# Contents

1 INTRODUCTION .................................................. 7
  1.1 THE MODELLING PROCESS. MOTIVATION FOR AUTOMATIC CALIBRATION ........................................... 7
  1.2 WHAT CAN BE MODELED WITH TRANSIN-IV ................. 9
  1.3 SCOPE AND ORGANIZATION. HOW TO USE THIS GUIDE. .... 10
  1.4 GETTING STARTED ............................................ 10

2 DIRECT MODEL EQUATIONS ..................................... 12
  2.1 PROGRAM CAPABILITIES ..................................... 12
  2.2 MODEL EQUATIONS ........................................... 13
    2.2.1 Flow equation ............................................. 13
    2.2.2 Parameterization .......................................... 15
    2.2.3 Numerical solution of flow equation ....................... 17
    2.2.4 The transport equation ................................... 22
    2.2.5 Numerical solution of solute transport equation ........ 24
    2.2.6 Non-linear problems solved by TRANSIN .................... 26

3 INVERSE PROBLEM THEORY ..................................... 30
  3.1 INTRODUCTION .............................................. 30
  3.2 CALIBRATION. DEFINING THE OBJECTIVE FUNCTION. ....... 32
    3.2.1 Formulation .............................................. 32
    3.2.2 The maximum likelihood method ......................... 34
  3.3 MINIMIZATION ALGORITHM .................................... 37
    3.3.1 Computation of derivatives ............................... 38
3.3.2 The Levenberg-Marquardt’s methods ........................................... 39
3.3.3 Marquardt parameter updating ...................................................... 40
3.3.4 Convergence criteria ................................................................. 42
3.3.5 ERROR ANALYSIS ................................................................. 45
3.3.6 MODEL SELECTION ............................................................... 46
3.3.7 Model selection criteria ............................................................ 46

3.4 PREDICTIONS. ERROR ANALYSIS ON THE PREDICTIONS .......... 48

3.5 IDENTIFIABILITY, UNIQUENESS AND STABILITY ....................... 49

4 USING TRANSIN ........................................................................... 51
4.1 INTRODUCTION ................................................................. 51
4.2 DISCRETIZATION ................................................................. 51
   4.2.1 Steps for constructing the FE grid ........................................... 52
   4.2.2 General rules for grid construction ....................................... 53
   4.2.3 Time discretization ............................................................. 57
4.3 PARAMETERIZATION. PRIOR ESTIMATION ............................... 62
4.4 VERIFICATION OF DISCRETIZATION ......................................... 67
   4.4.1 Input errors ................................................................. 67
   4.4.2 One, two or quasi-three dimensional grids ............................... 69
   4.4.3 Discretization size ............................................................ 70
   4.4.4 Defining direct problem convergence criteria ........................... 73
   4.4.5 Using changing convergence criteria ..................................... 74
4.5 SPECIFICATION OF OBSERVED DATA ....................................... 76
   4.5.1 General description ........................................................... 76
   4.5.2 Definition of devices and their spatial extension ...................... 76
   4.5.3 Methods for weighted summation over space .......................... 78
   4.5.4 Methods of weighted temporal summation ............................. 80
   4.5.5 Error structure of measurements ......................................... 80
4.6 INPUT FILE PREPARATION ...................................................... 82
4.7 BRIEF DESCRIPTION OF OUTPUT .......................................... 83
4.7.1 Mass balance ........................................ 85

4.8 RESTART PROCEDURE .................................. 86

4.9 DIFFICULTIES WITH TRANSIN .......................... 86

4.9.1 Input errors ........................................ 87
4.9.2 Control of problem size .............................. 88
4.9.3 Run-time errors ..................................... 89
4.9.4 Convergence problems ............................... 90
4.9.5 Direct problem ..................................... 90
4.9.6 Inverse problem ................................... 93
4.9.7 Checklist of stability and convergence issues .... 95

5 EXAMPLES .................................................. 97

5.1 INTRODUCTION .......................................... 97

5.2 EXAMPLE 1. TWO DIMENSIONAL FLOW AND TRANSPORT .... 97

5.2.1 Description of the example .......................... 97
5.2.2 Full description of input files ....................... 98
5.2.3 File DIM (EX1DIM.DAT) .......................... 100
5.2.4 File GRI (EX1GRI.DAT) .......................... 101
5.2.5 PAR File (EX1PAR.DAT) .......................... 103
5.2.6 File TIM (EX1TIM.DAT) .......................... 104
5.2.7 File OBS (EX1OBS.DAT) .......................... 104
5.2.8 File INI (EX1INI.DAT) .......................... 104
5.2.9 Results ............................................ 105
5.2.10 Output files ...................................... 105

5.3 EXAMPLE 2. CONVERGENT FLOW TRANSIENT TEST .... 107

5.3.1 Transforming the flow problem to one dimensional .... 109
5.3.2 Transforming the transport problem to one dimensional .... 110
5.3.3 Boundary conditions and model parameters .......... 110
5.3.4 Description of input files .......................... 111
5.3.5 Results of example 2 ................................ 114
5.3.6 Including matrix diffusion processes ........................................... 115
5.3.7 Matrix hypothesis .............................................................................. 116
5.3.8 Results using the different hypothesis .............................................. 117

5.4 EXAMPLE 1. CONSTANT ONE-DIMENSIONAL DRAINAGE FROM AN UNSATURATED MEDIUM ................................................................. 118

5.4.1 General description: non-linear flow inverse problem (IOFLLI=1,
IOEQT=1, IOINV=1) ............................................................................. 118
5.4.2 Direct problem .................................................................................. 118

5.5 Inverse problem .................................................................................. 119

5.6 Discretization and numerical conditioning description ....................... 119

5.7 Results .................................................................................................. 120

5.8 EXAMPLE 4. TRACER TEST IN A LOW PERMEABILITY, ANISOTROPIC, FRACTURED MEDIA ................................................................. 121

5.8.1 General description. Flow and transport (IOEQT=3), 3-D (IODIM=3), inverse problem (IOINV=1,2) .................................................. 121
5.8.2 Description of the physical medium .................................................. 121

5.9 Inverse problem .................................................................................. 122
5.9.1 Discretization and mathematical conditioning description ............... 123

5.10 Results .................................................................................................. 124

A INPUT DATA DESCRIPTION .................................................................... 132

A.1 INTRODUCTION ................................................................................... 132

A.2 Comments in input files ........................................................................ 132

A.3 FILENAME (1): DIM file ..................................................................... 134

A.3.1 Group A1. NAMES OF INPUT DATA FILES ................................. 134

A.3.2 Group A2. OUTPUT FILENAMES .................................................. 135

A.3.3 Group A3. TITLE AND OPTIONS .................................................. 137

A.3.4 Group A4. DIMENSIONS ................................................................. 139

A.3.5 Group A5. LOGARITHMIC TRANSFORMATION OPTIONS .......... 142

A.3.6 Group A6. OUTPUT OPTIONS ........................................................ 143

A.3.7 Group A7. MASS BALANCE .............................................................. 145
A.6.2 Group D2. TIME FUNCTIONS ........................................ 176
A.6.3 Group D3. TIME WEIGHTING PARAMETERS ..................... 177
A.7 FILENAME (5): OBS File .............................................. 177
  A.7.1 Group E1. GENERAL INFORMATION ON DEVICES AND
                  INTERPOLATION SYSTEMS .............................. 177
  A.7.2 Group E2. DESCRIPTION OF SPATIAL/TEMPORAL
                  INTEGRATION ............................................. 178
  A.7.3 Group E3. BASIC SPATIAL DATA ................................. 179
  A.7.4 Group E4. OBSERVATION INFORMATION ......................... 180
A.8 FILENAME (6): INI File .............................................. 181
  A.8.1 Group F1. INITIAL HEADS ..................................... 181
  A.8.2 Group F2. INITIAL CONCENTRATIONS .......................... 182
  A.8.3 Group F3. FLOW VELOCITY .................................... 182
  A.8.4 Group F4. BOUNDARY FLOW ................................... 183
A.9 CHARACTERISTICS OF THE NON-LINEAR FUNCTIONS .......... 183
  A.9.1 ASYMPTOTIC FREE AQUIFER TRANSMISSIVITY’ ............. 183
  A.9.2 “TWO LAYER STORAGE FUNCTION” ............................. 184
  A.9.3 LINEAR RETARDATION FUNCTION ............................... 185
  A.9.4 FREUNDLICH RETARDATION FUNCTION ........................ 186
  A.9.5 PARALLEL FREE AQUIFER TRANSMISSIVITY ................... 187
  A.9.6 SINK-SOURCE LIMITED CONDITION ........................... 188
  A.9.7 LINEAR RETENTION CURVE .................................. 190
  A.9.8 VAN GENUCHTEN RETENTION CURVE .......................... 191

B TRANSIN-IV VARIABLES ............................................... 193
  B.1 GENERAL RULES FOR VARIABLE NAMES .......................... 193
  B.2 DICTIONARY OF VARIABLES ..................................... 193
Chapter 1

INTRODUCTION

1.1 THE MODELLING PROCESS. MOTIVATION FOR AUTOMATIC CALIBRATION

In order to evaluate the need for automatic parameter estimation and properly assess its role in modeling, it is necessary, first, to set it in its appropriate framework: the modelling process.

The first step in building a numerical model is conceptualization. On the basis of site-specific data and general scientific knowledge, the modeler builds a picture of the real system and expresses it in terms of mathematical equations, which are the foundations of the numerical model. While the conceptual model can often be defined in very simple terms, it represents the basis for any subsequent effort. For a model to have any chance of reproducing reality, it must include the relevant processes and represent them by an adequate model structure. Since one can neither consider jointly all possible processes affecting water flow and solute transport nor represent accurately their spatial variability, one makes simplifying assumptions about both. This is what makes conceptualization difficult and very dependent on the modeler’s views about reality and the nature of evidence. In any case, more than one conceptual model may arise at this stage.

The second step in modelling involves representing the mathematical equations in a solvable form. We will refer to it as discretization, because it usually involves subdividing the domain and defining a solution grid and time increments. However, other tasks are also performed at this stage, including the choice of computer code(s), definition of resolution level for model parameters, etc. Discretization errors can be evaluated by following careful verification methodologies on both the computer code and its application to the specific problem.

Once the conceptual model has been defined, the numerical model should have been left in terms of a (hopefully small) finite number of unknown parameters. Calibration refers to the process of estimating them so as to obtain a good agreement between the computed and measured behavior of the system.
The need to revise the conceptual model or to select one among a set of alternatives arises at this stage. Model selection is usually performed by means of qualitative criteria. For example, models failing to fit available measurements or requiring absurd parameter values can be rejected. A more objective way of achieving the same goal is by means of model selection criteria, which are much more efficient in dealing with overparameterization. In either case, available data may not be sufficient for selecting a single model if they are fitted, with reasonable parameter values, by several conceptual models.

One is forced to plan for additional measurements whenever parameter uncertainties are too large or when widely different models have to be accepted. If these measurements are aimed at reducing parameter uncertainties, then one will be facing a problem of design for parameter estimation. On the other hand, design for model discrimination refers to the problem of defining the experiment and measurements that will allow us to select, among a set of alternative models, the one that best reproduces the real system.

The last step we will be discussing here includes formulation of predictions and evaluation of their uncertainties. This may involve a conceptualization on its own, because the stresses whose response is to be predicted may lead to significant changes in the natural system, so that the model used for calibration may no longer be valid. Hence, new uncertainties often arise when formulating predictions.

In reviewing back the modelling process, three sources of uncertainty can be identified. The first group comprises those associated with the conceptual model, including both the fact that several models may have been satisfactorily calibrated and that calibration assumptions may not be valid during prediction. The second group refers to parameter uncertainties. Finally, uncertainty in future natural or man-induced stresses causes model predictions to be uncertain. These latter uncertainties are external to modelling (although they can affect the model). Therefore, we will ignore this third group of uncertainties.

The discussion of the modeling process can be summarized in Figure 1.1. Inverse modelling, in its most strict sense, only refers to the third stage: calibration. Actually, we will see that TRANSIN can also be useful for performing error analysis, both on estimated parameters and predicted heads, selecting one model among a set of alternatives or planning for additional measurements. In any case, TRANSIN will not provide by itself any hints about the conceptual model. Yet quality of estimated parameters and the reliability of model predictions will be always limited by the validity of the conceptual model.

On the other hand, I contend that one cannot realistically carry out the process described in Figure 1.1 without using a powerful automatic calibration code such as TRANSIN. When performing manual calibration, one is never sure if disagreement between measured and computed heads and or concentrations (and in general concentrations) are caused by erroneous parameters or by an erroneous conceptual model. As a result, one tends to modify both, to neglect some data and to perpetuate erroneous concepts.
In our experience, the process tends to become anarchic and frustrating. On the other hand, our experience with automatic calibration is that it frees the modeler of the burden of having to deal with parameter modifications (sometimes at the cost of very large computer times). This allows him to concentrate on the really important issue, namely, the identification of the most appropriate conceptual model.

1.2 WHAT CAN BE MODELED WITH TRANSIN-IV

Describing TRANSIN-IV capabilities involves both the types of analyses that can be performed and the types of models to which the code can be applied. Among the first capabilities, following is a list of things that can be done:

- Simulation (i.e., standard direct solution of groundwater (linear or non-linear) flow and transport equations).
- Parameter estimation. It is the primary use of the code.
- Error analysis. That is, evaluation of uncertainty on model parameters.
- Sensitivity analysis: evaluation of how heads and concentrations will change in response to changes in model parameters.
- Model selection. That is, identification of the “best” among a set of alternative conceptual models.
- Experiment design: selection of the most informative experiment among a set of alternatives.

In addition, TRANSIN has a rather complete set of output options that facilitates postprocessing of results, both in terms of heads, concentrations and mass balances and both in terms of plane views and time evolutions.

Regarding the types of groundwater problems that can be solved, TRANSIN can solve virtually all types of flow and transport problems. It has been successfully applied to modelling the following types of real problems:

- 1-D laboratory experiments (Carrera et al., 1991)
- 2-D Horizontal aquifers (Samper et al, 1991, Custodio et al., 1993)
- Multilayer aquifers (DIT-DMAMI, 1991)
- Well hydraulics problems (Medina et al., 1994)
- Unsaturated transport (Guimerà, 1992)
- 3-D Double porosity solute transport.
1.3 SCOPE AND ORGANIZATION. HOW TO USE THIS GUIDE.

This document is a user’s guide for program TRANSIN. It is neither a book on Numerical Methods for groundwater flow and transport nor on groundwater modelling, although both topics are profusely used throughout the manual. It is organised in a somewhat standard fashion. Following this introductory chapter, Chapter 2 describes the numerical methods used by TRANSIN for solving the groundwater flow and transport equations. Chapter 3 introduces the theory of inverse modelling, including maximum likelihood formulation of the problem, derivation of objective function and minimization algorithms, model structure identification criteria, etc. Chapter 4 is devoted to the minimum the user should know to obtain a converging run with TRANSIN. Hence, it covers topics which are probably well known to many users (i.e., rules for finite element discretization) and others which are quite specific of TRANSIN (input of head measurements, error structures, parameterization, time functions, etc). Finally, Chapter 5 describes 3 examples covering a wide range of situations. These should be useful both for becoming acquainted with the code and as a reference for the user. Many practical hints are given there on how to approach every type of problem.

Input files description is given in Appendix 1. Following the instructions line by line (or card by card, as we call them there), one can prepare the input file. Input files for the examples presented in Chapter 6 are given in Appendix 4. Summaries of the corresponding output files are shown in Appendix 5. The text is completed with a list of subroutines and a list of variables. The latter can be particularly useful for checking the meaning of variables mentioned in the text.

The way to use this guide depends heavily on the user’s background on the Finite Element Method, on groundwater modelling and on the inverse problem. Ideally, the best would be to read through the whole manual. However, even though we have tried to adopt an informal tone, reading the whole text may become a boring and arduous task. On the other hand, we strongly discourage the user from jumping directly into Appendix 1 (detailed description of input files) to prepare input files, without having, at the very least read Chapter 4 and compared his (her) problem with one of the examples in Chapter 5. Moreover, a good understanding of the output cannot be gained without having a background on inverse problem theory (Chapter 3).

1.4 GETTING STARTED

TRANSIN-IV is written in standard FORTRAN-77. It has been used and checked in several computers as VAX, DEC-ALPHA, HP-735, Alliant series FX, CONVEX. The main body of the program can be compiled in any one of the above computers. There are a few machine-dependent subroutines that account for the consumed CPU time and the date of execution. So, to link the program one has to have these subroutines, as well as the IMSL library version 1.0.
Usually, one has not to compile and link the program, but when the problem dimension is very large (in terms of node points, elements, parameters, observation times, etc), the program dimension cannot be enough. In this cases, the program gives a message as:

PARTITION ERROR
------------------
IRMAX IS 700000 IT SHOULD BE GREATER OR EQUAL THAN 733000
IIMAX IS 55000 IT SHOULD BE GREATER OR EQUAL THAN 52300
ISMAX IS 30 IT SHOULD BE GREATER OR EQUAL THAN 20

GO TO THE BEGINNING OF THE MAIN PROGRAM AND CHANGE THE STATEMENT

PARAMETER (IRMAX=...,IIMAX=...,ISMAX=...)

THEN COMPIL AND LINK THE PROGRAM

In this cases you have to edit the file PRPR.FOR and change the corresponding dimensions. In the example you have to change the value of IRMAX to a value larger or equal than 733000.

After that, compile this file, and link the whole program with the IMSL routines.
Chapter 2

DIRECT MODEL EQUATIONS

2.1 PROGRAM CAPABILITIES

TRANSIN solves the inverse problem for linear and non linear flow and transport equations using linear finite elements for spatial discretization and weighted finite differences for time integration. All model parameters (transmissivity, hydraulic conductivity, storativity, areal recharge, prescribed head and flows at the boundary, internal nodes and leakage parameters, dispersion, diffusion, retardation, decay, porosity and external concentration in sinks or sources) may vary arbitrarily in space and, when appropriate, in time.

This chapter contains a description of the direct problem equations solved by TRANSIN. However, in examining these equations it is convenient to bear in mind TRANSIN capabilities, at least those regarding flow and transport domain dimensions and time regime.

The domains can be:

One dimensional: It can be convenient for simulating one-dimensional problems (e.g., laboratory experiments or higher dimension problems with parallel flow). It can be used for two-dimensional problems with radial symmetry too.

Two dimensional horizontal: It is convenient when variations of state variables in the vertical direction can be neglected. It is often used for aquifers with horizontal size much larger than vertical thickness.

Two dimensional vertical: It is used when vertical variations of state variables are much more significant than lateral variations. It can also be helpful for qualitative descriptions of vertical flow components.

Quasi-three-dimensional: Most often, it consists of several sub-horizontal layers, representing aquifers, connected by 1-D vertical elements that represent aquitards. However, it may also be used for fractured media or, in general, as a simplification of a 3D medium (Figure 2.1).
Three dimensional: Fully 3-D treatment. It is useful when the flow and/or transport are clearly and strongly three-dimensional.

• Two dimensional within a 3-D medium. It is a planar flow (no restrictions exist about position and direction of the plane) contained inside of a 3-D medium. It is useful to model flow and transport through pathways and fractures.

Notice that in one model one can mix with TRANSIN any combination of the problems described before.

which is controlled by variables IOTRS (flow regimes) and IORTS (transport regimes). Steady-state is assumed by setting the related variable to zero. If it is set to 1, then the regime is transient with arbitrary, user-defined, initial conditions. If these conditions are unknown, but one may assume, possibly by shifting backwards the initial time, that they are under steady-state, then the corresponding variable should be set to 2. Equations presented below correspond to this case, but the pure transient or steady-state regimes can be easily derived as particular cases.

2.2 MODEL EQUATIONS

2.2.1 Flow equation

This section contains a description of the generic flow equation solved by TRANSIN. Since a combination of alternatives related both to dimensionality (1D, 2D, 3D) and to the nature of flow (confined, unconfined, unsaturated) results in a particular formulation of the flow equation, we have adopted a compact generic form of the equation:

\[ \frac{\partial \theta}{\partial t} + \nabla \cdot (K \nabla h) + q = 0 \quad \text{on } \Omega \]  

where \( h[L] \) is piezometric head (\( h = p/\gamma + z = \psi + z \); \( p[F/L^2] \) is water pressure, \( \gamma[F/L^3] \) is specific weight, \( \psi \) is pressure head (\([L]\)) and \( z[L] \) is vertical position from a reference level). \( K \) is hydraulic conductivity tensor, \( q[T^{-1}, L/T, L^2/T] \) is an instantaneous recharge per element size (length for 1-D elements, area for 2-D elements and volume for 3-D elements), and \( \theta \) is the volumetric water content, defined as:

\[ \theta = \phi S_w f \]  

where \( \phi \) is porosity, \( S_w \) is the saturation degree and \( f \) is a factor that depends upon the dimensionality and type of the problem, as defined in table 2.1.

It is possible to solve equation (2.1) in terms of pressure heads (\( \psi = \frac{p}{\gamma} \)) employing TRANSIN. In such case, it is required to specify the gravity direction (it may or may not be the same as \( z \) direction).

Initial conditions for (2.1) have to be specified. When piezometric head is used as state variable, it is:

\[ h(x, 0) = h_0(x) \]  

(2.2)
Figure 2.1: Schematic representation of two types of discretization in a 3-D cubic medium. 
(a) divided in tetrahedrons b) divided in right triangle prisme. In the upper boundary is superposed a 2-D medium formed by triangles.

Table 2.1: Definition of factor $f$, in equation 2.2 according to the type and dimension of the problem ($h_b$ is the bottom of the aquifer, $b$ is saturated thickness, and $A$ is cross section).

<table>
<thead>
<tr>
<th>Type of flow</th>
<th>1-D</th>
<th>2-D</th>
<th>3-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confined</td>
<td>$A$</td>
<td>$b$</td>
<td>1</td>
</tr>
<tr>
<td>Unconfined</td>
<td>$A(h - h_b)$</td>
<td>$b(h - h_b)$</td>
<td>$-$</td>
</tr>
<tr>
<td>Unsaturated</td>
<td>$A$</td>
<td>$b$</td>
<td>1</td>
</tr>
</tbody>
</table>
where \( \mathbf{x} \) is the position vector, \( h_0 \) is an arbitrary function, specified by the user (it means, defined point to point), a solution of a previous problem or the steady state. Solution in terms of \( h \) is often required for saturated problems where one doesn’t need to distinguish between matric and gravitatory potential. On the other hand, unsaturated problems are solved in terms of the matric potential (\( \psi \)). Sometimes, it is also used when flow through vertical profiles is modeled. In such cases, initial conditions are given by:

\[
\psi(x, y, 0) = \psi_0(x, y)
\]

where \( \psi_0 \) has an analogous meaning as \( h_0 \).

Boundary conditions can be written as:

\[
(fK \nabla(h))\mathbf{n} = \alpha(H - h) + Q_0 \quad \text{on } \Gamma
\]

where \( \Gamma \) is the boundary of \( \Omega \), \( \mathbf{n} \) is the unit vector normal to \( \Gamma \) and pointing outwards, \( H \) is an external head, \( Q \) is prescribed flow and \( \alpha \) is a coefficient controlling the type of boundary condition; that means: \( \alpha = 0 \) for prescribed flow, \( \alpha = \infty \) for prescribed head and \( \alpha \neq 0, \infty \) for mixed condition, being \( \alpha \) in this case the leakage coefficient. The left-hand-side in (2.4) represents water flux entering (if positive) or leaving (if negative) the medium.

### 2.2.2 Parameterization

Flow equation (2.1) with initial and boundary conditions (2.2), (2.3) and (2.4) defines the flow system. In order to solve this equation, we need the parameters values (hydraulic conductivity, storativity, recharge, etc). In general, these parameters show spatial and temporal variations. One of the problems in groundwater hydrology is how to represent this variability. In practice, we are forced to rely on approximations. There are several methods for approximating parameter variation and many of them have been used in groundwater (see Carrera, 1987, Yeh, 1986 for reviews). The process of expressing physical parameters in terms of model parameters is called parameterization. In most cases, the parameterization can be expressed as:

\[
p(x, t) = \sum_{j=1}^{n} p_j f_j(x, t)
\]

where \( p_j \) are scalars called model parameters and \( f_j(x, t) \) are some interpolation functions. The parameterization procedure differs according to these functions. The most commonly used in groundwater have been:

- Zonation: A partition is made on the system. In every partition’s zone, the function \( f_j(x, t) \) has a predefined variation or it has a constant value (Carrera, 1984).

- Pilot points: This method was first proposed by De Marsily et al. (1984). In this case, the interpolation functions \( f_j(x, t) \) are defined as the kriging coefficients and \( p_j \) are the hypothetical parameters on a finite number of points (pilot points).
Discretization All the discrete values of the parameters are taken as model parameters. It can be seen as a particular case of zonation, when the partition coincides with the elements. This is the most frequently method in geostatistical formulations of the problem.

Other methods include splines, finite elements, etc.

While the actual choice of parameterization can bear important conceptual consequences, equation (2.5) implies that all options are equivalent from an algorithmic standpoint. Without loss of generality, we have adopted zonation because it is appropriate for including the type of geological information usually available. Moreover, zonation contains discretization as a particular case, which facilitates geostatistical treatment of the physical parameters. Some parameters (i.e., recharge, solute source terms, etc) may be time dependent. We consider independently the spatial and temporal variability. A generic parameter $p$ is expressed as the product of a spatial function with a time function:

$$p(x,t) = p(x)f_p(t)$$

(2.6)

where $f_p(t)$ is a piecewise linear function. The function $p(x)$ is piecewise defined (defined in every element). The temporal variation is predefined (i.e. we do not estimate it). In order to avoid stability and identifiability problems as much as possible, it seems necessary to reduce the number of model parameters. For every parameter, $p$, a partition of the finite element mesh is done, resulting in $n_p$ element subsets. Every one of these subsets is called a zone of parameter $p$, and a single real value (more than one for anisotropic conductivities) is assigned to every zone. Every one of these values is called a model parameter, and we will estimate these values. Although these will be the only ones we will estimate, the equation parameters are allowed to vary from one element to another. In short, a specific parameter $p$ may be expressed as:

$$p(x) = \sum_{j=1}^{n} p_{k(j)} \psi^p(x)$$

(2.7)

where $n$ is the number of elements, $k(j)$ is the zone of $p$ in element $j$ and

$$\psi^p_j(x) = \begin{cases} 0 & \text{if } x \notin \text{element } j \\ ct. & \text{if } x \in \text{element } j \end{cases}$$

(2.8)

In some type of problems the parameter $p_{k(j)}$ is a function of the state variable (conductivity for unsaturated media, storativity, etc). This is the particular property of the non linear problems. In general, for those problems, one can write (although, not always) $p_{k(j)}$ as the product of two factors:

$$p_{k(j)} = r_{k(j)} g_k(h)$$

(2.9)

where $r_{k(j)}$ is a constant value within elements belonging to the same zone $k$, which often represents an intrinsic property of the medium (e.g., intrinsic permeability). In turn, $g_k(h)$ is a function of the state variable ($h$) (e.g., relative permeability).

In conclusion, the parameter's variation, in the general case, (spatial, temporal and state variable dependence) is defined as:

$$p(x,t) = \left( \sum_{j=1}^{n} r_{k(j)} g_k(h) \psi^p_j(x) \right) f^p_k(t)$$

(2.10)
where we will estimate the values \( r_1, \ldots, r_{n_p} \) (called from now on zonal parameters). The set of all these values for all the physical parameters (transmissivity, recharge, etc) make up the so called model parameters.

Because of the way in which spatial variability affects the solution, the choice of an appropriate parameterization often becomes critical.

At the end of this chapter, we present a description of all type of non-linear functions, \( g_k \), that can be employed to model with TRANSIN, not only in the flow equation but also for the transport problem.

### 2.2.3 Numerical solution of flow equation

Once we defines the values of the parameters, we can solve the flow equation. In the general case, there is not analytical solution, so we have to employ numerical methods. In the groundwater context, finite differences or finite elements have been extensively used (Pinder and Gray, 1987, Huyakorn, 1984). In our case, we have selected the finite element method because it adapts easily to irregular boundaries. We solve the flow equation using a semidiscretization finite element method (Dautray and Lions, 1988) where the spatial and temporal variations are considered separately. Applying a finite element spatial discretization to the flow equation, an ordinary (sometimes non linear) differential system is obtained. This system is finally solved using finite differences. For the general case, after applying Galerkin’s method to equation (2.1), it results:

\[
A(h)h + D(h)\frac{\partial h}{\partial t} = b(h)
\]  

(2.11)

Where \( A, D \) are matrices depending on the elements’ shape, flow parameters and the state variable \( h \) and \( b \) is a vector depending on areal recharge, boundary parameters and state variable. Equation (2.14) is a ordinary differential equation system with constant coefficients, so, for linear cases, it can be solved analytically (Sahuquillo, 1983) and, for linear and non-linear cases, numerically. We have selected this second option using implicit finite differences with a weight parameter (Dautray and Lions, 1988). After applying this method, the system of equations obtained for the linear case is:

\[
\left( \theta_f A + \frac{1}{\Delta t} D \right) h^{k+1} = \theta_f b^{k+1} + (1 - \theta_f) b^k + (\theta_f - 1) A + \frac{1}{\Delta t} D \right) h^k
\]  

(2.12)

where \( \theta_f \) is the flow weight parameter \( 0 \leq \theta_f \leq 1 \); if \( \theta_f = 0 \) it is an explicit scheme; if \( \theta_f = 1 \) it is implicit; if \( \theta_f = 1/2 \) it is the Crank-Nicolson scheme of second order in time; for the rest of values, the scheme is of order 1, \( h^{k+1} \) is the head array in time \( k + 1 \). This system is solved sequentially starting with \( h_0 \) (initial conditions 2.2).

In non-linear cases, system 2.12 is non-linear and its solution has to be obtained iteratively. For this cases we have adopted the Newton-Raphson method because it offers various advantages when used in the frame of the inverse problem (Galarza and Carrera, 1996) compared with another methods such as Picard iteration.

In order to assemble the non-linear system, an approximation of the functions \( g_k \) (eq. 2.10), associated to matrices \( A \) and \( D \) and vector \( b \), becomes necessary. The procedure
adopted in TRANSIN consists of calculating $g_{k(j)}$ with the average value of $h$ in the element. Furthermore, it is evaluated at time $k + \varepsilon$ different than $k + \theta$ which is used for computing the explicit variable, $h^{k+\theta}$ (Pinder and Gray, 1977) as in 2.12. In summary, the system solved iteratively by the Newton-Raphson method is:

$$f(h) = A^{k+\varepsilon}h^{k+\theta} + \frac{D^{k+\varepsilon}}{\Delta t}(h^{k+1} - h^k) - b^{k+\theta} = 0$$  \hspace{1cm} (2.13)

$$A^{k+\varepsilon} = A(h^{k+\varepsilon})$$  \hspace{1cm} (2.14)

$$h^{k+\theta} = \theta_fh^{k+1} + (1 - \theta_f)h^k$$  \hspace{1cm} (2.15)

The basis of the procedure for solving (2.13) is updating the increment vector $\Delta h^{k+1,l+1}$, based on a previous approximation of $f(h^{k+1,l})$ such as indicated in (2.16):

$$\frac{\partial f(h^{k+1,l})}{\partial h^l}(h^{k+1,l+1} - h^{k+1,l}) = f(h^{k+1,l})$$  \hspace{1cm} (2.16)

System (2.16) is solved successively until convergence, obtaining $h^{k+1}$. For clarifying the structure of TRANSIN, let us point out that the user has to indicate explicitly the linear (eq. 2.12) or non-linear (eq. 2.13 to 2.16) nature of the problem, by means of the variable IOFFLI for flow (IOTRLI for transport). Nevertheless, TRANSIN allows the possibility of solving a linear problem treating it as non-linear (in this case only one Newton-Raphson iteration is needed to compute $h^{k+1}$). Furthermore, one could have problems where only part of its spatial domain is non-linear. In that cases, the non-linear way has to be adopted.

**Updating the state variable at time $k + 1(h^{k+1})$**

The algorithm used for applying the Newton-Raphson method is described below:

1.- Initialice the state variable $h^{k+1,l}$; $l = 1$.
2.- Compute the matrices of the physical problem $A(h)$, $D(h)$, $b(h)$.
3.- Compute the function $f(h)$ and the jacobian $\partial f(h)/\partial h$ (eq. 2.16).
4.- Update the state variable $h^{k+1,l+1} = h^{k+1,l} + \Delta h^l$.
5.- If Newton-Raphson method converges, increase $k$ and repeat the process.
6.- If Newton-Raphson method fails convergence, increases the time step increment $\Delta t$.
7.- If too many reductions of $\Delta t$ stop.
8.- Go to 2.
As it can be seen by examining the algorithm, TRANSIN contains mechanisms for improving the numerical conditions when convergence problems appear. Nevertheless, those problems are not only due to time discretization and in that case (reached the maximum number of time step reductions) the user has to check the conceptual model and the input data.

On the other hand, TRANSIN uses several criteria in order to assume reached convergence or divergence. Thus, convergence is assumed if:

$$|f(h^i_l)|_{i_{max}} < \text{RESIDMAXF}$$

or when the following conditions are simultaneously verified:

$$|\Delta h^i_l|_{i_{max}} < \text{DABSMAXF}$$

$$\frac{|\Delta h^i_l|_{i_{max}}}{|h^{k+1,l+1}_i - h^{k+1}_i|} < \text{DRELMAXF}$$

Where $i_{max}$ is the node to which the criterium is maximun. Parameters DABSMAXF, DRELMAXF and RESIDMAXF have to be defined by the user. Related to divergence, TRANSIN reduces the time increment when:

$$|\Delta h^{l+1}_i|_{i_{max}} > |\Delta h^i_l| ; l > 2$$

Referring to the initialization of the state variable, TRANSIN offers three possibilities according to the value of the variable IOPINITH. One is to extrapolate, at each node, the solution obtained for previous times. That alternative is suitable when the shape of solution is smooth in time, and it is the only one used when only a simulation is done. Also, it is used for times greater than the second solution one. Second method consists of using information related to the solution of the direct problem at previous inverse problem iteration. Based on that information, TRANSIN defines the initial value $h^{k+1,l}_i$ at each node $i$. The third alternative selects the one of the two previously described methods which in the previous time step performed better. Our experience (see Galarza et al., 1995) is that the last one is powerful in numerically difficult problems.

Finally, TRANSIN offers the possibility of relaxing the convergence criteria (RESIDMAX, DABSMAX and DRELMAX) of the direct problem at early iterations of the inverse problem, according to the value of the variable IOCRITRAP. Using that alternative often results in important savings of CPU time.

**Type of finite elements**

TRANSIN can manage some one, two and three dimensional elements. Combining appropriately those elements it can be applied to solve most of the linear and various of the non-linear flow and transport problems. The problem domain can be discretized using any of the elements depicted on Figures 2.3 and 2.4. These elements are one dimensional with linear or quadratic interpolation functions, triangular with linear or quadratic interpolation functions and rectangular with 4 or 8 nodes, thetrahedron with
linear interpolation functions, right triangle prisms with linear and quadratic interpolation functions and planer triangle in a 3-D medium with linear interpolation functions. Some integrals appearing during the application of the finite element method cannot be computed analytically. In this cases, a gaussian quadrature formula is employed. Only the integrals corresponding to elements of type 1, 3, 6, 9 and 10 can be computed analytically.

The user cannot use any of all these elements in a given problem, because some constrains are present. For inverse, non linear and 3-D problems, only types 1, 3, 6, 9, 10 and 11 can be used. This restrictions is in effect for efficient purposes only. The numerical formulae necessary in the other elements make these much more expensive from a computational point of view. For inverse problems, that usually take much more CPU time than simulation, this constraint could make impossible to use the inverse problem with these elements. For this reason, all types of elements can be used for simulation, but only types 1, 3, 6, 9, 10 and 11 can be used for solving the inverse problem. There are
Figure 2.3: Type of elements used in TRANSIN for 3-D domains.

TYPE 9. One gauss integration point

TYPE 10. One gauss integration point

TYPE 11. Six gauss integration points
two options for linear simulation. One can do the simulation computing numerically the integrals except in the case of element types 1, 3, 6, 9, 10 and 11 (card A3.3 in appendix 1, set IOINV equal to zero) and one can do analytically the integrals only for element types 1, 3, 6, 9, 10 and 11 (set IOINV less than zero in card A3.3). The last alternative also allows to solve non-linear simulations. Related to the selection of one or another type of elements, the main selection criteria are the shape of the domain and solution of the problem. In addition, it is important to keep in mind that, due to numerical constrains, in 3-D slender domains, it is better to use prisms than thetrahedrons.

### 2.2.4 The transport equation

Solute transport equation including saturated or unsaturated flow state, diffusion processes, dispersion, advection, first order reactions, retardation and matrix diffusion may be written as:

\[
\theta R \frac{\partial c}{\partial t} = \nabla (D \nabla c) - q \nabla c + q_r (c - c') - \theta \lambda R c
\]

\[
\sigma_m \phi_m D_m \frac{\partial c_m}{\partial z} \bigg|_{z=0} \quad \text{in } \Omega \quad (2.17)
\]

where \(\theta\) is the volumetric water content, \(\phi\) stands for aquifer’s porosity, \(c\) \([M/L^3]\) is solute concentration, \(t\) is time, \(q\) \([L/T]\) is Darcy’s flow, \(c'\) is external solute concentration in volumetrically, areally or lineary distributed sinks or sources, \(\lambda\) \([T^{-1}]\) is first order reactions parameter (e.g., radioactive decay), \(\phi_m\) is matrix porosity, \(D_m\) \([L^2/T]\) is matrix molecular diffusion, \(\sigma_m\) \([L^{-1}]\) is matrix specific surface, \(z\) is the coordinate axis perpendicular to the flow direction and \(R\) is retardation coefficient due to adsorption phenomena:

\[
R = 1 + K_d \rho \frac{1 - \phi}{\phi} \quad (2.18)
\]

where \(K_d\) \([L^3/M]\) is the adsorption equilibrium constant, \(\rho\) is solid bulk density and \(\mathbf{D}\) is dispersion tensor:

\[
\mathbf{D} = \begin{pmatrix}
D_{xx} & D_{xy} & D_{xz} \\
D_{yx} & D_{yy} & D_{yz} \\
D_{zx} & D_{zy} & D_{zz}
\end{pmatrix}
\]

\[
D_{xx} = \theta + \frac{\alpha_L q_x^2 + \alpha_T (q_y^2 + q_z^2)}{|q|} \quad (2.20)
\]

\[
D_{yy} = \theta + \frac{\alpha_T (q_x^2 + q_z^2) + \alpha_L q_y^2}{|q|} \quad (2.21)
\]

\[
D_{zz} = \theta + \frac{\alpha_L q_z^2 + \alpha_T (q_x^2 + q_y^2)}{|q|} \quad (2.22)
\]

\[
D_{xy} = (\alpha_L - \alpha_T) \frac{q_x q_y}{|q|} \quad (2.23)
\]

\[
D_{xz} = (\alpha_L - \alpha_T) \frac{q_x q_z}{|q|} \quad (2.24)
\]
\[ D_{yz} = (\alpha_L - \alpha_T) \frac{q_y q_z}{|q|} \]  \hspace{1cm} (2.25)

where \( D_d \ [L^2/T] \) is molecular diffusion, \( \alpha_L \ [L] \) is longitudinal dispersivity, \( \alpha_T \ [L] \) transversal dispersivity, \( q_x, q_y \) and \( q_z \) are Darcy’s flow components in \( x, y \) and \( z \) directions and \( |q| \) is its length. Darcy flow is computed as:

\[ q = -K \nabla h \]  \hspace{1cm} (2.26)

When flow is in steady state, \( \theta \) is constant and it is equal to \( \phi S_w b \), where \( S_w \) is saturation degree, \( \phi \) is the aquifer’s porosity and \( b \) is equal to 1 in 3D, aquifer thickness in 2D and cross section area in 1D. If flow is transient, \( \theta \) is a function of time. It is computed as:

\[ \theta(t) = \theta(t_0) + \int_{t(t_0)}^{h(t)} \frac{\partial \theta}{\partial h} dh \]

Equation (2.17) is solved with the following initial and boundary conditions. Initial conditions:

\[ c(x, 0) = c_0(x) \quad \forall x \in \Omega \]  \hspace{1cm} (2.27)

Boundary conditions:

Dirichlet:

\[ c = c_e \quad \text{in} \ \Gamma_1 \]  \hspace{1cm} (2.28)

and mixed:

\[ D \nabla cn = \beta (c_e - c) + M \quad \text{in} \ \Gamma_2 \]  \hspace{1cm} (2.29)

where \( \Gamma = \Gamma_1 \cup \Gamma_2 \) is \( \Omega \) contour and

\( \beta = 0 \) and \( M = 0 \) if output contour condition (mass flow equal to zero)

\( \beta = -qn \) on input boundary with external concentration \( c_e \)

\( \beta = 0 \) y \( M \neq 0 \) for input mass boundary conditions.

Matrix diffusion is associated with a diffusion type equation

\[ \phi_m R_m \frac{\partial c_m}{\partial t} = \phi_m D_m \frac{\partial^2 c_m}{\partial z^2} - \lambda \phi_m R_m c_m \begin{cases} 0 < z < b_m & t > 0 \\ 0 \end{cases} \]  \hspace{1cm} (2.30)

where \( R_m \) is matrix retardation coefficient and \( b_m \) is matrix thickness. This equation resembles the transport equation in solid medium including only diffusion and first order reactions processes. In this formulation we have made the hypothesis that matrix diffusion is one-dimensional and takes place in a direction perpendicular to the flow. Between the transport and the matrix diffusion equations we have to impose continuity conditions in the matrix-porous medium interface, i.e.:

\[ c_m(x, 0, t) = c(x, t) \quad \forall t > 0 \ \forall x \in \Omega \]  \hspace{1cm} (2.31)

\[ c_m(x, z, 0) = 0 \quad \forall x \in \Omega, \ 0 < z < b_m \]  \hspace{1cm} (2.32)

and

\[ \frac{\partial c_m}{\partial z} \bigg|_{z=b_m} = 0 \]  \hspace{1cm} (2.33)

This equation is solved using a semianalytical method, approximating the concentration in the matrix with a series expansion (Galarza et al., 1990).

It implies that the matrix medium has not to be discretized because knowing its properties \( b_m, \phi_m, A_m, D_m \) and the boundary conditions (concentration in the mobil water) using the analytical procedure lead to the solution of the matrix equation.
2.2.5 Numerical solution of solute transport equation

The finite element method applied to equation (2.16) as for flow equation leads to:

$$ Ec + F \frac{\partial c}{\partial t} = g $$  \hspace{1cm} (2.34)

Applying finite differences to (2.29) one obtains:

$$ \left( \theta_\tau E + \frac{1}{\Delta t} F \right) c^{k+1} = \theta_\tau g^{k+1} + (1 - \theta_\tau) g^k + \left( (\theta_\tau - 1) E + \frac{1}{\Delta t} F \right) c^k $$  \hspace{1cm} (2.35)

Where $\theta_\tau$ is the solute transport ponderation parameter (similar to $\theta_f$ for flow equation), the superscript stands for the time in which every term is evaluated. For linear problems, the matrices $E$, $F$ and the array $g$ are functions of the transport parameters (porosity, dispersivity, etc) and Darcy’s velocity. The linear system of equations (2.35) has a similar form to (2.12). The main difference is that all arrays in (2.12) are constant in time, whereas matrix $E$ is a function of time through Darcy’s velocity when flow is transient. In this case, $E$ is computed with $q$ evaluated at time $k + \theta_\tau$.

Although the flow equation may be solved independently of the transport one, in order to solve transport, the flow must have been previously computed in order to obtain Darcy’s velocity. Fortunately, they are not really coupled, so we compute first $h^{k+1}$, we use this array to compute Darcy’s velocity and we obtain $c^{k+1}$ later (this process is repeated at every time step).

On the other hand, TRANSIN allows to consider dependence of $R$ upon $c$. It fact leads to a nonlinear system, where matrices $E$ and $F$ are also functions of the non linear parameters ($g_k$ in equation 2.10). In order to solve the system we adopt identical procedure as in the flow equation. That is, full Newton-Raphson method is applied resulting a recurrent system, similar to (2.16), but whose unknown is now $c^{k+1}$.

Most of the difficulties associated to solute transport modeling are related to the stability criteria that must be jointly satisfied by the discretization and model parameters. Stability problems lead to smoothness of the contamination front and violation of the minimum principle. To avoid these problems, grid size and time increment should be controlled. In an one-dimensional mesh, these criteria are:

$$ \text{Courant number } C = \frac{v \Delta t}{\Delta x} \leq 1 $$  \hspace{1cm} (2.36)

$$ \text{Peclet number } P = \frac{v \Delta x}{\text{Dispersivity}} \leq 2 $$  \hspace{1cm} (2.37)

In two-dimensional problems these expressions are applied in the flow direction ($\Delta x$ symbolizes the mesh size in this direction). The size in the orthogonal direction is taken so as to satisfy:

$$ \frac{D_x}{\Delta x^2} \simeq \frac{D_y}{\Delta y^2} $$  \hspace{1cm} (2.38)

We have not adopted special precautions to take care of these criteria, but simply use enough fine discretizations. This is justified because accuracy normally leads to more stringent constrains than stability.
Computation of matrix diffusion term

The term of matrix diffusion in identity (2.16) is a function of the concentration in the rock matrix. The one-dimensional diffusion equation in the rock matrix is (equation 2.30)

\[
\frac{\partial c_m}{\partial t} = \frac{D_m}{R_m} \frac{\partial^2 c_m}{\partial z^2} - \lambda c_m \tag{2.39}
\]

With boundary conditions:

\[
c_m(x, 0, t) = c(x, t) \tag{2.40}
\]

\[
c_m(x, z, 0) = 0 \tag{2.41}
\]

\[
(\frac{\partial c_m}{\partial z})(x, b_m, t) = 0 \tag{2.42}
\]

Equation (2.39) resembles the transport equation (2.17) with only diffusion and first order reaction processes. Parameter \( R_m \) stands for matrix retardation.

We define the following dimensionless parameters

\[
t' = \frac{D_m t}{4R_m b_m^2} \quad \lambda' = \frac{4\lambda R_m b_m^2}{D_m} \quad z' = \frac{z}{2b_m} \tag{2.43}
\]

where \( b_m \) is matrix thickness. Substituting (2.43) into (2.39) gives:

\[
\frac{\partial c_m}{\partial t'} = \frac{\partial^2 c_m}{\partial z'^2} - \lambda' c_m \tag{2.44}
\]

Equation (2.44) has analytical solution (Carslaw and Jaegger, 1969)

\[
c_m = \frac{ch(\sqrt{\lambda'}/2(1 - 2z'))}{ch(\sqrt{\lambda'}/2)} - 4\sum_{1}^{\infty} \frac{\alpha_n}{\alpha_n^2 + \lambda'} e^{-(\alpha_n^2 + \lambda')t'} \sin(\alpha_n z') \tag{2.45}
\]

Where,

\[
\alpha_n = (2n - 1)\pi \tag{2.46}
\]

From equation (2.45), matrix diffusion term on equation (2.17) can be obtained as

\[
\phi_m D_m \frac{\partial c_m}{\partial z} \bigg|_{z=0} = K_m \frac{\partial c}{\partial t} + K'_m \tag{2.47}
\]

expressions for \( K_m \) and \( K'_m \) are obtain by derivation of (2.45)

\[
K_m = 8\phi_m R_m b_m \left\{ \sum_{1}^{\infty} \frac{\alpha_n^2}{(\alpha_n^2 + \lambda')^2} - \sum_{1}^{N} \frac{\alpha_n^2}{(\alpha_n^2 + \lambda')^2} e^{-(\alpha_n^2 + \lambda')\Delta t'} \right\} + \frac{\phi_m D_m}{2b_m} \sqrt{\lambda' \Delta t.th(\sqrt{\lambda'})} \tag{2.48}
\]

\[
K'_m = - \left[ I_o^{k-1} + 4\sum_{1}^{N} I_n^{k-1} e^{-(\alpha_n^2 + \lambda')\Delta t'} \right] \frac{\phi_m D_m}{2b_m} \tag{2.49}
\]
\( N \) = Number of employed terms to approximate the memory function (variable NTDMT in card A4.1).
\( \Delta t \) = Time increment used in the temporal discretization.

\[
I_0^k = I_0^{k-1} + \sqrt{\lambda' h/2} \left( c^k - c^{k-1} \right)
\]

\[
I_n^k = I_n^{k-1} e^{-(\alpha_n^2 + \lambda')\Delta t'} + \frac{\alpha_n^2}{(\alpha_n^2 + \lambda')^2} \left( 1 - e^{-(\alpha_n^2 + \lambda')\Delta t'} \right) \left( \frac{c^k - c^{k-1}}{\Delta t'} \right)
\]

2.2.6 Non-linear problems solved by TRANSIN

This section is devoted to clarify how non-linear problems are treated in TRANSIN. In a non-linear problem, parameters \( p(x, t) \) depend on the state variable, \( h \) for flow problem or \( c \) transport. As a consequence function \( g_k(h) \) equals 1 in a linear flow problem remaining as function of \( h \) if non linearity is present. We consider several types of \( g_k \) functions being, each one, associated to a particular physical flow or transport problem. They are briefly described here. Input data structure for these functions is presented in Appendix 1.

Characteristics of the functions \( g_k(v) \).

A non-linear function \( g_k \) is defined by: a) equation to which it belongs; b) primary physical parameter associated to the equation; c) secondary physical parameter associated (generic parameters) and d) numerical agreement parameters.

- **Equation to which the function belongs.** It may be flow or transport equation. There are not functions belonging simultaneously to both equations.

- **Physical parameter associated to the equation.** It refers to the term of the flow or transport equation which is calculated based on this function. It can be, for instance, hydraulic conductivity, storage term, sorption parameter, etc.

- **Secondary physical parameters associated (generic parameters).** These are parameters used for defining the function \( g_k \). Usually generic parameters have physical significance. This is the reason why TRANSIN includes the possibility of estimating both generic parameters and zonal parameters \( r_k(j) \) (see eq. 2.10).

- **agreement parameters.** Some of the functions \( g_k \) included in TRANSIN contain numerical undesirable properties in part of their domain. Those properties cause frequently convergence difficulties. For this reason we smooth the shape of these functions over then problematic sub-domain by employing spline curves that are defined through the so called agreement parameters.

**Mathematical expression.** It is a relationship between the so called non-linear parameter \( g_k \), the state variable, \( h, \psi \) or \( c \) and the generic parameters. Furthermore, the agreement parameters are used to redefine the relationship along part of the domain.

Finally, it is important to keep in mind that any nonlinear function is associated to a zonal parameter \( r_k(j) \) (see eq. 2.10) which in turn, is valid (or active) in the set of elements.
A synthetic description of functions $g_k$ included in TRANSIN is presented in table 2.2 and further details in section A1.9.
<table>
<thead>
<tr>
<th>Generic name of the function $g_k$</th>
<th>Equation</th>
<th>Term for computing</th>
<th>Form of calculation</th>
<th>Generic Parameters</th>
<th>Zonal parameters, $r_k$</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel transmissivity</td>
<td>Flow</td>
<td>$\nabla T h$</td>
<td>$T_{ij} = g_k r_k$; $r_k = k_{ij}^s$</td>
<td>No</td>
<td>Exist</td>
<td>Conductivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$g_k = h - h_b$; $h &gt; h_b$</td>
<td></td>
<td></td>
<td>Represents saturated thickness. It allows to simulate unconfined flow. It is used only for 2-D and 1-D simulations (Dupuit approx.)</td>
</tr>
<tr>
<td>Asymptotic transmissivity</td>
<td>Flow</td>
<td>$\nabla T h$</td>
<td>$T_{ij} = g_k r_k$; $r_k = k_{ij}^s$</td>
<td>No</td>
<td>Exist</td>
<td>Conductivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$g_k = h - h_b$; $h &gt; h_b$</td>
<td></td>
<td></td>
<td>Represents saturated thickness. It allows to simulate unconfined flow. It is used only for 2-D and 1-D simulations (Dupuit approx.)</td>
</tr>
<tr>
<td>Van Genuchten unsaturated</td>
<td>Flow</td>
<td>$k \nabla (\psi + z)$</td>
<td>$k_{ij} = g_k r_k$; $r_k = k_{ij}^s$</td>
<td>$S_r$, $S_m$</td>
<td>$\lambda$, $\psi_0$</td>
<td>Conductivity</td>
</tr>
<tr>
<td>conductivity</td>
<td></td>
<td></td>
<td>$h = S_e (1 - (1 - S_e^{1/\lambda})^{1/\lambda})^{1/2}$; $S_e &lt; 1 - \delta$</td>
<td></td>
<td></td>
<td>Represents relative permeability in an unsaturated flow. It is coupled with the Van Genuchten retention curve function.</td>
</tr>
<tr>
<td>Van Genuchten retention curve</td>
<td>Flow</td>
<td>$C = S_w S_a + \phi \frac{S_m S_r}{S_m - S_r}$</td>
<td>$C = k_{ij} r_k$; $r_k = S_s$</td>
<td>$S_r$, $S_m$</td>
<td>$\lambda$, $\psi_0$</td>
<td>Storage Coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$g_k = S_w + \phi \frac{S_m S_r}{S_m - S_r}$</td>
<td></td>
<td></td>
<td>Represents the storage capacity of a soil, proposed by Van Genuchten 1980</td>
</tr>
<tr>
<td>River aquifer function</td>
<td>Flow</td>
<td>$Q' = \alpha (h - H) + Q_0$</td>
<td>$Q' = g_k r_k$; $r_k = \alpha$</td>
<td>$Q_b$</td>
<td></td>
<td>Leakage Coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Gamma$ in the boundary</td>
<td>$g_k = (H - h) + \frac{Q_0}{\alpha}$; $h &gt; H - \frac{Q_0}{\alpha}$</td>
<td></td>
<td></td>
<td>Represents a sink-source term in the medium.</td>
</tr>
<tr>
<td>Quadratic retardation function</td>
<td>Transport</td>
<td>$R \phi S_w \frac{Q_0}{\Gamma}$</td>
<td>$R = g_k r_k$; $r_k = R_o$</td>
<td>$P$</td>
<td>Retardation Coefficient</td>
<td>The zoned parameter $R$ represents in this case the liquid density No operative in this version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\lambda R S_w \phi C$</td>
<td>$g_k = (1 - 2\phi)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$P = K_d / R_o$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2.2:** Description of the non-linear functions, $g_k$ (see eq. 2.10), considered for modelling flow and transport with TRANSIN. The meaning of the parameters is as follows (see also Appendix 1):

$T_{ij}$ = Component $ij$ of the transmissivity tensor: for linear problems is a zoned parameter.

$k_{ij}^s$ = Component $ij$ of the saturated conductivity tensor (zoned parameter).

$h_b$ = level of the bottom of the aquifer.

$s_r$ = residual saturation of the medium.

$s_m$ = maximum saturation of the medium.

$\psi_0$ = capillary head.
<table>
<thead>
<tr>
<th>Generic name of the function ( g_k )</th>
<th>Equation</th>
<th>Term for computing</th>
<th>Form of calculation</th>
<th>Generic Parameters</th>
<th>Generic zonal parameters, ( r_k )</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Langmuir retardation function</td>
<td>Transport</td>
<td>( R \delta S_w \frac{\partial c}{\partial t} ) ( \lambda RS_w \phi C )</td>
<td>( R = g_k r_k ); ( r_k = R_o ) ( g_k = \frac{P C}{(1 + PC)^2} + \frac{1}{1 + PC} ) ( P = K_d / R_o )</td>
<td>P</td>
<td>Retardation Coefficient</td>
<td>The zoned parameter ( R ) represents in this case the liquid density. No operative in this version.</td>
</tr>
<tr>
<td>Freundlich retardation function</td>
<td>Transport</td>
<td>( R \delta S_w \frac{\partial c}{\partial t} ) ( \lambda RS_w \phi C )</td>
<td>( R = g_k r_k ); ( r_k = R_o ) ( g_k = \frac{1}{P C (1/\rho) - 1} )</td>
<td>P</td>
<td>Retardation Coefficient</td>
<td>The zoned parameter ( R ) represents in this case the liquid density. No operative in this version.</td>
</tr>
<tr>
<td>Two-layer Storage function</td>
<td>Flow</td>
<td>( S \frac{\partial h}{\partial t} )</td>
<td>( S = r_k g_k ); ( r_k = S_s ) ( g_k = h - h_b ); ( h_b + \delta \leq h ) ( g_k = S_m ); ( h_b - \delta \geq h )</td>
<td>No Contains</td>
<td>Storage Coefficient</td>
<td>Allows to simulate a strong change in the storage coefficient when heads fall. Is often coupled to parallel or asymptotic transmissivity functions.</td>
</tr>
</tbody>
</table>

Table 2.2: (Cont.)
Chapter 3

INVERSE PROBLEM THEORY

3.1 INTRODUCTION

The inverse problem consists of estimating model parameters from measurements of the system response and appropriately weighted prior information on the parameters. While modeling in general, and inverse modelling in particular, are often viewed as simply running a computer program, the problem at hand is somewhat more complex. The problem should be viewed under a wider perspective since it involves several interacting steps (data acquisition and treatment, conceptualization, discretization, etc) of which estimation itself is not necessarily the most important nor the most time consuming. It is convenient to distinguish among the following concepts:

- Process Identification (PI): Selection of the physico (chemical) processes, and corresponding equations, to be included in the numerical model, which are considered relevant to the simulated output (Examples: Darcy’s flow, recharge mechanisms, etc.).

- Model Structure Identification (MSI): Definition of model geometry and variation patterns of the physical parameters (Examples: zones over which hydraulic conductivity is assumed constant; boundaries geometry and types; location of sources, etc.).

- Parameter Estimation (often referred to as model identification): Quantification of all model parameters. This implies assigning numerical values to the variables that define the physical parameters. Recall Section 2.2.

Admittedly, the separation of these three concepts is somewhat arbitrary. For example, the location of a boundary between two formations with different transmissivities may be a MSI issue or a parameter estimation issue. Recharge description may be a process identification or a parameter estimation issue (recharge rate can be estimated on the basis of available data and neglected if estimated value is very small). In fact, these simple examples point out that the three concepts depend on the modelling approach.
In general, as the level of detail of the approach increases, the three concepts become increasingly blended and the need to distinguish among them is reduced.

Traditionally, the inverse problem has been defined as that of estimating the model parameters from output measurements. Therefore, the inverse problem can be equated to parameter estimation. The other phases are often referred to as model conceptualization.

In our framework, process identification is usually not an issue, because both Darcy’s law and the continuity and Richard’s equations are widely accepted for groundwater’s flow. Also, the most important processes affecting the solute movement (advection, dispersion, sorption, etc) are relatively well know. Therefore, ignoring a process relevant to the study will be caused by misjudgements. However, storage and recharge mechanisms, as well as head dependence of aquifer parameters and flow behavior at the boundaries may require early conceptualization efforts. In transport, considering or not matrix diffusion and its implementation may be also a difficult task. In any case, these issues should be settled before formulating parameter estimation. Actually, one may attempt calibrating different conceptual models and select the one leading to the best results (see Section 3.5). However, hard data (quantitative) are rarely sufficient for unequivocal model selection, so that both process and model structure should be defined on the basis of qualitative data whenever possible. Having described adequately the physical mechanisms, model structure identification becomes the critical issue. In our context, defining the model structure can be equated to describing the physical system in terms of model parameters. One of the most important steps in the process is parameterization, which is described next.

Parameterization consists of expressing physical parameters as function of a, preferably small, number of model parameters. In our approach, parameterization is done by zonation. This involves defining a set of mutually exclusive subregions, comprising several elements (or nodes), in which each physical parameter is assumed to be constant or to vary in a prescribed manner.

One of the problems in parameter estimation is that many different conceptualizations may lead to a similar model performance (performance criteria will be defined in the next section). Furthermore, if the model structure is incorrect, its parameters may bear no relationship to their physical counterparts. Thus, a good match between measured and computed responses does not ensure that the prediction capabilities will be good, (see example 3 in section 5.4).

Neither precise rules nor a formal approach can be set for defining the conceptual model. As a matter of fact, one might argue that classical hydrogeology is oriented towards such objective. The hydrologist must incorporate information coming from many different fields (geochemistry, lithology, structural geology, sedimentology, edaphology, botanics, etc.) to create an internally coherent description of the system. Quantitative consistency will be tested through inverse modelling, but it is clear that this cannot achieve meaningful results if first of all the qualitative picture is not good. Admittedly, the level of detail required for inverse modelling, which involves geometry and types of boundaries and zonation patterns, is a good deal higher than what hydrogeologist are used to. Therefore, one must leave some room for uncertainty in the precise definition of the conceptual model. In what follows, we shall assume that such definition has been given.
and we will describe the statistical framework of TRANSIN (Section 3.2), optimization algorithms (Section 3.3) and error analysis (Section 3.4). Uncertainty in conceptual modelling is recognized when dealing with model selection (Section 3.5), prediction errors (Section 3.6) and instability problems (Section 3.7).

3.2 CALIBRATION. DEFINING THE OBJECTIVE FUNCTION.

The ideal statement for calibration would be to find the set of parameters that make the solution of the model equal to the true heads and/or concentrations. This ideal statement can never be achieved, due to some related errors that we can not know separately

1. The difference between the exact solution of flow and transport equations and true heads and concentrations. This difference comes from wrong parameters and erroneous conceptual model, in addition to the fact that flow and solute transport equations (2.1 and 2.17) are derived from physical system simplifications.

2. Differences between the exact solution of flow and transport equations and the numerical solution. In most of the cases no analytical solution is available and the equations must be solved numerically. This error is known as numerical error and is related to discretization, the numerical procedure and round-off errors.

3. The difference between true heads and concentrations at some specific points and times and the measured ones. Measurement errors are caused by wrong data interpretation, inappropriate measures, etc.

None of these errors can be known individually, but the sum of all them together, that is, the difference between computed and measured data. For this reason, although the right definition should be made as the difference between true value and exact solution, one has to be satisfied posing the problem in terms of differences between measured and computed data (sometimes called residuals). In the following, we will symbolize $z$ for computed values ($z$ is composed of both head and concentrations in the coupled case plus computed parameters) and $z^*$ for measured values (measured heads and concentrations and parameters’ prior information). We use the expression parameters prior information instead of measured parameters, because we rarely have real parameter measures, but rather indirect estimations of their value, often based on qualitative geological assessments.

3.2.1 Formulation

In order to define a good agreement between measured and computed values, one has to define mathematically what does “good agreement” mean. The usual way consists in defining an appropriate distance function between $z$ and $z^*$, and trying to minimize such distance. Many distances have been used in the context of groundwater hydrology (Yeh, 1986, Carrera, 1987). The methods could be divided in two groups: those which define a
distance without any additional consideration about the data and those which make use of a statistical framework (although they might look different in its conceptualization, most times they lead to similar algorithms). The most frequently used are:

1. Least squares. Even though a lot of formulations are not directly posed as a least squares problem, the final objective function resembles the one of least squares. The objective function to minimize is

\[ F = (z - z^*)^T(z - z^*) \]  

(2.0)

2. Generalized least squares. This method resembles the previous one, but it includes the possibility that measurements at different locations and times may have different levels of confidence, and that errors may be correlated. In this method every point has a weight related to its confidence’s degree. In this case the function is written as

\[ F = (z - z^*)^T W (z - z^*) \]  

(2.1)

where \( W \) is a weight symmetric positive definite matrix.

3. The maximum likelihood method (M.L.). This method consists in maximizing the probability of observing the measured data with respect to the parameters (i.e. maximizing the parameters likelihood). In contrast to the previous methods, one has to select a statistical model in order to be able to apply the method. In this approach, one maximizes the parameters likelihood (defined as proportional to the probability of observing the results being the parameters fixed) given the observed data. In this method, the parameters are fixed but unknown due to lack of knowledge about the physical system. We will use this method in the present work and it will be explained in more detail in subsequent sections.

4. Maximum a posteriori probability. It consists of maximizing the probability of the parameters given the observations. As in the previous method, it is necessary to define a statistical model in order to apply this one. In most cases, both methods lead to similar algorithms.

5. Some other methods less used in groundwater inverse problem are the method of moments, bayesian looser estimates, etc.

6. Methods (1) and (2), as well as (3) and (4) under the hypothesis of parameters gaussian distribution function, can be viewed as particular cases of a norm in \( \mathbb{R}^n \), more specifically, a \(||.||_2\) norm. Other authors have used different norms, for example \(||.||_1\) (Woodbury and Smith, 1988, Xiang et al., 1993):

\[ F = \sum_{i=1}^{n} |z_i - z^*_i| \]  

(2.2)

Each norm has advantages and disadvantages, \(||.||_2\) leads to giving excessive weight to data with large residuals, \(||.||_1\) is not differentiable at the origin, etc.

The first two groups of methods, (1) and (2), may be obtained as a particular case of ML when a normal distribution is assumed for head and concentration errors, as we will see
in the next section. The fact that ML and MAP need to establish a statistical framework, would seem a handicap with respect to least squares, for example. However, they include in a natural way the possibility of accounting for information about the estimated parameters. This information is particularly useful when dealing with uncertainty of estimates.

We have selected ML for three reasons: Prior information about parameters may be introduced quite easily (Edwards, 1972). It has been used successfully for flow parameters estimation (Carrera, 1984). ML leads naturally to criteria for identifying model structure (Carrera, 1984). As we will see below, ML may be thought as a \( \| \cdot \|_2 \) in \( \mathbb{R}^2 \). Its disadvantage is associated with data having a large residual, but its weight may be reduced considering large variances on those data. The maximum likelihood method is introduced in the next section.

### 3.2.2 The maximum likelihood method

**Definition of likelihood.** Let \( R \) be an observation set (head and concentration measurements) and let \( H \) be a hypothesis about the physical system (model parameters and flow and transport equations). Let \( f(R/H) \) be the probability density function of observing the results \( R \) given the hypothesis \( H \) according to a given statistical model. The likelihood of \( H \) given data \( R \) is defined as \( L(H/R) = k f(R/H) \), where \( k \) is an arbitrary constant. It should be noticed that the constant \( k \) takes only one value for all hypotheses given the same data and under the framework of the same statistical model.

The likelihood ratio between two hypotheses against some observations is defined as the ratio between likelihoods. It is noted as \( L(H_1, H_2/R) = \frac{L(H_1/R)}{L(H_2/R)} \).

The maximum likelihood method consists of finding the parameters that maximize the value \( L(H/R) \). The basis of this method stands on the likelihood axiom (Edwards, 1972): “Within the framework of a statistical model, all the information which the data provide concerning the relative merits of two hypothesis is contained in the likelihood ratio of those two hypotheses on the data, and the likelihood ratio is to be interpreted as the degree to which the data supports the one hypothesis against the other.”

As we have noted previously, one statistical model has to be defined in order to apply maximum likelihood theory. For this reason, additional hypotheses about data probability density function must be made, because in most of the cases we cannot know it exactly. The estimation criteria is based on the residuals, i.e., the difference between computed and measured values. The error structure being in general unknown. The only information that we can have about this, consists of a little idea about measurements error structure. Error sources belong to one of the following three groups: Conceptual, computational and measurement errors. Conceptual errors are due to erroneous simplifications about the most important physical phenomena (improper boundary conditions, erroneous analysis of the relevant processes, etc) and to parameters heterogeneity (the zones may not represent spatial variability accurately). Computational errors are due to the numerical algorithm. Their magnitude is associated with the spatial and temporal discretizations and parameter’s zonation. Measurements errors come from incorrect interpretation of the
data and from accuracy of measurement tools. In conclusion, the large number of error sources that are present may justify using the central limit theorem, i.e., assuming that errors follow a gaussian distribution. This will be the hypothesis to conform our statistical framework in which we will search the maximum likelihood parameters.

Maximum likelihood estimators have some nice properties. Let $\phi$ be a one to one transformation in the parameters’ space. The likelihood ratio between two different hypothesis is transformed by directly substituting $p$ by $\phi(p)$ when the set of parameters $p$ is transformed into a new one $\phi(p)$, i.e., if the maximum likelihood estimator of $p$ is $\hat{p}$, then the estimator for $\phi(p)$ is $\phi(\hat{p})$ (Edwards, 1972). This property allows us to do an appropriate transformation in order to convert a parameter’s distribution into a gaussian one for parameters not following a gaussian distribution. Transmissivities are the clearest example. They are usually assumed to be lognormally distributed. So, if instead of working with transmissivities we work with its logarithm, then the transformed parameters will have a gaussian distribution. Once we have obtained the maximum likelihood estimate for log-transmissivities, taking its anti-logarithm we obtain the transmissivity estimate.

With the hypothesis of residuals and prior information normal distribution, along with the independence of heads, concentrations and prior information, the likelihood function may be written as:

$$L = (2\pi)^{-\frac{n}{2}} (|C_h||C_c||C_p|)^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} \left( (h - h)^t C_h^{-1} (h - h) + (c - c)^t C_c^{-1} (c - c) + (p - p)^t C_p^{-1} (p - p) \right) \right]$$

(2.3)

where $h$ and $c$ are computed head and concentration arrays, respectively, $h^*$ and $c^*$ are measured head and concentration arrays, $p$ is the computed parameters array, $p^*$ is their prior information array, and $C_h$, $C_c$ and $C_p$ are heads, concentrations and parameters covariance matrices respectively (if different types of parameters have no correlation, the last matrix has a block diagonal structure, one block for every parameter type, being this the usual hypotheses) and $N$ is the total number of data (measured heads + measured concentrations + parameters prior information).

Usually, it is less difficult to know relative values of covariances than covariances themselves (i.e., it is easier to decide whether or not a measurement is more reliable than another, than assigning an absolute number to every measurement). For this reason, following Neuman and Yakowitz (1979), we will suppose that there exist some unknown scalars, $\tau_h, \tau_c, \tau_i \in \mathbb{R}, i=1, \ldots, n_t$ (number of parameters type, i=1 transmissivities, i=2 storage, etc) such that

$$C_h = \tau_h V_h, \quad C_c = \tau_c V_c, \quad C_i = \tau_i V_i$$

(2.4)

where $V_h$, $V_c$ and $V_i$ are known positive definite symmetric matrices. In equation (3.5) we are implicitly assuming that different parameter’s type have no correlation. In practice, $V_h$, $V_c$ and $V_i$ represent our best guess of the covariance matrices. In fact, (3.5) implies that the covariance structure is perfectly known. If $V_h$, $V_c$ and $V_i$ are indeed exact, then $\tau_h$, $\tau_c$ and $\tau_i$ would equal 1.

The maximum likelihood estimator consists of looking for the parameters that maximize (3.4). Instead of using (3.4), it is often taken the natural logarithm in both
sides of this equation leading to:

\[ S = \tau_h^{-1}F_h + \tau_c^{-1}F_c + \sum_i \tau_i^{-1}F_i + n_h ln(\tau_h) + n_c ln(\tau_c) + \sum_i n_i ln(\tau_i) + \]

\[ + ln|V_h| + ln|V_c| + \sum_i ln|V_i| + N ln(2\pi) \]  

(2.5)

in this expression, \(F_h = (h - h^*)V_h^{-1}(h - h^*)\), \(F_c = (c - c^*)V_c^{-1}(c - c^*)\) and \(F_i = (p_i - p_i^*)V_i^{-1}(p_i - p_i^*)\), \(F_h\) and \(F_c\) are the contribution of heads and concentrations to objective function and \(n_i\) the number of data of parameters type \(i\), \(n_h\) is the number of data heads and \(n_c\) the one for concentrations. It should be noted that in addition to equation parameters (model parameters), those that we want to estimate, in equation (3.6) appear some more unknown parameters (\(\tau_h, \tau_c, \tau_i\) in whose estimation we may be interested (even in the general case, \(V_h, V_c\) and \(V_p\) might be functions of some additional parameters, but those are not included in the code).

If we have a perfect knowledge of model statistical structure, some terms in equation (3.6) take constant values, so they can be excluded of the function to minimize. In addition, if parameters \(\tau_h, \tau_c\) and \(\tau_i\) are known, then minimization of (3.5) can be reduced to minimize (multiplying 3.6 by \(\tau_h\)):

\[ J = F_h + \lambda_c F_c + \sum_{i=1}^{n_t} \lambda_i F_i \]  

(2.6)

where \(\lambda_c = \frac{\tau_c}{\tau_h}\), \(\lambda_i = \frac{\tau_i}{\tau_h}\). Program TRANSIN minimizes \(J\) with respect to model parameters representing aquifer properties. In order to estimate parameters describing the error structure, it is necessary to perform several runs with different statistical parameters and choose the one that leads to the minimum \(S\) (3.6). Carrera and Neuman (1986a) show on a synthetic example that the parameters computed with the \(\lambda_i\) that minimize \(S\) are better than those computed with any other \(\lambda_i\).

In many cases, the relatively high computer cost of each run may discourage any attempt of estimating statistical parameters. In such cases, depending on the way \(\tau_h\), \(\tau_c\) and \(\tau_i\) are computed, \(S\) can be evaluated in different manners, each one having a different interpretation.

- First, assume that prior error structure is correct. This would imply \(\tau_h = \tau_c = \tau_i = 1\) in (3.6). Let \(S_1\) be the value of \(S\) (3.6) obtained with this assumption:

\[ S_1 = F_h + F_c + \sum_i F_i + \ln|V_h| + \ln|V_c| + \sum_i \ln|V_i| + N \ln(2\pi) \]  

(2.7)

- Second, assume that \(\lambda\) coefficients are correct. It can be shown that the maximum likelihood estimates of \(\tau_h, \tau_c\) and \(\tau_i\) are \(\tau_h = J/N, \tau_c = \tau_h/\lambda_c\) and \(\tau_i = \tau_h/\lambda_i\). Let \(S_2\) be the value of \(S\) (3.6) obtained with this assumption:

\[ S_2 = N + \ln|V_h| + \ln|V| + \sum_i \ln|V_i| + N \ln \left(\frac{J}{N}\right) - \]

\[ - n_c \ln(\lambda_c) - \sum_i n_i \ln(\lambda_i) + N \ln(2\pi) \]  

(2.8)
- Third, assume that $\lambda_c$, $\lambda_i$ are not correct but they are not far from the optimum. In this case, maximum likelihood estimates of $\tau_h$, $\tau_c$ and $\tau_i$ can be obtained by deriving (3.6) with respect to them, while imposing that the minimum condition on (3.7) is satisfied. This leads to $\tau_h = J_h/n_h$, $\tau_c = J_c/n_c$ and $\tau_i = J_i/n_i$, so that $S_3$ is written as:

$$S_3 = N + \ln |V_h| + \ln |V_c| + \sum \ln |V_i| + n_h \ln \frac{F_h}{n_h} + n_c \ln \frac{F_c}{n_c} + \sum n_i \ln \frac{F_i}{n_i} + N \ln(2\pi)$$

(2.9)

In the absence of any dependent information, and assuming that the problem is not overparameterized, the computation leading to the smallest $S$ should be chosen. However, the user may use his (her) own judgement to decide which is the best choice of $\tau_h$, $\tau_c$ and $\tau_i$.

### 3.3 MINIMIZATION ALGORITHM

At this stage, we have built the objective function (3.7) whose minimum we take as our solution to the inverse problem. Therefore, the calibration problem has been transformed into a search for the minimum of a function, i.e., an optimization problem. Many types of optimization methods have been proposed and used in order to compute the minimum of different objective functions (see Yeh, 1986, or Carrera, 1987, for reviews of this topic as applied to groundwater flow).

Attending to the order of required derivation of the objective function, optimization methods can be divided into four groups: Newton, gradient search, Gauss-Newton and non-derivative methods. In gradient methods, the gradient of the objective function (3.7) is required. In Newton’s method we must compute the gradient of (3.7) and its hessian matrix. In Gauss-Newton we have to compute the gradient of (3.7) and head and/or concentration jacobian matrices. Among all of them, we have chosen Marquardt’s method, which belongs to the Gauss-Newton group. Against Newton’s method, Marquardt’s does not need second order derivatives, whose computation is extremely demanding. Furthermore, an additional advantage of Marquardt’s method appears when solving non linear flow or transport problems. The coefficient matrix for computing jacobians of the Newton-Raphson method, used for solving the direct problem, is identical to the coefficient matrix for computing the jacobians of heads and/or concentrations needed in the inverse solution. It is particularly useful because jacobians are one of the most expensive computations. Gradient methods are less expensive per iteration than Marquardt’s method, but they have a slower convergence in some cases (Cooley, 1985). Non-derivative methods are most effective when fast objective function computation can be made, although they have been used quite succesfully by Woodbury and Smith (1988) in groundwater problems. In our case we have to solve numerically two partial differential equations (sometimes non-linear) for every single objective function value.
3.3.1 Computation of derivatives

Derivatives of (3.7) and jacobian matrices can be computed in three different ways (Carrera, 1987, Yeh, 1986):

- Using the adjoint state.
- Numerical computation using a finite difference approximation.
- Deriving directly the discretized flow and solute transport equations (2.15 and 2.30).

Finite differences approximation of head and concentration jacobians requires perturbing model parameters one at a time and solving the corresponding flow and transport problem. This implies solving \( n_p + 1 \) direct problems for each inverse problem iteration (possibly \( 2n_p + 1 \), \( n_p \) being the number of parameters). It is clear that for sizeable \( n_p \) values, this method becomes computationally prohibitive (more critical yet in non-linear cases), not to mention the difficulties associated to the choice of the perturbation size. The adjoint state method is thoroughly described by Townley and Wilson (1985), Carrera and Neuman (1986b) or Sun and Yeh (1990), among others. It is effective for obtaining derivatives of scalar functions with respect to model parameters, because it implies a computational cost comparable to that of a single direct problem. However, we are interested in obtaining derivatives of heads and concentrations at measurement points. This implies an enormous computational effort for the case of a large number of measurements (a frequent case in transient problems). In fact, whenever the number of measurements exceeds that of model parameters, a most frequent circumstance, the adjoint state becomes more expensive computationally than direct derivation, not to mention difficulties associated to the need of solving the problem backwards in time.

As a result, we conclude that the most easy way of computing the objective function gradient and the flow and transport jacobian matrices consists in taking the discretized flow and transport equations and deriving both members of equations (2.12) and (2.35) with respect to a parameter \( p \). In the general non-linear case, we have, for the flow equation:

\[
\left\{ \left( \theta f A^{k+\varepsilon} + \frac{D^{k+\varepsilon}}{\Delta t} \right) + \varepsilon \frac{\partial A^{k+\varepsilon}}{\partial h^{k+\varepsilon}} (\partial h^{k+1} + (1 - \theta) h^k) + \varepsilon \frac{\partial D^{k+\varepsilon}}{\Delta t} \frac{\partial h^{k+1} - h^k}{\partial p} \right\} \frac{\partial h^{k+1}}{\partial p} =
\]

\[
(1 - \varepsilon) \frac{\partial A^{k+\varepsilon}}{\partial h^{k+\varepsilon}} [(\theta - 1) h^k - \theta h^{k+1}) + (1 - \varepsilon) \frac{\partial D^{k+\varepsilon}}{\partial h^{k+\varepsilon}} \frac{1}{\Delta t} (h^{k+1} - h^k)] + (\theta - 1) \frac{\partial b^k}{\partial p} + \frac{1}{\Delta t} \frac{\partial D}{\partial p} (h^{k+1} - h^k) + \theta \frac{\partial b^{k+1}}{\partial p} + (1 - \theta) \frac{\partial b^k}{\partial p} + (\theta - 1) \frac{\partial A^{k+\varepsilon}}{\partial p} \frac{\partial h^k}{\partial p}
\]

As stated before, there is a common block of computations in the direct and inverse solution, that is, the coefficient matrix in the left hand side of (3.11) is equal to \( \partial f(h)/\partial h \).
in (2.16). The only particularity is that the matrix used in (3.11) for computing \( \frac{\partial h}{\partial p} \), corresponds to that obtained in the last iteration of the updating process represented in (2.16) (once obtained convergence), which leads to the solution of the direct problem at time \( k + 1 \) (\( h^{k+1} \)).

Thus, jacobians computation only requires backward and forward substitution, because it can take advantage of the LU factorization already performed for solving the direct problem. As a result, while equation (3.11) implies solving \( n_p \) systems, the cost of each one remains moderate.

The previous comment pointed out that the process of computing flow and transport derivatives is coupled with the direct problem (simulation), since heads are needed to compute \( \frac{\partial h}{\partial p} \) in (3.11). As an extra task, we need to compute matrix derivatives with respect to model parameters.

A similar equation to (3.11) can be derived for concentration derivatives from equation (2.35). There is, however, an important difference. Dependency of matrix \( E \) (equation 2.35) is not explicit in all parameters. This matrix depends on flow parameters through Darcy velocity (eq. 2.22) and boundary flows. For this reason, to compute matrix \( E \) derivatives with respect to flow parameters, one must derive \( q \) and boundary flows with respect to flow parameters and then apply the chain rule:

\[
\frac{\partial e_{ij}}{\partial p_f} = \frac{\partial e_{ij}}{\partial q_x} \frac{\partial q_x}{\partial p_f} + \frac{\partial e_{ij}}{\partial q_y} \frac{\partial q_y}{\partial p_f} + \frac{\partial e_{ij}}{\partial q_z} \frac{\partial q_z}{\partial p_f} + \frac{\partial e_{ij}}{\partial \omega} \frac{\partial \omega}{\partial p_f}
\]

(2.11)

where \( p_f \) is a flow parameter, \( q_x, q_y \) and \( q_z \) are \( q \) components, \( e_{ij} \) is the component \( i \) of matrix \( E \) and \( \omega \) stands for the boundary flow. Once equation (3.12) has been used for the computation of \( E \) and \( g \) derivatives, putting these values on the corresponding equation similar to (3.11), one obtains the derivatives of concentrations with respect to flow parameters (because of the explicit dependence of \( E \) and \( g \) on transport parameters, these derivatives are computed in an identical way to flow matrices derived with respect to flow parameters).

### 3.3.2 The Levenberg-Marquardt’s methods

This family of methods belong to the class of restricted step methods. These methods try to reduce the difficulties of some methods based on functional quadratic approximations. Levenberg-Marquardt’s methods solve a system like:

\[
\left( H^{(m)} + \mu^m I \right) \Delta p^{(m)} = -g^{(m)}
\]

(2.12)

where \( H^{(m)} \) is the hessian matrix of the objective function (usually an approximation) at point \( p^{(m)} \in \mathbb{R}^n, \mu^m \in \mathbb{R}^+, g^{(m)} \) is the gradient at point \( p^{(m)} \), and \( \Delta p^{(m)} \) is parameter increment, \( p^{(m+1)} = p^{(m)} + \Delta p^{(m)} \). The parameter \( \mu^m \) is updated at every \( (m) \) iteration following several empirical criteria described in next section. This parameter controls the parameters increment direction, \( \Delta p^m \). If \( \mu^m = 0 \), Marquardt’s method coincides with Gauss-Newton’s method. As \( \mu^m \) increases, the length of \( \Delta p^{(m)} \) decreases and its direction approaches that of the objective function gradient (Marquardt, 1963). The criteria we
use are similar to those proposed by Cooley (1985). Second order derivatives of residuals (differences between computed and measured data) are neglected in the computation of matrix $H$. Our implementation of Marquardt’s algorithm is inferred from equation (3.14) with an appropriate array scaling (Cooley, 1985):

\[
\left( E^{-1} H^{(m)} E^{-1} + \mu^m I \right) E^{-1} \Delta p^{(m)} = - E^{-1} g^{(m)}
\]

where

\[
H^m = J^h V_h^{-1} J_h + \lambda_c J^c V_c^{-1} J_c + \sum_i \lambda_i v_i^{-1}
\]

$g$ is computed as

\[
g^m = J^h V_h^{-1} (h - h^*) + J^c V_c^{-1} (c - c^*) + V_p^{-1} (p - p^*)
\]

\[
V_p = \begin{pmatrix}
\lambda_1 V_1 \\
\vdots \\
\lambda_n V_n
\end{pmatrix}
\]

where $n$ is the number of different parameter types in the estimation, $J_h$ and $J_c$ are head and concentration jacobian matrices respectively, $V_h$, $V_c$ and $V_i$ are known positive definite symmetric matrices containing parameters covariance structure and $\lambda_i$ as defined in section 3.2. $E = \sqrt{H_{ij} \delta_{ij}}$ where $H_{ij}$ stands for the element of row $i$, column $j$ of the matrix $H$.

### 3.3.3 Marquardt parameter updating

Parameter $\mu$ is updated depending on several empirical criteria. It should be noticed, that Marquardt’s minimizing direction ($\Delta p^k$ in equation 3.15) coincides with the one of Gauss-Newton method if $\mu = 0$. For increasing $\mu$, $\Delta p^k$ is getting smaller in norm and it approaches gradient descent direction (Marquardt, 1963). Updating criteria are based on three different assumptions:

- The algorithm must reduce the objective function value at every iteration.
- If objective function is nearly quadratic, a small value of $\mu$ should be taken, because Gauss-Newton method converges in one iteration when minimizing a quadratic function.
- The angle between parameter increment and minus gradient should not be larger than $90^\circ$.

Using the Taylor’s series expansion of function $F$ at point $p^k$:

\[
F^{m+1} = F^m + g^m \Delta p^m + \frac{1}{2} \Delta p^m^T H^m \Delta p^m + O \left( \|\Delta p^m\|^3 \right)
\]
If $F$ were quadratic, the last term of (3.17) would be zero, so if we define $\phi$ as

$$\phi = \frac{F^{m+1} - F^m}{g^{mt} \Delta p^m + \frac{1}{2} \Delta p^{mt} H^m \Delta p^m}$$

(2.17)

$\phi$ would be equal to 1.

One can avoid direct computation of $\Delta p^k H^k \Delta p^k$ (using the approximation of the Hessian matrix, eq. 3.15), because from Marquardt’s system (3.14):

$$\left( \frac{1}{2} E^{-1} H^m E^{-1} + \mu I \right) E \Delta p^m = -\frac{1}{2} E^{-1} g$$

(2.18)

where $E$ its the diagonal matrix composed by the square root of the diagonal elements of $H^k$. Multiplying both sides by $\Delta p^k E$ one obtains

$$\frac{1}{2} \Delta p^{mt} H^m \Delta p^m + \mu \Delta p^{mt} E^2 \Delta p^m = -\frac{1}{2} g^{mt} \Delta p^m$$

(2.19)

Parameters are not allowed to vary completely free. Its relative change is controlled with variables PERMX1 and PERMX2 (card A8.2). PERMX1 is the maximum relative change that is allowed for parameters not treated logarithmically (i.e., parameters which have variable IOLGXXX set to 0, card A5.1). PERMX2 is the maximum relative change that is allowed for parameters treated logarithmically (i.e., parameters with variable IOLGXXX set to 1). If the variation of one or several parameters is more than allowed by these variables, a reduction factor is defined for them all, $\alpha$, and all parameter increments are reduced by this factor. In short, the actual parameter increment that the code uses is $\Delta q^k = \alpha \Delta p^k$, obtaining from (3.20) (substituting $\Delta p^k$ by $\Delta q^k$):

$$\frac{1}{2} \Delta q^{mt} H^m \Delta q^m = -\mu \Delta q^{mt} E^2 \Delta q^m \alpha \Delta q^{mt} \Delta q^m$$

(2.20)

Expression (3.18) may then be written as:

$$\phi = \frac{F^{m+1} - F^m}{g^{mt} \Delta q^m (1 - \frac{1}{2} \alpha) - \mu \Delta q^{mt} E^2 \Delta q^m}$$

(2.21)

This expression requires only $O(n)$ operations because matrix $E^2$ is diagonal (instead of $O(n^2)$ operations required in 3.18).

In general $\phi \neq 1$. Depending on its distance to 1, parameter $\mu$ is divided or multiplied (values controlled for variables in card A8.1). If $\phi$ is near to 1 ($|\phi - 1| < \text{PHIMIN}$) the code considers that the objective function agrees quite well with its quadratic approximation and $\mu$ is divided by NUMAX (because $\mu=0$ is the best value for quadratic functions).

If PHIMIN $< |\phi - 1| < \text{PHIMAX}$, $\mu$ is divided by NUMIN. If $|\phi - 1| > \text{PHIMAX}$, $\mu$ is multiplied by NUMIN.

Another criterium is based on the basic assumption that objective function must be reduced at every iteration. If an iteration fails (the iteration is called failed iteration; failed iterations are controlled with variable NMTERF1 in card A8.2) to reduce the objective function value, $\mu$ is multiplied by NUMAX and the code computes a new parameter increment again (using equation 3.15). This is repeated until the objective function has
been reduced or until the number of failing iterations is greater than allowed (NMTERF1 in card A8.2) in which case the code stops and convergence has not been reached.

The last criterion tests the angle between arrays \( g^m \) and \( \Delta p^m \) (eq. 3.15). The direction of \(-g^m\) is the maximum descent direction of function \( F \), but most times \( \Delta p^m \) is beyond the neighborhood where this sentence is true. Despite this, ensuring that \( \Delta p \) aligns along a descent direction suggests that array \( \Delta p^m \) should not form an angle with \(-g^m\) close to 90°. This leads to another control consisting in multiplying \( \mu \) by NUMIN (remember that if \( \mu \) increases, the increment parameter direction approaches that of \(-g^m\)) if the angle between \( \Delta p^m \) and \(-g^m\) is close to 90° during MAXICOS iterations (usually \( MAXICOS=1 \)). This issue is controlled by variable COSMIN (Card A8.1). This variable is the cosine of the largest angle that we allow between \(-g^m\) and \( \Delta p^m \). For example, if we allow this angle up to 90°, we set COSMIN=0.

### 3.3.4 Convergence criteria

The final objective of any optimization algorithm consists in finding a convergent sequence \( \{p_n\} \in \mathbb{N} \) with limit \( \hat{p} \) (the minimum of \( f \)) such that the sequence of images of the objective function \( \{f(p_n)\} \) is decreasing to \( f(\hat{p}) \). This concept, although easily defined from an analytical point of view, involves a difficult numerical decision. For instance, nobody can generate an infinite sequence, only a finite number of their terms. This leads to the decision about the number of members of the sequence we should generate. The right answer, of course, would be to stop when \( ||p_n - \hat{p}|| \) would be small enough (\( \hat{p} \) being the supposed minimal point of our objective function, 3.7), but \( \hat{p} \) represents what we want to know, so we have to look for other criteria.

Another way would be imposing the mathematical minimum conditions (\( g = 0 \) is necessary and positive-definiteness of \( H \) is sufficient), but their practical implementation is very difficult. \( g \) never becomes identically zero because of roundoff and truncation errors. In fact, in real problems, the solution can be very close to the minimum and still \( g \) be large. Therefore, a number of criteria have to be used. They are briefly outlined below. In general, one can say that well-posed inverse problems converge well and that any of the criteria below may properly identify the minimum. On the other hand, unstable problems may be stopped by some of the criteria below before convergence has been reached (see Sections 3.7 and 4.9).

TRANSIN stops iterating when one of the following criteria is met (control variables are entered in Cards A8.1 and A8.2, see Appendix I):

1. **Number of iterations** (MAXITER). This criterion is defined for safety. If the problem stops for this reason, then it is likely that convergence has not been reached.

2. **Gradient’s norm** (GNORM): If the gradient becomes small, the solution is near the optimum and the iterative process stops. This variable depends on many factors and is difficult to choose before having run the problem at least once. We have found it useful only for synthetic problems.

3. **Relative reduction of gradient’s norm with respect to the first gradient’s**
norm (GNORM1): Because of the difficulties in setting an absolute limit for the norm of the gradient, it is often convenient to set a relative value. Hence, iterations are stopped if:

$$\frac{|| g_i ||}{|| g_1 ||} < \text{GNORM1}$$  (2.22)

where $|| g_i ||$ is gradient’s norm at iteration $i$ and $|| g_1 ||$ is the initial gradient’s norm. Theoretically, this is a poor criterion because it depends on the initial point (ideally, convergence should be independent of the point from which iterations are started). Yet, we have found this criterion to be the most reliable indicator of actual convergence. That is, if iterations are stopped as a result of (3.23), then it is likely that the solution is indeed close to a minimum. If initial parameter estimates were far from the solution, the user may want to restart the problem to get an improved solution.

4. Relative reduction of objective function per iteration (DMINF): The algorithm is stopped if the reduction in the value of the objective function becomes negligible, that is, if

$$\frac{|J^i - J^{i-1}|}{J^i} < \text{DMINF}$$  (2.23)

where $J^i$ is the objective function at the current iteration and $J^{i-1}$ is its value on the previous one. Actually, near the optimum or in areas of low sensitivity of the objective function to the parameters, reduction in the value of the objective function may be small and the computed parameters may still be far from their optimum value. In short, this criterion often indicates the inability of the method to further reduce the objective function, rather than indicating that convergence has been reached.

5. Maximum change in parameter values. The last criterion used by TRANSIN refers to the maximum parameter change per iteration. The process stops if the maximum relative increment of parameters is less than EPS.

Automatic calibration is not as simple as might be suggested by this section. It is complicated by two sets of problems. The first is related to conceptualization and the second to instability and convergence difficulties. The initial conceptualization derived by the modeler is rarely satisfactory. When calibrating the corresponding model, one is usually forced to revise the initial assumptions, often in the direction of increasing their complexity and the number of model parameters.

A large number of model parameters leads to the second set of problems, namely instability and lack of convergence. These can be assessed through the error analysis, which is discussed next, while model selection is reviewed in Section 3.5.

An schematic summary of the inverse problem process developed in TRANSIN is presented in figure 3.1. The general algorithm contains three nested loops; one for solving the direct problem at time $k$ (Newton-Raphson loop), one for updating the state variable in time (time marching loop) and one for updating the set of parameter values (Marquardt’s loop).
Figure 3.1: Schematic description of the general algorithm implanted in TRANSIN
3.3.5 ERROR ANALYSIS

Having reached this point, we are able to compute the model parameters that minimize (3.7). We have obtained some parameter values that are assumed to resemble the true ones. Of course, the latter can never be reached. In fact, we cannot even be sure about the distance (difference in norm) between our parameters and the true ones (except for synthetic cases). Despite this, we need to have an idea about the confidence of the estimated parameters. The first suggestion that one may make is to use the variance of the estimated parameters in order to analyze the confidence level.

Assymptotically, the maximum likelihood estimator follows a gaussian distribution. As a result, it is fully defined by its expectation and its covariance matrix. The expectation is the estimator itself and a lower bound of its covariance matrix may be computed as the inverse of Fisher’s information matrix, \( F \) (Bury, 1975):

\[
F_{ij} = \frac{1}{2} E \left[ \frac{\partial^2 S}{\partial \theta_i \partial \theta_j} \right]
\]

where \( S = -2ln(L) \) (eq. 3.6) and \( \theta \in \mathbb{R}^m \) is the parameter array, in which both model parameters (transmissivity, porosity, etc) and statistical parameters (\( \tau_h, \tau_c, \tau_i \), plus the additional ones defining the residual and prior information structure matrices \( V_h, V_c, V_i \)), are included. \( m \) equals the number of parameters (model and statistical). It should be remembered that the model parameters with respect to which we are deriving in expression (3.25) may not be the parameters themselves but a one-to-one transformation (usually logarithmic). Since we are interested in evaluating the uncertainty of model parameters, we restrict the covariance matrix to physical parameters and ignore, at this stage, statistical parameters.

In order to compute \( F \), one needs to derive equation (3.25) twice with respect to model parameters. However, terms containing second order derivatives of heads and concentration are neglected, both because they are computationally expensive and because they multiply residuals whose expected values are zero. This leads to:

\[
F = \tau_h^{-1} J_h^t V_h^{-1} J_h + \tau_c^{-1} J_c^t V_c^{-1} J_c + C_p^{-1}
\]

where \( J_h \) and \( J_c \) are head and concentration jacobian matrices respectively and \( C_p \) the model parameter covariance matrix. The advantage of this approximation is that no extra computational effort is required, because all the expressions in (3.26) have already been computed, as discussed in section 3.3. From equation (3.26), the following approximation for model parameters lower bound covariance matrix is obtained:

\[
\Sigma = E \left[ \tau_h^{-1} J_h^t V_h^{-1} J_h + \tau_c^{-1} J_c^t V_c^{-1} J_c + C_p^{-1} \right]^{-1}
\]

The above expression might be also obtained making a first order approximation for heads and concentrations. Matrix \( F \) and its inverse \( \Sigma \), which is a lower bound of the covariance matrix, become useful for assessing stability.
3.3.6 MODEL SELECTION

As discussed up to here, inversion consists of computing the most likely model parameters, given head and concentration measurements and parameters prior estimates. In practice, however, selection of relevant physical processes and parameterization (section 2.2.2) are rarely unambiguous. That is, more than one model structure may be available and the question then becomes how to choose the best one. Traditionally, model selection is performed by residual analysis. That is, modelers tend to select models with small residuals (differences between measured and computed data). In fact, the initial model structure is often modified after a preliminary calibration, so as to reduce residuals in the next calibration run. Unfortunately, this process tends to increase the number of model parameters, hence reducing the reliability of their estimates and their validity for model predictions. This suggests the use of objective model selection criteria.

3.3.7 Model selection criteria

These criteria have been derived in the field of time series analysis based on one of the following objective:
- Minimizing a distance between the real system and the model, using information theory.
- Seeking consistency, i.e. that the probability of selecting the right model tends to one as the number of data tends to infinity.
- Maximizing the a posteriori probability of selecting the right model.
- Cross validation methods in which the data set is divided into two disjoint subsets, using the first one for calibration and the second one for validation. None of these methods can be applied exactly. For this reason, all the methods we show below are approximated.


\[ AIC = S + 2M \]  \hspace{1cm} (2.27)

where \( S \) is defined in (3.6) and \( M \) stands for the number of estimated parameters.

The second criterion was developed simultaneously by several authors (Akaike, 1977, Rissanen, 1978, and Schwarz, 1978):

\[ BIC = S + M\ln N \]  \hspace{1cm} (2.28)

where \( N \) is the total number of data. Hannan (1980) derived:

\[ \phi = S + cM\ln(lnN) \]  \hspace{1cm} (2.29)

where \( c \) is a constant greater than 2. Usually it is taken as \( c = 2 \).

The last criterion is due to Kashyap (1982):

\[ d = S + M\ln(N/2\pi) + \ln|F| \]  \hspace{1cm} (2.30)
Where $F$ is Fisher’s information matrix (3.19).

All these criteria respect the parsimony principle, i.e., under equal conditions, the simpler model (with a smaller number of parameters) is chosen. It is not clear that this property is always desirable. One may sometimes wish to work with a number of parameters greater than that indicated by these criteria, based on both quantitative and qualitative information on aquifer heterogeneity. We resolve this problem partly by embedding the quantifiable prior information in the likelihood function, as discussed earlier. The effect of qualitative information can be considered by comparing only models that are consistent with this qualitative knowledge, while recognizing that simpler models (with fewer parameters) might lead to smaller values of the criteria. By comparing models of different complexity, one should be able to distinguish between their degrees of reliance on qualitative information (models with smaller criteria will be more “hard-data-based”).

Another interesting property of these criteria is that, while they tend to select the simpler model, more complexity may be built into the model as the data base increases. This is because as $N$ (the number of data) grows, so does the relative importance of the term $S$. Since minimization of this term leads to increasingly larger number of parameters, the overall minimum of (3.6) is displaced towards models with growing complexity as the data base increases.

Several comparisons done by Carrera (1984) and by Carrera and Neuman (1986c) with synthetic examples show that the last criterion seems to be the best, which agrees with the fact that the hypotheses under which it was deduced, are the most complete.
3.4 PREDICTIONS. ERROR ANALYSIS ON THE PREDICTIONS

Formulation of predictions involves a conceptualization of its own. Quite often, the stresses whose response is to be predicted lead to significant changes in the natural system, so that the structure used for calibration is no longer valid. Changes in the hydrochemical conditions or in the flow geometry may have to be incorporated into the model. While numerical models can be used for network design or as investigation tools, most models are built in order to study the response of the medium to several scenario alternatives. Therefore, uncertainty on future natural and man-induced stresses also cause model predictions to be uncertain. Finally, even if future conditions and conceptual model are exactly known, errors in model parameters will still cause errors in the predictions. In summary, three types of prediction uncertainties can be identified:

- Conceptual model uncertainties
- Stresses uncertainties
- Parameters uncertainties

The first group includes two types of problems. One is related to model selection during calibration. That is, more than one conceptual model may have been properly calibrated and data may not suffice to distinguish which one is the closest to reality. It is clear that such indetermination should be carried into the prediction stage because both models may lead to widely different results under future conditions. The second type of problems arise from improper extension of calibration to prediction conditions, that is, from not taking into account changes in the natural system or in the scale of the problem. The only way we think for dealing with this problem consists of analyzing carefully whether or not the assumptions in which the calibration was based are still valid under future conditions. In our experience, model uncertainties can be very large.

Uncertainties associated with future stresses, are evaluated by carrying out simulations under a number of alternative scenarios.

The last set of prediction uncertainties is the one associated with parameters uncertainties, which can be quantified quite well. If $P$ is a prediction to be made with the model ($P$ is a function of the parameters), a lower bound of its variance is given by:

$$Var(P) = \left(\frac{\partial P}{\partial p}\right)^t \Sigma_p \left(\frac{\partial P}{\partial p}\right) + \sigma^2_{p0}$$

where $\sigma^2_{p0}$ represents model errors independent of parameter uncertainties. As mentioned earlier, equation (3.32) represents a lower bound, so that actual uncertainty is larger than that. In order to overcome this limitation, Vecchia and Cooley (1987) have proposed an approach that takes into account non-linear effects, which are neglected in (3.32) and (3.27).
An alternative approach is to perform a sensitivity analysis. In this case, an acceptable range of variation of the parameters is derived from the error analysis of the estimation. Predictions are then made varying each parameter within such range. One of the limitations of this method is that it does not take into account correlations among parameters. These can be recognized in Monte Carlo methods.

### 3.5 IDENTIFIABILITY, UNIQUENESS AND STABILITY

The inverse problem (automatic calibration) may result in meaningless solutions for several reasons: improper conceptual model, incorrect discretization and/or poor convergence of the minimization algorithm. The issue of model selection has been discussed earlier, and adequate discretization will be the subject of Section 4.2. Poor convergence is associated to ill-posedness, in turn related to non-identifiability, non-uniqueness and/or instability.

Following is a definition of these terms, their effect on the solution of the inverse problem and ways to identify and mitigate them. For space reasons, we constrain ourselves to a sketchy description. Yet, the topic is extremely broad and, once again, well beyond the scope of this manual. Further details on their relevance to groundwater flow are discussed by Carrera and Neuman (1986b). Practical aspects, related to the application of TRANSIN, will be presented in Section 4.9.

Existence is the first requirement for well-posedness of any mathematical problem. It is usually ignored in the inverse problem literature because most modelers implicitly assume that a solution always exists, which can only be proven to be true when prior information is incorporated into the objective function. In the absence of prior information, a solution may not exist within any bounded parameter domain. Typically, this happens when the model includes a process which does not affect observed data. For example, if a prescribed head boundary is treated as a mixed boundary, the code may react by favouring an infinite leakage factor. Non-existence is easily detected because some parameters tend to take on absurd values. In fact, it often leads to numerical problems. To overcome this problem, one may be forced to change the conceptual model, fix the parameter value or include prior information.

Non-uniqueness refers to the existence of more than one set of parameters satisfying the first-order extremum conditions usually imposed for minimizing (3.7). However, for methodological purposes, it is convenient to distinguish between two types of non-uniqueness: non-identifiability and discrete non-uniqueness.

Non-identifiability refers to the possible existence of different parameter sets leading to the same heads and/or concentrations. For example, steady-state flow through a homogeneous medium does not depend separately on hydraulic conductivity and flow rate. Hence, multiplying all conductivities and sink/source terms by a constant does not affect computed heads, so that it will lead to identical objective functions in the absence of prior information. The effect of non-identifiability on the inverse problem is that a single solution cannot be sought, but rather infinite combinations. Non-identifiability can usually be determined by searching for the relevant combinations of parameters in
the equations governing the problem. Even if one fails to find the problem by simple mathematical arguments, non-identifiability can be easily detected because, in the absence of prior information, it leads to a singular information matrix. A singular information matrix can also be the result of parameters evolving towards absurd values (zero or infinite diffusivities, etc.). But these two effects can be easily distinguished by simply inspecting the values of parameters that led to singularity. Identifiability can be imposed by including prior information in the objective function or by fixing the value of the parameter causing non-identifiability to any reasonable value. In summary, non-identifiability can be easily detected and corrected, hence it rarely causes difficulties.

Discrete non-uniqueness refers to the possible existence of more than one local minimum. Inasmuch as most minimization algorithms only ensure local convergence, one can hardly ever be convinced that global convergence has been reached. In fact, a local minimum far from the global one may show large head residuals, hence leading the modeler to the erroneous conclusion that conceptualization should be revised. Detecting discrete non-uniqueness is not an easy matter, because convergence criteria, being local, cannot distinguish between global and local minima. In some cases, comparison of computed and measured heads and concentrations may indeed suggest that the solution is a local minimum and, at the same time, suggest how to change the parameters to ensure global convergence (see, e.g. Carrera et al., 1988). This approach works in some cases (in fact we have used it in all the cases in which we have been able to prove discrete non-uniqueness), but most often, especially for spatially distributed data, it does not. A more satisfactory proof of non-uniqueness is obtained by starting the iterative minimization procedure from different parameter sets. If they all lead to the same solution, then one tends to feel confident, though never positively sure, about global convergence. Otherwise, one may suspect discrete non-uniqueness. However, different stopping parameters for the solution algorithm are most often a consequence of poor stability, which is discussed next. Chances of uniqueness are improved by including prior information in the objective function.

Instability refers to the sensitivity of solution to input. Unstable inverse problems are characterized by elongated valleys or flat zones in the objective function. The exact location of the minimum in these areas is, first, difficult to find and, second, extremely sensitive to small perturbations in observed data or variations in error structure, discretization or solution approach. One of the problems with instability is that it causes convergence problems to most minimization algorithms. In fact, we are convinced that most of the examples of discrete non-uniqueness presented in the groundwater inverse problem literature were not such but rather highly unstable problems. Detection of instability is relatively straightforward; it can be identified by oscillating solutions (in spatially varying parameters), by high dependence on marginal issues (e.g., slight changes in discretization), by poor convergence rates or, most definitely, through the error analysis. High parameter correlations or large condition numbers (ratio of largest to smallest eigenvalue) indicate that the problem is unstable, and may suggest ways to change the problem set-up so as to improve stability (e.g., lump together two highly correlated parameters, fix one of them, etc.). Stability may also be aided by inclusion of prior information in the objective function. But, other than that, little can be done about it. In fact, instability may not be a consequence of the way the problem was set up, but rather of the insufficiency of data for estimating all parameters. Hence, it can only be resolved by collecting additional data.
Chapter 4

USING TRANSIN

4.1 INTRODUCTION

This chapter is devoted to explaining TRANSIN basic usage in some detail.

Taking full advantage of the code requires commanding a rather wide set of topics. These include basic hydrogeology, statistics and numerical methods. The first is needed for formulating meaningful conceptual models. Statistics helps in working with covariance matrices and other concepts that rarely get hydrogeologists excited. Finally, a minimum knowledge of numerical methods is required for being aware of numerical errors and how to reduce them. Providing an equally intensive description of all these issues is difficult. Considering that conceptualization is extremely problem dependent and that we assume that most users are either hydrogeologists or modelers with significant hydrogeological experience, no hints will be given here on how to build conceptual models.

Basic construction of the model structure is given in Sections 4.2 (discretization) and 4.3 (parameterization). Verification of the discretization is discussed in Section 4.4 and specification of observed heads in 4.5. Sections 4.6 through 4.8 are devoted to cursory descriptions of input and output files. Finally, a discussion of frequent difficulties encountered while running TRANSIN is presented in Section 4.9.

Expert modelers, as well as brave first-timers, may wish to go straight to Appendix I and prepare the input files. While not trying to discourage them, we would strongly advise first reading Sections 4.2.3 and 4.3, dealing with time discretization and parameterization, which are treated by TRANSIN in a rather different way from most other codes. Also, return to Section 4.9 when running into difficulties.

4.2 DISCRETIZATION

Discretization involves subdividing the spatial domain into a set of finite elements and the time domain into a set of time intervals. The procedure to build, enter and verify the finite element grid and time discretization is given in this section. Construction of the
finite element grid follows the usual guidelines for constructing a grid of linear, triangular and/or quadrilateral elements. Almost all of the criteria given in the next section are also valid for constructing 3-D grids. Nevertheless, it is clear that even fairly simple 3-D problems demand grids whose construction can become very time-intensive. Therefore, a 3-D grid generator is required in these cases. This section can be skipped by readers familiarized with FEM grid construction.

4.2.1 Steps for constructing the FE grid

Step 1: Construct a scale map of the region of interest, indicating all the major hydrologic properties of interest. Define and mark the physical boundaries of the area to be modeled.

Step 2: Mark in the locations of all pumping and observation wells, prior knowledge on the zones of transmissivity, storativity and other hydrologic parameters, and the types of boundary conditions (e.g., prescribed flow, prescribed head, etc.).

Step 3: Based on the information from Step 2, use judgement to draw the streamlines of the approximate flow pattern expected to result from pumping and boundary conditions.

Step 4: On the basis of the flow net obtained in Step 3, draw in a finite element grid, using the basic rules given in next section.

Step 5: Number the nodes. During numbering, keep the matrix bandwidth in mind. The bandwidth is minimized when the maximum difference between adjacent numbers nodes is minimized with respect to all elements in the domain. Remember that CPU time increases at a rate more than proportional to the problem bandwidth. Due to this, you should use a numbering scheme that minimizes this difference. In a multilayer problem, after numbering a layer continue the numbering onto the next layer. Although nodes may be numbered at random (but strongly not recommended), no intermediate number can be omitted.

Step 6: Establish a coordinate system and record sequentially the coordinates of all nodes. This can be done by use of a superimposed millimetered paper, by geometric considerations (if the grid is regular enough), by means of a digitizer or by a grid generator. The latter method is the easiest and quickest approach. While the coordinate system can be arbitrary, accuracy is gained by imposing the condition that the number of significant figures is similar to the range of coordinates variation. For instance, if X coordinate ranges between 7087453 and 7089622 m, the user should substract 7080000 (or 7087000) from all X coordinates, so that they range between 7453 and 9622 (or 453 and 2622). Moreover, if the medium is anisotropic and principal directions remain constant in space, input data and estimation are simplified by setting coordinate axes parallel to principal directions. For the TRANSIN code, it is most convenient to record the number and coordinates of each node on a separate line of a data file using the FORTRAN code format (I5, F10.a):

Node No.  X-Coordinate  Y-Coordinate  Z-Coordinate

<table>
<thead>
<tr>
<th>I5</th>
<th>F10.a</th>
<th>F10.a</th>
<th>F10.a</th>
</tr>
</thead>
</table>

where \(a\) is an arbitrary user defined number between 0 and 9 (i.e., use the first five columns, for typing the node number, next 10 columns for X coordinate, columns 16 through 25 for Y coordinate and columns 26 through 35 for Z coordinate).

Coordinates of omitted nodes will be interpolated linearly by TRANSIN. This can be used for simplifying input file preparation when a sequence of nodes are indeed aligned and equidistant, in which case only the first and the last nodal coordinates of the sequence have to be specified. Notice, however, that nodal properties (boundary conditions, etc.) of all interpolated nodes will be identical to those of the first node in the sequence, except for those where a non-zero default value has been defined (see Card B1.3 of Appendix 1).

When preparing the input file for multilayer problems, each layer should be represented on a X-Y plane (notice that Z coordinates are given). Geometrical consistency between layers is ensured through 1-D connections. It is clear that preprocessors must be used when some layers are not horizontal planes.

Notice also that in multilayer problems where the flow system is 3-D (\(z\) is defined in the input file), but is modeled as multiple 2-D layers TRANSIN handles this kind of problem as a 2-D problem; that is, if a specific problem does not contain full 3-D elements (type 9, 10 or 11), it has to be treated as a 2-D case (IODIM=2).

**Step 7:** Record the node numbers defining the corners of each element in a data file using the FORTRAN format:

\[
\text{Element No. Node 1 Node 2 Node 3 \ldots Node 8 Cross sect. area}
\]

\[
\begin{array}{ccccccc}
I5 & I5 & I5 & I5 & I5 & I5 & F10.a \\
\end{array}
\]

**Numbering must be counterclockwise around the element.** In this set of cards, all the elements of the mesh are defined including linear, triangular and quadrilateral.

**Step 8:** Check node and element data. This can be achieved using the rules described in Section 4.4.

### 4.2.2 General rules for grid construction

The following general rules may be used as guidelines in construction of the finite element grid (Step 4 of Section 4.2.1).

- a) The grid should conform to the boundaries and interfaces between materials with different properties.

- b) Triangles should not have any obtuse angles.

- c) When solving 2-D or 3-D problems try to avoid notably elongated elements. It is particularly critical when using the tetrahedral elements.
d) Although pumping and observation wells do not need to fall on nodal points, accuracy is generally better if they do.

e) Where possible, it is advisable that the mesh resemble the flow net. This is feasible for steady state regimes or when flow patterns do not change substantially over time.

f) In multilayer problems, it is convenient (though not required) to have grids with similar shape in the overlapping parts of the layers.

g) The key issue in preparing the grid is its basic size. Computer time increases nonlinearly with the number of nodes. This cost is particularly large when solving nonlinear problems. On the other hand, numerical errors in the solution of the direct problem also grow almost quadratically with grid size (actually, this is only true for the case of exact time discretization; see next section). Hence, a compromise between cost and accuracy must be reached. This makes it impossible to define absolute criteria for defining grid sizes. Bearing this relativism in mind, the following rules can be used:

   g.1) As a general rule, and aside from time discretization and singular points (pumping wells, etc.), the numerical as error for a given FE grid is of the same order of magnitude as that which would be made by interpolating the exact solution with the plane elements of the grid. In transient problems, the worst interpolation error with respect to time step should be taken as an upper bound. This rule leads to the following one.

   g.2) In general, elements ought to be small in the areas of large hydraulic gradients and/or marked head curvatures, but their size may grow away from these.

   g.3) Element size should increase smoothly away from the regions of large hydraulic gradients. As a rule of thumb, the area of any element should not be more than twice, the area of an adjacent element or its maximum dimension more than one and a half times that of the adjacent elements.

   g.4) Contacts between regions of largely different diffusivities (K/Ss or T/S) are an exception to the above rule. In fact, assuming that one needs accuracy, grid size perpendicular to the contact should be much smaller in the low diffusivity region than in the adjacent high diffusivity region.

   g.5) When solving problems in the unsaturated zone, the element size has to be increased away from:

   - Points where external sources of water are active.
   - Contacts between materials with different properties of retention or relative permeability.
   - Points where external imposed sinks are active.

   Selection of the size of the shorter element can be adopted based on results of several runs, with elements of different size.

   The largest element size must be small enough to adequately reproduce the pressure or moisture gradient.
g.6) A particular case of (g.4) is the discretization of the 1-D columns representing aquitards that connect different layers in multilayer models. As shown in Figure 2.1, the 1-D elements should be very small near the 2-D aquifers and grow exponentially towards the center of the aquitard. Moreover, if the layers simulate fractures (or thin aquifers), the size of the 1-D element adjacent to the 2-D layer should be particularly small to recognize the fact that half of its storage will be accounted for by the 2-D layer.

g.7) Element size near pumping wells depend on the location of observation points, time scale and required accuracy. It is very difficult to accurately simulate heads at pumping wells, which are controlled by local conditions and require element sizes comparable to the wellbore. No less than 5 elements should be used between the pumping well and the closest observation point. Their sizes should decrease towards the pumping well.

h) In order to solve the transport problem, the triangles in the mesh should be oriented such that their largest side coincides with flow direction. A comparison between analytical and numerical solutions is shown in figure 4.1. The numerical results were computed using two different triangular discretizations associated with a rectangular mesh (Kinzelbach, 1987). As can be observed, the best computed solution is obtained when the flow direction coincides with the hypotenuse.

i) To obtain accurate and stable numerical solutions to the transport equation, mesh size should be constrained. Similarly to the flow problem (rule g), accuracy requires that the mesh should be able to reproduce the contaminant’s front movement. This implies the following rules:

i.1) A minimum of four elements are needed to simulate accurately the contaminant front advance.

i.2) The previous issue implies that the grid should be refined around the sources.

i.3) One should refine the grid in interfaces with large parameter variations. For example, to model a narrow preferential path with large conductivities, the adjacent elements should also be narrow.

i.4) The mesh can be much coarser in zones that are not reached by the solute. Notice, however, that refinement is still needed upstream of the source.

j) Spatial stability of the numerical solution (oscillations and minimum principle violation) depends on the grid’s Peclet number, defined as:

\[ P_e = \frac{|q| \triangle L}{\phi D_m + \alpha L |q|} \]  

(4.1)

where \( \triangle L \) stands for nodal distance in flow direction. If \( D_m \) may be neglected, as is usually the case in solute transport problems, the Peclet number can be written as:

\[ P_e = \frac{\triangle L}{\alpha L} \]  

(4.2)
Figure 4.1: Directional effect in the triangular mesh:
a) Analytical solution
b) Rectangular triangles with hypotenuses parallel to flow direction.
c) Rectangular triangles with hypotenuses perpendicular to flow direction (Kinzelbach, 1987).
Traditionally, an upper bound of two has been imposed on (4.2) in order to ensure stability. This,

\[ P_e \leq 2 \] (4.3)

However, in practice larger values of the Peclet number can sometimes be achieved without notorious oscillations in the solution. In fact, when concentrations vary smoothly in space, this condition can be infringed by some orders of magnitude, without observable oscillations. Usually, the accuracy condition (rule i) is much more restrictive.

k) Transversal dispersivity controls the grid element extent in the direction perpendicular to flow. The ratio between longitudinal and transversal dispersivities should resemble the one of longitudinal and transversal element extents. However, this condition may be very restrictive, leading to very narrow elements. For this reason, transversal dispersivity often controls the required number of elements.

### 4.2.3 Time discretization

Temporal discretization consists of defining the time increments needed to solve discretized flow and transport equations (\( \Delta t \) in eq. 2.12 and 2.35) (IOTRS or IORTS equal to 1 or 2 in Card A3.3, Appendix I). The time domain must be discretized only for transient flow or transport problems. TRANSIN works with two superimposed time discretizations: observation and solution times. **Observation times** are defined for convenience and are used for specifying transient head and/or concentration measurements and temporal variation of stresses (prescribed external heads, areal recharge, prescribed nodal flow rates and prescribed concentration sources). On the other hand, **solution times** are defined for accuracy. Following is a description of how to specify both, along with details on time functions and rules for discretization.

**Observation and solution times**

As defined above, observation times are required for specifying time variability of stresses and head and/or concentration measurements. Some output data (piezometric and/or concentration measurements maps and mass balance) are only given at observation times. The interval between sequential observation times is termed the “**observation interval**”. The number of observation times is NINT (Card A4.1).

Solution times are required for solving flow and transport equations and are specified by subdividing each observation interval into several constant “**solution time steps**” (or intervals or time increments). The number of solution time steps per observation interval is given by the vector KINT (Card D1.1).

Thus, the length of the time increments within a particular observation interval, defined by the value of KINT (number of firstly proposed solution time steps), is called the **desirable time increment** (DTI). For linear problems, observation times and the number of solution time steps are enough data in order to define the time discretization.
Since the solution of non-linear problems is conditioned by a convergence process whose velocity is strongly dependent on the time increment, additional data is required by TRANSIN in this case. We defined the **maximum desirable time increment** (MDTI) within an observation interval to be the maximum value of the time increment which TRANSIN may adopt in order to solve the equation in the respective observation interval. To understand the usefulness of these variables DTI and MDTI, we describe a general sketch of the operatives necessary to obtain the solution of the equation in an observation time \((i)\) given that corresponding to time \((i-1)\). After having obtained the observation solution in time \((i-1)\), TRANSIN adopts the desirable time increment (DTI) in order to compute the first solution in the given observation interval. Time-dependent parameters are defined in TRANSIN (see sec. 4.2.3.2) at observation times. These parameters changes often cause convergence problems during the first solution time increment. Therefore, it is common to use a small value of the desirable time increment (DTI). It’s possible (actually very common) for DTI to be selected too large initially. Then, the program cannot reach convergence. In that case, TRANSIN will reduce DTI it several times if necessary until obtaining convergence. In general, it is not practical to maintain this small time increment. TRANSIN increases the time increment geometrically (with user-defined rate) at every solution time until reaching the maximum desirable time increment (MDTI). The MDTI solution time increment then maintained during the rest of the observation interval.

Finally, observation times are specified in the array TIME (Card D1.1). The arrangement of observation and solution times and intervals is depicted in Figure 4.2.

While TIME and KINT can be arbitrary, execution time per step is reduced if solution intervals are kept constant between successive observation intervals (Particularly in linear problems, because the flow matrix is constant in time and doesn’t need to be recalculated).

**Time functions**

As discussed in Section 4.2.3, time functions are specified for describing time variability of external stresses (areal recharge, prescribed heads, flow rates and concentration sources). As described in section 4.3, a time dependent recharge is defined as the product of a spatial dependent value and a time function. Notice that the same time function can be used for several elements and/or nodes, even if they belong to different parameter zones or represent different parameter types.

Time function values are specified at observation times by means of the array FNT (Card D2.1). Values at intermediate solution times are obtained by linear interpolation. However, the user should bear in mind that a weighted finite difference scheme is employed to calculate time derivatives. If weights are equal to one, a fully implicit scheme is used. In this case for mass balance purposes, the actual value of time-dependent parameters in TRANSIN calculations is step-wise constant, as shown in Figure 4.3.

Actual step changes can be represented using an observation interval with a single solution time just after the jump as seem in FNT(4) of Fig. 4.3.

**WARNING:** When solving transient problems with steady state initial conditions (IOTRS=2 for flow or IORTS=2 for transport in Card A3.3) the time function only
Figure 4.2: Specification of time discretization for linear (above) and non linear problems (below).

Figure 4.3: Definition and treatment of time functions. $\theta$ is the weight in the finite differences scheme of equations 2.15 and 2.30. In the picture the extreme cases are displayed, $\theta = 0$ and $\theta = 1$. These values are specified in card D3.1
applies to the transient part. While this can be used for specifying jumps from the steady-state values of stresses to their transient counterparts, the user should be careful when such jumps are not wanted (as can be deduced from the discussion in Section 4.3, this can be easily achieved by assigning adequate values to parameter coefficients, cards C1 through C3). The point to stress here is that FNT(1) does not represent the value of FNT at steady-state, but its value immediately after TIME(1), the beginning of transient simulation.

Rules for time discretization

As for spatial discretization, an optimal time discretization is reached by finding a compromise between accuracy (for non-linear problems), and CPU time. The main difference between time and spatial discretization is that modifying the latter can be very time-consuming, while the former can be easily changed, so that the effect of different time discretizations can be examined by the user with little input data effort (although it could be a big CPU effort). Hence, the rules described below can be used for defining initial values of time step sizes, which can be modified later by trial and error by the user or, because of convergence problems, by the code.

a) Definition of observation times. These are given as the times at which head data are available (see Section 4.5) and the times required for properly reproducing time variability of external actions. The latter may require drawing the actual time functions.

b) Preliminary definition of solution increments. Two criteria can be used for a preliminary rough definition of solution time steps:

b.1) Accumulated time discretization error rarely exceeds 10% of the largest head variation during a single solution time step. If sufficient head data are available, then one may estimate such head variation before actual simulation and derive time steps from them. For example, if maximum head change during a 30 day observation interval is 3 m and one does not want head errors to exceed 10 cm, then head change per solution time step should not exceed 1 m, which implies a 10 day solution time step or, equivalently, at least 3 solution steps for this observation time step (KINT).

b.2) The second criterion is based on the time required for perturbations to be transmitted through the medium. This time is controlled by the dimensionless time (flow equation):

\[ t_D = \frac{Kt}{S_s L^2} \]  

(4.4)

where \( L \) is the distance between the perturbation and the observation point, and \( K \) and \( S_s \) (note that for unsaturated problems \( S_s \) is replaced by an average value of the moisture capacity) stand for representative hydraulic conductivity and specific storage. Equation (4.4) leads to two different bounds for time increments. The first bound is based on the assumption that the dimensionless time computed with \( L = \Delta x \) (element size) should not be much great than unity. The second bound is obtained by imposing the solution that the computed dimensionless time increment...
using observation to pumping point distance \((d)\) should be taken as a fraction of one. The first of these assumptions leads to

\[
\Delta t \leq \frac{2S_s \Delta x^2}{K} \tag{4.5}
\]

for the first bound and the second assumption gives:

\[
\Delta t \leq \frac{0.1 S_s d^2}{K} \tag{4.6}
\]

b.3) Time discretization needed for transport equation. The first criterion is based on Courant’s number, which is defined as:

\[
C = \frac{|q| \Delta t}{\phi \Delta x} \tag{4.7}
\]

where \(|q|\) is Darcy’s velocity, \(\Delta x\) stands for mesh size and \(\phi\) is porosity. The classical stability condition is read as:

\[
C \leq 1 \tag{4.8}
\]

This condition may be relaxed when concentration fronts have been smoothed. It should be noticed that (4.8) implies that flow should not move more than one element in one time increment.

c) The above criteria may lead to very small time increments. These may be revised later during discretization verification.

d) After defining the solution time steps, their number for each observation interval (KINT) can be derived by simply dividing the interval length by the time step. At this stage, it is convenient to recall that, for linear problems, computer cost is reduced when the length of the solution time steps remain constant and a few solution time steps may be added for this purpose in each observation interval. For example, consider a problem with 7 observation intervals \((\Delta t = 30, 15, 20, 40, 50, 40, 80)\) which have been divided as \((\text{KINT}=5, 2, 4, 3, 5, 3, 7)\). Computer cost may be reduced, and accuracy improved using \((\text{KINT}=6, 3, 4, 4, 5, 4, 8)\), which implies a solution step of 5 for the first three observation times and of 10 for the last 4. The CPU time savings result because the method used for solving the linear systems of equations (2.15 and 2.30), takes advantage of the fact that system matrices do not change every solution time step when the time step is constant.

e) When solving non-linear problems, a preliminary run of TRANSIN is suitable in order to define the maximum desirable time increment. If the solution contains oscillations or it does not look smoothly continuous, the time discretization is probably not adequate. Furthermore, a notable number of time step reductions made by the code is an indication of an erroneous solution time step selection. On the other hand, when the solution is extremely variable in time, it is convenient to do several runs, using, every time, a discretization more refined than that used.
in the previous run. Then, based on examining the different solutions, one should choose the courses to discretization which leads to a solution visually equal to that obtained with greater levels of refinement.

Preliminary time steps can be defined with these rules. Final tuning of time discretization will be discussed in Section 4.4.

4.3 PARAMETERIZATION. PRIOR ESTIMATION

Aquifer parameters \((T, S, \text{ etc.})\) are defined over the flow domain and its boundary. Hence, they are functions of space (and/or time). Rather than estimating these functions, inverse modeling requires expressing them in terms of a discrete set of unknowns, called \textbf{model parameters}. The process of expressing aquifer parameters as functions of model parameters is termed parameterization. For a generic aquifer parameter varying in space over a given zone (let it be \(g(x)\)), most parameterizations can be expressed as (section 2.2.2):

\[ g(x) = \sum_{i=1}^{N} f_i(x) g_i \quad x \in \text{ zone I} \]  

(4.9)

where \(f_i(x)\) is a prespecified interpolation function and \(g_i\) is the corresponding model parameter. When \(g\) is not constant over time, expression (4.9) is multiplied by a time function.

Although different types of interpolation functions can easily be accommodated in TRANSIN, only zonation has been implemented so far because it is the best framework for geologically-based conceptual models. Hence, parameters are specified by zones, which may consist of one or several elements (or nodes) sharing some hydrogeological properties. Then, \(g(x)\) is defined as:

\[ g(x) = CF(N) \times G(I) \]  

(4.10)

where \(x\) belongs to element N (or to the boundary segment associated with node N), which in turn belongs to zone I. When \(g(x)\) depends on time, (4.10) is written as

\[ g(x, t) = CF(N) \cdot G(I) \cdot FT(t) \quad x \in \text{ zone I} \]  

(4.11)

where \(FT(t)\) stands for a time function value at time \(t\), in the given zone. Notice that aquifer parameters are taken as constant within a given element, although they can vary from one element to another within a given zone. Defining \(CF(N)\) for every element and every type of parameter can be tedious. However, it provides significant freedom to adapt the code to many different specific needs.

The expression (4.11) is a sufficiently general description of any physical parameter associated with a linear problem. Nevertheless, it is incomplete when referring to non-linear problems because of the dependence on the state variable, \(h(t)\) or \(c(t)\). Thus, for instance, the moisture capacity is a function of the moisture content which in turn depends on the pressure head. We propose (4.12) as a compact expression for a physical
parameter dependent upon the state variable,

\[ g(x, t) = CF(N) \cdot G(I) \cdot FT(t) \cdot FNL(I, h(t)) \quad x \in \text{zone I} \quad (4.12) \]

Although not all nonlinear parameters can be expressed strictly according to (4.12) the expression is suitable in order to transmit to the reader the treatment given by TRANSIN to nonlinearity. In (4.12), FNL is a factor depending on the state variable, \( h \) or \( c \). A significative difference between FNL and the rest of factors is that CF, G and FT are prescribed by the modeler. On the contrary with FNL, the user has to define the function FNL. TRANSIN will compute it with respect to time according to the current value of the state variable in the nodes of the elements. This relationship is called a “non-linear function” in TRANSIN. As described in chapter 2, TRANSIN offers several type of generic non-linear functions (see table 2.2). A non-linear function is defined by its type (i.e, van Genuchten retention curve), its agreement parameters and its generic parameters. (See Appendix A.9).

Thus, each zoned parameter, \( G(I) \) capable of adopting a non-linear form may have an associated number of a nonlinear function. This number, in turn, leads to a series of data related to the characteristics of the function, which is input by the user.

On the other hand, because of its physical nature, the generic parameters are treated as “class” of parameters such as transmissivity, storage coefficient or recharge, although they may have different physical meanings. Also, the generic parameters are considered as potential model parameters, that is, they can be considered unknowns.

A list of the most important variables related to parameters is shown in tables 4.1 and 4.2. As one can see, there are three groups of parameters: those that are defined by elements, i.e., on each element one may have a different value, and those parameters defined by nodes, i.e., on every node they can have different values. Generic parameters can belong either to element or nodes, according to their associated zoned parameter. It is important to note that one generic parameter may define more than one non-linear function if necessary. The nodal group of parameters is made up parameters related to boundary conditions (prescribed head, prescribed flow, leakage and external concentration). Consider external concentration, this parameter is included in the nodal group because external concentration may be associated with recharge (in this case it seems more natural defining this parameter by elements) or it may associated with infiltration through the boundary (in which case it seems more natural to define the parameter by nodes). Observe that all variables related to parameters are characterized by a root name (related to the name of the parameter) plus a suffix or prefix (see Appendix 2). Roots are:

- TRA Transmissivity (or hydraulic conductivity)
- STG Storage coefficient (or specific storage)
- ARR Areal recharge.
- CHP Prescribed head.
As an example, areal recharge at element number \( j \) is computed as:

\[
ARRC(LXARRT(j)) \times CFARRT(j) \times FNT(NFTARR(LXARRT(j)), t)
\]

where variable \( LXARR \) specifies the zone to which element \( j \) belongs to in steady state (\( LXARRT \) is the same for transient), \( ARRC \) is the model parameter \( G \) in equation (4.10), \( CFARR(j) \) is recharge element coefficient for steady state (as \( CFARRT \) for transient), \( NFTARR \) is the number of time function related to recharge zone \( LXARRT(j) \) and \( FNT \) is the value of that time function (see these variables in Table 4.1). It should be noticed that if no time function is specified for recharge, then the third multiplier in equation (4.14) is taken as one.

Analogously, to express the nodal parameter prescribed head at node \( i \) (\( h_p(i) \)) we’ll use the following structure:

\[
h_p(i) = CHPC(j) \cdot CFCHP(i) \cdot FNT(K, t)
\]

where \( j = IXCHP(i) \) \((4.15)\) and \( K = NFTCHP(j) \) \((4.16)\).

If we wanted now to define a non linear parameter, i.e., the total storativity in the element \( l \) (\( C(l) \)), which in turn belongs to the storage zone \( LXSTG(l) \), the expression is:

\[
C(l) = STGC(LXSTG(l)) \cdot CFSTG(l) \cdot FNL(PARACD, PRGC, \hat{h})
\]

Let’s develop arguments of \( FNL \). \( h \) is the average value of the stack variable of the state variable at the element \( l \). That is

\[
\hat{h} = \frac{1}{n} \sum_{j=1}^{n} h_j(t)
\]
where \( h_j(t) \) is the value of \( h \) at node \( j \) and time \( t \), and \( n \) is the number nodes of the element \( l \). In turn, generic parameter is allocated based on the variable NFNLPRG. This variable indicates the generic parameter order number(s) required for defining the function FNL. NFNLPRG has two arguments to be defined. The first one is the number which identifies the considered function (let’s call it NFNLSTG). The second argument, \( K \), considers the fact that more than one generic parameter can be used for defining the function number NFNLSTG. In summary, PRGC is defined as:

\[
PRGC = PRGC(NFNLPRG(NFNLSTG(LXSTG(l))), K) \quad (4.19)
\]

where \( K \) may vary between 1 to 8. It depends on the class of function employed, defined through the variable NFNLTYPE(NFNLSTG(LXSTG(l))). Thus, in Section A1.9 the physical meaning of the PRGC and \( K_{max} \) are given for each type of non-linear function. Variable PARCD is defined analogously as PRGC. At this stage, it is opportune to point out that the essential difference between PARACD and PRGC is that generic parameters contain physical significance and are defined by zones while agreements are geometrical variables and no defined in zonal structure.

\[
PARACD = PARACD(M, NFNLSTG(LXSTG(l))), m = 1, 3 \quad (4.20)
\]

Completely developed we have:

\[
STCG[LXSTG(t)] \times CFSTG(L) \times g \{PRGC[NFNLPRG(k, NFNLSTG(LXSTG(L)))]; PARACD[j, NFNLSTG(LXSTG(L))]; FNT[NFTPRG [M(t), NFNLSTG(LXSTG(L))], t]; h(t)\} \quad (4.21)
\]

In addition to specifying patterns of spatial variability through zones and nodal or element coefficients, the user must also define available prior information. This involves supplying a prior estimate value and a measure of its reliability (variance or standard deviation). As discussed in Chapter 3, the objective function penalizes parameter deviations from their prior estimates through the plausibility criteria:

\[
J_g = \frac{(g - g^*) V_g^{-1} (g - g^*)}{\lambda_g} \quad (4.22)
\]

where \( J_g \) multiplied by \( \lambda_g \) is the parameter type \( g \) contribution to objective function (recall equation 3.7), \( g^* \) is the vector of prior estimates (second column in Table 4.2) or their logarithmic transform (in fifth column of Table 4.2 is set to one), \( V_g \) is a matrix with the correlation structure of these estimates and \( g \) is the vector of computed values (or their logarithmic transform, third column in Table 4.1). Actually, with the only exception of transmissivities, \( V_g \) is taken by TRANSIN as diagonal, so that it is enough to specify the standard deviation (third column in Table 4.2) for each component of \( g \). Weight \( \lambda_g \) is specified by variables in the last column of Table 4.2 (if \( \lambda_g = 0 \), no prior information is available).

Notice that prior information is not used when only simulation is required (IOINV\( \leq 0 \) in card A3.3). The computed values of the parameters are given as the initial values at the input phase. It should be noticed that if estimation is done, but no prior information is given to the program, it takes the initial values of the parameters as prior information.
### Table 4.1: Most important variables dealing with model parameters and related to the direct problem.

<table>
<thead>
<tr>
<th>Param. type</th>
<th>Param.</th>
<th>Element or nodal coeff.</th>
<th>Model parameter&lt;sup&gt;3&lt;/sup&gt;</th>
<th>Number of zones</th>
<th>Time function</th>
<th>Element or nodal zone</th>
<th>Nonlinear function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmissivity&lt;sup&gt;1&lt;/sup&gt;</td>
<td>1</td>
<td>CFTRA</td>
<td>TRAC</td>
<td>NZTRA</td>
<td>—</td>
<td>LXTRA</td>
<td>NFNLTRTRA</td>
</tr>
<tr>
<td>Storage coeff.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>2</td>
<td>CFSTG</td>
<td>STGC</td>
<td>NZSTG</td>
<td>—</td>
<td>LXSTG</td>
<td>NFNLSTG</td>
</tr>
<tr>
<td>Areal recharge&lt;sup&gt;1&lt;/sup&gt;</td>
<td>3</td>
<td>CFARR&lt;sup&gt;4&lt;/sup&gt;</td>
<td>ARRC</td>
<td>NZARR</td>
<td>NFTARR</td>
<td>LXARR&lt;sup&gt;4&lt;/sup&gt;</td>
<td>NFNLARR</td>
</tr>
<tr>
<td>Presc. head&lt;sup&gt;2&lt;/sup&gt;</td>
<td>4</td>
<td>CFCHP&lt;sup&gt;4&lt;/sup&gt;</td>
<td>CHPCI</td>
<td>NZCHP</td>
<td>NFTCHP</td>
<td>IXCHP&lt;sup&gt;4&lt;/sup&gt;</td>
<td>—</td>
</tr>
<tr>
<td>Presc. flow&lt;sup&gt;2&lt;/sup&gt;</td>
<td>5</td>
<td>CFQQP&lt;sup&gt;4&lt;/sup&gt;</td>
<td>QQP</td>
<td>NZQQP</td>
<td>NFTQQP</td>
<td>IXQQP&lt;sup&gt;4&lt;/sup&gt;</td>
<td>—</td>
</tr>
<tr>
<td>Leakage&lt;sup&gt;2&lt;/sup&gt;</td>
<td>6</td>
<td>CFALF</td>
<td>ALFC</td>
<td>NZALF</td>
<td>—</td>
<td>IXALF</td>
<td>NFNLALF</td>
</tr>
<tr>
<td>Long. dispers.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>7</td>
<td>CFDSP</td>
<td>DLSLC</td>
<td>NZDSL</td>
<td>—</td>
<td>LXDSP</td>
<td>—</td>
</tr>
<tr>
<td>Transv. dispers.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>8</td>
<td>CFDSP</td>
<td>DSTC</td>
<td>NZDST</td>
<td>—</td>
<td>LXDSP</td>
<td>—</td>
</tr>
<tr>
<td>Molecular diff.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>9</td>
<td>CFDFM</td>
<td>DFMC</td>
<td>NZDFM</td>
<td>—</td>
<td>LXDFM</td>
<td>—</td>
</tr>
<tr>
<td>Porosity&lt;sup&gt;1&lt;/sup&gt;</td>
<td>10</td>
<td>CFPOR</td>
<td>PORC</td>
<td>NZPOR</td>
<td>—</td>
<td>LXPOR</td>
<td>—</td>
</tr>
<tr>
<td>Retard. coeff.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>11</td>
<td>CFCRD</td>
<td>CRDC</td>
<td>NZCRD</td>
<td>—</td>
<td>LXCRC</td>
<td>NFNLCRD</td>
</tr>
<tr>
<td>First order reac.</td>
<td>12</td>
<td>—</td>
<td>XLAMC</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>External conc.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>13</td>
<td>CFCOE</td>
<td>COEC</td>
<td>NZCOE</td>
<td>—</td>
<td>LXCOE</td>
<td>NFNLCOE</td>
</tr>
<tr>
<td>External conc.&lt;sup&gt;2&lt;/sup&gt;</td>
<td>13</td>
<td>CFCON&lt;sup&gt;4&lt;/sup&gt;</td>
<td>COEC</td>
<td>NZCOE</td>
<td>NFTCOE</td>
<td>IXCON&lt;sup&gt;4&lt;/sup&gt;</td>
<td>NFNLCOE</td>
</tr>
<tr>
<td>Generic parameters&lt;sup&gt;5&lt;/sup&gt;</td>
<td>14</td>
<td>1.0</td>
<td>PRGC</td>
<td>NZPRG</td>
<td>NFTPRG</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

<sup>1</sup> Parameter defined by elements.  <sup>2</sup> Parameter defined by nodes.  <sup>3</sup> On input, it contains the initial value. On output, it contains the estimated value.  <sup>4</sup> A different variable which name is the same plus a T at the end is employed in transient, e.g., CFARRT has the same meaning as CFARR but the former is for transient regime and the second for steady state.  <sup>5</sup> Parameters associated to non-linear functions.

### Table 4.2: Most important variables dealing with model parameters and related to inverse problem.

<table>
<thead>
<tr>
<th>Param. type</th>
<th>Prior Information</th>
<th>Standard deviation</th>
<th>Estimation index</th>
<th>Logarithmic estimation</th>
<th>Weighting parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmissivity&lt;sup&gt;1&lt;/sup&gt;</td>
<td>1</td>
<td>TRAM</td>
<td>STTRA</td>
<td>IVTRA</td>
<td>IOLGTRA</td>
</tr>
<tr>
<td>Storage coeff.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>2</td>
<td>STGM</td>
<td>STSTG</td>
<td>IVSTG</td>
<td>IOLGSTG</td>
</tr>
<tr>
<td>Areal recharge&lt;sup&gt;1&lt;/sup&gt;</td>
<td>3</td>
<td>ARRM</td>
<td>STAR</td>
<td>IVARR</td>
<td>IOLGARR</td>
</tr>
<tr>
<td>Presc. head&lt;sup&gt;2&lt;/sup&gt;</td>
<td>4</td>
<td>CHPM</td>
<td>STCHP</td>
<td>IVCHP</td>
<td>IOLGCHP</td>
</tr>
<tr>
<td>Presc. flow&lt;sup&gt;2&lt;/sup&gt;</td>
<td>5</td>
<td>QQPM</td>
<td>STQQP</td>
<td>IVQQP</td>
<td>IOLGQQP</td>
</tr>
<tr>
<td>Leakage&lt;sup&gt;2&lt;/sup&gt;</td>
<td>6</td>
<td>ALFM</td>
<td>STALF</td>
<td>IVALF</td>
<td>IOLGALF</td>
</tr>
<tr>
<td>Long. dispers.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>7</td>
<td>DSLM</td>
<td>STDSL</td>
<td>IVDSL</td>
<td>IOLGDSP</td>
</tr>
<tr>
<td>Transv. dispers.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>8</td>
<td>DSTM</td>
<td>STDST</td>
<td>IVDST</td>
<td>IOLGDSP</td>
</tr>
<tr>
<td>Molecular diff.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>9</td>
<td>DFMM</td>
<td>STDFM</td>
<td>IVDFM</td>
<td>IOLGDFM</td>
</tr>
<tr>
<td>Porosity&lt;sup&gt;1&lt;/sup&gt;</td>
<td>10</td>
<td>PORM</td>
<td>STPOR</td>
<td>IVPOR</td>
<td>IOLGPOR</td>
</tr>
<tr>
<td>Retard. coeff.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>11</td>
<td>CRDM</td>
<td>STCRD</td>
<td>IVCRD</td>
<td>IOLGCRD</td>
</tr>
<tr>
<td>First order reac.</td>
<td>12</td>
<td>XLAMM</td>
<td>STLAM</td>
<td>IVLAM</td>
<td>IOLGLAM</td>
</tr>
<tr>
<td>External conc.&lt;sup&gt;1&lt;/sup&gt;</td>
<td>13</td>
<td>COEM</td>
<td>STCOE</td>
<td>IVCOE</td>
<td>IOLGCOE</td>
</tr>
<tr>
<td>External conc.&lt;sup&gt;2&lt;/sup&gt;</td>
<td>13</td>
<td>COEM</td>
<td>STCOE</td>
<td>IVCOE</td>
<td>IOLGCOE</td>
</tr>
<tr>
<td>Generic flow&lt;sup&gt;3&lt;/sup&gt;</td>
<td>14</td>
<td>PRGM</td>
<td>STPRG</td>
<td>IVPRG</td>
<td>IOLGPFRG</td>
</tr>
<tr>
<td>Generic transport&lt;sup&gt;3&lt;/sup&gt;</td>
<td>14</td>
<td>PRGM</td>
<td>STPRG</td>
<td>IVPRG</td>
<td>IOLGPFRG</td>
</tr>
</tbody>
</table>

<sup>1</sup> Parameter defined by elements.  <sup>2</sup> Parameter defined by nodes.  <sup>3</sup> Parameters associated to non-linear functions.
The estimation of a given parameter is controlled by variables IVXXX and IOLGXXX (where XXX is a root name). These variables appear in table 4.2 on fourth and fifth columns, respectively. IVXXX controls if a given parameter will be estimated. If IVXXX(I) ≠ 0, then the component I of the parameter type XXX will be estimated. IOLGXXX controls if we handle the parameters logarithmically or not, if it is set equal to 1 all the parameters of type XXX will be handled logarithmically. For those parameters, the code will work with their logarithm during estimation process, although the code will print the parameters without transformation at the end.

4.4 VERIFICATION OF DISCRETIZATION

Verification refers to the process of ensuring that flow and transport equations are being solved with sufficient accuracy. This involves “code verification”, which refers to ensuring that adequate numerical methods are used and that they have been programmed without errors. TRANSIN has been extensively tested, so that we are pretty sure that most errors have been corrected (obviously, no code is error-free and if the user discovers some error, we would appreciate that he let us know). Regarding numerical methods, we are using a quite standard finite element method with weighted finite differences in time. This method has been extensively applied by many modelers and it is known to produce robust and fairly accurate results, provided that an adequate space and time discretization have been specified. Therefore, this section is devoted to the process of testing that grid input is error-free (4.4.1) and fine enough (4.4.2).

4.4.1 Input errors

Considering that grid definition is quite tedious, one should always assume that, even if an automatic grid generator has been used, input errors have been made. Most input errors can be identified by plotting the F.E. grid, preferably using a code that reads grid definition from files identical or very similar to TRANSIN input files. The plot (actually, several plots ought to be done) should not be restricted to the F.E. grid, but should also include areal zones (transmissivity, storage coefficient, porosity, dispersivity, etc) and nodal zones (prescribed head, prescribed flow, mixed condition boundaries, and sinks and sources). In fact, though more difficult, it is also convenient to plot element and nodal coefficients (CFXXX).

Plotting the grid will help to identify some obvious geometrical errors (such as errors 1 and 4 in Figure 4.4), but may not identify topological errors (such as 2 or 3 in Figure 4.4).

Counting nodal connections (NECON) may help for this purpose.
a) Correct grid
\[ \text{NUMNP} = 9 \]
\[ \text{NUMEL} = 8 \]
\[ \text{NECON} = 16 \]

b) Error 1: Giving node 5, instead of 7,
\[ \text{NUMNP} = 9 \]
\[ \text{NUMEL} = 8 \]
\[ \text{NECON} = 18 \]

c) Error 2: Element 6 is not well defined
\[ \text{NUMNP} = 9 \]
\[ \text{NUMEL} = 7 \]
\[ \text{NECON} = 16 \]

d) Error 3: One node belongs to none of the elements
\[ \text{NUMNP} = 10 \]
\[ \text{NUMEL} = 8 \]
\[ \text{NECON} = 16 \]

e) Error 4: Error in x coordinate of node 4
\[ \text{NUMNP} = 9 \]
\[ \text{NUMEL} = 8 \]
\[ \text{NECON} = 16 \]

Figure 4.4: Most frequent errors in F.E. grid definition.
4.4.2 One, two or quasi-three dimensional grids

For convenience, we will distinguish two types of connections; horizontal connections (connections belonging to triangular and quadrilateral elements) and vertical connections (connections belonging to 1-D elements). This nomenclature is employed, because in most cases 1-D elements are used to define connections between layers. Notice, however, that horizontal connections do not need to be actually horizontal and vertical connections may not be vertical. In order to check for topological errors, one should count the following variables:

\[\text{NUMNP} \quad \text{Number of nodes.}\]
\[\text{NUMEL} \quad \text{Number of triangular elements.}\]
\[\text{NECON} \quad \text{Number of horizontal connections.}\]
\[\text{NHOLES} \quad \text{Number of internal holes in the grid (normally, 0).}\]
\[\text{NCEXT} \quad \text{Number of connections defining the boundaries (both external and internal).}\]
\[\text{NGRIDS} \quad \text{Number of isolated grids (normally, 1).}\]

These variables must satisfy the following relationships.

\[\text{NECON} = \text{NUMNP} + \text{NUMEL} + \text{NHOLES} - \text{NGRIDS}\]  \hspace{1cm} (4.23)
\[\text{NECON} = (3 \times \text{NUMEL} + \text{NCEXT})/2\]  \hspace{1cm} (4.24)

If these conditions are not satisfied, then input data contains topological errors. Actually, counting the connections can be so tedious that what one often does is to compute \(\text{NECON}\) using (4.15) and check it using (4.16), or vice versa. Both equations should lead to the same value of \(\text{NECON}\).

Some of the most frequent input errors are displayed in Figure 4.4. Error 1 occurs almost inevitably when element definition is entered by hand. It consists of typing an erroneous node number when defining elements. Notice that the actual number of connections is 18, while applying (4.15) or (4.16) to the assumed grid (grid a in Figure 4.4) would lead to the number of corrections being 16. Obviously, careful inspection of the grid plot should allow easy identification of the error.

The second type of error in Figure 4.4 is also quite common when elements are defined by hand. Moreover, it may not be easy to identify by visual inspection because the look of the grid is not altered if one element is omitted. However, the number of connections computed with (4.15) or (4.16) and using \(\text{NHOLES} = 0\) and \(\text{NCEXT} = 8\) (values assumed by the user) would be 15 and 14.5(!), respectively. They are different from each other. Moreover, the second value is absurd. Notice that using \(\text{NHOLES} = 1\) and \(\text{NCEXT} = 11\) in equations (4.15) and (4.16) would lead to the right value, but this would imply that the user was inserting a hole in the grid on purpose. Identifying the missing element can be achieved by colouring the grid element by element, so that the missing one would be the one left uncoloured.
Error 3 in Figure 4.4, not unusual when gridding by hand (hence the advantage of F.D. or grid generators), consists in forgetting one node when defining the elements. In this case, equation (4.15) would lead to \( NECON = 17 \) while equation (4.14) would lead to 16.5 if \( NCEXT = 9 \) or to 16 if \( NCEXT = 8 \). Plotting node numbers along with the grid will help in pointing to where the error is taking place.

Finally, error 4 in Figure 4.4 is a purely geometrical error, possibly caused when digitizing nodal coordinates. This error may negatively affect the quality of results. TRANSIN will simply run with a geometry somewhat different from the one assumed.

4.4.3 Discretization size

The methodology of section 4.4.1 helps in defining error-free grids, but it does not ensure adequate precision. Truncation errors arise as a result of space and time discretization. These are reduced, at the cost of increased CPU time, by refining both the element and solution time increments. Hence, the appropriate level of refinement is based on accuracy-cost trade-offs, which implies that both CPU time availability and accuracy should be kept in mind.

CPU time requirements are discussed in Section 4.7. For now, let us simply state that making a preliminary run in simulation mode may be convenient both for grid verification and to obtain an estimate of CPU time. Also, it can reveal convergence problems for solving the direct problem, when it is non-linear. This case is discussed in section 4.9.4.1. Depending on the number of parameters and stability of the problem, total CPU time required for parameter estimation may range between 100 (few parameters) and 1000 (more than 100 parameters) times that needed for simulation.

The simplest approach for evaluating truncation errors of a given space-time discretization is by comparison with analytical solutions. In most practical problems with complex geometries no such analytical solution exists. However, in some cases, problem set-up can often be slightly modified so that comparison with analytical solutions is possible. For example, in the case of linear problems, if one is modeling the response to pumping or injection, it is possible to modify model parameters to make the aquifer isotropic and homogeneous, so that the numerical solution can be compared to Theis analytical solution. This would involve the following steps:

a) Assume that the aquifer is isotropic and homogeneous with constant transmissivity and storativity throughout the flow domain. Choose values for these parameters that are representative of average real values.

b) Choose a pumping well and a constant rate of pumping that is representative of pumping rates to be used to stress the aquifer. Choose two or three observation wells in the vicinity of the pumping well.

c) Choose a typical duration of pumping and appropriate observation intervals.

d) With a chosen space and time discretization (see Section 4.2), run the program in simulation mode assuming zero initial heads (IOTRS=1) and compute the
drawdowns at the observation wells at various observation times.

e) Compare results with Theis solution to see if the discretization error is acceptable. Errors in the TRANSIN solution compared to the “Theis” solution may be due to three possible causes:

i) Grid element density (shape, size and distribution of elements) is not adequate.

ii) Time discretization for numerical integration of equations is not adequate.

iii) Boundary effects may be affecting the solution. These should only affect late time data.

The easiest is to first try to improve the solution by using a finer integration time step. Increasing the density of elements near the wells is obviously a more tedious task. The easiest way to check for boundary effects is to plot head contours (using computed heads at all the nodes) to see if they remain circular. WARNING: This test may be unduly demanding if the main stresses are not caused by pumping.

Comparison with analytical solutions in nonlinear cases is much more difficult because these solutions are scarce and restrictive. Anyway, because this procedure is not best for testing the discretization error, we generally avoid comparison with analytical solutions for non-linear problems.

In fact, comparison to analytical solutions is an incomplete test. That is, it allows us to discard inadequate grids, but it does not ensure that the grid is adequate for the actual problem. In the Theis example above, the fact that small errors occur in the homogeneous problem does not grant equally good results in the heterogeneous case. If fact, narrow high transmissivity zones often require much finer grids than indicated by homogeneous simulation.

The best way of testing accuracy is by refining both element size and time discretization. This can be achieved as follows:

\textbf{Step 1:} Start with the time discretization derived in Section 4.2.3 and refine it by increasing the number of solution increments per observation interval (KINT in Card 27). For consistency of comparison, it is convenient to increase KINT by a constant factor (e.g., multiply it by two for every simulation run). Furthermore, the desirable time increment (DTDSR) and maximum desirable time increment (DTMXDSR?) variables have to be changed according to the value of KINT. In this sense it is important to keep in mind that for linear problems and those weakly non-linear, it is suitable to verify at all observation intervals, \( \alpha \), that:

\[
\frac{\text{TIME}(I+1) - \text{TIME}(I)}{\text{KINT}} = \text{DTMXDSR}(I) = \text{DTDSR}(J) \quad (4.25)
\]

When the differences in results between successive simulations are below an acceptable value, further refinement is probably not be worth the added computational effort, and one should stop refining the time step at this point. In
comparing solutions with different levels of time discretization, one should make sure that no new errors are introduced in the process. For example, if time functions are used for flow rates with a fully implicit method (THETA_F=1), variations of KINT will lead to variations in the volume of water pumped or injected into the model (recall Figure 4.3). Since these variations can be easily corrected by modifying the time function value, it may not be worthwhile paying much attention to them. Instead, constant or step-wise constant functions of time can be used to reduce or eliminate this variability during the time step identification process.

On the other hand, for non-linear problems, a qualitative indicator of the quality of the discretization, not only in time but also in space, is the number of time-step reductions and as well the average number of iterations required for reaching convergence in a time step. As empirical criteria, we would say that a number between 4 and 10 iterations per time step and no time reductions during the solutions indicates suitable discretization. Solutions obtained with more than 10 iterations and several reductions of time contain, surely, unacceptable numerical errors. Those reached with 3 or less could have an overrefined discretization.

**Step 2:** Refine the space discretization and follow the same approach as above (i.e., reduce element sizes by a constant factor until a sufficiently small difference between results for successive levels of refinement is observed). Ideally, an appropriate level of time refinement should be found for every spatial discretization because the appropriate solution time increment depends on element size. This can be done by repeating step 1 for every grid. Unfortunately, doing so may become exceedingly tedious, so that one may simply reduce time increments by a factor equal to the square of the grid size reduction factor (e.g., if nodal distances are being reduced by a 1.4 factor, reduce time increments by a factor of 2).

Step 2 can become very tedious if one is not using an automatic grid generator. Moreover, one may have to stop refining the grid if the simulation CPU time suggests that inverse problem runs would become nearly impossible (for example, a problem whose simulation needs 1 hour of CPU may lead to an inverse problem requiring no less than 5 days of CPU).

The last paragraph brings up the issues of how much numerical errors affect estimated parameters. Ideally, one might evaluate the effect of numerical errors by simply solving the inverse problem with increasing levels of refinement. As a rule of thumb, unstable problems may be sensitive to truncation errors, while well posed problems will not be very sensitive to such errors.

If the physical problem is linear, or it doesn’t contain dramatic slope changes in its time function which increase CPU time, being expensive reaching the solution, the process of selecting a discretization is completed by developing steps 1 and 2. Otherwise, it is desirable to carry out an additional step.

**Step 3:** Once found a suitable discretization examine additional time discretization alternatives for each observation interval (J). This consists of choosing a maximum desirable time increment greater than that defined based on KINT; which means:

\[
\frac{\text{TIME}(J+1) - \text{TIME}(J)}{\text{KINT}(J)} = \frac{\text{DTDS}(J)}{\text{DTMXDS}(J)}
\]
One can try to use $n$ between 1 and 5 depending on the length of the interval. The ideal is to choose the maximum $n$ that leads to an adequate solution. It is important to remember two aspects at this stage:

I- DTDS(J) is the first time increment used by the code after having obtained the solution at TIME(J) and, if well defined (see sec. 4.2.3.1) and no convergence problems appear, DTMXDS is the time increment used for computing solution at observation time $j + 1$. Then, the intermediate solutions will be obtained by using time increments varying between those two extremes values.

II- $n$ is in general, a different number for each $J$.

As one could suspect, step 3 is not less tedious than 1 and 2. Nevertheless, it is particularly useful when the desirable time increment selected based on steps 1 and 2 is smaller than the length of the observation interval.

As a final comment on this section, we want to point out that, what the modeler looks for are the parameters corresponding to the last iteration of the inverse problem. That means that although all previous set of parameters (those obtained in previous iterations of the inverse problem), excepting the last (supposing convergence has been reached) would be completely erroneous, it doesn’t matter. For that reason, and taking into account that, in non linear problems, the CPU time becomes a critical aspect, TRANSIN allows relaxation of the convergence criteria of the direct problem at early inverse iterations. The effect is that of obtaining inaccurate but useful numerical solutions for these iterations. It could seem contradictory with the process described previously, where one searches for a suitable discretization. Nevertheless, it is not because an adequate discretization and an adequate convergence criteria are required only in the last iteration of the inverse problem. In this sense, if the convergence criteria variations are well defined, a good numerical solution and a good set of parameter values are guaranteed by TRANSIN, at least, at the last inverse problem iteration. Selection of the convergence criteria is the subject of the next Section.

4.4.4 Defining direct problem convergence criteria

An appropriate selection of the convergence criteria is as important as defining the discretization. In this sense, an excessively relaxed criteria leads to an erroneous solution of the direct problem. Sometimes, moreover, the numerical process of solution degenerates until it is incapable of reaching convergence. On the other hand, excessively severe criteria demand many unnecessary iterations. Even if the restrictions described are clear, there are no formal rules for getting a compromise between accuracy and time savings. In the rest of section, we present some general criteria which one should respect in order to obtain an acceptably accurate solution.

Let us back up to equation (2.13). It represents the mass balance of water (or solute) in the medium. Since solution of (2.13) is obtained by means of a numerical process during which the value of $h$ is changing, this equation is never satisfied completely. That means:

$$ F(h) = \varepsilon_1 $$

where $\varepsilon$ is a vector of mass balance errors. Specifically, $\varepsilon_i$ is the error in the mass balance...
over the volume corresponding to node \( i \). As first approximation, one could compute the maximum \( \varepsilon \) permitted as:

\[
|\varepsilon_1|_i = \frac{V_m \cdot F}{n \cdot t}
\]

(4.27)

where \( V_m \) is an approximating of the amount of water entering or leaving the medium during the simulation. Since the product \( V_m \cdot F \) is the total water one expects to lose or to gain due to mass balance errors, \( F \) is a factor close to zero. (we suggest \( 0.01 > F > 0.001 \)). On the other hand, \( n \) is the total number of nodes in the grid and \( t \) the total time simulation.

In (4.27) one assumes that errors are uniformly distributed, and that volumes associated to all nodes are identical. In general, this is not true, especially when boundary conditions are concentrated. In these cases, it is suitable to be conservative when choosing \( F \). Notice that \( V_m \) can be obtained by examining the total mass balance, issued by the code (if requested) at the end of the simulation. Preliminary runs, used for defining the discretization, can be used also for defining \( F \).

Another way for defining \( \varepsilon_1 \) is to constrain errors in the state variable. Thus, the average head error at a node in the grid, induced by selection of a given \( \varepsilon_1 \), \( \Delta h \), is:

\[
\Delta h = \frac{|\varepsilon_1|_i t}{S_i V_i}
\]

(4.28)

where \( S \) and \( V_i \) are respectively the average specific storage coefficient and volume corresponding to node \( i \). If \( S \) is very variable in space and time (i.e., unsaturated problems), application of (4.28) is not advisable.

Changes in the variable

As explained in Chapter 2, the other convergence criteria used by TRANSIN are:

\[
\varepsilon_2 > |h^{k+1,l+1} - h^{k+1,l}|_i
\]

(4.29)

\[
\varepsilon_3 > \left| \frac{h^{k+1,l+1} - h^{k+1,l}}{h^{k+1,l+1} - h^k} \right|
\]

(4.30)

Then the code shall consider convergence when (4.29) and (4.30) are verified. Notice that the first criterion is unit dependent, therefore it may be very variable, analogously to the mass balance criterion. One can select \( \varepsilon_2 \) as a fraction (10\(^{-3}\) and 10\(^{-5}\) are usually adequate) of the maximum variation of \( h \) expected in a node during a time step. In turn, \( \varepsilon_3 \) is just the function employed for computing \( \varepsilon_2 \).

### 4.4.5 Using changing convergence criteria

As explained in the section 4.4.2, TRANSIN offers the possibility of changing the direct problem convergence criteria between two inverse problem iterations. The idea is to relax the precision when the actual set of model parameters is notably different than that one is looking for. Since the latter is the unknown of the problem, the idea expressed before has to be applied indirectly. That is, it is known that erroneous parameter sets lead to
big values of the objective function (eq. 3.7). Therefore, we propose adopt convergence criteria dependent upon the objective function according to the function presented in Figure 4.5.

being $\varepsilon_{\text{min}}$, those criteria defined by use of rules given in the previous section. In turn, $\varepsilon_{\text{max}}$ is defined as:

$$\varepsilon_{\text{max}} = N \varepsilon_{\text{min}}$$

Where $N$ is a number varying between 10 and 1000, depending on the strictness of $\varepsilon_{\text{min}}$. Anyway, it is unavoidable to do preliminary runs for testing $N$. One must choose the biggest $N$ which leads to solutions without notable oscillations or deformed solutions. The number of Newton-Raphson iterations is indicator of how favorable a given $N$ is, with respect to $N$ equal to 1. If the rate between total direct iterations is more than 1.5 we consider it is desirable to use the changeable convergence criteria. Defining $J_{\text{min}}$ and $J_{\text{max}}$ requires studying the range of variation of the objective function. As first step a magnitude of the mean residual error, $h_i - h_i^*$, considered clearly unacceptable, has to be defined by the modeler. Then, one should compute the objective function based on (3.7) and adopt this value as $J_{\text{max}}$. Afterwards, one must also define a value $h_i - h_i^*$ considered a good fit (in other words, if all errors of the observation data had this value one would have found the solution of the inverse problem). Now, we use that value for compute the objective function and adopt it as $J_{\text{min}}$. 

Figure 4.5: Relationship between values of the convergence criteria of the direct problem and magnitude of the objective function. It is proposed in order to save computations during the inverse problem solution where non-linear problems are solved.
4.5 SPECIFICATION OF OBSERVED DATA

4.5.1 General description

Measured (observed) heads and concentrations are the most important pieces of information to be entered in the program for parameter estimation. In previous versions of the code, the definition of observations has been limited to a value of head or concentration at one point in space and time. In TRANSIN IV the definition of the spatial and temporal observations has been extended, to allow a wide variety of possible measurements. It includes observations which are calculated as spatial and/or temporal integrals of heads or concentrations, as well of point spatial and temporal as before. This enables the use of data such as time integrated leakage to/from a river/lake or temporal averages of heads/concentrations in a certain area.

In previous versions, a value \( v \) (head or concentration) corresponding to an observation was calculated as:

\[
v = \sum_{n=1}^{N_n} \xi_n v_n \tag{4.31}
\]

where \( N_n \) is the number of nodes defining the element in which the observation point is located, \( v_n \) and \( \xi_n \) are head or concentration and interpolation function values at the nodes of this element, respectively.

With the new implementation, \( v \) is still calculated as a weighted summation of nodal values. However, to allow the use of observations which are not point values, the new general expression for \( v \) is computed as:

\[
v = \sum_{t=1}^{N_t} \sum_{u=1}^{N_u} W_u \sum_{bu=1}^{N_{bu,u}} W_{bu,u} \sum_{n=1}^{N_{n,bu,u}} \xi_{n,bu,u} v_{n,bu,u} \tag{4.32}
\]

where \( N_t \) is the number of integration times, \( N_u \) is the number of units constituting device, \( N_{bu,u} \) is the number of basic units constituting unit, \( N_{n,bu} \) is the number of nodes constituting basic unit, \( t \) is the integration time index, \( u \) is the unit index, \( bu \) is the basic unit index, \( W_u \) is the unit weight and \( W_{bu,u} \) is the basic unit weight.

Equation (4.32) is the basic equation for measured data in TRANSIN-IV. The meaning of some of the terms is explained in next sections. First, the ability for defining spatial integration will be explained. Then, the flexibility of including temporal integration data will be presented. However, the first step will be the definition of several basic terms that are used in the context of TRANSIN-IV observations: “device”, “datatype”, “unit” and “basic unit”.

4.5.2 Definition of devices and their spatial extension

- **Device.** A device most often means an instrument which is used to measure something. In the terms of TRANSIN-IV observations this meaning is slightly modified. A device is defined as a group of measurements of one data type (see below) carried out at one spatial location.
• **Data Type.** A data type is one of the two state variables, heads or concentrations, or any quantity (e.g. flux) that can be expressed as a function of these two state variables.

• **Unit.** A unit is a spatial entity (point, line, surface or volume). The spatial extension of a device is defined by one or more units. Table 4.3 shows the different unit types and the corresponding values of the variable IOUTYP (the Unit TYPe).

• **Basic Unit.** A basic unit is a spatial entity of the model (point, node, element or zone). The spatial extension of a unit (real world) is defined through basic units (model world). Table 4.3 shows the different basic unit types and the corresponding values of the variable IOBUTYP (the Basic Unit TYPe).

<table>
<thead>
<tr>
<th>IOUTYP</th>
<th>Unit Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Points</td>
</tr>
<tr>
<td>2</td>
<td>Lines</td>
</tr>
<tr>
<td>3</td>
<td>Surfaces (not operative in this version)</td>
</tr>
<tr>
<td>4</td>
<td>Volumes (not operative in this version)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IOBUTYP</th>
<th>Basic Unit Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Points</td>
</tr>
<tr>
<td>2</td>
<td>Nodes</td>
</tr>
<tr>
<td>3</td>
<td>Elements (not operative in this version)</td>
</tr>
<tr>
<td>4</td>
<td>Zones (not operative in this version)</td>
</tr>
</tbody>
</table>

**Examples of devices**

The following are examples of datatypes, unit types and basic unit types which are used to define three devices:

a) Head measurements at one well●Datatype: Head●Unit: Point defining well●Basic unit: Point defining well (in this case, unit and basic unit are identical)

It must be pointed that if at the same well there are several measurements at different deeps separated by packers, each one should be a different device. It is probably the most usual case that the well is represented by a point, however if the well is represented is the model as a segment, for instance, it can be also accomodated in TRANSIN-IV.

b) Average concentration in a supply system●Datatype: Concentration●Units: Subsets of wells geographically distributed●Basic unit type: Points defining wells of a given subset

c) Flow in a river●Datatype: Flow●Units: Lines defining each one of the streams●Basic units: Points defining each one of the lines
4.5.3 Methods for weighted summation over space

Since all weights $\xi_n$, $W_{bu}$ and $W_u$ in (4.32) depend only on the spatial structure of the device, they can be determined before any calculation of heads or concentrations are carried out. However, the weights depend on whether the observation is a spatial integration, a spatial average, a simple summation of unit values or a simple average of unit values. This is managed by IOCALTYP variable. Its allowed values and the corresponding method of weighted summation of units are listed in Table 4.4.

Table 4.4. Description of variable IOCALTYP

<table>
<thead>
<tr>
<th>IOCALTYP</th>
<th>Method of weighted summation of units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One point</td>
</tr>
<tr>
<td>2</td>
<td>Spatial integration</td>
</tr>
<tr>
<td>3</td>
<td>Spatial average</td>
</tr>
<tr>
<td>4</td>
<td>Simple summation (of spatial average of each unit) (not operative in this version)</td>
</tr>
<tr>
<td>5</td>
<td>Simple average (of spatial average of each unit) (not operative in this version)</td>
</tr>
<tr>
<td>6</td>
<td>User defined weights</td>
</tr>
</tbody>
</table>

**IOCALTYP=1.** The same method used in previous versions of TRANSIN is applied. The computed value of the measurement is calculated using equation (4.31).

**IOCALTYP=2.** Spatial integration over the units is carried out. As spatial integration cannot be carried out over one point, this method is only applicable if the units are not points, that is, when IOUTYP≠1. The value $v$ is determined as

$$v = \int \frac{Z \ast dV_{tot}}{V_{tot}} \approx \sum_{u=1}^{N_u} EXT_u \ast Z_u \approx \sum_{u=1}^{N_u} \sum_{bu=1}^{N_{bu,u}} EXT_{bu,u} \ast Z_{bu,u} \quad (4.33)$$

where $Z$ is the spatially distributed value of the current datatype. $Z_u$ is the spatial average of the data type over the unit whereas $Z_{bu}$ is the basic unit value of the current datatype. $EXT_u$ and $EXT_{bu}$ are the spatial extensions (1D: length, 2D: area, 3D: volume) of units and basic units, respectively. In case the basic units do not have any extension (IOBUTYP=1 or IOBUTYP=2, i.e., points), $EXT_{bu}$ are interpreted as the fraction of the unit extension which is best described by the basic unit. For instance, if a line (unit) is to be described by two points (basic units), $EXT_{bu}$ is the half length of the line for each point. This option has been only implemented only for lines defined by points or nodes (IOUTYP=2 and IOBUYTP=1 or IOBUYTP=2).

**IOCALTYP=3.** Spatial averaging over the units is carried out. As for spatial integration, spatial averaging is obviously only applicable in the units are not points, that is for options IOUTYP≠1. The value $v$ is computed as:

$$v = \frac{\int \frac{Z \ast dV_{tot}}{V_{tot}}}{\int \frac{dV_{tot}}{V_{tot}}} \approx \frac{\sum_{u=1}^{N_u} EXT_u \ast Z_u}{\sum_{u=1}^{N_u} EXT_u} \approx \frac{\sum_{u=1}^{N_u} \sum_{bu=1}^{N_{bu,u}} EXT_{bu,u} \ast Z_{bu,u}}{\sum_{u=1}^{N_u} \sum_{bu=1}^{N_{bu,u}} EXT_{bu,u}} \quad (4.34)$$

As for spatial integration, spatial averaging has been only implemented for lines defined by points or nodes.
**IOCALTYP=6.** The user defines weights $W_u$.

The way in which $v$ is determined depends on the value of the variable IOCALBU (CALculation type for Basic Unit) which determines how a basic unit value has to be calculated. The allowed values of this variable and the corresponding method of weighted summation of basic units are listed in Table 4.5. Furthermore, allowed values of IOUTYP and IOCALTYP are also shown.

**Table 4.5** Allowed relations between variables IOCALTYP and IOUTYP (above). Description of variable IOCALBU (below).

<table>
<thead>
<tr>
<th>IOCALTYP</th>
<th>Calculation type</th>
<th>IOUTYP=1 (POINT)</th>
<th>IOUTYP=2 (LINE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One point</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>2</td>
<td>Spatial integration</td>
<td>NO SENSE</td>
<td>OK</td>
</tr>
<tr>
<td>3</td>
<td>Spatial average</td>
<td>NO SENSE</td>
<td>OK</td>
</tr>
<tr>
<td>6</td>
<td>User defined weights</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

**IOCALBU=2.** Spatial integration over basic units. The measured value computed by TRANSIN, $v$, is given by:

$$v = \sum_{u=1}^{N_u} W_u \int_{V_u} Z_{bu,u}dV_u \approx \sum_{u=1}^{N_u} W_u \sum_{bu=1}^{N_{bu,u}} EXT_{bu,u} * Z_{bu,u} \quad (4.35)$$

**IOCALBU=3.** Spatial integration over basic units. In this case, the measured value computed by TRANSIN, $v$, is given by:

$$v = \sum_{u=1}^{N_u} \frac{\int_{V_u} Z_{bu,u}dV_u}{\int_{V_u} dV_u} \approx \sum_{u=1}^{N_u} W_u \frac{\sum_{bu=1}^{N_{bu,u}} EXT_{bu,u} * Z_{bu,u}}{\sum_{bu=1}^{N_{bu,u}} EXT_{bu,u}} \quad (4.36)$$

**IOCALBU=6.** In this case, the measured value computed by TRANSIN, $v$, is given by:

$$V = \sum_{u=1}^{N_u} W_u \sum_{bu=1}^{N_{bu,u}} W_{bu,u} Z_{bu,u} \quad (4.37)$$

The terms appearing in these equations have been described previously.
4.5.4 Methods of weighted temporal summation

In TRANSIN-IV, observations that are described by a time interval or that are weighted sums of observations at different times can be included in the objective function calculations. The method which is used to calculate the measured value of TRANSIN is defined by the variable IOINTTYP (INTerval TYPe). The accommodated valued and their meaning are listed in Table 4.6.

<table>
<thead>
<tr>
<th>IOINTTYP</th>
<th>Method of weighted temporal summation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Observations at a point in time</td>
</tr>
<tr>
<td>1</td>
<td>Integration or averaging by use of all simulation times between beginning and end of the interval</td>
</tr>
<tr>
<td>2</td>
<td>Integration or averaging by use of a number of equally distributed times (NUMTIM) which include the interval ends</td>
</tr>
</tbody>
</table>

Table yyy.4 Description of variable IOINTTYP

The measured value computed by TRANSIN is calculated in all cases, as:

\[ v = \sum_{t=1}^{N_t} W_t V_t \]  (4.38)

where \( V_t \) is the result of the spatial integration for simulation time number \( t \). What changes with IOINTTYP value is the way \( V_t \) is computed.

IOINTTYP=0. This option is used for the simple and most common case where the observation is defined as a point in time (default option). In this case \( W_t = 1.0 \), so \( V = V_t \).

IOINTTYP=1 or IOINTTYP=2. In this case, observation covers a time interval. Then, TRANSIN’s measured value is determined by:

\[ V = \frac{t_2 - t_1}{2} v_1 + \sum_{t=1}^{N_t-1} \frac{t_{i+1} - t_i}{2} v_i + \frac{t_{N_t} - t_{N_t-1}}{2} v_{N_t} \]  (4.39)

Notice that temporal averaging over an interval is different from simple averaging: the values at the interval ends are weighted differently.

4.5.5 Error structure of measurements

One must give the program a matrix with the error structure of measurements (section 3.2.2). However, this is not always straightforward. Some hints are given in this section. Many times it is very difficult to have full matrices errors, and it is usual to use them as diagonal (only standard deviation of error measurements taken into account).

In the absence of any indication to the opposite, standard deviations should be equal for all observation points. One way of assigning standard deviations is to subjectively
divide defined 95% confidence interval (Notice that this confidence intervals into 4 equal intervals should include not only measurement errors, but also model errors that cannot be attributed to the values of model parameters). Yet, a number of effects may lead to variable standard deviations. These are listed below.

**Unmodeled heterogeneity:** TRANSIN treats transmissivity (or conductivity) as constant by zones (actually, it can vary in a prescribed manner within each zone). However, true transmissivity is likely to vary randomly in space. This leads to a head uncertainty whose variance is proportional to the head gradient, log-T variance and log-T correlation distance (or zone size, whichever smaller). In practice, this effect can be significant only when the sizes of T-zones are very different from each other and when head gradients vary significantly across the flow field.

**Other unmodeled factors.** Some hydrogeologic effects are known to exist, but are not included in the model because of simplicity or because they are not known in sufficient detail. Examples of these include small pumpages, vertical head variations in 2-D models, etc. If they significantly affect some measurements, other model parameters may be displaced from their best values when striving to fit such measurements. The best way to minimize the negative effects of unmodeled factors is to reduce the weight (increase standard deviation) assigned to affected points.

**Numerical errors.** An idea of the size of numerical errors should be obtained when defining the finite element grid. These errors may vary in space as a consequence of varying grid size.

**Unpredictable measurement errors.** A large number of factors may affect measured data. Sometimes, random errors occur during typing, or because of errors in well head elevation, or because an unrecorded pumping test has been performed shortly before head measurement. Many of these errors can be corrected by careful inspection of data (drawing of contour lines, plotting head and concentration versus time curves, etc). However, a number of them may go on unnoticed. Hence, one should examine the contribution of each observation point to the objective function after running TRANSIN. Quite often, a large percentage of the head and concentration objective function can be explained by one or two points. Analyzing their data in detail may lead to identifying data errors. Otherwise, one may choose to reduce the weight given to data in order to minimize the negative effects of unidentified errors. This operation should be performed with extreme care, because if the abnormal data are indeed real, they may point to weaknesses of the conceptual model.

**Warning.** TRANSIN is able to use saturation degree as a measure. It is important to keep in mind two aspects when using this alternative. First, the range of this state variable is 0-1; therefore the range of the objective function in this case is very different than when using pressure head. Second, because the unsaturated flow equation is solved in terms of suctions, estimating the retention curve based on saturation measures is an inconsistent procedure.
4.6 INPUT FILE PREPARATION

The input files can be set up to run simulations, estimate parameter values or drive parameter sensitivity analyses.

Input data consists of up to the following six files:

- DIMension file. It contains all problem dimensions, options and objective function and convergence parameters.
- GRId file. It contains grid definition: Nodes, elements, boundary conditions and parameters.
- PARameter and coefficient file. It contains nodal and element coefficients as well as parameter definitions: (zonal values, estimation options, standard deviation, prior information, generic parameters and non linear function topology).
- TIMe file: It contains time discretization, time functions and time weighting parameters.
- OBServation file: It contains observed data and related options.
- INItial data file: It contains initial heads and concentrations as well as Darcy’s velocity and boundary flows if flow equation is not being solved.

The first three files must be defined in any problem. The filename for the dimension file is given interactively by the user. The remaining filenames are written in this file or they are created with a root plus a suffix. If the given filename for the dimension file does not contain a dot, “.”, the dimension filename will be understood as the input string (root) plus the suffix DIM.DAT (be careful with uppercase suffix in UNIX machines). The rest of filenames will be:

root + GRI.DAT (mesh file)
root + PAR.DAT (coefficients file)
root + TIM.DAT (time file)
root + OBS.DAT (observations file)
root + INI.DAT (initial data file)

It should be noticed that several problems may be solved in one TRANSIN run. This requires defining group A11 (DIM file) and may be motivated by the following purposes:

- Make a new run with different initial parameters. This is advisable to check uniqueness (see sections 3.7 and 4.9).
• Make a new run with different convergence parameters and using the last computed parameters as initial ones. It may be interesting to solve some convergence problems. Change the weighting coefficients of prior information ($\lambda_T$, $\lambda_S$, $\lambda_\phi$, etc). This may be required for identifying the optimal values of these coefficients. Moreover, convergence may be helped by solving a sequence of problems, starting with large weighting parameters and reducing them sequentially (see Section 4.9.4).

• Make a new run with new direct or inverse convergence parameters and new starting parameters. It may be useful in analyzing uniqueness.

In order to help in the understanding of preparation of main input files, some examples are presented in Chapter 5. Some of the corresponding input files are shown in Appendix 4. In addition, there are model files for each input file in Appendix 6. These are void files (only containing comments) that one can fill with the variable values. Under every line one can write the corresponding value of the variable. The names are shortened in some cases (do not coincide with the complete names that you may find in Appendix 1).

### 4.7 BRIEF DESCRIPTION OF OUTPUT

The output from TRANSIN consists of a main output file (Root + RES.OUT) and several secondary output files which can be used as inputs for postprocessing programs (Table 4.3).

- root + PLT.OUT: It contains computed and measured heads and concentrations time evolution at every observation point.

- root + MHH.OUT: It contains computed heads at every node at some observation times (specified by variable IOMHH in Card A6.1), together with head contour values as defined in group Card A10.

- root + MCC.OUT: It is similar to the previous one, for computed concentration.

The program first prints all input and interpolated data, as well as warning messages regarding dubious logic in input data. During preliminary runs, the user should always search the output file for “ERROR” and “WARNING” messages. “ERROR” messages will always stop execution.

At each iteration, the values of objective function, gradient, etc, are printed. As well as the Newton-Raphson process evolution for each time step, if required, according to the inverse iteration. Once the inverse iterative process has been finished, the program prints a statistical summary containing statistics on head and concentration residuals, objective function values, likelihood function, etc.

Following the statistical summary, computed parameters, their difference with prior estimates and the gradient of the objective function with respect to each of the unknown parameters are printed. After that, computed heads and head residuals are written at each observation point and time. Mass balance computations may be printed if so desired, as controlled by variables defined in Card A7.1. This is explained in next subsection.
Table 4.3: Output files

<table>
<thead>
<tr>
<th>File suffix</th>
<th>Controlling Variable</th>
<th>Type of information</th>
<th>Example</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>RES.OUT</td>
<td>-</td>
<td>Main output</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IOPLH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLT.OUT</td>
<td>IOPLH</td>
<td>Heads and/or conc.</td>
<td>1</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>IOPLC</td>
<td>vs time</td>
<td>2</td>
<td>5.8,5.10,5.11,5.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>5.23,5.24</td>
</tr>
<tr>
<td>MSH.OUT</td>
<td>IOMHH</td>
<td>Grid + flow zones</td>
<td>1</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MHH.OUT</td>
<td>IOMHH</td>
<td>Nodal heads at obs. times</td>
<td>1</td>
<td>5.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSC.OUT</td>
<td>IOMCC</td>
<td>Grid + Trpt. zones</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC.OUT</td>
<td>IOMCC</td>
<td>Nodal concent. at obs. times</td>
<td>1</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVM.OUT</td>
<td>IOCVM</td>
<td>Computed vs measured data</td>
<td>1</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>IOSEH</td>
<td>Sectional pictures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEC.OUT</td>
<td>IOSEC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Filenames are of the form Root + Suffix
2 Files displayed in Appendix IV
3 Figures produced with the corresponding file
Table 4.4: Scratch run time files

<table>
<thead>
<tr>
<th>Unit</th>
<th>Name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>94</td>
<td>F1H.DAT</td>
<td>Nodal heads printed each NTT observation times, where NTT=min (IOMHH,IOWRH). Notice that one (say IOMHH) has to be a multiple of the other (say IOWRH, or viceversa).</td>
</tr>
<tr>
<td>81</td>
<td>F2H.DAT</td>
<td>Heads at all observation points and every computation time.</td>
</tr>
<tr>
<td>83</td>
<td>F3H.DAT</td>
<td>Heads at all nodes and at some observation or computation times, depending on the values of variables IOBALH and IOBALGH.</td>
</tr>
<tr>
<td>93</td>
<td>F1C.DAT</td>
<td>Nodal concentrations printed each NTT observation times, where NTT=min (IOMHC,IOWRC). Notice that one (say IOMHC) has to be a multiple of the other (say IOWRC, or viceversa).</td>
</tr>
<tr>
<td>82</td>
<td>F2C.DAT</td>
<td>Concentrations at all observation points and every computation time.</td>
</tr>
<tr>
<td>84</td>
<td>F3C.DAT</td>
<td>Concentrations at all nodes and at some observation or computation times, depending on the values of variables IOBALC and IOBALGC.</td>
</tr>
<tr>
<td>85</td>
<td>ITE.SCR</td>
<td>Model parameters history through all iterations.</td>
</tr>
</tbody>
</table>

In the error analysis part of the program, the covariance matrix of computed parameters is printed. Variances of computed heads and concentrations are also printed. Finally, a principal components analysis of the covariance matrix, consisting of eigenvalues and eigenvectors, is also printed.

Additional details on output meaning can be found in chapter 5.

In addition to the output files employed by TRANSIN (Table 4.3), the code prints internal information in several scratch files that are deleted automatically at the end of each run. These are specified in Table 4.4.

4.7.1 Mass balance

Mass balance of flow and transport can be computed if the appropriate variables in group card A7 are activated. Mass balance computations are useful in order to check the amount of solute or water that moves through the aquifer. In addition, it allows us to find out some mistakes in the input data. For instance, we may see large outflows or inflows that are not possible from our knowledge of the aquifer. Furthermore, looking at mass balance may help us to improve our understanding of the system. As we have said previously, TRANSIN can manage mass balance computations for flow and transport. For each of them there are three different ways of obtaining the information (that can be activated simultaneously if desired):

**IOBALH:** This option allows displaying the flow status every IOBALH observation times. The balance is separated by zones of recharge, prescribed head, prescribed flow, leakage and storage. Storage shows the increment/decrement in water content in the model between this time and the previous one (if the storage is negative, a decrement in water content has been produced, and viceversa). The other zones represent the flow that has entered into or has left out the model (input or output flows, depending on the sign).
IOBALC: ¹Not operative at this version Same as IOBALH for solute transport. Solute mass variations are classified by porosity zones.

IOBALGH: Global mass balance. It is similar to the previous one, but it represents the flow integrated over time. It tells us the total amount of water that has left or entered into the model. It is also represented by zones.

IOBALGC: ¹Same as IOBALGH for solute transport.

IOBALDH: ¹Detailed flow mass balance. The same as IOBALH but instead of giving the information by zones, it gives the information by nodes. That is, it gives the same kind of information, but much more detailed. One should be careful when using this option, because very large files may be created. (Currently, this option is only available for steady-state)

4.8 RESTART PROCEDURE

In the framework of TRANSIN, the meaning of restart is somewhat different from “direct problem” programs. In our case it is not possible to change the type of boundary conditions, etc, beyond the possibilities that can be achieved by playing with the time functions. By restart, we mean the continuation of execution (or estimating parameters) under conditions related to the original problem, but somehow different.

Procedure 1 can be used to change initial values of parameters, stopping criteria, inverse iteration parameters, weighting coefficients, etc. It does not require stopping the execution. It is performed by repeating group cards A11 together with cards A5 to A10 or C4 to C16 as many times as required in the dimensions input file. It allows performing estimation with different weights, starting it from different initial parameters, etc. We have found this option very useful for estimation of statistical parameters and for convergence checking and we usually take advantage of it.

Procedure 2 is used when the iterative estimation process is stopped without having exhausted convergence possibilities. This may be the case when MAXITER is too small or GNORM1 too big. In these cases, iterations can be restarted by simply attaching the last computed parameters from the first run, which can be taken from the main output file, to the input data as new initial parameters.

4.9 DIFFICULTIES WITH TRANSIN

TRANSIN is not particularly easy to run. As a matter of fact, input data is very complex. Therefore, input data errors are frequent. Having managed to make the program execute without run-time errors, something that always generates an euphoric relief when one runs any code for the first time, does not mean that one is obtaining sensible results. Actually, it only means that the user is getting ready to face the real problems: first, ensuring that the code is really solving the problem we wanted to solve and, second, ensuring that the parameter estimation convergence problem is reached.

¹(1)
This pessimistic statement is not meant to discourage potential users from proceeding further, but rather to warn them about the need of doing with caution. In our experience, we have always ended up with satisfactory results.

Following is a discussion of frequent errors and difficulties.

### 4.9.1 Input errors

As a consequence of the complexity of input file preparation, input errors are very frequent. When one is not familiar with TRANSIN, it may be worth going over the input file, card by card, while checking input descriptions (Appendix I) before actually writing the file. This may reduce the chance of misunderstandings. The meaning of some variables only becomes clear once later cards have been specified. Going over the input file once it has been prepared helps in discovering accidental errors and enables a better understanding of the code, which will facilitate later identification of errors. Input errors can be of four types: those identified as fatal errors, those that only lead to warning messages, run-time errors and those that do not cause the program to stop. The four types are discussed below.

1. **Error messages.** An extensive, albeit incomplete, check of input errors takes place during the input phase of the code. TRANSIN does not always stop immediately after having found an error. Instead, in some parts of the input, it prints an error message and proceeds with execution until several errors have been identified. Hopefully, error messages are sufficiently clear, so that the user should be able to correct the input file accordingly (the user can also look up at appendix III, where the errors detected by TRANSIN are shown together with an explanation). Sometimes, the error does not take place where TRANSIN identifies it, but earlier, so that the code may be reading an erroneous card. For example, if the number of cards devoted to transmissivity zones in the PAR file is larger than needed, the code will start to read storativity zones where the input file still contains transmissivity data. Hence, if the user is convinced that TRANSIN’s error message does not make sense, he may need to check if the content of earlier cards has been properly printed (we recommend to set variable INPWR to 1 in card A3.2, in order to see how the code has interpreted our input data). If not, error messages may have been caused by unidentified errors (see below).

2. **Warning messages.** TRANSIN is an extremely forgiving code. This gives the user much freedom in combining the possibilities provided by the program so as to solve virtually any flow and transport inverse problem one can think of. Unfortunately, this also lends ample room for allowing perfectly meaningless problems to be executed inadvertently. In short, many errors do not cause the program to stop, but simply to print a warning message. These can be identified by searching the output file for them (text “WARNING” is printed when variable IOWAR is set equal to 1 in Card A3.2). This search should always be performed even if the problem appears to have run smoothly.

3. **Run time errors messages.** Despite of the extensive input error check, many errors may cause FORTRAN run time errors. When this happens, the first thing
to do is to examine the main output file. In this file we can see the input data as they have been interpreted by the program, which not always interprets data as one would like, and search for WARNING and ERROR messages. It is recommended when starting with a new problem (even if the user would know TRANSIN better than its authors), to set variables INPWR=1 (Card A3.2, the program prints all it reads), IOWAR=1 (Card A3.2, the code prints WARNING messages). If the FORTRAN run time error is of I/O type, check the line leading to the error (most compilers print the input file line number causing the input FORTRAN error). If the format of such line is consistent with the description in Appendix I, or if the run time error is not the I/O type, then the user must check the last part of the output file to make sure that printed information is consistent with what he (she) meant. For example, an error of the type “divide by zero” in the subroutine that solves the linear system of the discretized transport equation (2.30) may be a consequence of the system matrix being nearly singular. In several cases, we have found errors in transport boundary conditions. Another rather usual case, consists in forgetting to enter the aquifer thickness, which leads to a steady state equation.

4. **Unidentified input errors.** Even if the code has reached normal termination without errors, the user should assume that there may be a number of unidentified errors. Finding them may not be easy. But chances of their existence are reduced by following a systematic approach. First, the user should search the main output file for “WARNING” messages. Second, main output file should be thoroughly examined to make sure that the code has printed as input data exactly what the user meant. Third, the procedure described in Section 4.4 should be followed to verify discretization. Finally, several simulation runs with abundant output should be performed. By plotting contour lines, computed heads versus time and by printing out mass balances, the user should be able to ascertain whether or not the program is running as expected. Furthermore, these preliminary simulation runs help in finding a correct discretization, in defining adequate convergence criteria for direct problem, and in obtaining an early idea about the model’s behavior.

### 4.9.2 Control of problem size

The maximum allowable size of any problem is controlled by the dimensions of the arrays RV, IV and AS (whose dimensions are IRMAX, IIMAX and ISMAX respectively), defined in the PARAMETER sentence of the main program. The user should make sure that such dimensions are larger than the needed storage space for real, integer and character arrays of the program respectively.

These values are computed by TRANSIN (they are easily obtained as a sum of the dimensions of the different arrays needed for execution). If the problem size is larger than allowed, an error message will be printed, including the minimum dimension that should be given to vectors RV, IV and AS in the main program. The user should then change the dimensions, compile the main program and link the code again.
4.9.3 Run-time errors

Having executed successfully the input phase of the code, run-time FORTRAN errors are unusual. Robustness, rather than efficiency, has been stressed during code writing. Hence, most run time errors are caused by input or by programming errors. Three types of errors can be identified:

1. **Input errors**: As stated earlier, input error may pass unidentified. Hence, the user should first follow the approach described in Section 4.9.1 to make sure that input file is error-free.

2. **Identified errors**: A number of errors may cause the code to stop. Again, we hope that error messages are sufficiently clear for the user to correct them. Errors related to numerically singular matrices require further discussion. Poorly posed flow problems (the ones that lead to matrix $\theta_f A + \frac{1}{\Delta t} D$ in equation 2.15 to be numerically singular) occur when grid conductances vary over several orders of magnitude. This may happen when conductances (proportional to transmissivity over nodal distance squared) vary over orders of magnitude. For example, if one is modeling a confined regional aquifer and is interested in accurately modeling aquifer’s response in a pumping well (so that nodal distances range between cm’s and km’s), the matrix may be numerically singular. This problem may also occur when facing well hydraulics models in which well bore storage is represented by a 1-D element. A third situation frequently leading to ill-posedness of the direct problem is the one associated to fracture flow models in which fracture intersections are represented by highly conductive 1-D or 2-D elements. In order to overcome these difficulties, the user must reduce conductivity of the troublesome elements, provided that it is a fictitious conductivity, increase the size of smallest elements, reduce the size of the largest elements and/or reduce time steps. Another situation, not unfrequent, occurs when one forgets to put boundary conditions in flow or transport matrix for steady state problems. This omission leads to a singular matrix. When treating non-linear problems, it is common that effects of input-data or conceptual errors appear during the convergence. Since convergence problems may appear as consequence of various reasons, it will be discussed extensively in the next Section.

3. **FORTRAN errors**: These mean real trouble. Sometimes, they are caused by input errors. Hence, the first thing to do is to search the output file for “WARNING” messages. If no such messages are found, then examine the input file to make sure that input data has been properly read. “Overflow”, “division by zero” or “logarithm of zero” errors may occur in unstable problems, when one parameter, such as storativity, takes zero or even negative values. In such cases, which can be identified by looking at intermediate parameters, the user must reformulate the problem. In non-linear problems, it is not unfrequent run-time errors whose typical message is “high performance (for some compilers plotting overflow)” or “undefined exponentiation”, in the subroutine funnoli. It can be due to: I.- a wrong definition of a non linear function and II.- a particular evolution of the model parameters which lead to uncomputable numerical terms in the evaluation of particular non-linear functions (it occurs sometimes in the unsaturated non-linear functions). The first case occurs tipically during the first iteration of the inverse problem. The second
may occur at any of them. In any case, the procedure for removing the problem is to check all parameters defining the non-linear functions as well as the current values of model parameters (particularly whether some generic parameter are part of the set of model parameters). The careful analysis proposed shall to point out the cause of the error. Finally, if the error cannot be identified, then the user must resort to desperate means, including examination of the line where the error occur and/or contacting the code authors.

4.9.4 Convergence problems

This final section of the chapter is focused on describing and discussing the most frequent situations which hinder a normal convergence process of either the direct or the inverse problem. As preliminary comment, we would indicate that, although both, the inverse and the direct solution problem are convergence dependent, the nature of these processes is quite different and, therefore, each one of them has to be treated with a particular philosophy.

4.9.5 Direct problem

When convergence problems occur in the direct problem the first step one has to do is to request all information from the program about the convergence process. This is done by setting the variable IOWNR different than zero. Then, a new run is carried out, using the last set of model parameters printed in the main output file of the previous run. If the direct convergence problems occur at the first iteration of the inverse problem, this set of parameters corresponds to their initial values. During the new run, with IOWNR ≠ 0, TRANSIN prints detailed information of the Newton-Raphson process for each time step. Notice that, when IOWNR > 1 the code prints this information every IOWNR time steps. The information consists, basically of:

1.- Last absolute time solved.
2.- Current time step size.
3.- Desirable time step at this observation interval.
4.- Maximum desirable time step at this observation interval.
5.- For each direct problem iteration:
   - Current order iteration number.
   - Maximum residual error.
   - Node with maximum residual error.
   - Maximum change of the state variable.
   - Node with maximum change of the state variable.
When convergence is reached, it is indicated explicitly as well as the total number of direct iterations in the inverse calculation and the sum for of all iterations.

6. When convergence problems occur, a message is printed informing one that the time step length has to be reduced.

7. When the time step is reduced consecutively a certain user-defined number of times (MXNRTF; 8 is a commonly used value). The code stops asking the user to redefine the direct problem.

Based on the information described, we think that direct convergence problems may be solved by the user, without excessive effort. Once the information is obtained, one has to analyse it, having in mind that most direct convergence problems are due to an inadequate combination of: non-linear functions, space discretization, boundary conditions, model parameter values, time discretization, convergence criteria and/or initial conditions. As a general procedure, we suggest the following for identifying the source of convergence problems:

I- Check the maximum consecutive time step reductions (MXNRTF) permitted, prescribed in the DIM input data file. If the value of the parameter is less than 10 try increasing it to around 15. Whether the convergence difficulties dissappear completely, the problem is solved. Otherwise, if they appear again in the same time step or in another posterior one, back up to the original MXNRTF and go to II.

II- Study the causes which forced the program to reduce the time step several consecutive times. There are two possible ones: maximum permitted iteration number (ITRAPMX) exceeded and a positive increase of the change on the maximum state variable or mass balance residual between two successive iterations. If all reductions were caused by maximum iteration number exceeded, one can consider that one of the next situations are probably occurring:

a) The maximum iteration number is very small (a value around 13 is suitable).

b) The time discretization for the observation interval considered is very coarse. Try decreasing the time step one order of magnitude.

c) Excessively severe convergence criteria (DRELMX, DABSMX and RESIDMXF). Check the criterion selection given in section 4.4.3 and modify if necessary.

According to this reasoning, checking these conditions, and changing the appropriate variable, is the most suitable next step. The maximum iteration number ITRAPMX, and time discretization are easily modified in the TRANSIN input files. A new run containing a change (notice that in general two or more simultaneous changes are not desirable during this testing process) may confirm on discard those potential source of problems.

The only check to confirming a, b or c is that, after the change, the complete direct problem solution is reached, using the current set of model parameters. When hypotheses a to c cannot be confirmed or time step reductions were not caused
by exceeding ITRAPMX, another check is possible. It consists of investigating whether the convergence criteria is excessively relaxed. One typical symptom of this error is fast convergence (to a wrong solution) at early steps and, little by little, convergence problems appear until convergence cannot be reached. Again, criteria presented in section 4.4.3 are useful to validate or to eliminate this hypothetical source of problems. If discarded as a source of the problem, return to the original conditions of the problem and go to III.

III- Examine the last time containing solution was obtained. If that time is close (and posterior) to an observation time, it is possible that any time function (if exist) contains a dramatic slope change at this time, leading to convergence problems. If so, a way to confirm this is to do a new run relaxing or neglecting this slope change. If the convergence problems disappear, at least for the examined time, the hypothesis is confirmed and the next step is to study why that function is not numerically acceptable. Sometimes, it corresponds with excessively voluminous sink-source terms in the unsaturated zone. In general, what the modeler has to study is how the time function affects the non-linear parameters. Based on this, a redefinition of the problem is the next step.

IV- Another possible situation is that problems appear at the end of the observation interval. It may be an indicator of incorrectly defined maximum desirable time increment. If the mistake is defining the maximum time step to be too large an unending process of sequential increasing and reduction of the time step is produced. With a little luck (TRANSIN cannot identify this situation) the code will stop because MXNRTF is reached. Nevertheless, most of the time, it doesn’t stop and the CPU time trends to “infinity” before reaching the next observation time. A way to detect this problem is to periodically examine the main output file of TRANSIN. On the other hand, removing the problem is very easy by changing the maximum desirable time increment.

V- Examine whether most of the time step reductions were motivated by verifying the divergence criteria at the same node. This situation, which is quite frequent, is an indicator of problem are concentrated in a particular zone around the corresponding node. In this case, focus your attention on the actual pressure head or concentration at this node and its surrounding non-linear properties. Unsaturated simulations, for instance, present numerical problems when excessively dry states are reached. This is particularly critical in sands. If no conclusions can be reached from this check, or reductions were motivated differently by problems in several nodes it may be necessary to do a new run, ending the simulation just before convergence problems appear. A careful analysis of the output files and the non-linear properties of the problem is then carried out. Analyzing such aspects as stability of the solution in time and space, the presence of oscillations, logic of trends the solution and actual values of the non-linear parameters (conductivities, retardation coefficient, etc) may help at this stage. If one has suspicions about several matters as sources of problems, the first step is to list all of them and try to test each one separately, to single out the cause. For instance if one has doubts about the fitness of a localized area space discretization, because of dramatic contrast of non-linear properties in the neighborhood, first try a run reducing the contrast is the non-linear parameter.
before considering a new discretization.

After analyzing each possible source of error write down your conclusions. Finally, analyze jointly all conclusions to try to identify the error. Unfortunately some sources of convergence problems are detected only at this stage.

4.9.6 Inverse problem

If the inverse problem is well-posed (that is, if data contain sufficient information by unequivocally identifying all parameters), then convergence of the iterative parameter estimation algorithm is not likely to be a problem. In fact, this is usually the case when modeling synthetic problems (see the examples in Chapter 5), or when abundant data are available describing thoroughly stressed aquifers. On the other hand, when no solid conceptual model is known (this is the main difference between synthetic and real problems) or when data are scarce, chances are that instability and convergence difficulties will occur (recall Section 3.7).

Ironically, when facing unstable problems, it may not be easy to know whether convergence has been reached. As discussed in Section 3.3.4, the safest indicator of convergence is the reduction in gradient norm. If this is below 0.1 % of the initial gradient norm, chances are that full convergence has been reached. Convergence and well-posedness can be ascertained by trying a new initial point and by performing an error analysis.

When solving the inverse problem convergence to the same solution starting from two different initial can be considered a safe test of good convergence and uniqueness (this is just an empirical statement). Analogously, if the condition number of the estimation covariance matrix (ratio of largest to smallest eigenvalue) is reasonably small (say smaller than 1000), one may also conclude that convergence has been reached.

If convergence has been reached but the solution does not look good (i.e., differences between computed and measured heads are much larger than expected) then one may have to modify the conceptual model.

Before getting into modifications of the conceptual model, one may have to evaluate the contribution of each observation point to the objective function. This can be done by examining the “statistics on (weighted) residuals” item of the main output file. Severe errors are often made in processing head data. This can be due to measurement errors (actual measurements or topographical errors), typing mistakes, inappropriate treatment of head or concentration data (i.e., using a shallow well to represent average heads in a recharge zone), unmodeled factors (i.e., heads affected by pumping not included in the model, concentrations in a borehole that are not well-mixed, matrix diffusion, etc). Once the point troubles have been identified, one must either correct the cause, eliminate the corresponding data or simply reduce significantly the weight assigned to such point. This should lead to significant reduction of the objective function and, if everything else is correct, to improved convergence.

Lack of convergence can be caused by intrinsic limitations of minimization algorithms or by instability (or both!). As discussed earlier, the condition number of the covariance matrix can help in identifying an instability. If convergence problems are caused by the poor efficiency of the algorithm, the best solution may be to try to help the algorithm,
for example by reducing the number of estimated parameters or increasing the relative weight of prior information as discussed below. If poor convergence is caused by instability (almost certain for regional models when trying to estimate all parameters) then the first thing to do is reread Section 3.7. There, it is argued that instability is caused by insufficiency of data. Actually, large errors also make convergence difficult (notice that a good conceptual model helps convergence!). However, before revising the conceptual model, one should exhaust the possibility of using the current one. That is, one should force convergence. This can be achieved in two ways:

1. **Increase relative weight of prior information**: Since the prior information part of the objective function is the easiest to minimize, (because this part is quadratic) convergence behavior of the solution is significantly improved by increasing the weight given to prior information. This leads to solutions close to prior estimates at the cost to model match but, at least, a solution is obtained. Such solution can then be used as initial point when reducing the weight given to prior information (continuation method). **In summary**, minimize several inverse problems, starting with large weights (group A9) and, by performing several minimization cycles (group A11), reducing them gradually (for example, if weights should be around unity, the user may follow the sequence: 500, 100, 20, 5, 2, 1, or similar).

2. **Reduce the number of parameters**: This can be done by fixing some parameters to their prior estimates. Parameters to be fixed can be selected after a preliminary error analysis (with small weights). Selection can be done either by fixing parameters with very large variances or on the basis of an eigenanalysis. The second option is much better and should be chosen even if the user does not understand very well what an eigenanalysis is (a frequent and more understandable circumstance). The procedure can be summarized as follows:

   2.1 **Select the largest eigenvalues**: If the user has been careful in scaling the problem so that computed parameters are close to unity (recall that nodal and element coefficients should be chosen to ensure that parameters not log-transformed range around unity) and small weight values are used (say, less than 0.01), then 1.0 is an adequate threshold. That is, select all eigenvalues above unity.

   2.2 **Identify the largest components of the corresponding eigenvectors**: Eigenvectors are printed below the eigenvalue and their components follow the standard ordering within TRANSIN (that is, $T$, $S$, $q$, $H$, $Q$, $\alpha$, $\alpha_L$, $\alpha_T$, $D_d$, $\phi$, $\lambda$, $R$, $C_e$ and generic parameters). Hence, as illustrated below, each component of an eigenvector can be associated to a given parameter. The parameters with the largest components in each of the eigenvalues selected in step 2.1 are the ones to be fixed for subsequent estimation runs.

   2.3 **Example**: Assume a problem in which two $T$’s, one $q$, three $Q$’s and one $\alpha$ have to be estimated. Assume that only one eigenvalue has been selected during step 2.1 and that the components of the corresponding eigenvector are: $(0.6, 0.01, 0.04, 0.2, 0.7, 0.3, 0.1)$. Then, the parameter to be fixed is the second uncertain prescribed flow (Notice that 0.6 and 0.01 are associated to transmissivities, 0.04 to recharge, 0.2 to the first uncertain prescribed flow, etc.).
2.4 **Final run.** Once convergence has been achieved, the user may release the parameters that had been fixed. In some cases, specifically if poor convergence was caused by large residuals, this leads to improved results. Otherwise, unstable results may still be obtained, in which case it may be worth while to resort to the first approach (increase relative weights given to prior information), so as to find the smallest weights leading to stable parameters. In any case, a final run must be done, possibly in simulation mode (JOINV<0) to perform an error analysis with reduced weights. The discussion above leads to the issue of comparison between different runs. Let us state that the modeler should not give up his (her) responsibility to inspect the results. That is, when TRANSIN leads to physically meaningless parameters, the user should reject the run.

Modifications of the conceptual model may also help stability, but should only be attempted once a meaningful solution has been obtained from the first model. Deciding how to carry out such modifications falls beyond the scope of this manual because it is a decision based on the hydrologist’s views on the real system. Examining computed parameters, along with maps of head residuals and curves of measured and computed heads versus time can be useful (recall Section 3.5).

4.9.7 **Checklist of stability and convergence issues**

As a summary of the discussion above, the following issues should be examined when solving inverse problems.

- **Thorough check of input data.** Meaningless results and poor convergence are often caused by hard-to-identify input errors. Plot as much input information as possible and devote several preliminary runs to make sure that input data are correct. Subsequent runs will require less checking.

- **Perform error and sensitivity analyses.** These will help in evaluating quality of estimation and in identifying parameters leading to instability and convergence problems. Specifically, some of the parameters associated with largest eigenvalues may have to be fixed during early runs to ensure convergence.

- **Search for the largest contributions to objective function.** Sometimes, a very few observation points contribute to a large portion of head and/or concentration objective function. This may be a result of erroneous measured data (make sure that head measurements are not below the bottom of the well or, for instance, 10 m above its head and check its elevation datum), head variations caused by non-modeled factors or an improper conceptual model. Revision of the conceptual model or lowering the weight assigned to the troublesome point may improve results.

- **Make sure that a proper scaling has been used.** Representation of any aquifer parameter as the product of an element or node dependent coefficient times a zonal unknown model parameter can be made in an infinite number of ways (recall Section 4.3). When dealing with parameters that are not logarithmically transformed (those with IOLGXXX=0, group Cards A5) it is convenient to choose nodal or element coefficients so that zonal parameters fall around unity.
- **Alter weights given to prior information.** Using different weights may help convergence (always start with large weights), allow a rigorous error analysis and obtain a qualitative idea of the reliance of results on prior estimation.

- **Try different initial points.** This will lead to an increased (or decreased!) confidence on the estimation.

- **Qualitative analysis of estimated parameters.** Prior information may not be enough to ensure that estimated parameters are reasonable. Qualitatively examining their values may lead to improved conceptualizations, rejection of results (do not rely only on good-looking matches between measured and computed heads!) or revision of preliminary parameter assessments (do not trust TRANSIN blindly, but be ready to reassess your views on the system on the basis of TRANSIN’s results).

- **Plot maps of residuals (or average residuals).** Plotting maps of residuals (weighted) often suggests ways of modifying the conceptual model (specifically, zonation patterns) so as to improve results. Ideally, residuals should be uncorrelated. Hence, spatially correlated residuals indicate that either model parameters or the conceptual model are incorrect. Judging why the current model has not led to uncorrelated residuals can be interesting and can be taken as the first step to find ways of modifying the model.
Chapter 5

EXAMPLES

5.1 INTRODUCTION

The theory and most common details and mistakes in running TRANSIN have been explained in the previous sections. However, the actual problems are not seen until one tries to run the code. In addition, the code is extremely flexible, so that the full range of cases in which it can be applied may not be apparent at a first sight.

The objective of this chapter is to address these two points, however, the different type of problems that can be solved with TRANSIN-IV is huge, so an exhaustive illustration of them is not part of the scope of this manual. In this chapter, we deal with problems of all dimensionalities. The first example consists of a two dimensional problem, whose only aim is to describe the features of the code in an estimation problem, explaining exhaustively all input files. Example two shows an application to a radial flow case, that cannot be solved directly, but can be simulated with a proper interpretation of the parameters. This example also points out different treatments of the matrix diffusion, comparing a semianalytical solution with TRANSIN’s solution.

5.2 EXAMPLE 1. TWO DIMENSIONAL FLOW AND TRANSPORT

This example is aimed at familiarizing the user with the basic capabilities of the code and with some aspects of statistical parameter estimation and model selection.

5.2.1 Description of the example

The example is outlined in figure 5.1. The flow domain is a rectangular region of 900×2700 m², with impervious boundaries everywhere but on the right side, where head is prescribed to zero.

Transmissivity is assumed to be constant over the zones depicted in figure 5.1. Recharge is taken as constant in time, and increases towards the left (it is constant over every zone, figure 5.1). Therefore, flow takes place under steady-state conditions. The aquifer has a constant thickness of 10 m, an effective porosity of 0.1 and longitudinal and
transversal dispersivities of 100 and 50 m, respectively. The effect of molecular diffusion is neglected. Concentrations are initially equal to zero but at t=0 recharge starts having a 100 ppm concentration of a radioactive decaying solute.

With this example we will test TRANSIN capabilities when estimating model parameters. We arbitrarily chose some parameter values (those shown in table 5.1 under true parameter column) and made a simulation run. The computed heads and concentrations at some points in the mesh (called observation points, Figure 5.1) will be taken as actual heads and concentrations at those points. Measurements were defined as the actual data plus a random noise (to represent the measurement error).

It should be noticed that we have not taken into account numerical accuracy, because it is not important for our problem. If we were simulating a real aquifer, we would have to check both spatial and temporal discretizations.

![Figure 5.1: Up, transmissivity zones and observation points. Down, recharge zones, finite element grid and boundary conditions.](image)

### 5.2.2 Full description of input files

Following is a detailed account of how to prepare the input files for this example. Reading this section can become extremely tedious. It is only recommended for beginners, whom we would advise to try to actually fill in all variables as an exercise and then return to this section. Others may skip it entirely and only return for clarifications. The actual files (root EX1) are shown in Appendix 4.

Table 5.1. True parameters, initial parameters for estimation and computed parameters
<table>
<thead>
<tr>
<th>Parameter</th>
<th>True Parameter</th>
<th>Initial Value</th>
<th>Estimated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transm. 1 (m²/day)</td>
<td>1</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>Transm. 2 (m²/day)</td>
<td>2</td>
<td>1</td>
<td>1.99</td>
</tr>
<tr>
<td>Transm. 3 (m²/day)</td>
<td>3</td>
<td>1</td>
<td>3.01</td>
</tr>
<tr>
<td>Transm. 4 (m²/day)</td>
<td>4</td>
<td>1</td>
<td>4.20</td>
</tr>
<tr>
<td>Transm. 5 (m²/day)</td>
<td>100</td>
<td>1</td>
<td>100.2</td>
</tr>
<tr>
<td>Recharge 1 (mm/day)</td>
<td>0.4</td>
<td>0.4</td>
<td>–</td>
</tr>
<tr>
<td>Recharge 2 (mm/day)</td>
<td>0.3</td>
<td>0.3</td>
<td>–</td>
</tr>
<tr>
<td>Recharge 3 (mm/day)</td>
<td>0.2</td>
<td>0.2</td>
<td>–</td>
</tr>
<tr>
<td>Recharge 4 (mm/day)</td>
<td>0.1</td>
<td>0.1</td>
<td>–</td>
</tr>
<tr>
<td>Pres. head (m)</td>
<td>0</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>Long. dispers. (m)</td>
<td>100</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td>Trans. dispers. (m)</td>
<td>50</td>
<td>50</td>
<td>–</td>
</tr>
<tr>
<td>Porosity (-)</td>
<td>0.1</td>
<td>0.1</td>
<td>–</td>
</tr>
<tr>
<td>Ext. concen (mg/l)</td>
<td>100</td>
<td>100</td>
<td>–</td>
</tr>
</tbody>
</table>

Comment cards are included before every card to show how and when one can use comments. Observe that the comment character ("%") can be at any column in the record. When talking about record numbers, we do not count the records containing the comment character ("%") in next subsections.
First record (card A3.1) contains the title (Two dimensional ...). Record 2 (Card A3.2) contains variables defining some outputs, INPWR=1 means that input data (and interpolated data) will be written in file EX1RES.OUT, IOWAR=1 means that WARNING messages will be also written in the same file. Record 3 (card A3.3) contains the descriptions of problem type. These are: equation to be solved (IOEQT=3, flow and transport); inverse problem to be solved (IOINV=3 means flow and transport inversion, that is, heads and concentrations will be fitted), type of flow regime (IOTRS=0, steady state); transient transport regime with given initial conditions (IORTS=1); lumped numerical scheme for storage term (IOCNST=1); observation points do not coincide with nodal points (IOOBS=0); first order reactions will be included in transport equation to account for radioactive decay (IOLAM=1); age equation will not be solved (IOAGE=0). Problem dimensions are defined in the two next records (Cards A4.1 and A4.2). In card A4.1 we define: the number of elements (NUMEL=540); the number of nodes (NUMNP=308); the maximum number of nodes per element (in this case all elements are triangles, therefore LMXNDL is set equal to 3); transmissivity is isotropic (ISOT=1); matrix bandwidth (NBAND=11, maximum difference between node numbers in any element); the number of observation points (NUOBS=20), the total number of parameters to be estimated (flow + transport parameters, NPAR=5, five transmissivities) the number of observation times (NINT=21); the number of time functions (NFNT=0); no matrix diffusion effects will be taken into account (NTDMT=0); number of transmissivity zones with non diagonal covariance matrix, NPARTRA is set to zero, because only transmissivity standard deviations will be accounted for.

Next record consists of the number of zones for all parameters (card A4.2). Number of transmissivity zones is set to 5 (NZTRA=5); flow regime is steady state, therefore there are no storage zones (NZSTG=0); four recharge zones (NZARR=4); one prescribed head zone corresponding to the right side of the rectangle (fig. 5.1, NZCHP=1); no prescribed flow zones (NZQFP=0); no leakage zones (NZALF=0). One dispersivity zone (NZDSP=1), neither diffusion nor retardation zones (NZDFM=0, NZCRD=0 no retardation zones means retardation equal to 1 in equation 2.16); one porosity zone (NZPOR=1); one external concentration zone (NZCOE=1, this zone represents the recharge concentration); no matrix diffusion zones (NTDMT=0). Next record is composed of logarithmic estimation options (card A5.1). All are set equal to zero except for transmissivity (IOLGTRA=1, this means that transmissivity will be treated logarithmically). Card A5.1 (next record) contains output options. Steady state heads will be written in file EX1RES.OUT because IOWRH=1. Transient concentrations will be written in this file every 10 observation times (IOWRC=10). IOPLH is set equal to zero, because there is no temporal variation of heads. Transient concentrations at every observation time and every observation point are written in file EX1PLT.OUT, because IOPLC=1. Files EX1MSH.OUT and EX1MHH.OUT will be created because IOMHH=1 (file EX1MSH.OUT contains information about the mesh, and EX1MHH.OUT contains computed heads at every mesh node; these output files are prepared for postprocessing with a specific graphic program). Similarly, files EX1MSC.OUT and EX1MCC.OUT will be created because IOMHC=10 (this means that nodal concentrations will be written in file EX1MCC.OUT every 10 observation times).

Next record (Card A7.1) contains the variables that control mass balance output. In
this example IOBALH is set equal to 1 to show the flow mass balance. The transport mass balance will also be computed, therefore IOBALC=3 (this means that every three observation times the balance will be printed in file EX1RES.OUT). Next two cards (A8.1 and A8.2) define optimization parameters. Marquardt’s parameter is initially set equal to zero (XMARQ). This parameter is divided (when needed, section 3.3.3) by 2 (NUMIN), NUMAX is set equal to 5 (section 3.3.3). PHIMIN and PHIMAX (section 3.3.3) are taken as 0.01 and 0.4 respectively. Relative reduction in gradient with respect to the first gradient is $10^{-6}$ (GMNOR1, see section 3.3.4). Lower gradient norm is set equal to $10^{-8}$ (GMNOR, see section 3.3.4). Relative reduction of objective function is $10^{-12}$, DMINF (section 3.3.4). The minimum cosinus between $\Delta p^k$ and $g^k$ is taken as $10^{-5}$ (COSMIN, section 3.3.3).

The rest of optimization parameters are defined in the next record (card A8.2). The number of successive iterations where the cosinus criterion (section 3.3.3) can be violated is taken as 1 (MAXICOS). Log-transformed variables (only transmissivities in the present example) are allowed to have a relative change per iteration of 1.5 (one and a half magnitude order, variable PERMX1). The rest of variables to be estimated are allowed to have a maximum relative change per iteration of 0.8 (80%, variable PERMX2, in this example it does not matter, because all parameters are estimated logarithmically). EPS is set equal to $10^{-5}$ (section 3.3.4). The maximum number of iterations is 33 (MAXITER) and the algorithm is allowed to fail up to 8 iterations (NMTERF1, section 3.3.3) and variable IOWIT is set to 1, so the computed parameters will be written at each iteration (see file EX1RES.OUT in Appendix 5). Next record contains the weighting coefficients for flow parameters. Since we are not including the prior information of parameters, all weighting parameters are set equal to zero. Next record contains the weighting coefficients for transport parameters, which are also set equal to zero because no prior information is included.

As we have set IOMHH equal to 1, we must include in the next record the number of head contour lines that we want in the picture (NCUVH=12, card A10.1). Next two records contain the values of these contour lines (CUVH, eight values per record). Next record contains 10 as the number of concentration contour lines (NCUVC) and the next two lines contain their values (CUVC, eight values per line, you can see on figure 5.4 the values of the contour lines). This file ends with the control card A11.1. There is a -1 in this card, therefore the program execution is stopped at this moment and file EX1DIM.DAT is finished.

5.2.4 File GRI (EX1GRI.DAT)

The next file to be read by the code is the file GRI (EX1GRI.DAT). The first record contains boundary parameter default zones and default aquifer thickness. They are all set equal to zero in this example, and aquifer thickness is set to 10.

Next card, which consists of several records, contains nodal coordinates and aquifer thickness at every node. It should be noticed that not all nodal coordinates are given, because the grid is regular and we can take advantage of the interpolation feature of the program (look at appendixes 4 and 5, where we can compare the input grid in file EX1GRI.DAT with the final grid in file EX1RES.OUT). It should also be noticed that aquifer thickness at all nodes does not need to be given, because it is constant over
the aquifer and it coincides with the default value that has been given in the previous card (DFTACTH=10). Observe that despite TRANSIN interpolation features, the first and last node must always be given. Following the last node, card B1.3 begins. This card contains nodal boundary conditions, zones of boundary parameters (prescribed head, prescribed flow, leakance and external concentration) and matrix diffusion zones. Observe that not all nodes have to be given. In example 1 there are boundary conditions only on the right side. Rigorously speaking, there are boundary conditions on the four sides of the rectangle, but a no flow boundary condition has been adopted on the other three sides. This means \( T \nabla h_n = 0 \) (eq. 2.4 or 2.5) and this is the default taken by TRANSIN (IBCOD=0 in this card). So we have to enter only the values different from zero. It should be remembered that when a node number is missing, TRANSIN assigns to the corresponding variables the same number that was given in the last node number read, except for variables that have a non zero default value (see the explanation of this group card in Appendix 1). With respect to transport boundary conditions, a \( D \nabla c_n = 0 \) condition is taken on impervious boundaries and on outflow boundaries (eq. 2.24). Looking at file EX1GRI.DAT, it can be seen that we start card B1.3 with node 1 and give all values to zero (IBCOD, IBTCO, IXCHP, IXCHPT, IXQQP, IXQQPT, IXCON, IXCONT, IXDMT). Remember that it is not necessary because this is TRANSIN default. Next node is 295 that has a value of IBCOD=1 (prescribed head), IXCHP=1 (the zone of prescribed head which node 295 belongs to) and the rest of values in the record are zero. Notice that nodes between 1 and 294 will take the same values as node 1, i.e., all these nodes will take zero in all these variables. The card ends with last node number, 308, which has the same values as node 295. Nodes between 296 and 307 (both included) take the same values as node 295. It should be noticed that last node number (308 in this example) must always be given. Next record is a blank line, meaning that nodal information is finished.

According input description (Appendix 1), next records would contain matrix diffusion information. In the example they are missing, because we have given a value of zero to variable NTDMT (therefore, matrix diffusion is neglected) in file EX1DIM.DAT.

Element information begins in the next record. First card contains the default zonal information about the elements. This card is useful for parameters that have only one zone, or for parameters having more zones but with one much larger (i.e., more elements belong to this zone) than the others.

In this example, we have given LDDSP=1, LDPOR=1 and LDCOE=1, because these parameters (dispersivity, porosity and external concentration respectively) have only one zone. Next records (card B3.2) contain the elements. All elements are triangles, so LTYPE is set to 3 and LNNDEL is also set equal to 3. After them, element nodes are given in a counterclockwise fashion. Observe also that not all element numbers are given, because we are taking advantage of TRANSIN interpolation facility. You can see the differences between file EX1GRI.DAT in Appendix 4 and file EXIRES.OUT in Appendix 5, where interpolated data is written (because INPWR was set to 1 in card A3.2). Last card of this file (composed of several records), B3.3, contains the parameter zone number which every element belongs to. Missing elements take the same values as the last element introduced. For example, element 1 has LXTRA=1 (element 1 belongs to transmissivity zone number 1), LXARR=1 (element 1 belongs to recharge zone number 1). The rest of values in this record have not been given, because their zone number coincides with their default value.
All records in this card are similar. Last element must always be given. This file must end with a blank line.

5.2.5 PAR File (EX1PAR.DAT)

EX1PAR.DAT is the next file to be read by TRANSIN. Coefficients and zonal values are included in this file. First line is composed of default coefficient values. In our example, the only needed values are those of CFCHP and CFCONT. Variable CFCON is not needed, because the code is not going to compute steady state concentrations. If we were using steady state concentrations as initial conditions, then CFCON would have to be defined. For similar reasons, we do not need to assign a value to CFCHPT, because flow is steady state. The rest of variables are not necessary, because there are neither prescribed flow nor mixed boundaries. Since external concentration and prescribed head do not vary node to node, we need to give only the default values. Next record is blank, to indicate that nodal coefficient values are finished.

Next card defines element coefficient default values for flow parameters. Only transmissivity and steady state recharge have to be given. Since both parameters are constant over every zone and the value of the zonal parameter coincides with that of the physical parameter, we put simply DTRA=1.0 and DARR=1.0 and we do not need to specify the coefficients element wise in the next card (C2.2). Next line is left blank to show the program that element coefficients for flow equation are ended. Next record (card C3.1) contains element coefficient default values for transport equation parameters (dispersivity, porosity, diffusion, retardation and external concentration). Similarly to the previous card, as transport parameters do not vary from element to element, only default values are needed. Dispersivity, porosity and external concentration default coefficients are set to 1 (DDSP, DPOR and DCOE, Card C3.1). Next record is a blank line to indicate that coefficient values have been finished. It should be noticed that coefficient values for the parameters of our problem (in example 1, transmissivity, recharge, prescribed head, dispersivity, porosity and external concentration) must be not all zero, because a standard parameter value is computed as the product of two or three terms (section 4.3, parameter to be estimated, element or nodal coefficient and time function), so we have to be careful giving the three values correctly (this is a usual source of mistakes).

Next cards are devoted to the values to be estimated (often called parameters, although these values are only a part of the actual parameter). We refer to them as zonal parameters. All these cards are quite similar, every line consists of the zone number of a specific parameter, its value, its estimation index (IV, if zero, the parameter is not estimated), its standard deviation and its prior value. If the parameter is time dependent, its time function number is given before parameter prior information. For transmissivity, its isotropy index must be given after the zone number. For example, first line in card C4.1 reads: transmissivity zone number 1 is isotropic (ISOZ=1 in the second column), its initial value is 1.0, but the actual value will be estimated (IV ≠ 0), its standard deviation is 1.0 and its prior value is taken as 1.0 (equal to the initial value, because prior information is not given explicitly). Next cards are read similarly. The file ends, after external concentration zone. If we had included matrix diffusion processes, matrix diffusion zones of porosity, retardation and molecular diffusion should have been given after last external concentration zone.
5.2.6 File TIM (EX1TIM.DAT)

File EX1TIM.DAT contains the temporal information. The first NINT (number of observation time intervals read in card A4.1) records are all similar and then specify each observation time and the number of time increments in which the observation interval will be divided (recall the discussion about observation and computation times in section 4.2.3.1). For instance, record 1 contains the numbers: 1, 0, 10 and record 2: 2, 350, 10. This means that the first observation time is 0, the second observation time is 350 and 10 computation increments will be used between times 0 and 350, i.e., an increment of 35 \((350/10)\) will be taken in equation 2.30. No time functions are used in this example, so card D1.2 is missing. This file ends with the values of THETAF and THETAT, that are 0 and 1 respectively (THETAF has no relevance in our problem as flow is steady state and we are taking an implicit finite difference scheme for transport).

5.2.7 File OBS (EX1OBS.DAT)

The fifth file is EX1OBS.DAT. Its first record (card E1.1) defines the bottom level for head and concentration measurements. This means that if a head measurement has a value less than HBAS (in EX1OBS.DAT, HBAS=–100), then there is no actual measurement, i.e., this datum will not be taken into account, and similarly for concentration measurements and CBAS. Next card defines the default values for head and concentration standard deviation errors. In this example, both are set equal to 0.2. Steady state measurement data are given in next card.

Next card contains observation point coordinates, steady state measurements, standard deviation of error measurements and the name of the observation points (it can be left blank). When flow and transport are steady state, file EX1OBS.DAT is ended here with a blank line. In our case, transport is transient, so we must give the concentration measurements at all observation times and all observation points. For every time, a comment line or blank line (apart from those lines which contain the comment character) has to be given. All measurements at the current observation time are given after this line (one observation point per head and concentration). Once all observation times have been given, a blank line must be left and the observation data file is ended.

5.2.8 File INI (EX1INI.DAT)

The last input file is EX1INI.DAT, which contains the initial concentrations and/or heads. Only initial concentrations are given in this example. The file begins with a comment line (“Initial concentrations”, after 7 lines with the comment character that has been changed to $ in the first line). Card F2.2, with the initial concentrations at every node starts in the next record. It should be noticed that the first and last node must always be present in this card set. The rest of nodes can be missing if initial concentrations are equal at any node, as in our example. The program assigns the last input value entered in this card to missing nodes.

Once all input files are finished, one is ready to run TRANSIN. The first time one runs the code with a new group of input files, one has got a large probability (almost 1) of obtaining wrong results. The large number of parameters that one has to give to
TRANSIN facilitates input errors. But this large number of parameters is what allows TRANSIN to solve a very large number of different problems. When the code does not run adequately, the first thing to do consists in setting variable INPWR equal to 1 (card A3.2). With this variable, we can see on file EX1RES.OUT the input data as it has been interpreted by TRANSIN (who often disagree with our opinion).

5.2.9 Results

As described in the previous section, our problem consists in estimating five transmissivity values, using flow and transport data. The measurements were generated from a simulation with the true parameters (Table 5.1) and adding a noise with a standard deviation of 0.2. After performing an estimation with TRANSIN using these “measurements”, the parameters displayed in Table 5.1 were obtained. In this case, a synthetic example with few parameters and small data errors, the estimation is very good. These estimated parameters are one of the most relevant output result, but TRANSIN generates more information related to them as well as several output files to display different kind of graphics.

5.2.10 Output files

It should be noticed that all results (except the echo of input data and iteration information) are related to estimated parameters.

File EX1PLT.OUT contains heads and/or concentrations at every observation point (20 in our example) and every time. This information is useful to display the time variation of measured and computed heads and/or concentrations at every observation point. A few lines of this file can be see in Appendix 5. The type of curves that can be plotted using the data contained in this file are shown in figure 5.2.

Files EX1MSH.OUT and EX1MHH.OUT contain mesh information and the head values at every node respectively. The first file serves to plot information relative to the mesh, for example, figure 5.1 was generated using this file together with a graphical tool. The contents of these files are explained in detail in Appendix 5. File EX1MSH.OUT together with file EX1MHH.OUT can be used to plot head contour lines, as in figure 5.3. The corresponding files to plot concentration contour lines are EX1MCC.OUT and EX1MCC.OUT. The first one can be used to display the mesh and the zones of transport parameters. File EX1MCC.OUT contains nodal concentration values at some of the observation times, more precisely at observation time 10, 20 and 21 (although 21 is not multiple of 10, the last time is always written when variable IOMHC is different from zero). In Appendix 5 a small part of this file, showing its format, can be found. Typical pictures are shown in figure 5.4.

File EX1PLT.OUT is useful to compare the time evolution of computed and measured values at the observation points. But this file cannot be used for comparing steady state data, as the heads in our example. For such cases, file EX1CVM.OUT can be more adequate. This file is equal to file PLT in format, but the coordinate axis are the computed, and measured heads. If a perfect fit were reached, all dots would lie on a line of slope 1. A typical plot obtained with this file is displayed on figure 5.5.
Figure 5.3: Steady state head contour lines.

All these files are devoted to display several types of figures. File EX1RES.OUT serves a different purpose. It contains all the information about input data, minimization process and some statistical results. This file is shown in Appendix 5.

It starts with the title of the problem and the beginning of run data. After that, all input data are shown, because INPWR was set to 1 in file EX1DIM.DAT. It can be interesting to look at “GRID INFORMATION” and “ELEMENT INFORMATION” comparing the interpolated data with the input data (file EX1GRI.DAT in Appendix 4).

After all the input data, the iterations information is written. As we can see in the appendix, our problem has converged easily. All parameters point to a convergent process, something that is not always easy to ascertain. Following this information is the evolution of estimated parameters through all iterations, because IOWIT is set to one in card A8.2. Following is a short summary of the minimization process and estimated parameters as they have to be given for file EXIPAR.DAT. This can be useful for re-starting the problem beginning with these values.

The next thing one can see on this file are all the mass balance computations that we have selected in Card A7.1 (IOBALH=1, IOBALC=3). Flow mass balance computation is presented first, and following is the solute mass balance every three (IOBALC) observation times (absolute times are 700 days, 1400 days, etc). Look at the division of balance computations by zones. This allows us to see where most of the water enters or leave the aquifer.

Following you may find tables with the residuals of heads and concentrations (residuals= measured−computed). The most interesting tables are those of weighted residuals, because these have a strong relationship with the objective function. The most interesting columns are the last two. The column of “OBJ FUNC” is the contribution of every observation point to the objective function and last column gives the percentage of this contribution with respect to the total objective function. It is interesting to observe, for example, that point 8 contributes with 31% of the objective function. This happens often. In a real case we would have to investigate what is happening with this particular point (an error in the measured data, some important mistake in the conceptual model, etc).

After that there is a short overview of statistical results and model structure
identification. Also the covariance matrix is written. In general, we will look for large variances and large eigenvalues (this example was taken from Medina (1993), where a detailed study of this example can be found, including stability). In our example all of them have reasonable values. The file ends with the nodal heads and the nodal concentrations at the observation times that we have specified in card A6.1 of file EX1DIM.DAT.

5.3 EXAMPLE 2. CONVERGENT FLOW TRANSIENT TEST

This example aims to show TRANSIN features to simulate problems with radial symmetry. The problem consists in a convergent flow tracer test. An instantaneous pulse of tracer is injected at the injection well and water is pumped at the sampling well (Figure 5.6). The aquifer is confined and homogeneous with a constant thickness of 5 m, and it is assumed that a steady state flow regime has been reached. This problem might be treated as a fully three-dimensional problem, but it can be simplified. Taking advantage of symmetries, the problem can be reduced to one dimension. Vertical homogeneity and constant thickness in a confined aquifer allow us to think of the aquifer as a two-dimensional problem in a horizontal plane. Horizontal homogeneity leads to radial symmetry (for flow) that allows us to study the problem using only the radial coordinate. Concentrations do not have radial symmetry, but we will work with averaged (in $\varphi$) concentrations. Therefore, our mesh will be one dimensional, composed of 1-D elements. Yet, one must
be careful about the meaning and value of flow and transport parameters.
5.3.1 Transforming the flow problem to one dimensional

Transmissivity should be isotropic to ensure radial symmetry. Now, we will compare the equations that TRANSIN solves with the same equations written in polar coordinates (observe that head dependence on the angle \( \varphi \) can be neglected due to the radial symmetry). Flow equation for TRANSIN is (2.1, in one dimension):

\[
\frac{\partial}{\partial x} \left( T \frac{\partial h}{\partial x} \right) = S \frac{\partial h}{\partial t} \tag{2.0}
\]

Flow equation (5.1) can be written in polar coordinates (with \( T \) isotropic) as:

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r T \frac{\partial h}{\partial r} \right) + \frac{1}{r} \frac{\partial}{\partial \varphi} \left( \frac{1}{r} T \frac{\partial h}{\partial \varphi} \right) = S \frac{\partial h}{\partial t} \tag{2.1}
\]

As \( \partial h/\partial \varphi = 0 \) because of the symmetry, equation (5.2) may be written as (multiplying both sides by \( r \)):

\[
\frac{\partial}{\partial r} \left( r T \frac{\partial h}{\partial r} \right) = r S \frac{\partial h}{\partial t} \tag{2.2}
\]

If we write \( T' = r T \) and \( S' = r S \), equation (5.3) looks as:

\[
\frac{\partial}{\partial r} \left( T' \frac{\partial h}{\partial r} \right) = S' \frac{\partial h}{\partial t} \tag{2.3}
\]

This equation resembles exactly equation 5.1, but instead of having \( T \) and \( S \), we have new variables \( T' \) and \( S' \) related with those. So, if we wish to solve equation (5.2), we can use TRANSIN code, but giving to it \( r \cdot T \) as transmissivity and \( r \cdot S \) as storage coefficient.

This mathematical reasoning may also be made physically by simple mass balance considerations. From a geometric standpoint, groundwater flow (mass balance) is fully defined by the volumes associated to each node (half the one of adjacent elements) and by inter nodal distances and cross-sectional areas. When we view the aquifer as a line, the farther are the elements, the larger aquifer volume they have associated. For instance, an element joining nodes \( i \) and \( i + 1 \) (nodal coordinates are \( x_i \) and \( x_{i+1} \)) has associated a volume of \( b \cdot \pi (x_{i+1}^2 - x_i^2) \), where \( b \) stands for aquifer thickness. This may be written as:

\[
V = b \cdot \pi (x_{i+1} - x_i)(x_{i+1} + x_i) = 2\pi \cdot b \cdot l \left( \frac{x_{i+1} + x_i}{2} \right) \tag{2.4}
\]

where \( l \) is the element length. Let us denote \( r_i = \frac{x_{i+1} + x_i}{2} \), which is the coordinate of the middle point of the element. Internally, the program already computes \( b \cdot l \) when solving flow equation. In order to include the whole volume of the element, the program need to compute \( b \cdot l \) times \( 2\pi r_i \). It should be noticed that this coincides with what we are doing when solving equation (5.4). The only difference is the factor \( 2\pi \), but it is constant and it can be simplified in equation (5.4). Actually, one may multiply \( T \) and \( S \) by a constant in equation (5.4) and it remains unchanged. However, for reasons that will be clear later, we define \( T' = 2\pi r T \) and \( S' = 2\pi r S \). This example is useful to notice that TRANSIN model parameters are not always a direct translation from the physical ones, but an adequate interpretation of them.
5.3.2 Transforming the transport problem to one dimensional

The transport problem has no radial symmetry. However, sampling at the pumping well is equivalent to radial averaging, thus eliminating angular differences. This allows us to view the problem as if it had radial symmetry. Transport equation in polar coordinates may be written as (for equation 2.16):

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r D_{rr} \frac{\partial c}{\partial r} + D_{r\varphi} \frac{\partial c}{\partial \varphi} \right) + \frac{1}{r} \frac{\partial}{\partial \varphi} \left( D_{r\varphi} \frac{\partial c}{\partial r} + D_{\varphi\varphi} \frac{1}{r} \frac{\partial c}{\partial \varphi} \right) + q_r \frac{\partial c}{\partial r} + \frac{1}{r} q_r \frac{\partial c}{\partial \varphi} = \phi R b \frac{\partial c}{\partial t} \tag{2.5}
\]

where \( D_{rr}, D_{r\varphi}, D_{\varphi\varphi} \) stand for dispersivity components in polar coordinates, \( q_r, q_\varphi \) are Darcy velocity components in polar coordinates, \( \phi \) is porosity, \( R \) is retardation and \( b \) is cross sectional area. It is easy to prove that angular integrals (radial averaging) of \( \partial c/\partial \varphi \) and \( \partial^2 c/\partial \varphi^2 \) are zero. Therefore, equation (5.6) can be rewritten in terms of radially averaged concentrations as (we still write \( c \) for convenience, although it is concentration integrated over \( \varphi \)).

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r D_{rr} \frac{\partial c}{\partial r} \right) + q_r \frac{\partial c}{\partial r} = \phi R b \frac{\partial c}{\partial t} \tag{2.6}
\]

Multiplying both sides of equation (5.7) by \( r \) and substituting \( D_{rr} = \alpha \cdot q_r \) (where \( \alpha \) is dispersivity in radial direction), equation (5.7) may be written as:

\[
\frac{\partial}{\partial r} \left( rq_r \alpha \frac{\partial c}{\partial r} \right) + rq_r \frac{\partial c}{\partial r} = \phi R b r \frac{\partial c}{\partial t} \tag{2.7}
\]

If we define now \( q' = rq_r \) and \( b' = b \cdot r \), it results:

\[
\frac{\partial}{\partial r} \left( \alpha q' \frac{\partial c}{\partial r} \right) + q' \frac{\partial c}{\partial r} = \phi R b' \frac{\partial c}{\partial t} \tag{2.8}
\]

which is identical to the transport equation in one dimension. The only thing to have care of, consists in giving \( b' \) as aquifer cross sectional area and \( q' \) as Darcy’s velocity. To enter \( b' \) as aquifer cross sectional area is very easy, because it is an input data in file GRI. But Darcy’s velocity is computed by TRANSIN using Darcy’s law (equation 2.22) except when we are not solving flow equation. Fortunately, the transmissivity that the program is using to compute Darcy’s velocity is multiplied by \( r \) (recall previous section), so the code is actually computing \( q' \) instead of \( q \).

5.3.3 Boundary conditions and model parameters

Flow parameters are the transmissivity and the pumping flow at node 1 (figure 5.6). Transport parameters are longitudinal dispersion and porosity (diffusion and retardation are neglected). Flow boundary conditions are: prescribed flow equal to pumping at node
1 and prescribed head at the injection well. Because we are working with steady state head and we are only interested in concentrations, the prescribed head value is needless since changes in the value of the prescribed head leads to different head level, but head gradient remains invariable. Head values are unimportant for transport equation, only head gradient influences directly transport equation. For this reason, we might assign to $H$ an arbitrary value. Actually, not only we may assign to $H$ any value we want, but we do not need to solve the flow equation. It would suffice to define boundary flow and Darcy’s velocity at every element. Transport boundary conditions are: $D\nabla c_n = 0$ at the output boundary (pumping well, node 1), and $D\nabla c_n = Qc$ at the input boundary (injection borehole, node 93). Observe that on the input side, external concentration is zero, because we assume that the tracer we are injecting is not present in the aquifer.

Figure 5.6: One dimensional grid. Node 1 represents the pumping well and node 93 is the injection borehole

5.3.4 Description of input files

A complete description of input files was given in the first example of this chapter. For this reason, we will only give a full description of input cards that are new, in the present example. File EX2DIM.DAT looks very similar to EX1DIM.DAT. The only differences (apart from dimensions that are different for every example, number of nodes, number of elements, number of zones, etc) stand on the maximum number of nodes per element (LMXNDL=2, in Card A4.1) and in Card A3.3 where we have assigned a value of -1 for IOINV (any negative value would be ok) because we only want simulation on the current example (in Example 1 we did calibration). Moreover, we assign a value of 2 to IOEQT in the same card because we are not interested in solving the flow equation.

File EX2GRI.DAT displays the following main differences with that of example 1. Only X-coordinates are needed (radius of polar coordinates) in card B1.2. Observe that aquifer thickness is not given in columns 41-50, because aquifer thickness for one-dimensional elements is given in card B3.2 (actually, the cross sectional area of 1-D elements plays the role of aquifer thickness). In the next card, B1.3, boundary conditions are given. All variables are set to zero in node 1. This is used to initialize to zero all variables from node 1 to node 92. This is not particularly important, because all variables are
initialized to zero by default. Nevertheless, it must be kept in mind that when we assign some boundary conditions in one node, the next and following nodes will take the same values except that these nodes were explicitly provided (see section 5.2.2.2, explanation of file EX1GRI.DAT). The second and last record belonging to this card contains IBTCO=2 (mass flow). The zone number for external transient concentration is 1. Next card contains zone number defaults as in example 1. Card B3.2 has some differences with example 1. The type of all elements is 1. All of them have two nodes per element. And in columns 56-65, the cross sectional area of element is given. It should be noticed that these numbers are computed as: 

\[ b \cdot 2\pi \frac{x_{i+1}+x_i}{2}, \]

where \( b \) is the actual aquifer thickness (5 m) and \( x_i, x_{i+1}, \) are the coordinates of element nodes. The last card in this file, B3.3, is similar to that of example 1. In order to model adequately the solute transport equation in the aquifer, it is necessary to refine the grid near the injection well, because there will be the larger concentration gradients. If we wanted to solve accurately the flow equation, we should refine the grid near the pumping well. Even though we will not solve flow equation, we have refined both extremes of the finite element mesh (Figure 5.6).

File EX2PAR.DAT has small differences. These are restricted to flow coefficients which are missing (Card C2.2), because we do not solve flow equation. The rest of this file is similar omitting the part of flow parameters. It should be noticed, however, that if we were solving flow equation, we would have to give the values of transmissivity coefficient element by element, because this value is equal to the transmissivity multiplied by a factor \( 2\pi r \) (recall previous section). Parameter values are shown in Table 5.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>5 m.</td>
</tr>
<tr>
<td>Porosity</td>
<td>0.1</td>
</tr>
<tr>
<td>Molecular diffusion</td>
<td>0.0 m²/day.</td>
</tr>
<tr>
<td>Retardation</td>
<td>1.</td>
</tr>
<tr>
<td>Radial dispersivity</td>
<td>0.5 m.</td>
</tr>
<tr>
<td>Matrix Porosity</td>
<td>0.05 m²/day.</td>
</tr>
</tbody>
</table>
Matrix diffusion
0.001 m$^2$/day.

Well radius
0.2 m.

Injected mass
7.8 g.

Pumping rate
150 m$^3$/day.

Files EX2TIM.DAT and EX2OBS.DAT are very similar to their counterparts of example 1. It should be noticed that there are two ways of simulating the instantaneous tracer pulse. One consists in giving a concentration value as initial concentration in file EX2INI.DAT (the way we have selected). The other way consists in giving a zero initial concentration at all nodes and entering the tracer pulse as if it were entered to the aquifer with the flowing water. In this case, we have to define a time function for external concentration. The shape of such a function is depicted in figure 5.7. It has a non-zero value during a short time interval and then goes to zero. Observe that second and third observation times should be very close to simulate a pulse adequately (the closer these times, the better a pulse is simulated).

![Figure 5.7: Time function to simulate a pulse](image)

Some differences between files EX2INI.DAT and EX2INI.DAT are present. Node
number 93 has an initial concentration of 99.32 mg/l that corresponds to 7.8 g of tracer instantaneously injected. A zero value is assigned to the remaining nodes. Since we are not solving the flow equation, next two card sets containing velocities and boundary flows must be given. Pumping rate is \( Q = 150 \text{ m}^3/\text{day} \). As we will show, flow rate (Darcy’s velocity times cross sectional area) is constant and equal to \(-Q\). From equation (5.4) written for steady state heads \((S' = 0)\), one may obtain:

\[
T' \frac{\partial h}{\partial r} = \text{constant} \tag{2.9}
\]

Flow boundary condition at pumping well is:

\[
T \frac{\partial h}{\partial r} \bigg|_{r=r_w} = \frac{1}{2}Q2\pi r_w \text{flow per unit length } \tag{2.10},
\]

where \( r_w \) stands for well radius. From this equation one obtains

\[
T' \frac{\partial h}{\partial r} \bigg|_{r=r_w} = 2\pi r_w T \frac{\partial h}{\partial r} \bigg|_{r=r_w} = Q \tag{2.11}
\]

looking simultaneously at equations (5.10) and (5.12) we conclude that

\[
T'(r) \frac{\partial h}{\partial r}(r) = Q \quad \forall r \in (0, +\infty) \tag{2.12}
\]

Observe that the right hand side of equation (5.13) is equal to \(-q'(r)\), so \(-q'(r)\) is constant and equal to \(Q\).

Card group F3, which contains Darcy’s velocity, starts with a title card (F3.1, “VELOCITY”) and following it are the velocities at all elements forming card F3.2. As flow rate is constant, it is enough to enter the first and last element, because the rest of elements are interpolated (missing elements have the same velocity value as the last element given). WARNING: This file does not contain Darcy velocities, but rather \(2\pi r T \partial h/\partial r = T' \partial h/\partial r = -Q\). In our case, this is equal to total flow rate, which is constant, unlike Darcy velocity which is inversely proportional to \(r\). Next card group, F4, contains nodal boundary flow rates. It starts with a title card (Card F4.1), and following are the flow values at every node (card F4.2). Only the first and last node have a non zero value. Node 1 is outflow node, so its value is negative and last node is an inflow boundary, so it takes the same value as node 1, but positive (it takes the same value, because flow is steady state). Node number 2 must be given to indicate that from node 2 to node 92, all of them have a zero value. Otherwise, these would be assigned the same value as node 1 (in this case, however, this would not have any special effect, because no boundary condition is given at these nodes, look at Card B1.3).

5.3.5 Results of example 2

Computed concentrations are displayed in figure 5.8. In this figure we compare TRANSIN results with a semi analytical solution obtained by numerical inversion of the Laplace transform solution of the problem (Benet and Carrera, 1992). As we can see, the agreement is very good, which is a consequence of the fine discretization.
Figure 5.8: Comparison between TRANSIN and TRAZADOR results for the case of no matrix diffusion.

5.3.6 Including matrix diffusion processes

TRANSIN has the possibility of including matrix diffusion processes in the transport equation. Even though one can include directly matrix diffusion processes with small changes in the input data, we want to use an alternative approach to show the flexibility of the code. This approach consists in preparing a new model capable to simulate matrix diffusion using only the transport equation without the matrix diffusion term.

Matrix diffusion is simulated simply with a diffusion type equation (like the flow one, eq. 2.25). Therefore, we can mix transport equation (2.16) without matrix diffusion term, together with another transport equation (2.25). Boundary conditions (2.26 to 2.28) are no difficult to implement.

These equations take place on different parts of the aquifer, so we need different grids to discretize their domains. Due to the continuity condition (eq. 2.26), these grids should have common nodes. The whole grid is displayed in figure 5.9. Horizontal mesh is controlled by the same transport equation as in the previous section and it represents the aquifer. Vertical mesh is controlled by diffusion equation and it models the solid matrix.

Condition (2.26) is straightforward, and so is condition (2.27). Boundary condition (2.28) has to be imposed on upper nodes. Some observations should be made about the whole mesh. The first vertical node should be close to the horizontal line, because the
aquifer volume simulated by TRANSIN includes both the one represented by horizontal elements and one half of the first vertical element, i.e., horizontal mesh nodes have more volume associated than in the case of horizontal mesh alone (Figure 5.6). For this reason, the first vertical element should be small. In addition, the larger concentration gradients take place at the interface between horizontal (aquifer) and vertical (matrix) grids.

In addition, the larger concentration gradients are likely to occur at the interface between flowing water and the matrix. This points to a more refined grid in this zone, as we may see on figure 5.9. Although it is not necessary to refine in both extremes of the vertical grid, we have refined both in the same way. In this way there are smaller differences between element lengths than refining only one extreme. In the latter case, last vertical element would be very much larger than the first one. In next section we will discuss some features of this approach, comparing it with the matrix diffusion option includes in TRANSIN. In next section we will compare results obtained using different meshes.

5.3.7 Matrix hypothesis

In this section we will compare three different conceptual models, that differ on the assumptions made about the underlying rock matrix. First hypothesis thinks of the matrix as if it were composed of a large number of layers in which there is matrix diffusion. The second hypothesis treats the matrix as if it were made of a large number of spheres where diffusion takes place. And the third hypothesis consists in thinking of the matrix as formed by a number of cylinders. Depending on these hypothesis, one will find different breakthrough curves.

The conceptual models have in common the distance where diffusion takes place. They differ on the number and shape of particles (which fill the bulk of the aquifer) and on the surface where the diffusion takes place. For the first model particles are small layers, for the second particles are spheres and for the last one are cylinders. The surface where diffusion take place is constant in vertical for the layered model, and it varies with the distance to the flowing water (aquifer) for the other two models.

How can we model the matrix? As diffusion equation takes place in the matrix, we will model the matrix using vertical lines (composed of a number of nodes) from every node of the horizontal grid (aquifer). The shape of this particular grid is depicted in figure 5.9. Water flows along the aquifer (horizontal line in figure 5.9, observe that this part is the same as figure 5.6). In this part of the grid, the tracer is affected by diffusion, dispersion and advection. Along the vertical lines, only diffusion is present. Apart from the diffusion coefficient, another important parameter is the diffusion surface. Observe that the area of such surface increases with the distance, because the farther we are from the pumping well, the larger the associated volume. Also, depending on the conceptual model, the area of the surface is reduced in vertical (for the layered conceptual model is constant, and in the other cases is proportional to \(r\), cylinders, or \(r^2\), spheres). For this reason, all input files will be equal for all these models, except card D2.2, where variable ACTH takes different values depending on the conceptual model. Only the input files for one conceptual model (spheres) are included in Appendix 4.
5.3.8 Input file differences with previous case (no matrix diffusion).

Apart from the obvious ones, number of nodes, elements, coordinates of them, an extra zone of diffusion and porosity (those related with matrix), etc, there are some more differences that may be not so evident. Cross sectional area of vertical elements must be given in file EX22GRID.DAT, as well as a zonal number must be assigned to porosity and diffusion for vertical elements (zone 2 in the example). The values of these zones must be given on file EX22PAR.DAT (see the values on table 5.2). In addition, zero flow rate must be assigned to vertical nodes in file EX22INI.DAT.

5.3.8 Results using the different hypothesis

The results for the three conceptual models are displayed on figures 5.10, 5.11 and 5.12. Because of the fine discretization employed, (both in horizontal and in vertical) the agreement between the finite element simulation and the measurements (actually the semianalytical solution using Laplace transform) is almost perfect.

A comparison between the three conceptual models is depicted on figure 5.13. As one can see in the layered model, the effect of matrix diffusion is observed before than in the other models, because layered model has a smaller diffusion surface, therefore the effect

Figure 5.9: Finite element grid used for the simulation of matrix diffusion processes.
of matrix diffusion is slightly reduced.

All results displayed in this example, were prepared using the *PLT.OUT files. The RES file was used only for checking purposes (check that all input data was ok).

5.4 EXAMPLE 1. CONSTANT ONE-DIMENSIONAL DRAINAGE FROM AN UNSATURATED MEDIUM

5.4.1 General description: non-linear flow inverse problem (IOFLLI=1, IOEQT=1, IOINV=1)

5.4.2 Direct problem

A 0.5 long (1 m section, CFTRA(I)=1.00) unsaturated medium (IOFLSAT=1) is considered. Initial conditions correspond to the saturation (h(I)=0) and boundary conditions consist of a constant flow extraction of 0.046 (QQPC(1); CFPOR(i)=1.0) is 0.35 and its retention curve can be assimilated to that proposed by van Genuchten 1980, adopting next parameters: λ=0.2, residual saturation=0.05, maximum saturation=0.95, capilar head=1.0. On the other hand, specific storage coefficient (STGC(1), CFSTG(1)=1.0) is 0.005 and saturated conductivity (TRAC(1,1)) is 0.1. The problem is modeled considering that flow is mainly 1-D. Component (IODIM=1), and gravity components are neglected (IOPRHED=1).

Non linear terms

Relative permeability, as demanded by TRANSIN, was associated to the saturated one through the variable NFNLTRA(NZ=1)=2 (notice that, because of the one dimensionality of the problem, ISOZ(1)=1)) and NFNLTIP(NFNLTRA(1))=8 being

- \( \lambda = PRGC(NFNLPRG(1,2)) \)
- Residual Saturation= \( PRGC(NFNLPRG(2,2)) \)
- Maximum saturation= \( PRGC(NFNLPRG(3,2)) \)
- Capilar head= \( PRGC(NFNLPRG(4,2)) \)
- PARACD(1,2)=0.0.

Analogously, \( j_n \) the storage term, nonlinearity was defined through the variable NFNLSTG(NZ=1)=1, coupled with NFNLTIP(NFNLSTG(1))=102 and

- \( \lambda = PRGC(NFNLPRG(1,1)) \)
- Residual Saturation= \( PRGC(NFNLPRG(2,1)) \)
- Maximum saturation = PRGC(NFNLPRG(3,1))
- Capillary head = PRGC(NFNLPRG(4,1))

This information is involved in the final part of the PAR file.

## 5.5 Inverse problem

Once the simulation starts, a serie of values of the pressure head evolution at 4 observation points (see OBJ file) is stored (a certain noise was added to those values), took from the results of TRANSIN (PREFIX=CAPINV1) in simulation mode. Then, a change in parameters conductivity, storage coefficient and outflow are introduced, in order to run TRANSIN in inverse problem mode, based on previous measures. The new values of zoned parameters are: \(K^s=10.0\), \(S_s=0.5\) and \(Q=-0.06\).

## 5.6 Discretization and numerical conditioning description

A 50 elements, one-dimensional grid, uniformly long is defined in GRI file. It is considered an admissible refinement degree according to a previous sensitivity analysis of the variable NUMEL. On the other hand a 1.0 units time was simulated, using 0.1 as the prior length of the time step (KINT=1 and DTMXDS=1) for all of the 10 uniform observation interval defined.

Regarding to the direct problem conditioning, values of the convergence criterion used in simulation mode are: \(DRELMX=10^{-8}\) \(DABSMX=10^{-8}\) \(RESIDMXF=10^{-8}\)

Moreover, when running in inverse problem mode, variable convergence criteria option was adopted (IOCRITRAP=1), being the criteria corresponding to relaxed conditions: \(DRELMX=10^{-2}\) \(DABSMX=10^{-2}\) \(RESIDMX=10^{-2}\)

Initial values for the state variable in the direct problem were computed according to the option 2 (IOPINITH=2). It consists of using information of the solutions of all previous simulations for finding the solution at current inverse iteration.

Finally, the inverse problem stopping criteria (valid for the rest of the problems presented) is: \(GMNOR1=0.1\times10^{-8}\) \(GMNOR=0.5\times10^{-9}\) \(DMINF=0.1\times10^{-9}\) \(EPS=0.1\times10^{-7}\) \(MAXITER=10\) \(NMTERF1=11\)

As explained in chapter 4, there exist various types of criteria for selecting the previous variables. Thus, if the most relevant aspect is CPU time, perhaps values of RESIDMXF, DABSMX and DRELMX for simulation mode become excessively rigorous. On the other hand, stopping criteria in the inverse problem are were reference points. The previous comment attends to indicate that, probably, different sets of parameters could lead to similar or better results than those presented in the next section.
Regarding to the conditioning of the inverse problem, it is important to mention that prior information was included (XLAMTRA=XLAMQQP=XLAMHEAD=10). It result (as we will show) in a ratio between the parameter and total objection function close to $10^{-2}$, at the first inverse iteration. As measured, we introduced the real parameter values.

5.7 Results

The evolution of pressure heads along the sample for various solution times is presented in Figure 5.1. A comparison it is also shown also in the figure between measured and computed values. On the other hand, the evolution of the calibration process is presented in the file CAPINV1RES.OUT.

This file contains, I) input data, II) inverse problem evolution indicator, III) direct problem evolution indicators, IV) error analysis and V) pressure head evolution on the nodes of the grid at observation times. It can be seen that the inverse problem is solved in 10 Marquardt iterations but it falls iteration 3, 6 and 9. The objective function (search for FNEW) vary from 15400 to 627. In addition to this, gradient norm (search for GNORM) starts by $1.26 \times 10^6$ and finish by 771. Model parameters, in turn, become close to the real values (0.21 vs 0.1 for $K^s$, -0.01 vs -0.006 for Q and 0.16 vs 0.05 for $S_s$). The reason why the absolute real values are not reached is the noise introduced in the previous information.

Regarding to the inverse problem parameters evolution, we believe that it is a normal behavior when solving problems. Although, we want call for the reader attention in two aspects. First, the gradient norm doesn’t descend from $10^{-7}$ until the 5 iteration, at which objective function is close to the minimum one. Second, the inverse problem stop criteria were no succesfull in the sense that 5 iterations would be enough. A value of DMINF close to $10^{-3}$ would avoid 5 unuseful inverse iterations.

Finally, we will comment the direct problem convergence process. In this sense, the main quality indicator is the Newton-Raphson iteration number at each inverse iteration. It can be seen in the file that direct iteration number start by 33 at first inverse one (search for general accumulative raphson iterations) and afterwards remain almost constant around 11. These facts indicates that the option chosen for initializing the state variable was correct (IOPINITH=2). The reason for concluding this is the difference between the first and the rest of the inverse iterations. That is, so it uses extrapolation. Nevertheless, in the second and rest, it has stopped previous solutions and define, successfully, the initial values based on that solutions. Notice that IOCRITRAP=2 leads to a save of $\approx 210$ direct iterations (the total number in this problem was 137).
5.8  EXAMPLE
TRACER TEST IN A LOW PERMEABILITY, ANISOTROPIC, FRACTURED MEDIA

5.8.1 General description. Flow and transport (IOEQT=3), 3-D (IODIM=3), inverse problem (IOINV=1,2)

This example is part of a serie of tracer and hydraulic test, carried out in the frame of the BERROCAL project, having the hydrological caracterization of the Berrocal area as basic objective. Interpretation of the field test was made with several models and under variated hypotesis. In the most sophisticated conceptual model, the medium is treated as an anisotropic low permeability matrix, crossed by several fracture systems. This matrix is composed basically by granite, gneiss and micas, with some mineral intercalations such as uranium and torio. TRANSIN was sucessfully employed for calibrating the set of field test, when adopting this complex conceptual model.

5.8.2 Description of the physical medium

A convex 3-D polihedra (see figure 5.6) taken from the bulk medium was selected as model domain. All boundaries of the volume are formed by planes, whose dip and direction were defined based on the boundary conditions, resulting from the regional flow conceptualization. The surface boundary, for instance, reproduce the most significant topographic forms. It is assumed in the groundwater conceptual model a constant water recharge in the surface (x-z upper planes). Also, vertical x-y plane (see Fig. 5.6 and 5.7) corresponds to a regional water divide, and the flow direction contains no x components in the vicinity of the z-y boundaries (z-y planes). Then the water recharge travels within the medium, being the natural regional flow -y, z component. Once crossed the volume, the water finds two boundaries. One is the bottom of the aquifer, which coincides with the lower x-z plane. Part of the water reaching this boundary evacuates the medium through itself. Another boundary is the subvertical x-y plane, which coincides with an uranium-quarz dike. Since this is a faster pathway, the water goes down through the dike, arriving, finally, to the intersection with the bottom boundary, getting out in this area. The movement of the water in the bulk occurs through two media; the low permeable rock matrix and fracture systems. Under natural conditions, the processes described govern the groundwater flow, leading to a piezometric system similar to their presented in Figure 5.7, which is assumed stationary.

Hydraulic test

As mentioned before, a serie of boreholes were built in the medium studied. For the proposes of our example, we consider only those identified as S-2, S-13 and S-15 (see fig. 5.6 and 5.8). The test presented here consists of a pumping test in an isolated section of the S-2, collecting piezometric head measures in several isolated sections of S-15, S-13 and S-2 itself.
Tracer Test

Once stabilized the piezometric head field, after pumping, a serie of tracers were introduced in borehole S-15, expecting to recover it in the pumping test. The tracer used are: deuterium, renio, phloxina, uranina and gadolinio. In all cases, certain amount of mass was recovered. Detailed characteristics of both, hydraulic and tracer test, can be found in (referencia).

5.9 Inverse problem

The objective of the calibration exercise was to characterize: conductivity tensor of the rock matrix, conductivity tensor of the fracture systems, storage coefficient of the medium, porosity, if relevant, matrix diffusion parameters. The first step consisted of modelling the flow system. For this propose, various hypothesis relate to anisotropy degree and boundary conditions was made. Then several TRANSIN runs in inverse mode were made, using measures during the transient pumping test (IOTRS=1) as observation data. In addition to this, several TRANSIN runs in inverse mode were also made, using stabilized heads (IOTRS=0) after pumping as observation data. The flow system calibration was done using drowdown because the uncertainty asociated to the absolute piezometric heads.

The combined analysis of results arise from runs described, led modellers to adopt a particular flow system, in order to advance to the next step: tracer test modelling.

Once flow defined, five different inverse transport exercise were developed. They consisted of callibratin each one of the tracer breakthrough. The calibration process included runs where neglected and runs where considered matrix diffusion processes. The case we present in this manual corresponds to tracer deuterio with no matrix diffusion.

Because description of complete relevant details of the tests can be found in (reference) we omit to include it here. Nevertheless, we consider to coment, in the next of this section, an interesting particularity of the tracer test.

In order to introduce the solute mass in the bulk, a free water flushing was made in the borehole zone where the tracer was put. Because the flushing time period were no despicable in order to conceptualize the tracer injection, that period should be modelled. On the other hand, it forces the modeler to consider transient flow. This assumption results in a expensive CPU cost. In order to avoid this cost, the modelling task was divided in two steps. In the first one, we modelled the tracer flushing considering transient transport and transient flow (IOTRS=1, IORTS=1). This run was made in simulation mode (IOINV=-1), that is, no calibration was done. Then we use the solute concentration in the medium at the end of the simulation as initial conditions of the new run (second step). Now the flow was assumed in steady state and transport parameters were identified by doing the run in inverse mode.
5.9.1 Discretization and mathematical conditioning description

A mixed 8157 elements grid was defined for modelling the tracer test. It contains three types of element: right triangle prisms (LTYPE=11), which represents the rock matrix, 2-D triangles in the 3-D space (LTYPE=10) which may represent both the phreatic zone and the fracture systems, and linear elements (LTYPE=1) which represents boreholes. Regarding to the phreatic zone, a plane formed by triangles was coupled to the 3-D grid, in the upper X-Z boundary. The function of this layer is to take in account the surface flow. About the fractures zones, two (triangle elements) planes were included in the volume domain, according to the characteristics of the fracture zones (see fig. 5.7(2)). For the 2-D elements, the equation solved by TRANSIN is

\[ S \frac{\partial h}{\partial t} = \nabla(T\nabla h) + Q \]  

(5.1)

where \( S \) is very small for fracture systems, and it is equal to the porosity in the phreatic zone. On the other hand, \( T \) is a particular property of each medium. This equation is coupled with that solved in the 3-D medium

\[ S_s \frac{\partial h}{\partial t} = \nabla(k\nabla h) \]  

(5.2)

Boreholes were treated as an uniform head medium, assigning a high transmissivity value to the 1-D elements (only those which belong to the particular isolated zone) representing the holes. A general view of the main characteristics of the grid can be found in Figure 5.6, 5.7 and 5.8. It is notable in the grid the high refinement degree in the hole zones compared with that of the boundary. This is due, obviously, to the major gradients occurring in this area, not only in heads but also in concentrations. The boundary conditions in the grid consists of: surface recharge in the upper XZ plane (ARRC(1)=0.22 x 10^{-3}, CFARR(i)=2-D element area); no flow through Y-Z planes (IBCOD(i)=0); no flow through the vertical X-Y plane (water divide); prescribed head in the bottom plane X-Z (CHPC(1)=900, CFHPC(i)=variable) and; prescribed head in the subvertical X-Y plane (uranium-Quarz dike, CHPC(2)=1.04; CFCHP(i)=variable). Moreover, a prescribed flow condition was assigned to the node 1339 (QQPC(1)=-0.1414, CFQQP(1339)=1) which coincides with the pumping isolated zone. The rest of the parameters can be found in the PAR file included in the appendix 4.

Regarding to the transport problem, as discussed in the previous section, the solute injection was included in the model through the initial concentration. On the other hand a zone of porosity and dispersivity was assigned for each one of the phreatic zone, rock matrix and fracture systems.

The time simulation of the model was 49 days, adopting no uniform desirable time lengths (vary between 0.025 and 1 day), according to the evolution of the test.

Input-output files of this example, presented in appendix 4 and 5, corresponds to the final inverse TRANSIN run of the deuterio tracer test. Model parameters are: matrix longitudinal dispersivity, fracture systems longitudinal dispersivity, matrix transversal dispersivity, fracture systems transversal dispersivity, matrix porosity and fracture porosity. The calibration was based on measures obtained in the tracer test.
Including previous model parameter information. Stopping criteria adopted in this case were:

- $GMNOR1 = 10^{-1}$
- $GMNOR = 0.4$
- $DMINF = 0.1 \times 10^{-4}$
- $EPS = 0.1 \times 10^{-3}$
- $PERMX1 = 1.0$
- $PERMX2 = 0.4$

### 5.10 Results

Graphic results of the flow inverse problem result is presented in Figure 5.8. As general comment of this Figure, we can stress two points: the fitness is, for all observation points, very good. Second, this model is the final result of a systematic work where several conceptual models were considered, partially described in the previous section. This work demanded several TRANSIN runs in simulation and calibration mode. As indicated before the flow system corresponding to the Fig. 5.7 used for modelling the transport problem. Figure 5.9 is the breakthrough of deuterium in the pumping test. It is clear in the Figure the high quality of the fitness. Notice also that this final model was used successfully for predicting, additionally, the breakthrough of phloxina. Examining the file CAPINV4RES.DAT, reports that convergence was reached in 4 iterations. As interesting aspect of the inverse problem evolution, we can mention the likeness between all components of the gradient vector at each inverse iteration. Furthermore, the convergence rate is quite similar for all of them. Those facts are clear indicators of the adequate conditioning of the inverse problem. We mean, the set of model parameters was successfully chosen and values assigned to the rest (i.e, velocities) was adequate. In this sense, one can conclude that this is a very robust model taking into account that it reproduces not only the tracer breakthrough but also the pressure head evolution in several points. Furthermore, the structure of the model is quite coherent with all available information about the Berrocal site. Anyway, as inherent to any mathematical model, uncertainty is also present here. It has two main components: related to the conceptual model and related to the parameter values. In this case, that associated to the conceptual model is clearly bigger. Think that hydrogeology of the Berrocal site is very complex and hypothesis done for developing the conceptual model are very simplificative. Regarding to the uncertainty associated to parameters, examining the error analysis at the end of the RES file show that it is low and, thus, acceptable.
Figure 5.10: Comparison between TRANSIN and TRAZADOR results with matrix diffusion on a layered matrix.

Figure 5.11: Comparison between TRANSIN and TRAZADOR results with matrix diffusion on a spherical matrix.
Figure 5.12: Comparison between TRANSIN and TRAZADOR results with matrix diffusion on a cylindrical matrix.

Figure 5.13: Comparison of matrix diffusion effects with the different models.
Figure 5.14: Evolution of the pressure head along the medium for various solution times in the example 1 (left). Comparison between “measured” and computed pressure heads for several observation points on the same problem (right).
Figure 5.15: Main characteristics of the model domain of Problem 4. 1- Basic 2-D grid used for conforming the 3-D Card. 2- Model domain and 3-D finite element grid. 3- Model domain including the solution position of the fractured zones. 4- Model domain including the relative position of the boreholes.
Figure 5.16: Natural flow system illustrations in the Berrocal. It was defined based in several measures, assumed as average heads.
Figure 5.17: Comparison between measured and computed pressure heads in isolated sections of the holes S-13 and S-15 during the pumping test. At the end of this test (once stabilized heads) the tracer test was carried out.
Figure 5.18: Deuterium breakthrough measured (dots) and computed by TRANSIN (continuous line) at the obtained from this exercise were successfuly used for predicting another conservative tracer tests.
Appendix A

INPUT DATA DESCRIPTION

A.1 INTRODUCTION

As mentioned in Section 4.6, input data consists of up to seven files. This appendix contains a description of data to be specified in every line. For organizational purposes, input description is arranged in “groups” and “cards”. In general, each card defines the contents on one line of the input file, although it may have to be repeated to specify sequential sets of data. For example, card B1.2, which contains nodal coordinates, has to be repeated as many times as needed to ensure that coordinates of all nodes are either read or interpolated by the program. The seven files are:

1. FILENAME(1). Main file (Cards “A”).
2. FILENAME(2). Grid definition (Cards “B”).
3. FILENAME(3). Parameters definition (Cards “C”).
4. FILENAME(4). Time and time functions (Cards “D”).
5. FILENAME(5). Measured data (Cards “E”).
6. FILENAME(6). Initial conditions (Cards “F”).
7. FILENAME(7). A priori covariance matrix of parameters (Cards “G”).

The name of the main file is specified interactively in the screen (logical unit number 5). The names of the other input data files are given in the main file or are created from the name of the main input file. Following is a full description of every input file.

Lists with error messages and TRANSIN variables are included in appendixes 2 and 3.

A.2 Comments in input files

At any record of an input file one or more comment lines can be written. To tell the program that a line is a comment line (therefore, this line will be ignored, the percent sign (%) must be placed in that line. It should be noticed that the percent sign can be at the beginning, at the end or in the middle of the line (see input files in appendix IV
for a better understanding). Be careful that if you are using the percent sign as comment character, it can not be used in a title of some input cards of TRANSIN (Cards A3.1, F1.1, F2.1, F3.1, F4.1). By default the percent sign is the comment character. You can change it whenever you want by writing a comment line including in it old-comment character - new-comment character. For example, if you are using % as comment sign and you want to change it to $, put % --->$ (without blank spaces inside this expression). You can see this on appendix 4 in example EX1. It is recommended to use only one character as comment in all files, because if one changes the comment sign in one file, this will be the comment character in the next files.
A.3 FILENAME (1): DIM file

This file contains problem dimensions, output options, convergence criteria, etc.

There are two possibilities to enter the name of this file. If the user enters interactively the full filename, this file must contain the names of the remaining files (Card Groups A1 and A2). On the other hand, if the user enters a name which doesn’t have the character “.”, the program interprets that the user has entered the prefix (root) for all filenames, which are generated by adding a suffix. The suffixes are (Recall sections 4.6 and 4.7): DIM.DAT, PAR.DAT, TIM.DAT, OBS.DAT, INI.DAT, COV.DAT for the input files and RES.OUT, MCC.OUT, PLT.OUT .... for the output files. In this second case you must omit Card group A1 and A2.

This file has been assigned the logical unit number 10 and consists of eleven groups:

Group A1 The names of input data files
Group A2 The names of output data files
Group A3 Options
Group A4 Dimensions
Group A5 Log transform options
Group A6 Output options
Group A7 Mass balance
Group A8 Inverse problem parameters
Group A9 Inverse weighting coefficients and direct problem convergence parameters
Group A10 Contour line data
Group A11 Control data

A.3.1 Group A1. NAMES OF INPUT DATA FILES

Omit if filenames prefix has been given as name for DIM file.

Card A1.1. Grid definition filename (GRI file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(2)</td>
<td>File which contains grid variables: Nodal variables Element variables Unit number of this file is 11 (IUGRID).</td>
</tr>
</tbody>
</table>
Card A1.2. Parameters definition filename (PAR file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(3)</td>
<td>File which contains zonal parameters, coefficients (both of elements and nodes) and non-linear functions parameters. Its unit number is 12 (IUPAR).</td>
</tr>
</tbody>
</table>

Card A1.3. Filename for time information (TIM file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(4)</td>
<td>File with time functions, time intervals and time weighting parameters ($\theta_f$ and $\theta_t$ in equations 2.12 and 2.35). This file will not be open (leave a blank Card) for steady state problems. Its unit number is 13 (IUTIM).</td>
</tr>
</tbody>
</table>

Card A1.4. Filename observation data (OBS file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(5)</td>
<td>File with observation data. This file MUST be created if the inverse problem is to be solved or if the user wants some output values (see card A.6.1). Its unit number is 14 (IUOBS).</td>
</tr>
</tbody>
</table>

Card A1.5. Initial conditions filename (INI file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(6)</td>
<td>File which contains initial conditions and the velocities and flow (these last if no flow equation is solved). In case that several problems are going to be solved, the initial conditions for all problems must be supplied. Its unit number is 12 (IUCAL).</td>
</tr>
</tbody>
</table>

A.3.2 Group A2. OUTPUT FILENAMES

Omit if filenames prefix has been given as name for DIM file.

Card A2.1. Main output filename (RES.OUT file)
<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(9)</td>
<td>Filename for the main output file Its unit number is 25.</td>
</tr>
</tbody>
</table>

Card A2.2. Time evolution file (PLT.OUT file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(10)</td>
<td>Filename for the PLT output file. This file contains the time evolution of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>computed and measured (if available) heads and/or concentrations at every</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>device.</td>
</tr>
</tbody>
</table>

Card A2.3. Meshplot filename for heads (MSH.OUT file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(11)</td>
<td>Filename for the MSH output file. It contains the nodes and elements of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mesh as well as the zonal information of flow parameters.</td>
</tr>
</tbody>
</table>

Card A2.4. Nodal heads at some control times (MHH.OUT file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(12)</td>
<td>Filename for the MHH output file. It is composed of several blocks of data,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>each one corresponding to a control time (these are specified in card A6.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>with variable IOMHH). Each block contains computed heads at every mesh node</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and the values of head contour lines specified in card A10.2. This is also</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>repeated for each flow problem.</td>
</tr>
</tbody>
</table>

Card A2.5. Meshplot filename for concentrations (MSH.OUT file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(13)</td>
<td>Filename for the MSH output file. It contains the nodes and elements of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mesh as well as the zonal information of transport parameters.</td>
</tr>
</tbody>
</table>

Card A2.6. Nodal concentrations at some control times (MCC.OUT file)
Filename for the MCC output file. It is composed of several blocks of data, each one corresponding to an observation time (these are specified in card A6.1 with variable IOMCC). Each block contains computed concentrations at every mesh node and the values of concentration contour lines specified in card A10.4. This is also repeated for each flow problem.

Card A2.7. Sections output filename (SEC.OUT file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(15)</td>
<td>Filename for the SEC output file. This file contains the time evolution of heads and/or concentrations at some linear sections defined in group A11.</td>
</tr>
</tbody>
</table>

Card A2.8. Computed vs measured output filename (CVM.OUT file)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>A20</td>
<td>FILENAME(16)</td>
<td>Filename for CVM output file. File for displaying measured versus computed data (heads and/or concentrations).</td>
</tr>
</tbody>
</table>

A.3.3 Group A3. TITLE AND OPTIONS

Card A3.1. Problem title

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>A80</td>
<td>TITLE</td>
<td>Problem title</td>
</tr>
</tbody>
</table>

Card A3.2. Options controlling printing of input data

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>INPWR</td>
<td>Control of input data printing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INPWR=0 Nothing is printed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INPWR=1 Prints input and interpolated data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INPWR&gt;1 The same plus default assignments</td>
</tr>
</tbody>
</table>
6-10 I5 IOWAR Controls the printing of warning messages. IOWAR=0, the code does not print warnings. IOWAR=1, it prints warning messages. The latter should be preferred, unless the user is interested in obtaining a “beautiful” printout.

11-15 I5 IOPART If different from zero, it prints some information about variables allocation. This variable should be zero, except for code debugging purposes.

Card A3.3. Problem definition options

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOEQT</td>
<td>Type of problem to be solved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOEQT=0 Checking data only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOEQT=1 Only flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOEQT=2 Only transport</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOEQT=3 Flow and transport</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>IOINV</td>
<td>Inverse problem option (section 2.2.3.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOINV&lt;0 Only simulation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOINV=1 Estimation of flow parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOINV=2 Estimation of transport parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOINV=3 Estimation of flow and transport parameters</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IOTRS</td>
<td>Flow regime</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0, Steady state flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1, Transient flow with prescribed initial conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2, Transient flow with steady-state initial conditions.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IOCNSF</td>
<td>Scheme for storage term in flow problem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If 0 consistent scheme, else lumped</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>IORTS</td>
<td>Transport regime</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0, Steady state transport</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1, Transient transport with prescribed initial conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2, Transient with steady-state initial conditions.</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>IOCNST</td>
<td>Scheme for mass storage term in transport problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If 0 consistent scheme, else lumped.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Consistent scheme is not operative in the current version of the code.</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>IOVAR</td>
<td>If equals 1, the code does a statistical analysis of the estimation [suggested 1].</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>IOFOD</td>
<td>If zero, there are no first order reactions (e.g. radioactive decay)</td>
</tr>
</tbody>
</table>
36-40  I5  IODIM  Maximum dimension of any element included in the problem. Notice that one could have 1-D, 2-D and 3-D elements simultaneously in the same problem.

41-45  I5  IOFLLI  Set to 0 when the flow problem is linear, otherwise set to 1. On the other hand, think that a flow problem is considered as non linear when, at least, one flow function from those defined in section A1.9 is included in the cards of group C17.

46-50  I5  IOTRLI  Idem to IOFLLI, in the case of transport. (Not operative in this version)

51-55  I5  IOPRHED  Indicates whether the flow state variable is pressure (set to 1) or piezometric head (set to 0). If IOPRHED=1 and IODIM≤2 the gravity direction has to be defined in PAR (2-D elements) and/or GR1 (1-D elements) files. When IODIM=3 it is assumed -z as the gravity direction.

56-60  I5  IOINV_GS  If different from zero, geostatistical inversion is allowed.

A.3.4 Group A4. DIMENSIONS

Card A4.1. Problem dimensions (elements, nodes, etc)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NUMEL</td>
<td>Number of elements.</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>NUMNP</td>
<td>Number of nodal points.</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>LMXNLDL</td>
<td>Maximum number of nodes per element.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>ISOT</td>
<td>Maximum hydraulic conductivity (transmissivity) tensor anisotropy degree in the problem. It coincides with the maximum value adopted by the variable ISOZ(NZ), defined for each conductivity zone (NZ) in cards of the group C4.</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>NBAND</td>
<td>Bandwidth (maximum difference between the numbers of any two nodes belonging to the same element)</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>NPAR</td>
<td>Total number of parameters to be estimated. It does not include the generic parameters estimated in the Problem.</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>NINT</td>
<td>Number of observation times, including initial time (steady-state) as the first observation time (ignore if IOTRS + IORTS=0 in Card A3.3).</td>
</tr>
</tbody>
</table>
36-40  I5  NFNT  Number of time functions used for describing time
dependence of all transient parameters (ignore if
IOTRS + IORTS=0 in Card A3.3).

41-45  I5  NPARF  Number of flow parameters to be estimated. It
does not include the generic parameters estimated
in the problem.

46-50  I5  NTDMT  If different from zero, matrix diffusion is taken into
account. See Section 2.2.5.1

51-55  I5  NFNL  Total number of non-linear functions required for
modelling the problem. Notice that NFNL is
at less, equal but, in general, greater than the
type functions number employed, because several
functions can belong to the same type. This
variable is relevant only if a non-linear problem
is considered (IOFLLI\neq 0 or IOTRLI\neq 0

56-60  I5  NBLCVP  Number of blocks making the covariance matrix of
parameters. If covariance matrix of parameters is
diagonal, a zero must be set.

61-65  I5  IOFLSAT  Indicates if one consider the possibility that part
of the domain
reaches unsaturated state. If so (IOFLSAT=1),
the problem is non linear (IOFLLI=1) and one
relationship between pressure (IOPRHED=1) or
piezometric (IOPRHED=0) head, and saturation
degree, from those defined in section A1.9, has to
be defined in a non linear function form (group
C17).

66-70  I5  IOFMLF  It controls the way the nonlinear contribution to a
parameter that depends on the head is computed
(Remember that a parameter value is computed
as parZ*CFpar*FTpar*FNLpar, where parZ is the
zonal value, CFpar is the coefficient value, FTpar
is the time function value and FNLpar is the non
linear value). It may be:
1.- FNLpar = FNL(εh_{m+1} + (1-ε)h_m)
2.- FNLpar = εFNL(h_{m+1} + (1-ε)FNL(h_m)
3.- FNLpar = FNL(h_{m+1})
Where v^j_m can be the average of the head h at
time j in one element if the parameter is defined
by elements, or the value of the head h at time j
in one node if the parameter is defined by nodes,
ε is the variable EPSFLU defined in card D3.1
and FNL is the nonlinear function that defines
the dependence of the parameter with the state
variable. Notice that all three cases coincide if
EPSFLU = 1
It controls the way the nonlinear contribution to a parameter that depends on the concentration is computed (Remember that a parameter value is computed as \( \text{parZ} \times \text{CFpar} \times \text{FTpar} \times \text{FNLpar} \), where \( \text{parZ} \) is the zonal value, \( \text{CFpar} \) is the coefficient value, \( \text{FTpar} \) is the time function value and \( \text{FNLpar} \) is the non linear value). It may be:

1. \( \text{FNLpar} = \text{FNL}(\varepsilon c_{m}^{k+1} + (1 - \varepsilon)c_{m}^{k}) \)
2. \( \text{FNLpar} = \varepsilon \text{FNL}(c_{m}^{k+1}) + (1 - \varepsilon)\text{FNL}(c_{m}^{k}) \)
3. \( \text{FNLpar} = \text{FNL}(c_{m}^{k+1}) \)

Where \( c_{m}^{j} \) can be the average of the concentration at time \( j \) in one element if the parameter is defined by elements, or the value of the concentration at time \( j \) in one node if the parameter is defined by nodes, \( \varepsilon \) is the variable \( \text{EPSTRA} \) defined in card D3.1 and \( \text{FNL} \) is the nonlinear function that defines the dependence of the parameter with the state variable. Notice that all three cases coincide if \( \text{EPSTRA} = 1 \)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NDEVS</td>
<td>Number of DEVICES</td>
</tr>
<tr>
<td>5-10</td>
<td>I5</td>
<td>NUMTOBS</td>
<td>Total number of observations (the sum of all observations of each device)</td>
</tr>
<tr>
<td>10-15</td>
<td>I5</td>
<td>NUMTNOD</td>
<td>Total number of grid nodes needed to define the devices</td>
</tr>
<tr>
<td>15-20</td>
<td>I5</td>
<td>NUMTIT</td>
<td>Total number of times needed to define all the data (if measurements are of point in space and time, this number is equal to NUMTOBS)</td>
</tr>
<tr>
<td>20-25</td>
<td>I5</td>
<td>NBANDCOV</td>
<td>Bandwidth of the inverse of the covariance matrix of data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOSMLF</td>
<td>If 1, the flow problems are solved simultaneously, otherwise are solved sequentially</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>IOSMLT</td>
<td>If 1, the transport problems are solved simultaneously, otherwise are solved sequentially</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IORDCH</td>
<td>If 1, the transport problems are a radioactive chain. IOSMLT has to be set to 1.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>NPBFL</td>
<td>Number of flow problems to be solved</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>NPBTP</td>
<td>Number of transport problems to be solved</td>
</tr>
</tbody>
</table>
If 0, constant water content defined by elements is used
If 1, variable water content defined by elements is used
If 2, variable water content defined by nodes is used
(values 1 and 2 have only sense in transient transport with transient flow).

Card A4.2. Number of zones of all parameters

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZTRA</td>
<td>Number of transmissivity zones (minimum=1 if IOEQT ≠ 2)</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>NZSTG</td>
<td>Number of storage coefficient zones (minimum=1 in transient flow)</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>NZARR</td>
<td>Number of areal recharge zones</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>NZCHP</td>
<td>Number of prescribed head zones (used for defining prescribed head or leakage boundaries.)</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>NZQQP</td>
<td>Number of prescribed flow zones</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>NZALF</td>
<td>Number of leakage (α) zones</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>NZDSP</td>
<td>Number of dispersivity zones.</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>NZDFM</td>
<td>Number of molecular diffusion zones</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NZPOR</td>
<td>Number of porosity zones</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NZFOD</td>
<td>Number of first order decay coefficient zones</td>
</tr>
<tr>
<td>51-55</td>
<td>I5</td>
<td>NZCRD</td>
<td>Number of retardation coefficient zones</td>
</tr>
<tr>
<td>56-60</td>
<td>I5</td>
<td>NZCOE</td>
<td>Number of external concentration zones</td>
</tr>
<tr>
<td>61-65</td>
<td>I5</td>
<td>NZDMT</td>
<td>Number of matrix diffusion zones</td>
</tr>
<tr>
<td>66-70</td>
<td>I5</td>
<td>NZPRG</td>
<td>Total number of generic parameter zones. It is only active for non-linear flow or transport problems.</td>
</tr>
<tr>
<td>71-75</td>
<td>I5</td>
<td>NPARFPRG</td>
<td>Number of uncertain generic parameter zones involved in the non linear flow inverse problem.</td>
</tr>
<tr>
<td>76-80</td>
<td>I5</td>
<td>NPARPRG</td>
<td>Total number of uncertain generic parameter zones involved in the inverse problem.</td>
</tr>
</tbody>
</table>

A.3.5 Group A5. LOGARITHMIC TRANSFORMATION OPTIONS

It consists of one card that defines which parameters will be treated logarithmically (see section 4.3).
Card A5.1. Logarithmic estimation options.

It is useful for some parameters like transmissivities and storage coefficient. Skip a line for simulation runs (leave it blank). Also, includes an option for using heads or saturations as observation data in the inverse problem.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOLGTRA</td>
<td>Transmissivity log-scaling index (Suggested, 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- Linear scaling will be used for estimation of transmissivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- Log-scaling will be used</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>IOLGSTG</td>
<td>Same as IOLGTRA for storage (Suggested, 1)</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IOLGARR</td>
<td>Same as IOLGTRA for areal recharge (Suggested, 0)</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IOLGCHP</td>
<td>Same as IOLGTRA for prescribed head (Suggested, 0)</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>IOLGQQP</td>
<td>Same as IOLGTRA for prescribed head (Suggested, 0)</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>IOLGALF</td>
<td>Same as IOLGTRA for leakage (Suggested, 1)</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>IOLGDSP</td>
<td>Same as IOLGTRA for dispersivity (Suggested, 0)</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>IOLGDFM</td>
<td>Same as IOLGTRA for molecular diffusion (Suggested, 1)</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>IOLGPOR</td>
<td>Same as IOLGTRA for porosity (Suggested, 0)</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>IOLGFOD</td>
<td>Same as IOLGTRA for linear decay</td>
</tr>
<tr>
<td>51-55</td>
<td>I5</td>
<td>IOLGCRD</td>
<td>Same as IOLGTRA for retardation</td>
</tr>
<tr>
<td>56-60</td>
<td>I5</td>
<td>IOLGCOE</td>
<td>Same as IOLGTRA for external concentration (Suggested, 0)</td>
</tr>
<tr>
<td>61-65</td>
<td>I5</td>
<td>IOLGPRG</td>
<td>Same as IOLGTRA for generic parameters.</td>
</tr>
<tr>
<td>66-75</td>
<td>I5</td>
<td></td>
<td>Space not used. Set to 0.</td>
</tr>
<tr>
<td>75-80</td>
<td>I5</td>
<td>IOSUCHUM</td>
<td>Indicates if measures are given in terms of pressure or piezometric heads (IOSUCHUM=0), or in terms of saturation degree (IOSULCHUM=1). It set to 1, the flow problem should be unsaturated state (IOFLSAT≠0).</td>
</tr>
</tbody>
</table>

A.3.6 Group A6. OUTPUT OPTIONS

Card A6.1. Output options

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td></td>
<td>Not used</td>
</tr>
</tbody>
</table>
If 1, computed and measured heads in every observation point at all observation times are written in FILENAME (10) (*PLT.OUT). If it is greater than 2, computed values are written at all solution times.

Warning: when solving non-linear problems and or TRANSIN is running in inverse mode, it is suitable to use IOPLH=1. Otherwise, if the solution number becomes great, it is possible to fill the memory space available during the execution.

Similar to IOPLH for concentrations.

If non zero, computed values of heads at all nodes are written in FILENAME (12) every IOMHH observation times. If non zero, Cards A10.1 and A10.2 must be entered.

Similar to IOMHH for concentrations.

If non zero, the time evolution of computed heads through IOSEH linear sections is written in file SEC.OUT as specified in group card A11.

If non zero, the time evolution of computed concentrations through IOSEC linear sections is written in file SEC.OUT as specified in group card A11.

If 1, computed vs measured values of heads are written in Filename (13) (CVM.OUT)

If 1, computed vs measured values of concentrations are written in Filename (13) (CVM.OUT)

If 1, information about the evolution of optimization process is shown in RES file (objective function, gradient, etc.)

If 1, the history of estimated parameters is written in RES file

If 1, darcy’s flow is computed and written in Filename (9) (RES.OUT). Of course darcy’s flow is computed even if this option is not set when transport equation is solved.
## A.3.7 Group A7. MASS BALANCE

Card A7.1. Mass balance options (See section 4.7.1).

This card allow us to have a description of mass balance (flow or concentration) through all observation times (see section 4.7.1).

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOBALH</td>
<td>Flow mass balance computations option. Mass balance is computed by zones (see section 4.7.1). Mass balance is written in file RES.OUT. IOBALH =0 the code computes the flow mass balance every IOBALH observation times. IOBALH &lt;0 the code computes the flow mass balance every IOBALH computation times.</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>IOBALC</td>
<td>Same as IOBALH for solute mass balance.</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IOBALGH</td>
<td>If non zero, the code computes a global flow mass balance (the integral of all flow mass balances at every time step). This option should be used only under transient conditions. The mass balance is written in file RES.OUT.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IOBALGC</td>
<td>Mass balance is written in file RES.OUT. IOBALGC =0 nothing. IOBALGC =1 the same as IOBALGH. IOBALGC =2 global mass balance is not computed, but the temporal mass balance (IOBALC) is computed by integration in the time step. IOBALGC=3 global mass balance is computed and the temporal mass balance (IOBALC) is computed at the last time in the time step (see section 4.7.1).</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>IOBALDH</td>
<td>If not zero, the temporal flow mass balance is computed at every node (quite large files may result with this option). The balance is computed at the times defined by IOBALH.</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>IOBALDC</td>
<td>Same as IOBALDH for concentrations.</td>
</tr>
</tbody>
</table>

Leave blank the next four Cards if no estimation will be made.
A.3.8 Group A8. ITERATION PARAMETERS AND STOPPING CRITERIA

Inverse algorithmic parameters and stopping criteria are entered in this group. Some of them are difficult to assign. Therefore, a suggestion (default) is given for everyone. Details on the meaning of variables can be found in sections 3.3.3 and 3.3.4. The two cards of this group can be left blank if no inverse problem is being solved (i.e., if IONV ≤ 0).

Card A8.1. Optimization parameters

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>XMARQ</td>
<td>Initial value of Marquardt’s parameter (0.0)</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>NUMIN</td>
<td>Marquardt’s parameter is divided by this number on appropriate iterations (see section 3.3.3). Default value is 2.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>NUMAX</td>
<td>Marquardt’s parameter is multiplied by this number on failed iterations (see section 3.3.3). Default value is 5.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>PHIMIN</td>
<td>If the ratio between the actual change in objective function and its quadratic approximation is smaller than PHIMIN, the program considers that the objective function quadratic approximation is poor and Marquardt parameter is multiplied by NUMIN (0.1).</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>PHIMAX</td>
<td>When the above ratio is greater than PHIMAX, the program considers that the objection function quadratic approximation is good and Marquardt parameter is divided by NUMIN (0.8)</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.0</td>
<td>GMNOR1</td>
<td>Convergence criterion. Algorithm stops if $\frac{|g_i|}{|g_1|} &lt; GMNOR1$</td>
</tr>
<tr>
<td>51-60</td>
<td>F10.0</td>
<td>GMNOR</td>
<td>Algorithm stops if gradient norm becomes smaller than GMNOR ($10^{-6}$)</td>
</tr>
<tr>
<td>61-70</td>
<td>F10.0</td>
<td>DMINF</td>
<td>Algorithm stops if $\frac{</td>
</tr>
<tr>
<td>71-80</td>
<td>F10.0</td>
<td>COSMIN</td>
<td>XMARQ is multiplied by NUMIN if the cosinus of the angle between the gradient and the parameters increment is less than COSMIN during MAXICOS iterations (0.01). See section 3.3.3</td>
</tr>
</tbody>
</table>
Card A8.2. Inverse algorithmic parameters (continuation)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>MAXICOS</td>
<td>Number of successive iterations where the previous criterion can be violated (Def=1).</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>PERMX1</td>
<td>Maximum change per iteration in log-transformed variables. Def=1.0</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>PERMX2</td>
<td>Maximum relative change per iteration for the rest of parameters. The use of this variable is tricky. If PERMX2&lt;1, then no parameter will change of sign during the estimation process. If the user desires so, then 0.9 is an appropriate value. However, if one is not sure about the sign of some parameters (which could be the case for some sink/source terms), then 1.5 can be adequate. In this latter case, however, numerical problems may be caused by changing signs in variables which must be positive (e.g., storativity, dispersivity, porosity, etc). This problem can be overcome by specifying such parameters as log-transformed (see card A5.1)</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>EPS</td>
<td>Convergence criterion. Algorithm stops if maximum relative change in one parameter is smaller than EPS (Def=10^{-3})</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>MAXITER</td>
<td>Maximum number of iterations (section 3.3.4)</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NMTERF1</td>
<td>Maximum number of failed iterations</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>IOWIT</td>
<td>If 1, it writes the history of estimated parameters through the iteration process.</td>
</tr>
</tbody>
</table>

A.3.9 Group

A9. WEIGHTING COEFFICIENTS AND DIRECT PROBLEM CONVERGENCE PARAMETERS

Weighting coefficients for the terms of objective function related to parameters prior information ($\lambda_T$, $\lambda_S$, $\lambda_q$, etc).

Card A9.1. Weighting coefficients for flow parameters

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>XLAMTRA</td>
<td>Coefficient ($\lambda_T$) of transmissivity objective function.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>XLAMSTG</td>
<td>Coefficient ($\lambda_S$) of storage coefficient objective function.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>XLAMARR</td>
<td>Coefficient ($\lambda_q$) of areal recharge objective function.</td>
</tr>
</tbody>
</table>
Card A9.2. Weighting coefficients for transport parameters

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>XLAMDSP</td>
<td>Coefficient $\lambda_a$ of dispersivity objective function.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>XLAMDFM</td>
<td>Coefficient $\lambda_D$ of molecular diffusion objective function.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>XLAMPOR</td>
<td>Coefficient $\lambda_\phi$ of porosity objective function.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>XLAMFOD</td>
<td>Coefficient $\lambda_\lambda$ of linear decay objective function.</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.0</td>
<td>XLAMCRD</td>
<td>Coefficient $\lambda_R$ of retardation objective function.</td>
</tr>
<tr>
<td>51-60</td>
<td>F10.0</td>
<td>XLAMCOE</td>
<td>Coefficient $\lambda_{ce}$ of external concentration objective function.</td>
</tr>
<tr>
<td>61-70</td>
<td>F10.0</td>
<td>XLAMPRGT</td>
<td>Coefficient of uncertain generic parameters involved in the transport inverse problem.</td>
</tr>
</tbody>
</table>

Card A9.3. Weighting coefficients for state variables objective function

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>XLAMHED</td>
<td>Coefficient $\lambda_h$ of heads objective function. It has to be nonzero if you are making inversion using head data (i.e., IOINV=1 or 3).</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>XLAMCON</td>
<td>Coefficient $\lambda_c$ of concentration objective function. It has to be nonzero if you are making inversion using concentration data (i.e., IOINV=2 or 3).</td>
</tr>
</tbody>
</table>

Card A9.3. Direct problem convergence parameters

If flow and transport are linear problems, skip these lines and continue to group A10.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>FCTNCV</td>
<td>Time increment increasing factor which the program employs for modifying the time increment whether it is less than that desirable $(\text{TIME}(JH) - \text{TIME}(J)/\text{KINT}(J))$ or than the maximum desirable one $(\text{DTMXDS}(J))$. Recommended 1.5.</td>
</tr>
</tbody>
</table>
11-20 F10.0 FCTDEC Time increment decreasing factor which the program employs for modifying the time increment when convergence problems appear. Recommended 0.5.

21-30 F10.0 FCTINC This factor increases the desirable time increment of a generic observation interval j (only for this inverse iteration), when this is less than the actual time increment. Anyway, the desirable time increment can be as maximum equal to the maximum desirable. (DTHSDS(j)). Recommended 1.5.

31-40 F10.0 DRELMAX Maximum value of \[|(h_{k+1,l+1}^{i+1} - h_{k+1,l}^{i+1})/(h_{k}^{i+1,l+1} - h_{k}^{i})|\] considered admissible in order to assume convergence, being verified simultaneously the criterium given by DABSMAX. I is direct iteration, K is time step level and i indicates node. Recomended value: 10^{-5}.

41-50 F10.0 DABSMAX Maximum value of \[|h^{k+1,l+1} - h^{k+1,l}|\] considered admissible in order to assume convergence, being verified simultaneously the criterium given by DRELMAX. This parameter may vary orders of magnitud depending on the problem (see criteria selection in cap. 4).

51-60 F10.0 RESIDMAXF Maximum value of the mass balance error over the volume represented by a node, \[|f_{i}(h)|_{max}\] (see eq. 2.13), considered admissible in order to assume convergence of the non-linear flow equation (see selection criteria in section 4.4.3).

61-70 F10.0 RESIDMXT Maximum value of the mass balance error over the volume represented by a node, considered admissible in order to assume convergence of the non-linear transport equation.

71-75 I1 IOPINITH Indicates the option selected for initializing pressure or piezometric heads \(h^{k+1,0}\) at the beginning of the direct problem iterative process, for each time step;

IOPINIT=0: extrapolation criterium is used.
IOPINIT=3: the program defines \(h^{k,0,m}\) (m indicates inverse problem iteration order) based on the solution of the direct problem obtained in previou inverse iterations.
IOPINIT=1 or 2 the program sets the best of the previous alternatives described, applied in the time \(k-1\).

Warning: If TRANSIN is running in simulation mode or MAXITER=1 use IOPINITH=0
76-80   I1   IOCRITRAP   Option of treatment on the direct problem convergence criteria. If set to 0 they are considered constants; when set to 1 they may vary at each iteration of the inverse problem. If only simulation is made, or MAXITER=1 it has to be equal to 0. Warning: if set to 1 card A9.5 has to be entered.

Card A9.4. Direct problem convergence parameters (continuation)

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1-10   | F10.0  | ZEROF    | Minimum value of $|h^{k+1,l+1} - h^k|$ considered valid in order to apply the convergence criterium represented by the variable DRELMAX, read in card A9.3. Suggested value $10^{-10}$.
| 11-20  | F10.0  | ZEROT    | Idem to ZEROF but in the transport equation.
| 21-25  | I5     | ITRAPMX  | Maximum number of direct problem iterations permitted in order to look for convergence, before reducing the time step. Suggested value: 15.
| 26-30  | I5     | MXNRTF   | Maximum consecutive time step reductions permitted in order to achieve convergence, before stopping. Suggested value: 6.
| 31-35  | I5     | MXNRTT   | Idem to MXNRTF, for the transport equation.
| 36-45  | F10.0  | FCTDVNR  | Testing factor. Set to 1.0 in all cases.
| 46-50  | I5     | MINCAT   | Minimum number of consecutive convergences, achieved after a time step reduction, required before starting to increase the time increment, when it is less than that desirable. Suggested value: 5.
| 51-55  | I5     | IOWNR    | Option of printing information about the direct problem iterative process evolution. If set to N, the code will print detailed information of the process every N solution times. Warning: Furthermore, this variable is used as criteria for writing the saturation degree state in the file FOR088.DAT when the flow is unsaturated. Again, if set to N, the code will print saturation degree at geometrical center of each element, every N solution times.
| 56-65  | I5     | DHITMX   | Testing Factor. Set to 1.0 in all cases.
| 66-75  | I5     | DCITMX   | Testing Factor. Set to 1.0 in all cases.

Card A9.5. Direct problem convergence parameters (continuation)

Complete only when IOCRITRAP $\neq 0$, otherwise omit. This card is used for defining the variation of the direct problem convergence criteria, according to the status of the
inverse problem. For a detailed explanation, please refer to section 4.4.4 and Figure 4.5.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>OBJHED1(J_{min})</td>
<td>Value of the objective function considered close to that corresponding to the minimum of the inverse problem.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>RESIDMX1F(c^{min}_1)</td>
<td>Mass balance error per unit time permitted over the volume corresponding to a node, having guarantee of suitable accuracy.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>DABSMX1(c^{min}_2)</td>
<td>Maximum value of the absolute change of the state variable between two direct iterations considered admissible in order to assume convergence, having guarantee of an accurate solution.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>DRELMX1(c^{min}_3)</td>
<td>Idem to DABSMX1 but considering relative changes of the state variable.</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.0</td>
<td>OBJHED2(J_{max})</td>
<td>Value of the objective function considered notably bigger than that expected to find in the minimum of the inverse problem.</td>
</tr>
<tr>
<td>51-60</td>
<td>F10.0</td>
<td>RESIDMX2F(c^{max}_1)</td>
<td>Idem to RESIDMX1F but admitting certain degree of precision error in the solution.</td>
</tr>
<tr>
<td>61-70</td>
<td>F10.0</td>
<td>DABSMX2(c^{max}_2)</td>
<td>Idem to DABSMX1 but admitting certain degree of precision error in the solution.</td>
</tr>
<tr>
<td>71-80</td>
<td>F10.0</td>
<td>DRELMX2(c^{max}_3)</td>
<td>Idem to DRELMX1 but admitting certain degree of precision error in the solution.</td>
</tr>
</tbody>
</table>

A.3.10 Group A10. HEAD CONTOUR LINES

This group of cards is devoted to define the head contour lines output. This group should be entered only when IOMHH + IOMHC \neq 0. In this group the number of desired contour lines and their values is defined. If several problems are solved (flow or transport), these cards have to be repeated for each one.

Card A10.1. Number of head contour lines

Omit if IOMHH=0 in card A6.1.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NCUVH</td>
<td>Number of head contour lines</td>
</tr>
</tbody>
</table>

Card A10.2. Head contour lines values

Omit this Card if NCUVH\leq0 or if IOMHH=0.
For \( I = 1, \ldots, \text{NCUVH} \), head contour values are entered in this card. Write eight values per line and use as many lines as needed to write all NCUVH values.

**Card A10.3. Concentration contour lines**

Omit if IOMCC=0, in card A6.1.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NCUVC</td>
<td>Number of concentration contour lines values</td>
</tr>
</tbody>
</table>

**Card A10.4. Concentration contour lines values**

Omit this Card if NCUVC \( \leq 0 \) or if IOMCC=0.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>8F10.0</td>
<td>CUV(I)</td>
<td>Same as CUVH(I), for concentrations</td>
</tr>
</tbody>
</table>

**A.3.11 Group A12. END CARDS**

This group defines the end of the program. These cards are useful to start an inverse problem from different initial points, to make several simulations runs, etc (section 4.8).

**Card A12.1. Stopping card**

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IPROCESS</td>
<td>If less than zero the program will stop. Otherwise it will continue.</td>
</tr>
</tbody>
</table>

**Card A12.2. Continuation option**

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOINIPAR</td>
<td>If 1 the program reads new initial parameter (Cards C4 through C16). If 2, it reads the variables from Cards A5 through A10 again. If 3, it reads both card groups, beginning with model parameters. If 0, program stops.</td>
</tr>
</tbody>
</table>
A.4 FILENAME(2): GRI File

This file contains the grid definition. The grid is defined by its elements, nodes, boundary conditions and parameter zones. This file consists of three groups:

(Cards B1) Node definitions: Coordinates, boundary conditions and zones related to nodes.

(Cards B2) Matrix diffusion thickness and truncation error in its approximation.

(Cards B3) Element definitions: Element nodes and zones associated with every element.

The first card of every group contains default values. If a default value is non-zero, it will be assigned to the corresponding variable unless it is specified otherwise in subsequent cards.

A.4.1 Group B1. NODAL DATA

Card B1.1. Default nodal data

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IDCHP</td>
<td>Default value for steady-state prescribed head zone</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>IDCHPT</td>
<td>Same for transient</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IDQQP</td>
<td>Default value for steady state prescribed flow zone</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IDQQPT</td>
<td>Same for transient</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>IDALF</td>
<td>Default value for leakage zone</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>IDALFT</td>
<td>Same for transient</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>IDCON</td>
<td>Default value for steady state external concentration zone</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>IDCONT</td>
<td>Same for transient</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>IDDMT</td>
<td>Default value for matrix diffusion zone</td>
</tr>
<tr>
<td>46-55</td>
<td>F10.0</td>
<td>DFTACTH</td>
<td>Aquifer thickness default value</td>
</tr>
<tr>
<td>56-65</td>
<td>F10.0</td>
<td>DEFHBAS</td>
<td>Default value for the bottom of the aquifer</td>
</tr>
<tr>
<td>66-75</td>
<td>F10.0</td>
<td>DEFCBAS</td>
<td>Default value for the bottom level of concentration</td>
</tr>
</tbody>
</table>

Card B1.2. Nodal coordinates and thickness

Nodal coordinates and aquifer thickness are specified in this Card. In general, provide one Card per node in sequential order. If some nodes are missing the program will generate their coordinates by linear interpolation. In addition, the code assigns as thickness value the last value given for missing nodes if thickness default value was set equal to zero, otherwise default value is assigned.

For example, assume that DFTACTH=0 in previous card and the file contains:
Node .... 23  Coordinates....  100.  250.  1000.  Thickness  3.2
Node .... 26  Coordinates....  400.  850.  1900.  Thickness  4.6

The program will operate with:

Node .... 23  Coordinates....  100.  250.  1000.  Thickness  3.2
Node .... 24  Coordinates....  200.  450.  1300.  Thickness  3.2
Node .... 25  Coordinates....  300.  650.  1600.  Thickness  3.2
Node .... 26  Coordinates....  400.  850.  1900.  Thickness  4.6

But if DFTACTH=7.0 in previous card, then the program will operate with:

Node .... 23  Coordinates....  100.  250.  1000.  Thickness  3.2
Node .... 24  Coordinates....  200.  450.  1300.  Thickness  7.0
Node .... 25  Coordinates....  300.  650.  1600.  Thickness  7.0
Node .... 26  Coordinates....  400.  850.  1900.  Thickness  4.6

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>N</td>
<td>Node number</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>X(N)</td>
<td>X-coordinate of node N</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>Y(N)</td>
<td>Y-coordinate of node N</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>Z(N)</td>
<td>Z-coordinate of node N</td>
</tr>
<tr>
<td>36-45</td>
<td>F10.0</td>
<td>BTRA(N)</td>
<td>Thickness of node N for 2-D elements. Ignore for 1-D elements without matrix diffusion. Matrix diffusion area per unit length associated to node N for 1-D elements with matrix diffusion. See section 2.2.5.1. Warning: thickness defined here is valid only for transport equation (eq. 2.17)</td>
</tr>
<tr>
<td>46-55</td>
<td>F10.0</td>
<td>HBASE(N)</td>
<td>Bottom level of the aquifer in node N($h_b$). It should be defined only in problems under the following conditions: 1. flow equation 2. In node N, Dupuit approx. is adopted (that is, free aquifer treated as 1-D or 2-D medium). 3. The problem is considered non-linear using non-linear functions type 1,2 or 21 (see appendix 1, sec. A1.9) for computing transmissivity in, at least, one element associated to node N.</td>
</tr>
<tr>
<td>56-65</td>
<td>F10.0</td>
<td>CBASE(N)</td>
<td>Bottom level of concentration in node N($c_b$). Used only in some transport nonlinear functions.</td>
</tr>
</tbody>
</table>

To end this group, the last node must be given.
Card B1.3. Boundary conditions and parameter zones

In general provide one card per node in sequential order. Notice that the term boundary condition is used here in a generic manner to identify nodes where actions (prescribed head, pumping, etc) are taken, regardless of whether such nodes belong to the topological boundary of the grid. This card set consists of as many lines as needed to specify boundary conditions at all nodes. **Repeat this group as many times as problems (one problem may include flow and transport)**

For missing nodes, boundary condition parameters are kept equal to those specified for the first node of the sequence when no default value (zero) has been provided in Card B1.1, otherwise default value is assigned to missing nodes. In addition, boundary conditions (defined with IBCOD and IBTCO) have no user defined default values. Both, IBCOD and IBTCO are initialized to zero, i.e., no boundary condition. For instance, assume that IDCHP=1, IDCHPT=2, IDQQP=0 in Card B1.1. If the file contains:

```
Node IBCOD IBTCO IXCHP IXCHPT IXQQP etc...
1 1 0 0 0 1 ....
4 0 0 2 0 0 ....
```

The program will operate with:

```
Node IBCOD IBTCO IXCHP IXCHPT IXQQP etc...
1 1 0 0 0 1 ....
2 1 0 1 2 1 ....
3 1 0 1 2 1 ....
4 0 0 2 0 0 ....
```

Nodes 2 and 3 are missing in the input file. IBCOD takes the same value as the last node given, i.e., the same value as node 1. IXCHP takes the default values in nodes 2 and 3, because default value is not zero. But IXQQP takes the same value as node 1 which is the last given value, because its default is zero (IDQQP).

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>N</td>
<td>Node number</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>IBCOD(N)</td>
<td>Flow boundary condition code (Eq. 2.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- No flow ($\alpha = 0, Q = 0$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- Prescribed head ($\alpha = \infty$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2- Prescribed flow ($\alpha = 0, Q \neq 0$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3- Mixed boundary condition ($\alpha = 0, Q \neq 0$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4- Mixed boundary with flow term ($\alpha \neq 0, Q \neq 0$)</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IBTCO(N)</td>
<td>Transport boundary condition code (Eq. 2.24)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- No flow ($\beta = 0, M = 0$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- Prescribed concentration ($\beta = \infty$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2- Mass flow ($\beta \neq 0, M = 0$)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4- Input mass ($\beta = 0, M \neq 0$)</td>
</tr>
</tbody>
</table>
16-20  I5  IXCHP(N)  Leave blank if IBCOD(N)=0, 2 or if IOTRS=1. Prescribed head zone to which node N belongs in steady state.

21-25  I5  IXCHPT(N)  Leave blank if IBCOD(N)=0 or if IOTRS=0. Prescribed head zone to which node N belongs in transient.

26-30  I5  IXQQP(N)  Leave blank if IBCOD(N)=0, 1 or 3 or if IOTRS=1. Prescribed flow zone to which node N belongs in steady state.

31-35  I5  IXQQPT(N)  Leave blank if IBCOD(N)=0, 1 or 3 or if IOTRS=0 Prescribed flow zone to which node N belongs in transient state.

36-40  I5  IXALF(N)  Leave blank if IBCOD(N)=0, 1 or 2. Mixed B.C. zone (parameter $\alpha$ in equation 2.4).

41-45  I5  IXALFT(N)  Idem to IXALF for transient problems.

46-50  I5  IXCON(N)  External concentration zone to which node N belongs in steady-state. Leave blank if IBCO(N)=0.

51-55  I5  IXCONT(N)  External concentration zone to which node N belongs in transient. Leave blank if IBCO(N)=0.

56-60  I5  IXDMT(N)  Diffusion matrix zone number. Leave blank if NTDMT=0 in card A4.1.

To end this group, the last node must be given.

### A.4.2 Group B3. ELEMENT DATA

Default parameters, connections between nodes, number of nodes per element, element cross sectional area (only for 1-D elements) and parameter zones are defined in this group.

This group must be ended with the last element.

#### Card B3.1. Default value of parameter zones.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>LDTRA</td>
<td>Default transmissivity zone number.</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>LDSTG</td>
<td>Default storage coefficient zone number.</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>LDARR</td>
<td>Default steady-state areal recharge zone number.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>LDARRT</td>
<td>Default transient areal recharge zone number.</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>LDDSP</td>
<td>Default dispersivity zone number.</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>LDDFM</td>
<td>Default molecular diffusion zone number.</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>LPOR</td>
<td>Default porosity zone number.</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>LDCRD</td>
<td>Default retardation zone number.</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>LDCOE</td>
<td>Default external concentration zone number.</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>LDFOD</td>
<td>Default first order decay zone number.</td>
</tr>
</tbody>
</table>
Card B3.2. Element type, nodes and thickness

This card is repeated as many times as needed for specifying all the elements.

Element type, number of element nodes, node numbers and cross sectional area (one dimensional elements) are read in this card. In general, one card is provided for each element in sequential order. If elements are missing, they are generated automatically by linear interpolation of node numbers, so care must be taken because wrong node numbers may be generated. Element type, number of element nodes and cross sectional area of the element are assigned to the previous given value. WARNING: Element numbers must be provided increasingly and nodes must be provided counterclockwise.

For example, if one Card looks like: Element number 1 Nodes 1 10 100 Cross-sect. area 1.0 and the next. Element number 5 Nodes 9 14 120 Cross-sect. area 5.0

It is equivalent to: Element number 1 Nodes 1 10 100 Cross-sect. area 1.0 Element number 2 Nodes 3 11 105 Cross-sect. area 1.0 Element number 3 Nodes 5 12 110 Cross-sect. area 1.0 Element number 4 Nodes 7 13 115 Cross-sect. area 1.0 Element number 5 Nodes 9 14 120 Cross-sect. area 5.0

Notice that node numbers are not interpolated, but increased in one unit sequentially.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NE</td>
<td>Element number</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>LTYPE(NE)</td>
<td>Type of element (section 2.2.3.2):</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.- Linear onedimensional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.- Quadratic one dimensional (only if IOINV=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.- Linear triangle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.- Six nodes triangle (only if IOINV=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.- Quadrilateral of 4 nodes and 4 Gauss integration points (only if IOINV=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.- Quadrilateral of 4 nodes and 1 Gauss integration point (only if IOINV=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.- Quadrilateral of 8 nodes and 9 Gauss integration point (only if IOINV=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8.- Quadrilateral of 8 nodes and 4 Gauss integration point (only if IOINV=0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.- 3-D Thetrahedrom of 4 nodes with linear interpolation functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10.- Plane triangle contented within a 3-D medium, with linear interpolation functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11.- Right triangle prism (3-D) with 8 Gauss integration points (Further details in Figure 2.4 and Sec. 2.2.3.2)</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>LNNDEL(NE)</td>
<td>Number of nodes in element NE</td>
</tr>
</tbody>
</table>
Node numbers of element NE corners. Enter LNNDEL(NE) numbers in counterclockwise order.

Cross sectional area of element NE (Only for one-dimensional elements)

Card B3.3. Definition of parameter zones

In this card, the zones for parameters defined by elements (see table 4.1) are provided. Supply as many lines as necessary to specify all parameter zones. **Repeat this group as many times as problems (one problem may include flow and transport)**

For missing elements, zone numbers are kept equal to those specified for the first element of the sequence when no default value (zero) has been provided in Card B3.1, otherwise default value is assigned to missing elements. For instance, assume that LDTRA=2, LDSTG=1, LDARR=0 in Card B3.1. If the file contains:

<table>
<thead>
<tr>
<th>Element</th>
<th>LXTRA</th>
<th>LXSTG</th>
<th>LXARR</th>
<th>etc...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>.....</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>.....</td>
</tr>
</tbody>
</table>

the code interpret this as:

<table>
<thead>
<tr>
<th>Element</th>
<th>LXTRA</th>
<th>LXSTG</th>
<th>LXARR</th>
<th>etc...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>.....</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>.....</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>.....</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>.....</td>
</tr>
</tbody>
</table>

Elements 2 and 3 are missing in the input file. LXTRA takes the default value (2), because it is non zero. For the same reason, LXSTG takes the default value (1 in this case). On the other hand, LXARR takes the same value as the previous element given, i.e., the same value as element 1, because default recharge is set equal to zero.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NE</td>
<td>Element number.</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>LXTRA(NE)</td>
<td>Transmissivity zone number for element NE. Required only if EOEQJ≠2.</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>LXSTG(NE)</td>
<td>Storage coefficient zone number. Required only if IOTRS≠0.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>LXARR(NE)</td>
<td>Steady state areal recharge zone. Required only if IOTRSLNZARR≠0.</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>LXARRT(NE)</td>
<td>Transient areal recharge zone number. Required only if IOTRSLNZARR≠0.</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>LXDSP(NE)</td>
<td>Dispersivity zone number. Required only if NZDSP≠0.</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>LXDFM(NE)</td>
<td>Molecular diffusion zone number for element NE. Required only if NZDFM ≠ 0.</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>LXPOR(NE)</td>
<td>Porosity zone number. Required only if IOEQT ≠ 1 or IOFLSAT ≠ 0.</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>LXCRD(NE)</td>
<td>Retardation zone number. Required only if NZCRD ≠ 0.</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>LXCOE(NE)</td>
<td>External concentration zone number. Required only if NZCOE ≠ 0.</td>
</tr>
<tr>
<td>51-55</td>
<td>I5</td>
<td>LXFOD(NE)</td>
<td>First order decay zone number. Required only if NZFOD ≠ 0.</td>
</tr>
</tbody>
</table>

To end this group, the last element must be given.
A.5 FILENAME(3): PAR File

This file contains element coefficients, nodal coefficients, zonal parameters, matrix diffusion parameters, non-linear functions and gravity direction. The first three card groups are the nodal coefficients for both equations, element coefficients for flow equation and element coefficients for transport equation, respectively. The first Card of each group contains default values. If a default value is different from zero, this will be the default value of this coefficient for all elements or nodes unless specified otherwise in the subsequent cards.

The card groups in this file are:

- Group C1. Nodal coefficients.
- Group C2. Element coefficients for flow equation.
- Group C3. Element coefficients for transport equation.
- Group C4. Transmissivity zonal parameters.
- Group C5. Storage coefficient zonal parameters.
- Group C6. Areal recharge zonal parameters.
- Group C7. Prescribed head zonal parameters.
- Group C8. Prescribed flow zonal parameters.
- Group C9. Leakance zonal parameters.
- Group C10. Dispersivity zonal parameters.
- Group C13. Linear decay parameter.
- Group C15. External concentration zonal parameters.
- Group C15bis. Matrix diffusion parameters.
- Group C17. Non-linear functions definition.
- Group C18. Gravity direction.
A.5.1 Group C1. NODAL COEFFICIENTS

Card C1.1. Nodal coefficients default values.

Nodal variables are initialized with these values. If you solve only flow equation (respectively, transport equation) and the default values for flow variables (respectively, transport variables) are nonzero, the program will issue a warning message.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>DCHP</td>
<td>Default value for steady-state prescribed head nodal coefficient.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>DCHPT</td>
<td>Default value for transient prescribed head nodal coefficient.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>DQQP</td>
<td>Default value for steady state prescribed flow nodal coefficient.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>DQQPT</td>
<td>Default value for transient prescribed flow nodal coefficient.</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.0</td>
<td>DALF</td>
<td>Default value for leakage nodal coefficient.</td>
</tr>
<tr>
<td>51-60</td>
<td>F10.0</td>
<td>DALFT</td>
<td>Idem for transient state.</td>
</tr>
<tr>
<td>61-70</td>
<td>F10.0</td>
<td>DCOE</td>
<td>Default value for steady state external concentration nodal coefficient.</td>
</tr>
<tr>
<td>71-80</td>
<td>F10.0</td>
<td>DCOET</td>
<td>Default value for transient external concentration nodal coefficient.</td>
</tr>
</tbody>
</table>

Card C1.2. Nodal coefficients

A warning message will be issued whenever a non-zero value is specified for a variable of a problem that is not being solved (e.g. if a concentration coefficient is given for a flow problem).

Repeat this card as many times as necessary.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>N</td>
<td>Node number.</td>
</tr>
<tr>
<td>6-14</td>
<td>F9.0</td>
<td>CFCHP(N)</td>
<td>Steady state prescribed head nodal coefficient. Required only if IBCOD(N)=1, 3 or 4 and IOTRS≠1</td>
</tr>
<tr>
<td>15-23</td>
<td>F9.0</td>
<td>CFCHPT(N)</td>
<td>Transient prescribed head nodal coefficient. Required only if IBCOD(N)=1, 3 or 4 and IOTRS≠1.</td>
</tr>
<tr>
<td>24-32</td>
<td>F9.0</td>
<td>CFQQP(N)</td>
<td>Steady state prescribed flow nodal coefficient. Required only if IBCOD(N)=2 or 4 and IOTRS≠1.</td>
</tr>
<tr>
<td>33-41</td>
<td>F9.0</td>
<td>CFQQPT(N)</td>
<td>Transient prescribed flow nodal coefficient. Required only if IBCOD(N)=0 or 4 and IOTRS≠0.</td>
</tr>
</tbody>
</table>
42-50  F9.0  CFALF(N)  Leakage nodal coefficient for steady state. Required only if IBCOD(N)=3 or 4 and IOTRS≠0.
51-58  F9.0  CFALFT(N)  Idem to CFALF for transient state. Required only if IBCOD(N)=3 or 4 and IOTRS≠1.
59-67  F9.0  CFCON(N)  Steady state external concentration nodal coefficient. Required only if IBTCO(N)≠0 and IORTS≠1.
68-76  F9.0  CFCONT(N)  Transient external concentration nodal coefficient. Required only if IBTCO(N)≠0 and IORTS≠1.

Leave a blank line to end this group.

**A.5.2 Group C2. ELEMENT COEFFICIENTS FOR FLOW EQUATION**

Omit this group if only transport is being solved (i.e., if IOEQT=2 in Card A3.3).

**Card C2.1. Element coefficients and default values of flow equation parameters**

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>DTRA</td>
<td>Default transmissivity coefficient value.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>DSTG</td>
<td>Default storage coefficient value.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>DARR</td>
<td>Default steady-state areal recharge coefficient value.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>DARRT</td>
<td>Default transient areal recharge coefficient value.</td>
</tr>
</tbody>
</table>

**Card C2.2. Element coefficients for flow equation**

Only coefficient values different from default values are needed.

Repeat this Card as many times as necessary and end with a blank card.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>N</td>
<td>Element number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>CFTRA(N)</td>
<td>Transmissivity coefficient for element N.</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>CFSTG(N)</td>
<td>Storage coefficient.</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>CFARR(N)</td>
<td>Steady state areal recharge coefficient.</td>
</tr>
<tr>
<td>36-45</td>
<td>F10.0</td>
<td>CFARRT(N)</td>
<td>Transient areal recharge coefficient.</td>
</tr>
</tbody>
</table>

Leave a blank line to end this group.
A.5.3 Group

C3. ELEMENT COEFFICIENTS FOR TRANSPORT EQUATION

Omit this group if you are solving only flow equation (i.e., if IOEQT=1 in Card A3.3), with the exception that IOFLSAT≠0 (in that case, porosity coefficient, CFPOR, has to be defined).

Card C3.1. Default values for transport equation coefficients

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>DDSP</td>
<td>Default dispersivity coefficient value.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>DDFM</td>
<td>Default molecular diffusion coefficient value.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>DPOR</td>
<td>Default porosity coefficient value.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>DCRD</td>
<td>Default retardation coefficient value.</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.0</td>
<td>DCOE</td>
<td>Default external concentration coefficient value.</td>
</tr>
<tr>
<td>51-60</td>
<td>F10.0</td>
<td>DFOD</td>
<td>Default first order decay coefficient value.</td>
</tr>
</tbody>
</table>

Card C3.2. Element coefficient parameter values for transport equation

Only values different from the default ones must be given.

Repeat this Card as many times as necessary and end with a blank card.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>N</td>
<td>Element number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>CFDSP(N)</td>
<td>Dispersivity coefficient for element N.</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>CFDFM(N)</td>
<td>Molecular diffusion coefficient.</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>CFPOR(N)</td>
<td>Porosity coefficient.</td>
</tr>
<tr>
<td>36-45</td>
<td>F10.0</td>
<td>CFCRD(N)</td>
<td>Retardation coefficient.</td>
</tr>
<tr>
<td>46-55</td>
<td>F10.0</td>
<td>CFCOE(N)</td>
<td>External concentration coefficient.</td>
</tr>
<tr>
<td>46-55</td>
<td>F10.0</td>
<td>CFFOD(N)</td>
<td>First order decay coefficient.</td>
</tr>
</tbody>
</table>

Leave a blank line to end this group.

Zonal parameters and estimation options are defined in next group of cards, C4 through C15. Some considerations should be made for all of them. Zonal parameter values are always stored in variables XXXC, where XXX can be any of the root names defined in section 4.3 (TRA, STG, etc). For inverse problem runs, this is the initial value of estimated parameters as provided by the user, and the estimated value at the end of the process. Prior information is stored in variables XXXM. If prior information of a parameter is not given by the user, then the code assumes that prior information is equal to the initial value.
A.5.4 Group C4. DEFINITION OF TRANSMISSIVITY ZONES

Transmissivity or conductivity zone parameters, anisotropy degree, estimation index, measurement errors and non-linear function associated are defined in this group. Its organization is somewhat different from that of other groups, taking into account that this parameter involves anisotropy degree, a variable which doesn’t take place in other kind of parameters, such as storage coefficient. Furthermore, as indicated in what follows, depending on the anisotropy degree, a specific number of zonal parameters has to be defined.

This group of cards must be omitted if you are solving only transport equation, that is, if IOEQT=2 in card A3.3. The structure of this group is explained in the next: for each transmissivity or conductivity zone of the domain, all of its tensor components have to be introduced, one card for each component, sequentially. In addition to this, information associated to each component (i.e., estimation option, prior information etc) is wrote in the same card.

The number of components (in a generic zone NZ) of the tensor is in correspondence with the variable ISOZ(NZ) as indicated in table A1.2. This is entered in the information related to the first component of the tensor, as illustrated in the card description. In table A1.2 is showed the number and meaning of the tensor components, according to the anisotropy degree (ISOZ(NZ)) and to the spatial dimension of the elements to which the tensor of NZ zone are assigned.

Card C4.1. First component of the tensor, anisotropy degree and related information

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Transmissivity (conductivity) zone number.</td>
</tr>
<tr>
<td>6-10</td>
<td>I5</td>
<td>ISOZ(NZ)</td>
<td>Anisotropy degree of the NZ transmissivity zone.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>See table A1.2.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>TRAC(1,NZ)</td>
<td>First component of the transmissivity tensor in zone NZ.</td>
</tr>
<tr>
<td>21-25</td>
<td>F10.0</td>
<td>IVTRA(1,NZ)</td>
<td>Estimation option of the first tensor component in the NZ transmissivity zone. If set to 1 the parameter will be estimated. If set to 0 it will not.</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>STTRA(1,NZ)</td>
<td>Ignore if IVTRA(1,NZ)=0. Standard deviation of prior estimation of the first tensor component in zone NZ.</td>
</tr>
<tr>
<td>36-45</td>
<td>F10.0</td>
<td>TRAM(1,NZ)</td>
<td>Prior estimation of the first transmissivity tensor component in zone NZ. If not given, it is taken as TRAC(1,NZ). Omit if IVTRA(1,NZ)=0.</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFNLTRA(NZ)</td>
<td>Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details.</td>
</tr>
<tr>
<td>51-55</td>
<td>I5</td>
<td>NFTTRA(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>
Card C4.2. $j^{th}$ component of transmissivity tensor and related information

This card is repeated for $j$ varying from 2 to $j_{max}$ (for defining $j_{max}$ see table A9.2). If $j_{max}=1$ omit this card group.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>TRAC($j$,NZ)</td>
<td>$j^{th}$ transmissivity tensor component in zone NZ.</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>IVTRA($j$,NZ)</td>
<td>Estimation option of the $j^{th}$ transmissivity tensor component in zone NZ.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- No estimation is done.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- It will be estimated.</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>STTRA($j$,NZ)</td>
<td>Standard deviation of the $j^{th}$ transmissivity tensor component in zone NZ.</td>
</tr>
<tr>
<td>36-45</td>
<td>F10.0</td>
<td>TRAM($j$,NZ)</td>
<td>Prior information of the $j^{th}$ transmissivity tensor component in zone NZ.</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFNLTRA(NZ)</td>
<td>Non-linear function number (defined in group C17) associated with zone NZ.</td>
</tr>
</tbody>
</table>

A.5.5 Group C5. STORAGE COEFFICIENT ZONES

This group consists of NZSTG cards that are read sequentially, starting from the first zone. If IORTS=0 or when solving only transport equation, this group must be omitted and the user should proceed to the next group. Variables ending with “M” store measured values and variables ending with “C” store computed values. If only simulation is done “M” variables are ignored.

Card C5.1. Storage coefficient zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>STGC(NZ)</td>
<td>Estimated value of storage coefficient at zone NZ. If IVSTG(NZ)=0 (see below), the code works with STGC(NZ) as the actual value of storage coefficient at zone NZ. Otherwise, this value is estimated and STGC(NZ) is the initial value.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVSTG(NZ)</td>
<td>Estimation option for storativity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- STGC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- STGC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STSTG(NZ)</td>
<td>Ignore if IVSTG(NZ)=0. Standard deviation of STGM(NZ) (or that of its logarithm if IOLGSTG was set to 1 in card A5.1).</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>STGM(NZ)</td>
<td>Prior estimate of storage coefficient at zone NZ. If not given, it is taken as STGC(NZ).</td>
</tr>
</tbody>
</table>
A.5.6 Group C6. RECHARGE ZONES

This group consists of NZARR Cards that are read sequentially, starting from the first zone. If NZARR=0, this group must be omitted. Variables ending with “M” represent measured values (prior estimates) and variables ending with “C” represent computed values. If only simulation is done both variables have the same value.

Card C6.1. Areal recharge zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>ARRC(NZ)</td>
<td>Estimated value of areal recharge coefficient at zone NZ</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVARR(NZ)</td>
<td>Estimation option for areal recharge</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- ARRC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- ARRC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STARR(NZ)</td>
<td>Ignore if IVARR(NZ)=0. Standard deviation of ARRM(NZ)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or of its logarithm if IOLGARR was set to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>ARRM(NZ)</td>
<td>Prior estimate of areal recharge at zone NZ. If not given, it is taken as ARRC(NZ).</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLARR(NZ)</td>
<td>Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details. (Not operative in this version).</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTARR(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

A.5.7 Group C7. PRESCRIBED HEAD ZONES

This group consists of NZCHP cards that are read sequentially, starting from the first zone. If NZCHP=0, this group must be omitted.

Card C7.1. Prescribed head zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>CHPC(NZ)</td>
<td>Estimated value of prescribed head at zone NZ.</td>
</tr>
</tbody>
</table>
16-20  I5  IVCHP(NZ)  Estimation option for prescribed head:
0-  CHPC(NZ) is fixed and will not be estimated
1-  CHPC(NZ) will be computed
21-30  F10.0  STCHP(NZ) Ignore if IVCHP(NZ)=0. Standard deviation of
CHPM(NZ) or of its logarithm if IOLGCHP was
set to 1 in card A5.1.
31-40  F10.0  CHPM(NZ) Prior estimate of prescribed head at zone NZ. If
not given, it is taken as CHPC(NZ).
41-45  I5  NFNLCHP(NZ) Non-linear function number (defined in group
C17) associated with zone NZ. See sections A1.9
and 2.2.2 for details. (Not operative in this
version).
46-50  I5  NFTCHP(NZ) Time function number for zone NZ.

A.5.8 Group C8. PRESCRIBED FLOW ZONES
This group consists of NZQQP cards that are read sequentially, starting from the first
zone. If NZQQP=0 this group must be omitted.

Card C8.1. Prescribed flow zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>QQPC(NZ)</td>
<td>Estimated value of prescribed flow at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVQQP(NZ)</td>
<td>Estimation option for prescribed flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0-  QQPC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-  QQPC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STQQP(NZ)</td>
<td>Ignore if IVQQP(NZ)=0. Standard deviation of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QQPM(NZ) or of its logarithm if IOLGQQP was</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>set to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>QQPM(NZ)</td>
<td>Prior estimate of prescribed flow at zone NZ.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If not given, it is taken as QQPC(NZ).</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLQQP(NZ)</td>
<td>Non-linear function number (defined in group</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C17) associated with zone NZ. See sections A1.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and 2.2.2 for details. (Not operative in this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>version).</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTQQP(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

A.5.9 Group C9. LEAKAGE ZONES
This group consists of NZALF cards that are read sequentially, starting from the first
zone. If NZALF=0 this group must be omitted.
Card C9.1. Leakage zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>ALFC(NZ)</td>
<td>Estimated value of leakage at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVALF(NZ)</td>
<td>Estimation option for leakage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- ALFC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- ALFC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STALF(NZ)</td>
<td>Ignore if IVALF(NZ)=0. Standard deviation of ALFM(NZ) or of its logarithm if IOLGALF was set to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>ALFM(NZ)</td>
<td>Prior estimate of leakage at zone NZ. If not given, it is taken as ALFC(NZ).</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLALF(NZ)</td>
<td>Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details.</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTALF(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

A.5.10 Group C10. DISPERSIVITY ZONE PARAMETERS

This group consists of $2 \times NZDSP$ cards that are read sequentially, starting from the first zone. All the zones of longitudinal dispersivity are read first (card C10.1) and all the zones of transversal dispersivity are read next (card C10.2). In one-dimensional problems, even though transversal dispersivity has no sense, card C10.2 must be supplied (the supplied values do not matter to the program).

This group must be omitted if NZDSP=0 or IODSP=0 or only flow equation is solved.

This group has two subgroups, the first contains all the longitudinal dispersivity zones and the second defines all the transversal dispersivity zones. Variables ending with "M" store measured values and variables ending with "C" store computed values. If only simulation is done, the former are ignored.

Card C10.1. Longitudinal dispersivity zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>DSLC(NZ)</td>
<td>Estimated value of longitudinal dispersivity at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVDSL(NZ)</td>
<td>Estimation option for longitudinal dispersivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- DSLC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- DSLC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STDSL(NZ)</td>
<td>Ignore if IVDSL(NZ)=0. Standard deviation of DSLM(NZ) or of its logarithm if IOLGDSP was set to 1 in card A5.1.</td>
</tr>
</tbody>
</table>
31-40 F10.0 DSLM(NZ) Prior estimate of longitudinal dispersivity at zone NZ. If not given, it is taken as DSLC(NZ).

41-45 I5 NFNLDLSL(NZ) Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details. (Not operative in this version).

46-50 I5 NFTDSL(NZ) Time function number for zone NZ.

Card C10.2. Transversal dispersivity zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>DSTM(NZ)</td>
<td>Estimated value of transversal dispersivity at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVDST(NZ)</td>
<td>Estimation option for transversal dispersivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- DSTM(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- DSTM(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STDST(NZ)</td>
<td>Ignore if IVDST(NZ)=0. Standard deviation of DSTM(NZ) or of its logarithm if IOLGDSP was set to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>DSTM(NZ)</td>
<td>Prior estimate of transversal dispersivity at zone NZ.</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLDST(NZ)</td>
<td>Non-linear function number (defined in group C17) associated with zone NZ.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>See sections A1.9 and 2.2.2 for details. (Not operative in this version).</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTDST(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

A.5.11 Group C11. MOLECULAR DIFFUSION ZONES

This group consists of NZDFM cards that are read sequentially, starting from the first zone. This group must be omitted if NZDFM=0. Variables ending with “M” represent measured values (prior estimates) and variables ending with “C” represent computed values. If only simulation is done, the former are ignored.

Card C11.1. Diffusion zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>DFMC(NZ)</td>
<td>Estimated value of molecular diffusion at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVDFM(NZ)</td>
<td>Estimation option for molecular diffusion</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- DFMC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- DFMC(NZ) will be computed</td>
</tr>
</tbody>
</table>
21-30  F10.0  STDFM(NZ)  Ignore if IVDFM(NZ)=0. Standard deviation of DFMM(NZ) or of its logarithm if IOLGDFM was set to 1 in card A5.1.
31-40  F10.0  DFMM(NZ)  Prior estimate of molecular diffusion value at zone NZ. If not given, it is taken as DFMC(NZ).
41-45  I5   NFNLDFM(NZ)  Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details. (Not operative in this version).
46-50  I5   NFTDFM(NZ)  Time function number for zone NZ.

A.5.12 Group C12. POROSITY ZONES

This group consists of NZPOR cards that are read sequentially, starting from the first zone. Variables ending with “M” store measured values and variables ending with “C” store computed values. If only simulation is done, both variables share the same address space. Warning: this group has to be defined although the flow equation does not be solved when IOFLSAT $\neq$ 0.

Card C12.1. Porosity zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>PORC(NZ)</td>
<td>Estimated value of porosity at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVPOR(NZ)</td>
<td>Estimation option for porosity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- PORC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- PORC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STPOR(NZ)</td>
<td>Ignore if IVPOR(NZ)=0. Standard deviation of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORM(NZ) or of its logarithm if IOLGPOR was</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>set to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>PORM(NZ)</td>
<td>Prior estimate of porosity at zone NZ. If not</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>given, it is taken as PORC(NZ).</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLPOR(NZ)</td>
<td>Non-linear function number (defined in group</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C17) associated with zone NZ. See sections A1.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and 2.2.2 for details. (Not operative in this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>version).</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTPOR(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

A.5.13 Group C13. FIRST ORDER LINEAR DECAY COEFFICIENT

This group consists of one card containing the value for the first order reactions coefficient. It should be noted that there is a unique value for the whole aquifer. This card must be
omitted if IOFOD=0 (card A3.3). Variables ending with "M" store measured values and variables ending with "C" store computed values. If only simulation is done, the former are ignored.

Card C13.1. First order decay coefficient

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>FODC(NZ)</td>
<td>Estimated value of first order decay coefficient.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVFOD(NZ)</td>
<td>Estimation option for first order decay coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- FODC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- FODC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STFOD(NZ)</td>
<td>Ignore if IVFOD(NZ)=0. Standard deviation of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FODM(NZ) or its logarithm if IOLGFOFOD was set</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>FODM(NZ)</td>
<td>Prior estimate of first order decay coefficient. If</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>not given, it is taken as FODC.</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLFOD(NZ)</td>
<td>Non-linear function number (defined in group</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C17) associated with zone NZ. See sections A1.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and 2.2.2 for details. (Not operative in this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>version).</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTFOD(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

A.5.14 Group C14. RETARDATION COEFFICIENT ZONES

This group consists of NZCRD cards that are read sequentially, starting from the first zone. This group must be omitted if NZCRD=0. Variables ending with "M" store measured values and variables ending with "C" store computed values. If only simulation is done, the former are ignored.

Card C14.1. Retardation coefficient zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>CRDC(NZ)</td>
<td>Estimated value of retardation at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVCRD(NZ)</td>
<td>Estimation option for retardation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0- CRDC(NZ) is fixed and will not be estimated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1- CRDC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STCRD(NZ)</td>
<td>Ignore if IVCRD(NZ)=0. Standard deviation of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CRDM(NZ) or its logarithm if IOLGCRD was set</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>CRDM(NZ)</td>
<td>Prior estimate of retardation coefficient at zone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NZ. If not given, it is taken as CRDC(NZ).</td>
</tr>
</tbody>
</table>
41-45 I5 NFNLCRD(NZ) Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details. (Not operative in this version).

46-50 I5 NFTCRD(NZ) Time function number for zone NZ.

A.5.15 Group C15. EXTERNAL CONCENTRATION ZONES

This group consists of NZCOE cards that are read sequentially, starting from the first zone. Omit this group if NZCOE=0 (card A3.3). Variables ending with “M” store measured values and variables ending with “C” store computed values. If only simulation is done, the former are ignored.

Card C15.1. External concentration zones

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>COEC(NZ)</td>
<td>Estimated value of external concentration at zone NZ.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IVCOE(NZ)</td>
<td>Estimation option for external concentration 0- COEC(NZ) is fixed and will not be estimated 1- COEC(NZ) will be computed</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>STCOE(NZ)</td>
<td>Ignore if IVCOE(NZ)=0. Standard deviation of COEM(NZ) or of its logarithm if IOLGCOE was set to 1 in card A5.1.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>COEM(NZ)</td>
<td>Prior estimate of external concentration at zone NZ. If it is not given, it is taken as COEC(NZ).</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>NFNLCOE(NZ)</td>
<td>Non-linear function number (defined in group C17) associated with zone NZ. See sections A1.9 and 2.2.2 for details. (Not operative in this version).</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>NFTCOE(NZ)</td>
<td>Time function number for zone NZ.</td>
</tr>
</tbody>
</table>

NOTE: If boundary condition (IBTCO) is 1, 2 or 3, external concentration has concentration units. If boundary condition (IBTCO) is 4, it has mass per unit time units.

A.5.16 Group C16. MATRIX DIFFUSION

Zone numbers for porosity, retardation and matrix diffusion in the grid nodes influenced by matrix diffusion are defined in this group. Omit this group if NTDMT=0.
Card C16.1. Matrix diffusion zones

Repeat this card as many times as matrix diffusion zones and give them in ascendent order.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Number of zone.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>ESPDMT(NZ)</td>
<td>Thickness of zone number NZ.</td>
</tr>
<tr>
<td>15-20</td>
<td>I5</td>
<td>MTDFM(NZ)</td>
<td>Molecular diffusion zone number of matrix diffusion zone NZ.</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>MTCRD(NZ)</td>
<td>Retardation zone number of matrix diffusion zone NZ.</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>MTFOD(NZ)</td>
<td>First order decay zone number of matrix diffusion zone NZ.</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>MTPOR(NZ)</td>
<td>Porosity zone number of matrix diffusion zone NZ.</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>NTDMT(NZ)</td>
<td>Number of terms used in the computation of matrix diffusion zone NZ.</td>
</tr>
<tr>
<td>41-45</td>
<td>I5</td>
<td>IGEOM(NZ)</td>
<td>Geometry of matrix diffusion zone NZ (see section 2.2.5.1) .</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.- Slabs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.- Spheres.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.- Cylinders.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.- Arbitrary. In this case, coefficients $\alpha_n^2$ and $a_n$ must be given in the next lines.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>AN_MZ1</td>
<td>Coefficient $\alpha_n^2$ of the matrix diffusion term approximation</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>AN_MZ2</td>
<td>Coefficient $a_n$ of the matrix diffusion term approximation</td>
</tr>
</tbody>
</table>

A.5.17 Group C17. GENERIC PARAMETERS

This group consists of NZPRG cards. At each card, data relate to the generic parameter PRGC(NZ) are given. NZ is 1 for the first card and NZPRG for the last, increased sequentially. The group is defined only if a non-linear function is included in the problem (I0FLLI$\neq 0$ or I0TRLI$\neq 0$). The physical meaning given to each PRGC should be controlled by the user, according the the non-linear functions data structure presented in cards group C17 and topology, defined in Section A1.9.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NZ</td>
<td>Zone number</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>PRGC(NZ)</td>
<td>Estimated value of generic parameter at zone NZ.</td>
</tr>
</tbody>
</table>
A.5.18 Group C18. NONLINEAR FUNCTIONS

In this group of cards, the user has to enter information about characteristics of the non-linear functions, which, jointly to the variable NFNL***, entered in the previous group, determine the non-linear problem. This information is based on the properties of the non-linear functions, described in detail in section A1.9 of this appendix. Repeat this card NFNL times.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NF</td>
<td>Order number of this non-linear function. It has to vary from 1 to NFNL, increased consecutively. This number is in correspondence with the variable NFNL***(NZ) (i.e., *** may be TRA), associated to zonal parameters.</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>NFNLPRG(1,NF)</td>
<td>Generic parameter zone number of the first generic parameter (sec. A1.9) involved in the non-linear function whose order number is NF.</td>
</tr>
</tbody>
</table>
A.6 FILENAME(4): TIM file

This file contains information regarding time discretization and time functions. Therefore, it can be omitted in steady-state problems, that is, if IOTRS + IORTS = 0 (see card A3.3). The file consists of three group cards: (1) times at which observations are made and number of solution time steps between observations (Card D1.1), (2) time functions (Card D2.1), and (3) temporal weighting parameters (Card D3.1).

A.6.1 Group D1. TIME DISCRETIZATION

This group consists of a single card set (D1.1), which is repeated NINT times, one per observation time, in sequential order (i.e., for NI= 1, ..., NINT). Details on the meaning of TIM and KINT arrays are given in section 4.2.3.1.

Card D1.1. Observation times

This card defines the observation times and time steps between observation times.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NI</td>
<td>Number of time observation.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>TIME(NI)</td>
<td>NI–th observation time.</td>
</tr>
</tbody>
</table>
15-20  I5    KINT(NI)  Number of time steps between successive
observation times TIME(NI) and TIME(NI+1).

21-30  F10.0  DTMXDS(NI) Maximum time step size permitted along the
observation interval NI. For non-linear problems
examine the criterion given in section 4.4.2.

31-35  I5    ISOLEQ(NI,1) Type of flow problem solution:
=0 Standard transient flow is computed if IOTRS
is not zero.
=1 Steady state flow is solved with updated
parameters.
=2 New head initial conditions are read.
=3 New head initial conditions are set to zero.
=4 Flow equation is not solved in this time step.

36-40  I5    ISOLEQ(NI,2) Type of transport problem solution:
=0 Standard transient transport is computed if
IORTS is not zero.
=1 Steady state transport is solved with updated
parameters.
=2 New transport initial conditions are read.
=3 New transport initial conditions are set to zero.
=4 Transport equation is not solved in this
timestep.

41-45  I5    ISOLEQ(NI,3) Flow problem number.

46-50  I5    ISOLEQ(NI,4) Transport problem number.

A.6.2 Group D2. TIME FUNCTIONS

Time functions are used for defining time variability of recharge, external heads and
concentrations, prescribed fluxes, etc. Details on the meaning of time functions are
given in Section 4.2.3.2. Time functions are entered one by one, 7 values for record
file, increasingly in time.

The first card value is FNT(1,1), corresponding to time function number one at
observation time one. This value is followed by FNT(1,2), FNT(2,3), until FNT(1,7),
after which you must skip to next record till the value of FNT(1,NINT), when the first
time function is finished. To begin with the next time function you must start in the
subsequent record with FNT(2,1) and successively to end all time functions. For example,
in a problem with 3 time functions and 11 observation times, card D2.1 is repeated six
times (that is 3 sets, one per time function, of 2 cards, the first containing 7 values and
the the second 4).

Card D2.1. Time functions

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1-70   | 7F10.0 | FNT(K,I) | Value of time function number K at time I. (I=1, ...
NINT; K=1, ..., NFNT) |
A.6.3 Group D3. TIME WEIGHTING PARAMETERS

This group contains the weighting parameters used in the numerical solution of flow and transport equations. These parameters must belong to the interval \([0,1]\).

Card D3.1. Time weighting parameters

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>THETAF</td>
<td>Temporal weighting parameter for flow matrices ((\theta_f) in eq. 2.13). Remember that (0 \leq \theta_f \leq 1).</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>THETAT</td>
<td>Temporal weighting parameter for transport matrices ((\theta_t) in eq. 2.13). Remember that (0 \leq \theta_t \leq 1).</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.0</td>
<td>EPSFLU</td>
<td>Temporal weighting parameter for flow state variable ((\varepsilon_f) in eq. 2.13). Remember that (0 \leq \varepsilon_f \leq 1).</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.0</td>
<td>EPSTRA</td>
<td>Temporal weighting parameter for transport state variable ((\varepsilon_t) in eq. 2.13). Remember that (0 \leq \varepsilon_t \leq 1).</td>
</tr>
</tbody>
</table>

A.7 FILENAME (5): OBS File

This file contains heads and concentrations measurements, or any other quantity that can be expressed as function of these variables. Details on the definition of measurements are given in Section xxx. OBS file must be included if inverse problem is being solved \((\text{IOINV}\neq0)\) or if TRANSIN is run in simulation mode and the user wants to plot simulation results at observation devices \((\text{IOPLH}\text{ or IOPLC} \neq 0)\).

A.7.1 Group E1. GENERAL INFORMATION ON DEVICES AND INTERPOLATION SYSTEMS

Card E1.1. General information on device

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NODEV</td>
<td>Device number</td>
</tr>
<tr>
<td>6-15</td>
<td>A10</td>
<td>DEVNAME</td>
<td>Device name</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>IODATTPY</td>
<td>Data type measured at device</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-Head level/pressure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2-Concentration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3-Water content (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4-Water flux (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5-Mass flux (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>21-25</td>
<td>I5</td>
<td>IFLAG</td>
<td>Describes whether the measurements of this device will be used as calibration data (in this case IFLAG≠ 0). Not used if only simulations are done.</td>
</tr>
<tr>
<td>26-30</td>
<td>I5</td>
<td>IOCALTYT</td>
<td>Method of spatial integration (see section ??)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-Point in space measurement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2-Spatial integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3-Spatial average</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4-Simple summation over unit values (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5-Spatial average over unit values (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6-User defined weights</td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>IOINTTYP</td>
<td>Method of temporal integration (see section ??)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0-Point in time measurement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-Average of all simulation times between beginning and end of time interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2-Average of equally distributed time intervals between beginning and end of time interval</td>
</tr>
<tr>
<td>36-40</td>
<td>I5</td>
<td>IOTINT</td>
<td>Number of integration times. Skip (leave blank) if IOINTTYP equals 0 or 1. If IOINTTYP equals 2, IOTINT is the number of integration time intervals (minimum 2).</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.0</td>
<td>STDEV</td>
<td>Default value for standard deviation of measurement errors in the current device</td>
</tr>
<tr>
<td>51-55</td>
<td>I5</td>
<td>IOCOVTYP</td>
<td>Type of covariance matrix for the observations of current device</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-Independent observations (diagonal covariance matrix)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2-Autoregressive errors (ARMA models) values (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3-Exponential model (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4-Gaussian model (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5-Linear model (not operative in this version)</td>
</tr>
<tr>
<td>56-65</td>
<td>F10.0</td>
<td>CORRCOEKF</td>
<td>Correlation coefficient in ARMA models. Skip (leave blank if IOCOVTYP equals 1)</td>
</tr>
<tr>
<td>65-70</td>
<td>I5</td>
<td>IPROB</td>
<td>Flow/transport problem to which the device is related to</td>
</tr>
</tbody>
</table>

**A.7.2 Group E2. DESCRIPTION OF SPATIAL/TEMPORAL INTEGRATION**

Skip this entire group if IOCALTYT=1.
Card E2.1. Unit type

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOUTYP</td>
<td>Unit type</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0. This unit is not taken into account</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. Point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Surface (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. Volume (not operative in this version)</td>
</tr>
</tbody>
</table>

End this card with IOUTYP=-2 (Previous unit was the last defining this device)

Card E2.2. Unit weight and basic units weighting method

Skip this card if IOCALTYP/=6.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>WEIGHTU</td>
<td>Unit weight</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>IOCALBU</td>
<td>Weighting method for basic units</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-Point measurement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2-Spatial integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3-Spatial averaging</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4-Simple summation over basic unit values (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5-Spatial averaging over basic unit values (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6-User defined weights</td>
</tr>
</tbody>
</table>

End this card with IOUTYP=-2 (Previous unit was the last one defining device)

Card E2.3. Basic units weights

Skip this card if IOCALBU/=6.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-70</td>
<td>7F10.0</td>
<td>WEIGHTU</td>
<td>Basic units weights. Write 7 values per line and use as many lines as needed to write all basic unit weights.</td>
</tr>
</tbody>
</table>

Repeat group 2 once for every unit defining device NODEV.

A.7.3 Group E3. BASIC SPATIAL DATA

Card E3.1. Definition of basic unit type and location
<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>IOBUTYP</td>
<td>Basic units type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1-Point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2-Node</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3-Element</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4-Zone (not operative in this version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-1 Finished unit, but not device</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>BUDAT(1)</td>
<td>X-coordinate of basic unit if IOBUTYP=1. Node</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>number if IOBUTYP=2. Element number if IOBUTYP=3</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>BUDAT(2)</td>
<td>Y-coordinate of basic unit if IOBUTYP=1. Skip</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(leave blank) if IOBUTYP=2 or 3</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>BUDAT(3)</td>
<td>Z-coordinate of basic unit if IOBUTYP=1. Skip</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(leave blank) if IOBUTYP=2 or 3</td>
</tr>
</tbody>
</table>

Repeat card E3.1 once for each basic unit. End this card with –1, after each group of basic units which together constitute a device.

### A.7.4 Group E4. OBSERVATION INFORMATION

#### Card E4.1. Observation information

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>NOOBS</td>
<td>Observation number for device</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>VOBS</td>
<td>Observation value</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>STDEVOBS</td>
<td>Standard deviation of observation (by default,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>equal to STDEV)</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>TOBS1</td>
<td>Observation time if IOINTTYP=0. If observation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>covers a time interval, marks the beginning of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the interval.</td>
</tr>
<tr>
<td>36-45</td>
<td>F10.0</td>
<td>TOBS2</td>
<td>Skip (leave blank) if IOINTTYP=0. If observation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>covers a time interval, marks the end of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>interval.</td>
</tr>
<tr>
<td>46-50</td>
<td>I5</td>
<td>IOTINT</td>
<td>Skip (leave blank) if IOINTTYP=0. If IOINTTYP=1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or 2, IOTINT indicates temporal integration (=1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or averaging (=2, by default)</td>
</tr>
</tbody>
</table>

Sort observations according to TOBS1, beginning with the first observation. End group E4 with –1.

Repeats groups E1 to E4 as many times as devices in the model.
A.8 FILENAME (6): INI File

This file contains the initial conditions for flow and transport equations and the flow velocity and the boundary flows when we are only solving transport equation. Node numbers must be introduced in increasing order. The value is assigned equal to the last introduced value for missing nodes or elements. For instance, if the code finds the following records for card F1.2:

<table>
<thead>
<tr>
<th>Node</th>
<th>Initial head</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>3.0</td>
</tr>
</tbody>
</table>

then the code will assume that nodes 3 and 4 have the same value as node 2, i.e., the program will work as if we had written:

<table>
<thead>
<tr>
<th>Node</th>
<th>Initial head</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>3.0</td>
</tr>
</tbody>
</table>

This rule is followed in all group cards of this file (F1, F2, F3 and F4).

A.8.1 Group F1. INITIAL HEADS

This group consists of two cards. In the first one a title or a comment is defined for the initial conditions and initial heads are defined in the next one. Omit this group if no flow equation is being solved or flow is in steady state or initial conditions are assumed as steady state, i.e., if IOEQT=2 or IOTRS=2 or IOTRS=0.

Card F1.1. Comment line

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>A80</td>
<td>TITULO</td>
<td>Title for the flow initial conditions (It can be a blank line).</td>
</tr>
</tbody>
</table>

Card F1.2. Initial heads

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>I</td>
<td>Node number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>HCAL(I)</td>
<td>Pressure (IOPRHED=1) or piezometric (IOPRHED=0) head.</td>
</tr>
</tbody>
</table>
A.8.2 Group F2. INITIAL CONCENTRATIONS

This group consists of two cards. In the first one a title or a comment is defined for the initial conditions and initial concentrations are defined in the next one. Omit this group if no transport equation is being solved or transport is in steady state or initial conditions are assumed as steady state, i.e., if IOEQT=1 or IORTS=2 or IORTS=0.

Card F2.1. Comment line

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>A80</td>
<td>TITULO</td>
<td>Title for the concentration initial conditions (It can be a blank line).</td>
</tr>
</tbody>
</table>

Card F2.2. Initial concentration

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>I</td>
<td>Node number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>CCAL(I)</td>
<td>Concentration at I–th node number.</td>
</tr>
</tbody>
</table>

If the program will solve the flow equation (i.e., if IOEQT≠2), omit next groups.

A.8.3 Group F3. FLOW VELOCITY

This group is made of two cards. The first one consists of a comment line or a title. In the second one, the flow velocity is defined.

Card F3.1. Comment line

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>A80</td>
<td>TITULO</td>
<td>Title for flow velocities (It can be a blank line).</td>
</tr>
</tbody>
</table>

Card F3.2. Flow velocity

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>N</td>
<td>Element number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>VD(1,N)</td>
<td>Flow velocity at the N–th element, in the X direction.</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.0</td>
<td>VD(2,N)</td>
<td>Flow velocity at the N–th element, in the Y direction (in 1-D elements, this value is not used).</td>
</tr>
<tr>
<td>26-35</td>
<td>F10.0</td>
<td>VD(3,N)</td>
<td>Flow velocity at the N-th element, in the Y direction (in 1-D or 2-D elements, this value is not used).</td>
</tr>
</tbody>
</table>
A.8.4 Group F4. BOUNDARY FLOW

This group is made of two cards. The first one consists of a comment line or a title. In the second one, the boundary flow is defined.

Card F4.1. Comment line

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>A80</td>
<td>TITULO</td>
<td>Title for boundary flow (It can be a blank line).</td>
</tr>
</tbody>
</table>

Card F4.2. Boundary flow

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>I</td>
<td>Node number.</td>
</tr>
<tr>
<td>6-15</td>
<td>F10.0</td>
<td>CAUDAL(I)</td>
<td>Boundary flow at node I.</td>
</tr>
</tbody>
</table>

A.9 CHARACTERISTICS OF THE NON-LINEAR FUNCTIONS

In this section, a detailed description of the non-linear functions, used for defining non-linear flow and transport problems in TRANSIN, is presented. In order to introduce the content of the section, let us remember that any zonal parameter, susceptible of becoming non-linear, is characterized through a variable whose name structure is NFNL***NZ, where *** indicates the type of zonal parameter (i.e., TRA for transmissivity, STG for storage coefficient, etc) and NZ is the order zone number. NFNL*** is an integer which indicates which one is the non linear function affecting the considered zonal parameter. Then, once identified NFNL***, type and characteristics of the non-linear function are implicitly achieved, because, each function is completely defined by its type (NFNLTYPE(NFNL***NZ)), its generic parameters (PRGC(NFNLPRG(k,NFNL***NZ)),k=1...8) and its agreement parameters (PARACD(j,NFNL***NZ))j=1,3).

A.9.1 ASYMPTOTIC FREE AQUIFER TRANSMISSIVITY’

- Function type 1
- NFNLTIP(NFNLtra(nz)) = 1

Description

It is a relationship between the piezometric head in a free aquifer (h) and its saturated thickness B(h). This function allows model aquifers with extreme low piezometric heads, being the phreatic level near or under the bottom of the aquifer.
Figure A.1: Relationship between pressure head \((h)\) and saturated thickness given by function type 1.

The function is valid for 1-D or 2-D horizontal discretized media and the equation solved is based on the Dupuit approximation.

**Mathematical expression**

\[
T = KB(h)
\]

where \(T\) and \(K\) (zonal parameter = TRAC(IS,NZ), IS indicates tensor component) are transmissivity and hydraulic conductivity tensor respectively.

\[
B(h) = (h - h_b) \quad \text{for} \quad h \geq (h_b + \delta)
\]

\[
B(h) = \delta e^{\left(\frac{h-h_b-\delta}{\delta}\right)} \quad \text{for} \quad h < (h_b + \delta)
\]

\(h_b\) is the bottom of the aquifer.

**Parameters**

No generic parameters
Agreement parameters: \(\delta = \text{PARACD}(1,\text{NFNLTRA}(NZ))\)
Another parameters: \(h_b\) is the average aquifer bottom level. It is defined by nodes in cards of the group B1.2. (HBASE).

**A.9.2 “TWO LAYER STORAGE FUNCTION”**

- Function type 21

**Description**

Relationship between the storage coefficient of a free aquifer and the piezometric head considering two extremes values for the coefficient. One corresponds to a high conductive medium (upper layer), valid when levels remain higher than the bottom of this medium. Another corresponds to a low conductive medium (down layer), active...
Figure A.2: Relationship between storage factoring (F) and piezometric head given by function type 21.

when levels fall under that bottom. This function is useful for modelling the drying process in the aquifer, being used in 1-D or 2-D horizontal models.

Mathematical definition

\[ S = S_y \cdot F \]

where \( S_y \) is the high conductive medium storage coefficient (STGC(NZ)) and \( F \) is a factor depending on \( h \).

\[ F = \begin{cases} h - h_b; & (h_b + \delta) \leq h \\ S_m; & (h_b + \delta) \geq h \end{cases} \]

\[ F = \left[ (1 - S_m)(3x - x^3)/4 + \frac{1 + S_m}{2} \right]; \quad (h_b - \delta) < h < (h_b + \delta) \]

where \( x = \frac{h - h_b}{\delta} \)

Parameters

No generic parameters.

Agreement parameters: \( S_m = \text{PARACD}(1,\text{NFNLSTG}(NZ)) \)
\( \delta = \text{PARACD}(2,\text{NFNLSTG}(NZ)) \)

Other parameters: \( h_b \) is the average aquifer bottom level. It is defined by nodes in cards of the group B1.2. (HBASE).

A.9.3 LINEAR RETARDATION FUNCTION

Warning: no operative in this version

Description
Linear relationship between retardation coefficient (R in eq. 2.17) and solute concentration (c).

**Mathematical definition**

\[ R = R_z R_0 \]  
where the zonal parameter, \( R_z = (\text{CRDC(NZ)}) \), corresponds to the sorption equilibrium constant.

\[ R_0 = \left( \frac{1}{R_z} + \frac{P C}{\phi} \right) \]

**Parameters**

- Main zonal parameter: \( R_z = \text{CRDC(NZ)} \)
- Secondary zonal parameters: \( \psi = \text{PORC(nz)} \)
- Generic parameters: \( P = \text{PRGC(NFNLPKG(1,NFNLCRD(NZ)))} = \text{density.} \)
- No agreement parameters.

### A1.9.4. “LANGMUIR RETARDATION FUNCTION”

- Warning: Not operative in this version

**Description**

Relationship between retardation coefficient (R in eq. 2.17) and solute concentration (c) according to the “Langmuir expression”.

**Mathematical definition**

\[ R = R_z R_0 \]  
where \( R_z \) is zonal parameter (CRDC(NZ)).

\[ R = \left[ \frac{P C}{(1 + P C)^2} + \frac{1}{(1 + P C)} \right] ; \quad P = \frac{K_d}{R_z} \quad K_d \text{ is sorption coefficient} \]

**Parameters**

- Generic Parameters: \( P = \text{PRGC(NFNLPKG(1,NFNLCRD(NZ)))} = \frac{K_d}{R_z} \)
- No agreement parameters.

### A.9.4 FREUNDLICH RETARDATION FUNCTION

Warning: Not operative in this version

**Description**
Relationship between retardation coefficient (R in eq. 2.17) and solute concentration (c) according to the “Freundlich expression”.

Mathematical definition

\[ R = R_z R_0; \] where the zonal parameter, \( R_z = (CRDC(NZ)) \), corresponds to the sorption equilibrium constants.
\[ R_0 = \left( \frac{1}{R_z} + \frac{PC^n - 1}{\phi} \right) \]

Parameters

- Main zonal parameter: \( R_z = CRDC(NZ) \)
- Secondary zonal parameter: \( \phi = PORC(NZ) \)
- Generic parameters: \( P = PRGC(NFNLPRG1,NFNLPRG(NZ)) \)
\( n = PRGC(NFNLPRG2,NFNLPRG(NZ)) \).
- No agreement parameters.

A.9.5 PARALLEL FREE AQUIFER TRANSMISSIVITY

- Function type 2
- \( NFNLTYPE(NFNTRA(NZ))=2 \)

Description

It is a relationship whose basic description is identical as function type 1. The only difference is that saturated thickness in function type 1 tends to zero when the aquifer is drained while that of function type 2 tends to a threshold value (b).

Mathematical definition

\[ T = KB(h) \] where \( T \) and \( K \) (zonal parameter=TRAC(IS,NZ), IS indicates tensor component) are transmissivity and hydraulic conductivity tensor respectively.

\[ B(h) = (h - h_b) \quad (h - h_b) > \delta \]

\[ B(h) = \text{Parabolic agreement function; } -\delta \leq (h - h_b) \leq \delta \]

\[ B(h) = b \quad (h - h_b) < -\delta \]

Parameters

No generic parameters

Agreement parameters: \( PARACD1(NFNTRA(NZ))=\delta, \)
\( PARACD(2,NFNTRA(NZ))=b. \)
A.9.6 SINK-SOURCE LIMITED CONDITION

- Function type 41 or 42
- NFNLTYPE(NFNLALF(NZ))=41 or 42

Description

It relates to a sink-source water moving from or to the medium modeled, according to the current pressure or piezometric head. It is often used for modelling interaction between rivers, lakes or aquitards and porous media. The relationship between flow and head is linear but there exist a maximum in-flow value.
Mathematical definition

The sink-source term, \( q \) in equation (2.1), is defined as:

\[
q = \alpha(H - h) + Q_b; \quad h > H - \frac{Q_0}{\alpha} + \delta
\]

\[
q = Q_0 + Q_b; \quad h < H - \frac{Q_0}{\alpha} - \delta
\]

\[
q = \text{parabolic agreement function}; \quad H - \frac{Q_0}{\alpha} - \delta < h < H - \frac{Q_0}{\alpha} + \delta
\]

PARAMETERS

Main zonal parameter \( \alpha = \text{ALFC}(NZ) \). This parameter involves the non linear function number through the variable NFNL***; that is NFNLALF(NZ), NZ=IXALFT(J) and \( J \) is the node where the boundary condition is prescribed.

Secondary zonal parameter: \( H=\text{CHPC}(IXCHPT(J)), \ Q_b=\text{QQPC}(IXQQPT(J)) \)

Generic parameters: \( Q_0=\text{PRGC}(\text{NFNLPRG}(1,\text{NFNLALF}(NZ))) \)

Agreement parameters: \( \delta = \text{PARACD}(1,\text{NFNLALF}(NZ)) \)

A1.9.8. “RELATIVE PERMEABILITY OF VAN GENUCHTEN”

Function type 4

Description

The function gives the relative permeability as a function of the pressure head in an unsaturated medium (suction). The function is coupled with non linear function type 24 (retention curve of van Genuchten).

Mathematical expression

\[
K = K^s K^r(h); \quad \text{where } K^s \text{ (zonal parameter= TRAC(IS,NZ); IS indicates tensor component number) is the saturated conductivity tensor and } K^r \text{ is the relative permeability.}
\]

\[
K^r = \sqrt{\frac{S_w - S_{\text{min}}}{S_{\text{max}} - S_{\text{min}}}} \left\{ 1 - \left( 1 - \left[ \frac{S_w - S_{\text{min}}}{S_{\text{max}} - S_{\text{min}}} \right]^{\frac{1}{\lambda}} \right)^2 \right\}^{\frac{1}{2}}; \quad \frac{S_w - S_{\text{min}}}{S_{\text{max}} - S_{\text{min}}} \leq (S_{\text{max}} - \delta)
\]

\[
K^r \equiv \text{agreement parabolic function}; \quad \frac{S_w - S_{\text{min}}}{S_{\text{max}} - S_{\text{min}}} > (S_{\text{max}} - \delta)
\]

where \( S_w \) is the actual saturation degree, defined based on the retention curve proposed by van Genuchten 1980:

\[
S_w = S_{\text{min}} + (S_{\text{max}} - S_{\text{min}}) \left[ 1 + \left( \frac{h}{h_o} \right)^{\frac{1}{\lambda}} \right]^{-\lambda}
\]
Figure A.5: Pressure head ($h$) and relative conductivity, ($K^r$) given by function type 4.

$S_{min}$ is residual saturation, $S_{max}$ is maximum saturation degree $h_0$ is capillary pressure and $\lambda$ is an empirical parameter.

**Parameters**

Generic Parameters: $\lambda = \text{PRGC(NFNLPRG(1,NFNLTRA(NZ)))}$.

$S_{min} = \text{PRGC(NFNLPRG(2,NFNLTRA(NZ)))}$

$S_{max} = \text{PRGC(NFNLPRG(3,NFNLTRA(NZ)))}$

$h_0 = \text{PRGC(NFNLPRG(4,NFNLTRA(NZ)))}$.

Agreement Parameters: $\delta = \text{PARACD(1,NFNLTRA(NZ))}$

Warning: It is suitable to use $\delta=0.0$ except for problems where $\lambda > 0.6$ and the medium is saturated in part of its domain; in that case, recommended 0.05.

**A.9.7 LINEAR RETENTION CURVE**

Function type 22

**Description**

Relates pressure head ($h$) and specific yield ($\frac{\partial S_w}{\partial h}$; $S_w$ is saturation degree) in a linear form.

where $S_{max}$ is maximum saturation, $S_{min}$ is residual saturation and $\theta_s$ is moisture capacity.

**Mathematical expression**

$\text{Storativity} = (S_w S_s + \phi \frac{\partial S_w}{\partial h})$ see table 2.2. Where $S_w$ and $\partial S_w/\partial h$ is computed based on the function presented in figure A1.9.5, $S_s$ is specific storage coefficient and $\phi$ is porosity.
Parameters

- Main zonal parameters STGC(NZ)=S_s; this parameter involves the non-linear function number through the variable NFNL***; that is NFNLSTG(NZ), NZ=LXSTG(J) and J is element number.

- Secondary zonal parameter= PORC(LXPOR(J))=φ.

  Generic parameters: \( \theta_s = \text{PRGC(NFNLPRG(1,NFNLSTG(NZ)))} \).
  \( S_{min} = \text{PRGC(NFNLPRG(2,NFNLSTG(NZ)))} \).
  \( S_{max} = \text{PRGC(NFNLPRG(3,NFNLSTG(NZ)))} \).
  \( h_0 = \text{PRGC(NFNLPRG(4,NFNLSTG(NZ)))} \).

 Agreement parameters: \( \delta \)

A.9.8 VAN GENUCHTEN RETENTION CURVE

Function type 24

Description

Relationship between pressure head (h) and saturation degree, according to the proposed by van Genuchten (1980). It is coupled with function type 4 (van Genuchten relative permeability).

Mathematical definition

Storativity term \( (S_wS_s + \varphi \frac{\partial S_w}{\partial \varphi}) \), where \( S_w \) is the current saturation degree, \( S_s \) is specific storage coefficient and \( \varphi \) is porosity (see table 2.2). This term is computed by

Figure A.6: Relationship between saturation degree \( (S_w) \) and pressure head \( (h) \), given by function type 22.
means of the function:

\[ S_w = S_{min} + (S_{max} - S_{min}) \left[ 1 + \left( \frac{h}{h_0} \right)^{\frac{1}{\lambda}} \right]^{-\lambda} \]

\( S_{min} \) is residual saturation, \( S_{max} \) is maximum saturation \( h_0 \) is capillary head and \( \lambda \) is an empirical coefficient.

**Parameters**

Main zonal parameter, STGC(NZ)=\( S_s \); this parameter involves the non-linear function number through the variable NFNLSTG(NZ); NZ=LXSTG\( (j) \); \( j \) indicates element.

- Secondary zonal parameters: \( \phi=PORC(LXPOR(j)) \).

Generic Parameters: 
\( \lambda=PRGC(NFNLPRG(1,NFNLSTG(NZ))) \). 
\( S_{min}=PRGC(NFNLPRG(2,NFNLSTG(NZ))) \). 
\( S_{max}=PRGC(NFNLPRG(3,NFNLSTG(NZ))) \). 
\( h_0=PRGC(NFNLPRG(4,NFNLSTG(NZ))) \).

![Diagram of saturation degree (\( S_w \)) vs. pressure head (\( h \))](image.png)

Figure A.7: Relationship between saturation degree (\( S_w \)) and pressure head (\( h \)), given by function type 24.
Appendix B

TRANSIN-IV VARIABLES

B.1 GENERAL RULES FOR VARIABLE NAMES

Most of TRANSIN variable names (mainly variables related to model parameters) are constructed following simple rules. Variable names are divided into prefix, root and sufix. Combination of these parts leads to variable names. Learning these rules is advantageous, because one can identify most of variables very easily. Following is the list of roots, prefixes and sufixes employed in the construction of TRANSIN variable names.

ROOT:

PREFIX:
CF.- Element or nodal coefficient IO.- Different options IX.-
IZ.- Zone number of a nodal parameter IU.- Parameters dimension
IV.- Logical unit number of input files IX.- Parameter estimation index LX.-
NF.- Zone number of a parameter defined by elements NFNL.- Non-linear function NZ.- Number of zones
ST.- Parameter standard deviation XLAM.- Coefficient of parameter objective function

SUFIX:
C.-

Computed parameter values M.- Measured parameter values

B.2 DICTIONARY OF VARIABLES

In this section most of TRANSIN’s variables are listed including a brief description about them.

ACTH(NUMEL).- Aquifer thickness at every two dimensional element. Cross sectional
area of every one-dimensional element. (Input variable Card B3.2).

**AFLU(NUMNP,NBAND1).**- Matrix of finite elements equations for flow problem ("stiffness or conductance matrix", $A$ in equation 2.15). No boundary conditions are included in it.

**AFLUDSC(NUMNP,NBAND1).**- Coefficient matrix of flow system (2.15), most often in its decomposed form.

**ALFC(IZALF).**- Computed leakance zonal parameter. (Input variable Card C9.1).

**ALFM(IZALF).**- Measured (prior information) leakance zonal parameter. (Input variable Card C9.1).

**AREA(NUMEL).**- Element’s area.

**ARRC(IZARR).**- Computed areal recharge zonal parameter. (Input variable Card C6.1).

**ARRM(IZARR).**- Measured (prior information) areal recharge zonal parameter. (Input variable Card C6.1).

**ATRA(NUMNP,NBAND2).**- Matrix of finite elements equations for transport problem ("stiffness or conductance matrix", $E$ in equation 2.30). No boundary conditions are included in it.

**ATRADSC(NUMNP,NBAND2).**- Coefficient matrix of flow system (eq. 2.30), most often in its decomposed form.

**BFLU(MAXNEOP).**- Right hand side of flow discretized equation (2.15).

**BFOBS(LMXNDL,NUOBS).**- Basis functions values at observation points.

**BTRA(MAXNEOP).**- Right hand side of transport discretized equation (2.30).

**CAUDAL(NUMNP).**- Input/output flow at every node.

**CBAS.**- Concentration base line: the program assumes that a measurement at a given point in space and time is missing if its value is smaller than CBAS (card E1.1). (Input variable Card E1.1).

**CCAL(NUMNP).**- Computed concentrations.

**CCALAN(NUMNP).**- Computed concentrations in the previous time step.

**CFALF(NUMNP).**- Leakance nodal coefficient. (Input variable Card C1.2).

**CFARR(NUMEL).**- Steady-state recharge element coefficient. (Input variable Card C2.2).

**CFARRT(NUMEL).**- Transient recharge element coefficient. (Input variable Card C2.2).

**CFCHP(NUMNP).**- Steady state prescribed head nodal coefficient. (Input variable Card C1.2).

**CFCHPT(NUMNP).**- Transient prescribed head nodal coefficient. (Input variable Card C1.2).
CFCOE(NUMEL).- External concentration element coefficient. (Input variable Card C3.2).

CFCON(NUMNP).- Steady state external concentration nodal coefficient. (Input variable Card C1.2).

CFCONT(NUMNP).- Transient external concentration nodal coefficient. (Input variable Card C1.2).

CFCRD(NUMEL).- Retardation element coefficient. (Input variable Card C3.2).

CFDFM(NUMEL).- Molecular diffusion element coefficient. (Input variable Card C3.2).

CFDSP(NUMEL).- Dispersivity element coefficient. (Input variable Card C3.2).

CFPOR(NUMEL).- Porosity element coefficient. (Input variable Card C3.2).

CFQQP(NUMNP).- Steady state prescribed flow nodal coefficient. (Input variable Card C1.2).

CFQQPT(NUMNP).- Transient prescribed flow nodal coefficient. (Input variable Card C1.2).

CFSTG(NUMEL).- Storage element coefficient. (Input variable Card C2.2).

CFTRA(NUMEL).- Transmissivity element coefficient. (Input variable Card C2.2).

CHPC(IZCHP).- Computed prescribed head zonal parameter. (Input variable Card C7.1).

CHPM(IZCHP).- Measured (prior information) prescribed head zonal parameter. (Input variable Card C7.1).

CJAC(NUOBS,NINT,NPAR).- Jacobian matrix of concentrations with respect to estimated parameters at the observation points.

COBS(NUOBS).- Measured concentrations at the observation points. (Input variable Card E2.1).

COBSC(NUOBS).- Computed concentrations at the observation points.

COEC(IZCOE).- Computed external concentration zonal parameter. (Input variable Card C15.1).

COEM(IZCOE).- Measured (prior information) external concentration zonal parameter. (Input variable Card C15.1).

COSDIRX.- Cosinus of the angle formed between the gravity direction and axis X.

COSDIRY.- Cosinus of the angle formed between the gravity direction and axis Y.

COSTHETA.- Angle formed by the direction of the 1-D element NE (defined from node with local order, J, minor to that with major) and gravity direction.

CRDC(IZCRD).- Computed retardation coefficient zonal parameter. (Input variable Card C14.1).
**CRDM(IZCRD)**.- Measured (prior information) retardation coefficient zonal parameter. (Input variable Card C14.1).

**DABSMAX**.- Maximum value of $|h_{k+1,l+1} - h_{k,l+1}|$ considered admissible in order to assume reached convergence, being verified simultaneously the criterium given by DRELMX.

**DABSMX1**.- Maximum value of the absolute change of the state variable between two iterations of Newton-Raphson, considered admissible in order to assume convergence, having guarantee of an accurate solution.

**DABSMX2**.- Idem to DABSMX1 but admitting certain degree of precision error in the solution.

**DADQX(LMXNDL,MAXNEOP)**.- Derivatives of element matrices with respect to $x$ component of Darcy’s velocity.

**DADQY(LMXNDL,MAXNEOP)**.- Derivatives of element matrices with respect to $y$ component of Darcy’s velocity.

**DALF**.- Leakance default value. (Input variable Card C1.1).

**DARR**.- Steady state recharge default value. (Input variable Card C2.1).

**DARRT**.- Transient recharge default value. (Input variable Card C2.1).

**DCHP**.- Steady state prescribed head default value. (Input variable Card B1.1).

**DCHPT**.- Transient prescribed head default value. (Input variable Card C1.1).

**DCITMX**.- Testing Factor. Set to 1.0 in all cases.

**DCOE**.- Steady state external concentration default value. (Input variable Card C1.1).

**DCOET**.- Transient external concentration default value. (Input variable Card C1.1).

**DCRD**.- Retardation coefficient default value. (Input variable Card C3.1).

**DDSP**.- Dispersivity default value. (Input variable Card C3.1).

**DDFM**.- Molecular diffusion default value. (Input variable Card C3.1).

**DERC(NUMNP,NPAR,2)**.- Concentration derivatives with respect to estimated parameters.

**DERH(NUMNP,NPARF,2)**.- Head derivatives with respect to estimated flow parameters.

**DFLU(NUMNP,ID)**.- Matrix of finite elements equations for flow problem related to storage term (matrix $D$ in equation 2.15) If flow scheme is consistent, ID=NBAND1, if lumped, ID=1.

**DFMC(IZDFM)**.- Computed molecular diffusion zonal parameter. (Input variable Card C11.1).

**DFMM(IZDFM)**.- Measured (prior information) molecular diffusion zonal parameter. (Input variable Card C11.1).
DHITMX.- Testing Factor. Set to 1.0 in all cases.

DPOR.- Porosity default value. (Input variable Card C3.1).

DQQP.- Steady state prescribed flow default value. (Input variable Card C1.1).

DQQPT.- Transient prescribed flow default value. (Input variable Card C1.1).

DERELMAX.- Maximum value of $|{(h_i^{k+1,l+1} - h_i^{k+1,l})/(h_i^{k+1,l+1} - h_i^k)}|$ considered admissible in order to assume reached convergence, being verified simultaneously the criterium given by DABSMAX.

DRELMX1.- Idem to DABSMX1 but considering relative changes of the state variable.

DRELMX2.- Idem to DRELMX1 but admitting certain degree of precision error in the solution.

DSLC(IZDSP).- Computed longitudinal dispersivity zonal parameter. (Input variable Card C10.1).

DSLM(IZDSP).- Measured (prior information) longitudinal dispersivity zonal parameter. (Input variable Card C10.1).

DSTC(IZDSP).- Computed transversal dispersivity zonal parameter. (Input variable Card C10.2).

DSTG.- Storage default value. (Input variable Card C2.1).

DSTM(IZDSP).- Measured (prior information) transversal dispersivity zonal parameter. (Input variable Card C10.2).

DTRA.- Transmissivity default value. (Input variable Card C2.1).

DTRA(NUMNP,ID).- Matrix of finite elements equations for transport problem related to mass storage term (matrix $F$ in equation 2.30). If transport scheme is consistent, ID=NBAND1, if lumped, ID=1.

EPSFLU.- Temporal weighting parameter for flow matrices.

EPSTRA.- Temporal weighting parameter for transport matrices.

ERRDMS.- Maximum truncation error when computing the series for matrix diffusion. (Input variable Card B1.5).

ESPDMT(NZDMT).- Matrix diffusion thickness at every matrix diffusion zone. (Input variable Card B1.4).

FCTNCV.- Time increment increasing factor which the program employs to modify the time increment when it is less than the desirable or than the maximum desirable. Recommended 1.5.

FCTDEC.- Time increment decreasing factor which the program employs to modify the time increment when convergence problems appear. Recommended 0.5.

FCTDVNR.- Testing factor. Set to 1.0 in all cases.

FCTINC.- This factor increase the time increment desirable (only for this inverse iteration) when this is less than the actual time increment. Anyway, the time increment
desirable can be as maximum equal to the maximum desirable. Recommended 1.5.

**FNT(NFNT,NINT).-** Time functions. (Input variable Card A4.1).

**HBAS.** Head base line: the program assumes that a measurement at a given point in space and time is missing if its value is smaller than HBAS (card E1.1). (Input variable Card E1.1).

**HBASE.** Bottom level of the aquifer in node N. It should be defined only in problems under follow conditions:
1. flow equation
2. In node N, Dupuit aprox is adopted (that is, free aquifer treated as 1-D or 2-D medium.
3. The problem is considered non-linear using non-linear functions type 1 or 6 for computing transmissivity in, at least, one element associated to node N.

**HCAL(NUMNP).-** Computed heads.

**HCALAN(NUMNP).-** Computed heads in the previous time step.

**HESS(NPAR*(NPAR+1)/2).-** Hessian matrix of objective function (actually an approximation, eq. 3.16).

**HJAC(NUOBS,NINT,NPARf).-** Jacobian matrix of heads with respect to estimated parameters at the observation points.

**HOBS(NUOBS).-** Measured heads at observation points. (Input variable Card E2.1).

**HOBSC(NUOBS).-** Computed heads at observation points.

**IBCOD(NUMNP).-** Flow Boundary condition index (Input variable Card B1.3).
- If IBCOD=0- No flow ($\alpha = 0$, $Q = 0$)
- If IBCOD=1- Prescribed head ($\alpha = \infty$)
- If IBCOD=2- Prescribed flow ($\alpha = 0$, $Q \neq 0$)
- If IBCOD=3- Mixed boundary condition ($\alpha = 0$, $Q \neq 0$)
- If IBCOD=4- Mixed boundary with flow term ($\alpha \neq 0$, $Q \neq 0$)

**IBTCO(NUMNP).-** Transport boundary condition index (Input variable Card B1.3).
- If IBTCO=0- No flow ($\beta = 0$, $M = 0$)
- If IBTCO=1- Prescribed concentration ($\beta = \infty$)
- If IBTCO=2- Mass flow ($\beta \neq 0$, $M = 0$)
- If IBTCO=4- Input mass ($\beta = 0$, $M \neq 0$)

**INPWR.** Control of inpt data printing (Input variable Card A3.2).
- If INPWR=0 Nothing
- If INPWR=1 It prints input and interpolated data
- If INPWR>1 The same plus default assignments.

**IOARR.** No recharge if IOARR=0.

**IOCRITRAP.** Option of treatment on the direct problem convergence criteria. It set to 0 they are considered constants; when set to 1 they may vary at each iteration of the inverse problem. If only simulation is made, or are unic iteration of the inverse H has to be equal to 0.

**IODFM.** No molecular diffusion if IODFM=0.
IODIM.- Maximum dimension of one element included in the problem.

IODSP.- No dispersivity if IODSP=0.

IOFLLI.- Set to 0 when the flow problem is linear, otherwise set to 1. On the other hand, it is non linear when, at least, one flow function defined in Section A1.9 is included in the problem.

IOHOB.- No head data if IOHOB=0.

IOCOB.- No concentration data if IOCOB=0.

IOEQT.- Type of problem to be solved.
  (IOEQT=0 Checking data only
   IOEQT=1 Only flow
   IOEQT=2 Only transport
   IOEQT=3 Flow and transport)

IOFLSAT.- Indicates if one consider the possibility that part (or all) of the domain reach unsaturated state.

IOFOBJ.- If fixed to 0 the code solves completely the direct problem at each iteration of the inverse’s one. If set to 1 it computes completely the simulation only for successful iterations of the inverse.

IOINV.- Inverse problem option.

IOINV≥0 Only simulation computing numerically (Gauss quadrature) the finite element method integrals.

IOINV=1 Flow parameters estimation

IOINV=2 Transport parameters estimation

IOINV=3 Flow and transport parameters estimation

IOINV<0 Only simulation, computing analytically the finite element method integrals).

IOLAM.- If IOLAM=0 no first order decay occur. (Input variable Card A3.3).

IOLGPRG.- Same as IOLGTRA for generic parameters.

IOOBS.- If IOOBS=1, observation points coincide with nodes, otherwise observation points must be given. (Input variable Card A3.3).

IOPINITH.- Indicates the option selected for initializing pressure or piezometric heads \( h^{k+1,0} \) at the beginning of the Newton-Raphson iterative process, for each time step.

IOPRHED.- Indicates whether the flow state variable is pressure (set to 1) or piezometric head (set to 0). If IOPRHED=1 and IODIM≤2 the gravity direction has to be defined in file. (When IODIM=3 it is assumed -Z as the gravity direction).

IOSUCHUM.- Indicates if measures are given in terms of pressure or piezometric heads (IOSUCHUM=0), or in terms of saturation degree (IOSUCHUM=1).

IORTS.- Transport regime (Input variable Card A3.3).
IORTS=0 Steady state transport
IORTS=1 Transient transport with prescribed initial conditions
IORTS=2 Transient with initial conditions equals to the steady problem.

IOTRLI.- IDEM to IOFLLI, in the case of transport.

IOTRS.- Flow regime (Input variable Card A1.5).
IOTRS=0 Steady state flow
IOTRS=1 Transient flow with prescribed initial conditions
IOTRS=2 Transient flow with initial conditions equals to the steady problem.

IOSTC.- If equal to 1, concentration standard deviations are equal for all observation points. (Input variable Card A3.3).

IOSTH.- If equal to 1, head standard deviations are equal for all observation points. (Input variable Card A3.3).

IOWAR.- Control of warning printing (Input variable Card A3.2).
  IOWAR=0 it does not write warnings.
  IOWAR=1 it writes warnings

IOWNR.- Option of printing information about the direct problem iterative process evolution. If set to N, the code will print detailed information of the process every N solution times.

ISOT.- Maximum anisotropy index in all transmissivity zones. (Input variable Card A4.1).

ISOZ(IZTRA).- Degree of anisotropy at every zone. (Input variable Card C4.1).
ISOZ=1 Isotropic
ISOZ=2 Anisotropic with principal directions parallel to coordinate axes
ISOZ=3 Anisotropic with arbitrary principal directions

ITRAPMX.- Maximum number of direct problem iterations, premissed in order to get convergence, before reducing the time step.

IUCAL.- Logical unit number of initial conditions file (FILENAME(6)).

IUDIM.- Logical unit number of dimensions and options file (FILENAME(1)).

IUGRI.- Logical unit number of mesh file (FILENAME(2)).

IUOBS.- Logical unit number of observations file (FILENAME(5)).

IUPAR.- Logical unit number of parameters and coefficients file (FILENAME(3)).

IUTIM.- Logical unit number of time information file (FILENAME(4)).

IVALF(IZALF).- Leakance estimation index. If not zero in a zone, parameter ALFC will be estimated in this zone. (Input variable Card C9.1).
IVARR(IZARR).- Recharge estimation index. If not zero in a zone, parameter ARRC will be estimated in this zone. (Input variable Card C6.1).

IVCHP(IZCHP).- Prescribed head estimation index. If not zero in a zone, parameter CHPC will be estimated in this zone. (Input variable Card C7.1).

IVCOE(IZCOE).- External concentration estimation index. If not zero in a zone, parameter COEC will be estimated in this zone. (Input variable Card C15.1).

IVCRD(IZCRD).- Retardation estimation index. If not zero in a zone, parameter CRDC will be estimated in this zone. (Input variable Card C14.1).

IVDFM(IZDFM).- Molecular diffusion estimation index. If not zero in a zone, parameter DFMC will be estimated in this zone. (Input variable Card C11.1).

IVDSL(IZDSP).- Longitudinal dispersivity estimation index. If not zero in a zone, parameter DSLC will be estimated in this zone. (Input variable Card C10.1).

IVDST(IZDSP).- Transversal dispersivity estimation index. If not zero in a zone, parameter DSTC will be estimated in this zone. (Input variable Card C10.2).

IVLAM.- First order decay coefficient estimation index. If not zero, parameter XLAMC will be estimated. (Input variable Card C13.1).

IVPOR(IZPOR).- Porosity estimation index. If not zero in a zone, parameter PORC will be estimated in this zone. (Input variable Card C12.1).

IVPRG(NZ).- Estimation option for generic parameter at zone NZ.

IVQQP(IZQQP).- Prescribed flow estimation index. If not zero in a zone, parameter QQPC will be estimated in this zone. (Input variable Card C8.1).

IVSTG(IZSTG).- Storage estimation index. If not zero in a zone, parameter STGC will be estimated in this zone. (Input variable Card C5.1).

IVTRA(ISOT,IZTRA).- Transmissivity estimation index. If not zero in a zone, parameter TRAC will be estimated in this zone. (Input variable Card C4.1).

IXALF(NUMNP).- Leakance zone number of every node. (Input variable Card B1.3).

IXALFT.- Idem to IXALF for transient problems.

IXCHP(NUMNP).- Prescribed head zone number of every node (steady state). (Input variable Card B1.3).

IXCHPT(NUMNP).- Prescribed head zone number of every node (transient). (Input variable Card B1.3).

IXCON(NUMNP).- External concentration zone number of every node (steady state). (Input variable Card B1.3).

IXCONT(NUMNP).- External concentration zone number of every node (transient). (Input variable Card B1.3).

IXDMT(NUMNP).- Matrix diffusion zone number of every node. (Input variable Card B1.3).
IXQQP(NUMNP).- Prescribed flow zone number of every node (steady state). (Input variable Card B1.3).

IXQQPT(NUMNP).- Prescribed flow zone number of every node (transient). (Input variable Card B1.3).

IZALF.- Used only in DIMENSION statements. It coincides with NZALF when NZALF ≠ 0, otherwise its value is set to 1.

IZARR.- Used only in DIMENSION statements. It coincides with NZARR when NZARR ≠ 0, otherwise its value is set to 1.

IZCHP.- Used only in DIMENSION statements. It coincides with NZCHP when NZCHP ≠ 0, otherwise its value is set to 1.

IZCOE.- Used only in DIMENSION statements. It coincides with NZCOE when NZCOE ≠ 0, otherwise its value is set to 1.

IZCRD.- Used only in DIMENSION statements. It coincides with NZCRD when NZCRD ≠ 0, otherwise its value is set to 1.

IZDFM.- Used only in DIMENSION statements. It coincides with NZDFM when NZDFM ≠ 0, otherwise its value is set to 1.

IZDSP.- Used only in DIMENSION statements. It coincides with NZDSP when NZDSP ≠ 0, otherwise its value is set to 1.

IZQQP.- Used only in DIMENSION statements. It coincides with NZQQP when NZQQP ≠ 0, otherwise its value is set to 1.

IZSTG.- Used only in DIMENSION statements. It coincides with NZSTG when NZSTG ≠ 0, otherwise its value is set to 1.

IZTRA.- Used only in DIMENSION statements. It coincides with NZTRA when NZTRA ≠ 0, otherwise its value is set to 1.

KINT(NINT).- Number of solution time increments, between successive observation times. Number of solution times between observation times TIME(I) and TIME(I+1) is defined by KINT(I). (Input variable Card D1.1).

KXX(LMXNDL,NUMEL).- Node numbers of every element (counterclockwise order). (Input variable Card B3.2).

LMXNDL.- Maximum number of nodes per element. (Input variable Card A4.1).

LOBS(NUOBS).- Element to each observation point belongs to.

LXARR(NUMEL).- Recharge zone number of every element (steady state). (Input variable Card B3.3).

LXARRT(NUMEL).- Recharge zone number of every element (transient). (Input variable Card B3.3).

LXCOE(NUMEL).- External concentration zone number of every element. (Input variable Card B3.3).

LXCRD(NUMEL).- Retardation zone number of every element. (Input variable Card
LXDFM(NUMEL).- Molecular diffusion zone number of every element. (Input variable Card B3.3).

LXDSP(NUMEL).- Dispersivity zone number of every element. (Input variable Card B3.3).

LXPOR(NUMEL).- Porosity zone number of every element. (Input variable Card B3.3).

LXSTG(NUMEL).- Storage zone number of every element. (Input variable Card B3.3).

LXTRA(NUMEL).- Transmissivity zone number of every element. (Input variable Card B3.3).

MAINF.- Logical unit number of main output file.

MAXNEOP.- Maximum of NUMNP, NUOBS y NPAR.

MINCAT.- Minimum number of consecutive convergences achieved after a time step reduction, required before starting to increase the time increment when it is less than that desirable.

MTPOR(NZDMT).- Porosity zone number of every matrix diffusion zone. (Input variable Card C16.1).

MTCRD(NZDMT).- Retardation zone number of every matrix diffusion zone. (Input variable Card C16.1).

MTDFM(NZDMT).- Molecular diffusion zone number of every matrix diffusion zone. (Input variable Card C16.1).

MXNRTF.- Maximum consecutive time reductions permitted in order to achieve convergence.

MXNRTT.- Idem to MXNRTF, for the transport equation.

NBAND.- Matrices bandwidth (maximum difference between node numbers in any element). (Input variable Card A4.1).

NBAND1.- NBAND+1

NBAND2.- 2× NBAND+1

NFNL.- Total number of non-linear functions required for modeling the problem.

NFNLARR.- Non-linear function order number affecting the recharge in elements assigned to the NZ recharge zone (not operative at this version).

NFNLCOE.- Non-linear function order number (from those defined in group C17) affecting the external concentration in nodes or elements assigned to the NZ concentration zone number. Not operative at this version.

NFNLCRD.- Non-linear function order number (from those defined in group C17), affecting the retardation coefficient in elements assigned the NZ retardation CORF, zone.
**NFNLPRG.** - Zone number of the generic parameter involved in the non-linear function whose order number is NF.

**NFNLSTG.** - Non-linear function order number (from those defined in group C17) affecting the storativity of elements assigned to zone NZ of the storage coefficient.

**NFNLTP.** - Type of non-linear function of the function whose order is NF. It may be:
1. Asymptotic free aquifer transmissivity.
2. Two layer storage function.
3. Linear retardation function.
5. Freundchly retardation function.
6. Parallel free aquifer transmissivity.
7. Sink source limited condition.
8. Relative permeability of van Genuchten.
9. Linear retention curve.
10. van Genuchten retention curve.

**NFNLTRA.** - Non-linear function order number (from those defined in group C17) affecting the transmissivity tensor component in zone NZ.

**NFNT.** - Number of time functions used for describing time dependence of transient parameters. (Input variable Card A4.1).

**NFTARR(IZARR).** - Time function number at every recharge zone. (Input variable Card C6.1).

**NFTCHP(IZCHP).** - Time function number at every prescribed head zone. (Input variable Card C7.1).

**NFTCOE(IZCOE).** - Time function number at every external concentration zone. (Input variable Card C15.1).

**NFTQQP(IZQQP).** - Time function number at every prescribed flow zone. (Input variable Card C8.1).

**NINT.** - Number of observation times, including initial time (steady-state) as the first observation time. (Input variable Card A4.1).

**NPAR.** - Total number of parameters to be estimated. (Input variable Card A4.1).

**NPARF.** - Flow parameters to be estimated. (Input variable Card A4.1).

**NPARFP.** - Number of uncertain generic parameter zones involved in the non-linear flow problem.

**NPARPRG.** - Total number of uncertain generic parameter zones.

**NPARTRA.** - Number of transmissivities zones to be estimated with non-diagonal prior information covariance matrix. It is unemployed if the prior information covariance matrix is diagonal or if NZTRA is equal to 1. (Input variable Card A4.1).

**NTDMT.** - Number of terms included in the series expansion for solving matrix diffusion equation. It should be set to 0 if no matrix diffusion processes will be included. (Input
NUMEL.- Number of elements.
NUMNP.- Number of nodes. (Input variable Card A4.1).
NUOBS.- Number of observation points. (Input variable Card A4.1).
NZALF.- Number of leakance zones. (Input variable Card A4.2).
NZARR.- Number of recharge zones. (Input variable Card A4.2).
NZCHP.- Number of prescribed head zones. (Input variable Card A4.2).
NZCOE.- Number of external concentration zones. (Input variable Card A4.2).
NZCRD.- Number of retardation coefficient zones. (Input variable Card A4.2).
NZDFM.- Number of molecular diffusion zones. (Input variable Card A4.2).
NZDMT.- Number of matrix diffusion zones. (Input variable Card A4.2).
NZDSP.- Number of dispersivity zones. (Input variable Card A4.2).
NZPOR.- Number of porosity zones. (Input variable Card A4.2).
NZPRG.- Total number of generic parameter zones. It is only active for non-linear flow or transport problems.
NZQQP.- Number of prescribed flow zones. (Input variable Card A4.2).
NZSTG.- Number of storage coefficient zones. (Input variable Card A4.2).
NZTRA.- Number of transmissivity zones. (Input variable Card A4.2).
OBJHED1.- Value of the objective function considered close to those corresponding to the minimum.
OBJHED2.- Value of the objective function considered notably bigger than that expected finding in the minimum of the problem.
PAR(NPAR).- Parameter’s increment at every iteration.
PARACD.- Value of the first agreement parameter involved in the non-linear function whose order number is NF.
PARAUX(NPAR).- Array with computed parameters at the previous step of Marquardt’s algorithm. During the first iteration it contains the initial values of parameters.
PORC(IZPOR).- Computed porosity zonal parameter. (Input variable Card C12.1).
PORM(IZPOR).- Measured (prior information) porosity zonal parameter. (Input variable Card C12.1).
PRGC(NZ).- Estimated value of generic parameter at zone NZ.
PRGM(NZ).- Prior information of the generic parameter corresponding to zone NZ.
QQPC(IZQQP).- Computed prescribed flow zonal parameter. (Input variable Card
QQPM(IZQQP).- Measured (prior information) prescribed flow zonal parameter. (Input variable Card C8.1).

QX2(NUMEL).- Square of x component of Darcy's velocity, divided by Darcy's velocity modulus at every element.

QXY(NUMEL).- Product of x and y components of Darcy's velocity, divided by Darcy's velocity modulus at every element.

QY2(NUMEL).- Square of y component of Darcy's velocity, divided by Darcy's velocity modulus at every element.

RESIDMXF.- Maximum value of the mass balance error, over the volume represented by a node, $|f_i(h)|_{max}$ (see eq. 2.13), considered admissible in order to assume reached convergence of the non-linear flow equation.

RESIDMXT.- Maximum value of the mass balance error, over the volume represented by a node, considered admissible, in order to assume reached convergence of the non-linear transport equation.

RESIDMX1F.- Mass balance error, per unit time permissed over the volume corresponding to a node, having guarantee of suitable accuracy.

RESIDMX2F.- Idem to ResidMX1F but admitting certain degree of precision error in the solution.


STARR(IZARR).- Standard deviation errors of recharge prior information (ARRM). (Input variable Card C6.1).

STCHP(IZCHP).- Standard deviation errors of prescribed head prior information (CHPM). (Input variable Card C7.1).

STCOBS(NUMOBS).- Concentration standard deviation errors at observation points. (Input variable Card E2.1).

STCOE(IZCOE).- Standard deviation errors of external concentration prior information (COEM). (Input variable Card C15.1).


STDSL(IZDSP).- Standard deviation errors of longitudinal dispersivity prior information (DSLM). (Input variable Card C10.1).

STDST(IZDSP).- Standard deviation errors of transversal dispersivity prior information (DSTM). (Input variable Card C10.2).

STGC(IZSTG).- Computed storage zonal parameter. (Input variable Card C5.1).
STGM(IZSTG).- Measured (prior information) storage zonal parameter. (Input variable Card C5.1).

STHOBS(NUOBS).- Head standard deviation errors at observation points. (Input variable Card E2.1).

STLAM.- Standard deviation errors of first order reactions prior information (XLAMM). (Input variable Card C13.1).

STPOR(IZPOR).- Standard deviation errors of porosity prior information (PORM). (Input variable Card C12.1).

STPRG.- Ignore if IVPRG(NZ)=0. Standard deviation of the prior information of PRGC(NZ).

STQQP(IZQQP).- Standard deviation errors of prescribed flow prior information (QQPM). (Input variable Card C8.1).

STSTG(IZSTG).- Standard deviation errors of storage prior information (STGM). (Input variable Card C5.1).

STTRA(ISOT,IZTRA).- Standard deviation errors of transmissivity prior information (TRAM). (Input variable Card C4.3).

THETAF.- Temporal weighting parameter for flow equation. It must belong to interval [0,1] (Input variable Card D3.1).

THETAT.- Temporal weighting parameter for transport equation. It must belong to interval [0,1] (Input variable Card D3.1).

TIME(NINT).- Observation times. (Input variable Card D1.1).

TRAC(ISOT,IZTRA).- Computed transmissivity zonal parameter. (Input variable Card C4.3).

TRAM(ISOT,IZTRA).- Measured (prior information) transmissivity zonal parameter. (Input variable Card C4.2).

VD(2,NUMEL).- Darcy’s velocity at every element.

WORK(—).- Workspace array used in some subroutines. It is dimensioned with the maximum value between (NPAR × (NPAR+1))/2) and NUMNP × (NBAND+1).

X(NUMNP).- Nodal X-coordinates. (Input variable Card B1.2).

XLAMALF.- Coefficient λ_α of mixed B.C. objective function. (Input variable Card A9.1).

XLAMARR.- Coefficient λ_q of areal recharge objective function. (Input variable Card A9.1).


XLAMCHP.- Coefficient λ_H of prescribed head objective function. (Input variable Card A9.1).

XLAMCON.- Coefficient λ_c of relative weight between head and concentration objective
functions. It must be nonzero if concentration data is employed for estimation. (Input variable Card A9.2).

**XLAMCOE.** - Coefficient $\lambda_{ce}$ of external concentration objective function. (Input variable Card A9.2).

**XLAMCRD.** - Coefficient $\lambda_R$ of retardation objective function. (Input variable Card A9.2).

**XLAMDFM.** - Coefficient $\lambda_D$ of molecular diffusion objective function. (Input variable Card A9.2).

**XLAMDSP.** - Coefficient $\lambda_\alpha$ of dispersivity objective function. (Input variable Card A9.2).

**XLAMLAM.** - Coefficient $\lambda_\lambda$ of linear decay objective function. (Input variable Card A9.2).

**XLAMM.** - Measured (prior information) linear decay parameter. (Input variable Card A9.2).

**XLAMPOR.** - Coefficient $\lambda_\phi$ of porosity objective function. (Input variable Card A9.2).

**XLAMPRGF.** - Coefficient of uncertain generic parameters involved in the flow problem.

**XLAMPRGT.** - Coefficient of uncertain generic parameters involved in the transport problem.

**XLAMQQP.** - Coefficient $\lambda_Q$ of prescribed flow objective function. (Input variable Card A9.1).

**XLAMSTG.** - Coefficient $\lambda_S$ of storage coefficient objective function. (Input variable Card A9.1).

**XLAMTRA.** - Coefficient $\lambda_T$ of transmissivity objective function. (Input variable Card A9.1).

**XNORVD(NUMEL).** - Darcy’s velocity modulus at every element.

**XOBS(NUOBS,NINT).** - X-coordinates of observation points. (Input variable Card E2.1).

**Y(NUMNP).** - Nodal Y-coordinates. (Input variable Card B1.2).

**YOBS(NUOBS,NINT).** - Y-coordinates of observation points. (Input variable Card E2.1).

**Z(NUMNP).** - Nodal Z-coordinates. (Input variable Card B1.2).

**ZEROF.** - Minimum value of $|h^{k+1,l+1} - h^{k+1,l}|$ considered valid in order to apply the convergence criterium represented by the variable DRELMAX, read in card A9.3.

**ZEROT.** - Idem to ZEROF but in the transport equation.

**ZOBS(NUOBS,NINT).** - Z-coordinates of observation points. (Input variable Card E2.1).