 Output control parameters

<KPMESH>, <KPHEAD>,<KPVEL>, <KRESTAR>, <KWRITH>, <KPMASB> 0, 1, 1, 0, 0, 1

<ECHO_TO_OUTPUT> .false.

The group consists of integer switches that control the output of various quantities to BINARY files, and of the logical switch <ECHO_TO_OUTPUT> which defines if output is additionally generated in the general output file in ASCII format.

The values of the <KPHEAD>, <KPVEL> and <KPMASB> switches determine time intervals between every data output. For example: <KPHEAD> equal to 5 results in an output of nodal heads at every fifth time-step. <KPHEAD> equal to 0 means that no output is desired. If steady-state flow is simulated, <KPHEAD>, <KPVEL> and <KPMASB> have to be 1 to obtain a single output of heads, velocities and fluid mass balance.

☞ If the value of one of the above described integer switches is greater than zero and the logical switch <TIME_STEP_CONTROL>of group 2 is set to .true., the output will be done at the specified target times (specification in group 11 or 20).

1) **KPMESH** Controls output of grid information to the general output file. For KPMESH = 0 no grid information is given in the general output file while for KPMESH = 1 nodal coordinates and for KPMESH = 2 nodal coordinates and the incidences of the elements are printed.

2) **KPHEAD** Controls output of nodal heads in BINARY form to unit 40. The binary file corresponding to unit 40 will automatically be named <problem prefix>o.hds.

3) **KPVEL** Controls output of elemental velocities in BINARY form to unit 43. The corresponding file will automatically be named <problem prefix>o.vel.

4) **KRESTAR** Variable that specifies if initial conditions have to be read from a BINARY file. If KRESTAR = 1, heads from a previous simulation are used as initial conditions for the present simulation. The initial head values, calculated in a previous simulation, must be in a file called <problem prefix>.hin. (unit 15). (Note: This is an input file for the

simulation. Therefore the filename consists of the problem prefix without the letter "o" appended to it). If KRESTAR = 0 the initial piezometric heads are defined in the general input file (group 6). This switch is used in combination with KWRITH.

5) **KWRITH** Variable that specifies if head values of the last simulated time step are to be written to a BINARY file. If KWRITH = 1, the head values of the last simulated time step will be written to a file named <problem prefix>o.hen. If it is equal to 0, the head values are not written to this file. The switch, in combination with KRESTAR, enables the user to continue a former simulation. If the subsequent simulation is called <new problem prefix>, the generated file, <problem prefix>o.hen must be renamed <new problem prefix>.hin, and in the new simulation the switch KRESTAR must be set to 1.

6) **KPMASB** Controls output of the fluid mass balance into the general output file. The following applies to this switch:

> 0: output at every KPMASBth time step. If TIME_STEP_CONTROL is used, output will be done at target times.

= 0: no output of the fluid mass balance.

9) logical switch **ECHO_TO_OUTPUT** *Output into the general output ASCII file.*

TRUE: Heads and velocities (according to the values of the switches KPHEAD and KPVEL) will also be written to the general ASCII output file <problem prefix>o.out. This enables a direct visual examination of the results which is not possible by means of the BINARY file. However, depending on the size of the problem, ASCII output can result in a very large general output file, the size of which can be up to several megabytes. This switch has no effect on the other switches in group 3.

FALSE (default): Results are only written to BINARY files.

DATA GROUP 4 Grid data

The data input for group 4 depends on the switch <GEN_GRID> set in group 2. Several options for reading or generating the grid are offered. A regular grid can be automatically generated with given values of nodal coordinates. It can also be partially generated by duplicating a previously defined two-dimensional xy-slice in z-direction. The grid can also be entirely read from data files.

Two choices of three-dimensional elements are offered: block elements (8 nodes) or triangular prism elements (6 nodes) (see variable NLN in group 2). For triangular prisms, it is assumed that the triangles lie in the xy-plane. If they do not, some warnings will be issued when the user defines different zones by using coordinates for flow properties, boundary conditions etc. If those warnings appear, nodal numbers should be used to define zones.

According to the various options available the following description is divided into several cases:

(I) Input data if **GEN_GRID** is **.true.** (the grid is automatically generated)

<variable_space>   .false.

1) logical switch **variable_space**   *Specification of variable or constant grid space.*

TRUE: Variable grid space. The values of Dx, Dy or Dz are not constant throughout the entire grid.

Enter:

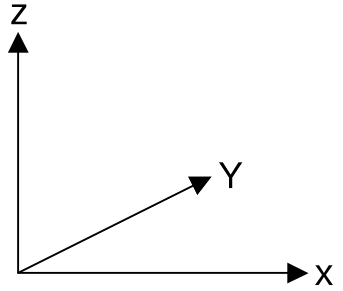
<x_1>, <x_2>, ... , <x_{nx}>	0, 1.0, 1.3, 1.7 ... , 29.4, 30
<$y_{1>}$, <y_2>, ... , <y_{ny}>	0, 0.8, 1.2, 2.3 ... , 9.65, 9.7
<z_1>, <z_2>, ... , <z_{nz}>	0. 0.1, 0.12, 0.17

2) **xi(maxnx):** x-coordinates of the nodes, from x=1 to x=nx. [L]

3) **yi(maxny):** y-coordinates of the nodes, from y=1 to y=ny. [L]

4) **zi(maxnz):** z-coordinates of the nodes, from z=1 to z=nz. [L]

Coordinate system:



FALSE: Uniform grid space. The value of Dx (internodal space in x-direction) is uniform within the grid. The same holds for Dy and Dz (internodal space in y- and z-direction). The space dimension does not have to be the same in the x-, y- and z-dimensions.

Enter:

<xl>, <yl>, <zl>	30, 20, 1

2) **xl, yl, zl**: Total domain lengths in x-, y- and z-direction [L]. The grid will be generated according to these dimensions and the number of nodes in x-, y- and z-directions (nx, ny, nz) specified in group 2.

(II) Input data if **GEN_GRID** is **.false.** (Grid is partially or entirely read from a data file)

<iwhole>	0

1) **iwhole:** Integer switch that specifies if the entire grid or only one xy-slice, composed of 2D-elements, is to be read.

If <iwhole> is equal to 0, only a horizontal xy-slice has to be read (nodal coordinates and incidences of elements) and the 3D-grid is generated by duplicating this slice in z-direction for specified z-values. The xy-slice can either be composed of rectangular (if 3D-block elements are to be constructed) or triangular elements (if 3D triangular prisms are to be constructed).

If <iwhole> is equal to 1, the entire 3D-grid has to be read from a binary file (output of a grid generator). If <iwhole> is equal to 2, the entire 3D-grid has to be read from an ASCII file

iwhole=0: One xy-slice is read. The below specified input files must contain the following data:

The ASCII file **<problem prefix>.nod** must contain (unit 11 in the program):

nndsl:	Number of nodes for one xy-slice
x(maxnn), y():	x- and y-coordinates for each of the nndsl nodes [L], [L]
nz:	Number of xy-slices (in z-direction)
zi(maxnz):	z-location of each of the nz slices [L]

The ASCII file **<problem prefix>.inc** must contain (unit 12 in the program):

nesl: Number of elements for one xy-slice

in(maxnln,maxne): Nodal incidences for each 2D-element lying in the xy plane

Note: These input files have to be in the same directory as the input file <problem prefix>.in

iwhole=1: The entire 3D-grid is read from binary files. The below specified input files must contain the following data:

The binary file **<problem prefix>.nod** must contain (unit 11 in the program):

nn, x(maxnn), y(), z(): Number of nodes and the x-, y- and z- coordinates for each node. The coordinates have to be entered in the following order: x(1), y(1), z(1), x(2), etc..

The binary file **<problem prefix>.inc** must contain (unit 12 in the program):

ne, in(maxnln, maxne): Number of 3D-elements and nodal incidences for each element. They have to be entered in the following order: in(1,1), in(2,1),...., in(nln,1), in(1,2), etc..

iwhole=2: The entire 3D-grid is read from ASCII files. The below specified input files must contain the following data:

The ASCII file **<problem prefix>.nod** must contain (unit 11 in the program):

nn, x(maxnn), y(), z(): Number of nodes and the x-, y- and z- coordinates for each node. The coordinates have to be entered in the following order: x(1), y(1), z(1), x(2), y(1),z(1) etc..

The ASCII file **<problem prefix>.inc** must contain (unit 12 in the program):

ne, in(maxnln, maxne): Number of 3D-elements and nodal incidences for each element. They have to be entered in the following order: in(1,1), in(2,1),...., in(nln,1), in(1,2), etc.. Note, that the order of nodes has to follow the description given in group 8.

DATA GROUP 5 Physical flow parameters

It might be that the domain is composed of several materials with different flow properties. The assignment of properties to zones is done by specifying either nodal coordinates or element numbers that correspond to the boundaries of zones of different materials. The data can optionally be read from a BINARY file (logical switch <K_RAND> is true), or from an ASCII file (logical switch <K_SURFER> is true), or it can be entered into this group by the user

<K_RAND>, <K_SURFER>	.false., .false.

1) logical switches **K_RAND and K_SURFER** *How to enter flow properties*

☞ If one of these two switches is true, the other one must be false! If both are false, the data has to be defined by the user.

There are three cases to consider:

(I) Input if **K_RAND** is **.true.** and **K_SURFER** is **.false.**

Elemental hydraulic conductivity values are read from a BINARY file (unit 25) named <problem prefix>.gen, which is created by a random field generator. TBC is compatible with the output from FGEN [*Robin et al.*, 1992] but other generators can also be used. The following three data lines must be entered:

Enter:

<ss>, <porb>	1e-4, 0.2
<convf>	1e-6
<vert>	10

2) **ss** Specific storage coefficient [1/L], assumed to be constant for the entire domain.

3) **porb** Effective porosity [-], assumed to be constant for the entire domain.

4) **convf** Conversion factor that adjusts the units of hydraulic conductivity obtained by the random generator to TBC-units:

$$k_{f,\,TBC} = \text{convf} * k_{f,\,\text{random generator}}$$

5) **vert** Anisotropy factor of the vertical hydraulic conductivity. It is defined as the ratio between horizontal and vertical hydraulic conductivity.

(II) Input if **K_RAND** is **.false.** and **K_SURFER** is **.true.**

Both horizontal and vertical hydraulic conductivities are read from separate ASCII files compatible to the Surfer Grid Node Editor format (©Golden Software).

For each horizontal slice the following input is required for the horizontal hydraulic conductivity:

Enter:

<convf>, <ss>, <porb>	1.0e-6, 1e-4, 0.2
<surffile>	data.srf

2) **convf** Conversion factor that adjusts the units of hydraulic conductivity obtained by surfer to TBC-units:

$$k_{f,\,TBC} = \text{convf.} \cdot k_{f,\,Surfer}$$

3) **ss** Specific storage coefficient [1/L], assumed to be constant for the entire domain.

4) **porb** Effective porosity [-], assumed to be constant for the entire domain.

5) **surffile** Name of the ASCII file which has to be read. This file has to be in the same directory as the general input file <problem prefix>.in.

Furthermore for each horizontal slice the following input is required for the vertical hydraulic conductivity:

Enter:

<convf>	1.0e-6
<surffile>	data.srf

6) **convf** Conversion factor that adjusts the units of hydraulic conductivity obtained by surfer to TBC-units:

$$k_{f,\,TBC} = \text{convf.} \cdot k_{f,\,Surfer}$$

The conversion factor may also be used to easily specify a vertical anisotropy, eg:

$k_{f,\ vert} = \text{convf.} \cdot k_{f,\ horz}$

7) **surffile** Name of the ASCII file which has to be read. This file has to be in the same directory as the general input file <problem prefix>.in.

☞ Option (II) allows to declare inactive elements inside the rectangular model area by setting the horizontal **and** vertical hydraulic conductivities of the respective elements to zero. In this case TBC automatically flags the inactive elements and nodes and assigns a piezometric head value of –9999 to them. A node is only flagged as inactive if all surrounding elements are inactive.

(III) Input if K_RAND and K_SURFER are .false.

The hydraulic conductivities are not read from files but need to be defined by the following routine:

Enter:

<use_coord>	.true.

2) logical switch **use_coord** *Use of coordinates or element numbers to define flow property zones*

TRUE: The flow property zones are specified by their x-, y- and z-coordinates (should only be used in combination with gen_grid = true). Enter:

<nzones_prop>	1
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	0, 30, 0, 20, 1, 10
<kxx>, <kyy>, <kzz>, <stor>, <por>	0.1, 0.1, 0.01, 1e-4, 0.2

2a) **nzones_prop** Number of zones that have different flow properties.

For each zone, specify the data described below:

2b) **xfrom, xto, yfrom, yto, zfrom, zto**

Dimensions of the zone are described by its extent in the x-,y- and z-directions. Note that these coordinates must correspond to grid coordinates. Otherwise an error message will be issued.

2c) **kxx(maxkzn), kyy(), kzz(), stor(maxpzn), por()**

Hydraulic conductivities [L/T] in the x-, y- and z-directions, specific storage coefficient [1/L] and effective porosity [-] of the defined zone. It is assumed that the principal directions of the hydraulic conductivity tensor coincide with the x-, y- and z-axes.

FALSE: The different flow property zones are specified by their element numbers. Enter:

<nzones_prop>	1
<i1>, <i2>, <ckx>, <cky>, <ckz>, <ss>, <porb>	1, 750, 0.1, 0.1, 0.01, 1e-4, 0.2

2a) **nzones_prop** Number of zones that have different flow properties.

For each zone enter the following data:

2b) **i1, i2, ckx, cky, ckz, ss, porb**:

Element index of the first element lying in the zone, element index of the last element lying in the zone, hydraulic conductivity [L/T] in the x-, y- and z-directions, specific storage coefficient [1/L] and effective porosity [-] for these elements.

☞ Please note that a single porosity value has to be used for the whole model domain if phase exchange processes are considered. Furthermore, this porosity value has to be equal to the value of <vmob> specified in GROUP 15.

DATA GROUP 6 Initial piezometric head data

(A) If piezometric heads from a previous simulation are used (switch **KRESTAR greater than 0** in group 3), ONLY the following variable needs to be entered:

Enter:

<trestar>	120

1) **trestar** Starting time of the present simulation, that is the time at which the piezometric heads - used as initial ones - were computed in the previous simulation. The initial heads are automatically read from the file <problem prefix>.hin.

The input for this group is completed. ➔ *Proceed to next group.*

(B) If no previously calculated heads are used (switch **KRESTAR equal to 0** in group 3), the initial hydraulic heads have to be entered. This can either be done for the entire domain or for different zones.

Enter:

<DEFAULT_IC>	.true.

1) logical switch **DEFAULT_IC** *Default initial head in the domain*

TRUE: The default initial head specified below is valid for the entire domain.

<hinit>	4.0

2) **hinit** Default initial head in the domain [L].

FALSE: There are zones with initial heads different from the default initial head (should only be used in combination with gen_grid = true).

<hinit>	4.0
<nzones>	2
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	1, 10, 1, 3, 1, 2
<hzone>	5
	12, 20, 5, 8, 1, 10
	3.8

2) **hinit** Default initial head in the domain [L] is valid, where no other value is defined.

3) **nzones** Number of zones in which initial heads are different from the default head <hinit>.

For each zone specify:

4) **xfrom, xto, yfrom, yto, zfrom, zto**

Coordinates specifying the extent of the zone. The coordinates must correspond to nodal coordinates [L].

5) **hzone**

Initial head for the zone delimited by the above defined coordinates [L].

DATA GROUP 7 First-type boundary conditions for flow

In this group Dirichlet boundary conditions (first-type boundary conditions) are considered.

<DIRICHLET_BC> .true.

1) logical switch **DIRICHLET_BC** *First-type boundary conditions (prescribed heads) are present in the domain*

TRUE: There are first-type boundary conditions in the domain.

➔ *Continue in this group.*

FALSE: There are no first-type boundary conditions in the domain.

The input for this group is completed. ➔ *Proceed to next group.*

Enter:

<use_coord> .true.

2) logical switch **use_coord** *Use of coordinates or nodal numbers to define first-type conditions.*

(A) TRUE: Nodal coordinates are used to define zones of first-type boundary conditions (should only be used in combination with gen_grid = true).

Enter:

<boundary_file> .false.

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: The boundary conditions are read from the input file <problem prefix>.dbc.

The data in the input file <problem prefix>.dbc must have the following format:

```
<nzones>                                   1
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>, <hpres>
                                           1, 10, 2, 5, 0, 1, 1.7
```

2b) **nzones** Number of zones with different first-type boundary conditions.

For each zone, the following must be entered:

2c) **xfrom, xto, yfrom, yto, zfrom, zto, hpres**

Coordinates specifying the extent of the zone with prescribed boundary conditions. The coordinates must correspond to nodal coordinates [L].

Prescribed hydraulic head in the respective zone [L].

FALSE: First-type boundary conditions have to be entered into this group.

```
<nzones>                                        2
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>
                                                1, 10, 2, 5, 0, 1
<hpres>                                         1.7
                                                10, 12, 1, 5, 0, 1
                                                2.4
```

2b) **nzones** Number of zones with different first-type boundary conditions.

For each zone, the following must be entered:

2c) **xfrom, xto, yfrom, yto, zfrom, zto**

Coordinates specifying the extent of the zone with prescribed boundary conditions. The coordinates must correspond to nodal coordinates [L].

2d) **hpres**

Prescribed hydraulic head in the defined zone [L].

(B) FALSE: Nodal numbers are used to define zones with first-type boundary conditions.

Enter:

<boundary_file>	.false.

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: Boundary conditions are read from the input file <problem prefix>.dbc.

All data in the input file <problem prefix>.dbc must have the format described in the next paragraph (FALSE).

FALSE: First-type boundary conditions have to be entered into this group.

<ndc>	3
<jm>, <hbc1>	234, 1.5
	548, 2.1
	1012, 1.6

2b) **ndc** Number of nodes with first-type boundary conditions

For each first-type node specify the following:

2c) **jm(maxndc), hbc1()**

Node index, jm(), and prescribed head value, hbc1() [L].

DATA GROUP 8 Second-type boundary conditions for flow

<FLUX_BC>	.true.

1) logical switch **FLUX_BC** *Second-type boundary conditions (prescribed flux) are present in the domain.*

TRUE: Second-type boundary conditions are present in the domain.

➔ *Continue in this group.*

FALSE: Second-type boundary conditions are not present in the domain.

The input for this group is completed. ➔ *Proceed to next group.*

Enter:

<use_coord>	.true.

2) logical switch **use_coord** *Use of coordinates or nodal numbers to define second-type conditions.*

(A) TRUE: Nodal coordinates are used to define zones of second-type boundary conditions (should only be used in combination with gen_grid = true).

Enter:

<nzones>	2
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	1, 1, 4.3, 9.5, 0, 1
<num_plane>, <flux_norm>	4, 3.2
	1, 10, 4, 4, 0, 1
	1, -4.5

2a) **nzones** Number of different second-type boundary condition zones

For each zone specify:

2b) **xfrom, xto, yfrom, yto, zfrom, zto**

Coordinates specifying the extent of the zone with prescribed boundary conditions in x-, y- and z- direction. The coordinates must correspond to nodal coordinates [L].

2c) **num_plane, flux_norm**

Index of face (of every 3D element in the zone) which the second-type boundary conditions are applied to and fluid flux density that is normal to this face.

The following conventions are used:

a) For 3D block elements:

Face 1: Front face (XZ), nodes 1,2,6,5

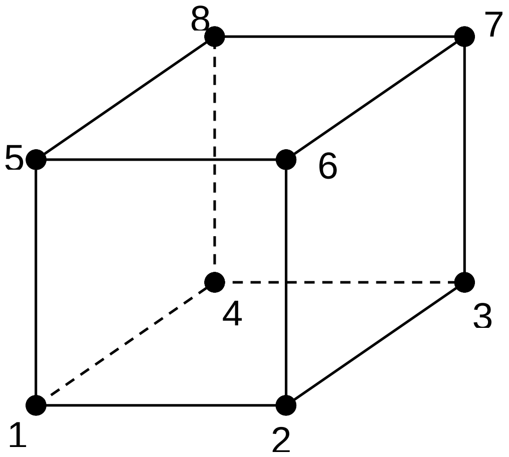
Face 2: Right face (YZ), nodes 2,3,7,6

Face 3: Back face (XZ), nodes 4,3,7,8

Face 4: Left face (YZ), nodes 1,4,8,5

Face 5: Bottom face (XY), nodes 1,2,3,4

Face 6: Top face (XY), nodes 5,6,7,8



b) For 3D triangular prisms:

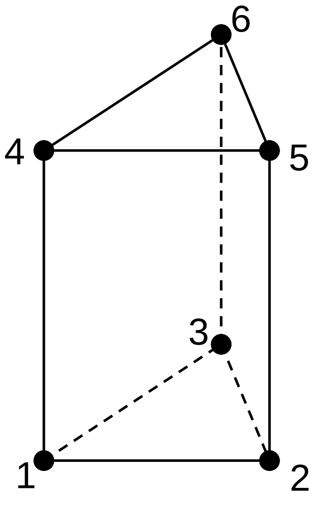
Face 1: Bottom face (XY), nodes 1,2,3

Face 2: Top face (XY), nodes 4,5,6

Face 3: Side face (XY), nodes 1,2,5,4

Face 4: Side face (XY), nodes 1,3,6,4

Face 5: Side face (XY), nodes 2,3,6,5



c) For fluid flux:

By convention, positive flux indicates fluid flowing into the domain and negative flux fluid flowing out of the domain.

(B) FALSE: Element numbers are used to define zones with second-type boundary conditions.

Enter:

<boundary_file>	.false.

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: Boundary conditions are read from the input file <problem prefix>.2tb.

All data in the input file <problem prefix>.2tb must have the format described in the next paragraph (FALSE).

FALSE: Second-type boundary conditions have to be entered into this group.

<nofelem>	3
<iel>, <num_plane>, <flux_norm>	25, 6, 6.e-7
	26, 6, 6.e-7
	27, 6, 6.e-7

2b) **nofelem** Number of elements with second-type boundary conditions

For each second-type element specify the following:

2c) **iel** Element number

2c) **num_plane, flux_norm**

Index of element face on which the second-type boundary conditions are applied to and fluid flux density that is normal to this face.

The conventions explained in 2A are used

☞: All boundaries that are not defined as first- or second-type boundaries in group 7 and 8 represent no-flux boundaries.

DATA GROUP 9 Source and sink data for the fluid

<WELLS>	.true.

1) logical switch **WELLS** *Presence of fluid sources and/or sinks in the domain.*

TRUE: Sources and sinks are present in the domain.

➔ *Continue in this group.*

FALSE: There are neither sources nor sinks present in the domain.

The input for group 9 is complete. ➔ *Proceed to next group.*

<iter_flux>	.true.
<qfac>	1.0
<nwell>	2
<use_coord>	.true.

2) logical switch **iter_flux** *Iteration for variable flux distribution along the well screen.*

TRUE (default): A variable flux distribution along the well screen is computed by an iterative process during the simulation.

FALSE: The flux is assumed to be uniform along the well screen. However, after the discretization, it is internally weighted according to the hydraulic conductivity and the grid size (Dx, Dy, Dz) around the well.

3) **qfac** Convergence factor. Convergence is attained when the back-calculated flux at the well nodes differs less then a factor qfac from the prescribed flow rate (see below). It must be entered, but is only used if <iter_flux> is .true. .

4) **nwell** Number of sources/sinks (wells).

5) logical switch **use_coord** *Nodal coordinates are used for the input*

TRUE: Nodal coordinates are used for the input (should only be used in combination with gen_grid = true).

For each source/sink enter the following:

<flowrate>	-1.7
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	5, 5, 2, 2, 0, 1

5a) **flowrate (maxwell)** Prescribed flow rate [L^3/T] for the well. By convention, the flow rate for an injection well is positive and for a withdrawal well negative.

5b) **xfrom, xto, yfrom, yto, zfrom, zto**

Extent of the well screen in the x-, y- and z-directions. The wells are assumed to be vertical. Therefore, <xfrom> has to be equal to <xto> and equal to the x-location of the well. The same holds for <yfrom> and <yto>. <zfrom> has to be smaller than <zto>.

FALSE: Nodal numbers are used for the input

For each source/sink, the following must be entered:

<flowrate>	-1.7
<nn_well>	3
<jq>	102, 202, 302
<jslice>	1,2,3

5a) **flowrate (maxwell)** Prescribed flow rate [L^3/T] for the well. By convention, the flow rate is positive for an injection well and negative for a withdrawal well.

5b) **nn_well ()** Number of nodes for the present well.

5c) **jq(maxwell, maxwelln)** List of node indices that form the present well.

5d) **jslice(maxwell, maxwelln)** List of slices in which the nodes for the present well specified before by **jq** (5c) are located.

DATA GROUP 10 Observation well data

The output of computed heads (at the location of the observation well) and of computed concentrations (flux averaged along an imaginary passive vertical well) in the domain is possible by means of observation wells. The calculated heads in the observation well(s) are written to the file <problem prefix>.obs which is automatically created when observation wells are specified. The flux averaged concentrations of all defined species are written to the file <problem prefix>.cob. Output is done for every time step.

☞ Observation wells do not affect the result of the computation.

<OBS_WELLS> .true.

1) logical switch **OBS_WELLS** *Presence of observation wells in the domain.*

TRUE: Observation wells are present.

➔ *Continue in this group.*

FALSE: There are no observation wells present in the domain.

The input for group 10 is completed. ➔*Proceed to next group.*

<USE_COORD> .true.

2) logical switch USE_COORD *Use coordinates to define positions of observation wells or use node numbers.*

TRUE: Positions of observation wells are defined by coordinates.

Enter:

<nobs>	2
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	3, 3, 7, 7, 0, 3
	8, 8, 2, 2, 4.2, 6.7

3a) **nobs** Number of observation wells.

For each observation well enter the following:

4a) **xfrom, xto yfrom, yto, zfrom, zto**

Extent of the well screen in the x-, y- and z- directions. The wells are assumed to be vertical. Therefore, <xfrom> has to be equal to <xto> and equal to the x-location of the well. The same holds for <yfrom> and <yto>. <zfrom> has to be smaller than <zto>. Coordinates must coincide with nodal coordinates.

FALSE: Positions of observation wells are defined by node numbers.

Enter:

<nobs>	2
<nn_obsw>	3
<jobs>	3455, 3555, 3655
	2
	439, 539

3b) **nobs** Number of observation wells.

For each observation well enter the following:

4b) **nn_obsw(nobs)** Number of nodes belonging to the observation well.

5b) **jobs(nobs,nn_obsw)** List of node indices that form the present observation well.

DATA GROUP 11 Time specifications

<STEADY_STATE> .false.

1) logical switch **STEADY_STATE** *Steady-state or transient flow simulation.*

TRUE: Steady-state flow is simulated.

The input for this data group is complete. ➔ *Proceed to next group.*

FALSE: Transient flow is simulated.

➔ *Continue in this group*.

<nts> 2

<tw>, <istep> 1, 0

2) **nts** Number of time steps. If an automatic time step control is desired (logical switch <TIME_STEP_ CONTROL> in group 2 is true), <nts> refers to the number of target times.

3) **tw** Time-weighting factor for the flow simulation. It specifies the use of hydraulic heads to calculate the flow and it must have a value between 0 and 1. <tw>=0 means explicit, <tw>=1 means implicit formulation of the flow equation.

4) **istep** Generation of the time step procedure. If <TIME_STEP_CONTROL> (group 2) is true, heads, concentrations, etc. are calculated at target times. Target times have to be defined here. If <TIME_STEP_CONTROL> (group 2) is false, simulation times have to be specified here. <istep> can have the values 0, 1 or 2.

If **istep = 0** enter:

<target_time> 10, 22, 38

4a) **target_time (maxnt)** Values of all nts target or simulation times.

If **istep = 1** enter:

<tmin>, <delta> 0, 5

4b) **tmin, delta** Enter the initial time, <tmin> [T], and the constant size of the time-step, <delta> [T]. All nts target or simulation times are computed according to

<tmin> + n ≅ <delta>.

If **istep = 2** enter:

<tmin>, <delta>, <tinc>, <dtmax> 20, 5, 1.2, 12

4a) **tmin, delta, tinc, dtmax**

Enter the initial time, <tmin> [T], the size of the initial time-step, <delta> [T], the increment of the time-step, <tinc>, and the allowed maximum size of the time-step, <dtmax> [T]. Target or simulation times are computed internally by increasing the size of the time-step by a factor <tinc> until all nts simulation or target times are computed. After n simulation steps the actual time-step size is either <delta> · <tinc>^{n} or <dtmax>. The smaller one is used.

☞ If <TIME_STEP_CONTROL> (group 2) is set to .false., the input for this group is completed. ➔*Proceed to next group*

If <TIME_STEP_CONTROL> (group 2) is set to .true., the program internally calculates simulation times and uses time step sizes that may have to be changed as the simulation progresses, for example in the case of transient flow. Target times only represent times at which an output is desired. The simulation stops when the last target time is reached. ➔*Continue in this group.*

The following input data is required:

<delta>	5
<control_head>	.true.
<dhead_allowed>	0.1

5) **delta**

Initial size of the time step [T] that is internally used.

6) Logical switch **control_head**

Control of head variations between two successive simulation times

TRUE: The variation of the head values between two time steps is calculated to check whether the next time step needs to be longer or shorter etc..

FALSE The head variation is not considered.

7) **dhead_allowed**

The maximum allowed nodal variation of head values between two time steps. If a simulated nodal variation is greater than <dhead_allowed>, the next time step size [T] will be shorter. Otherwise, the next time-step can either be the same or longer.

DATA GROUP 12 Orthomin solver data for flow

<ISOLV_INFO>	.false.
<CHECK_RESIDUAL>	.false.
<ORDER2>	.false.

1) **RESID_ERR, RELAT_ERR, ABSOL_ERR**

The prescribed maximum residual, relative and absolute errors for the iterative flow equation solver. The ORTHOMIN procedure is stopped when one of these criteria is true.

Note: An accurate transport calculation requires an accurate flow simulation.

2) logical variable **ISOLV_INFO** *Extra information about the iterative procedure*

TRUE: More information about the iterative procedure is provided after each ORTHOMIN iteration step. This is useful in case of arising problems.

FALSE (default): No extra information is provided.

3) logical variable **CHECK_RESIDUAL** *Checking of the residual norm*

TRUE: During the ORTHOMIN procedure the residual norm of each element is computed and it is checked if it is smaller than RESID_ERR.

FALSE (default): The residual norm is not computed and it is not checked if it is smaller than RESID_ERR.

4) logical variable **ORDER2** *Second-order factorisation*

TRUE: A second-order factorisation of the flow equation is performed.

FALSE (default): A first-order factorisation, where no extra-bands are added to the original matrix, is performed.

****

☞ At this point the data input for flow is complete. ☜

****

DATA GROUP 13 Control parameters for transport simulation

<XTERMS>	.true.
<CVOLUME>	.true.
<MAXITER>, <TOLEX>	10, 1.0e-5

1) logical switch **XTERMS** *Computation of dispersive cross terms for finite differences.*

TRUE (default): Dispersive cross terms are computed explicitly when a FD (Finite Differences) representation is chosen (group 2).

FALSE: Dispersive cross terms are ignored when a FD representation is chosen.

2) logical switch **CVOLUME** *Use of a control volume approach for finite differences.*

TRUE (default): A control volume approach (as opposed to a standard finite differences approach) is used. No effect when FE (Finite Elements) is selected (group 2).

FALSE: No control volume approach for FD.

4) **MAXITER, TOLEX**:

Maximum number of multi-step iterations for the coupling of physical transport with biochemical and chemical reactions. Convergence tolerance for the iterative process of the transport-chemistry . If no iteration is desired, define a large <TOLEX>.

DATA GROUP 14 Output control parameters for transport

<KRESTARC>, <KWRITHC>, <KPCONC>, <KWRITEDAT>, <KPMASBC> 0, 0, 0, 10, 10

1) **KRESTARC** Variable that specifies whether initial conditions for transport simulation are to be read from a BINARY file. If KRESTARC is equal to 1, concentrations from a previous simulation are used as initial concentrations for the current simulation. The initial concentration values, calculated in a previous simulation, must be in a file called <problem prefix>.cin (unit 16). (Note: This is an input file for the simulation. Therefore, the filename consists of the problem prefix without the letter o appended to it). If KRESTARC is equal to 0, the initial concentration values have to be defined in group 16. This switch is used in combination with KWRITHC.

2) **KWRITHC** Variable that specifies whether concentration values of the last simulated time step are to be written to a BINARY file. If KWRITHC is equal to 1, the concentration values of the last simulated time step will be written in a file named <problem prefix>o.cen (unit 51). If KWRITHC is equal to 0, the concentration values are not written into a file. This switch, in combination with KRESTARC, enables the user to continue a former simulation. If the subsequent simulation is called <new problem prefix>, the generated file, <problem prefix>o.cen, must be renamed in <new problem prefix>.cin, and the switch KRESTARC must be set to 1 in the subsequent simulation.

3) **KPCONC** If KPCONC is equal to 1, nodal concentrations are written into a BINARY file (unit 41) named <problem prefix>o.con. For KPCONC = 0 no output takes place.

4) **KWRITEDAT** If KWRITEDAT is equal to <n>, nodal concentrations for every <n>th time step are written in surfer format to an ASCII file named <n>o.dat (unit 81). For KWRITEDAT = 0 no output takes place.

5) **KPMASBC** Controls output of the mass balance. Output is written into the general output file and into an ASCII file named <n>o.mba (unit 65). The following applies to this switch.

> 0, output at every <KPMASBC>th time step. If TIME_STEP_CONTROL (group 2) is used, output will be at all target times.

= 0, no output of the solute mass balance.

DATA GROUP 15 Transport parameters

A set of transport parameters has to be entered for each zone of the porous medium which was defined in group 5. The zones have to be in the same order as in group 5. If more zones have to be used in order to define dispersivities, set the logical switch <DEFAULT_IC> (described below) to .false..

For each zone of group 5 enter:

<al>, <ath>, <atv>	0.5, 0.05, 0.01
<dstar>	0.0
<bdens>	2.65

1) **al(maxpznc), ath(), atv()** Longitudinal dispersivity [L], transverse dispersivities [L] (horizontal and vertical) in the zone.

2) **dstar()** Effective coefficient of molecular diffusion [L^2/T] in the zone. Note: <dstar> is an effective diffusion coefficient that does not include effects of porosity.

3) **bdens ()** Density of aquifer solids [M/L^3] in the zone. Density is used to compute the retardation factor and the adsorption coefficient of a reactive solute, based on its distribution coefficient.

Enter:

<DEFAULT_IC>	.true.

4) logical switch **DEFAULT_IC** *Distribution of initial dispersivities in the domain.*

TRUE: The initial dispersivities defined above are valid in the specified zone.

FALSE: There are zones with dispersivities different from the values defined above. The additional zones have to be specified:

Enter:

<nzone>	1
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	6, 8, 1, 2, 4.2, 6.7
<alzone>, <atzone>, <atvzone>	0.3, 0.03, 0.006

4a) **nzone** Number of zones with different dispersivity values.

For each zone enter:

4b) **xfrom, xto, yfrom, yto, zfrom, zto**

Extent of the zone in the x-, y- and z-directions. The coordinates must correspond to nodal coordinates [L].

4c) **alzone, atzone, atvzone**

Longitudinal dispersivity, horizontal and vertical transverse dispersivities [L] for the zone delimited by the above defined coordinates.

Enter:

<vmob>, <vbio>, <vmat>	0.25, 0.005, 0.745
<napl_interaction>	.true.

5) **vmob, vbio, vmat** Relative volumes of mobile phase, biophase and matrix

6) logical switch **napl_interaction** *The solubility of a species present in a NAPL (Non Aqueous Phase Liquid) may be influenced by other NAPL constituents. The NAPLs are defined in group 16.*

TRUE: The solubility of the i`th single NAPL constituent ($c_{sat,\ i}$) depends on its molar fraction according to the following equation:

$$c_{sat,i} = c_{sat\_\max,i} \cdot mf_i$$

where $c_{sat\_max,\ i}$ is the solubility of the i`th constituent from a pure phase and mf_i its molar fraction in the mixed phase.

☞ The mass unit has to be mole if NAPL interaction is calculated.

FALSE: NAPL composition has no effect on solubilities of the constituents.

DATA GROUP 16 Initial condition data for transport

In this group all transport species (mobile and immobile), their properties and their initial concentrations are defined. This is only necessary if <KRESTARC> in group 14 is 0.

☞ The unit of mass has to be mole, if NAPL interaction (group 15) or if chemistry is calculated (group 22).

(I) If **KRESTARC = 1** (group 14), initial conditions are read from the file <problem prefix>.cin (unit 16).

➔ *Proceed to next group*

(II) If **KRESTARC = 0** (group 14), initial conditions are not read from a file. It has to be entered here:

<nsp>	3

1) **nsp** Number of transport species.

For each species enter the follwing data:

<spname>	“DOC”
<ioutspec>	1
<phase>, <exspec>, <exchc>, <cmolfac>	1, 0, 0, 1
<kd> or <csat> if required (see definition below)	
<clambda>	0.
<DEFAULT_IC>	.true.
<cinit>	1.0e-6

2) **spname** Name of the species [A20].

3) **ioutspec** Parameter that controls output of species concentrations. If <ioutspec>≠0, the concentration of the species at each observation well (defined in group 10) will be written to the output file <problem prefix>o.cob. If <ioutspec>=0, the species concentration is not written to the file.

4) **phase** This parameter indicates the phase which hosts the species. The following convention applies:

-1: mobile phase, non reactive (tracer);

1: mobile phase, reactive;

2: biophase;

3: matrix;

4: NAPL (Non Aqueous Phase Liquid)

A chemical constituent that is present in more than one phase is treated as several species. For example a constituent that is located in the reactive mobile phase as well as in the biophase is treated as two species when using TBC. For each phase in which the constituent resides all parameters [numbers 2) - 15) in this group] have to be defined.

5) **exspec, exchc**

The exchange between different species can be simulated based on a linear gradient-flux law.

<exspec> is the index number of the exchange species. <exchc> is the exchange coefficient. All exchange processes have to be defined only once.

For example: If DOC in phase 1 (mobile) is the second species and DOC in phase 2 (biophase) is the fifth species in our list of species, it could either be entered:

A) species 2: <exspec>=5 and <exchc>=0.001

species 5: <exspec>=0 and <exchc>=0

or

B) species 2: <exspec>=0 and <exchc>=0

species 5: <exspec>=2 and <exchc>=0.001

In this case both formulations are equivalent. If species in phase 3 (matrix) or phase 4 (NAPL) take part in an exchange process, parameters have to be entered for these species in order to consider the effects of adsorption and maximum solubility.

6) **cmolfac** In TBC concentrations of the various organic constituents have to be provided in mass of organic carbon per given volume. <cmolfac> is then the number of carbon atoms of the respective organic constituent. For instance, <cmolfac> for benzene (C_6H_6) is 6, as one molecule of benzene contains 6 molecules of carbon. It is only used for calculation of NAPL interaction. An arbitrary value can be assigned to <cmolfac> if there is no NAPL interaction.

7) **kd** Distribution coefficient of the species [L^3/M] based on a linear adsorption isotherm. It is only entered for species associated with <phase> 3 (matrix).

8) **csat** Maximum solubility of a species in water [M/L^3]. It is only entered for species associated with <phase> 4 (NAPL).

9) **clambda** First-order decay rate. There is no first order decay if <clambda>=0.

10) logical switch **DEFAULT_IC** *Default initial concentration in the entire domain.*

TRUE: The default initial concentration specified below is valid in the entire domain.

10a) **cinit** Default initial concentration for the domain [M/L^3]. Concentrations refer to the specific volume of the phase where the species is located. Concentrations of NAPL constituents refer to the matrix volume.

➔ *Proceed with next species*

FALSE: There are zones with an initial concentration different from the default initial concentration that is specified below.

10a) **cinit** Default initial concentration for the domain [M/L^3]. Concentrations refer to the specific volume of the phase in which the species is in. Concentrations of NAPL constituents refer to the matrix volume.

Enter:

<FILEINIT>	.true.

10b) logical switch **FILEINIT** *Concentration data is read from a file.*

TRUE: Initial concentration data are read from a file.

For each model layer enter:

<convf>	1.0
<fname>	example.ico

10c) **convf** conversion factor for multiplication of the input data

10d) **fname** Name of the ASCII-file which contains the initial concentration distribution (no predefined name) for a single layer. <convf> and <fname> have to be specified separately for each model layer.

➔ *Proceed with next species*

FALSE: The initial concentrations are specified by the user

Enter:

<nzone>	1
<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	2, 8, 0, 2, 4.2, 6.7
<czone>	1.8e-6

10c) **nzone** Number of zones with a concentration different from the default initial concentration <cinit>.

For each zone the following must be entered:

10d) **xfrom, xto, yfrom, yto, zfrom, zto**

Extension of the zone in the x-, y- and z-directions. The coordinates must correspond to nodal coordinates [L].

10e) **czone** Initial concentrations for the zone delimited by the coordinates defined above.

➔ *Proceed with next species*

DATA GROUP 17 First-type boundary conditions for transport

In this group the first-type boundary conditions for transport (prescribed concentrations) are assigned to one or several zones by specifying the delimiting coordinates or nodal indices. The concentrations for all species in this/these zone(s) have to be pre-defined by the user.

<DIRICHLET_BC> .true.

1) logical switch **DIRICHLET_BC** *First-type boundary conditions (prescribed concentrations) for transport are present in the domain.*

TRUE: There are first-type boundary conditions in the domain.

➔ *Continue in this group.*

FALSE: There are no first-type boundary conditions in the domain.

The input for this group is complete. ➔ *Proceed to next group.*

<USE_COORD>	.true.
<boundary_file>	.true.
<nbc1_zones>	1
<iconbc1>	.false.

2) logical switch **USE_COORD** *Use of nodal coordinates or nodal indices to prescribe first-type conditions for transport*

(I) TRUE: Nodal coordinates are used to define zones with first-type boundary conditions.

Enter:

<boundary_file>	.false.

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: First-type boundary conditions for transport are read from the input file <problem prefix>.1tt.

The data in the input file <problem prefix>.1tt must have the format described below

FALSE: First-type boundary conditions for transport have to be entered into this group.

3) **nbc1_zones** Number of zones with first-type boundary conditions.

For each zone, define the following:

4) Logical switch **iconcbc1 (maxznbc1)**

Specifies whether or not in this zone the initial concentrations (defined in group 16) represent the prescribed concentrations.

(A) TRUE: The initial concentrations represent the prescribed concentrations. Enter the following line:

<ton_bc1>, <toff_bc1> 0, 10

4a) **ton_bc1(maxznbc1, maxsp), toff_bc1()**

Initial time at which the initial concentrations represent the pre-defined values [T].

Final time at which the initial concentrations represent the pre-defined values [T].

Concentrations are set to zero if no concentrations are defined outside this time interval.

(B) FALSE: The initial concentrations for at least one of the mobile species do not represent the pre-defined concentrations. Enter the following:

<nbc1_conc> 2

<conc_bc1>, <ton_bc1>, <toff_bc1> 2.7e-5,0,10

[species #1, interval #1]

7.3e-5,10,20

[species #1, interval #2]

3.6e-4, 0, 14

[species #2,interval #1]

2.7e-5, 14, 20

[species #2, interval #2]

4a) **nbc1_conc (maxznbc1, maxsp)**

Number of time intervals with the pre-defined concentrations of the species.

For each species and for each time interval enter the following:

4b) **conc_bc1(maxnbc1, maxcobc1, maxsp), ton_bc1(), toff_bc1()**

Pre-defined concentration [M/L3] of the species during the time interval.

Start of time interval with pre-defined concentration [T].

End of time interval with pre-defined concentration [T].

The location of the zone has to be entered by specifying the delimiting nodal coordinates:

<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto>	0, 8, 2, 12, 4.2, 6.7

5) **xfrom, xto, yfrom, yto, zfrom, zto**

Extension of the zone in x-, y- and z-direction. The coordinates must correspond to nodal coordinates [L].

(II) FALSE: Nodal indices are used to define zones with first-type boundary conditions.

Enter:

<boundary_file>	.false.

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: First-type boundary conditions for transport are read from the input file <problem prefix>.1tt.

The data in the input file <problem prefix>.1tt must have the format described below

FALSE: First-type boundary conditions for transport have to be entered into this group.

3) **nbc1_zones** Number of zones with first-type boundary conditions.

For each zone, define the following:

4) Logical switch **iconcbc1 (maxznbc1)**

Specifies whether or not in this zone the initial concentrations (defined in group 16) represent the pre-defined concentrations.

(A) TRUE: The initial concentrations represent the pre-defined concentrations.

Enter the following line:

<ton_bc1>, <toff_bc1>	0, 10

4a) **ton_bc1(maxznbc1, maxsp), toff_bc1()**

Initial time at which the initial concentrations represent the pre-defined values [T].

Final time at which the initial concentrations represent the pre-defined values [T].

Concentrations are set to zero if no concentrations are defined outside this time interval.

(B) FALSE: The initial concentrations for at least one of the mobile species do not represent the prescribed concentrations. Enter the following:

<nbc1_conc>	2
<conc_bc1>, <ton_bc1>, <toff_bc1>	2.7e-5, 0, 10
	[species #1, interval #1]
	7.3e-5, 10, 20
	[species #1, interval #2]
	3.6e-4, 0, 14
	[species #2, interval #1]
	2.7e-5, 14, 20
	[species #2, interval #2]

4a) **nbc1_conc (maxznbc1, maxsp)**

Number of time intervals with the pre-defined concentrations of the species.

For each species and for each time interval enter the following:

4b) **conc_bc1(maxnbc1, maxcobc1, maxsp), ton_bc1(), toff_bc1()**

Pre-defined concentration [M/L3] of species during the time interval.

Start of time interval with pre-defined concentrations [T].

End of time interval with pre-defined concentrations [T].

The location of the zone has to be entered by specifying the delimiting nodal indices:

<nbc1>	6
<node_bc1>	5, 6, 7, 105, 106, 107

5) **nbc1(maxznbc1, maxsp)** Number of nodes belonging to the zone.

6) **node_bc1(maxznbc1, maxsp)** Nodal indices of all nodes in the zone.

☞ Concentration values are required for all fixed head inflow boundaries (group 7). If these concentrations are not specified here, TBC automatically assigns a concentration value of zero to all species.

DATA GROUP 18 Third-type boundary conditions for transport

In this group the third-type boundary conditions for transport (prescribed concentrations) are assigned to one or several zones by specifying the delimiting coordinates or elemental indices. The concentrations for all species in this/these zone(s) have to be prescribed by the user.

☞ By default, all boundaries that are not specified as first- or third-type boundaries represent no-flux boundaries for transport. If the third-type condition is chosen and a concentration of zero is specified in this group, the third-type boundary reverts to a second-type boundary (prescribed dispersive flux) for transport.

☞ Note that third-type boundary conditions for transport are only admissible in connection with positive or zero 2nd type fluxes (Data Group 8)

<CAUCHY_BC> .true.

1) logical switch **CAUCHY_BC** *Third-type boundary conditions (prescribed flux) for transport are present in the domain.*

TRUE: There are third-type boundary conditions in the domain.

➔ *Continue in this group.*

FALSE: There are no third-type boundary conditions in the domain.

The input for this group is complete. ➔ *Proceed to next group.*

Enter:

<USE_COORD>	.true.

2) logical switch **USE_COORD** *Use of coordinates or element indices to define third-type boundary conditions for transport.*

(I) TRUE: Coordinates are used to define zones with third-type boundary conditions.

Enter:

<boundary_file>	.false.
<nbc3_zones>	1

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: Third-type boundary conditions for transport are read from the input file <problem prefix>.3tt.

The data in the input file <problem prefix>.3tt must have the format described below

FALSE: Third-type boundary conditions for transport have to be entered into this group.

3) **nbc3_zones** Number of zones with third-type boundary conditions.

For each zone, define the following:

Enter:

<giveflux>

4) Logical variable **giveflux**

Specifies if the fluid flux is computed from elemental darcy fluxes or whether the 2nd type fluxes specified in group 8 are used.

(A) TRUE: The 2nd type fluxes specified in group 8 are used to calculate 3rd type boundary mass fluxes.

(B) FALSE: The incoming fluid flux that is normal to the boundary is computed internally. It is equal to the computed elemental Darcy flux normal to the third-type boundary face.

Enter:

<iface_bc3>	3
<iconcb3>	.false.

5) **iface_bc3()** Element face which the third-type boundary condition is applied to. It is specified according to the convention described in group 8.

6) Logical switch **iconcb3**

Specifies whether third-type boundary conditions are represented by the initial concentrations (group 16) of each of the nodes in the zone.

(A) TRUE: The third-type boundary conditions are represented by the initial concentrations of all mobile species defined in group 16. Enter the following:

<ton_bc3>, <toff_bc3>	0, 10

6a) **ton_bc3(maxznbc3, maxcobc3, maxsp), toff_bc3()**

Initial time at which the initial concentrations represent the third-type concentration values [T] (should be equal to the initial simulation time).

Final time at which the initial concentrations represent the third-type concentration values [T].

Concentrations are set to zero if no concentrations are defined outside this time interval.

(B) FALSE: The initial concentrations for at least one of the mobile species do not represent the prescribed third-type concentration values. Enter the following:

<nbc3_conc>	2
<conc_bc3>, <ton_bc3>, <toff_bc3>	2.7e-5, 0, 10
	[species#1, interval#1]
	7.3e-5, 10, 20
	[species#1, interval#2]
	3.6e-4, 0, 14
	[species #2, interval#1]
	2.7e-5, 14, 20
	[species #2, interval#2]

6a) **nbc3_conc(maxznbc3, maxsp)**

Number of time intervals with prescribed third-type concentration values of the species

For each species and each time interval enter the following:

6b) **conc_bc3(maxnbc3, maxcobc3, maxsp), ton_bc3(), toff_bc3()**

Prescribed concentration [M/L3] of the species during the time interval.

Start of time interval with prescribed concentration [T].

End of time interval with prescribed concentration [T].

The location of the zone has to be entered by specifying the delimiting nodal coordinates:

<xfrom>, <xto>, <yfrom>, <yto>, <zfrom>, <zto> 0, 8, 2, 12, 4.2, 6.7

7) **xfrom, xto, yfrom, yto, zfrom, zto**

Extension of the zone in the x-, y- and z-directions. The coordinates must correspond to nodal coordinates [L].

(II) FALSE: Element indices are used to define zones with third-type boundary conditions

Enter:

<boundary_file>	.false.
<nbc3_zones>	1
<element_wise>	.true.

2a) logical switch **boundary_file** *Boundary conditions are read from a special file or from the standard input file.*

TRUE: Third-type boundary conditions for transport are read from the input file <problem prefix>.3tt.

The data in the input file <problem prefix>.3tt must have the format described below

FALSE: Third-type boundary conditions for transport have to be entered into this group.

3) **nbc3_zones** Number of zones with third-type boundary conditions.

4) logical switch **element_wise** *Boundary conditions are provided in zones or element by element*

(A) TRUE: Third-type boundary conditions for transport are provided in element by element mode

For each element, define the following:

Enter:

<iel_bc3>, <iface_bc3>,<conc_bc3>

5) **iel_bc3()** Indices of the element with third-type boundary conditions

6) **iface_bc3()** Element face which the third-type boundary condition is applied to. It is specified according to the convention described in group 8.

7) conc_bc3(maxnbc3, maxcobc3, maxsp)

Prescribed concentrations [M/L^3] of all species (mobile + immobile). The order has to be that defined in group 16. Input values for immobile species have to be provided, but they are not used by the program (dummy variables)

☞ time intervals with different concentrations are not possible with the element by element input mode

(B) FALSE: Third-type boundary conditions for transport are provided in zones

For each zone, define the following:

Enter:

<giveflux>

5) Logical variable **giveflux**

Specifies if the fluid flux is computed from elemental darcy fluxes or whether the 2nd type fluxes specified in group 8 are used.

(A) TRUE: The 2nd type fluxes specified in group 8 are used to calculate third type boundary mass fluxes.

(B) FALSE: The incoming fluid flux that is normal to the boundary is computed internally. It is equal to the computed elemental Darcy flux normal to the third-type boundary face.

Enter:

<iface_bc3>	3
<iconcb3>	.false.

6) **iface_bc3()** Element face which the third-type boundary condition is applied to. It is

specified according to the convention described in group 8.

7) Logical switch **iconcb3**

Specifies whether third-type boundary conditions are represented by the initial concentrations (group 16) of each of the nodes in the zone.

(A) TRUE: The third-type boundary conditions are represented by the initial concentrations of all mobile species defined in group 16. Enter the following:

<ton_bc3>, <toff_bc3>   0, 10

7) ton_bc3(maxznbc3, maxcobc3, maxsp), toff_bc3()

Initial time at which the initial concentrations represent the third-type concentration values [T] (should be equal to the initial simulation time).

Final time at which the initial concentrations represent the third-type concentration values [T].

Concentrations are set to zero if no concentrations are defined outside this time interval.

(B) FALSE: The initial concentrations for at least one of the mobile species do not represent the prescribed third-type concentration values. Enter the following:

<nbc3_conc>   2

<conc_bc3>, <ton_bc3>, <toff_bc3>

2.7e-5, 0, 10

[species#1, interval#1]

7.3e-5, 10, 20

[species#1, interval#2]

3.6e-4, 0, 14

[species #2, interval#1]

2.7e-5, 14, 20

[species #2, interval#2]

7a) **nbc3_conc(maxznbc3, maxsp)**

Number of time intervals with prescribed third-type concentration values of the species

For each species and each time interval enter the following:

7b) **conc_bc3(maxnbc3, maxcobc3, maxsp), ton_bc3(), toff_bc3()**

Prescribed concentration [M/L3] of the species during the time interval.

Start of time interval with prescribed concentration [T].

End of time interval with prescribed concentration [T].

The location of the zone has to be entered by specifying the respective elements:

<nbc3>	6
<iel_bc3>	5, 6, 7, 105, 106, 107

8) **nbc3** Number of elements belonging to this zone.

9) **iel_bc3** Indices of all elements in the zone

DATA GROUP 19 Concentration at injection wells

<INJWELL>

1) logical switch **INJWELL** *Concentration at injection wells.*

TRUE: Non-zero species concentrations are considered for the injection wells.

➔ *Continue in this group.*

FALSE: There are no injection wells in the domain, or the concentration of all species in the injection water is zero.

The input for this group is completed. ➔ *Proceed to next group.*

Enter:

<ninjwell>	1	
<iwellid>	3	
<ninjspec>	1	
<ninjspnr>	3	
<ninjc>	2	
<cincj>, <toninjc>, <toffinjc>	3.4e-6, 5, 12	[species #1, interval #1]
	8.5e-6, 13, 20	[species #1, interval #2]
<ninjc>	1	
<cincj>, <toninjc>, <toffinjc>	4.0e-7, 10, 20	[species #2]

☞ The flow data for the injection and withdrawal wells was entered in group 9. The numbering system of the wells must be the same when assigning concentrations to the injected water in this group.

2) **ninjwell** Number of injection wells at which species with non-zero concentrations are considered. This number may be smaller than the total number of injection wells in the domain, if clear water is injected in some wells, but it cannot be larger.

For each injection well enter the following:

3) **iwellid(maxwell)** Index number of the well that has to be described (in the same order as in group 9).

4) **ninjspec** Number of species injected at the described well.

For each species enter the following data. (If no data is given for one or more species, the default injection concentration is 0.):

5) **ninjspnr** Index number of species according to their sequence in group 16 where the species were defined. Index numbers

match the order of definition (1=defined first, 2=second etc.)

6) **ninjc(maxwell, maxsp)** Number of time intervals with different injection concentrations for the current species.

For each interval, enter:

7) **cincj(maxwell, maxsp, maxinjc), toninjc(), toffinjc()**

Injection concentration [M/L3].

Initial time at which the injection concentration is applied [T].

Final time at which the injection concentration is applied [T].

DATA GROUP 20 Orthomin solver and time data for transport

<RESID_ERRC>, <RELAT_ERRC>, <ABSOL_ERRC>	1.0e-30, 1.0e-6, 1.0e-25
<ISOLV_INFOC>	.false.

1) **RESID_ERRC, RELAT_ERRC, ABSOL_ERRC**

The prescribed maximum residual, relative and absolute errors for the iterative equation solver of transport. The ORTHOMIN procedure is stopped when any of these criteria is true.

2) Logical variable **ISOLV_INFOC**

Additional information about the iterative procedure

TRUE: Additional information about the iterative procedure is provided after each ORTHOMIN iteration step. This is useful in case of arising problems.

FALSE (default): No additional information is provided.

☞ If a steady-state simulation for flow was chosen in group 11 (switch **STEADY_STATE** is set to **.true.**) time specifications must be provided here.

➔ *Continue in this group.*

If the switch **STEADY_STATE** is set to .false. the input for this group is completed.
➔ *Proceed to next group.*

Enter:

<nts>	100
<istep>	1

3) **nts** Number of time steps. If an automatic time step control is desired (logical switch <TIME_STEP_ CONTROL> in group 2 is true), <nts> refers to the number of target times.

4) **istep** Generation of the time step procedure. If <TIME_STEP_CONTROL> (group 2) is true, heads, concentrations, etc. are calculated at target times. Target times have to be defined here. If <TIME_STEP_CONTROL> (group 2) is false, simulation times have to be specified here. <istep> can have the values 0, 1 or 2.

If **istep = 0** enter:

<target_time> 10, 22, 38

4a) **target_time (maxnt)** Values of all nts target or simulation times.

If **istep = 1** enter:

<tmin>, <delta> 20, 5

4a) **tmin, delta** Enter the initial time, <tmin> [T], and the constant time step, <delta> [T]. All nts target or simulation times are computed according to

<tmin> + n ≅ <delta>.

If **istep = 2** enter:

<tmin>, <delta>, <tinc>, <dtmax> 20, 5, 1.2, 12

4a) **tmin, delta, tinc, dtmax**

Enter the initial time, <tmin> [T], the initial time step, <delta> [T], the increment of the time step, <tinc>, and the allowed maximum size of the time step, <dtmax> [T]. Target or simulation times are computed internally by increasing the size of the time step by a factor <tinc> until all nts simulation or target times are computed. After n simulation steps the actual time step size is either <delta>*<tinc>^{n} or <dtmax>. The smaller one is used.

Continue by entering:

<twc>, <pectol>, <courtol>	1.0, 2.0, 1.0

5) **twc, pectol, courtol**

Time-weighting factor for the transport simulation (0≤ twc ≤ 1; twc = 0 corresponds to an explicit, twc = 1 to an implicit formulation of the transport equation; default: twc = 1)

If the maximum Peclet number <pectol> is exceeded (default: 2.0) warning messages are issued.

If the maximum Courant number <courtol> is exceeded (default: 1.0) warning messages are issued.

If **<STEADY_STATE>** (group 11) and **<TIME_STEP_CONTROL>** (group 2) are **.true.** enter the following:

<deltac>	1.0
<control_conc>	.true.
<dconc_allowed>	0.1

5a) **deltac** Initial size of the time step [T] that is internally used in order to start the simulation.

5b) logical switch **control_conc**

Control of concentration variation between two successive simulation times

TRUE: The variation of the concentration values between two time steps is calculated to check if the next time step can be longer or has to be shorter.

FALSE The concentration variation is not considered.

5c) **dconc_allowed** The maximum allowed nodal variation of concentration values between two time steps. If there is a simulated nodal variation that is greater than <dconc_allowed>, the next time step size [T] will be shorter. Otherwise, the next time-step can either be the same or longer.

Always enter:

<UPSTRVEL>	.true.
<ALMAX>, <BTMAX>, <GAMMAX>	1.0, 1.0, 1.0

6) logical switch **UPSTRVEL** *Upstream weighting of velocities.*

TRUE (default): Upstream-weighting of velocities.

FALSE: No upstream-weighting of velocities. ALMAX, BTMAX and GAMMAX are read but set to zero.

7) **ALMAX, BTMAX, GAMMAX**

Upstream-weighting factors for the x-, y- and z-directions. They range from 0 (no upstream-weighting) to 1 (full upstream-weighting). For mostly convective flow the default is: 1.0, 1.0, 1.0, i.e. full upstream-weighting in all directions.

****

☞ At this point the data input for transport is complete. ☜

****

DATA GROUP 21 Biochemical reactions

<CONVERSION> 1.0e-3

1) **conversion** If only biochemical calculations are performed, either gram or mole can be used as mass unit. In this case define a conversion factor of 1. The units used for microbial parameters are equal to the units used in the transport part of the input file.

If chemistry will be calculated, concentrations MUST be expressed in mol / l. For example if concentrations in the transport part have been expressed in mmol / l (=mol/m^3) and additional chemistry calculations are desired (logical switch **COMPCHEM** in group 22 is set to **.true.**), a conversion factor of 1.0e-3 in this group allows to transfer the concentrations into mol / l. In this case all concentrations defined in data groups 21, 22 and 23 have to be expressed in mol / l.

<K_BIOCHEM> .true.

2) logical switch **K_BIOCHEM** *Simulation of biochemical processes.*

TRUE: Biochemical processes will be considered. ➔ *Continue in this group.*

FALSE: No biochemical processes are simulated.

The input data for this group is completed. ➔ *Proceed to next group.*

Enter:

<dmaxkap> 0.0

<nreac> 2

3) **dmaxkap** Maximum microbial capacity [mol C / unit volume of biophase] per element (sum of microbial groups). If <dmaxcap> = 0, microbial growth is not limited by microbial density.

4) **nreac** Total number of reaction equations that describe microbial growth or decay. Each equation has the form:

$$\frac{dX}{dt} = v\max \cdot X \cdot \prod_{m=1}^{nrmonod} \frac{C_m}{dmonod_m + C_m} \cdot \prod_{i=1}^{nrinhi} \frac{dinhibit_i}{dinhibit_i + C_i}$$

X is the concentration of the microbial species that is growing or decaying, and C_i are the concentrations of species that affect microbial growth. The meaning of <vmax>, <nrmonod> and <nrinhi> as well as <dmonod> and <dinhibit> is explained below.

For each growth or decay reaction enter the following data:

<nr>, <growth>, <vmax>, <nrmonod>, <nrinhi>	3, .true., 1.5, 2, 1
<ispec>, <dmonod>, <dyield>	1, 0.4e-3, -10.0
<ispec>, <dmonod>, <dyield>	2, 0.2e-3, -10.0
<ispec>, <dinhibit>	4, 0.6e-3

5) **nr** Index number of the microbial species that is growing or decaying. Index numbers have to match the order of definition in group 16.

6) logical variable **growth** Specifies whether microbial growth or decay by this reaction takes place

TRUE: Microorganisms are growing by this reaction.

FALSE: Microorganisms are not growing, their mass does not change due to this reaction. Substrates are consumed proportional to dX/dt.

7) **vmax** Maximum growth rate of the microbial species [1/T]. A negative value indicates microbial decay.

8) **nrmonod** Number of Monod-terms affecting microbial growth [int]. For example: if growth of a certain microbial species depends both on oxygen and carbon concentrations, two Monod-terms have to be considered in the growth equation.

9) **nrinhi** Number of inhibition-terms affecting microbial growth [int]. For example: if growth of a certain microbial species is inhibited by the presence of oxygen, one inhibition term has to be considered.

For each Monod-term enter the following data:

10) **ispec** Index number of the species [int] for which a Monod-term or a yield coefficient will be specified. Index numbers have to match the order of definition in group 16.

11) **dmonod** Monod concentration (also called half-velocity concentration) [M/L^3] for the species. If the species does not effect microbial growth and a yield coefficient will be specified (see next item) enter <dmonod> = 0.

12) **dyield** Turnover coefficient for the species. It links microbial growth to species consumption or production. By convention the turnover coefficient is negative if the species is consumed by microbial growth and it is positive if the species is produced. As the turnover coefficient refers to microbial growth, its unit is [M (species)/ M(cell carbon)].

If the species does not reside in the biophase (defined in group 16), the program internally converts the concentrations into mass per unit volume of biophase.

If heterotrophic microorganisms are considered, the turnover coefficient for organic carbon corresponds to the yield coefficient. For other species or other types of microorganisms the value of the turnover coefficient can usually be derived from stoichiometry of the microbially mediated reactions.

In case of microbial decay (i.e. negative growth rates), the turnover coefficient may be used to specify organic carbon return from dead biomass to the pool of bio-available organic carbon in the domain. Notice that the return coefficient has also to be negative (organic carbon is produced while growth rates are negative, i.e. dyield = -1 means that organic carbon from dead biomass is completely returned to the organic carbon pool). Furthermore the respective Monod-concentration should be zero, although it does not effect the results.

For each inhibition-term enter the following data:

13) **ispec** Index number of the species [int] for which an inhibition term will be specified. Index numbers have to match the order of definition in group 16.

14) **dinhibit** Inhibition concentration for this species [M/L^3].

Enter the parameters for the Equation solver (DGEAR):

<com>	****Dgear solver data *****
<meth>, <miter>, <tol>, <hdtmin>,<hdtmax>	2, 1, 1.0e-5, 1.0e-4, 1.0e-3

For further information on the DGEAR solver see (Hindmarsh, 1974)

15) **com** Comment line for the dgear-solver, e.g. ****Dgear solver data *****

16) **meth** Solution method. If <meth> = 1, the Adams method is used. If <meth> = 2 (default), the Stiff method or the backward differentiation formula are used.

17) **miter** Indicator for the iteration method:

If **<miter> = 0**, functional iteration without using partial derivatives is applied.

If **<miter> = 1**, (default) the Chord method with an analytical Jacobian matrix is used.

If **<miter> = 2**, the Chord method with a Jacobian matrix that is calculated internally with a finite difference method is used.

If **<miter> = 3**, the Chord method is used. The Jacobian matrix is replaced by a diagonal matrix approximation based on a directional derivative.

If **<miter> = -1 or -2**, the same method as for 1 or 2, respectively, is applied, but now by using a banded jacobian matrix.

18) **tol** Maximum relative error <tol> must be greater than 0 but in general should not be larger than 1.0e-3.

19) **minfact** The starting time step <hdt> [T] for the DGEAR solver is computed internally according to

$$< hdt > = < \min fact > \cdot < transport\ time\ step >$$

<transport time step> is the time interval used for transport calculations. It is defined in group 11 or 20. <minfact> should be less than 0.01. After each transport time step <hdt> is replaced by <hdt> $\cdot 1.5^n$ with n being the number of the actual transport step.

20) **maxfact** Factor for calculating the maximum <hdt>, i.e. <hdt> $\cdot$ 1.5^n is not allowed to exceed <maxfact> <transport time step>. Setting <maxfact> = <minfact> leads to a constant <hdt> for all transport steps.

****

☞ At this point the data input for biology is complete. ☜

****

DATA GROUP 22 Equilibrium Chemistry

In this group the data for equilibrium chemistry is specified. Before entering the required parameters the user must have worked out a chemical reaction system according to the component principle as described in (Morel, 1983). Two major steps are necessary to specify the reaction system:

- A set of **components** has to be selected. The components consist of one or more chemical species. Components provide mass or charge balance equations for the chemical system. Their values are not changed during equilibrium calculations.
- A **matrix** has to be assembled in which all chemical species are expressed in terms of the components. The link between chemical species and components is given by the law of mass action.

Additionally the link between transport species defined in group 16 and the chemical species has to be specified by the user.

<COMPCHEM>	.true.

1) logical switch **COMPCHEM** *Computation of chemical equilibrium reactions.*

TRUE: Compute chemical equilibrium reactions. ➔ *Continue in this group.*

FALSE: Do not compute chemical equilibrium reactions.

The input for this group is completed. ➔ *Proceed to next group.*

☞ The required species concentrations for chemical calculations are mol / l (M). Therefore the concentrations of the transport species defined in group 16 have also to be provided in mol/l (or e.g. mmol/l). The <conversion> factor specified in group 21 can be used for conversion of e.g. mmol/l to mol/l. However, it cannot be used for conversion of per gram units to molar units, as this would require a specific conversion factor for each of the species.

<grtitle>	***chemical system information***
<nspec>, <nckomp>, <ncequa>	6, 3, 3

2) **grtitle** comment line

3) **ncspec** Number of chemical species [int].

4) **nckomp** Number of components [int].

5) **ncequa** Number of equilibrium reactions [int].

☞ The chemical species defined in this group include the species that take part in chemical equilibrium reactions as well as those species that react kinetically. The kinetic reactions will be defined in group 23. The following relation **must always be valid**:

<ncspec> = <nckomp> + <ncequa> + <nksp>

<nksp> is the number of kinetically reacting species which will be defined in group 23.

Enter information about the chemical species:

<grtitle>	***species information***					
<cname$_1$>, <cname$_2$>,..., <cname$_{ncspec}$>	“H^+”,	“H_2CO_3”,	“HCO_3^-”,	“CO_3^{2-}”,	“$CaCO_3$”,	“Ca^{2+}”
<ioutspec$_1$>, <ioutspec$_2$>,.., <ioutspec$_{ncspec}$>	1,	1,	1,	1,	1,	1
<icharge$_1$>, <icharge$_2$>,..., <icharge$_{ncspec}$>	+1,	0,	-1,	-2,	0,	2
<chemc$_1$>, <chemc$_2$>,..., <chemc$_{ncspec}$>	0.2e-7,	0.7e-4,	0.2e-2,	0.77e-5;	0.1e-1;	0.4e-4

6) **grtitle** Comment line

7) **cname (ncspec)** Names of the <ncspec> chemical species.

8) **ioutspec (ncspec)** Control parameter for the <nspec> chemical species.

If ioutspec()=0: concentrations of the species will not be added to the output file <problem prefix>o.cob.

If ioutspec()=1: concentrations of the species will be added to the output file <problem prefix>o.cob.

9) **icharge (ncspec)** Charge of the <ncspec> chemical species (used for calculating ionic strength). Assign a value of –99 to icharge to identify minerals. For all minerals with icharge = -99, a special input procedure is applied which allows to vary the start values for the Newton-Raphson iteration node by node.

10) **chemc (ncspec)** Initial guess for the concentration values of the <ncspec> chemical species used for the Newton-Raphson iterations [mol/l]. In principle the results of the iteration are independent of the initial guess, but one should be aware that solver problems may arise if the first estimates deviate too largely from the final values. As a rule of

thumb the initial guess should be in the correct order of magnitude. Initial concentrations of zero are not allowed, except for minerals.

If icharge(i) of a species i was assigned a value of –99 (see item 9 above), then the value of chemc(i) is used as a pointer which points at the transport species in group 16 with the number chemc(i). The initial values of this transport species provided in data group 16 is then used as start values of the chemical species i. (This implies that all minerals to be processed in this way **must** appear as transport species in group 16)

The values are used for the first equilibrium calculation only. For all following steps and in a case of simulation restart, the final values of the preceding calculation are taken as initial guess.

Enter information about the components:

<grtitle>	***component information***
<kname$_1$>, <kname$_2$>, ...,<kname$_{nckomp}$>	“TOTH”, “TOTCO$_3$”, “TOTCa”

11) **grtitle** Comment line

12) **kname (nckomp)** Names of the <nckomp> chemical components.

☞ The transport species were defined in group 16. It is, however, not necessary that each chemical species corresponds to one of the transport species. Instead transport species might be linear combinations of chemical species, if they show similar transport properties. The following example illustrates possible relations between transport species, chemical species and components. It describes the reactions of calcite with ground water assuming the dissolution of calcite being a slow kinetic reaction and the dissociation reactions being fast equilibrium reactions.

There are 7 species ins the system:

H^+, OH^-, H_2CO_3, HCO_3^-, CO_3^{2-}, Ca^{2+} and $CaCO_3(s)$.

We consider 3 equilibrium reactions:

$$H_2O \Leftrightarrow H^+ + OH^-$$
$$H_2CO_3 \Leftrightarrow HCO_3^- + H^+$$
$$HCO_3^- \Leftrightarrow CO_3^{2-} + H^+$$

and one kinetic reaction:

$$\frac{dCaCO_3}{dt} = k_{forward} \cdot Ca^{2+} \cdot CO_3^{2-} - k_{backward}$$

As we specify the kinetic reaction in terms of $CaCO_3$ we have to remove this species from the equilibrium system. The remaining 6 species together with the 3 equilibrium reaction equations require the definition of 3 components (remember: # components + # equations = # species).

We select TOTH, TOTH$_2$CO$_3$, and TOTCa for components and assemble the species-component matrix:

species↓\components ➔	**TOTH**	**TOTH$_2$CO$_3$**	**TOTCa**
H^+	1	0	0
OH^-	-1	0	0
H_2CO_3	0	1	0
HCO_3^-	-1	1	0
CO_3^{2-}	-2	1	0
Ca^{2+}	0	0	1

The 3 components provide 3 balance equations:

TOTH = $1\cdot H^+ - 1\cdot OH^- - 1\cdot HCO_3^- - 2\cdot CO_3^{2-}$

TOTCO$_3$ = $1\cdot H_2CO_3 + 1\cdot HCO_3^- + 1\cdot CO_3^{2-}$

TOTCa = $1\cdot Ca^{2+}$

-TOTH represents the alkalinity, and TOTH$_2$CO$_3$ the total inorganic carbon (TIC).

One could now have defined all 7 chemical species (H^+, OH^-, H_2CO_3, HCO_3^-, CO_3^{2-}, Ca^{2+} and CaCO$_3$(s)) as transport species. In order to save storage and computational effort, however, one might for example also use alkalinity and TIC as transport species. By this the number of transport species is reduced from 7 to 4 (alkalinity, TIC, Ca^{2+}, CaCO$_3$(s)).

Enter information about the assembly of the components from transport species (exemplified for the case that chemical species and transport species are **not** identical:

<grtitle>	***Assembly of components from transport species defined in group 16***	
<ianz>	1	;TOTH
<isp>, <fact>	1, 1.0	;alkalinity
<ianz>	1	;TOTH$_2$CO$_3$
<isp>, <fact>	2, 1.0	;TIC
<ianz>	1	;TOTCa

<isp>, <fact>	3, 1.0	;Ca^{2+}

13) **grtitle** comment line

For each component enter the following:

14) **ianz** Number of transport species necessary to assemble the component

For all transport species used for the assembly of the respective component, enter:

15) **isp, fact** Index number of the transport species [int]. Index numbers have to match the order of definition in group 16.

Weighting factor for the transport species according to the component definition.

Note, that in our example $CaCO_3$ reacts kinetically and does not belong to a component.

Enter information about the relation between chemical species and transport species:

<grtitle>	***Relation between chemical and transport species***	
<ianz>	4	
<nr>	1	;alkalinity
<bspeccomp$_1$>,..., <bspeccomp$_{ncspec}$>	1, -1, 0, -1, -2, 0, 0	
<nr>	2	;TIC
<bspeccomp$_1$>,..., <bspeccomp$_{ncspec}$>	0, 0, 1, 1, 1, 0, 0	
<nr>	3	;Ca^{2+}
<bspeccomp$_1$>,..., <bspeccomp$_{ncspec}$>	0, 0, 0, 0, 0, 1, 0	
<nr>	4	;$CaCO_3$
<bspeccomp$_1$>,..., <bspeccomp$_{ncspec}$>	0, 0, 0, 0, 0, 0, 1	

16) **grtitle** comment line

17) **ianz** Total number of transport species that take part in the chemical reactions (equilibrium **and** kinetic).

For each of these transport species, enter:

18) **nr** Index number of transport species [int]. Index numbers have to match the order of definition in group 16.

19)**bspeccomp(nr, ncspec)** coupling factors between transport species and all chemical species. The order of the chemical species must match the order specified in item 7).

That means for our example:

alkalinity= $1\cdot H^{+}-1\cdot OH^{-}+0\cdot H_2CO_3-1\cdot HCO_3^{-}-2\cdot CO_3^{2-}+0\cdot Ca^{2+}+0\cdot CaCO_3$

TIC = $0\cdot H^{+}+0\cdot OH^{-}+1\cdot H_2CO_3+1\cdot HCO_3^{-}+1\cdot CO_3^{2-}+0\cdot Ca^{2+}+0\cdot CaCO_3$

Ca^{2+} = $0\cdot H^{+}+0\cdot OH\text{-}+0\cdot H_2CO_3+0\cdot HCO_3^{-}+0\cdot CO_3^{2}+1\cdot Ca^{2+}+0\cdot CaCO_3$

$CaCO_3$ = $0\cdot H^{+}+0\cdot OH^{-}+0\cdot H_2CO_3+0\cdot HCO_3^{-}+0\cdot CO_3^{2}+0\cdot Ca^{2+}+1\cdot CaCO_3$

Finally the chemical equilibrium reactions have to be specified. Using the mass action law, the general form of a chemical equilibrium reaction becomes:

$$\prod_{i=1}^{\max\_invol}(ac_i\cdot c_i)^{<sc_i>} =< rconst0 >$$

where c is the concentration of the chemical species and ac its activity coefficient which is computed internally using the Davies approximation. <sc> and <rconst0> are described below. The index "max_invol" is the maximum number of species involved in a single reaction. It was **set to 4** in TBC. This value should be sufficient in most cases. If not, the value of 'max_invol' can be changed in the source code (file: chems.f).

In our example we assume the following three reversible equilibrium reactions:

$(H^{+})\cdot(OH^{-})$ = constant1

$(H^{+})\cdot(HCO_3^{-})/(H_2CO_3)$ = constant2

$(H^{+)}\cdot(CO_3^{2-})/(HCO_3^{-})$ = constant3

Enter:

<grtitle> ***equilibrium reactions***

<rconst0>, <dHD0>, <T>, <chem-spec$_1$>, <sc$_1$>,..., <chem-spec$_{maxinvol}$>,<sc$_{maxinvol}$>

-14, 13345, 25, 1, 1, …. 2, 1, 0, 0, 0 , 0

-6.351, 2247, 25, 1, 1, …. 3, -1, 4, 1, 0, , 0

-10.33, 3617, 25, 1, 1, …. 4, -1, 5, 1, 0, , 0

20) **grtitle** Comment line

For each reaction enter

21) **rconst0, dHD0, T**,

Logarithm of the reaction constant

Free enthalpy of the reaction (for temperature correction)

Temperature at which <rconst0> and <dHD0> were measured.

For each species taking part in the specified reaction, enter:

22) **chem-spec, sc** Index number of the chemical species taking part in the specified reaction. Index numbers have to match the order of definition in item 7) of this group.

Exponent of this species in the mass action law.

If a chem-spec “i” with an exponent of 0 is provided, the reaction only occurs, if species “i” is available. Use this option for dissolution / precipitation of minerals.

Enter:

<grtitle>	***groundwater temperature***
<Tground>	12

23) **grtitle** comment line

24) **Tground** Actual groundwater temperature [°C]. This value is needed for the correction of temperature dependent reaction constants. Enter -1 for no correction.

Enter:

<grtitle>	***parameters for the Newton-Raphson equation solver***
<maxits>, <tolf>, <tolx>	50000, 1.0e-8, 1.0e-10

25) **grtitle** comment line

26) **maxits, tolf, tolx** Maximum number of iterations that is allowed for the Newton-Raphson equation solver.

Convergence criterion for the mass action laws and component equations. Its value should be ≤ 1e-8.

Tolerance for concentrations. (≤ 1e-10)

DATA GROUP 23 Kinetic chemistry

Kinetic reactions have to be specified for all chemical species that are specified in group 22 **and** that do not take part in an equilibrium reaction. Both forward and backward reactions can be considered.

Enter:

<COMPKINET> .true.

1) logical switch **COMPKINET** *Computation of kinetic chemistry.*

TRUE: Kinetic reactions are considered. This is only possible if chemical species are defined in group 22

➔ *Continue in this group.*

FALSE: Kinetic reactions are not considered.

Input is completed. ➔ *Proceed to next group.*

Enter:

<grtitle>	***specification of kinetic reactions***
<nkspec>	1
<grtitle>	***

2) **grtitle** ***comment line***

3) **nkspec** Number of kinetic reactions.

4) **grtitle** ***comment line***

The general form of a kinetic reaction in TBC is:

$$\frac{dc}{dt} = rckin1 \prod_{i=1}^{\max\_invol} (ac_i \cdot c_i)^{sc_i} + rckin2 \prod_{i=1}^{\max\_invol} (ac_i \cdot c_i)^{sc_i}$$

where c_i are the concentrations of the i chemical species involved in the kinetic reaction and ac_i are their activity coefficients which are computed internally using the Davies approximation. c is the concentration of the species which the kinetic reaction is specified for. The meaning of <sc>, <rckin1> and <rckin2> is described below. The maximum number of species involved in a single reaction ('max invol'), was **set to 4** in TBC. This value should be sufficient in most cases. If not, the value of 'max invol' can be changed in the source code (file: chems.f).

EXAMPLE:

We assume the dissolution of calcite to be a kinetic reaction which can be described by the following equation:

$$\frac{dCaCO_3}{dt} = 10^6 \cdot \left[Ca^{2+}\right] \cdot \left[CO_3^{2-}\right] - 5 \cdot 10^{-3}$$

Remember that all species involved in this reaction have to be specified in group 22.

For each kinetic reaction the following input is required:

<kkomp>, <kinetspec>	4, 7
<idepend>	7
<rckin1>, <nksp$_1$>, <sc$_1$>,...,<nksp$_{maxinvol}$>, <sc$_{maxinvol}$>	1e+6, 6, 1, 5, 1, 0, 0, 0, 0
<rckin2>, <nksp$_1$>, <sc$_1$>,...,<nksp$_{maxinvol}$>, <sc$_{maxinvol}$>	-5e-3, 0, 0, 0, 0, 0, 0, 0, 0
<stoich$_1$>, <stoich$_2$>,..., <stoich$_{ncspec}$>	0, 0, 0, 0, -1, -1, 0

5) **kkomp** Index number of the species for which the kinetic reaction is defined ($CaCO_3$ in our example) according to the definition of transport species in group 16.

6) **kinetspec** Index number of the species for which the kinetic reaction is defined ($CaCO_3$ in our example) according to the definition of chemical species in group 22.

7) **idepend** If the species for which the kinetic reaction is defined ($CaCO_3$ in our example) is a mineral, two options can be specified by the value of <idepend>:

1. <idepend> = kintespec: The program checks whether the mineral is actually available or not. In our example, the kinetic dissolution of $CaCO_3$ is only calculated if $CaCO_3$ is present at the actual grid node.

2. <idepend> = 0: The kinetic reactions is calculated irrespective of the availability of the mineral. Mineral concentrations below zero can occur due to dissolution.

8) **rckin1** Reaction constant for the forward reaction (<rckin1> should usually be > 0).

For each chemical species involved in the forward reaction equation, enter:

nksp Index number of the chemical species. Index numbers have to match the order of definition in group 22.

sc Exponent of the species concentration with which it appears in the forward reaction equation.

9) **rckin2** Reaction constant for the backward reaction (<rckin2> should usually be < 0). For irreversible reactions, rckin2 = 0.

For each chemical species involved in the backward reaction equation, enter:

nksp Index number of the chemical species. Index numbers have to match the order of definition in group 22.

sc Exponent of the species concentration with which it appears in the backward reaction equation.

10) **stoich (nkspec,ncspec)**

Stoichiometric coefficient that links the concentration change of the species for which the kinetic reaction was build up with the concentration changes of all others species involved in the reaction. <stoich> has to be specified for all reactive species defined in group 22 and in the same order of definition.

Convention: If a species is consumed/produced due to the increase/decrease of the species for which the kinetic reaction was build up, <stoich> is negative. If a species is consumed/produced due to the decrease/increase of the species for which the kinetic reaction was build up, <stoich> is positive. If a reactive species defined in group 22 is not affected by the kinetic reaction <stoich> is 0. <stoich> is always 0 for the species for which the kinetic reaction was build up

EXAMPLE: 1 mol $CaCO_3$ consists of 1 mol Ca^{2+} and 1 mol CO_3^{2-}. Therefore, <stoich> = -1 for Ca^{2+} and CO_3^{2-}. The negative signs reflects that precipitation (production) of $CaCO_3$ results in a decrease in Ca^{2+} and CO_3^{2-} concentrations.

****

☞ At this point the data input for chemistry is completed. ☜

****

****

☞ Congratulations! Data input is completed. ☜

****

Description of the input and output files

Files that are always opened and used:

unit 10: file **tbc.fil** that contains

a) the subdirectory where input data is found and where output data is written to

b) file prefix for the simulation

tbc.fil has to be in the same subdirectory as the program TBC.

unit 55: main input datafile ('prefix.in')

unit 66: main output listing ('prefixo.out')

Files that are opened and used depending on the simulation requirements:

Input files:

unit 11: input of nodal coordinates ('prefix.nod')

unit 12: input of element incidences ('prefix.inc')

unit 15: input file containing heads for a subsequent simulation (binary) ('prefix.hin')

unit 16: input file containing concentrations for subsequent simulation (binary) ('prefix.cin')

unit 25: input file containing elemental hydraulic conductivities from a random field generator (binary) ('prefix.gen')

unit 25 (alternatively): input files (ASCII) containing horizontal and vertical hydraulic conductivities compatible to the Surfer Grid Node Editor format (©Golden Software, no predefined name or extension)

unit 99: input file containing Dirichlet (first-type) boundary conditions for flow (ASCII) (‘prefix.dbc’)

unit 98: input file containing second-type boundary conditions for flow (ASCII) (‘prefix.2tb’)

unit 25: input files (ASCII) containing initial concentrations. Separate files for each species and each layer (no predefined name or extension)

unit 99: input file containing first-type boundary conditions for transport (ASCII) (‘prefix.1tt’).

unit 99: input file containing third-type boundary conditions for transport (ASCII) (‘prefix.3tt’).

Output files:

unit 40: output file containing heads at specified output times (binary) ('prefixo.hds')

unit 41: output file containing concentrations at specified output times (binary) ('prefixo.con) unit 42: output file containing water budgets (binary) ('prefixo.bal'). It is used if a steady-state flow simulation is performed with TBC and if subsequent transport simulations based on this flow simulation will be performed with a separate code.

unit 43: output file containing elemental velocities (binary) ('prefixo.vel').

unit 50: output file containing of heads of the time step to be use for a subsequent simulation (binary) ('prefixo.hen')

unit 51: output file containing concentrations of the time step to be used for a subsequent simulation (binary) ('prefixo.cen')

unit 60: output file containing nodal fluxes at observation wells (ascii) ('prefixo.obs')

unit 63: output file containing concentrations at observation wells (ascii) ('prefixo.cob')

unit 64: output file containing piezometric heads, concentrations of transport species and concentrations of chemical species of all nodes for timestep “n” (ascii) ('<n>.dat')

unit 65: output file containing mass balance calculations for transport species (ascii) ('prefixo.mba')

unit 92: output file containing flux-averaged concentrations at wells (ascii) ('prefixo.wec')

Description of the array sizes defined with parameter statements in file 'tbc.prm'

maxbc2: Maximum number of nodes with second-type boundary faces.

maxinjc: Maximum number of injection time intervals for concentrations.

maxkzn: Maximum number of zones with different hydraulic conductivities. If a random generator is used, maxkzn must be at least equal to the number of elements, otherwise set it equal to maxpzn.

maxnb: Maximum number of connections for a node: 3d blocks (27 for FE, 7 for FD)/ 3d prisms (variable)

maxndc: Maximum number of Dirichlet nodes

maxne: Maximum number of 3d elements

maxnja: Maximum number of nodal connections for the entire grid. The nodal connections for each node, including itself are determined and added up in the set_iaja routine. For example, for a 7-point finite difference template, each node has 7 connections, including itself.

maxnja2: Maximum number of nodal connections for the entire grid for the decomposed matrix. Maxnja2 is equal to maxnja when a first-order decomposition is selected. Otherwise, for a second-order decomposition, its value must be set higher.

maxnln: Maximum number of nodes per 3d element, (6 for triangular prisms, otherwise 8).

maxnn: Maximum number of nodes.

maxnt: Maximum number of target time steps.

maxntcrl: Maximum number of generated time steps.

maxnx: Maximum number of nodes in x-direction.

maxny: Maximum number of nodes in y-direction.

maxnz: Maximum number of nodes in z-direction.

maxobw: Maximum number of observation (passive) wells.

maxobwn: Maximum number of observation well nodes (Sum for all the wells). It is defined as

maxobw · maxnz where assuming that the wells are extended in z-direction.

maxpzn: Maximum number of zones with different properties (or materials).

maxwell: Maximum number of injection / extraction wells.

maxwelle: Maximum number of elements containing well nodes.

maxwelln: Maximum number of well nodes (sum for all the wells). It is automatically set to

maxwell · maxnz assuming that the wells are extended in z-direction.

maxsp: Maximum number of species defined in group 16.

maxreac: Maximum number of microbiological growth reactions.

maxkspec: Maximum number of kinetic reactive chemical species.

maxkreac: Maximum number of kinetic reactions for each kinetic reactive chemical species.

nglmax: Maximum number of differential equations in dgear.

maxcequa: Maximum number of chemical equilibrium reactions.

maxckomp: Maximum number of chemical components.

maxcspec Maximum number of chemical species.

north: Number of ORTHOMIN orthogonalizations before restart (use 4).

Array dimensions for the transport simulation.

☞ Note: If only flow is simulated, the following parameters can all be set equal to 1 to reduce the amount of required storage.

maxcobc1: Maximum number of time intervals with different concentration values for first-type boundary zones.

maxcobc3: Maximum number of time intervals with different concentration values for third-type boundary zones.

maxnbc1: Maximum number of first-type nodes for transport.

maxnbc3: Maximum number of elements having faces with third-type boundary conditions for transport.

maxnec: Maximum number of 3d elements. (=maxne)

maxnnc: Maximum number of nodes. (=maxnn)

maxpznc: Maximum number of zones with different properties (or materials).

maxznbc1: Maximum number of different first-type zones for transport.

maxznbc3: Maximum number of different third-type zones for transport.

Trouble-shooting

Due to the lack of a graphical interface, TBC is not as easy to handle as e.g. most of the modern office software. Study the manual carefully and take a look at the example file. Here is some very brief advice that may help you.

1) Error in the input file - the program doesn't start.

Have a look at the general output file **<problem prefix>o.out**. The input data which TBC has read from the general input file **<problem prefix>.in**, is written with some comments to this file. Compare the output with your input. Usually the output file will end where the error has occurred.

Sometimes the input file is correct but the program still stops issuing an error message concerning problems with array dimensions. In this case, check the general output file **<problem prefix>o.out** and the array sizes in the file **tbc.prm**. It may be that, e.g., you are using more species than allowed. Increase the array size and recompile the program.

2) The program runs but leads to unexpected or obviously erroneous results.

TBC does not check whether the input data makes sense physically! So check your parameters. Another source for erroneous results may be numerical accuracy or stability problems. Try to increase the accuracy of the different equation solvers and to fulfil the different numerical stability and accuracy criteria (e.g. Courant and Peclet number). Especially equilibrium reactions in connection with mineral precipitation or dissolution require very accurate flow and transport simulation results. Sometimes it may be also helpful to compile TBC with debug information and to run a debugger.

References

Chapelle, F.H., 1993. Ground-water microbiology and geochemistry. John Wiley & Sons, Inc., New York, 424 pp.

Hindmarsh, A.C., 1974. GEAR: Ordinary differential equation system solver. UCID-30001, Revision 3, Lawrence Livermore Laboratories, Livermore, Ca.

Kinzelbach, W., 1992. Numerische Methoden zur Modellierung des Transports von Schadstoffen im Grundwasser. Schriftenreihe gwf Wasser Abwasser, Band 21. R. Oldenbourg Verlag, München, 343 pp.

Morel, F.M.M., 1983. Priciples of aquatic chemistry. John Wiley and Sons, New York.

Schäfer, D., Schäfer, W. and Kinzelbach, W., 1998a. Simulation of Reactive Processes Related to Biodegradation in Aquifers 1. Structure of the 3D Reactive Transport Model. J. Contaminant Hydrol. 31(1/2), 167-186

Schäfer, D., Schäfer, W. and Kinzelbach, W., 1998b. Simulation of Reactive Processes Related to Biodegradation in Aquifers 2. Model Application to a Column Study on Organic Carbon Degradation. J. Contaminant Hydrol. 31(1/2), 187-209

Schlegel, H.G., 1985. Allgemeine Mikrobiologie. Georg Thieme Verlag, Stuttgart, 571 pp.

Stumm, W. and Morgan, J.J., 1996. Aquatic chemistry. John Wiley & Sons, Inc., New York, 1022 pp.

Acknowledgements

The development of the program TBC was funded by the DFG (German Research Foundation).