Università degli Studi di Parma
Facoltà di Ingegneria
Dottorato di Ricerca in Ingegneria Civile - XXII Ciclo
Curriculum: Protezione Idraulica del Territorio (ICAR/02)

Marco D'Oria

Characterization of Aquifer Hydraulic Parameters: from Theis to Hydraulic Tomography

Dissertazione per il conseguimento del titolo di Dottore di Ricerca

Tutore: Prof. Ing. Maria Giovanna Tanda
Co-Tutori: Dott. Michael N. Fienen, Dott. Andrea Zanini
Coordinatore del Dottorato: Prof. Ing. Paolo Mignosa

Parma, Gennaio 2010

Don't look where you fall, but where you slipped.

African Proverb.

Acknowledgments

This dissertation contains the results of the Ph.D. activity that I performed during the last three years. The work it would be more difficult without the encouragement from many people.

Special thanks give to my Advisor, Prof. Maria Giovanna Tanda who suggested and gave me the chance to start the Ph.D.; she always gave me valuable advice and she supported and helped me when it was necessary.

Many thanks to the other Professors and Colleagues, that I have met in this period, for their kindness and helpfulness; thanks to Dr. Andrea Zanini that, in addition, helped me with the field activity.

I'm also particularly thankful to Dr. Michael Fienen that I have met during my six months exchange visitor program at the USGS Wisconsin Water Science Center (Middleton, WI, USA). He was (and still is) a rich source of suggestions and encouragements; I would also like to thank him for his remarkable hospitality and for the review of this dissertation.

Many thanks to AIPO (Agenzia Interregionale per il fiume PO) that gave me the opportunity to perform tests and to collect data in the well field of the Boretto Research Site.

Finally, I wish to thank my parents for their encouragement and all my friends, the old ones and those that I have met in these years without whom I would never have enjoyed many moments.

Parma, January 2010

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Groundwater is in many parts of the world an important source of fresh water for several purpose such as domestic and industrial use and irrigation.

Pollution and bad management of groundwater are only two of the problems that affect the aquifers around the world. Detailed information about the spatial distribution of hydraulic properties in subsurface are of crucial importance for a proper management of groundwater and for the prediction of the solutes transport in aquifer and therefore for the design of effective remediation systems. Different methods have been used for characterizing aquifer hydraulic parameters but the most used are the interpretation of the pumping tests. They consist of measuring the drawdowns in an observation well due to the extraction of a constant rate of water from a different well. The data collected in this way are then fitted with analytical solutions (e.g., Theis solution) that assume aquifer homogeneity so providing average values of the hydraulic parameters without considering any spatial distribution. One refers to these methods as traditional aquifer tests. If more than one observation point is available, the traditional analyses can highlight the presence of some heterogeneity.

In the last 15 years, to remove the homogeneity hypothesis and to investigate the spatial distribution of aquifer hydraulic properties, a technique called Hydraulic Tomography has been developed. It consists of sequential aquifer tests

in which the stress location is sequentially moved and the hydraulic responses are monitored in other locations. The data collected are then used to solve an inverse problem and to obtain information about the spatial variability of the aquifer hydraulic parameters.

In this work after a discussion of the traditional aquifer tests and an overview of the inverse methods applied to the hydraulic tomography, a Bayesian Geostatistical approach (conditioned on direct head data) is considered and tested with tomographic data in transient flow conditions and with both constant and non-stationary boundary conditions. Traditional analyses and the hydraulic tomography approach are then applied to a real case of the well field of the AIPO Boretto Research Site (Northern Italy) to test the methodologies on a field application.

To date, the application of the Bayesian Geostatistical approach to inverse problems (in particular on real problems) is limited by the lack of tools available for the scientific and technical community. For this reason the USGS (United States Geological Survey) is sponsoring a project to incorporate the Bayesian approach as a module of the industry standard software package PEST for the parameters estimation. In this work the kernel of the Bayesian PEST developed by the Writer is described in the last chapter. This module is doubtless a good way to spread the Bayesian Geostatistical inverse procedure to the modelers community.

Chapter 2

Traditional Aquifer Tests

2.1 Introduction

In this chapter characterization of aquifer hydraulic parameters (such as hydraulic conductivity or transmissivity and specific storage or storativity) using traditional aquifer tests is described. A brief overview of the hydraulic parameter definitions is reported before the introduction of the tests.

2.2 Hydraulic Parameter Definitions

2.2.1 Hydraulic Conductivity and Transmissivity

The hydraulic conductivity K $\left[\mathrm{LT}^{-1}\right]$ is the proportionality constant in Darcy's law that, for the flow through a porous medium column, can be written as:

$$\mathrm{v} = -\mathrm{Ki} \tag{2.1}$$

it states that the specific discharge v $\left[\mathrm{LT}^{-1}\right]$ in a porous medium is proportional to the hydraulic gradient i []. The hydraulic conductivity is the volume of fluid that will move through a unit cross-sectional area in a unit time under a unit hydraulic gradient; thus it is a measure of the material capacity to transmit

water. The hydraulic conductivity depends on both the properties of the porous medium and the fluid:

$$\mathrm{K} = k\frac{\gamma}{\mu} \tag{2.2}$$

where:

k is the intrinsic permeability of the porous media $\left[\mathrm{L}^2\right]$

γ is the specific weight of the fluid $\left[\mathrm{ML}^{-2}\mathrm{T}^{-2}\right]$

μ is the dynamic viscosity of the fluid $\left[\mathrm{ML}^{-1}\mathrm{T}^{-1}\right]$

Darcy's law as expressed in the Equation 2.1 considers the porous medium homogeneous and isotropic and K is a scalar; in the general case of anisotropic and heterogeneous medium the hydraulic conductivity is a symmetric tensor:

$$\overline{\overline{\mathrm{K}}} = \begin{bmatrix} \mathrm{K}_{xx} & \mathrm{K}_{xy} & \mathrm{K}_{xz} \\ & \mathrm{K}_{yy} & \mathrm{K}_{yz} \\ & & \mathrm{K}_{zz} \end{bmatrix} \tag{2.3}$$

Transmissivity T $\left[\mathrm{L}^2\mathrm{T}\right]$is the product of the hydraulic conductivity and the saturated thickness of the homogeneous aquifer; in general the integral of the hydraulic conductivity over the saturated thickness must be considered. It is a measurement of the ability of an aquifer to transmit groundwater throughout its entire saturated thickness. For confined aquifer the saturated thickness coincides with the total aquifer thickness. Usually transmissivity is considered when the groundwater flow is essentially horizontal, commonly when the lateral extensions of the aquifer are much greater than its thickness (Dupuit approximation). In the case of anisotropic aquifer the transmissivity is a symmetric tensor:

$$\overline{\overline{\mathrm{T}}} = \begin{bmatrix} \mathrm{T}_{xx} & \mathrm{T}_{xy} \\ & \mathrm{T}_{yy} \end{bmatrix} \tag{2.4}$$

2.2.2 Specific Storage and Storativity

The specific storage S_s $\left[L^{-1}\right]$ is the amount of water that a unit volume of aquifer releases from storage under a unit decline in head remaining in fully saturated conditions. In a confined aquifer the storage is dependent on both the compressibility of the aquifer material and the water.

Storativity S is the integral of the specific storage over the saturated thickness. S is dimensionless and more generally for a saturated confined aquifer is the volume of water released from storage per unit cross-sectional area of the aquifer and per unit decline in the component of the hydraulic head normal to the considered area.

2.3 Traditional Aquifer Tests

Pumping tests are the most common methods involved in the hydraulic characterization of aquifers. Usually a constant rate of water is extracted from a well and the changes in water level (drawdowns) are monitored in one or more observation wells and in the extraction well itself. The hydraulic parameters are then estimated by matching the collected data with analytical solutions developed under several approximations. Traditionally, the practically used methods are: the one developed by Thiem (1906) for steady state solution (see also Slichter (1899)) and the most important one developed by Theis (1935) under transient conditions. Although both the solutions are based on various simplified assumptions, they are still often used because of their simplicity. Omitting the steady state solutions, here the Theis analysis and assumptions are described; the methods mentioned below are all based on this approximate solution.

Theis (1935) was the first that had quantified the drawdown in a confined aquifer due to a pumping test as function of the extracted constant flow rate Q $\left[L^3T^{-1}\right]$, the distance r [L] between observation and extraction well and the time t [T]:

$$s(r,t) = \frac{Q}{4\pi \mathrm{T}} \int_{r^2\mathrm{S}/4\mathrm{T}t}^{\infty} \left(\frac{e^{-\tau}}{\tau}\right) \mathrm{d}\tau \tag{2.5}$$

T and S are the transmissivity and the storativity as above described respectively. The integral expression is known as exponential integral and it is usually indicated as well function $W(u)$. The explicit form of the well function is given by an infinite series:

$$W(u) = \int_{u}^{\infty} \left(\frac{e^{-\tau}}{\tau}\right) \mathrm{d}\tau = -0.5572 - \ln u + u - \frac{u^2}{2 \cdot 2!} + \frac{u^3}{3 \cdot 3!} - ... \tag{2.6}$$

where:

$$u = \frac{r^2 \mathrm{S}}{4\mathrm{T}t} \tag{2.7}$$

is dimensionless and the drawdown can be consequently expressed as:

$$s(r,t) = \frac{Q}{4\pi \mathrm{T}} W(u) \tag{2.8}$$

and $W(u)$ is dimensionless too.

The Theis solution 2.5 is valid under these assumptions: (1) the aquifer is homogeneous and isotropic; (2) the aquifer has infinite areal extent; (3) the well is fully penetrating; (4) the diameter of the well is infinitesimal; (5) T is constant in time and space; (6) water removed from storage is discharged instantaneously with decline in head; as already mentioned above, despite these restrictions, the Theis formula is successfully applied to many groundwater problems.

Effectively, applying Equations 2.7 and 2.8, S and T can be determined if s is measured for one value of r and several values of t, or for one value of t and several values of r, and if the discharge Q is known.

2.3.1 Drawdown-Time Analysis

Assuming that the drawdowns due to a pumping test are known for a given distance from the pumping well and at several times, transmissivity and storativity can be determined through a least square approach that minimize the sum of

square differences between observed and theoretical drawdowns:

$$f(t) = \sum_{i=1}^{m} \left[\bar{s}(r, t_i) - s(r, t_i)\right]^2 = min \tag{2.9}$$

where $\bar{s}(r, t_i)$ is the observed drawdown at distance r from pumping well and at time t_i; $s(r, t_i)$ is the theoretical drawdown at same location and time of the observed one, calculated by means of the Theis formula; m is the total number of drawdown data recorded in the observation interval during which the pumping rate remain constant.

2.3.2 Drawdown-Distance Analysis

Drawdown-Distance analyses allow to obtain independent estimates of transmissivity and storativity of an aquifer respect to the ones predicted with the previous described method and can be used to confirm Drawdown-Time results. Assuming that drawdowns due to a pumping test are known for a given time and at least two observation points, minimization of the sum of square differences between observed and theoretical drawdowns can be written as:

$$f(r) = \sum_{j=1}^{n} \left[\bar{s}(r_j, t) - s(r_j, t)\right]^2 = min \tag{2.10}$$

where $\bar{s}(r_j, t)$ is the observed drawdown at distance r_j from pumping well at time t; $s(r_j, t)$ is the theoretical drawdown at same distance r_j and time of the observed one, calculated by means of the Theis formula; n is the total number of observation point available during the pumping test at the same time t.

According to Wu et al. (2005), with this method, the not uniform spatially distribution of hydraulic parameters is considered and the correct effective hydraulic properties (for an equivalent homogeneous formation of the considered one) can be predicted. Possible variations in time of hydraulic parameters and any asymptotic behavior can also be highlighted with this method.

2.3.3 Drawdown-Time-Distance Analysis

Drawdown-Time-Distance analyses are able to take account of both the time and spatial variations of transmissivity and storativity. With this method, drawdowns due to a pumping test are assumed known at several distances from the pumping well (two or more observation points) and at several times; estimation of hydraulic parameters is performed minimizing the function:

$$f(r,t) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[\bar{s}(r_j, t_i) - s(r_j, t_i)\right]^2 = min \tag{2.11}$$

where $\bar{s}(r_j, t_i)$ is the observed drawdown at distance r_j from pumping well at time t_i; $s(r_j, t_i)$ is the theoretical drawdown at same distance r_j and time t_i of the observed one, calculated by means of the Theis formula; the other symbol are already defined.

It is wise to remind here that all the methods described above consider the aquifer as homogeneous; in this way average properties (of an equivalent homogeneous aquifer) are defined on a large volume that encloses the pumping and observation wells without providing any detailed spatial variability information of the hydraulic parameters (Butler and Liu, 1993; Yeh and Liu, 2000).

When several observations are available both in time and distance, different results coming from the procedures 2.3.1, 2.3.2, 2.3.3 highlight the existence of aquifer heterogeneities but, of course, no indication of the spatial distribution of such heterogeneities can be obtained from the equivalent homogeneous approach of the Theis procedure.

Chapter 3

Hydraulic Tomography

3.1 Introduction

Literally, tomography identifies a technique to obtain a plane section image of a solid object. The term is derived, in fact, from the Greek words "$\tau o \mu o \varsigma$" that means slice or section and "$\gamma \rho \alpha \varphi \iota \alpha$" that means to write (Merriam-Webster, 2009). Historically, the concept of tomography was developed in the area of medicine for imaging of human body through the use of X-rays (e.g. computer tomography, CT or CAT) (Sharma, 1997). Tomographic methods are also used in archeology, material science, oceanography, geophysics and other sciences. Geophysical methods, like gravity methods, magnetic methods, seismic methods, gamma-ray methods, electrical methods and so on, are, from several years and often, used to measure and image the physical properties of the subsurface (Hoover et al., 1996). In geosciences, observations are only possible at the surface or in boreholes; instead in medicine and some other sciences the source and the observation points can be positioned around the whole unknown body allowing to collect a set of 2-D slices combined to form a 3-D image (Gottlieb and Dietrich, 1995; Doser et al., 1998). This substantial difference limits the amount of information that can be collected with geophysical methods.

In hydrogeology, the physical quantities measured with geophysical methods

are indirectly related to the flow phenomena (Dietrich et al., 1998); for this reason general assumptions or empirical relationships are necessary to interpret geophysical data in term of hydraulic parameters with the effect of errors and further uncertainties in the estimation (Doser et al., 1998). Therefore, these *indirect* methods are often useful for qualitative characterization of hydraulic parameters in groundwater rather than quantitative. Alternative to the geophysical methods but still based on the tomographic concepts, the hydraulic tomography procedure, for the estimation of the spatial distribution of the hydraulic properties in the subsurface, has been quickly developing in the last years (Brauchler et al., 2003). The hydraulic tomography belongs to the *direct* methods because directly relates hydraulic parameters to quantities characterizing the groundwater flow, as water levels or drawdowns.

3.2 Overview

Hydraulic tomography is a recent technique for investigating the spatial distribution of hydraulic properties in subsurface. This method has potential to yield information on the spatial variations in groundwater hydraulic properties between wells at a level of detail that was not previously possible (Butler, 2005). Practically, during an hydraulic tomography test, water is extracted from or injected into a well and the drawdowns or more generally the changes in head are recorded at multiple wells at different locations (Zhu and Yeh, 2005). Afterward, sequentially the stress location is moved and the aquifer response is monitored in the other available observation points. In this way, different sets of independent data are collected at each location without the installation of additional wells. For a three-dimensional characterization of the aquifer heterogeneity, wells can also be divided into many vertical intervals using packers (Bohling et al., 2002). The aquifer stimulation takes place at each of these intervals and the hydraulic responses are monitored at other intervals in the same well and in the other locations (Yeh and Liu, 2000).

With N different stressing positions and assuming that each one is used ei-

ther for stimulation or observation, a total of N tests can be performed and $N(N-1)$ sets of independent head observations can be collected (Fienen, 2007). The availability of a large number of observed data is useful in the application of the inverse methodology used to incorporate the tomographic information. Consequently, the inverse problem, very often undetermined with a number of unknowns that greatly exceeds the number of measurements, with not unique solution and so ill-posed, is better constrained (Fienen et al., 2008). Moreover, the estimate will be more accurate and closer to reality than the traditional inverse approaches (Yeh and Lee, 2007).

In the last 15 years, several researchers have worked on inverse approaches for hydraulic tomographic data. Bohling (1993) is one of the first researchers that has introduced the term Hydraulic Tomography. The Author has used a trajectory-based approach to hydraulic tomography essentially founded on an iterative least square method; the result is a set of resistivities (the inverse of the hydraulic conductivities) of the field. Tosaka et al. (1993) for the purpose of identifying subsurface permeability distribution have developed a method named Hydropulse Tomography in which, as observation data, they have used hydraulic pressure responses collected at multiple points in a highly transient, multi-well interference testing. Gottlieb and Dietrich (1995) have investigated the possibility to identify the distribution of the permeability of water saturated soil by means of a series of pumping tests with different location of sinks and sources. The Authors have proposed a direct inverse approach of the head data based on a least squares method. Butler et al. (1999) have discussed new techniques for the measurement of drawdowns using small diameter tubings and their implications for the hydraulic tomography. An asymptotic solution to the flow equation has been applied by Vasco et al. (2000) to the inversion of pressure data collected by means of interference tests in boreholes. A travel time based hydraulic tomographic approach has been explored by Brauchler et al. (2003); the inversion is based on the relationship between the peak time of a recorded pressure curve and the diffusivity of the investigated system.

Geostatistical inversion methods, conditioned on direct head data, have been

applied both in steady state (Yeh and Liu, 2000; Fienen et al., 2008) and transient hydraulic tomography (Zhu and Yeh, 2005; Castagna and Bellin, 2009). In particular, Fienen et al. (2008) have highlighted the importance that discontinuities in hydraulic conductivity can have on the solution of tomographic inverse problems and have discussed how to subdivide the parameter field into zones with no correlation among hydraulic characteristics to obtain the best parameters estimation based on the available data. Castagna and Bellin (2009) have used a a pilot points approach to estimate the hydraulic parameters and a genetic algorithm to overcome the non-linearity between observation data and hydraulic parameters in the inverse problem. The non-linearity has been instead solved by Yeh and Liu (2000) and Zhu and Yeh (2005) using a linear estimator successively improved (Successive Linear Estimator (Yeh et al., 1996)) and have included the data sets collected in tomographic way sequentially; Fienen et al. (2008) have overcome the non-linearity by means of the Quasi-Linear method developed by Kitanidis (1995) and have considered all the data sets at the same time during the inversion.

In order to reduce the computational burden needed to analyze transient data, Bohling et al. (2002) have suggested the steady shape, unsteady flow analysis of tomographic pumping tests. Steady shape is a condition in which drawdown is continuing to change with time but the hydraulic gradient not. Transient data can be analyzed with the computational effort of a steady state model. With the same aim at reducing the computational effort in managing transient data, Li et al. (2005); Zhu and Yeh (2006) have explored the inversion of temporal moments of drawdown instead of drawdown itself. The Authors state that the first two temporal moments are sufficient to characterize the well hydrographs.

Sandbox experiments have been conducted by Liu et al. (2002); Yin and Illman (2009); Illman et al. (2007); Liu et al. (2007) in order to evaluate the performances and to highlight the power of some of the inverse approaches previously mentioned. The Authors have verified how different methods work under realistic conditions where experimental errors (in measurement of drawdowns or pumping rates or uncertainties associated with boundary conditions, etc.) are

present in the data used in the inversion process.

Pumping tests in tomographic way have been performed and analyzed in recent works on field applications by means of different inverse methodologies (Li et al., 2007; Straface et al., 2007; Bohling et al., 2007; Li et al., 2008; Bohling, 2009).

3.3 Methodology

In this section, the governing equations and a methodology for implementing hydraulic tomography are presented. A Quasi-Linear Geostatitistical, Bayes theorem based, approach represents the kernel of the inverse problem (Kitanidis, 1995); a constant but unknown mean value about which the estimate hydraulic conductivity field varies is also assumed. Pumping tests and observation data, collected in tomographic way, are considered during transient flow conditions; a constant value of the specific storage is also estimated on the entire domain. Both constant and changing boundary conditions are explored. The prior knowledge about the tests is limited to the flow pumping rates, the location of the wells and the observed drawdowns. The prior information about the parameters is instead restricted to the choice of a variogram (or a covariance model) with a single free parameter. Epistemic uncertainty is also considered by means of an epistemic error term estimated together with the variogram parameter. The forward model required in the groundwater inversion and that provide the relationship between hydraulic parameters and observations is MODFLOW_2005 (Harbaugh, 2005); an adjoint version of the same model MODFLOW_2005-Adjoint (Clemo, 2007) is used to compute the sensitivity (Jacobian) matrix. The optimization procedure named *"line search"* (Zanini and Kitanidis, 2008; Fienen, 2007), to stabilize and enforce the solution of the inverse problem during the linearization, is always adopted in this work.

3.3.1 Governing Equations

Three-Dimensional Groundwater Flow in Confined Aquifer. The governing flow equation for confined aquifers is developed from application of the continuity principle (i.e. law of mass conservation) to an elemental control volume (Bear and Verruijt, 1987). Applying the Darcy's Law and integrating the conservation of mass under constant density the equation become:

$$\nabla \cdot \left(\overline{\overline{\mathrm{K}}} \nabla h \right) = \mathrm{S}_s \frac{\partial h}{\partial t} \tag{3.1}$$

Equation 3.1 is the three-dimensional general diffusion equation for a heterogeneous and anisotropic material. Discharge or recharge (pumping well or injection well) to or from the control volume is represented as volumetric flux per unit volume Q $\left[\mathrm{T}^{-1}\right]$:

$$\nabla \cdot \left(\overline{\overline{\mathrm{K}}} \nabla h \right) = \mathrm{S}_s \frac{\partial h}{\partial t} \mp Q \tag{3.2}$$

where:

t is the time [T]

$h(x, y, z, t)$ is the piezometric head [L]

x, y, z are the spatial coordinates [L]; x and y in the horizontal plane and z along the vertical and positive upward

$\overline{\overline{\mathrm{K}}}$ is the saturated hydraulic conductivity tensor $\left[\mathrm{LT}^{-1}\right]$

S_s is the specific storage $\left[\mathrm{L}^{-1}\right]$

Initial and Boundary Conditions. Specification of initial and boundary conditions is required for the uniqueness of the solution of the differential partial equation 3.1 (De Smedt, 1998).

Initial conditions for unsteady state aquifer problems are generally specified when time is zero ($t = 0$) on the whole domain:

$$h\left(x,y,z,0\right)=h_0\left(x,y,z\right) \tag{3.3}$$

where h_0 represent a known function of x, y and z.

Conditions are required at every point of the boundary of the physical flow domain. There are several types of boundary conditions used in solving flow problems in aquifers but the most common ones are specified head boundaries (Dirichlet boundary condition):

$$h\left(x_b,y_b,z_b,t\right)=h_b\left(t\right) \tag{3.4}$$

where (x_b, y_b, z_b) is a point on the boundary and h_b is a known function of time; and specified flux boundaries (Neumann boundary condition):

$$q_n\left(x_b,y_b,z_b,t\right)=-\mathrm{K}\frac{\partial h}{\partial n} \tag{3.5}$$

where n represents the direction perpendicular to boundary and q_n $\left[\mathrm{L}^2\mathrm{T}^{-1}\right]$ is the flux component normal to the boundary itself, positive if enter the flow domain regardless of the sense of n. A mixture of the two previous type is applied when potential and normal flux component are related to each other.

3.3.2 Inverse Method: the Bayesian Approach

The Bayesian Geostatistical method allows to estimate a set of parameters that gives the best reproduction of observations and that is constrained using the prior information, characterized by geostatistical functions, on the structure of the parameters themselves. The first developments of the method go back to Kitanidis and Vomvoris (1983); Hoeksema and Kitanidis (1984) for applications in linear problems; a Quasi-Linear extension is then addressed by Kitanidis (1995).

3.3.2.1 Bayes theorem

Bayes theorem, in terms of random variables and their probability function, states:

$$p(\mathbf{s} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{s})\, p(\mathbf{s})}{p(\mathbf{y})} \tag{3.6}$$

where $\mathbf{s}$ and $\mathbf{y}$ are the state (uncertain quantities) and data (measured quantities) variables vectors.

In the Equation 3.6, $p(\mathbf{s}|\mathbf{y})$ is the posterior probability density function evaluated as the product of the likelihood function $p(\mathbf{y} \mid \mathbf{s})$ and the prior probability distribution function $p(\mathbf{s})$ normalized with respect to the total probability $p(\mathbf{y})$.

The prior probability distribution represent knowledge about the unknown quantities a priori, that is, before any observed data have been considered; however, in interpolation and inverse problems it is reasonable to infer the structure of $\mathbf{s}$, represented by the prior, from the data. One refers in this case to *"empirical Bayes"* methods.

The likelihood function indicates how likely a particular $\mathbf{s}$ is to produce an observed sample; the total probability (prior of the data), instead, equal to:

$$p(\mathbf{y}) = \int p(\mathbf{y} \mid \mathbf{s})\, p(\mathbf{s}) \tag{3.7}$$

is, except in special case, difficult or impossible to calculate if not numerically; but for all practical purposes it is just a normalization constant chosen as to make the integral of the posterior probability density function equal to 1.

3.3.2.2 Linear Model

A simple but very important case in inverse estimation processes is the one that involve a problem with linear relations between unknowns and data. Some problems although non linear can be solved by successively linearizing the above mentioned relations using an iterative procedure (Quasi-Linear methods). A simplest and popular approach is to adopt a Gaussian prior distribution and a Gaussian likelihood function so that the posterior probability density function is also Gaussian.

Let's start considering the following relation (measurement equation):

$$\mathbf{y}=\mathbf{h}(\mathbf{s})+\mathbf{r} \tag{3.8}$$

The Equation 3.8 relates the data (observations) vector $\mathbf{y}$ $[n_{obs} \times 1]$ to the vector of the unknowns (parameters) $\mathbf{s}$ $[n_{par} \times 1]$; $\mathbf{h}(\mathbf{s})$ $[n_{obs} \times 1]$ represents the function (forward model) that, for a given $\mathbf{s}$, provides the modeled values at the same locations and times of the observed data. Errors in the conceptual model, errors due to the numerical solution of the model and, mainly, errors when measuring data, are considered by means of $\mathbf{r}$ $[n_{obs} \times 1]$, the epistemic error vector. Epistemic uncertainties are supposed to be a random process with zero mean and covariance matrix $\mathbf{R}$ $[n_{obs} \times n_{obs}]$; a priori $\mathbf{s}$ and $\mathbf{r}$ are uncorrelated.

In case the relation between parameters and observations is linear, $\mathbf{h}(\mathbf{s})$ can be substituted with $\mathbf{Hs}$ where the matrix $\mathbf{H}$ $[n_{obs} \times n_{par}]$ is, in this case, independent from $\mathbf{s}$. The measurement equation can be rewritten as:

$$\mathbf{y} = \mathbf{Hs} + \mathbf{r} \tag{3.9}$$

The vector $\mathbf{s}$ of the unknowns is a priori assumed with a random multi-Gaussian distribution with mean:

$$E[\mathbf{s}] = \mathbf{X}\boldsymbol{\beta} \tag{3.10}$$

and covariance:

$$E\left[(\mathbf{s} - \mathbf{X}\boldsymbol{\beta})(\mathbf{s} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}\right] = \mathbf{Q}_{\mathrm{ss}} \tag{3.11}$$

where the symbol E means the expected value, $\mathbf{X}$ $[n_{par} \times p]$ is a known matrix (of base functions), $\boldsymbol{\beta}$ $[p \times 1]$ is a vector of p drift coefficients and $\mathbf{Q}_{\mathrm{ss}}$ $[n_{par} \times n_{par}]$ is the covariance matrix. The matrix $\mathbf{X}$ associates each value of the $\mathbf{s}$ vector with the corresponding mean value selected from the $\boldsymbol{\beta}$ vector (in this way it can be taken account of zones with different mean or parameters of different type); a drift, that is, a trend information a priori known about $\mathbf{s}$ can be also expressed

by the same matrix.

The probability density function of $\mathbf{s}$ for a given $\boldsymbol{\beta}$ assumes, accordingly, the form:

$$p(\mathbf{s}|\boldsymbol{\beta}) = \frac{1}{\sqrt{(2\pi)^{n_{par}} \det(\mathbf{Q}_{ss})}} \exp\left[-\frac{1}{2}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mathbf{Q}_{ss}^{-1}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})\right] \tag{3.12}$$

Assuming $\boldsymbol{\beta}$ a priori unknown, with its prior probability $p(\boldsymbol{\beta}) \propto \mathbf{1}$ (uniform over all space), $\mathbf{s}$ and $\boldsymbol{\beta}$ are estimated together and it is possible to write:

$$p(\mathbf{s},\boldsymbol{\beta}) = \frac{1}{\sqrt{(2\pi)^{n_{par}} \det(\mathbf{Q}_{ss})}} \exp\left[-\frac{1}{2}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mathbf{Q}_{ss}^{-1}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})\right] \tag{3.13}$$

The likelihood function of the errors, also assumed multi-Gaussian, can be written as:

$$p(\mathbf{y}|\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^{n_{obs}} \det(\mathbf{R})}} \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{s})^{\mathrm{T}} \mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{s})\right] \tag{3.14}$$

applying the Bayes theorem, removing $p(\mathbf{y})$ together with the other constants, the posterior density probability function becomes:

$$p(\mathbf{s}|\mathbf{y}) \propto \exp\left[-\frac{1}{2}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mathbf{Q}_{ss}^{-1}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})\right] \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{s})^{\mathrm{T}} \mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{s})\right] \tag{3.15}$$

The posterior is multi-Gaussian too and after a simple manipulation becomes:

$$p(\mathbf{s}|\mathbf{y}) \propto \exp\left[-\frac{1}{2}\left((\mathbf{s}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mathbf{Q}_{ss}^{-1}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta}) - (\mathbf{y}-\mathbf{H}\mathbf{s})^{\mathrm{T}} \mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{s})\right)\right] \tag{3.16}$$

The posterior values of $\mathbf{s}$ and $\boldsymbol{\beta}$ are the ones that maximizes the 3.16; it is convenient and equivalent, instead of maximize the posterior probability, to

minimize its negative logarithm:

$$\mathcal{L} = -\ln p(\mathbf{s}|\mathbf{y}) \tag{3.17}$$

definitely, the objective function to minimize assumes the form:

$$\mathcal{L} \propto (\mathbf{s} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}} \mathbf{Q}_{\mathrm{ss}}^{-1} (\mathbf{s} - \mathbf{X}\boldsymbol{\beta}) + (\mathbf{y} - \mathbf{H}\mathbf{s})^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{y} - \mathbf{H}\mathbf{s}) \tag{3.18}$$

The posterior mean values $\hat{\mathbf{s}}$ and $\hat{\boldsymbol{\beta}}$, that minimize $\mathcal{L}$, can be obtained setting to zero the derivatives $\frac{\partial \mathcal{L}}{\partial \mathbf{s}}$ and $\frac{\partial \mathcal{L}}{\partial \beta}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{s}} = \left(\hat{\mathbf{s}} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{Q}_{\mathrm{ss}}^{-1} - (\mathbf{y} - \mathbf{H}\hat{\mathbf{s}})^{\mathrm{T}} \mathbf{R}^{-1}\mathbf{H} = 0 \tag{3.19}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = -\left(\hat{\mathbf{s}} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{Q}_{\mathrm{ss}}^{-1}\mathbf{X} = 0 \tag{3.20}$$

The Equation 3.19 can be manipulated to obtain the best estimate:

$$\hat{\mathbf{s}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \left(\mathbf{Q}_{\mathrm{ss}}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1} \mathbf{H}^{T}\mathbf{R}^{-1} \left(\mathbf{y} - \mathbf{H}\mathbf{X}\hat{\boldsymbol{\beta}}\right) \tag{3.21}$$

or the equivalent form:

$$\hat{\mathbf{s}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}} \left(\mathbf{H}\mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right)^{-1} \left(\mathbf{y} - \mathbf{H}\mathbf{X}\hat{\boldsymbol{\beta}}\right) \tag{3.22}$$

Introducing the vector $\boldsymbol{\xi}$ $[n_{par} \times 1]$ such that:

$$\mathbf{y} - \mathbf{H}\mathbf{X}\hat{\boldsymbol{\beta}} = \left(\mathbf{H}\mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right)\boldsymbol{\xi} \tag{3.23}$$

the Equation 3.22 can be expressed as:

$$\hat{\mathbf{s}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}}\boldsymbol{\xi} \tag{3.24}$$

The best estimate $\hat{\mathbf{s}}$ is a superposition of the calculated mean and fluctuations about the mean itself. Substituting the 3.24 into 3.20 and after simple manipulations it is possible to write:

$$(\mathbf{HX})^{\mathrm{T}}\boldsymbol{\xi} = 0 \tag{3.25}$$

Combining Equations 3.23 and 3.25 the solution ($\boldsymbol{\xi}$ and $\hat{\boldsymbol{\beta}}$) is obtained by solving the $[(n_{par}+p)\times(n_{par}+p)]$ system of linear equations:

$$\begin{bmatrix} \mathbf{H}\mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}}+\mathbf{R} & \mathbf{HX} \\ \mathbf{X}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi} \\ \hat{\boldsymbol{\beta}} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \tag{3.26}$$

The system 3.26 is also known as the ordinary cokriging system usually derived by finding a Best Linear Unbiased Estimation (BLUE methods).

Known $\boldsymbol{\xi}$ the best estimate $\hat{\mathbf{s}}$ is obtained by means of the 3.24.

3.3.2.3 Quasi-Linear Geostatistical Approach

So far, the function $\mathbf{h}(\mathbf{s})$ in the observation Equation 3.8 is considered linear. For weakly nonlinear problems, $\mathbf{h}(\mathbf{s})$ can be successively linearized about a candidate solution $\mathbf{s}_k$ following the Quasi-Linear geostatistical approach (Kitanidis, 1995).

At each iteration k in the linearization process, the function $\mathbf{h}(\mathbf{s})$ becomes:

$$\mathbf{h}(\mathbf{s}) \approx \mathbf{h}(\mathbf{s}_k) + \tilde{\mathbf{H}}(\mathbf{s}-\mathbf{s}_k) \tag{3.27}$$

where, now, the sensitivity matrix $\tilde{\mathbf{H}}$ $[n_{obs}\times n_{par}]$ is a function of $\mathbf{s}$ and must be evaluated at each linearization as $\tilde{\mathbf{H}}_k = \left.\frac{\partial \mathbf{h}(\mathbf{s})}{\partial \mathbf{s}}\right|_{\mathbf{s}_k}$.

The observations vector can be corrected as:

$$\tilde{\mathbf{y}}_k = \mathbf{y} - \mathbf{h}(\mathbf{s}_k) + \tilde{\mathbf{H}}_k\mathbf{s}_k \tag{3.28}$$

and the linearized objective function to minimize can be expressed as:

$$\mathcal{L}' \propto (\mathbf{s}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}\mathbf{Q}_{\mathrm{ss}}^{-1}(\mathbf{s}-\mathbf{X}\boldsymbol{\beta}) + \left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{H}}_k\mathbf{s}\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{H}}_k\mathbf{s}\right) \tag{3.29}$$

The Equation 3.29 is formally identical to the Equation 3.18 so that the

solution can be achieved solving in the same way of the 3.26 the linear system:

$$\begin{bmatrix} \tilde{\mathbf{H}}_k \mathbf{Q}_{\mathrm{ss}} \tilde{\mathbf{H}}_k^{\mathrm{T}} + \mathbf{R} & \tilde{\mathbf{H}}_k \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} \tilde{\mathbf{H}}_k^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_{k+1} \\ \boldsymbol{\beta}_{k+1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{y}}_k \\ \mathbf{0} \end{bmatrix} \tag{3.30}$$

The new estimation of $\mathbf{s}$ is then similarly to the 3.24:

$$\mathbf{s}_{k+1} = \mathbf{X}\boldsymbol{\beta}_{k+1} + \mathbf{Q}_{\mathrm{ss}} \tilde{\mathbf{H}}_k^{\mathrm{T}} \boldsymbol{\xi}_{\mathrm{k+1}} \tag{3.31}$$

The procedure is iteratively repeated increasing k by one until convergence is achieved. In this work the iterations are stopped when the improvement in the objective function are negligible.

3.3.2.4 Covariance Model and Structure Selection

According to Fienen et al. (2008), in this work an approximation, a limiting case, of the exponential covariance model is adopted:

$$R(d) = \sigma^2 \exp\left(-\frac{d}{l}\right) \tag{3.32}$$

where σ^2 is the variance, d is the separation distance between nodes and l is the integral scale. Assuming that $l \to \infty$ and $d > 0$, the covariance model becomes:

$$R(d) = \sigma^2 \left(1 - \frac{d}{l}\right) = \theta l - \theta d \tag{3.33}$$

where $\theta = \frac{\sigma^2}{l}$. In this case the variogram model assume the form:

$$\gamma(d) = R(0) - R(d) = \theta d \tag{3.34}$$

that is a linear model.

Assuming l to be constant and sufficiently large (10 times the $\max(d)$) and using the relation $\sigma^2 = \theta l$ the prior covariance model 3.32 can be rewritten as:

$$R(d) = \theta l \exp\left(-\frac{d}{l}\right) \tag{3.35}$$

The above covariance model is a valid covariance and have just a parameter θ that must be estimate. According to this model each term of the covariance matrix in the prior density function will be:

$$\mathbf{Q}_{\mathrm{ss_{i,j}}}(\theta) = \theta l \exp\left(-\frac{d_{i,j}}{l}\right) \tag{3.36}$$

Moreover, in this work, the epistemic errors in the Equation 3.8 are assumed to be independent and identically distributed (uncorrelated) with variance σ_R^2; in this case the covariance matrix $\mathbf{R}$ in the Equation 3.14 assumes the form:

$$\mathbf{R} = \sigma_R^2 \mathbf{I} \tag{3.37}$$

where $\mathbf{I}$ $[n_{obs} \times n_{obs}]$ is the identity matrix.

In the above derivation of the geostatistical approach the structural parameters are considered known but the choice of the right parameters is crucial to reach a good solution of the problem. At this point, two structural parameters must be estimated: the prior covariance model parameter θ (Equation 3.36) and the epistemic error variance σ_R^2 (Equation 3.37).

The estimation of parameters can be based on the examination of residuals, that is, the differences between observed and predicted values; in this work both the parameters are estimated analyzing the orthonormal residuals as proposed by Kitanidis (1991, 1997). The orthonormal residuals vector $\boldsymbol{\epsilon}$ $[n_{obs} - p \times 1]$ is the $\boldsymbol{\delta}$ $[n_{obs} - p \times 1]$ residuals vector normalized by the vector of the standard errors $\boldsymbol{\sigma}$ $[n_{obs} - p \times 1]$. The $i-$th residual value δ_i is the difference between the $i-$th observed value and the $i-$th estimated value using only the first $i-1$ measurements.

According to Kitanidis (1997), when the correct parameters are used the ϵ residuals are independent identically distributed with Gaussian distribution, zero mean and variance 1. For this reason if the parameters are not known, the correct selection of the parameters is such that:

$$Q_2 = \frac{1}{n_{obs} - p} \sum_{i=p+1}^{n_{obs}} \epsilon_i^2 = 1 \tag{3.38}$$

Since two parameter must be estimated, there may be more than one set of parameters that satisfies the Equation 3.38. Another constraint is needed; it is reasonable to select the values of the parameters that result in small estimation error, i.e. the residuals $\boldsymbol{\delta}$ are as small as possible. A good agreement between the model and the data is described by:

$$cR = Q_2 \exp\left(\frac{1}{n_{obs} - p} \sum_{i=p+1}^{n_{obs}} \ln\left(\sigma_i^2\right)\right) \tag{3.39}$$

To summarize, the correct parameters are the two that minimize the Equation 3.39 with the constraint 3.38.

Effectively, once a reasonable value for σ_R^2 is set, it is possible to determine the ratio $\frac{\theta}{\sigma_R^2}$ that minimize cR (Equation 3.39) and then adjust the parameters by multiplying them by the value Q_2 (Kitanidis, 1991).

3.3.3 Forward Model

The forward problem that provides the relation between the hydraulic parameters and the observations needed in the inverse procedure is solved by means of MODFLOW_2005 (Harbaugh, 2005). MODFLOW_2005 is the most recent version of the finite-difference groundwater model MODFLOW (McDonald and Harbaugh, 1988). The partial differential Equation 3.2 of the three-dimensional groundwater flow is simulated using a block-centered finite-difference approach. The program is divided into packages that allow the user to simulate specific hydrologic features of the model independently.

The model requires a set of instructions and data files to perform the simulation; files that not depend on the parameters that must be estimated are edited once at the beginning; files that contain parameters that change during the inversion process are automatically updated before the new forward run. The output

data needed to the inverse model are automatically read after each forward run.

Computational inefficiencies in MODFLOW_2005 have been encountered using lumped parameters for the definition of the variables in the LPF package (Layer Property Flow); these are discussed in the Appendix A.

3.3.4 Sensitivity Matrix: the Adjoint State Method

In nonlinear models, like the above described, the calculation of the sensitivity (Jacobian) matrix is required. It represents the rate of changes in model results to the changes in model parameters $\tilde{\mathbf{H}} = \frac{\partial \mathbf{h(s)}}{\partial \mathbf{s}}$. This is often the computationally most costly step. The simplest way to calculate the Jacobian is the perturbation method based on the direct differentiation: the forward problem is repeated with a small variation of each parameter in each run. The sensitivity matrix is the ratio between the variations of all measured quantities and the variation of each parameter. In this way at least $n_{par} + 1$ runs are required and often the number of the parameters highly exceeds the number of observations n_{obs}.

The adjoint state method can be used to efficiently compute the Jacobian in undetermined problems (e.g. Sykes et al., 1985; Townley and Wilson, 1985; Sun, 1994). In this way to calculate all the elements of the sensitivity matrix it is needed to solve the forward problem once and the adjoint problem n_{obs} times; in total $n_{obs} + 1$ runs are required.

The process is essentially composed of three steps (Clemo, 2007):

1. solution of the flow problem;
2. calculation of the adjoint state for each observation using the flow solution;
3. determination of the sensitivity of each observation to the parameters by summing the product of the adjoint state with the derivative of the groundwater flow equation with respect to the parameters for each time step of the flow equation.

In this work the MODFLOW_2005-Adjoint code is used to calculate the sensitivity of each observation to each parameter to create a Jacobian matrix; the

code supports different type of parameters: horizontal hydraulic conductivity *HK*, vertical hydraulic conductivity *VK*, specific storage *SS*, specific yield *SY* and many others (for a complete list see Clemo (2007)).

Also in MODFLOW_2005-Adjoint inefficiencies have been encountered using lumped parameters; these are discussed in the Appendix A together with those encountered in MODFLOW_2005.

3.3.5 Inversion Code

In this work, the geostatistical inversion is performed by means of an improved version of the MATLAB (Mathworks, 2002) code originally developed by Fienen (2007). With the original code, based on steady-state analysis of pumping tests in tomographic way, only the hydraulic conductivity field is estimate. The new version includes, in addition, the estimation of the specific storage. Moreover the Fienen code is developed to use the MODFLOW_2000 (Harbaugh et al., 2000) and an adjoint version of the same code as forward model and sensitivity calculation model respectively. The version of the inverse code used here, instead, is based on the new 2005 versions of both the MODFLOW codes.

The routines, of the original code, that write the MODFLOW_2005 and MODFLOW_2005-Adjoint files that depend on the parameters and must be successively update, have also been changed. In fact, in the original inversion code, the use of lumped parameters is assumed for the definition of the variable in the Layer Property File package. Due to inefficiencies in MODFLOW_2005 and MODFLOW_2005-Adjoint using lumped parameters (see Appendix A for more details), the current inversion code assumes distributed parameters defined through multiplication matrices (see Harbaugh (2005, chapter 8) for more information about the use of parameters in MODFLOW family codes).

Finally, the selection of the appropriate structural parameters, the optimal ratio of $\frac{\theta}{\sigma_R^2}$ that minimize cR, is in the improved inverse code version an automatic process instead of a manually operation by trial and error with a graphical selection of the optimum (Fienen, 2007). An unconstrained nonlinear optimization

algorithm (*fminsearch*) available in the MATLAB (Mathworks, 2002) toolbox is used to perform this selection. See Appendix E for more details about *fminsearch.*

3.3.5.1 Optimization Procedure: *"line search"*

For problems with substantial non-linearity, the linearization procedure of the Quasi-Linear approach could give oscillation of the solution at each iteration k. The estimation of the parameters at the iteration $k+1$, $\mathbf{s}_{k+1}$, could be too much different from the previous one $\mathbf{s}_k$. An optimization procedure can be adopted to drive the solution at each iteration. This procedure, *"line search"* (Zanini and Kitanidis, 2008; Fienen, 2007), consists of considering a linear combination of the new estimate $\mathbf{s}_{k+1}$ and the previous one $\mathbf{s}_k$:

$$\tilde{\mathbf{s}}_{k+1} = \alpha \mathbf{s}_k + (1-\alpha)\,\mathbf{s}_{k+1} \tag{3.40}$$

where α is chosen so that the objective function:

$$\mathcal{L}'' \propto (\tilde{\mathbf{s}}_{k+1} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}\, \mathbf{Q}_{\mathrm{ss}}^{-1}\, (\tilde{\mathbf{s}}_{\mathbf{k+1}} - \mathbf{X}\boldsymbol{\beta}) + \left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{H}}_k \mathbf{s}\right)^{\mathrm{T}} \mathbf{R}^{-1} \left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{H}}_k \mathbf{s}\right) \tag{3.41}$$

has a minimum.

In this way the objective function 3.41 has a monotonic decrease at each linearization iteration.

An automatic procedure is used to perform the line searching: the unconstrained nonlinear optimization algorithm (*fminsearch*) available in the MATLAB (Mathworks, 2002) toolbox (see Appendix E). It is important to highlight here that it is not necessary during the line search procedure to reach the real minimum of the objective function, but the idea is just to drive the function to reach the minimum value. For this reason, the maximum number of iterations in the optimization procedure is set to a small value (from 3 to 5) so that the computational time is limited.

3.3.5.2 Non-negativity of hydraulic parameters

The hydraulic conductivities and the specific storage cannot be negative. To enforce non-negativity of these parameters it is possible to work in a logarithm estimation space $\mathbf{s}$. Then, after the estimation, with a back-transformation the parameters come back to the physical space $\mathbf{z} = \exp(\mathbf{s})$. This transformation makes the problem non-linear even if the governing equation are linear, but ensure a physically correct solution. The sensitivity matrix, that is calculated in the physical space (by means of the MODFLOW_2005_Adjoint software) need also to be converted in the estimation space during the inversion procedure. Indicating with $\tilde{\mathbf{H}}$ the sensitivity matrix in the estimation space and with $\hat{\mathbf{H}}$ the sensitivity in the physical space, using the chain rule it is possible to write:

$$\tilde{\mathbf{H}} = \frac{\partial \mathbf{h}(\mathbf{s})}{\partial \mathbf{s}} = \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{s}} = \hat{\mathbf{H}} \frac{\partial \exp(\mathbf{s})}{\partial \mathbf{s}} = \hat{\mathbf{H}} \exp(\mathbf{s}) \tag{3.42}$$

3.3.5.3 Inversion Steps

Assuming that n_{obs} observations due to N pumping tests performed in tomographic way are available and a forward model able to reproduce each test and to provide predictions at the same positions and times of the n_{obs} is set up, the inversion procedure is based on the following steps:

1. **Initialization of the variables**. The parameter vector $\mathbf{s}$ is initialized to $\mathbf{s}_0$ based on prior information. The structural parameters θ and σ_R^2 are initialized so that the ratio $\frac{\theta}{\sigma_R^2}$ is less than 1; in this way at the beginning a smooth solution is achieved that is more stable.

2. **Running of the forward model and the adjoint model**. Using the current parameters estimation vector $\mathbf{s}_k$ (at the beginning $\mathbf{s}_k = \mathbf{s}_0$), the forward model and the adjoint model run N times and provide the vector of predictions $\mathbf{h}(\mathbf{s}_k)$ and the sensitivity matrix $\tilde{\mathbf{H}}_k = \frac{\partial \mathbf{h}(\mathbf{s})}{\partial \mathbf{s}} \mid_{\mathbf{s}_k}$.

3. **Calculation of the best estimate**. By means of $\mathbf{h}(\mathbf{s}_k)$ and $\tilde{\mathbf{H}}_k$ and the Equations 3.26 and 3.31 a new candidate for the parameter vector $\mathbf{s}_{k+1}$ is calculated. The *"line search"* procedure is performed to drive the objective function 3.41 to reach the minimum value (during the line search the forward model runs again). With the new estimation of $\mathbf{s}$ ($\tilde{\mathbf{s}}_{k+1}$ of the Equation 3.40) the step 2 is repeated until the improvement in the objective function is below a threshold value.

4. **Estimation of the structural parameters**. At this point, the solution has converged for the fixed structural parameters. The orthonormal residuals are calculated and the metrics Q_2 and cR are evaluated. The current structural parameters are both multiplied by Q_2 and the steps 2 and 3 are repeated so that when the solution has converged Q_2 results equal to 1. The automatic optimization algorithm (*fminsearch* (Appendix E)) chooses the new ratio $\frac{\theta}{\sigma_R^2}$ of the structural parameters and the steps 2, 3 and 4 are repeated until the minimum of cR is reached. The last evaluated parameter vector is the best estimation of $\mathbf{s}$ achievable with the available observations.

3.4 Synthetic cases

Following the above described methodology, several illustrative examples of synthetic cases have been performed to show the capability of the inversion procedure. Pumping tests and observation data, collected in tomographic way, are considered during transient flow conditions; the hydraulic conductivity field and a single value of the specific storage are estimated on the entire domain. Both constant and non-stationary boundary conditions are explored. The prior knowledge about the tests is limited to the flow pumping rates, the location of the wells and the observed drawdowns. No correlation is assumed between the hydraulic conductivity field and the single value of the specific storage and the variance of this last parameter is set to a small value[1] equal to 10^{-4}.

[1] Actually the variance must be zero, but a zero value renders the covariance matrix singular and the inverse problem ill-conditioned

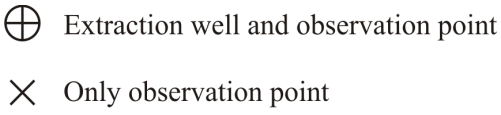

Figure 3.1: Symbols used to identify well and observation locations.

The computational domain is 31×31 meters discretized in 961 square elements 1 m on each side; a single confined layer is considered. The four sides of the aquifer model are specified head boundaries; in case of constant boundary conditions the same value of head is specified on the entire boundary from the beginning to the end of the run, in case of non-stationary boundary conditions the specified heads stay on an inclined plane that shifts its position and changes its slope during the run time. The pumping rate values are variable from $0.001\,\mathrm{m^3 s^{-1}}$ to $0.020\,\mathrm{m^3 s^{-1}}$. A total run of 1200 seconds is performed for each case and four observations in time are considered for each observation point: after about 40 sec, 315 sec, 695 sec and 1200 sec from the beginning of the pumping tests. The times of the observations are chosen so that the drawdown curves are well described; the use of more observations in time is computationally inefficient due to the calculation of the sensitivity and, as showed by Zhu and Yeh (2005), use of many heads at a given observation location at different times provides overlapping information. The number of wells and observation points is variable for each case; Figure 3.1 shows the symbols used to identify well and observation locations. The observation head values for each synthetic case, used in the estimation process, are obtained with an *a priori* forward run using the true hydraulic conductivity field and the true specific storage by means of MODFLOW_2005.

For each case the mean value $\overline{\mathrm{K}}$ and standard deviation value σ_{K} of the hydraulic conductivity field and the specific storage value S_s are listed for the true field and the estimated one; also the value of θ and σ_R^2 are showed for each test together with the MSE (Mean Square Error)[2] of the hydraulic conductivity field.

[2]The Mean Square Error MSE is one way to quantify the difference between an estimator x_e and the true value x_t of the quantity being estimated: MSE$=\frac{1}{n}\sum_{i=1}^{n}(x_t - x_e)^2$ where n is the size of the sample.

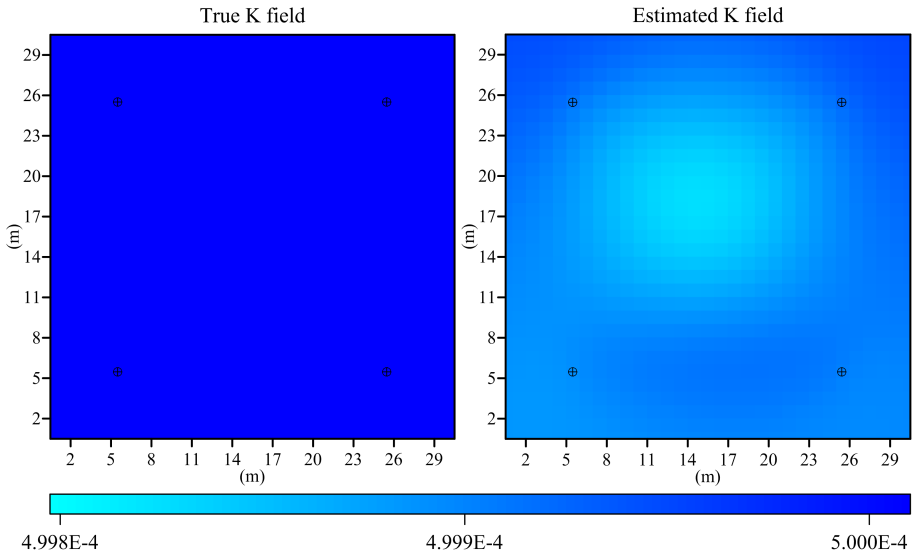

Figure 3.2: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an homogeneous aquifer and constant boundary conditions.

In Figures 3.2 and 3.3 cases of homogeneous hydraulic conductivity fields with constant boundary conditions and non-stationary boundary conditions are reported respectively. The aim of these synthetic cases is to test the ability of the methodology to reproduce the correct hydraulic conductivity field and the specific storage value in cases of no variations around the mean values. The small value estimated for the structural parameter θ imply that the estimated hydraulic conductivity field is almost homogeneous; the estimated mean values are satisfactory and the variations around the estimated mean value are very small. The standard deviation and the MSE of the estimated hydraulic conductivity field in case of non-stationary boundary conditions is of the same order of magnitude of the constant boundary conditions case and proves that, even if the observations are affect by the changing boundary conditions, the estimation is correct.

In Figures 3.4-3.9 cases of random generated hydraulic conductivity fields with variations on different orders of magnitude and with constant boundary conditions and non-stationary boundary conditions are shown. In all the cases there is a good reproduction of the hydraulic conductivity field and also the

Table 3.1: True and estimated characteristics of the homogeneous aquifer of Figure 3.2.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.999 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	0	$2.736 \cdot 10^{-8}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.999 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$8.300 \cdot 10^{-11}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$7.545 \cdot 10^{-14}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$1.059 \cdot 10^{-14}$		

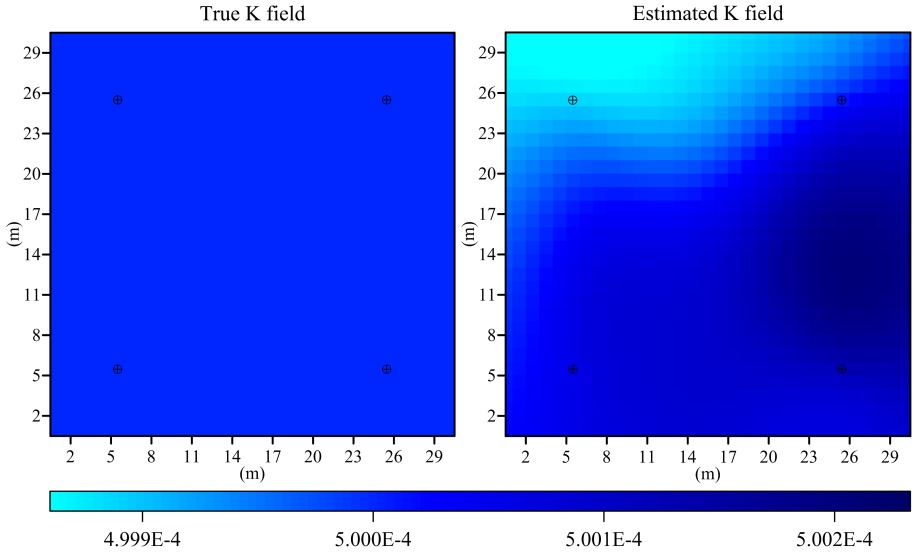

Figure 3.3: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an homogeneous aquifer and non-stationary boundary conditions.

Table 3.2: True and estimated characteristics of the homogeneous aquifer of Figure 3.3.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.999 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	0	$9.629 \cdot 10^{-8}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$5.001 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$2.010 \cdot 10^{-9}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$2.000 \cdot 10^{-12}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$1.433 \cdot 10^{-14}$		

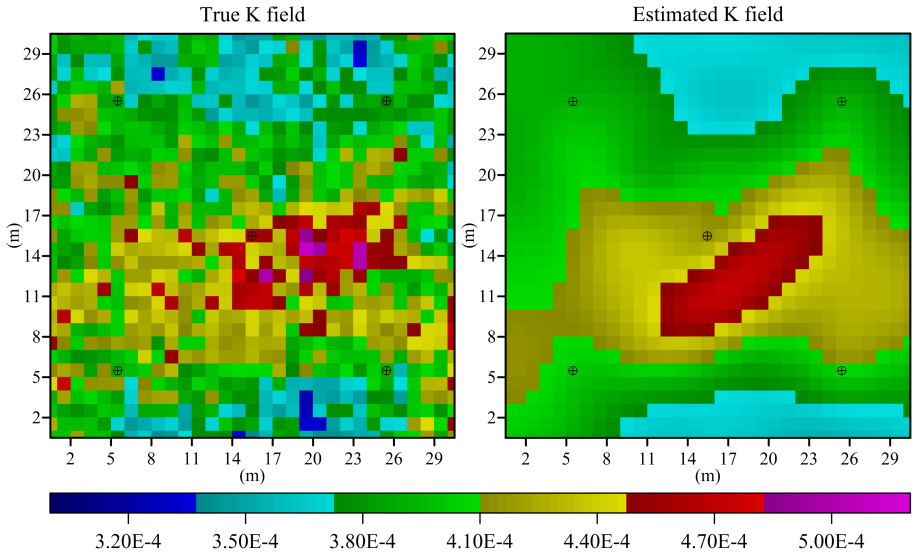

Figure 3.4: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for a random generated formation and constant boundary conditions.

mean value and the standard deviation of the hydraulic conductivity field are correctly estimated as well as the specific storage values. In all the cases the use of non-stationary boundary conditions gives better results in comparison to the correspondent case with constant boundary conditions (comparing the MSE). The fact of considering non-stationary boundary conditions implies a changing in time flow and hydraulic heads in the aquifer that, in the opinion of the Writer, gives additional information about the heterogeneity in the observation points independently of the pumping tests.

Figure 3.10 shows the true and the estimated hydraulic conductivity field for a centered homogeneous inclusion in an homogeneous field with one order of magnitude of variation and with non-stationary boundary conditions. In this case four well/observation locations, outside of the inclusion, are used. The mean and the standard deviation values of the hydraulic conductivity field and the specific storage value are estimated with the correct order of magnitude but there is a trend to underestimate the values. Considering one more well/observation point located into the inclusion gives better results as showed in Figure 3.11 and

Table 3.3: True and estimated characteristics of the random generated formation of Figure 3.4.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$4.012 \cdot 10^{-4}$	$4.007 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$3.118 \cdot 10^{-5}$	$2.604 \cdot 10^{-5}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$5.012 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$2.400 \cdot 10^{-3}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$3.770 \cdot 10^{-12}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$4.194 \cdot 10^{-10}$		

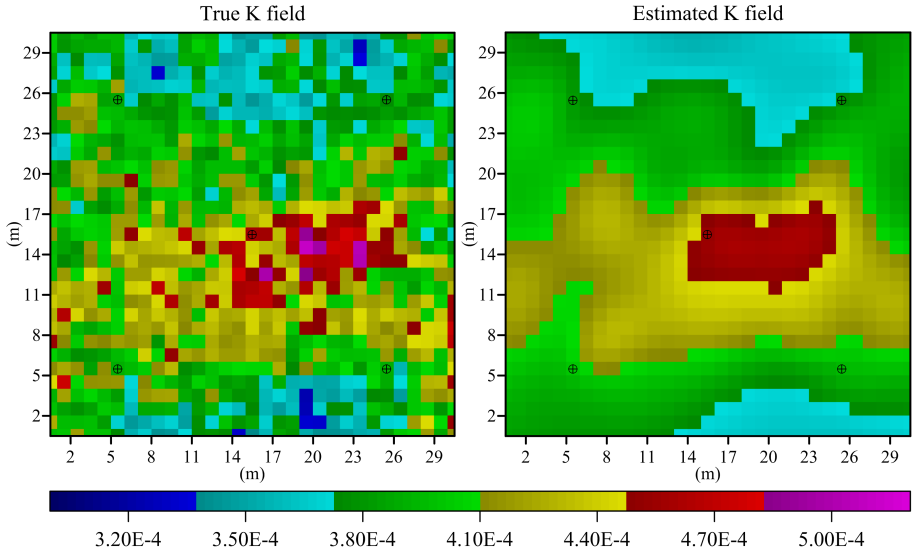

Figure 3.5: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for a random generated formation and non-stationary boundary conditions.

Table 3.4: True and estimated characteristics of the random generated formation of Figure 3.5.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$4.012 \cdot 10^{-4}$	$3.996 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$3.118 \cdot 10^{-5}$	$2.556 \cdot 10^{-5}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$5.002 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$4.766 \cdot 10^{-3}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$2.899 \cdot 10^{-13}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$3.925 \cdot 10^{-10}$		

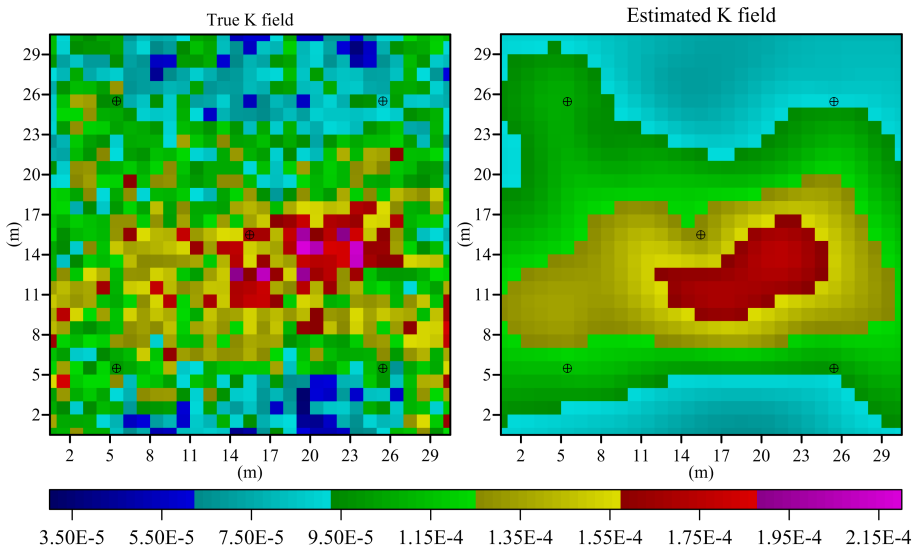

Figure 3.6: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for a random generated formation and constant boundary conditions.

Table 3.5: True and estimated characteristics of the random generated formation of Figure 3.6.

	True	Estimated	
$\overline{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$1.110 \cdot 10^{-4}$	$1.089 \cdot 10^{-4}$	
$\sigma_{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$3.118 \cdot 10^{-5}$	$2.589 \cdot 10^{-5}$	
$\mathrm{S}_s\ \left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$5.001 \cdot 10^{-4}$	
$\theta\ \left(\mathrm{ms}^{-2}\right)$	$1.232 \cdot 10^{-2}$	$\sigma_R^2\ \left(\mathrm{ms}^{-2}\right)$	$1.010 \cdot 10^{-11}$
$\mathrm{MSE}\ \left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$3.895 \cdot 10^{-10}$		

Table 3.6: True and estimated characteristics of the random generated formation of Figure 3.7.

	True	Estimated	
$\overline{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$1.110 \cdot 10^{-4}$	$1.082 \cdot 10^{-4}$	
$\sigma_{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$3.118 \cdot 10^{-5}$	$2.735 \cdot 10^{-5}$	
$\mathrm{S}_s\ \left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.923 \cdot 10^{-4}$	
$\theta\ \left(\mathrm{ms}^{-2}\right)$	$2.606 \cdot 10^{-2}$	$\sigma_R^2\ \left(\mathrm{ms}^{-2}\right)$	$0.600 \cdot 10^{-11}$
$\mathrm{MSE}\ \left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$3.715 \cdot 10^{-10}$		

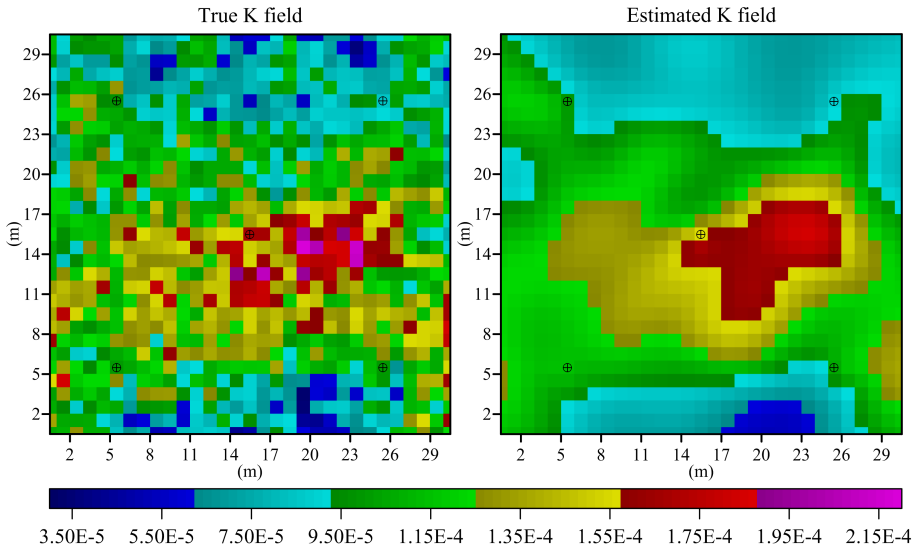

Figure 3.7: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for a random generated formation and non-stationary boundary conditions.

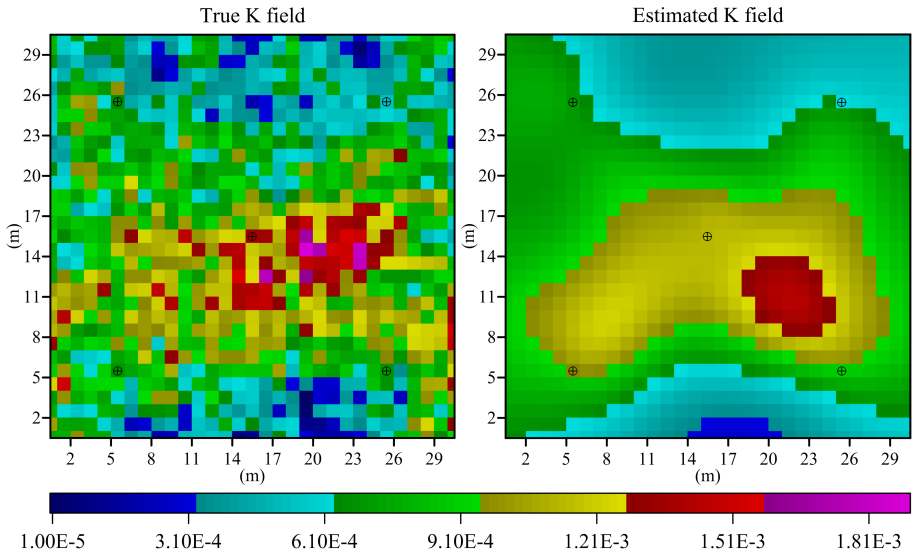

Figure 3.8: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for a random generated formation and constant boundary conditions.

Table 3.7: True and estimated characteristics of the random generated formation of Figure 3.8.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$7.921 \cdot 10^{-4}$	$7.826 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$3.118 \cdot 10^{-4}$	$2.594 \cdot 10^{-4}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$5.179 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$3.070 \cdot 10^{-2}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$5.000 \cdot 10^{-12}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$4.558 \cdot 10^{-8}$		

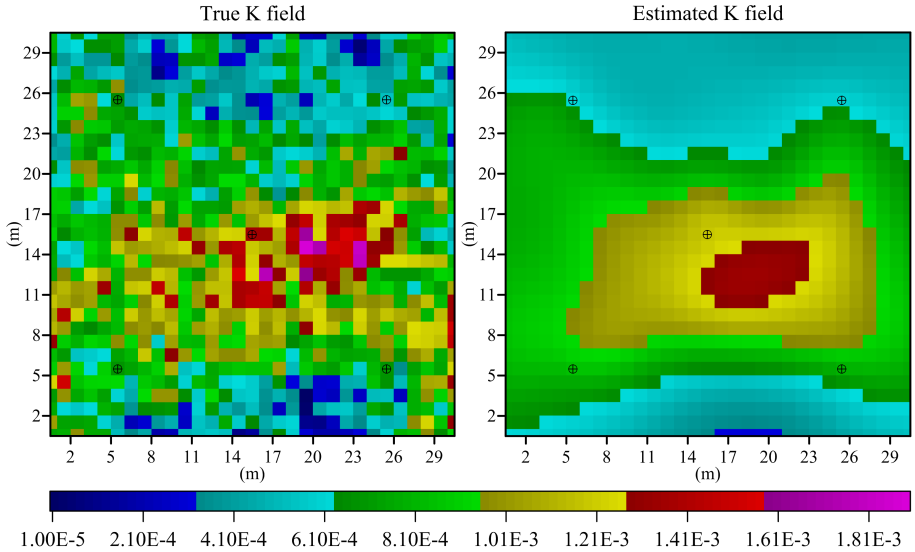

Figure 3.9: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for a random generated formation and non-stationary boundary conditions.

Table 3.8: True and estimated characteristics of the random generated formation of Figure 3.9.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$7.921 \cdot 10^{-4}$	$7.616 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$3.118 \cdot 10^{-4}$	$2.520 \cdot 10^{-4}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$5.030 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$1.329 \cdot 10^{-2}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$0.190 \cdot 10^{-12}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$3.963 \cdot 10^{-8}$		

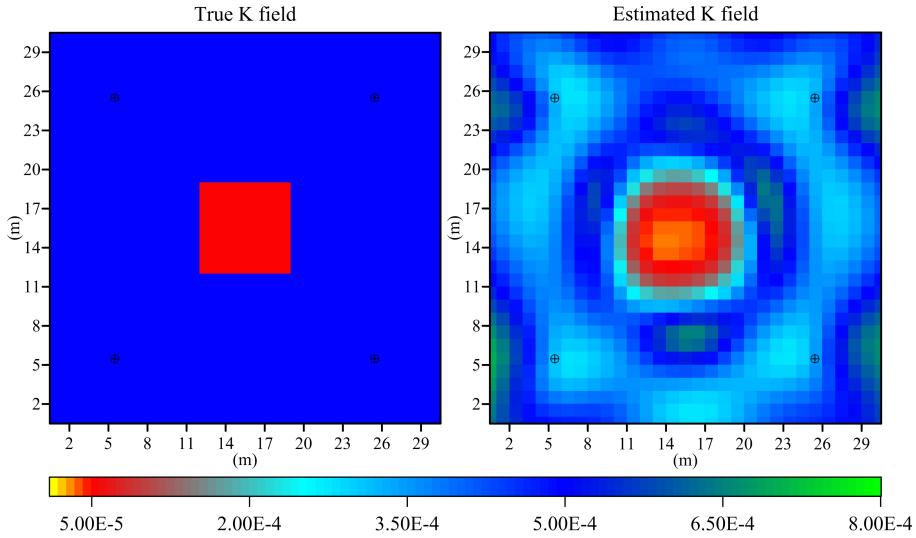

Figure 3.10: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an aquifer with central inclusion and non-stationary boundary conditions.

Table 3.10 where the value of MSE is decreased. This behavior can be expected because information is increased by the presence of a new data set coming from pumping in the additional well. The location of the additional point is important in order to collect independent data respect to the previous monitoring set: better distributed on the interesting area, more information is collected to the inverse procedure because the sensitivity matrix contains a greater number of elements which remarkably differ from zero.

The case depicted in Figure 3.12 considers the same field of the Figures 3.10

Table 3.9: True and estimated characteristics of the aquifer with central inclusion of Figure 3.10.

	True	Estimated	
$\overline{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$4.771 \cdot 10^{-4}$	$3.977 \cdot 10^{-4}$	
$\sigma_{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$0.990 \cdot 10^{-4}$	$1.291 \cdot 10^{-4}$	
$\mathrm{S}_s\ \left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.266 \cdot 10^{-4}$	
$\theta\ \left(\mathrm{ms}^{-2}\right)$	$2.067 \cdot 10^{-1}$	$\sigma_R^2\ \left(\mathrm{ms}^{-2}\right)$	$1.300 \cdot 10^{-11}$
$\mathrm{MSE}\ \left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$1.766 \cdot 10^{-8}$		

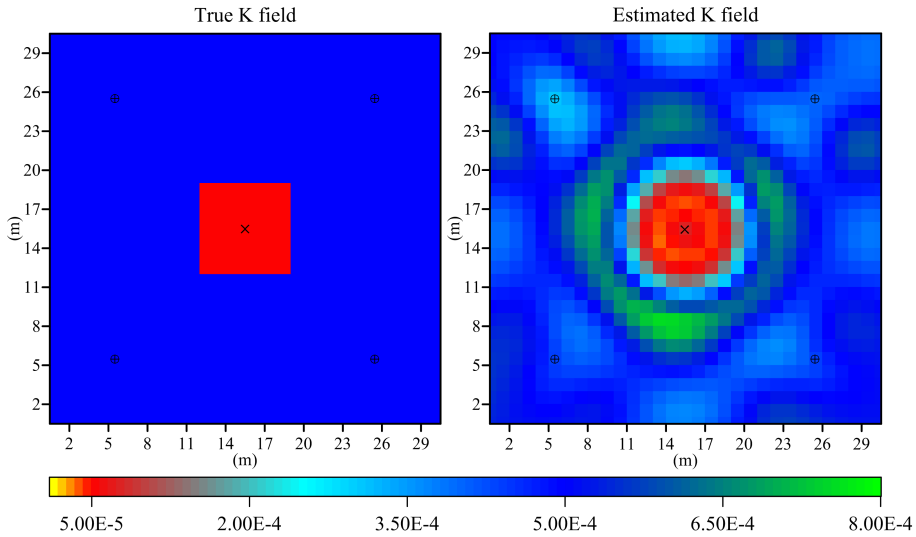

Figure 3.11: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an aquifer with central inclusion and non-stationary boundary conditions.

Table 3.10: True and estimated characteristics of the aquifer with central inclusion of Figure 3.11.

	True	Estimated	
$\overline{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$4.771 \cdot 10^{-4}$	$4.671 \cdot 10^{-4}$	
$\sigma_{\mathrm{K}}\ \left(\mathrm{ms}^{-1}\right)$	$0.990 \cdot 10^{-4}$	$1.346 \cdot 10^{-4}$	
$\mathrm{S}_s\ \left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.785 \cdot 10^{-4}$	
$\theta\ \left(\mathrm{ms}^{-2}\right)$	$1.850 \cdot 10^{-1}$	$\sigma_R^2\ \left(\mathrm{ms}^{-2}\right)$	$0.800 \cdot 10^{-11}$
$\mathrm{MSE}\ \left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$0.962 \cdot 10^{-8}$		

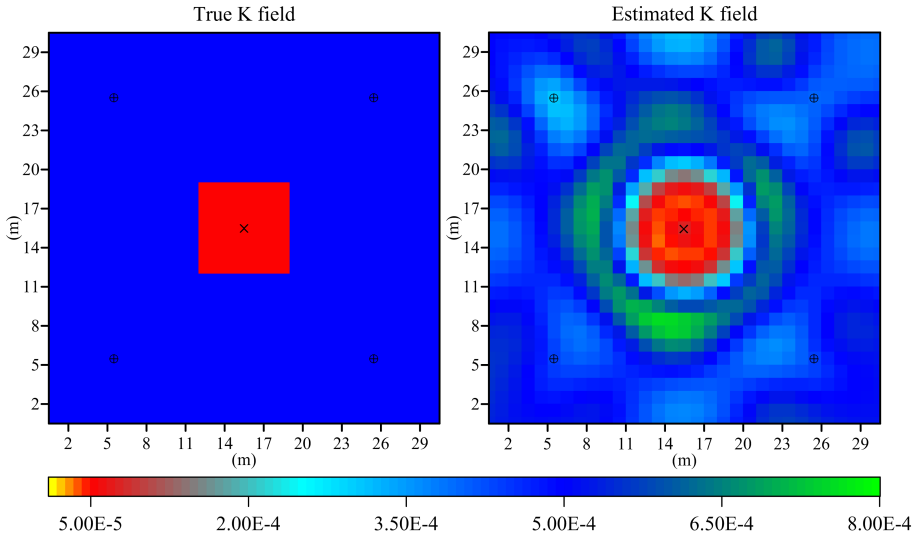

Figure 3.12: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an aquifer with central inclusion and non-stationary boundary conditions.

and 3.11 with the difference that now in the central inclusion there is only an observation point (not used as extraction well). The results are very similar to those shown in Figure 3.11 where the observation point located in the inclusion is used also as extraction well. Also in this case, considering an additional observation point located close to the inclusion gives additional and better sensitivity values to the parameters of the area surrounding the point itself. This suggests that if a net of wells is already available and there is a suspect of an inclusion in a given location an additional observation point close to the inclusion improves the solution of the inverse problem. As described in Butler et al. (1999) there are new cost efficient techniques to collect head data in aquifer without the need of well excavations and that can be used for accurately and rapidly measuring the drawdowns.

Figure 3.13 shows the true and estimated hydraulic conductivity field of the same case of Figure 3.12 with the only difference that the observed drawdowns (the measurements) are corrupted with a random small error before using them in the estimation process. The inverse solution is still accurate and the variance

Table 3.11: True and estimated characteristics of the aquifer with central inclusion of Figure 3.12.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$4.771 \cdot 10^{-4}$	$4.642 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$0.990 \cdot 10^{-4}$	$1.311 \cdot 10^{-4}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.817 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$2.175 \cdot 10^{-1}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$6.000 \cdot 10^{-12}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$0.892 \cdot 10^{-8}$		

Table 3.12: True and estimated characteristics of the aquifer with central inclusion of Figure 3.13.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$4.771 \cdot 10^{-4}$	$4.797 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$0.990 \cdot 10^{-4}$	$1.166 \cdot 10^{-4}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.785 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$1.412 \cdot 10^{-1}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$1.000 \cdot 10^{-7}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$0.577 \cdot 10^{-8}$		

of the epistemic errors increases to take account of the erroneous measurements and confirm the importance of the estimation of the structural parameters.

Cases of a lateral homogeneous inclusion in an homogeneous field with one order of magnitude of difference between the hydraulic conductivities and with non-stationary boundary conditions are shown in Figures 3.14 and 3.15. As shown in Figure 3.15 and Table 3.14, also in this case, the use of one more observation point in the middle of the field gives better results in comparison to the case of 3.14.

3.5 Conclusions

In this chapter a Bayesian Geostatistical inverse method applied to hydraulic tomography is described. The approach follows the Quasi-Linear method developed by Kitanidis (1995). The methodology is tested by means of different synthetic cases in transient flow conditions and with constant and non-stationary

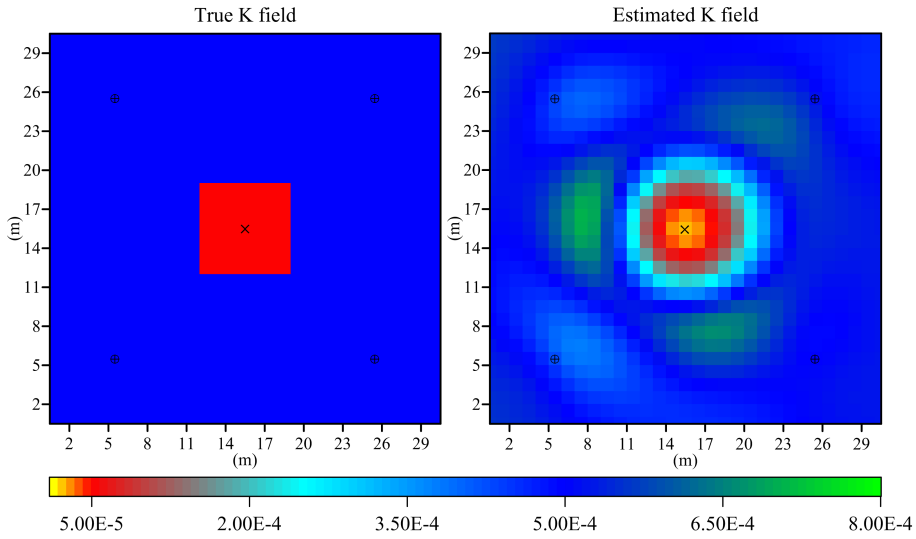

Figure 3.13: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an aquifer with central inclusion and non-stationary boundary conditions.

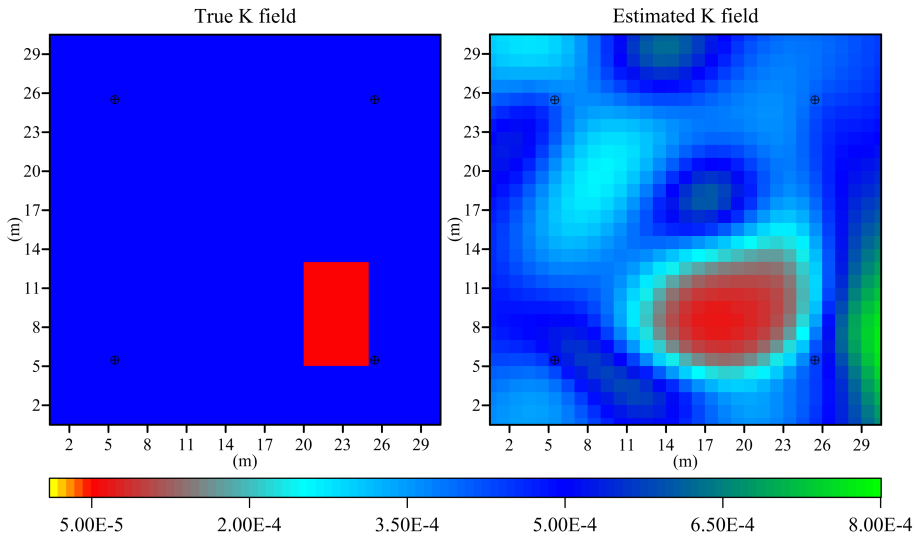

Figure 3.14: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an aquifer with lateral inclusion and non-stationary boundary conditions.

Table 3.13: True and estimated characteristics of the aquifer with lateral inclusion of Figure 3.14.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$4.813 \cdot 10^{-4}$	$4.797 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$0.899 \cdot 10^{-4}$	$1.335 \cdot 10^{-4}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$3.972 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$0.417 \cdot 10^{-1}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$2.900 \cdot 10^{-11}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$2.343 \cdot 10^{-8}$		

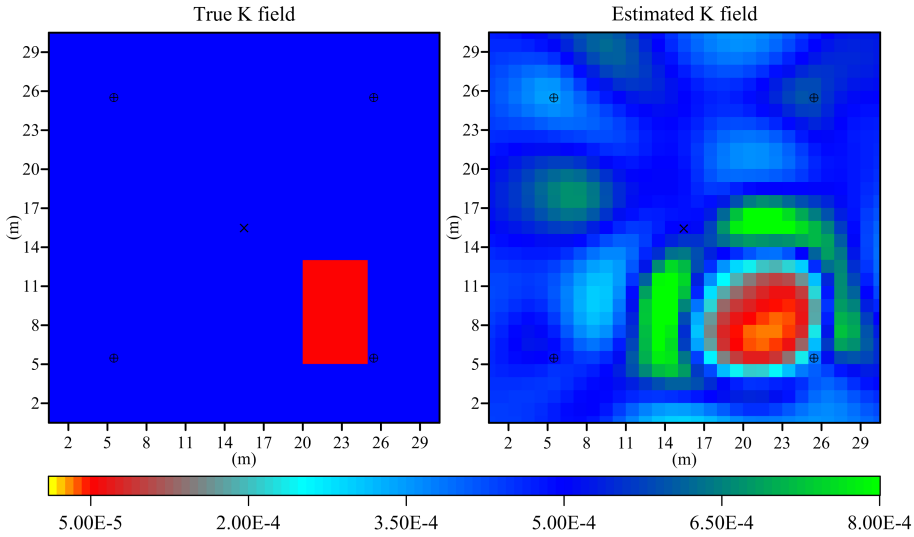

Figure 3.15: True and estimated hydraulic conductivity field $\left(\mathrm{ms}^{-1}\right)$ for an aquifer with central inclusion and non-stationary boundary conditions.

Table 3.14: True and estimated characteristics of the aquifer with lateral inclusion of Figure 3.15.

	True	Estimated	
$\overline{\mathrm{K}}$ $\left(\mathrm{ms}^{-1}\right)$	$4.813 \cdot 10^{-4}$	$4.680 \cdot 10^{-4}$	
σ_{K} $\left(\mathrm{ms}^{-1}\right)$	$0.899 \cdot 10^{-4}$	$1.379 \cdot 10^{-4}$	
S_s $\left(\mathrm{m}^{-1}\right)$	$5.000 \cdot 10^{-4}$	$4.799 \cdot 10^{-4}$	
θ $\left(\mathrm{ms}^{-2}\right)$	$1.157 \cdot 10^{-1}$	σ_R^2 $\left(\mathrm{ms}^{-2}\right)$	$5.020 \cdot 10^{-12}$
MSE $\left(\mathrm{m}^2\mathrm{s}^{-2}\right)$	$1.327 \cdot 10^{-8}$		

boundary conditions. During the tests of the methodology it seemed clear the importance of the estimation of the structural parameters in order to achieve a good solution and so an automatic procedure is implemented.

The use of transient flow and non-stationary boundary conditions is new in the application of the method to the hydraulic tomography.

Up to the present (on the basis of the available literature) the Quasi-Linear approach (as developed by Kitanidis (1995)) is used in steady-state hydraulic tomography (Fienen et al., 2008) and transient hydraulic tomography using temporal moments of drawdowns (Li et al., 2005, 2007). Obviously for the steady-state case and, as clearly specified, for the transient case using temporal moments of drawdowns, over the time period of the test, the boundary conditions, except for the pumping, do not change. Also the other methodologies described in Section 3.2 are all tested with constant boundary condition. Considering transient flow and non-stationary boundary conditions the inverse method must be conditioned on the direct head data and at least 3 or 4 observations in time for each observation point must be considered to properly describe the drawdown curve. This implies that also the sensitivity of observations to parameters must be calculated in the same locations and times. The increasing performance of computers and the use of the adjoint state method for the calculation of the sensitivity together with the properly use of the MODFLOW_2005-Adjoint (inefficiencies of the code and the correct formulation to use are described in Appendix A) allow the computation of the sensitivity in an acceptable time and therefore the possible use of non-stationary boundary conditions.

The use of non-stationary boundary conditions in the methodology, verified on synthetic models, extends the application of the hydraulic tomography to actual field applications where very often during a pumping test it can be unrealistic to consider the aquifer isolate from all that surrounds it. An application of the methodology to a field case is described in the next chapter.

Chapter 4

Field Work: the Boretto Research Site

4.1 Introduction

The Agency for the Po River Management (AIPO, Italy) owns the Boretto Research Site; it consists of two hydraulic laboratories, a geothecnical laboratory and a well field. At present, the Department of Civil, Environmental and Landscape Engineering and Architecture (DICATeA) of the University of Parma (Italy) has an agreement with AIPO about the development of research programs in one of the hydraulic laboratories and in the well site. The Research Site is located in Boretto (RE) a small village in the Emilia Romagna Region in the Northern Italy. The site is pretty close to the Po River (about 3 km), the area is flat and mainly addressed to agriculture activities with the exception of some industrial activities. The Figure 4.1 shows the location of the Boretto Research Site.

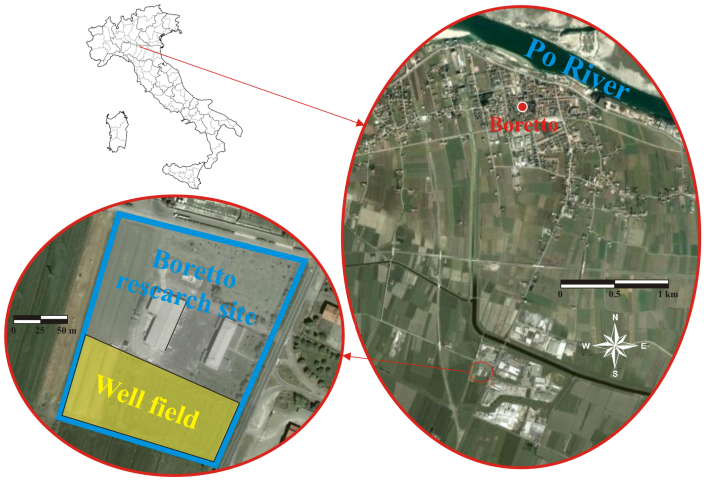

Figure 4.1: Location of the Boretto Research Site.

4.2 The Well Field

The well field, within the Boretto Research Site, is in a flat area with a mean elevation of 21.5 m a.s.l and covers an area of about 11.000 m^2 (about 160 m x 70 m). The test site is located in an alluvial deposit close to the Po River; the main aquifer is confined and 16 m thick; it is mainly constituted by sand and comprised between two clay layers. The upper clay formation extends from the ground surface to a depth of about 15 m. A stratigraphic column showing the vertical sequence of underground materials has been obtained during the well drilling in the location P1 (Figure 4.2) and is depicted in the Appendix B.

The well field facility was established in June 2008 and it consists of one main well and four observation points. Both the well and the boreholes have a plastic casing and are screened over the entire extension of the confined aquifer. The locations of the wells in the field (in a local reference system) and their main characteristics are reported in Figure 4.2 and in Table 4.1.

In Table 4.2 the mutual distances between wells, useful for all the next analyses, are reported.

Figure 4.2: Location of wells of the Boretto field.

Table 4.1: Main features of the wells of the Boretto field.

Name	Diameter (mm)	Well head (m a.s.l.)	Depth (m)	Screen Interval (m)	Location	
					x (m)	y (m)
P1	125	21.50	30.00	15.00 - 30.00	124.92	68.39
P2	125	21.27	30.00	15.00 - 30.00	99.79	39.74
P3	125	21.31	30.00	15.00 - 30.00	171.12	58.29
P4	125	21.19	30.00	15.00 - 30.00	23.53	60.06
P5	330	21.49	35.00	15.00 - 30.00	124.02	57.40

Table 4.2: Mutual distances between wells of the Boretto field.

	Distances (m)				
	P1	P2	P3	P4	P5
P1	-	38.12	47.29	101.74	11.02
P2	38.12	-	73.71	78.92	29.99
P3	47.29	73.71	-	147.61	47.10
P4	101.74	78.92	147.61	-	100.53
P5	11.02	29.99	47.10	100.53	-

4.3 Available Instruments and Capabilities

The Boretto well field is equipped with the following instruments:

- 1 submersible pump with 22 kW of power; it is able to extract a maximum flow rate of 30 ls^{-1};
- 1 submersible pump with 1 kW of power and small diameter (100 mm) able to extract a maximum flow rate of 3 ls^{-1};
- 5 submersible pressure and temperature probes with integrated data logging;
- 2 magnetic flow meters (*Mag-Flow*);
- 1 manual water level meter (contact gauge).

With the above described instruments, in the Boretto well field there are the capabilities to conduct traditional pumping tests and sequential tests too (in tomographic way). The five pressure probes are permanently installed one for each well and the water levels can be measured and stored continuously and with selectable sampling interval during the tests. The collected data can be downloaded from the probes via infrared interface (IrDA) powered by means of a laptop without the removal of the sensors. The first pump is permanently installed in the main well P5 and one of the magnetic flow meter, dedicate to it, measures the extracted water flow rate. The small pump, instead, can be easily moved around the observation wells and the extracted water flow rate is measured by means of the second *Mag-Flow*. The manual water level meter is useful to set the reference level values of the pressure sensors and to verify their proper operation. More information and data sheets about the instruments are provided in the Appendix C.

Dipole tests in which the water extracted out from one well is injected into another one can also be performed without too much difficulties.

Table 4.3: Main characteristics of the pumping tests.

Test	Test Duration (hours)	Pumping Well	Pumping Rate (ls^{-1})
1	22.87	P1	1.615
2	23.20	P2	2.800
3	13.02	P3	2.750
4	17.21	P4	2.560
5	26.75	P5	15.24

4.4 The Well Field Tests

At the present, about ten pumping tests have been performed in the Boretto well field; in this work only the five tests conducted in tomographic way are presented. The tests start on January 2009 and finished on March 2009.

During the sequential tests, water was extracted from one of the five wells and the drawdown curves was recorded at all the wells contemporaneously. At the end of each test and after the recovering of the aquifer, the stress location was moved and the aquifer responses was still monitored in all the wells. The procedure was repeated until all the wells were used for extraction. The sampling rate of the drawdown was variable from 0.5 to 30 samples per minute. The pumping rate was enforced by a gate valve and maintained constant during each pumping test; the extracted flow rate was measured by means of a *Mag-Flow* and recorded by a data logger (6 samples per minute). The extracted water was moved away in an open channel that does not interfere with the aquifer.

With reference to Figure 4.2, Table 4.3 shows the main characteristics of each pumping test.

Figures 4.3-4.7 show the drawdowns versus time for each pumping test in the five wells.

As observable in Figures 4.3-4.7, the drawdowns-time curves do not exhibit an asymptotic behavior at large time as would be expected for an homogeneous aquifer with infinite areal extent. This behavior can be explained by the fact that the Po River water level changed during the tests. As shown in Figure 4.8 for a

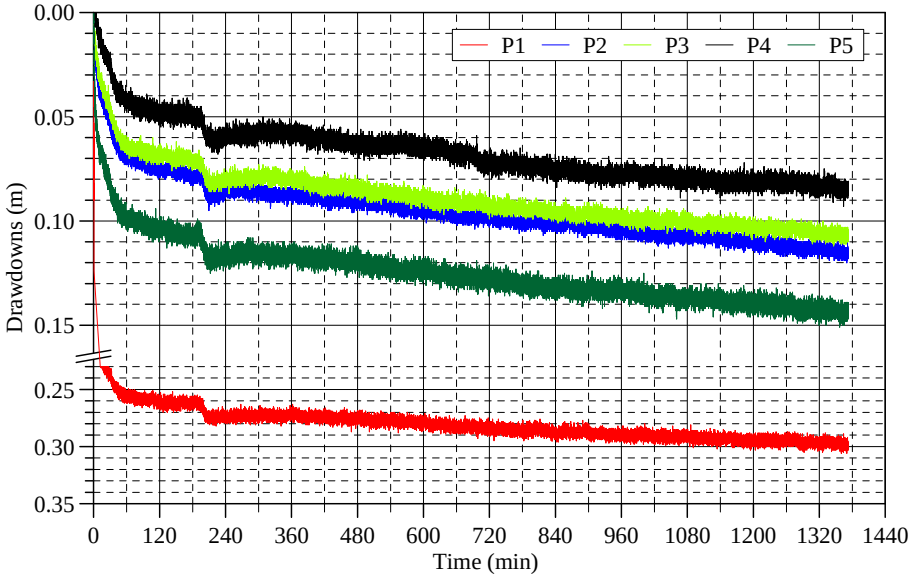

Figure 4.3: Drawdowns versus time for pumping test 1.

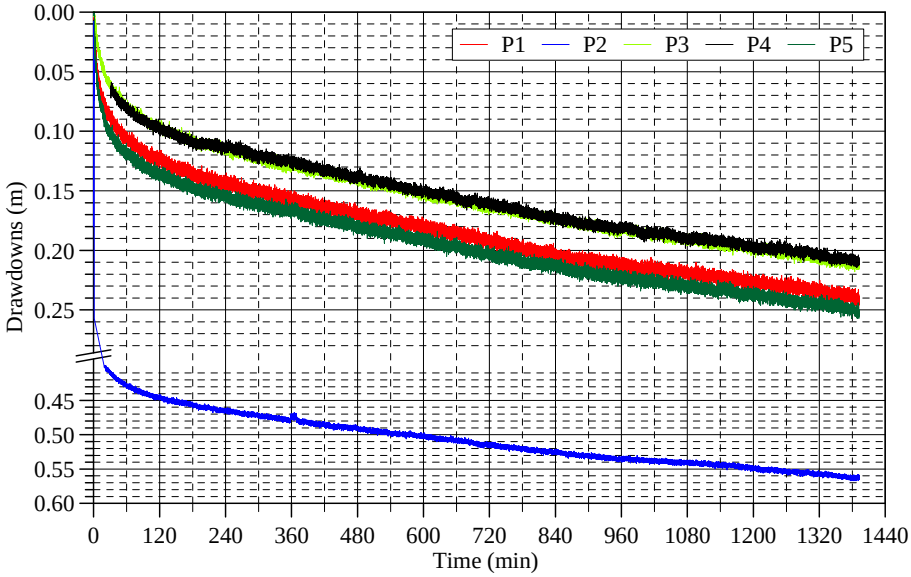

Figure 4.4: Drawdowns versus time for pumping test 2.

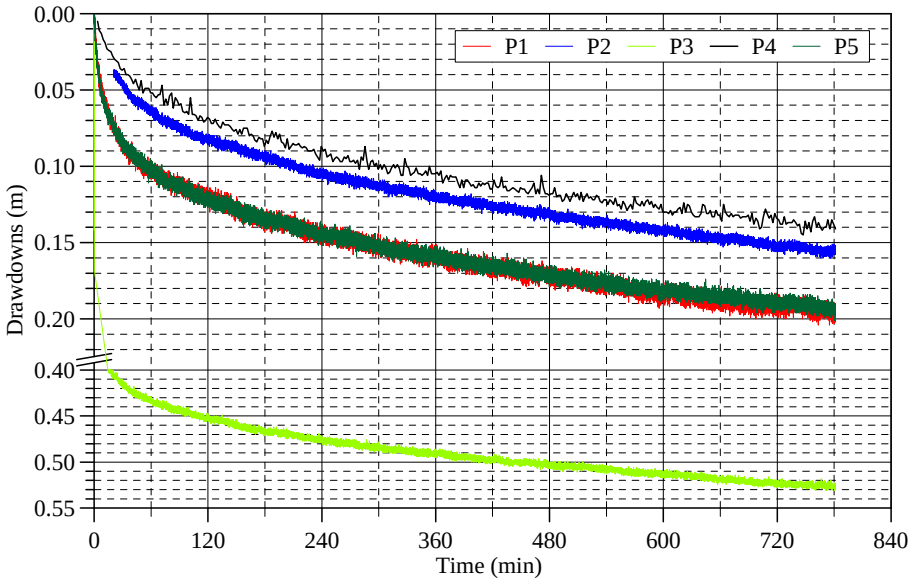

Figure 4.5: Drawdowns versus time for pumping test 3.

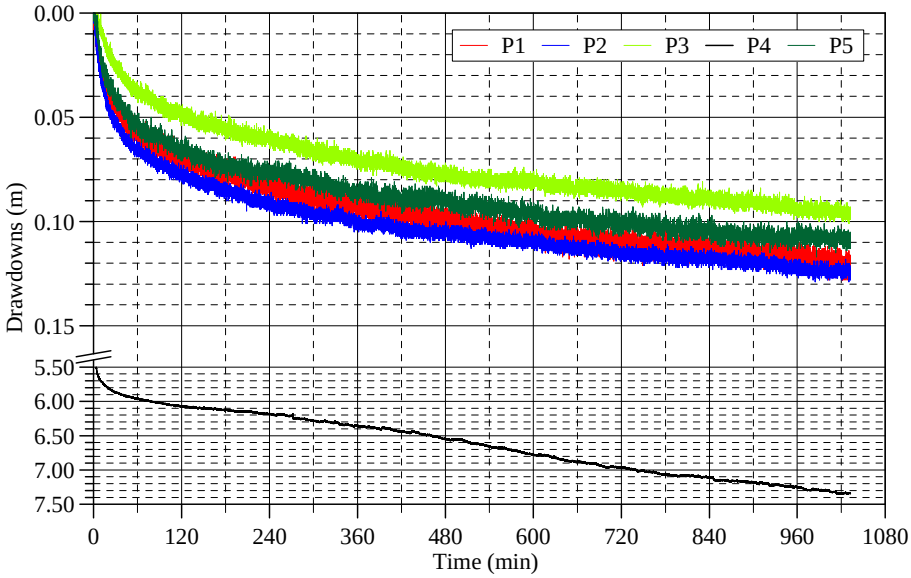

Figure 4.6: Drawdowns versus time for pumping test 4.

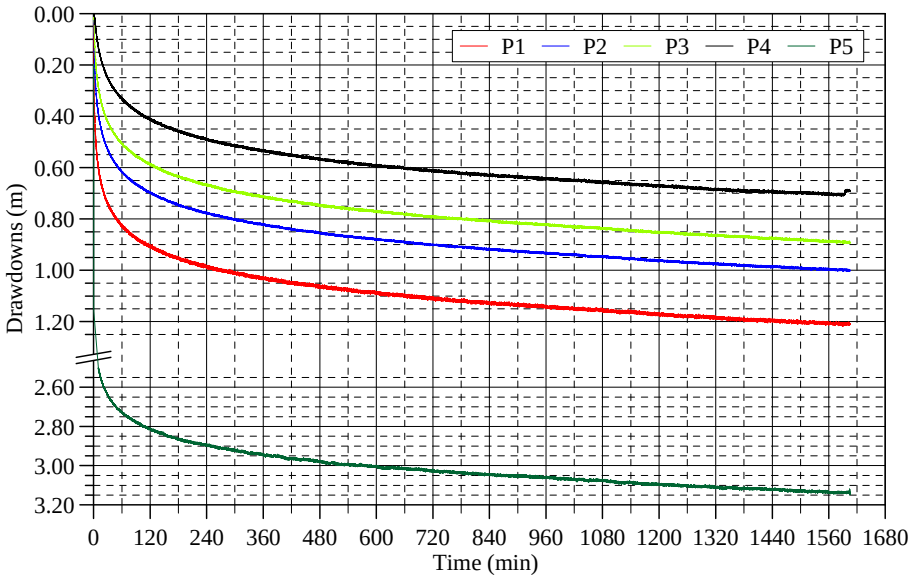

Figure 4.7: Drawdowns versus time for pumping test 5.

representative time interval (a period without pumping tests) and one well in the field, there is a close relation between the river water level and the groundwater level in the field. This appears not strange recalling that the aquifer is located in an alluvial deposit and it is very close to the Po River. The Po River is therefore a boundary condition for the aquifer and its variations are not negligible at the distance of the well field and during the pumping tests.

4.5 Aquifer Characterization

4.5.1 Traditional Aquifer Analyses

The data collected during the pumping tests were all analyzed in traditional ways (Drawdown-Time, Drawdown-Distance, Drawdown-Time-Distance methods) as described in Section 2.3. The raw data, as recorded by means of the pressure probes, were converted in term of drawdowns and filtered by means of the MATLAB (Mathworks, 2002) System Identification Toolbox *idfilt* function in order to cut off the higher frequencies of the signal. An example of the data filtered and

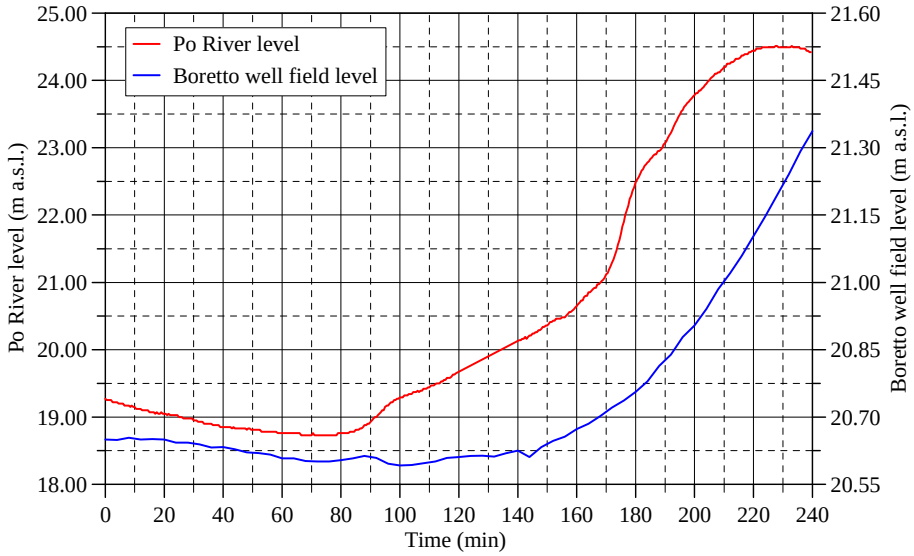

Figure 4.8: Po River water level and well field groundwater level versus time for a representative time interval.

before the filtering is reported in Figure 4.9. See Appendix E for more details about *idfilt*.

Because of the strict connection between the Po River water level and the groundwater level in the well field, two different sets of traditional analyses were conducted. The first set considers no influence of the boundary conditions on the observed drawdowns during each pumping test; the second set takes account of the fact that part of the differences in the groundwater level during the pumping tests was due to the changing boundary conditions. All the analyses were conducted by means of automatic codes developed by the Writer in MATLAB (Mathworks, 2002) using the constrained nonlinear optimization routine *fmincon* to minimize the sum of square differences between observed and theoretical drawdowns. See Appendix E for more details about *fmincon*.

Traditional Analyses Considering no Influence of the Boundary Conditions. In this set of analyses the measured drawdowns in all the observation wells were considered; the measured drawdowns at the pumping wells were not

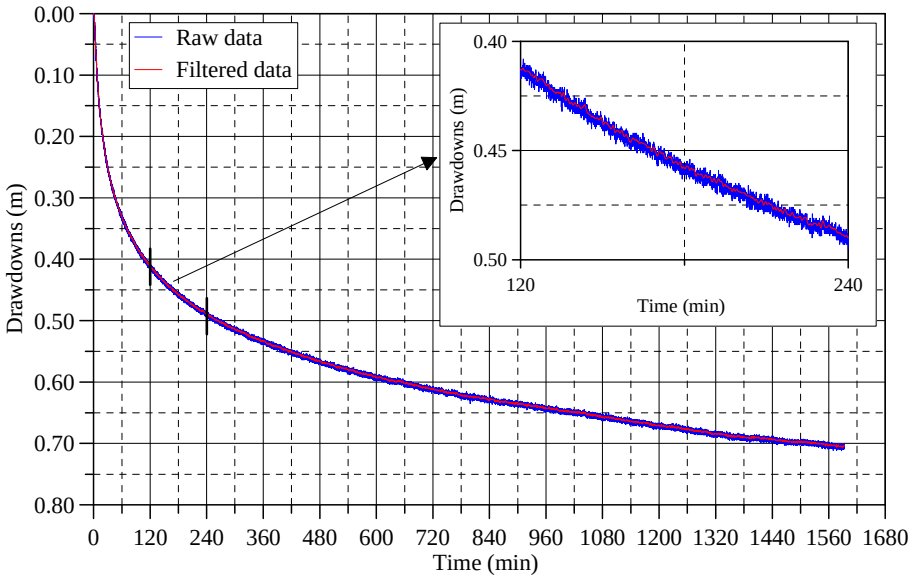

Figure 4.9: Example of drawdown data filtered and before the filtering.

used in the calculations. A total number of 20 drawdown curves were therefore available for the analyses. The Tables 4.4 and 4.5 show the results of the Drawdown-Time analyses (transmissivity and hydraulic conductivity, storativity and specific storage) based on the 20 drawdown hydrographs separately analyzed; the mean value and standard deviation for each test and for each observation points and the mean of the whole results are also showed, the second row of the tables contains the wells used as observation points, the second column the pumping wells.

In the Drawdown-Distance analyses for each pumping test the hydraulic properties were estimated using the drawdowns at different times but contemporaneously in all the observation points. About one estimation every five minutes was performed and the resulting Figures are showed and discussed in Appendix D, here in Tables 4.6 and 4.7 only the temporal mean and standard deviation for each test of the estimated values and the total mean and standard deviation are reported.

In the end the hydraulic parameters were estimated by means of the Drawdown-

Table 4.4: Drawdown-Time analyses (using measured drawdowns): Transmissivity and Hydraulic Conductivity for each pumping test.

Transmissivity T $\left(m^2s^{-1} \cdot 10^{-3}\right)$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Stdev
1	P1	-	8.27	8.42	8.15	8.16	8.25	0.12
2	P2	4.82	-	4.53	4.51	4.91	4.69	0.20
3	P3	6.04	5.77	-	5.84	6.25	5.98	0.22
4	P4	9.00	9.84	9.52	-	10.05	9.61	0.46
5	P5	10.24	10.31	10.37	10.55	-	10.37	0.13
	Mean	8.42	8.55	8.21	7.26	7.34		
	Stdev	2.52	2.05	2.58	2.66	2.25		
						Total	7.78	2.21
Hydraulic Conductivity K $\left(ms^{-1} \cdot 10^{-4}\right)$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Stdev
1	P1	-	5.17	5.26	5.09	5.10	5.16	0.08
2	P2	3.01	-	2.83	2.82	3.06	2.93	0.12
3	P3	3.78	3.60	-	3.65	3.91	3.74	0.14
4	P4	5.63	6.15	5.95	-	6.28	6.00	0.29
5	P5	6.40	6.44	6.48	6.59	-	6.48	0.08
	Mean	4.70	5.34	5.13	4.54	4.59		
	Stdev	1.58	1.28	1.61	1.66	1.40		
						Total	4.86	1.38

Table 4.5: Drawdown-Time analyses (using measured drawdowns): Storativity and Specific Storage for each pumping test.

Storativity S $(\cdot 10^{-4})$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Stdev
1	P1	-	8.42	7.11	7.16	16.70	9.85	4.61
2	P2	48.17	-	45.35	45.12	49.05	46.92	1.98
3	P3	15.01	20.86	-	7.46	15.27	14.65	5.50
4	P4	7.65	6.11	6.70	-	6.69	6.79	0.64
5	P5	6.67	5.14	5.12	4.93	-	5.47	0.81
	Mean	17.68	10.13	11.23	10.92	22.83		
	Stdev	16.24	7.28	9.87	8.88	20.37		
						Total	14.56	13.00
Specific Storage S_s $(m^{-1} \cdot 10^{-5})$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Std
1	P1	-	5.26	4.44	4.48	10.44	6.16	2.88
2	P2	25.87	-	16.24	15.09	32.91	22.53	8.44
3	P3	9.38	13.04	-	4.66	9.54	9.16	3.44
4	P4	4.79	3.82	4.19	-	4.18	4.24	0.40
5	P5	4.17	3.21	3.20	3.08	-	3.42	0.51
	Mean	11.05	6.33	7.02	6.83	14.27		
	Stdev	10.15	4.55	6.17	5.55	12.73		
						Total	9.10	8.12

Table 4.6: Drawdown-Distance analyses (using measured drawdowns): Mean values of Transmissivity and Hydraulic Conductivity for each pumping test.

Test	T $(m^2s^{-1} \cdot 10^{-3})$	Stdev (T) $(m^2s^{-1} \cdot 10^{-3})$	K $(ms^{-1} \cdot 10^{-4})$	Stdev (K) $(ms^{-1} \cdot 10^{-4})$
1	10.20	0.54	6.37	0.34
2	9.26	0.14	5.79	0.09
3	8.91	7.41	5.57	4.63
4	9.45	1.01	5.91	0.63
5	10.69	0.29	6.68	0.18
Mean	9.70		6.06	
Stdev	0.73		0.45	

Table 4.7: Drawdown-Distance analyses (using measured drawdowns): Mean values of Storativity and Specific Storage for each pumping test.

Test	S ($\cdot 10^{-4}$)	Stdev (S) ($\cdot 10^{-4}$)	S_s ($m^{-1} \cdot 10^{-5}$)	Stdev (S_s) ($m-1 \cdot 10^{-5}$)
1	2.97	1.04	1.85	10.42
2	4.55	14.42	2.84	144.18
3	4.91	2.50	3.07	24.97
4	6.94	1.21	4.34	12.07
5	4.50	0.41	2.81	4.11
Mean	4.77		2.98	
Stdev	1.42		0.89	

Table 4.8: Drawdown-Time-Distance analyses (using measured drawdowns): Mean values of Transmissivity and Hydraulic Conductivity, Storativity and Specific Storage for each pumping test.

Test	T ($m^2s^{-1} \cdot 10^{-3}$)	K ($ms^{-1} \cdot 10^{-4}$)	S ($\cdot 10^{-4}$)	S_s ($m^{-1} \cdot 10^{-5}$)
1	9.52	5.95	4.08	2.55
2	7.97	4.98	13.07	8.17
3	8.07	5.98	13.88	9.00
4	9.55	5.97	6.79	4.24
5	10.59	6.62	4.69	2.93
Mean	9.14	5.90	8.54	5.38
Stdev	1.11	0.58	4.66	3.00

Time-Distance analysis. The whole data set for each test was processed so that the sum of the square differences between observed and theoretical drawdowns (calculated in all the observation points and for the entire duration of the test) was minimized. In this way a mean value of the hydraulic aquifer property for each test was estimated and the results are summarized in Table 4.8.

Traditional Analyses Considering the Influence of the Boundary Conditions. As shown in Figure 4.8 the closeness of the Po River affects the groundwater level in the Boretto well field. The groundwater level variations in the field present a non-zero phase shift and non-zero attenuation respect to the

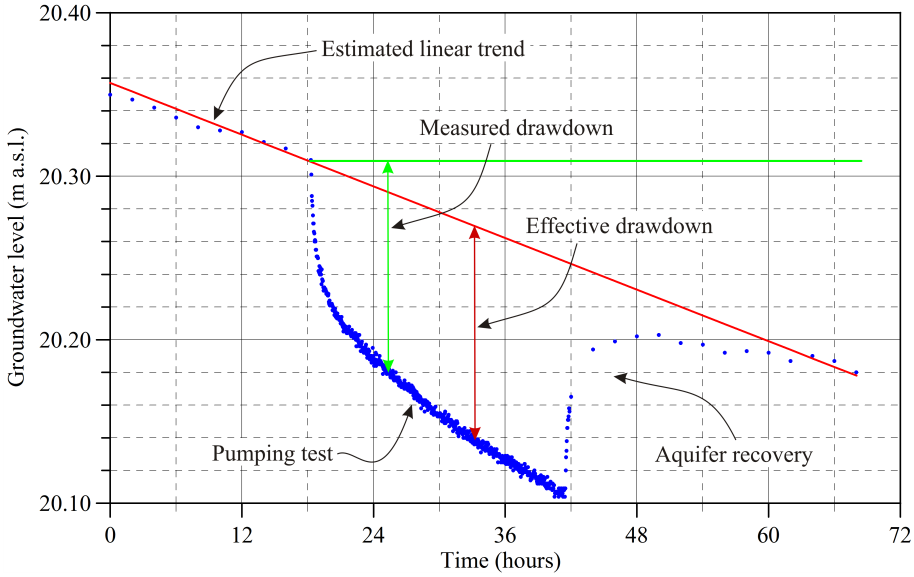

Figure 4.10: Example of mean trend of the groundwater during a pumping test due to changing boundary conditions.

water river level.

To take account of the influence of the changing Po River water level (a boundary conditions for the aquifer) the variations of the field groundwater level due to the boundary conditions and during each test were considered linear with a slope estimated on the basis of the mean trend of the groundwater level recorded in the wells immediately before the test and the mean trend immediately after the recovering of the aquifer. Assuming the aquifer homogeneous (as assumed in the traditional analyses) and therefore the system linear, the effective drawdowns were then estimated referring not to the groundwater level at the beginning of each pumping test but to the groundwater level estimated in the field as if the pumping test was not performed. An example of this linear interpolation and estimation of effective drawdowns is showed in Figure 4.10.

Then the analyses performed were exactly the same described in the previous paragraph with the difference that the effective estimated drawdowns were used in the calculation. The results are tabulated in Tables 4.9-4.13 and the subscript

Table 4.9: Drawdown-Time analyses (using effective drawdowns): Transmissivity and Hydraulic Conductivity for each pumping test.

Transmissivity T_e $\left(m^2s^{-1} \cdot 10^{-3}\right)$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Stdev
1	P1	-	13.29	13.45	14.04	12.44	13.30	0.66
2	P2	9.04	-	9.06	9.35	9.12	9.15	0.14
3	P3	9.77	10.45	-	10.64	10.14	10.25	0.38
4	P4	13.81	14.79	15.57	-	15.82	15.00	0.90
5	P5	11.03	11.14	11.23	11.53	-	11.23	0.21
	Mean	10.92	12.42	12.33	11.39	11.88		
	Stdev	2.10	1.99	2.81	1.98	2.97		
						Total	11.79	2.22
Hydraulic Conductivity K_e $\left(ms^{-1} \cdot 10^{-4}\right)$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Stdev
1	P1	-	8.31	8.41	8.77	7.78	8.31	0.41
2	P2	5.65	-	5.66	5.84	5.71	5.72	0.09
3	P3	6.11	6.53	-	6.65	6.33	6.41	0.24
4	P4	8.63	9.25	9.73	-	9.89	9.37	0.56
5	P5	6.89	6.96	7.02	7.20	-	7.02	0.13
	Mean	6.82	7.76	7.71	7.12	7.43		
	Stdev	1.31	1.24	1.75	1.24	1.85		
						Total	7.37	1.39

"e" is used with the hydraulic parameters to make distinctive the use of the effective estimated drawdowns.

The results of the Drawdown-Time analyses show that the estimated value of the aquifer hydraulic parameters vary slightly considering different pumping and observation locations. This means that the aquifer presents heterogeneity but it is difficult to detect any spatial distribution of the hydraulic parameters. As also found by Straface et al. (2007) the variations of the means and standard deviation in the estimation of K or T are generally smaller than those of S or S_s. Different values of the hydraulic parameters are also estimated considering the effective drawdowns instead of the measured drawdowns, highlighting the importance and

Table 4.10: Drawdown-Time analyses (using effective drawdowns): Storativity and Specific Storage for each pumping test.

Storativity S_e $(\cdot 10^{-4})$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Stdev
1	P1	-	1.19	1.24	2.05	1.35	1.46	0.40
2	P2	7.48	-	7.19	6.02	8.26	7.24	0.93
3	P3	4.54	8.50	-	3.96	4.65	5.41	2.08
4	P4	3.21	2.05	3.10	-	2.41	2.70	0.56
5	P5	4.15	3.57	3.77	3.94	-	3.86	0.24
	Mean	4.84	3.83	3.83	4.00	4.17		
	Stdev	1.84	3.27	2.49	1.62	3.06		
						Total	4.13	2.29
Specific Storage S_{se} $(m^{-1} \cdot 10^{-5})$								
Test	Pump. Well	P1	P2	P3	P4	P5	Mean	Std
1	P1	-	0.74	0.77	1.28	0.84	0.91	0.25
2	P2	4.67	-	4.49	3.76	5.16	4.52	0.58
3	P3	2.84	5.31	-	2.48	2.91	3.38	1.30
4	P4	2.01	1.28	1.94	-	1.51	1.69	0.35
5	P5	2.59	2.23	2.35	2.46	-	2.41	0.15
	Mean	3.03	2.39	2.39	2.50	2.60		
	Stdev	1.15	2.04	1.55	1.01	1.91		
						Total	2.58	1.43

Table 4.11: Drawdown-Distance analyses (using effective drawdowns): Mean values of Transmissivity and Hydraulic Conductivity for each pumping test.

Test	T_e $(m^2s^{-1} \cdot 10^{-3})$	Stdev (T_e) $(m^2s^{-1} \cdot 10^{-3})$	K_e $(ms^{-1} \cdot 10^{-4})$	Stdev (K_e) $(ms^{-1} \cdot 10^{-4})$
1	10.19	0.53	6.37	0.33
2	9.23	0.56	5.77	0.35
3	8.54	7.50	5.34	4.68
4	9.37	1.01	5.85	0.63
5	10.68	0.26	6.67	0.16
Mean	9.60		6.00	
Stdev	0.84		0.53	

Table 4.12: Drawdown-Distance analyses (using effective drawdowns): Mean values of Storativity and Specific Storage for each pumping test.

Test	S_e $(\cdot 10^{-4})$	Stdev (S_e) $(\cdot 10^{-4})$	S_{se} $(m^{-1} \cdot 10^{-5})$	Stdev (S_{se}) $(m{-}1 \cdot 10^{-5})$
1	7.93	3.04	4.95	30.24
2	9.76	21.85	6.10	218.53
3	11.49	2.18	7.18	21.83
4	14.27	5.21	8.92	52.12
5	5.27	0.60	3.30	5.98
Mean	9.74		6.10	
Stdev	3.42		2.14	

Table 4.13: Drawdown-Time-Distance analyses (using effective drawdowns): Mean values of Transmissivity and Hydraulic Conductivity, Storativity and Specific Storage for each pumping test.

Test	T_e $(m^2s^{-1} \cdot 10^{-3})$	K_e $(ms^{-1} \cdot 10^{-4})$	S_e $(\cdot 10^{-4})$	S_{se} $(m^{-1} \cdot 10^{-5})$
1	10.70	6.68	5.65	3.53
2	8.20	5.12	32.66	20.41
3	9.74	7.63	5.80	4.53
4	12.71	7.94	5.14	3.21
5	10.81	6.75	4.81	3.01
Mean	10.43	6.83	10.81	6.94
Stdev	1.65	1.10	12.22	7.55

the difficulty to take account of non-stationary boundary conditions.

Analyzing the data by means of the Drawdown-Distance method, it is possible to see (Tables 4.6, 4.7, 4.11 and 4.12) that the estimated mean transmissivity values (or hydraulic conductivities) are almost the same considering measured or effective drawdowns; more variations are instead present in the estimation of the storativity values. This behavior can be explained considering the Jacob approximation of the Theis solution; according to this solution, in a semi-log graph $(\ln(t),\ s(r,t))$ the drawdowns s are disposed on a straight line and the value of the transmissivity is proportional to the slope of the line itself while the value of the storativity is the intersection between the straight line and the x axis. The analysis of drawdowns using several observation points (at each set time) doesn't change the slope of the straight line using effective instead of measured drawdowns; it is just a translation of the reference system. This implies that the transmissivity values are almost the same whereas the storativity values suffer of more variations. This approach seems able to estimate a reasonable value of transmissivity independently of the influence of the boundary conditions.

Drawdown-Time-Distance analyses give estimated hydraulic parameters that should represent mean values in space and time but also in this case potential heterogeneity is difficult to detect and to locate certainly.

Concluding, the assumption of homogeneity in all the traditional aquifer tests yields to results of questionable interpretation and a method able to take account of the spatial variability of the aquifer properties is needed to overcame these difficulties.

4.5.2 Hydraulic tomography

To remove the hypothesis of homogeneous aquifer and to obtain a detailed knowledge of the hydraulic properties of the Boretto well field, the data collected during the pumping tests were analyzed by means of the hydraulic tomography approach described in Section 3.3. The forward aquifer model, the inversion procedure and the results are reported in succession.

4.5.2.1 Forward Aquifer Model

To apply the inverse procedure, a forward model able to reproduce the observed values at the same locations and times for each pumping tests was needed. The forward model was always MODFLOW_2005 and the sensitivity was always calculated by means of MODFLOW_2005-Adjoint.

To take into account the variations of the well field groundwater level due to the changing Po River water level, a model of the aquifer that extends until the river itself in North direction was developed (this model is referred to as regional model); the model was extended 1 km from the well field in the other directions. The use of a fine grid over the entire domain was computationally prohibitive so a telescopic mesh refinement procedure was used to better detail the model in the area of the well field (this model is referred to as local model); the boundary conditions for the local model were obtained from the larger model (the regional model) that encompasses the model in the area of interest. The Figure 4.11 shows the extents of the models.

The regional model had an extent of 2261 m $\times$ 4015 m, the local model had instead an extent of 261 m $\times$ 135 m in x and y direction respectively and totally covers the well field. The regional model had a grid with variable spacing: constant spacing of 9 m in the area of the well field (along x and y) and gradually increasing grid spacing away from this area; in the river direction the grid spacing first increases and then again decreases to better simulate the changing of the boundary condition. The local model had instead a regular grid of constant spacing of 3 m in both the plane directions. Both the models presented one confined layer with a thickness of 16 m. A summary of the characteristics of both the models are tabulated in Table 4.14.

Five couples (one regional and one local model) of models were developed, one for each pumping test; the differences among them were in the boundary conditions, the position of the extraction wells and the location of the observation points according to the considered field test. All the forward runs were performed in transient state with an extension in time of 3 hours starting from the beginning

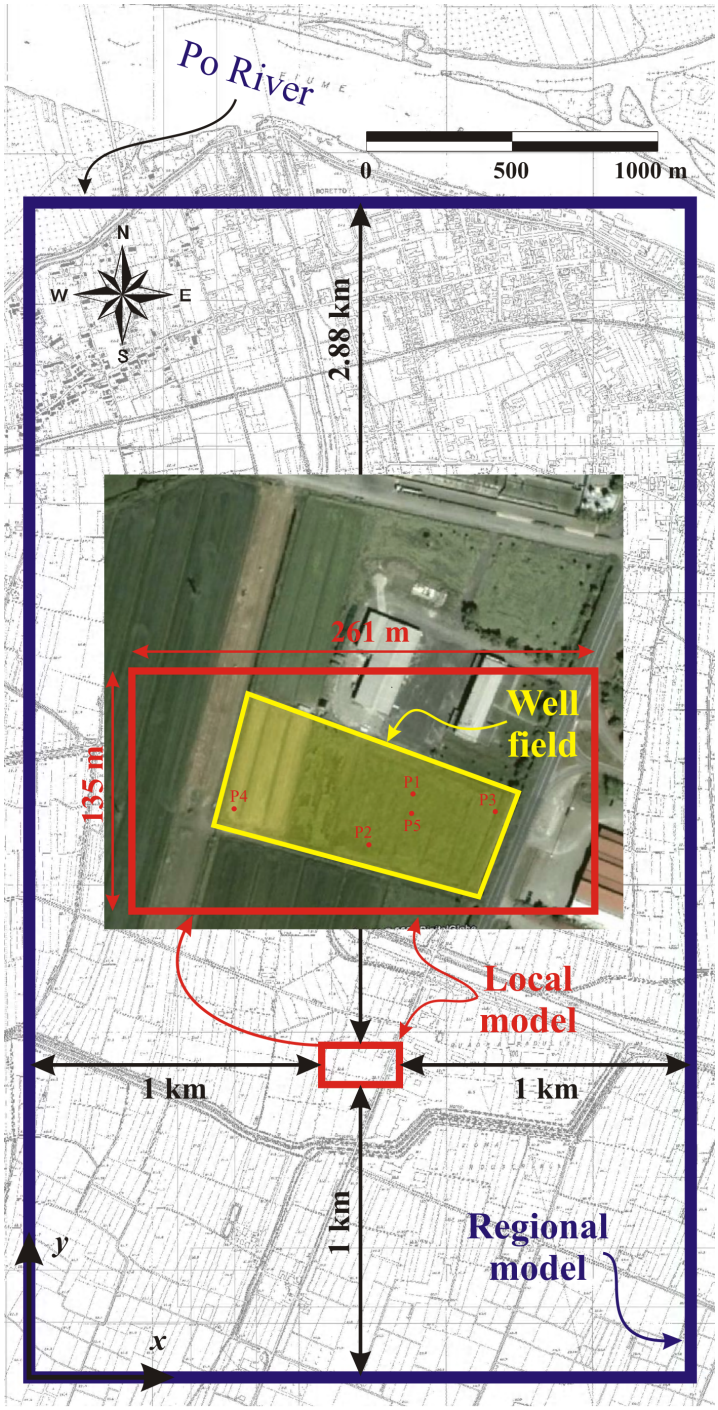

Figure 4.11: Extents of regional model and local model.

Table 4.14: Characteristics of regional model and local model.

	Regional model	Local model
Extent in x (m)	2261	261
Extent in y (m)	4015	135
Number of rows	57	45
Number of columns	55	87
Number of nodes	3135	3915
Grid spacing in x (m)	9-239	3
Grid spacing in y (m)	9-384	3

of each pumping test.

Each regional model had specified transient head boundary conditions on all the four edges; the northern boundary reproduced the river level during each test (with the level variation supposed linear for the duration of the test), the southern boundary level was instead estimated assuming a linear variation between the river water level and the field groundwater level at the beginning and the end[1] of each test and extending the straight line until the boundary. The same linear variation was used to define the eastern and western boundaries. The boundary conditions of each local model (specified heads) were defined reading and interpolating the hydraulic heads of the regional model after its run in the nodes that constitute the perimeter of the local model.

4.5.2.2 Inversion procedure

The parameters estimated with the inversion procedure were: the hydraulic conductivity in each of the 3915 nodes of the local model, a single value for the specific storage on the entire local model, and one hydraulic conductivity value and one specific storage value for the outer zone of the regional model that surrounds the local model. The covariance model for the hydraulic conductivity field of the local model was the linear one (as limiting case of the exponential model) as described in Subsection 3.3.2.4; no correlation was assumed between

[1] The groundwater level at the end of the test was the one estimated as if the pumping test was not performed and due only to the changing river level. See Section 2.3.

the hydraulic conductivity parameters of the local model and the other estimated parameters; the variances of these last parameters were set to a small value[2] equal to 10^{-4}. For each observation point four observations in time were considered: after about 5 min, 20 min, 1 hour and 3 hours from the beginning of the pumping tests.

The inversion procedure followed these steps:

1. **Initialization of the variables**. The hydraulic parameters of the local model and regional model were initialized to a constant value according to the information obtained by means of the traditional aquifer tests. The structural parameters were initialized to $\theta = 10^{-6}$ and $\sigma_R^2 = 10^{-4}$ so that the ratio $\frac{\theta}{\sigma_R^2}$ was less than 1.

2. **Running of forward regional models and adjoint models**. Using the current parameter estimation (at the beginning the initialization values), the forward regional model and the adjoint regional model run 5 times (one for each test) and provided the hydraulic heads needed as boundary conditions in the local model and the sensitivity of the hydraulic conductivity and specific storage of the outer zone of the regional model;

3. **Running of forward local models and adjoint models**. Using the current parameter estimation (at the beginning the initialization values), the local forward model and the adjoint local model run 5 times with the boundary conditions provided by the corresponding regional model. The predictions in the observation points (a total of 80 observations) and the sensitivity of the hydraulic conductivity field and specific storage of the local model were calculated;

4. **Calculation of the best estimate**. Solving the cokriging Equation 3.30 and by means of the Equation 3.31 a new candidate for the parameter vector was calculated. The *"line search"* procedure was performed to drive the objective function 3.41 to reach the minimum value. During the *"line search"*

[2] Actually the variance must be zero, but a zero value renders the covariance matrix singular and the inverse problem ill-conditioned

the forward models must run; the hydraulic conductivities estimated for the local models were scaled each time (each cell of the regional model assumed the mean value of the corresponding 9 cells of the local model) to became the hydraulic conductivities of the inner zone of the regional models. With the new estimation of the parameters the steps 2-4 were repeated until the improvement in the objective function was below the value 0.1.

5. **Estimation of the structural parameters**. At this point the orthonormal residuals were calculated and the metrics Q_2 and cR were evaluated. The automatic optimization algorithm (*fminsearch* (Appendix E)) chose the new ratio $\frac{\theta}{\sigma_R^2}$ of the structural parameters and the steps 2, 3, 4 and 5 were repeated until the minimum of cR was reached. The last evaluated parameter vector was the best estimation of parameters achievable with the available observations.

4.5.2.3 Results

The Figure 4.12 shows the best estimation of the hydraulic conductivity field of the local model (the Boretto well field) and the Table 4.15 reports the mean values of all the hydraulic parameters estimated.

The estimated hydraulic conductivity field is consistent with the values calculated by means of the traditional aquifer tests; the variations are of the same order of magnitude and this fact is consistent with the alluvial nature of the aquifer. The estimated constant value of the specific storage of the local model is also very close to the values estimated by means of the traditional aquifer test. The Bayesian Geostatistical approach applied to the hydraulic tomography is doubtless a good tool to reveal the spatial distribution of the aquifer hydraulic parameters and it produces more and easily to interpret information than traditional aquifer tests.

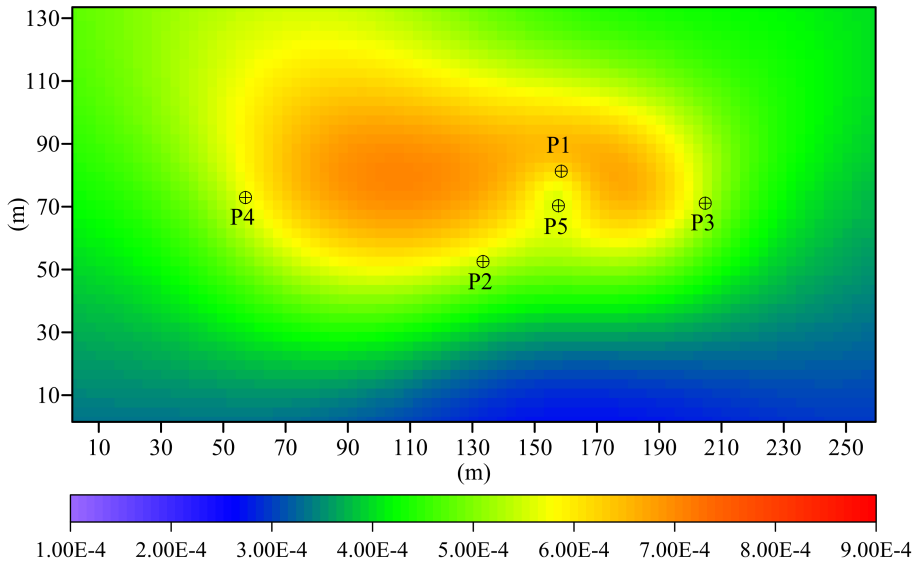

Figure 4.12: Estimated hydraulic conductivity field of the local model (Boretto well field).

Table 4.15: Estimated mean values of the Boretto well field hydraulic parameters.

Regional model		Local model	
K $(ms^{-1} \cdot 10^{-4})$	S_s $(m^{-1} \cdot 10^{-5})$	K $(ms^{-1} \cdot 10^{-4})$	S_s $(m^{-1} \cdot 10^{-5})$
13.38	8.79	4.51	5.18

4.6 Conclusions

In this chapter, traditional aquifer tests and a Bayesian Geostatistical method for hydraulic tomography are applied to the data collected during a field work in the well field of the Boretto Research Site. The closeness of the well field to the Po River causes not negligible variations of the field groundwater level during the pumping tests that require to be considered during the analyses. The results of the traditional aquifer tests are of questionable interpretation and even if they can highlight the presence of heterogeneities, it is difficult to locate them. These

analyses, easily to be implemented, are obviously useful to have an idea of the order of magnitude of the aquifer hydraulic parameters.

More information about the parameters are revealed analyzing the data collected by means of sequential aquifer tests and the use of a Bayesian Geostatistical approach, applied to hydraulic tomography, is definitely a good way to estimate the spatial distribution of the aquifer hydraulic parameters.

Chapter 5

Implementation of Bayesian Geostatistical Inverse Method in PEST

5.1 Introduction

As explained in the previous sections, the Bayesian Quasi-Linear Geostatistical method (Kitanidis, 1995) is an inversion technique able to provide the most likely and minimum variance estimation of parameters from collected data and constrained by the prior information of the spatial structure of the parameters themselves. To date, all applications of this method have been performed by means of tools that each researcher has developed for personal use and the methodology has been tested by very few works on real problems. The spread of the method could be possible if the modelers have available a good and user friendly tool that incorporates the geostatistical inverse approach.

For this reason the USGS (United States Geological Survey) is sponsoring a project for incorporating the Bayesian approach into the industry standard software package PEST (Doherty, 2008b) in the form of a module. In this way,

modelers that are already familiar with PEST can choose the optional geostatistical inversion with just few variations.

The input data needed for the Bayesian module in PEST are based on the JUPITER API framework. JUPITER API (the Joint Universal Parameter IdenTification and Evaluation of Reliability Application Programming Interface) provides resources for computer programs designed to analyze process models. For more information about JUPITER API see Banta et al. (2006).

In the following, the slight differences between the Kitanidis (1995) method and the one implemented here in PEST are described and the programming tools used in the implementation are described as well. A brief overview of the current PEST capabilities is also reported.

5.2 PEST Overview

PEST is the acronym for Parameter ESTimation; it is a free software developed by Dr. John Doehrty that can be used to support data interpretation, model calibration and predictive analysis with any pre-exisisting model. This feature of PEST (the model independence) is very powerful; the model does not need to be adapted to PEST. PEST is able to write the model input files, to read the model output files and to run the model without the necessity to have the source code.

PEST is based on the Gauss-Marquardt-Levenberg method that is a robust and powerful non-linear estimation technique developed for and already tested on many complex environmental models. The parameters are adjusted during the estimation process until the discrepancies between selected model outputs and a complementary set of observed values is reduced to a minimum in the weighted least squares sense.

PEST needs three types of files in input:

- template files, that provide the instructions needed to PEST to write the input model files before each model run;
- instruction files, in which information needed to PEST to read model output

file after each run is provided;

- control file, that contains the names of all template and instruction files and input and output model files to which they pertain; observation values, initial value of the parameters and all the other PEST variables that control the implementation of the Gauss-Marquardt-Levenberg method.

More information about the current version of PEST, the estimation algorithm and implementation can be found in Doherty (2008b,a).

5.3 Bayesian Module in PEST

Implementation of the Bayesian Quasi-Linear Geostatistical method as a PEST module, is based on a slight different cokriging system respect to the one described in Subsection 3.3.2. In Equation 3.26, indeed, it has been assumed that the drift coefficients $\boldsymbol{\beta}$ are a priori unknown. Sometime, prior, even if diffuse, information about the mean is available and can be incorporated into the method obtaining in addition an improvement of the numerical stability of the inverse process.

Assuming that the drift coefficient $\boldsymbol{\beta}$ follow a multi-Gaussian distribution with mean $\bar{\boldsymbol{\beta}}$ and covariance matrix $\mathbf{Q}_{\beta\beta}$, the prior probability density function of $\boldsymbol{\beta}$, using the symbols already defined in Chapter 3, becomes:

$$p(\boldsymbol{\beta}) = \frac{1}{\sqrt{(2\pi)^p \det\left(\mathbf{Q}_{\beta\beta}\right)}} \exp\left[-\frac{1}{2}\left(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{Q}_{\beta\beta}^{-1}\left(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}}\right)\right] \tag{5.1}$$

Taking into account the prior information on $\boldsymbol{\beta}$, the prior probability density function for $\mathbf{s}$ becomes (Nowak and Cirpka, 2004):

$$p(\mathbf{s}) \propto \exp\left[-\frac{1}{2}\left(\mathbf{s}-\mathbf{X}\bar{\boldsymbol{\beta}}\right)^{\mathrm{T}} \mathbf{G}_{\mathrm{ss}}^{-1}\left(\mathbf{s}-\mathbf{X}\bar{\boldsymbol{\beta}}\right)\right] \tag{5.2}$$

where $\mathbf{G}_{\mathrm{ss}} = \mathbf{Q}_{\mathrm{ss}} + \mathbf{X}\mathbf{Q}_{\beta\beta}\mathbf{X}^{\mathbf{T}}$ is the prior covariance and $\mathbf{X}\bar{\boldsymbol{\beta}}$ is the prior mean. The posterior density probability function becomes:

$$p(\mathbf{s}|\mathbf{y}) \propto \exp\Big[-\left(\mathbf{s}-\mathbf{X}\bar{\boldsymbol{\beta}}\right)^{\mathrm{T}}\mathbf{G}_{\mathrm{ss}}^{-1}\left(\mathbf{s}-\mathbf{X}\bar{\boldsymbol{\beta}}\right)-\left(\mathbf{y}-\mathbf{H}\mathbf{s}\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{H}\mathbf{s}\right)\Big] \quad (5.3)$$

and the best estimate $\hat{\mathbf{s}}$ and the posterior mean $\hat{\boldsymbol{\beta}}$ values are obtained maximizing the 5.3 or minimizing its negative logarithm by means of solving the system:

$$\begin{bmatrix} \mathbf{H}\mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}}+\mathbf{R} & \mathbf{H}\mathbf{X} \\ \mathbf{X}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} & -\mathbf{Q}_{\beta\beta}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi} \\ \hat{\boldsymbol{\beta}} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ -\mathbf{Q}_{\beta\beta}^{-1}\bar{\boldsymbol{\beta}} \end{bmatrix} \quad (5.4)$$

where $\boldsymbol{\xi} = \left(\mathbf{H}\mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}}+\mathbf{R}\right)^{-1}\left(\mathbf{y}-\mathbf{H}\mathbf{X}\hat{\boldsymbol{\beta}}\right)$ and the best estimate can be calculate by means of $\hat{\mathbf{s}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}}\boldsymbol{\xi}$. The equation system 5.4 for the special case of $\mathbf{Q}_{\beta\beta}^{-1} = 0$ (fully unknown mean) give the classic ordinary cokriging system 3.26 already introduced in Subsection 3.3.2.

In the Quasi-Linear approach, it is possible to monitor the progress of the optimization calculating at each iteration the objective function that must be minimized:

$$\bar{\mathcal{L}} \propto \left(\mathbf{s}-\mathbf{X}\boldsymbol{\beta}\right)^{\mathrm{T}}\mathbf{G}_{\mathrm{ss}}^{-1}\left(\mathbf{s}-\mathbf{X}\boldsymbol{\beta}\right)+\left(\mathbf{y}-\mathbf{H}\mathbf{s}\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{H}\mathbf{s}\right) \quad (5.5)$$

The calculation of the objective function of the Equation 5.5 requires the computation of the inverse of the matrix $\mathbf{G}_{\mathrm{ss}} = \mathbf{Q}_{\mathrm{ss}} + \mathbf{X}\mathbf{Q}_{\beta\beta}\mathbf{X}^{\mathbf{T}}$ $[n_{par} \times n_{par}]$ that, in case the number of parameters is high, requires a computational effort not negligible.

A computationally most efficient form of the objective function is derived in (Nowak and Cirpka, 2004) where the 5.5 is rearranged and assumes the form:

$$\bar{\mathcal{L}}' \propto \boldsymbol{\xi}^{\mathrm{T}}\left(\mathbf{G}_{\mathrm{yy}}\right)\boldsymbol{\xi} \quad (5.6)$$

where $\mathbf{G}_{\mathrm{yy}} = \mathbf{H}\mathbf{Q}_{\mathrm{ss}}\mathbf{H}^{\mathrm{T}} + \mathbf{R} + \mathbf{H}\mathbf{X}\mathbf{Q}_{\beta\beta}\mathbf{X}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}$; that is the objective function computed into the Bayesian PEST module.

5.3.1 Implementation

The current version of PEST is written in FORTRAN-90 (Chapman, 2008). The new Bayesian module is implemented in the same language to allow a complete compatibility with the current one. A new control file that contains the control variables for the inverse algorithm, the prior information, the observation values, the parameters (location, initial values and at which $\boldsymbol{\beta}$ value are associated) and all the other PEST variables needed in the application of the Bayesian method must be provided. The control file follows a subset of the JUPITER input file protocol (Banta et al., 2006); an example of the control file is showed in the Appendix F. The Bayesian module is split in sub-modules, each one developed for a specific purpose. In this work only the implementation in PEST of the kernel of the Bayesian approach is described, omitting all the input/output modules, initialization modules and error modules that have not developed by the Writer. The source code of these parts of the Bayesian PEST module is reported in Appendix F. Particular attention has been given in the storage of the parameters covariance matrix $\mathbf{Q}_{ss}$.

5.3.1.1 The Parameters Covariance Matrix $\mathbf{Q}_{ss}$

The models of covariance functions available in the Bayesian PEST are the nugget-effect model, the linear model (as limit case of the exponential as described in Subsection 3.3.2.4) and the exponential model. The three models are function of the structural parameters (one value for nugget and linear and two for the exponential) and both linear and exponential model require the separation distances between all the parameters locations that are associated with the same mean value (the i-th value of the $\boldsymbol{\beta}$ vector). The separation distances, even if the structural parameters must be estimated, don't change during the entire process and are evaluated and stored in the matrix called $\mathbf{Q}_0$ $[n_{par} \times n_{par}]$ once at the beginning of the inverse procedure. In case of nugget model the corresponding value of $\mathbf{Q}_0$ is simply set to 1. To express no correlation between parameters, the corresponding $\mathbf{Q}_0$ values are instead set to 0. The parameters covariance

matrix $\mathbf{Q}_{\text{ss}}$ $[n_{par} \times n_{par}]$ is then function of $\mathbf{Q}_0$ and the structural parameters $\boldsymbol{\theta}$ $[p \times (1\,or\,2)]$ and is evaluated every time the $\boldsymbol{\theta}$ changes.

Parameters lumped into zones of piecewise homogeneity can also be considered using the nugget model framework but in this case the structural parameter represents the variance of the unknown parameters. Actually the variance must be zero (each lumped parameter is equal to its mean) but a zero value renders the covariance matrix singular and the inverse problem ill-conditioned, so it is suggested to use a small value but not null; it could be equal to 10^{-4} to reach a compromise.

In the implementation of the Bayesian PEST two different ways for storing the covariance matrix are available: the Full form and the Compressed form, the differences are discussed below.

Full Covariance Matrix. In the full storage form of the covariance matrix, $\mathbf{Q}_0$ (and then the corresponding $\mathbf{Q}_{\text{ss}}$) is a single matrix that contains the covariances for all the parameters. To express that parameters associated with different means (different $\boldsymbol{\beta}$ value) have no correlation, the $\mathbf{Q}_0$ matrix is filled with zeros in the positions that represent the covariance between these uncorrelated elements. The order by which the parameters are indicated into the PEST control file is not important and it is the software that looks for which parameters are correlated each other based on the $\boldsymbol{\beta}$ associations supplied by the modeler. Based always on the $\boldsymbol{\beta}$ associations are the choice of the models of the covariance function and consequently the calculation of the $\mathbf{Q}_{\text{ss}}$ matrix and the correct shape of the base functions matrix $\mathbf{X}$. The one described here is the more general form of covariance matrix.

Compressed Covariance Matrix. The covariance matrix, as described above, in case of parameters of different type uncorrelated each other or parameters of the same type split up in multiple zones always uncorrelated, contains a series of zeros that must be stored increasing the required computer memory. If the parameters are ordered based on the $\boldsymbol{\beta}$ associations (and the parameters are

supposed uncorrelated for different $\boldsymbol{\beta}$), the covariance matrix then assume the form:

$$\mathbf{Q}_{\mathrm{ss}} = \begin{bmatrix} \mathbf{Q}_{\mathrm{ss}}^1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{Q}_{\mathrm{ss}}^2 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \mathbf{Q}_{\mathrm{ss}}^i & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \mathbf{Q}_{\mathrm{ss}}^p \end{bmatrix} \tag{5.7}$$

in which the matrix $\mathbf{Q}_{\mathrm{ss}}$ is now a block diagonal matrix and can be partitioned in p blocks (sub-matrices), where p is the maximum dimension of the vector $\boldsymbol{\beta}$ (the number of different means). The same is valid for the $\mathbf{Q}_0$ matrix.

If the Bayesian PEST user selects the compressed storage, an option in the control file, p matrices, one for each $\boldsymbol{\beta}$ value, are stored allowing to neglect all the zero values that indicate uncorrelation between parameters that are associated with different $\boldsymbol{\beta}$ values, and to reduce the otherwise required memory. Then the products between the covariance matrix and other matrices are performed selecting the correct covariance submatrices and the correct corresponding portion of the matrix involved in the operation. The only restriction asked to the user is to supply the parameters in the control file in blocks for each $\boldsymbol{\beta}$ association rather than random order.

To better simulate the area of interest, often the number of parameters greatly exceeds the number of available measurements and a fine discretization of the parameters themselves is required. The dimensions of the covariance matrix, that directly depends on the number of unknowns, become high and problems in its storage can be encountered as well as memory overflow during the multiplications that involve the covariance matrix itself.

However, in most common cases (e.g. a finite difference scheme), the computational grid is regular; this means, in the most general point of view that considering a three-dimensional space of coordinates x, y, z (x and y in the horizontal plane and z along the vertical), the grid spacings, in the three directions,

Table 5.1: Characteristics of the $\mathbf{Q}_0^i$ matrix reported in Figure 5.1.

n_r	n_c	n_l	$\triangle x$ (m)	$\triangle y$ (m)	$\triangle z$ (m)
3	4	3	1.0	1.5	2.0

$\triangle x$, $\triangle y$ and $\triangle z$ are constant along the corresponding direction (but can be different each other). If this is the case the $i-$th block of the covariance matrix (in case of compressed form) assumes a particular form: it is a symmetric block Toeplitz[1] matrix ($1-$st level), in which each block is a symmetric block Toeplitz matrix ($2-$nd level) constituted by blocks of Toeplitz symmetric matrices ($3-$rd level). For a grid constituted by n_r rows (along y), n_c columns (along x) and n_l layers (along z), the block Toeplitz matrices are constituted by $[n_l \times n_l]$ blocks at the $1-$st level and $[n_r \times n_r]$ blocks at the $2-$nd level; the Toeplitz symmetric matrix at the $3-$rd level has instead dimensions $[n_c \times n_c]$. The same is valid for the $i-$th block $\mathbf{Q}_0^i$ of the $\mathbf{Q}_0$ matrix. An example of three levels block Toeplitz symmetric matrix $\mathbf{Q}_0^i$ obtained with the characteristics in Table 5.1 is reported in Figure 5.1.

Due to the properties of the symmetric Toeplitz matrices and as easily verifiable looking at the Figure 5.1, only the first column (or row) of $\mathbf{Q}_0^i$ can be stored and then all the other columns (or rows) can be constructed rearranging the elements of the first one. In this way the storage requirement for the $i-$th block of $\mathbf{Q}_0$ can be strongly reduced.

In the Bayesian PEST, the possibility to take advantage of the Toeplitz matrices properties is a further option of the compressed storage option. In case the user chooses this option (available only if the parameters are distributed on a regular grid), the list of parameters must be provided in order, row by row, starting from a corner of the grid and consecutively for each layer. Moreover,

[1] A T $[n \times n]$ symmetric Toeplitz matrix has the form:

$$\mathrm{T} = \begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_{n-1} \\ t_1 & t_0 & t_1 & \dots & t_{n-2} \\ t_2 & t_1 & t_0 & \dots & t_{n-3} \\ \dots & \dots & \dots & \dots & \dots \\ t_{n-1} & t_{n-2} & t_{n-3} & \dots & t_0 \end{bmatrix}$$

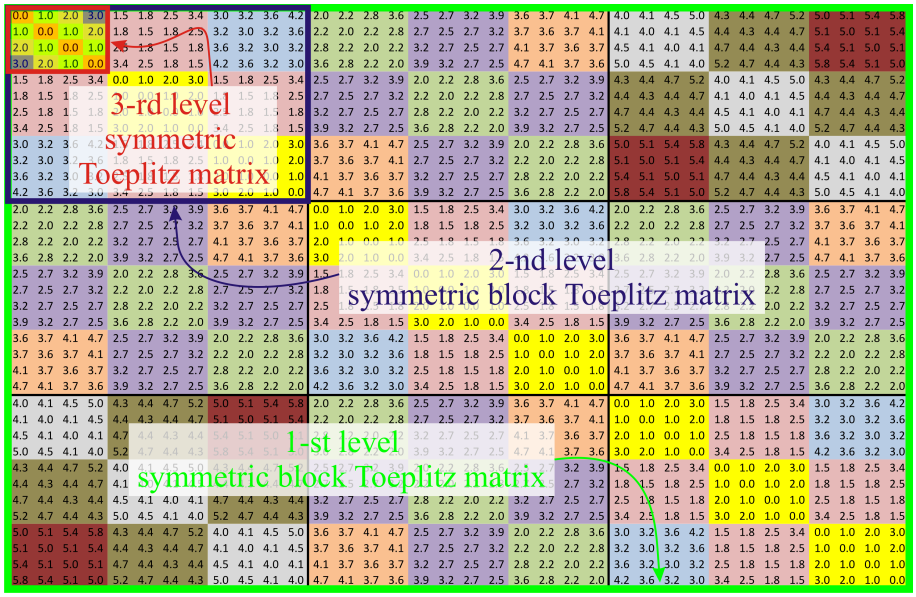

Figure 5.1: Example of three levels block Toeplitz symmetric matrix.

the number of rows, columns and layer must be provided by the user. The $i-$th block $\mathbf{Q}_0^i$ is represented by a vector of distances between the first and all the others parameters and accordingly $\mathbf{Q}_{\rm ss}^i$ is a vector of covariances calculated on the basis of the covariance function model associated with the $i-$th $\boldsymbol{\beta}$. Then, the products that involve the block of the covariance matrix that is stored in vector form are performed iteratively constructing one column of the matrix per time and multiplying it by the other matrix involved in the operation. The algorithm written to rearrange the vector of covariances to obtain all the columns of the corresponding covariance matrix is reported in Appendix F in form of FORTRAN code. The use of the Toeplitz option for the $i-$th block of the covariance matrix, do not imply that all the blocks must be stored in Toeplitz form (vector form) but can coexist the storage of other blocks entirely (matrix form).

5.3.1.2 Operations with Matrices and Vectors

Implementation of the Bayesian Geostatistical inverse method in PEST, requires a set of matrix and vector operations like: matrix-matrix products, matrix-

vector products, calculation of inverse and transpose of matrices. Some of these operations can be performed by means of the intrinsic functions available in FORTRAN-90 (Chapman, 2008) but often these function are not very efficient. In this work most of the matrices and vectors operations are performed using the BLAS (Basic Linear Algebra Subprograms) routines (Blas, 1997) and the LAPACK (Linear Algebra PACKage) routines (Anderson et al., 1999) (both are written in FORTRAN-77 (Ellis, 1990)).

The BLAS provide standard building blocks for performing basic vector and matrix operations. The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. The BLAS are efficient and optimized for fast calculations.

LAPACK routines are written so that as much as possible of the computation is performed by calls to the BLAS. The routines are designed for efficiently solving the most commonly occurring problems in numerical linear algebra.

5.3.1.3 Solution of the Cokriging System

The cokriging system of Equation 5.4 can be rewritten in the general form:

$$\mathbf{A}\,[n_{obs}+p \times n_{obs}+p] \cdot \mathbf{x}\,[n_{obs}+p \times 1] = \mathbf{B}\,[n_{obs}+p \times 1] \tag{5.8}$$

in which $\mathbf{A}$ is a symmetric matrix. The subroutine of LAPACK *"DSYSV"* (Anderson et al., 1999) solves real symmetric linear systems (in the form $\mathbf{A} \cdot \mathbf{x} = \mathbf{B}$) using the diagonal pivoting method with the partial pivoting strategy of Bunch and Kaufman (1997) and is used in the Bayesian PEST code. This subroutines makes only use of the upper (or lower) triangular part of $\mathbf{A}$, instead of the whole matrix and therefore computational time can be saved. The solution method is stable and its stability has been proved in Higham (1997).

5.4 Conclusions

Implementation of the Bayesian Quasi-Linear Geostatistical inverse method into the industry standard software package PEST is doubtless a good way to spread the method to the modelers community. The success of these highly parametrized methods is also sustained by the increasing performance of computers. At the present the Bayes module is not complete; the kernel of the geostatistical method is written and almost tested. Algorithms to increase the stability of the method (like the Levenberg-Marquardt (Nowak and Cirpka, 2004) for strong non-linearities or high variability of the parameter field) are about to be incorporate in the module. Also a method for the calculation of the sensitivity will be embedded in the Bayesian PEST.

Chapter 6

Conclusions

The main objective of this work is the estimation of spatial variations of the aquifer hydraulic parameters by means of information that can be obtained directly on the field. This is important for a proper management of groundwater but in particular for an accurate prediction of solute transport in aquifer.

A classic approach to collect data are the pumping tests in which a constant water rate is extracted from one well and the caused drawdowns are measured in one or more observation points. If the observation points can be sequentially used to extract or inject water, different sets of independent data can be collected without the installation of additional wells; these are called hydraulic tomography tests. Traditional methods or inverse methods based on tomographic data can be used to estimate the hydraulic properties in subsurface.

As shown in Chapter 4 for an application to a real case, traditional analyses, that assume aquifer homogeneity, provide average values of the hydraulic parameters and even if they can highlight the presence of heterogeneities, it is difficult to locate them and so to reproduce a spatial distribution of the aquifer hydraulic properties. Moreover, in field applications, during a pumping test often the regional flow conditions are not constant; the drawdowns are affected from these non-stationary boundary conditions and traditional analyses cannot take account of these variations. Therefore the results of traditional analyses

are of questionable interpretation and the estimation of the aquifer hydraulic parameters depends on the particular pumping and observation position or on the adopted methodology. Anyway, traditional analyses still remain important for a first and quick estimation of the parameters and to have an idea of the order of magnitude of the parameters themselves. In addition, these results can be used as starting point for the next and more accurate analyses.

On the other hand, inverse methods based on the hydraulic tomography data are able to provide detailed information about the spatial distribution of hydraulic properties in subsurface. In this work a Bayesian Quasi-Linear Geostatistical approach is considered with the aim at inverting the direct head data collected in tomographic way; this method is able to provide the most likely and minimum variance estimation of parameters from collected data and constrained by the prior information of the spatial structure of the parameters themselves. The methodology is tested by means of different synthetic cases during transient flow conditions with constant but also with non-stationary boundary conditions with excellent results. In Chapter 4, the field application of the inverse procedure together with a model able to reproduce the regional flow highlights the power of the hydraulic tomography to identify the aquifer hydraulic parameters and gives more detailed information than the traditional analyses.

The application of the Bayesian Quasi-Linear Geostatistical approach to the hydraulic tomography together with transient flow conditions and non-stationary boundary conditions is new in the scientific literature. The bottleneck in the application of the method is the calculation of the sensitivity of observations to parameters but the increasing performance of computers and the use of the adjoint state method allow the computation of the Jacobian in an acceptable time.

The use of non-stationary boundary conditions in the methodology extends the application of the hydraulic tomography concept to field cases where very often it is unrealistic to consider the studied portion of aquifer isolated from all that surrounds it.

Up to the present, the application of Geostatistical approaches to inverse

problems is limited by the lack of tools available for the scientific and technical community and the only tests are performed by means of tools that each researcher has developed for personal use. In this work, as part of a project sponsored by the USGS (United States Geological Survey) to incorporate the Bayesian Quasi-Linear Geostatistical approach as a module of the industry standard software package PEST for the parameters estimation, the kernel of the Bayesian PEST developed by the Writer is described. Particular attention is given to the storage of the parameter covariance matrix and to matrices and vectors operations to save both memory space and computational time. The opportunity of using the Bayesian PEST with different applications concerning groundwater problems and not only, implies a diligent analysis of all the routines to avoid many use limitations assuring at the same time a correct estimation of the parameters. This module is doubtless a good way to spread the Bayesian Geostatistical inverse procedure to the modelers community.

Appendix A

Efficiency of Lumped Parameter in MODFLOW_2005

In this Appendix the large computational effort needed in the numerical codes MODFLOW_2005 and MODFLOW_2005-Adjoint using lumped parameters for the definition of variables in the Layer Property Flow package is discussed. Parameters are a useful tool to represent aquifer properties in both codes and are the only option available in the adjoint version. Substantial gains in efficiency are achieved by removing logical comparison of character strings that represent the names and types of the parameters. These controls are often invoked inside loops that pass through all model nodes, the parameters themselves and all the time steps. Substantial computational time can be saved by skipping certain character comparisons or removing them from loops. An alternative formulation already available in the current implementation of the code can also alleviate the efficiency degradation due to character comparisons in the special case of distributed parameters defined through multiplication matrices.

A.1 Introduction

Applying the geostatistical methodology above described (see Section 3.3), computational inefficiencies in MODFLOW_2005 (Harbaugh, 2005) and MODFLOW-_2005-Adjoint (Clemo, 2007) using lumped parameters for the definition of variables in the LPF package (Layer Property Flow) have been encountered. The use of parameters in the MODFLOW family of codes is discussed in Harbaugh (2005, chapter 8). Lumped parameters, as defined for MODFLOW, allow for the use of a single variable value applied to multiple model nodes through the definition of zone numbers. This is especially useful when models are lumped into zones of piecewise homogeneity, but even in distributed cases in which each model node has a distinct value (as the highly parametrized inversion code used in this dissertation), using parameter definitions enables interaction with the model using a parameter value file (Harbaugh, 2005, p. 8-23). In MODFLOW_2005-Adjoint, which is used for sensitivity matrix calculation geared toward undetermined parameter estimation problems, either lumped or distributed parameters must be used.

The parameters, in both the codes, are defined with a specific name and with a string that identifies the type (e.g. *HK* for hydraulic conductivity, *VK* for vertical hydraulic conductivity, *SS* for specific storage). Both codes are mainly written in Fortran-90 (Chapman, 2008). As the scale of the problem increases in terms of number of parameters, substantial time is expended on the logical comparison of character strings. In MODFLOW_2005, string comparisons are needed to ensure that there are no illegal duplicates in the definition of the parameters and to assure that all the nodes that require a parameter of a given type have a defined one. In MODFLOW_2005-Adjoint, logical string comparisons are also used to identify the parameter types required during the simulation within loops through all the cells nested in loops through all the time steps, if the simulation is transient. Moreover, string comparisons are used to select different cases in some subroutines and often are used with combinational logical operators (e.g. AND, OR, NOT) that require a series of comparisons at the same time. Skipping

some or many of these character comparisons greatly increases the computational efficiency.

The modifications and analysis described here pertain to the lumped parameter option because this is the mode in which the number of character strings associated with parameters reaches large numbers. In the distributed case, described for MODFLOW_2005-Adjoint, only one parameter can be named for each parameter type and the distinct values are defined through a multiplier file. In this way, the scaling issues described are alleviated. However, the distributed parameter approach is only applicable to cases in which the sensitivity to input values in each node are desired. In many cases, the most efficient practical solution for calculating sensitivities to a large number of parameters using MODFLOW_2005-Adjoint is to use the distributed parameters option even if some post-processing is required to consolidate subareas into lumped zones.

Modifications made to both MODFLOW_2005 and MODFLOW_2005-Adjoint, limited to revisions of character comparisons, and the computational efficiency gains obtained through the modifications for the lumped parameter case are discussed here. All the codes were compiled for IA-32 using the Intel Fortran Compiler 10.0 and the simulations were performed with the same machine, an Intel Core2 Duo with a CPU speed of 2.53 GHz and 4 GB of RAM running Windows Vista.

A.2 String comparisons in Fortran

Fortran allows the comparison of character strings in logical expressions. Character comparisons are based on the collating sequence of characters on the computer, i.e. the order in which they occur within a specific character set (e.g. the ASCII character set) (Chapman, 2008, p. 156-160). String comparisons begin with the first character in each string; if they are the same, then the second two are compared and so on. The comparison ends either when a difference is found or the end of the string is reached. The strings must be of the same length; to ensure this condition when comparing strings of different lengths, the shorter

string is padded with blanks. For these reasons, when using string comparisons to select cases, it is better to use strings that start with different letters. In this way, in case of different strings, the comparison ends after one operation instead of at least two. This is not the typical case when assigning parameter names, however. Typically, names have an identifier and a sequential number such as "HK0001", "HK0002", and so on. In this example, it would take six comparisons to conclude that "HK0001" is different than "HK0002" whereas if, for example, integers were used, a single comparison would reveal the difference.

A.3 MODFLOW_2005 Optimization

As already above mentioned, MODFLOW_2005 is the most recent version of the finite-difference groundwater model MODFLOW (McDonald and Harbaugh, 1988). The definition of variables in the LPF package (horizontal hydraulic conductivity, HK; vertical hydraulic conductivity, VK; specific storage, SS; and so on) using lumped parameters can be a useful tool to represent aquifer properties. However, substantial additional computational time is required at the beginning of the run to process the data. This time increases quickly with the number of the parameters. The time expended is due to two controls/checks in MODFLOW_2005 in which string comparisons are involved: the first assures that there are no illegal duplicates in the definition of the parameters; and the second assures that all the nodes of each layer that require a parameter of a given type, have a defined one (through the zone file). Searching for duplicates requires, for any new parameter read in the LPF file, the comparison of the current parameter name with all the previous parameters already defined. Checking for complete definition of parameters of a given type for one layer of cells requires a loop through parameters to find matching parameter type, a loop through clusters associated with this parameter, and a loop through zones listed for this cluster.

If MODFLOW_2005 is used as the forward model in a parameter estimation procedure, MODFLOW must be run many times as the inverse algorithm updates parameter values. In this context, model input files are generated using a

program that should not change throughout the progress of the algorithm. Using a single run, up front, to verify the parameter configuration obviates the need for subsequent runs to perform these expensive checking procedures. Making this checking optional in MODFLOW_2005 requires little modification. The idea is to have a flag in the model that enables or disables the two checks described. With these changes, substantial computational time can be saved when iteratively running MODFLOW_2005 in parameter estimation following an initial quality-assurance run with full checking.

A modification can be easily implemented in which the variable NPLPF at the head of the LPF input file could have two functions, depending on its sign. If NPLPF is positive, all checking would proceed as in place in the current version. If NPLPF is negative, the parameter integrity checking is skipped.

A series of transient runs with 60 time-steps, one confined layer, and one extraction well have been performed in order to evaluate the saved computational time. Different configurations varying the number of parameters and nodes (the number of nodes was equal to the number of parameters) have been analyzed. Each configuration has been run with and without the controls described above, and a version using the official code with distributed parameters defined through multiplication matrices (in this case just one named parameter was necessary) has also been run. Figure A.1 shows the computational times for each configuration. The lines diverge very quickly as soon as the number of parameters increases. In the extreme case of 1 000 000 nodes, the model version without checking for parameters is 7.5 times faster than with checking. Even at more modest numbers of parameters, increases in efficiency are about six-fold. For example, consider the calculation of a Jacobian matrix (required for gradient-based inverse modeling) using forward perturbations which requires as many model runs as the number of parameters. For 10 000 parameters, the run-times are 0.006, 0.025, and 0.1 minutes from most to least efficient implementations in Figure A.1. Multiplied by the number of parameters (10 000) the total Jacobian run-times are 60, 250, and 1 000 minutes, respectively. Although all individual run-times are small, the impact on scaling of the inverse problem is substantial. An alternative method to

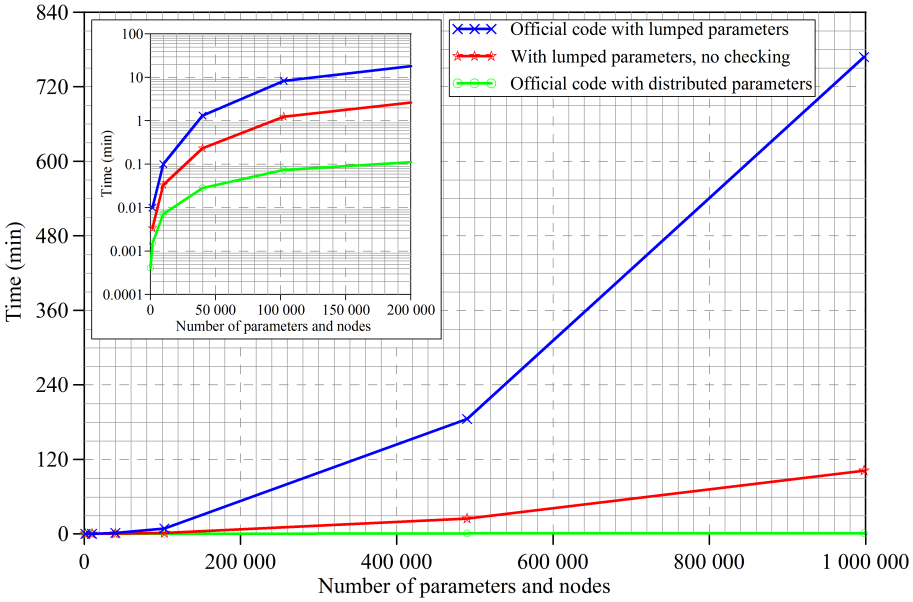

Figure A.1: MODFLOW_2005: Time versus number of parameters for the official code using lumped parameters, the code modified to skip checking using lumped parameters, and a version with distributed parameters. Results are shown in linear scale in the main plot and the early time is shown in log scale in the inset plot.

calculate the Jacobian matrix by adjoint state is discussed below, although it is not always an option. A version of the official code without the use of parameters was also run and the computational time was within a fraction of a second of the distributed parameters case so the results are not presented. The run results are identical as expected because no modifications are involved in the solution of the model equations.

A.4 MODFLOW_2005 Adjoint Optimization

As already mentioned in Section 3.3.4 MODFLOW_2005-Adjoint is able to calculate the sensitivity of each observation to each parameter creating a Jacobian matrix; the code supports different type of parameters: horizontal hydraulic conductivity *HK*, vertical hydraulic conductivity *VK*, specific storage *SS*, specific

yield *SY*, and many others (for a complete list see Clemo (2007)). Each parameter in MODFLOW_2005-Adjoint is identified by two character strings: one is the parameter name and the other is the parameter type. In addition to the use described in Section A.3 at the beginning of the simulation, the strings with the names are used to determine which parameters the sensitivity calculation must be performed for; the strings with the types are often used to make decisions, in different subroutines, on which algorithm must be run. During the adjoint process, logical comparisons of strings to identify the parameter types are often nested inside loops through all the nodes and through all the time steps. Moreover, many times, these comparisons are combined with combinational logical operators (e.g. AND, OR, NOT) that require more than one string comparison on the same program line. Another string comparison involved in the adjoint process, only in the case of horizontal hydraulic conductivity (types *HK* and *YHK*) or horizontal anisotropy (type *HANI*) parameters, is used to select two different cases in the derivatives calculation. In this case the two strings are *CC* and *CR* and, because they start with the same letter, two character comparisons are needed at each iteration. The *CC* and *CR* strings indicate whether arrays are column- or row-major.

A.4.1 Description of modifications

Two different modified versions of the MODFLOW_2005-Adjoint has been implemented. In one version, the derivatives calculation has been optimized so that the comparison between *CC* and *CR,* described above, are not required. In the original version of the code, the program invokes a subroutine in which the returned results depend on the passed argument string *CC* or *CR*. In the modified version, the subroutine call has been eliminated; the subroutine has been split into two custom loops, one for *CC* and one for *CR,* and incorporated directly into the main program. In this way, the *CC* and *CR* string comparisons are avoided. The code becomes repetitious with respect to *CC* and *CR,* but substantial computational time is saved. This modification is generally applicable

to all parameter types.

The second modified version of the code has been customized just for the specific case of one confined layer and parameter type equal to *HK*. In the original version of MODFLOW_2005-Adjoint, logical string comparisons are used to identify the parameter types required during the simulation within loops over all the nodes. These loops are then nested in other loops over all the time steps and are often used with combinational logical operators (e.g. AND, OR, NOT). All string comparisons pertaining to parameter type have been fully removed, therefore assuming all parameters are of the type *HK*. The string comparisons for determining the parameter type involved in IF statements and with combinational logical operators have been also skipped because the parameter type was known *a priori*. These modifications are specific to the parameter types used in one given application, so extension to any other parameter type combinations would require a similar modification of the source code.

A.4.2 Test cases

In the MODFLOW_2005-Adjoint case, a series of transient runs with 60 time-steps, one confined layer and one extraction well has been performed. Different configurations varying the number of parameters and nodes (the number of nodes was equal to the number of parameters) have been analyzed. In all the cases the sensitivity has been calculated only for the hydraulic conductivity *HK*, in just one observation point and for four different times within the total simulation time. For each configuration, four different runs have been performed: the first one using the original MODFLOW_2005-Adjoint code; the second with the optimized derivative calculation so that the comparison between *CC* and *CR*, described above, is not required; the third one customizing the code just for the specific case of one confined layer and parameter type equal to *HK*; and the forth one using distributed parameters instead of lumped. The results of the simulations are depicted in Figure A.2, the lines that represent the three different code versions, using lumped parameters, diverge quickly as the number of parameters

increases. Skipping some of the string comparisons substantially reduces the simulation time and a strong optimization/customization of the code can considerably reduce the computational time. In the extreme case of 40 000 nodes, the model version without the *CC* and *CR* comparison is 3.4 times faster than the original code and the only *HK* code version is 6.3 times faster than the original one. Even at more modest numbers of parameters, increases in efficiency are substantial, improving scalability as the number of parameters increases.

Using distributed parameters and multiplication matrices greatly alleviates the efficiency degradation due to character comparisons because a very small number of parameters are defined and sensitivities are calculated with respect to multipliers. In this way, the character comparisons are minimal and a huge gain in efficiency is realized. Not all applications call for distributed parameters, but due to linearity of sensitivities, distributed sensitivity values can be added together in a post-processing step for zones that are considered lumped. A typical case where this can occur is one in which a focused region of a model is parametrized in a distributed way but is surrounded by a single homogeneous buffer zone.

A.5 Conclusions

The numerical codes MODFLOW_2005 and MODFLOW_2005-Adjoint can make use of lumped parameters for the definition of variables in the LPF package. Accompanying the conceptual benefits of enabling the application of a single variable value to multiple model nodes is a cost in overhead related to the comparison of character strings used to define the name and type of each lumped parameter. This overhead increases quickly with the number of the parameters in MODFLOW_2005 and also the number of the nodes in MODFLOW_2005-Adjoint.

The definition of variables in the LPF package using lumped parameters is a useful tool to represent aquifer properties when their numbers are limited. The scalability of this concept is hampered, however, by what rapidly becomes an enormous number of inefficient character comparisons as the number of named parameters increases. This inefficiency can be avoided through the use of dis-

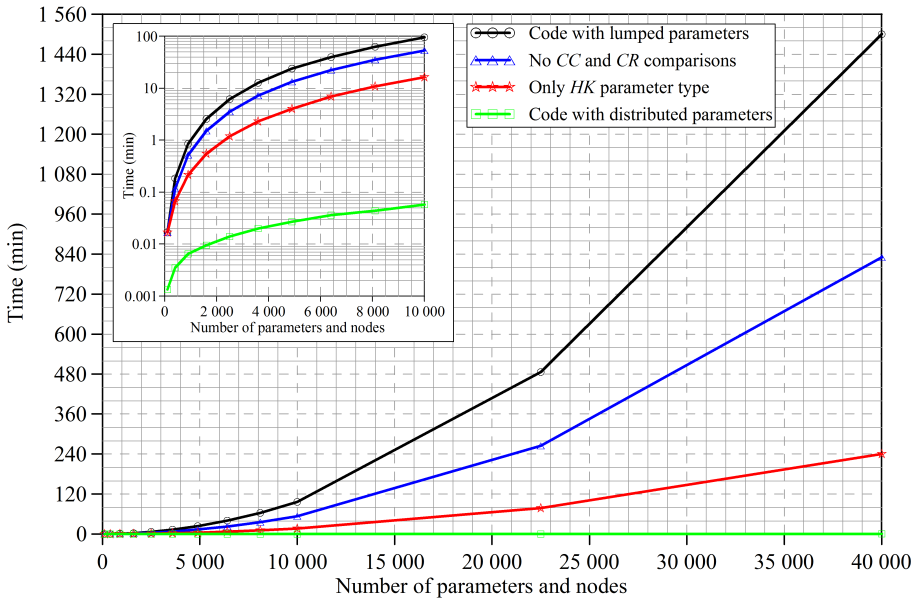

Figure A.2: MODFLOW_2005-Adjoint: Time versus number of parameters for the original code, the code modified to skip the *CC* and *CR* comparison, the only *HK* parameter type version with lumped parameters, and the code with distributed parameters. Results are shown in linear scale in the main plot and the early time is shown in log scale in the inset plot.

tributed parameters, already implemented in the existing code, in which the sensitivity is calculated for each model node rather than for named parameters. This approach works well provided that every node in the model is an estimable parameter in a parameter estimation problem. In some cases, such as a focused area of interest in which each node is an estimable parameter surrounded by a lumped buffer zone, the intermediate approach allowed by using named parameters is still useful. It is in this context that the efficiency improvements are the most useful.

Further modifications can be implemented in the definitions of the parameters in MODFLOW_2005 and MODFLOW_2005-Adjoint to avoid string comparisons, such as using integers rather than characters for the parameter names and parameter types. The use of hash tables and some efficient hash functions could also speed up the detection of duplicate records and verification of complete definition of parameters (Knuth, 2009, p. 506-549). MODFLOW_2005-Adjoint could also be parsed such that a custom subroutine for each type of parameter would be defined and subroutine selection would occur outside of loops.

Appendix B

Stratigraphic Column of the Boretto Well Field

In this Appendix, the stratigraphic column of the Boretto well field is shown. The vertical sequence of underground materials has been obtained during the well drilling in the location P1 (Figure 4.2); it starts from the ground level and goes on until a depth of 50 m.

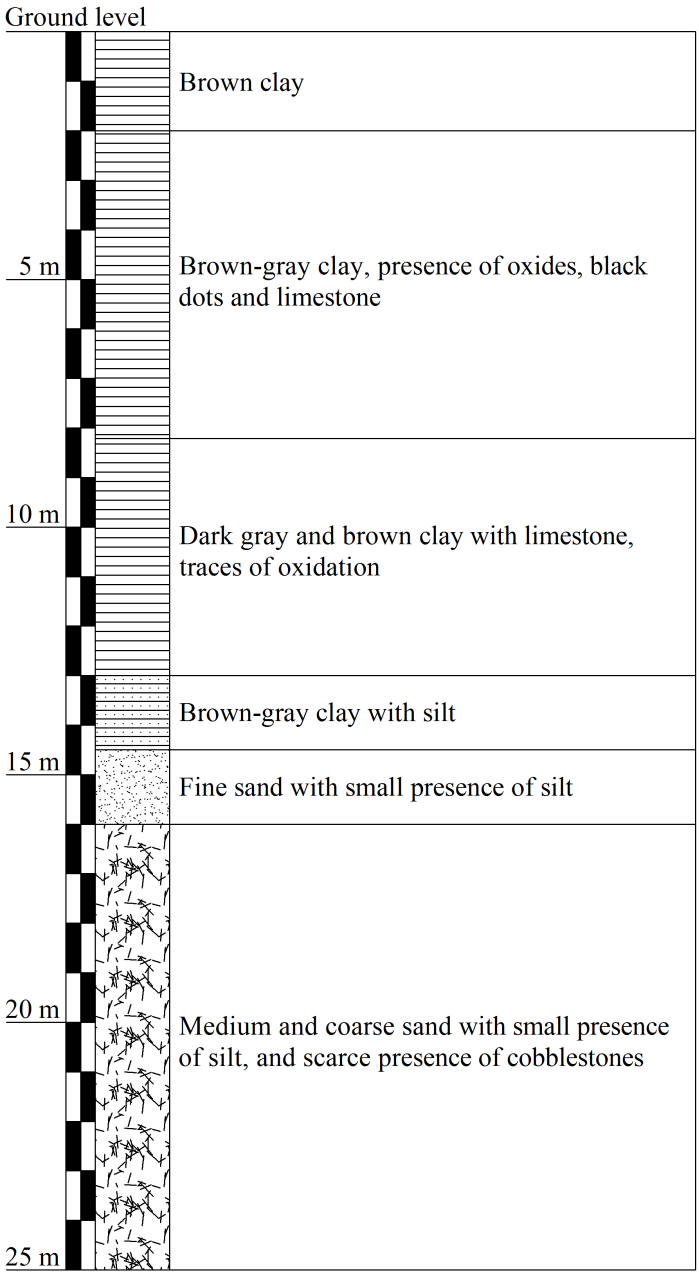

Figure B.1: Stratigraphic column of the Boretto well field: part 1.

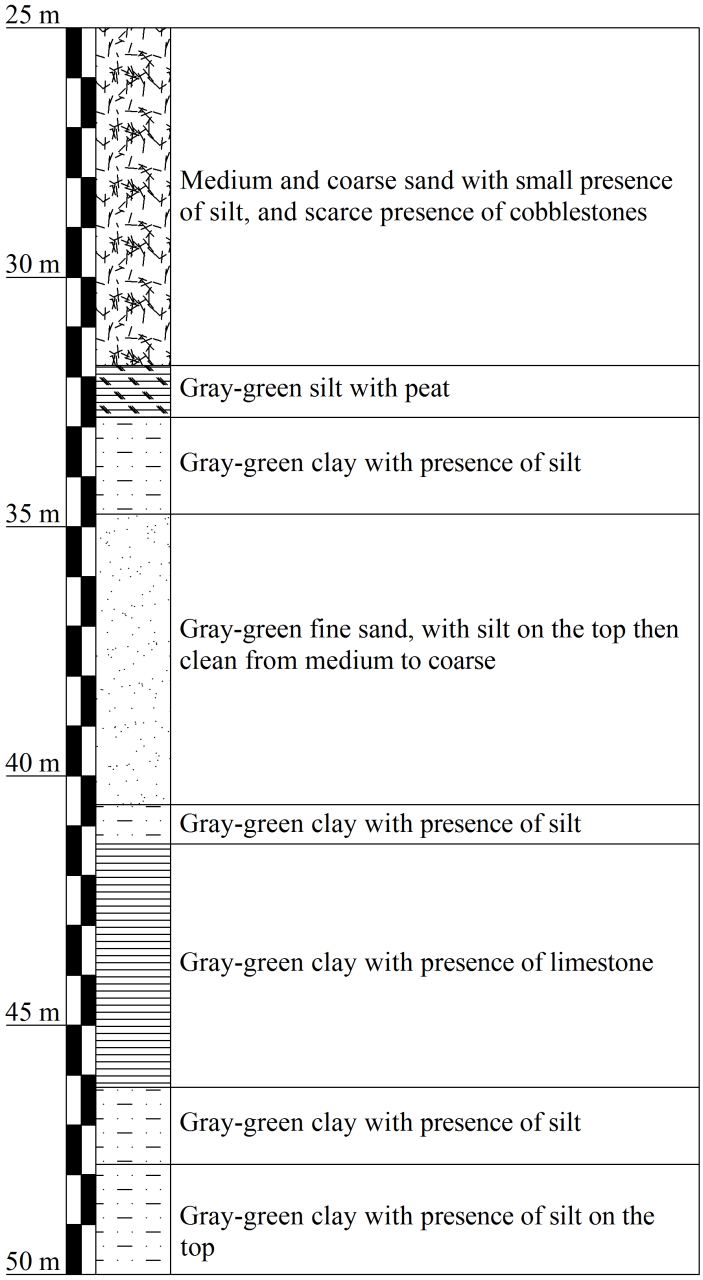

Figure B.2: Stratigraphic column of the Boretto well field: part 2.

Appendix C

Well field instruments

C.1 Pressure and temperature probes

The pressure and temperature probes are the submersible OTT Orpheus Mini; they have been designed for the reliable monitoring and storage of water level and temperature. The main application of OTT Orpheus Mini is the installation in groundwater pipes and wells. OTT Orpheus Mini is equipped with a rugged, ceramic-capacitive measuring cell and a precise temperature sensor. A data logger, which can be configured individually, stores and manages the monitored measured values in a 4 MB non-volatile memory (corresponds to approx. 500,000 measured values). The connexion for data retrieval and start-up and the power supply of the OTT Orpheus Mini are provided by the communication unit. The kevlar-reinforced pressure probe cable with pressure compensation capillary in combination with a desiccant cartridge prevents reliably measuring errors by compensating barometric pressure fluctuations. When downloading data the data logger transfers the measured values via a RS-485 connexion to the communication unit, which establishes a non-contact transmission of the data via infrared interface (IrDA) to the reading unit or laptop. Figure C.1 shows the OTT Orpheus Mini and Table C.1 reports the main characteristics of the pressure and temperature probe.

Figure C.1: OTT Orpheus Mini groundwater data logger.

Table C.1: Main characteristics of the OTT Orpheus Mini.

Measuring range	0-40 m water column
Resolution, pressure	0.01 % FS
Accuracy, pressure	±0.05 % FS
Resolution, temperature	0.1 °C
Accuracy, temperature	±0.5 °C
Interface	Infrared (IrDA)
Memory	4 MB
Number of measured values	approx. 500000
Reading interval	1 second 24 hours
Storage interval	1 second 24 hours

Figure C.2: OTT Contact Gauge.

Table C.2: Main characteristics of the OTT Contact Gauge.

Measuring range	0-50 m
Measuring accuracy	0.1 % of the measured value
Measuring tape type	2-stranded
Labelling	Meter scale: black; cm divisions and dm numbers

C.2 Contact gauge

The manual water meter available in the Boretto well field is the OTT Contact Gauge. The electric contact gauges are used for fast and accurate measurements of water level in groundwater areas. They are suitable for both control measurements as well as for continuous monitoring of pump tests. Figure C.2 shows the OTT Contact Gauge and Table C.2 reports its main characteristics.

C.3 Magnetic flow meters

The two magnetic flow meters (*Mag-Flow*) used in the Boretto well field are *Fischer & Porter* production. The electromagnetic flowmeter can be used to

Figure C.3: *Fischer & Porter* magnetic flowmeter.

Table C.3: Main characteristics of the *Fischer & Porter* magnetic flowmeters available in the Boretto well field.

DN	100 mm	50 mm
Warm-Up time	30 min	30 min
$Q_{max\,DN}$	240 m^3h^{-1}	60 m^3h^{-1}
Accuracy	$Q > 0.05\ Q_{max\,DN}$ ± 0.05 % VM	$Q > 0.05\ Q_{max\,DN}$ ± 0.05 % VM
	$Q < 0.05\ Q_{max\,DN}$ ± 0.0002 % $Q_{max\,DN}$	$Q < 0.05\ Q_{max\,DN}$ ± 0.0002 % $Q_{max\,DN}$

accurately measure the flowrate of liquids which have an electrical conductivity greater then $5\,\mu$S/cm. The flowmeters are the model COPA-XM and present a linear and accurate flowrate metering independent of flow profile. Figure C.3 shows one of the used flow meters and Table C.3 reports the main characteristics of both the *Mag-Flow*s.

Appendix D

Traditional Aquifer Tests

In this Appendix, the results of the Drawdown-Distance analyses for each pumping test performed in the Boretto Research Site are reported. The hydraulic properties were estimated using the drawdowns at different times but contemporaneously in all the observation points. About one estimation every five minutes was performed using both measured and effective drawdowns. The results are showed for the hydraulic conductivties in Figures D.1-D.5 and for the specific storage values in Figures D.6-D.10.

In Figures D.1-D.10 it is possible to see that, by means of the Drawdown-Distance analysis, the estimated hydraulic conductivity values have small vari-

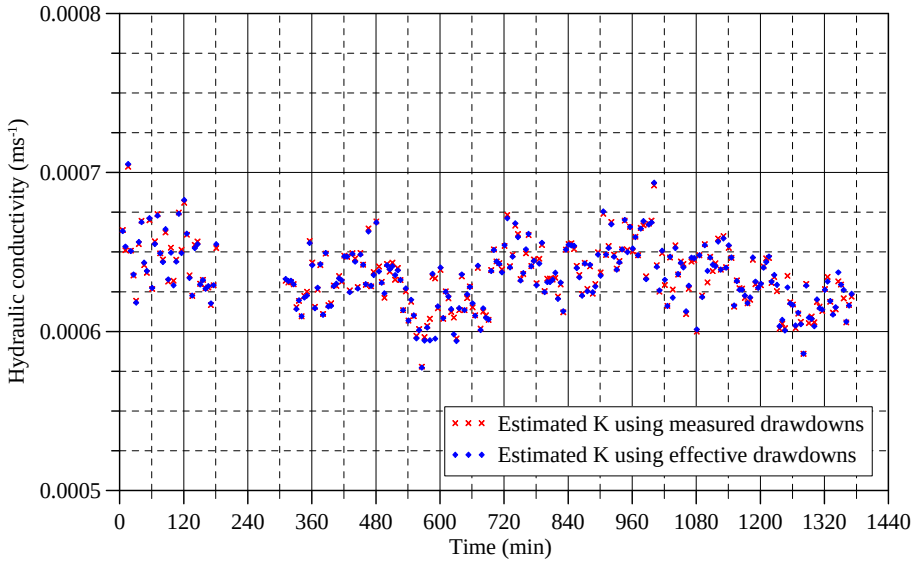

Figure D.1: Hydraulic conductivity versus time for pumping test 1 using the Drawdown-Distance analysis with measured and effective drawdowns.

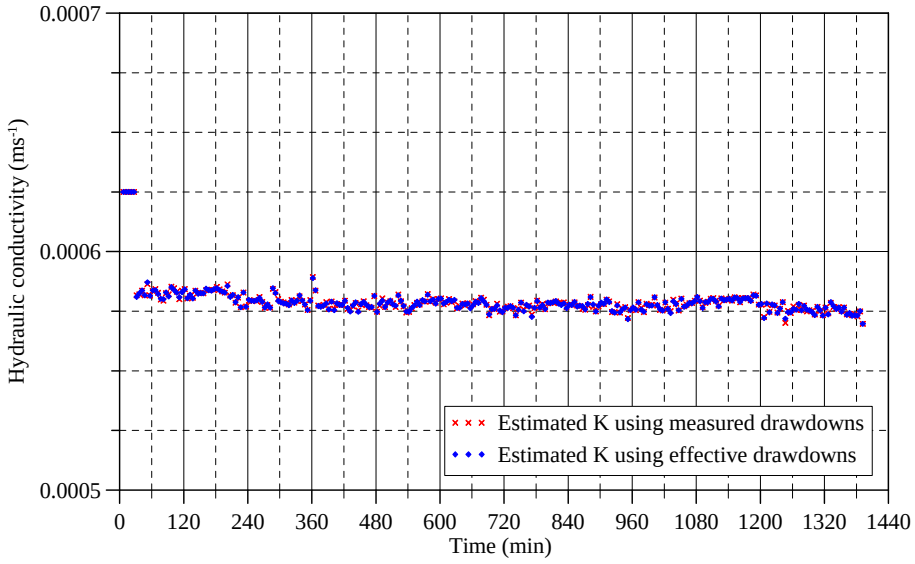

Figure D.2: Hydraulic conductivity versus time for pumping test 2 using the Drawdown-Distance analysis with measured and effective drawdowns.

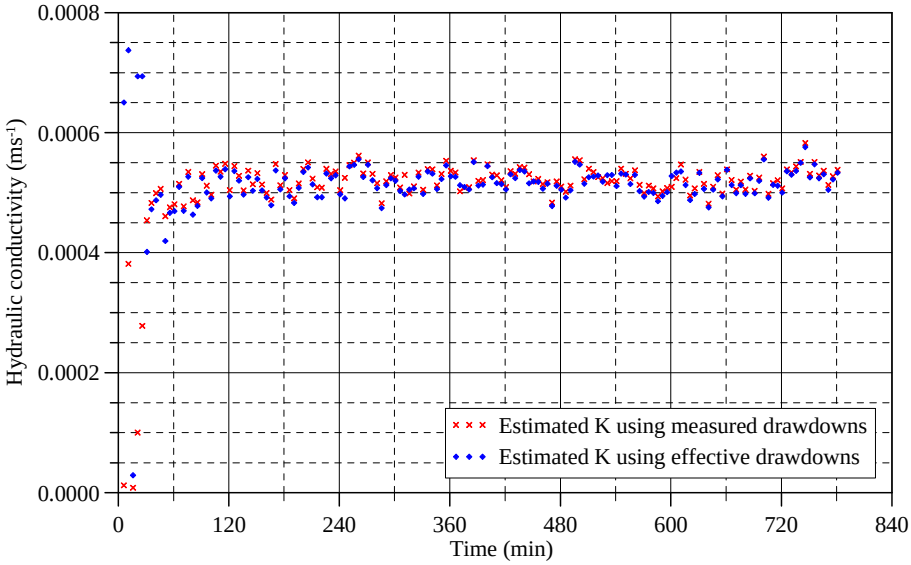

Figure D.3: Hydraulic conductivity versus time for pumping test 3 using the Drawdown-Distance analysis with measured and effective drawdowns.

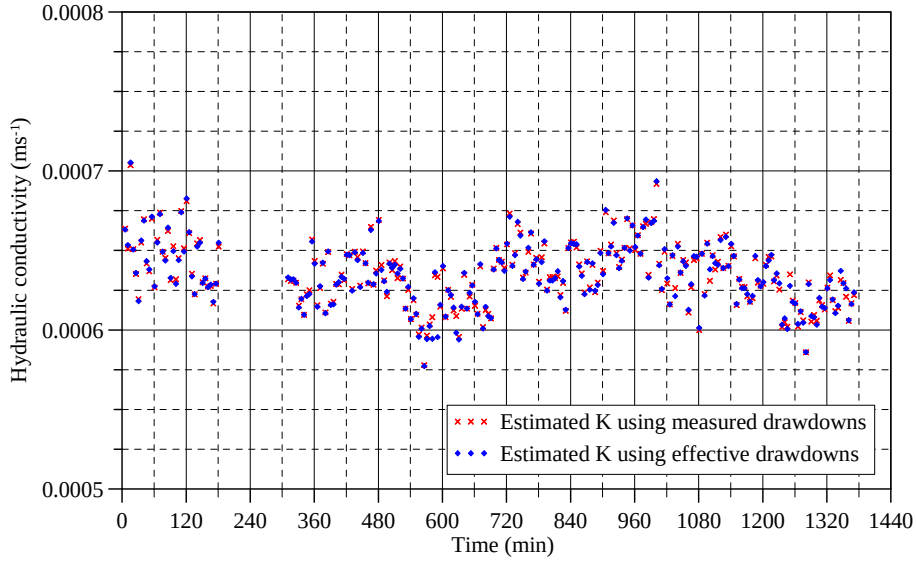

Figure D.4: Hydraulic conductivity versus time for pumping test 4 using the Drawdown-Distance analysis with measured and effective drawdowns.

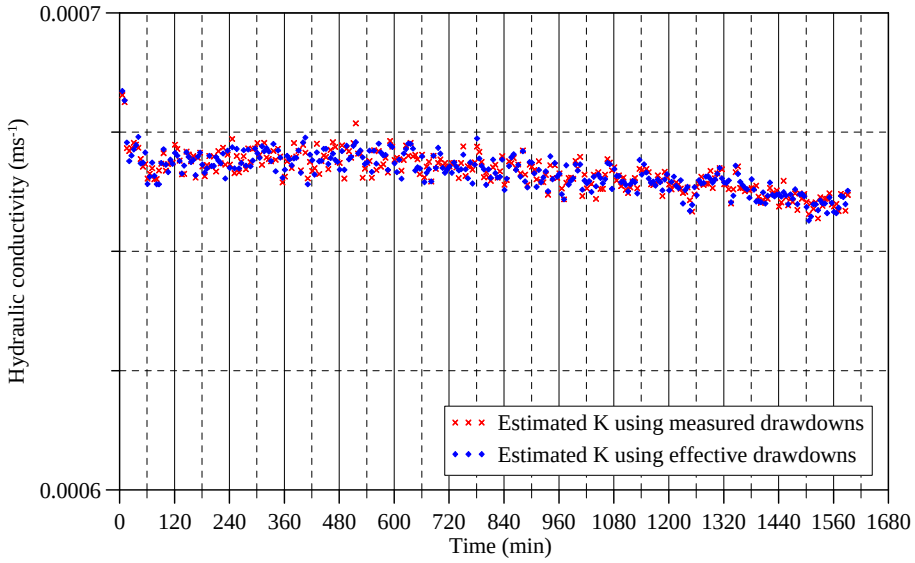

Figure D.5: Hydraulic conductivity versus time for pumping test 5 using the Drawdown-Distance analysis with measured and effective drawdowns.

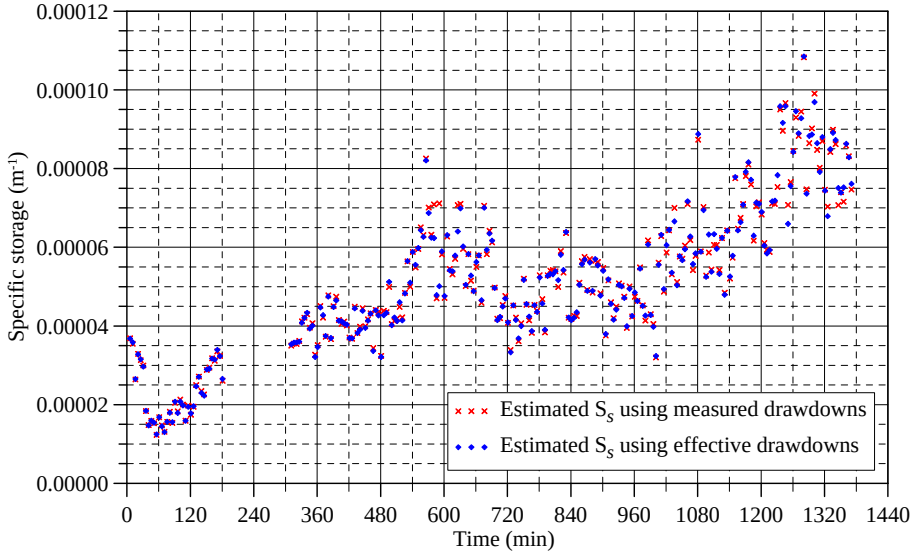

Figure D.6: Specific storage versus time for pumping test 1 using the Drawdown-Distance analysis with measured and effective drawdowns.

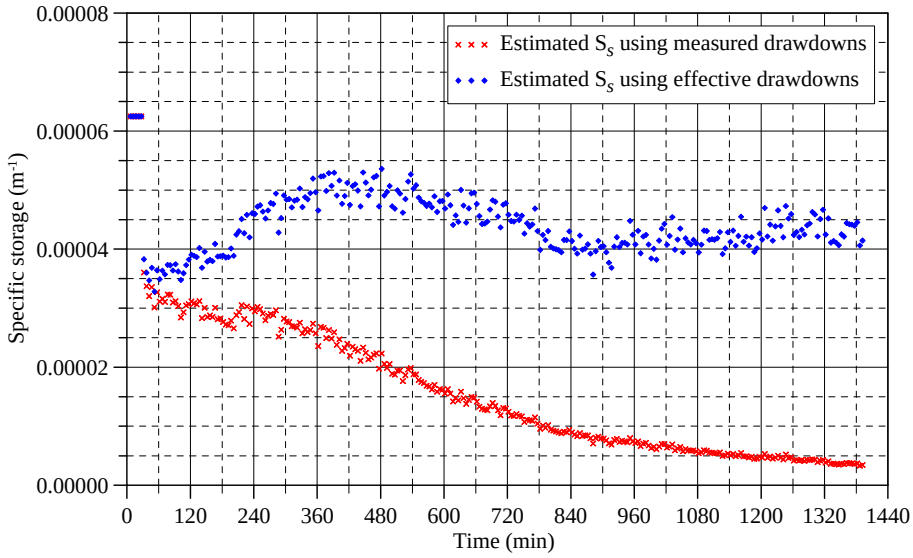

Figure D.7: Specific storage versus time for pumping test 2 using the Drawdown-Distance analysis with measured and effective drawdowns.

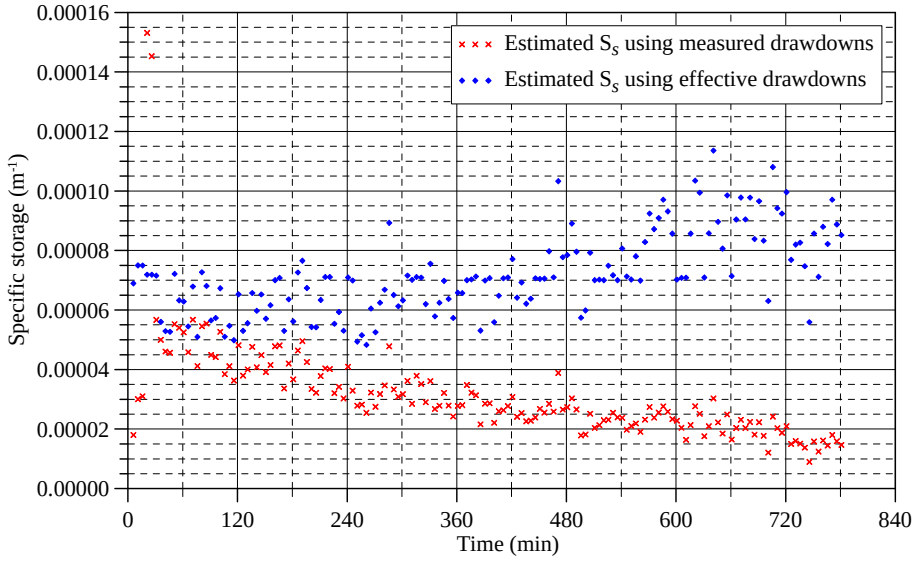

Figure D.8: Specific storage versus time for pumping test 3 using the Drawdown-Distance analysis with measured and effective drawdowns.

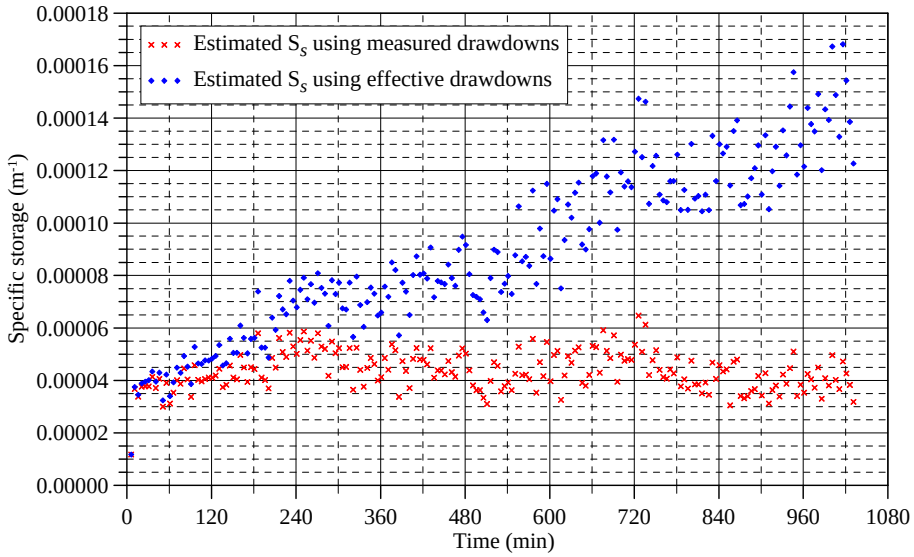

Figure D.9: Specific storage versus time for pumping test 4 using the Drawdown-Distance analysis with measured and effective drawdowns.

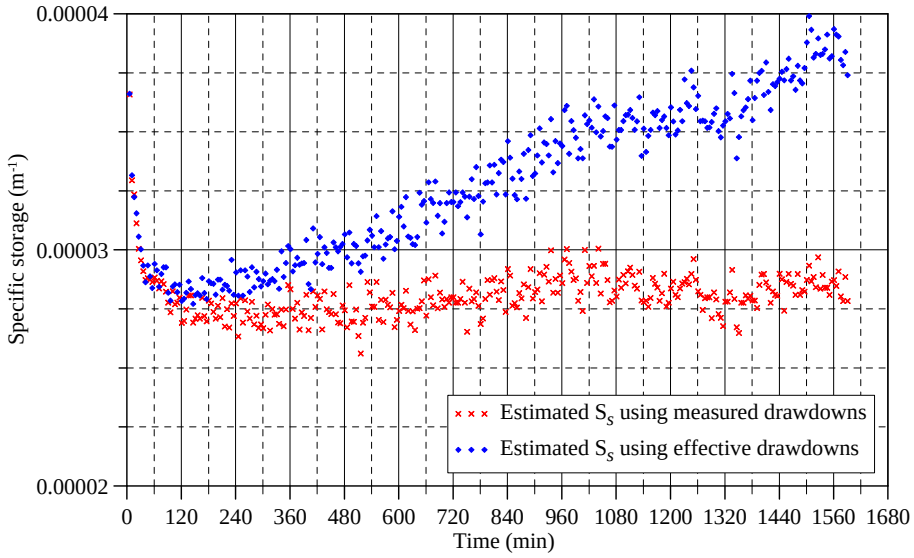

Figure D.10: Specific storage versus time for pumping test 5 using the Drawdown-Distance analysis with measured and effective drawdowns.

ations in time and they are almost the same considering measured or effective drawdowns. More variations are instead present in the estimation of the storativity values. This behavior can be explained considering the Jacob approximation of the Theis solution[1]:

$$s(r,t) = \frac{Q}{4\pi \mathrm{T}} \ln \frac{2.25\mathrm{T}t}{\mathrm{S}r^2} \tag{D.1}$$

where Q $\left[\mathrm{L}^3\mathrm{T}^{-1}\right]$ is the extracted constant flow rate, r [L] is the distance between observation and extraction well, t [T] the time and T and S are the transmissivity and the storativity respectively.

According to the Jacob solution, in a semi-log graph $(\ln(t),\, s(r,t))$ the drawdowns s are disposed on a straight line and the value of the transmissivity is proportional to the slope of the line itself while the value of the storativity is the intersection between the straight line and the x axis. The analysis of drawdowns using several observation points (at each set time) doesn't change the slope of the straight line using effective instead of measured drawdowns; it is just a translation of the reference system. This imply that the transmissivity values are almost the same instead the storativity values suffer of more variations. This approach seems able to estimate a quite correct value of transmissivity independently of the influence of the boundary conditions.

[1] According to Theis: $s(r,t) = \frac{Q}{4\pi \mathrm{T}} W(u)$ with $u = \frac{r^2 \mathrm{S}}{4\mathrm{T}t}$; the Jacob approximation is obtained for large t.

Appendix E

MATLAB functions *fminsearch*, *fmincon* and *idfilt*

In this Appendix more details about the MATLAB functions *fminsearch*, *fmincon* and *idfilt* are reported.

E.1 *fminsearch*

The MATLAB (Mathworks, 2002) algorithm *fminsearch* is used to find a minimum of a scalar function of several variables. The algorithm starts at the point x_0 (an initial estimate) and returns a value x that is a local minimizer of the function to minimize. The unconstrained non-linear optimization procedure of *fminsearch* uses the Nelder-Mead (Lagarias et al., 1998) simplex search method, a derivative-free method that does not use numerical or analytic gradients.

If n is the length of x, a simplex in n-dimensional space is characterized by the $n+1$ distinct vectors that are its vertices. In two-space, a simplex is a triangle; in three-space, it is a pyramid. At each step of the search, a new point in or

near the current simplex is generated. The function value at the new point is compared with the function values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, giving a new simplex. This step is repeated until the diameter of the simplex is less than the specified tolerance.

E.2 *fmincon*

The MATLAB (Mathworks, 2002) algorithm *fmincon* is used to find a minimum of constrained nonlinear function of several variables starting at an initial estimate. The constrained nonlinear optimization procedure of *fmincon* uses a gradient-based method that is designed to work on problems where the objective and constraint functions are both continuous and have continuous first derivatives; the Hessian matrix is needed during the optimization procedure. There are three algorithms used by *fmincon:* 1) Trust-Region-Reflective Optimization; 2) Active-Set Algorithm; 3) Interior-Point Algorithm. In this work the Trust-Region-Reflective Optimization is used; it is an algorithm based on the interior-reflective Newton method (Coleman and Li, 1994, 1996). Each iteration involves the approximate solution of a large linear system using the method of preconditioned conjugate gradients (PCG).

E.3 *idfilt*

The MATLAB (Mathworks, 2002) function *idfilt* is used to apply pass-band and other custom filters to time-domain or frequency-domain data. In this work the filter is defined as pass-band and for time domain data. The time domain filters in the pass-band case are calculated as cascaded Butterworth pass-band and stop band filters. The Butterworth filter is a type of signal processing filter designed to have a frequency response which is as flat as possible in the pass-band. For time domain data, the filtering is carried out as a CAUSAL filter. Causal filters typically introduce a phase shift in the results. For this reason, the option NONCASUAL is used in this work to have zero-phase filter.

Appendix F

Bayesian PEST Source Code and Control File

In this Appendix the source code of the subroutines of the Bayesian PEST developed by the Writer is reported; BLAS (Basic Linear Algebra Subprograms) routines and LAPACK (Linear Algebra PACKage) routines used in the implementation of the code are not reported here and can be found in Blas (1997) and Anderson et al. (1999). An example of the Bayesian PEST control file that follows a subset of the JUPITER input file protocol (Banta et al., 2006) is shown at the end.

Subroutine to set up the Q_0 matrix, X matrix and R matrix

subroutine bxq_make_X0_Q0_R0_InvQbb(d_PAR,cv_S,d_S, &
& cv_PAR,d_XQR,cv_A,d_OBS,nobs,d_PM,Q0_All,cv_PM)
!– Subroutine to create covariance matrix Q0, X matrix, R0 matrix, and if necessary Qbb^-1 and beta0*Qbb^-1 !
!– The covariance matrix can be: (The flag (cv_A%Q_compression_flag) control this)
! - the full Q0 matrix ——————–> (cv_A%Q_compression_flag) = 0

```
! - compressed form (in block for each beta) —> (cv_A%Q_compression_flag)
= 1
!– In case of compressed form we have 3 different types:
!(The flag Q0_All(x)%Toep_flag control this)
! - full matrix for the specified beta ————————> (Q0_All(x)%Toep_flag) = 0
! - just a vector for the specified beta ————————> (Q0_All(x)%Toep_flag) = 1
! - 1 value in case of nugget for the specified beta ——-> (either if the Toep_flag
is [1] or [0])
use bayes_pest_control
use jupiter_input_data_support
use utilities implicit none
! declarations
type(d_param),intent(in) :: d_PAR
type(cv_struct),intent(in) :: cv_S
type(d_struct), intent(in) :: d_S
type(cv_algorithmic), intent(in) :: cv_A
type(cv_param), intent(in) :: cv_PAR
type (cv_prior_mean), intent(in) :: cv_PM
type(Q0_compr), intent(inout) :: Q0_All(:)
type(d_observ), intent(in) :: d_OBS
type(kernel_XQR),intent(inout) :: d_XQR
type(d_prior_mean), intent(inout) :: d_PM
integer, intent(in) :: nobs
integer, pointer :: cnp(:)
integer :: i,j,k,p      ! local counters
double precision :: ltmp     ! Temporary value of Lmax
character (len=ERRORWIDTH) :: retmsg
! Allocate memory for X and initialize to 0
allocate(d_XQR%X(cv_PAR%npar,cv_PAR%p))
d_XQR%X = 0. ! matrix
```

```
select case (cv_A%Q_compression_flag) !Select the compressed or not form of Q0 matrix
case(0) !Calculate full Q0 matrix
!*****************************************************************
!********************** Make full Q0 and X matrix ******************
!*****************************************************************
!Allocate memory for cnp
!**** Cnp is a counter to verify that in case of not nugget
! variogram, the number of parameters is gt 1
allocate(cnp(cv_PAR%p)) cnp=0 ! 2 or more means no problem, 1 found only one parameter, 0 no parameters found
select case (cv_A%store_Q)
case (.TRUE.) ! Allocate memory for Q and initialize to 0
allocate(d_XQR%Q0(cv_PAR%npar,cv_PAR%npar)) d_XQR%Q0=0. ! matrix
!*** Start to fill the Q0 matrix based on the variogram type and to fill X matrix with 1
!*** to associate the correct beta to each parameter with a loop over all the parameters
!*** In the same loop are 3 control.
!*** 1. To check that each beta has at least one parameter defined
!*** 2. To verify that in case of not nugget variogram, the number of parameters is gt 1
!*** 3. To avoid that the same beta corresponds to parameters of different type.
do i=1, cv_PAR%npar !Loop over all the parameters
if (cv_S%var_type(d_PAR%BetaAssoc(i))==0) then ! put 1 on diagonal for nugget
d_XQR%Q0(i,i)=1.
cnp(d_PAR%BetaAssoc(i))= 2
do j=i+1, cv_PAR%npar
!*** This control avoid that the same
```

```
if (d_PAR%BetaAssoc(i).eq.d_PAR%BetaAssoc(j)) then !*** This control avoid that the same
!*** beta corresponds to parameters of different type.
if (d_PAR%Group_type(i).ne.d_PAR%Group_type(j)) then
write(retmsg,30) d_PAR%BetaAssoc(i), i, j
30 format('Error: Beta association value ',i6, ' corresponds to different parameter types.'
& ' Check rows ',i6,' and ',i6,' of the parameter table. Excecution stopped.')
call utl_writmess(6,retmsg)
stop
endif
endif
enddo !Finished control
else ! all other variograms require distances
cnp(d_PAR%BetaAssoc(i))= cnp(d_PAR%BetaAssoc(i))+1
do j=i+1, cv_PAR%npar
if (d_PAR%BetaAssoc(i).eq.d_PAR%BetaAssoc(j)) then !Search in the parameters list the associated parameters and calculate the distances
if (d_PAR%Group_type(i).ne.d_PAR%Group_type(j)) then !*** This control avoid that the same beta corresponds
!*** to parameters of different type
write(retmsg,40) d_PAR%BetaAssoc(i), i, j
40 format('Error: Beta association value ',i6, ' corresponds to different parameter types.'
& ' Check rows ',i6,' and ',i6,' of the parameter table. Excecution stopped.')
call utl_writmess(6,retmsg)
stop
endif ! Finished control
do k = 1,cv_PAR%ndim
! calculate the distances
```

```
d_XQR%Q0(i,j) = d_XQR%Q0(i,j) + (d_PAR%lox(i,k) - d_PAR%lox(j,k))**2
!Here the squared distance
enddo
d_XQR%Q0(i,j) = sqrt(d_XQR%Q0(i,j)) ! Now calculate the sqrt
d_XQR%Q0(j,i)=d_XQR%Q0(i,j) ! Because the Q0 matrix is symmetric
endif
enddo
endif
d_XQR%X(i,d_PAR%BetaAssoc(i))= 1.  !Fill the X matrix to associate the
correct beta to each parameter
enddo
if (minval(cnp).eq.0) then
write(retmsg,10) minloc(cnp)
10 format('Error: No parameters correspond to beta association value',i6,
& '. Excecution stopped.')
call utl_writmess(6,retmsg)
stop
elseif (minval(cnp).eq.1) then
write(retmsg,20) minloc(cnp)
20 format('Error: Found only one parameter that corresponds to beta association
value',i6,
& '. Variogram type must be nugget. Excecution stopped.')
call utl_writmess(6,retmsg)
stop
endif
d_XQR%L = 10 * maxval(d_XQR%Q0) ! Define L as 10 times the maximum
distance
case (.FALSE.) ! We need to address this option
allocate(d_XQR%Q0(1,1)) d_XQR%Q0 = UNINIT_REAL
end select ! d_A%store_Q
case(1)
```

```
!Calculate compressed form of Q0 matrix
!(in block for each beta or vector for each beta if toepl_flag is 1 and 1 value for nugget)
!*****************************************************************
!******* Make the Q0_C matrix or vector in case of Toeplitz or single value
!******* in case of nugget and X matrix ******************************
!*****************************************************************
d_XQR%L = 0. !Initialize the 10 times maximum distance in the Q0_C matrices
select case (cv_A%store_Q)
case (.TRUE.)
do p = 1, cv_PAR%p !Loop for each beta that correspond to each different Q0_C
if (cv_S%var_type(Q0_All(p)%BetaAss)==0) then !  Q0_C is just a single 1 for nugget
allocate (Q0_All(p)%Q0_C(1,1)) !Allocation Just a value
Q0_All(p)%Q0_C(1,1) = 1.
else
select case (Q0_All(p)%Toep_flag) ! Toep_flag [0] full Q0 matrix for that beta [1] just vector with distances for that beta
case(0) !full matrix for this beta —> allocate the matrix [npar * npar] for the p-th beta
allocate (Q0_All(p)%Q0_C(Q0_All(p)%npar,Q0_All(p)%npar))
!Allocation Q0_All(p)%Q0_C = 0.
!Initialization
do i =1, Q0_All(p)%npar
do j=i+1, Q0_All(p)%npar
do k = 1,cv_PAR%ndim
! calculate the distances
Q0_All(p)%Q0_C(i,j) = Q0_All(p)%Q0_C(i,j) + &
& (d_PAR%lox(Q0_All(p)%Beta_Start+i-1,k) - &
& d_PAR%lox(Q0_All(p)%Beta_Start+j-1,k))**2 !Here the squared distance.
```

```
!Beta_Start identify where in the parameter list, starts the value with the p-th
beta association
enddo
Q0_All(p)%Q0_C(i,j) = sqrt(Q0_All(p)%Q0_C(i,j)) ! Now calculate the sqrt
Q0_All(p)%Q0_C(j,i) = Q0_All(p)%Q0_C(i,j) ! Because the Q0 matrix is
symmetric
enddo
enddo
case(1) !just a vector for this beta —>
allocate the matrix [npar * 1] for the p-th beta
allocate (Q0_All(p)%Q0_C(Q0_All(p)%npar,1)) !Allocation a vector
Q0_All(p)%Q0_C = 0. !Initialization
do j=2, Q0_All(p)%npar d
o k = 1,cv_PAR%ndim
! calculate the distances
Q0_All(p)%Q0_C(j,1) = Q0_All(p)%Q0_C(j,1) + &
& (d_PAR%lox(Q0_All(p)%Beta_Start,k) - &
& d_PAR%lox(Q0_All(p)%Beta_Start+j-1,k))**2 !Here the squared distance.
!Beta_Start identify where in the parameter list, starts the value with the p-th
beta association
enddo
Q0_All(p)%Q0_C(j,1) = sqrt(Q0_All(p)%Q0_C(j,1)) ! Now calculate the sqrt
enddo
end select !Q0_All(p)%Toep_flag
ltmp = maxval(Q0_All(p)%Q0_C) !Temporary value of maximum distance in
the p-th Q0_C matrix
if (ltmp.gt.d_XQR%L) d_XQR%L = ltmp
endif ! cv_S%var_type(Q0_All(p)%BetaAss)==0
enddo !p = 1, cv_PAR%p
d_XQR%L = 10 * d_XQR%L !before here L was just the maximum distance in
all the Q0_C matrices
```

```
case (.FALSE.) ! We need to address this option
allocate(d_XQR%Q0(1,1)) d_XQR%Q0 = UNINIT_REAL
end select ! d_A%store_Q
! Make the X matrix. The 1 values are associated in the order the parameter list
do i=1, cv_PAR%npar
d_XQR%X(i,d_PAR%BetaAssoc(i))= 1.  !Fill the X matrix to associate the
correct beta to each parameter
enddo
!*****************************************************************
!******* End Make the Q0_C matrix
!*****************************************************************
end select ! (cv_A%Q_compression_flag)
!*****************************************************************
!*****************************************************************
!** The next lines are valid for both the full and compressed form of Q0 cases *
!*****************************************************************
!*****************************************************************
!*****************************************************************
!**************** Make the R0 matrix. ********************************
!*****************************************************************
allocate(d_XQR%R0(nobs,nobs)) d_XQR%R0 = 0.    ! array
do i=1,nobs
d_XQR%R0(i,i) = 1./(d_OBS%weight(i)**2)
enddo
!*****************************************************************
!******** End Make the R0 matrix. **********************************
!*****************************************************************
!*****************************************************************
!***** Calculate the inverse of the Qbb matrix and InvQbb * beta0 **********
!************ We do that just if the betas_flag is not 0 ******************
!*****************************************************************
```

```
if (cv_PM%betas_flag .ne. 0) then !Calculate the inverse of the Qbb matrix
call INVGM(cv_PAR%p,d_PM%Qbb,d_PM%InvQbb)
!Calculate the product of inverse of the Qbb matrix and beta_0
d_PM%InvQbbB0=matmul(d_PM%InvQbb,d_PM%beta_0)
endif
!*********************************************************************
!***** End Calculate the inverse of the Qbb matrix and InvQbb * beta0 ********
!*********************************************************************
end subroutine bxq_make_X0_Q0_R0_InvQbb
```

Subroutine to set up the Q_{ss} matrix, make the operation to solve the Bayesian system and calculate the objective functions

```
subroutine bmo_mat_ops(d_XQR, d_S, d_PM, cv_PAR, cv_OBS,d_OBS, &
& cv_S, cv_A, d_A, d_PAR,Q0_All,cv_PM, c_iter)
implicit none
! declarations
type(kernel_XQR), intent(in) :: d_XQR
type(cv_struct), intent(in) :: cv_S
type(d_struct), intent(inout) :: d_S
type(d_prior_mean), intent(in) :: d_PM
type(cv_param), intent(in) :: cv_PAR
type(cv_observ), intent(in) :: cv_OBS
type(cv_algorithmic),intent(inout) :: cv_A
type(d_algorithmic), intent(inout) :: d_A
type(d_param), intent(inout) :: d_PAR
type(d_observ), intent(in) :: d_OBS
type(Q0_compr), intent(in) :: Q0_All(:)
type (cv_prior_mean), intent(in) :: cv_PM
integer, intent(in) :: c_iter
double precision, pointer :: Q0_tmp(:), TMP(:,:), Qrow(:)
```

```
double precision, pointer :: Qss(:,:), TVP(:),TMP1(:,:)
double precision, pointer :: LHS(:,:), RHS (:) , Soln(:), C_S(:)
integer :: ierr, i, j, k, cc, p, it, start_v, end_v
select case (cv_A%Q_compression_flag) !Select if the Q0 matrix is compressed
or not
case(0) !Full Q0 matrix
! Qss is the full matrix, made up of the kernel (Q0) multiplied by the appropiate
current theta values
!*****************************************************************
!Make Qss (Q) based on Q0 and variogram type
!*****************************************************************
select case (cv_A%store_Q)
case (.TRUE.)
allocate(Qss(cv_PAR%npar,cv_PAR%npar)) ! Allocation Qss = 0.
! Initialization
do i=1, cv_PAR%npar !Loop over all the parameters
select case (cv_S%var_type(d_PAR%BetaAssoc(i)))
case (0) ! means nugget —> just multiply by theta1
Qss(i,i)=d_S%theta(d_PAR%BetaAssoc(i),1,c_iter)*d_XQR%Q0(i,i)
case (1) ! means linear —> we need the maximum distance and theta1
do j=i, cv_PAR%npar
if (d_PAR%BetaAssoc(i).eq.d_PAR%BetaAssoc(j)) then !Search in the param-
eters list the associated parameters
Qss(i,j)=d_S%theta(d_PAR%BetaAssoc(i),1,c_iter)* &
& d_XQR%L*exp(-d_XQR%Q0(i,j)/d_XQR%L)
Qss(j,i)=Qss(i,j) ! Because Qss is symmetric
endif
enddo
case (2) ! means exponential —> we need theta1 and theta2
do j=i, cv_PAR%npar
```

```
if (d_PAR%BetaAssoc(i).eq.d_PAR%BetaAssoc(j)) then !Search in the parameters list the associated parameters
Qss(i,j)=d_S%theta(d_PAR%BetaAssoc(i),1,c_iter)* &
& exp(-d_XQR%Q0(i,j)/d_S%theta(d_PAR%BetaAssoc(i),2,c_iter))
Qss(j,i)=Qss(i,j) ! Because Qss is symmetric
endif
enddo
end select ! Variogram type
enddo
case (.FALSE.) ! We need to address this option
allocate(Qrow(cv_PAR%npar))
end select ! store_Q
!*****************************************************************
! End make Qss
!*****************************************************************
!*****************************************************************
! Make Qsy which is Qss*Ht
!*****************************************************************
allocate(d_A%Qsy(cv_PAR%npar,cv_OBS%nobs)) ! Allocation
d_A%Qsy = UNINIT_REAL ! Initialization
call dgemm('n','t',cv_PAR%npar, cv_OBS%nobs, cv_PAR%npar,
& 1.D0, Qss, cv_PAR%npar, d_A%H, cv_OBS%nobs,
& 0.D0, d_A%Qsy, cv_PAR%npar)
!*****************************************************************
! End make Qsy
!*****************************************************************
case(1) !Compressed form of Q0 matrix
!*****************************************************************
!Make Qsy which is Qss*Ht based on Q0 and variogram type. Qss is calculated on fly
!*****************************************************************
```

```
select case (cv_A%store_Q)
case (.TRUE.)
allocate(d_A%Qsy(cv_PAR%npar,cv_OBS%nobs)) ! Allocation
d_A%Qsy = UNINIT_REAL ! Initialization
do p = 1, cv_PAR%p !Loop for each beta that correspond to each different Q0_C (each beta has a separate Q0_C)
select case (cv_S%var_type(Q0_All(p)%BetaAss)) !Here the selection of the variogram type
case (0) ! means nugget —> just transpose the correct portion of H and multiply by theta1
d_A%Qsy(Q0_All(p)%Beta_Start:Q0_All(p)%Beta_Start+Q0_All(p)%npar-1,:) = &
& d_S%theta(Q0_All(p)%BetaAss,1,c_iter)* &
& (transpose(d_A%H(:,Q0_All(p)%Beta_Start:Q0_All(p)%Beta_Start+ &
& Q0_All(p)%npar-1))) !Portion of H(p)
case (1) ! means linear —> we need the maximum distance and theta1.
!We have 2 option: Toeplitz or not
select case (Q0_All(p)%Toep_flag) !Selection of Toeplitz [1] or not [0]
case(0) !Means no Toeplitz.....just compressed form. Q0(p) is the full matrix for the p-th beta
allocate (TMP(Q0_All(p)%npar,Q0_All(p)%npar))
allocate (TMP1(Q0_All(p)%npar,cv_OBS%nobs))
start_v = Q0_All(p)%Beta_Start
end_v = Q0_All(p)%Beta_Start+Q0_All(p)%npar-1
do it=1,Q0_All(p)%npar
TMP(it,:)= exp(-Q0_All(p)%Q0_C(it,:)/d_XQR%L)
enddo
call dgemm('n','t',Q0_All(p)%npar, cv_OBS%nobs, Q0_All(p)%npar, &
(d_S%theta(Q0_All(p)%BetaAss,1,c_iter)*d_XQR%L), &
& TMP, Q0_All(p)%npar, &
& d_A%H(:,start_v:end_v), cv_OBS%nobs, &
```

```
& 0.D0, TMP1, Q0_All(p)%npar)
if (associated(TMP)) deallocate(TMP)
do it =1,Q0_All(p)%npar
d_A%Qsy(start_v+it-1,:) = TMP1(it,:)
enddo
if (associated(TMP1)) deallocate(TMP1)
case(1) !Means Toeplitz. Q0(p) is just a vector with the distances
start_v = Q0_All(p)%Beta_Start
end_v = Q0_All(p)%Beta_Start+Q0_All(p)%npar-1
call toep_mult(Q0_All(p),d_A%H(:,start_v:end_v), cv_OBS%nobs, &
& (d_S%theta(Q0_All(p)%BetaAss,1,c_iter)),d_XQR%L, &
& d_XQR%L, d_A%Qsy)
end select !Q0_All(p)%Toep_flag)
case (2) ! means exponential —> we need theta1 and theta2. We have 2 option:
Toeplitz or not
select case (Q0_All(p)%Toep_flag) !Selection of Toeplitz [1] or not [0]
case(0) !Means no Toeplitz.....just compressed form. Q0(p) is the full matrix for
the p-th beta
allocate (TMP(Q0_All(p)%npar,Q0_All(p)%npar))
allocate (TMP1(Q0_All(p)%npar,cv_OBS%nobs))
start_v = Q0_All(p)%Beta_Start
end_v = Q0_All(p)%Beta_Start+Q0_All(p)%npar-1
do it=1,Q0_All(p)%npar
TMP(it,:)= exp(-Q0_All(p)%Q0_C(it,:)/ &
& (d_S%theta(Q0_All(p)%BetaAss,2,c_iter)))
enddo
call dgemm('n','t',Q0_All(p)%npar, cv_OBS%nobs, Q0_All(p)%npar, &
(d_S%theta(Q0_All(p)%BetaAss,1,c_iter)),TMP, Q0_All(p)%npar, &
& d_A%H(:,start_v:end_v), cv_OBS%nobs, &
& 0.D0, TMP1, Q0_All(p)%npar)
if (associated(TMP)) deallocate(TMP)
```

```
do it =1,Q0_All(p)%npar
d_A%Qsy(start_v+it-1,:) = TMP1(it,:)
enddo
if (associated(TMP1)) deallocate(TMP1)
case(1) !Means Toeplitz. Q0(p) is just a vector with the distances
start_v = Q0_All(p)%Beta_Start
end_v = Q0_All(p)%Beta_Start+Q0_All(p)%npar-1
call toep_mult(Q0_All(p),d_A%H(:,start_v:end_v), cv_OBS%nobs, &
& (d_S%theta(Q0_All(p)%BetaAss,1,c_iter)), &
& (d_S%theta(Q0_All(p)%BetaAss,2,c_iter)),1.D0 , &
& d_A%Qsy)
end select !Q0_All(p)%Toep_flag)
end select !(cv_S%var_type(Q0_All(p)%BetaAss))
enddo
case (.FALSE.) ! We need to address this option
allocate(Qrow(cv_PAR%npar))
end select ! store_Q
!*****************************************************************
! End make Qsy
!*****************************************************************
end select !(cv_A%Q_compression_flag)
!*****************************************************************
!*****************************************************************
!* The next lines are valid for both the full and compressed form of Q0 cases
!*****************************************************************
!*****************************************************************
!*****************************************************************
! Make Qyy which is H*Qss*Ht + sig*R0 = H*Qsy + sig*R0
!*****************************************************************
allocate(d_A%Qyy(cv_OBS%nobs,cv_OBS%nobs)) ! Allocation
d_A%Qyy=d_XQR%R0 !R0: to pass this to the multiplication subroutine,
```

```
!but the matrix will be overwritten with the real Qyy
call dgemm('n', 'n', cv_OBS%nobs, cv_OBS%nobs, cv_PAR%npar, &
1.D0, d_A%H, cv_OBS%nobs, d_A%Qsy, cv_PAR%npar, &
d_S%sig(c_iter), d_A%Qyy, cv_OBS%nobs)
!*************************************************************
! End make Qyy !***********************************************
!*************************************************************
! Make H*X
!*************************************************************
allocate(d_A%HX(cv_OBS%nobs,cv_PAR%p)) ! Allocation
call dgemm('n', 'n', cv_OBS%nobs, cv_PAR%p, cv_PAR%npar, &
1.D0, d_A%H, cv_OBS%nobs, d_XQR%X, cv_PAR%npar, &
0.D0, d_A%HX, cv_OBS%nobs)
!*************************************************************
! End make H*X !***********************************************
!*************************************************************
! Make Hsold which is H*d_PAR%pars_old *** Is a vector (nobs)
!*************************************************************
allocate(d_A%Hsold(cv_OBS%nobs)) !Allocation
call dgemm('n', 'n', cv_OBS%nobs, 1, cv_PAR%npar, &
1.D0, d_A%H, cv_OBS%nobs, d_PAR%pars_old, cv_PAR%npar, &
0.D0, d_A%Hsold, cv_OBS%nobs)
!*************************************************************
! End make Hsold
!*************************************************************
!*************************************************************
! Make RHS which is y' and -InvQbbB0 *** Is a vector (nobs + p)
!*************************************************************
allocate(RHS(cv_OBS%nobs+cv_PAR%p)) !Allocation
RHS= 0. !Initialization (IMPORTANT THAT IS = 0.)
RHS(1:cv_OBS%nobs) = d_OBS%obs - d_OBS%h + d_A%Hsold
```

```
if (cv_PM%betas_flag .ne. 0) then
RHS(cv_OBS%nobs+1:cv_OBS%nobs+cv_PAR%p) = - d_PM%InvQbbB0
endif
!*****************************************************************
! End Make RHS !************************************************
!*****************************************************************
! Make LHS which is Qyy HX and (HX)t -InvQbb *** Is a matrix (nobs + p *
nobs + p)
! Here we store the upper side of the matrix because symmetric
!*****************************************************************
allocate(LHS(cv_OBS%nobs+cv_PAR%p, cv_OBS%nobs+cv_PAR%p)) !Al-
location
LHS = 0. !Initialization (IMPORTANT THAT IS = 0.)
LHS(1:cv_OBS%nobs,1:cv_OBS%nobs)= d_A%Qyy
LHS(1:cv_OBS%nobs,cv_OBS%nobs+1:cv_OBS%nobs+cv_PAR%p) &
& = d_A%HX
if (cv_PM%betas_flag .ne. 0) then
LHS(cv_OBS%nobs+1:cv_OBS%nobs+cv_PAR%p,cv_OBS%nobs+ &
& 1:cv_OBS%nobs+cv_PAR%p)= - d_PM%InvQbb
endif
!*****************************************************************
! End Make LHS ! Here we stored the upper side of the matrix because symmetric
!*****************************************************************
!*****************************************************************
! Scale LHS which is C_S'*LHS*C_S Is a matrix (nobs+p*nobs+p) LHS is
overwritten by the scaled matrix
! Scaled using the matrix C_S(ii)=abs(LHS(i,i))**(-0.5)
! Here we calculate the upper side of the matrix because symmetric, the lower
side is not correct
!but we don't use that in the solution of the system LHS * InvC_S*Soln = RHS
! RHS is C_S * RHS and is overwritten too
```

```
!*****************************************************************
allocate(C_S(cv_OBS%nobs+cv_PAR%p)) !This is the vector with the diagonal values of the scaling matrix
C_S = UNINIT_REAL !Initialization
do i=1,cv_OBS%nobs+cv_PAR%p
if (LHS(i,i).eq.0.) then !If the value on the LHS diagonal is 0 then the scaling factor will be 1
C_S(i) = 1.
else
C_S(i) = abs(LHS(i,i))**(-0.5) !We use the absolute value because Qbb is negative
endif
LHS(i,:) = C_S(i) * LHS(i,:)
LHS(:,i) = C_S(i) * LHS(:,i)
RHS(i) = C_S(i) * RHS(i)
enddo
!*****************************************************************
! End Scale LHS
!*****************************************************************
!*****************************************************************
! Calculate the solution of the system LHS * InvC_S*Soln = RHS
! Note that here LHS and RHS are scaled using the C_S matrix *** the original was overwritten
!*****************************************************************
allocate(Soln(cv_OBS%nobs+cv_PAR%p))
Soln=RHS !Initialization, Soln will be overwritten with the solution of the system
call SLVSSU(cv_OBS%nobs+cv_PAR%p,LHS,Soln) ! Call to solve LHS * Soln = RHS
!*** Warning: On exit LHS is overwritten by the factorization used for the solution ***
!*** The solution of this system is C_S^-1 * Soln. Soln must be rescaled.
```

```
!Rescaling the solution and assign the values to ksi and beta_hat
allocate(d_A%ksi(cv_OBS%nobs)) !Allocation
allocate(d_A%beta_hat(cv_PAR%p)) !Allocation
do i=1,cv_OBS%nobs !Rescaling the solution and assign to ksi
d_A%ksi(i) = C_S(i) * Soln(i)
enddo
do i=cv_OBS%nobs+1,cv_OBS%nobs+cv_PAR%p !Rescaling the solution and
assign to beta_hat
d_A%beta_hat(i-cv_OBS%nobs) = C_S(i) * Soln(i)
enddo
!*****************************************************************
! End Calculate the solution
!*****************************************************************
!*****************************************************************
! Calculate best estimate s_hat which is d_PAR%pars = X*beta_hat + Qsy *
ksi
!*****************************************************************
call DGEMV('n',cv_PAR%npar,cv_PAR%p,1.D0, &
& d_XQR%X,cv_PAR%npar,d_A%beta_hat,1,0.D0,d_PAR%pars,1)
call DGEMV('n',cv_PAR%npar,cv_OBS%nobs,1.D0, &
d_A%Qsy,cv_PAR%npar,d_A%ksi,1,1.D0,d_PAR%pars,1)
!*****************************************************************
! End Calculate best estimate
!*****************************************************************
!*****************************************************************
! Make Gyy which is Qyy + HX * Qbb * HXt
!*****************************************************************
allocate(d_A%Gyy(cv_OBS%nobs,cv_OBS%nobs))
d_A%Gyy = d_A%Qyy
if (cv_PM%betas_flag .ne. 0) then
!Here we add HX * Qbb * HXt to Qyy only if we have prior mean informations
```

```
allocate(TMP(cv_OBS%nobs,cv_PAR%p))
TMP = UNINIT_REAL ! matrix
call dgemm('n','n',cv_OBS%nobs,cv_PAR%p,cv_PAR%p, &
1.D0,d_A%HX,cv_OBS%nobs,d_PM%Qbb,cv_PAR%p, &
0.D0, TMP, cv_OBS%nobs)
call dgemm('n','t',cv_OBS%nobs,cv_OBS%nobs,cv_PAR%p, &
1.D0,TMP,cv_OBS%nobs,d_A%HX,cv_OBS%nobs, &
1.D0, d_A%Gyy, cv_OBS%nobs)
if (associated(TMP)) deallocate(TMP)
endif
!*****************************************************************
! End Make Gyy
!*****************************************************************
!*****************************************************************
! Calculate the objective functions
!*****************************************************************
!Total objective function phi_T= 1/2 ksit * Gyy * ksi
allocate(TVP(cv_OBS%nobs))
TVP = UNINIT_REAL ! vector (temporary vector)
call DGEMV('n',cv_OBS%nobs,cv_OBS%nobs,1.D0, &
d_A%Qyy,cv_OBS%nobs, d_A%ksi,1,0.D0,TVP,1)
call DGEMV('t',cv_OBS%nobs,1,5.0D-1,d_A%ksi,cv_OBS%nobs, &
TVP,1,0.D0,d_PAR%phi_T,1)
if (associated(TVP)) deallocate(TVP)
!Measurement objective function phi_M = (y-h(s))t * R^-1 * (y-h(s))
!We use just a loop because R0 is diagonal *** Must change if allow full R0
matrix
d_PAR%phi_M = 0.
do i = 1, cv_OBS%nobs
d_PAR%phi_M = d_PAR%phi_M + (1./(d_S%sig(c_iter)* &
d_XQR%R0(i,i)))*((d_OBS%obs(i) - d_OBS%h(i))**2)
```

```
enddo
!Regularization objective function phi_R = phi_T - phi_M
d_PAR%phi_R = d_PAR%phi_T - d_PAR%phi_M
!************************************************************
! End Calculate the objective functions
!************************************************************
if (associated(Qss)) deallocate(Qss)
if (associated(Qrow)) deallocate(Qrow)
if (associated(Q0_tmp)) deallocate(Q0_tmp)
end subroutine bmo_mat_ops
```

Subroutine to make Q_{sy} in case of compressed form of Q and Toeplitz option

```
!************************************************************
!***** Subroutine to make Qsy in case of Toeplitz matrix *************
!************************************************************
subroutine toep_mult(Q0,H,nobs,theta_1,theta_2,Lmax,Qsy)
type(Q0_compr), intent(in) :: Q0
double precision, intent(in) :: H(nobs,Q0%npar)
double precision, intent(inout) :: Qsy(:,:) !Q0%npar,nobs)
double precision, intent(in) :: theta_1,theta_2,Lmax
integer, intent(in) :: nobs
double precision :: Qtmpb(Q0%npar),Qtmpg(Q0%npar)
double precision :: Qtmpl(Q0%npar),Qv(Q0%npar),TMP(Q0%npar)
integer :: ncol,nbl,nlay integer :: blkg,blkl integer :: i,l,k,p,it,jt
!Note: In case of linear variogram theta_1 must be theta_1,
!theta_2 and Lmax must be the 10 times the maximum distance in Q0_All
!In case of exponential variogram theta_1 must be theta_1,
!theta_2 must be theta_2 and Lmax must be 1
!Qsy = 0.
```

```
ncol=Q0%Ncol
nbl=Q0%Nrow
nlay=Q0%Nlay Qv=0.
Qtmpb=Q0%Q0_C(:,1)
Qtmpg=Q0%Q0_C(:,1)
Qtmpl=Q0%Q0_C(:,1)
new_block=.true.
blkg=1
blkl=1
i=0
do p=1, Q0%npar !Index for all the columns of the matrix
if (i/ncol.eq.1) then
if(blkg/nbl.eq.1) then
blkl = blkl+1
Qtmpb(1:(ncol*nbl)) = Q0%Q0_C((ncol*nbl*(blkl-1))+ &
& 1:((ncol*nbl)*(blkl-1))+(ncol*nbl),1)
Qtmpb((ncol*nbl)+1:(ncol*nbl*nlay)) = Qtmpl(1:(ncol*nbl*nlay)-(ncol*nbl))
Qtmpl = Qtmpb
Qtmpg = Qtmpb
blkg = 1
i=1
new_block=.true.
else
blkg = blkg+1
do l= 1,(nlay*ncol*nbl),(ncol*nbl)
Qtmpb(l:l+ncol-1)= Qtmpl((ncol*(blkg-1))+l:((ncol-1)*(blkg-1))+l+ncol)
Qtmpb(l+ncol:ncol*nbl+(l-1)) = Qtmpg(l:(ncol*nbl)-ncol+(l-1))
enddo
Qtmpg=Qtmpb
new_block=.true.
i=1
```

```
endif
else
i=i+1
endif
if (new_block) then
Qv=Qtmpb
new_block=.false.
else
do l= 1,(nlay*ncol*nbl),(ncol*nbl)
do k=1,nbl
Qv(ncol*(k-1)+l)= Qtmpg(ncol*(k-1)+i+l-1)
Qv(ncol*(k-1)+1+l:ncol*(k-1)+ncol+l-1)= &
& Qtmpb(ncol*(k-1)+l:ncol*(k-1)+ncol-2+l)
enddo
enddo
endif
Qtmpb=Qv
!*************************************************************
! Here, for each p, Qv is the vector that contain the value of the p-th columnn
of the matrix Q **********
!** From here Qv is available to be used in some calculation ***************
!*************************************************************
!**** Here we calculate for each column of Q0 (Qv of the p-th iteration in this
subroutine) H*Qt that is (Q*Ht)t **
!**** We assign the result to the p-th row (instead of column) of Qsy to obtain
Q*Ht
!**** that, at the end of the loop, is the Qsy for the specified beta *********
!*************************************************************
TMP = (theta_1*Lmax*exp(-Qv/theta_2))
call DGEMV('n',nobs,Q0%npar,1.D0,H,nobs,TMP,1,0.D0, &
& Qsy(Q0%Beta_start+p-1,:),1)
```

```
enddo !End of loop for each column of the entire Q matrix
end subroutine toep_mult
!*************************************************************
!**** End Subroutine to make Qsy in case of Toeplitz matrix *********
!*************************************************************
```

Subroutine to invert a matrix

```
subroutine INVGM(N,A,InvA) ! Subroutine to invert a matrix using DGETRI
integer, intent (in) :: N
integer :: INFO, LWORK
double precision, intent (in) :: A(N,N)
double precision, intent (inout) :: InvA(N,N)
integer :: IPIV(N)
double precision :: WORK (N)
InvA = A
call DGETRF(N,N,InvA,N,IPIV,INFO)
call DGETRI(N,InvA,N,IPIV,WORK,-1,INFO)
LWORK=INT(WORK(1))
call DGETRI(N,InvA,N,IPIV,WORK,LWORK,INFO)
end subroutine INVGM
```

Subroutine to solve a symmetric system $\mathbf{A} \cdot \mathbf{x} = \mathbf{B}$ upstored

```
subroutine SLVSSU(N,A,X) ! Subroutine to solve a symmetric
! system A*X=B upstored
integer ,intent (in) :: N
integer :: INFO, LWORK
double precision, intent (inout) :: A(N,N)
double precision, intent (inout) :: X(N)
integer :: IPIV(N)
double precision :: WORK (N)
```

```
call DSYSV('U',N,1,A,N,IPIV,X,N,WORK,-1,INFO)
LWORK=INT(WORK(1))
call DSYSV('U',N,1,A,N,IPIV,X,N,WORK,LWORK,INFO)
write(*,*) info
end subroutine SLVSSU
```

Bayesian PEST control file

```
BEGIN algorithmic_cv KEYWORDS
    lm_lambda=0.5 step_max_ds_lin=0.4
    lm_factor=3.162277 step_min_ds_lin=0.1
    lm_add=0.25 step_min_ds_neg=0.02
    lm_gamma=4. step_min_dL_neg=0.1
    lm_excessor=3. it_max_num_outer=10
    lm_beta_force=4. lm_max=1E+12
    chi2_max=0.9 step_max_ds_try=1.2
    chi2_min=0.1 theta_conv=0.005
    theta_cov_form=1 Q_compression_flag=1
END algorithmic_cv

BEGIN prior_mean_cv KEYWORDS
    prior_betas=0 beta_cov_form=2
END prior_mean_cv

BEGIN prior_mean_data TABLE
  nrow=4 ncol=6 columnlabels
  BetaAssoc beta_0 beta_cov_1 beta_cov_2 beta_cov_3 beta_cov_4
    1   0.1   5.5   65.   3.1   4.1
    2   0.5   65.   4.1   0.3   8.1
    3   1.   3.1   0.3   3.2   4.
    4   1.3   4.1   8.1   4.   2.
```

```
END prior_mean_data

BEGIN structural_parameter_cv TABLE
  nrow=4 ncol=7 columnlabels
 BetaAssoc prior_cov_mode trans_theta var_type alpha_trans struct_par_opt
num_theta_type
    1   2   1   0   20   1   1
    2   2   0   2   40   0   2
    3   2   1   1   30   1   1
    4   2   0   2   25   0   2
END structural_parameter_cv

BEGIN structural_parameters_data TABLE
  nrow=4 ncol=3 columnlabels
  BetaAssoc theta_0_1 theta_0_2
    1   0.1   -22.1
    2   2.    0.02
    3   0.02   0.3
    4   0.09   0.06
END structural_parameters_data

BEGIN structural_parameters_cov TABLE
  nrow=6 ncol=1 columnlabels
  theta_cov_1
    11.1
    11.9
    1.9
    111.9
    0.1
    0.2
```

```
END structural_parameters_cov

BEGIN epistemic_error_term KEYWORDS
    sig_0 = 1.349E-03 sig_opt = 1
END epistemic_error_term

BEGIN parameter_cv KEYWORDS
    ndim=3
END parameter_cv

BEGIN Q_compression_cv TABLE
  nrow=4 ncol=5 columnlabels
  BetaAssoc Toep_flag Nrow Ncol Nlay
    1   0
    2   1   3   2   2
    3   0
    4   1   2   2   4
END Q_compression_cv

BEGIN parameter_groups TABLE
  nrow=4 ncol=2 columnlabels
  groupname grouptype
    pargp_uno 1
    pargp_dos 2
    pargp_san 1
    pargp_yon 2
END parameter_groups

BEGIN parameter_data TABLE
  nrow=35 ncol=8 columnlabels
  ParamName StartValue GroupName BetaAssoc SenMethod x1 x2 x3
```

K2_1 1 pargp_dos 2 1 1 1 1

K2_2 2 pargp_dos 2 1 2 1 1

K2_3 3 pargp_dos 2 1 1 2 1

K2_4 4 pargp_dos 2 1 2 2 1

K2_5 5 pargp_dos 2 1 1 3 1

K2_6 6 pargp_dos 2 1 2 3 1

K2_7 7 pargp_yon 2 1 1 1 12.5

K2_8 8 pargp_yon 2 1 2 1 12.5

K2_9 9 pargp_yon 2 1 1 2 12.5

K2_10 10 pargp_yon 2 1 2 2 12.5

K2_11 11 pargp_yon 2 1 1 3 12.5

K2_12 12 pargp_yon 2 1 2 3 12.5

K1_1 50.2 pargp_dos 1 1 2.321 10.21 5.20e-0

K1_2 50.2 pargp_dos 1 1 3.321 10.21 5.20e-0

K1_3 50.2 pargp_dos 1 1 3.331 10.21 5.20e-0

K4_1 111 pargp_uno 4 1 1.5 1.5 6

K4_2 112 pargp_san 4 1 3 1.5 6

K4_3 113 pargp_san 4 1 1.5 3 6

K4_4 114 pargp_uno 4 1 3 3 6

K4_5 115 pargp_san 4 1 1.5 1.5 8

K4_6 116 pargp_uno 4 1 3 1.5 8

K4_7 117 pargp_san 4 1 1.5 3 8

K4_8 118 pargp_san 4 1 3 3 8

K4_9 119 pargp_san 4 1 1.5 1.5 10

K4_10 120 pargp_san 4 1 3 1.5 10

K4_11 121 pargp_san 4 1 1.5 3 10

K4_12 122 pargp_san 4 1 3 3 10

K4_13 123 pargp_san 4 1 1.5 1.5 12

K4_14 124 pargp_san 4 1 3 1.5 12

K4_15 125 pargp_san 4 1 1.5 3 12

K4_16 126 pargp_san 4 1 3 3 12

```
    K8974 9.8 pargp_uno 3 1 5 2.21 5.20e-3
    K8975 0.958 pargp_san 3 1 5.21 3.21 5.20e-3
    K8976 0.98334 pargp_san 3 1 5.1 2.221 5.20e-3
    K8977 0.98432 pargp_san 3 1 5.321 0.21 5.20e-3
END parameter_data

BEGIN observation_groups TABLE
  nrow=2 ncol=1 columnlabels
  groupname obsgp_oden obsgrp_dva
END observation_groups

BEGIN observation_data TABLE
  nrow=2 ncol=4 columnlabels
  ObsName ObsValue GroupName Weight
    K79TRIT 0.05 obsgrp_oden 1.00000001
    K98O18 0.223343114 obsgrp_dva 0.00000172892
END observation_data

BEGIN model_command_lines TABLE
  nrow=1 ncol=1 columnlabels
  Command read_trit_interp.exe
END model_command_lines

BEGIN model_input_files TABLE
  nrow=1 ncol=2 columnlabels
  TemplateFile ModInFile
    TRIT_Data_Master_interptxt.tpl TRIT_Data_Master_interp.txt
END model_input_files

BEGIN model_output_files TABLE
  nrow=1 ncol=2 columnlabels
```

```
  InstructionFile ModOutFile
    tritout.ins trit.out
END model_output_files
```

Bibliography

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D., 1999. LAPACK Users' Guide Release 3.0. Society for Industrial and Applied Mathematics, http://www.netlib.org/lapack/lug/node153.html.

Banta, E. R., Poeter, E. P., Doherty, J. E., Hill, M. C., 2006. Jupiter: Joint universal parameter identification and evaluation of reliability– an application programming interface (api) for model analysis. Tech. rep., U.S. Geological Survey Techniques and Methods Book 6, Section E, Chapter 1.

Bear, J., Verruijt, A., 1987. Modeling Groundwater Flow and Pollution. Springer.

Blas, 1997. BLAS (Basic Linear Algebra Subprograms). A Quick Reference Guide. University of Tennesse, Oak Ridge National Laboratory, Numerical Alghorithms Group LTD. http://www.netlib.org/blas.

Bohling, G. C., 1993. Hydraulic tomography in two-dimensional, steady-state groundwater flow (abstract). Eos Transactions AGU Spring Meeting Supplement 74 (16), 141.

Bohling, G. C., 2009. Sensitivity and resolution of tomographic pumping tests in an alluvial aquifer. Water Resources Research 45 (W02420).

Bohling, G. C., Butler, J. J., Zhan, X. Y., Knoll, M. D., 2007. A field assessment of the value of steady shape hydraulic tomography for characterization of aquifer heterogeneities. Water Resources Research 43 (W05430).

Bohling, G. C., Zhan, X. Y., Butler, J. J., Zheng, L., 2002. Steady shape analysis of tomographic pumping tests for characterization of aquifer heterogeneities. Water Resources Research 38 (12), 1324.

Brauchler, R., Liedl, R., Dietrich, P., 2003. A travel time based hydraulic tomographic approach. Water Resources Research 39 (12), 1370.

Bunch, J., Kaufman, L., 1997. Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems. Mathematics of Computation 31 (137), 163–179.

Butler, J., Liu, W., 1993. Pumping Tests in Nonuniform Aquifers: The Radially Asymmetric Case. Water Resources Research 29 (2), 259–269.

Butler, J. J., 2005. Hydrogeological methods for estimation of spatial variations in hydraulic conductivity. In: Rubin, Y., Hubbard, S. (Eds.), Hydrogeophysics. Springer, Netherlands, pp. 23–58.

Butler, J. J., McElwee, C. D., Bohling, G. C., 1999. Pumping tests in networks of multilevel sampling wells; motivation and methodology. Water Resources Research 35 (11), 3553–3560.

Castagna, M., Bellin, A., 2009. A bayesian approach for inversion of hydraulic tomographic data. Water Resources Research 45 (W04410).

Chapman, S. J., 2008. Fortran 95/2003 for scientists and engineers, 3rd Edition. McGraw-Hill Science/Engineering/Math, Boston, MA.

Clemo, T. M., 2007. MODFLOW-2005 ground water model - user guide to the adjoint state based sensitivity process (ADJ). Tech. Rep. BSU CGISS 07-01, Boise State University http://cgiss.boisestate.edu/pubs/CGISS_Techreports.html.

Coleman, T., Li, Y., 1994. On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds. Mathematical Programming 67 (2), 189–224.

Coleman, T., Li, Y., 1996. An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. SIAM Journal on Optimization 6, 418–445.

De Smedt, F., 1998. Two- and Three-Dimensional Flow of Groundwater. In: Delleur, J. (Ed.), The Handbook of Groundwater Engineering. CRC Press., School of Civil Engineering Purdue University West Lafayette, Indiana, pp. 18–1–18–36.

Dietrich, P., Fechner, T., Whittaker, J., 1998. An integrated hydrogeophisical approach to subsurface characterization. In: Herbert, M., Covar, K. (Eds.), Groundwater Quality: Remediation and Protection. Vol. 250. IAHS Publ., Wallingford, UK, pp. 513–520.

Doherty, J., 2008a. PEST, model independent parameter estimation. Addendum to user manual: 5th edition. http://www.sspa.com/PEST/.

Doherty, J., 2008b. PEST, model independent parameter estimation. User manual: 5th edition. http://www.sspa.com/PEST/.

Doser, D. I., Crain, K. D., Baker, M. R., Kreinovich, V., Gerstenberger, M. C., 1998. Estimating Uncertainties for Geophysical Tomography. Reliable Computing 4 (3), 241–268.

Ellis, T., 1990. Fortran 77 Programming With an Introduction to the Fortran 90 Standard, 2nd Edition. Addison-Wesley Publishing Company, Reading, MA.

Fienen, M. N., 2007. Inverse methods for nearfield hydrogeologic characterization. Ph. D. dissertation, Stanford University.

Fienen, M. N., Clemo, T. M., Kitanidis, P. K., 2008. An interactive Bayesian geostatistical inverse protocol for hydraulic tomography. Water Resources Research 44 (W00B01).

Gottlieb, J., Dietrich, P., 1995. Identification of the permeability distribution in soil by hydraulic tomography. Inverse Problems 11 (2), 353–60.

Harbaugh, A. W., 2005. MODFLOW-2005, the U.S. Geological Survey modular ground-water model – the ground-water flow process. Tech. rep., U.S. Geological Survey Techniques and Methods: 6-A16.

Harbaugh, A. W., Banta, E. W., Hill, M., McDonald, M. G., 2000. MODFLOW-2000, the U.S. Geological Survey modular ground-water model–user guide to modularization concepts and the ground-water flow process. Tech. rep., United States Geological Survey: Open File Report 00-92.

Higham, N., 1997. Stability of the Diagonal Pivoting Method with Partial Pivoting Symmetric Linear Systems. SIAM Journal on Matrix Analysis and Applications 18 (1), 52–65.

Hoeksema, R. J., Kitanidis, P. K., 1984. An application of the geostatistical approach to the inverse problem in two-dimensional groundwater modeling. Water Resources Research 20 (7), 1003–1020.

Hoover, D. B., Klein, D. P., Campbell, D. C., 1996. Geophysical Methods in Eploration and Mineral Enviromental Investigations. http://pubs.usgs.gov/of/1995/ofr-95-0831/CHAP3.pdf.

Illman, W. A., Liu, X. Y., Craig, A., 2007. Steady-state hydraulic tomography in a laboratory aquifer with deterministic heterogeneity: Multi-method and multiscale validation of hydraulic conductivity tomograms. Journal of Hydrology 341 (3-4), 222–234.

Kitanidis, P. K., 1991. Orthonormal residuals in geostatistics - model criticism and parameter-estimation. Mathematical Geology 23 (5), 741–758.

Kitanidis, P. K., 1995. Quasi-linear geostatistical theory for inversing. Water Resources Research 31 (10), 2411–2419.

Kitanidis, P. K., 1997. Introduction to Geostatistics: Applications in Hydrogeology. Cambridge University Press, New York, NY.

Kitanidis, P. K., Vomvoris, E. G., 1983. A geostatistical approach to the inverse problem in groundwater modeling (steady state) and one-dimensional simulations. Water Resources Research 19 (3), 677–690.

Knuth, D. E., 2009. The Art of Computer Programming, Volume 3: Sorting and Searching, 1st Edition. Addison-Wesley Professional, Reading, MA.

Lagarias, J., Reeds, J., Wright, M., Wright, P., 1998. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal on Optimization 9 (1), 112–147.

Li, W., Englert, A., Cirpka, O. A., Vanderborght, J., Vereecken, H., 2007. Two-dimensional characterization of hydraulic heterogeneity by multiple pumping tests. Water Resources Research 43 (4).

Li, W., Englert, A., Cirpka, O. A., Vereecken, H., 2008. Three-dimensional geostatistical inversion of flowmeter and pumping test data. Ground Water 46 (2), 193–201.

Li, W., Nowak, W., Cirpka, O. A., 2005. Geostatistical inverse modeling of transient pumping tests using temporal moments of drawdown. Water Resources Research 41 (8), 1–13.

Liu, S., Yeh, T.-C. J., Gardiner, R., 2002. Effectiveness of hydraulic tomography: Sandbox experiments. Water Resources Research 38 (4), 1034.

Liu, X., Illman, W. A., Craig, A. J., Zhu, J., Yeh, T.-C. J., 2007. Laboratory sandbox validation of transient hydraulic tomography. Water Resources Research 43 (W05404).

Mathworks, 2002. MATLAB.

McDonald, M. G., Harbaugh, A. W., 1988. A Modular Three-Dimensional Finite-Difference ground-water flow model. Vol. 06-A1 of USGS Techniques of Water Resources Investigations. United States Geological Survey (USGS), Reston, Virginia.

Merriam-Webster, 2009. Merriam-Webster Online Dictionary, Merrian-Webster Online. http://www.merriam-webster.com/dictionary/tomography.

Nowak, W., Cirpka, O. A., 2004. A modified Levenberg-Marquardt algorithm for quasi-linear geostatistical inversing. Advances in Water Resources 27 (7), 737–750.

Sharma, P. V., 1997. Environmental and Engineering Geophysics. Cambridge Univ. Pr., Cambridge, UK.

Slichter, C. S., 1899. Theoretical investigation of the motion of ground water. 19th annual report, U.S.G.S.

Straface, S., Yeh, T.-C. J., Zhu, J., Troisi, S., Lee, C. H., 2007. Sequential aquifer tests at a well field, Montalto Uffugo Scalo, Italy. Water Resources Research 43 (7).

Sun, N.-Z., 1994. Inverse problems in groundwater modeling. Theory and applications of transport in porous media ; v. 6. Kluwer Academic, Dordrecht ; Boston.

Sykes, J. F., Wilson, J. L., Andrews, R. W., 1985. Sensitivity analysis for steady state groundwater flow using adjoint operators. Water Resources Research 21 (3), 359–371.

Theis, C. V., 1935. The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage. Transactions of the American Geophysical Union 2, 519–524.

Thiem, G., 1906. Hydrologische Methoden. Gebhart, Leipzig.

Tosaka, H., Masumoto, K., Kojima, K., 1993. Hydropulse tomography for identifying 3-D permeability distribution. In: Proceedings of the 4th Annual International Conference on High Level Radioactive Waste Management; Apr 26-30 1993; Las Vegas, NV, USA. High Level Radioactive Waste Management. New York, NY, USA : Publ by ASCE, pp. 955–959.

Townley, L. R., Wilson, J. L., 1985. Computationally efficient algorithms for parameter estimation and uncertainty propagation in numerical models of groundwater flow. Water Resources Research 21 (12), 1851–60.

Vasco, D. W., Keers, H., Karasaki, K., 2000. Estimation of reservoir properties using transient pressure data: An asymptotic approach. Water Resources Research 36 (12), 3447–3465.

Wu, C. M., Yeh, T.-C. J., Zhu, J. F., Lee, T. H., Hsu, N. S., Chen, C. H., Sancho, A. F., 2005. Traditional analysis of aquifer tests: Comparing apples to oranges? Water Resources Research 41 (9).

Yeh, T.-C. J., Lee, C.-H., 2007. Time to change the way we collect and analyze data for aquifer characterization. Ground Water 45 (2), 116–118.

Yeh, T.-C. J., Liu, S., 2000. Hydraulic tomography: Development of a new aquifer test method. Water Resources Research 36 (8), 2095–2105.

Yeh, T.-C. J., Minghui, J., Hanna, S., 1996. An iterative stochastic inverse method: Conditional effective transmissivity and hydraulic head fields. Water Resources Research 32 (1), 85–92.

Yin, D., Illman, W. A., 2009. Hydraulic tomography using temporal moments of drawdown recovery data: A laboratory sandbox study. Water Resources Research 45 (W01502).

Zanini, A., Kitanidis, P., 2008. Geostatistical inversing for large-contrast transmissivity fields. Stochastic Environmental Research and Risk Assesment. Springer-Verlag.

Zhu, J. F., Yeh, T.-C. J., 2005. Characterization of aquifer heterogeneity using transient hydraulic tomography. Water Resources Research 41 (W07028).

Zhu, J. F., Yeh, T.-C. J., 2006. Analysis of hydraulic tomography using temporal moments of drawdown recovery data. Water Resources Research 42 (2).