

Model-Reduced Inverse Modeling

Model-Reduced Inverse Modeling

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen

op donderdag 13 april 2006 om 10.30 uur

door

Petrus Theodorus Maria VERMEULEN

HBO-ingenieur hydrologie en waterbeheer
geboren te Breukelen

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. A.W. Heemink

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof.dr.ir. A.W. Heemink, Technische Universiteit Delft, promotor
Prof.dr. G.J. Olsder, Technische Universiteit Delft
Prof.dr.ir. N.C. van de Giesen, Technische Universiteit Delft
Prof.dr.ir. J.D. Jansen, Technische Universiteit Delft
Prof.dr.ir. M.F.P. Bierkens, Universiteit Utrecht
Prof.dr. A.E.P. Veldman, Rijks Universiteit Groningen
Dr.ir. C.B.M. Te Stroet, TNO Bouw en Ondergrond

Model-Reduced Inverse Modeling

Ph.D. thesis Delft University of Technology - with ref. - with summary in Dutch

ISBN-10: 90-9020536-5

ISBN-13: 978-90-9020536-6

NUR 934, 919

Copyright © 2006 by Peter Vermeulen

All rights reserved

Cover design by Peter Vermeulen. It shows on the one side, the concept of *model reduction* which reduces the title in the upward direction; on the other hand, the principle of *upscaling* which coarsens the title downwards. Both methodologies are described in this thesis.

Printed in the Netherlands by Printpartners IPSKAMP B.V. Enschede

This project was carried out with financial support of TNO Build Environment and Geoscience



Voorwoord

DE DRANG OM TE LEREN en vooral te begrijpen waren mijn belangrijkste drijfveren om te starten met mijn promotieonderzoek. Als iemand mij nu zou vragen of deze drang nu dan enigszins verminderd is, moet ik teleurstellend antwoorden. Het doen van onderzoek blijkt een vicieuze cirkel van waaruit het overigens moeilijk ontsnappen is, zeker als het meezit maar nog meer als het tegenzit. Ik wil mijn promotor Arnold Heemink hierom dan ook bedanken voor zijn laagdrempelige begeleiding, stimulans en zijn betogen om het belang van resultaten breder te zien, dan die voor mijn eigen toepassing. Tegelijkertijd wil ik mijn 'excuses' aanbieden voor het feit dat ik wel eens te enthousiast was en hem hiervoor zelfs in het weekend stoorde. Dank ben ik ook verschuldigd aan Chris te Stroet, die mij het vertrouwen en de ruimte gaf. Het is spijtig dat we geen directe collega's meer zijn. Marc Bierkens en Wouter Zijl wil ik bedanken voor hun interesse, inhoudelijke discussies en gedegen commentaren op manuscripten. Zeker ook mijn familie en vrienden voor hun belangstelling en hun bereidheid om teksten te corrigeren. Verder dank ik prof.dr. G.J. Olsder, prof.dr.ir. N.C. van de Giesen, prof.dr. A.E.P. Veldman en prof.dr.ir. J.D. Jansen, voor het zitting nemen in de promotiecommissie. Tevens wil ik TNO bedanken voor het medefinancieren van dit onderzoek en het beschikbaar stellen van alle benodigde faciliteiten. In het bijzonder bedank ik mijn paranifmen en tevens kamergenoten, Aris Lourens en Wilbert Berendrecht. We hebben met z'n drieën een reputatie hoog te houden gezien het enthousiasme, de (on)zin, liederen en andere geluiden die we regelmatig produceren.

Tot slot, moet ik diep, diep door de knieën voor alle tijd die ik heb moeten ontnemen van mijn Erna, Aniek en Stijn. De vele zondagen, waarbij Aniek zei: "Wij gaan even naar de kinderboerderij, kun jij weer even lekker werken", zijn nu voorbij. Het is mooi geweest.

Peter Vermeulen

P.s. Eigenlijk is het meer dan bijzonder dat dit proefschrift er nu ligt;

"... In het belangstellingsonderzoek gaf Peter aan geen interesse te hebben voor cijfermatige beroepen, voor natuurwetenschappelijke beroepen en voor literaire activiteiten ..."

bron: Beroepskeuzevoorlichting, 12 februari 1986

Contents

Voorwoord	5
Notation	11
1 General introduction	13
1.1 Problem Definition	15
1.2 Methodologies	17
1.2.1 Upscaling	17
1.2.2 Model Reduction	18
1.3 Inverse modeling	21
1.4 Research Objectives	22
1.5 Thesis outline	22
2 Limitations to Upscaling of Groundwater Flow Models	25
2.1 Governing Equations and Error Measures	28
2.1.1 Fine-Scale Equations	28
2.1.2 Coarse-Scale Equations	29
2.1.3 Error Measures	30
2.2 Upscaling Hydraulic Parameters	32
2.2.1 Power Averaging (GEO method)	32
2.2.2 Darcian Methods (HNF method)	32
2.2.3 Homogenization (TEN & PTEN methods)	34
2.2.4 Local-Global Method (LG method)	35
2.2.5 Global-Local Methods (GL method)	35
2.2.6 Vertical Hydraulic Transmissivity	36
2.2.7 Examples	36
2.2.8 Conclusions	39
2.3 Upscaling Boundary Conditions	40
2.3.1 Dirichlet Conditions	40
2.3.2 Neumann Conditions	42
2.3.3 Cauchy Conditions	43
2.3.4 Conclusions	44
2.4 Real-World Case	44
2.4.1 Introduction	44
2.4.2 Results	46
2.5 Conclusions and Recommendations	47

2A	A Standard Seven-Point Stencil	48
2B	Finite Difference Simulation with Tensorial Hydraulic Transmissivities	49
2C	Periodic Boundary Conditions	51
3	Low-dimensional Modeling of Numerical Groundwater Flow	53
3.1	Reduced Model Methodology	57
3.1.1	Formulation of Groundwater Flow	57
3.1.2	Formulation of a Reduced Model Structure	57
3.1.3	Pattern Identification (EOF)	57
3.1.4	Physical Meaning of Patterns	61
3.2	Reduced Model Formulations	63
3.2.1	State-Space Projection Method	63
3.2.2	Galerkin Projection Method	65
3.2.3	Initial Conditions	67
3.2.4	Discussion	68
3.3	Real-world Case	69
3.3.1	Description of the Original Model	69
3.3.2	Purpose of the Low-dimensional Model	70
3.3.3	Preparation Phase	70
3.3.4	Simulation Phase	74
3.3.5	Efficiency	79
3.4	Comparison of the EOF and Lanczos method	80
3.4.1	Lanczos Vectors	80
3.4.2	Lanczos Reduction Method	83
3.4.3	Synthetic Cases	84
3.5	Conclusions and Recommendations	86
4	Inverse Modeling of Groundwater Flow using Model Reduction	89
4.1	Methodology	91
4.1.1	Formulation of Groundwater Flow	91
4.1.2	Reduced Model	92
4.1.3	Inverse Modeling	94
4.2	Application to a Synthetical Problem	96
4.2.1	Problem Description	96
4.2.2	Snapshot Simulation	96
4.2.3	Resulting Patterns	98
4.2.4	Sensitivity of $\hat{J}$	98
4.3	Application to a Real-World Case	102
4.3.1	Description of the Original Model	102
4.3.2	Results	103
4.4	Conclusions	106
4A	First Derivative of the Hydraulic Conductance	107
4B	Derivation of the Reduced Adjoint State Variable	107

5	Model-Reduced Variational Data Assimilation	109
5.1	Inverse Modeling	112
5.2	Reduced Model Methodology	113
5.2.1	Introduction	113
5.2.2	Methodology	113
5.2.3	Reduced Model Structure	114
5.2.4	Pattern Identification	116
5.2.5	Generic Reduced Model Formulation	117
5.2.6	Adjoint of the Reduced Model	118
5.2.7	Efficiency	119
5.3	Synthetic Cases	119
5.3.1	Introduction	119
5.3.2	Problem Description	120
5.4	Conclusions	127
5A	Derivation of the Reduced Adjoint State Variable	128
6	Inversion of Reduced Nonlinear Incremental Models	131
6.1	Classic Methodology	134
6.1.1	Forward Model	134
6.1.2	Inverse Model	134
6.2	Reduced Model Methodology	136
6.2.1	Introduction	136
6.2.2	Incremental Galerkin Projection Method	137
6.2.3	Incremental Sequential Projection Method	140
6.2.4	Snapshots	140
6.2.5	Forward Model	141
6.2.6	Adjoint Model	141
6.3	Computational Efficiency	142
6.4	Examples	143
6.4.1	Introduction	143
6.4.2	Case I	143
6.4.3	Case II	145
6.5	Real-world Case	146
6.5.1	Configuration	146
6.5.2	Results	146
6.6	Conclusions	146
6A	Hydraulic Conductance	148
6B	Derivation of the Original Adjoint Model	148
6C	Derivation of the First Derivative of the Hydraulic Conductance	148
6D	Derivation of the Reduced Incremental Adjoint Model	149
6E	Perturbation of the Hydraulic Conductance with a Pattern Direction	150

7 Summary and Conclusions	151
7.1 Upscaling	152
7.2 Model Reduction Methodologies	153
7.2.1 Scenario Analysis	153
7.2.2 Inverse Modeling	155
Samenvatting en Conclusies	157
Index	173
Curriculum vitae	179

Notation

In general, lowercase and uppercase *italic* type represent variables or functions. Vectors and matrices are represented by a lowercase and uppercase Sans serif **bold** type font. The page on which the symbol is introduced or explained is given in brackets.

General Conventions

i, j, k	indices
x, y, z	spatial coordinates
z	scalar
$\mathbf{z}$	vector
$\mathbf{Z}$	matrix
$\underline{\mathbf{z}}$	low-dimensional vector
$\hat{\mathbf{z}}$	approximated vector
$\bar{\mathbf{z}}$	mean vector
$\tilde{\mathbf{z}}$	linearized quantity
$(\cdot)$	specific parameter
$\langle(\cdot)\rangle$	mean quantity
$\ (\cdot)\ $	absolute quantity
$(\cdot)^T$	transpose
$(\cdot)^{-1}$	inverse, i.e. $1/(\cdot)$
$(\cdot)^a$	posterior
$(\cdot)^b$	background
$(\cdot)^{\text{cr1}}$	aligned to columns, rows or layers
$(\cdot)^d$	Dirichlet condition
$(\cdot)^e$	evaluation
$(\cdot)^f$	Cauchy condition
$(\cdot)^o$	observation
$(\cdot)^p$	partitioned matrix
$(\cdot)^t$	truth
$(\cdot)^{xyz}$	towards an x, y or z direction
$(\cdot)^*$	neighbouring grid cell
$(\cdot)_{i,j,k}$	specific element within vector

Lowercase roman symbols

c	coefficient	[50]
$\mathbf{d}$	corrected snapshot	[58]
f	power factor	[32]
h	thickness aquifer	[29]
h'	level correction	[43]

k	permeability	[29]
k^{aa}	tensorial components within k	[29]
n^d	number of Dirichlet conditions	[40]
n^e	number of snapshots	[58]
n^g	number of groups	[70]
n^l	number of Lanczos vectors	[82]
n^m	number of grid nodes	[63]
n^o	number of observations	[92]
n^p	number of patterns	[60]
n^q	number of source terms	[80]
n^t	number of time steps	[70]
n^u	number of estimate variables	[114]
q	source/sink term	[28]
r	time-dependent coefficients	[57]
t	time	[57]
t^p	preparation time	[79]
t^e	simulation time	[105]
t^∇	t^e to compute ∇J	[105]
t^∂	time to compute PDE	[105]
$\mathbf{u}$	Darcy velocity	[28]
w	observational weight	[92]
$\mathbf{x}$	general state vector	[112]
$\mathbf{y}$	general estimated observation	[112]

Uppercase roman symbols

$\mathbf{A}$	system matrix	[64]
$\mathbf{B}$	operator	[50]
C	block conductance	[48]
$\mathbf{E}$	snapshots	[58]
E	efficiency	[80]
$\mathbf{F}$	projection matrix	[116]
$\mathbf{G}$	covariance matrix	[58]
H	operator that maps the model field to observation space	[112]

I	identity	[60]	Uppercase greek symbols	
J	objective/cost function	[92]	Γ	Dirichlet condition [120]
K	hydraulic conductivity	[57]	Λ	set of eigenvalues [59]
L	set of Lanczos vectors	[81]	Ξ	relative mass error [76]
M	dynamics operator	[112]	Ω	model domain [29]
N	matrix	[65]	Calligraphical	
P	set of patterns	[60]	$\mathcal{C}$	fictitious model complexity [80]
$\mathbf{P}^{\text{rot}}$	set of rotated patterns	[62]	$\mathcal{F}$	function [63]
Q	summation of fluxes	[50]	$\mathcal{L}$	set of parameters [93]
R	vertical resistance	[48]	$\mathbb{R}^n$	n -dimensional Euclidean space [60]
R^2	regression coefficient	[80]	$\mathcal{S}$	model subspace [58]
S	storage coefficient	[57]	$\mathcal{X}$	original model space [58]
T	tridiagonal matrix	[81]	$\mathcal{Z}$	zone definition [116]
T	transmissivity	[29]	∇	gradient of scalar [94]
V	set of eigenvectors	[59]	$\nabla \cdot$	gradient of vector [28]
V	grid block volume	[31]	Acronyms	
W	weight matrix	[92]	ADJ	Adjoint [135]
Lowercase greek symbols			AR	Autoregressive [112]
α, β, γ	factor	[91]	CFA	Common Factor Analysis [55]
α^e	factor for snapshots	[96]	CS	Coherent Structures [55]
δ	step size	[96]	EOF	Empirical Orthogonal Function [55]
δ^ϕ	head difference	[31]	FDM	Finite Difference Method [135]
ε^ϕ	relative head error	[31]	GEO	Geometric Average [32]
ε^d	relative dissipation error	[31]	GL	Global-Local [35]
ε^u	relative error of Darcian flow	[31]	GPM	Galerkin Projection Method [65]
η	gradient loop	[95]	HNF	Head No-Flow [32]
θ	error of mass conservation	[76]	IGPM	Incremental GPM [137]
ι	Picard iteration	[134]	ISPM	Incr. Sequential Proj. Method [140]
κ	pattern loop	[95]	KL	Karhunen-Loève Decomposition [133]
λ	eigenvalue	[59]	LG	Local-Global [35]
λ'	criterion	[61]	MIPS	Million Instructions Per Second [15]
μ	snapshot simulation loop	[95]	NAP	Dutch Ordnance Datum [63]
ξ	upscaling factor	[29]	ODE	Ordinary Differential Equation [63]
o	angle	[62]	PCA	Principal Component Analysis [55]
ρ	participation factor	[82]	PDE	Partial Differential Equation [55]
ϱ	percentile	[71]	POD	Proper Orth. Decomposition [55]
σ	variance	[31]	PTEN	Prior Tensorial [35]
τ	original model simulations	[120]	REV	Representative Elem. Volume [35]
ϕ	hydraulic head	[28]	RMAE	Relative Mean Absolute Error [74]
φ	relative eigenvalue	[59]	RRMSE	Relative Root Mean-Square [74]
φ^e	expected relative variance	[60]	SSPM	State-Space Projection Method [63]
φ^h	original relative variance	[60]	SSPM ^a	Alternative SSPM [65]
φ^l	reduced relative variance	[60]	TEN	Tensorial [34]
φ^r	reconstructed relative variance	[60]		
ω	distribution index	[42]		

1

General introduction



PREDICTION OF SPECIFIC BEHAVIOR is one of the main issues that mankind can be engaged in. It gives a sense of authority, which is an important reason why children and adults like to watch reruns on television. Moreover, a reliable prediction of your opponent's behavior may mean the difference between winning and losing, or surviving and dying. Therefore, people nowadays are sensitive of elements in society that claim to predict a specific behavior in the future. As a result, an overwhelming number of scientific studies try to predict different kinds of physical, and even emotional behavior by numerical simulations. It is like modern witchcraft behind a computer without the risk of dying at the stake.

We can divide numerical simulation models into two groups of model structures. (1) Models with a structure that is purely based upon measured data for a certain time window. These models predict a phenomenon by assuming that a future behavior is similar to its behavior during the measured time window (e.g. referred to as 'black-box' models or time-series models). An advantage of these type of models is that no knowledge is necessary whatsoever of the interaction of phenomena because this is captured by the data. Though they compute fast, a disadvantage is that we need measurements first before we are able to predict. (2) In contrast, deterministic models are based upon physical descriptions of phenomena and can be used instantaneously to predict. Nonetheless, also these descriptions contain parameters that are often uncertain or valued by field experiments and/or heuristic assumptions. However, they can be optimized by assimilating additional measurements (data assimilation). Therefore, to develop a reliable prediction, measurements are necessary as well.

To use one of the above mentioned numerical simulation models, we need to describe our 'reality' (i.e. the reality that we can observe). Generally, this means that we sample our 'reality' in a time- and/or spatial domain according to a mesh of, for example rectangles or triangles, and neglect information that falls outside the sampling frequency. Compared to the pixels on our television screen, our 'reality' is represented better whenever we increase the number of pixels. Unfortunately, the number of pixels is limited as computer resources are limited (e.g. computational performance CPU, memory RAM and disk space). The main object of the research presented in this thesis is therefore to examine and develop alternative methods with low-computational costs that conserve the detailed information of

a high number of pixels. We have distinguished two methodologies that we have applied to a numerical simulation of 3D groundwater flow. However, the method implementations can easily be extended towards other fields of science.

- We have described an approach in which we simply reduce the number of pixels and compute average quantities for our model parameters in such a way that the resulting model optimally represents a solution of the original model. This approach is known as *upscaling*.
- The second methodology is known as *model reduction* in which we use a deterministic model structure to generate synthetic experimental data. With those empirical data we build a reduced model with a structure comparable to a 'black-box' model. Basically, it rewrites a deterministic model to a 'black-box' model and reduces the complexity of the original deterministic model such that its essential part remains in the 'black-box' model.

This chapter starts with the problem definition and a brief description of the two techniques. It is followed by the research objectives and concludes with an outline of the thesis.

1.1 Problem Definition

In 1965, Intel co-founder Gordon Moore predicted that the number of transistors (i.e. integrated circuits) that can be fitted onto a square inch of silicon doubles every twelve months (Moore's Law [Moore, 1965]). Moreover, each time transistor size shrinks, integrated circuits become cheaper and perform better. For example, in 1965, a single transistor cost more than a euro. By 1975, the cost of a transistor had dropped to less than a euro cent, while transistor size allowed for almost 100,000 transistors on a single die. From 1979 to 1989, processor performance went from about 1.5 million instructions per second (MIPS), to almost 50 MIPS on the i486's. The next ten years it went to over 1,000 MIPS on the Intel® Pentium® III. Today's Intel® processors, run at 3.2 GHz and higher, deliver over 10,000 MIPS, and can be manufactured in high volumes with transistors that cost less than 1/10,000th of a euro cent. Today's microprocessors run everything from toys to traffic lights, and a musical birthday card today has more computing power than the fastest mainframes of a few decades ago. So, why should we bother about model reduction in the first place?

Numerical modeling of physical processes requires a lot of computer resources. These models are conceptual approximations that describe physical systems by use of mathematical equations. The applicability or usefulness of these models depends on how closely the mathematical equations approximate the physical system being modeled. Moreover, it depends on the scale on which we study different phenomena and that approximation scale is always related to the computing power available. Furthermore, we need additional computing power to express the uncertainties within a model through conceptual and mathematical errors. Due

to Moore's Law, scientific researchers approach every two years a different equilibrium between 'reality' and its approximation by numerical simulations. Nevertheless, why should we wait another two years, if we can increase our computing power by model reduction right now?

Model reduction is only possible whenever we sustain a different model formulation with a limited application. The underlying mathematical model definition is limited to its assumptions of physical processes, but if we consider the application itself we observe that model results are 'bounded' or 'structured' caused by the underlying mathematical formulations. Consider a car for which the structure is given by a mathematical expression, in fact this is our model definition. Whenever we move the car, which is the application of the model, we recompute its position instead of reconstructing the entire car for each different location. Mathematically, we need to capture its structure and formulate an equation to compute its position. This phenomenon can be illustrated by considering the generalized equation for a sphere (i.e. $x_i^2 + x_{i+1}^2, \dots, x_n^2 \leq r^2$) for which we compute the solution for a particular valued r and $n = 2$ (i.e. a circle), see Figure 1.1a. We observe only that a part of the rectangular-shaped 2D domain is occupied by possible solutions of x_1, x_2 for a given value for r . The *emptiness of space* is in that case one minus the volume of a circle divided by the volume of the coordinate system that surrounds it (i.e. a *span*), so

$$\text{emptiness of space} = \left(1 - \frac{\pi r^2}{4r^2}\right) \cdot 100\% \quad (1.1)$$

which yields 21%. If we compute the *emptiness of space* for a sphere ($n = 3$), that exists in a 3D domain (its volume is given by $4/3\pi^3$), the emptiness is increased to 48% (see Figure 1.1b) and to 70% for a *hypersphere* ($n = 4$). Strangely enough, whenever n approaches infinity, the volume of the n D sphere approaches zero, regardless of the value of r . This is an important insight because we can consider r as our typical application (i.e. the scenario that we want to compute with our model,

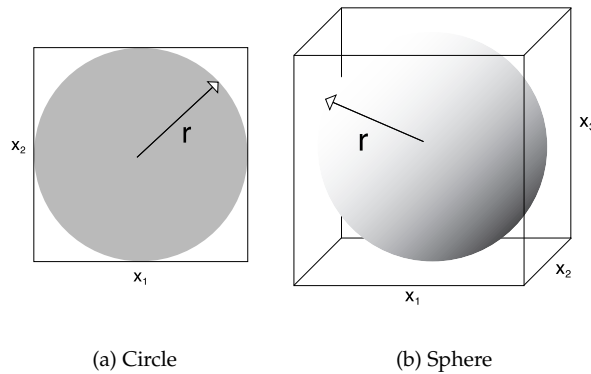


Figure 1.1: Presentation of (a) a circle in a 2D space and (b) a sphere in a 3D space.

for example we compute another circle with a new value for the radius r) and the coefficients x_i as the unknowns in our mathematical model. In other words, the emptiness of space is such enormous that it is hardly possible for example to hit a high-dimensional hypersphere with a dart.

By truncating the *emptiness of space* we achieve a different definition of the original model. As a result the new model formulation becomes optimal in the sense that it contains no emptiness regarding the considered application. Since nowadays models compute millions of unknowns, it is obvious that these model reductions will save a huge amount of computational resources. These savings could especially benefit the process of parameter optimization because this requires an enormous effort from the model. In this thesis we describe some methodologies to find these type of new models.

1.2 Methodologies

This thesis describes two methodologies to reduce the computational demands of large-scale numerical simulations models. Both methods reduce the ‘size’ of the problem and through that, introduce some numerical discrepancy between the original model and its derivative model. Both methods reduce the problem in its spatial domain, the first method reduces it locally (i.e. it reduces the number of *pixels* locally), the second method reduces the problem globally (i.e. according to its behavior).

1.2.1 Upscaling

Reducing the number of pixels (*upsampling*) is the most straightforward approach in decreasing the computational demands for a model in terms of storage capacity

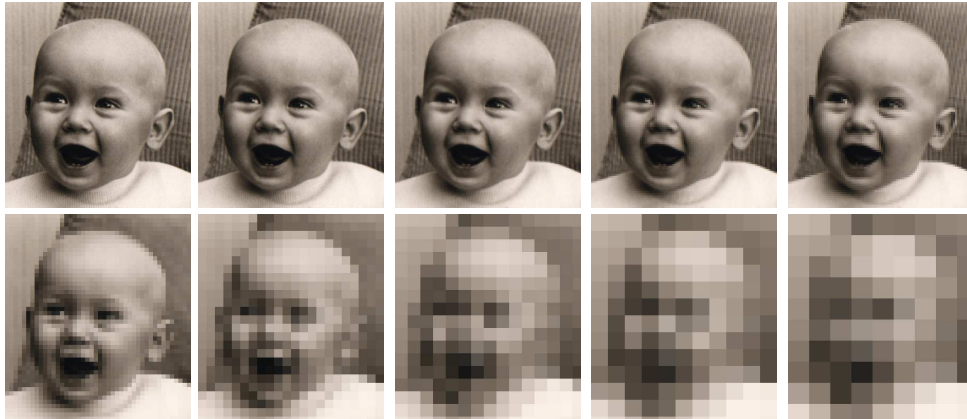


Figure 1.2: Reduction of the pixel resolution (*upsampling*) of a picture that represents a baby face. The upsampling factor increases from the first picture on the top row towards the right from 1 to 5. For the bottom row, it increases from 10 to 50 with increments of 10. It is obvious that an upsampling of 5 can be carried out without serious loss of information.

and CPU requirements. It is unavoidable that the accuracy of the upscaled model is always less than the original model, simply because we need to describe the same amount of information by fewer pixels. Since the efficiency of a simulation model often relates quadratically to the number of pixels, it is worthwhile to compute the highest achievable efficiency that yields an acceptable loss in accuracy. Therefore we need to define a mathematical algorithm that conserves most of the relevant information and neglects the remaining part. Consider a problem that needs to reduce the number of pixels for a picture of a baby face. We formulated a mathematical algorithm that expresses each upscaled pixel as the mean grey tint of all original pixels within that upscaled pixel. Whenever we applied an upscaling factor of 5 we still obtained an accurate description of the baby face but we achieved an efficiency of $5^2=25$, see Figure 1.2. Moreover, higher efficiencies could be achieved whenever we were satisfied with a global appearance of the baby face. In that case, we lost details, but we kept insight in the global behavior. Therefore, the upscaling factor had to be attuned to the model intentions. Moreover, a spatial distribution of upscaling factors is also possible. In such cases we can apply a coarsening of pixels for those areas only that possess no detail (for example the bald head of the baby), and do not disturb the remaining areas.

In this thesis we will apply upscaling algorithms to the numerical simulation of groundwater flow. Although this is quite different from the upscaling of a baby face, the objective and consequences are similar.

1.2.2 Model Reduction

The second methodology is based upon model reduction. Roughly speaking, the goal of model reduction is to replace initial data by data that are optimal in terms of storage capacity. We achieve this by suppressing redundant data that exist within multi-dimensional or *multivariate* data: information that is organized such that each datum in the data set is identified with a point in $\mathbb{R}^n$. It belongs therefore in the category of data compression. *Principal Component Analysis* (PCA) is the most effective way to analyze such multivariate data sets. It ‘breaks up’ the initial data sets into different components (i.e. the principal components) that all together represent the original data set again. The original data can be obtained again by multiplying the principal components with appropriate weight factors (i.e. a linear combination). Since each principal component represents a specific importance to the overall solution, we can ignore those principal components that are of minor interest to the solution. In this way, we represent our initial data set with even fewer principal components, without a significant loss of accuracy.

The above mentioned method can best be illustrated by the following example. Imagine that we want to compress the number of letters of the alphabet to reduce the number of keys on a typewriter. Instead of 26 keys we try to find other, more abstract letters to put on a typewriter, that can reconstruct each real letter by pressing these with different strengths. First, we represent or describe each letter within the alphabet by a grid of 5×5 squares. Within the grid we draw the shape of each

letter by assigning a value of one or a value of zero to the squares within the grid, see Figure 1.3. We refer to these grids as *snapshots* which describe all possible appearances (i.e. the *solutions*) that need to be represented by the abstract letters (i.e. the *principal components*).

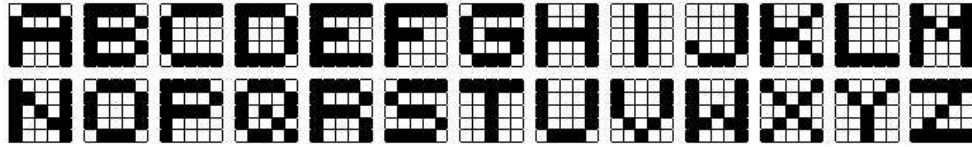


Figure 1.3: Represented letters of the alphabet on a grid of 5×5 squares. Black is assigned to a value of one, white to a value of zero.

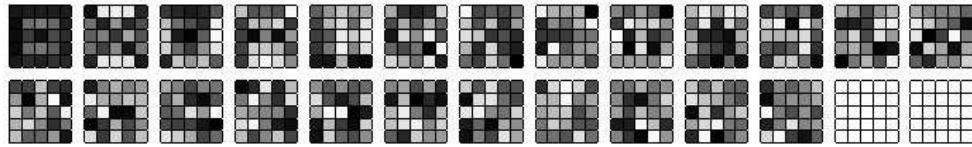


Figure 1.4: Eigenletters that explain parts of the real letters of the alphabet. The relative importance of each eigenletter decreases from the first picture on the top row towards the right, until the last column on the bottom row. The eigenletters are valued between 0–1.3, a value that corresponds to a colour on the greyscale between white and black.

When we apply the *Principal Component Analysis* to the letters, we find the abstract letters, which we refer to as *eigenletters* that explain a particular part of any letter of the alphabet (see Figure 1.4). The last two eigenletters are ‘empty’, because there was no more information left that was not yet captured by the previous eigenletters 1–24. Moreover, when we analyzed the relative importance for each eigenletter, we observed only that the eigenletters 1–16 explained some relevant amount of information. They described $> 99\%$ of the information. Although it is difficult to observe any correspondence between the eigenletters and the real letters of the alphabet, the relative importance of each eigenletter is a good criterion whether this eigenletter is necessary to reconstruct any letter of the alphabet. The main thought of model reduction is now to focus on the important eigenletters (*modes*) and neglect the other ones. In Figure 1.5 we have depicted the reconstruction of each letters of the alphabet by applying an optimal pressure (*weight factors*) on the eigenletters in the sense that the error between the reconstructed letter and the real one is minimal. We observe that this error is only zero when we use all 24 eigenletters. Nevertheless when we apply a filter which assigns black to the values between 0.85–1.15 and white to the remaining values, we can read already each letter fairly well by 16 eigenletters, see Figure 1.6. So, in model reduction there is always a relation between accuracy and efficiency, and high efficiencies can be obtained whenever we can ignore a large number of modes. In the example with the alphabetic letters we achieved an efficiency of $26/16=1.625$.

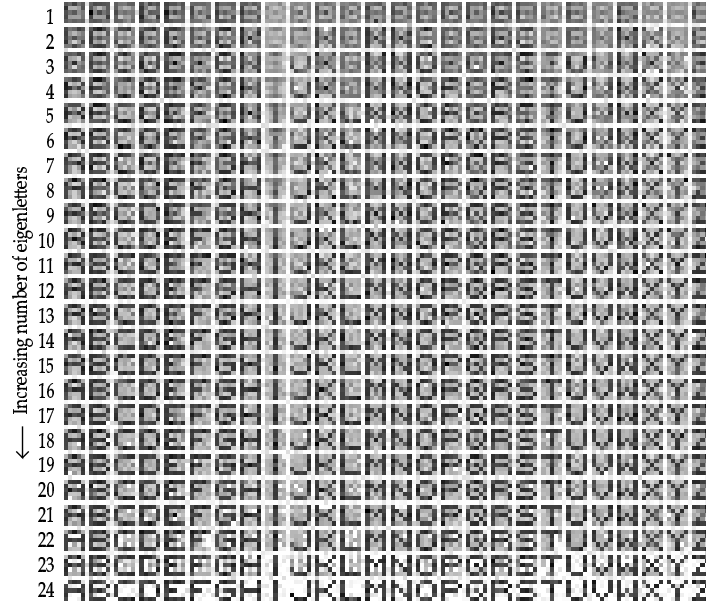


Figure 1.5: Reconstruction of the real letters of the alphabet by use of the eigenletters (Figure 1.4) and weight factors that are optimal in the sense that the error between the real letters and the reconstructed letters is minimal.



Figure 1.6: Applying a filter upon the results in Figure 1.5 which assigns a black color to a value between 0.85–1.15 and a white color to the remaining values.

In this thesis, we will apply model reduction to the numerical simulation of groundwater flow. We describe the implementation of the *Principal Component Analysis* and develop several numerical models that compute the weights for the principal components. Nevertheless, most of the concepts and mathematical elaborations are applicable to other fields of science.

1.3 Inverse modeling

The objective of *inverse modeling* is to find an optimal set of model parameters that yield the lowest objective function value. This function could be for example an elevation map of a mountainous area, see Figure 1.7. The optimal set of ‘model parameters’ will be in that case those coordinates that represent the lowest point of the valley, that is where the mountain cabin is. Now, imagine a hiker that wants to return fastest from a particular location on the mountain to the cabin. Unfortunately, he does not have an elevation map because that was too expensive. Moreover, there are poorly visible sights, so he needs to reevaluate whether he is going downhill, at every footstep. Imagine again that each footstep consumes the same amount of time that a single model simulation does. So, the advantage of a low-dimensional model, as described in the previous subsection, could really benefit the speed of the hiker. Moreover, there is no guarantee that the hiker arrives at the right mountain cabin. To ensure that the current cabin is the lowest one in the region, the hiker needs to be ‘dropped’ at different places.

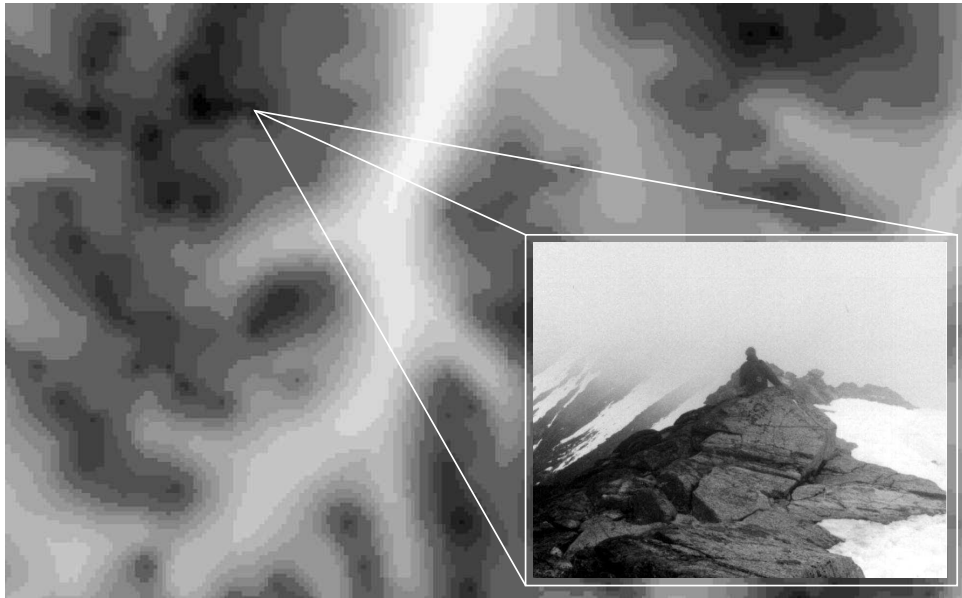


Figure 1.7: Inverse modeling is comparable to hiking through the mountains where one wants to return to the bottom of a valley as fast as possible but at low-visible sights, unfortunately. The figure shows the elevation, graduated from black to white, of the National Park Jotunheimen, Norway.

1.4 Research Objectives

The main objective of the research presented in this thesis was to develop a mathematical formulation for 3D groundwater flow to achieve a solution with a negligible loss of accuracy, at low-computational costs. More specifically, the primary objectives were to investigate existing upscaling techniques and model-reduction procedures and implement and qualify their performances specifically for high-dimensional 3D real-world cases. Model reduction gave a negligible loss in accuracy and gained a rate of efficiency that was rather encouraging. In contrast, the performance of upscaling techniques yielded rather poor results in terms of accuracy. Hence we focused on model reduction for the remaining part of this thesis. Therefore, our secondary objectives were to investigate and implement possibilities for model reduction and inverse modeling and describe their requirements and limitations. Furthermore, we have extended this by developing a generic approach to model reduction to reduce nonlinear (coupled) models.

1.5 Thesis outline

We start this thesis by stating some elementary definitions for the simulation of 3D groundwater flow. Furthermore, the second chapter is mainly devoted to upscaling methods for this type of model. It discusses a variety of methods for upscaling the hydraulic conductivity (i.e. a measure for the permeability of the subsoil) that are mentioned in literature. Moreover, it draws some attention to the upscaling of boundary conditions (e.g. the interaction between groundwater and surface water elements) which is a rather new approach. We applied both upscaling techniques to several synthetic 2D cases that differ in their permeability distribution and to a steady-state (i.e. time invariant) high-dimensional 3D real-world application of the entire province of Noord-Brabant, the Netherlands. However, we were not able to achieve reliable results in combination with high efficiencies. This was the main reason to focus specifically on model reduction in the remaining part of this thesis.

In Chapter 3 we introduce and elaborate several approaches for model reduction in groundwater flow. The most promising ones use several simulations with the original model (i.e. an empirical data which we call *snapshots*), to formulate a reduced model that operates in fewer dimensions and computes more efficiently. We illustrated the performance of three different methodologies in terms of efficiency and accuracy and applied them to a transient 3D real-world case of the entire province of Noord-Brabant, the Netherlands. Finally, we compare these techniques with another methodology that makes no use of snapshots.

A new element in model reduction is its incorporation in inverse modeling. Inverse modeling is known as computing the optimal set of model parameters that yield the highest reliability of a model. Since this process is rather time consuming we used in Chapter 4 a reduced model to find these optimal model parameters. We applied it to several transient synthetic 1D cases and a transient 3D real-world case near the village of Staphorst, the Netherlands.

Chapter 5 gives a generic formulation for a reduced model that can be used for inverse modeling of a variety of systems, such as atmospheric or oceanic systems. It yields a reduced model that is very time efficient and handles nonlinearities (i.e. a physical concept whereby the solution itself influences the parameters in the system) as well as models that are coupled (i.e. different models that exchange boundary conditions). We have applied the method for several model inversions of different 1D and 2D synthetic cases.

Finally, we improved the efficiency of the reduced model as described in Chapter 4 by including the estimate variable into the reduced model, and compared its performance and accuracy with the generic formulation as given in Chapter 5. We applied them to several 2D synthetic cases and one 3D real-world case of the entire province of Noord-Brabant, the Netherlands.

2

Limitations to Upscaling of Groundwater Flow Models

Abstract. Different upscaling methods for groundwater flow models are investigated. A suit of different upscaling methods is applied to several synthetic cases with structured and unstructured porous media. Although each of the methods applies best to one of the synthetic cases, no performance differences was observed if the methods were applied to a real 3D case. Furthermore, we focus on boundary conditions such as Dirichlet, Neumann and Cauchy conditions, that characterize the interaction of groundwater with e.g. surface water, recharge. It follows that the inaccuracy of the flux exchange between boundary conditions on a fine scale and the hydraulic head on a coarse scale causes additional errors that are far more significant than the errors due to an incorrect upscaling of the heterogeneity itself. Whenever those errors were reduced, the upscaled model was improved by 70%. It thus follows that in practice, whenever we focus on predicting groundwater heads, it is more important to correctly upscale the boundary conditions than hydraulic conductivity.

*This chapter is mainly adapted from Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Limitations to Upscaling of Groundwater Flow Models dominated by Surface Water Interaction, **Water Resources Research**, submitted 2006*

ALTHOUGH FASTER COMPUTERS have been developed in recent years, they tend to be used to solve even more detailed problems, e.g. examples of decreasing *problem scales* are pollution problems, and conservation of small nature conservation areas. One of the reasons is that the *information scale* has increased enormously. For example, the online available database in the Netherlands contains $\approx 400,000$ drillings ($\approx 10/\text{km}^2$) of the subsoil (0–100 m). This makes it possible to define an accurate description of the geology on a local scale (< 250 m). Moreover, through modern observation techniques (e.g. landuse by satellite images, digital-terrain-models obtained by laser altimetry) we are able to map and level the surface water most accurately (< 25 m), see Figure 2.1.

A recent development is that detailed numerical models are build that address small problem scales on a regional scale. They can be used for optimizing management problems that consider regional water problems. In many cases this will yield enormous models that can not be resolved within acceptable time constraints. It is worthwhile mentioning that the time efficiency of a simulation model can be

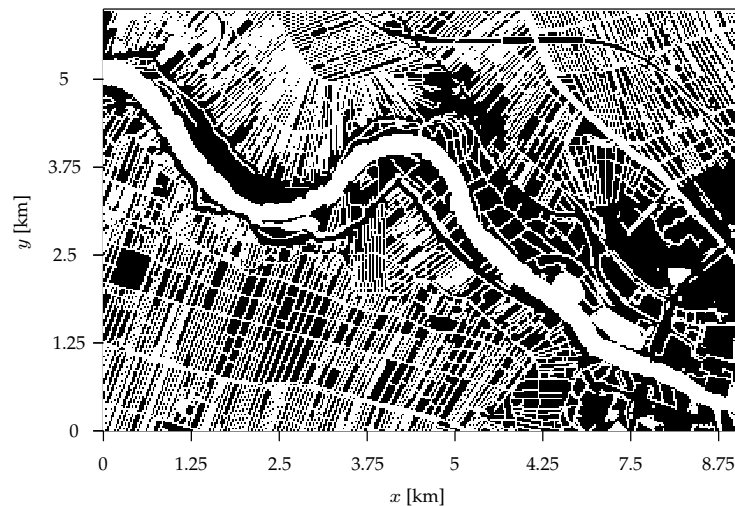


Figure 2.1: Detailed layout of the surface water elements (white) in a ‘polder’ region (below sea level) in the Netherlands. Resolution of the grid is 25×25 m.

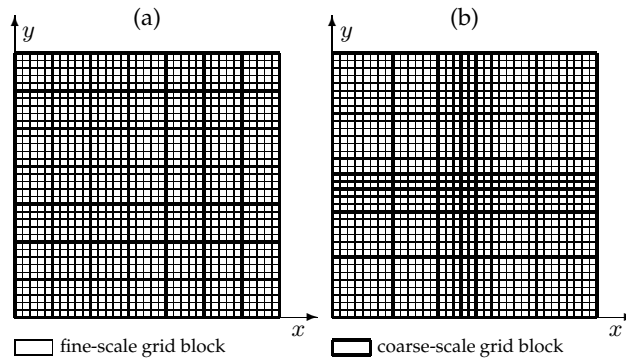


Figure 2.2: Schematic showing fine- and coarse-scale grid blocks for (a) uniform upscaling of the entire model domain, and (b) non-uniform upscaling of parts of the domain.

increased with (1) a more time-efficient solver [Mehl & Hill, 2001], and/or (2) a reduced model that is capable of simulating the important behavior of the original model [Vermeulen *et al.*, 2004a,b]. The first technique requires extra computer resources (i.e. memory and disk space) and becomes impracticable whenever the model under consideration contains too many nodes. The second method is restricted because it assumes a limited application of the model, e.g. it can not compute other scenarios than those that are closely related to the ones that were used to compute the reduced model.

A coarsening of grid blocks in each direction (*upscaling*) reduces the computational times and disk space significantly because they relate quadratically to the coarsening. In this chapter we refer to this type of coarsening as *uniform upscaling* because the entire model domain is subject to coarsening, see Figure 2.2a. However, it reduces the overall scale of detail, and if that is not desired, the alternative is to keep the fine-scale grid in the area of interest, and coarsen only those areas that are further away from it. We refer to this as *non-uniform upscaling*, see Figure 2.2b. For both of these coarsening approaches we need to obtain model parameters that are referred to as *upscaled, effective, equivalent* and/or *homogenized* parameters. They are defined such that they reproduce the global behavior of the aquifer while keeping the local behavior as close as possible to ‘reality’. The term reality is here referred to as *that* model solution that would have been obtained whenever we were able to compute the entire model without coarsening. Unfortunately, the coarsening of grid blocks is not without any consequence because detail will be lost that relates to boundary conditions and subgrid heterogeneity of geology.

Excellent reviews on upscaling geology and hydraulic parameters are given by Renard & de Marsily [1997]; Farmer [2002]; Sánchez-Villa *et al.* [1995]; Neuman & Federico [2003], among others. However, this literature does not provide a comparison between different upscaling methods on a range of different hydraulic transmissivity distributions. Moreover, many papers are devoted to upscaling of hydraulic transmissivities in reservoir engineering [Durlofsky, 1991; Durlofsky *et al.*, 1997; Chen

et al., 2003; Wen *et al.*, 2003; Holden & Nielsen, 2000]. These type of models are often focused on an accurate prediction of oil production at the boundaries of the reservoir, and model impulses are rather simple as they are formed by an injector and a producer. Moreover, the quality of these upscaled models is usually assessed by comparing breakthrough times for two-phase flows (i.e. the oil pressure and water saturations) for specified grid blocks with well injectors and well producers. In groundwater hydrology, however, we have a densely distributed system of source terms that influence the system throughout the model domain. To ensure that the flow rate for these source terms is correct, we urge to have local agreement. Our statement is that one can not simply upscale the surrounding and assume that these source terms are met. Since this is common practice, especially in hydrologic modeling, this chapter should be an eye-opener for those that construct models with an overwhelming number of (nonlinear) source terms.

The objective of this chapter is to present a review of existing upscaling techniques for hydraulic transmissivity and test their performances to single-phase groundwater flow which is discretized by means of finite-differences (i.e. rectangular volumes). It tries to answer for which type of media which upscaling method performs best, and if there is a single method that performs 'best' in practice. Furthermore, this chapter addresses the issue of upscaling boundary conditions and quantifies the importance of this.

This chapter is structured as follows: Section 2.1 briefly gives the governing equation that describes three-dimensional groundwater flows for the fine and coarse blocks. It also defines several error criteria for quantifying the performances of the upscaling methods. Section 2.2 is devoted to upscaling techniques for the hydraulic transmissivity and illustrates their performances to several synthetic cases. Section 2.3 describes the consequences of block coarsening on Dirichlet, Neumann and Cauchy conditions. Section 2.4 shows the overall performance of the upscaling methodologies to a large-scale real-world application. Finally, conclusions and recommendation are formulated in Section 2.5.

2.1 Governing Equations and Error Measures

2.1.1 Fine-Scale Equations

The equation that describes steady state incompressible, single-phase groundwater flow in 3D porous media can be represented by the combination of the equation of continuity ($\nabla \cdot \mathbf{u} = -q$) and Darcy's Law ($\mathbf{u} = -\mathbf{K} \nabla \phi$) [Strack, 1989]

$$\nabla \cdot [\mathbf{K}(\mathbf{x}) \nabla \phi] = q(\phi) ; \mathbf{x} \in \Omega, \quad (2.1)$$

where ϕ [L] is the hydraulic head, $\mathbf{u}$ is the Darcy velocity [LT^{-1}] (also called *specific discharge*), q is a source/sink term [T^{-1}] that can be hydraulic head dependent, and $\mathbf{K}$ is the hydraulic tensorial permeability [LT^{-1}] for a fine-scale point $\mathbf{x}$ subject to

domain Ω . The tensor can be represented by its matrix of components

$$\mathbf{K} = \begin{bmatrix} k^{xx} & k^{xy} & k^{xz} \\ k^{yx} & k^{yy} & k^{yz} \\ k^{zx} & k^{zy} & k^{zz} \end{bmatrix}. \quad (2.2)$$

From a macroscopic point of view [Bear, 1972] it can be shown that $\mathbf{K}$ is symmetric ($k^{xy} = k^{yx}$, $k^{xz} = k^{zx}$, $k^{yz} = k^{zy}$) and positive definite ($k^{xx}k^{yy}k^{zz} \geq (k^{xy})^2, (k^{xz})^2, (k^{yz})^2$; $k^{xx}, k^{yy}, k^{zz} > 0$). This assures that energy is always dissipated during flow. However, most of the models assume quasi 3D flow, i.e. the head gradient does not vary vertically within aquifers and does not vary horizontally within aquitards (Dupuit-Forchheimer flows). As a consequence of this, the tensor components $k^{xz} = k^{zx} = k^{yz} = k^{zy} = 0$ and the model is simulated by the quantity $\mathbf{T}$ [L^2T^{-1}] which is the transmissivity defined as $\mathbf{T} = \mathbf{K}h$, where h [L] is the thickness of the aquifer/aquitard. A solution for these Dupuit-Forchheimer flows can be obtained by means of a grid-centered finite-difference discretization [McDonald & Harbaugh, 1988], elaborated in Appendix 2A.

2.1.2 Coarse-Scale Equations

It has been shown by Bourgeat [1984] under certain conditions (a uniform flow through fractured media within an infinite domain), that the coarse-scale equation for the hydraulic head is of the same form as the fine-scale equation, but for which the fine-scale hydraulic permeability tensor $\mathbf{K}$ is replaced by the coarse-scale hydraulic permeability tensor $\underline{\mathbf{K}}$. This in itself is a significant finding, as a homogenized version of a partial differential equation is not necessarily of the same form as the original equation (e.g. the homogenized version of the Stokes equation for flow in dilute porous media is the Brinkman equation). The coarsening itself can be given by

$$\mathbf{x} = \xi \cdot \underline{\mathbf{x}}; \mathbf{x}, \underline{\mathbf{x}} \in \Omega, \quad (2.3)$$

where ξ is the upscaling factor in two directions (x, y), and $\underline{\mathbf{x}}$ is a coarse-scale point within the identical domain Ω . We underline each coarse-scale variable to distinguish it from its fine-scale equivalents. Furthermore, $\underline{\mathbf{K}}$ itself is positive definite and symmetric, just as the fine-scale hydraulic transmissivity [Mei & Auriault, 1989]. An important aspect of coarsening is that $\underline{\mathbf{K}}$ becomes often anisotropic, even if the original hydraulic transmissivity $\mathbf{K}$ is isotropic. Recently, numerical methods became available that cope with the off-diagonal components in a tensor in a finite-difference scheme [Aavatsmark et al., 1996; Lee et al., 1998; Anderman et al., 2002], see Appendix 2B and the greybox on page 30. A final remark is that we want to reproduce the fine-scale solution with a coarse-scale equation most optimally. The discussion whether the upscaled hydraulic parameters reflect the intrinsic property of the medium is not the purpose of this chapter.

Numerical Model Schemes and Intrinsic Hydraulic Transmissivity

Numerical simulation of groundwater flow models involves partitioning of aquifers into meshes of rectangles (**finite differences**) as done in e.g. MODFLOW [McDonald & Harbaugh, 1988] or meshes of triangles or rectangles (**finite elements**) as done in e.g. MICROFEM [Hemker & Nijsten, 1996]. An exception to these is the **Multi-Layered Analytical Element Method** [Strack, 1989] which devides the model domain into a collection of analytical elements instead. In this thesis we use meshes of rectangles (**grid blocks**) which gives us two options: (1) we compute the hydraulic head within the mesh at grid nodes (**node-centered scheme**) or (2) at grid centroids (**grid-centered scheme**). The node-centered scheme defines the hydraulic conductance between grid blocks as the **arithmetic mean** of transmissivity. The grid-centered scheme uses the **harmonic mean** and the ‘true’ hydraulic conductance (**intrinsic hydraulic transmissivity**) will be probably somewhere in between [Renard & de Marsily, 1997]. For a checker-board medium for which the intrinsic hydraulic conductance is known as the **geometric mean** [Warren & Price, 1961], we observe quit significant differences between the two schemes, see Figure 2.3. The node-centered scheme is known as a **upper-bound method** [Penman, 1988]) as it overestimates the hydraulic conductance, the grid-centered scheme is known as a **lower-bound method** [Duvaut & Lions, 1976]) because it underestimates the hydraulic conductance. The differences between the two schemes vanishes whenever we refine the mesh infinitely. Zijl & Trykozko [2001] proposed to use both schemes for numerical simulations to capture the **intrinsic hydraulic transmissivity** of the medium. Nevertheless, we focus only on the upscaling of hydraulic transmissivity for lower-bound methods, because they are most frequently used. Moreover, the discrepancy between the schemes becomes irrelevant when the ratio between hydraulic transmissivities within a coarse grid block decreases, see Figure 2.4.

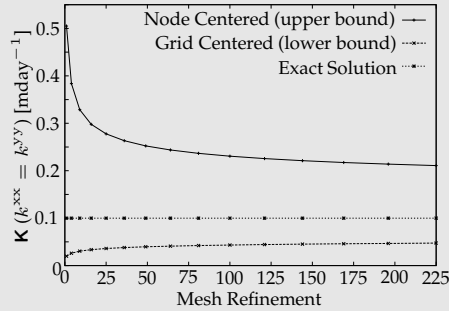


Figure 2.3: The graph shows the hydraulic permeability $\mathbf{K}$ for a checker-board media versus the refinement of the mesh for a grid-centered and node-centered finite-difference scheme in which $\mathbf{K}_1 = 0.01 \text{ mday}^{-1}$ and $\mathbf{K}_2 = 1 \text{ mday}^{-1}$.

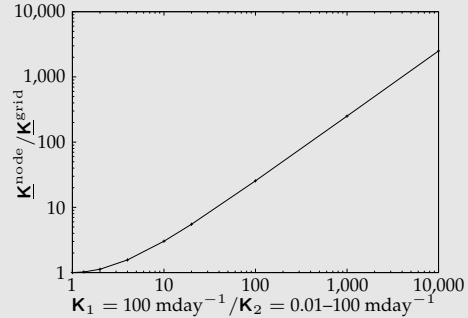


Figure 2.4: The graph shows the ratio of the coarse-scale hydraulic permeability $\underline{\mathbf{K}}$ obtained by a node- and grid-centered scheme versus the ratio of hydraulic permeabilities $\mathbf{K}_1/\mathbf{K}_2$ within the checker-board media.

2.1.3 Error Measures

A complete equivalence, however, between the fine and coarse-scale model is never possible because the grid coarsening introduces numerical errors simply because the degrees of freedom are reduced by ξ^2 , and the flow follows a different ‘path’.

Ames [2002] stated that to prevent numerical dispersion, the size of the grid blocks should be about the inverse of the gradient of the logarithm of transmissivity. This error is negligible for the hypothetical case of uniform flow that is aligned to a network. Therefore, several methods align their meshes to the local flow conditions [Cao & Kitanidis, 1999; Mansell *et al.*, 2000]. However, an accurate and efficient implementation for a large-scale, multi-layered system is often limited because the final mesh becomes a mixture of concessions that will not guarantee an overall improvement. In general, however, this error (numerical dispersion) will be smaller than other errors discussed in the following sections. To quantify the error we use three criteria:

1. **Conservation of the Hydraulic Head:** ultimately, the upscaled hydraulic head $\underline{\phi}$ should be equal to the spatial averaged hydraulic head on the fine scale:

$$\delta^\phi(\underline{\mathbf{x}}) = \|\underline{\phi}(\underline{\mathbf{x}}) - \frac{1}{V(\underline{\mathbf{x}})} \int_{V(\underline{\mathbf{x}})} \phi(\mathbf{x}) d\mathbf{x}\|, \quad (2.4)$$

where $V(\underline{\mathbf{x}})$ corresponds to a coarse grid block volume. However, we accept larger errors for grid blocks that describe high variances on a fine scale, and a relative error ε^ϕ can be defined as:

$$\varepsilon^\phi(\underline{\mathbf{x}}) = \frac{\delta^\phi(\underline{\mathbf{x}})}{2\sigma[\mathbf{x} \in V(\underline{\mathbf{x}})]} \cdot 100\%, \quad (2.5)$$

where σ is the standard deviation of the fine-scale hydraulic heads within a coarse grid block.

2. **Conservation of Continuity:** this requirement is especially important in transport problems in which velocity plays a dominant role. It prescribes that the upscaled Darcy flow $\underline{\mathbf{u}}$ equals the averaged Darcy flow of the corresponding fine-scale model [Renard & de Marsily, 1997]. Again, we define it as a relative error:

$$\varepsilon^u(\underline{\mathbf{x}}) = \frac{\left\| \underline{\mathbf{u}}(\underline{\mathbf{x}}) - \frac{1}{V(\underline{\mathbf{x}})} \int_{V(\underline{\mathbf{x}})} \mathbf{u}(\mathbf{x}) d\mathbf{x} \right\|}{\left\| \frac{1}{V(\underline{\mathbf{x}})} \int_{V(\underline{\mathbf{x}})} \mathbf{u}(\mathbf{x}) d\mathbf{x} \right\|} \cdot 100\%. \quad (2.6)$$

3. **Conservation of Dissipation:** this criterion is the equality of energy dissipated by the hydraulic head [Bø, 1994]. It is defined as the rate of dissipation of mechanical energy per unit weight of fluid. The relative dissipation error is given by:

$$\varepsilon^d(\underline{\mathbf{x}}) = \frac{\left\| [-\nabla \cdot \underline{\phi} \underline{\mathbf{u}}](\underline{\mathbf{x}}) - \frac{1}{V(\underline{\mathbf{x}})} \int_{V(\underline{\mathbf{x}})} [-\nabla \cdot \phi \mathbf{u}](\mathbf{x}) d\mathbf{x} \right\|}{\left\| \frac{1}{V(\underline{\mathbf{x}})} \int_{V(\underline{\mathbf{x}})} [-\nabla \cdot \phi \mathbf{u}](\mathbf{x}) d\mathbf{x} \right\|} \cdot 100\%. \quad (2.7)$$

Unfortunately, energy is lost when discretizing the flow domain, so that the numerical dissipation is always larger than the exact dissipation for discrete block-centered schemes. This leads to a $\underline{\mathbf{K}}$ that should be larger than $\mathbf{K}$ for isotropic homogeneous domains [Sánchez-Villa *et al.*, 1995; Duvaut & Lions, 1976], see Figure 2.10.

2.2 Upscaling Hydraulic Parameters

This section describes the most common upscaling techniques as mentioned in literature, for the hydraulic permeability $\mathbf{K}$, and the hydraulic conductances between grid blocks. They can be roughly categorized into (semi) problem dependent and problem independent techniques.

2.2.1 Power Averaging (GEO method)

It's known, from analytical solutions, that the upscaled hydraulic permeability for flow parallel to the strata is the arithmetic mean of the fine-scale hydraulic permeabilities, and for flow perpendicular to the strata, the harmonic mean, e.g. Wiener [1912]; Matheron [1967], among others. Journel *et al.* [1986] proposed the general equation

$$\underline{k}(\mathbf{x}) = \left\{ \frac{1}{V(\mathbf{x})} \int_{V(\mathbf{x})} [k(\mathbf{x})]^f dV(\mathbf{x}) \right\}^{1/f}, \quad (2.8)$$

where $f = -1$ corresponds to the harmonic mean, $f = 1$ to the arithmetic mean. Desbarats [1992] demonstrated that, for moderately heterogeneous two-dimensional systems, the upscaled hydraulic transmissivity could be estimated accurately by means of a spatially optimized f that best fits numerical simulations. However, one of the few exact results, for two-dimensional flow, is the rule of geometric averaging ($f = 0$) [Matheron, 1967]:

$$\underline{k}(\mathbf{x}) = \exp \left\{ \frac{1}{V(\mathbf{x})} \int_{V(\mathbf{x})} \ln [k(\mathbf{x})] dV(\mathbf{x}) \right\}. \quad (2.9)$$

The rule of geometric averaging is only satisfied for a permeability that has an isotropic log-normal distribution or a checkerboard binary design [Warren & Price, 1961], see Figure 2.3. Moreover, it holds only for uniform (parallel) flow fields and is not satisfied for example for radial flow that is discretized by rectangles.

2.2.2 Darcian Methods (HNF method)

This approach is like laboratory measurements of local properties of the porous medium, but by use of simulated experimental results. This method was first introduced by Warren & Price [1961], and after all, it simply consists of numerically simulating the experiment of Darcy upon a coarse grid block isolated from the total model domain, see Figure 2.5. The advantage of Darcian methods is that the *shape*

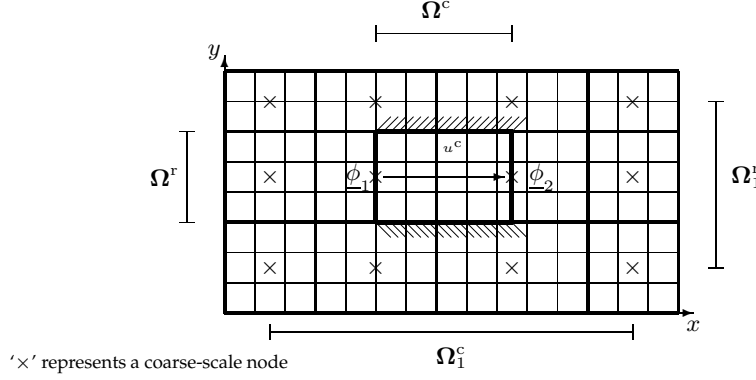


Figure 2.5: Local model domain $\Omega^c \times \Omega^r$ for the computation of a local averaged Darcian flux u^c between two Dirichlet conditions with coarse grid block pressures ϕ_1 and ϕ_2 and a 'sealed-off' boundary conditions perpendicular to those pressures. The local domain $\Omega_1^c \times \Omega_1^r$ is given to account for one 'borderring' or 'jacket cell'.

of hydraulic transmissivities affects the final upscaled hydraulic transmissivity, that power averaging lacks (see Subsection 2.2.1).

The coarse-scale conductance $\underline{C}^c$ between adjacent blocks along the column direction would now be obtained by simulating a coarse grid block with domain $\Omega^c \times \Omega^r$ with 'sealed-off' boundary conditions along Ω^r and a constant pressure drop along Ω^c . The conductance will be the ratio between the averaged local flow and the averaged gradient in the coarse grid block [Rubin & Gómez-Hernández, 1990], so

$$\underline{C}^c(\mathbf{x}) = \frac{\langle u^c \rangle \Omega^r}{\langle \nabla \phi^c \rangle \Omega^c}. \quad (2.10)$$

The averaged quantities along the column direction are defined as:

$$\langle u^c \rangle = \frac{1}{V(\mathbf{x})} \int_{V(\mathbf{x})} u^c(\mathbf{x}) d\mathbf{x} \quad (2.11a)$$

$$\langle \nabla \phi^c \rangle = \frac{1}{V(\mathbf{x})} \int_{V(\mathbf{x})} \nabla \phi^c(\mathbf{x}) d\mathbf{x}, \quad (2.11b)$$

with $\mathbf{x} \in \{\Omega^c \times \Omega^r\}$. The coarse-scale block conductance along the row direction $\underline{C}^r$ is derived analogously. Gómez-Hernández & Journal [1990] suggested that 'jacket-cells' (i.e. extension of the flow region larger than the coarse block, see Figure 2.5) will improve the upscaled hydraulic transmissivity. It is possible that these extra 'borderrings' will improve the results [Wen et al., 2003], however, there is a simple example that illustrates the risk of using those 'jacket cells' or 'borderrings'. Consider a case in which a zero hydraulic transmissivity barrier divides a coarse grid block into two pieces and extends unto the boundary of the coarse grid block. Whenever no 'jacket-cells' were used, the upscaled hydraulic transmissivity be-

comes zero. If they, however, were used some flow will be allowed, and an incorrect non-zero hydraulic transmissivity results.

2.2.3 Homogenization (TEN & PTEN methods)

A disadvantage of Darcian methods (see Subsection 2.2.2) is that they yield an up-scaled hydraulic transmissivity that depends on the imposed boundary conditions. In contrast to this, homogenization is an upscaling method [Bensoussan *et al.*, 1978], that yields a homogenized hydraulic transmissivity that is independent of the chosen boundaries. These boundary conditions assume that the region under study is immersed in a large-scale pressure field and the system itself is surrounded by periodic replications of itself on all sides [Durlflosky, 1991]. It seems an unrealistic situation because no natural medium is periodic. However, there is no reason to believe that these *periodic* boundary conditions are less arbitrary than ‘sealed-off’ type, uniform type of boundaries, or effective flux boundary conditions [Wallstrom *et al.*, 2002].

Periodic boundary conditions prescribe specific correspondences between pressure and velocity on opposite faces of an isolated coarse grid block. This means that the fluxes through opposite boundaries should be equal and opposite, and the heads along that direction should be equal to each other minus a given drop in pressure. Perpendicular to this, the heads for opposing boundaries should be equal, see Appendix 2C. A model with these boundary conditions needs to be solved twice, (†) with a hydraulic gradient in the column direction, and (‡) in the row direction. From these solutions we compute the averaged velocities (Equation 2.11a) and pressure gradients (Equation 2.11b) and obtain the components of the tensorial hydraulic permeability by solving:

$$\begin{bmatrix} \langle \nabla \phi^c \rangle^\dagger & \langle \nabla \phi^r \rangle^\dagger & 0 & 0 \\ 0 & 0 & \langle \nabla \phi^c \rangle^\dagger & \langle \nabla \phi^r \rangle^\dagger \\ \langle \nabla \phi^c \rangle^\ddagger & \langle \nabla \phi^r \rangle^\ddagger & 0 & 0 \\ 0 & 0 & \langle \nabla \phi^c \rangle^\ddagger & \langle \nabla \phi^r \rangle^\ddagger \end{bmatrix} \begin{bmatrix} k^{xx} \\ k^{xy} \\ k^{yx} \\ k^{yy} \end{bmatrix} = - \begin{bmatrix} \langle u^c \rangle^\dagger \\ \langle u^r \rangle^\dagger \\ \langle u^c \rangle^\ddagger \\ \langle u^r \rangle^\ddagger \end{bmatrix}, \quad (2.12)$$

where $\langle \cdot \rangle^\dagger$ represents the average quantities obtained by the first simulation, and $\langle \cdot \rangle^\ddagger$ those from the second. A solution of Equation (2.12) yields always a symmetric and positive definite tensor [Durlflosky, 1991]. Instead of flux averaging, we can apply dissipation averaging and whenever they distinct less, it is a measure for the necessity of periodic boundary conditions [Bensoussan *et al.*, 1978].

Periodic boundaries are not completely generic as they assume periodicity for a block scale which is not equivalent to the intrinsic property of the medium. Again, ‘bordersings’ can be expected to provide an improved tensorial hydraulic permeability because the effects of larger-scale hydraulic transmissivity connectivity are accounted for [Wen *et al.*, 2003]. Nevertheless, the size of the grid block should essentially capture the intrinsic property of the medium (Representative Elementary

Volume REV). To approach this condition slightly, we introduce a different strategy (PTEN) by applying periodic boundary conditions to all fine-scale grid blocks and include 1 ‘jacket-cell’. We assume that the intrinsic property of the medium could be described by those 3×3 cells. In reality this REV size should be space dependent. The upscaled tensorial hydraulic permeability within a coarse-scale grid block becomes the geometric mean as defined in Equation (2.9).

2.2.4 Local-Global Method (LG method)

The Local-Global method [Chen *et al.*, 2003] is a slight adjustment to the previously mentioned methods (see Subsection 2.2.2 and 2.2.3). It can be seen as a minimization problem for which the upscaled hydraulic transmissivity minimizes the differences in the velocity fields generated by the coarse and fine scale [Holden & Nielsen, 2000]. The method can be briefly summarized by stating that the boundary conditions for the local model domains are extracted from subsequent iterations of the global solution. The local model domain is extended towards the centroid of the neighbouring grid blocks, and the global heads ϕ are projected on the fine-scale boundary domain, see Figure 2.5. Those boundary pressures ϕ^d for the local domain $\Omega_1^c \times \Omega_1^r$ are used to obtain pressures in between. We weigh them harmonically by their transmissivities, so the boundary pressure for the i^{th} column, between ϕ_1^d and ϕ_2^d along the x direction becomes

$$\phi_i^d = \phi_1^d + (x_i - x_1^d) \frac{\phi_2^d - \phi_1^d}{x_2^d - x_1^d} \frac{\sum_i T_i}{T_i}; \quad x_1^d < x_i < x_2^d, \quad (2.13)$$

where $\sum_i T_i$ is the total harmonic transmissivity between x_1^d and x_2^d . Chen *et al.* [2003] solved the local problem twice by assuming a global flow along the column direction (interpolating ϕ_i^d linearly in one direction and harmonically in the perpendicular direction), and a global flow along the row direction (reverse interpolation of ϕ_i^d). They stated that by doing so the upscaled hydraulic transmissivity will be representative for more different cases. The LG method is efficient whenever the effort put into the local flow simulations is less than a single simulation of the original fine-scale model. Chen *et al.* [2003] suggested therefore to apply only these simulations for areas with a significant drop in pressure.

2.2.5 Global-Local Methods (GL method)

As shown in the previous subsections, the upscaled hydraulic conductance is not unique and depends on the local flow conditions. So, the best thing we can do is use these local flow conditions to compute the upscaled conductances as the ratio between the averaged flux and averaged pressure gradient in each coarse-scale grid block (Equation 2.10). The upscaled parameters are optimal for the used boundary conditions and are, however, no guarantee that different flow conditions will be simulated correctly [Bierkens & van der Gaast, 1998]. This is the main disadvantage of this Global-Local strategy as it leads to the paradoxical situation where one has to know *a priori* the local solution on a global scale, to determine the upscaled

block hydraulic conductance. In this chapter, we assume that we have the ‘luxury’ of knowing the exact fine-scale solution and therefore the results obtained by this method should be interpreted as being the best possible for this method.

2.2.6 Vertical Hydraulic Transmissivity

The 3D domain is often discretized as quasi-3D, which means that stratified layered media are assumed. This is as a set of 2D models that exchange water vertically over intermediate layers with poor hydraulic transmissivity without horizontal flow components (Dupuit-Forchheimer flows). The assumption of quasi-3D flow behavior holds if the vertical transmissivity of the separating layers (aquitards) is very small compared to the horizontal transmissivity of the aquifers. For upscaling 3D flow, the amount of water that is vertically exchanged over a coarse grid block, should be equal to the total exchange in the fine-scale model, so

$$\underline{C}_i^l (\phi_i - \phi_{i+1}) = \int_{V(\mathbf{x})} [C_i^l (\phi_i - \phi_{i+1})] (\mathbf{x}) d\mathbf{x}, \quad (2.14)$$

where i denotes the model layer, and $\underline{C}_i^l$ is the vertical conductance between two overlaying aquifers i and $i + 1$ (Equation 2.25). Again, it follows that the upscaled vertical conductance is a function of the fine-scale solution. This could be easily implemented for the GL method. For the LG method we have extended the local model towards 3D to apply Equation (2.14). The other methods, however, compute the vertical conductance as

$$C_k^{zc} = \int_{V(\mathbf{x}^c)} C_K^{zz}(\mathbf{x}) d\mathbf{x} \quad (2.15)$$

which is only valid however, whenever $d(\phi_k - \phi_{k+1})/d\mathbf{x} \approx 0$.

2.2.7 Examples

This subsection describes the performance of the upscaling techniques mentioned in the previous subsections for different heterogeneous media. We denote the different techniques by their acronyms, given in their corresponding subsections.

Distributions of Hydraulic Transmissivity

We have defined three synthetic cases with different distributions of hydraulic transmissivities. All of these cases are two-dimensional (102 columns by 102 rows) and possess a Dirichlet condition around the entire model domain. In the middle of the model we have positioned an extraction well with rate $q = 100 \text{ m}^3 \text{ day}^{-1}$. Since upscaling is most sensitive to the amount of distortion of the flow direction compared to the axis of the network, we simulated a diagonal flow from the lower left corner unto the upper right corner, by applying Dirichlet conditions around the entire model domain. The first transmissivity field is that of a strong anisotropic medium, see Figure 2.6a. It yields a strong preferential flow that is rotated 45° clockwise to the axis of the network (Figure 2.7a). The second case, is a multi

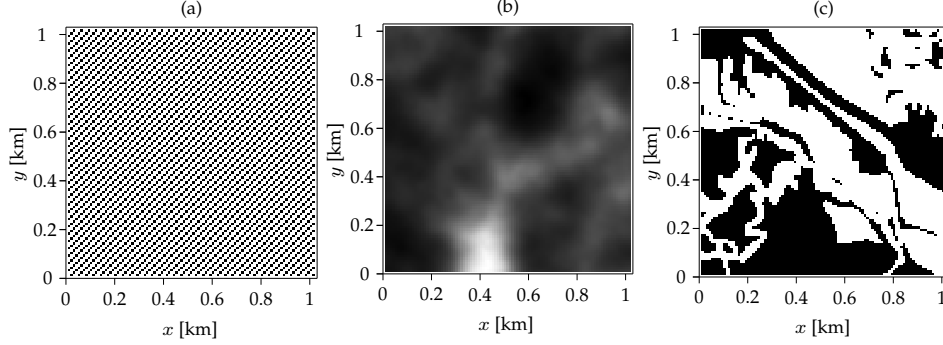


Figure 2.6: Different maps of the fine-scale hydraulic transmissivities; (a) an anisotropic medium with $T = 1 \text{ m}^2\text{day}^{-1}$ (black) and $T = 100 \text{ m}^2\text{day}^{-1}$ (white), (b) a multi Gaussian random field with lognormal distribution (<http://www.math.umd.edu/~bnk/bak/generate.cgi>) between $T = \exp(-2) \text{ m}^2\text{day}^{-1}$ (black) and $T = \exp(2) \text{ m}^2\text{day}^{-1}$ (white), and (c) a channelling system with a dual hydraulic transmissivity identical to (a).

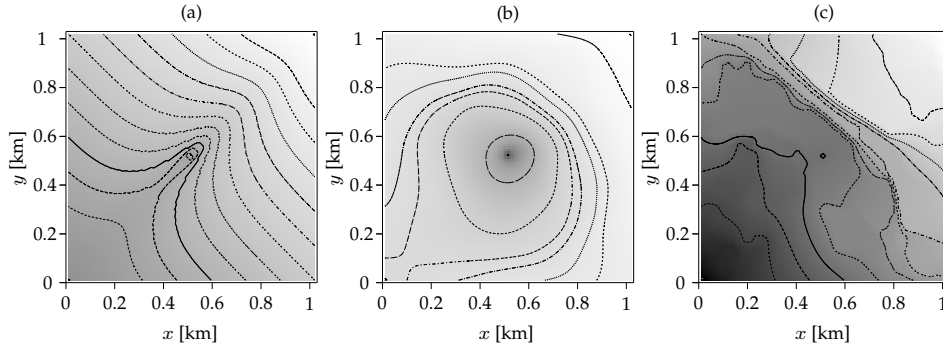


Figure 2.7: Different maps of the hydraulic head that migrates by a 45° rotation through (a) an anisotropic medium, (b) a multi Gaussian random field, and (c) a channelling system, see Fig 2.6a-c.

Gaussian random field (see Figure 2.6b) that has an irregular shaped drawdown (Figure 2.7b). The last synthetic case is that of a dual hydraulic transmissivity system that is derived from a real-world channelling system [Snepvangers & te Stroet, 2005] (Figure 2.6c). The flow field shows sharp irregular shaped discontinuities (Figure 2.7c).

Results

We have applied a uniform upscaling and increased the size of the grid blocks subsequently by the upscaling factors $\xi = 2\text{--}20$. We excluded the first and last row and column to avoid the artificial shift of Dirichlet conditions caused by the grid coarsening, see forthcoming Subsection 2.3.1. We have computed the relative-mean-absolute errors for the hydraulic head ($\langle \|\varepsilon^\phi\| \rangle$) for all these cases and plotted them in Figure 2.8a-c.

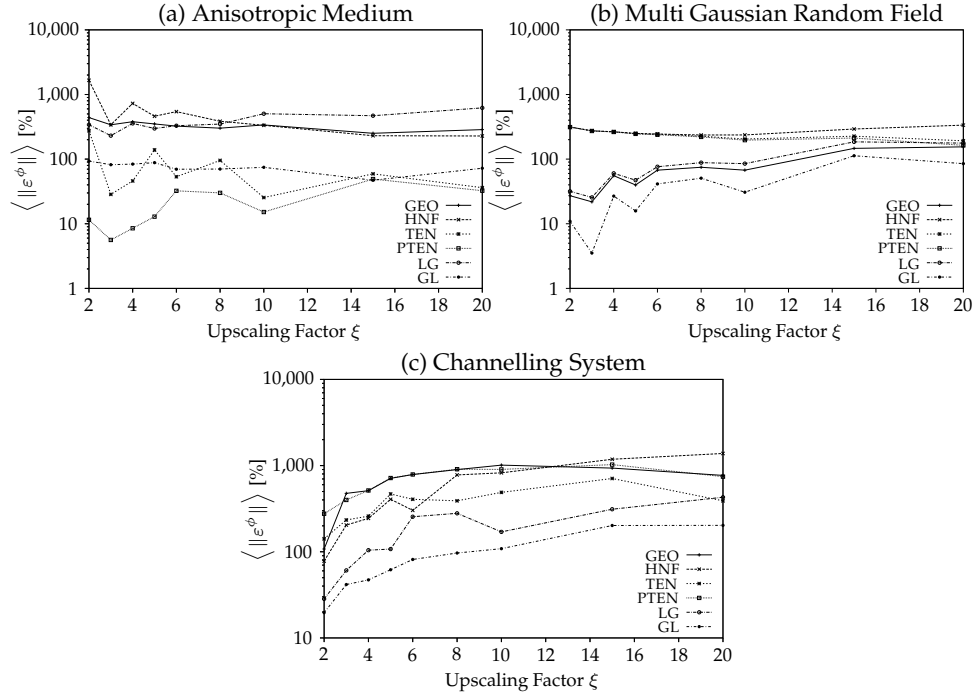


Figure 2.8: Graphs showing the relative-mean-absolute error $\langle \|\varepsilon^\phi\| \rangle$ for different upscaling techniques versus the upscaling factor ξ for (a) an anisotropic medium, (b) a multi Gaussian random field, and (c) a channelling system, see Figure 2.6a-c.

- Anisotropic Medium** The tensorial methods TEN and PTEN, yield the most accurate results because the medium is anisotropic and diagonal preferential flow is present, see Figure 2.8a. The PTEN method performs better than TEN because it captures the intrinsic periodicity of the medium (i.e. 3×3 grid blocks) instead of periodicity on a coarse block scale. The worst performance for this medium is given by the HNF and GEO methods because they can not cope with the anisotropic character of the flow field. The performance of HNF increases with ξ , which sounds rather contradictory but can be explained because the effects of the ‘sealed-off’ boundaries reduces as the isolated block increases. The GL method performs better because realistic boundary conditions are used but it can not cope with the diagonal preference of the flow. This is one of the reasons why the LG method differs so much from the GL method.
- Multi Gaussian Medium** The second case is most accurately upscaled by the GL, LG and GEO method, see Figure 2.8b. It is expected that the latter performs very well because the rule of geometric averaging is satisfied for a log-normal medium. In contrast to the periodic medium, the tensorial methods TEN, PTEN, and the Darcian method HNF were not able to yield reliable re-

sults. Because they perform rather similar, we conclude that the off-diagonal components of the tensors, for this type of medium, are relatively small compared to the diagonal components. Now they all compute the harmonic mean of transmissivity which is an underestimation of upscaled hydraulic transmissivity, see Figure 2.3.

- **Channelling Medium** This synthetic case is the only one where the HNF method performs rather well, see Figure 2.8c. A reason for this is that the system shows strong drops in pressure that are mainly aligned to the network; the harmonic mean is therefore a good approximation of the hydraulic conductance. At the same time, this phenomena causes the GEO method to perform poorly. The PTEN performs rather inaccurate because it assumed an REV size of 3×3 cells for the entire model domain. This is, however, incorrect for most parts of the model field though it is difficult to express the correct REV for each grid block. In this context, the TEN method performs slightly better because it uses an REV size which is based upon the coarse grid blocks, which is often larger than 3×3 cells. Again, the performance of the LG and GL methods are the best.

2.2.8 Conclusions

From the results obtained by the synthetic cases, it is obvious that the inaccuracy of a coarsened model increases rapidly with the upscaling factor ξ . The smallest upscaling factor of $\xi = 2$ results already in relative errors $\langle \|\epsilon^\phi\| \rangle > 10\%$ and only a limited number of upscaling algorithms manage to obtain reliable results, see Table 2.1.

Table 2.1: Maximal upscaling factors ξ that yield ‘acceptable’ performances ($\langle \|\epsilon^\phi\| \rangle \leq 100\%$) for different cases (Figure 2.6) and different upscaling algorithms for hydraulic transmissivity.

Heterogeneity	Upscaling Techniques					
	GEO	HNF	TEN	PTEN	LG	GL
Anisotropic			3,4,6–20	≤ 20		≤ 20
Multi Gaussian	≤ 10				≤ 10	≤ 10
Channelled		≤ 2			≤ 5	≤ 10

The anisotropic medium can be most successfully upscaled over a wide range of upscaling factors with tensors, but however, an anisotropic medium (e.g. complex crossbeddings, dipping layers not aligned to the coordinate system) that is perfectly periodic is hypothetical and not realistic whatsoever. The multi Gaussian distribution can be best simulated with the GEO method. The channelling system with the HNF method, though for very small upscaling factors. The LG and GL methods perform rather well for both media. Unfortunately, real-world media are a ‘mixture’ of the mentioned hydraulic transmissivity fields, and hence **there is no single ‘best’ upscaling method!** This is also confirmed by a practical application in Section 2.4. The best thing we can do is determining throughout the model domain which technique suits the local hydraulic transmissivity field most optimally. Moreover, the

network configuration should be a function of the hydraulic transmissivity variance [Garcia *et al.*, 1990] and/or existing periodic structures [Zijl & Trykozko, 2001] which is, however, often impossible or difficult to implement for groundwater flow models.

2.3 Upscaling Boundary Conditions

This subsection describes the consequences of grid coarsening to the behavior of Dirichlet (constant), Neumann (linear) and Cauchy (nonlinear) conditions.

2.3.1 Dirichlet Conditions

Dirichlet conditions are ‘open’ boundary conditions that exchange Darcy flow u^d with the world outside the model domain Ω . It is modeled by defining a hydraulic head on the boundary and the Darcy flow along the column direction becomes

$$u^d = C^c (\phi^d - \phi), \quad (2.16)$$

where ϕ^d represents the ‘fixed’ head on the boundary. Whenever we apply a grid coarsening we transform each coarse grid block into a Dirichlet condition whenever there is at least one fine-scale Dirichlet condition in it. Therefore the shape and the individual boundary parameters C^c, ϕ^d need to be replaced by averaged quantities, such that the upscaled Dirichlet conditions yield a similar Darcy flow $\underline{u}^d$ by a comparable flow field, so

$$\underline{u}^d \approx \sum_{i=1}^{n^d} u_i^d(\mathbf{x}) ; \mathbf{x} \in \underline{\mathbf{x}}, \quad (2.17)$$

where n^d is the number of fine-scale Dirichlet conditions within a single coarse grid block (e.g. $n^d = 3$ within Figure 2.9). To approximate $\underline{u}^d$, we define a local fine-scale model between the coarse grid block with fine-scale Dirichlet conditions and its adjacent grid block along the column direction, see Figure 2.9a. The model is determined by ‘sealed-off’ boundary conditions around its model domain $\Omega_1^c \times \Omega_1^r$ except for each row at the utmost right column, see Figure 2.9b. The head at the boundary is the averaged head for all fine-scale Dirichlet conditions within the coarse grid cell ϕ^d , minus an arbitrary chosen $\Delta\phi$. The coarse-scale conductance $\underline{C}^c$ along the column direction follows from

$$\underline{C}^c = \underline{u}^d \Delta\phi^{-1}. \quad (2.18)$$

The conductance along the row direction $\underline{C}^r$ is defined analogously. In this chapter we refer to these computations as *Dirichlet Simulations* and we observed an important improvement of the upscaled model in terms of dissipation, see Figure 2.10. This can be explained because we artificially shifted the node of the coarse grid block towards the centroid of gravity of the fine-scale Dirichlet conditions. However, because the actual flow at these Dirichlet conditions was radial, we overestimated the Darcy flow by assuming ‘sealed-off’ boundaries along the axis of the

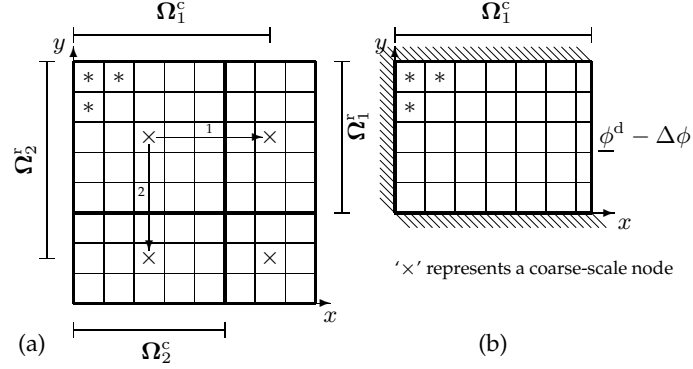


Figure 2.9: (a) Schematic showing the local model domains ($\Omega_1^c \times \Omega_1^f, \Omega_2^c \times \Omega_2^f$) for computing the coarse grid-block conductances aligned to (1) the column and (2) the row direction of the model network, subject to fine-scale Dirichlet conditions (*). (b) Schematic showing the isolated fine-scale model with 'sealed-off' boundaries and a drop in pressure $\phi^d - \Delta\phi$.

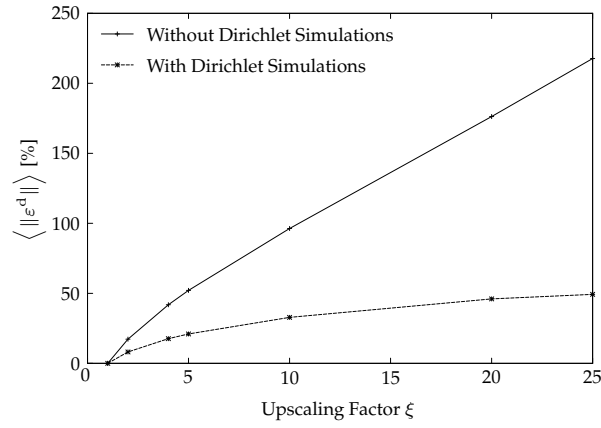


Figure 2.10: Graph showing the relative-mean error in dissipation $\langle \|\varepsilon^d\| \rangle$ versus the upscaling factor ξ with and without Dirichlet Simulations for a homogeneous medium ($T = 100 \text{ m}^2 \text{ day}^{-1}$).

local fine-scale model. An improvement can be expected whenever we impose an accurate representation of the hydraulic heads on the local boundary conditions [Chen *et al.*, 2003].

2.3.2 Neumann Conditions

Neumann conditions are head-independent point sources (e.g. precipitation, extraction wells) that guarantee a similar flux exchange for the fine-scale and coarse-scale model. Nonetheless, the actual position of a Neumann condition can be shifted in the coarse-scale model, simply because the node does not need to resemble the actual well position. We can reduce this error by distributing the strength of the source terms q among neighbouring coarse grid blocks, such that they resemble the centroid of gravity for the Neumann condition. We refer to these as *Neumann Displacements*.

A simple method is to compute a distribution index ω_i by a numerical experiment where the model domain is determined by the centers of the coarse-scale blocks that neighbour the specific Neumann condition, see Figure 2.11a-b. The amount of water that will be released from each corner determines the corresponding Neumann index ω_i , and the strength for each neighbouring coarse grid block becomes, $q_i = \omega_i * q$. The strategy shows some resemblance with the method proposed by Durlafsky *et al.* [2000], however, they related the hydraulic conductances to the Neumann condition instead.

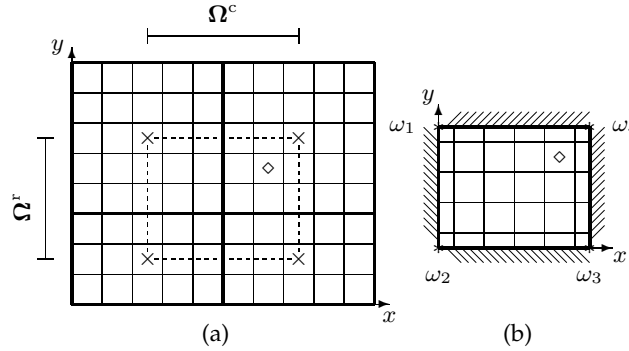


Figure 2.11: (a) Local model domain ($\Omega^c \times \Omega^f$) for encountering a displacement of a Neumann condition (◇) onto neighbouring nodes (×) within a coarse mesh. (b) Corresponding isolated fine-scale model used to compute the Neumann indices ω_i for each corner with Dirichlet conditions (*).

Applying Neumann Displacements showed to improve an upscaled model significantly (almost an order of magnitude), see Figure 2.12. More improvement can be expected whenever the local fine-scale model is extended with 'jacket-cells' and/or simulated in 3D. However, it should be mentioned that the method can have a drawback too because non-activeness within the local fine-scale model can cause an unbalanced distribution of the Neumann indices. In the worst case, no indices can be computed as all corner points are inactive on a fine scale.

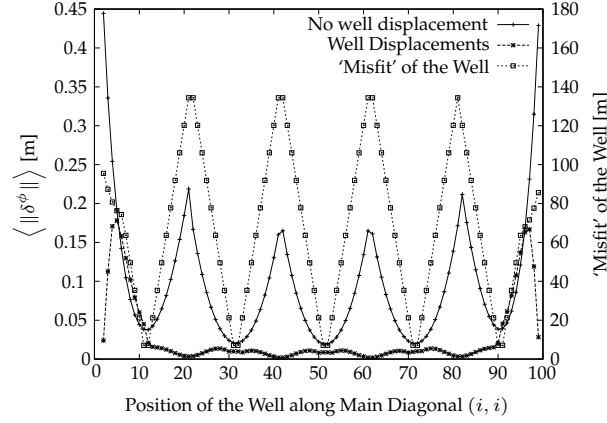


Figure 2.12: Graph of the mean-absolute error of the hydraulic head $\langle \|\delta\phi\| \rangle$ versus the minimal distance between the exact well location to the nearest coarse grid block, i.e. the 'misfit' (an upscaling factor $\xi = 20$ was used for the simulations).

2.3.3 Cauchy Conditions

The most important boundary condition in low-land areas such as the Netherlands is the surface-water system, see Figure 2.1. The flux q_i between surface water and groundwater is modeled as a Cauchy condition. Moreover, often this condition is nonlinear, e.g. a drainage flux is limited to a specified drainage level, and infiltration from rivers occurs often easier than exfiltration. They can generally formulated as:

$$\Delta h_i = h_i^f - \phi_i, \quad (2.19)$$

$$q_i = C_i^f(\Delta h_i) \Delta h_i, \quad (2.20)$$

where h^f is the level that determines eventually the value for C_i^f which is the conductance between the considered external force and the aquifer. Here it is a nonlinear function of Δh_i (Figure 2.13a). Most iterative solvers use a Picard iteration that sequentially reformulates and solves a linear system after reexamining Δh_i . Whenever we coarsen the grid blocks, this evaluation takes place between coarse- and fine-scale parameters and yields an over- or underestimation of the interaction, see $\Delta \underline{h}_i$ in Figure 2.13b. However, it is only possible to compute an averaged coarse scale $\underline{h}_j$ whenever a set of conductances C_i^f on the fine scale are equal, which is rarely the case. An alternative is to use the fine-scale solution (or a good approximation of it) ϕ_i and obtain a prior estimation of the coarse-scale solution $\hat{\phi}_j$. We can now approximate a corrected level h_i' such that the coarse-scale solution exchanges approximately the same flux within the external system, thus

$$\Delta h_i' = (\hat{\phi}_j - \phi_i) \quad (2.21)$$

$$\underline{q}_i = C_i \left[(h_i^f + \Delta h_i') - \underline{\phi}_i \right] \approx q_i \quad (2.22)$$

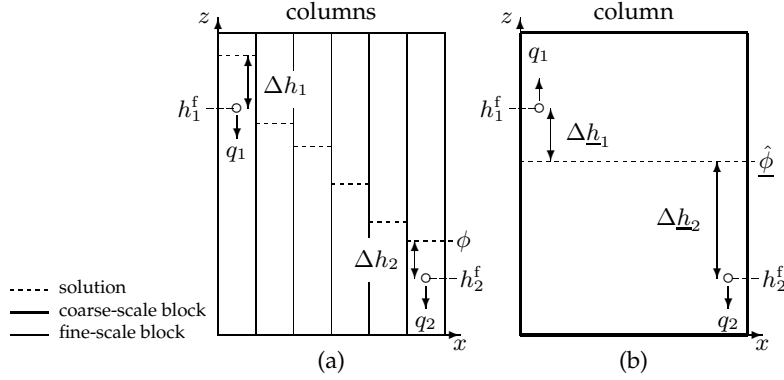


Figure 2.13: (a) Schematic showing the pressure differences for the computation of a drainage flux q_1, q_2 within a fine-scale model (a) and a coarse-scale model (b).

We refer to this method as *Cauchy Corrections* and we show in Section 2.4 that these yield significant improvements for an upscaled real-world model.

2.3.4 Conclusions

The errors of upscaling boundary conditions emerge simply because we coarsen a mesh, and the existing Dirichlet, Neumann and Cauchy conditions need to be grouped together. This yields an inaccuracy that could be reduced by computing average quantities that ensure that fluxes from boundary conditions are more-or-less equal to those within the fine-scale model. However, they yield average correction factors that are based upon prior simulations and it is questionable whether these average quantities remain valid for different scenarios. This is a new line of research that has not been reported on before. As will shown in the following section, errors caused by upscaling of boundary conditions are often much larger than those of hydraulic parameters.

2.4 Real-World Case

2.4.1 Introduction

The real-world case is a regional steady-state three-dimensional groundwater flow model that describes the entire region of the province of Noord-Brabant in the Netherlands ($\approx 10,700 \text{ km}^2$). The model consist of 320 rows, 535 columns and 9 model layers, so the total number of nodes is 1,540,800. Each grid block is a square with $\Delta x \equiv \Delta y = 250 \text{ m}$. The first model layer contains the influence of an intense surface water network that consists of ‘polders’ (i.e. a specific low-land area where a ‘fixed’ surface-water level is maintained) and natural drainage systems situated in higher areas. Absolute levels for these nonlinear boundary conditions were obtained by accurate laser altimetry. In Figure 2.14a we have depicted a map that shows the irregularity of the boundary conditions and the highly-detailed solution of the hydraulic head. The model is furthermore characterized by a detailed de-

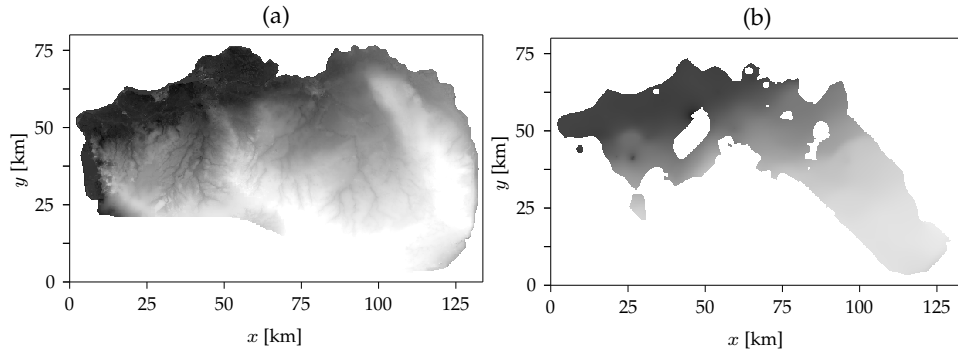


Figure 2.14: Map of the hydraulic head [m+MSL] within (a) model layer 1 and (b) model layer 9 for the real-world case. Grayshade ranges from black = 5 m-NAP to white = 30 m+NAP.

scription of the precipitation and evapotranspiration rate. These were obtained by combining data from rain gauge stations throughout the model domain with accurate land use classification from satellite images. The subsurface modeling was based upon thousands of drillings that were available for the region under consideration (<http://www.dinoloket.nl>). The statistics of the hydraulic transmissivities for the aquifers and the vertical resistances of the aquitards are given in Table 2.2.

Table 2.2: Statistics for the horizontal transmissivities and vertical resistances for the regional groundwater flow model of the province of Noord-Brabant in the Netherlands.

Layer	Transmissivities T [$\text{m}^2\text{day}^{-1}$] in the aquifers			
	Minimum	Maximum	Mean	St.Dev.
1	1.22	382.4	99.81	92.82
2	0.0	37,590	1,078	2,185
3	0.0	1,395,000	1,347	26,980
4	0.0	245,200	1,294	10,280
5	0.0	2,770	127.1	303.9
6	0.0	9,683	299.8	693.7
7	0.0	7,075,000	452.8	2,581
8	0.0	39,550	967.8	2,718
9	0.0	19,930	1,467	2,467
Layer	Vertical Resistances R [days] between aquifers			
	Minimum	Maximum	Mean	St.Dev.
1-2	1.29	206,600	3,232	8,007
2-3	5.37	307,700	16,040	37,910
3-4	0.47	98,430	2,660	10,020
4-5	0.45	56,150	702	2,552
5-6	1.65	304,900	1,930	13,490
6-7	0.47	1,042,000	9,187	55,730
7-8	0.47	854,700	13,260	67,410
8-9	2.99	30,880	1,274	3,631

2.4.2 Results

We have sequentially solved the real-world case for different upscaling factors $2 \leq \xi \leq 10$. We applied a variety of upscaling techniques for the hydraulic transmissivities (Subsection 2.4) and depicted the relative-mean error for the hydraulic head in Figure 2.15a. As we expected, the hydraulic transmissivity field of the real-world case is of such a ‘mixture’ of periodic media, multi Gaussian distributions and sharp discontinuities that all upscaling methods perform almost identical. In fact we should say: ‘just as bad’ because the smallest upscaling factor $\xi = 2$ yields an error of $\langle \|\varepsilon^\phi\| \rangle \approx 100\%$ already. This means that the mean error of the predicted head for this scaling operation, is just as large as twice the total standard deviation within the fine-scale solution (Equation 2.5). Furthermore, the relative-mean-averaged error in Darcy flow $\langle \|\varepsilon^u\| \rangle$, does not distinguish significantly between the upscaling methods. A small upscaling of $\xi = 2$ yields an error in Darcy flow of $\approx 30\%$, see Figure 2.15b. These bad performances are mainly caused by significant errors caused by upscaling of the boundary conditions such as Dirichlet, Neumann and Cauchy conditions. In Section 2.4 we suggested several techniques for the upscaling of these conditions and we have added these techniques

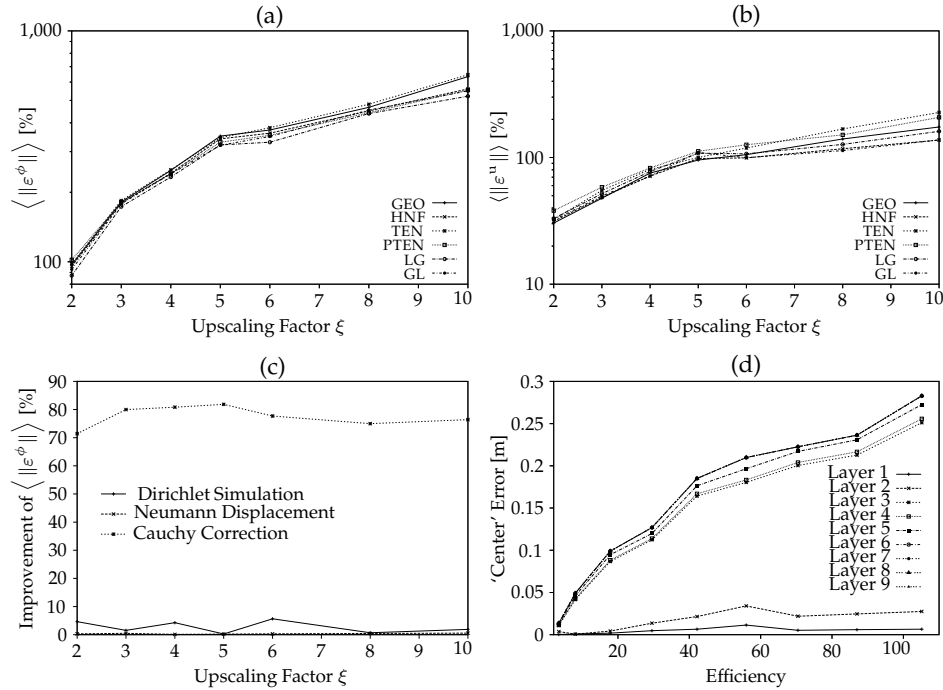


Figure 2.15: Graph (a-b) showing the inaccuracy of uniform upscaling for different upscaling techniques for hydraulic transmissivities, (c) shows the improvement of the error through corrections applied to Dirichlet, Neumann and Cauchy conditions, and (d) shows the error for different model layers at the center of a fine-scale mesh surrounded by a coarsened mesh.

and plotted their improvements in Figure 2.15c. Techniques that proved to work rather well for synthetical problems, appear to be of low relevance for our real-world case. The improvement given by *Dirichlet Simulations* is maximally 10% and *Neumann Displacements* have, however, a negligible effect. In contrast to these, the *Cauchy Corrections* have a significant effect and improve the coarsened model upto $\approx 70\text{--}80\%$! This again shows that reliable upscaling for groundwater flow models is only possible whenever we know the fine-scale solution. Since a fine-scale solution can not be computed, however, it can be approximated by subsequently solving a coarsened model with a local refinement (*non-uniform upscaling*, see Figure 2.2b), and collecting only the results from those fine regions to form a limited part of the global solution. It is important that the models overlap to determine the zone for which the solution can be copied. Again, it should be understood that this type of upscaling needs to be carried out with care because there is a clear trade-off between efficiency (ratio between the number of nodes in the upscaled model and the fine-scale model) and accuracy of the hydraulic head in the fine-scale part of the coarsened model. High efficiencies increase the error significantly, especially for aquifers that have a significant resistance to the boundary conditions (in our case that appeared to be $\approx 20,000$ days), see Figure 2.15d.

2.5 Conclusions and Recommendations

This chapter describes the error caused by upscaling of meshes in groundwater flow modeling that uses block centered finite-differences. The chapter consists of two parts:

- **Part I** describes the upscaling of hydraulic transmissivities by means of the most important and promising techniques mentioned in literature. For three synthetic cases that all differ in their heterogeneity distribution, it is clear that the different upscaling techniques have their preferences for which type of distribution they perform optimally.
- **Part II** is devoted to the error that evolves through upscaling of boundary conditions. This error emerges because we coarsen a mesh and reduce the number of degrees of freedom. Therefore the existing Dirichlet, Neumann and Cauchy conditions within a model need to be grouped. This yields an inaccuracy that was, for several synthetic cases, successfully reduced to acceptable levels by computing average quantities that ensure that the fluxes from the upscaled source terms are more-or-less equal to the fluxes in the fine-scale model.

We applied the above mentioned techniques to a real-world three-dimensional case and observed that the differences between the upscaling techniques for the hydraulic transmissivity were negligible. Since real-world media are a 'mixture' of different distributions of heterogeneity, **there is no single upscaling technique that**

performs best. In practice we should vary between different techniques throughout the model domain which is, however, impracticable and often impossible. Moreover, all techniques performed poorly. The techniques we introduced for the upscaling of Dirichlet and Neumann conditions were not relevant for this 3D case. The main improvements were obtained by use of the Cauchy corrections that increased the accuracy of the coarsened model for $\approx 70\text{--}80\%$! Nonetheless, it needs the entire fine-scale solution which is often not available.

So, to achieve a reliable coarsened model, we need to know the fine-scale solution. This implies a paradoxical situation where one has to know the solution to acquire it. A possible option is to use an a priori analysis of localized ‘small’ models to acquire this fine-scale solution. With these results an accurate global model can be constructed for the current application, however, it is questionable for which other applications than the one used to obtain an upscaled model from it, can be still used. This topic is a new line of research that has not been reported on before.

2A A Standard Seven-Point Stencil

A standard seven-point stencil can be used to solve a 3D groundwater flow model (Equation 2.1), and can be written as:

$$\begin{aligned} & C_{i-\frac{1}{2},j,k}^c [\phi_{i-1,j,k} - \phi_{i,j,k}] + C_{i+\frac{1}{2},j,k}^c [\phi_{i+1,j,k} - \phi_{i,j,k}] \\ & + C_{i,j,k-\frac{1}{2}}^r [\phi_{i,j,k-1} - \phi_{i,j,k}] + C_{i,j,k+\frac{1}{2}}^r [\phi_{i,j,k+1} - \phi_{i,j,k}] \\ & + C_{i,j,k-1}^l [\phi_{i,j,k-1} - \phi_{i,j,k}] + C_{i,j,k+1}^l [\phi_{i,j,k+1} - \phi_{i,j,k}] = q(\phi)_{i,j}, \end{aligned} \quad (2.23)$$

where $\phi_{i,j,k}$ denotes the hydraulic head in grid block i, j, k , and C^c, C^r are the harmonic hydraulic block conductances between adjacent grid blocks along the column and row direction, respectively. Along the column direction these are defined as:

$$C_{i+\frac{1}{2},j,k}^c = \frac{2T_{i,j,k}T_{i+1,j,k}\Delta y_j}{T_{i,j,k}\Delta x_{i+1} + T_{i+1,j,k}\Delta x_i}, \quad (2.24)$$

where $T [L^2T^{-1}]$ is the transmissivity defined as the aquifer thickness h times the hydraulic tensorial permeability along the column direction k^{xx} . The block conductance along the row direction C^r is defined analogously. The vertical conductance C^l is defined as

$$C_{i,j,k}^l = \frac{\Delta x_i \Delta y_j}{\left[\frac{1}{2}h(k^{zz})^{-1}\right]_{i,j,k} + R_{i,j,k} + \left[\frac{1}{2}h(k^{zz})^{-1}\right]_{i,j,k+1}}, \quad (2.25)$$

where $R [T]$ is the vertical resistance of an aquitard in between two aquifers. The mentioned elaboration is extensively described by McDonald & Harbaugh [1988] and implemented in the three-dimensional groundwater flow simulation program MODFLOW.

2B Finite Difference Simulation with Tensorial Hydraulic Transmissivities

Several investigators have developed approaches for the incorporation of tensorial hydraulic transmissivities into finite difference reservoir simulators. In this chapter we refer to these as nine-points stencils because we incorporate only tensors for the x, y plane and neglect the components that relate to the z direction, thus $k^{zx} = k^{zy} = k^{xz} = k^{yz} = 0$. Recent methods [Aavatsmark *et al.*, 1996; Edwards & Rogers, 1998] are based upon a flux-continuous approach and reduce to the usual seven-point stencil. A practical implementation of this is given by Lee *et al.* [1998]. In essence the method examines the fluxes in a ‘control volume’, that is a dual mesh in between four neighbouring grid blocks denoted by their hydraulic heads $\phi_1, \phi_2, \phi_3, \phi_4$, see Figure 2.16. However, if $\mathbf{K}$ is discontinuous between adjacent grid blocks, then linear interpolation of pressure is not valid across the cell faces because the pressure gradient is discontinuous. Continuous pressure are incorporated in the grid centered approximation by introducing a mean pressure $\bar{\phi}_i$ at the face that divides neighbouring cells. To remove the ambiguity, however, in computing the cell face fluxes, we assume that the pressure is constant within each quadrant and is determined by the grid block center pressures ϕ_i . By reexpressing the interface pressures $\bar{\phi}_i$ by the four cell centered pressures ϕ_i , the flux equation

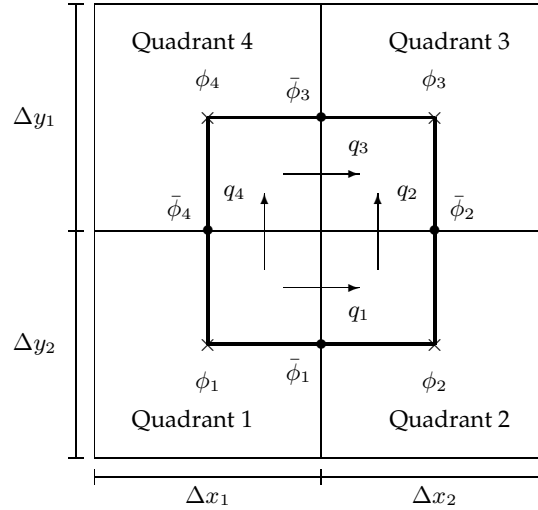


Figure 2.16: Control Volume and associated variables that were used to construct the flux-continuous finite-difference scheme (after Lee *et al.* [1998]).

within each quadrant can be written in a matrix-vector form as:

$$\mathbf{A}\mathbf{q} = -\Delta\phi, \text{ with} \quad (2.26a)$$

$$\mathbf{A} = \begin{bmatrix} \hat{k}_1^{yy} + \hat{k}_2^{yy} & -\hat{k}_2^{xy} & 0 & -\hat{k}_1^{xy} \\ -\hat{k}_2^{xy} & \hat{k}_2^{xx} + \hat{k}_3^{xx} & -\hat{k}_3^{xy} & 0 \\ 0 & -\hat{k}_2^{xy} & \hat{k}_3^{yy} + \hat{k}_4^{yy} & -\hat{k}_4^{xy} \\ -\hat{k}_1^{xy} & 0 & -\hat{k}_4^{xy} & \hat{k}_1^{xx} + \hat{k}_4^{xx} \end{bmatrix}, \quad (2.26b)$$

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T, \text{ and} \quad (2.26c)$$

$$\Delta\phi = \begin{bmatrix} \phi_2 - \phi_1 & \phi_3 - \phi_2 & \phi_3 - \phi_4 & \phi_4 - \phi_1 \end{bmatrix}^T, \quad (2.26d)$$

where $\hat{k}_1^{xx}$ denotes the xx component of the tensor within the first quadrant of the control volume. They are normalized as

$$\hat{k}^{xx} = k^{xx} \frac{\Delta y \Delta x^{-1}}{\det \mathbf{K}} \quad (2.27a)$$

$$\hat{k}^{yy} = k^{yy} \frac{\Delta x \Delta y^{-1}}{\det \mathbf{K}} \quad (2.27b)$$

$$\hat{k}^{xy} = \hat{k}^{yx} = k^{xy} (\det \mathbf{K})^{-1} \quad (2.27c)$$

$$\det \mathbf{K} = k^{xx} k^{yy} - (k^{xy})^2. \quad (2.27d)$$

This normalization is necessary because we define $\mathbf{B} = \mathbf{A}^{-1}$ and let

$$\mathbf{C}\phi = \mathbf{q}, \quad (2.28)$$

where the linear coefficients in $\mathbf{C}$ for each quadrant i are defined as

$$c_{i,1} = -b_{i,1} - b_{i,4} \quad (2.29a)$$

$$c_{i,2} = +b_{i,1} - b_{i,2} \quad (2.29b)$$

$$c_{i,3} = +b_{i,2} + b_{i,3} \quad (2.29c)$$

$$c_{i,4} = -b_{i,3} + b_{i,4}. \quad (2.29d)$$

This means that half of the total flow for an arbitrary reference location ϕ_0 can now be computed as the summation of the fluxes caused by pressure drops within the four neighbouring grid blocks in the control volume, so

$$q_i = - \sum_{l=1}^4 c_{i,l} (\phi_l - \phi_0). \quad (2.30)$$

The total flux Q out of the grid block is a summation of the fluxes towards the neighbouring eight grid blocks and is given by

$$Q_{i,j} = q_1^{(i,j)} + q_4^{(i,j)} + q_2^{(i-1,j)} - q_1^{(i-1,j)} - q_3^{(i-1,j+1)} - q_2^{(i-1,j+1)} - q_4^{(i,j+1)} + q_3^{(i,j+1)} \quad (2.31)$$

where $(\cdot)^{(i,j)}$ denotes the index of the grid blocks. The resulting block conductances will be a summation of

$$C_{i+\frac{1}{2},j}^c = +c_{1,2}^{(i,j)} + c_{4,2}^{(i,j)} - c_{4,3}^{(i,j+1)} + c_{3,3}^{(i,j+1)} \quad (2.32a)$$

$$C_{i,j+\frac{1}{2}}^r = -c_{2,2}^{(i-1,j+1)} - c_{3,2}^{(i-1,j+1)} - c_{4,1}^{(i,j+1)} + c_{3,1}^{(i,j+1)} \quad (2.32b)$$

$$C_{i+\frac{1}{2},j-\frac{1}{2}}^{cr} = c_{1,3}^{(i,j)} + c_{4,3}^{(i,j+1)} \quad (2.32c)$$

$$C_{i+\frac{1}{2},j+\frac{1}{2}}^{rc} = c_{3,2}^{(i,j+1)} + c_{4,2}^{(i,j+1)}. \quad (2.32d)$$

The eight flow terms yield a nine-point stencil that can not be solved by standard numerical solvers used in groundwater. *Edwards* [2000] developed an *operator splitting* of the finite-difference equations. The method is based upon the splitting of the nine-point stencil $\mathbf{A}^{(9)}$ into a pentadiagonal matrix $\mathbf{A}^{(5)}$ and a residual matrix. A consistent splitting is done at flux level whereby each flow is splitted into a normal-gradient component and a tangential-gradient component. It guarantees a diagonal dominance, thus the residual matrix $\mathbf{A}^{(9-5)}$ becomes

$$\mathbf{A}^{(9-5)} = \mathbf{A}^{(9)} - \mathbf{A}^{(5)}. \quad (2.33)$$

The system of hydraulic heads can now be solved by an iteration cycle whereby the diagonal flow components will be updated after each iteration ι , so

$$\mathbf{A}^{(5)}\phi^\iota = \mathbf{A}^{(9-5)}\phi^{\iota-1}. \quad (2.34)$$

The mentioned operator splitting has been recently implemented in the groundwater simulation model MODFLOW [McDonald & Harbaugh, 1988] to simulate horizontal anisotropic behavior [Anderman *et al.*, 2002].

2C Periodic Boundary Conditions

Suppose an isolated coarse grid block consists of 3 columns and 3 rows. To simulate periodic boundaries we have to add two extra columns and rows to express the correspondences between opposing boundaries, see Figure 2.17. The harmonic conductances between the columns 2–4 and between the rows 2–4 will be computed as usual according to Equation (2.24). The first constraint of periodic boundaries is that the horizontal flux inside the isolated block (between column 1 and 2) should be equal to the flux outside the opposing boundary (between column 4 and 5). We apply this constraint to all rows j at the left side of the isolated block (column 1), so the hydraulic head at that particular position $\phi_{1,j}$ is subject to

$$\phi_{1,j}C_{1+\frac{1}{2},j}^c = \phi_{2,j}C_{1+\frac{1}{2},j}^c + \left[(\phi_{n,j} - \phi_{n-1,j})C_{n-\frac{1}{2},j}^c\right], \quad (2.35)$$

where n expresses the total number of columns (five in Figure 2.17) and the conductance C^c between the boundary and the adjacent node is expressed as:

$$C_{1+\frac{1}{2},j}^c = \frac{4T_{2,j}\Delta y_j}{T_{2,j}\Delta x_2} \quad (2.36)$$

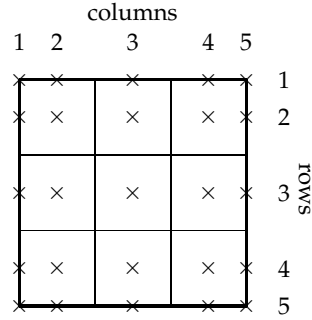


Figure 2.17: Illustration of an isolated coarse grid block to apply periodic boundary conditions to.

because $T_{1,j} = T_{2,j}$ and $\Delta x_1 = 0$ m. The elaboration along the row direction is analogously. The second constraint of periodic boundaries is that the pressure drop along the direction of flow is constant. This can be imposed on the system by simply stating

$$\phi_{n,j} = \phi_{1,j} + \Delta\phi, \quad (2.37)$$

where $\Delta\phi \neq 0$ along the current direction of flow and $\Delta\phi = 0$ perpendicular to this direction.

3

Low-dimensional Modeling of Numerical Groundwater Flow

Abstract. Numerical models are often used for simulating groundwater flow. Written in state-space form, the dimension of these models is of the order of the number of grid cells used and can be very high ($>$ million). As a result, these models are computationally very demanding, especially if many different scenarios have to be simulated. In this chapter we introduce a model reduction approach to develop an approximate model with a significantly reduced dimension. The reduction method is based upon several simulations of the large-scale numerical model. By computing the covariance matrix of the model results, we obtain insight into the variability of the model behavior. Moreover, by selecting the leading eigenvectors of this covariance matrix, we obtain the spatial patterns that represent the directions in state space where the model variability is dominant. These patterns are also called Empirical Orthogonal Functions (EOFs). We can project the original numerical model onto those dominant spatial patterns. The result is a low-dimensional model that is still able to reproduce the dominant model behavior.

This chapter is adapted from

- (1) Vermeulen, P.T.M, A.W. Heemink, & C.B.M. te Stroet, *Low-dimensional modeling of numerical groundwater flow*, **Hydrological Processes**, 18, 1487-1504, 2004
- (2) Vermeulen, P.T.M, A.W. Heemink, & C.B.M. te Stroet, *Reduced Models for Linear Groundwater Flow Models using Empirical Orthogonal Functions*, **Advances in Water Resources**, 27, 57-69, 2004

THE ACCESS TO DATA AND ITS ACCURACY, allows the use of numerical groundwater models with a huge amount of grid cells (often > 1 million). Such large and often transient models have long computational times. Apparently, the advantage of digital data changes into a disadvantage when we need to evaluate the model for many different cases, for example to optimize a pumping strategy. Especially in such a case, our objective should not be to formulate a very accurate mathematical model, but to formulate a model that is useful for the intended study. The main idea of a reduction method is to seek a simpler model, that is able to represent the numerical groundwater model for the intended study. From this perspective, *Newman* [1996a] stated: “We may wish to sacrifice some of the correctness of the model to make the equations easier to solve, or to allow a faster computation”. Such a simple model falls in the category of spectral methods that represent the solution to a problem as a truncated series of patterns and time-dependent coefficients (i.e. a reduced model structure). There are different approaches mentioned in literature to specify patterns, such as:

- Analytical patterns/functions can be used, such as Fourier functions, wavelets or polynomials. The disadvantage of such functions is that they have to be defined a priori and that they are non-economical regarding the amount of analytical patterns required [*Cazemier et al.*, 1998; *Hooimeijer*, 2001];
- More economical is the technique of computing patterns as an eigenvalue decomposition of the system matrix differential equations (‘modal’ decomposition). *Sahuquillo* [1983] was one of the first to use a part of the eigenvalues and eigenvectors of the system matrix, to compute the influence functions of pumpage on the piezometric head. To compute them within the entire domain the complete set of eigenvectors is necessary. When the number of cells is relatively high this can be computed efficiently with the Lanczos algorithm. This has been done for the groundwater flow equation [*Dunbar & Woodbury*, 1989], in simulating groundwater flow in a dual-porosity media [*Zhang & Woodbury*, 2000a], and in the solution of radionuclide decay chain transport in dual-porosity media [*Zhang & Woodbury*, 2000b]. A significant reduction of the dimension of the problem can be obtained if the system matrix contains relatively few dominant eigenmodes;

- The last technique uses specific model results (empirical data) to define the patterns. This data-driven technique produces a minimal set of patterns which span a specific data set optimally. In fact, the basis formed by the patterns is optimal in the sense that no other basis of the same dimension has a smaller mean-square error [Holmes *et al.*, 1996]. The EOF analysis has been used for large data sets to find the dominators to depict the modes of variation [Reyment & Jöreskog, 1993] and, caused by the limited number of patterns, the behavior of a complex system can be analysed more easily. In Ghanem [1998], these patterns are used to model the heterogeneity in random porous media as a superposition of scales of heterogeneity that are statistically uncorrelated, and in Hooimeijer [2001] they were used to represent a reduced morphological model. In different fields of science, this pattern identification technique is also called Coherent Structures (CS), Principal Component Analysis (PCA), Common Factor Analysis (CFA) or Proper Orthogonal Decomposition (POD).

Hooimeijer [2001] gave an extensive overview of the existing pattern identification techniques (data driven and analytical) and selected the EOF technique as being most suitable to develop a reduced morphological model. The original model was not known explicitly and the best results were obtained with a linear autoregressive model in which the EOFs were multiplied by coefficients. These coefficients were fitted to the data, however, Hooimeijer [2001] stated: “It seems to be more elegant and natural to project the original equations directly onto the EOFs. In order for this method to work, the entire equation set of the original model must be written in matrix format.” In this chapter we describe two different methodologies that fulfill that condition; these are

1. the *State-Space Projection Method* (SSPM) which is a straightforward projection of the state-space formulation of system equations written in matrix format [Vermeulen *et al.*, 2001, 2002]. Although the projection consumes more time to complete, it has an advantage from a numerical point of view [Vermeulen *et al.*, 2004a]; We present additionally the *Alternative State-Space Projection Method* (SSPM^a) which can be seen as an intermediate form between the SSPM and the following method.
2. a more efficient and sophisticated method known as the *Galerkin Projection Method* (GPM). It substitutes a reduced model structure into the partial differential equation (PDE) and projects the outcome on the patterns. This method is not new and most of it has been mathematically proven by Newman [1996a]. The technique has been applied in a wide range of scientific fields, such as turbulence and image processing [Sirovich, 1987], rapid thermal chemical vapor deposition [Adomaitis, 1995], in fluid dynamics for a lid-driven cavity with a rotating rod [Hoffman Jørgensen & Sørensen, 2000], in flow reactors governed by nonlinear equations [Park & Cho, 1996], and heat transfer problems [Park *et al.*, 1999]. In groundwater hydrology, this projection technique has been

described and applied by Vermeulen *et al.* [2003, 2004b] and in petroleum engineering by Heijn *et al.* [2004].

We present these types of projection strategies to construct a low-dimensional groundwater model, see Figure 3.1. It starts with the definition of scenarios that need to be computed, e.g. different pumping strategies for extraction wells. With these in mind we compute several model evaluations (i.e. the snapshots [Sirovich, 1987]) with the original high-dimensional model. These yield a truncated set of patterns by means of a pattern identification technique (EOF) that forms the basis of a low-dimensional model according to a chosen projection technique (SSPM, SSPM^a or GPM). We call these processes the *preparation phase*. For each of the scenarios within the scenario definition, we can simulate the time-dependent coefficients by means of the reduced model. We refer to this as the *simulation phase* that finally yields results after a *reconstruction phase*. In practice, verification of the reduced model can lead to a refinement of the snapshots and/or to the selection of more or different patterns.

This chapter outlines briefly, in Subsection 3.1, the original model equations and gives a description of the model-reduced methodology. Subsection 3.2 elaborates extensively on the mathematical definition for the reduced models SSPM, SSPM^a and GPM, respectively. In Subsection 3.3 we illustrate the performance and differences between the three methods by means of a real-world case. Finally, we give some conclusions and recommendation in Subsection 3.5.

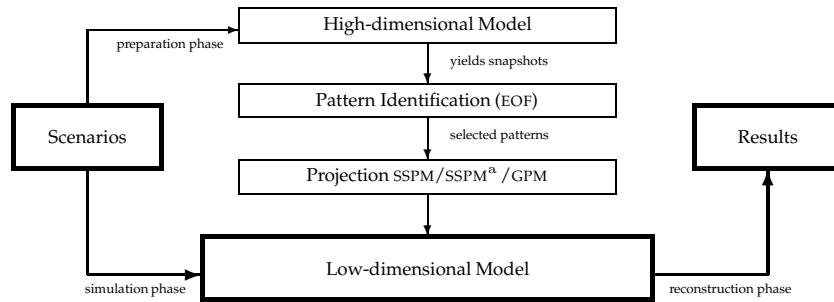


Figure 3.1: Model reduction methodology.

3.1 Reduced Model Methodology

3.1.1 Formulation of Groundwater Flow

To describe groundwater flow in three dimensions with a uniform density and viscosity, we use the partial differential equation based upon Darcy's law and the equation of continuity:

$$\frac{\partial}{\partial x} \left(K^{xx} \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(K^{yy} \frac{\partial \phi}{\partial y} \right) + \frac{\partial}{\partial z} \left(K^{zz} \frac{\partial \phi}{\partial z} \right) - q + C^f (\phi - h^f) = S \frac{\partial \phi}{\partial t}, \quad (3.1)$$

where ϕ represents the hydraulic head [L], K^{xx} , K^{yy} , K^{zz} are the hydraulic permeabilities in x , y , z direction [LT^{-1}], S is the specific storage coefficient, or volume of water released from storage per aquifer per unit decline in head [L^{-1}], t is time [T], q is a fluid source/sink term, or the volumetric rate at which water is added to or removed from the system [T^{-1}], C^f represents the specific conductance between a boundary condition (e.g. surface water) and the hydraulic head [$(LT)^{-1}$], and h^f is the corresponding water level [L] in the boundary condition.

3.1.2 Formulation of a Reduced Model Structure

A reduced model structure is based upon the assumption that the hydraulic head can be expressed in terms of a set of spatially distributed patterns with time-varying coefficients (Rayleigh-Ritz procedure). Hence, this approximated hydraulic head $\hat{\phi}$ can be described by:

$$\hat{\phi}(x, y, z, t) = \phi^b(x, y, z) + \sum_i p_i(x, y, z) r_i(t) \quad (3.2)$$

where p_i is the i^{th} pattern value [–], r_i is the i^{th} time-dependent coefficient [L], and ϕ^b is the background hydraulic head [L]. For this type of model structure, the spatial distribution of the patterns is captured once during the *preparation phase*. During the actual *simulation phase* we need to compute only i coefficients r over time t . The background hydraulic head ϕ^b is computed by means of those time-independent boundary conditions. These conditions are not described by patterns and can not be varied within the low-dimensional model.

A relationship exists between the patterns and the model parameterization described by K^{xx} , K^{yy} , K^{zz} , S , C^f . Whenever one of those parameters is changed, we need to identify a new reference head and a set of patterns. In Chapter 4, however, a methodology is presented that handles these dependencies more efficiently.

3.1.3 Pattern Identification (EOF)

Among others, the pattern identification technique described here is known as Proper Orthogonal Decomposition (POD) or Empirical Orthogonal Functions (EOFs). It can be classified as one that examines the linear relationship between variables with the aim of reducing the dimensionality of the problem. It falls into the category of *Spectral methods* or *Multivariate methods* and can be seen as reconstructing or

Definitions

A system is considered to be **linear** whenever the behavior of its state does not affect any parameter that influences the system, and thus the state recursively. Otherwise we speak of **nonlinear systems**. Linear systems can be solved ‘directly’ though nonlinear systems need to be solved by a recursive algorithm (e.g. a **Picard iteration** or **Newton iteration**). The algorithm reformulates a linear system for each iteration that is affected by nonlinear terms. Another, more important advantage of linear systems is that they support the **superposition principle**. This principle composes a signal by adding different signals multiplied by specific weight factors. The proposed reduced model structure (Equation 3.2) uses this decomposition. The weight factors can be obtained by a linear or even nonlinear reduced models.

modeling a ‘signal’ optimally in terms of patterns or modes. The methods consist of computing a low-dimensional subspace $\mathcal{S}$ (described by patterns) for the original high-dimensional model space $\mathcal{X}$. Eventually, the application of a reduced model is always limited to the quality of the subspace $\mathcal{S}$ and the number of patterns that describes or *spans* it. Moreover, it is not possible to reconstruct a ‘signal’ different from the ones used to derive the patterns. However, if we consider a linear system, we are able to construct different ‘signals’ by the superposition principle.

To compute EOFs, we create an empirical data set in which n^e snapshots of the original steady-state or transient model are arranged in vectors ϕ_i . Each element within a snapshot represents the hydraulic head (the ‘signal’) for a specific location in the model. Since the reduced model computes only that part that needs to be added to the background hydraulic head ϕ^b (Equation 3.2), we correct each snapshot vector by subtracting ϕ^b , so

$$\mathbf{d}_i = \phi_i - \phi^b ; i \in \{1, \dots, n^e\}. \quad (3.3)$$

We weigh each vector $\mathbf{d}_i$ with its vector length (i.e. normalising):

$$\mathbf{e}_i = \frac{1}{\|\mathbf{d}_i\|} \mathbf{d}_i ; i \in \{1, \dots, n^e\}. \quad (3.4)$$

The necessity for weighing is because the irregularity that exists within a snapshot is more important than its amplitude [Newman, 1996a]. For example a snapshot that contains the influence for precipitation can become dominant due to its larger vector components (*factor loadings*), compared to another snapshot vector that e.g. represents the influence of a single extraction well. Eventually, the patterns (known as *basis functions*) follow from an eigenvalue decomposition of a covariance matrix $\mathbf{G}$ (‘outer product’, ‘tensor product’ or ‘autocorrelation tensor’) constructed from the snapshots [Sirovich, 1987]:

$$\mathbf{G} = \frac{1}{n^e} \mathbf{E} \mathbf{E}^T, \quad (3.5)$$

where the corrected snapshots are collected in the matrix $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{n^e}]$. To shorten the calculation time necessary for solving the eigenvalue problem for this

often very high-dimensional covariance matrix, we define a different covariance matrix $\underline{\mathbf{G}}$ as the inner product ('dot product' or 'sum of products') of a matrix [Krysl *et al.*, 2001]:

$$\underline{\mathbf{G}} = \mathbf{E}^T \mathbf{E}, \quad (3.6)$$

and solve a 'reduced' eigenvalue problem:

$$\underline{\mathbf{G}} \mathbf{v}_i = \mathbf{v}_i \lambda_i; \quad i \in \{1, \dots, n^e\}, \quad (3.7)$$

for which we transform the eigenvectors $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n^e}]$ into patterns $\mathbf{P}$ by [Golub & van Loan, 1989]:

$$\mathbf{P} = \mathbf{E} \mathbf{V} \Lambda^{-\frac{1}{2}}, \quad (3.8)$$

where $\Lambda^{-\frac{1}{2}}$ is the square root of the inverse of a matrix with eigenvalues λ_i on the main diagonal. The eigenvectors and eigenvalues are equal to those that would have been obtained by an eigenvalue decomposition of $\underline{\mathbf{G}}$. In Figure 3.2a-c we have depicted a graphical interpretation of the mentioned identification technique. In fact, there is an infinite number of patterns (also known as 'coordinate systems') that describe or *span* the snapshots, however, the linear basis from the eigenvalue decomposition is optimal as it has a smaller mean-square error than any other linear basis of the same dimensions [Holmes *et al.*, 1996]. It is also known as the optimal solution to an *orthogonal regression* problem.

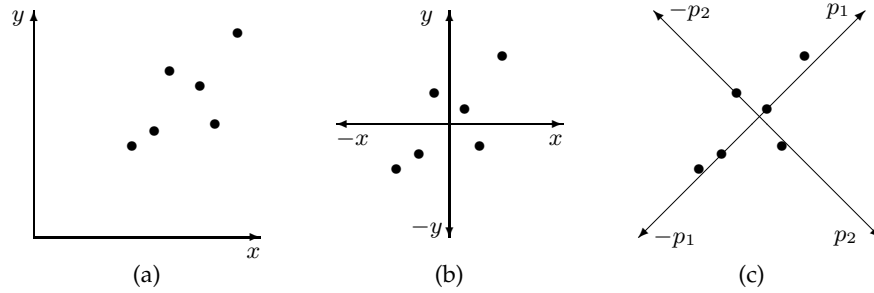


Figure 3.2: Hypothetic schematic example showing the pattern identification technique that consists of (a) a set of snapshots (•) that are described or spanned by their original coordinate system ($x \times y$). The snapshots are (b) centered (zero-averaged for their background state) and (c) can be optimally spanned by a different coordinate system ($p_1 \times p_2$) in terms of a mean-square error.

The eigenvalues λ_i provide a measure for the relative 'energy' associated with a corresponding eigenvector $\mathbf{p}_i$ (i.e. a single EOF). This measure can also be interpreted as the relative amount of 'time' that the hydraulic head spends along the corresponding pattern i . Its relative importance is given by:

$$\varphi_i = \frac{\lambda_i}{\sum_{j=1}^{n^e} \lambda_j} \cdot 100\%; \quad i \in \{1, \dots, n^e\}. \quad (3.9)$$

From a mathematical point of view, the reduced model should be able to reproduce at least that amount of model variance as explained totally by the selected set of n^p patterns. We call this the *expected* variance φ^e defined as:

$$\varphi^e = \sum_{i=1}^{n^p} \varphi_i. \quad (3.10)$$

The number of patterns n^p should be at least such that the expected variance is more-or-less equal to the *reconstructed* variance φ^r :

$$\varphi^r = \varphi^h (\varphi^l)^{-1} \cdot 100\%, \quad (3.11)$$

where φ^h and φ^l are the total variances computed with the high and low-dimensional model, respectively. When $\varphi^e \neq \varphi^r$, the low-dimensional model lacks significant patterns and more patterns should be added. We collect n^p patterns such that their corresponding $\varphi_1 > \varphi_2 > \dots > \varphi_{n^p}$ and they totally explain at least the given φ^e amount of model variance. We store them as columns within the span $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{n^p}\}$. This span is *orthonormal*, which means that all patterns are perpendicular to each other and have a unit length ($\mathbf{P}^T \mathbf{P} = \mathbf{I}$). It can be interpreted as a description or span of a high-dimensional coordinate system that is defined by a limited amount of axes. This typical information is used to construct a numerical model than can, however, operate only within the numerical subspace $\mathcal{S} \in \mathbb{R}^{n^p}$ described by the set of patterns. For example, when we use only pattern $\mathbf{p}_1$ to represent the snapshots, we reduce a two-dimensional system $\mathbb{R}^2$ into a more efficient one-dimensional one $\mathbb{R}^1$, see Figure 3.3a. However, the error of a model that operates within this one-dimensional space, occurs simply because the pattern is not able to capture all snapshots accurately. Such an error is comparable to the linear regression coefficient R^2 . Whenever we use $\mathbf{p}_2$ instead, which explains less variance than $\mathbf{p}_1$, we observe that the ‘misfit’, and thus the error, increases, see Figure 3.3b. Thus, the accuracy of a reduced model increases with the number of patterns in-

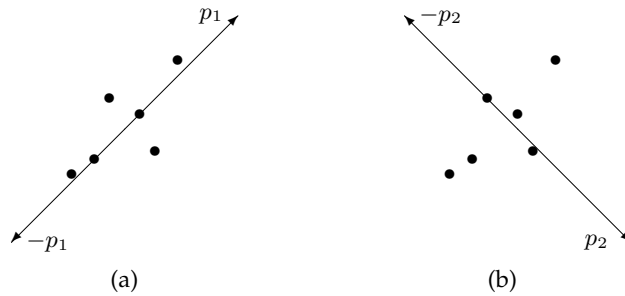


Figure 3.3: Hypothetic schematic example showing the concept of model reduction in which (a) a model is constructed that operates along the one-dimensional space of $\mathbf{p}_1$, and (b) along the one-dimensional space of $\mathbf{p}_2$ which explains less snapshot variance than the former.

Alternative Pattern Selection

Sometimes we need to simulate the hydraulic head most accurately near specific locations only, e.g. near a well or within nature conservation areas. For those situations we define an alternative selection criterion λ'_i that is expressed for each pattern i as the total of its elements j within n^P selected locations throughout the model, so:

$$\lambda'_i = \sum_{j=1}^{n^P} \|p_{i,j}\| ; i \in \{1, \dots, n^m\}. \quad (3.12)$$

Instead of λ_i we can use λ'_i to sort the patterns and use this reorganization to determine the set of patterns within the matrix $\mathbf{P}$ by means of Equation (3.9).

involved, and the highest accuracy is achieved by $\varphi^e = 100\%$, however this yields to lowest efficiency of the model. So, there is always a trade-off between efficiency and accuracy. Though there are realistic situations in which the time efficiency increases by $\varphi^e = 99.9\%$ without any decrease in accuracy. This particular aspect depends on the distribution of φ_i , e.g. many patterns that describe a very small amount of variance can be eliminated without affecting the quality of the reduced model,

3.1.4 Physical Meaning of Patterns

Consider a one-dimensional problem dimensioned by 101 columns, an extraction well in the middle and two Dirichlet conditions on opposing sides. We can determine the subspace $\mathcal{S}$ by computing the original model for several time steps and vary the well strenght, see Figure 3.4a. This plot shows some of the solutions of the original model. The pattern identification technique seeks for another ‘coordinate system’ that spans these in terms of number of modes or patterns most optimally. So, we remove the background state and normalize the snapshots to

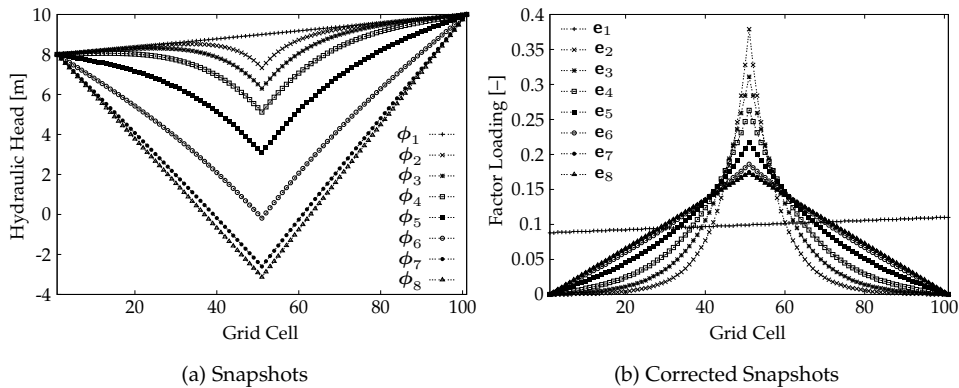


Figure 3.4: Graphs showing (a) the snapshots ϕ_i and (b) the corrected snapshots $\mathbf{e}_i$ that are background state centered and normalized.

avoid that those with greatest variance and amplitude will dominate the results, see Figure 3.4b. In Figure 3.5a we have plotted the factor loadings of the leading six patterns. They have reduced the dimensionality of the problem from 101 onto 6, however, the patterns are not easily to interpret. That is, whether they can be associated with realistic physical processes in the system. *Richman* [1986] mentioned that a *rotation* of patterns increases the interpretability of the individual patterns. He also demonstrated that unrotated patterns are primarily a function of the geometrical shape of the (sub)domain and that they become intermixed as their eigenvalues are very closely spaced. Basically, it means that the elements of the patterns should contain mostly small values (i.e. a lot of zeros) such that they become *simple* in their appearance (referred to as *simple structures*). In that particular case, each pattern affects relatively a few components within ϕ , or alternatively each component ϕ is affected by relatively few patterns. A popular scheme for rotation, is the *Varimax Rotation* as suggested by *Kaiser* [1958]. The algorithm maximizes the overall variance by means of an iterative orthogonal rotation sequence that rotates two patterns at a time while holding the other patterns constant, so

$$\mathbf{P}^{\text{rot}} = \mathbf{P} \begin{bmatrix} \cos(o) & -\sin(o) \\ \sin(o) & \cos(o) \end{bmatrix}, \quad (3.13)$$

where o is the rotation angle. This rotation continues until the increase of the overall variance drops below a preset value. We have depicted the rotated patterns $\mathbf{P}^{\text{rot}}$ in Figure 3.5b and the extraction well is more distinguishable. Somehow, they become more comparable to analytical elements used and proposed by *Strack* [1989]. Instead of representing only one feature (e.g. a single well or line element), a pattern can represent combined features within distorted hydraulic permeability fields. However, for the mentioned rotation sequence there is no other reason than ‘pattern interpretability’. It influences only the amplitude of the time-dependent

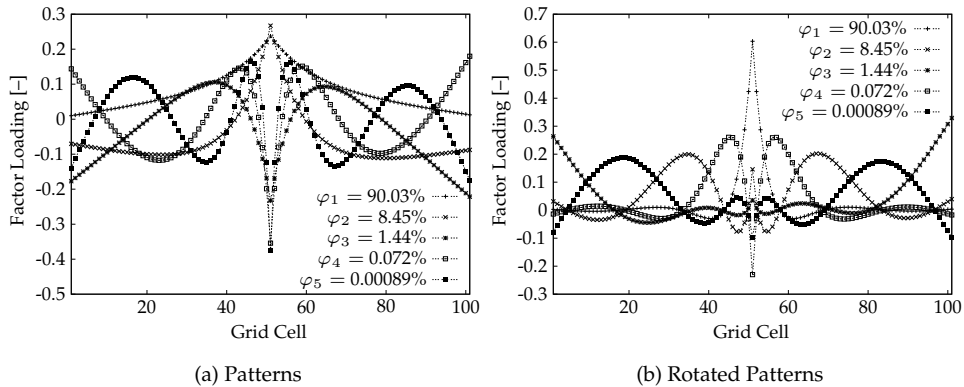


Figure 3.5: Graphs showing the factor loadings of (a) the pattern vectors $\mathbf{p}_i$ that explain totally $\varphi^\circ = 100\%$ of the model variance as present in the snapshots, and (b) the rotated vectors $\mathbf{p}_i^{\text{rot}}$.

coefficients, it does not affect the quality of the reduced model at the end.

3.2 Reduced Model Formulations

In the forthcoming subsections we describe two different formulations for a dynamical model that simulates the time-dependent coefficients r_i (Equation 3.2). We define an ordinary differential equation (ODE) that describes dr/dt :

$$\mathcal{F}(K^{xx}, K^{yy}, K^{zz}, S) r - \underline{q} = \frac{dr}{dt}, \quad (3.14)$$

where $\underline{q}$ is a reduced term that expresses the boundary conditions within the reduced model, comparable to q in Equation (3.1), and $\mathcal{F}$ is a linear function that depends on the system properties of the original model (i.e. the variables K^{xx} , K^{yy} , K^{zz} , S). The function describes the internal relation of the coefficients r_i . In this chapter we discuss two strategies to compute dr/dt :

- the *State-Space Projection Method* (SSPM) which is a linear coordinate transformation of the original system equations written in matrix format (see Subsection 3.2.1). Although the projection consumes more time to complete, it has an advantage from a numerical point of view, and it leads to a reduced model that computes faster than the following method;
- the *Galerkin Projection Method* (GPM) which substitutes the original state variable ϕ in the original PDE with the reduced model structure and projects the outcome onto the patterns (see Subsection 3.2.2). This method can be thought of as the more elementary method of the two, because it results in an ODE for dr/dt before projection.

3.2.1 State-Space Projection Method

One way to solve the continuous formulation for groundwater flow is to discretize the PDE in space and time by use of finite differences. We discretize the PDE (Equation 3.1) for each grid cell in a mesh that consists of n^m nodes wherein each node is indexed by a row (i), column (j) and layer (k) number. For each node we write an equation that expresses the finite-difference approximation for the space and time derivative of the hydraulic head, so

$$\begin{aligned} & \left(-C_{i,j-\frac{1}{2},k}^r - C_{i,j+\frac{1}{2},k}^r - C_{i-\frac{1}{2},j,k}^c - C_{i+\frac{1}{2},j,k}^c - \right. \\ & \left. - C_{i,j,k-\frac{1}{2}}^l - C_{i,j,k+\frac{1}{2}}^l - \frac{1}{\Delta t_i} \Delta x_j \Delta y_i S_{i,j,k} - \Delta x_j \Delta y_i C_{i,j,k}^f \right) \phi_{i,j,k}(t_i) + \\ & + C_{i,j-\frac{1}{2},k}^r \phi_{i,j-1,k}(t_i) + C_{i,j+\frac{1}{2},k}^r \phi_{i,j+1,k}(t_i) + C_{i-\frac{1}{2},j,k}^c \phi_{i-1,j,k}(t_i) + \\ & + C_{i+\frac{1}{2},j,k}^c \phi_{i+1,j,k}(t_i) + C_{i,j,k-\frac{1}{2}}^l \phi_{i,j,k-1}(t_i) + C_{i,j,k+\frac{1}{2}}^l \phi_{i,j,k+1}(t_i) = \\ & = q_{i,j,k}(t_i) - \frac{1}{\Delta t_i} \Delta x_j \Delta y_i S_{i,j,k} \phi_{i,j,k}(t_{i-1}) - \Delta x_j \Delta y_i C_{i,j,k}^f h_{i,j,k}^f(t_i), \quad (3.15) \end{aligned}$$

where

$$C_{i\pm 1,j,k}^c = \frac{2\Delta x_j \Delta z_k^2 K_{i,j,k}^{yy} K_{i\pm 1,j,k}^{yy}}{K_{i,j,k}^{yy} \Delta z_k \Delta y_{i\pm 1} + K_{i\pm 1,j,k}^{yy} \Delta z_k \Delta y_i}, \quad (3.16a)$$

$$C_{i,j\pm 1,k}^r = \frac{2\Delta y_i \Delta z_k^2 K_{i,j,k}^{xx} K_{i,j\pm 1,k}^{xx}}{K_{i,j,k}^{xx} \Delta z_k \Delta x_{j\pm 1} + K_{i,j\pm 1,k}^{xx} \Delta z_k \Delta x_j}, \quad (3.16b)$$

$$C_{i,j,k\pm 1}^l = \frac{\Delta x_j \Delta y_i}{R_{i,j,k}}. \quad (3.16c)$$

The C^r, C^c, C^l [L^2T^{-1}] are the confined harmonic mean hydraulic conductances between nodes along the rows, columns and layers, respectively, $\Delta x, \Delta y, \Delta z$ [L] are the dimensions of a grid cell along those directions, R [T] is the vertical resistance between two aquifers. Since Equation (3.15) involves up to seven unknown values of ϕ , the equations for the entire model grid must be solved simultaneously at each time step t_i . This entire set of equations can be summarized in matrix form as:

$$\left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S}\right) \phi(t_i) = -\frac{1}{\Delta t} \mathbf{S} \phi(t_{i-1}) + [\mathbf{q} - \mathbf{C}^f \mathbf{h}^f](t_i), \quad (3.17)$$

where $\mathbf{A}$ is a heptadiagonal system matrix (*operator*) that contains all the conductances that affect ϕ . The set of equations are often solved by means of an iterative procedure [McDonald & Harbaugh, 1988]. It is also possible to rewrite the equation such that

$$\phi(t_i) = -\frac{1}{\Delta t} \left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S}\right)^{-1} \mathbf{S} \phi(t_{i-1}) + \left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S}\right)^{-1} [\mathbf{q} - \mathbf{C}^f \mathbf{h}^f](t_i). \quad (3.18)$$

The advantage of Equation (3.18) is that we find only the solution vector $\phi(t_i)$ by matrix multiplication. Though it is highly unusual that someone would ever simulate such formulation as it is computationally very demanding. Its 'dreary' characteristic with the inverse of $\mathbf{A} - \frac{1}{\Delta t} \mathbf{S}$ can be overcome whenever we multiply the entire equation with the selected set of patterns $\mathbf{P}$. We call this linear operation a *projection* or a *change-of-coordinates* in which the matrix $\mathbf{P}^T$ acts as a *reductor* that reduces an n^m dimensional vector into an n^p dimensional vector. $\mathbf{P}$ itself, however, acts as a *reconstructor* that performs the reverse operation, so

$$\mathbf{r} = \mathbf{P}^T \phi, \quad (3.19a)$$

$$\phi = \mathbf{P} \mathbf{r}. \quad (3.19b)$$

To reduce a matrix we pre multiply it with $\mathbf{P}^T$ and post multiply it with $\mathbf{P}$. Equation (3.18) looks now as

$$\mathbf{r}(t_i) = -\frac{1}{\Delta t} \mathbf{P}^T \left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S}\right)^{-1} \mathbf{S} \mathbf{P} \mathbf{r}(t_{i-1}) + \mathbf{P}^T \left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S}\right)^{-1} \mathbf{b}(t_i), \quad (3.20a)$$

$$\mathbf{b} = \mathbf{q} - \mathbf{C}^f (\mathbf{h}^f - \phi). \quad (3.20b)$$

Unfortunately, we cannot solve Equation (3.20a) as long as we need to compute the inverse, therefore we rewrite it as:

$$\mathbf{r}(t_i) = -\frac{1}{\Delta t} \left[\left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)^{-T} \mathbf{P} \right]^T \mathbf{SPr}(t_{i-1}) + \left[\left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)^{-T} \mathbf{P} \right]^T \mathbf{b}(t_i), \quad (3.21)$$

and, instead of computing $\left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)^{-T}$ (i.e. the transpose of the inverse) completely, we define a matrix $\mathbf{N}$ with dimension $[n^m \times n^p]$ as:

$$\mathbf{N} = \left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)^{-T} \mathbf{P}, \quad (3.22)$$

for which we solve each vector $\mathbf{n}_i$ in $\mathbf{N}$, so:

$$\left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)^T \mathbf{n}_i = \mathbf{p}_i, \quad i \in \{1, \dots, n^p\}. \quad (3.23)$$

It should be noticed that it is not difficult to implement this as $\left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)^T = \left(\mathbf{A} - \frac{1}{\Delta t} \mathbf{S} \right)$ and hence we need to exchange the right-hand-side within Equation (3.17) by $\mathbf{p}_i$. We can now rewrite Equation (3.21) as:

$$\mathbf{r}(t_i) = \underline{\mathbf{A}}\mathbf{r}(t_{i-1}) + \left(\mathbf{N}^T \mathbf{b} \right)(t_i), \quad (3.24a)$$

$$\underline{\mathbf{A}} = -\frac{1}{\Delta t_i} \mathbf{N}^T \mathbf{SP}. \quad (3.24b)$$

We call this a reduced model that simulates a finite-difference approximation of $d\mathbf{r}/dt$ (Equation 3.14) based upon a state-space formulation of groundwater flow. The computation of $\mathbf{r}(t_i)$ consists finally of a time-dependent projection of $\mathbf{b}(t_i)$ with $\mathbf{N}^T[n^p \times n^m]$, and a multiplication of the previous coefficient $\mathbf{r}(t_{i-1})$ with a reduced system operator $\underline{\mathbf{A}}[n^p \times n^p]$.

3.2.2 Galerkin Projection Method

The Galerkin Projection Method can be thought of as more elementary than the SSPM method because it results in an ODE for $d\mathbf{r}/dt$ before the reduction algorithms.

Alternative Low-dimensional Model SSPM^a

The reductor $\mathbf{P}^T$ and reconstructor $\mathbf{P}$ can be applied directly to Equation 3.17, so:

$$(\underline{\mathbf{A}} - \underline{\mathbf{S}}) \phi(t_i) = \underline{\mathbf{S}}\phi(t_{i-1}) + \mathbf{P}^T \left[\mathbf{q} - \mathbf{C}^f \mathbf{h}^f \right](t_i), \quad (3.25a)$$

$$\underline{\mathbf{A}} = \mathbf{P}^T \mathbf{A} \mathbf{P}, \quad (3.25b)$$

$$\underline{\mathbf{S}} = -\frac{1}{\Delta t_i} \mathbf{P}^T \mathbf{S} \mathbf{P}. \quad (3.25c)$$

Because Equation 3.25a is low dimensional we can solve it directly instead of iteratively. We refer to this method as the SSPM^a and it is the least sophisticated method but easiest to program. It can be seen as a intermediate formulation between the SSPM and the GPM.

It substitutes the hydraulic head ϕ within the original PDE (Equation 3.1) by the reduced model structure (Equation 3.2). This results in the formulation:

$$\begin{aligned} \frac{\partial}{\partial x} \left[K^{xx} \frac{\partial \left(\phi^b + \sum_{i=1}^{n^p} p_i r_i \right)}{\partial x} \right] + \frac{\partial}{\partial y} \left[K^{yy} \frac{\partial \left(\phi^b + \sum_{i=1}^{n^p} p_i r_i \right)}{\partial y} \right] + \\ + \frac{\partial}{\partial z} \left[K^{zz} \frac{\partial \left(\phi^b + \sum_{i=1}^{n^p} p_i r_i \right)}{\partial z} \right] - q - C^f \left[\left(\phi^b + \sum_{i=1}^{n^p} p_i r_i \right) - h^f \right] = \\ = S \frac{\partial \left(\phi^b + \sum_{i=1}^{n^p} p_i r_i \right)}{\partial t}. \end{aligned} \quad (3.26)$$

Eventually, we need an equation that describes dr/dt (Equation 3.14), and therefore we rearrange Equation (3.26) such that r becomes isolated from the second order spatial derivative, so:

$$\begin{aligned} \sum_{i=1}^{n^p} \left[\underbrace{\frac{\partial}{\partial x} \left(K^{xx} \frac{\partial p_i}{\partial x} \right)}_{u_i^c} + \underbrace{\frac{\partial}{\partial y} \left(K^{yy} \frac{\partial p_i}{\partial y} \right)}_{u_i^r} + \underbrace{\frac{\partial}{\partial z} \left(K^{zz} \frac{\partial p_i}{\partial z} \right)}_{u_i^l} - \underbrace{C^f p_i}_{u_i^f} \right] r_i = \\ = q - C^f (h^f - \phi^b) + \sum_{i=1}^{n^p} \underbrace{S p_i}_{u_i^s} \frac{\partial r_i}{\partial t}. \end{aligned} \quad (3.27)$$

A main advantage in this formulation is that the entire term bracketed in the left-hand-side of the equation, can be computed in advance. It should be noticed that the elimination of ϕ^b herein is only sustained for these type of confined linear models as K^{xx} , K^{yy} , K^{zz} , S are all independent of ϕ^b . The removal of ϕ^b that affects C^f is corrected by adjusting the surface water level h^f by ϕ^b . Whenever we apply a finite-difference approximation for the pattern derivative of space, we can compute for each pattern p_i the variables u^c , u^r , u^l , u^f , u^s for each grid cell i, j, k in the mesh as:

$$u_{i,j,k,n}^c = \Delta x_j^{-1} \left(C_{i,j,k}^c \frac{p_{i,j,k,n} - p_{i,j+1,k,n}}{\frac{1}{2}(\Delta x_j + \Delta x_{j+1})} - C_{i,j-1,k}^c \frac{p_{i,j,k,n} - p_{i,j-1,k,n}}{\frac{1}{2}(\Delta x_j + \Delta x_{j-1})} \right), \quad (3.28a)$$

$$u_{i,j,k,n}^r = \Delta y_i^{-1} \left(C_{i,j,k}^r \frac{p_{i,j,k,n} - p_{i+1,j,k,n}}{\frac{1}{2}(\Delta y_i + \Delta y_{i+1})} - C_{i-1,j,k}^r \frac{p_{i,j,k,n} - p_{i-1,j,k,n}}{\frac{1}{2}(\Delta y_i + \Delta y_{i-1})} \right), \quad (3.28b)$$

$$u_{i,j,k,n}^l = C_{i,j,k}^l (p_{i,j,k,n} - p_{i,j,k+1,n}) - C_{i,j,k-1}^l (p_{i,j,k,n} - p_{i,j,k-1,n}), \quad (3.28c)$$

$$u_{i,j,k,n}^f = C_{i,j,k}^f p_{i,j,k,n}, \quad (3.28d)$$

$$u_{i,j,k,n}^s = S_{i,j,k} p_{i,j,k,n}, \quad (3.28e)$$

with

$$C_{i,j\pm 1,k}^r = (\Delta x_j + \Delta x_{j\pm 1}) / \left(\frac{\Delta x_j}{K_{i,j,k}^{xx} \Delta z_k} + \frac{\Delta x_{j\pm 1}}{K_{i,j\pm 1,k}^{xx} \Delta z_k} \right), \quad (3.29a)$$

$$C_{i\pm 1,j,k}^c = (\Delta y_i + \Delta y_{i\pm 1}) / \left(\frac{\Delta y_i}{K_{i,j,k}^{yy} \Delta z_k} + \frac{\Delta y_{i\pm 1}}{K_{i\pm 1,j,k}^{yy} \Delta z_k} \right), \quad (3.29b)$$

$$C_{i,j,k\pm 1}^l = (R_{i,j,k\pm 1})^{-1}, \quad (3.29c)$$

where $p_{i,j,k,n}$ is the n^{th} pattern value out of n^p [–] for grid cell i, j, k , and C^r, C^c, C^l are the specific confined harmonic mean hydraulic conductances [T^{-1}] between nodes along the rows, columns and layers, respectively. The entire Equation (3.27) can now be summarized for the entire model mesh in matrix form, as:

$$\underbrace{(\mathbf{U}^c + \mathbf{U}^r + \mathbf{U}^l - \mathbf{U}^f)}_{\mathbf{U}} \mathbf{r} = \mathbf{U}^s \frac{d\mathbf{r}}{dt} + \underbrace{\mathbf{q} - \mathbf{C}^f (\mathbf{h}^f - \phi^b)}_{\mathbf{b}} \quad (3.30)$$

where $\mathbf{U}$ has dimension $[n^m \times n^p]$. The mathematical dimension (i.e. number of elements within a vector) within Equation (3.30) is still equal to the original amount n^m , though the number of the unknown elements in vector $\mathbf{r}$ is n^p . In words: we have more equations than unknown variables. To eliminate all superfluous equations, we project the entire Equation (3.30) upon the selected set of pattern, so:

$$\underbrace{\mathbf{P}^T \mathbf{U}}_{\mathbf{A}} \mathbf{r} = \underbrace{\mathbf{P}^T \mathbf{U}^s}_{\mathbf{S}} \frac{d\mathbf{r}}{dt} + \mathbf{P}^T \mathbf{b}. \quad (3.31)$$

This is an ODE for a reduced model for groundwater flow. We can solve it by applying an implicit Euler scheme to the time derivative of $\mathbf{r}$, resulting in:

$$\left[\mathbf{A} - \frac{1}{\Delta t_i} \mathbf{S} \right] \mathbf{r}(t_i) = -\frac{1}{\Delta t_i} \mathbf{S} \mathbf{r}(t_{i-1}) + \mathbf{P}^T \mathbf{b}(t_i). \quad (3.32)$$

This formulation consists of three time-independent matrices $\mathbf{A}$ and $\mathbf{S}$ both with a small dimension $[n^p \times n^p]$, and the projection matrix $\mathbf{P}^T$ with dimension $[n^p \times n^m]$. During the actual simulation $\mathbf{b}$ should be multiplied with $\mathbf{P}^T$ and after that it is easy to obtain the coefficients $\mathbf{r}(t_i)$ because all matrices and vectors in the equation are low dimensional.

3.2.3 Initial Conditions

For transient simulations the initial condition $\phi(t_0)$ can be different from the background state ϕ^b . It is fairly easy to project $\phi(t_0)$ upon the patterns as:

$$\mathbf{r}(t_0) = \mathbf{P}^T [\phi(t_0) - \phi^b], \quad (3.33)$$

where $\mathbf{r}(t_0)$ is the initial condition for the low-dimensional model. It is only meaningful if we can indeed *span* $\phi(t_0)$ by the n^p selected patterns:

$$\phi(t_0) - \phi^b = (c\mathbf{p})_1 + (c\mathbf{p})_2 + \dots + (c\mathbf{p})_{n^p}, \quad (3.34)$$

where c_i are coefficients which should be found. Because ϕ_0 is often a steady-state solution of the full-dimensional model, it is necessary to include at least steady-state evaluations of the full-dimensional model in the snapshot vectors.

3.2.4 Discussion

In this subsection we discuss some mathematical limitations for the reduced model formulations. These concern time discretization and the inclusion of numerical errors.

Time Discretization

There is an important difference in the reduced model formulations that affects the implementation of time steps.

- *State-Space Projection Method* (SSPM): within the mathematical formulation of this method the time step Δt_i is included within the computation of the matrices $\mathbf{A}$ and $\mathbf{N}$ (see Subsection 3.2.1). To avoid these Δt_i -dependent computations, as they completely undermine the gained computational effort, we suggest a more practical solution. Hereby the concerned matrices are computed for a specific Δt_i that is capable to follow the evolution of the desired time steps; this could be the smallest Δt within the simulation, e.g. 10 days. Whenever Δt_i changes during the simulation into 30 days, we solve $\mathbf{r}$ for the intermediate periods (10 and 20 days) without recomputing $\mathbf{N}^T \mathbf{b}$ (Equation 3.24a). This will hardly increase the computational time, and there is another motivation to support this strategy. Inherent to the method of finite differences, the evolution of ϕ in time ($\partial\phi/\partial t$) is discretized linearly between adjacent time steps and the smaller Δt the more $\Delta\phi/\Delta t = \partial\phi/\partial t$;
- *Galerkin Projection Method* (GPM) and *Alternative State-Space Projection Method* (SSPM^a): a change in Δt_i does not affect the reduced model that severe as within the SSPM. There is no limitation for a variation in Δt_i as the finite-difference approximation of $d\mathbf{r}/dt$ has been computed after the actual projection (Equation 3.31). As a result, Δt can be varied unlimited at low computational costs as it affects only low-dimensional matrices (Equation 3.32).

Numerical Errors

There is an important difference between the SSPM, SSPM^a and GPM that concerns the stage at which the actual projection is applied. The SSPM^a and GPM can be

summarized by the following sequence of equations:

$$\mathbf{A}\phi = \mathbf{b} \xrightarrow{\text{projecting}} \mathbf{P}^T \mathbf{A} \mathbf{P} \mathbf{P}^T \phi = \mathbf{P}^T \mathbf{b} \Leftrightarrow \mathbf{r} = \underbrace{(\mathbf{P}^T \mathbf{A} \mathbf{P})^{-1}}_{\hat{\mathbf{A}}} \mathbf{P}^T \mathbf{b}, \quad (3.35)$$

where $\mathbf{A}$ reflects the system matrix and $\mathbf{b}$ the model forcing terms. It should be noticed that $\hat{\mathbf{A}}$ is projected upon the patterns prior to the computation of an inverse. This is very efficient because the dimension is first significantly reduced, but it can lead to serious errors in $\hat{\mathbf{A}}$. For this particular error the SSPM is more accurate, as shown by the following sequence of equations:

$$\phi = \mathbf{A}^{-1} \mathbf{b} \xrightarrow{\text{projecting}} \mathbf{P}^T \phi = \mathbf{P}^T \mathbf{A}^{-1} \mathbf{P} \mathbf{P}^T \mathbf{b} \Leftrightarrow \mathbf{r} = \underbrace{\mathbf{P}^T \mathbf{A}^{-1} \mathbf{P}}_{\hat{\mathbf{A}}} \mathbf{P}^T \mathbf{b}. \quad (3.36)$$

The advantage of this SSPM is its inclusion of $\mathbf{A}^{-1}$. Therefore we assign $\hat{\mathbf{A}}$ as the true, and $\hat{\hat{\mathbf{A}}}$ as an approximation through its a priori projection. The ‘extra’ mathematical error caused by the difference between $\hat{\mathbf{A}}$ and $\hat{\hat{\mathbf{A}}}$ is negligible whenever

- we include all snapshot variance by use of the SSPM^a or GPM. In other words: the reduced model should explain $\varphi^e = 100\%$. In practice this will be never the case, as patterns with tiny eigenvalues are numerically ill defined. The SSPM method introduces errors caused by these (unreliable) patterns, as they contribute to the inverse at their full vector dimension (n^m). The SSPM^a and GPM suffer less because the vector dimension of the unreliable patterns is reduced significantly (n^p) before the inverse computation;
- we neglect the round-off errors because they accumulate with the number of grid cells n^m and patterns n^p . This error will play a more important role for the SSPM as we need to solve $\mathbf{P}^T (\mathbf{A} - \frac{1}{\Delta t} \mathbf{S})^{-1}$ iteratively (Equation 3.23).

3.3 Real-world Case

3.3.1 Description of the Original Model

To evaluate the accuracy and suitability of a low-dimensional model for ground-water flow, we use a supra-regional model of the province of Noord-Brabant, the Netherlands. This province is characterized by regional areas with free-floating water tables (south, south-east) and regions with intense dewatering systems (north, north-west). The latter dewaterers and recharges in proportion to the difference between ϕ and h^f . Hence, the model is strictly linear. Finally, an open boundary condition has been modeled all around by applying Dirichlet conditions for each model layer.

The underground is a complex system of succeeding high- ($> 5,000 \text{ m}^2 \text{ day}^{-1}$) and extremely low ($< 10^{-6} \text{ m}^2 \text{ day}^{-1}$) hydraulic transmissivities. The model has

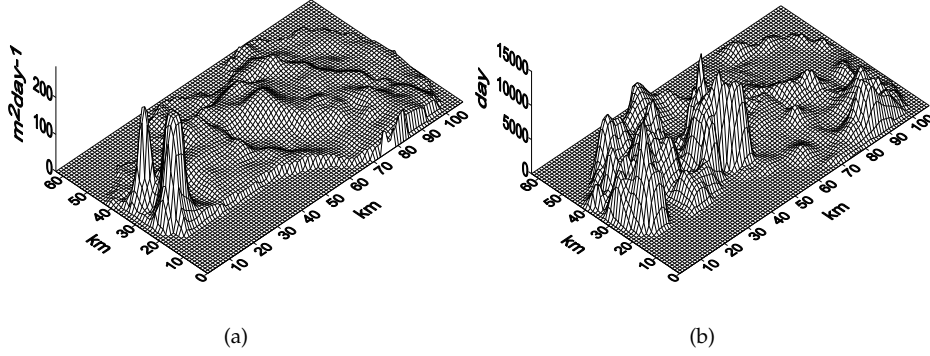


Figure 3.6: Spatial structure of (a) the hydraulic conductance $K^{xx} \Delta z$ for model layer 1, and (b) the vertical resistance R between model layer 1–2.

9 model layers which contain the highly permeable sediments (aquifers), see Figure 3.6a. The low hydraulic permeability in between the aquifers is processed as a vertical conductance, see Figure 3.6b. Within each model layer (2–9) several wells are active (770 in total). The model network consists of 107 columns and 64 rows, each dimensioned 1×1 km. The entire model contains 61,632 model cells, in which 32,949 are active. All others are inactive (no-flow) through an irregular boundary and/or several aquifers which thin out.

3.3.2 Purpose of the Low-dimensional Model

The main purpose of the reduced model was to evaluate whether it could simulate the long-term effects of change in precipitation/evaporation, together with the impact of variable pumping rates for all extraction wells simultaneously for the model layers 2–9. So, the reduced model should be able to vary $n^g = 9$ groups (recharge/evaporation and 8 layers with wells) independently. We want to combine the groups randomly and simulate their combined effects for $n^t = 150$ time steps with $\Delta t = 10$ days. To be able to do so, we need to find patterns which are suitable to simulate at least those scenarios at the desired time scale.

3.3.3 Preparation Phase

Computing Snapshots

To find appropriate patterns to simulate the desired scenarios, we filled a data set with an ensemble of snapshot vectors obtained by the original model. To minimize the data set, we accounted for the linearity of the model. As a result of that, we were able to compute a combined effect for different groups by combining their individual responses (i.e. the principle of *superposition*). Whenever we consider a steady-state simulation, we need to fill the data set with the steady-state responses for each group; so $n^e = n^g = 9$. For our transient simulation we need the patterns to capture the entire response behavior over time; so $n^e = \alpha \cdot n^g$. For each group the

factor α defines the number of snapshots that is necessary to describe its response. To determine which time steps are significant, we separately compute a ‘block’-impulse response for each group. Depending on their gradient in ϕ between the last two time steps, we select model cells with different percentiles (ϱ) for that gradient and plot their corresponding response. We observe that the response, caused by the collection of wells within model layer 2 (Figure 3.7a), reaches its equilibrium at $t \approx 300$ days. At that time we assume that all ($\varrho = 99\%$) model cells have at least reached their steady-state solution. We can represent the behavior of the response efficiently by computing the full-dimensional model at only four time steps: $t_1 = 10, t_2 = 20, t_3 = 70, t_4 = 320$ days. This time strategy does not satisfy the precipitation group which response reaches its steady state for $\varrho = 99\%$ after $t \approx 5,000$ days (Figure 3.7b). Its shape is different from Figure 3.7a and can be described best by nine time steps: $\Delta t_{i+1} = 2\Delta t_i$ with $\Delta t_1 = 10$ days. The chosen time steps are not by definition the best, different combinations will improve or worsen the performance of the low-dimensional model. The most important issue herein is that the frequency should be high initially, and may be decreased as time elapses to obtain a set that fully characterizes the impulse-response optimally [Park & Cho, 1996]. This happens slowly for the recharge component, as the phreatic storage coefficient is much larger ($S = 0.27$) than for confined aquifers where the wells are mainly operating ($S = 0.0007$).

The background hydraulic head ϕ^b is computed with time-independent boundary conditions. These are stored within $\mathbf{q}^b$ and can be e.g. part of the precipitation, wells, dewatering systems, and Dirichlet conditions. In our real-world case all dewatering systems and the Dirichlet conditions along most of the boundaries within each model layer are part of $\mathbf{q}^b$. The ϕ^b is included as a snapshot vector in the data set, so $n^e = 42$.

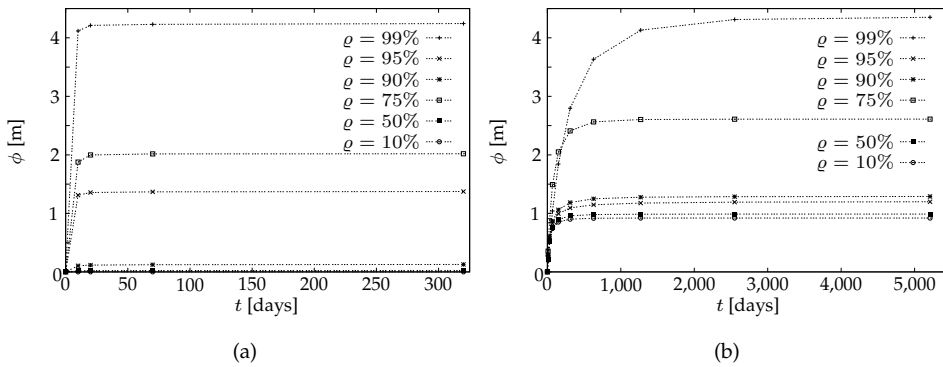


Figure 3.7: Graphs showing the response caused by (a) wells within model layer 2, and (b) by precipitation. The responses are given for different model cells depending on their gradient in ϕ between the last two time steps. Different time series are plotted for various percentiles (ϱ) of that gradient.

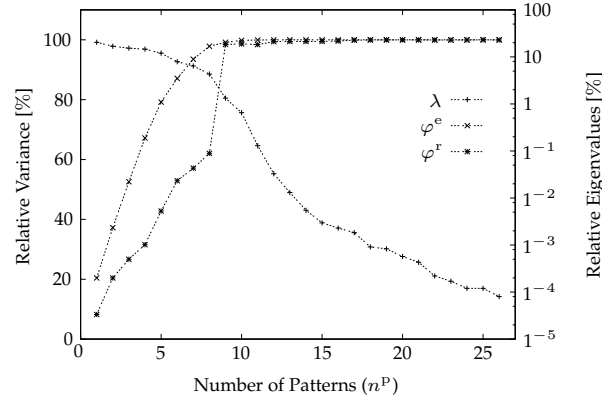


Figure 3.8: Graph of the number of patterns n^P versus the relative eigenvalues λ , the expected φ^e and the reconstructed variance φ^r .

Resulting Patterns

With the $n^e = 42$ snapshot vectors from the previous subsection, we compute the patterns (EOFs) as described in Subsection 3.1.3. The maximal explained variance for a pattern ($\lambda_1 = 20.42\%$) declines rapidly ($\lambda_{>9} \ll 1\%$), see Figure 3.8. The number of patterns should be chosen such that φ^e is more-or-less equal to the *reconstructed* variance φ^r . Otherwise, the low-dimensional model lacks significant patterns, and yields unreliable model predictions. In our case, it is remarkable that the behavior of the full-dimensional model can be described with $\varphi^e = 99.780\%$ by only 10 patterns, see Figure 3.8. Apparently the spatial distribution of ϕ is complex but its behavior in time is not complex at all [Sirovich, 1987; Park & Cho, 1996; Hoffman Jørgensen & Sørensen, 2000]. The cost of it in terms of accuracy is negligible, see Subsection 3.3.4 for elaboration. The spatial distribution of a pattern with the

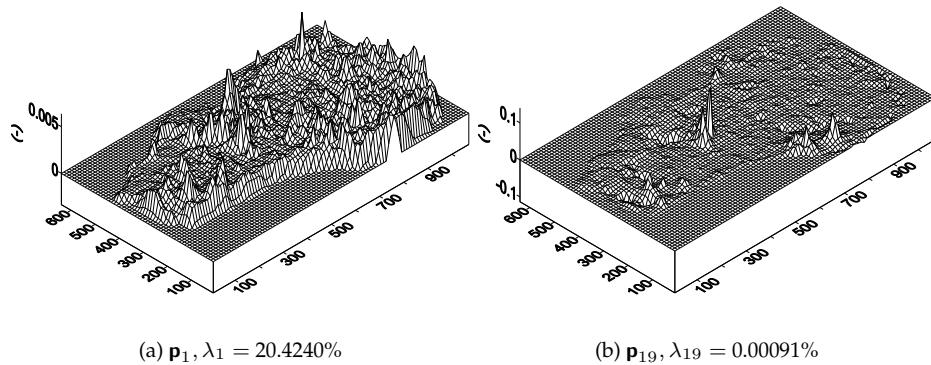


Figure 3.9: Spatial distribution of (a) the most important pattern structure, and (b) the least important pattern structure.

largest eigenvalue ($\mathbf{p}_1, \lambda_1 = 20.42\%$, Figure 3.9a) shows a strong similarity to the global model behavior, e.g. we can observe the impact of precipitation. For a less important pattern ($\mathbf{p}_{19}, \lambda_{19} = 0.00091\%$, Figure 3.9b), such physical processes can be still recognized, though locally. If we look at patterns as if they represent such physical processes, we should expect most of the coefficients r to be zero whenever we compute $\mathbf{q}$, for example only for precipitation (Figure 3.10). Unfortunately this is not the case and almost all coefficients are unequal to zero; they range between -17 and 10. We conclude that the meaning of patterns is not directly related to physical processes. We refer back to Subsection 3.1.4 in which is explained that *rotation* of patterns may increase their interpretability.

When we look at the distribution of the coefficients r computed with a $\mathbf{q}$ for the $n^g = 9$ groups individually, we observe that all main coefficients vary for each case (Figure 3.11). The absolute value of r is mainly caused by the magnitude of the corresponding boundary condition; the wells in model layer 4 are significantly larger than those in other layers. On the other hand, the coefficients $r_{i>11}$ are much less compared to $r_{i\leq 11}$, which corresponds to their small λ s.

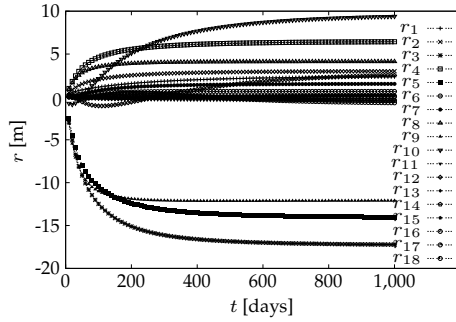


Figure 3.10: Graph of the coefficients r versus t when $\mathbf{q}$ represents precipitation only.

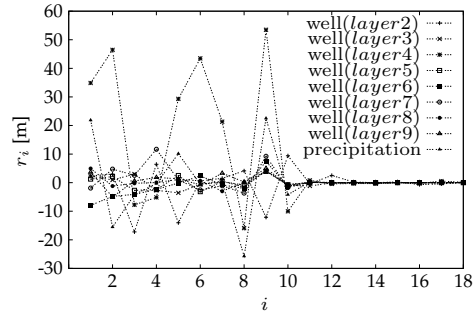


Figure 3.11: Graph of the values for the main coefficients r_i after $n^t = 150$ time steps for each of the $n^g = 9$ groups.

3.3.4 Simulation Phase

Accuracy

The accuracy of a reduced model is expressed as the relative-mean-absolute error RMAE, and the relative-root-mean-square error RRMSE [Overschee van & de Moor, 1996]:

$$\text{RMAE} = 100 \cdot \frac{1}{n^t} \sum_{i=1}^{n^t} \frac{\|\phi(t_i) - \hat{\phi}(t_i)\|}{\|\phi(t_i) - \phi^b\|} [\%], \quad (3.37a)$$

$$\text{RRMSE} = 100 \cdot \frac{1}{n^t} \sum_{i=1}^{n^t} \sqrt{\frac{\|\phi(t_i) - \hat{\phi}(t_i)\|^2}{\|\phi(t_i) - \phi^b\|^2}} [\%]. \quad (3.37b)$$

Both errors give a percent error conditioned on the difference between ϕ and the background hydraulic head ϕ^b . In this approach we accept larger errors as the difference between ϕ and ϕ^b increases. We have computed the RMAE and the RRMSE for the SSPM, SSPM^a and GPM over $n^t = 150$ time steps, see Figures 3.12a-b. For all methods the RMAE and RRMSE decline eventually whenever we include more patterns. The strong reduction in RMAE and RRMSE between $n^p = 9$ and $n^p = 10$ can be explained because the reduced model simulates scenarios with 9 independent groups (Subsection 3.3.2) and hence beyond that point, the accuracy gathers strength. Of course, it is strongly related to the purpose of the reduced model, but an error of RMAE = 0.5% and RRMSE = 1.0% should be acceptable. It should be remarked that the SSPM reaches this criterion at $n^p = 16$, whereas the GPM reaches this accuracy at $n^p = 22$ and the SSPM^a at $n^p = 28$. The latter are affected by the additional numerical error (Subsection 3.2.4, item 1), that can increase the inaccuracy temporary (e.g. $n^p = 13$) but declines eventually whenever n^p is enlarged. This is

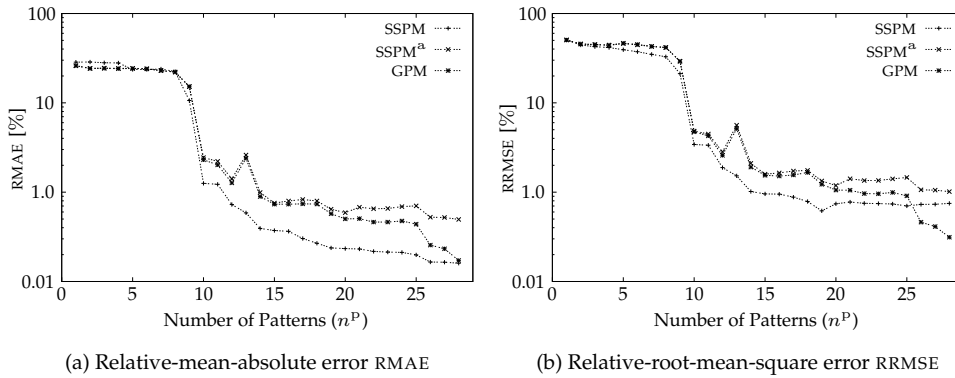


Figure 3.12: Graphs of n^p versus the errors RMAE and RRMSE for the SSPM, SSPM^a and GPM over $n^t = 150$ time steps.

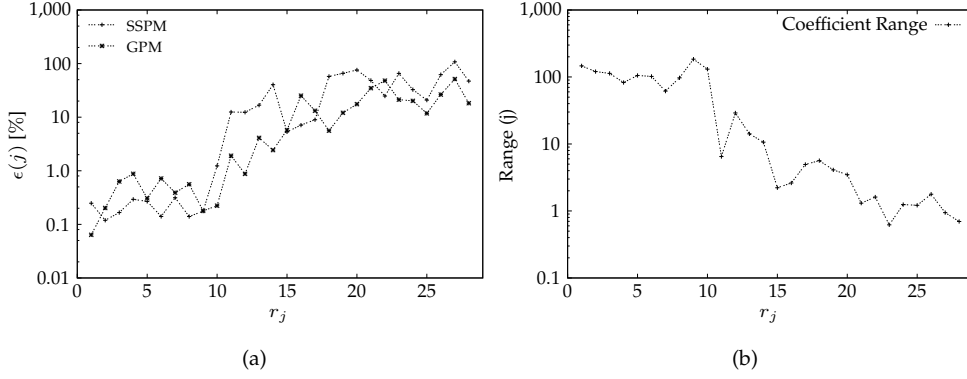


Figure 3.13: Graphs of the time-dependent coefficient r_j versus (a) the relative error $\epsilon(j)$ for the SSPM and GPM, and (b) the range for each coefficient r_j over $n^t = 150$ time steps for the SSPM.

supported because for $n^p = 28$ the RMAE becomes almost similar to the SSPM and GPM. This does not occur with the SSPM^a which is more sensitive to the inclusion of unreliable patterns with small eigenvalues ($n^p > 20$, see Subsection 3.2.4 item 1–2). A reason for this could be that the SSPM^a uses pattern values itself, instead of the pattern derivative of space.

To analyse the SSPM and GPM, we computed $\mathbf{r}$ directly from the results obtained by the original model and marked them as the truth, so:

$$\mathbf{r}^t(t_i) = \mathbf{P}^T [\boldsymbol{\phi}(t_i) - \boldsymbol{\phi}^b]. \quad (3.38)$$

To quantify how well each of the time-dependent coefficients r , computed with the different projection methods, resembles r^t we computed

$$\epsilon(j) = 100 \times \sqrt{\frac{\sum_{i=1}^{n^t} [r_j^t(t_i) - r_j(t_i)]^2}{\sum_{i=1}^{n^t} [r_j^t(t_i)]^2}} [\%] \quad (3.39)$$

for each coefficient r_j separately. Whenever we plot $\epsilon(j)$ for the SSPM and GPM (Figure 3.13a), we see that the coefficients r_j ; $j \in \{18, \dots, 28\}$ show a large inaccuracy (50–110%). They correspond to patterns with very small eigenvalues ($\varphi_i < 0.001\%$, Figure 3.8) that are numerically unreliable. This phenomenon affects the SSPM more seriously than the GPM. It should be noticed, however, that patterns with tiny φ values contribute limited to the complete solution of $\hat{\phi}$, as their r_j range (maximum r_j value minus minimum r_j value) is significantly (almost two orders of magnitude) smaller, see Figure 3.13b.

Mass Conservation

The results of a groundwater model are often used to determine the velocity vectors which can be fed into a transport model. To evaluate whether the results, obtained

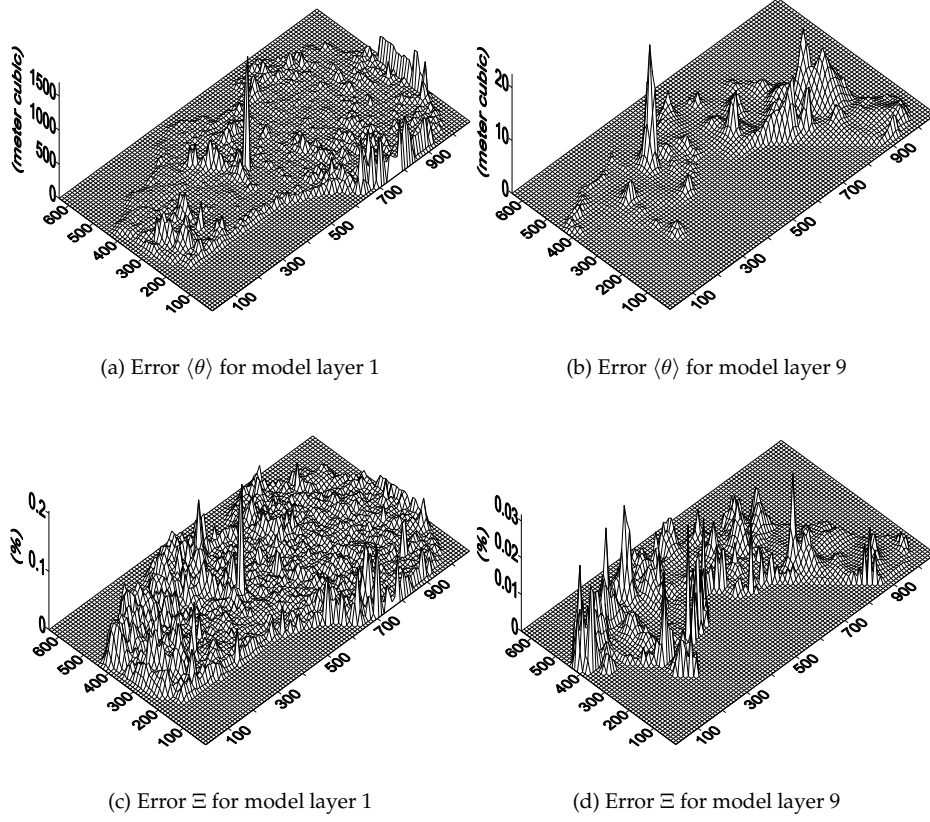


Figure 3.14: Spatial distribution of the errors θ and Ξ over $n^t = 150$ time steps with $n^p = 19$.

by a low-dimensional model are accurate enough, we compute an additional error which quantifies the difference in mass distribution, so:

$$\Xi_k = \frac{1}{n^t} \sum_{i=1}^{n^t} \frac{\theta_k(t_i)}{\sum_{j=1}^6 \|(q_j)_k\|(t_i)} \cdot 100\% ; k \in \{1, \dots, n^m\}, \quad (3.40a)$$

$$\theta_k(t_i) = \sum_{j=1}^6 \|(q_j)_k - (\hat{q}_j)_k\|(t_i), \quad (3.40b)$$

where $(q_j)_k$ is the flow $[\text{L}^3\text{T}^{-1}]$ over six cell faces j for grid cell k and $(\hat{q}_j)_k$ is the equivalent obtained by the reduced model. The error θ_k has been computed as a time average for each grid cell k and plotted for model layer 1 and 9, see Figure 3.14a-b. For a single grid cell the maximum value for θ is $\approx 1,560$ and $\approx 2,121 \text{ m}^3\text{day}^{-1}$ for model layer 1 and 9, respectively. Most of the grid cells possess a value for θ which is considerable less. Whether this error in the mass balance is small enough, such that the coupling with a transport model and/or particle

tracking program is sustained, we computed the relative error Ξ (Equation 3.40a), see Figure 3.14c-d. The value for Ξ is acceptable within most of the model domain with some isolated peaks which are still less than $\approx 0.2\%$. Table 3.1 presents the total water balance for the low- and full-dimensional model, subdivided for each model layer. The error in the water balance is for each model layer within acceptable ranges. Within the low-dimensional model there is $\approx 1.7 \cdot 10^6 \text{ m}^3$ more inflow and $\approx 2.3 \cdot 10^6 \text{ m}^3$ more outflow which is only $\approx 0.16\%$ of the total water balance.

Table 3.1: Total in- and outflow for the original and reduced model ($n^p = 19$).

Layer	Original Model		Reduced Model		Difference [%]	
	In [10^6 m^3]	Out [10^6 m^3]	In [10^6 m^3]	Out [10^6 m^3]	ΔIn	ΔOut
1	770.9	-751.1	772.5	-753.1	0.201	0.264
2	446.1	-705.4	446.2	-705.4	0.039	0.003
3	182.8	-349.5	182.8	-349.7	0.033	0.056
4	143.3	-225.7	143.3	-225.7	0.007	0.027
5	121.9	-160.5	121.9	-160.5	-0.006	0.012
6	103.0	-206.2	103.0	-206.5	-0.001	0.008
7	85.3	-103.7	85.2	-103.7	-0.030	-0.003
8	117.9	-299.7	117.9	-299.8	-0.033	0.007
9	29.7	-139.8	29.7	-139.8	-0.099	-0.008
Total	2,000.9	-2,941.6	2,002.6	-2,943.9	0.084	0.079

Results

In Figure 3.15a we plotted the mean hydraulic head

$$\bar{\phi} = \frac{1}{n^t} \sum_{i=1}^{n^t} \phi(t_i) \quad (3.41)$$

over $n^t = 150$ time steps, as computed with the original model and with the reduced model (Figure 3.15b). The model is based upon a SSPM with $n^p = 16$ patterns and it is very well capable of simulating hydraulic heads within a complex distribution of hydraulic transmissivities and external sources. It is interesting what RMAE (Equation 3.37b) looks like as we compute it for each grid cell k as:

$$\text{RMAE}_k = 100 \cdot \frac{1}{n^t} \left(\frac{\sum_{i=1}^{n^p} \|\phi_k(t_i) - \hat{\phi}_k(t_i)\|}{\Delta\phi_k} \right) [\%], \quad (3.42a)$$

$$\Delta\phi_k = \max (\|\phi_k(t_i) - \phi_k^b\|) ; i \in \{1, \dots, n^t\}. \quad (3.42b)$$

For each grid cell k the relative error RMAE_k is weighed based on $\Delta\phi_k$, which is the maximum fluctuation for grid cell k . The results of these computations are given for model layer 1 in Figures 3.16a-b, and although the $\text{RMAE} \approx 0.5\%$ (Figure 3.12a), we see that RMAE_k actually varies. The enlarged errors appear especially in the regions with an intense dewatering system ($\text{RMAE}_k \approx 2\%$) and near the main extraction wells ($\text{RMAE}_k \approx 6\%$). The disturbance of the hydraulic head by external sources is less within model layer 9, and hence the RMAE_k is significant reduced.

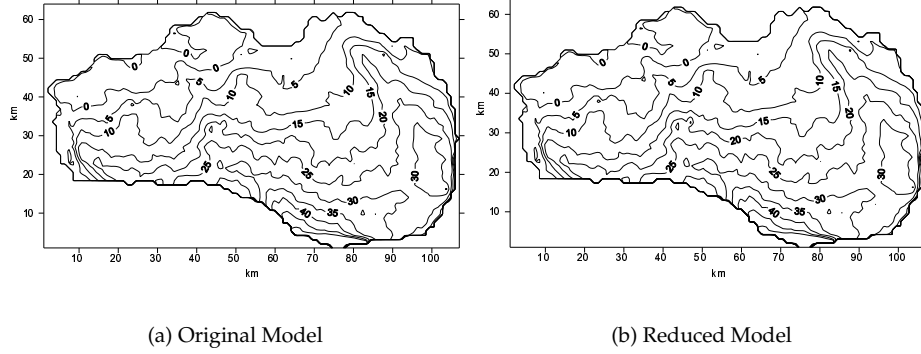


Figure 3.15: Mean hydraulic heads for model layer 1 over $n^t = 150$ time steps for (a) the original model and (b) a reduced SSPM model ($n^p = 16$).

In Figure 3.17a we have plotted a time series for a specific grid cell i, j, k that has $\text{RMAE}_{(i,j,k)} \approx 6\%$. Most of the time $\hat{\phi}$ follows the behavior of ϕ accurately, but at several time steps the difference is enlarged (e.g. $t = 350$ days; $\hat{\phi} - \phi \approx 0.15$ m). These (temporary) errors are caused by a lack of patterns, as we explain only $\varphi^e = 99\%$ with the selected set of $n^p = 16$ patterns. The absence of some or all of the rejected patterns (φ_i ; $i \in \{17-28\}$) is the cause of this specific misbehavior - and we could include them to decrease the error. It should be mentioned that this error occurs only very locally within the model and most of the grid cells perform rather well with a $\text{RMAE}_k \ll 6\%$. In Figure 3.17b we show a time series with $\text{RMAE}_k \approx 0.5\%$. Here, the reduced model simulates the behavior of the model almost perfectly.

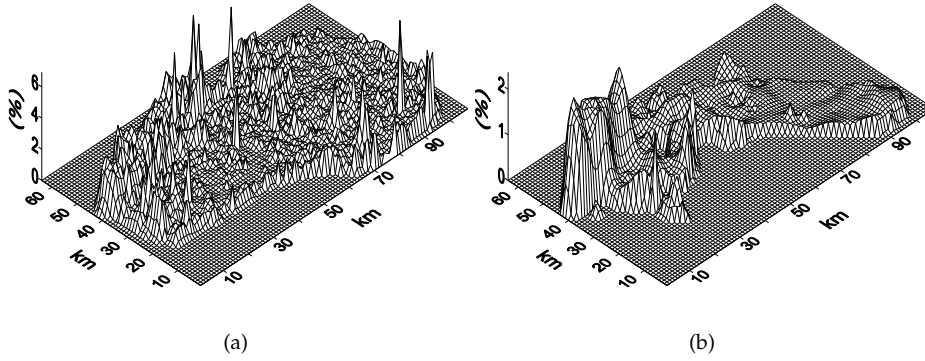


Figure 3.16: Spatial distribution of the relative-mean-absolute error RMAE_k for (a) model layer 1 and (b) model layer 9.

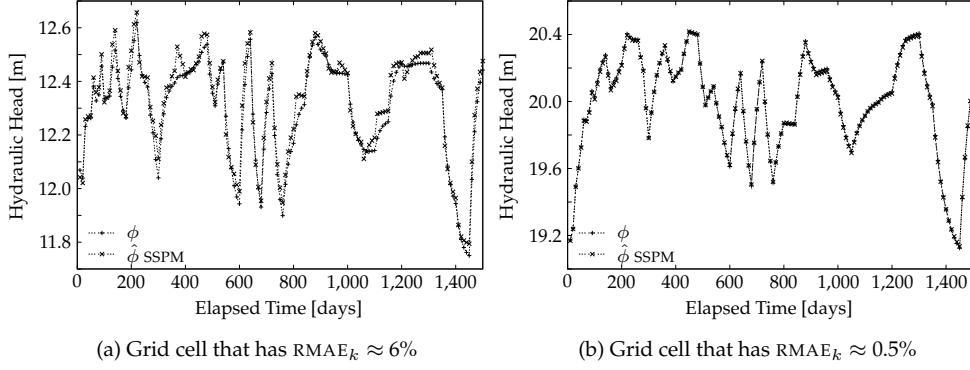


Figure 3.17: Time series of ϕ and $\hat{\phi}$ (SSPM) for different grid cells in the model.

3.3.5 Efficiency

Preparation Time

The key concept of a reduced model is to compute time-independent matrices in advance and to compute as little as possible during the actual simulation. The SSPM, SSPM^a and GPM formulations differ, hence each requires a different computational effort to prepare their matrices $\mathbf{A}$, $\mathbf{S}$, $\mathbf{N}$. We have plotted the preparation time t^P for those methods and they are almost linearly related to n^P (Figure 3.18a). For an acceptable reduced model $t^P \approx 26$ s ($n^P = 16$) and $t^P \approx 13$ s ($n^P = 24$) for the SSPM and GPM, respectively. The SSPM consumes most time because the original model needs to be solved $n^P = 16$ times. For all methods, t^P summed with the total time taken by the computation of the $n^e = 43$ snapshots is still significantly less than one simulation of the original model (≈ 240 s). Therefore it is wise to create a reduced

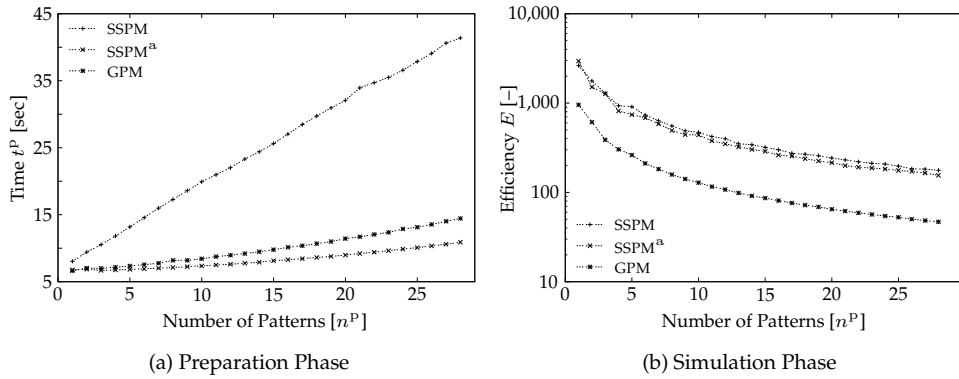


Figure 3.18: Graph of the number of patterns n^P versus (a) the preparation time t^P , and (b) the efficiency E of the actual simulation.

model for a limited range of scenarios, and recreate it whenever the specifications (conductances and/or positions of external sources) for the scenarios changes. In that case, fewer patterns are required for scenarios which benefits the simulation time.

Simulation Time

To quantify the efficiency E of the reduced models, we divided the total time taken by the iterative procedure to solve $\phi(t_i)$, by the total time taken by the reduced models to solve $\mathbf{r}(t_i)$. It results in an $E \approx 625$ ($n^p = 16$) and an $E \approx 70$ ($n^p = 24$) for the SSPM and GPM, respectively. The SSPM method contains one matrix multiplication less and profits from the effort put in the preparation phase (see previous Subsection). Although the efficiency decreases whenever we add more patterns (Figure 3.18b), the gained time reduction remains enormous for all methods.

With a hypothetical model we examined how E relates to n^p , n^m , the number of external sources n^q , and the complexity of the original model $\mathcal{C}$. We took a one-layered model with constant grid cell dimensions and a constant conductance of $100 \text{ m}^2\text{day}^{-1}$ ($\mathcal{C} = 1$). We put a Dirichlet condition on the left and right edges, and recharge within each grid cell and distributed five independent extraction wells throughout the model, so $n^q = n^m$. We have varied n^m and n^p and found by regression (regression coefficients $R^2 > 0.98$) the following relationship for:

$$E(\text{SSPM}) = \mathcal{C} \frac{n^m}{n^q} 0.40 (n^m)^{0.48} \frac{5}{n^p} \text{ and} \quad (3.43a)$$

$$E(\text{GPM}) = \mathcal{C} \frac{n^m}{n^q} 0.12 (n^m)^{0.41} \frac{5}{n^p}. \quad (3.43b)$$

The ratio n^m/n^q determines the efficiency that can be obtained by multiplying only the non-zero elements within $\mathbf{b}$ (Equations 3.24a & 3.32). It is purely a gained time reduction through an optimized multiplication routine which can be significant. An important conclusion, however, is that the efficiency increases with n^m and decreases with n^p . We have plotted a graph of the efficiency E for $\mathcal{C} = 1$ and $n^q/n^m = 1$ in Figures 3.19a & 3.19b. Although $E(\text{GPM})$ is approximately one order of magnitude less than $E(\text{SSPM})$, both methods remain still efficient when n^p increases ($n^p \approx 100$). The total efficiency is most likely higher for practical applications. Namely, for our realistic case study it appeared that $n^m/n^q = 7.1$ and $\mathcal{C} \approx 5$.

3.4 Comparison of the EOF and Lanczos method

This subsection describes briefly the Lanczos model reduction method and gives several synthetic examples to illustrate the differences between the Lanczos method and the previous reduced models based on EOF.

3.4.1 Lanczos Vectors

The idea behind the Lanczos vectors [Lanczos, 1950] is that for general systems, which are often diffuse in nature, the smallest eigenvalues Λ of the governing sys-

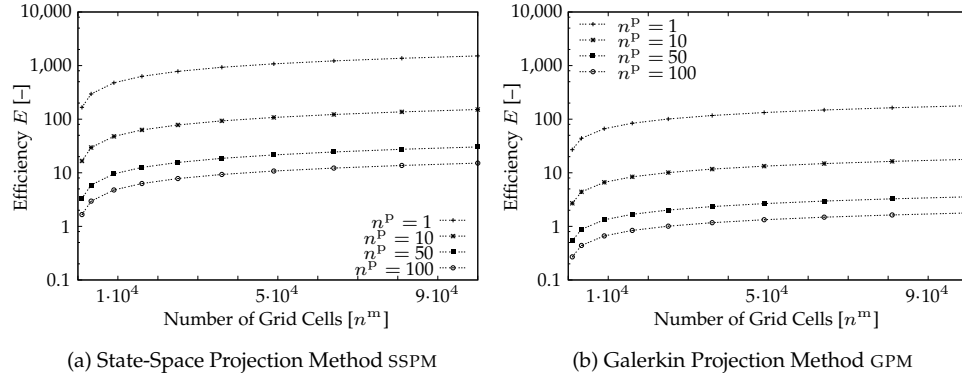


Figure 3.19: Graph of the efficiency E versus the number of grid cells n^m for a hypothetical model with $n^m/n^q = 1.0$ and $C = 1$. The results are presented for different number of patterns n^p for (a) the SSPM and (b) the GPM.

tem matrices form the significant part of the solution [Dunbar & Woodbury, 1989], so

$$\mathbf{A}\mathbf{V} = \mathbf{S}\mathbf{V}\mathbf{\Lambda}, \quad (3.44)$$

where $\mathbf{A}$ is known as the ‘stiffness’ matrix that holds the hydraulic conductances, and $\mathbf{S}$ is known as the ‘capacity’ matrix that represents the storage coefficients. For nonsymmetric matrices the Arnoldi method [Arnoldi, 1951] is developed. The eigenvalue decomposition (3.44) is, however, difficult to implement as matrices are often large. This is solved by the Krylov sequence which is recursive in nature and gives an additional largest eigenvector after each step in the recursion. This avoids the computational burden of determining all the eigenvalues at once and then sort through them to find the smallest ones and discard the rest. Consequently, a scheme that uses $\mathbf{A}^{-1}$ instead as the governing matrix will yield the smallest eigenvalues therefore we rewrite Equation (3.44) as

$$\mathbf{A}^{-1}\mathbf{S}\mathbf{L} = \mathbf{L}\mathbf{T} \quad (3.45)$$

where $\mathbf{T}$ is a tridiagonal matrix with good approximations of the reciprocal eigenvalues λ_i^{-1} , thus

$$\mathbf{T} = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n^1} \\ & & & \beta_{n^1} & \alpha_{n^1} \end{bmatrix} \quad (3.46)$$

where n^1 are the number of Lanczos vectors. The Lanczos vectors $\mathbf{L} = [\mathbf{l}_1, \dots, \mathbf{l}_{n^1}]$ describe an orthonormal basis for a Krylov space ($\mathbf{L}^T \mathbf{S} \mathbf{L} = \mathbf{I}$). The Lanczos vectors which are left out, represent a dynamic behavior within the model that dissipate more quickly than others. These Lanczos vectors have been reported on between 1987–1994 [Dunbar & Woodbury, 1989; Zhang & Woodbury, 2000a,b; Nour-Omid, 1987].

Krylov/Lanczos Sequence

The Krylov sequence uses orthogonal transformations to compute the tridiagonal matrix $\mathbf{T}$ by a recursive scheme, that is, each recursion results in one Lanczos vector $\mathbf{l}_i$ and one more row and column in $\mathbf{T}$ with the components α_i, β_i . The $\mathbf{S}$ orthogonality requirement for a Lanczos vector results in an equation that yields a new Lanczos vector $\mathbf{l}_i$ that is based upon the previous Lanczos vector $\mathbf{l}_{i-1}$ and its corresponding components $\alpha_{i-1}, \beta_{i-1}$, see Figure 3.20. The starting Lanczos vector $\mathbf{l}_1$ should contain states that are of primary interest to the problem under consideration. In this example the first Lanczos vector is equal to a normalized steady-state problem without the initial boundary conditions $\phi(t_0)$. Even though there are several references to $\mathbf{A}^{-1}$, the procedure does not require this inversion of $\mathbf{A}$. Instead of computing $\mathbf{A}^{-1} \mathbf{p}$ we can obtain the results by solving $\mathbf{A} \mathbf{x} = \mathbf{p}$ for the original model. An important issue for the Krylov/Lanczos sequence is the stopping criterion. Dunbar & Woodbury [1989] used a ‘participation factor’ $\rho_i = \mathbf{l}_i^T [\mathbf{b} - \mathbf{A} \phi]$ that describes the contribution of the Lanczos vector to the initial forcing vector without Dirichlet conditions, thus $\mathbf{b} - \mathbf{A} \phi$. Whenever $\mathbf{l}_i$ is orthogonal to $\mathbf{b} - \mathbf{A} \phi$, $\rho_i = 0$ and the recursion can be terminated; the renewed Lanczos vector is irrelevant for the current forcing definitions. If more than one source is present, the vector $\mathbf{b}$ should contain a sum of terms [Nour-Omid, 1987]. Finally, the Krylov/Lanczos sequence is subject to loss of orthogonality through round-off errors. This can be monitored during the recursion by computing $\mathbf{l}_{i+1}^T \mathbf{S} \mathbf{l}_i$. When loss of orthogonality has occurred the newly generated Lanczos vector is orthogonalized against all preceding vectors by use of the Gram-Schmidt procedure.

```

 $\mathbf{x} = \mathbf{A}^{-1}[\mathbf{b} - \mathbf{A} \phi](t_0)$ 
 $\mathbf{l}_0 = 0$ 
 $\beta_1 = (\mathbf{x}^T \mathbf{S} \mathbf{x})^{\frac{1}{2}}$ 
 $\mathbf{l}_1 = \mathbf{x} \beta_1^{-1}$ 
 $\mathbf{y} = \mathbf{S} \mathbf{l}_1$ 
For  $i=1, 2, \dots$ 
   $\mathbf{r} = \mathbf{A}^{-1} \mathbf{p} - \beta_i \mathbf{l}_{i-1}$ 
   $\alpha_i = \mathbf{l}_i^T \mathbf{S} \mathbf{x}$ 
   $\mathbf{x} = \mathbf{x} - \alpha_i \mathbf{l}_i$ 
   $\mathbf{y} = \mathbf{M} \mathbf{x}$ 
   $\beta_{i+1} = (\mathbf{x}^T \mathbf{S} \mathbf{x})^{\frac{1}{2}}$ 
  Terminate loop if desired
   $\mathbf{l}_{i+1} = \mathbf{x} \beta_{i+1}^{-1}$ 
   $\mathbf{y} = \mathbf{y} \beta_{i+1}^{-1}$ 
End i

```

Figure 3.20: Krylov/Lanczos Algorithm [Dunbar & Woodbury, 1989]

$$\mathbf{l}'_i = \mathbf{l}_i - \frac{\mathbf{l}_1^T \mathbf{S} \mathbf{l}_i}{\mathbf{l}_1^T \mathbf{S} \mathbf{l}_1} \mathbf{l}_1 - \dots - \frac{\mathbf{l}_{i-1}^T \mathbf{S} \mathbf{l}_i}{\mathbf{l}_{i-1}^T \mathbf{S} \mathbf{l}_{i-1}} \mathbf{l}_{i-1} \quad (3.47)$$

where $\mathbf{l}'_i$ is the orthogonalized renewed Lanczos vector. Experiences showed that this partial reorthogonalization is usually required after 5–10 Lanczos steps.

3.4.2 Lanczos Reduction Method

The Lanczos reduction method reduces the size of the original model by means of a Rayleigh-Ritz procedure and is therefore comparable to the SSPM, SSPM^a and GPM methods. However, this method uses Lanczos vectors $\mathbf{L}$ to reduce the system, instead of patterns $\mathbf{P}$. So, we write the first-order differential equation for ground-water as

$$\mathbf{A}\phi - \mathbf{S}\frac{\partial\phi}{\partial t} = \mathbf{b} - \mathbf{A}\phi_0, \quad (3.48)$$

where $\mathbf{A}\phi_0$ corrects the right-hand-side vector $\mathbf{b}$ for initial conditions because they took no part in the Lanczos recursion. We define an approximate head $\hat{\phi}$ as

$$\hat{\phi} = \phi_0 + \mathbf{L}\mathbf{r}. \quad (3.49)$$

Multiplication of Equation (3.48) with $\mathbf{L}^T \mathbf{S} \mathbf{A}^{-1}$ gives:

$$\mathbf{L}^T \mathbf{S} \mathbf{A}^{-1} \mathbf{A} \mathbf{L} \mathbf{r} - \mathbf{L}^T \mathbf{S} \mathbf{A}^{-1} \mathbf{S} \mathbf{L} \frac{\partial \mathbf{r}}{\partial t} = \mathbf{L}^T \mathbf{S} \mathbf{A}^{-1} (\mathbf{b} - \mathbf{A}\phi_0), \quad (3.50)$$

which reduces to a low-dimensional ODE

$$\mathbf{r} - \mathbf{T} \frac{\partial \mathbf{r}}{\partial t} = \mathbf{L}^T \mathbf{S} \mathbf{A}^{-1} (\mathbf{b} - \mathbf{A}\phi_0), \quad (3.51)$$

because

$$\mathbf{A}^{-1} \mathbf{A} = \mathbf{I} \quad (3.52a)$$

$$\mathbf{L}^T \mathbf{S} \mathbf{L} = \mathbf{I} \quad (3.52b)$$

$$\mathbf{T} = \mathbf{L}^T \mathbf{S} \mathbf{A}^{-1} \mathbf{S} \mathbf{L}. \quad (3.52c)$$

This model operates in Lanczos space and could be easily solved by a time-integration technique (e.g. Euler) such that

$$\left(\mathbf{I} - \frac{1}{\Delta t} \mathbf{T} \right) \mathbf{r}(t_i) = -\frac{1}{\Delta t} \mathbf{T} \mathbf{r}(t_{i-1}) + \mathbf{L}^T \mathbf{S} \mathbf{A}^{-1} [\mathbf{b}(t_i) - \mathbf{A}\phi(t_0)]. \quad (3.53)$$

The system $\mathbf{T}$ could be easily decoupled by means of a further eigenvalue decomposition of $\mathbf{T}$ [Dunbar & Woodbury, 1989]. The transient model starts with zero initial conditions $\mathbf{r}(t_0) = 0$, however, if boundary conditions are time dependent, the reduced model suggests that the right-hand-side vector needs to be evaluated for each time step. In Dunbar & Woodbury [1989]; Zhang & Woodbury [2000b] different procedures are described to avoid this. They decompose the time-dependent vector into spatial and temporal components just like the Rayleigh-Ritz concept for reduced models. Furthermore, they group the boundary conditions into parts that have identical history patterns [Gambolati, 1993; Dunbar et al., 1994]. These techniques are based on the principle of superposition and resemble strongly the procedure described earlier for the computation of snapshots, see Subsection 3.3.3. The Lanczos approach is very general because it yields patterns that are dependent to the geometry of the current source terms but independent to the source strength and time discretization.

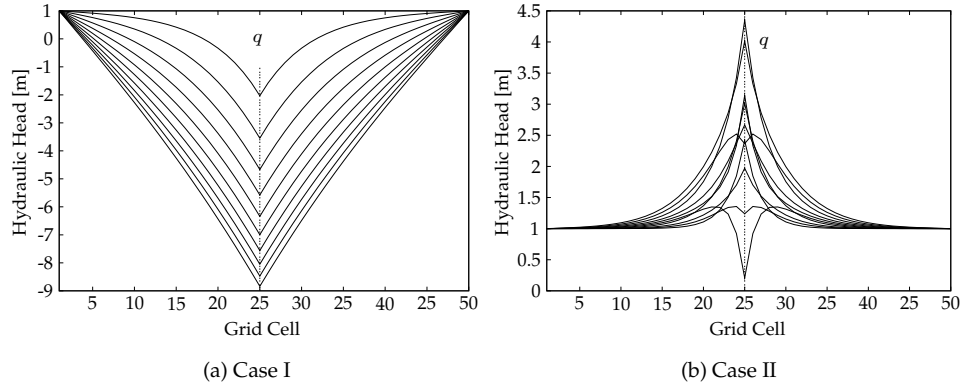


Figure 3.21: Computed hydraulic head ϕ for the synthetic case I & II.

3.4.3 Synthetic Cases

This subsection illustrates the performance and accuracy of the EOF and Lanczos decomposition by means of two synthetic cases (Figure 3.21a-b). Both cases are one dimensional (50 columns) and possess an extraction well in the middle (grid cell 25). For Case I the extraction rate was time-constant and equal to $100 \text{ m}^3 \text{ day}^{-1}$. For Case II we defined the extraction rate as a random variable that varied between $-125 \text{ m}^3 \text{ day}^{-1}$ and $125 \text{ m}^3 \text{ day}^{-1}$.

We found five main patterns by means of an EOF analysis which explained totally $\varphi^e \approx 100\%$ of the model variance (Figure 3.22a). We computed five Lanczos vectors and plotted them in Figure 3.22b. The corresponding participation factor ρ , however, did not give us a good criterion whether to terminate the Lanczos recursion. Nevertheless, both reduced model were capable of reconstructing the original ‘signal’ very well (Figure 3.23). The Lanczos method is ‘least’ accurate at the location of the extraction well and the EOF method is ‘most’ accurate at that position instead. This can be explained because EOF modes are most sensitive to areas with high variances. As a result those particular areas are well represented by the patterns **P**. In contrast to this, Lanczos modes **L** will yield a maximum error in the first several time steps, or the moment the extraction rate is changed for a particular location. That is because the Lanczos vectors which are left out represent a dynamic behavior within the model that dissipates more quickly than others. Therefore, the accuracy of Lanczos modes will increase when the time discretization (Δt) is increased.

The difference between the EOF method and the Lanczos method depicts more clearly within Case II (Figure 3.21b). Since Lanczos vectors are independent of the strength of steady-state boundary conditions, the leading Lanczos vectors are identical to those obtained in Case I. They differ only in their participation factors ρ , see Figure 3.24a. Moreover, the accuracy of this type of model behavior is clearly less predicted by these Lanczos modes (Figure 3.24b). The randomness of the extraction

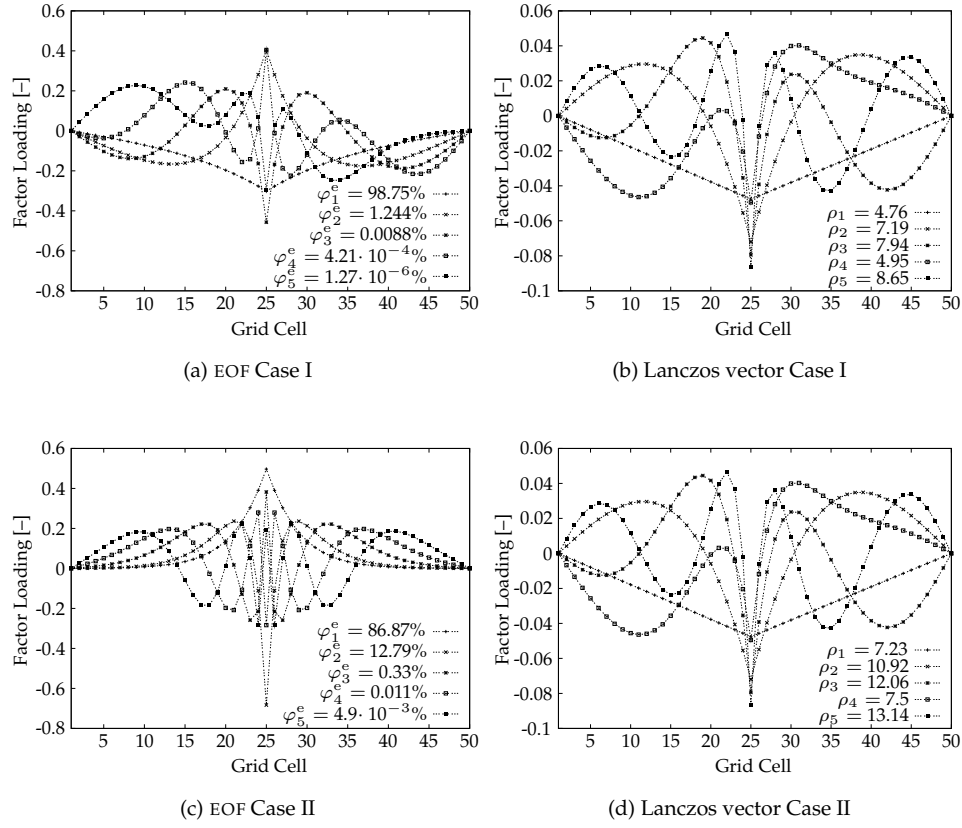


Figure 3.22: Computed EOF and Lanczos vectors for the synthetic case I & II.

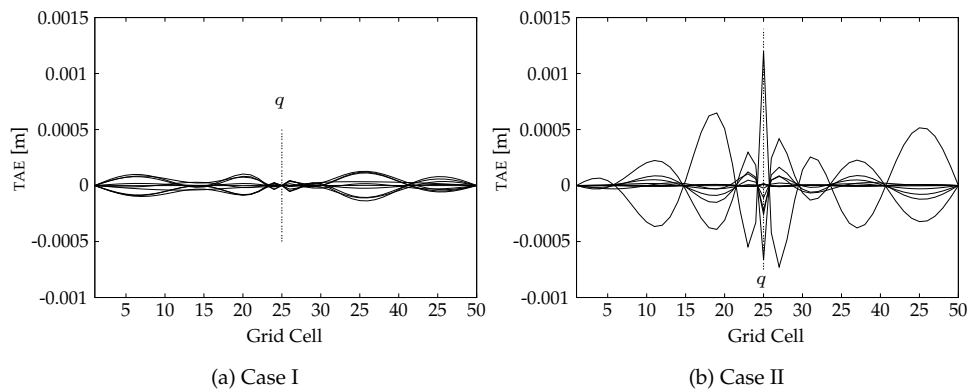


Figure 3.23: Computed total-absolute error TAE of the hydraulic head ϕ for the synthetic case I & II.

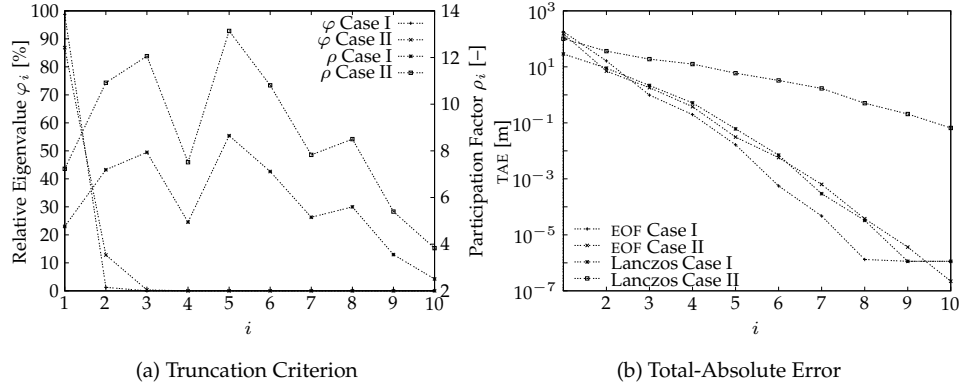


Figure 3.24: Graph of (a) the criterion used for truncation of the EOF and Krylov spaces, and (b) the total-absolute errors for the EOF and Lanczos methods for the synthetic cases I & II.

rates manifests a dynamic model behavior close to the well, that is not captured by the Lanczos decomposition. In contrast to this, the EOF method is very well capable of simulating this behavior. The EOF modes differ significantly from the modes obtained in Case I and describe the transient behavior rather than the steady-state solution, see Figure 3.22c.

3.5 Conclusions and Recommendations

This chapter elaborates several methods to develop a reduced model of a linear numerical groundwater flow model. We defined a reduced model structure that consists of a set of spatial patterns with time-varying coefficients. The patterns are called Empirical Orthogonal Functions (EOFs), and they span a subspace of model results that captures most of the relevant information of the original model. The time-varying coefficients are derived by projecting the original model onto the orthogonal EOFs, starting from a partial differential equation (*Galerkin Projection Method* GPM) or from a state-space formulation (*State-Space Projection Method* SSPM). We have used the two different projection strategies to simulate different scenarios with a realistic case study. Both methods were able to simulate the scenarios within an acceptable accuracy constraint (relative-mean-absolute Error $\text{RMAE} < 0.5\%$). To obtain this accuracy we needed more patterns for a GPM ($n^p = 24$) than for a SSPM ($n^p = 16$). This is caused by an additional error which affects only the GPM, but decreases whenever we include more patterns. Due to its mathematical formulation, the SSPM is numerically more sensitive to round-off errors, especially with patterns that explain a very small amount of variance ($\phi_i \ll 0.001\%$) of the original model. Another consequence of its formulation is that the reduced model contains one matrix multiplication less than the one created by a GPM. Therefore we achieved a larger time reduction with a SSPM than with a GPM. Specific for the realistic case study, explained in this chapter, we achieved an efficiency (time reduction) of ap-

proximately 625 and 70, respectively. For both methods, the efficiency increases with the number of grid cells and the complexity of the model.

We illustrated the differences between the EOF method and the Lanczos method which lacks the use of snapshots. It decomposes the governing system matrices and uses the smallest eigenvalues to reconstruct the significant part of the solution. The Lanczos vectors are independent of the strength of steady-state boundary conditions, however, they are less capable of reconstructing a behavior that reaches no steady-state equilibrium. The EOF method is very well capable of simulating such a behavior.

In future research we intend to focus on the possibility to apply the discussed reduced model to non-linear 3D groundwater flow models. Experiences from other sciences showed that the reduced model is also valuable in Reduced Order Kalman Filters [Heemink *et al.*, 2001] and parameter estimation [Delay *et al.*, 2001; Park & Cho, 1996]. The latter is the main topic of the forthcoming chapters.

4

Inverse Modeling of Groundwater Flow using Model Reduction

Abstract. Numerical groundwater flow models often have a very high number of model cells ($\gg$ million). Such models are computationally very demanding, which is disadvantageous for inverse modeling. This chapter describes a low-dimensional formulation for groundwater flow that reduces the computational burden necessary for inverse modeling. The formulation is a projection of the original groundwater flow equation on a set of orthogonal patterns (i.e. a Galerkin Projection). The patterns (Empirical Orthogonal Functions) are computed by a decomposition of the covariance matrix over an ensemble of model solutions. Those solutions represent the behavior of the model as a result of model impulses and the influence of a chosen set of parameter values. For an interchangeable set of parameter values, the patterns yield a low-dimensional model, as the number of patterns is often small. An advantage of this model is that the adjoint is easily available and most accurate for inverse modeling. For several synthetic cases the low-dimensional model was able to find the global minimum efficiently, and the result was comparable to that of the original model. For several cases, our model even converged where the original model failed. Our results demonstrate that the proposed procedure results in a 60% time reduction to solve the groundwater flow inverse problem. Greater efficiencies can be expected in practice for large-scale models, with a large number of grid cells, that are used to compute transient simulations.

*This chapter is adapted from Vermeulen, P.T.M, A.W. Heemink, & J.R. Valstar, Inverse Modeling of Groundwater Flow using Model Reduction, **Water Resources Research**, 41, W06003, doi:10.1029/2004WR003698, 2004*

GROUNDWATER MODELERS ARE CHALLENGED to simulate the natural system with numerical models. Nowadays, these models consist of large model networks that describe reality in more and more detail. As a consequence, the computational demands are increased, which is especially undesirable for inverse modeling. Here we adjust a set of variables (e.g. parameter values) such that they decrease the difference (objective function value) between the measured and the computed heads. Extensive reviews of inverse models in geohydrology are given in: *Carrera & Neuman* [1986]; *Cooley* [1985]; *Yeh* [1986], among others. Several techniques compute sensitivities (i.e. the first derivatives of heads with respect to a single estimate of parameters) [*Cooley*, 1985; *Cooley & Hill*, 1992; *Hill*, 1990; *Mehl & Hill*, 2003] to find an approximate gradient of the objective function. Other, more sophisticated and efficient techniques use the adjoint method [*Courant & Hilbert*, 1953; *Townley & Wilson*, 1985] to find the exact gradient for all estimates simultaneously. Beside these minimization methods there are different methods of assigning parameter values. Most commonly they are assigned to predefined zones while another, recently developed, technique applies parameter values according to influence functions (representer functions) [*Bennet*, 1992; *Valstar et al.*, 2004]. Other recently developed geostatistical techniques generate hundreds of possible solutions with different structures of the estimate variables to address the notion of uncertainty of the solutions [*Delay et al.*, 2001; *Ginn & Cushman*, 1990; *McLaughlin & Townley*, 1996]. It is beyond the scope of this chapter to compare the differences between the mentioned techniques, as this chapter describes a different approach that could affect the time efficiency of those techniques positively.

Roughly, the time efficiency can be increased by (1) use of a more time-efficient solver [*Mehl & Hill*, 2001], (2) applying a coarse grid and/or a locally refined grid [*Mehl & Hill*, 2003; *Wen et al.*, 2003; *Bennet et al.*, 1996], and/or (3) formulating a low-dimensional model that is capable of simulating the important behavior of the original model [*Cazemier et al.*, 1998; *Hoffman Jørgensen & Sørensen*, 2000; *Krysl et al.*, 2001; *Newman*, 1996a; *Park & Cho*, 1996]. For groundwater hydrology *Vermeulen et al.* [2004a,b] obtained such a model by selecting a set of patterns (Empirical Orthogonal Functions (EOF)) that are most representative of the behavior of the original model. Recently, *Delay et al.* [2001] have used EOF analysis to determine the uncertainty of a stochastic inversion method. In this chapter, the EOFs can be seen as a

description of the *subspace* in which a possible solution of the original model exists. For hydrological models, it appears that in many directions of the original space the solution is constant. The shape of a well drawdown (its direction in space) will not change dramatically for a change in its amplitude: it will not move around the model domain. This phenomenon makes it possible to *project* the original partial differential equation upon a given set of patterns to create a low-dimensional model. Because often the number of patterns is small, such a model reduces the computation time needed (with approximately 2–3 orders of magnitude, Vermeulen *et al.* [2004b]). This makes the reduced model suitable for inverse modeling problems [Park *et al.*, 1999], by which the model needs to be evaluated recurrently to find the set of variables with the minimal valued objective function.

In our previous work [Vermeulen *et al.*, 2004b], the patterns are representative for the model behavior for one particular set of parameter values. In this chapter, the patterns are extended such that they include the sensitivity with respect to different sets of parameter values. The set of parameter values can now be sequentially perturbed, while the set of patterns remains constant and yields a reduced model, over and over again, which is accurate for that combination of parameter values. Another advantage of the reduced model is that its adjoint, which is used to obtain the gradient of the objective function, is easily available (in contrast to that of the original model).

In this chapter we describe briefly the method to create a reduced model by means of the Galerkin Projection in Section 4.1. Section 4.2 then explains the process of inverse modeling by use of a reduced model. Finally, section 4.3 describes the performance and the resulting reduction in computation time for a realistic three-dimensional inverse modeling problem.

4.1 Methodology

4.1.1 Formulation of Groundwater Flow

Three-dimensional groundwater flow, with a uniform density and viscosity, can be described by Darcy's law and the equation of continuity. This yields the following partial differential equation PDE [McDonald & Harbaugh, 1988]:

$$\frac{\partial}{\partial x} \left[\alpha K^{xx} \frac{\partial \phi}{\partial x} \right] + \frac{\partial}{\partial y} \left[\alpha K^{yy} \frac{\partial \phi}{\partial y} \right] + \frac{\partial}{\partial z} \left[\alpha K^{zz} \frac{\partial \phi}{\partial z} \right] - \gamma q = -\beta S \frac{\partial \phi}{\partial t} \quad (4.1)$$

where α, β, γ are estimates of parameters [–], K^{xx}, K^{yy}, K^{zz} are the hydraulic permeabilities [LT^{-1}] along the x, y, z direction, respectively. ϕ is the hydraulic head [L], S is the storage coefficient [L^{-1}], t is time [T], and q is a fluid source/sink term [T^{-1}]. One way to solve Equation (4.1) is to use the finite-difference approach to discretize the equation for a mesh of grid cells n^m in space and to solve the entire set of n^m equations in discrete time.

In many situations, the model becomes more reliable when a value for α, β, γ can be found that reduces the difference between a set of observations ϕ^o and its

corresponding simulated valued ϕ . A quantity that determines this difference is the sum of weighed squared residuals:

$$J = \frac{1}{n^t n^o} \sum_{i=1}^{n^t} \sum_{j=1}^{n^o} [\phi_j^o(x, y, z, t_i) - \phi_j(x, y, z, t_i, \alpha, \beta, \gamma)]^2 w_j \quad (4.2)$$

where J is the objective function [L^2], $\phi_j^o(x, y, z, t_i)$ is the j^{th} observation out of n^o for time step i located at position (x, y, z) , and w_j is the weight factor for observation j . There are several ways to find the optimal values for the estimate variables that minimize J [Carrera & Neuman, 1986; Cooley, 1985; Tarantola, 1987; Press et al., 1992]. In this chapter, we use a reduced model to limit the computation time of this minimization problem.

4.1.2 Reduced Model

Model Structure

Assume that ϕ can be expressed as a linear combination that can be written by the following equation:

$$\hat{\phi}(x, y, z, t_i, \alpha, \beta, \gamma) = \sum_{i=1}^{n^p} p_i[x, y, z, \phi_j(\mathcal{L}_j)] r_i(\alpha, \beta, \gamma, t_i) \quad (4.3)$$

where $\hat{\phi}$ is the approximated hydraulic head [L], p_i is the i^{th} pattern value out of n^p [-], and r_i is the i^{th} time-dependent coefficient [L]. It should be noticed that r_i depends on the current values for the estimate variables α, β, γ and p_i depends on ϕ which is computed for several sets of estimate variables $\mathcal{L}_j \in \{\alpha_j, \beta_j, \gamma_j\}$. It seems disadvantageous that the patterns need to be computed in advance, but for the actual model simulation, this only concerns a limited number of coefficients r_i . For this reason the following ordinary differential equation is defined that describes dr/dt :

$$\mathcal{F}[\alpha K^{xx}, \alpha K^{yy}, \alpha K^{zz}, \beta S] r - \gamma \underline{q} = \frac{dr}{dt}. \quad (4.4)$$

This equation contains a linear function $\mathcal{F}$ that depends on the system properties of the original model ($K^{xx}, K^{yy}, K^{zz}, S$) and the estimate variables α, β . The variable $\underline{q}$ is a reduced boundary condition [LT^{-1}]. For inverse modeling, the function $\mathcal{F}$ is recomputed for each perturbation of α and/or β . Thereafter, the actual simulation of dr/dt and the evaluation of the objective function (Equation 4.2) yield a reduction in CPU time as this type of model has less dimensions than the original model describing Equation (4.1).

Pattern Identification (EOFs)

Patterns (Empirical Orthogonal Functions) are the eigenvectors of a covariance matrix that is computed from an ensemble of snapshot vectors [Park & Cho, 1996; Park

et al., 1999; Hoffman Jørgensen & Sørensen, 2000; Vermeulen *et al.*, 2004a,b]. Snapshots are n_j^e specific result vectors ϕ_i^j that are obtained by the original model given a set of estimate variables $\mathcal{L}_j$. They are collected as vectors in a matrix $\mathbf{D}_j = [\phi_1^j, \phi_2^j, \dots, \phi_{n_j^e}^j]$. This matrix contains the behavior of the original model with source terms (e.g. wells, recharge and rivers) for a certain set of estimate variables $\mathcal{L}_j$. The original model is computed again for different sets of estimate variables and the total collection of snapshot vectors ($n^e = n_1^e + n_2^e + \dots + n_j^e$) becomes the matrix $\mathbf{E} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_j]$ with dimension $[n^m \times n^e]$. Following the procedure as described in Section 3.1.3 on page 57, it yields a pattern $\mathbf{p}_i$ as the normalized eigenvector of $\mathbf{E}\mathbf{E}^T$ and its corresponding relative importance φ_i (Equation 3.9). A collection of pattern vectors $\mathbf{p}_i$ are then stored as vectors within an orthonormal *projection* matrix $\mathbf{P}$ with dimension $[n^m \times n^p]$.

An important difference with the patterns elaborated earlier in Chapter 3 is that the patterns here represent an ‘extended’ subspace $\mathcal{S}$. They capture the influence of the estimation variables on the hydraulic head too, and increase the application of the reduced model for inverse modeling. Nonetheless, the dimension of the reduced model will increase but remains still significantly lower than the dimension of the original model.

Reduced Model by means of a Galerkin Projection

The Galerkin Projection method finds an expression for $\mathcal{F}$ (Equation 4.4) by substituting the hydraulic head ϕ (Equation 4.1) by $\hat{\phi}$ (Equation 4.3), yielding:

$$\sum_{i=1}^{n^p} \left\{ \frac{\partial}{\partial x} \left[C^x(\alpha) \frac{\partial p_i}{\partial x} \right] + \frac{\partial}{\partial y} \left[C^y(\alpha) \frac{\partial p_i}{\partial y} \right] + \frac{\partial}{\partial z} \left[C^z(\alpha) \frac{\partial p_i}{\partial z} \right] \right\} r_i - \gamma q = -\beta S \sum_{i=1}^{n^p} p_i \frac{\partial r_i}{\partial t}. \quad (4.5)$$

where C^x [T^{-1}] and C^z [T^{-1}] become

$$C^x(\alpha)_i = (\Delta x_i + \Delta x_{i+1}) / \left[\frac{\Delta x_i}{(\alpha K^{xx} h)_i} + \frac{\Delta x_{i+1}}{(\alpha K^{xx} h)_{i+1}} \right] \quad (4.6a)$$

$$C^z(\alpha)_i = \alpha R_i^{-1} \quad (4.6b)$$

for a discretized mesh of grid cells i . Here h is the thickness of the aquifer, R [T] is the vertical resistance between two aquifers. All of them can be affected by the estimate variable α . The computation for C^y is identical to the computation of C^x . It can be written in matrix notation, in which the second-order differential of the pattern derivative of space can be computed in advance (more extensively elaborated in Section 3.2.2 on page 65), so

$$\mathbf{U}(\alpha)\mathbf{r} - \gamma\mathbf{q} = -\beta\mathbf{S}\mathbf{P}\frac{d\mathbf{r}}{dt}, \quad (4.7)$$

where the matrix $\mathbf{U} [n^m \times n^p]$ is a function of the vector α , the matrix $\mathbf{S} [n^m \times n^m]$ and vector $\mathbf{q} [n^m \times 1]$ are influenced by the vectors β, γ , respectively. Equation (4.7) still operates within $\mathbb{R}^{n^m}$ and all superfluous equations can be eliminated by multiplying (*projecting*) each term in by the projection matrix $\mathbf{P}^T$:

$$\underbrace{\mathbf{P}^T \mathbf{U}(\alpha)}_{\mathbf{A}(\alpha)} \mathbf{r} - \underbrace{\mathbf{P}^T \gamma \mathbf{q}}_{\mathbf{q}(\gamma)} = - \underbrace{\mathbf{P}^T \beta \mathbf{S} \mathbf{P}}_{\mathbf{S}(\beta)} \frac{d\mathbf{r}}{dt}. \quad (4.8)$$

This low-dimensional ODE operates within $\mathbb{R}^{n^p}$ and is solved for each time step t_i with an implicit Euler scheme for the time derivative of $\mathbf{r}$:

$$\left[\mathbf{A}(\alpha) - \frac{1}{\Delta t} \mathbf{S}(\beta) \right] \mathbf{r}(t_i) = - \frac{1}{\Delta t} \mathbf{S}(\beta) \mathbf{r}(t_{i-1}) - \mathbf{q}(t_i)(\gamma). \quad (4.9)$$

4.1.3 Inverse Modeling

The computed coefficients r_i (Equation 4.9) are used to reconstruct $\hat{\phi}$ for the locations of observations. This results in an approximate objective function $\hat{J}$:

$$\hat{J} = \frac{1}{n^t n^o} \sum_{i=1}^{n^t} \sum_{j=1}^{n^o} \left\{ \left[\phi_j^o(x, y, z, t_i) - \hat{\phi}_j(x, y, z, t_i, \alpha, \beta, \gamma) \right] w_j \right\}^2. \quad (4.10)$$

A popular method to minimize the objective function $\hat{J}$ is to compute the gradient $\nabla \hat{J}_j$ for each estimate variable j in which $\hat{J}$ declines at a certain location in parameter space (i.e. the current values of α, β, γ). The gradient can be obtained by perturbing the n^u estimate variables independently (by means of a finite-difference approximation) and calculating the gradient of the objective function. This requires $n^u + 1$ normal simulations with the model (i.e. a forward run). However, the gradient can be most efficiently obtained by the *adjoint* method [Courant & Hilbert, 1953]. This requires one forward run with Equation (4.9), and one reverse simulation (i.e. one adjoint run) that leads to a reduced adjoint state variable $\underline{\lambda}$ [-] (elaborated in Appendix 4B). The $\nabla \hat{J}$ is obtained by:

$$\begin{aligned} \nabla \hat{J} = \frac{\Delta \hat{J}}{\Delta \alpha} &= \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T \left[\frac{\partial \mathbf{A}}{\partial \alpha} \mathbf{r}(t_i) \right]; \\ \frac{\Delta \hat{J}}{\Delta \beta} &= \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T \left[- \frac{\partial \mathbf{S}}{\partial \beta} \left(\frac{\mathbf{r}(t_i) - \mathbf{r}(t_{i-1})}{\Delta t} \right) \right]; \\ \frac{\Delta \hat{J}}{\Delta \gamma} &= \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T \left[\frac{\partial \mathbf{q}(t_i)}{\partial \gamma} \right], \end{aligned} \quad (4.11)$$

wherein

$$\frac{\partial \underline{\mathbf{A}}}{\partial \underline{\alpha}} = \mathbf{P}^T \left[\frac{\partial}{\partial x} \left(\frac{\partial \mathbf{C}^x}{\partial \underline{\alpha}} \frac{\partial \mathbf{P}}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial \mathbf{C}^y}{\partial \underline{\alpha}} \frac{\partial \mathbf{P}}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial \mathbf{C}^z}{\partial \underline{\alpha}} \frac{\partial \mathbf{P}}{\partial z} \right) \right], \quad (4.12a)$$

$$\frac{\partial \underline{\mathbf{S}}}{\partial \underline{\beta}} = \mathbf{P}^T \frac{\partial \mathbf{S}}{\partial \underline{\beta}} \mathbf{P}, \quad (4.12b)$$

$$\frac{\partial \mathbf{q}(t_i)}{\partial \underline{\gamma}} = \mathbf{P}^T \frac{\partial \mathbf{q}(t_i)}{\partial \underline{\gamma}}. \quad (4.12c)$$

Notice that the exchange of the variables α, β, γ appears in the dimensions of the original model, before the projection with $\mathbf{P}^T$. Equation (4.12a) is elaborated in Appendix 4A). This is disadvantageous for the final efficiency (see Subsection 4.3.2), but currently unavoidable.

Once $\nabla \hat{J}$ is known, there are various methods to search along that gradient to reach a minimum (i.e. a line search). In this chapter we implemented Variable Metric Method (Quasi-Newton) [Press *et al.*, 1992]. The method uses information from the gradient to obtain an estimate of the second-order derivative of the objective function (often addressed as the Hessian). This Hessian adjusts the gradient, and from the acquired minimum along that renewed direction, the inverse modeling sequence reiterates. The entire process can be enumerated as follows (Figure 4.1):

(1) The original model is evaluated for specific situations (*snapshots*) that represent the model behavior and the influence of the estimate variables. (2) From the snapshots a set of orthogonal patterns (EOFs) is computed. (3) A number of patterns is selected to fulfill the expected model variance (φ^e) and with that set of patterns a reduced model is created. (4) The time-dependent coefficients obtained by one single forward simulation are then used for the adjoint run (backward run), to compute the reduced adjoint state variable. This variable provides the gradient of the objective function. (5) The reduced model is simulated several times to search along the obtained gradient towards a minimum of the objective function (i.e. a line search). (6) Finally, it depends on the progress of the inverse modeling whether one needs to proceed to step 3 (gradient loop, η), 2 (pattern loop, κ) and/or 1 (snapshot simulation loop, μ).

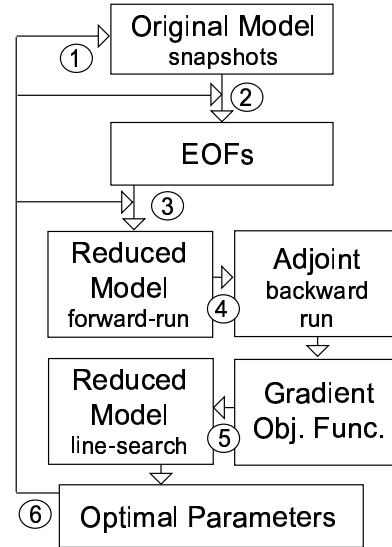


Figure 4.1: Methodology for inverse modeling by use of a reduced model. Numbers refer to steps described in the text.

4.2 Application to a Synthetical Problem

4.2.1 Problem Description

A one-dimensional synthetic model was considered that estimated the variable α (affecting the transmissivity) which varies within two predefined zones (Figure 4.2). The zones were defined by the grid cells $\mathbf{Z}_1 \in \{1-50\}$ and $\mathbf{Z}_2 \in \{52-101\}$. The well rate varied randomly ($-50 \leq q \leq 50 \text{ m}^3\text{day}^{-1}$) for the period $10 \leq t \leq 1,000$ days, and was kept constantly ($q = -15 \text{ m}^3\text{day}^{-1}$) for the succeeding period $1,000 < t \leq 2,000$ days. The model was simulated with $\alpha \equiv \beta \equiv \gamma = 1$ and $\Delta t = 10$ days for each time step, yielding $n^t = 200$ time steps in total. A set of synthetic observations were obtained by recording ϕ at all time steps for the grid cells 25 and 76.

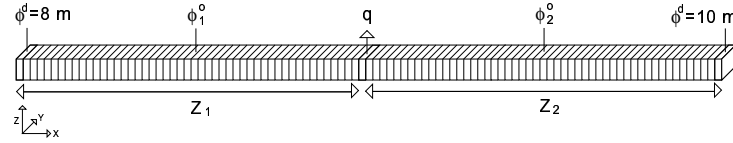


Figure 4.2: Synthetic problem under consideration: 101 grid cells along a single row with an extraction well positioned in the middle and a Dirichlet condition on both edges. Each grid cell is dimensioned by: $\Delta x = \Delta y = 10 \text{ m}$ with $T = 100 \text{ m}^2\text{day}^{-1}$ and $S = 0.21$. The model is divided into two zones $\mathbf{Z}_1$ and $\mathbf{Z}_2$ and two observation wells ϕ_1^o, ϕ_2^o , one observation within each zone.

4.2.2 Snapshot Simulation

In this subsection the snapshots are determined for the estimate variable α . The procedure will be identical for the variables β, γ . First, the sensitivity of α can be expressed by simulating the original model with different values for α [Park & Cho, 1996]. A simple approach is to define a lower and upper boundary for each estimate variable (i.e. a *snapshot boundary value*). These are not the true boundaries of the variable, but they determine a range for which the snapshots are currently representative. For example, these snapshot boundary values can be defined as:

$$\alpha_i^e = \exp [\ln (\alpha_i) \pm \delta_i] \quad (4.13)$$

where δ_i is a step size that determines the width of the snapshot boundary for the estimate variable α_i . The log transformation is applied for reasons of convenience, as the estimate variable α_i is also log-transformed for the minimization process.

For the synthetical problem we defined the lower and upper snapshot boundary values as $\alpha \equiv \delta = 1$. Several snapshot simulations were computed by sequentially perturbing one variable and keeping the others at their lower boundary. This yielded a $n^u + 1$ set of estimate variables ($\mathcal{L}$) defined by: $\mathcal{L}_1 \in \{\alpha_1^e = 0.36; \alpha_2^e = 0.36\}$, $\mathcal{L}_2 \in \{\alpha_1^e = 0.36; \alpha_2^e = 2.72\}$, $\mathcal{L}_3 \in \{\alpha_1^e = 2.72; \alpha_2^e = 0.36\}$. Mathematically, this method has a drawback: the pattern identification technique (Subsection 4.1.2) is sensitive to any correlation in the snapshots [Cazemier *et al.*, 1998]. This

occurs when a specific variable affects a limited zone and generates almost identical values for ϕ for the remaining zones. The *Latin Hypercube Sampling* (LHS) method reduces this correlation [Iman & Shortencarier, 1984]. It divides the range of each estimate variable into i non-overlapping intervals on the basis of equal probability. One value is randomly selected from each interval with respect to the probability density in the interval. For the synthetic problem, the LHS method generated $n^u = 2$ sets of estimate variables valued by $\mathcal{L}_1 \in \{\alpha_1^e = 0.41, \alpha_2^e = 1.43\}$ and $\mathcal{L}_2 \in \{\alpha_1^e = 0.94, \alpha_2^e = 0.52\}$. These were used to compute two snapshot simulations.

A single snapshot simulation i captured the influence of q (i.e. the boundary condition and the computation of draw-down by a single well) with respect to the variables in $\mathcal{L}_i$. A snapshot for the boundary condition was computed as the steady-state solution, whereby $q = 0 \text{ m}^3\text{day}^{-1}$. This snapshot $\phi_1(\mathcal{L}_i)$ represents the gradient through the system caused by the boundary condition on both ends of the model (Figure 4.3). Thereafter, the extraction rate for the well increased instantaneously up to $q = 50 \text{ m}^3\text{day}^{-1}$ and 4 snapshots were recorded at intermediate time intervals t ($t_1 = 10, t_2 = 40, t_3 = 350, t_4 = 5,000$ days) until a new steady-state situation was reached. The chosen time intervals were not by definition the best as different combinations improved or worsened the reduced model at the end. The most important point is that the recording frequency was high initially and decreased as time elapsed. Eventually, these snapshots were captured for each combination of estimate variables defined by $\mathcal{L}_1$ and $\mathcal{L}_2$, yielding $n^e = n^u \times 5 = 10$ snapshots as indicated in Figure 4.3a. Finally, the snapshot $\phi_1(\mathcal{L}_i)$ is subtracted from the corresponding snapshots ($\phi_j(\mathcal{L}_i); j \in \{2-5\}$), such that they reflect the influence of the well solely with respect to each combination of variables, see Figure 4.3b.

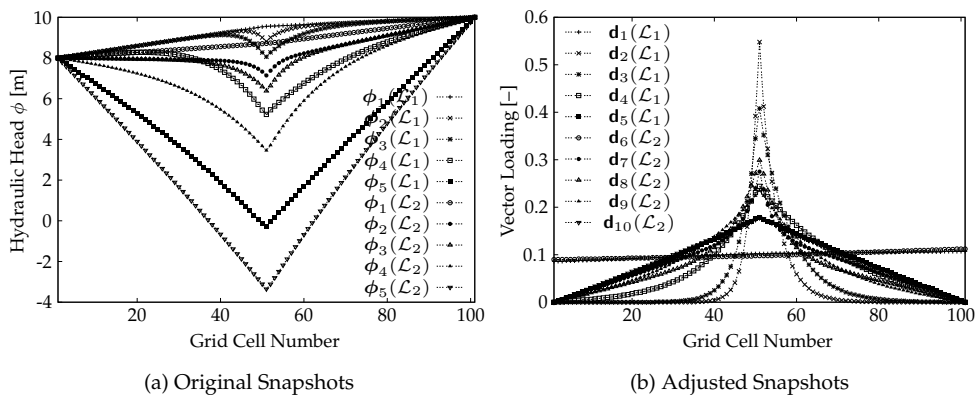


Figure 4.3: Graph of the (a) selected snapshot vectors $\phi_i(\mathcal{L}_j)$ and (b) the adjusted vectors $\mathbf{d}_i(\mathcal{L}_j)$ for two combinations of the estimate variables α computed with the original model.

4.2.3 Resulting Patterns

The collection of $n^e = 10$ snapshots yielded 9 patterns with eigenvalues $\varphi_i > 0\%$. The maximal explained variance for a pattern is 84.27% and it declines rather rapidly for the other ones. When we take five main patterns together, they explained already $> 99.9\%$ (Figure 4.4a). Within the spatial structure of $\mathbf{p}_1$ and $\mathbf{p}_2$ the solution of ϕ is still recognizable. All other patterns ($\varphi_i \leq 5\%$) describe a numerical behavior of the system that does not clearly reflect a hydrological phenomenon, see Figure 4.4b. They can be *rotated* to increase their interpretability, see Subsection 3.1.4 on page 61.

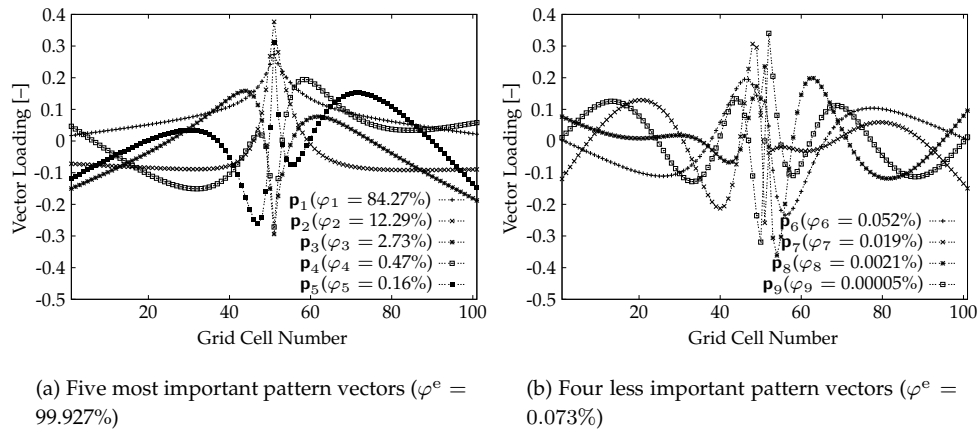


Figure 4.4: Graphs of the pattern vectors $\mathbf{p}_i$ and their corresponding relative importance φ_i .

4.2.4 Sensitivity of $\hat{J}$

Influence of the Number of Patterns

In Figure 4.5a the surface of the objective function J (Equation 4.2) is depicted. The minimum of the objective function ($\min(J) = 0.34 \cdot 10^{-4} \text{ m}^2$) is located at $\alpha = 1$. The surface of the approximate objective function $\hat{J}$ (4.10) computed with $n^p = 9$ patterns is accurate for a significant area around the minimum of the function, see Figure 4.5b. Although the minimum value is increased ($\min(\hat{J}) = 0.59 \cdot 10^{-1} \text{ m}^2$), its location is almost identical to $\min(J)$. It is remarkable that the resemblance between the functions exceeds approximately 3 times the range of the snapshot boundary values for the estimate variables (see Subsection 4.2.2). When α_1, α_2 were forced towards more extreme values ($-3 \geq \ln(\alpha_i) \geq 3$), the difference between J and $\hat{J}$ increased, but the shape of the surface remained mostly similar. This can be seen more clearly in a cross section of the objective function for $\ln(\alpha_2)$ equals zero, see Figure 4.7a-b. The efficiency of a reduced model increases with a decrease in the number of patterns n^p [Vermeulen *et al.*, 2004b]. Therefore, it is a challenge to reduce n^p such that the resulting approximate objective function is still accurate

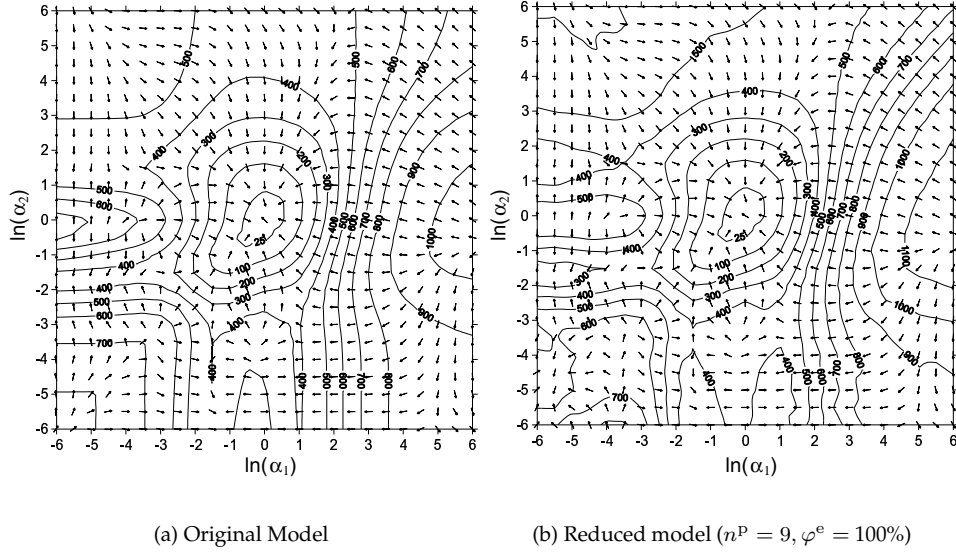


Figure 4.5: Surface of the objective function J [m²] for α_1, α_2 computed with (a) the original model and (b) the reduced model ($\mathcal{L}_1 \in \{\alpha_1^e = 0.41, \alpha_2^e = 1.43\}$ and $\mathcal{L}_2 \in \{\alpha_1^e = 0.94, \alpha_2^e = 0.52\}$). The arrows represent the gradient ∇J .

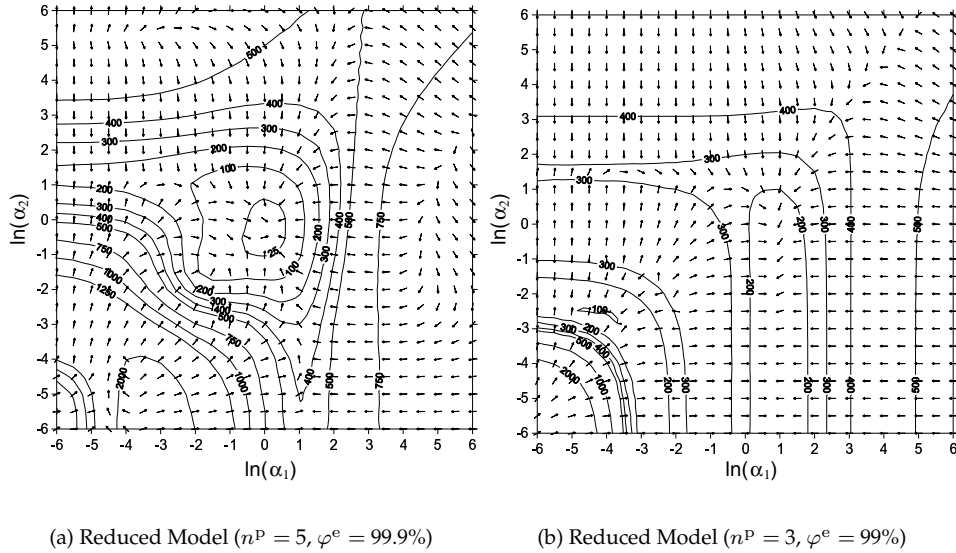


Figure 4.6: Surface of the approximate objective function $\hat{J}$ [m²] for α_1, α_2 computed with a reduced model ($\mathcal{L}_1 \in \{\alpha_1^e = 0.41, \alpha_2^e = 1.43\}$ and $\mathcal{L}_2 \in \{\alpha_1^e = 0.94, \alpha_2^e = 0.52\}$). The arrows represent the gradient $\nabla \hat{J}$.

enough. As most of the variance occurs near the well, the variance is less near a observation location. This offers the possibility of neglecting some tiny patterns without severely affecting the shape of the objective function. In Figure 4.6a the surface of $\hat{J}$ is depicted ($n^p = 5$; $\varphi^e = 99.927\%$); thus each pattern in Figure 4.4b is ignored. The surface shape still shows a strong resemblance with J , but $\min(\hat{J}) = 6.5 \text{ m}^2$ and is located at $\ln(\alpha_1) = 0.095$, $\ln(\alpha_2) = -0.083$. Nevertheless, with half the number of patterns the optimal value is closely to its global minimum. From this point, more patterns can be added to improve the estimation (pattern loop, κ). There is a limitation, however, as with the amount of patterns ($n^p = 3$; $\varphi^e = 99\%$) one decreases the accuracy and reliability of the $\hat{J}$, see Figure 4.6b.

Influence of the Snapshot Simulations

The accuracy of $\hat{J}$ can be traced back to the snapshot simulations (Subsection 4.2.2) as they form the real bases on which the reduced model is founded. The chosen snapshot boundary values (Equation 4.13) are herein most determined as one can see by expanding $\delta_i = 4$. It will increase the number of patterns and the range of resemblance, but eventually it increases the difference at the minimum, ($\min(\hat{J}) = 1.59 \text{ m}^2$, see Figure 4.7c). For this synthetic problem this does not seem to be a problem, but numerical experiences for more complex systems with many estimate variables showed that $\delta_i \geq 2$ yielded unreliable objective functions. For such a wide range of parameter values, the assumed linear relation over δ is not valid anymore.

It is unlikely that the initial value α^0 is equal to the optimal value α^t . Whenever the initial snapshot simulation ($\delta_i = 1$) is computed around $\ln(\alpha_1) = -4$ and $\ln(\alpha_2) = 0$, the shape of $\hat{J}$ is only accurate for the interval $-6 \leq \ln(\alpha_1) \leq 1$ (Figure 4.7d). The location of $\min(\hat{J})$ has been even shifted in case $\ln(\alpha_1) = 4$, $\ln(\alpha_2) = 0$ and $\delta_i = 1$ (Figure 4.7e). These outcomes limit a reduced model to a certain band-

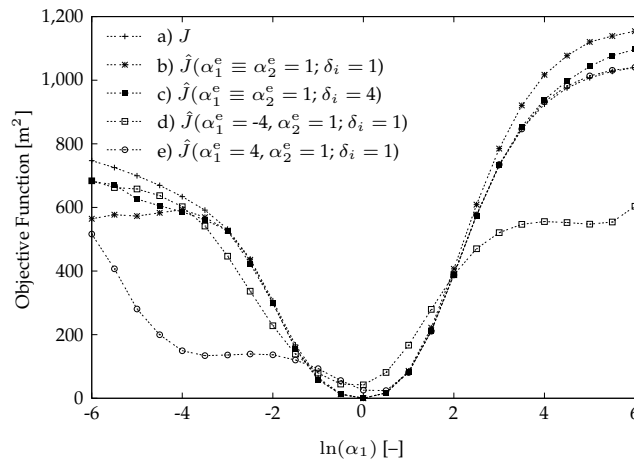


Figure 4.7: Cross section of the objective functions J and $\hat{J}$ versus α_1 ($\ln(\alpha_2) = 0$) for different values for α_1^e and α_2^e that were used to compute snapshots and collect these patterns such that $\varphi^e = 100\%$.

width to estimate α . Therefore the reduced model needs to be updated after a minimum for the current reduced model is achieved.

Minimization Experiments

This subsection extends the number of estimate variables for the one-dimensional synthetic problem (Subsection 4.2.1) to nine, see Figure 4.8. The rate for each well varied randomly and for each observation well a set of observations was obtained by recording ϕ for all time steps. For twelve test cases (simulations 1–12) the initial estimate variables (α_i^0) were varied randomly and corresponding snapshots were computed as described in Subsection 4.2.2. It yielded $n^e = 9 \times 4 \times 9 = 324$ snapshots. The minimization procedure as outlined in subsection 4.1.3 was used to estimate the optimal value for those estimate variables.

In Table 4.1 the results of the minimization are given. Both the original and the reduced model succeeded in getting the correct optimal values for the estimate variables (simulations 1–3). Their initial values (α_i^0) are within close reach (0.09–10) of their optimal values ($\|\alpha^t\| = 1$). They needed approximately $\eta = 9$ –19 gradient iterations. More gradient iterations ($\eta = 22$ –32) were necessary for the reduced model as the reach of α_i^0 increases (simulations 4–5). For specific values of α_i^0 the original model was trapped by a local minimum, as the reduced model succeeded in finding the *global* minimum (simulations 6–8). In this case, the reduced model has the benefit of the lack of detail in the objective function. Another example is given in simulation 9, in which both models fail to find the global minimum. Whenever the reduced model starts the inverse modeling for this case, by rejecting initially the less important patterns (i.e., their lumped contribution to the explained snapshot variance is less than 1%), it proceeds eventually towards the global minimum, see simulation 10. Unfortunately, this is not a rule of thumb as this strategy can work the other way around, compare simulations 11–12. In the end, it can be said that the reduced model is capable of finding an optimal solution for realistic values of α (0.01–100). Beyond that, the method may fail, just like the original model, but it may also succeed although the procedure is more ambiguous.

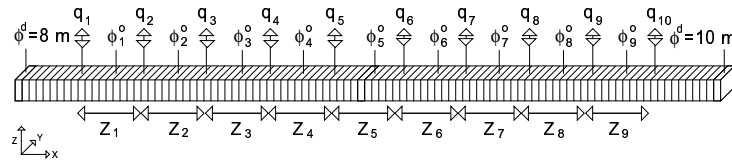


Figure 4.8: Synthetic problem under consideration: 101 grid cells along a single row with a Dirichlet condition on both edges. Each grid cell is dimensioned by: $\Delta x = \Delta y = 10$ m with $T = 100 \text{ m}^2 \text{ day}^{-1}$ and $S = 0.21$. The model is divided into nine zones (Z_1 – Z_9), each contains an observation well ϕ_i^o and an injection and/or extraction well q_i .

Table 4.1: Several values for the prior estimate α_i^0 and the corresponding objective function value J^0 [m^2], and its value after η minimization steps for the original model (J) and the reduced model ($\hat{J}$)[†].

Simulation	1	2	3	4	5	6
α_1^0	0.098	1.35	0.67	15.3	0.51	0.17
α_2^0	0.11	3.39	0.76	3.97	0.63	7.65
α_3^0	0.29	0.20	12	0.38	63	0.07
α_4^0	1.70	2.03	0.90	0.64	0.84	3.24
α_5^0	0.94	0.08	3.74	0.10	9.02	0.01
α_6^0	2.54	2.80	4.01	0.41	10.1	5.58
α_7^0	0.06	0.23	10.78	1.20	52.6	0.09
α_8^0	1.65	4.31	0.84	1.60	0.74	11.4
α_9^0	0.89	4.26	0.09	8.99	0.02	11.2
J^0	534.22	101.85	48.68	44.82	115.3	410.46
$J^0(\eta)$	0.004(10)	0.006(12)	0.008(12)	0.008(26)	0.001(22)	12.36(6) [#]
$\hat{J}(\eta/n^p), \mu = 1$	0.009(9/40)	0.01(15/35)	0.06(19/35)	10.51(6/34)	3.49(20/34)	0.30(20/38)
$\hat{J}(\eta/n^p), \mu = 2$	-	-	-	0.01(20/35)	0.04(12/32)	0.004(3/37)
$\hat{J}(\eta/n^p), \mu = 3$	-	-	-	-	-	-

Simulation	7	8	9	10	11	12
α_1^0	94.1	4.45	0.17	0.17	3.36	3.36
α_2^0	9.95	2.05	0.01	0.01	4.28	4.28
α_3^0	0.20	102.9	0.11	0.11	1066	1066
α_4^0	0.48	1632	0.03	0.03	6	6
α_5^0	0.02	10.96	5.80	5.80	103.4	103.4
α_6^0	0.23	0.88	6.72	6.72	118.7	118.7
α_7^0	1.37	373.5	0.01	0.01	859.2	859.2
α_8^0	2.18	281.7	14.37	14.37	5.17	5.17
α_9^0	38.9	7.38	70.2	70.2	0.06	0.06
J^0	101.89	66.77	831.84	831.84	62.73	62.73
$J^0(\eta)$	12.40(9) [#]	15.29(22) [#]	15.31(9) [#]	15.31(9) [#]	13.78(12) [#]	13.78(12) [#]
$\hat{J}(\eta/n^p), \mu = 1$	0.69(20/34)	10.05(20/19)	11.86(11/41)	12.03(14/18) ⁺	45.93(14/10) ⁺	5.45(20/24)
$\hat{J}(\eta/n^p), \mu = 2$	0.004(6/36)	6.67(10/33)	7.94(14/29)	0.79(20/17) [*]	9.94(15/15) [*]	2.82(7/38)
$\hat{J}(\eta/n^p), \mu = 3$	-	0.002(20/34)	7.70(4/33) [#]	0.009(4/39)	3.12(20/35) [#]	0.007(20/37)

[†] The reduced model is based upon a number of patterns (n^p) that describe $\varphi^e = 100\%$ of the snapshot variance, unless stated otherwise; [#] Local minimum; ⁺ Here $\varphi^e = 99.9\%$; ^{*} Here $\varphi^e = 99.99\%$.

4.3 Application to a Real-World Case

This section examines to what extent the reduced model can reduce the computation time needed for inverse modeling. For this purpose we consider a 3D problem with a realistic parameterization based on a situation near the town Staphorst in the Netherlands.

4.3.1 Description of the Original Model

The model area is approximately 30 km² translated into $n^m = 34,224$ grid cells (92 rows and 93 columns) divided over 4 model layers. The model layers 1–3 have a closed boundary condition. Model layer 4 has an open boundary condition

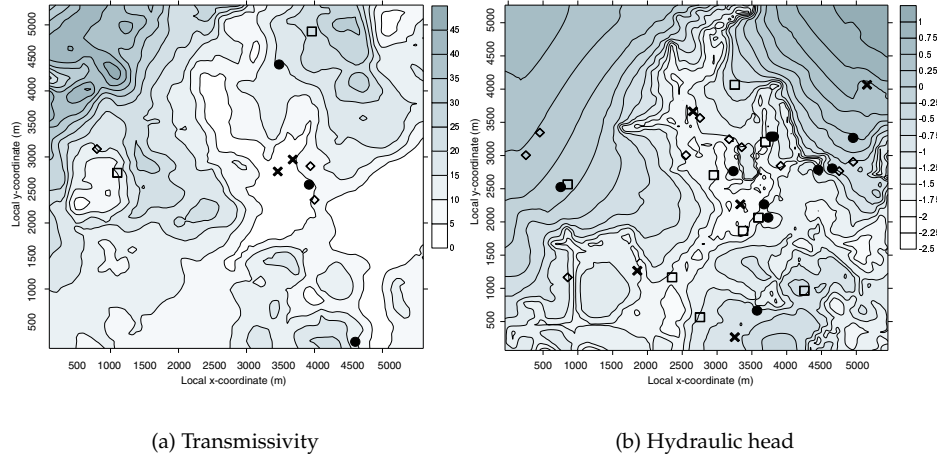


Figure 4.9: Distribution of (a) transmissivity T [$\text{m}^2\text{day}^{-1}$] and (b) the computed hydraulic head ϕ [m] for time step 100. The locations of observations within model layer 1–4 are expressed by the symbols $\bullet$, $\diamond$, $\square$, $\times$, respectively.

(Dirichlet Condition). The underground is characterized by a detailed distribution of transmissivities that vary between 0–50 (Figure 4.9a), 0–33, 330–540 and 625–1,370 $\text{m}^2\text{day}^{-1}$ for the model layers 1–4, respectively. The resistance for the aquitards in between varies between 15–175, 280–520 and 310–1,000 days. The top of the hydrological system is characterized by an area that has an intense surface water system (modeled by a linear relationship between ϕ and the water level) and a surrounding area with a free-floating water table. The distribution of precipitation was computed by detailed land survey images. There are ten independent extraction wells, distributed throughout the district (Figure 4.9a). The model was simulated over a period of $n^t = 200$ time steps ($\Delta t = 10$ days), and a transient solution is given for model layer 1 for time step 100 in Figure 4.9a. The estimate variable α was optimized for three zones in model layer 4, for two zones in model layer 3, and for one zone in model layer 2. This yielded six estimate variables ($n^u = 6$). Furthermore, $n^o = 35$ observation wells were selected that were measured for each time step, see Figure 4.9b.

4.3.2 Results

Snapshot Simulation

The boundary conditions within the model were Dirichlet conditions (model boundary), Cauchy conditions (surface water), and Neumann conditions (precipitation and the extraction wells). To isolate the effects of these boundary conditions, each snapshot simulation contained a steady-state solution without those boundary conditions. Subsequently a transient impulse response with nine time steps was computed for the precipitation ($\Delta t_{k+1} = 2\Delta t_k$ with $\Delta t_1 = 10$ days), and ten transient

impulse responses were computed for each well, independently ($\Delta t_1 = 10$, $\Delta t_2 = 20$, $\Delta t_3 = 50$, $\Delta t_4 = 500$ days). This yielded $1+9+(4 \times 10)=50$ snapshots for each snapshot simulation. Each of them was carried out with another set of estimate variables $\mathcal{L}_i \in \{\alpha_1^e, \dots, \alpha_6^e\}$ that was generated by the LHS method (Subsection 4.2.2). Eventually the total number of snapshots was $n^e = 6 \times 50 = 300$, which is 1.5 times the number of total time steps for a single forward model simulation ($n^t = 200$).

Table 4.2: Number of snapshot simulation loops (μ) and gradient loops (η) for inverse modeling of six unknown estimate variables α for the original model (a) and the reduced model (b).

(a) Original Model							
η	α_1	α_2	α_3	α_4	α_5	α_6	J [m ²]
0	1	1	1	1	1	1	27.47
14	0.13	7.40	0.05	19.80	0.22	0.22	0.00051

(b) Reduced Model							
μ/η	α_1	α_2	α_3	α_4	α_5	α_6	$\hat{J}$ [m ²]
0/0	1	1	1	1	1	1	27.34
1/5	0.06	8.46	0.005	3.31	0.08	0.14	8.33
2/10	0.14	6.65	0.05	19.52	0.21	0.21	0.59
0.14	7.39	0.05	20.1	0.23	0.21	0.024	

Accuracy of the Estimated Variables

First, the estimate variables were obtained by use of the original model. Within $\eta = 14$ gradient loops, it resulted in a wide range of estimate variables values (Table 4.2a) with $\min(J) = 0.00051$ m², see Figure 4.10a. Second, the estimate variables were obtained by a reduced model, see Figure 4.10b and Table 4.2b. For the first snapshot loop ($\mu = 1$, $\delta_i = 1$) a reduced model was constructed with $\varphi^e = 99.9\%$, $n^p = 36$ that was able to reduce the objective function and estimate the variables towards the proper direction. After $\eta = 5$ gradient loops the objective function became stable and another snapshot simulation was performed ($\mu = 2$, $\delta_i = 1$)

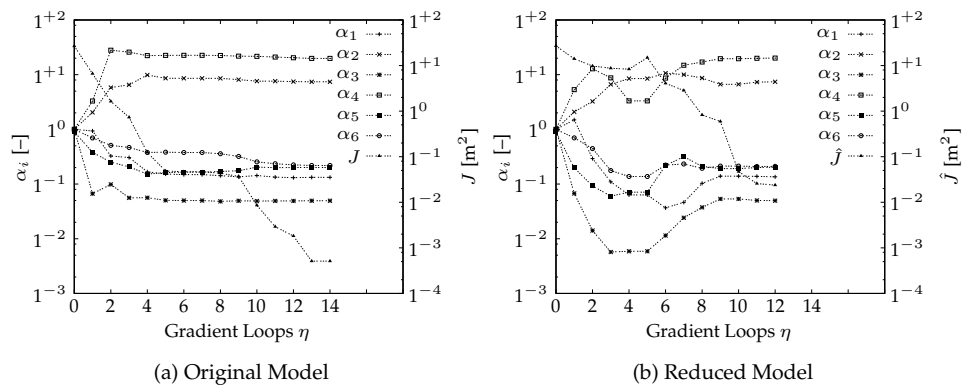


Figure 4.10: Graph of the progress of α_1 – α_6 and $\hat{J}$ versus the number of gradient loops η to compute the gradient of the objective function by use of (a) the original model, and (b) the reduced model.

around the current values of the estimate variables. Notice that the objective function increased initially ($\eta = 6$), as the current location in parameter space was not a minimum anymore for the updated reduced model ($\varphi^e = 99.99\%$, $n^p = 63$). Nevertheless, the objective function decreased significantly after $\eta = 10$ gradient loops, and the estimate variables were close to their optimal values. A final snapshot simulation was carried out ($\mu = 3$, $\delta_i = 0.5$), and the final reduced model ($\varphi^e = 100\%$, $n^p = 103$) yielded almost identical values for the estimated variables as found with the original model. As the estimate variables did not change significantly within this third snapshot simulation loop, the inverse modeling was terminated.

Time Efficiency

All computations for this chapter have been carried out on a Pentium 4 ® (2.4GHz) processor and a single forward or backward run with the original model consumed approximately $t^e = 100$ s ($n^t = 200$), see Figure 4.11. It took approximately $t^\nabla = 2 \times t^e = 200$ s to compute ∇J , and the entire parameter optimization with the original model took approximately $14 \times 200 = 2,800$ s.

The time it took the reduced model to compute $\nabla \hat{J}$ depended on the number of patterns involved. It can be calculated as the sum of the time taken to construct a reduced model (preparation time t^p), twice the simulation time of the reduced model (i.e. the forward and adjoint run; $2 \times t^e$) and the time t^∂ to compute $\partial \mathbf{A} / \partial \alpha$ (Equation 4.12a). The total time is dominated by t^∂ which is almost quadratically related to n^p as it involves two nested ‘do-loops’ over n^p , see Figure 4.11. As the inverse modeling took $n^p = 36$ patterns for the first snapshots simulation loop ($\mu = 1$), 63 and 103 for the following snapshot simulation loops ($\mu = 2, 3$), the final computation time can be computed as 5×60 s 5×100 s 2×195 s = 1,190 s, see Table 4.2b and Figure 4.12. Hence, the total inverse modeling, described in Subsection 4.3.2, resulted in a final time reduction of $1 - (1,190 / 2,800) \times 100\% \approx 60\%$.

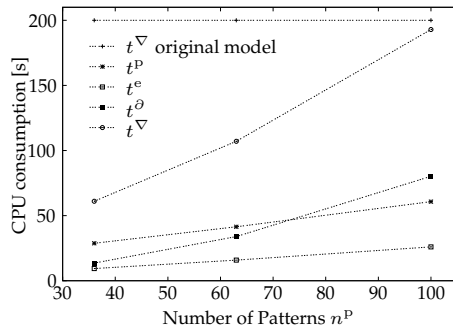


Figure 4.11: Graph of the CPU consumption for different processes for inverse modeling versus the number of patterns (n^p).

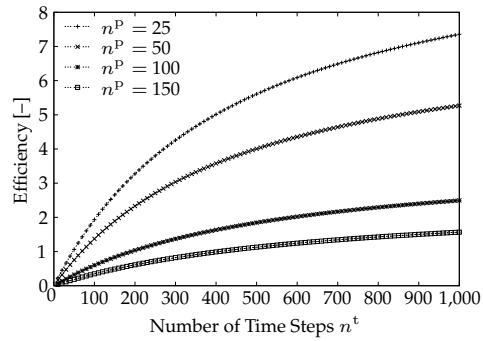


Figure 4.12: Graph of the ratio of computation times by use of the original and reduced model, to compute the gradient of the objective function, versus the number of time steps within a simulation.

4.4 Conclusions

This chapter describes a new reduced model for inverse modeling of groundwater flow. The proposed model consists of a linear combination of a set of patterns and time-varying coefficients. The patterns (Empirical Orthogonal Functions EOFs) depend both on the response of the hydraulic head ϕ with respect to source terms (e.g. wells and rivers) and system properties (e.g. horizontal/vertical conductances and storage coefficients). These fields of system properties could be defined by zones as done in this chapter and/or by stochastic simulation. By selecting the dominant patterns, we have rewritten the original PDE into another one that operates within the numerical space that is spanned by the selected patterns. Simulating this type of model yields a reduction in CPU time as the number of state variables is small. Since the patterns are based upon different parameter values, the resulting reduced model accurately simulates a wide range for those parameter values. Another advantage is that the adjoint of the reduced model can be obtained fairly easy.

It is not possible to quantify a general rate of efficiency for the proposed procedure, as it depends strongly on the application. For example, the efficiency increases whenever the estimate variable appears only to be the storage coefficient and/or the recharge/well rate. The latter could really benefit the minimization program MODMAN [Greenwald, 1998], which answers groundwater management questions that maximize or minimize a user-defined objective function. The efficiency increases even more when the number of time steps in the original model is increased, see Figure 4.12. This makes this type of inverse modeling highly suitable for long transient simulations. On the other hand, the method is strongly sensitive with respect to the number of patterns (i.e. the complexity of the problem). Such a model may possess many non-linearities between the hydraulic head and the external source terms, that will increase the computation time of the adjoint dramatically. Comparable to other inverse modeling techniques this could be disadvantageous, even though such non-linearities may not affect the objective function at all.

In future research we intend to focus on the possibility to update the estimate variables in the reduced model. This will truly increase the final efficiency as it will eliminate the necessity to construct a reduced model each time an estimate variable changes. This is the main topic of the next chapter.

4A First Derivative of the Hydraulic Conductance

In Equation (4.12a) the first derivative of the hydraulic conductance C^x with respect to the estimate variable α is used defined as

$$\frac{\partial C_i^x}{\partial \alpha_i} = \frac{\partial C_i^x}{\partial T_i} \cdot \frac{\partial T_i}{\partial \alpha_i} + \frac{\partial C_i^x}{\partial T_{i+1}} \cdot \frac{\partial T_{i+1}}{\partial \alpha_{i+1}}. \quad (4.14)$$

This can be computed with finite differences as

$$\frac{\partial C_i^x}{\partial \alpha_i} = \frac{\Delta x_i (\Delta x_i + \Delta x_{i+1})}{T_i^2 \left(\frac{\Delta x_i}{T_i} + \frac{\Delta x_{i+1}}{T_{i+1}} \right)^2} \cdot \frac{T_i}{\exp(\alpha_i)} + \frac{\Delta x_{i+1} (\Delta x_i + \Delta x_{i+1})}{T_{i+1}^2 \left(\frac{\Delta x_i}{T_i} + \frac{\Delta x_{i+1}}{T_{i+1}} \right)^2} \cdot \frac{T_{i+1}}{\exp(\alpha_{i+1})}. \quad (4.15)$$

For the sake of simplicity $C_i^x(\alpha)$ is denoted as C_i^x , and $\partial T / \partial \alpha$ is equal to $T / \exp(\alpha)$ as the estimate variable α is log transformed. To include this in the computation of the second-order differential for the pattern derivative of space we compute the elements within $\underline{\mathbf{A}}$ as

$$\frac{\partial \underline{A}_{l,k}}{\partial \alpha} = \sum_{m=l}^{n^p} \sum_{j=k}^{n^p} \sum_{i=1}^{n^m} p_{m,i} \Delta x_i^{-1} \cdot \left(\frac{\partial C_i^x}{\partial \alpha_i} \frac{p_{j,i} - p_{j,i+1}}{\frac{1}{2} \Delta x_i + \Delta x_{i+1}} - \frac{\partial C_{i-1}^x}{\partial \alpha_{i-1}} \frac{p_{j,i} - p_{j,i-1}}{\frac{1}{2} \Delta x_i + \Delta x_{i-1}} \right). \quad (4.16)$$

4B Derivation of the Reduced Adjoint State Variable

In Equation (4.11) the reduced adjoint state variable $\underline{\lambda}$ is introduced to compute the gradient of the objective function ($\nabla \hat{J}$) with respect to the estimate vectors α, β, γ . To derive $\underline{\lambda}$, we write Equation (4.8) as:

$$\underline{\mathbf{A}}\mathbf{r} - \underline{\mathbf{S}} \frac{d\mathbf{r}}{dt} + \underline{\mathbf{q}} = 0. \quad (4.17)$$

Then the objective function $\hat{J}$ is defined as:

$$\hat{J} = \hat{J}^\bullet + \underline{\lambda}^T \left[\underline{\mathbf{A}}\mathbf{r} - \underline{\mathbf{S}} \frac{d\mathbf{r}}{dt} + \underline{\mathbf{q}} \right] \quad (4.18)$$

This inclusion of $\hat{J}^\bullet \equiv \hat{J}$ is necessary to include the state variable r of the reduced model in the following expansion. To find an expression for $\Delta \hat{J}$, a first order Tay-

lor's series expansion is applied to the variables $\underline{\lambda}$, $\mathbf{r}$ and α in (4.18), yielding:

$$\begin{aligned} \frac{\partial \hat{J}}{\partial \hat{J}} \Delta \hat{J} &= \underline{\lambda}^T \left[\underbrace{\mathbf{A}\mathbf{r} - \underline{\mathbf{S}} \frac{\partial \mathbf{r}}{\partial t} + \underline{\mathbf{q}}}_{=0} \right] + \frac{\partial \underline{\lambda}^T}{\partial \underline{\lambda}^T} \Delta \underline{\lambda}^T \left[\underbrace{\mathbf{A}\mathbf{r} - \underline{\mathbf{S}} \frac{\partial \mathbf{r}}{\partial t} + \underline{\mathbf{q}}}_{=0} \right] \\ &+ \underline{\lambda}^T \left[\underbrace{\mathbf{A}\mathbf{r} - \underline{\mathbf{S}} \frac{\partial \mathbf{r}}{\partial t} + \underline{\mathbf{q}}}_{=0} \right] + \frac{\partial \hat{J}}{\partial \mathbf{r}} \Delta \mathbf{r} + \underline{\lambda}^T \left[\underbrace{\mathbf{A}\mathbf{r} - \underline{\mathbf{S}} \frac{\partial \mathbf{r}}{\partial t} + \underline{\mathbf{q}}}_{=0} \right] \\ &+ \underline{\lambda}^T \left[\underline{\mathbf{A}} \frac{\partial \mathbf{r}}{\partial \mathbf{r}} \Delta \mathbf{r} - \underline{\mathbf{S}} \frac{\partial \mathbf{r}}{\partial \mathbf{r}} \Delta \mathbf{r} + \underline{\mathbf{q}} \right] + \underline{\lambda}^T \left[\frac{\partial \underline{\mathbf{A}}}{\partial \alpha} \Delta \alpha \mathbf{r} - \frac{\partial \underline{\mathbf{S}}}{\partial \beta} \Delta \beta \frac{\partial \mathbf{r}}{\partial t} + \frac{\partial \underline{\mathbf{q}}}{\partial \gamma} \Delta \gamma \right] \end{aligned} \quad (4.19)$$

which simplifies to

$$\Delta \hat{J} = \frac{\partial \hat{J}}{\partial \mathbf{r}} \Delta \mathbf{r} + \underline{\lambda}^T \left[\underline{\mathbf{A}} \Delta \mathbf{r} - \frac{1}{\Delta t} \underline{\mathbf{S}} \Delta \mathbf{r} \right] + \underline{\lambda}^T \left[\frac{\partial \underline{\mathbf{A}}}{\partial \alpha} \Delta \alpha \mathbf{r} - \frac{\partial \underline{\mathbf{S}}}{\partial \beta} \Delta \beta \frac{\partial \mathbf{r}}{\partial t} + \frac{\partial \underline{\mathbf{q}}}{\partial \gamma} \Delta \gamma \right]. \quad (4.20)$$

The reduced adjoint state variable $\underline{\lambda}^T$ is solved by letting:

$$\frac{\partial \hat{J}}{\partial \mathbf{r}} \Delta \mathbf{r} + \underline{\lambda}^T \underline{\mathbf{A}} \Delta \mathbf{r} - \underline{\lambda}^T \frac{1}{\Delta t} \underline{\mathbf{S}} \Delta \mathbf{r} = 0, \quad (4.21)$$

which yields a simple expression as $\Delta \mathbf{r}$ can be removed, and $\underline{\lambda}^T$ is solved backward in time by an implicit Euler scheme:

$$\left(\underline{\mathbf{A}} - \frac{1}{\Delta t} \underline{\mathbf{S}} \right)^T \underline{\lambda}(t_i) = -\frac{1}{\Delta t} \underline{\mathbf{S}} \underline{\lambda}(t_{i+1}) - \frac{\partial \hat{J}}{\partial \mathbf{r}}(t_i) \quad (4.22)$$

where

$$\frac{\partial \hat{J}}{\partial \mathbf{r}}(t_i) = \frac{\partial \hat{J}}{\partial \hat{\phi}}(t_i) \cdot \frac{\partial \hat{\phi}}{\partial \mathbf{r}}(t_i) \quad (4.23a)$$

$$\frac{\partial \hat{J}}{\partial \hat{\phi}}(t_i) = 2 \sum_{i=1}^{n^o} w_i \left[\phi_i^o(t_i) - \hat{\phi}_i(t_i) \right] \quad (4.23b)$$

$$\frac{\partial \hat{\phi}}{\partial \mathbf{r}}(t_i) = - \sum_{j=1}^{n^p} p_j \quad (4.23c)$$

yields a n^p -dimensional vector. An important difference between Equation (4.9) and Equation (4.22) is the transpose sign T on the left-hand side of the equation.

5

Model-Reduced Variational Data Assimilation

Abstract. This chapter describes a new approach to variational data assimilation. It does not require an implementation of the adjoint of the tangent linear approximation of the original model, though it has a comparable computational efficiency. In classical variational data assimilation, the adjoint implementation is used to efficiently compute the gradient of the criterion to be minimized. Our approach is based on model reduction. Using an ensemble of forward model simulations, the leading EOFs are determined to define a subspace. The reduced model is created by projecting the original model onto this subspace. Once this reduced model is available, its adjoint can be implemented very easily and be used to approximate the gradient of the criterion. The minimization process can now be solved completely in reduced space with negligible computational costs. If necessary, the procedure can be repeated a few times by generating new ensembles closer to the most recent estimate of the parameters. The reduced model based method has been tested on several nonlinear synthetic cases for which a diffusion coefficient was estimated.

*This chapter is adapted from Vermeulen, P.T.M & A.W. Heemink, Model-Reduced Variational Data Assimilation, **Monthly Weather Review**, accepted, December 2005*

DATA ASSIMILATION is a methodology to combine the results of a large-scale numerical model with the measurements available to obtain an optimal reconstruction of the state of the system. It can be applied both for state estimation and for model calibration. In this chapter we concentrate our attention on calibration problems: the estimation of a large number of constant model parameters from data.

Variational data assimilation or ‘the adjoint method’ has been used very often for model calibration, e.g. *Carrera & Neuman* [1986]; *Lardner & Song* [1995]; *Ullman & Wilson* [1998]; *Heemink et al.* [2002]. The method is based on optimal control theory. Here, a number of unknown ‘control’ parameters are introduced into the numerical model. Using the available data, these control parameters are identified by minimizing a certain objective function that measures the difference between the model results and the data. To obtain a computationally efficient procedure, the function is minimized with a gradient-based algorithm that determines the gradient by solving the adjoint problem. Variational data assimilation requires implementation of the adjoint model. Even with the use of the adjoint compilers which have become recently available [*Kaminski et al.*, 2003] this is a tremendous programming effort for existing model systems. This is a serious practical disadvantage that hampers new applications of the method.

A number of existing data assimilation schemes are based on Kalman filtering. The Ensemble Kalman Filter [*Evensen*, 1994] is based on a representation of the probability density of the state estimate by a finite number of randomly generated states. The scheme is very easy to implement and does not require the adjoint implementation of the model. The Kalman filter can also be used for parameter estimation problems by augmenting the state vector with the uncertain parameters [*Jazwinski*, 1970]. Generally this approach performs well in case of noise parameters that slowly vary in time (and, therefore, act like states). For constant parameters not affected by noise however, the filter is biased [*Ljung*, 1979]. It is known [*Jazwinski*, 1970] that even for simple linear systems the Kalman filter tends to diverge from the optimal results when the system is not controllable (i.e. when not all states are affected by noise, which is the case for constant parameters).

In this chapter we introduce a new approach to variational data assimilation with a comparable computational efficiency that does not require implementation

of the adjoint of the original model. Our approach is based on model reduction. An ensemble of forward model simulations is used to determine an approximation of the covariance matrix of the model variability. A limited number of leading eigenvectors (EOFs) of this matrix are selected to define a model subspace. By projecting the original model onto this subspace, an approximate linear model is obtained. Once this reduced model is available, its adjoint can be implemented very easily and the minimization process can be solved completely in reduced space with negligible computational costs. If necessary, the procedure can be repeated a few times by generating new ensembles closer to the most recent estimate of the parameters.

The computational costs of the reduced model approach are dominated by the generation of the ensemble of forward model simulations. To obtain an accurate reduced model, however, often a significantly smaller simulation period can be chosen for the ensemble of model simulations than that used in solving the minimization problem. For the original model, the period chosen for the simulations is related to the time scale of this model, while that chosen for the reduced model in the minimization problem is the period over which data is available.

Model-reduced variational data assimilation is computationally very efficient if the number of uncertain control parameters is much smaller than the dimension of the state. Then the controllable subspace is very likely to be of low rank and an accurate reduced model can indeed be obtained. An efficient implementation can also be obtained when the number of data locations is much smaller than the dimension of the state. Then the observable subspace (i.e. the one that can be observed from the data) is very likely to be of low rank and as a result an accurate reduced model can be obtained that is able to reproduce the model results at least in the data locations.

Model-reduced variational data assimilation is more robust than the classical approach. While the adjoint of the tangent linear approximation of the original model produces the exact local gradient, the model reduction approach is based on a statistically linearized model and tends to produce a spatially averaged gradient. As a result, the model-reduced approach is less sensitive to local minima.

Our model-reduced approach was inspired by the work done in reduced-order control, e.g. *Ravindran [2002]; Antoulas, et al. [2004]; Atwell & King [2004]; Doren et al. [2005]*. In control problems, an additive low-dimensional control input is introduced into the model and some performance measure is optimized by minimizing a given performance index. We have modified this approach for large-scale parameter estimation problems, which generally result in highly nonlinear minimization problems.

In this chapter we first summarize briefly in Subsection 5.1 the classical inverse modeling methods (finite-difference method FDM and the adjoint method ADJ). In Subsection 5.2 we describe in detail our model-reduced variational data assimilation approach. A number of case studies are discussed in Subsection 5.3. Here, too, the spatially varying diffusion coefficient in a diffusion model is identified. We fi-

nally present a number of conclusions and some ideas for further improvement of the method in Subsection 5.4.

5.1 Inverse Modeling

A discrete model for the evolution of an atmospheric, oceanic, or coupled system from time t_{i-1} to time t_i is governed by the equation

$$\mathbf{x}(t_i) = M_i [\mathbf{x}(t_{i-1})], \quad (5.1a)$$

$$\mathbf{y}(t_i) = H [\mathbf{x}(t_i)], \quad (5.1b)$$

where $\mathbf{x}$ is the forecast state vector for an n^m -dimensional domain, M_i is a dynamics operator which includes the inputs. The dynamics M_i of the model evolution in a computer simulation or prediction is commonly nonlinear and deterministic; H is an operator that maps the model field to observation space, and $\mathbf{y}$ is the estimated observation. We refer to such a model as *autoregressive* (AR) because the current value of the state $\mathbf{x}(t_i)$ depends, at least in part, on the previous value $\mathbf{x}(t_{i-1})$. The terminology ‘autoregressive’ rises to the fact that $\mathbf{x}$ has a regression on its own past.

In many situations, the forecast becomes more reliable when a set of model parameters within M_i can be found that reduces the misfit between a set of observations $\mathbf{y}^o$ and its estimated observation value $\mathbf{y}$. A quantity that determines this difference is, for example, the weighed sum of squared residuals:

$$J(\alpha_j) = \sum_{i=1}^{n^t} [\mathbf{y}^o(t_i) - \mathbf{y}(t_i)]^T \mathbf{W}^o [\mathbf{y}^o(t_i) - \mathbf{y}(t_i)], \quad (5.2)$$

where $\mathbf{W}^o$ is the observational weight matrix and J is the objective function value that is minimized with respect to a estimate variable α_j , and n^t is the number of time steps in the discrete model. The presented discrete model (Equation 5.1a) can now be rewritten as

$$\mathbf{x}(t_i) = M_i [\mathbf{x}(t_{i-1}), \alpha_j]. \quad (5.3)$$

There are several ways to find an optimal value for α_j such that J is minimal [Press *et al.*, 1992; Tarantola, 1987]. Several techniques compute the gradient of J (denoted as ∇J) as a finite-difference approximation of the sensitivity of J subject to a perturbation of an estimate variable α_j , so:

$$\nabla J_j \equiv \frac{\Delta J}{\Delta \alpha_j} \approx \frac{J(\alpha_j + \Delta \alpha_j) - J(\alpha_j)}{\Delta \alpha_j}. \quad (5.4)$$

A disadvantage of this method is that for each perturbed variable it requires a simulation of the model, and an additional simulation for a reference simulation. From a computational point of view, the method is therefore restricted to a limited number

of variables. The adjoint or variational method [Courant & Hilbert, 1953] is more efficient as it simultaneously finds the gradient of the objective function with respect to all variables:

$$\nabla J_j \equiv \frac{\Delta J}{\Delta \alpha_j} = \sum_{i=1}^{n^t} -[\lambda(t_i)]^T \left[\frac{\partial M_i[\mathbf{x}(t_{i-1})]}{\partial \alpha_j} \right], \quad (5.5)$$

where $\lambda(t_i)$ is the solution of the adjoint model. It needs one forward simulation and a second additional simulation backwards in time with the adjoint model. It is independent of the number of variables to be estimated. However, implementation of the adjoint method is very complicated. Besides that, the minimization can fail as the objective function is not strictly convex and the exact gradient is sensitive to local minima.

5.2 Reduced Model Methodology

5.2.1 Introduction

The model-reduced method used in this chapter falls into the category of *spectral methods*. These methods describe the original state as a truncated series of known basis functions and independent coefficients (i.e. the Galerkin method/Rayleigh-Ritz method). Briefly, this type of model is based upon the simulation of specific model configurations (i.e. the method of *snapshots*, [Sirovich, 1987]). By applying functions that describe the variance in these snapshots, the superfluous state-space dimensions can be eliminated. These functions are widely known as Proper Orthogonal Decompositions (POD) or Empirical Orthogonal Functions (EOF). Eventually, the model operates within the remaining dimensions $\mathbb{R}^{n^p}$ in which $n^p \ll n^m$, and therefore consumes significantly less time. This type of model has been extensively mathematically described [Newman, 1996a,b] and successfully applied within different fields of science, among others, the fields of turbulence and image processing [Sirovich, 1987], rapid thermal chemical vapour deposition [Adomaitis, 1995], fluid dynamics [Hoffman Jørgensen & Sørensen, 2000] and groundwater flow [Vermeulen *et al.*, 2004a,b]. This concept was extended for a groundwater application to handle an inverse modeling problem [Vermeulen *et al.*, 2005a]. The method was restricted to a single model system matrix that needs to be explicitly available. In this chapter a quite generic approach is conducted that does not need this constraint.

5.2.2 Methodology

A disadvantage of the classical inverse methods is that they need to evaluate the original model for each gradient computation, see Figure 5.1a. As a result, a lot of effort is put into single gradient estimation. In contrast to this we propose to exploit more of the information contained in the model simulation, by creating an approximate linear reduced model. The flow chart (Figure 5.1b) shows that the original model is used to generate several snapshots. These snapshots form the basis of a

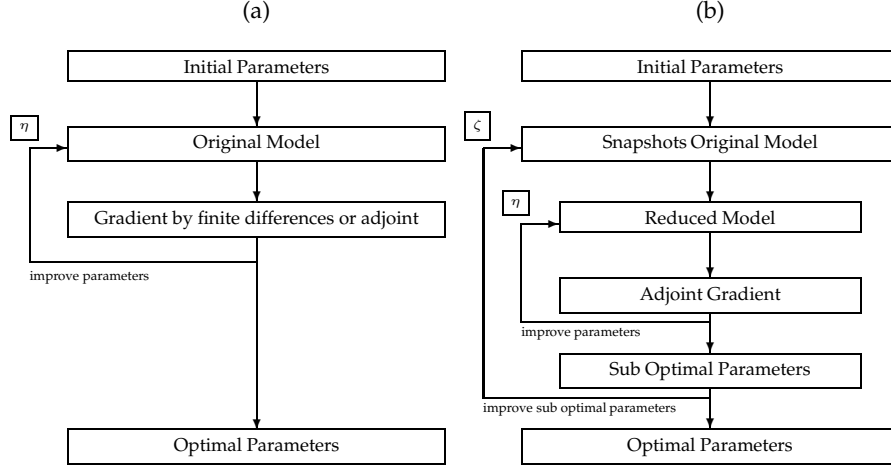


Figure 5.1: Flow chart of the methodology of inverse modeling by use of (a) the classical methods and (b) a reduced model method. The number of gradient iterations are denoted by η and outer iteration cycles by ζ .

reduced model for which it is easy to implement an adjoint algorithm. This model is used to improve the parameters by repeating an inner loop η , until a minimal objective function value is found. This is optimal for the linearized reduced model but can be sub-optimal for the original model. The outer loop ζ is entered again and the reduced model is adjusted to reflect the updated parameters. This sequence continues and converges when the maximal innovation of the parameters between two adjacent ζ cycles is less than a given terminal criterion.

5.2.3 Reduced Model Structure

We start by linearizing the nonlinear high-order model (Equation 5.3) with respect to the variable α_j :

$$\tilde{\mathbf{x}}(t_i) = M_i [\tilde{\mathbf{x}}(t_{i-1}), \boldsymbol{\alpha}^b] + \sum_{j=1}^{n^u} \frac{\partial M_i [\mathbf{x}^b(t_{i-1}), \boldsymbol{\alpha}^b]}{\partial \tilde{\alpha}_j} \Delta \alpha_j, \quad (5.6)$$

where $\tilde{\mathbf{x}}$ is the linearized state vector, n^u is the number of estimate variables α_j , $\tilde{\alpha}_j$ is the estimate variable used for the linearization, and $\mathbf{x}^b$ is the background state for which the corresponding estimate variables $\boldsymbol{\alpha}^b$ are linearized, thus

$$\mathbf{x}^b(t_i) = M_i [\mathbf{x}^b(t_{i-1}), \boldsymbol{\alpha}^b]. \quad (5.7)$$

For a simple simulation model, the function $\partial M / \partial \alpha_j$ can be computed analytically and a comparable model is constructed in reduced resolution for an incremental 4D-VAR [Lawless *et al.*, 2005]. However, for the majority of simulation models, the partial differential can not be computed explicitly, and the function can be ap-

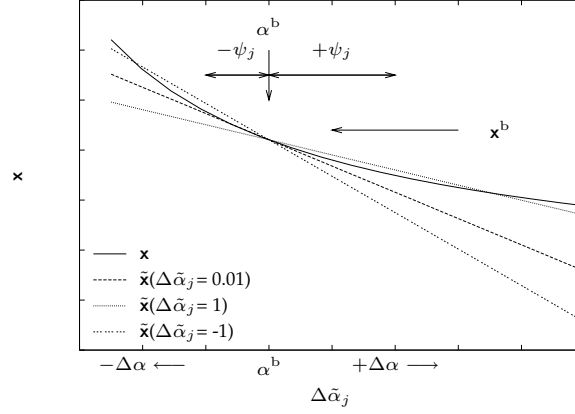


Figure 5.2: Graph of a computed forecasted state vector $\mathbf{x}$ versus $\Delta\alpha_j$, and the forecasted state vector $\tilde{\mathbf{x}}$ with the linearized model. The definition of ψ_j and $\mathbf{x}^b$ are plotted as well.

proximated by

$$\frac{\partial M_i}{\partial \tilde{\alpha}_j} \approx \frac{M_i[\mathbf{x}^b(t_{i-1}), \alpha_j^b + \Delta\tilde{\alpha}_j] - M_i[\mathbf{x}^b(t_{i-1}), \alpha_j^b]}{\Delta\tilde{\alpha}_j}. \quad (5.8)$$

Small values for $\Delta\tilde{\alpha}_j$ in here increase the accuracy for a limited range (ψ_j), whereas large values can increase the overall accuracy for a wider range. A sign change will perform a forward or backward approximation, illustrated in Figure 5.2. From a practical point of view, the sign of $\Delta\tilde{\alpha}_j$ can depend on the direction of the parameter update, and/or its size can decline during the iteration cycle η . The value for ψ_j should depend more-or-less reciprocally on the chosen $\Delta\tilde{\alpha}_j$.

A model can be reduced whenever the state $\tilde{\mathbf{x}}$ can be reconstructed as the linear combination

$$\hat{\mathbf{x}}(t_i) = \mathbf{x}^b(t_i) + \mathbf{P}\mathbf{r}(t_i), \quad (5.9)$$

where $\hat{\mathbf{x}}$ is the approximate linearized state, $\mathbf{P}$ is a matrix of basis vectors that span a subspace $\mathcal{S}$ of the state space $\mathcal{X}$, and $\mathbf{r}$ is a reduced time-varying state vector that is computed by

$$\mathbf{r}(t_i) = \mathbf{N}_i \mathbf{r}(t_{i-1}) + \sum_{j=1}^{n_u} \frac{\partial \mathbf{N}_i}{\partial \tilde{\alpha}_j} \Delta\alpha_j, \quad (5.10a)$$

where

$$\mathbf{N}_i = \mathbf{P}^T \frac{\partial M_i[\alpha^b]}{\partial \mathbf{x}^b(t_{i-1})} \mathbf{P}, \quad (5.10b)$$

$$\frac{\partial \mathbf{N}_i}{\partial \tilde{\alpha}_j} = \mathbf{P}^T \frac{\partial M_i[\alpha_j^b]}{\partial \tilde{\alpha}_j} \mathbf{P}, \quad (5.10c)$$

where $\mathbf{N}_i$ is a reduced dynamics operator. It can be computed by Equation (5.10b), however, in this chapter we assume that the vector function M_i is an unknown quantity and therefore we elaborate another strategy to obtain the matrix $\mathbf{N}_i$. This model requires less computational time, as it simulates a reduced state within the dimensions of a subspace $\mathcal{S} \in \mathbb{R}^{n^p}$ of the original state space $\mathcal{X} \in \mathbb{R}^{n^m}$ with $n^p \ll n^m$. Note that $\mathbf{r}$ represents the response to a perturbation of α_j , and Equation (5.10a) is therefore referred to as a reduced perturbation model because $\mathbf{r}$ remains zero as $\Delta\alpha_j$ equal zero.

5.2.4 Pattern Identification

Pattern identification consists of computing a subspace $\mathcal{S}$ for the original model space $\mathcal{X}$. Among others, this technique is known as Proper Orthogonal Decomposition (POD) or Empirical Orthogonal Functions. These functions can be consisely described as eigenvectors of a covariance matrix constructed from empirical data that are acquired by means of original model simulations. The application considered in this chapter prescribes a reduced model that includes a dependency on α_j . Hence, the snapshots should provide insight in the sensitivity of $\mathbf{x} \in \mathcal{X}$ with respect to $\alpha_j \in \mathcal{Z}$. The projection matrix $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_{n^u}\}$ spans the parameter space $\mathcal{Z}$ and maps each estimate variable to the model field. Each element within the vector $\mathbf{z}_i$ is given a value of one to reflect zone i that is affected by α_i ; a zero value is assigned to the remaining elements. As in the classic finite-difference method (Equation 5.4), each sensitivity subject to α_j is computed along this corresponding definition $\mathbf{z}_j$. However, when we perturb a single parameter, it probably influences only a limited area within the entire model domain and yields a snapshot vector that contains many elements that are close to zero. This strategy is disadvantageous for the quality of the basis vectors as the algorithm for this is sensitive to any correlation in the snapshots [Cazemier *et al.*, 1998]. To avoid this, the snapshots should reflect the influence of all parameters simultaneously [Vermeulen *et al.*, 2005a]. A practical method for this is to apply a coordinate transformation by an orthonormal projection matrix $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_{n^u}\}$. The matrix can be chosen artificially, and transforms $\Delta\tilde{\alpha}$ into

$$\Delta\tilde{\beta} = \mathbf{F}^T \mathbf{Z} \Delta\tilde{\alpha}. \quad (5.11)$$

Instead of perturbing along the original definition of each variable $\mathbf{z}_j \Delta\tilde{\alpha}_j$ (Equation 5.8), we perturb along the vectors $\mathbf{f}_j$ by $\Delta\tilde{\beta}_j$, see Figure 5.3. So a single snapshot vector $\mathbf{e}$ becomes

$$\mathbf{e}_j(t_i) = \frac{\partial M_i[\mathbf{x}^b(t_{i-1}), \alpha_j^b]}{\partial \tilde{\beta}_j} \approx \frac{M_i[\mathbf{x}^b(t_{i-1}), \alpha^b + \Delta\tilde{\beta}_j \mathbf{f}_j] - M_i[\mathbf{x}^b(t_{i-1}), \alpha^b]}{\Delta\tilde{\beta}_j}. \quad (5.12)$$

The collection of normalized snapshot vectors are then collected in a matrix $\mathbf{E} = \{\mathbf{e}_1(t_1), \dots, \mathbf{e}_n(t_{n^t})\}$. We solve the eigenvalue problem for the covariance matrix

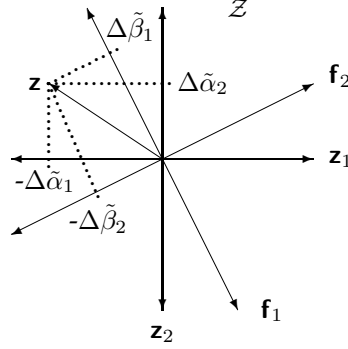


Figure 5.3: Definition of the parameter space $\mathcal{Z} \in \mathbb{R}^2$ with vector $\mathbf{z}$ spanned by the original basis $\mathbf{z}_i$ and coefficients $\Delta\tilde{\alpha}_i$, and the alternative basis $\mathbf{f}_i$ that determines $\mathbf{z}$ by the coefficients $\Delta\tilde{\beta}_i$.

$\mathbf{E}^T \mathbf{E}$ and we scale each eigenvalue λ_i according to

$$\varphi_i = \lambda_i / \text{sum}(\lambda) \times 100\%. \quad (5.13)$$

This expresses the relative amount of variance φ_i that each eigenvector $\mathbf{p}_i$ represents. These steps are more extensively elaborated in Subsection 3.1.3 on page 57. The basis becomes $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{n^p}]$ and contains a selected set of eigenvectors. The number of patterns n^p depends on the desired accuracy of the reduced model and is expressed by the expected variance $\varphi^e = \sum_{i=1}^{n^p} \varphi_i$.

5.2.5 Generic Reduced Model Formulation

A reduced model operates within the subspace $\mathcal{S}$ that is spanned by $\mathbf{P}$. The reduced model formulation was initially presented in Equation (5.10a) and can be written as a partitioned matrix

$$\begin{bmatrix} \mathbf{r} \\ \Delta\beta \end{bmatrix} (t_i) = \begin{bmatrix} \mathbf{N} & \mathbf{N}^\beta \\ \mathbf{0} & \mathbf{I} \end{bmatrix}_i \begin{bmatrix} \mathbf{r} \\ \Delta\beta \end{bmatrix} (t_{i-1}), \quad (5.14)$$

where $\Delta\beta$ is the projected prior estimate (Equation 5.11), $\mathbf{N}$ and $\mathbf{N}^\beta$ are reduced dynamics operators that can be computed by a linear transformation (i.e. a change of coordinates) of $\partial M_i / \partial \mathbf{x}_j^b$ and $\partial M / \partial \tilde{\beta}_j$ (Equation 5.12) upon a set of basis vectors $\mathbf{P}$, so

$$\mathbf{N}_i = \mathbf{P}^T \frac{\partial M_i[\alpha^b]}{\partial \mathbf{x}^b(t_{i-1})} \mathbf{P}, \quad (5.15a)$$

$$\mathbf{N}_i^\beta = \mathbf{P}^T \left\{ \frac{\partial M_i[\mathbf{x}^b(t_{i-1}), \alpha_1^b]}{\partial \tilde{\beta}_1}, \dots, \frac{\partial M_i[\mathbf{x}^b(t_{i-1}), \alpha_{n^p}^b]}{\partial \tilde{\beta}_{n^p}} \right\}. \quad (5.15b)$$

For these applications, in which the operator M_i is not explicitly available, it can be approximated by linearizing M_i with respect to $\mathbf{x}^b$ by finite differences:

$$\left[\frac{\partial M_i[\boldsymbol{\alpha}^b]}{\partial \mathbf{x}^b(t_{i-1})} \right]_j \approx \frac{M_i[\mathbf{x}^b(t_{i-1}) + \epsilon \mathbf{i}_j, \boldsymbol{\alpha}^b] - M_i[\mathbf{x}^b(t_{i-1}), \boldsymbol{\alpha}^b]}{\epsilon} \quad (5.16)$$

where ϵ is the interval for which the partial differential is linearized, $j \in \{1, \dots, n^m\}$ and $\mathbf{i}_j$ is the j^{th} column of the identity matrix $\mathbf{I}$ with dimension $[n^m \times n^m]$. It approximates the partial differential by slightly perturbing the function M_i in the direction of each node j that is represented by $\epsilon \mathbf{i}_j$. It yields a matrix $\partial M_i / \partial \mathbf{x}^b(t_{i-1})$ (Jacobian) which needs to be post multiplied with $\mathbf{P}$. To avoid such a huge Jacobian, we do not need it anyhow, we incorporate the post multiplication with $\mathbf{P}$ by perturbing only along the pattern directions, so Equation (5.16) becomes

$$\frac{\partial M_i[\boldsymbol{\alpha}^b]}{\partial \mathbf{x}^b(t_{i-1})} \mathbf{p}_j \approx \frac{M_i[\mathbf{x}^b(t_{i-1}) + \epsilon \mathbf{p}_j, \boldsymbol{\alpha}^b] - M_i[\mathbf{x}^b(t_{i-1}), \boldsymbol{\alpha}^b]}{\epsilon}. \quad (5.17)$$

The reduced operator $\mathbf{N}_i$ (Equation 5.15a) can now be written as:

$$\mathbf{N}_i = \mathbf{P}^T \left\{ \frac{\partial M_i[\boldsymbol{\alpha}^b]}{\partial \mathbf{x}^b(t_{i-1})} \mathbf{p}_1, \dots, \frac{\partial M_i[\boldsymbol{\alpha}^b]}{\partial \mathbf{x}^b(t_{i-1})} \mathbf{p}_{n^p} \right\}. \quad (5.18)$$

The dimension of the reduced model depends eventually on the number of estimate variables n^u and the number of basis vectors n^p . Since the reduced model is autoregressive and operates in the low-dimensional space $\mathbb{R}^{n^u + n^p}$, it needs less computational time compared to the original model that operates in $\mathbb{R}^{n^m}$.

5.2.6 Adjoint of the Reduced Model

The results of the simulations are used to minimize an approximate objective function for the reduced model, defined as:

$$\hat{J}(\Delta \boldsymbol{\alpha}) = \sum_{i=1}^{n^t} \left[\mathbf{P}^T (\mathbf{y}^o(t_i) - H[\mathbf{x}^b(t_i)]) - \mathbf{r}(t_i, \Delta \boldsymbol{\beta}) \right]^T \mathbf{P}^T \mathbf{W}^o \mathbf{P} \left[\mathbf{P}^T (\mathbf{y}^o(t_i) - H[\mathbf{x}^b(t_i)]) - \mathbf{r}(t_i, \Delta \boldsymbol{\beta}) \right], \quad (5.19)$$

where the observation value $\mathbf{y}^o(t_i)$ is projected on the basis $\mathbf{P}$ and needs to be corrected for the background state $\mathbf{x}^b(t_i)$ which is mapped on the observational space by the operator H because $\mathbf{r}$ yields a perturbation upon the background state (Equation 5.9). It seems, however, illogical to optimize

$$\Delta \boldsymbol{\alpha} = \mathbf{Z}^T \mathbf{F} \Delta \boldsymbol{\beta} \quad (5.20)$$

instead of $\Delta \boldsymbol{\beta}$. The main reasons for optimizing $\Delta \boldsymbol{\alpha}$ are that it hardly affects the efficiency, distinguishes the method from others more clearly, and allows direct comparison of the gradient to other algorithms. The objective function (Equation 5.19)

is minimized with respect to $\Delta\alpha_j$ by means of a reduced adjoint-based gradient algorithm

$$\nabla \hat{J}_j \equiv \frac{\Delta \hat{J}}{\Delta \alpha_j} = \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T - \left\{ \begin{bmatrix} \mathbf{N}^\beta \\ \mathbf{0} \end{bmatrix}_i \mathbf{F}^T \mathbf{d}(\mathbf{z}_j \alpha_j) \right\}, \quad (5.21)$$

where $\mathbf{z}_j$ is the j^{th} vector in $\mathcal{Z}$ that maps the j^{th} estimate variable on the model field, $\mathbf{F}$ maps the estimate variables on the coordinate system that is affected by $\Delta\beta_j$, and $\underline{\lambda}(t_i)$ is the reduced adjoint state model. The reduced adjoint state is easy to implement through the linear character of the reduced model (elaborated in Appendix 5A). Once the gradient is obtained, the reduced model (Equation 5.14) is used again to explore the evaluation of $\hat{J}$ along the direction of the gradient (i.e. a line search is carried out). From a new location along that direction, a different gradient is determined (Equation 5.21). As both models are low dimensional, the minimization requires a negligible amount of simulation time.

5.2.7 Efficiency

The efficiency of the minimization process is mainly influenced by

- **Snapshots:** The maximal number of snapshot simulations n^e is equal to the number of estimate variables times the number of time steps. However, it is difficult to give a minimal number because there is a trade-off between the effort put in the snapshot simulations and the ‘ $\hat{J}$ -reduction capacity’ of the reduced model evolved. For example, it is possible to start with half the maximal number of simulations such that the minimization procedure will first focus on the main directions and add more snapshots as the process proceeds.
- **Outer Iteration:** The number of outer iterations ζ is influenced by the chosen abortion criterion γ . It should be chosen not too small ($\gamma \geq 0.1$), as this causes jumping of sub-optimal posterior estimates around the optimal global solution. Since the reduced model probably overestimates α_j^t through the applied linearizations, it probably yields a redundancy of ζ whenever a threshold ψ_j is applied. This threshold prevents the model from updating the variables to extraordinary values, far beyond the background estimates α_j^b , see Figure 5.2.

The next subsection will illustrate that high efficiencies can be achieved.

5.3 Synthetic Cases

5.3.1 Introduction

Generally, large-scale numerical models can be used to describe a particular transport through a certain medium. The generic time-dependent diffusion equation can be written as a partial differential equation for a discretised numerical scheme,

so

$$\mathbf{S} \frac{\partial \mathbf{x}}{\partial t} = \nabla^2 \kappa \mathbf{x} + \mathbf{q}(\mathbf{x}, t) ; \mathbf{x} \in \Omega, \quad (5.22)$$

where κ [T^{-1}] contains the diffusion coefficient, $t \in [0, \infty)$ [T] is the time, $\mathbf{S}$ [-] is a storage coefficient, and $\mathbf{q}$ [LT^{-1}] is a linear and/or nonlinear Neumann condition that determines imposed fluxes. The operator $\nabla^2 = \sum_{i=1}^n \partial^2 / \partial x_i^2$ and is the Laplacian for the current model domain $\Omega \in \mathbb{R}^{n^m}$. A solution to the partial differential equation can be obtained by imposing certain boundary conditions and applying a finite-difference approximation for the space and time derivative of $\mathbf{x}$.

In the next subsections several synthetic cases are described. The main purpose of this is to illustrate the methodology and to demonstrate its efficiency and robustness compared to classic minimization algorithms, i.e. the method of finite differences (Equation 5.4) and the adjoint-based method (Equation 5.5). The parameter minimization for each case is computed by the Variable Metric Method (Quasi Newton) [Press *et al.*, 1992] which adjusts the acquired gradient by an approximated Hessian. The iteration cycle is aborted when the maximal innovation of the parameters

$$\gamma = \sum_j^u \| [\alpha_j^b]^\eta - [\alpha_j^b]^{\eta-1} \| \leq 0.1. \quad (5.23)$$

Unless stated differently, the number of evaluations of the original model τ is used to compare the different methods. This includes the number of evaluations that were eventually needed for the line searches. Another quality criterium is the relative accuracy of the posterior estimate α_j^a . This is determined by:

$$\varepsilon = 100\% \cdot \|\alpha^t\| - \sqrt{\frac{1}{n^t} \sum_{j=1}^{n^t} \begin{cases} [\alpha_j^a]^2 & \text{if } \alpha_j^a \geq 1 \\ [\alpha_j^a]^{-2} & \text{if } 0 < \alpha_j^a < 1, \end{cases}} \quad (5.24)$$

where α^t is the ‘true’ value for the estimate variable, which is equal to 1 for all cases.

5.3.2 Problem Description

The synthetic cases in the following subsections increase in complexity with respect to the number n^u of estimate variables $\mathcal{Z} \in \mathbb{R}^{n^u}$. The corresponding network consisted of a number of columns and rows dimensioned by $\Delta x \equiv \Delta y = 10$ m, and κ and $\mathbf{S}$ were kept at 1 day^{-1} and 0.27 for the entire model domain Ω , respectively. In each case, the estimate α_j affected the parameter κ . Two linear Neumann conditions were defined near the center of the model that varied randomly between $-2 \leq \mathbf{q}(t_i); i \in \{1, \dots, n^t\} \leq 2 \text{ mday}^{-1}$. The time-invariant states for the Dirichlet conditions were $\Gamma(t_i); i \in \{1, \dots, n^t\} = 0$ m and located at the entire left side of the model domain. Each case was simulated for 10 time steps with $\Delta t_i = 10$ days. Finally, a set of synthetic measurements $\mathbf{y}^o(t_i); i \in \{1, \dots, n^t\}$ was generated for different locations.

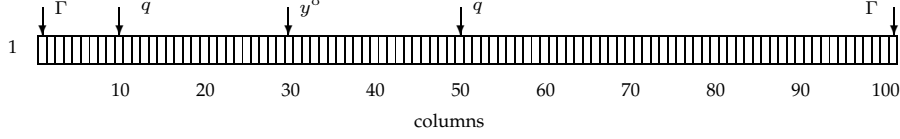


Figure 5.4: Model network for Case I ($\mathcal{Z} \in \mathbb{R}^1$), the location of boundary conditions ($\mathbf{q}, \Gamma$) and the observation $\mathbf{y}^\circ$ is given.

Case I - $\mathcal{Z} \in \mathbb{R}^1$

In this case, a one-dimensional model was discretised by 101 columns, and two Neumann conditions that render the system, see Figure 5.4. The estimate variable α_1 was defined for the entire domain, so $\mathbf{z}_1 = 1$.

To illustrate the pattern identification procedure as described in Subsection 5.2.4, the snapshots $\mathbf{E}$ were generated according to Equation (5.12). It should be noticed that for this one-dimensional parameter case $\Delta\alpha_1 = \Delta\beta_1$. The resulting snapshots ($n^p = 10$) contained the variance of the original state vector related to the Dirichlet and Neumann boundary condition, and a perturbation of the estimate variable, see Figure 5.5a. In essence, the reduced model needs to represent only this model behavior. The resulting patterns are depicted in Figure 5.5b and explained $\varphi^e = 99.99\%$ of the snapshot variance. Only four of them described some significant variance ($\varphi_i > 1\%$). For this application we have defined $\varphi^e = 99.99\%$ which resulted in six patterns. We neglected the others though most of the variance of the original model could be described with respect to the boundary conditions and the estimate variable. With the selected set of patterns $\mathbf{P}$, a reduced model (Equation 5.14) was constructed.

For several values of $\Delta\alpha$ the approximate objective function $\hat{J}$ (Equation 5.19)

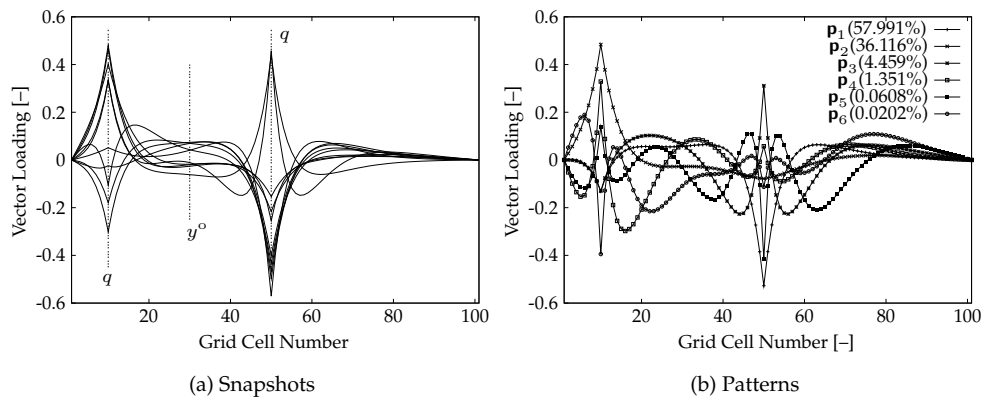


Figure 5.5: Graph of (a) the snapshot vectors given by $\mathbf{e}_i$; $i \in \{1 - 10\}$ and the position of the Neumann conditions $\mathbf{q}$ and the observation location $\mathbf{y}^\circ$. (b) Depicts the set of basis vectors $\mathbf{p}_i$ with their corresponding relative importance φ_i explaining $\varphi^e = 99.99\%$.

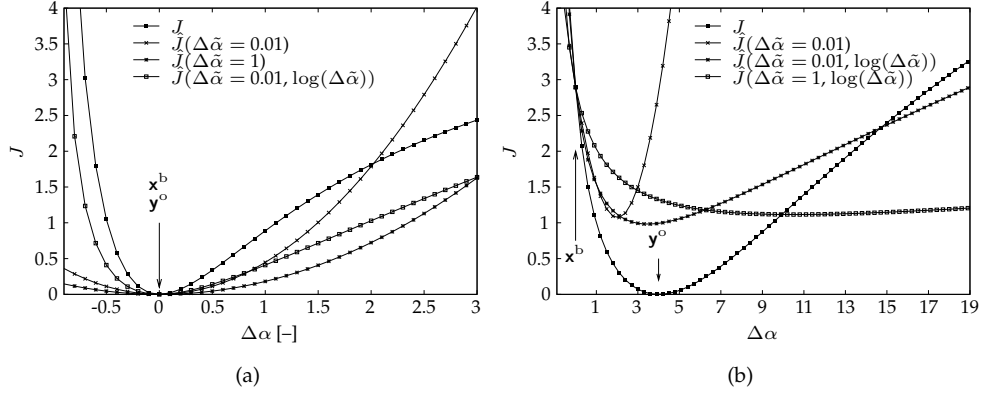


Figure 5.6: Graph of the objective function J and $\hat{J}$ versus a guess value $\Delta\alpha$. The $\hat{J}$ is computed by different reduced models by use of (a) a linearized state vector $\mathbf{x}^b$ equal to the observation $\mathbf{y}^o$, and (b) whereby $\mathbf{x}^b \neq \mathbf{y}^o$. Both models are defined by $\varphi^e = 99.99\%$.

was evaluated, see Figure 5.6a. For reasons of comparison, the objective function J (Equation 5.2) is also depicted. The shape of $\hat{J}$ is strictly parabolic because the reduced model is linear and the objective function is quadratic. Moreover, its shape is sensitive to $\Delta\tilde{\alpha}$ and whenever we increase it, we observed that $\hat{J}$ becomes flattened, see e.g. $\hat{J}(\Delta\tilde{\alpha} = 1)$. Nevertheless, the location of $\min(\hat{J})$ remains identical to that of $\min(J)$. To improve the trajectory of the fit between the two objective functions, a scaling of $\Delta\tilde{\alpha}$ can be applied according to the expected influence of the estimate variables on the objective function. E.g., in this example α affects κ , and κ affects $\mathbf{x}$ logarithmically. Hence a corresponding scaling operation is recommended, so $\Delta\tilde{\alpha} = \log(\Delta\tilde{\alpha})$. It can be seen in Figure 5.6a that this scaling operation improves the fit between J and $\hat{J}$.

In realistic problems, the point of linearization is not identical to the optimal parameters that yield the least error with the observations, i.e. $\mathbf{x}^b \neq \mathbf{y}^o$ depicted in Figure 5.6b. In this case a relative small value for $\Delta\tilde{\alpha} = 0.01$ underestimated the optimal location. A relative large log-transformed value for $\Delta\tilde{\alpha} = 1$ shifted the location of $\min(J)$ extraordinary. For the synthetic case, the relative small log-transformed value $\Delta\tilde{\alpha} = 0.01$ performed best. It can be stated that $\Delta\tilde{\alpha}$ relates to ε (Equation 5.24), which is unfortunately not known beforehand. A relevant future research direction is fine tuning this parameterization.

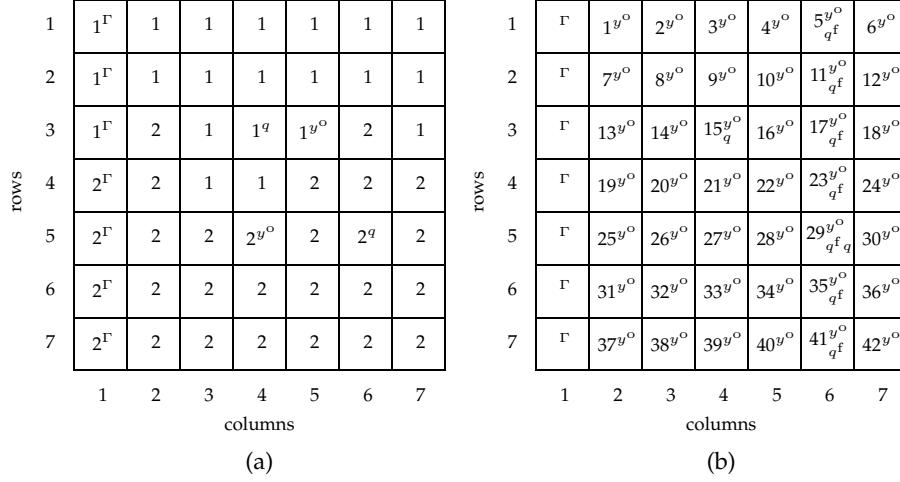


Figure 5.7: Model network for Case II & III ($\mathcal{Z} \in \mathbb{R}^{42}$), the location of linear Neumann conditions q , nonlinear Neumann conditions q^f , Dirichlet conditions Γ , and observations y^o are given. The definition of the zones are given in the grid cell centers.

Case II - $\mathcal{Z} \in \mathbb{R}^2$

In this case, the diffusion Equation (5.22) is used to estimate two variables that affect the parameter κ for two zones. Within each of these zones a synthetic observation was created, see Figure 5.7a. The objective function J is depicted in Figure 5.8 for the range $0.1 \leq \log(\alpha_j) \leq 3.1$, in which the optimal value is located at $\alpha^t = 1$. For the situation in which the prior estimates were assumed to be $\alpha_1 = 2.5$ and $\alpha_2 = 0.3$, this minimal value was eventually found after $\eta = 11$ iterations with the original adjoint method (Equation 5.5), see Table 5.1a.

To optimize the prior estimates with the reduced model, we constructed a model for $\alpha_1^b = 2.5$ and $\alpha_2^b = 0.3$. It yielded a reduced model with six patterns that describe $\varphi^e = 99.99\%$ of the snapshot variance. Combined with two estimate variables, the reduced model operated eventually in $\mathbb{R}^{6+2}$, instead of $\mathbb{R}^{42 \times 2}$. This reduced model decreased $\hat{J}^\eta$ significantly, and found a new sub-optimum at $[\alpha_1^a]^\eta = 1.53$ and $[\alpha_2^a]^\eta = 0.84$, see Figure 5.9a. The process reentered the outer loop ζ (Figure 5.1), and hence the reduced model and the surface of $\hat{J}$ changed. The sub optimum from the previous model $\hat{J}^\eta$, did not appear to be an optimum for the updated model and after another minimization loop, a different suboptimum was found again at $[\alpha_1^a]^{\eta+1} = 0.97$ and $[\alpha_2^a]^{\eta+1} = 1.037$. This sequence was reiterated, and after $\eta = 4$ cycles the optimal values for α_j were found, see Table 5.1b. Compared to the classic methods, almost an identical number of simulations of the original model were needed. For a different situation in which less variance was included ($90.39\% \leq \varphi^e \leq 95.62\%$), the reduced model still converged, see Table 5.1c. An explanation for this is that for the latter case, the surface of $\hat{J}$ did not depend on the patterns that were eliminated. These patterns describe some model variance, irrelevant to the observations.

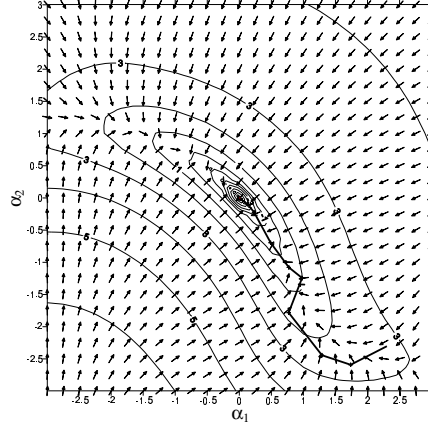


Figure 5.8: Surface of J for $\log(\alpha_1)$ with respect to $\log(\alpha_2)$, the arrows represent the adjoint gradient ∇J .

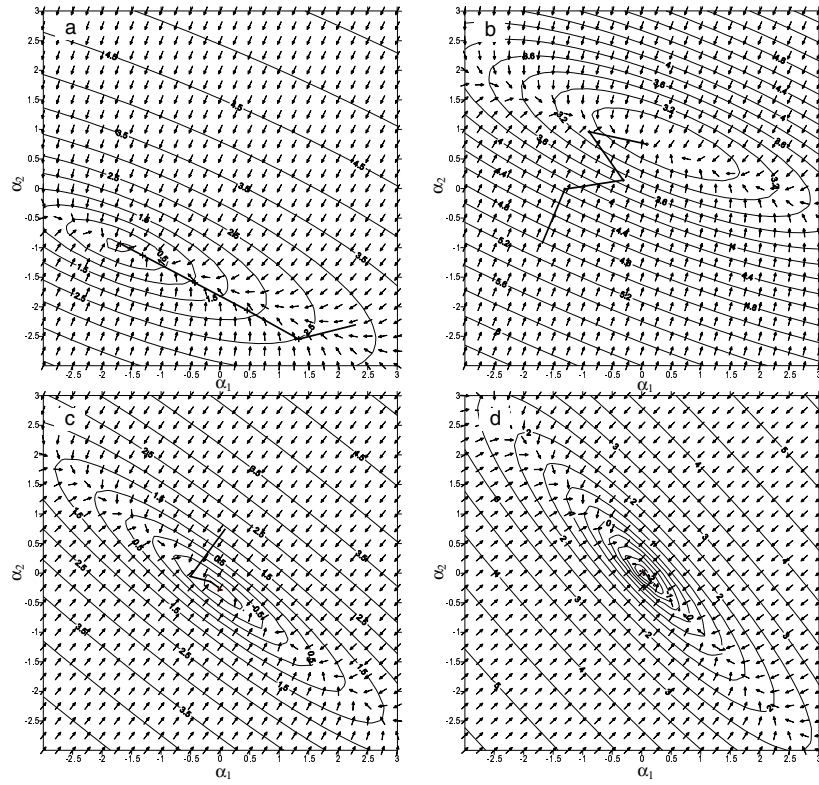


Figure 5.9: Surface of $\hat{J}$ for subsequent outer iterations (a) $\varsigma = 1$, (b) $\varsigma = 2$, (c) $\varsigma = 3$, (d) $\varsigma = 4$ for $\log(\alpha_1)$ with respect to $\log(\alpha_2)$. The arrows represent the gradient $\nabla \hat{J}$. The reduced model is linearized for $\alpha_1^b = 2.5$ and $\alpha_2^b = 0.3$ and explains $\varphi^e = 99.99\%$ model variance.

Table 5.1: Results for Case II for a minimization of two estimate variables α_1 and α_2 with (a) the classic methods and (b-c) a reduced model method. This is expressed by the number of gradient iterations η , the number of outer iteration cycles ζ , the number of simulations with the original model τ , the relative estimation error ε , and the innovation γ . For each ζ -cycle, $\hat{J}^b$ and $\hat{J}^a$ express the prior and posterior objective function value, respectively.

(a) Original Model – [†] finite-difference gradient (Eq. 5.4); [‡] adjoint gradient (Eq. 5.5)								
η	τ	α_1	α_2	J^η		ε	γ	$\mathbb{R}^n$
0		2.50	0.30	0.4557		115.55		42
6 [†]	26	1.01122	0.99225	$0.3684 \cdot 10^{-4}$		0.1597	$0.496 \cdot 10^{-3}$	
11 [‡]	32	0.99988	1.00022	$0.2274 \cdot 10^{-7}$		$0.4804 \cdot 10^{-2}$	$0.6366 \cdot 10^{-3}$	

(b) Reduced Model – $\varphi^e = 99.99\%$								
ζ	τ	α_1	α_2	$\hat{J}^b$	$\hat{J}^a$	ε	γ	$\mathbb{R}^n$
1	7	2.50	0.30	0.3603	0.09657	194.63	1.50	8
2	14	1.5259	0.8353	0.1215	$0.5471 \cdot 10^{-2}$	37.14	0.974	8
3	21	0.9733	1.0373	$0.4179 \cdot 10^{-3}$	$0.7051 \cdot 10^{-4}$	3.24	0.5526	8
4	28	0.9996	0.9997	$0.6218 \cdot 10^{-6}$	$0.5704 \cdot 10^{-7}$	0.031	0.0375	8

(c) Reduced Model – $90.39 \leq \varphi^e \leq 95.62\%$								
ζ	τ	α_1	α_2	$\hat{J}^b$	$\hat{J}^a$	ε	γ	$\mathbb{R}^n$
1	4	2.50	0.30	0.3603	0.10736	194.63	1.50	4
2	8	1.4689	0.9198	0.1523	0.01125	29.27	1.031	4
3	12	1.0006	1.0028	$0.1913 \cdot 10^{-4}$	$0.23437 \cdot 10^{-5}$	0.1708	0.468	4
4	16	0.9992	1.0009	$0.1996 \cdot 10^{-6}$	$0.18753 \cdot 10^{-6}$	0.0873	0.00191	4

Case III - $\mathcal{Z} \in \mathbb{R}^{42}$

The efficiency and performance of the minimization problem depends for most of the part on the dimension and complexity of the parameter space $\mathcal{Z}$. Furthermore, it relates to the identifiability of estimates and whether the prior estimate α_j is already near its optimal value α_j^* . To illustrate this, we applied a parameter minimization to a 42-dimensional parameter problem, so $\mathcal{Z} \in \mathbb{R}^{42}$. For this case, each active cell (i.e. each grid cell without a Dirichlet condition) possesses an observation $\mathbf{y}^o$ and is part of an individual zone $\mathbf{z}_j$, see Figure 5.7b. Furthermore, several conditions $\mathbf{q}^f$ were added that describe a flux that behaves nonlinearly to the state, so:

$$\mathbf{q}^f(\mathbf{x}, t) = \begin{cases} \mathbf{x} \cdot 0.1 & ; \text{ if } \mathbf{x} \leq 0 \\ 0 & ; \text{ if } \mathbf{x} > 0 \end{cases} . \quad (5.25)$$

To obtain insight in the robustness of the reduced model, we generated different samples of prior estimates from which the objective function was minimized. A constrained Monte Carlo sampling scheme was used known as the *Latin Hypercube Sampling* (LHS) [McKay et al., 1979; Iman & Shortencarier, 1984] This method divides the range of each α_j into n non-overlapping intervals on the basis of equal probability. One value is randomly sampled from each interval with respect to the probability density in the interval. The values are then paired in a random manner with the other valued α s. We sampled 5 times from 10 different log-normal distri-

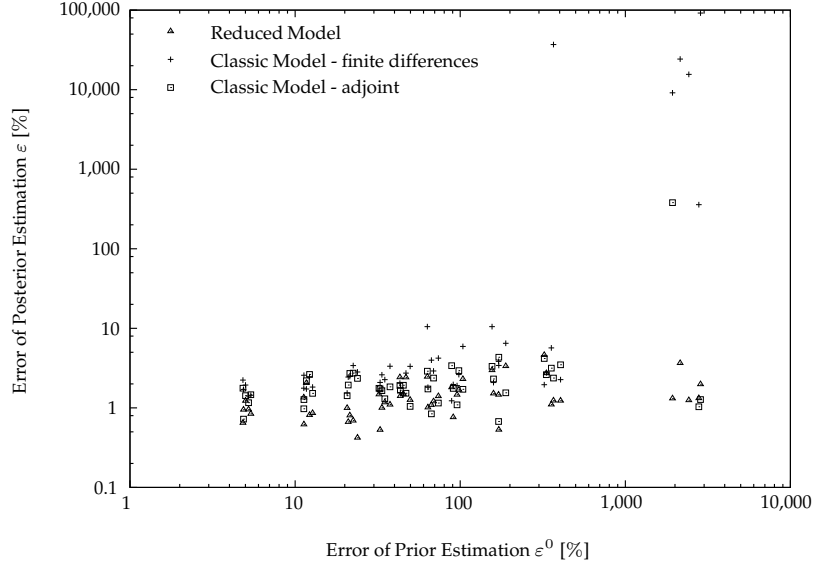


Figure 5.10: Graph of the relative error ε of the prior estimate versus the posterior estimate, for the reduced model ($\varphi^e = 99.99\%$) as well as for two classic methods.

butions $\mathcal{N}^{\log}(\mu, \sigma)$, in which $\mu = 1$ and $\sigma^2 \in \{0.1, 0.2-1, 10\}$. For those samples, the minimization results are depicted in Figure 5.10.

For relative simple disturbances of the prior estimate ($\varepsilon \leq 60\%$), each of the methods described in this chapter succeeded in computing the correct posterior estimate value. Since the adjoint method yields an exact gradient, it is sensitive to local disturbances in the surface of J and this is the main reason of failure when $\varepsilon > 1,000\%$. Although the finite-difference method is more sensitive to this phenomenon, its robustness can be artificially increased by starting with a large finite-difference step size and decreasing the step size while proceeding. Of course, these results are specific for the current application, and can be rather different from another applications. Nevertheless, the reduced model converged fairly constantly and was not sensitive to the relative error ε of the prior estimate. The explanation for this result is twofold: (1) the applied linearization ‘flattens’ the objective function, and (2) the reduced model describes a part of the total model variance, and local disturbances in the objective function are mainly caused by less important and local variances that are not present in the reduced model.

The efficiency of the reduced model is expressed by τ and represents the number of model simulations of the original model and, as expected, τ increases with ε , see Figure 5.11. This figure shows that for the reduced model τ is almost one order of magnitude less than for the finite-difference method. Moreover, the reduced model offers an efficiency comparable to the adjoint method, without the burden of implementation of the adjoint.

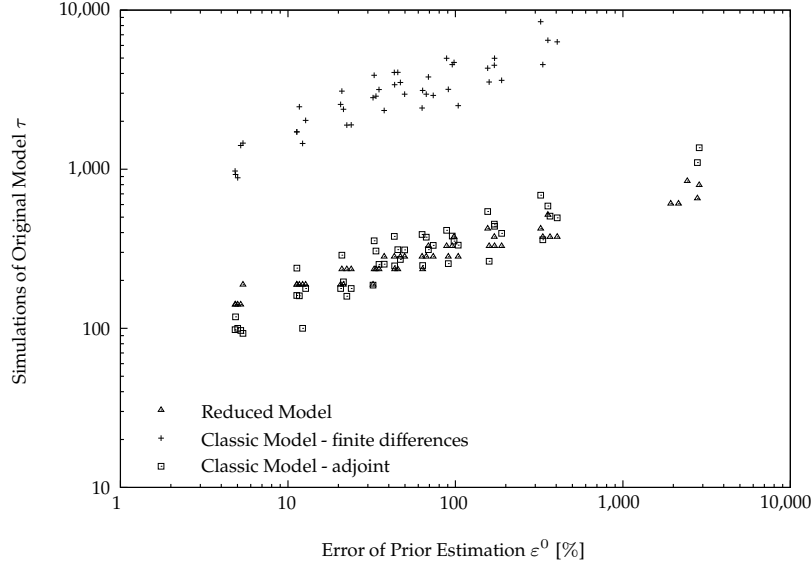


Figure 5.11: Graph of the relative error ε of the prior estimate versus the number of original simulations (τ) for the reduced model ($\varphi^e = 99.99\%$) and for two classic methods.

5.4 Conclusions

This chapter describes a new approach to the calibration of complex numerical simulation models. The main advantages of the proposed concept are that it is generic in its kind, easy to implement, applicable to complex models that consists of multiple systems, and very time efficient compared to current classic minimization sequences.

The concept is based on the computation of a reduced model by use of a number of simulations (i.e. snapshots) of the original model. Those snapshots describe the variance of the original model with respect to its dynamics and parameters. By selecting the main eigenvectors of the covariance of the snapshots, a reduced model is created that operates within a reduced space. It models the effect of perturbations of the parameters with a negligible amount of simulation time. As the adjoint algorithm for this reduced model is easy to implement, the reduced model approach is capable of finding the sub-optimal set of parameters that yields a minimal objective function value for the reduced model. With these parameters, the sequence is reiterated until the parameter values converge.

The algorithm was tested on several nonlinear synthetic cases in which the diffusion coefficient has been estimated in a diffuse type model. The results were compared with two widely used minimization algorithms, the method of finite differences and the adjoint method. From several minimization problems that increased in complexity, the reduced model always converged in contrast to the classic methods. These methods failed if the prior estimate was too distorted compared to its true value. Moreover, the reduced model approach required more or less the same

number of simulations of the original model as the adjoint method. Since the results from the synthetical cases are promising, some issues can be addressed to improve the method's efficiency even more:

- Since the approximation of the Hessian is used to update the parameters within the inner iteration cycle η , a comparable algorithm can be applied to the outer iteration loop ζ .
- The accuracy of the reduced model is directly related to the step size and direction of the linearization (Equation 5.8). These could be exploited during the outer iteration by adding some intelligence.
- The patterns could be varied over time because this can reduce the overall number of patterns. On the other hand, time propagating phenomena (e.g. waves) are inefficiently described by POD [Hooimeijer, 2001]. For these cases, the POD should be time varying and during the reduced model simulation, the time-dependent coefficients should be transformed from the subspace described by a POD for time step i into the subspace of the adjacent POD for time step t_{i+1} .
- To minimize the objective function, the reduced model needs only to forecast the state accurately for the observations. Hence, it is possible to construct only a reduced model for the observations. A main advantage in this is that the effort of the algebraic computations (i.e. the eigenvalue decomposition and linear transformations) is reduced, and eventually fewer patterns will be needed.

5A Derivation of the Reduced Adjoint State Variable

In Equation (5.21), the reduced adjoint state variable $\underline{\lambda}_{i+1}$ is introduced to compute the gradient of the approximate objective function $\hat{\mathcal{J}}$ with respect to each estimate $\Delta\alpha_j$, so:

$$\hat{\mathcal{J}}(\Delta\alpha_j) = \hat{\mathcal{J}}^{\bullet}(\Delta\alpha_j) + \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T \{ \mathbf{r}^p(t_i) - \mathbf{N}_i^p [\mathbf{r}^p(t_{i-1})] \}, \quad (5.26a)$$

where $\hat{\mathcal{J}}^{\bullet} \equiv \hat{\mathcal{J}}$ and the partitioned matrix $\mathbf{N}^p$ and vector $\mathbf{r}^p$ are defined as

$$\mathbf{N}_i^p = \begin{bmatrix} \mathbf{N} & \mathbf{N}^{\beta} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}_i \quad (5.26b)$$

$$\mathbf{r}^p(t_{i-1}) = \begin{bmatrix} \mathbf{r} \\ \Delta\beta \end{bmatrix} (t_{i-1}), \quad (5.26c)$$

where $\Delta\beta$ represents a projected perturbation of the original perturbed estimate variables (Equation 5.11). After a Taylor series expansion for the variables, $\underline{\lambda}(t_i)$, $\mathbf{r}^p(t_i)$, $\mathbf{r}^p(t_{i-1})$ and $\Delta\alpha_j$, the expression yields:

$$\begin{aligned}\Delta\hat{J} &= \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T \left[-\mathbf{N}_i^p \Delta\mathbf{r}^p(t_{i-1}) + \Delta\mathbf{r}^p(t_i) + \frac{\partial\mathbf{r}(t_i)}{\partial\Delta\alpha_j} \Delta\alpha_j \right] \\ &+ \left[\frac{\partial J}{\partial\mathbf{r}^p(t_i)} \right]^T \Delta\mathbf{r}^p(t_i).\end{aligned}\quad (5.27)$$

The reduced adjoint state can be solved by re-expressing $\mathbf{r}^p(t_{i-1})$ to be $\mathbf{r}^p(t_i)$:

$$-[\underline{\lambda}(t_i)]^T \mathbf{N}_i + [\underline{\lambda}(t_{i-1})]^T = \left[\frac{\partial J}{\partial\mathbf{r}^p(t_i)} \right]^T, \quad (5.28)$$

which yields the reduced adjoint model $\underline{\lambda}(t_i)$; $i \in \{n^t, \dots, 1\}$ to be solved backwards in time:

$$\underline{\lambda}(t_{i-1}) = \mathbf{N}_i^T \underline{\lambda}(t_i) + \mathbf{S}(t_{i-1}), \quad (5.29a)$$

$$\mathbf{S}(t_{i-1}) = \begin{bmatrix} \partial J / \partial\mathbf{r}(t_{i-1}) \\ \partial J / \partial\Delta\beta \end{bmatrix}, \quad (5.29b)$$

$$\frac{\partial J}{\partial\mathbf{r}(t_{i-1})} = -2\mathbf{P}^T \mathbf{W}^o \mathbf{P} \left[\mathbf{r}(t_{i-1}) - \mathbf{P}^T \{ \mathbf{y}^o(t_{i-1}) - H[\mathbf{x}^b(t_{i-1})] \} \right] \quad (5.29c)$$

$$\frac{\partial J}{\partial\Delta\beta} = 0, \quad (5.29d)$$

where $\underline{\lambda}(t_{n^t})$ equals zero. The gradient of $\Delta J / \Delta\alpha_j$ eventually becomes:

$$\frac{\Delta J}{\Delta\alpha_j} = \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^T \frac{\partial\mathbf{r}^p(t_i)}{\partial\Delta\alpha_j}, \quad (5.30a)$$

where

$$\frac{\mathbf{r}^p(t_i)}{\partial\Delta\alpha_j} = \frac{\partial\mathbf{r}^p(t_i)}{\partial\beta_j} \cdot \frac{\partial\beta_j}{\partial\alpha_j} \quad (5.30b)$$

$$\frac{\mathbf{r}^p(t_i)}{\partial\Delta\beta_j} = - \begin{bmatrix} \mathbf{N}^\beta \\ \mathbf{0} \end{bmatrix}_i \quad (5.30c)$$

$$\frac{\partial\beta_j}{\partial\alpha_j} = \mathbf{F}^T \mathbf{d}(\mathbf{z}_j \alpha_j), \quad (5.30d)$$

where $\mathbf{F}$ is a transformation matrix that maps the first derivative of the mapping by means of $\mathbf{z}_j$ of the j^{th} estimate variable on the model field.

6

Inversion of Reduced Nonlinear Incremental Models

Abstract. Despite increasing computational resources many high-dimensional applications are impractical for model inversions. In this paper two methods are presented that are promising for high-dimensional model inversion. The methods draw on POD (Proper Orthogonal Decomposition) and yield reduced models that describe a truncated behaviour of the original model. We utilize POD differently for the two methods that differ in efficiency and implementation. The first method (IGPM) applies a POD to an existing partial differential equation, the second method (ISPM) applies it to an auto-regressive formulation of a discrete model. Both method were applied to several synthetic cases and a real-world case, and by comparing them to classic inverse methodologies (i.e. the method of finite differences and the adjoint method). The two POD methods appeared to be computationally robust and more efficient for a wide range of prior estimates. Moreover, the implementation of an adjoint for the ISPM method is easy.

*This chapter is adapted from Vermeulen, P.T.M, A.W. Heemink, & C.B.M. Te Stroet, Model Inversion of Transient Nonlinear Groundwater Flow Models using Model Reduction, **Water Resources Research**, submitted 2006*

COMPUTING A MODEL that satisfies (inverts) a given observation optimally is known as *inverse modeling*. It is widely accepted that a model becomes thereby more reliable, though the model should be thoroughly reviewed to ensure the reasonableness of the results. On the other hand, there are an infinite number of objective functions (calibration criteria) to represent the error of calibration, and the computed calibration parameters are largely influenced by how this objective function is formed. Hence, the results of the minimization are not unique, and subject to the chosen objective function and the estimate variables. So, a single model inversion should be embedded into another inversion cycle that varies the calibration variables, e.g. conceptualization, model parameterizations [Poeter & Hill, 1997]. Extensive reviews of inverse models in geohydrology are given in: Carrera & Neuman [1986]; Cooley [1985]; Bennet [1992], among others. It is beyond the scope of this chapter to compare the differences between the techniques, but it is essential for all of them that the model inversions can be solved within a limited amount of time.

With regard to groundwater modeling, the time efficiency can be increased by (1) use of a more time-efficient solver [Mehl & Hill, 2001], (2) applying a coarse grid and/or locally refined grid [Mehl & Hill, 2003; Wen *et al.*, 2003] and/or (3) formulating a reduced model. This chapter describes the last item, which falls in the category of spectral methods. It is worthwhile mentioning that all methods can benefit from parallel computing. For example, in hydrology this is being exploited by Wu *et al.* [2002] and for reactive transport by Hammond *et al.* [2005]. Efforts to couple models are being designed such that they can be partitioned on different processors [Winter *et al.*, 2004].

Model reduction represents the solution to a problem as a truncated series of known basis functions and independent coefficients (Galerkin Method), thoroughly elaborated by Newman [1996a,b]. Roughly speaking, the goal of these methods is to replace the initial data by data that are optimal in terms of storage capacity. We achieve this by suppressing redundant data that exists within multi-dimensional data sets. It 'breaks up' the initial data sets into different components (i.e. the basis functions) that all together represent the original data set again. This can be obtained by multiplying the components with appropriate coefficients. The basis functions are computed by use of the Proper Orthogonal Decomposition (POD)

(also known as Principal Component Analysis (PCA), Empirical Orthogonal Functions (EOFs), or Karhunen Loève decompositions (KL), although the original concept goes back to *Pearson* [1901]). These functions represent the space of interest as 'organized' spatial features (coherent structures) that are optimal for reconstructing and modeling a signal [*Sirovich*, 1987; *Holmes et al.*, 1996]. By projecting the original partial differential equation (PDE) upon a truncated series of basis functions (Galerkin Projection), it yields a reduced model that is low dimensional because the set of basis functions is often limited. Recently this type of model was introduced in numerical groundwater flow modeling by *Vermeulen et al.* [2004a,b] (see Chapter 3). It has been applied in other fields of science [*Cazemier et al.*, 1998; *Hoffman Jørgensen & Sørensen*, 2000; *Park & Cho*, 1996]. *Park et al.* [1999] mentioned its usefulness for model inversion and implemented a reduced adjoint model for the minimization of a heat source function. *Vermeulen et al.* [2005a] implemented an approach for linear groundwater flow in which the model inversion is performed in reduced space and the results are then reconstructed to the original space to update the reduced model (dual approach), see Chapter 4. Quite similar approaches were conducted recently in optimal control design [*Atwell & King*, 2004; *Doren et al.*, 2005] and atmospheric modeling in which the minimization was done in observation space [*Courtier*, 1997]. The latter relates to the 'balancing' principle of systems as it truncates spaces that have no influence on observations [*Lee et al.*, 2000; *Newman & Krishnaprasad*, 1998].

To make the reduced model even more time efficient, an 'incremental' approach can be used in which the original nonlinear model equation is replaced by a reduced linear incremental system [*Courtier et al.*, 1994; *Lawless et al.*, 2005; *Vermeulen & Heemink*, 2006a]. With this system we are able to find the minimum of an objective function by interchanging only the estimate variables within the reduced model. This chapter describes two different implementations of these incremental approaches for groundwater models (finite differences).

The chapter starts by overviewing the classical methodologies for inverse modeling. Subsection 6.2 is devoted to the reduced model methodologies whereby two different implementations of the POD are evaluated. In Subsection 6.3 a note is given on their computational efficiency. In Subsection 6.4 several synthetic cases are used to express the performance of the methodologies in terms of efficiency and robustness. The subsection is followed by the results obtained by a real-world case in Subsection 6.5. Subsection 6.6 gives conclusions and recommendations.

6.1 Classic Methodology

6.1.1 Forward Model

The algorithms in this chapter are elaborated for three-dimensional groundwater flow which can be described by:

$$\sum_{i=1}^3 \frac{\partial}{\partial x_i} \left[C^{\mathbf{x}_i}(\alpha_j, x) \frac{\partial \phi(x)}{\partial x_i} \right] - C^f(x, \phi) \phi(x) - q(x) - C^f(x, \phi) h_t(x) = -S(x) \frac{\partial \phi(x)}{\partial t}, \quad (6.1)$$

where ϕ is the hydraulic head [L] at location x in the computational domain $x \in \Omega$, $C^{\mathbf{x}_i}$ is the hydraulic conductance [$L^2 T^{-1}$] aligned to x_i which depends on an estimate variable α_j (Appendix 6A), S is the storage coefficient [–], t is the time [T], q is a specific source term [LT^{-1}] for a portion of the model domain $x \in \partial\Omega^q$, and C^f [T^{-1}] is the total sum of linear and/or nonlinear specific conductances for a portion of the model domain $x \in \partial\Omega^f$ that depends on $\phi(x)$ and $h^f(x)$ where h^f is a specific reference level [L].

The PDE (Equation 6.1) can be solved by means of a finite-difference discretization for a mesh of grid cells n^m in space and time with boundary conditions $\phi^d(x)$; $x \in \partial\Omega^d$ as done in *McDonald & Harbaugh* [1988]. This discretized model can be written in matrix form as

$$\mathbf{A}[\alpha] \phi(t_i) = \mathbf{b}(t_i), \quad (6.2a)$$

with

$$\mathbf{A}[\alpha] = \mathbf{C}[\alpha] - \frac{1}{\Delta t} \mathbf{S} - [\mathbf{C}^f]^\iota(t_i), \quad (6.2b)$$

$$\mathbf{b}(t_i) = -\frac{1}{\Delta t} \mathbf{S} \phi(t_{i-1}) + \mathbf{q}(t_i) - [\mathbf{C}^f]^\iota \mathbf{h}^f(t_i), \quad (6.2c)$$

where $\phi \in \mathbb{R}^{n^m}$ is a vector of the nodal hydraulic heads, $\mathbf{C}$ is a heptadiagonal coefficient matrix that contains the hydraulic conductances C at the nodes of a spatial network, $\mathbf{S}$ and $[\mathbf{C}^f]^\iota$ are both diagonal matrices for the elements of the network in which the latter depends in case of nonlinearity with respect to ϕ on an internal iteration cycle ι (i.e. a Picard iteration), $\mathbf{q}$ and $\mathbf{h}^f$ are vectors for source/sink terms and reference levels, respectively.

6.1.2 Inverse Model

The objective function value J describes the weighed sum of squared residuals:

$$J(\alpha_j)^\eta = \sum_{i=1}^{n^t} \left(\{\mathbf{y}^o(t_i) - H[\phi(t_i)]\}^T \mathbf{W}^o \{\mathbf{y}^o(t_i) - H[\phi(t_i)]\} \right), \quad (6.3)$$

where J is the objective function value for the η^{th} iteration subject to the j^{th} estimate variable α_j , $\mathbf{W}^o$ is the observational weight matrix, $\mathbf{y}^o(t_i)$ are the observations

for the i^{th} time step out of n^t , and H is an operator that maps the model field to observation space. To minimize J , two classic methods are described concisely and implemented for purposes of comparison with the reduced model methodologies (see Subsection 6.2).

- The finite-difference method (FDM) approximates the first derivative of J with respect to the variable α_j [Cooley, 1985], see Equation (5.4). The method is easy to implement, but a disadvantage is that its efficiency decreases with the number of α s, see Figure 6.1a.
- A more sophisticated and time-efficient method computes the exact gradient of J with respect to all α s, simultaneously. This method is known as the adjoint method (ADJ) or 4D-variational method [Courant & Hilbert, 1953; Townley & Wilson, 1985], so:

$$\left[\frac{\Delta J}{\Delta \alpha_j} \right]^\eta = \sum_{i=1}^{n^t} \left\{ \lambda^T(t_i) \left(\frac{\partial \mathbf{A}}{\partial \alpha_j} \right) \phi(t_i) \right\}, \quad (6.4)$$

where $\lambda(t_i)$ is called the adjoint model (concisely derived in Appendix 6B). For each gradient computation it needs to solve the forward model (Equation 6.2a) and the adjoint model backwards in time (Equation 6.23), and is independent of the number of α s, see Figure 6.1b. However, a disadvantage is that the adjoint model is often hard to implement, especially when the forward model contains nonlinearities.

The classic inversion methods put a lot of effort into a single gradient estimation η that vanishes at the next iteration $\eta+1$. The proposed methodology in this chapter puts this effort into model reduction.

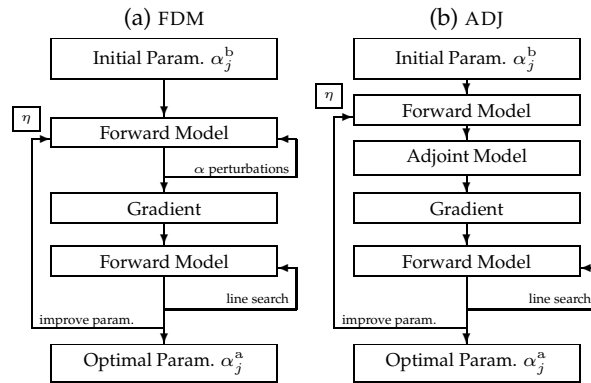


Figure 6.1: Flow chart of classic inversion methods; (a) the finite-difference method (FDM), and (b) the adjoint method (ADJ). The number of gradient computations is expressed by η .

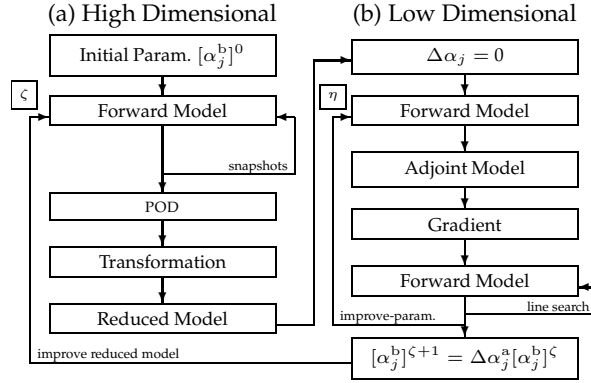


Figure 6.2: Flow chart of the reduced model methodology that exists of (a) a high-dimensional part to compute PODs that transform the original model into a reduced model and (b) a low-dimensional part used for the model inversion.

6.2 Reduced Model Methodology

6.2.1 Introduction

The Galerkin method is a discretization scheme for PDEs that is based upon the assumption that the spatial distribution of the original state vector ($\phi \in \mathbb{R}^{n^m}$) is very complex in contrast to its behavior in time. For these situations, an approximate state $\hat{\phi}$ [L] can be computed by separating the original state into a spatial and time-variant component:

$$\hat{\phi}(x, \alpha^b + \Delta\alpha, t_i) = \phi^b(x, \alpha^b, t_i) + \mathbf{P}(x)\mathbf{r}(\Delta\alpha, t_i), \quad (6.5)$$

where $\phi^b(x, \alpha^b, t_i)$ [L] is the background state as a function of the background estimates α , $\mathbf{r}$ [L] is the reduced state equivalent of ϕ , and $\mathbf{P}$ [–] is a truncated set of spatial basis vectors (i.e. the projection matrix) that span a subspace $\mathcal{S}$ of the original state space $\mathcal{X}$, see for elaboration Subsection 3.1.3 & 4.1.2. In this chapter we derive different ‘incremental’ nonlinear models to simulate the coefficients $\mathbf{r}$ as a function of a perturbation of the estimate variables $\Delta\alpha$. We apply a transformation of the original set of equations by use of the PODs that yield an ordinary differential equation (ODE) for $\mathbf{r}$ (referred to as ‘reduced model’), see Figure 6.2a. These models can then be used, at low computational costs, to optimize a set of estimate variables by minimizing an approximate objective function $\hat{J}$:

$$\hat{J}(\Delta\alpha_j^a)^\zeta = \sum_{i=1}^{n^t} \left[\mathbf{P}^T \left\{ \mathbf{y}^o(t_i) - H[\phi^b(t_i)] \right\} - \mathbf{r}(\Delta\alpha_j^a, t_i) \right]^T \mathbf{P}^T \mathbf{W}^o \mathbf{P} \left[\mathbf{P}^T \left\{ \mathbf{y}^o(t_i) - H[\phi^b(t_i)] \right\} - \mathbf{r}(\Delta\alpha_j^a, t_i) \right], \quad (6.6)$$

where ζ is the outer iteration index. An advantage is that the minimization of $\hat{J}$ with respect to $\Delta\alpha_j^a$ can be done only with the reduced model (Figure 6.2b) and

consumes a negligible amount of computation time as $\mathbf{r} \in \mathbb{R}^{n^p}$ with $n^p \ll n^m$. It yields an optimal set of posterior estimates $[\Delta\alpha_j^a]^\eta$ for the current η^{th} reduced model that can still be sub optimal for the original model. Therefore, the outer loop $\eta + 1$ is entered again with $[\alpha_j^b]^{\eta+1} = [\Delta\alpha_j^a\alpha_j^b]^\eta$ which yields an updated set of snapshots and a different reduced model. This iteration cycle continues until a given termination criterion γ is met for the innovation of n^u estimate variables, thus

$$\sum_{j=1}^{n^u} \|[\alpha_j^b]^{\eta+1} - [\alpha_j^b]^\eta\| \leq \gamma. \quad (6.7)$$

We describe two different formulations based upon (1) a transformation of the governing PDEs (Incremental Galerkin Projection Method IGPM), and (2) a transformation of an autoregressive representation of the governing system of PDEs (Incremental Sequential Projection Method ISPM). The last method is extensively elaborated in Chapter 5. The two methods differ in their forward and backward (adjoint) model formulations and above all, their snapshots requirements.

6.2.2 Incremental Galerkin Projection Method

Snapshots

In previous publications [Vermeulen *et al.*, 2004a,b, 2005a] effort was put into the computation of snapshots that unravel the model behavior for specific boundary conditions subject to α_j^b . For simple linear models this can be advantageous caused by the superposition principle (see grey-box on page 58). However, for nonlinear models this principle is not valid and another method that defines a snapshot distortion vector α_i^e is needed. This distortion vector affects the parameter T in the original model and yields a snapshot series $\mathbf{E}[\alpha_i^e](t_i)$; $i \in \{1, \dots, n^t\}$ for the entire time domain. After that, another vector α_{i+1}^e is defined that yields a different snapshot series (single estimation), see Figure 6.3a. From a mathematical point of view, the subsequent vectors α_i^e should be uncorrelated, and a method that can resolve this is the *Latin HyperCube Sampling* (LHS) [Iman & Shortencarier, 1984], see Subsection 4.2.2 on page 96. The PDF used in this chapter is based upon a log-normal distribution of α_j^b characterized by a log-transformed deviation factor $\exp(\delta_j)$. Large values for δ_j will increase the subspace $\mathcal{S}$ and yield a reduced model that is less accurate but gives a broad averaged view of $\hat{J}$. Tiny values for δ_j , however, de-

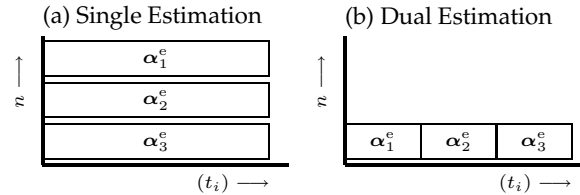


Figure 6.3: (a) Single and (b) dual estimation methodology for the computation of n -snapshot simulations for the time domain (t_i) ; $i \in \{1, \dots, n^t\}$.

crease the subspace and compute an accurate $\hat{J}$ for a limited parameter space. In practice, the robustness of the method is improved when δ_j is sequentially evolved as $[\delta_j]^{\eta+1} < [\delta_j]^\eta$.

Forward Model

The linearization of the high-order model is performed by including an extra term that expresses the dependency of the system dynamics on α_j^b (elaborated in Appendix 6C). Linearization needs to be performed for a certain distortion vector α^b that yields a specific state, which we call the background state $\phi^b(\alpha^b)$ (i.e. a fixed point approximation). This is the normal procedure and our adjustment is that we interchange this background state recurrently by the current state ϕ to reflect the current distortion vector α^a . This expression is obviously superfluous to the original model, but it yields eventually an incremental reduced model with respect to $\Delta\alpha_j$. So, including the extra term, substituting the state vector with the Galerkin method (Equation 6.5), and applying a finite-difference approximation in space and time, Equation (6.1) becomes

$$\sum_{i=1}^3 \left\{ \underbrace{\frac{\partial}{\partial x_i} \left[\mathbf{C}^{x_i}([\alpha^b]^\eta) \frac{\partial \mathbf{P}}{\partial x_i} \right]}_{\mathbf{n}_i} + \sum_{j=1}^{n_p} \underbrace{\frac{\partial}{\partial x_i} \left[\frac{\partial \mathbf{C}^{x_i}([\alpha^b]^\eta)}{\partial \tilde{\alpha}_j} \frac{\partial \mathbf{P}}{\partial x_i} \right]}_{\mathbf{u}_{ij}} \Delta\alpha_j \right\} \mathbf{r}(t_i) - \frac{1}{\Delta t} \mathbf{SPr}(t_i) - [\mathbf{C}^f]^\iota \mathbf{Pr}(t_i) = -\frac{1}{\Delta t} \mathbf{SPr}(t_{i-1}) + \mathbf{q}(t_i) - [\mathbf{C}^f]^\iota \mathbf{h}^f(t_i), \quad (6.8)$$

where $\tilde{\alpha}_i$ is the interval for which the partial differential equation was linearized (see Figure 5.2). The expression can be simplified by computing the second-order differentials of the pattern derivative of space and store the results in $\mathbf{N} = \{\mathbf{n}_1, \dots, \mathbf{n}_n\}$ and $\mathbf{U}_j = \{\mathbf{u}_{1j}, \dots, \mathbf{u}_{nj}\}$. More elaboration can be found in Subsection 4.1.2 on page 93 and the Equations 3.28a-c on page 66). Each term in Equation (6.8) can now be projected by multiplication with the basis vectors $\mathbf{P}^T$, yielding their reduced equivalents

$$\underline{\mathbf{N}} = \mathbf{P}^T \mathbf{N}, \quad (6.9a)$$

$$\underline{\mathbf{U}}_j = \mathbf{P}^T \mathbf{U}_j, \quad (6.9b)$$

$$\underline{\mathbf{S}} = \mathbf{P}^T \mathbf{SP}, \quad (6.9c)$$

$$\left(\underline{\mathbf{C}}^f \right)^\iota = \mathbf{P}^T (\mathbf{C}^f)^\iota \mathbf{P}, \quad (6.9d)$$

$$\underline{\mathbf{q}}(t_i) = \mathbf{P}^T \mathbf{q}(t_i), \quad (6.9e)$$

$$\underline{\mathbf{h}}^f(t_i) = \mathbf{P}^T \mathbf{h}^f(t_i). \quad (6.9f)$$

It yields the following formulation of the IGPM:

$$\left[\mathbf{N} + \sum_{j=1}^{n^p} \mathbf{U}_j \Delta \alpha_j - \frac{1}{\Delta t} \mathbf{S} - (\mathbf{C}^f)^t \right] \mathbf{r}(t_i) = -\frac{1}{\Delta t} \mathbf{S} \mathbf{r}(t_{i-1}) + \mathbf{q}(t_i) - (\mathbf{C}^f)^t \mathbf{h}^f(t_i), \quad (6.10)$$

for which the model is initialised by computing $\mathbf{r}(t_0) = \mathbf{P}^T \phi(t_0)$ and defining a prior incremental estimate $\Delta \alpha_j^b$. The model is nonlinear for the matrix $\mathbf{C}^f$, which needs to be evaluated iteratively in the original dimensions and projected on the basis vectors repeatedly (Equation 6.9d). However, this computation needs to be done only for that portion of the model domain $x \in \Omega^f$. For large-scale models however, it is doubtful whether this iteration is necessary, because of the projection these slight adjustments within $\mathbf{C}^f$ will have a limited influence upon $\hat{J}$. This is a heuristic assumption that increases the efficiency of a reduced model even more. A final remark is that a Dirichlet condition (i.e. a ‘constant head’ cell) should be modeled as a linear boundary condition with a high conductance C^f such that $\phi(t_i) \approx \phi^d$.

Adjoint Model

An implementation of the adjoint for the reduced model is comparable to that of the original model and yields

$$\frac{\Delta \hat{J}_j}{\Delta \alpha_j} = \sum_{i=1} [\underline{\lambda}(t_i)]^T \mathbf{U}_j \mathbf{r}(t_i), \quad (6.11)$$

where $\underline{\lambda}(t_i)$ $[-]$ is the reduced adjoint model (elaborated in Appendix 6D) and should be computed backwards in time to represent the reduced system dynamics with respect to the partial derivative $\partial \hat{J} / \partial \mathbf{r}(t_i)$. To ensure that $\mathbf{C}^{x_i} + (\partial \mathbf{C}^{x_i} / \partial \tilde{\alpha}_j^b) \Delta \alpha_j$ remains positive during the optimization procedure, $\Delta \alpha_j$ is log-transformed. As a consequence, the matrix $\mathbf{U}_j$ needs to be recomputed now for each perturbation of α_j , which will decrease the overall efficiency of the method. We have avoided this by linearizing the partial differential equation for α_j^b . Due to this, the adjoint gradient will become less accurate beyond this ‘fixed point’ and the matrix $\mathbf{U}_j^f$ should be updated during the minimization. A simple procedure is to recompute only this matrix when no further improvement of the objective function is achieved (i.e. the ξ iteration). From the synthetic examples, however, the matrix $\mathbf{U}_j$ was still accurate for a reasonable parameter space.

6.2.3 Incremental Sequential Projection Method

Governing Equation

The method is extensively elaborated in Subsection 5.2 for a generic application. It writes the original PDE (Equation 6.1) as an autoregressive model with an extra term which linearizes the nonlinear model with respect to the estimate variable α_j :

$$\phi(t_i) = M_i[\alpha^b, \phi(t_{i-1})] + \sum_{j=1}^{n_u} \frac{\partial M_i[\alpha^b, \phi(t_{i-1})]}{\partial \tilde{\alpha}_j} \Delta \alpha_j, \quad (6.12)$$

where M_i is the system dynamics operator, and a function of the background estimate α^b . It is defined as:

$$M_i[\alpha^b, \phi(t_{i-1})] = [\mathbf{A}_i^p(\alpha^b)]^{-1} \mathbf{b}^p(t_i), \quad (6.13)$$

where $\mathbf{A}^p$ and $\mathbf{b}^p$ could represent a collection of system matrices that are influenced by a collection of different (non)linear boundary conditions, e.g. a coupled surface-water model:

$$\mathbf{A}^p = \begin{bmatrix} \mathbf{A}_i(\alpha^b) & 0 & 0 \\ 0 & \mathbf{A}_{i+1}(\alpha^b) & 0 \\ 0 & 0 & \mathbf{A}_n(\alpha^b) \end{bmatrix} \quad (6.14a)$$

$$\mathbf{b}^p(t_i) = \begin{bmatrix} \mathbf{b}_i \\ \mathbf{b}_{i+1} \\ \mathbf{b}_n \end{bmatrix} (t_i). \quad (6.14b)$$

Thus, the time-variant dynamics operator M_i which we are looking for represents the system dynamics **and** the boundary conditions and source/sink terms within the entire set of n models. It will be a full matrix with dimension $\sum_{i=1}^n [n^m]_i^2$ that will be impossible to compute for realistic problems. However, in reduced space this matrix can be obtained fairly easily as shown in the following subsection.

6.2.4 Snapshots

The usage of snapshots here is twofold; (1) they are needed to describe the model behavior, and (2) they represent $\partial M_i[\phi(t_{i-1}), \alpha^b] / \partial \tilde{\alpha}_j$. The latter can be obtained by the classic method of finite differences (Equation 5.4) which sequentially samples each estimate variable α_j . As mentioned earlier, however, it is wise to perturb all variables simultaneously (see Subsection 6.2.2). Hence, we introduced in Subsection 5.2.4 the partial differential $\partial M_i[\phi(t_{i-1}), \alpha^b] / \partial \tilde{\beta}_j$, and instead of perturbing along the vectors that map the estimate variables on the model field ($\mathbf{z}_j \Delta \tilde{\alpha}_j$), we perturb along $\mathbf{f}_j \Delta \tilde{\beta}_j$:

$$\mathbf{e}_j = H \left[\frac{\partial M_i[\phi(t_{i-1}), \alpha^b]}{\partial \tilde{\beta}_j} \right] \approx H \left[\frac{M_i[\phi(t_{i-1}), \alpha_j^b + \Delta \tilde{\beta}_j \mathbf{f}_j] - M_i[\phi(t_{i-1}), \alpha_j^b]}{\Delta \tilde{\beta}_j} \right], \quad (6.15)$$

where $\Delta\tilde{\beta}$ is defined in Equation 5.11, and the vector $\mathbf{f}_j$ is the j^{th} vector of an orthonormal matrix $\mathbf{F}$, and $M_i[\phi(t_{i-1}), \alpha_j^b + \Delta\tilde{\beta}_j \mathbf{f}_j]$ is elaborated in Appendix 6E. From a mathematical point of view, $\Delta\tilde{\beta}_j$ should be chosen with care as it can influence the diagonal dominance of the operator M_i . Most iterative solvers should satisfy that condition, which can potentially be satisfied by *splitting* the system matrices into a diagonal dominant part and a remaining part [Edwards, 2000]. Finally, the snapshot vectors $\mathbf{e}_j$ will contain only, after an operation with H , the results for the locations of observation n^o . An advantage is that the pattern identification (Subsection 3.1.3) is efficient because the snapshot vectors are low dimensional $\mathbb{R}^{n^o}$. Moreover, the snapshot vectors describe less variance yielding fewer patterns, compared to those that were needed to describe the variance for the entire model domain. However, the patterns $\mathbf{P}$ should contain n^m elements to proceed with the next subsection, though they are nonzero only for the locations of the observations.

6.2.5 Forward Model

The autoregressive model (Equation 6.12) can be projected by $\mathbf{P}^T$ and notated as a partitioned matrix yielding

$$\begin{bmatrix} \mathbf{r} \\ \Delta\beta \end{bmatrix} (t_i) = \begin{bmatrix} \mathbf{N} & \mathbf{N}^\beta \\ \mathbf{0} & \mathbf{I} \end{bmatrix}_i \begin{bmatrix} \mathbf{r} \\ \Delta\beta \end{bmatrix} (t_{i-1}), \quad (6.16a)$$

where

$$\mathbf{N}_i^\beta = \mathbf{P}^T \{ \mathbf{e}_1, \dots, \mathbf{e}_{n^u} \}, \quad (6.16b)$$

$$\mathbf{N}_i = \mathbf{P}^T H \left[\frac{\partial M_i[\alpha^b]}{\partial \phi^b(t_{i-1})} \right] \mathbf{P}. \quad (6.16c)$$

The operator $\mathbf{N}_i$ can be rewritten as:

$$\mathbf{N}_i = \mathbf{P}^T \left\{ H \left[\frac{\partial M_i[\alpha^b]}{\partial \phi^b(t_{i-1})} \mathbf{p}_1 \right], \dots, H \left[\frac{\partial M_i[\alpha^b]}{\partial \phi^b(t_{i-1})} \mathbf{p}_{n^p} \right] \right\}. \quad (6.17)$$

This has been extensively elaborated in Subsection 5.2.5. The reduced model is eventually initialised by $\mathbf{r}(t_0) = 0$ and a projected prior estimate $\Delta\beta \neq 0$. Because the formulation of the model is autoregressive, it lacks the need for any Picard iteration ι that emerges for nonlinear boundary conditions.

6.2.6 Adjoint Model

An important advantage of the ISPM is that an implementation of the adjoint is relatively simple (elaborated in Appendix 5A) and yields the exact gradient

$$\frac{\Delta \hat{J}_j}{\Delta \alpha_j} = \sum_{i=1} [\lambda(t_i)]^T - \begin{bmatrix} \mathbf{N}^\beta \\ \mathbf{0} \end{bmatrix}_i \mathbf{F}^T \mathbf{z}_j d\alpha_j. \quad (6.18)$$

Since the reduced model operates with $\Delta\beta_j$ s instead of $\Delta\alpha_j$ s, it is necessary to project $\mathbf{z}_j d\alpha_j$ onto $\mathbf{F}$. Furthermore, an improved fit between J and $\hat{J}$ can be obtained by applying an appropriate scaling for the estimate variables. Because the estimate variable α_j is log-related to the observation, the appropriate scaling here is a log-transformation such that $\log(\Delta\tilde{\alpha}_j) \mapsto \Delta\tilde{\beta}_j$ (Equation 5.11).

6.3 Computational Efficiency

The primary factors that determine the efficiency of the IGPM & ISPM are quantified by the number of original model simulations τ which is influenced by the

- *Snapshots*

- The maximum number of snapshot simulations n^e is equal to $n^u \times n^t$. However, it is difficult to mention the minimum number because there is a trade-off between the effort put in the snapshot simulations and the ‘ $\hat{J}$ -reduction capacity’ of the evolved reduced model. For example, it is possible to start with half the maximum number of simulations such that the minimization procedure will focus itself on the main directions first, and add more snapshots as the process proceeds.
- More efficiency can be obtained by interchanging α_i^e during the model simulation (dual estimation), see Figure 6.3b. Unfortunately, this is not possible for the ISPM because all snapshots are needed for the computation of $\partial M_i[\phi(t_{i-1}), \alpha^b]/\partial \tilde{\beta}_j$ (Equation 6.15).
- Especially, for the ISPM it is more time efficient to avoid the time dependency of matrix $\mathbf{N}_i$ (Equation 6.17). This can easily be done by applying a constant time-discretization $\Delta t(t_i); i \in \{1, \dots, n^t\} = c$ and linearizing the nonlinear conductance matrix $\mathbf{C}^f$ for the initial state $\phi^b[\alpha^b](t_0)$. The error that arises will affect the objective function, however, it is mainly averaged away after projection.

- *Outer Iterations*

The number of outer iterations is determined by the chosen termination criterion γ (Equation 6.7). It should be chosen not too small ($\gamma \geq 0.1$), as this causes jumping of sub-optimal posterior estimates around the global optimal solution. Since the reduced model probably overestimates α_j^t caused by the applied linearization, it probably yields a redundancy of η whenever a threshold ψ_j is applied. It prevents the model from updating the variables to extraordinary values, far beyond background estimation α_j^b that was used for the linearization.

6.4 Examples

6.4.1 Introduction

This section illustrates the IGPM & ISPM by means of twin experiments in which observations are generated from a run of the original model defined to be the ‘truth.’ We have used a synthetic 2D model to generate a set of observations (first experiment). This model consisted of a network of 7 columns by 7 rows with $\Delta x \equiv \Delta y = 10$ m, $\mathbf{T} = 100 \text{ m}^2 \text{ day}^{-1}$, and $\mathbf{S} = 0.27$ for the entire model domain. The left side of the model was determined by Dirichlet conditions ($\phi^d = 0$ m). A drainage element that behaved nonlinearly was active along the entire fourth column for which the flux q^f was computed as

$$q^f(x, \phi) = \begin{cases} \phi(x) < h^f(x) & ; 0 \\ \phi(x) \geq h^f(x) & ; C^f[\phi(x) - h^f(x)] \end{cases}, \quad (6.19)$$

where h^f is a drainage level [L] ($h^f(t_i); i \in \{1, \dots, n^t\} = -1$ m), and C^f is the drainage conductance ($\mathbf{C}^f = 10 \text{ m}^2 \text{ day}^{-1}$). The system was simulated for 10 time steps with $\Delta t(t_i); i \in \{1, \dots, n^t\} = 10$ days. We have applied the IGPM & ISPM ($\varphi^e = 99.99\%$) to this problem in order to minimize $\hat{J}$ from an incorrect prior estimate (second experiment). For all cases the observational weight matrix $\mathbf{W}^o = \mathbf{I}$. We used the Quasi-Newton method to proceed during the minimization [Press *et al.*, 1992], and we aborted the outer iteration cycle η by $\gamma = 0.1$. For purposes of comparison, we have also implemented the FDM ($\Delta \tilde{\alpha}_j = 0.01$, see Equation (5.4)) and the ADJ method. Unless stated differently, the number of evaluations of the original model τ is used to compare the different methods. This included the number of evaluations that were needed for the necessary *line searches*.

6.4.2 Case I

Configuration

For this particular case, we defined two observation wells y_i^o and two pumping wells q_i that varied randomly in strength such that $-2 < q_i(t_i); i \in \{1, \dots, n^t\} < 2 \text{ mday}^{-1}$. Two zones were defined for which the prior estimates were $\alpha_1^0 = 0.1, \alpha_2^0 = 10$, see Figure 5.7a.

Results

The ADJ method minimized J within $\eta = 9$ gradient iterations and consumed in total $\tau = 30$ original model simulations, see Figure 6.4a. The surface of $\hat{J}$, as computed by IGPM (Figure 6.4b1) resembles the surface of J near the locations for which the snapshots were computed (denoted by the black stars). The inaccuracy increases further away because the snapshots and the applied linearization subject to α_j^b , are both representative for a limited space (approximated by the grey circle with radius ψ). Nevertheless, a sub-optimal minimum was found (after another ξ iteration denoted by the term ‘Proj.’ in Figure 6.4b1). An additional second outer iteration η was eventually carried out to obtain the closest fit between apriori es-

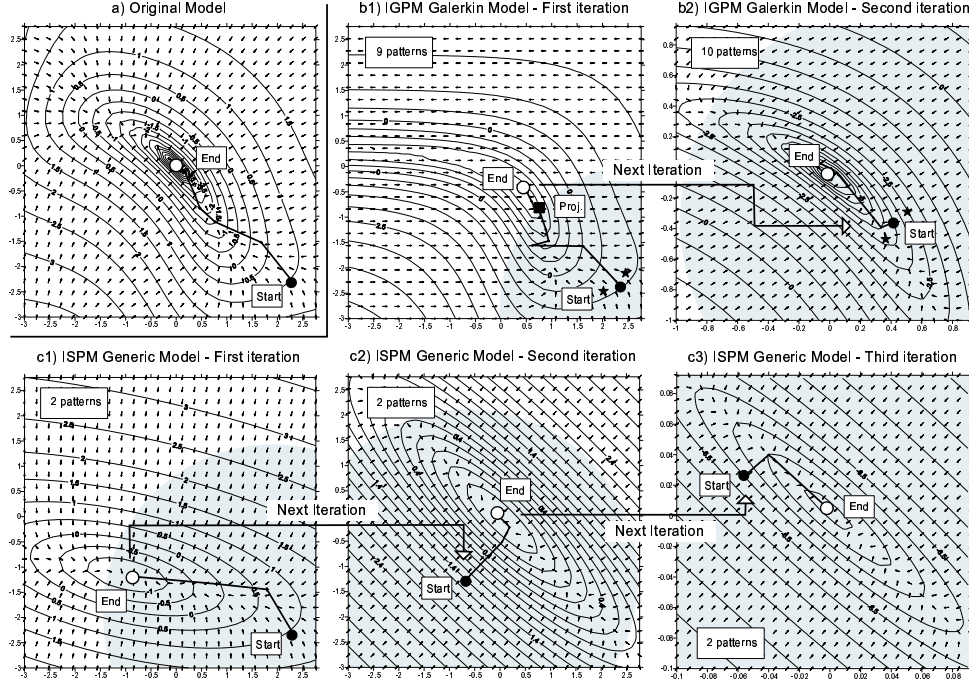


Figure 6.4: The surface of $\log(J)$ and the accompanying gradient (arrows) for the minimization of two prior estimates ($\log(\alpha_1)$ and $\log(\alpha_2)$ presented on the y-axis and x-axis, respectively) that starts at the $\bullet$ and ends in the $\circ$. An ADJ for the original model is depicted in (a), in (b1-2) the results from a IGPM minimization are plotted, and in (c1-3) the ISPM performance is illustrated. The grey-area within the figures (b1-2) and (c1-3) depict the domain of the reduced model in estimation space (ψ), and the black stars $\star$ represent the chosen combinations of variables within the snapshots.

timated and the ‘truth,’ see Figure 6.4b1-2. The surface of $\hat{J}$ for the ISPM methods looks quite different compared to the others (Figure 6.4c1) because it evolves from a linear model for which α_j was log-transformed before the projection. Therefore, the $\log \hat{J}$ yields a perfect quadratic function for which the innovation of the variables needs to be restricted (grey area with radius ψ) because the model tends to overestimate the variables. So, instead of approaching the minimum mainly from one direction, it approaches the ‘truth’ by decreasingly *jumping* around it from side to side. Hence, it needs more outer iterations η to find the optimal solution, see Figure 6.4c2–3.

Because the IGPM needs to describe the variance on the locations of observations **and** on the locations where a boundary condition exists, it results in a ten-dimensional model. This is still significantly less than the original dimensions, but the ISPM method needs only two dimensions as it describes the estimate variables subject to the observational variance that exists within two observations. Above all, both methods IGPM and ISPM used $\tau = 4$ and $\tau = 9$ original simulations respectively, which is less than the original ADJ that consumed $\tau = 30$.

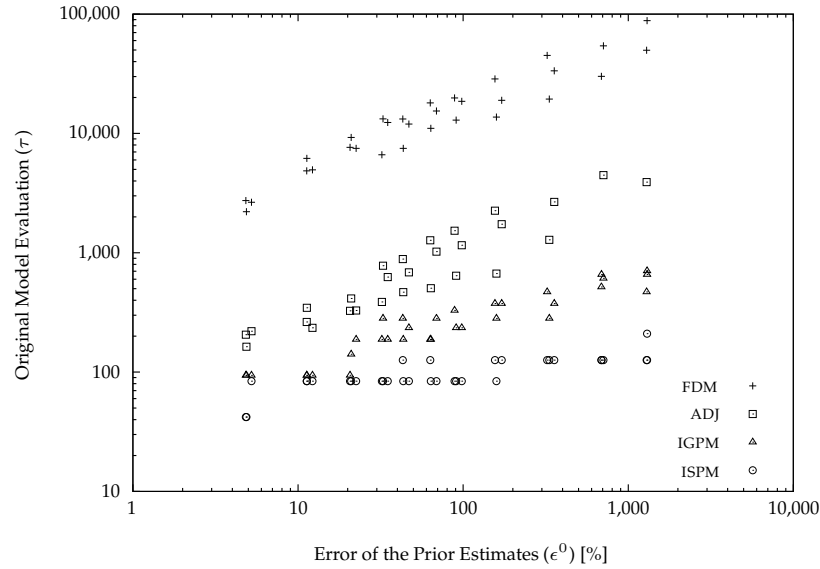


Figure 6.5: Minimization efficiencies for different methodologies (FDM, ADJ, IGPM & ISPM), expressed in the number of model evaluation τ , for 33 samples of 42 incorrect prior estimates.

6.4.3 Case II

Introduction

To obtain insight in the efficiency and robustness (i.e., to be able to find the correct posterior estimates) of the methods, 33 samples of α_j^0 were generated by means of the LHS method [Iman & Shortencarier, 1984] from which J and $\hat{J}$ were minimized. The quality of these minimizations were determined by the relative error ε between the prior estimate and the posterior estimate (Equation 5.24). Since the ‘truth’ is $\|\alpha^a\| = 1$, the equivalent ε^0 expresses the error of the prior estimates.

Configuration

The identical model configuration as described for Case I was extended with 42 observations throughout the model, see Figure 5.7b.

Results

The robustness (the ability to find the correct posterior estimates) of the methods is mutually comparable for slight errors of the prior estimate, see Figure 6.5. The IGPM and ISPM appear to be more robust for increased values for ε^0 because they compute a ‘rough’ surface of the objective function that makes them less sensitive to local minima. They are based on a statistically linearized model that tends to produce a spatially averaged gradient. For all methods τ increases with ε^0 and both the IGPM & ISPM need less τ compared to the FDM and ADJ methods. The IGPM is approximately two times more efficient than the ISPM because the latter needs additional model evaluations to determine the coefficients of the system dynamics.

These results are specific for the synthetic case considered, and will be influenced in practice by a number of factors (e.g. the number of nodes, the number of estimate variables, the number of observations and the complexity of the nonlinearity inside the system). It is reasonable to suggest that all methods will be equally influenced by these factors and that the IGPM & ISPM remain their efficiencies for other cases.

6.5 Real-world Case

6.5.1 Configuration

This case has a realistic configuration that describes the entire region of the province of Noord-Brabant in the Netherlands ($\approx 10,700 \text{ km}^2$). The model consist of 64 rows, 107 columns and 9 model layers, so the total number of nodes is 61,632. The first model layer contains the influence of an intense surface water network that consists of wet-lands and natural dewatering systems situated in higher areas. Absolute levels for these nonlinear boundary conditions were obtained by accurate laser altimetry. The model is further characterized by a detailed description of the precipitation and evapotranspiration rate. These were obtained by combining rain gauge stations throughout the model domain with accurate land use classification from satellite images.

6.5.2 Results

We have minimized an objective function with respect to a change in transmissivity within each model layer (Equation 6.20). It was based upon 234 synthetic measurements distributed throughout the model and observed for 36 time steps. For the various minimization techniques as described in this chapter, the evolution of the objective function value for this minimization is depicted in Figure 6.6. All methodologies were able to find the correct values for the estimate variables but they differ significantly in efficiency. The results found here, for a realistic case, are similar to the results described for the synthetic cases in the previous subsections. That is: most efficient is the IGPM method; it is almost twice as efficient as the ISPM method. The IGPM method is less efficient than the ADJ method. However, the former is easy to program and twice as efficient as the FDM method.

6.6 Conclusions

In this chapter two different methodologies have been described for solving nonlinear inverse modeling problems. They were derived by a linear transformation, known as a ‘change of coordinates’, that projects a system of equations from a high dimension into a low dimension described by patterns (POD). These PODs were based upon several evaluations of the original model that provided insight in the observational behavior, subject to the chosen estimate variables. By truncation of unimportant patterns, which represented a tiny amount of model behavior, it yielded a low-dimensional model. The first model was based upon a projection of

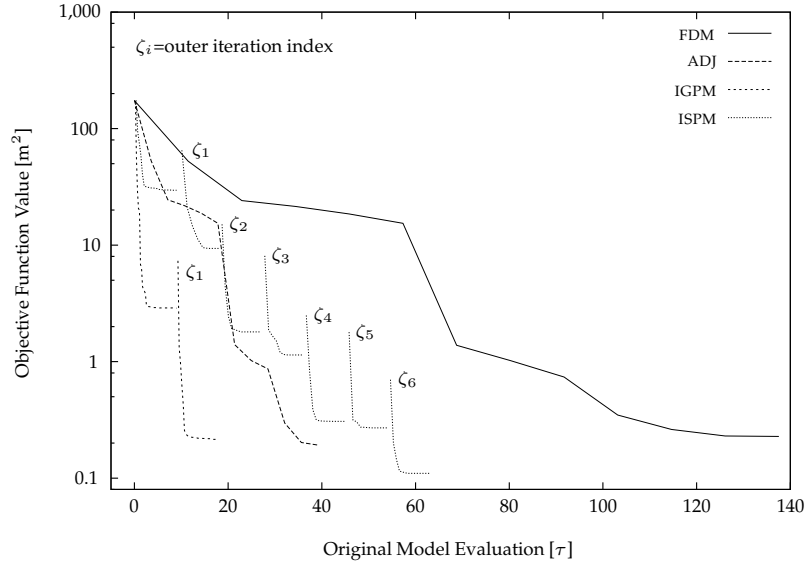


Figure 6.6: Evolution of the objective function value subject to nine estimate variables for different methodologies (FDM, ADJ, IGPM & ISPM) versus the number of model evaluation τ .

the original PDE (IGPM method) and the second model resulted from a projection of an autoregressive representation of the original system (ISPM). Both methods were applied to several synthetic cases and they appeared to be more computationally efficient than the classic inverse methodologies (method of finite differences and the adjoint method). The advantage of the IGPM method is that it conserves the system dynamics in reduced space and is therefore more accurate, from a mathematical point of view. On the other hand this can be disadvantageous too, as it includes some analytical derivatives that are inconvenient to program and nonlinear aspects that causes an extra inner-iteration loop. In this perspective the ISPM is more flexible and faster because it can handle these nonlinearities more effectively. Above all, both methods operate in fewer dimensions than the original model and have the ability to become more robust than the classic inverse methodologies, because they focus on the main directions in observational space and tend to become less sensitive to local distortions of the objective function. This should be examined in future.

6A Hydraulic Conductance

In Equation (6.1) the hydraulic conductance is introduced that is aligned with a direction x_i . For a location $x \in \Omega$ this can be computed as

$$C^{x_i}(x) = [\Delta w_i(x) + \Delta w_i(x^*)] \left[\frac{\Delta w_i(x)}{\alpha_j T(x)} + \frac{\Delta w_i(x^*)}{\alpha_j T(x^*)} \right]^{-1}, \quad (6.20)$$

where Δw_i [L] is the width of the grid cell along the i^{th} direction, T [L²T⁻¹] is the transmissivity, and $(\cdot)^*$ denotes the neighbor grid cells along the current direction i .

6B Derivation of the Original Adjoint Model

Equation (6.4) introduces the adjoint model $\lambda(t_i)$ that can be concisely elaborated by stating that:

$$J = J^\bullet + \sum_{i=1}^{n^t} [\lambda(t_i)]^T [\mathbf{A}\phi(t_i) - \mathbf{b}(t_i)], \quad (6.21)$$

where $J^\bullet \equiv J$. By applying a Taylor's series expansion for the variables $\phi(t_i)$, $\phi(t_{i-1})$, α_j , and $\lambda(t_i)$ it yields the expression:

$$\Delta J = \sum_{i=1}^{n^t} [\lambda(t_i)]^T \left\{ \mathbf{A}\Delta\phi(t_i) - \frac{1}{\Delta t} \mathbf{S}\Delta\phi(t_{i-1}) + \frac{\partial \mathbf{A}}{\partial \alpha_j} \phi(t_i) \Delta\alpha_j \right\} + \frac{\partial J}{\partial \phi(t_i)} \Delta\phi(t_i), \quad (6.22)$$

from which a model can be derived that simulates the adjoint state vector backwards in time so:

$$\mathbf{A}^T \lambda(t_i) = -\frac{1}{\Delta t} \mathbf{S} \lambda(t_{i+1}) - 2\mathbf{W}^o [\mathbf{y}^o(t_i) - \mathbf{H}\phi(t_i)]. \quad (6.23)$$

Eventually, the corresponding gradient becomes:

$$\frac{\Delta J}{\Delta \alpha_j} = \sum_{i=1}^{n^t} [\lambda(t_i)]^T \left(\frac{\partial \mathbf{A}}{\partial \alpha_j} \right) \phi(t_i). \quad (6.24)$$

The time dependency of $\mathbf{A}$ is caused by the nonlinear conductance $(\mathbf{C}^f)^t$ (Equation 6.2b).

6C Derivation of the First Derivative of the Hydraulic Conductance

Within the elaboration of the reduced Galerkin model, the partial derivative of the hydraulic conductance C_i with respect to the estimate variable $\Delta\alpha_j$ is introduced.

For each location $x \in \Omega$ and a specific direction i this can be expressed as:

$$\frac{\partial C_i(x)}{\partial \alpha_j} = \frac{\partial C_i(x)}{\partial T^b(x)} \cdot \frac{\partial T^b(x)}{\partial \alpha_j}, \quad (6.25)$$

and with (Equation 6.20) this yields:

$$\begin{aligned} \frac{\partial C_i(x)}{\partial \alpha_j} = & \frac{\Delta w_i(x) [\Delta w_i(x) + \Delta w_i(x^*)]}{[T^b]^2(x) \left[\frac{\Delta w_i(x)}{T^b(x)} + \frac{\Delta w_i(x^*)}{T^b(x^*)} \right]^2} \cdot T^b(x) + \\ & + \frac{\Delta w_i(x^*) [\Delta w_i(x) + \Delta w_i(x^*)]}{[T^b]^2(x^*) \left[\frac{\Delta w_i(x)}{T^b(x)} + \frac{\Delta w_i(x^*)}{T^b(x^*)} \right]^2} \cdot T^b(x^*), \end{aligned} \quad (6.26)$$

where the background transmissivity is defined as:

$$T^b = [\alpha_j^b]^\eta T, \quad (6.27)$$

that depends on the background estimation α_j^b for the current outer iteration cycle η .

6D Derivation of the Reduced Incremental Adjoint Model

The formulation of a reduced adjoint model for Equation (6.10) can be concisely derived by a Taylor's series expansion for the variables $\underline{\lambda}$, $\mathbf{r}(t_i)$, $\mathbf{r}(t_{i+1})$ and α_j . It yields eventually a reduced adjoint model:

$$\left[\mathbf{N} + \sum_{j=1}^{n^u} \underline{\mathbf{u}}_j \Delta \alpha_j - \frac{1}{\Delta t} \underline{\mathbf{S}} - \underline{\mathbf{C}}^f \right]^\top \underline{\lambda}(t_{i-1}) = -\frac{1}{\Delta t} \underline{\mathbf{S}} \underline{\lambda}(t_i) + \frac{\partial \hat{J}}{\partial \mathbf{r}}(t_{i-1}), \quad (6.28)$$

that needs to be solved backwards in time whereby

$$\frac{\partial \hat{J}}{\partial \mathbf{r}(t_{i-1})} = -2\underline{\mathbf{W}}^o \left[\mathbf{P}^\top \left(\mathbf{y}^o(t_{i-1}) - H[\phi^b(t_{i-1})] \right) - \mathbf{r}(t_{i-1}) \right], \quad (6.29)$$

where the reduced weight matrix $\underline{\mathbf{W}}^o$ is computed as:

$$\underline{\mathbf{W}}^o = \mathbf{P}^\top \mathbf{W}^o \mathbf{P}. \quad (6.30)$$

Eventually, the gradient can be obtained by applying

$$\frac{\Delta J}{\Delta \alpha_j} = \sum_{i=1}^{n^t} [\underline{\lambda}(t_i)]^\top \underline{\mathbf{u}}_j \mathbf{r}(t_i). \quad (6.31)$$

6E Perturbation of the Hydraulic Conductance with a Pattern Direction

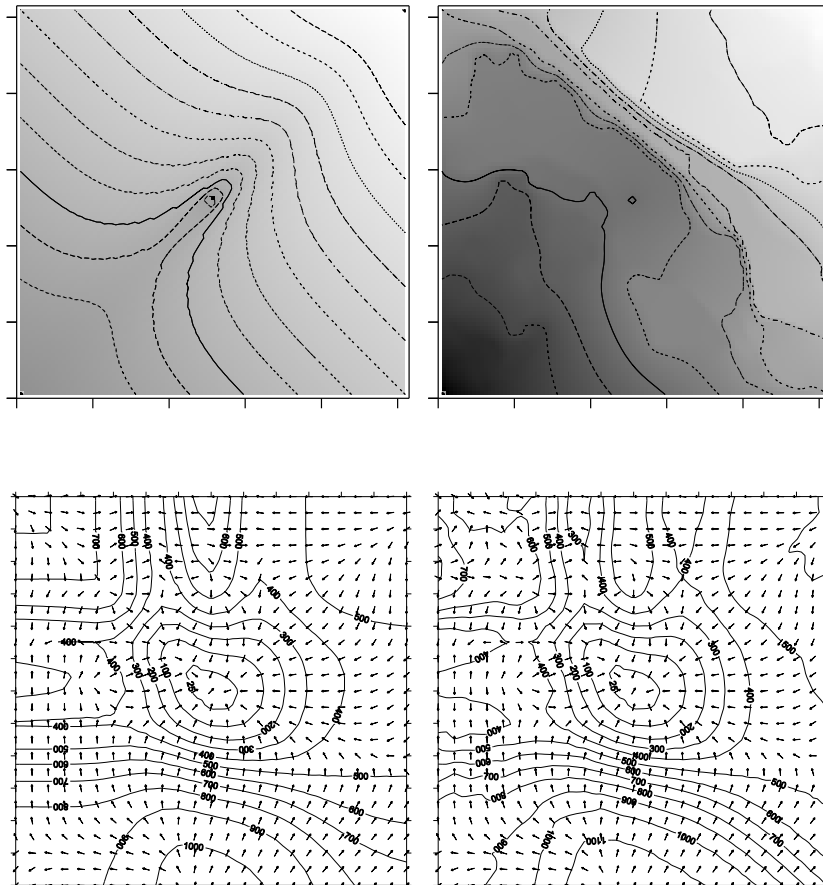
In Equation (6.15) the operator is defined as $M_i[\phi^b, \alpha_j^b + \Delta\tilde{\beta}_j \mathbf{f}_j]$ for each location $x \in \Omega$. It can be computed for the elements $C_{ij}(x)$ in matrix $\mathbf{A} \in \mathbf{A}^p \in M_i$ as:

$$C_{ij}(x) = [\Delta w_i(x) + \Delta w_i(x^*)] \left[\frac{\Delta w_i(x)}{T_j^b(x) + \{\Delta\tilde{\beta} f T^b(x)\}_j} + \frac{\Delta w_i(x^*)}{T_j^b(x^*) + \{\Delta\tilde{\beta} f T^b(x^*)\}_j} \right]^{-1}, \quad (6.32)$$

where T_j^b is defined in Equation (6.27) and the usage of $(\cdot)^*$ is explained in Appendix 6A.

7

Summary and Conclusions



NUMERICAL MODELING OF PROCESSES has become the main application of computing power for research in different fields of science. In general we can say that the accuracy of these models relates to the scale on which phenomena are studied. So, the smaller the scale the more accurate the model may become. Unfortunately, it is impossible to decrease the scale repeatedly without facing a dramatic increase of the need for disk space and computation times. Therefore, there is a need for alternative methods that simulate such models more efficiently. The main objective of this thesis was to develop techniques that represent the original model with a negligible loss of accuracy, but at low-dimensional costs. Within this frame, we searched for generic methodologies that can serve different applications. We described two methods that reduce the problem in its spatial domain; (1) a method which reduces the problem locally by increasing the scale, we call this *upscaling*; and (2) a method which reduces the problem globally by reducing its freedom of movement with respect to its behavior, we call this *model reduction*.

7.1 Upscaling

Reducing the number of computational nodes (*upscaling*) is the most straightforward approach in decreasing the computational demands for a model. However, the accuracy decreases as well because the same amount of information is described with fewer computational nodes. The main challenge is to define the algorithms for upscaling that conserve the accuracy best and ignore only those parts of the model that are irrelevant. In this thesis we developed and applied several algorithms for upscaling the numerical simulation of groundwater flow. We can divide them into two categories that compute an equivalent of the

1. Transmissivity Field.

We applied different methods to a variety of transmissivity fields. These methods varied between numerical experiments and statistical assumptions. It appeared that each of these techniques performed only optimally for that typical distribution for which they were developed. However, when we applied these techniques to a real-world case, we observed that differences between the upscaling techniques were negligible. Moreover, all techniques performed rather poorly and unacceptable errors arose already with small

upscaling factors. We obtained the best results by attuning the upscaled parameters to the corresponding flow field.

2. Boundary Conditions.

We addressed the issue that water, which exchanges between external boundary conditions (e.g. surface water) and the model itself, should be identical, no matter what level of upscaling is applied. Such an error emerges because we reduced the number of computational nodes in our model, which creates the need for boundary conditions to be grouped together. This problem was solved by interchanging the original boundary conditions by those that exchange a similar amount of water. Nonetheless, the main condition for this method was that we had to acquire the entire flow field.

The above mentioned results implied that to obtain a most accurate coarsened model, we needed the original flow field. A practical solution to solve this, is to obtain the original flow field by collecting different flow fields from a number of models. From each of these a different area was collected that was excluded from upscaling. It remains questionable whether other flow fields than the one used to obtain an upscaled model can still be simulated in the future. This makes it necessary to adjust the upscaled parameters again that annul the gained computational effort. This was one of the reasons to focus for the rest of the thesis on the second method: *model reduction*.

7.2 Model Reduction Methodologies

Roughly speaking, the goal of model reduction is to reconstruct a specific behavior that is optimal in terms of storage capacity and computational effort. We can achieve this by suppressing any redundancy in the behavior by applying a *Principal Component Analysis*, which is comparable to data compression. It “breaks up” a specific behavior into different components (i.e. the principal components) from which the original behavior can be resolved again by multiplying the principal components with appropriate weight factors. Consequently, the principal components serve a specific behavior. Whenever this changes into extraordinary values, they become incorrect. Within this thesis we developed several mathematical expressions to compute the weight factors. This yielded a variety of different models that we refer to as *reduced models*. Each of them serves a different purpose in which we distinguish scenario analysis and inverse modeling.

7.2.1 Scenario Analysis

A main purpose of a reduced model is scenario analysis. This model is used frequently to evaluate the consequences of different interferences. For this particular application the reduced model methodology consists of the following two steps:

Evaluation of the Principal Components

It takes relatively much effort to determine the behavior of a model by model simulations to which we refer to as *snapshots*. It takes a thorough knowledge of the underlying model as well as a solid survey of future scenarios. Moreover, the choices that are made to compute the snapshots determine the quality of a reduced model eventually. To minimize the number of snapshots we considered the *principle of superposition* to be valid. This principle accounts for effects of individual interferences that can be added up to form a lumped effect of several interferences. The principal components are computed for these snapshots so that they represent the main behavior of the model most optimally with the smallest number of components.

Formulation of a Reduced Model

The principal components enable us to apply a transformation of a model with many computational nodes into a model with fewer computational nodes. We refer to the latter type of model as *reduced model*. It operates within a different coordinate system and therefore their state variables are not comparable anymore to the original ones. We call them weight factors and we have developed two different formulations to compute them. Briefly summarized these are the:

- *State-Space Projection Method (SSPM)*.
This method reduces the original system of equations written in matrix notation by a matrix multiplication with the principal components. The method is generic and simple to implement;
- *Galerkin Projection Method (GPM)*.
This method substitutes the state variables within the differential equation by the linear combination of the principal components and their corresponding weight factors. The outcome is projected again onto the principal components to form a reduced system of differential equations. The method can be seen as the more elementary of the two.

We applied the two strategies to simulate different scenarios with a realistic three-dimensional groundwater problem. Both methods could simulate the scenarios within a tight accuracy constraint. Their efficiency was approximately two orders of magnitude. Besides that, the GPM was slightly less efficient than the SSPM. A disadvantage of both strategies is the existence of nonlinearity in the model, i.e. the state variables influence the system of equations and vice versa. This undermines the validity of the *principle of superposition* and force us to increase the number of snapshots to capture all crossterms between scenarios. It is questionable whether the exposure of nonlinearity is such that an approximation by a linear reduced model becomes unacceptable. For these cases the value of a fast approximation should be weighed by the value of a slow and more accurate solution.

7.2.2 Inverse Modeling

A new element in model reduction is its incorporation in inverse modeling. Inverse modeling is defined as computing the optimal set of model parameters that yields a minimal error of the model compared to measurements (i.e. the minimal objective function value). For this specific purpose the reduced model methodology consists of two steps:

Evaluation of the Principal Components

We computed snapshots that describe the behavior of the model as well as its relationship to a change of specific model parameters. In this way we simulated the model as if it was formulated for a specific time window, and subsequently interchanged several values for the model parameters. This results in an increased number of snapshots and probably more principal components than those obtained by the scenario analysis. An important advantage is that there are more degrees of freedom left in the acquired reduced model that can be used to perform the inverse modeling.

Formulation of a Reduced Model

The following formulations for a reduced model have in common that they minimize a reduced objective function. They yield a set of model parameters that are optimal for the current reduced model, but can still be sub-optimal to the original model. From here the entire determination of a reduced model is restarted, so that it reflects the updated set of parameters. This sequence continues until a given termination criterion is met for the innovation of the selected parameters. We distinguished three different reduced-model formulations to perform this procedure:

- *Galerkin Projection Method (GPM).*
This method (see page 159) prescribes that the updating of each model parameter is performed within the original model. This concept is most accurate but undermines the efficiency of a reduced model because it is relatively computational demanding to reformulate a reduced model repeatedly.
- *Incremental Galerkin Projection Method (IGPM).*
This method is an extension of the GPM in which we added an extra differential (the *increment*) that expresses the analytical dependence of the state to a set of model parameters. Such a formulation is more efficient than the GPM, although it involves a significant programming effort.
- *Incremental Sequential Projection Method (ISPM).*
This method computes solely the current state from the previous one (i.e. an *auto-regressive* model). We have added a differential (the *increment*) that expresses the numerical dependence of the state to a set of model parameters. Such a model is simple and efficient and the corresponding adjoint model is easy to implement. The novel aspect is that such a model is acquired relatively easily with the principal components. The methodology is generic and

therefore applicable to different simulation models that behave (non)linearly and/or include coupled systems.

We have tested the above mentioned reduced models on twin experiments. First, we generated observations by simulating the original model. Second, we distorted the initial model parameters and used the observations to recover the initial model parameters by use of a reduced model. We compared these results to similar experiments that were performed with: (1) the method of finite differences (FDM) that is easy to program but less efficient, and (2) the adjoint method (ADJ) that is hard to program but more efficient. In most of the experiments the reduced models recovered the initial set of parameters unanimously. They appeared to be more efficient than the FDM method and moreover, the IGMP and ISPM approximated the efficiency of the ADJ method. The main drawback of the last method is its programming effort and because of that, the ISPM is an improved alternative.

Peter Vermeulen

Samenvatting en Conclusies

Model-Gereduceerd Invers Modelleren

HET NUMERIEK SIMULEREN van wetenschappelijke processen is een belangrijke toepassing geworden van de sterk toegenomen rekenkracht van computers. We kunnen stellen, dat de nauwkeurigheid hiervan gerelateerd is aan de schaal waarop we het proces bestuderen. Hoe kleiner de schaal, hoe groter de kans dat het model hier ook nauwkeuriger van zal worden. Helaas is het onmogelijk om de schaal onbeperkt te verkleinen, zonder hiervan direct de nadelen te ondervinden van de toegenomen benodigde opslag- en rekencapaciteit. Het noodzaakt hierom naar alternatieve methoden te zoeken, die meer efficiënt zijn. Het hoofddoel van dit proefschrift was daarom ook, om verschillende methodes te ontwikkelen, die het oorspronkelijke model konden vervangen door één met een grotere rekensnelheid en zonder noemenswaardig verlies van nauwkeurigheid. We hebben hierbij gezocht naar generieke methodieken die kunnen dienen voor verschillende toepassingen. We hebben twee hoofdthema's beschreven die beide het probleem in de ruimtelijke betekenis reduceren: (1) een methode die het probleem lokaal reduceert door de schaal ter plekke te vergroten, dit noemen we *opschalen*; en (2) een methode waarmee we het probleem globaal reduceren door de bewegingsruimte van het model te beperken, dit noemen we *modelreductie*.

Opschalen

Het verminderen van het aantal rekenknooppunten (*opschalen*) is de meest voor de hand liggende werkwijze om het rekenproces van een model te versnellen. De nauwkeurigheid van een opgeschaald model is hierdoor altijd minder, omdat we nu met minder rekenknooppunten dezelfde hoeveelheid informatie moeten beschrijven. Een belangrijke uitdaging ligt nu in het bepalen van die opschalingsregels, die de nauwkeurigheid het best conserveren en alleen de minder relevante delen van een model verwaarlozen. In dit proefschrift hebben we verschillende opschalingsregels ontwikkeld en toegepast op het numeriek simuleren van grondwaterstroming. We kunnen deze methoden onderscheiden in twee categorieën die de opgeschaalde equivalent berekenen van de

1. Hydraulische Doorlatendheid van de Ondergrond.

We hebben hierbij verschillende opschalingsregels toegepast voor een verscheidenheid aan doorlatendheidsverdelingen. Deze methodes varieerden tussen numerieke experimenten en statistische aannamen. Hieruit volgde, dat iedere opschalingsregel het best presteerde voor alleen die typische verdeling van de doorlatendheid, waarvoor de opschalingsregel ook ontwikkeld

was. De verschillen tussen de opschalingsregels verdwenen nagenoeg, zodra we deze toepasten op een realistisch probleem. Het bleek zelfs, dat alle opschalingsregels reeds slechte resultaten opleverden bij een kleine opschalingsfactor. Om het opschalen toch zo goed mogelijk te doen, dienen we de equivalente doorlatendheid af te stemmen op het bijbehorende stromingspatroon.

2. Externe Randvoorwaarden van het Model.

Een voorwaarde voor opschalen is, dat de uitwisseling tussen de externe randvoorwaarden (bijvoorbeeld het oppervlaktewater) en het model na opschalen, ook overeen moet komen met deze in het oorspronkelijke model. Een fout hierin ontstaat, doordat het verminderen van het aantal rekenknooppunten ons noodzaakt om externe invloeden te groeperen. Deze fout kan gereduceerd worden door equivalente opgeschaalde randvoorwaarden te berekenen, die gelijke uitwisseling waarborgen, zoals die in het oorspronkelijk model ook voorkomt. Voorwaarde hiervoor was, dat het gehele stromingspatroon, op de oorspronkelijke schaal aanwezig diende te zijn.

Bovenstaande resultaten impliceren beiden, dat een nauwkeurig opgeschaald model alleen verkregen kan worden door gebruik te maken van het aanwezige stromingspatroon. Een praktische invulling hiervoor is om dit stromingspatroon te verkrijgen door van verschillende, lokaal opgeschaalde modellen hun stromingspatronen 'bijeën te rapen', waarbij telkens dát deel overgenomen wordt, dat geen onderdeel was van een zekere opschaling. Het zal altijd dubieus blijven of een dergelijk opgeschaald model nauwkeurig genoeg blijft voor de simulatie van andere stromingspatronen. Dit noodzaakt namelijk wederom een aanpassing van de opgeschaalde parameters, dat uiteindelijk de gewonnen rekensnelheid volledig teniet doet. Dit was één van de redenen om ons in de rest van het proefschrift te richten op het tweede onderdeel: *modelreductie*.

Gereduceerd Model

Ruw gezegd, is het doel van modelreductie erop gericht om een bepaald gedrag zodanig te herschrijven, zodat deze minder opslag en rekentijd vergt. We kunnen dit verkrijgen door het toepassen van een *Principal Component Analysis*, dat vergelijkbaar is met data compressie. Het 'breekt' een bepaald gedrag in een minimum aan componenten van waaruit het oorspronkelijke gedrag, of afgeleiden hiervan, herleid worden door de voornaamste componenten te vermenigvuldigen met de juiste gewichtsfactoren. De componenten dienen hierom een vooraf gedefinieerd doel en wanneer dit significant verandert, verliezen ze hun geldigheid.

De berekening van de gewichtsfactoren is in dit proefschrift afgeleid voor verschillende wiskundige formuleringen en leverde een variëteit aan modellen op, die ieder worden aangeduid als *gereduceerd model*. Ieder van hen diende een ander doel waarbinnen we het onderscheid hebben gemaakt tussen scenario analyse en invers

modelleren.

Scenario Analyse

Een belangrijke toepassing van modelreductie is scenario analyse. Hierbij dient het model regelmatig geëvalueerd te worden om de consequenties van bepaalde ingrepen te kwantificeren. Voor deze specifieke toepassing bestaat een gereduceerd model uit de volgende twee stappen:

1. Bepaling van de Voornaamste Componenten.

Het kost relatief veel tijd om het gedrag van een model, door middel van modelsimulaties (*momentopnamen*) te berekenen. Het vergt bovendien een gedegen kennis van het onderliggende model en een helder overzicht van de toekomstige scenario's. Daarnaast bepalen de keuzes die hierin gemaakt worden in belangrijke mate de kwaliteit van het gereduceerde model. Om de hoeveelheid momentopnamen te minimaliseren zijn we uitgegaan van het *superpositie beginsel*, dat toestaat dat individuele effecten van ingrepen opgeteld kunnen worden om zodoende een totaal effect te kunnen vormen. De voornaamste componenten worden voor deze momentopnamen berekend waarmee we het belangrijkste gedrag van het model zo optimaal mogelijk, dus met een minimaal aantal componenten, vertegenwoordigen.

2. Formulering van een Gereduceerd Model.

De voornaamste componenten stellen ons in staat om een model met veel rekenknooppunten, te transformeren naar een model met weinig rekenknooppunten. Dit laatste noemen we een gereduceerd model en aangezien deze, na de transformatie binnen een ander coördinatensysteem opereert, zijn haar oplossingen niet meer gelijk aan deze van het oorspronkelijke model. We noemen ze dan gewichtsfactoren en voor de berekening hiervan hebben we een tweetal formuleringen beschreven. Kort samengevat zijn dit de

- *State-Space Projection Method* (SSPM).

Deze methode reduceert het aantal rekenknooppunten door deze, weergegeven in matrix- en vectornotatie, te vermenigvuldigen met de voornaamste componenten. De methode is generiek en eenvoudig te implementeren;

- *Galerkin Projection Method* (GPM).

Deze methode vervangt de toestandsvariabele in de oorspronkelijke differentiaalvergelijking door een lineaire combinatie van de voornaamste componenten met haar gewichtsfactoren. Het resultaat hiervan wordt geprojecteerd op de componenten en resulteert in een gereduceerde differentiaalvergelijking. De methode is hierom meer elementair dan de SSPM, maar vergt meer aanpassingen in de bestaande modelcode.

We hebben beide strategieën toegepast op een realistisch grondwaterstromings-model en beiden waren in staat om een verscheidenheid aan scenario's te berekenen, zonder noemenswaardig verlies van nauwkeurigheid. Hun efficiëntie bedroeg bijna twee ordes van grootte waarbij de GPM hierbij minder efficiënt was dan de SSPM. Een nadeel van beide strategieën kan echter de eventuele aanwezigheid van niet-lineariteiten zijn; een situatie waarbij de toestandsvariabele het systeem beïnvloedt en andersom. Hierdoor is het *superpositie principe* niet langer geldig en zijn meer voornaamste componenten nodig om alle kruistermen tussen de scenario's te kunnen beschrijven. Het is echter de vraag of het effect van niet-lineariteit zich dermate manifesteert, dat een gereduceerd lineair model te onnauwkeurig wordt. De waarde van een snelle schatting moet in dat geval afgewogen worden tegen de langdurige inspanning van een 'correcte' numerieke berekening.

Invers Modelleren

Een nieuwe ontwikkeling op het gebied van modelreductie is haar toepassing in het invers modelleren. Dit is een berekeningswijze waarbij een combinatie van modelparameters wordt berekend die optimaal is, omdat de fout van het model ten opzichte van een aantal metingen, de doelfunctie, hierbij minimaal is. Voor deze specifieke toepassing bestaan de onderdelen voor een gereduceerd model uit de:

1. Bepaling van de Voornaamste Componenten.

We creëren hier momentopnamen die zowel het gedrag van een model vertegenwoordigen alsook een verandering van bepaalde modelparameters. Hiervoor simuleren we het model zoals het gedefinieerd is voor een specifiek tijdsinterval en verwisselen we herhaaldelijk de waarde van de geselecteerde modelparameters. Dit resulteert in een grotere reeks van momentopnamen en hoogstwaarschijnlijk ook in een groter aantal voornaamste componenten, dan die benodigd waren bij de scenario analyse. Belangrijk voordeel hierbij is, dat het gereduceerde model meer graden van vrijheid kent, die gebruikt kunnen worden voor de invers modellering.

2. Formulering van een Gereduceerd Model.

De gereduceerde modellen die hierna beschreven worden, hebben alledrie gemeen, dat ze in staat zijn om een gereduceerde doelfunctie te minimaliseren. Ze leveren een combinatie van parameters die optimaal zijn, maar eventueel nog sub-optimaal kunnen zijn voor het oorspronkelijke model. De gehele bepaling van een gereduceerd model herhaalt zichzelf, zodat deze de aangepaste parameters reflecteert, waarna het de doelfunctie eventueel verder kan verlagen. Dit proces herhaalt zichzelf totdat de verbetering van de modelparameters minder is dan een opgegeven afbreekcriterium. Voor de uitwerking van deze procedure hebben we een drietal gereduceerde modellen geformuleerd, dit zijn de

- *Galerkin Projection Method* (GPM).
Bij deze methode (zie pagina 159) wordt iedere aanpassing van een modelparameter in het oorspronkelijke model gedaan. Hierna wordt het gereduceerde model opnieuw geformuleerd. Dit is nadelig voor de rekkentijd, aangezien het relatief veel tijd kost om een gereduceerd model herhaaldelijk te moeten formuleren.
- *Incremental Galerkin Projection Method* (IGPM).
De methode is een uitbreiding van de GPM waarbij we een extra differentiaal toevoegen (de *increment*) die de analytische afhankelijkheid van de toestandsvariabele naar de modelparameters vertegenwoordigt. Een dergelijke methode kent een grotere efficiëntie dan de GPM, maar betekent nog steeds een aanzienlijke programmeerinspanning.
- *Incremental Sequential Projection Method* (ISPM).
Deze methode berekent de nieuwe toestand uitsluitend op basis van de vorige (een *auto-regressief* systeem). We hebben hieraan een extra differentiaal toegevoegd, die de numerieke afhankelijkheid van de toestand naar de modelparameters vertegenwoordigt (de *increment*). Het model is eenvoudig, efficiënt en haar adjointmodel is eenvoudig te implementeren. Het hernieuwde aspect hierbij is, dat we dit type model op relatief eenvoudige wijze kunnen verkrijgen, voor een willekeurige (niet)-lineaire (gekoppelde) modellen, door gebruik te maken van de voornaamste componenten.

We hebben deze strategieën toegepast op tweeledige experimenten. Hierbij hebben we eerst met het oorspronkelijke model een synthetische meetset gegenereerd. Daarna hebben we deze gebruikt om een aangebrachte verstoring van de initiële modelparameters te herstellen met een gereduceerd model. De resultaten hiervan hebben we vergeleken met deze, die we verkregen door soortgelijke experimenten uit te voeren met: (1) de eindige-verschillen methode (FDM) die eenvoudig te implementeren, maar weinig efficiënt is, en (2) de adjointmethode (ADJ) die lastiger te implementeren, maar zeer efficiënt is. Alle gereduceerde modellen vonden in verreweg de meeste experimenten de juiste initiële modelparameters terug. Tevens bleken ze alle drie meer efficiënt dan de FDM methode en kwamen de IGPM en ISPM hierbij zelfs in de buurt van de ADJ methode. Bij deze laatste staat haar lastige implementatie vaak in de weg, waardoor vooral de ISPM een verbeterd alternatief is.

Bibliography

The numbers in brackets refer to the page(s) on which a publication is cited. A page number may appear more than once when a publication is cited more than once on the same page.

- Aavatsmark, I., T. Barkve, T. Bøe, Ø., & T. Mannseth, Discretization on non-orthogonal, quadrilateral grids for inhomogeneous, anisotropic media, *J. Comput. Phys.*, 127, 2–14, 1996. [29,49]
- Adomaitis, R.A., RTCVD Model Reduction: A Collocation on Empirical Eigenfunctions Approach. Technical Report T.R. 95–64, Inst. Systems Research, 1995. [55,113]
- Ames, W.F., *Numerical Methods for Partial Differential Equations*, Academic Press, San Diego, CA, 1992. [31]
- Anderman, E.R., K.L. Kenneth, M.C. Hill, J. Valstar, & R.M. Neupauer, *Modflow-2000, The U.S. Geological Survey Modular GroundWater Model - Documentation of the Model Layer Variable Direction Horizontal Anisotropy (LVDA) Capability of the Hydrogeological Unit Flow (HUF) Package*, U.S. Geological Survey, Open-File Report 02–409, Denver, Colorado, 2002. [29,51]
- Antoulas, A., S. Sørensen, & K.A. Gallivan, Model Reduction of Large-Scale Dynamical Systems *Lecture Notes In Computer Science*, 3038, 740–747, 2004. [111]
- Arnoldi, W., The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.*, 9, 17–29, 1951. [81]
- Atwell, J.A., & B.B. King, Reduced Order Controllers for Spatially Distributed Systems Via Proper Orthogonal Decomposition, *SIAM J. On Scientific Computing*, 26(1), 128–151, 2004. [111,133]
- Bear, J., *Dynamics of Fluids in Porous Media*, American Elsevier, New York, 1972. [29]
- Bennet, A.F., *Inverse Methods in Physical Oceanography*. Cambridge University Press, 1992. [90,132]
- Bennet, A.F., B.S. Chua, & L.M. Leslie, Generalized Inversion of a global numerical weather prediction model, *Metereology and Atmospheric Physics*, 60, 165–178, 1996. [90]

- Bensoussan, A., J.-L., Lions, & G. Papanicolau, *Asymptotic Analysis for Periodic Structures*, North Holland Publishing Company, Amsterdam, 1978. [34,34]
- Bierkens, M.F.P., & J.W.J. van de Gaast, Upscaling hydraulic conductivity: theory and examples from geohydrological studies, *Nutrient Cycl. in Agroecosyst.*, 50, 193–207, 1998. [35]
- Bøe, O., Analysis of an upscaling method based on conservation of dissipation, *Transp. in Por. Media*, 17, 77–86, 1994. [31]
- Bourgeat, A., Homogenized behavior of two-phase flows in naturally fractured reservoirs with uniform fractures distribution. *Comput. Math. Appl. Mech. Eng.*, 47, 205–216, 1984. [29]
- Cao, J., & P.K. Kitanidis, Adaptive grid simulation of groundwater flow in heterogeneous aquifers, *Adv. in Water Res.*, 22, 681–696, 1999. [31]
- Carrera, J., & S.P. Neuman, Estimation of aquifer parameters under transient and steady-state conditions. 1. Maximum-likelihood method incorporating prior information. *Water Resources Research*, 22(2), 199–210, 1986. [90,92,110,132]
- Cazemier, W., R.W.C.P. Verstappen, & A.E.P. Veldman, Proper Orthogonal Decomposition and Low-Dimensional Models for Driven Cavity Flows, *Phys. of Fluids*, 10, 1685–1699, 1998. [54,90,96,116,133]
- Chen, Y., L.J. Durlofsky, M. Gerritsen & X.H. Wen, A coupled local-global upscaling approach for simulating in highly heterogeneous formations, *Adv. in Water Res.*, 26, 1041–1060, 2003. [28,35,35,35]
- Cooley, R.L., A comparison of several methods of solving nonlinear regression groundwater flow problems: *Water Resour. Res.*, 21(10), 1525–1538, 1985. [90,90,92,132,135]
- Cooley, R.L., & M.C. Hill, A comparison of three Newton-like nonlinear least-squares methods for estimating parameters of groundwater flow models, in *Computational Methods in Water Resources 9th, vol. 1: Numerical methods in water resources*, Elsevier, eds. Russell, T.F., Ewing, R.E., Brebbia, C.A., Gray, W.G., & Pinder, G.F., 1, 379–386, 1992. [90]
- Courant, R., & D. Hilbert, *Methods of Mathematical Physics*, Wiley Interscience, New York, 1953. [90,94,113,135]
- Courtier, P., Dual formulation of four-dimensional variational assimilation, *Quart. J. Roy. Met. Soc.*, 123, 2449–2461, 1997. [133]
- Courtier, P., Thepaut, J.N., & A. Hollingsworth, A strategy for operational implementation of 4D-Var, using an incremental approach, *Quart. J. Roy. Met. Soc.*, 120, 1367–1387, 1994. [133]

- Delay, F., A. Buoro, & G. de Marsily, Empirical Orthogonal Functions Analysis Applied to the Inverse Problem in Hydrogeology: Evaluation of Uncertainty and Simulation of New Solutions, *Math. Geology*, 33(8), 927–949, 2001. [87,90,90]
- Desbarats, A.J., Spatial averaging of hydraulic conductivity in three-dimensional heterogeneous porous media, *Math. Geol.*, 24(3), 249–267, 1992. [32]
- Doren, J. van, R. Markovinović, & J.D. Jansen, Reduced-Order Optimal Control of Water Flooding Using Proper Orthogonal Decomposition, *Comp. Geosciences*, 9(4), 2005. [111,133]
- Dunbar, W.S., & A.D. Woodbury, Application of the Lanczos algorithm to the solution of the groundwater flow equation, *Water Resour. Res.*, 25(3), 551–558, 1989. [54,82]
- Dunbar, W.S., A.D. Woodbury, & B. Nour-Omid, Comment on 'On time integration of groundwater flow equations by spectral methods' by G. Gambolati, *Wat. Resour. Res.*, 25(3), 551–558, 1994. [83]
- Duvaut, G., & J.-L., Lions, *Inequalities in Mechanics and Physics*, Springer-Verlag, Berlin, 1976. [30,32]
- Durlofsky, L.J., Numerical Calculation of Equivalent Grid Block Permeability Tensors for Heterogeneous Porous Media, *Wat. Resour. Res.*, 27(5), 699–708, 1991. [28,34,34]
- Durlofsky, L.J., R.C. Jones, & W.J. Milliken, A nonuniform coarsening approach for the scale-up of displacement processes in heterogeneous porous media, *Adv. in Water Res.*, 20(5–6), 335–347, 1997. [28]
- Durlofsky, L.J., W.J. Milliken, & A. Bernath, Scaleup in the Near-Well Region, *SPE J.*, 1, 110–117, 2000. [42]
- Edwards, M.G., & C.F. Rogers, Finite Volume discretization with imposed flux continuity for the general tensor pressure equation, *Comp. Geosc.*, 2, 259–290, 1998. [49]
- Edwards, M.G., M-Matrix flux splitting for general full tensor discretization operators on structured and unstructured grids, *J. of Comp. Physics*, 160, 1–28, 2000. [51,141]
- Evensen, G., Sequential Data Assimilation with A Nonlinear Quasi-Geostrophic Model Using Monte-Carlo Methods to Forecast Error Statistics. *J. Of Geophysical Research-Oceans*, 99(C5), 10143–10162, 1994. [110]
- Farmer, C.L., Upscaling: a review, *Int. J. for Num. Methods in Fluids*, 40(40), 63–78, (DOI: 10.1002/fld.267), 2002. [27]

- Garcia, M.H., A.G. Journel, & K. Aziz, *An automatic grid generation and adjustment method for modeling reservoir heterogeneities*. Tech. Rep., Stanford Center for Reservoir Forecasting, Report 3, 1990. [40]
- Ghanem, R., Scales of fluctuation and the propagation of uncertainty in random porous media, *Water Resour. Res.*, 34(9), 2123–2136, 1998. [55]
- Gambolati, G., On time integration of groundwater flow equations by spectral methods, *Wat. Resour. Res.*, 29, 1257–1267, 1993. [83]
- Ginn, T.R., & J.H. Cushman, Inverse methods for subsurface flow: A critical review of stochastic techniques, *Stochastic Hydrol. Hydraul.*, 4(1), 1–26, 1990. [90]
- Greenwald, R.M., Documentation and Users Guide: MODMAN, an Optimization Module for MODFLOW, Version 4.0, *HSI GeoTrans, Freehold, New Jersey*, 1998. [106]
- Golub, G., & A. van Loan, *Matrix Computations*, John Hopkins University Press 2nd edition, 1989. [59]
- Gomez-Hernandez, J.J., & A.G. Journel, Stochastic characterization of grid-block permeabilities: from point values to block tensors, In *2nd European Conference on the Mathematics of Oil Recovery*, Arles, France, 1990. [33]
- Hammond, G.E., A.J. Valocchi, & P.C. Lichtner, Application of Jacobian-free Newton-Krylov with physics-based preconditioning to biogeochemical transport. *Adv. in Water Res.*, 28, 359–376, 2005. [132]
- Heemink, A.W., M. Verlaan, & A.J. Segers, Variance reduced ensemble Kalman filtering, *Mon. Weather Rev.* 129(7), 1718–1728, 2001. [87]
- Heemink, A.W., E.E.A. Mouthaan, & M.R.T. Roest, Inverse 3D Shallow Water Flow Modelling of The Continental Shelf. *Continental Shelf Research*, 22(3), 465–484, 2002. [110]
- Heijn, T., R. Markovinović, & J.D. Jansen, Generation of Low-order Reservoir Models Using System-Theoretical Concepts, *SPE J.*, 202–218, 2004. [56]
- Hemker, C.J., & G.J. Nijsten, *MicroFem*, 1996. [30]
- Hill, M.C., Relative efficiency of four parameter-estimation methods in steady-state and transient groundwater flow models, in *Computational Methods in Subsurface Hydrology, International Conference on Computational Methods in Water Resources*, eds. Gambolati, G., Rinaldo, A., Brebbia, C.A., Gray, W.G. & Pinder, G.F., 103–108, 1990. [90]
- Hoffmann Jørgensen, B., & J.N. Sørensen, Proper Orthogonal Decomposition and Low-Dimensional Modelling. *ERCOTAC Bulletin*, 46, 44–51, 2000. [55,72,90,93,113,133]

- Holden, L., & B.F. Nielsen, Global Upscaling of Permeability in Heterogeneous Reservoirs; The Output Least Squares (OLS) Method, *Transp. in Por. Media*, 40, 115–143, 2000. [28,35]
- Holmes, P., J.L. Lumley, & G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Univ. Press, 1996. [55,59,133]
- Hooimeijer, M.A., *Reduction of Complex Computational Models*, PhD Thesis, Printed by SIECRA REPRO, Delft, The Netherlands, 2001. [54,55,55,55,128]
- Iman, R.L., & M.J. Shortencarier, *A Fortran 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use with Computer Models*. NUREG/CR-3624, Technical Report SAND83-2365, Sandia National Laboratories, Albuquerque, NM, 1984. [97,125,137,145]
- Jazwinski, A.H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970. [110,110]
- Journel, A.G., C.V. Deutsch, & A.J. Desbarats, Power averaging for block effective permeability, *SPE 15128*, Soc. of Petroleum Engineers, 1986. [32]
- Kaiser, H.F., The Varimax criterion for analytical rotation of factor analysis, *Psychometrika*, 23, 187, 1958. [62]
- Kaminski, T., R. Giering, & M. Scholze, An Example of an Automatic Differentiation-Based Modelling System. *Lecture Notes In Computer Science*, 2668, 95–104, 2003. [110]
- Krysl, P., S. Lall, & J.E. Marsden, Dimensional Model Reduction in Non-linear Finite Element Dynamics of Solids and Structures. *Int. J. Numer. Meth. Engng*, 51, 479–504, 2001. [90,59]
- Lanczos, C., An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Nat. Bur. Stand*, 45, 255–282, 1950. [80]
- Lardner, R.W., & Y. Song, Optimal Estimation of Eddy Viscosity and Friction Coefficients for a Quasi-3-Dimensional Numerical Tidal Model. *Atmosphere-Ocean*, 33(3), 581–611, 1995. [110]
- Lawless, A.S., S. Gratton, & N.K. Nichols, Approximate iterative methods for variational data assimilation, *Int. J. Numer. Methods in Fluids*, 47, 1129–1135, 2005. [114,133]
- Lee, S.H., L. Durlofsky, M.F. Lough, & W.H. Chen, Finite-difference simulation of geologically complex reservoirs with tensor permeabilities, *SPE Res. Eval. Eng.*, 1(6), 567–574, 1998. [29,49,49]

- Lee, K.S., E. Yongtae, J.W. Chung, J. Choi, & D. Yang, A control-relevant model reduction technique for nonlinear systems, *Computers and Chem. Eng.*, 24, 309–315, 2000. [133]
- Ljung, L., Asymptotic-Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems. *IEEE Transactions On Automatic Control*, 24(1), 36–50, 1979. [110]
- Mansell, R.S., L. Ma, L.R. Ahuja, & S.A. Bloom, Adaptive Grid Refinement in Numerical Models for Water Flow and Chemical Transport in Soil: A Review, *Vadose Zone J.*, 1, 222–238, 2002. [31]
- Matheron, G. *Éléments pour une Théorie des Milieux Poreux*, Masson, Paris, 1967. [32,32]
- McDonald, M.G., & A.W. Harbaugh, *A Modular Three-Dimensional Finite-Difference Groundwater Flow Model*, U.S. Geological Survey, Open-File Report 83–875, Book 6, Chapter A1, 1988. [29,48,51,64,91,134]
- McKay, M.D., W.J. Conover, & R.J. Beckman, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 221, 239–245, 1979. [125]
- McLaughlin, D. & L.R. Townley, A reassessment of the groundwater inverse problem, *Water Resour. Res.*, 32, 1131–1161, 1996. [90]
- Mehl, S.W., & M.C. Hill, MODFLOW-2000, the U.S. Geological Survey Modular Groundwater Model. User Guide to the Link-AMG (LMG) Package for Solving Matrix Equations Using an Algebraic Multigrid Solver, *U.S. Geological Survey*, 2001. [27,90,132]
- Mehl, S.W. & M.C. Hill, Locally refined block-centered finite-difference groundwater models, *Evaluation of parameter sensitivity and the consequences for inverse modelling and predictions*, eds. Karel Kovar, Hrkal Zbynek, *IAHS Publication 277*, 227–232, 2003. [90,90,132]
- Mei, C.C., & J.-L. Auriault, Mechanics of heterogeneous porous media with several spatial scales, *Proc. R. Soc. London, Ser. A*, 426, 391–423, 1989. [29]
- Moore, G., Cramming more components onto integrated circuits, *Electronics Magazine*, 19 April 1965. [15]
- Neuman, S.P. & V. Di Federico, Multifaceted nature of hydrogeologic scaling and its interpretation, *Reviews of Geophysics*, 41(3), 1014, doi:10.1029/2003RG000130, 2003. [27]

- Newman, A.J., *Model reduction via the Karhunen-Loève expansion Part I: An exposition*. Technical Report T.R. 96-32, Inst. Systems Research, University of Maryland in College Park, MD, 1996a. [54,55,58,90,113,132]
- Newman, A.J., *Model reduction via the Karhunen-Loève expansion Part II: Some Elementary Examples*. Technical Report T.R. 96-33, Inst. Systems Research, University of Maryland in College Park, MD, 1996b. [113,132]
- Newman, A.J., & P.S. Krishnaprasad, Nonlinear Model Reduction for RTCVD. *Technical Report T.R. 98-10*, Inst. Systems Research, University of Maryland in College Park, MD, 1998. [133]
- Nour-Omid, B., Lanczos method for heat conduction analysis, *Int. J. Numer. Methods Eng.*, 24, 251-262, 1987. [82]
- Nour-Omid, B., Application of the Lanczos method, *Comput. Phys. Commun.*, 53, 157-168, 1989. [82]
- Overschee, P. van, & B. de Moor, *Subspace Identification for Linear Systems. Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996. [74]
- Park, H.M., & C.H. Cho, Low dimensional modeling of flow reactors. *Int. J. Heat and Mass Transfer* 39(16), 3311-3323, 1996. [55,71,72,87,90,93,96,133]
- Park, H.M., O.Y. Chung, & J.H. Lee, On the solution of inverse heat transfer problem using the Karhunen Loève Galerkin method, *Int. J. Heat and Mass Transfer* 42(1), 127-142, 1999. [55,91,93,133]
- Pearson, K., On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2, 609-629, 1901. [133]
- Penman, J., Dual and complementary variational techniques for the calculation of electromagnetic fields, *Adv. in Elect. and Electron Phys.*, 70, 315-364, 1988. [30]
- Poeter, E.P., & M.C. Hill, Inverse Methods: A Necessary Next Step in Groundwater Modeling, *Ground Water*, 35, (2), 250-260, 1997. [132]
- Press, W.H., S.A. Teukolsky, W.T. Vetterling, & B.P. Flannery, *Numerical Recipes in Fortran. The Art of Scientific Computing*. Cambridge University Press, 1992. [92,95,112,120,143]
- Ravindran, S.S., Control of Flow Separation over a Forward-Facing Step by Model Reduction, *Computer Methods in Applied Mechanics and Engineering*, 191(41-42), 4599-4617, 2002. [111]
- Renard, Ph., & G. de Marsily, Calculating Equivalent Permeability: A Review, *Adv. in Water Res.*, 20(5/6), 253-278, 1997. [27,31]

- Reyment, R.A., & K.G. Jöreskog, *Applied Factor Analysis in the Natural Sciences*, Cambridge University Press, 1993. [55]
- Richman, M.B., Review Article, Rotation of Principal Components, *J. of Climatology*, 6, 293–335, 1986. [62]
- Rubin, Y. & J. Gómez-Hernández, A stochastic approach to the problem of upscaling of conductivity in disordered media: Theory and unconditional numerical simulation, *Wat. Resour. Res.*, 22(4), 691–701, 1990. [33]
- Sahuquillo, A., An eigenvalue numerical technique for solving unsteady linear groundwater models continuously in time, *Water Resour. Res.* 19, 87–93, 1983. [54]
- Sánchez-Villa, X., J.P. Girardi, & J. Carrera, A synthesis of approaches to upscaling of hydraulic conductivities, *Wat. Resour. Res.*, 31(4), 867–882, 1995. [27,32]
- Sirovich, L., Turbulence and the dynamics of coherent structures; Part I: Coherent Structures. *Quarterly Appl. Math.*, 45(3), 561–571, 1987. [55,56,58,72,113,113,133]
- Snepvangers, J.J.J.C., & C.B.M. te Stroet, Mapping curvilinear structures with local anisotropic kriging, *Math. Geology*, 37(6), 635–650, 2005. [37]
- Strack, O.D.L., *Groundwater Mechanics*, Prentice Hall, Inc., Englewood Cliffs, New-Jersey, 1989. [28,30]
- Tarantola, A., *Inverse problem theory*. Elsevier, 2 edition, 1987. [92,112]
- Townley, L.R., & J.L. Wilson, Computationally efficient algorithms for parameter estimation and uncertainty propagation in numerical models of groundwater flow. *Water Resour. Res.*, 21(12), 1851–1860, 1985. [90,135]
- Ullman, D.S., & R.E. Wilson, Model Parameter Estimation from Data Assimilation Modeling: Temporal and Spatial Variability of the Bottom Drag Coefficient. *J. of Geophysical Research Oceans*, 103(C3), 5531–5549, 1998. [110]
- Valstar, J.R., D.B. McLaughlin, C.B.M. te Stroet, & F.C. van Geer, A representer-based inverse method for groundwater flow and transport applications, *Water Resour. Res.*, 40, W05116, doi:10.1029/2003WR002922, 2004. [90]
- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Model Reduction In Groundwater Hydrology, in *Final Proc. Int. Conf. on Modelling and Simulation*, Eds. F. Ghassemi, P. Whetton, R. Little & M. Littleboy, Canberra, Australia 1, 407–412, 2001. [55]
- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Reduction of large-scale numerical groundwater flow models, in *Final Proc. XIV Int. Conf. on Computational Methods in Water Resources*, Eds. S.M. Hassanizadeh, R.J. Schotting, W.G. Gray, & G.F. Pinder, Delft, the Netherlands, 1, 397–404, 2002. [55]

- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Reduction of large-scale groundwater flow models via the Galerkin Projection, in *13th IFAC Symposium on System Identification*, Eds. P.M.J. Van den Hof, B. Wahlberg, & S. Wieland, Rotterdam, the Netherlands, 1425–1430, 2003. [56]
- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Low-dimensional modeling of numerical groundwater flow, *Hydrological Processes*, 18, 1487–1504, 2004a. [27,55,90,93,113,133,137]
- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Reduced Models for Linear Groundwater Flow Models Using Empirical Orthogonal Functions. *Adv. in Water Res.*, 27(1), 57–69, 2004b. [27,56,90,91,93,98,113,133,137]
- Vermeulen, P.T.M., A.W. Heemink, & J.R. Valstar, Inverse Modeling Using Model Reduction. *Wat. Resour. Res.*, 41, W06003, doi:10.1029/2004WR003698, 2005a. [113,116,133,137]
- Vermeulen, P.T.M., & A.W. Heemink, Model-Reduced Variational Data Assimilation, *Month. Weath. Rev.*, accepted January, 2006a. [133]
- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Model Inversion of Transient Nonlinear Groundwater Flow Models Using Model Reduction, *Wat. Resour. Res.*, submitted, 2006b. [109]
- Vermeulen, P.T.M., A.W. Heemink, & C.B.M. te Stroet, Limitations to Upscaling of Groundwater Flow Models dominated by Surface Water Interaction, *Wat. Resour. Res.*, submitted, 2006c. [25]
- Wallstrom, T.C., M.A. Christie, L.J. Durlofsky, & D.H. Sharp, Effective Flux Boundary Conditions for Upscaling Porous Media Equations, *Transp. in Por. Media*, 46, 139–153, 2002. [34]
- Warren, J.E., & H.S. Price, Flow in Heterogeneous Porous Media, *Soc. of Petroleum Engineers J.*, 153–169, 1961. [30,32,32]
- Wen, X.H., L.J. Durlofsky, & M.G. Edwards, Use of border regions for improved permeability upscaling, *Math. Geology*, 35, 521–547, 2003. [28,33,34,90,132]
- Wiener, O., Die theorie des Mischkorpers für das Feld der Stationären Stromung I. Die Mittelwertsätze für Kraft, Polarisierung und Energie, *Der Abhandlungen der Mathematisch-Physischen Klasse der Königlichen Sächsischen Gesellschaft der Wissenschaften*, 32, 509–604, 1912. [32]
- Winter, C.L., E.P. Springer, K. Costigan, P. Fasel, S. Mniszewski, G. Zyvoloski, Virtual watersheds: Simulating the water balance of the Rio Grande basin. *Comp. in Science and Engineering*, 6, 3, 18–26, 2004. [132]

- Wu, Y.S., K. Zhang, C. Ding, K. Pruess, E. Elmroth, & G.S. Bodvarsson, An efficient parallel-computing method for modeling nonisothermal multiphase flow and multicomponent transport in porous and fractured media. *Adv. in Water Res.*, 25, 243-261, 2002. [132]
- Yeh, W.W.G., Review of Parameter Identification Procedures in Groundwater Hydrology: The Inverse Problem, *Water Resour. Res.*, 22, 95-108, 1986. [90]
- Zhang, K., & A.D. Woodbury, The Arnoldi reduction technique for efficient direct solution of radionuclide decay transport in dual-porosity media, *J. Contaminant Hydrology* 44, 387-416, 2000a. [54,82]
- Zhang, K., & A.D. Woodbury, Application of the Lanczos algorithm to the simulation of groundwater flow in dual-porosity media. *Adv. in Water Res.* 23, 579-589, 2000b. [54,82]
- Zijl, W., & A. Trykozko, Numerical Homogenization of the Absolute Permeability Using the Conformal-Nodal and Mixed-Hybrid Finite Element Method, *Transp. in Por. Media*, 44, 33-62, 2001. [30,40]

Index

A

- adjoint, 91, 106, 109, 111
 - compilers, 110
 - derivation, 148
 - method, 90, 94, 110, 113, 135
 - model, 135
 - reduced, 94, 95
 - derivation, 107, 128, 149
 - model, 118, 139, 141
 - run, 95, 105, 141
- analytical
 - derivatives, 147
 - elements, 62
 - functions, 54
 - solution, 32
- anisotropic, 29
 - behavior, 51
 - medium, 36–39
- aquifer, 27, 29, 36, 45, 47, 70
- aquitard, 29, 36, 48
- Arnoldi method, 81
- autoregressive
 - model, 118, 140, 141
 - representation, 137, 147

B

- background
 - estimates, 119, 136, 140
 - hydraulic head, 57
 - state, 67, 114, 118, 136, 138
 - transmissivity, 149
- balancing, 133
- basis, 55, 59, 117
 - function, 58, 113, 132
 - vector, 115
- borderring, 33, 34
- Brinkman equation, 29

C

- Cauchy conditions, 43
 - correction of, 44, 47

coarse-scale

- Cauchy condition, 43
- conductance, 33
- Dirichlet condition, 40
- equation, 29
- grid, 90, 132
- grid block, 32, 34, 35, 42, 51
 - volume, 31
- model, 30, 39
- Neumann condition, 42
- point, 29

coarsening, 27, 40

conservation

- mass, 75
- of continuity, 31
- of dissipation, 31
- of hydraulic head, 31
- of nature areas, 26, 61

coordinate

- system, 59–61, 119
- transformation, 63, 116

- covariance matrix, 58, 92, 111, 116

- crossbeddings, 39

D

Darcy

- experiment of, 32
- flow, 31, 46
- law, 28, 57, 91
- velocity, 28

- database, 26

- degrees of freedom, 30, 47

- dipping layers, 39

- Dirichlet condition, 139

- simulation of, 40, 47
- upscaling of, 40

- Dupuit-Forchheimer flow, 29

E

effective flux boundary condition, 34
 efficiency, 61, 80, 86, 98, 106, 119, 126, 139, 142
 computational, 110, 133
 of the method, 35
 time, 26, 90, 105, 132
 trade-off between, 61
 eigenmodes, 54
 eigenvalue decomposition, 54, 58, 81, 128
 eigenvector, 54, 59, 92
 leading, 111
 normalized, 93
 empirical data, 55, 58, 116
 evapotranspiration, 45, 146

F

factor
 deviation, 137
 loading, 58, 62
 scaling, 29
 weight, 92
 finite-difference, 28, 63
 approximation, 63, 68, 94
 block-centered, 47
 discretization, 29, 134
 equation, 51
 flow reactor, 55
 fluid
 dynamics, 55, 113
 source term, 57, 91

G

Galerkin
 method, 113, 132, 136
 projection method, 55, 63, 65, 93
 gradient
 adjoint-based, 119
 average, 33
 head, 29
 hydraulic, 34
 loop, 95, 104
 normal component, 51
 of objective function, 94
 pressure, 49
 tangential component, 51
 transmissivity, 31
 Gram-Schmidt procedure, 82
 grid-centered scheme, 30

H

heptadiagonal, 64, 134
 Hessian, 95, 120, 128
 homogenized, 27, 29
 hydraulic transmissivity, 34
 hydraulic
 conductance, 30
 gradient, 34
 head, 28, 48
 transmissivity, 28, 47
 parameters, 27
 permeability, 32
 tensor, 28

I

image processing, 55, 113
 information scale, 26
 intrinsic
 periodicity, 38
 property, 29, 34
 inverse, 69
 modeling, 21, 90, 94, 112
 iteration cycle, 51, 115, 120, 128, 134, 149

J

jacket cell, 33
 Jacobian, 118

K

Kalman Filter, 110
 ensemble, 110
 reduced order, 87
 Karhunen-Loève decomposition, 133
 Krylov
 sequence, 81
 Krylov space, 82

L

Lanczos method, 54, 80
 laser altimetry, 26, 44, 146
 Latin Hypercube Sampling, 97, 125, 137
 line search, 95, 120, 143
 linear
 approximation, 109, 111
 autoregressive model, 55
 boundary condition, 139
 coefficients, 50

- combination, 18, 115
- coordinate transformation, 63, 117
- function, 63, 92
- interpolation, 35, 49
- model, 66, 137
- relationship, 57, 100, 103
- system, 43, 58, 110
- linearized
 - model, 111
 - reduced model, 114
 - state vector, 114
- lower-bound method, 30

M

- mean
 - absolute error, 37
 - arithmetic, 30, 32
 - geometric, 35
 - harmonic, 30, 32, 39, 64
 - pressure, 49, 77
 - relative error, 46
 - square error, 55, 59
- minimum, 95
 - global, 100
 - local, 101, 145
 - sub-optimal, 143
- model
 - adjoint, 110, 113, 135
 - autoregressive, 55
 - reduced, 118, 141
 - behavior, 73, 91, 121, 137, 140
 - calibration, 110
 - coarse-scale, 30
 - coarsened, 39, 47
 - digital-terrain, 26
 - discrete, 112, 134
 - domain, 27, 32, 40, 45
 - local, 35
 - enormous, 26
 - fine-scale, 31
 - local, 40, 42
 - full-dimensional, 71
 - groundwater flow, 44
 - low-dimensional, 57
 - mathematical, 54
 - morphological, 55
 - networks, 90
 - numerical, 26, 54, 110
 - parameter, 27, 110, 112

- predictions, 72
- reduced, 27, 60
 - adjoint, 118, 119, 129, 139, 141
 - formulation, 63, 93, 117
 - incremental, 138
 - methodology, 57, 113, 136
 - perturbation, 116
 - purpose of, 70
 - structure, 57, 92, 114
- reduction, 111
- scheme, 30, 82
- simulation, 26, 111
- solution, 27
- space, 58, 116
- steady-state, 58
- subspace, 111
- supra-regional, 69
- surface-water, 140
- transient, 58
- transport, 75
- variance, 60, 95, 123
- Modflow, 30, 48, 51
- Multi Gaussian random field, 37, 38, 46
- multivariate
 - data, 18
 - methods, 57

N

- Neumann condition
 - displacements, 42, 47
 - upscaling of, 42
- Newton iteration, 58
- node-centered scheme, 30
- nonlinear, 43
 - boundary condition, 44, 146
 - conductance matrix, 142
 - equation, 55
 - function, 43
 - inverse modeling, 146
 - synthetic cases, 127
 - system, 58
- Noord-Brabant, 44, 69, 146
- numerical
 - aspects, 68
 - dispersion, 31
 - dissipation, 32
 - errors, 30, 74
 - experiences, 100
 - ill defined, 69

point of view, 55, 63
 solver, 51
 subspace, 60

O

objective function, 90, 92, 94, 134
 approximate, 94, 136
 cross section, 98
 surface, 98
 user-defined, 106
 observation, 92
 estimated, 112
 space, 112, 133, 135
 synthetic, 96, 123
 technique, 26
 well, 101, 143
 observational
 behavior, 146
 variance, 144
 weight, 112
 matrix, 134, 143
 ordinary differential equation, 63, 83, 92, 136
 orthogonal
 decomposition, 55, 57, 113, 132
 function, 57, 90
 rotation, 62
 transformation, 82
 orthogonal regression, 59
 orthonormal, 60, 93, 116

P

partial differential, 114, 118
 partial differential equation, 29, 55, 57, 86, 91, 119,
 133, 138, 139
 partial orthogonalization, 82
 participation factor, 82
 pattern, 54, 59
 alternative selection, 61
 analytical, 54
 derivative of space, 66, 93, 107, 138
 identification, 57
 interpretability, 62
 loop, 95, 100
 number of, 60
 rotated, 62
 vectors, 93
 periodic
 boundary conditions, 34, 51, 52
 medium, 38

replications, 34
 structures, 40

phase

preparation, 56, 57, 70
 reconstruction, 56
 simulation, 56, 57, 74
 single, 28

physical processes, 62, 73

Picard iteration, 43, 58, 134, 141

posterior estimate, 119, 120, 126, 137, 142

precipitation, 42, 45, 58, 70, 71, 103

Principal Component Analysis, 18

prior

estimate, 43, 117, 123, 125
 projection, 63, 95
 to an inverse, 69

problem

adjoint, 110
 calibration, 110
 control, 111
 dimension, 54, 57
 eigenvalue, 58, 116
 heat transfer, 55
 inverse modeling, 89, 91, 146
 local, 35
 management, 26
 minimization, 35, 92, 111, 125, 127
 pollution, 26
 regional water, 26
 scale, 26
 synthetic, 47, 96, 100, 101
 three-dimensional, 102
 transport, 31
 projection, 64
 matrix, 67, 93, 94, 116, 136
 technique, 56
 time-dependent, 65
 pumping strategy, 54, 56

Q

quasi-3D flow, 29

Quasi-Newton, 95, 120, 143

R

rank, 111
Rayleigh-Ritz, 83, 113
 procedure, 57
reconstructor, 64
recursive, 81
 algorithm, 58
reductor, 64
regression, 80
Representative Elementary Volume, 35, 39
reservoir
 engineering, 27
 simulator, 49

S

satellite images, 26, 45, 146
sealed-off boundaries, 33, 34, 38, 41
simple structure, 62
snapshots, 56, 58, 70, 96, 103, 113, 137, 140, 142
 boundary value, 96, 100
 distortion vector, 137
 series, 137
 simulation loop, 95, 104
 variance, 101
 vector, 58
solver
 iterative, 43, 141
 numerical, 51
 time-efficient, 27, 90, 132
span, 55, 61, 68, 115, 116
spectral methods, 54, 57, 113, 132
standard deviation, 31, 46
Staphorst, 102
Stokes equation, 29
subspace, 58, 61, 91, 115, 116, 137

 controllable, 111
 extended, 93
 observable, 111
superfluous, 67, 94, 113, 138
superposition, 55, 58, 70, 137

T

time
 interval, 97
 series, 78
twin experiments, 143

U

uniform
 density, 57, 91
 flow, 31, 32
 type of boundaries, 34
upper-bound method, 30
upscaling, 27
 boundary conditions, 40
 factor, 29
 geology, 27
 hydraulic parameters, 32
 methods, 27
 non-uniform, 27
 uniform, 27

V

Variable Metric Method, 95, 120
variational method, 113
Varimax rotation, 62

Z

zone, 90, 96

Curriculum vitae

PETER VERMEULEN werd geboren op 8 maart 1970 in Breukelen. In 1986 behaalde hij zijn HAVO diploma aan de Thorbecke Scholengemeenschap te Arnhem. Vervolgens ging hij Tropische Cultuurtechniek studeren aan de International Agrarische Hogeschool Larenstein IAHL te Velp. In 1991 doorliep hij in een periode van 6 maanden een tropenstage in Tanzania. Hier verrichtte hij, voor de Stichting Nederlandse Vrijwilligers SNV, in het kader van het onderzoeksprogramma Small-Scale Irrigation Program Dodoma SSIPDO, een uitgebreide studie naar het landgebruik en eventuele mogelijkheden voor irrigatie in het dorp Chamkoroma. Na terugkomst verzette hij zijn bakens, om in het laatste jaar van zijn studie, zich te richten op Hydrologie. Zijn afstudeerwerk, getiteld *Pompen of vernatten, effectenstudie in de Barsbeker Binnenpolder*, werd verricht bij de toenmalige Waterleiding Maatschappij Overijssel WMO (nu Vitens B.V.). In juli 1992 ontving hij zijn diploma, waarna hij zijn verplichte 'activiteiten' van de Militaire Dienstplicht, combineerde met werkzaamheden voor de toenmalige afdeling Grondwater en Geo-Energie TNO-GG. Aansluitend hieraan werkte hij een half jaar voor de Meetkundige Dienst van Rijkswaterstaat te Delft, waarna hij in 1993 een vaste aanstelling kreeg bij het TNO-GG. Na een aaneengesloten periode van 7 jaar verrichtte hij verschillende lokale/regionale modelstudies waarna het TNO-NITG (fusie tussen TNO-GG en de Rijks Geologische Dienst RGD) hem de kans bood om zijn kennis verder te verdiepen. In maart 2001 is hij hierom in dienst getreden als promovendus aan de faculteit Electrotechniek, Wiskunde en Informatica van de Technische Universiteit Delft. Sinds maart 2005 is hij wederom teruggekeerd als wetenschappelijk onderzoeker/projectleider bij de Business Unit Bodem en Grondwater van de divisie TNO Bouw en Ondergrond te Utrecht.