

10-15-2009

# Model Reduction for Simulation, Optimization and Control

Oleg Edward Roderick  
*Portland State University*

Follow this and additional works at: [https://pdxscholar.library.pdx.edu/open\\_access\\_etds](https://pdxscholar.library.pdx.edu/open_access_etds)



Part of the [Mathematics Commons](#)

Let us know how access to this document benefits you.

---

## Recommended Citation

Roderick, Oleg Edward, "Model Reduction for Simulation, Optimization and Control" (2009). *Dissertations and Theses*. Paper 5984.

<https://doi.org/10.15760/etd.7854>

This Dissertation is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: [pdxscholar@pdx.edu](mailto:pdxscholar@pdx.edu).

DISSERTATION APPROVAL

The abstract and dissertation of Oleg Edward Roderick for the Doctor of Philosophy in Mathematical Sciences were presented October 15, 2009, and accepted by the dissertation committee and the doctoral program.

COMMITTEE APPROVALS:




Dacian N. Daescu, Chair



Gerardo A. Lafferriere



Bin Jiang



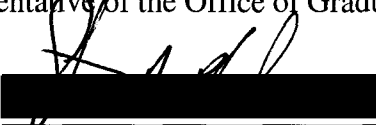
Rolf Könenkamp



Dean Atkinson

Representative of the Office of Graduate Studies

DOCTORAL PROGRAM APPROVAL:



Steven A. Bleiler, Director  
Mathematical Sciences Ph.D. program

## ABSTRACT

An abstract of the dissertation of Oleg Edward Roderick for the Doctor of Philosophy in Mathematical Sciences presented October 15, 2009.

Title: Model Reduction for Simulation, Optimization and Control

Many tasks of simulation, optimization and control can be performed more efficiently if the intermediate complexity of the numerical model is reduced. In our work, we investigate model reduction, as applied to reaction-transport systems of atmospheric chemistry. We use a Proper Orthogonal Decomposition-based approach to extract information from a set of model observations, and to project the model equations onto a reduced order space chosen in such a way that the essential model behavior is preserved in the solution of the reduced version. We examine and improve many features of the method. In particular, we show how to measure sensitivities of the model reduction process, and use the results to select the placement and weighting of observations to best reproduce specific events in the full model behavior; we also develop novel techniques allowing to take into account multiple events. We show how to construct reduced models to replace the full model in iterative parameter optimization procedures so that fewer steps and lower computational budget are needed. The result of the study is a more complete understanding of how to perform tasks of simulation and optimization of nonlinear models using model reduction tools.

MODEL REDUCTION FOR SIMULATION,  
OPTIMIZATION AND CONTROL

by

OLEG EDWARD RODERICK

A dissertation submitted in partial fulfillment of the  
requirements for the degree of

DOCTOR OF PHILOSOPHY  
in  
MATHEMATICAL SCIENCES

Portland State University  
2009

## ACKNOWLEDGEMENTS

I would like to thank my parents, Irina and Ric Roderick for advice and support over the years of graduate school, and for providing me with inspiring personal examples of reliability and persistence. I am grateful to Alisa Vaynrub, the love of my life; our commitment to each other has given meaning to my work of the last few years. It has been a pleasure and an honor learning from Dacian Daescu, my adviser at Portland State University, an excellent applied scientist, and a good teacher. I am equally grateful for an opportunity to work with Mihai Anitescu, my supervisor at Mathematics and Computer Science Division, Argonne National Laboratory. His exceptional competence, erudition, creativity and intellectual curiosity make him the person I learned from the most in this early stage of my career. I am grateful to the members of the Ph.D. committee Gerardo Lafferiere, Bin Jiang, Rolf Koenkamp and Dean Atkinson for the insightful suggestions that helped to improve this work, and also to the department chair Marek Elzanowski for his advice and support. I would also like to mention a number of friends, all of them young specialists in their fields, that helped form my interest in interdisciplinary applied studies: Alexei Soldatov, Ilya Safro, Anna Kelbert, Anton Dragunov and Sergey Jukovskiy. In addition to contributions from the wonderful people I mentioned, this work was made possible by financial assistance provided by the University and the Laboratory.

## CONTENTS

Acknowledgements	...	i
List of tables	...	iv
List of figures	...	v
Notation	...	viii
1. Introduction	...	1
2. SVD-based model reduction	...	16
2.1 Method of snapshots	...	19
2.2 Sensitivity analysis	...	29
2.2.1 Derivative information	...	30
2.2.2 Sensitivity by interpolating models	...	39
2.3 Selective model reduction	...	46
2.4 Improvements on the method of snapshots	...	53
2.4.1 Weighting and metric change, event targeting	...	55
2.4.2 Snapshot placement	...	63
2.4.3 Additional suggestions	...	66
3. A posteriori error estimation	...	81
3.1 Error induced by perturbation	...	82
3.2 Error induced by model reduction	...	86
4. Adjoint analysis	...	92
4.1 Differentiation of an ODE model	...	94
4.2 Differentiation of a PDE model	...	98

5. Optimization	...	102
5.1 Descent direction methods	...	105
5.2 Reduced model in optimization	...	112
5.2.1 Single reduced model	...	114
5.2.2 Multiple reduced models	...	120
6 Numerical tools	...	125
6.1 Preprocessing	...	126
6.2 Model integration	...	129
7 Examples	...	137
7.1 Stratospheric chemistry mechanism	...	139
7.2 Test optimization problem	...	143
7.3 ‘Brusselator’ model	...	149
7.4 Molenkamp-Crowley problem	...	159
7.5 Lorenz model	...	167
7.6 Charney-DeVore model	...	184
7.7 SAPRC-99 model	...	192
7.7.1 Examination of the chemical mechanism	...	193
7.7.2 Reduction of the chemical model	...	206
7.7.3 Measurement of the reduced model performance	...	221
8. Conclusions	...	233
References	...	237

## LIST OF TABLES

8.1 List of model species used in SAPRC-99 mechanism	...	202
8.2 Inspection of SAPRC-99 model dynamics	...	204
8.3 Effectiveness of reduced model for perturbed initial conditions	...	226
8.4 Effectiveness of reduced model: reproduction of individual species; 2.5% perturbation	...	227



## LIST OF FIGURES

1.1 Illustration of the data assimilation process	...	15
2.1 Targeting an event with two types of weighting	...	80
5.1 Model reduction in an iterative search	...	124
7.1 Chapman-like mechanism, performance of the reduced model	...	142
7.2 Convergence of the full and reduced optimization searches, problems of dimension 10, 20, 30 with random parameters	...	148
7.3 First 50 eigenvalues of the ‘Brusselator’ model covariance matrix		154
7.4 Relative error for the reduced version of the ‘Brusselator’ model for different dimensions	...	155
7.5 Comparative performance of the full and reduced solutions for the ‘Brusselator’ model	...	156
7.6 Comparative performance of the full and reduced solutions for the ‘Brusselator’ model; non-uniform placement of snapshots	...	157
7.6. Comparative performance of the full and reduced solutions for the ‘Brusselator’ model; iterative recovery of correct initial conditions		158
7.7 First 50 eigenvalues of the Molenkamp-Crowley model covariance matrix	...	163
7.8 Comparative performance of the full and reduced solutions for the Molenkamp-Crowley model	...	164
7.9 Performance of a reduced solution for the Molenkamp-Crowley model: snapshots taken from the exact rotation of the profile	...	165

7.10 Comparative performance of the full and reduced solutions for the Molenkamp-Crowley model: iterative recovery of correct initial conditions, step 10	166
7.11 First 40 eigenvalues of the Lorenz model covariance matrix	178
7.12 Comparative performance of the full and reduced solutions for the Lorenz model	179
7.13 Reduced Lorenz model, first-order sensitivity information	180
7.14 Full Lorenz model, first-order sensitivity information	181
7.15 Weighted reduction of the Lorenz model: performance of the model; snapshot weighting and placement scheme	182
7.16 Weighted reduction of the Lorenz model: performance of the model; snapshot weighting chosen by polynomial interpolation	183
7.17 Comparative performance of the full and reduced solutions for the Charney-DeVore model, stable setup: fast transient period not shown	189
7.18 An example of the Charney-DeVore model solution trajectories: two distinct steady states	190
7.19 Performance of the reduced solution in reproducing the chaotic features of the Charney-DeVote model: behavior during transient stage; placement of the attractors	191
7.20 SAPRC99 solution, 72 hours	200
7.21 SAPRC99 right-side Jacobian sparsity pattern	201
7.22 First 50 eigenvalues of the SAPRC99 covariance matrix	215

7.23 Performance of the reduced model solution for SAPRC99 model: unmodified reduction setup ...	216
7.24 Fast manifold of the SAPRC99 model: distribution over time and Components ...	217
7.25 Performance of the reduced model solution for SAPRC99 model: no snapshots during transient intervals ...	218
7.26 Performance of the reduced model solution for SAPRC99 model: slow manifold targeting ...	219
7.27 Performance of the reduced model solution for SAPRC99 model: selective model reduction ...	220
7.28 Relative error for the reduced SAPRC-99 model for different dimensions ...	229
7.29 Relative error for the reduced (slow manifold targeting) SAPRC-99 model for different dimensions ...	230
7.30 Effectiveness of of reduced model for perturbed initial conditions: error variability in reproduction of species of interest ...	231
7.31 Effectiveness of reduced model: error variability in reproduction of individual species; 2.5% perturbation ...	232

## NOTATION

$u$	model state
$t, t_i, T, \tau$	time
$x, x_i$	spatial coordinates
$p \in P$	parameters, in the parameter space
$f(u, t, p)$	right-side term of an ODE
$\Omega$	spatial domain
$\partial\Omega$	spatial domain boundary
$u_0$	initial model state
$t_0$	initial integration time
$F(u, t, p)$	differential-algebraic operator of the reaction-transport PDE
$\mathfrak{S}$	output function, cost function, quantity of interest
$n$	model state dimension
$k$	reduced model state dimension
$\Phi, (\phi_1, \phi_2, \dots)$	subspace basis
$\Pi$	projection to subspace
$\hat{u}$	reduced model state
$q, q_i$	coordinates of reduced model state in the subspace
$e, E, \theta, \Theta$	error
$J$	right-side expression Jacobian
$H$	right-side expression Hessian

$U_o, [u(t_1), u(t_2), \dots]$	snapshots, model state observations
$N$	number of snapshots, observations
$\mu$	mean of the snapshot ensemble
$C$	correlation matrix of the snapshot ensemble
$\lambda_i$	eigenvalues of the correlation matrix
$u^*$	adjoint variable
$\Gamma$	interpolating polynomial model
$M^S, M^F$	manifolds with slow and fast dynamics
$g, g^r$	repaired vector with unreliable data
$\langle \cdot \rangle$	vector inner product
$\  \cdot \ , \  \cdot \ _\Lambda$	vector, or matrix norm
$\Lambda$	matrix used to define metric
$W, (w_1, w_2, \dots)$	snapshot weights
$\nabla \mathfrak{J}, \nabla_p \mathfrak{J}$	output function gradient
$d, d^{(k)}$	direction vector in iterative optimization
$\alpha, \alpha^{(k)}$	step size in iterative optimization
$p, p^{(k)}$	intermediate parameter value in iterative optimization
$p_{\min}$	optimal (minimizing) parameter value
$\hat{\mathfrak{J}}$	reduced model output function

## CHAPTER 1

### INTRODUCTION

The main theoretical motivation of our study is the existence of basic, sometimes well-studied problems of applied mathematics with a numerical solution that becomes very computationally expensive with the increase in the size of the problem. Colloquially, the problems “do not scale well”: the increased number of dimensions, degrees of freedom, or points in the discretization grid results in too many intermediate variables and operations for a numerical solution to be obtained within an available computational budget.

There is, however, a possibility that the intermediate complexity of such problem is not necessary to obtain the answer. The list of variables, in particular, can be reduced, due to redundancy (many of the variables are strongly correlated to each other through linear combinations or a general functional dependency), or irrelevance (many of the variables have an influence on the answer that is smaller than the required precision). Correspondingly, the inputs, intermediate parameters, and the solutions of the involved equations can be limited (at least approximately) to manifolds of much smaller dimension than the spaces declared in the definition of the problem. Then numerical solutions can be directly improved through the combined use of *factor importance analysis* (to decide which features of the problem are negligible), and *model reduction* (to replace the complete problem with a simplified version).

Model reduction can be treated as projection of the data set, extraction of statistical information, data compression, or a form of factor importance analysis. Altogether, there are 8-10 distinct approaches to reduction [38]. We use a process based on projection of the model dynamics onto a reduced dimension subspace chosen to best capture the observed information on the full model behavior at a selection of time instances. The method we used to select the subspace is based on Proper Orthogonal Decomposition, POD (a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables). The correlation matrix for the POD is based on a selection of observed model states. This approach has appeared in areas such as image processing [36], fluid dynamics [115], acoustics [68], circuit development [90], behavioral science [93].

In our study, we extend the POD-based model reduction procedure to include traditional and novel tools for an improved representation of various features of the full model dynamics in the reduced model. We perform factor importance analysis (i.e ranking of variables, or data components by importance in the context of a particular output) using first-order sensitivity information, and elements of sampling-based statistical learning. In addition, we perform factor importance analysis on the reduction process itself, obtaining the sensitivity information that was not available explicitly previously. As an additional contribution to the field of study, we show how model reduction may be used to improve iterative solution methods for model-constrained optimization problems.

We shall now describe our applied area of interest in more detail; and then overview the organization of the thesis.

Our specific models of interest are the reaction-transport systems that arise in the study of atmospheric chemistry processes and air pollution forecasting. The main subjects of study are large-dimensional ODEs modeling chemical processes. We assume the solutions to be smooth with respect to such system parameters as the initial conditions, with no bursting behavior. The chemical reaction ODEs may be augmented by simple transport terms, resulting in advection-diffusion-reaction PDEs. The problems of this class appear in many areas of applied industrial significance, and the mathematical content of our work can be extended to complex problems of other forms.

Some research was performed recently on improved model reduction, associated error estimation and sensitivity analysis (Petzold et. al. [54], [85]; Willcox et. al. [9], [19]), but many technical questions remain unanswered. To our knowledge, the application of model reduction to optimization problems of atmospheric chemistry was not examined.

In a general framework, we consider a dynamical system modeled by a parameter-dependent system of ordinary differential equations:

$$\begin{aligned} \frac{du}{dt} &= f(u, t, p) \\ u(t_0) &= u_0(p) \end{aligned} \tag{1.1}$$



where  $n$  is the dimension of the system;  $u = (u_1, u_2, \dots, u_n)^T$  is the state of the model, that is, a vector of individual chemical species concentrations  $u_i$ ;  $f(u, t, p)$  is the chemical reaction term, and initial conditions  $u_0(p)$  are dependent on the parameters  $p = (p_1, p_2, \dots, p_m)$ .

If the parameters are time-independent, the equations (1.1) can be reformulated without the loss of generality so that that parameters only appear in the initial conditions. The reformulation can be done by appending all the parameters appearing in the expression  $f(u, t, p)$  to the list of variables  $u$ . For an appropriately redefined term  $f$  and the list of parameters  $p$ , the ODE (1.1) is written as

$$\begin{aligned} \frac{du}{dt} &= f(u, t) \\ u(t_0) &= p \end{aligned} \tag{1.2}$$

The full model (including transport effects) is based on a generic scalar transport equation

$$\frac{\partial u}{\partial t} + \nabla \cdot \varphi(x, u, t, \nabla u) = \mathcal{S}(x, u, t) \tag{1.3}$$

where  $\varphi$  is called the *flux*, and  $\mathcal{S}$  the *source*. The advection-diffusion-reaction model is a particular case of (1.2), with advection and diffusion taken into account in the flux term, and chemical reactions included in the source term. The convection vector field  $\omega$  and a diffusion coefficient matrix  $K$  are generally allowed to depend on time  $t$  and spatial variable  $x$ , but not on the model state  $u$ . The system of equations is written as follows:

$$\begin{aligned}
u &= u(x, t) \\
\frac{\partial u}{\partial t} &= -\nabla \cdot (\omega u) + \nabla \cdot (K \nabla u) + f(u, t) \\
u(x, t_0) &= p
\end{aligned} \tag{1.4}$$

for  $x \in \Omega \subseteq R^3$ ,  $t \geq t_0$ , with the appropriate boundary conditions on  $x \in \partial\Omega$ .

The described system is used in the studies of atmospheric pollution. The vector  $u$  lists concentrations of chemically active species, such as ozone, nitrogen oxides, hydrocarbons and radicals. The wind patterns are described by  $\omega$ , hence the spatial dependence. The chemical reaction term  $f$  may include emissions and depositions, and is typically quadratic with respect to components of  $u$ . Stiff transients in the time evolution of the model state are related to daylight cycle and photolytic reactions.

The computational task of solving the advection-diffusion-reaction system essentially consists of integrating a very large ODE. A standard approach follows the method of lines. The system of partial differential equations (1.4) with the chosen boundary conditions is discretized on a fixed spatial grid (Eulerian; uniform, adaptive, or related to geographical features). The resulting system of ODEs, with an explicitly available, sparse Jacobian is then passed to a numerical solver.

Technical difficulties arise due to stiffness (time derivative of  $u$  has components that may vary by several orders of magnitude); and a large dimensionality of the system. This has led to the use of special time integration techniques (time or operator splitting, implicit-explicit methods, approximate matrix factorization approaches) [116].

Meaningful problems could have a number of species in the 30-100 range, and a discrete state vector of the size on the order of  $10^7$  due to the size of the spatial grid. For example, a family of General Circulation Models for global weather prediction uses horizontal resolution of down to 250 kilometers, and up to 30 horizontal layers, resulting in about 800,000 grid points. One typical benchmark test of GEOS-chem [129], [130], a global model of atmospheric composition includes 350 reactions with 90 chemical species, 30 horizontal layers with 6500 grid points in each, and simulated a time interval of 1 year. Even for the smaller problems that we use as examples, and modest requirements on accuracy, the number of grid points can exceed 10,000.

The usual tasks associated with large models include prediction of the future behavior; recovery of the true state of the system based on the incomplete observations; inverse problems such as recovery of the parameters that lead to a particular state of the system; and analysis of sensitivity of the problem solution to changes in the components and parameters. We are particularly interested in the parameter optimization problem in the context of data assimilation.

The subject of data assimilation in atmospheric science is well-described in [59]. In general, data assimilation is a process of estimation of a true state of the system based on (imprecise) observational or simulated data. Some form of data assimilation is required in all environmental sciences, studies of ocean dynamics, and weather prediction. The idea is to use the existing actual observations of the environment to gradually adjust the values of the parameters until the model is

stable and consistent with the available data over shorter periods of time, then use it for long-term predictions. The process is multi-step, and there may be many criteria of reliability that a model has to satisfy.

In Figure 1.1 we provide a simplified visualization of the assimilation process. At each step of the cycle, a current estimate of correct parameters is obtained as a solution to an optimization problem. Our main research interest is the following basic form of this optimization problem. Given a general model

$$F(u, t, p) = 0 \tag{1.5}$$

with a particular example given by (1.4), find such values of parameters  $p$  that the difference

$$\mathfrak{S} = \|u(x, t) - u_o(x, t)\| \tag{1.6}$$

between the simulated state of the system  $u(x, t)$ , and the observed state  $u_o(x, t)$  is minimal in some appropriate norm  $\|\cdot\|$ . In other words, the task is to fit the model parameters (in our case, model initial conditions) to observations.

The computational difficulty of the optimization problem depends on the number of parameters and the complexity of the underlying model. As stated in (1.4), the complex dependence on parameters is present only in the reaction part of the model. We mean to use a very simple description of transport, so that the large size of the grid merely amplifies the computational cost of the ODE. Therefore, we shall primarily study the behavior, sensitivities, and opportunities for complexity reduction for the model (1.1), and then apply the results in the context of the PDE.

For many chemical systems with reactions and transport, the large dimension of the equations (1.1) does not reflect the true number of degrees of freedom of the model. The chemical dynamics can be simplified: it may be done even as the reaction equations are derived. The components that produce a small overall effect on the state of the system could be partially absorbed ('data lumped', [80]) into the constructed state variables, and partially neglected. The same could be done for the components that remain almost constant, for the components that oscillate rapidly around some mean value, and for components that (due to various reasons) are not described reliably by the numerical model. Because of the simplification, the precision of the simulation will suffer, but we may be allowed a moderate error anyway, because in practice the equation parameters and the initial conditions are deduced from the already imprecise observations.

Any model reduction method ranks either the involved state components, the interactions between the state components, or the particular times in the evolution of the system by importance, and eliminates the less important ones from the system. There is a variety of ways to reduce complexity suggested in the literature: see [38], [33], [45], [52] for the overview. They include approaches based on the experimental insight, omission of components chosen by sensitivity analysis, omission of components chosen by a greedy algorithm that minimizes the loss of quality after reduction, and simplification of the model equations. In addition, for linear ODEs, the optimal control theory offers a number of additional techniques, such as Hankel reduction and balanced truncation (see [2], [3], [103], [114]).

In this study, we simplify the dynamics of the model by projecting the equations onto a reduced subspace. A model reduction process results in a reduced system of (ordinary differential, partial differential, or algebraic) equations with the solution  $\hat{u} \in S \subset R^n$ , that is an approximation of the solution  $u \in R^n$  of the full system. Informally, we shall refer to  $u, \hat{u}$  as the *full*, and the *reduced* versions of the same mathematical model. The reduced system solution  $\hat{u}$  evolves in an optimal subspace  $S$  of dimension  $k < n$ :

$$S = \text{span}(\phi_1, \phi_2, \dots, \phi_k) \tag{1.7}$$

where the basis vectors  $\phi_i$  are chosen so that the important features of the behavior of the full model are preserved in the reduced model. The characterization of this subspace is the essence of model reduction.

We expect the reduced system to be optimal in some sense. Due to a wide variety in the models' behavior, and its reproduction, general descriptions of what we expect as a result of model reduction are not effective. The practical definitions of a good quality of reduced system need to be goal-oriented. The basic requirement is that the error  $e = \|u - \hat{u}\|$  is minimal in some norm  $\|\dots\|$ .

We may use a problem-specific definition of the norm, even allow the error (1.7) to be sub-optimal, though still reasonably small, in the cases when the reduced model best satisfies some problem-specific requirement on the behavior of the solution. Such a requirement may consist of a faithful reproduction of some *output*  $\mathfrak{S}$  (also known as *cost*, *merit function*, or the *quantity of interest*):

$$\begin{aligned} \mathfrak{I}: R^n &\rightarrow R \\ \mathfrak{I}(u) &\approx \mathfrak{I}(\hat{u}) \end{aligned} \tag{1.8}$$

A more advanced treatment of the subject would also call attention to a number of additional features of analytical, physical or algebraic nature, such as the regularity of the solution: smoothness, existence of bounds on the numerical values of the derivatives; boundedness and positivity of the state components, conservation of mass; the preservation of periodicity and other symmetries. We will be mostly interested in optimal reproduction of the output (1.8); other features in the reduced model behavior can be addressed as long as they can be quantified.

Our practical requirements for a numerical solution of the reduced model are relatively high computational speed and numerical stability. The achieved numerical advantage over the use of the full model can be partial, leading to reduction only in some model components, only over some regions of space, or only over some intervals of time. In our test examples, we encounter a number of scenarios with non-ideal, but acceptable performance:

- Model reduction decreases the dimension of the problem significantly, perhaps by over 90%; the error introduced by reduction is low on a certain time interval, then starts to deteriorate.
- Model reduction decreases the dimension of the problem by 50-90%. The error in almost every individual component is small, but in some components it accumulates rapidly: all the time, or perhaps just over some intervals of time.

- Model reduction decreases the dimension of the problem by 50-90%. The error is generally small, but the reduced model does not reproduce the behavior over some time intervals correctly, leading to instability or unacceptably large error in any integration that includes the problematic intervals.

When deciding whether it is advantageous to use model reduction at all, we have to take into account the computational cost of creating the reduced model, and the fact that sparsity of the full model is necessarily lost in reduction. The error and sensitivity analysis of the reduced model also carry an additional computational cost. We shall allow the analytic and pre-processing tools to take a significant time, as long as they are to be applied only once. The main strength of our work lies in the problems where the constructed reduced model is then re-used multiple times for different sets of parameters.

The main content of our work is an extensive examination of features of the existing POD-based approach to model reduction. We introduce specific improvements for many aspects that previously were only abstractly characterized as important for the process. In particular, we explain how to use the results of goal-oriented model factor importance analysis to select the weighting, snapshot placement, and metric for POD-based reduction, and develop techniques for basis selection that take into account the different behavior of distinct model components under reduction.



We use such tools as adjoint differentiation, analytic differentiation of linear algebra procedures, and high-order interpolation to collect information on the sensitivity of the full and the reduced models that is inaccessible by simpler techniques of factor importance analysis. We revise an existing approach to error estimation in the projected systems, and derived an additional estimate, taking into account both the errors introduced by perturbation of the model inputs, and the errors introduced by reduction. We use model reduction to improve the computational efficiency of the descent optimization methods applied to initial conditions recovery problems. We implemented reduction-based optimization for a number of small test examples, and for a larger atmospheric chemistry model.

Our numerical examples show that the developed techniques may be applied to both test models with basic transport and interaction effects, and the atmospheric chemistry models of high complexity (chemical mechanism SAPRC-99 is used as a central example).

All theoretical results that are required for implementation of numerical experiments are given in sufficient detail for the readers to reproduce and modify the procedures according to their needs. This led to a number of technical remarks not directly related to the topic of model reduction: for example, the description of conjugate gradient methods, or the procedure for obtaining the positive definite matrix from unreliable data. On the other hand, if we do not improve, or study the properties of a particular procedure, and the reader is likely to use a standard

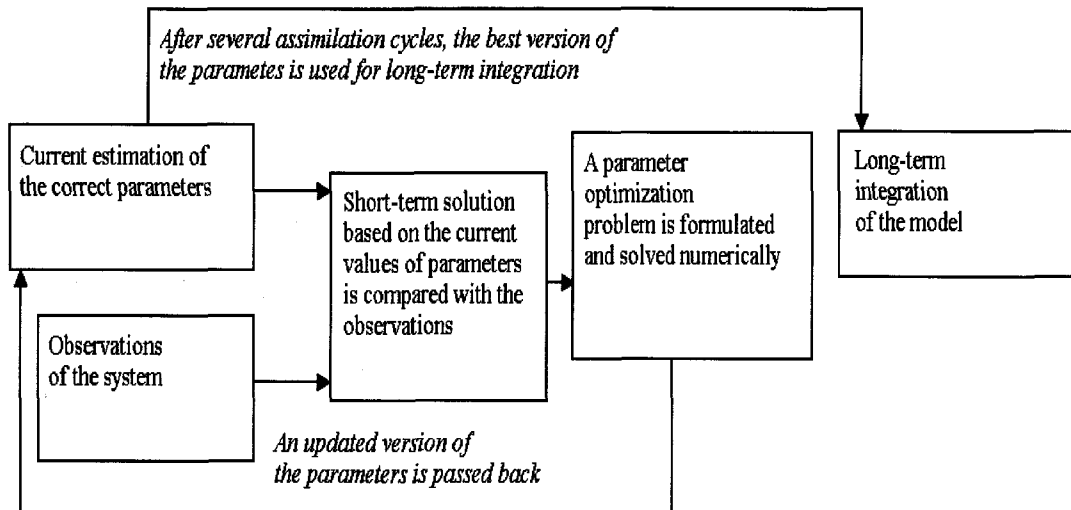
implementation, we do not present a complete description. For example, the topic of adjoint differentiation is given without a full reference to Hilbert space theory.

In some calculations, the number of variables, the dimensions of the arrays, and the indexing of summations are not specified: usually because there is more than one possibility, or the expression is not meant to be evaluated completely. In such cases, we write that the expression is ‘schematic’.

Now that we have provided some comments on the nature, scale and relevance of our central problems, the rest of the thesis material is organized as follows:

- In Chapter 2, SVD-based model reduction, we explain our approach to dimension reduction, perform sensitivity analysis of the reduction process, and introduce a number of goal-oriented improvements of the approach.
- In Chapter 3, A posteriori error estimation, we describe how to measure an error introduced by model reduction on the solution of the full model.
- In Chapter 4, Adjoint analysis, we explain how to efficiently differentiate the functional aspect of the model output using the adjoint system of differential equations.
- In Chapter 5, Optimization, we explain the use of model reduction in solving optimization problems.
- In Chapter 6, Numerical tools, we overview some of the standard numerical tools and practices for the problems of atmospheric chemistry.
- In Chapter 7, Examples, we apply the developed tools to a number of numerical models. In particular:

- In Section 7.1, Stratospheric chemistry mechanism, we pre-process a small chemical model and reduce the dimension from 5 to 2
- In Section 7.2, Test optimization problem, we solve a generic ODE-constrained optimization problem by iterative descent methods using full and reduced models.
- In Section 7.3, ‘Brusselator model’, we reduce a reaction-diffusion model by over 90%.
- In Section 7.4, Molenkamp-Crowley problem we discretize a transport-only problem, reduce it by over 80%, and solve an associated initial conditions optimization problem.
- In Section 7.5, Lorenz model, we apply our model reduction sensitivity analysis tools to a test model with advection and reaction effects.
- In Section 7.6, Charney-DeVore model, we present an example of dynamics that is in principle not correctly reproduced by model reduction.
- In Section 7.7, SAPRC-99 model, we apply reduction to a complex chemical mechanism, measure and improve the performance of the reduced model, and provide a tabulation of properties of chemical species in the context of model reduction.
- In Chapter 8, Conclusions, we summarize the main outcomes and findings of the performed work.



**Figure 1.1** Illustration of the data assimilation process

## CHAPTER 2

### SVD-BASED MODEL REDUCTION

The procedure of model reduction consists of constructing a truncated projection of the model equations onto a low-dimensional subspace in the space of model states. Let a full model be described by equations

$$\begin{aligned} F(u, t) &= 0 \\ F : R^n \times [t_0, T] &\rightarrow R^m \end{aligned} \tag{2.1}$$

with the solution  $u(x, t) \in R^n$ . The reduction in dimensionality of the system state is achieved by obtaining the solution  $\hat{u}(x, t) \in R^n$  as a linear combination of  $k < n$  basis vectors

$$\hat{u}(x, t) = \sum_i q_i(x, t) \phi_i + \mu = \Phi q + \mu \tag{2.2}$$

where  $\mu$  is the optional shift of the coordinate reference point. The basis of the reduced space is defined by the matrix

$$\Phi = [\phi_1, \phi_2, \dots, \phi_k] \in R^{n \times k} \tag{2.3}$$

the columns of which are the vectors  $\phi_i$ . The corresponding projection matrix of rank  $k$  is defined by

$$\Pi = \Phi \Phi^T \in R^{n \times n} \tag{2.4}$$

The reduced solution  $\hat{u}$  satisfies the projected version of (2.1)

$$\Pi F(\hat{u}, t) = 0 \tag{2.5}$$

Formally, the reduction is applied to the model state, and the modifications to the system of equations appear as a consequence. In principle, reduction of

complexity could also be performed on the algebraic structure of the right-side equations. We do not perform this additional reduction, since our main models were already pre-processed at the construction stage, with elimination of the insignificant, or the redundant elements that could be identified by inspection of the mathematical structure of equations.

The relations (2.2), (2.4), (2.5), after some algebraic simplifications, result in a system of equations for the coordinates  $q_i(x, t)$  of the reduced system state in the new basis. In particular, for a model described by a system of  $n$  ordinary differential equations

$$\begin{aligned} \frac{du}{dt} &= f(u, t) \\ u(t_0) &= p \end{aligned} \tag{2.6}$$

the reduced solution  $\hat{u}(t)$  satisfies the system of  $n$  equations

$$\begin{aligned} \frac{d\hat{u}}{dt} &= \Phi\Phi^T f(\hat{u}, t) \\ \hat{u}(t_0) &= \Phi\Phi^T(p - \mu) + \mu \end{aligned} \tag{2.7}$$

and the coordinates  $q_i$  of the expansion (2.2) are subject to the *reduced* system of  $k$  equations:

$$\begin{aligned} \frac{dq}{dt} &= \Phi^T f(\Phi q + \mu, t) \\ q(t_0) &= \Phi^T(p + \mu) \end{aligned} \tag{2.8}$$

An explicit expression for the Jacobian of the full system (2.6):

$$J = (J_{ij}) = \left( \frac{\partial f_i}{\partial u_j} \right) \tag{2.9}$$

leads to an expression for the reduced Jacobian of the system (2.8):

$$\hat{J} = (\hat{J}_{ij}) = \left( \frac{\partial \Phi^T f(\Phi q + \mu, t)_i}{\partial q_j} \right) = \Phi^T J(\Phi q + \mu, t) \Phi \quad (2.10)$$

The Jacobian matrix (2.10) of the reduced model is in general dense, although the Jacobian of the full system (2.9) may be sparse.

Note that the motivation to replace the full model with the reduced model for the tasks of simulation consists of two parts. First, equations (2.8) are of smaller dimension than equations (2.6), resulting in an improvement in integration time, and in the volume for the optimal initial conditions search. Second, equations (2.7) are expected to be structurally less complex than equations (2.6).

Once the basis  $\Phi$  is selected, the projected equations (2.8) are sufficient for reduction of a space-discretized PDE with an ODE term. Most of the rest of the material in this section will (directly or indirectly) concern the choice of the subspace basis  $\{\phi_i\}$ .

Given a general expectation that the reduced solution should be a high-fidelity reproduction of the state of the full model, and of its sensitivities, over a range of parameter values, we shall now introduce a projection that optimally (in the least squares sense) reproduces the state of the full model at some given time instances. We will then modify and enhance it. Our main tools are based on singular value decomposition, and closely related to such concepts as principal component analysis, covariance analysis, Karhunen-L  ve expansion, Hotelling transform of

stochastic process theory; and the principle of empirical orthogonal eigenfunctions of the interpolation theory.

## 2.1 METHOD OF SNAPSHOTS

Principal component analysis (PCA) is formally defined as an orthogonal linear transformation to a new coordinate system, such that the first coordinate axis is the direction of the greatest variance of the data by any projection, the second coordinate axis is the direction of the second greatest variance, and so on [80]. In the terms of PCA, model reduction by projection and truncations consists of keeping the first few components of the system state in new coordinates, and ignoring the rest. It is generally expected that the first few components will contain the most important aspects of the data. The degree to which this expectation is true depends on the specific problem, and the choice of the data set. The PCA approach to model reduction used to our work is the discrete version of proper orthogonal decomposition (POD), also known as the method of snapshots. We shall now introduce the method in sufficient technical detail, and explain in what sense this approach is optimal.

### **Singular value decomposition**

Given a (rectangular) matrix  $A \in R^{n \times m}$ ,  $n < m$ , of full rank  $n$ , we can find an approximation  $\hat{A} \in R^{n \times m}$  of rank  $k < n$  such that the error  $\|A - \hat{A}\|_2$  is minimal, using the singular value decomposition



$$A = \sum_{i=1}^n \sigma_i u_i v_i^T = U \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_n \\ & & & \vdots \\ 0 & & & 0 \end{pmatrix} V^T \quad (2.11)$$

where  $U \in R^{n \times n}, V \in R^{m \times m}$  are unitary orthogonal matrices consisting of column vectors  $u_i$  and  $v_i$  (known as *left* and *right singular vectors*);  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$  are the singular values of matrix  $A$ . Multiplying (2.11) by  $u_i, v_i$ , we obtain

$$\begin{aligned} Av_i &= \sigma_i u_i \\ A^T u_i &= \sigma_i v_i \end{aligned} \quad (2.12)$$

implying that  $u_i, v_i$  are eigenvectors of matrices  $AA^T$  and  $A^T A$  correspondingly, with eigenvalues  $\lambda_i$  satisfying

$$\lambda_i = \sigma_i^2 \quad (2.13)$$

The optimal low-rank approximation is defined by a truncated version of decomposition (2.10):

$$\hat{A} = \sum_{i=1}^k \sigma_i u_i v_i^T = U \begin{pmatrix} \sigma_1 & \dots & 0 & 0 \\ \vdots & \ddots & & \\ 0 & & \sigma_k & \vdots \\ 0 & & \dots & 0 \end{pmatrix} V^T \quad (2.14)$$

In practice, SVD is computed through a series of orthogonalization (QR) decompositions by the Gram-Schmidt process with reordering reduction. The error introduced by SVD is

$$E = \|A - \hat{A}\|_2 = \sum_{i=k+1}^n \sigma_i \quad (2.15)$$

We refer to [42] for derivation of the properties of SVD.

■

### Method of snapshots

Consider a set of  $N$  snapshots, or exact observations of the model at arbitrary times  $t_1, t_2, \dots, t_N$  (time instances do not have to be in any particular order).

The corresponding model states are organized as column vectors in the matrix

$$U_o = [u(t_1), u(t_2), \dots, u(t_N)] \in R^{n \times N} \quad (2.16)$$

The matrix

$$C = (U_o - \mu)(U_o - \mu)^T \quad (2.17)$$

is known as the *correlation matrix* of the data set (2.16) if  $\mu = 0$ , and the *covariance matrix* (of variability around the mean) if the term  $\mu$  is defined as the mean of the observed model states:

$$\mu = \frac{1}{N} \sum_{j=1}^N u(t_j) \quad (2.18)$$

Subtraction of the term  $\mu$  effectively results in a zero-mean ensemble of data. The covariance matrix is used in the probability theory as a discrete version of the covariance function for the stochastic process [75]. The term  $\mu$  is occasionally treated as optional, and can be omitted in notation:

$$C = U_o U_o^T \quad (2.19)$$

The basis  $\Phi = [\phi_1 \ \phi_2 \ \dots \ \phi_k]$  for the reduced model is then obtained as  $k$  dominant eigenvectors of the covariance matrix,

$$C\phi_i = \lambda_i\phi_i, \quad \lambda_1 > \lambda_2 > \dots > \lambda_k, \quad i = 1, 2, \dots, k \quad (2.20)$$

This choice of basis is justified in the following theorem, adapted from [117].

**Theorem 2.1 Optimality of the POD basis**

The solution of the eigenvalue problem for the correlation matrix (2.20) is also a solution to the following optimization problem: minimize

$$E_{dist} = \sum_{j=1}^N \|u(t_j) - \hat{u}(t_j)\|_2 \quad (2.21)$$

or the distance of the time-dependent data set  $u(t_i)$  set from its reduced representation  $\hat{u}(t_i)$ , given that  $\hat{u}(t)$  is subject to (2.2) rewritten in the form

$$\hat{u}(t_j) = \sum_{i=1}^k \langle u(t_j), \phi_i \rangle \phi_i \quad (2.22)$$

and that the basis vectors  $\{\phi_i\}$  are orthonormal:

$$\langle \phi_i, \phi_j \rangle = \delta_{ij} \quad (2.23)$$

where  $\langle \cdot, \cdot \rangle$  is a Euclidean inner product.

**Proof**

Substituting (2.22) into (2.21), and simplifying by orthonormality, we obtain an a version of (2.21)

$$E_{dist} = \sum_{j=1}^N \|u(t_j) - \hat{u}(t_j)\|_2 = \sum_{j=1}^N \sqrt{\|u(t_j)\|^2 - \sum_{i=1}^k |\langle u(t_j), \phi_i \rangle|^2} \quad (2.24)$$

which leads to an equivalent formulation of an optimization problem: maximize the alignment of the data set with the new coordinate vectors

$$E_{align} = \sum_{j=1}^N \sum_{i=1}^k \left| \langle u(t_j), \phi_i \rangle \right|^2 \quad (2.25)$$

subject to (2.23). For the case  $k=1$ , the associated Lagrangian functional is written as

$$L = \sum_{j=1}^N \left| \langle u(t_j), \phi_1 \rangle \right|^2 + \lambda(1 - \|\phi_1\|_2^2) \quad (2.26)$$

The necessary condition for the solution of the optimization problem is

$$\nabla L = 0 \quad (2.27)$$

Index manipulation performed on the matrix  $U_o$  in the expression

$$\frac{\partial L}{\partial \phi_{1i}} = 2 \sum_{j=1}^N \left( \sum_{l=1}^n U_{oij} \phi_{1l} \right) U_{oij} - 2\lambda \phi_{1i} = 2 \sum_{l=1}^n \sum_{j=1}^N (U_o U_o^T)_{il} - 2\lambda \phi_{1i} \quad (2.28)$$

rewrites (2.27) as an eigenvalue problem

$$U_o U_o^T \phi = \lambda \phi \quad (2.29)$$

Since the matrix  $U_o U_o^T$  is symmetric positive definite, the problem (2.29) has a set of  $n$  non-negative eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$  and an orthonormal set of corresponding eigenvectors  $\{\phi_i\}$ .

We shall now show that the eigenvector  $\phi_1$  corresponding to the highest eigenvalue  $\lambda_1$ , maximizes (2.25). By index manipulation, we obtain an alignment estimate:

$$\begin{aligned}
\sum_{j=1}^N \left| \langle u(t_j), \phi_1 \rangle \right|^2 &= \left\langle \sum_{j=1}^N \left( \sum_{l=1}^n U_{ol} \phi_{1l} \right) u(t_j), \phi_1 \right\rangle = \\
&= \left\langle \sum_{l=1}^n \left( \sum_{j=1}^N U_{o\dots j} U_{o\dots j}^T \phi_{1l} \right) u(t_j), \phi_1 \right\rangle = \langle U_o U_o^T \phi_1, \phi_1 \rangle = \lambda_1
\end{aligned} \tag{2.30}$$

Now compare  $\phi_1$  with an arbitrary normalized vector with a representation

$\tilde{\phi} = \sum_{l=1}^n \langle \tilde{\phi}, \phi_l \rangle \phi_l$ . The alignment of this vector to the data set (2.16) is written as:

$$\begin{aligned}
\sum_{j=1}^N \left| \langle u(t_j), \tilde{\phi} \rangle \right|^2 &= \sum_{l=1}^n \sum_{l'=1}^n \left( \left\langle \sum_{j=1}^N \langle u(t_j), \phi_l \rangle u(t_j), \phi_{l'} \right\rangle \langle \tilde{\phi}, \phi_l \rangle \langle \tilde{\phi}, \phi_{l'} \rangle \right) = \\
&= \sum_{l=1}^n \sum_{l'=1}^n \langle \lambda_l \phi_l, \phi_{l'} \rangle \langle \tilde{\phi}, \phi_l \rangle \langle \tilde{\phi}, \phi_{l'} \rangle = \sum_{l=1}^n \lambda_l \left| \langle \tilde{\phi}, \phi_l \rangle \right|^2 \leq \lambda_1 \sum_{l=1}^n \left| \langle \tilde{\phi}, \phi_l \rangle \right|^2 = \lambda_1
\end{aligned} \tag{2.31}$$

The reasoning is generalized for  $k = 2, 3, \dots$  by induction. The second vector  $\phi_2$  for the orthonormal basis maximizes (2.25) with an additional constraint  $\langle \phi_1, \phi_2 \rangle = 0$ , and turns out to be the second eigenvector of (2.29), the step (2.31) is repeated for  $\phi \perp \text{span}(\phi_1)$ , and so on. The general form of (2.31) is an error estimate similar to (2.15):

$$E_{align} = \sum_{j=1}^N \sum_{i=1}^k \left| \langle u(t_j), \phi_i \rangle \right|^2 = \sum_{i=1}^k \lambda_i \tag{2.32}$$

To decide on the dimension of the reduced model using (2.32), we measure the fraction of 'eigenvalue energy' of the model captured by a basis of dimension  $k$ :

$$E_k = \sum_{i=1}^k \lambda_i / \sum_{i=1}^N \lambda_i \tag{2.33}$$

and select  $k$  so that  $E_k \approx 1$  (within a margin of 1%, 0.1%, etc). For the eigenvalue distributions following the power law, this can be achieved for very small values of  $k$ .

We note that in practice the dimension of the problem often exceeds the number of available observations,  $N < n$ . If that is the case, using the large matrix in (2.29) is computationally inconvenient. We can instead solve an eigenvalue problem with a smaller matrix:

$$U_o^T U_o \phi' = \lambda' \phi' \quad (2.34)$$

and find the leading eigenvalues and corresponding eigenvectors of  $U_o U_o^T$  from the relationships

$$\begin{aligned} \lambda_i &= \lambda'_i \\ \phi_i &= \frac{1}{\sqrt{\lambda_i}} U_o \phi'_i \end{aligned} \quad (2.35)$$

Compare (2.34) with (2.12) to see that  $\phi'$  are the right singular vectors, and  $\phi$  are the left singular vectors of the covariance matrix.

We refer to [117] for a more sophisticated explanation of proper orthogonal decomposition, in the context of Hilbert-Schmidt operator theory.

■

The basis provided by the method of snapshots is an attractive choice for model reduction. It satisfies several empirically expected characteristics at once: the reduced model retains the characteristics of the data set that contribute the most to its variance; the directions of the coordinate axis in the new subspace are optimally

aligned with the data; also, if (2.18) is included in the process, the data mean is best represented. The method uses only standard linear algebra operations, and does not depend on the nonlinearity or generic complexity of the underlying model equations (2.1).

The numerical stability of the reduced model remains an unresolved implementation issue. It has been observed that the method of snapshots occasionally leads to unstable systems of ODEs, even when the original system is stable [21], [53], but an efficient method to ensure stability has not yet been developed. A general *a priori* characterization of stable reduced systems is also not available.

Formally, a reduced model does not have to inherit linear stability: even with  $u \approx \hat{u}$ ,  $J(u) \approx J(\hat{u})$ , the solutions of the full and projected equations can produce structurally different phase portraits. For example, an orthogonal projection of a sink may produce a saddle point:

$$J = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & \alpha \\ 0 & -\alpha & \alpha^2 - 1 \end{pmatrix}, \quad \Phi = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \Phi^T J \Phi = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad (2.36)$$

In general, the eigenvalues of the full and reduced Jacobian matrices  $J, \Phi^T J \Phi$  are not related in any obvious way, unless we enforce additional conditions on the Jacobian  $J$  (such as require it to be symmetric, or negative definite: neither is typical for chemical reaction systems). In Section 2.4.3 we suggest a sampling approach to address this issue. We collect snapshot information,

detect the snapshot content that is likely leading to instability, then reject some of the snapshots and build a reduced model that is relatively more likely to be stable. To our knowledge, the only formal approach to ensuring ODE stability under projection is valid locally, near a single critical point, and at a cost of optimal representation of the snapshots [95]; not applicable for our tasks.

While the projected version of the first derivative information is inconclusive, reduced equations preserve desirable properties of the second derivative, important in the context of convex optimization. The  $n$  Hessian matrices of the full model are given by

$$H_l = \left( \frac{\partial^2 f_l}{\partial u_i \partial u_j} \right) \quad (2.37)$$

and the corresponding reduced model Hessians by

$$\hat{H}_l = \left( \frac{\partial f(\Phi q + \mu, t, p)_l}{\partial q_i \partial q_j} \right) = \Phi^T H_l(\Phi q + \mu, t, p) \Phi \quad (2.38)$$

We refer to the following simple theorem, adapted from [42].

**Theorem 2.2**

If the matrix (2.37) is positive definite for all  $u$ , so is its lower-rank projection (2.38). In addition, the projection has a lower condition number.

**Proof**

For a positive definite matrix (2.37), the lower-rank projection is also positive definite: since  $z^T H z \geq 0$  is true for every vector, it must also be true for every vector  $\hat{z} := \Phi z$  of the reduced space, and  $z^T \Phi^T H \Phi z \geq 0$ .



The eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ ,  $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_k$  of the symmetric matrices (2.37), (2.38) are subject to interlacing inequalities

$$\lambda_i \geq \hat{\lambda}_i \geq \lambda_{n-k+i} \quad i = 1, 2, \dots, k \quad (2.39)$$

Then the reduced Hessian  $\hat{H}$  has a lower condition number  $\kappa$ :

$$\kappa(H) = \lambda_1 / \lambda_n, \quad \kappa(\hat{H}) = \hat{\lambda}_1 / \hat{\lambda}_k \quad (2.40)$$

For a short proof of the inequalities (2.39), we use a form of Rayleigh's principle. Let  $v_1, v_2, \dots, v_n$ ,  $\hat{v}_1, \hat{v}_2, \dots, \hat{v}_k$  be the corresponding eigenvectors of  $H, \hat{H}$ .

For  $i = 1, 2, \dots$ , take any vector  $s_i$  in the subspace  $span(v_1, v_2, \dots, v_i) \cap (span(\Phi^T \hat{v}_1, \Phi^T \hat{v}_2, \dots, \Phi^T \hat{v}_{i-1}))^\perp$ .

Note that  $\Phi s_i \in (span(\hat{v}_1, \hat{v}_2, \dots, \hat{v}_{i-1}))^\perp$ . To obtain the left side of (2.39), we write out the Rayleigh's quotients for the Hermitian matrices:

$$\lambda_i \geq \frac{(\Phi s_i)^T H (\Phi s_i)}{(\Phi s_i)^T (\Phi s_i)} = \frac{s_i^T \hat{H} s_i}{s_i^T s_i} \geq \hat{\lambda}_i \quad (2.41)$$

For the right side, replace  $H, \hat{H}$  with  $-H, -\hat{H}$  correspondingly.

■

A similar characterization for the Hessian of the output function

$$H_{\mathfrak{S}} = \left( \frac{\partial^2 \mathfrak{S}}{\partial u_i \partial u_j} \right) \quad (2.42)$$

is an open question. Empirically, a such Hessian computed for the reduced model has a lower condition number than the Hessian computed for the full model. The

formal statement on the subject, unfortunately, is not available. We mention the topic again, in Chapter 5.

Because of its desirable properties and easy implementation, POD remains the most often used tool in dimensionality reduction of systems with nonlinear dynamics. At the same time, there is a growing body of empirical results and counter-examples that demonstrate the limitations of the method ([6], [33], [52], etc). At the current state of the field of study, there is a need for a better understanding of the process, and perhaps for a hybrid, or modified approach.

In response to the critical materials, we point out that the method of snapshots is essentially a data compression tool applied to first-order correlations in the observations of the model. It sometimes fails to detect and reproduce such implicit features of the full model as stability and nonlinear sensitivities, especially if the features are not strongly present in the snapshots. Once the features of interest are identified by techniques of factor importance analysis, the method of snapshots may be modified to better reproduce them in the reduced model. We will introduce such suggestions for improvement to the extent needed by our applied problems.

## 2.2 SENSITIVITY ANALYSIS

The overall effect of the model parameters  $p$  on the reduced model solution  $\hat{u}$  is a combination of the parametric dependence of the full model solution  $u$ , and the details of the reduction procedure (such as the placement and the contents of snapshots). The questions of the relative importance of the parameters, the

snapshots, or of the choices in the reduction process may be hard to answer by inspection.

The dependence of the reduced model on the individual parameters, and on the intermediate steps of the reduction process may be formally characterized by derivative information, or estimated by a statistical analysis based on many runs of the model, perhaps many runs of the model reduction process. Both approaches produce simplified, local characterizations of the model sensitivities. To estimate the global behavior, the analysis needs to be applied to representative subsets of the parameter space.

The main reason for sensitivity analysis of the model reduction process is the availability of techniques that allow us to amplify the quality of reproduction of the important model components in the reduced model solution; we need a better understanding of importance than is available by inspection. Establishing the importance of each parameter in the reproduction of the output function by the reduced model is also important for characterization of validity of using the same reduced model for different sets of parameters: the region of acceptable values is necessarily narrower in the important parameters.

### 2.2.1 DERIVATIVE INFORMATION

In this section, we describe how to obtain the partial derivatives of an arbitrary output function  $\mathfrak{J}$  applied to the full and the reduced models. For practical purposes, we are primarily interested in first-order differentiation, but higher order

derivatives are also possible to obtain. Much of the construction is based on the fact that it is possible to differentiate the singular value decomposition of a matrix analytically (though not explicitly). The rest of the procedure consists of differentiation of ODE solutions, a standard task accomplished either by direct or the adjoint method, the latter explained in more detail in Chapter 4.

The procedure is computationally expensive, but useful, because it allows to obtain the sensitivities of the reduction method itself. To our knowledge, this sensitivity information that was not examined, or available previously (except possibly a calculation by finite differences, requiring many runs of the reduction process).

For a selected output function  $\mathfrak{J}$ , the first-order derivative can be expressed by chain rule:

$$\frac{d\mathfrak{J}(\hat{u})}{dp} = \frac{d\mathfrak{J}(\hat{u})}{\underset{(1)}{d\hat{u}}} \cdot \frac{d\hat{u}}{\underset{(4)}{d\Phi}} \cdot \frac{d\Phi}{\underset{(2)}{du}} \cdot \frac{du}{\underset{(3)}{dp}} \quad (2.43)$$

The expression (2.43) is schematic: depending on the type of partial derivative needed, it may not have to be evaluated completely; correct indexing and the times at which the components are evaluated are provided as needed by specific tasks. Higher-order derivative information may be expanded similarly, with a higher computational cost.

We assume that an explicit, differentiable expression for the output function is available, providing the term (1) in (2.43). We also allow  $\mathfrak{J}$  to be defined in a form of comparison between the full and the reduced models, as in (1.6). The type

of an output function that depends on  $u$  and  $\hat{u}$  leads to a slightly more complicated form of the complete derivative:

$$\frac{d\mathfrak{I}(u, \hat{u})}{dp} = \frac{\partial \mathfrak{I}(u, \hat{u})}{\partial \hat{u}} \cdot \frac{d\hat{u}}{d\Phi} \cdot \frac{d\Phi}{du} \cdot \frac{du}{dp} + \frac{\partial \mathfrak{I}(u, \hat{u})}{\partial u} \cdot \frac{du}{dp} \quad (2.44)$$

The terms (2), (3), (4) in (2.43), (2.44) are not available explicitly, and require additional techniques.

### Differentiation of the singular value decomposition

The term (2) in (2.44) represents the sensitivity of the reduced space basis with respect to the contents of the snapshots:

$$\frac{d\Phi}{du} = \left[ \frac{d\Phi}{du(t_1)}, \frac{d\Phi}{du(t_2)}, \dots, \frac{d\Phi}{du(t_N)} \right] \quad (2.45)$$

We use a procedure suggested by Papadoupolo et. al. [84] that differentiates a converged SVD. Since the procedure is relatively new, and has not been applied to practical problems in our field of study, we present it in some detail. Given the decomposition of a full-rank matrix  $A$ :

$$A = U\Sigma V^T \quad (2.46)$$

let  $a_{ij}$ ,  $u_{ij}$ , **Error! Objects cannot be created from editing field codes.** be elements of **Error! Objects cannot be created from editing field codes.**,  $U$ ,  $V$  correspondingly, and **Error! Objects cannot be created from editing field codes.** the diagonal elements of  $\Sigma$ . We differentiate (2.47) with respect to the elements  $a_{ij}$  of the matrix:

$$\frac{\partial A_{ij}}{\partial a_{ij}} = \frac{\partial U}{\partial a_{ij}} \Sigma V^T + U \frac{\partial \Sigma}{\partial a_{ij}} V^T + U \Sigma \frac{\partial V^T}{\partial a_{ij}} \quad (2.47)$$

Also, differentiating the orthogonality conditions  $U^T U = 0$  ,  $V^T V = 0$  , we obtain:

$$\begin{aligned}\frac{\partial U^T}{\partial a_{ij}} U + U^T \frac{\partial U}{\partial a_{ij}} &= 0 \\ \frac{\partial V^T}{\partial a_{ij}} V + V^T \frac{\partial V}{\partial a_{ij}} &= 0\end{aligned}\tag{2.48}$$

We denote the terms of (2.48) by

$$\begin{aligned}\Omega_U^{ij} &= U^T \frac{\partial U}{\partial a_{ij}} \\ \Omega_V^{ij} &= \frac{\partial V^T}{\partial a_{ij}} V\end{aligned}\tag{2.49}$$

We then multiply (2.47) by  $U^T$  ,  $V$  on the left and on the right correspondingly:

$$U^T \frac{\partial A}{\partial a_{ij}} V = \Omega_U^{ij} \Sigma + \frac{\partial \Sigma}{\partial a_{ij}} + \Sigma \Omega_V^{ij}\tag{2.50}$$

Notice that  $\Sigma$  is diagonal, and  $\Omega_U^{ij}, \Omega_V^{ij}$  are anti-symmetric by definition, with zeros on the diagonal:

$$U^T \frac{\partial A}{\partial a_{ij}} V = \frac{\partial \Sigma}{\partial a_{ij}}\tag{2.51}$$

The matrix  $\frac{\partial A}{\partial a_{ij}}$  has only one non-zero component:

$$\frac{\partial \sigma_k}{\partial a_{ij}} = u_{ik} v_{jk}\tag{2.52}$$

Substitution into (2.49) results in systems of two linear equations for each index pair  $(k, l)$ :

$$\begin{cases} \sigma_l(\Omega_U^{ij})_{kl} + \sigma_k(\Omega_V^{ij})_{kl} = u_{ik}v_{jl} \\ \sigma_k(\Omega_U^{ij})_{kl} + \sigma_l(\Omega_V^{ij})_{kl} = -u_{il}v_{jk} \end{cases} \quad (2.53)$$

uniquely defining the components  $(\Omega_U^{ij})_{kl}, (\Omega_V^{ij})_{kl}$ , assuming singular values  $\sigma_k, \sigma_l$  do not coincide. Taking anti-symmetry into account, there are  $n(n-1)/2$  distinct systems, resulting in the expressions

$$\begin{aligned} \frac{\partial U}{\partial a_{ij}} &= U\Omega_u^{ij} \\ \frac{\partial V}{\partial a_{ij}} &= -V\Omega_v^{ij} \end{aligned} \quad (2.54)$$

To apply the procedure to the specific SVD used in model reduction, set

$$A = (U_o)(U_o)^T, \quad U = \Phi, \text{ then}$$

$$\frac{\partial(\phi_k)_i}{\partial(u(t_i))_j} = \left(\frac{\partial U}{\partial a_{ij}}\right)_{kl} \quad (2.55)$$

To differentiate an alternative eigenvalue problem described by (2.34), (2.35), set  $A = (U_o)^T(U_o)$ ,  $V = -\Phi$  instead. We refer to the original paper for a treatment of the procedure in more detail, including degenerate cases with rank deficiency, and repeated eigenvalues.

■

### Differentiation of the model state

Finding the term  $\frac{du}{dp}$ , (3) in (2.44), is a standard task of sensitivity analysis

[22], [120]. Assuming smoothness of the full model solution with respect to time,

and to the initial conditions, the derivative information can be obtained by direct differentiation of the ODE (2.6), or by the adjoint method.

By the direct approach, the sensitivity term  $y(t) = \frac{du}{dp_j}$  satisfies the system

$$\begin{aligned} \frac{dy}{dt} &= \frac{d}{du} f(u,t)y(t) + \frac{d}{dp_j} f(u,t) \\ (u(t_0))_i &= 0 \quad i \neq j \\ (u(t_0))_j &= 1 \end{aligned} \tag{2.56}$$

solved for every parameter  $p_j, j = 1, 2, \dots, m$ . The terms  $\frac{d}{du} f(u,t,p), \frac{d}{dp_j} f(u,t,p)$  are available explicitly.

By the adjoint method, the sensitivity  $\left. \frac{du}{dp} \right|_{t=T}$  is found component-wise for

each  $u_i, i = 1, 2, \dots, n$ :

$$\frac{du_i}{dp} = -u^*(T) \frac{\partial}{\partial p} f(x,T) + u^*(t_0) \frac{du_0(p)}{dp} \tag{2.57}$$

with the *adjoint variable*  $u^*$  defined by the system of ODEs solved backwards in time:

$$\begin{aligned} \frac{du^*}{dt} &= -\left( \frac{df}{du} \right)^T u^* \\ (u^*(T))_k &= \delta_{ik} \end{aligned} \tag{2.58}$$

A justification and a more detailed description of the adjoint method is given in Chapter 4, where we also explain how to obtain the derivatives of the output function  $\mathfrak{F}$  with respect to parameters  $p$ .



■

### Differentiation with respect to subspace basis

The term  $\frac{d\hat{u}}{d\Phi}$ , (4) in (2.44) is found by differentiation of (2.2):

$$\frac{d\hat{u}(t)}{d\Phi_{ij}} = \frac{d}{d\Phi_{ij}}(\Phi q + \mu) = \frac{d\Phi}{d\Phi_{ij}} q(t) + \Phi \frac{dq}{d\Phi_{ij}} \quad (2.59)$$

The term  $Y = \frac{dq}{d\Phi_{ij}}$  is obtained numerically, by differentiation of (2.6). The right-

side simplifies as follows:

$$\begin{aligned} \frac{d}{d\Phi_{ij}}(\Phi^T f(\Phi q + \mu, t)) &= \\ &= \frac{d\Phi^T}{d\Phi_{ij}} f(\Phi q + \mu, t) + \Phi^T \left( \frac{d\Phi}{d\Phi_{ij}} q(t) + \Phi \frac{dq}{d\Phi_{ij}} \right) J_{ij}(\Phi q + \mu, t) \end{aligned} \quad (2.60)$$

leading to a system of equations:

$$\begin{aligned} \frac{dY}{dt} &= \frac{d\Phi^T}{d\Phi_{ij}} f(\Phi q + \mu, t) + \Phi^T \left( \frac{d\Phi}{d\Phi_{ij}} q(t) + \Phi Y \right) J_{ij}(\Phi q + \mu, t) \\ Y(t_0) &= \frac{d\Phi^T}{d\Phi_{ij}} (p + q) \end{aligned} \quad (2.61)$$

Note that the trajectory  $q(t)$  needs to be recorded, possibly interpolated at missing

time instances. The matrices  $\frac{d\Phi}{d\Phi_{ij}}$ ,  $\frac{d\Phi^T}{d\Phi_{ij}}$  are mostly sparse, the only nonzero

component being unity in position  $(i, j)$ ,  $(j, i)$  correspondingly. An adjoint

approach to differentiation is less appropriate here: we may need the sensitivity of  $\hat{u}(t)$  at multiple time instances.

■

Since the derivative information will be mostly used to find model elements of extremely high relative importance, a moderate numerical error in the computation of derivative is allowed.

Selecting which components of the arrays in the expression (2.44) to evaluate depends on the available computational budget and the specific practical task. In particular, the first-order derivatives  $\frac{d\mathfrak{S}(\hat{u})}{du(t_i)}$ ,  $\frac{d\mathfrak{S}(\hat{u})}{dp_j}$  characterize the

influence of the snapshots and the parameters on the performance of the reduced model. The first-order derivative  $\frac{d\Phi}{dp_j}$  characterizes the dependence of the subspace basis on the parameters.

We can also obtain a first-order estimate of the sensitivity of the reduced model with respect to a small change in the *placement* of a particular snapshot, in effect, characterizing the relative importance of a small time interval  $(\tau - \varepsilon, \tau + \varepsilon)$ .

Let the matrix of observations be  $U_o = [u(t_1), u(t_2), \dots, u(t_{N-1}), u(\tau)]$ . Then

$$\frac{d\mathfrak{S}(\hat{u})}{d\tau} = \frac{d\mathfrak{S}(\hat{u})}{d\hat{u}} \cdot \frac{d\hat{u}}{d\Phi} \cdot \frac{d\Phi}{du(\tau)} \cdot \frac{du}{d\tau} = \frac{d\mathfrak{S}(\hat{u})}{d\hat{u}} \cdot \frac{d\hat{u}}{d\Phi} \cdot \frac{d\Phi}{du(\tau)} \cdot f(u, \tau, p) \quad (2.62)$$

Some sensitivity information on the model can be obtained without differentiation of the reduction procedure, but then even an *a posteriori* information

on how the reduced model preserves the full model sensitivities will be lost. Using the expression (2.62) is an improvement on characterizing the importance of the time interval by the quantity  $\frac{du}{d\tau} = f(u, \tau, p)$  alone. We also note a distinction

between the expression  $\frac{d\mathfrak{I}(\hat{u})}{d\tau}$ , and a derivative  $\frac{\partial\mathfrak{I}(\hat{u})}{\partial\hat{u}} \cdot \frac{d\hat{u}}{d\tau} = \frac{d\mathfrak{I}(\hat{u})}{d\hat{u}} \cdot \hat{f}(\hat{u}, \tau, p)$ .

The latter is an incomplete, computationally cheaper estimate of the importance of time  $(\tau - \varepsilon, \tau + \varepsilon)$  for the performance of the reduced model; this interpretation requires an assumption that the basis  $\Phi$  is fixed, which is not a valid unless it happens that all the components of  $\frac{d\Phi}{du(\tau)}$  are very small.

For the sensitivity of the Jacobian (2.10) of the reduced model, we have the chain rule expansion

$$\begin{aligned} \frac{d\hat{J}}{dp} &= \frac{d\hat{J}}{d\Phi} \cdot \frac{d\Phi}{du} \cdot \frac{du}{dp} = \\ &= \left( \frac{d\Phi^T}{d\Phi} \cdot J(\Phi q - \mu, t) + \Phi^T H(\Phi q - \mu, t) \cdot q \cdot \Phi + \Phi^T J(\Phi q - \mu, t) \right) \cdot \frac{d\Phi}{du} \cdot \frac{du}{dp} \end{aligned} \quad (2.63)$$

Even with the explicit, sparse forms of both the Jacobian  $J$  and the Hessian  $H$  of the full model, the evaluation of (2.63) is computationally expensive. This complexity presents a practical barrier to obtaining further sensitivity information related to the stability of the reduced system.

In particular, obtaining the sensitivities for the eigenvalues of the Jacobian of the ODE would improve our understanding of numerical stability of the reduced model. However, this information would require an evaluation of (2.63), and

possibly a differentiation of the Jordan canonical decomposition, for which an analytic procedure is not available (to our knowledge).

### 2.2.2 SENSITIVITY BY INTERPOLATING MODELS

We shall now describe how to characterize the influence of the parameters on the output of the function through statistical regression. The idea is to treat the complete process, from model reduction to the evaluation of the output function as an action of a single functional on the inputs that may include model parameter values, or features in the reduction setup. This functional is not available explicitly, and has to be interpolated based on outputs (and, possibly, derivative information) of a few instances of model reduction with different sets of inputs. Once the interpolation is constructed in an explicit form, it can be used to predict the outputs of the model reduction process, and the sensitivities of these outputs.

The idea of estimating the effect of individual inputs on the output by constructing an interpolating model appears in statistical learning theory, and has been recently successfully applied in the study of the coupled multi-physics systems [91] (where the function on the parameters can be obtained explicitly, but is very complicated), and of the social networks [91] (where the function is not deterministically available).

While the evaluations of the interpolating model are computationally cheap, its construction is computationally expensive, so it should be treated as a one-time analysis tool. Its construction based on typical runs of reduction is meant to improve

our general understanding of importance of individual model elements that is not captured by the first-order derivative information; moderate error in importance assessment is allowed.

### Time-independent interpolation

For a fixed placement of the snapshots, and a fixed dimension of the reduced model, the output function is dependent on the parameters (in our case, initial conditions) alone. This dependence can be estimated by an interpolating model  $\Gamma(p) \approx \mathfrak{S}(p)$ , meant to represent the combined effect of the two ODE solvers, the singular value decomposition, and the output function on the list of parameters. Clearly, the replacement of  $\mathfrak{S}$  with  $\Gamma$  adds another tier of model reduction, with an expected loss of quality.

For a basis of multivariable functions

$$\Psi = \{\psi_1(p_1, p_2, \dots, p_m), \psi_2(p_1, p_2, \dots, p_m), \dots\} \quad (2.64)$$

we define

$$\Gamma(p) = \sum_i \chi_i \psi_i(p) \quad (2.65)$$

We normally use a polynomial basis, with each function defined by:

$$\psi_j(p) = \psi_j(p_1, p_2, p_3, \dots) = \prod_l \rho^{(k_l)}(p_l) \quad (2.66)$$

where  $\rho^{(k_l)}$  is a single-variable polynomial of order  $k_l$ . The set of polynomials  $\rho$  is arbitrary; for example, we can use a trivial choice  $\rho^{(k)}(p_i) = p_i^k$ , leading to the multivariable basis

1

$$\begin{array}{cccccccc}
p_1 & p_2 & p_3 & \dots & & & & \\
p_1^2 & p_1 p_2 & p_1 p_3 & \dots & p_2^2 & p_2 p_3 & \dots & \\
p_1^3 & p_1 p_2^2 & p_1 p_3^2 & \dots & p_1 p_2 p_3 & p_1 p_2 p_4 & \dots & p_2^3 \dots
\end{array} \tag{2.67}$$

A more sophisticated case would be a well-conditioned, orthogonal set, such as a family of Hermite, or Chebyshev polynomials.

Depending on the required precision of the interpolation, and the available computational budget, the basis  $\Psi$  may include just the linear polynomials, or a complete set of multi-variable polynomials up to a fixed maximal total degree. In the presence of additional information about the relative importance of the specific parameters, the basis may be adaptive, with higher-order polynomials only in some of the most important variables, and linear polynomials in all the rest; for more details on basis truncation, we refer to our work in [91].

The coefficients  $\chi_i$  are found by collocation based on a sample

$$p^{(1)}, p^{(2)}, \dots, p^{(i)} = (p_1^{(i)}, p_2^{(i)}, \dots, p_m^{(i)}) \tag{2.68}$$

from the parameter space  $P$ . The system of linear collocation equations

$$\begin{array}{l}
\mathfrak{I}(p^{(1)}) = \sum_i \chi_i \psi_i(p^{(1)}) \\
\mathfrak{I}(p^{(2)}) = \sum_i \chi_i \psi_i(p^{(2)}) \\
\vdots
\end{array} \tag{2.69}$$

requires at least as many rows as there are polynomials in the basis  $\Psi$ , and, correspondingly, many full runs of the model reduction process. The number of the required runs may be even larger if the system is intentionally over-determined for a

better condition number and decreased bias in the interpolation. The system is then solved in the least squares sense [42].

The task is to make the procedure computationally feasible through the use of additional sensitivity information. In [91], [92], we introduce two approaches: reduce the number of variables, and make use of the derivative information.

To reduce the number of variables, we partition the parameter set  $p$  into ‘important’ and ‘unimportant’, and define  $\Gamma$  as a function of the important parameters only. The reduction in the number of variables used in (2.65) can be also achieved through data lumping, in which the set of parameters  $p = (p_1, p_2, \dots, p_m)$  gets replaced with a set of a few of their linear combinations  $s = (s_1, s_2, \dots, s_k)$ ; the interpolating model is then defined as  $\Gamma = \sum_i \chi_i \psi_i(s)$ . It is natural to use an already obtained POD-based projection, and define  $s^T = \Phi p^T$ .

Note that in this case the relative importance of the variable  $s_i$  is also a measure of the relative importance of the eigenvector  $\phi_i$ , and can be used to select the subspace basis in a way not equivalent to capturing most of the eigenvalue energy, (2.33). This observation may be useful for the cases where an addition of a few non-dominant eigenvectors to the basis improves the quality of the reduced model.

Since differentiation is a linear operator, we can augment the collocation system (2.67) with first-order derivatives of every equation. The idea is to fit the interpolated output function to sample output values, and also to fit the (explicitly

available) derivative of the interpolated output function to the derivatives of sample output values. The augmented system is written as

$$\begin{aligned}\mathfrak{I}(p^{(i)}) &= \sum_i \chi_i \psi_i(p^{(i)}) \quad i=1,2,\dots,n \\ \frac{d}{dp_j} \mathfrak{I}(p^{(i)}) &= \sum_i \chi_i \frac{d}{dp_j} \psi_i(p^{(i)}), \quad i=1,2,\dots,n, \quad j=1,2,\dots,n\end{aligned}\tag{2.70}$$

The derivatives  $\frac{d}{dp_j} \psi_i$  of the polynomial basis functions are available explicitly.

The left-side expressions  $\frac{d}{dp_j} \mathfrak{I}(p^{(i)})$  can be obtained as shown in (2.44); moderate error is allowed. The block of  $n$  derivatives provides an additional  $n$  rows for each row of the original system (2.67), resulting in a fewer required sample points. As long as the derivative information can be computed at the overhead of less than  $n+1$  full runs of the reduction process, using (2.70) instead of (2.69) is computationally justified. Empirically, the computational overhead for the calculation of the derivative is 100–300%; [91] suggests a theoretical bound of 500%.

We note that it is possible to avoid the collocation procedure through the use of an orthogonal basis  $\Psi$ : the coefficients  $\chi_i$  may be obtained by the Galerkin method. In that case, to find a specific  $\chi_k$ , we multiply the expression (2.64) by  $\psi_k$ , and integrate by parts over the set of possible parameters (here,  $P$  must be a bounded region):

$$\int_P \mathfrak{I}(p) \psi_k(p) dp = \int_P \psi_k(p) \sum_i \chi_i \psi_i(p) dp = \chi_k \cdot 1 + \sum_{i \neq k} \chi_i \cdot 0\tag{2.71}$$



$$\mathfrak{I}(p)\psi_k(p) - \int_p \frac{d\mathfrak{I}(p)}{dp} \psi_k^\uparrow(p) dp = \chi_k \quad (2.72)$$

where  $\psi_k^\uparrow(p) = \int \psi_k(p) dp$  is the explicitly available primitive of the polynomial  $\psi_k(p)$ . This approach requires only one run of the reduction process to compute  $\mathfrak{I}(p)\psi_k(p)$ , but possibly many runs (including the derivative information) in the calculation of  $\int_p \frac{d\mathfrak{I}(p)}{dp} \psi_k^\uparrow(p) dp$ , making it practical only if a very sparse representative set of points is chosen for the quadrature.

■

### Time-dependent interpolation

Now suppose that the sensitivity information is needed at a large number of time instances. For that case, we offer a simple scheme with a separation of time and state, while a more general approach lies outside of scope of our work. To build an interpolating model  $\Gamma(p, t) \approx \mathfrak{I}(p, t)$ , we combine an expansion in terms of the parameters  $\Psi = \{\psi_1(p_1, p_2, \dots, p_m), \psi_2(p_1, p_2, \dots, p_m), \dots\}$  with an expansion in terms of time and frequencies  $\Xi = \{\xi_1(\omega_1, t), \xi_2(\omega_2, t), \dots\}$ , and write (2.64) as

$$\Gamma(p, t) = \sum_i \chi_i \psi_i(p) \cdot \sum_j \omega_j \xi_j(t) \quad (2.73)$$

The collocation procedure based on a sample from  $P \times [t_0, T]$  will have a form

$$\mathfrak{I}(p_i, t^{(k)}) = \sum_i \chi_i \psi_i(p_i) \cdot \sum_j \omega_j \xi_j(t^{(k)}), \quad k=1,2,\dots, \quad i=1,2,\dots,n \quad (2.74)$$

We suggest using a Fourier expansion

$$\xi_j(t) = \sin\left(\frac{\pi t}{\omega_j}\right), \quad \omega_j = j \quad (2.75)$$

and augmenting the system (2.73) with partial derivatives with respect to parameters  $p$  only.

■

We note that by manipulation of the polynomial basis (2.64) and the inputs sample (2.68), we can build the interpolating model  $\Gamma$  to have any degree of nonlinearity (if an increase in computational budget is acceptable). We can also modify the setup to achieve increased precision in representation of response of the model to some selected parameters by assigning them more high-order polynomials in the basis. We can use the interpolating model to describe the aspects of the parametric dependence of the reduced model that would otherwise involve computationally expensive sampling in a large dimensional space, such as finding the values of the parameters for which the reduced model solution at time  $T$  stays within fixed bounds.

Another possible task is estimating the influence of a single parameter on the variance of the output using the ratio

$$S(p_i) = \frac{\text{var}[E[\mathfrak{Z}_i]]}{\text{var}[\mathfrak{Z}]} \quad (2.76)$$

where each quantity  $E[\mathfrak{Z}_i]$  is defined on a sample  $\Omega$  from the parameter space  $P$ :

$$E[\mathfrak{Z}_i] = E[\mathfrak{Z}_i](p_i) = \frac{1}{|\Omega|} \int_{\Omega} \mathfrak{Z}(p_1, p_2, \dots, p_m) dp_1 dp_2 \dots dp_{i-1} dp_{i+1} \dots dp_m \quad (2.77)$$

The quadrature required in (2.75) has to be evaluated multiple times to collect a sufficient sample for evaluation of variance. To make the process efficient, we suggest evaluating (2.75) with  $\mathfrak{S}$  replaced with an interpolation  $\Gamma$ , see [92] for additional details.

We conclude this section with a remark that the results of sensitivity analysis presented in this section are open to interpretation: the meaning of factor importance is very problem-specific, and cannot be deduced from theoretical information alone. Informally, we define the time-varying importance (or an ‘index of importance’, as in [92] or [55]) of a particular variable as an absolute value of the corresponding first-order derivative of the output function, taken at a representative time instance, or averaged over the evolution of the model. If an interpolating expansion  $\Gamma$  was used, the importance is defined by the magnitudes of the coefficients at the polynomials in which the variable is present. We expect that the variable is important for the dynamics of the model if the ‘index of importance’ is high, or changes significantly with time, more information is provided in comments to a specific example, in Section 7.5.

### 2.3 SELECTIVE MODEL REDUCTION

We suggest that the POD-based model reduction may be extended to the cases when the elimination of complexity is applied selectively: only to some components of the model state, or only to some time intervals. The proposed

manipulation of the reduction method is *intrusive*: it may cause the loss of desirable properties, in particular, the optimality of fit to the observations.

The motivation for the approach comes from an observation on the limitations of the general method: while a feature of interest may be well reproduced in a reduced model, we cannot expect an arbitrary list of features to be simultaneously preserved under a single model reduction. This would not invalidate the idea of model reduction if we were to use different versions of the reduced model in combination, switching between them as needed during the simulation process, applying different reductions to different groups of components, or using an average of several reduced models.

The features of interest that prompt selective model reduction are identified by inspection, or by sensitivity analysis. We will primarily attempt to apply a higher-quality reduction to the factors strongly influencing the output function, and a lower-quality reduction to the factors that are not essential in the full model.

We will discuss a number of options, while a complete characterization of the effect of combining an arbitrary number of reducing projections lies outside of the scope of our study.

A first simple case is based on combining two reductions of the same covariance matrix. This setup is of practical use in the cases where it is unclear by inspection of several (short-term) solutions, which dimension of the reduced model is preferable.

#### **Combination of two truncations of the same basis set**

Let the full list of the covariance matrix eigenvectors be  $\{\phi_1, \phi_2, \dots, \phi_n\}$ . The subspace basis sets

$$\begin{aligned}\Phi^{(k_1)} &= (\phi_1, \phi_2, \dots, \phi_{k_1}) \\ \Phi^{(k_2)} &= (\phi_1, \phi_2, \dots, \phi_{k_1}, \dots, \phi_{k_2})\end{aligned}\tag{2.78}$$

produce the corresponding reduced model solutions  $\hat{u}^{(k_1)}, \hat{u}^{(k_2)}$ . In the special case where full model dynamics are combined with a single reduction, we set  $k_2 = n$ , then  $\Phi^{(n)}$  is a coordinate change matrix of full rank,  $\hat{u}^{(n)} = u$ .

Based on the alignment error estimate (2.32), the solution  $\hat{u}^{(k_2)}$  provides a more precise reproduction of the snapshots. While a formal statement describing the advantage in quality resulting from using  $\hat{u}^{(k_2)}$  rather than  $\hat{u}^{(k_1)}$  is not available, we can provide a partial characterization. The difference  $e = \hat{u}^{(n)} - \hat{u}^{(k)}$  satisfies the system

$$\begin{aligned}\frac{de}{dt} &= \sum_{i=1}^k \phi_i \phi_i^T (f(\hat{u}^{(n)}) - f(\hat{u}^{(k)})) + \sum_{i=k+1}^n \phi_i \phi_i^T f(\hat{u}^{(k)}) \\ e(t_0) &= \sum_{i=k+1}^n \phi_i \phi_i^T u_0(p)\end{aligned}\tag{2.79}$$

so for a sufficiently small  $k$ , even with  $f(\hat{u}^{(n)}) \approx f(\hat{u}^{(k)})$  the term  $\sum_{i=k+1}^n \phi_i \phi_i^T f(\hat{u}^{(k)})$  is not negligible.

Suppose that a lower quality reduction  $\Phi^{(k_1)}$  is applied to the state components  $u^{(1)} = (u_1, u_2, \dots, u_l)$ , and a higher-cost, higher quality reduction  $\Phi^{(k_2)}$  is applied to  $u^{(2)} = (u_{l+1}, u_{l+2}, \dots, u_n)$ ; note that model state components can be

renumbered without the loss of generality. A combined application of  $\Phi^{(k_1)}, \Phi^{(k_2)}$  to the full model equations (2.6) produces a system

$$\begin{aligned} \frac{d\hat{u}_i}{dt} &= \left( \Phi^{(k_1)} \Phi^{(k_1)T} f(\hat{u}, t, p) \right)_i & i \leq l \\ \frac{d\hat{u}_i}{dt} &= \left( \Phi^{(k_2)} \Phi^{(k_2)T} f(\hat{u}, t, p) \right)_i & i > l \\ \hat{u}_i(t_0) &= \left( \Phi^{(k_1)} \Phi^{(k_1)T} (u_0(p) - \mu) + \mu \right)_i & i \leq l \\ \hat{u}_i(t_0) &= \left( \Phi^{(k_2)} \Phi^{(k_2)T} (u_0(p) - \mu) + \mu \right)_i & i > l \end{aligned} \quad (2.80)$$

The combined projection basis  $\Phi^C$  can be obtained by replacing a  $l \times (k_2 - k_1)$  submatrix of  $\Phi^{(k_2)}$  with zeros:

$$\Phi^C = \begin{pmatrix} \Phi^{(k_2)}_{1,1} & \cdots & \Phi^{(k_2)}_{1,k_1} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ \Phi^{(k_2)}_{l,1} & & \Phi^{(k_2)}_{l,k_1} & 0 & \cdots & 0 \\ \Phi^{(k_2)}_{l+1,1} & \cdots & \Phi^{(k_2)}_{l+1,k_1} & \Phi^{(k_2)}_{l+1,k_1+1} & & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \Phi^{(k_2)}_{n,k_2} \end{pmatrix} \quad (2.81)$$

In this setup, only the first  $k_1$  vectors of the column space of  $\Phi^C = (\phi_1^C, \phi_2^C, \dots, \phi_{k_2}^C)$  are orthonormal. The matrix  $\Phi^C$  can then be normalized to the form  $\Phi^N = (\phi_1^N, \phi_2^N, \dots, \phi_{k_2}^N)$  using a standard Gram-Schmidt process:

$$\begin{aligned} \phi_i^N &= \phi_i^C, \quad i \leq k_1 + 1 \\ \phi_i^N &= \phi_i^C - \sum_{j=1}^{i-1} \langle \phi_i^N, \phi_j^C \rangle \phi_j^C \end{aligned} \quad (2.82)$$

The same dimension of the reduced models  $\hat{u}$  and  $\hat{u}^{(k_1)}$  leads to approximately the same computational cost. The orthogonal form  $\Phi^N$ , however, can be truncated to the selective basis

$$\Phi^S = (\phi_1^N, \phi_2^N, \dots, \phi_k^N), \quad k_1 \leq k < k_2 \quad (2.83)$$

resulting in a computationally cheaper model  $\hat{u}^S$ . The association of the eigenvectors  $\{\phi_i\}$  to the ordered set of the covariance matrix eigenvalues is preserved exactly in the first  $k_1$  components of (2.83), and approximately afterwards. Because of orthogonality, the distorted vectors  $\phi_{k_1}^N, \phi_{k_1+1}^N, \dots, \phi_k^N$  will not align with the first dominant eigenvectors. The alignment estimate (2.32) is still applicable: a selective solution  $\hat{u}^S$  reproduces the snapshots information with an alignment error of at worst  $\sum_{i=1}^{k_1} \lambda_i$ .

■

### Combination of two basis sets

We use the same steps as before to combine two unrelated reduced models.

Given the distinct sets of snapshots

$$\begin{aligned} U_{OI} &= [u_I(t_1^I), u_I(t_2^I), \dots, u_I(t_{N_I}^I)] \\ U_{OII} &= [u_{II}(t_1^{II}), u_{II}(t_2^{II}), \dots, u_{II}(t_{N_{II}}^{II})] \end{aligned} \quad (2.84)$$

leading to the covariance matrices

$$\begin{aligned} C_I &= (U_{OI} - \mu)(U_{OI} - \mu)^T \\ C_{II} &= (U_{OII} - \mu)(U_{OII} - \mu)^T \end{aligned} \quad (2.85)$$

and the distinct eigenvalue and eigenvector sets

$$\begin{aligned}
(\lambda_1^I, \lambda_2^I, \dots, \lambda_n^I), \quad (\phi_1, \phi_2, \dots, \phi_n) &= \Phi^{(I)} \\
(\lambda_1^{II}, \lambda_2^{II}, \dots, \lambda_n^{II}), \quad (\phi_1, \phi_2, \dots, \phi_n) &= \Phi^{(II)}
\end{aligned} \tag{2.86}$$

truncated to reduced space dimensions  $k_I, k_{II}$ .

We apply the reductions, correspondingly, to the groups of components  $u^{(I)} = (u_1, u_2, \dots, u_l)$ ,  $u^{(II)} = (u_{l+1}, u_{l+2}, \dots, u_n)$ . The system of equations is written in the same way as (2.80), leading to the combined projection basis

$$\Phi^C = \begin{pmatrix} \Phi^{(I)}_{1,1} & \cdots & \Phi^{(I)}_{1,n} \\ \vdots & & \vdots \\ \Phi^{(I)}_{l,1} & \cdots & \Phi^{(I)}_{l,n} \\ \Phi^{(II)}_{l+1,1} & \cdots & \Phi^{(II)}_{l+1,n} \\ \vdots & & \vdots \end{pmatrix} \tag{2.87}$$

leading to the normalized matrix  $\Phi^N \in R^{n \times k_{II}}$  and the truncated normalized matrix  $\Phi^S \in R^{n \times k}$  as shown in (2.82), (2.83).

This setup is of practical use in the cases where the correlations between the components of the ODE solution are slightly different over the time intervals  $(t_1^I, t_{N_I}^I), (t_1^{II}, t_{N_{II}}^{II})$ . This may be due to an insufficiently representative set of snapshots, non-periodic behavior of the solution, or the changing mean model state, to name a few issues that arise in the problems of atmospheric chemistry. We refer to [80] for more details on the difficulties in the Principal Component Analysis.

The matrices  $C_I, C_{II}, \Delta C = C_{II} - C_I$  are symmetric, so the difference between the two sets of eigenvalues (2.85) is bounded by a Weilandt-Hoffman inequality:

$$\sum_{i=1}^{k_I} (\lambda_i^{II} - \lambda_i^I)^2 \leq \|\Delta C\|_F \leq \sqrt{n} \|\Delta C\|_2 \tag{2.88}$$



and furthermore, by a ‘minimax’ characterization,

$$|\lambda_i^I - \lambda_i^H| \leq \|\Delta C\|_2, \quad i = 1, 2, \dots \quad (2.89)$$

We measure  $\|\Delta C\|_2$  to decide if the two given covariance matrices with a corresponding distinct reduced models are nevertheless sufficiently close to each other to be used in selective combination. For a sufficiently small value we shall define

$$\lambda_i = \frac{|\lambda_i^I - \lambda_i^H|}{2}, \quad i = 1, 2, \dots \quad (2.90)$$

as a combined estimate for the set of eigenvalues corresponding the eigenvectors (2.88). If required for formal characterization of the process, it is possible to estimate the combined covariance matrix based on the information from (2.87), (2.90).

■

### **Alternating reduced models in time**

In comparison with a setup that is selective *by component*, selection *by time interval* is straightforward, and does not involve violating the optimality conditions. This modification to the reduction procedure is based on an understanding that different reductions may best represent the behavior of the full model over different time intervals. Given the reductions  $\Phi^{(I)}, \Phi^{(H)}$ , we write the combined system of equations as

$$\begin{aligned}
\frac{d\hat{u}}{dt} &= \Phi^{(I)}\Phi^{(II)T} f(\hat{u}, t, p), \quad t_0 < t < t^{(I)} \\
\frac{d\hat{u}}{dt} &= \Phi^{(II)}\Phi^{(II)T} f(\hat{u}, t, p) \quad t^{(I)} < t \leq t^{(II)} \\
\hat{u}(t_0) &= \Phi^{(I)}\Phi^{(I)T} (u_0(p) - \mu) + \mu
\end{aligned} \tag{2.91}$$

To avoid the loss of significant components of the model state during the transition from one reduced model to another, we patch with the full model dynamics over a short time interval:

$$\frac{d\hat{u}}{dt} = f(u, t, p), \quad t^{(I)} - \varepsilon \leq t \leq t^{(I)} + \varepsilon \tag{2.92}$$

■

We have now introduced three operations to manipulate the subspace basis sets: combination of different orders of the same basis; selective combination of different basis sets by the state component; and by the time interval. For the sufficiently close participating reductions, the end product of the selective model reduction can be treated as a result of the straightforward POD reduction based on some covariance matrix. In that sense, the proposed additional tools do not introduce implementation difficulties not already inherent to the method of snapshots.

## 2.4 IMPROVEMENTS ON THE METHOD OF SNAPSHOTS

We will now discuss the features of the method of snapshots that were either chosen arbitrarily, or never introduced explicitly in the previous sections: namely, weighting of components, weighting of snapshots, and the placement of

snapshots. This will complete the discussion of the aspects of POD-based model reduction sufficient for our work.

While the sensitivity of the reduced model behavior to such features has been widely acknowledged, the corresponding goal-oriented tuning of model reduction is still a relatively new topic, and remains under development ([88], [9]). We provide our perspective on weighting and snapshot selection: consistent with the information obtained by sensitivity analysis, and avoiding standard choices that ignore the dynamics of the model (as noted in [33], [38], [52]).

The POD-based reduction method, as described in Section 2.1, produces a reduced order approximation  $\hat{u}(t)$  on the time interval  $[t_0, T]$ , for the purposes of fast approximate reproduction of behavior of the output function,  $\mathfrak{I}(\hat{u}) \approx \mathfrak{I}(u)$ . The method did not take into account the relative importance of particular time instances or parts of the model state for the output function; in Section 2.2 we suggested measurements of that importance. In Section 2.3 we introduced an example of *intrusive* modification of the method consisting of post-processing the reduced subspace basis for selective treatment of different model elements under reduction.

We shall now explain some of the possible *non-intrusive* modifications of the method that take into account the sensitivity and importance information during the construction of the reduced model.

While the importance of choosing the weights and the snapshots has been acknowledged in the field of study, and some authors have noted the advantages of *goal-oriented* model reduction (for example, [9]), the standard choices of such

features are usually very simple, so our systematic examination of options is an improvement on the existing practices.

#### 2.4.1 WEIGHTING AND METRIC CHANGE, EVENT TARGETING

We shall now discuss the traditional modifications to the process of extracting the essential data from the set of observations, introduced to take into account that model state components, and time intervals from which they were taken may have different importance for the representation of the model behavior.

Historically, the basic technique known as snapshot weighting was suggested as an extension of the idea that the snapshots do not have to be distinct. To represent a greater relative importance of the particular model state at the time  $t = t_i$ , and improve the quality at which it is reproduced by the reduced model, the corresponding snapshot can be repeated several times without modifying the rest of the procedure:

$$U_O = [u(t_1), u(t_2), \dots, u(t_i), \dots, u(t_i), \dots, u(t_i), \dots, u(t_N)] \quad (2.93)$$

If each snapshot  $u(t_i)$  is repeated  $w_i$  times, the optimality condition (2.21) may be rewritten as

$$E_{dist} = \sum_{j=1}^N w_j \|u(t_j) - \hat{u}(t_j)\|_2 \quad (2.94)$$

An immediate generalization is to allow the *snapshot weights*  $w_i$  in (2.95) to be arbitrary nonnegative real numbers, resulting in a more flexible representation of relative importance.

Another generalization of the method of snapshots leads to *component weighting*. To take into account a characteristic of importance  $a_i \geq 0$  assigned to each model state component  $u_i$ , the condition (2.21) may be rewritten as

$$E_{dist} = \sum_{j=1}^N \|u(t_j) - \hat{u}(t_j)\|_{\Lambda} \quad (2.95)$$

where  $\|\cdot\|_{\Lambda}$  is a *weighted metric* with an inner product

$$\langle v, v' \rangle_{\Lambda} = (\Lambda^{1/2}v) \cdot (\Lambda^{1/2}v') \quad (2.96)$$

induced by a diagonal matrix  $\Lambda$  with entries  $a_i$  on the diagonal.

### Weighted proper orthogonal decomposition

The method of snapshots can be adapted to produce a basis corresponding to the weighted optimality conditions (2.94), (2.95). We refer to [34] for the full discussion of the details of *dual-weighted POD method*, and provide a brief exposition here. Given a set of distinct snapshots  $U_o$ , a metric matrix  $\Lambda$  (usually diagonal, but it is sufficient to make it symmetric positive definite), and a (diagonal) normalized weights matrix

$$W = \text{diag}(w_1, w_2, \dots, w_n), \quad \sum_{i=1}^n w_i = 1 \quad (2.97)$$

we formulate a weighted version of the optimality condition. We now seek a basis

$\Phi = (\phi_1, \phi_2, \dots, \phi_k)$  such that

$$E_{dist} = \sum_{j=1}^N w_j \|u(t_j) - \hat{u}(t_j)\|_{\Lambda} = \sum_{j=1}^N w_j \sqrt{\Lambda \|u(t_j)\|^2 - \sum_{i=1}^k \Lambda \langle u(t_j), \phi_i \rangle^2} \quad (2.98)$$

is minimal. We obtain the optimal basis as a solution to an eigenvalue problem

$$C\Lambda\phi_i = \lambda_i\phi_i \quad (2.99)$$

where the weighted covariance matrix is defined as

$$C = (U_o - \mu)W(U_o - \mu)^T \quad (2.100)$$

For the cases where the number of snapshots is smaller than the state dimension, we solve the eigenvalue problem

$$W^{1/2}(U_o - \mu)^T \Lambda(U_o - \mu)W^{1/2} = \lambda'_i \phi'_i \quad (2.101)$$

and find the eigenvalues and eigenvectors of  $C$  from the relationships

$$\begin{aligned} \lambda_i &= \lambda'_i \\ \phi_i &= \frac{1}{\sqrt{\lambda'_i}} U_o W^{1/2} \phi'_i \end{aligned} \quad (2.102)$$

The equations for the reduced model solution  $\hat{u} = \sum_{i=1}^k q_i(t)\phi_i$  are written as

$$\begin{aligned} \frac{d\hat{u}(t)}{dt} &= \Phi\Phi^T \Lambda f(\hat{u}, t) \\ \hat{u}(t_0) &= \Phi\Phi^T \Lambda(p - \mu) + \mu \end{aligned} \quad (2.103)$$

with the coordinates  $q(t)$  satisfying

$$\begin{aligned} \frac{dq}{dt} &= \Phi^T \Lambda f(\Phi q + \mu, t) \\ q(t_0) &= \Phi^T \Lambda(p + \mu) \end{aligned} \quad (2.104)$$

■

## Weights selection

The weights  $w = (w_1, \dots, w_n)$  are selected based on the expected properties of the chemical system. The standard choice in existing literature is to obtain the weights from the sensitivity information of the unmodified reduced model. In our basic experiments, we used the weighting scheme

$$w_i = \left\| \frac{d\mathfrak{S}(u)}{du(t_i)} \right\|_2 = \left\| \Lambda^{-1} \cdot \frac{d\mathfrak{S}(u)}{du(t_i)} \right\|_\Lambda \quad (2.105)$$

evaluated as explained in Section 2.2, normalized by  $w := w / \sum_{i=1}^N w_i$ . Using a complete form of (2.44) results in a weighting scheme

$$w_i = \left\| \frac{d\mathfrak{S}(\hat{u})}{du(t_i)} \right\|_2 \quad (2.106)$$

When the computational budget allows it, model reduction and the subsequent weight estimation (2.106) may be repeated multiple times, with the obtained weights used in the next reduction. Note that starting at the second iteration we will be differentiating a weighted reduction, so the technical details will change. Specifically, the differentiation of the singular value decomposition of the weighted covariance matrix (2.55) will use the values  $A = (U_o - \mu)W(U_o - \mu)^T \Lambda$ ,  $U = \Phi$ . The calculation of sensitivity for the weighted reduced model will include the (schematic) term

$$\frac{d\Phi}{d(W^{1/2} \cdot u \cdot \Lambda^{1/2})} = (\Lambda^{1/2})^{-1} \cdot \frac{d\Phi}{d(\sqrt{w_i} \cdot u(t_i))} \quad (2.107)$$

instead of the term  $\frac{d\Phi}{du}$ .

The expression (2.106) estimates the combined importance of the snapshot  $u(t_i)$  and the attached weight  $w_i$ . We note that because in this estimate the effects of the data, and of the weight are not separated, the sensitivity analysis of the modified reduction procedure may in fact be comparatively less informative. In principle, in the existing framework for finding derivatives, it is possible to formulate and solve an optimization problem leading to the set of weights that best reproduce an output function, but only at a fixed time and *for a fixed ensemble of snapshots*.

We note that unintentional weighting of snapshots may be present even in a direct application of the method, since a sampling of states of the complex systems, with interactions happening at speed of multiple scales, will occasionally produce very similar snapshots. In our practice, this may lead to an implementation problem, related to the *information inflation* issues (of sampling theory). Repetition of a particular model state provides no new information on the correlations of the system and unnecessarily increases the influence of some correlations on the outcome. In this case, a smaller set of snapshots, or a weighting scheme that amplifies the unique snapshots, may improve the performance of the reduced model.

■

### **Metric selection**

The sensitivity of the method of snapshots to the choice of the inner product is noted in [34], [53]. For the model equations, an application of the matrix  $\Lambda$  is



essentially a change of coordinates. However, at the stage of selecting the subspace basis, the metric influences the obtained eigenvalues and eigenvectors, potentially resulting in significant distortion (or improvement) of the reduced model behavior. In the existing literature, the choice of the matrix  $\Lambda$  receives relatively little attention: it is usually selected *a priori*, based on very general expectations of its effects on the reduced model behavior.

According to (2.98), the metric directly determines the relative precision with which the reduced model will reproduce the individual components in the snapshots. This suggests a diagonal form of the metric, and the schemes similar to (2.105), (2.106):

$$\Lambda = \begin{pmatrix} \Lambda_{1,1} & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & \Lambda_{n,n} \end{pmatrix}, \quad \Lambda_{i,i} = \left| \frac{d\mathfrak{S}(u(T))}{du_i(T)} \right| \quad (2.108)$$

$$\Lambda_{i,i} = \left| \frac{d\mathfrak{S}(\hat{u}(T))}{du_i(T)} \right| \quad (2.109)$$

The entries of  $\Lambda$  can be also chosen by inspection of the snapshots. To assess how the explicit characteristics of the chemical system influence the performance of the reduced model, we will use the schemes based on the average amount of the particular component in the system, and the variance of that amount:

$$\Lambda_{i,i} = \frac{1}{N} \sum_{j=1}^N u_i(t_j) \quad (2.110)$$

$$\Lambda_{i,i} = \text{var}[(U_o)_i] = \frac{1}{N} \left| \sum_{j=1}^N u_i(t_j) - \mu_i \right|^2 \quad (2.111)$$

The suggestions for the metric selection presented here are only the first approach, our experiments show the need for a better understanding of the role of the metric. In practice, large deviation of  $\Lambda$  from the identity matrix  $I$  may leads to numerical instability of the reduced model; this implementation concern is possible to detect, but it still invalidates the reduction process. In practice, the schemes (2.108), (2.109), (2.110), (2.111) need to be additionally tuned to the form

$$\Lambda_{stable} = I + \Lambda / \eta \quad (2.112)$$

where  $\eta$  is some large constant. Note that the distinctions in relative importance of model state components are preserved in (2.112).

■

### **Event targeting**

The application of snapshot weighting and metric change allows us to achieve customized representation in the reduced model of selected elements in the full model evolution. We use a somewhat limited, but still widely applicable definition of an *event of interest* as a rectangular region

$$(u_1, u_2, \dots, u_l) \times (T_1, T_2) \in R^n \times [t_0, T] \quad (2.113)$$

or a selection of model state elements on an interval in time; model state components renumbered without the loss of generality.

If a chosen event of interest lies within an identified time interval, and includes only a few model state components, we suggest that the representation of such an event in the reduced model can be amplified or dampened using an *event*

*targeting* approach. The approach consists of 4 steps, performed for each rectangular region such as (2.113):

- assign greater importance to selected components by metric change,
- assign greater importance to the time instances falling into selected interval by snapshot weighting,
- dampen the importance of the rest of the components by metric change,
- dampen the importance of the rest of the snapshots by snapshot weighting.

The implementation for these 4 steps may be as follows:

$$\begin{aligned}
 \Lambda_{i,i} &:= \Lambda_{i,i} \cdot c & 1 \leq i \leq l \\
 w_j &:= w_j \cdot c' & T_1 \leq t_j \leq T_2 \\
 \Lambda_{i,i} &:= \Lambda_{i,i} \cdot 1/c & i > l \\
 w_j &:= w_j \cdot 1/c' & t_j < T_1, t_j > T_2
 \end{aligned} \tag{2.114}$$

where  $c, c'$  are either the empirically chosen constants, or component-dependent, time-dependent estimates of importance similar to (2.105), (2.106); (2.108), (2.109), (2.110), (2.111). Additional steps, such as metric post-processing (2.112) are also possible.

The resulting effect for one rectangular region is visualized in Figure 2.1; note the additional events (light-grey regions) that got amplified, although not as much as the targeted event (dark-grey region). As a remark on implementation, we note that the consequences of snapshot weighting and metric change are not completely predictable: for example, the relative magnitude of their effects on the reduced model behavior is not available *a priori*. It may happen that snapshot weighting

provides excessive amplification of the event, and the metric change that is sufficiently large to compensate for it leads to numerical instability. In practice, that means that the event targeting procedure cannot be made fully automatic: it needs to be additionally tuned with the awareness of the properties of the model.

■

#### 2.4.2 SNAPSHOT PLACEMENT

To apply an unmodified method of snapshots, we did not need to specify how the time instances  $t_1, \dots, t_N$  should be placed. Ideally, we would like to obtain an effective reduced model valid for a long period of time, based on a few snapshots taken from a relatively short time interval. In practice, the states of the full model are obtained by physical measurements, or by integration of the full model. The choice of snapshots may then be limited by the availability of physical sensors, or by the computational budget. Our freedom in choosing the snapshots may be limited to omitting some of the available observations, or adding a few additional ones (by integration of the full model, or by interpolation).

The available literature ([6], [33], [52], [88], etc) lists a number of cases indicating that the performance of the reduced model is sensitive to the choice of snapshots, that a uniform placement of snapshots is not always optimal, and that the omission of some snapshots from a large set may in fact lead to improvement in performance. Constructing additional examples is straightforward: a dense

placement of snapshots in the transient state of the chemical system will very likely spoil the performance of the reduced model.

While a completely automatic optimal snapshot placement technique is not available, the already developed material on sensitivity analysis is sufficient for comparison and gradual improvement of snapshot sets. We suggest to start with a uniformly placed set of snapshots (2.16), and compute the first-order sensitivities

$$S_i = \left| \frac{d\mathfrak{S}(u(T))}{du(t_i)} \right|, \quad i = 1, 2, \dots, N \quad (2.115)$$

If at least one of the values  $S_i$ ,  $S_{i+1}$  is higher than an empirically established threshold  $S$ , the time interval  $(t_i, t_{i+1})$  is assigned  $l$  additional snapshots:

$$U_o := U_o \cup \left\{ u \left( (l-1) \frac{t_{i+1} - t_i}{l} \right) \right\} \quad (2.116)$$

We have mostly experimented with high threshold values for  $S$  and moderate values of  $l$ , resulting in a moderately increased density of snapshots over very few time intervals, and a sparse uniform distribution of snapshots over the rest of the observed time. If the computational budget allowed it, a sensitivity estimate

$$\hat{S}_i = \left| \frac{d\mathfrak{S}(\hat{u}(T))}{du(t_i)} \right|, \quad i = 1, 2, \dots, N \quad (2.117)$$

can be used instead of (2.115).

We note that adding additional snapshots to regions of high importance presents a trade-off between expected quality and observed effectiveness. After a few augmentations like (2.116), the size of the set of snapshots  $U_o$  becomes very

large, capturing more of the information on the model behavior, but also increasing the computational cost of linear algebra operations, and the chance of numerical instability.

To compensate for this latter risk, we suggest rejecting snapshots in the regions of potential instability. Geometrically, any projection to a subspace is a smooth movement of every point. We can predict the consequences of this *a priori* unknown movement on the reduced model solution by examining the consequences of moving the snapshots in an arbitrary direction. The maximal distance of the movement is  $E_{dist}$  (2.21), bounded as shown in (2.32). Given the reduction basis  $\Phi$  obtained from the current set of snapshots, we evaluate the (first few) eigenvalues of the perturbed Jacobian

$$\bar{J}^{(i)} \approx J(u + \varepsilon_1 E_{dist}, t_i + \varepsilon_2 \frac{t_{i+1} - t_i}{2}) = \Phi^T J(u + \varepsilon_1 E_{dist}, t_i + \varepsilon_2 \frac{t_{i+1} - t_i}{2}) \Phi \quad (2.118)$$

for a small sample of values  $-1 < \varepsilon_1 < 1$ ,  $-1 < \varepsilon_2 < 1$ . The eigenvalues are then used to assess the linear stability of the model ODEs in the region. If for some of the Jacobians (2.118) obtained in the neighborhood of the snapshot, the first (dominant) eigenvalues have non-negligible real parts, the snapshot  $u(t_i)$  should be removed from the set  $U_o$ .

The procedure of accumulation and rejection of snapshots may be repeated several times, until snapshot ensemble of moderate size and good stability properties is collected. The smallest effective number of snapshots is the dimension  $k$  of the reduced space: with fewer snapshots, the rank deficiency of the covariance matrix

will not allow us to achieve the alignment (2.33). We suggest that the number of snapshots is too large if many of them have almost identical contents, thus leading to unintended weighting, as discussed in Section 2.4.1. To avoid the repetition of snapshot contents, we suggest an additional measurement of the distance

$$\|u(t_i) - u(t_{i+1})\|_2 \tag{2.119}$$

for the neighboring snapshots.

We have now concluded the description of the basic tools used to improve the model reduction process; in Chapter 7 we apply them as needed to achieve acceptable performance of particular reduced models. In the remainder of the chapter we will list additional suggestions for improvement.

### 2.4.3 ADDITIONAL SUGGESTIONS

We shall now describe a number of modifications to the reduction procedure that resulted in some empirical improvement of performance, but are not applicable widely, because they are too problem-specific, or lack mechanisms of quality assessment and control, or resolve performance issues already better addressed by other techniques. The content of this section should be viewed as a collection of suggestions and open questions for future research. Topics mentioned here include the treatment of multiple timescales of the model dynamics in the context of reduction; processing of data that is numerically unreliable due to imperfect observations, or the integration errors; and direct manipulation of eigenvalues of the covariance matrix using the change of metric.

## Use of slow-fast dynamics

Schemes for the numerical integration of models of atmospheric chemistry (n particularly, represented by stiff ODEs) often include separate treatment of the model state components depending on their rate of change in concentration. We need to provide a general description of the practice before we explain how to include it in the model reduction process.

If by inspection it is possible to partition the individual chemical concentrations into groups with very different magnitudes of the rates of change, it is said that the model exhibits the *slow-fast behavior*; either on the whole simulation interval, or during specific periods of time. In general, the definition is subjective, with thresholds for the slow and fast behavior are set empirically. In our applied problem, though, the distinction is easy to observe, with changes in relative concentration for the fast interactions orders of magnitude larger than for the slow ones.

As a rule, in our applied problems, extremely fast chemical interactions between species typically also end quickly, exhausting the reagents participating in the interaction. The model then tends to evolve more slowly, until the concentration of the reagents builds up again, or a time-dependent external effect starts the fast interaction again. The evolution of an individual specie in time may include short, semi-periodic intervals of fast change in concentration (*fast transient intervals*), and longer intervals with slow, monotonous change. The transient intervals for different species are not exactly concurrent in time; it is more convenient to describe the



regions of the fast and the slow behavior of the model as the *slow* and the *fast* manifolds  $M^S, M^F$ , that is, listings of the slow and the fast model state components  $u^S, u^F$  on different time intervals:

$$M^S = \bigcup u^S(t) \times (T_1, T_2), \quad M^F = \bigcup u^F(t) \times (T_1, T_2) \quad :$$

$$u^S = (u_{i_1}, u_{i_2}, \dots, u_{i_l}), \quad u^F = (u_{j_1}, u_{j_2}, \dots, u_{j_{n-l}}), \quad \left\| \left( \frac{du(t)}{dt} \right)_i \right\| \ll \left\| \left( \frac{du(t)}{dt} \right)_j \right\|, T_1 < t < T_2 \quad (2.120)$$

Note that this definition assumes that the model behavior has been observed at every time instance. In the absence of this information, we approximate the manifolds by rectangular regions aligned with the model state coordinates:

$$M^S \approx \text{span}(e_{i_1}, \dots, e_{i_l}) \times \bigcup (T_1, T_2)$$

$$M^F \approx \text{span}(e_{j_1}, \dots, e_{j_{n-l}}) \times \bigcup (T_1, T_2) \quad (2.121)$$

where  $(e_1, \dots, e_n)$  is the Euclidean basis. In this representation, the status of a slow, or a fast specie is a permanent label.

On each interval  $(T_1, T_2)$ , the model ODEs can be broken up into  $l$  “slow” and  $n - l$  “fast” equations.

$$\frac{du^S}{dt} = f^S(u, t)$$

$$\varepsilon \cdot \frac{du^F}{dt} = f^F(u, t) \quad (2.122)$$

where  $\varepsilon \approx 0$ , and the functions  $f^S : R^l \times (T_1, T_2) \rightarrow R^n$ ,  $f^F : R^{n-l} \times (T_1, T_2) \rightarrow R^n$  produce outputs of approximately the same order of magnitude in every component. The format (2.122) (sometimes refined to also distinguish *fast*, *intermediate* and *slow* species) allows modifications of the integration process for a more stable and

accurate solution: for example, different integration schemes for the slow and the fast species [100], [101]; steady-state assumption  $\frac{du^F}{dt} = 0$  after the end of the transient period [131], or different error tolerances for the slow and the fast manifold.

In the context of model reduction, at least a basic awareness of the slow-fast behavior should be used to guide snapshot placement, selection of weighting, and other modifications of the reduction process.

For our applied problems, the slow and fast manifolds contain information of different complexity and different numerical reliability.

On the fast manifold, the error in the integration of the full model is relatively larger. The covariance information collected primarily on the fast manifold is empirically less reliable (possibly because the correlations between the components observed on the transient interval are only valid for a short period of time). The existence of the fast dynamics requires correcting our schemes of importance assessment of snapshots and components. While high values of derivatives lead us to place more snapshots of greater weight on fast transient intervals, it is more sensible to avoid it. The search for sources of instability suggested in (2.118) usually rejects the snapshots on the fast transient intervals, confirming the suggestion made in [105] that it is better to take snapshots from the slow manifold. Some sources ([1], [88]) suggest an iterative procedure with multiple runs of the reduction process, and adjustments of the snapshot set. While

improvement of the snapshot set via a converged optimization process is a theoretically sound idea, for our purposes it is not computationally efficient.

We experimented with modified weighing schemes that take into account the slow-fast behavior. The idea was to base the weight of the snapshot both on the sensitivity information and on the distance from the fast manifold, both in time and the model state. This approach is potentially attractive, because it can process snapshot information automatically, but not fully developed because we lack an effective description of the fast manifold boundaries; the estimate (2.121) captures too much of the slow manifold.

The steady-state assumption for the species with almost negligible concentrations fits well into the model reduction procedure, as a preprocessing operation on the snapshots. The resulting reduced model solution does not reproduce the steady state behavior; but there is a small improvement in the quality and numerical stability.

The scheme (2.122) can also be used as a setup for selective reduction described in Section 2.3. The fast species have a more complex evolution structure, and may require a basis of relatively greater dimension to represent them; their evolution is better represented if the basis is created from the snapshots taken on the manifold. On the other hand, the slow manifold species behavior can be well reproduced by using a subspace of low dimension, based on uniformly (or even arbitrarily) placed snapshots.

Correct processing of the slow-fast behavior in the full model is an interesting open question in the study of model reduction. It provides a basic example of the need to balance the representation of two types of information: reliable and with few degrees of freedom versus unreliable, complex and inherently multi-dimensional. One practical approach consists of gathering a lot of information on complex behavior (transient intervals), and then filtering it by rejection or dampening of most of the snapshots.

The next topic of discussion is based on an extreme case of slow-fast behavior, where the information obtained on the transient intervals is not reliable at all, needs to be completely rejected, and then replaced with some surrogate data obtained using assumptions on the overall structure of the model.

■

### **Data omission and recovery**

In Section 2.3, we suggested a selective application of different reductions to some of the model state components, or over some time intervals. Another approach consists of selectively omitting the some data altogether and performing a projection to the reduced subspace. The omitted data needed to solve the reduced order equations is then guessed or interpolated based on the additional assumptions. The expected outcome is an improvement in some of the aspects of the reduced model performance at the cost of theoretical optimality. The setup is practical in the cases where some data is either inconsequential, or cannot be reliably reproduced by the reduced order equations.

As noted above, fast manifold is a basic example of the full model element with unreliable data. In the simplest case, the behavior of the species identified as fast is unreliably reproduced during a transient period, and almost constant soon after. A steady-state assumption for fast species may be combined with model reduction. The reduced model equations are written as

$$\begin{aligned} \frac{d\hat{u}^S}{dt} &= \Phi\Phi^T f^S(u, t) & \hat{u}^S(t_0) &= \Phi\Phi^T p^F \\ \frac{d\hat{u}^F}{dt} &= 0 & \hat{u}^F(t_0) &= \Phi\Phi^T \left( \frac{1}{N} \sum_{i=1}^N u_o^F(t_i) + \mu \right) - \mu \\ p^F &= u^F(t_0) \end{aligned} \tag{2.123}$$

Note that this approach to reduction is intrusive: the reduced model (2.123) is aligned with the information contained in the snapshots, but only in the slow components. The fast components are forced to be steady-state, thus their evolution is reproduced by a constant.

We can now recover the lost information using either a POD-based approach called *gappy data recovery* [36], [122], or *Kriging interpolation* [44]. The use of gappy data recovery is motivated by the fact that a least-squares optimal low-dimensional approximation of the data was already constructed during model reduction. Kriging interpolation is a more generic statistical learning approach, known to perform better than gappy data recovery if large portions of data are unreliable.

To indicate the components that contain unreliable or missing data, we create a *mask vector*  $\eta \in R^n$  of zeros and ones:  $\eta_i = 1$  corresponds to the reliable

component of data  $u_i$ ,  $\eta_i = 0$  to an unreliable component. Using the mask vector, we define a *gappy inner product* and a *gappy norm*:

$$\begin{aligned}\langle v, v' \rangle_\eta &= (\eta \cdot v) \cdot (\eta \cdot v') \\ \|v\|_\eta^2 &= \eta \cdot v \cdot \eta \cdot v\end{aligned}\tag{2.124}$$

Let  $g = u(T)$  be a model state vector with unreliable components, and  $\Phi = (\phi_1, \phi_2, \dots, \phi_k)$  the POD basis,  $k \leq n$ . We assume that the basis  $\Phi$  is based on completely reliable data. The gappy POD is a two-stage procedure. First, an intermediate repaired vector  $g^r$  is constructed as a best fit of the reliable components of  $g$  to the basis  $\Phi$ . Specifically, we express the repaired vector in the POD basis:

$$g^r = \sum_i b_i \phi_i\tag{2.125}$$

We define the difference between the original and the repaired vector in the gappy norm, so that only the reliable components are compared:

$$e = \|g - g^r\|_\eta\tag{2.126}$$

The coefficients  $b = (b_1, b_2, \dots, b_k)^T$  are then chosen so that (2.126) is minimal. The critical point

$$\frac{d}{db_i} \|g - g^r\|_\eta = 0\tag{2.127}$$

is found as a solution of the system of linear equations

$$\begin{aligned}
Ab &= B \\
A_{ij} &= \langle \phi_i, \phi_j \rangle_\eta, \quad B = \langle g, \phi_j \rangle_\eta
\end{aligned} \tag{2.128}$$

Now the entries of vector  $g^r$  are used to replace the unreliable data in  $g$ , producing a repaired vector

$$\begin{aligned}
g_i^R &= g_i^r & \eta_i &= 0 \\
g_i^R &= g_i & \eta_i &= 1
\end{aligned} \tag{2.129}$$

Since  $g^r = g + A^{-1}e$ , we will use the value

$$\|A^{-1}\|_2 = \left\| (\Phi\Phi^T)^{-1} \right\|_\eta \tag{2.130}$$

to estimate the quality of the approximation. The expression (2.130) also characterizes the advantages of using a larger POD basis for gappy data recovery.

The procedure for data recovery we have just outlined may be modified to a more flexible fuzzy logic formulation. The mask vector  $\eta$  will consist of values between zero and one, characterizing different degrees of data reliability.

We set  $g_i^R = g_i^r$  in the step (2.129) if the corresponding mask vector component  $\eta_i$  is below some small positive threshold. This approach is equivalent to using a weighted POD basis for gappy data recovery.

As well as for the POD-based gappy data recovery, Kriging interpolation relies on correlations between model state components, as captured by the covariance matrix. The approach does not use the proper orthogonal decomposition. Instead, the unreliable state vector is assumed to be stochastically dependent on the

available snapshot data, and is replaced with a *best linear unbiased estimator*, in the Gauss-Markov sense.

Given a set of snapshots  $U_O = (u(t_1), u(t_2), \dots, u(t_N))$  we replace the unknown, or unreliable state vector  $g = u(T)$  by a repaired version expressed as a weighted average of the snapshot states:

$$g^r = \sum_{i=1}^N w_i u(t_i), \quad \sum_{i=1}^N w_i = 1 \quad (2.131)$$

The covariance function  $c : N^n \rightarrow R$  is represented by the covariance matrix  $C$ :

$$c(u(t_i), u(t_j)) = C_{ij} \quad (2.132)$$

We now define a *Kriging error*

$$E = \sum_{i=1}^N \sum_{j=1}^N w_i w_j c(u(t_i), u(t_j)) - 2 \sum_{i=1}^n w_i c(u(t_i), u(T)) \quad (2.133)$$

In statistical terms, we treat both  $g$  and  $g^r$ , and explain the quantity (2.133) as

$$E = \text{var}[g^r - g] \quad (2.134)$$

The weights  $w = (w_1, w_2, \dots, w_N)$  defining the repaired vector (2.131) are chosen so that  $E$  is minimal, leading to a *simple kriging interpolation*

$$w = C^{-1} \cdot \begin{pmatrix} c(u(t_1), u(T)) \\ \vdots \\ c(u(t_N), u(T)) \end{pmatrix} \quad (2.135)$$

The main difficulty in adapting the Kriging procedure to our case is that the correlations of the snapshots with the unknown state vector are not available.. The simplest suggestion is to set all unknown correlations to be equal to each other:



$$c(u(t_i), u(T)) = \frac{1}{N} \quad (2.136)$$

We may also assume that the strength of correlation depends on the distance  $|T - t_i|$  between the snapshots. Then we set

$$c(u(t_i), u(T)) = \exp(-|T - t_i|) \quad (2.137)$$

and then normalize the obtained weights. Alternatively, we can base the estimate of the correlation on similarity on the reliability of the data, as identified by the mask vector  $\eta$ :

$$c(u(t_i), u(T)) = \exp(-\|u(t_i) - u(T)\|_\eta) \quad (2.138)$$

Since the reliable components of  $g$  should not be replaced with the interpolated values, the final version of the repaired vector  $g^R$  is still obtained by (2.131).

We refer to [44] for additional comments on the performance of POD-based and Kriging recovery procedures. We have now introduced the two methods of data recovery that can be used on the solution of the reduced model equations, to compensate for the unreliable data in snapshots, or to correct solver errors.

■

### Direct eigenvalue editing

Suppose that the (weighted) covariance matrix  $(U_o - \mu)W(U_o - \mu)^T$  (2.100) has positive eigenvalues  $\lambda_1 > \lambda_2 > \dots > \lambda_n$  with the corresponding eigenvectors  $\phi_1, \dots, \phi_n$ . Let the corresponding weighted observation matrix  $(U_o - \mu)W^{1/2}$  have a

singular value decomposition  $(U_o - \mu)W^{1/2} = U\Sigma V^T$ . Then, for an arbitrary set of nonnegative numbers  $\xi_1 > \xi_2 > \dots > \xi_n$  we set

$$\Lambda = KK^T, \quad K = V \begin{pmatrix} \sqrt{\frac{\xi_1}{\lambda_1}} & \dots & 0 \\ \vdots & \sqrt{\frac{\xi_2}{\lambda_2}} & \\ 0 & & \ddots \end{pmatrix} \quad (2.139)$$

Then the matrix  $(U_o + L)W(U_o + L)^T \Lambda$  will have eigenvalues  $\xi_1 > \xi_2 > \dots > \xi_n$  and eigenvectors  $\phi_1, \dots, \phi_n$ . To allow  $\lambda_i \approx 0$ , we replace the corresponding term  $\sqrt{\frac{\xi_i}{\lambda_i}}$  with 1, forcing  $\xi_i = \lambda_i$ .

We suggest running the model reduction procedure until the eigenvalues of the unmodified covariance matrix are obtained. One possible metric change is to increase the distances between the eigenvalues without changing their ordering. We suggest a scheme

$$\xi_i = \varepsilon^{n-i+1} \lambda_i \quad (2.140)$$

for a small constant  $\varepsilon > 0$ .

The quality estimate (2.32) is applicable, resulting in an observation that *in a new metric*, the same quality of alignment can be achieved by a reduced model of smaller dimension. This does not result in an improvement in quality in the Euclidean metric, since the constants  $m, M$  in the metric equivalence relationship

$m\|v\| \leq \|v\|_\Lambda \leq M\|v\|$  are correspondingly the lowest and the highest eigenvalue of  $\Lambda$ ,

$m = \frac{\xi_n}{\lambda_n}, M = \frac{\xi_1}{\lambda_1}$ . If we use the scheme (2.140),  $m = 1, M = \varepsilon^n$ . However, if  $M$  is not

too large, the described metric change is justified by providing a reduced model of smaller dimension, with almost the same performance. We also find that such change results in improved stability. This observation is partially justified by the eigenvectors of secondary importance being well aligned with the trajectories during the fast transient periods.

For very different magnitudes of the original and the edited eigenvalues, the numerical error in (2.133) may lead to a non-positive definite matrix, which cannot be used to define a metric. In that case, we recommend approximation by the nearest positive definite matrix, by a Chen-McInray procedure for finding symmetric positive-definite matrices from imperfect measurements [112], [28]. Omitting theoretical details, to solve the approximate system

$$AX \approx B \tag{2.141}$$

for a positive definite matrix  $X$ , we introduce symmetric decompositions:

$$\begin{aligned} P &= A^T A \\ Q &= B^T B \end{aligned} \tag{2.142}$$

and Schur decompositions:

$$\begin{aligned} P &= U_P (D_P)^2 U_P^T \\ \tilde{Q} &= U_P^T Q U_P \end{aligned} \tag{2.143}$$

We then define

$$\tilde{Q} = D_P U_P^T Q U_P D_P = U_{\tilde{Q}} (D_{\tilde{Q}})^2 U_{\tilde{Q}}^T \quad (2.144)$$

with diagonal matrices  $D_P, D_{\tilde{Q}}$ . The least squares optimal solution of (2.141) is given by

$$X = U_P D_P^{-1} U_{\tilde{Q}} D_{\tilde{Q}} U_{\tilde{Q}}^T D_P^{-1} U_P^T \quad (2.145)$$

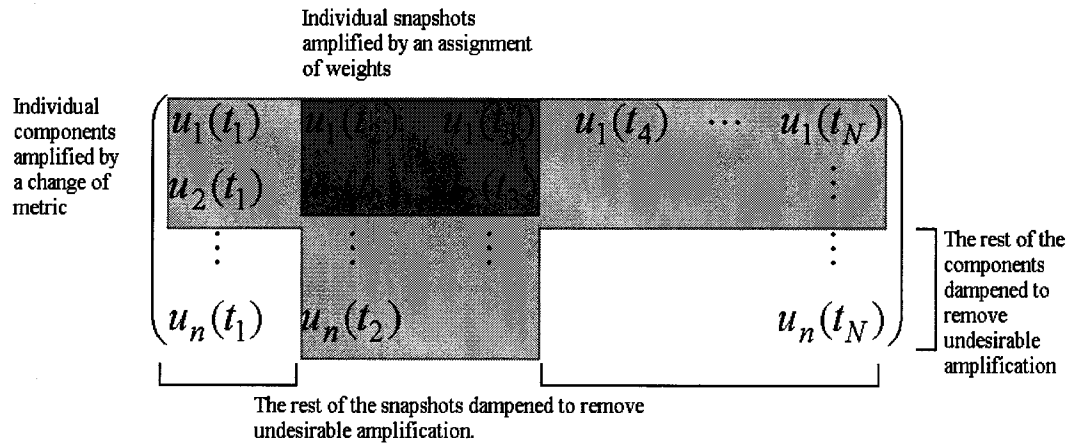
To find a positive definite matrix closest to a given matrix  $B$ , we set  $A = I$ .

As a measurement of quality, we suggest the difference

$$\|\Lambda - \Lambda'\|_2 \quad (2.146)$$

between the matrix  $\Lambda$  obtained by (2.139) and its nearest positive definite approximation  $\Lambda'$  obtained by (2.144). If the value is large, the metric change is numerically inefficient.

■



**Figure 2.1 Targeting an event with two types of weighting**

## CHAPTER 3

### A POSTERIORI ERROR ESTIMATION

To our knowledge, there are no available *a priori* techniques to predict the magnitude of a maximal error introduced by reduction. Because of this, we view the error analysis as a means to characterize and validate the performance of an already constructed reduced model, describe its expected behavior over a range of parameters, or to informally describe the expected performance of a family of reduced models.

An *a posteriori* first-order approach has been developed by Petzold et. al. in [54], [55], [125]. We will now review the available material, simplifying it to fit our basic case where dependence on parameters appears only in the initial conditions. We derive an additional estimate that takes into account both sources of error (perturbation of model inputs and projection) at the same time.

The approach consists of three steps:

- Estimation of the error in the full model solution caused by a perturbation in the initial conditions.
- Search for the directions of the maximal error growth in the parameter space.
- Estimation of how well the reduced model preserves these directions.

The results of the three steps are put together to describe of an error introduced by a combined effect of the approximation by a reduced-order model, and the perturbations in the initial conditions.

Some elements of the approach are computationally expensive, so the method is not effective for the purposes of adaptive reduced model construction and improvement. We note that for many existing tools of model reduction not based on the POD, even such a limited technique is not available.

The estimation of perturbation-induced errors is based on the *adjoint sensitivity analysis*, and *small-sample statistical condition estimation* (SCE). A minimal amount of information required to implement adjoint differentiation of an ODE-based model was provided in Section 2.2; again, we refer to Chapter 4 for additional details.

### 3.1 ERROR INDUCED BY PERTURBATION

We introduce a perturbation  $\delta p$  to the parameters  $p$  of a simple initial value problem

$$\begin{aligned} \frac{du}{dt} &= f(u) \\ u(0) &= p \end{aligned} \tag{3.1}$$

resulting in a perturbed system

$$\begin{aligned} \frac{d\bar{u}}{dt} &= f(\bar{u}) \\ u(0) &= p + \delta p \end{aligned} \tag{3.2}$$

A perturbation error

$$\bar{e}(t) = \bar{u}(t) - u(t) \tag{3.3}$$

can be approximated by a first-order Taylor expansion, resulting in a system of equations

$$\begin{aligned}\frac{d\bar{e}(t)}{dt} &= J(u(t), t)\bar{e} \\ \bar{e}(0) &= \delta p\end{aligned}\tag{3.4}$$

where  $J$  is the Jacobian of the right-side function (2.9), and the trajectory  $u(t)$  is obtained by integrating (3.1).

A straightforward approach to examining the growth of perturbation error in various directions is to solve (3.4) multiple times; impractical for large dimensions of the system. There exists an alternative estimate of the error at an arbitrary fixed time  $T$ , based on the following standard result from statistics [60]. The derivation of the result lies outside of scope of our work.

### Small-sample estimate

For any vector  $y \in R^n$  and a unit vector  $x \in R^n$  chosen randomly and uniformly, the expected value of  $|x^T y|$  is proportional to the norm of  $y$  with a coefficient depending only on the dimension  $n$ :

$$E[|x^T y|] = W_n \|y\|_2\tag{3.5}$$

with the *Wallis factor*  $W_n \approx \sqrt{2/\pi(n-2)}$  formally defined as

$$W_n = \begin{cases} \frac{1 \cdot 3 \cdot \dots \cdot (n-2)}{\pi \cdot 2 \cdot 4 \cdot \dots \cdot (n-1)} & n \text{ odd} \\ \frac{2 \cdot 2 \cdot 4 \cdot \dots \cdot (n-2)}{\pi \cdot 1 \cdot 3 \cdot \dots \cdot (n-1)} & n \text{ even} \end{cases}\tag{3.6}$$



Then the expected value (3.5), found as an average for a small sample of values of  $x$ , may be used to estimate the norm  $\|y\|_2$ :

$$\|y\| \approx \frac{W_\eta}{W_n} \sqrt{\sum_{i=1}^n |x_i^T y|^2} \quad (3.7)$$

with a relative error of size  $\zeta$  occurring with a probability on the order of  $\zeta^{-\eta}$ , making sample size  $\eta = 4$  sufficient for practical purposes. In Chapter 7 we occasionally use samples of the size  $n$  for factor importance analysis; the limiting factor on error estimation is not the computational budget, but rather the development cost.

■

To use the estimate (3.7) we obtain the values  $x^T e(T)$  by adjoint differentiation of (3.4). We introduce an adjoint variable  $\bar{e}^*$  as a solution of a homogeneous system

$$\begin{aligned} \frac{d\bar{e}^*}{dt} &= -J(u(t), t)\bar{e}^* \\ \bar{e}^*(T) &= x \end{aligned} \quad (3.8)$$

And integrate it backwards in time to obtain an expression

$$x^T \bar{e}(T) = (\bar{e}^*)^T(0) \delta p \quad (3.9)$$

Since the solution of the adjoint system that is independent of  $\delta p$ , it can be used in the estimate

$$\bar{e}(t) \approx \frac{W_\eta}{W_n} \sqrt{\sum_{i=1}^n \left( (\bar{e}_i^*)^T(0) \delta p \right)^2} \quad (3.10)$$

where  $\bar{e}_i^*(t)$  are solutions of the family of the adjoint equations (3.8) based on a random sample  $(x_1, x_2, \dots, x_\eta)$ :

$$\begin{aligned} \frac{d\bar{e}_i^*}{dt} &= -\nabla_u \mathfrak{J} \bar{e}_i^* \\ \bar{e}_i^*(T) &= x_i \end{aligned} \tag{3.11}$$

### Directions of highest sensitivity

A standard tool for finding the directions of the largest error growth is the *singular vector analysis* (SV) (suggested in the studies of dynamical systems predictability), based on the dominant eigenspace of the observed error matrix  $E_O = (E(t_1), E(t_2), \dots, E(t_N))$  with columns defined by a solution of

$$e(t) = E(t) \delta p \tag{3.12}$$

at a selection of time instances. Note the similarity to the development of the POD-based model reduction: the dominant directions of error growth are singular vectors of the error correlation matrix  $EE^T$ .

Alternatively, a single leading eigenvector can be found as a solution to an optimization problem. We will then define the maximal error created by a unit perturbation in the initial conditions as

$$\bar{e}^{\max} = \sup_{\|\delta p\|=1} \|\bar{e}(t)\| \tag{3.13}$$

and use an SCE approach to reformulate the problem as

$$\bar{e}^{\max}(T) \approx \max_{\|\delta p\|=1} \frac{W_\eta}{W_n} \sqrt{\sum_{i=1}^{\eta} (|x_i^T \bar{e}_i| \delta p)^2} \tag{3.14}$$

Using a fixed sample  $(x_1, x_2, \dots, x_\eta)$ , we treat the expression on the right side of (3.14) as a function of  $\delta p$

$$\bar{E}(\delta p) = \sqrt{\sum_{i=1}^{\eta} (|x_i^T \bar{e}_i | \delta p)^2} \quad (3.15)$$

requiring only  $\eta$  integrations of the adjoint system (3.8) to evaluate. This function has an explicit derivative

$$\frac{d\bar{E}}{d\delta p} = \frac{\sum_{i=1}^{\eta} (|x_i^T \bar{e}_i | \delta p)^2}{\sqrt{\sum_{i=1}^{\eta} (|x_i^T \bar{e}_i | \delta p)^2}} \quad (3.16)$$

allowing us to find  $\min_{\|\hat{\phi}\|=1, \hat{\phi} \in P} \bar{E}$  by performing a search in the parameter space  $P$ .

A framework for solving such optimization problems is provided in Chapter 5.

■

### 3.2 ERROR INDUCED BY MODEL REDUCTION

We shall now estimate the error introduced solely by the model reduction. Suppose that a reduced model is created by projecting the model ODEs with the matrix  $\Pi = \Phi\Phi^T$  and adjusting to zero-mean ensemble with a shift term  $-\mu$ . We define the reduction error

$$\theta(t) = u(t) - \hat{u}(t) \quad (3.17)$$

Then  $\theta(t)$  satisfies the system

$$\begin{aligned} \frac{d\theta}{dt} &= u(t) - \hat{u}(t) \\ \theta(0) &= -(I - \Pi)(p - \mu) \end{aligned} \quad (3.18)$$

where  $I$  is the identify matrix. We approximate the right-hand side of (3.26) by the a first-order Taylor expansion:

$$\frac{d\theta}{dt} \approx J(\hat{u}(t), t)\theta - (I - \Pi)f(\hat{u}(t), t) \quad (3.19)$$

Let  $E \in R^{n \times n}$  be the solution of a (fundamental) homogeneous equation

$$\begin{aligned} \frac{dE}{dt} &= J(\hat{u}, t)E \\ E(0) &= I \end{aligned} \quad (3.20)$$

where  $I$  is the identity matrix. Note that (3.18) can be written as

$$\theta(T) = - \int_0^T E(T)E^{-1}(\tau)(I - \Pi)f(\hat{u}(\tau), \tau)d\tau - E(T)(I - \Pi)(p - \mu)d\tau \quad (3.21)$$

Technically, this expression is a more complicated form of (3.12). We can now arrive at an estimate of the value  $x^T \theta(T)$  for a randomly chosen unit vector  $x$ :

$$x^T \theta(T) = - \int_0^T x^T E(T)E^{-1}(\tau)(I - \Pi)f(\hat{u}(\tau), \tau)d\tau - x^T E(T)(I - \Pi)(p - \mu)d\tau \quad (3.22)$$

We can now introduce an adjoint variable  $e^*$  as a solution to the ODE

$$\begin{aligned} \frac{de^*}{dt} &= -(J(\hat{u}(t), t))^T e^* \\ e^*(T) &= x \end{aligned} \quad (3.23)$$

A combination of (3.20), (3.22) and (3.23) results in an expression

$$x^T \theta(T) = - \int_0^T (e^*)^T(\tau)(I - \Pi)f(\hat{u}(\tau), \tau)d\tau - (e^*)^T(T)(I - \Pi)(p - \mu)d\tau \quad (3.24)$$

The corresponding SCE estimate is written as

$$\begin{aligned}
\|\theta(T)\| &\approx \frac{W_\eta}{W_n} \sqrt{\sum_{i=1}^{\eta} |x_i^T \theta(T)|^2} = \\
&= \frac{W_\eta}{W_n} \sqrt{\sum_{i=1}^{\eta} \left| \int_0^T (e_i^*)^T(\tau)(I - \Pi)f(\hat{u}(\tau), \tau)d\tau - (e_i^*)^T(T)(I - \Pi)(p - \mu)d\tau \right|^2} \quad (3.25)
\end{aligned}$$

and requires  $\eta$  evaluations of the adjoint ODE (3.23).

### Combined error

For a complete report on the available tools on the propagation of error under model reduction, we also derive an estimate for the model with combined effects of model reduction and perturbation in the initial conditions. We denote the solution of such model as  $\bar{u}$ . We write the overall error  $\Theta(t)$  as the difference

$$\Theta(t) = u - \bar{u} = u - \hat{u} + \bar{u} - u + \bar{u} - \hat{u} = \underbrace{\theta(t)}_{(1)} + \underbrace{\bar{e}(t)}_{(2)} + \underbrace{\bar{\theta}(t)}_{(3)} \quad (3.26)$$

The terms (1), (2), (3) in (3.26) are described, correspondingly, by ODEs

$$\begin{aligned}
\frac{d\theta}{dt} &= J(\hat{u}(t), t)\theta - (I - \Pi)f(\hat{u}(t), t) \\
\theta(0) &= -(I - \Pi)(p - \mu)
\end{aligned} \quad (3.27)$$

$$\begin{aligned}
\frac{d\bar{e}(t)}{dt} &= J(u(t), t)\bar{e} \\
\bar{e}(0) &= \delta p
\end{aligned} \quad (3.28)$$

$$\begin{aligned}
\frac{d\bar{\theta}}{dt} &= J(\hat{u}(t), t)\bar{\theta}(t) \\
\bar{\theta}(0) &= -(I - \Pi)\delta p
\end{aligned} \quad (3.29)$$

Note that integration of the full and the reduced model equations is required to obtain  $u(t)$ ,  $\hat{u}(t)$ . The corresponding (fundamental) adjoint ODEs are (3.8), (3.23) and (3.8) again:

$$\begin{aligned}\frac{de^{(1)*}}{dt} &= -(J(\hat{u}(t), t))^T e^{(1)*} \\ e^{(1)*}(T) &= x\end{aligned}\tag{3.30}$$

$$\begin{aligned}\frac{d\bar{e}^{(2)*}}{dt} &= -J(u(t), t)\bar{e}^{(2)*} \\ \bar{e}^{(2)*}(T) &= x\end{aligned}\tag{3.31}$$

$$\begin{aligned}\frac{de^{(3)*}}{dt} &= -J(\hat{u}(t), t)e^{(3)*} \quad \Rightarrow \quad e^{(3)*} = e^{(2)*} \\ e^{(3)*}(T) &= x\end{aligned}\tag{3.32}$$

The assembled SCE estimate is written as

$$\begin{aligned}\|\Theta(T)\| &= \frac{W_\eta}{W_n} \sqrt{\sum_{i=1}^{\eta} |Z(i)|^2}, \\ Z(i) &= \left[ \int_0^T (e_i^{(1)*})^T(\tau)(I - \Pi)f(\hat{u}(\tau), \tau)d\tau - (e_i^{(1)*})^T(T)(I - \Pi)(p - \mu)d\tau \right] + \dots \\ &+ \left[ (\bar{e}_i^{(2)*})^T(0)\delta p \right] + \dots \\ &+ \left[ (\bar{e}_i^{(3)*})^T(0)\delta p - \int_0^T (e_i^{(3)*})^T(T)(I - \Pi)\delta p d\tau \right]\end{aligned}\tag{3.33}$$

The obtained expression is bulky, but more convenient than three separate SCE estimates. It requires  $2\eta$  evaluations of the adjoint ODEs.

### Comparing the responses of full and reduced models

We shall now complete the chapter with the last remark on *a posteriori* characterization of the model to which both perturbation and reduction were applied. To compare the perturbation responses of the full and reduced models  $u, \hat{u}$ , we construct the error correlation matrices  $E_o^T E_o$  and  $\hat{E}_o^T \hat{E}_o$ , where the perturbation-induced error is observed for both full and the reduced models:

$$E_o = (\bar{e}(t_1), \bar{e}(t_2), \dots, \bar{e}(t_N)) \quad (3.34)$$

$$\hat{E}_o = (\hat{e}(t_1), \hat{e}(t_2), \dots, \hat{e}(t_N)) \quad (3.35)$$

For the error correlation matrices, we denote the corresponding sets of eigenvectors as

$$\begin{aligned} V &= (v_1, v_2, \dots, v_n) \\ \hat{V} &= (\hat{v}_1, \hat{v}_2, \dots, \hat{v}_n) \end{aligned} \quad (3.36)$$

According to SV (see comments before (3.12)), the dominant eigenspaces of the correlation matrices contain the major parts of the evolution of a perturbation-induced error. A computationally practical *similarity index*  $S$  is suggested in [55] as a measure of the difference between the first few eigenvectors from the sets (3.21):

$$I_1 = \sum_{i=1}^l \sum_{j=1}^l \langle v_i, \hat{v}_j \rangle \quad (3.37)$$

Another similarity index compares the errors induced in the full model, and in the reduced model, by a small perturbation along the dominant eigenvector of the full model:

$$I_2 = \min \left( \frac{\max_{\|\hat{\phi}\|=\varepsilon, \|\hat{\phi}\|_{v_1}} \bar{\hat{e}}}{\max_{\|\hat{\phi}\|=\varepsilon, \|\hat{\phi}\|_{v_1}} \bar{e}}, \frac{\max_{\|\hat{\phi}\|=\varepsilon, \|\hat{\phi}\|_{v_1}} \bar{e}}{\max_{\|\hat{\phi}\|=\varepsilon, \|\hat{\phi}\|_{v_1}} \bar{\hat{e}}} \right) \quad (3.38)$$

The expression (3.38) can be evaluated using the previously derived SCE estimates.

■

In conclusion of this chapter, we note that the presented error estimation techniques can be used to ensure (experimentally) that a constructed reduced model adequately approximates the magnitudes of the model state components and reject

the reduced model that may fail to do so. Some of the available sources suggest alternative approaches to *a posteriori* error estimation [82], [47]. As far as we know, they are approximately equivalent in performance to the material presented above.



## CHAPTER 4

### ADJOINT ANALYSIS

In this chapter, we present technical information on adjoint differentiation of the output functions of ODE-based models with respect to parameters. The task is well described in the available literature: see, for example, [22], [69]. The adjoint differentiation of ODEs was used in the material of Section 2.2, Chapter 3. We now provide a more complete and specific description, sufficient for our tasks of factor importance analysis and iterative optimization.

#### Adjoint operators in Hilbert spaces

The theoretical basis for adjoint differentiation is a part of Hilbert space theory. For *Hilbert spaces*  $H_1, H_2$  with an inner product  $\langle \cdot, \cdot \rangle$  and a continuous linear operator  $M : H_1 \rightarrow H_2$  there exists a unique, continuous linear operator  $M^* : H_2 \rightarrow H_1$  such that for any two vectors  $v_1 \in H_1, v_2 \in H_2$

$$\langle Mv_1, v_2 \rangle = \langle v_2, M^*v_1 \rangle \quad (4.1)$$

The operator  $M^*$  is called the adjoint of  $M$ . When the operator  $M$  is represented by multiplication by a matrix,  $Mv = M \cdot v$ ,  $M^*$  corresponds to the complex conjugate of that matrix,  $M^*v = M^* \cdot v$ . We write the relationship between the model state  $u$  and the parameters  $p$  as

$$u = G(p) \quad (4.2)$$

In practice, the relationship is defined implicitly, by the generic model equations (2.1); we have to assume it is differentiable. A perturbation  $\delta u$  in the state, due to a

perturbation  $\delta p$  in the parameters can be approximated by a first-order Taylor expansion, written as an inner product

$$\delta u = \langle g, \delta p \rangle \quad (4.3)$$

where  $g = \nabla_p G$ , the result of differentiating the right-side of (4.2) with respect to parameters. The expression (4.3) is called the *tangent linear equation* to (4.2). The variation of the cost function  $\mathfrak{J}$  with respect to  $u$  is

$$\delta \mathfrak{J} = \langle \nabla_u \mathfrak{J}, \delta u \rangle = \langle \nabla_u \mathfrak{J}, g \delta p \rangle = \langle g^* \nabla_u \mathfrak{J}, \delta p \rangle \quad (4.4)$$

for an operator  $g^*$  adjoint to  $g$ . Compare with the variation of the cost function with respect to  $p$

$$\delta \mathfrak{J} = \langle \nabla_p \mathfrak{J}, \delta p \rangle \quad (4.5)$$

to obtain the adjoint equality for the gradient:

$$\nabla_p \mathfrak{J} = g^* \nabla_u \mathfrak{J} \quad (4.6)$$

The term  $\nabla_u \mathfrak{J}$  is available from the definition of the cost function, but the adjoint operator  $g^*$  is generally not explicit. In the particular case where the state  $u$  is subject to a system of ODEs (2.6) with parametric dependence in the initial conditions, the adjoint variable  $u^* = g^* u$  is solution to the (adjoint) ODEs:

$$\frac{du^*}{dt} = - \left( \frac{df}{du} \right)^T u^* = - (J(u(t), t))^T u^* \quad (4.7)$$

with the initial conditions chosen to satisfy a version of (4.4).

■

The decision to use the adjoint method is based mainly on the dimensions of the problem inputs and outputs and the available computational budget. The advantage of the adjoint method is that the adjoint system only needs to be solved once to produce all components of the gradient. Given a scalar output function and multiple input parameters, adjoint differentiation is more efficient than the direct differentiation shown in Section 2.2, (2.56) that would require additional integration of the model for each parameter.

We note that while the theoretical foundations of adjoint differentiation are straightforward, the development of the adjoint operator of a given problem can sometimes be challenging. In the following sections, we provide details on differentiation of our basic ODE (reaction model), and PDE (reaction-transport model). We refer to the existing extensive literature for additional examples: [24], [35], [97], [119].

#### 4.1 DIFFERENTIATION OF AN ODE MODEL

So far, we have not explained how the adjoint term  $g^*$  is obtained, and did not justify the expression (4.7). We will now explain how to implement differentiation for the cost function given in the generalized form of (2.6):

$$\mathfrak{J}(p) = \int_{t_0}^{\tau} \|u(t) - u_o(t)\|^2 dt \quad (4.8)$$

Here  $u_o(t)$  is the exact observations of the model, and  $\|\cdot\|$  is an arbitrary norm. We will assume that the ODE has the form (1.2), with  $p = u(t_0)$ . We use the notation

$$\mathfrak{J}(p) = G_N(u(T)) + \int_{t_0}^T g(u(t)) dt \quad (4.9)$$

$$g(u(t)) = \|u(t) - u_o(t)\|^2, \quad t_0 < t < T; \quad G_N(u(T)) = \|u(T) - u_o(T)\|^2$$

A separate term  $G_N$  is introduced so that the presented material may be easily adapted for the case where the output function is defined only on a single time instance:

$$\mathfrak{J} = \|u(T) - u_o(T)\| \quad (4.10)$$

### First-order adjoint differentiation

We shall now repeat the steps (4.2) - (4.6). For a perturbation  $\delta p$  in the parameters  $p$ , the corresponding perturbation in the state is  $\delta u = \bar{u} - u$  satisfies an ODE

$$\begin{aligned} \frac{d\delta u}{dt} &= f(\bar{u}, t) - f(u, t) \\ \delta u(t_0) &= \delta p \end{aligned} \quad (4.11)$$

Expanding the expression by first-order Taylor series, we rewrite (4.11) as

$$\begin{aligned} \frac{d\delta u}{dt} &= \frac{\partial f(u, t)}{\partial u} \delta u \\ \delta u(t_0) &= \delta p \end{aligned} \quad (4.12)$$

The expression (4.4) is written as

$$\delta \mathfrak{J} = \langle \nabla_p \mathfrak{J}, \delta p \rangle = \langle \nabla_{u(T)} G_N, \delta u(T) \rangle + \int_{t_0}^T \langle \nabla_{u(t)} g(u(t)), \delta u(t) \rangle dt \quad (4.13)$$

To find the non-explicit terms in the expression, we multiply (4.12) by a so far unspecified *adjoint variable*  $u^*$  and integrate by parts over the interval  $t_0 \leq t \leq T$ :

$$\left\langle \delta u, u^* \right\rangle_{t_0}^T - \int_{t_0}^T \left\langle \delta u, \frac{du^*}{dt} \right\rangle dt = \int_{t_0}^T \left\langle \frac{\partial f(u, t)}{\partial u}, u \right\rangle dt \quad (4.14)$$

We shall now define  $u^*$  as a solution of the adjoint problem with a terminal (instead of the initial) condition:

$$\begin{aligned} \frac{du^*}{dt} &= - \left( \frac{\partial f(u, t)}{\partial u} \right)^T u^* + \nabla_{u(t)} g(u(t)) \\ u^*(T) &= \nabla_{u(T)} G_N(u(T)) \end{aligned} \quad (4.15)$$

We substitute  $u^*$  into (4.14), written for a particular case  $\delta u = \delta u(0) = \delta p$ :

$$\left\langle \delta p, -u^*(t_0) \right\rangle = \left\langle \delta u(T), -u^*(T) \right\rangle + \int_{t_0}^T \left\langle \nabla_{u(t)} g(u(t)), \delta u(t) \right\rangle dt \quad (4.16)$$

Then (4.13) can be simplified to an expression

$$\nabla_p \mathfrak{J} = -u^*(t_0) \quad (4.17)$$

which is a particular case of (4.6).

The complete computational procedure required to find the derivative of the output function with respect to the initial conditions consists of integrating the *direct* model equations (2.6) forward in time, to find the trajectory  $u(t)$ , and then integrating the *adjoint* model equations (4.15) backward in time to find the adjoint initial state  $u^*(t_0)$ . If the integration of (4.15) requires variable time step in the numerical solver, then at least for some time instances the exact model state  $u(t_i)$  will have to be interpolated from the nearby values.

■

## Second-order adjoint differentiation

The developed procedure can be repeated again to obtain a selection of Hessian vector products (i.e. directional second derivatives), resulting in extension of the sensitivity analysis of Section 2.2 to the second order. The evaluation of every component is not feasible in practice due to computational expense. We refer to the [120] for additional information.

We introduce another perturbation  $\delta p$  into the initial conditions of the direct model (2.6) and note the effect on the adjoint model (4.15). We redefine the terms  $\delta u, \delta u^*$  as the resulting perturbations in the direct and the adjoint variables  $u, u^*$  correspondingly.

The perturbations are described by the equations

$$\begin{aligned} \frac{d\delta u}{dt} &= \left( \frac{\partial f}{\partial u} \right) \delta u \\ \delta u(0) &= p \end{aligned} \tag{4.18}$$

$$\begin{aligned} \frac{d\delta u^*}{dt} &= - \left( \frac{\partial f(u, t)}{\partial u} \right)^T u - \left( \frac{\partial^2 f(u, t)}{\partial u^2} \right)^T u^* + \nabla_u^2 g(u(t)) \cdot \delta u \\ \delta u^*(T) &= 0 \end{aligned} \tag{4.19}$$

The expression (4.19) is called the *second order adjoint model* to (2.6). We denote perturbed adjoint variable by  $\bar{u}^* = u^* + \delta u^*$ , and the perturbed initial conditions by  $\bar{p} = p + \delta p$ . The derivative of the output function (4.8) with respect to the initial conditions can be approximated to the first order by an expansion

$$\nabla_{\bar{p}} \mathfrak{J}(\bar{u}) = \nabla_p \mathfrak{J} + \nabla^2 \mathfrak{J} \cdot \delta p \tag{4.20}$$

By the first-order adjoint analysis

$$\nabla_{\bar{p}} \mathfrak{J}(\bar{u}) = \bar{u}^*(t_0) \quad (4.21)$$

Applying (4.20), (4.21) for a specific case  $t = t_0$  we conclude

$$\delta u^*(t_0) = \nabla^2 \mathfrak{J} \cdot \delta p \quad (4.22)$$

The term  $\delta u^*(t_0)$  is obtained by integrating (4.18) backwards in time. Setting the vector  $\delta p$  to the values of coordinate vectors  $e^{(1)}, e^{(2)}, \dots, e^{(n)}$  with

$$e^{(i)}_i = 1; \quad e^{(i)}_j = 0: \quad i \neq j \quad (4.23)$$

we obtain, from (4.21) a list of  $n$  equations each defining a column of the Hessian  $\nabla^2 \mathfrak{J}$ .

■

## 4.2 DIFFERENTIATION OF A PDE MODEL

While most of our analysis of the chemical transport is done for the model discretized to a system of ordinary differential equations, this may not be the best form for differentiation. We shall now develop an adjoint operator for the full reaction-transport PDE (1.4), with parametric dependence in the initial conditions:

$$\begin{aligned} \frac{du}{dt} &= -\nabla \cdot (\omega u) + \nabla \cdot (K \nabla u) + f(u, t), \quad t_0 \leq t \leq T \\ u(x, t_0) &= u_0(x) = p(x), \quad x \in \Omega \end{aligned} \quad (4.24)$$

For the conditions on the boundary  $\partial\Omega$  we shall use either the prescribed value (Dirichlet), or the zero normal derivative (Neumann) forms:

$$u(x,t) = u_{\partial\Omega}(x,t), \quad x \in \partial\Omega \quad (4.25)$$

$$\frac{\partial u(x,t)}{\partial \bar{n}} = 0, \quad x \in \partial\Omega \quad (4.26)$$

where  $\bar{n}$  is the outward normal vector of the spatial domain.

We define the output function using an un-discretized form of (4.8):

$$\mathfrak{J}(p) = \int_{t_0}^T \int_{\Omega} \|u(x,t) - u_o(x,t)\|^2 dt \quad (4.27)$$

and use the notation

$$\mathfrak{J}(p) = \int_{t_0}^T \int_{\Omega} g(u(t)) dt \quad (4.28)$$

$$g(u(t)) = \|u(x,t) - u_o(x,t)\|^2$$

For the perturbation in the state  $\delta u(x,t)$ , the corresponding perturbation in the output function can be expressed to the first order as

$$\delta \mathfrak{J} = \langle \nabla_u \mathfrak{J}, \delta u \rangle = \int_{t_0}^T \int_{\Omega} \delta u(x,t) \cdot \frac{\partial g}{\partial u} dx dt \quad (4.29)$$

where  $\delta u$  is due to a perturbation in the initial conditions  $\delta u(x,0) = \delta p$ , and satisfies the tangent linear model to (4.24):

$$\begin{aligned} \frac{d\delta u}{dt} &= -\nabla \cdot (w\delta u) + \nabla \cdot (K\nabla \delta u) + \left(\frac{df}{du}\right)\delta u \\ \delta u(x, t_0) &= \delta p(x) \end{aligned} \quad (4.30)$$

with boundary conditions corresponding to (4.25), (4.26):

$$\delta u(t, x) = 0 \quad (4.31)$$

or



$$\frac{\partial u(x,t)}{\partial \vec{n}} = 0 \quad (4.32)$$

For convenience of notation, we record (4.30) as

$$\begin{aligned} \frac{d\delta u}{dt} &= L(u)\delta u \\ L(u) : L^2((t_0, T) \times \Omega) &\rightarrow L^2((t_0, T) \times \Omega) \end{aligned} \quad (4.33)$$

i.e. record the right-side of the PDE as an operator action. We multiply (4.33) by an adjoint variable  $u^*$  and integrate over  $(t_0, T) \times \Omega$ :

$$\int_{t_0}^T \int_{\Omega} \frac{d\delta u}{dt} \cdot u^* dx dt = \int_{t_0}^T \int_{\Omega} L(u)\delta u \cdot u^* dx dt \quad (4.34)$$

By (4.1), there exists an operator  $L^*$  such that

$$\int_{t_0}^T \int_{\Omega} L(u)\delta u \cdot u^* dx dt = \int_{t_0}^T \int_{\Omega} \delta u \cdot L^*(u)u^* dx dt \quad (4.35)$$

To specify  $L^*$ , we integrate the left-side of (4.35) by parts:

$$\int_{t_0}^T \int_{\Omega} (-\nabla \cdot (w\delta u)) \cdot u^* dx dt = \int_{t_0}^T \int_{\Omega} \delta u \cdot (\nabla \cdot (wu^*)) dx dt + \int_{t_0}^T \int_{\partial\Omega} w\delta u u^* \cdot \vec{n} dx dt \quad (4.36)$$

$$\begin{aligned} \int_{t_0}^T \int_{\Omega} (\nabla \cdot (K\nabla \delta u)) \cdot u^* dx dt &= \\ &= \int_{t_0}^T \int_{\Omega} \delta u \cdot (\nabla \cdot (K\nabla u^*)) dx dt + \int_{t_0}^T \int_{\partial\Omega} \delta u \cdot K\nabla u^* \cdot \vec{n} dx dt + \int_{t_0}^T \int_{\partial\Omega} K\nabla \delta u \cdot u^* \cdot \vec{n} dx dt \end{aligned} \quad (4.37)$$

where  $\vec{n}$  is the outward normal vector of the spatial domain. The last term of the expression has already been evaluated in differentiation of an ODE:

$$\int_{t_0}^T \int_{\Omega} \left( \frac{df}{du} \delta u \right) \cdot u^* dx dt = \int_{t_0}^T \int_{\Omega} \delta u \cdot \left( \left( \frac{df}{du} \right)^T u^* \right) dx dt \quad (4.38)$$

The expressions (4.36), (4.37), with appropriate boundary conditions for the adjoint variable, provide a form of the adjoint differential operator

$$L^*(u)u^* = \nabla \cdot (wu^*) + \nabla \cdot (K\nabla u^*) + \left(\frac{df}{du}\right)^T u^* \quad (4.39)$$

We shall now integrate (4.34) by parts, using (4.35):

$$\int_{t_0}^T \int_{\Omega} \left( \frac{du^*}{dt} + L^*(u)u^* \right) \cdot \delta u dx dt = \int_{\Omega} \delta u \cdot u^* dx \Big|_{t_0}^T \quad (4.40)$$

The adjoint variable  $u^*(x, t)$  is defined as the solution of the adjoint PDE

$$\begin{aligned} \frac{du^*}{dt} &= -L^*(u)u^* - \frac{\partial g}{\partial u}, \quad t_0 < t < T, \quad x \in \Omega \\ u^*(x, T) &= 0, \quad x \in \Omega \\ u^*(x, t) &= 0, \quad \nabla u^* \cdot \vec{n} = 0, \quad x \in \partial\Omega \end{aligned} \quad (4.41)$$

Then a special case of (4.27) with  $\delta u(0, x) = \delta p(x)$  simplifies to

$$\delta \mathfrak{S} = \int_{\Omega} \delta p(x) \cdot u^*(t_0) dx \quad (4.42)$$

and the derivative is expressed by

$$\frac{d\mathfrak{S}}{dp(x)} = u^*(t_0, x) \quad (4.43)$$

## CHAPTER 5

### OPTIMIZATION

As explained in the material of Chapter 2, while the process of creating the reduced model has essentially the same computational cost as generation of snapshots, the factor importance analysis and improvement of performance of the reduced model may require multiple integrations of the direct and the adjoint model equations. The computational cost may be unacceptably large, making the use of the full model more attractive. That is why model reduction is mainly motivated by applications that require multiple uses of the same model. In such applications, the reduced model replaces the full model multiple times.

One such application is the study of sensitivities of the full model: it is easier to compute derivatives, or sample the solutions in the reduced space. In addition, the POD-reduced model has an advantage of already identified and separated dominant factors.

Another application is the iterative solution of optimization problems, where a reduced model of sufficient quality is used to estimate the full model state and its derivatives (of first, but also, possibly, of the second order) at every iteration. This idea is central to our work.

Our central subject of study is the initial conditions optimization. The task is usually to choose the initial conditions of the ODE system so that the best match of an output function to some expected values can be achieved. This class of problems (i.e. improvement of the model performance by making the distance of the

numerical solution from the observed data optimally small) arises in the process of data assimilation, as explained in Chapter 1. The material presented here may also be generalized to a wider class of optimization problems.

Given a general system of algebraic-differentiation equations (2.1) with a particular case of an ODE

$$\begin{aligned}\frac{du}{dt} &= f(u, t, p) \\ u(t_0) &= u_0(p)\end{aligned}\tag{5.1}$$

we introduce an output function

$$\begin{aligned}\mathfrak{J} &= \mathfrak{J}(u(t), p) : R^{n \times n} \rightarrow R \\ \mathfrak{J} &= \int_{t_0}^T \|u(t) - u_o(t)\|^2 dt\end{aligned}\tag{5.2}$$

also known as *cost*, or *merit* function in this context. Since the solution  $u(t)$  is determined by the choice of the parameters, we formulate the *parameter optimization problem* on the parameter set  $P$ :

$$\begin{aligned}\min_p \mathfrak{J}(u(t), p) \\ p = (p_1, p_2, \dots) \in P \subseteq R^n\end{aligned}\tag{5.3}$$

subject to (5.2).

In principle, the parameter space  $P$  may be equal to the whole state space  $R^n$ . In our practical problems we expect that each parameter component  $p_i$  is restricted to an interval of empirically reasonable values:

$$(1 - \varepsilon_i)(p^{(0)})_i \leq p_i \leq (1 + \varepsilon_i)(p^{(0)})_i\tag{5.4}$$

for a set of constants  $(\varepsilon_1, \varepsilon_2, \dots)$  of moderate magnitude, resulting in representation of  $P$  as a rectangular region.

To simplify the notation, we will redefine  $\mathfrak{I}, F$  as expressions dependent only on the parameters:

$$\begin{aligned}\mathfrak{I} &:= \mathfrak{I}(p) : R^n \rightarrow R \\ F &:= F(p) : R^n \rightarrow R^n\end{aligned}\tag{5.5}$$

We can also reformulate (5.2) so that the parameters only appear in the initial conditions:

$$\begin{aligned}\frac{du}{dt} &= f(u, t) \\ u(t_0) &= p\end{aligned}\tag{5.6}$$

The fact that (5.2) is replaced with (5.6) causes no loss of generality, and is of little practical significance in our main application.

In the scope of our study, we treat (5.4) as a generic model-constrained nonlinear optimization problem, that is at least locally convex and solvable by sequential unconstrained minimization techniques (extensively described in [37], [41], [46], [57], etc). While in practice our techniques also apply to the problems that are only approximately convex, in the scope of this study we are mainly interested in the problems that can be solved in reasonable time using the simplest iterative search based on first-order derivative information.

We will now introduce a group of such simple descent methods for nonlinear, model constrained optimization. We will discuss applying the same method to the problems based on the full, and on the reduced versions of the model.

While some derivative information is available analytically, we will mostly use adjoint differentiation, as explained in Chapter 4. We will identify a number of possibilities to include model reduction into an otherwise computationally expensive optimization process. The practical applications of the material are provided in Chapter 7.

## 5.1 DESCENT DIRECTION METHODS

For an optimization problem (5.4), subject to (5.6), we would like to find a critical point  $p_{\min}$  such that  $\nabla_p \mathfrak{J} = 0$ . Once the point is located, it will be identified as a minimum by convexity.

A widely applied *feasible direction method* constructs a sequence  $p^{(k)}$ ,  $k = 0, 1, 2, \dots$  of approximations to the minimizing set of values. For each of the points  $p^{(k)} \in P$ , starting with the initial guess  $p^{(0)}$ , we choose a direction vector  $d^{(k)} \in P$  that satisfies the descent condition

$$\nabla_p \mathfrak{J}(p) \cdot d^{(k)} < 0 \tag{5.7}$$

We shall now briefly explain some of the most commonly used methods.

### Iterative descent search

In the most basic case, the feasible direction vector  $d^{(k)}$  is defined as the *direction of steepest descent*

$$d^{(k)} = -\nabla_p \mathfrak{J}(p) \tag{5.8}$$

The next point  $p^{(k+1)}$  in the sequence is obtained from  $p^{(k)}$  by following the feasible direction for a small positive distance  $a^{(k)}$ :

$$p^{(k+1)} = p^{(k)} + a^{(k)} d^{(k)} \quad (5.9)$$

with a necessary condition

$$\mathfrak{I}(p^{(k)} + a^{(k)} d^{(k)}) < \mathfrak{I}(p^{(k)}) \quad (5.10)$$

In effect, we find the minimizer as a linear combination of the feasible direction vectors:

$$p_{\min} = p^{(0)} + \lim_{k \rightarrow \infty} \sum_{i=1}^k a^{(i)} d^{(i)} \quad (5.11)$$

If at least an approximation of the cost function Hessian  $\nabla^2 \mathfrak{I} = \nabla_p (\nabla_p \mathfrak{I})$  is available, we can instead define  $d^{(k)}$  as a *Newton-Rhapson direction*

$$\nabla^2 \mathfrak{I} \cdot d^{(k)} = -\nabla_p \mathfrak{I}(p) \quad (5.12)$$

resulting in a faster convergence. For a faster version of (5.11), it may be acceptable to evaluate only the diagonal entries of the Hessian, and replace the rest with zeros [48].

An optimal value of the scalar  $a^{(k)}$  is found by *line search*, i.e. by solving a one-dimensional optimization problem

$$\min_{a^{(k)} > 0} \mathfrak{I}(p^{(k)} + a^{(k)} d^{(k)}) \quad (5.13)$$

with the corresponding condition for the critical point

$$\frac{d\mathfrak{I}(p^{(k+1)})}{da^{(k)}} = 0 \quad (5.14)$$

Depending on the computational budget, the value  $\alpha^{(k)}$  may also be obtained using a second-order approximation:

$$\frac{d\mathfrak{I}(p^{(k+1)})}{d\alpha^{(k)}} \approx (\nabla_p \mathfrak{I})^T d^{(k)} + \alpha^{(k)} (d^{(k)})^T (\nabla_p^2 \mathfrak{I})(d^{(k)}) \quad (5.15)$$

resulting in an expression

$$\alpha^{(k)} = \frac{-(\nabla_p \mathfrak{I})^T d^{(k)}}{(d^{(k)})^T (\nabla_p^2 \mathfrak{I})(d^{(k)})} \quad (5.16)$$

In practice, however, it may be more effective to find an approximate value  $\alpha^{(k)}$  by direct search; for example, by division of the interval  $0 \leq \alpha^{(k)} \leq 1$  into subintervals, and evaluations of  $\mathfrak{I}(p^{(k)} + \alpha^{(k)} d^{(k)})$  on their ends. In computational cost, it is approximately equivalent to (5.16) with derivatives evaluated by finite differences.

■

### Conjugate gradient method

A more sophisticated option is the *conjugate gradient* search. It is essentially an extension of the steepest descent method (5.8) based on minimization of the

residual  $R_k = p^{(0)} - \sum_{i=0}^k \alpha^{(i)} \nabla_p \mathfrak{I} d^{(i)}$  over the space spanned by feasible directions

$d^{(1)}, d^{(2)}, \dots, d^{(k)}$  [48]. The method requires evaluating

$$\begin{aligned} p^{(k+1)} &= p^{(k)} + \alpha^{(k)} d^{(k)} & r^{(k+1)} &= r^{(k)} - \alpha^{(k)} \nabla_p \mathfrak{I} d^{(k)} \\ \beta^{(k+1)} &= \frac{\left( (r^{(k+1)})^T \cdot (r^{(k+1)}) \right)}{\left( (r^{(k)})^T \cdot (r^{(k)}) \right)} \\ d^{(k+1)} &= r^{(k+1)} + \beta^{(k+1)} d^{(k)}, & d^{(0)} &= p^{(0)} \\ \alpha^{(k+1)} &= \frac{\left( (r^{(k)})^T \cdot (r^{(k)}) \right)}{\left( (d^{(k)})^T \cdot (\nabla_{p^{(k)}} \mathfrak{I}) \cdot (d^{(k)}) \right)} \end{aligned} \quad (5.17)$$



where  $r^{(i)}$  are the residual vectors, and  $\beta^{(i)}$  are scalars.

The method can be adjusted to include the Newton search direction (5.12):

$$\alpha^{(k+1)} = \left( - \left( \nabla_{p^{(k)}} \mathfrak{F} \right)^T \cdot \left( r^{(k)} \right) \right) / \left( \left( d^{(k)} \right)^T \cdot \left( \nabla^2 \mathfrak{F} \right) \cdot \left( d^{(k)} \right) \right) \quad (5.18)$$

This requires a different definition of the residual  $\beta$ ; there are several options, for example

$$\beta^{(k+1)} = \left( \left( r^{(k+1)} \right)^T \cdot \left( r^{(k+1)} - r^{(k)} \right) \right) / \left( \left( r^{(k)} \right)^T \cdot \left( r^{(k)} \right) \right) \quad (5.19)$$

■

### Derivative information and convergence

We will now refer to a number of standard results on the expected performance of iterative searches for the minimizer; the reason for this material is to identify bounds on convergence that can later be observed for the full and the reduced models in comparison of their performance. Consider a generic descent search  $p^{(k+1)} = p^{(k)} + \alpha^{(k)} d^{(k)}$ , with a descent direction  $d^{(k)}$  that does not deviate very much from direction of the steepest descent  $-\nabla_p \mathfrak{F}(p^{(k)})$ ; specifically, we require bounds

$$\begin{aligned} - \left( d^{(k)} \right)^T \cdot \nabla_p \mathfrak{F}(p^{(k)}) &\geq \delta \left\| d^{(k)} \right\| \cdot \left\| \nabla_p \mathfrak{F}(p^{(k)}) \right\| \\ K_1 \left\| \nabla_p \mathfrak{F}(p^{(k)}) \right\| &\leq \left\| d^{(k)} \right\| \leq K_2 \left\| \nabla_p \mathfrak{F}(p^{(k)}) \right\| \end{aligned} \quad (5.20)$$

for some constants  $0 < \delta \leq 1$ ,  $0 < K_1 \leq K_2$ . Note that the characterization (5.20) covers the methods where the feasible direction is obtained from the direction of

steepest descent by multiplication by a positive definite matrix. Let the scalar  $\alpha^{(k)}$  be an approximate solution of the minimization problem (5.13):

$$\mathfrak{I}(p^{(k)} + (\alpha^{(k)} + \delta\alpha)d^{(k)}) \leq \mathfrak{I}(p^{(k)}) \quad (5.21)$$

for small positive values of  $\delta\alpha$ . By local convexity, the Hessian  $\nabla_p^2 \mathfrak{I}(p)$  is symmetric positive definite in a neighborhood of the minimizer. We refer to [48], [57], [109] for the following property.

### Theorem 5.1 Bounds on Hessian

There exist positive constants  $m, M$  (correspondingly the minimal and the maximal values of the Rayleigh coefficient of the Hessian) such that

$$m\|d\|^2 \leq d^T \nabla_p^2 \mathfrak{I}(p) d \leq M\|d\|^2 \quad (5.22)$$

for an arbitrary vector  $d$  and point  $p$ . The constants also bound the output function and its first derivative, by the following expressions:

$$\frac{m}{2}\|p - p_{\min}\|^2 \leq \mathfrak{I}(p) - \mathfrak{I}(p_{\min}) \leq \frac{M}{2\|p - p_{\min}\|^2} \quad (5.23)$$

$$m\|p - p_{\min}\| \leq \nabla_p \mathfrak{I}(p) \leq M\|p - p_{\min}\| \quad (5.24)$$

■

Theorem 5.1 leads to the following linear convergence result.

### Theorem 5.2 Convergence of iterative search

If a descent sequence  $p^{(0)}, p^{(1)}, p^{(2)}, \dots$  is contained in a compact subset  $B(p_{\min})$  of some convex set, then it converges to the minimum point  $p_{\min}$ . In particular, there exist constants  $K > 0$ ,  $0 < L < 1$  such that

$$\mathfrak{I}(p^{(k)}) - \mathfrak{I}(p^{(k+1)}) \geq K \|\nabla_p \mathfrak{I}(p^{(k)})\|^2 \quad (5.25)$$

$$\mathfrak{I}(p^{(k+1)}) - \mathfrak{I}(p_{\min}) \leq L(\mathfrak{I}(p^{(k)}) - \mathfrak{I}(p_{\min})) \quad (5.26)$$

Setting  $D = \mathfrak{I}(p^{(0)}) - \mathfrak{I}(p_{\min})$ , we also have the estimate in terms of previously defined  $m, M$  :

$$\mathfrak{I}(p^{(k+1)}) - \mathfrak{I}(p_{\min}) \leq DL^k \quad (5.27)$$

$$\|p^{(k+1)} - p_{\min}\|^2 \leq \frac{2D}{m} L^k \quad (5.28)$$

$$\|\nabla_p \mathfrak{I}(p^{(k+1)})\|^2 \leq \frac{2M^2 D}{m} L^k \quad (5.29)$$

■

We refer to [109] for a similar characterization of the quadratic convergence of the Newton search (5.13).

### **Theorem 5.3 Convergence of Newton-Rhapson search**

If the derivative of the output function satisfies the Lipschitz condition

$$\|\nabla_p \mathfrak{I}(p) - \nabla_p \mathfrak{I}(p')\| \leq \gamma \|p - p'\|, \quad \gamma < 1 \quad (5.30)$$

and the Hessian satisfies

$$\|(\nabla_p^2 \mathfrak{I}(p))^{-1}\| \leq \beta \quad (5.31)$$

$$\|(\nabla_p^2 \mathfrak{I}(p))^{-1} \nabla_p \mathfrak{I}(p)\| \leq \nu \quad (5.32)$$

Then the Newton descent sequence defined by (5.13) converges to a critical point  $p_{\min}$ , and at every step  $k$

$$\|p^{(k)} - p_{\min}\| \leq v \frac{h^{2^k-1}}{1-h^{2^k}}, \quad h = \frac{v\beta\gamma}{2} < 1 \quad (5.33)$$

■

It is instructive to consider a quadratic approximation of the output function  $\mathfrak{J}(p) \approx A + Bp + Hp^2$ . For this form of the output function, the quantities  $m, M$  that appear in the convergence estimates (5.28), (5.29) are correspondingly the lowest and the highest eigenvalue of the constant Hessian  $H$ ; the ratio  $\frac{M}{m}$  is the condition number of the Hessian. Furthermore, the conjugate gradient search converges on the minimizer exactly, and in a number of steps equal to  $n$ , the dimension of the parameter space [48].

Based on the reasoning about a quadratic approximation, the convergence properties of an optimization problem approximately depend on the condition number of the Hessian, the magnitude of the Hessian; and additionally on the dimension of the space in which the search is performed. Thus we expect faster convergence for the relatively lower condition number, Hessian magnitude and dimension of the problem.

While an *a priori* formal statement on convergence of the iterative searches for the minimization problems subject to *reduced versions* of the model is not available, we believe that the observed improvement in these three characteristics is not accidental. Additional remarks on the need for further research are provided in Section 5.2.

While the output function (1.6), (4.8), (5.3) is convex in  $u$  by triangular inequality for the norm  $\|\cdot\|$ , it is not necessarily convex in  $p$ . The convergence of the iterative search also depends on whether the first guess for the initial conditions  $p^{(0)}$  is sufficiently good for local convexity.

■

## 5.2 REDUCED MODEL IN OPTIMIZATION

The usual justification of using a reduced model instead of the full model is as follows. If the factor importance analysis has successfully identified the most prominent features of the model-defined output function, and an enhanced POD-based reduction method has successfully preserved such features in the behavior of the reduced model, then the reduced model and the full model are effectively equivalent. Once the reduced model is constructed, the calculations required for evaluating the model, obtaining derivatives by direct or adjoint method, performing factor importance analysis by interpolating models, etc, are performed relatively faster due to smaller dimension.

The computational savings at each step of the descent optimization method may add up to a significant improvement in efficiency. In addition, there is an empirical evidence that for a sufficient quality of the reduced model, a descent search converges in relatively fewer steps. More specifically, while an effective characterization of the relationship between the derivatives of the full and the projected reduced versions of the model is not available, empirically the magnitudes

of the derivatives, and the condition number of the Hessian decrease after reduction, leading to tighter inequalities in Theorems 5.2, 5.3.

Motivated by the expectations of better performance, we arrive at the following practical question. Given that we have identified some of the most important states and state space directions in the model evolution, and constructed a reduced model which preserves the states and directions, can we apply this information to perform a significantly cheaper search for the minimizer?

The material presented up to this point leads to suggesting that, given a POD-based reduced model solution  $\hat{u}(t)$  approximating, with sufficient quality, the full model (5.6) with solution  $u(t)$  with initial conditions  $p$ , the problems of minimizing the cost function  $\mathfrak{J}$

$$\min_{p \in P} \mathfrak{J}(u) \tag{5.34}$$

$$\min_{p \in \hat{P}} \mathfrak{J}(\hat{u}) \tag{5.35}$$

are qualitatively equivalent; this is informally understood as “have approximately the same answer”. We do not require the full cost function  $\mathfrak{J}$  and the reduced cost function  $\hat{\mathfrak{J}} = \mathfrak{J}(u)$  to coincide for all values of  $p$ , and will tolerate a moderate error in  $p_{\min}$ .

To make the definition efficient, we suggest making an additional requirement that only a short iterative search is needed to find a true minimizer  $p_{\min}$  starting from an approximate minimizer  $\hat{p}_{\min}$  as the first guess. In practice, the

difference this makes is trivial, but it covers the situation where the value of the minimizer cannot be recovered based on the reduced model alone, since  $p_{\min} \notin \hat{P}$ .

The main implementation issue is the choice of the scheme according to which a reduced model (or multiple reduced models) will replace the full model in the iterative search (partially, or completely). In the next two sections, we introduce the possible schemes, varying depending on the extent to which the model reduction is used. The material is straightforward, and relies on the already developed mathematical content.

### 5.2.1 SINGLE REDUCED MODEL

Suppose we have constructed a reduced model, and verified (using material of Chapters 2 and 3) that it has a solution of sufficient quality,  $\hat{u}(t) \approx u(t)$ , and correctly reproduces the sensitivities of the output function,  $\nabla_p J(u) \approx \nabla_p J(\hat{u})$ . Then, instead of the optimization problem (5.34), we can solve one of the two version of (5.35): either use the full model equations, but restrict the region of search by reduction:

$$\min_{p \in \hat{P}} \mathfrak{J}(u) \tag{5.36}$$

or use the reduced model equations (and the region of search will be restricted automatically):

$$\min_{p \in P} \mathfrak{J}(\hat{u}) \sim \min_{p \in \hat{P}} \mathfrak{J}(\hat{u}) \tag{5.37}$$

Since the result of iterative search  $\hat{p}_{\min}$  is at best very close (but still distinct) from the solution to (5.35), an additional search for the final answer may be performed:

$$\min_{p \in P} \mathfrak{J}(u): \quad p^{(0)} = \hat{p}_{\min} \quad (5.38)$$

The scheme (5.36) requires a short explanation.

### Reduction of parameter space

The minimal use of model reduction in optimization consists of accepting that the projection  $\Pi = \Phi\Phi^T$  approximately preserves the descent directions in the model state space, and performing the search for the optimal initial conditions  $p_{\min}$  based on the full model equations, but only in the reduced space

$$\hat{P} = \Phi\Phi^T P \quad (5.39)$$

Note that the result of the converged search is going to be the projection of the true minimizer onto the reduced space. Since the approach does not formally require  $\hat{u}(t) \approx u(t)$ , the reduced model does not have to be constructed at all, except for the purposes of analysis and improvement of the basis  $\Phi$ .

The steepest descent and the Newton search directions (5.8), (5.12) are written, correspondingly, as

$$d^{(k)} = -\Phi\Phi^T \nabla_p \mathfrak{J}(u) \quad (5.40)$$

$$d^{(k)} = -\Phi\Phi^T \left( \nabla^2 \mathfrak{J}(u) \right)^{-1} \nabla_p \mathfrak{J}(u) \quad (5.41)$$

with the derivatives computed by adjoint differentiation of the full model.

This approach requires the least development effort, and does not depend on the stability of the reduced model. On the other hand, the improvement of the



iterative optimization process is due only to the smaller dimension of the search space. We note that reducing only the parameter space is very appropriate for the tasks that require large-scale sampling of  $\mathfrak{I}(p)$ ,  $p \in P$ , for example, for the purposes of factor importance analysis.

■

If the scheme (5.37) is used, the integrations required at each step of the search are performed in the reduced-order subspace. In particular, the adjoint differentiation required to obtain  $\nabla_p \mathfrak{I}(\hat{u})$  is performed as follows. For the reduced direct ODEs

$$\begin{aligned} \frac{d\hat{u}}{dt} &= \Phi \Phi^T f(\hat{u}, t) \\ \hat{u}(t_0) &= \Phi \Phi^T (p - \mu) + \mu \end{aligned} \tag{5.42}$$

the adjoint variable  $\hat{u}^*$  satisfies the ODEs

$$\begin{aligned} \frac{d\hat{u}^*}{dt} &= - \left( \frac{\partial f}{\partial \hat{u}} \right)^T \Phi^T \Phi \hat{u}^* \\ \hat{u}^*(T) &= \Phi \Phi^T \nabla_{\hat{u}^*(T)} G_N(T) \end{aligned} \tag{5.43}$$

resulting in

$$\nabla_p \mathfrak{I}(\hat{u}) = \hat{u}^*(t_0) \tag{5.44}$$

We set

$$\hat{u}^*(t) = \sum_i \hat{q}_i^*(t) \phi_i = q \Phi \tag{5.45}$$

and solve the projected equations:

$$\begin{aligned}\frac{d\hat{q}^*}{dt} &= -\Phi^T \left( \frac{\partial f}{\partial \hat{u}} \right)^T \Phi \hat{q}^* \\ \hat{q}^*(T) &= \Phi^T \nabla_{\hat{u}^*(T)} G_N(T)\end{aligned}\tag{5.46}$$

The approach using a single reduced model has an inherent weakness of being valid only locally, at an unknown distance from the minimizer. At best, the available reduced model was validated against a small set of full model solutions based on a sample of values of the initial conditions. The reduced model is going to be used, however, on a trajectory of a descent method, the steps  $\hat{p}^{(i)}$  of which eventually become a dense sample in the neighborhood of the point  $\hat{p}_{\min}$ , relatively far from the original guess. In terms of the unknown surface learning theory, the efficiency of the reduced model as a data compression tool may deteriorate, due to exposure to new data that did not participate in compression [93]. This possibility calls for a characterization of the performance of the reduced model at a given step of the optimization process.

We note a restriction on using sophisticated schemes of error estimation: calculations on the scale of SCE-based method (described in Chapter 3) will take away the computational advantage gained by model reduction.

We suggest enhancing the iterative search by occasionally integrating the full model based on the current version of the initial conditions  $p^{(k)}$ , perhaps over only a part of the time interval  $(t_0, T)$ . Specifically, we measure the performance of the reduced model at step  $k$  by comparing the value  $\mathfrak{F}^{(k)}$  defined by

$$\begin{aligned}\mathfrak{S}^{(k)} &= \int_{t_0}^{t_1} \|u(t) - u_o(t)\|^2 dt \\ \frac{du}{dt} &= f(u, t) \\ u(t_0) &= p^{(k)}\end{aligned}\tag{5.47}$$

with the similar quantity  $\hat{\mathfrak{S}}^{(k)}$  already available at this step:

$$\begin{aligned}\hat{\mathfrak{S}}^{(k)} &= \int_{t_0}^{t_1} \|\hat{u}(t) - u_o(t)\|^2 dt \\ \frac{du}{dt} &= f(u, t) \\ u(t_0) &= p^{(k)}\end{aligned}\tag{5.48}$$

The choice of  $t_1$  depends on the available computational budget. In the case of the models of atmospheric chemistry exhibiting semi-periodic behavior, the interval  $(t_0, t_1)$  should include at some fast transient behavior.

As a measure of control, we observe the differences

$$\Delta\mathfrak{S}^{(k)} = \left| \hat{\mathfrak{S}}^{(k)} - \mathfrak{S}^{(k)} \right|\tag{5.49}$$

$$\Delta u^{(k)} = \left| \int_{t_0}^{t_1} \|\hat{u}(t)\| dt - \int_{t_0}^{t_1} \|u(t)\| dt \right|\tag{5.50}$$

and reject the reduced model if they exceed an experimentally established threshold.

If the model was rejected, a particular time interval at which the reduced model encountered performance problems can be identified by examining the differences

$$e(t) = u(t) - \hat{u}(t)\tag{5.51}$$

Alternatively, if the computational budget allows it, we can set  $t_1 = T$ , and measure the approximate descent direction  $\nabla_p \mathfrak{J}^{(k)}$ , assembled component-wise:

$$\left(\nabla_p \mathfrak{J}^{(k)}\right)_i \approx \frac{\mathfrak{J}^{(k+1)} - \mathfrak{J}^{(k)}}{\left(p^{(k+1)} - p^{(k)}\right)_i} \quad (5.52)$$

and compare it with the accumulated descent

$$D^{(k)} = \sum_{i=1}^k \alpha^{(i)} d^{(i)} \quad (5.53)$$

We then measure the angle

$$\gamma = \cos^{-1} \left( \frac{D^{(k)} \cdot \nabla_p \mathfrak{J}^{(k)}}{\|D^{(k)}\| \|\nabla_p \mathfrak{J}^{(k)}\|} \right) \quad (5.54)$$

and reject the model if  $\gamma$  exceeds an experimentally established threshold, which we set to be approximately constant for a sufficiently large value of  $k$  (for the first few steps of the method we allow the descent directions to be very different).

For a more flexible performance characterization, we can use pairs of thresholds on the values  $\Delta \mathfrak{J}^{(k)}$ ,  $\Delta \mathbf{u}^{(k)}$ ,  $\gamma$ :

$$\begin{aligned} \Delta \mathfrak{J}^{(k)} &\leq \Delta \mathfrak{J}^{(k)}_{low} \leq \Delta \mathfrak{J}^{(k)}_{high} \\ \Delta \mathbf{u}^{(k)} &\leq \Delta \mathbf{u}^{(k)}_{low} \leq \Delta \mathbf{u}^{(k)}_{high} \\ \gamma &\leq \gamma_{low} \leq \gamma_{high} \end{aligned} \quad (5.55)$$

The idea is to identify the step on the trajectory as problematic when a ‘low’ threshold is exceeded, and to abort the iterative search when a ‘high’ threshold is exceeded.

In practice, it may be possible to identify by inspection when the answer obtained via reduction-based search (5.37) is unreasonable: either the value  $\hat{p}_{\min}$

lies outside of the region (5.4), or reduced model did not reproduce the critical point correctly, and  $\|\nabla_p \mathfrak{J}\|_{p=p_{\min}} \gg 0$ . In that case, the measurements (5.49), (5.50), (5.54) may be applied retroactively, to identify a which step the reduction-based search has diverged from the correct trajectory.

Depending on the dimension of the model and the computational effort of adjoint differentiation, it may be more efficient to reject and revise the used reduction during the intermediate steps rather than evaluate the quality of the final step and then completely restart the process. Now that we have a simple scheme of rejecting a reduced model based on poor performance, we have to decide what to replace it with. We discuss some of the options in the next section.

### 5.2.2 MULTIPLE REDUCED MODELS

Suppose that a reduction-based search (5.37) has failed the suggested empirical tests, and was become inefficient somewhere between the steps  $k_1$  and  $k_2$  of the search. The complete information on the search trajectory up to step  $k_2$  is available. We then set

$$p^{(0)} = p^{(k)} \tag{5.56}$$

for some  $k_1 \approx k < k_2$ , and restart the iterative search, but using a different model.

The simplest suggestion is to switch to using the full model dynamics, as in (5.36). On the other extreme, the most sophisticated (and computationally expensive) choice is to switch to a reduced model constructed specifically to

perform better on the parameters in the neighborhood of  $p^{(k_1)}, p^{(k_2)}$ . If due to availability of additional experimental data, or a sufficient computational budget, we have access to a set of distinct snapshots collections  $U_o = [u(t_1), u(t_2), \dots]$ , we choose the one based on the initial conditions  $p = u(t_0) \approx u(t_1)$  that are the closest to  $p^{(k)}$ .

Ideally, we would like to construct a hierarchy of the versions of the reduced model: the first one assuming no *a priori* knowledge about the performance of a model under reduction, each of rest tuned using results of factor importance analysis on a previous version. Due to constraints on computational cost, the number of models in the hierarchy cannot be very large, and the factor importance analysis can only be very elementary.

Based on the argument that a subspace basis of higher dimension captures more relevant features (accepted in [95], [117]; disputed for some case studies in [6], [33]), we suggest preparing a collection of reduced models of different dimensions based on the same set  $(\phi_1, \phi_2, \dots, \phi_n)$  of the covariance matrix eigenvectors. The idea is then to use a higher-dimensional model after a low-dimensional one failed.

A more flexible approach is to build reduced models of improved local validity. This requires obtaining additional observational data, or generating additional full model snapshots based on the point in the parameter space at which the search has failed. This requires integration of the full model with the initial

conditions  $u(t_0) = p^{(k)}$ . If we accept a reasonable reduced model quality  $u(t) \approx \hat{u}(t)$  valid at least until the step  $k_1$  of the method, we can instead use a cheaper set of snapshots

$$\hat{U}_o = [\hat{u}(t_1), \hat{u}(t_2), \dots] \quad (5.57)$$

based the reduced model solution with  $\hat{u}(t_0) = p^{(k_1)}$ .

Finally, we can use the fact that the initial conditions, and the model states at any other time  $t \neq t_0$  are taken from the same space  $R^n$ , and apply the method of snapshots to the set

$$U_o = [p^{(k_0)}, p^{(k_0+1)}, \dots, p^{(k_1-1)}, p^{(k_1)}] \quad (5.58)$$

producing a reduced model that optimally reproduces the search trajectory between some step  $k_0$  and the step  $k_1$ .

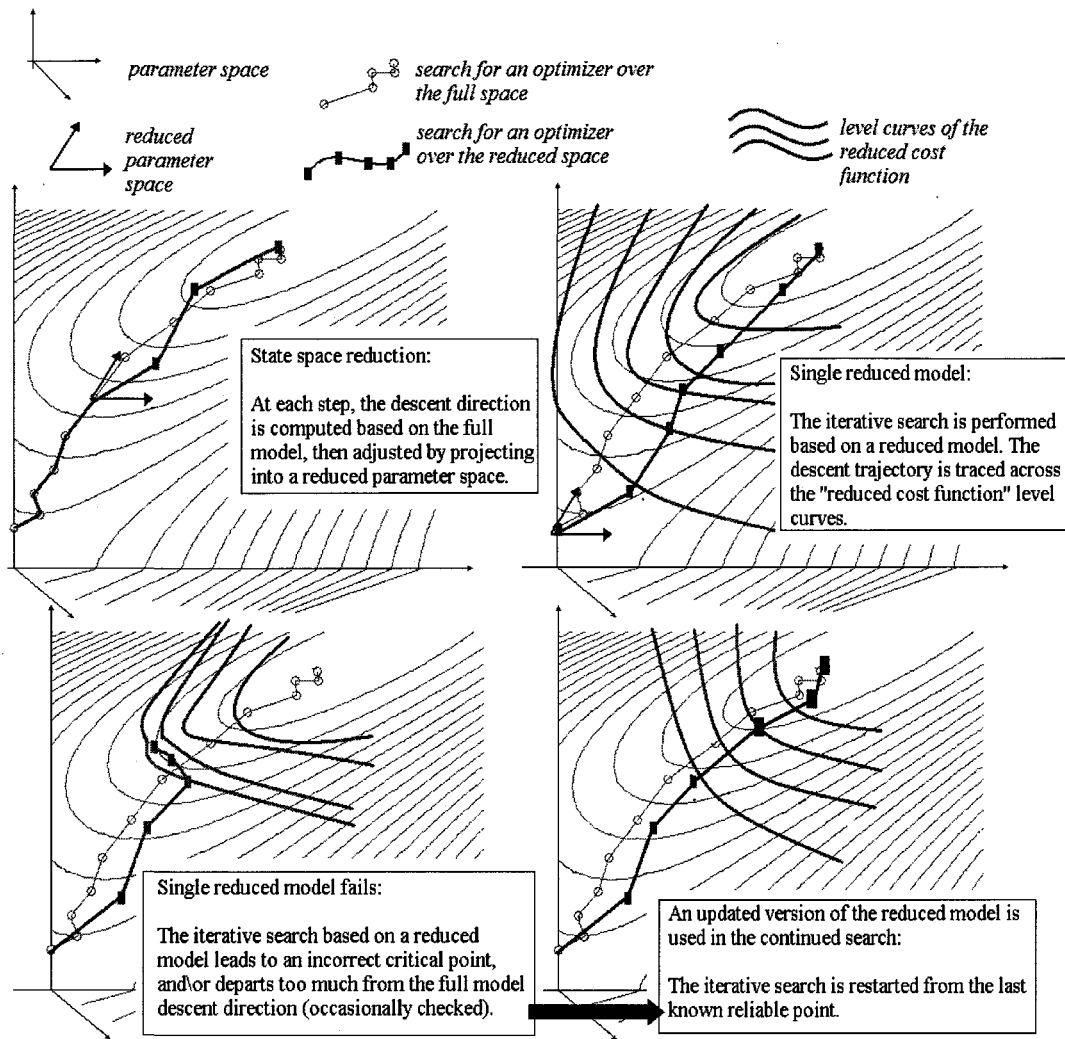
The idea of building an adaptive sequence of reduced models to solve large-scale problems has received some attention in recent literature. For example, Ravindran [88] uses a sequence of revised versions of the reduced model to solve a control problem: at each step, the snapshots for the next reduction are taken from the observations on the model state corresponding to the current set of controls. In terms of our schemes, the idea is equivalent to solving the reduced optimization problem (5.37), generating new snapshots as in (5.57) with the initial conditions equal to the best available estimate for  $p_{\min}$ , and repeating the steps until the procedure converges. For our applied problems, such a long search is not effective,

because the possible improvement in quality is not justified by the increased computational cost.

In [8], [10], Bashir et. al. use considerations similar in content to our Chapter 3; they generate new snapshots based on integrating the model with initial conditions equal to the dominant eigenvector of the error Hessian matrix. Again, the regular use of this suggestion in our work is prevented by associated cost; it is not effective to compute the complete error information. Overall, we prefer fast, model-oriented schemes with moderate precision to slower, automatic schemes with high precision.

As a minimal summary of this chapter, we visualize the schemes for the use of model reduction in the iterative search in Figure 5.1. In the upper half of the picture, we show how a single version of the reduced space is used to arrive at an approximation for the minimizer: the scheme is either (5.36) or (5.37). In the lower half of the picture we show how, if needed, the search can be restarted based on a different version of the reduced model: see remarks around (5.56).





**Figure 5.1 Model reduction in an iterative search.**

## CHAPTER 6

### NUMERICAL TOOLS

The models used in chemical kinetics can be very complex. The behavior of the numerical simulation, and its sensitivity to parameters may depend on the choices made in modeling the speed and time-dependence of chemical reactions, the spatial discretization scheme, and the solver chosen for the ODE.

In most cases, the nature of our applied problems allows us to use standard numerical solutions, since moderate numerical error is allowed, and the issues with numerical stability cannot be resolved by the choice of the correct solver alone. A reader interested in the extensions of our research should be aware that the choice of the solver is relevant, in particular, for factor importance analysis presented in Chapter 2 and adjoint differentiation presented in Chapter 4. Our assumption that the full model trajectory  $u(t)$  is precise may have to be replaced with an understanding of model reduction process that takes into account the solver-induced error with its own sensitivities. A fundamental question of optimal choice of the numerical tools to integrate the reduced models is largely unanswered. In our practical applications, however, it is not a primary issue.

In this chapter we provide a brief overview of the choices made in the study of our applied problems. In particular, we explain how a set of chemical equations and processes is automatically rewritten into a system of differential equations; explain our choice of the numerical solvers for integrating the stiff ODEs of

atmospheric chemistry, and of the discretization scheme for modeling the convection-diffusion transport.

## 6.1 PREPROCESSING

The chemical systems appearing in our applied problems have a large number of reactions; even with most components being neutral with respect to each other, the full record of such system is inconvenient for direct manipulation and analysis. In standard practice, a list of chemical equations is turned into a mathematical model by preprocessing software. Under automated processing, each chemical reaction contributes to the production and the loss terms of the model. For example, the basic reactions



turn, correspondingly, into components of the ODE:

$$\begin{aligned}
 (1) \quad & \frac{d[A]}{dt} = \dots - k[A][B] \quad \frac{d[B]}{dt} = \dots - k[A][B] \quad \frac{d[C]}{dt} = \dots + k[A][B] \\
 (2) \quad & \frac{d[A]}{dt} = \dots - k[A] \quad \frac{d[B]}{dt} = \dots + k[A] \quad \frac{d[C]}{dt} = \dots + k[A] \\
 (3) \quad & \frac{d[A]}{dt} = \dots - k[A][B] \quad \frac{d[B]}{dt} = \dots - k[A][B] \\
 & \frac{d[C]}{dt} = \dots + k[A][B] \quad \frac{d[D]}{dt} = \dots + k[A][B] \\
 (4) \quad & \frac{d[A]}{dt} = \dots - k[A] \quad \frac{d[B]}{dt} = \dots + k[A]
 \end{aligned}
 \tag{6.2}$$

where  $[A]$  denotes the concentration of the reaction component  $A$ , and  $k$  is the rate of the reaction. Removal of reagents from the system is modeled by a reaction  $A \rightarrow M$ , where  $M$  is an inert medium that does not participate in reactions. The possibility of depletion of a particular specie is usually ignored. The system may be enhanced with additional details, such as photon consumption and emission, heat emission, and change in the reaction rates. The resulting model does not have to conserve mass or concentrations in general, though an inspection of the list of equations allows to identify which species are supposed to be mass-conservative, and use the information to later validate the numerical solution.

Our examples of chemical models were generated using the kinetic preprocessor package KPP [99], [131]. Its capabilities provide a good example of standard preprocessing procedures.

KPP records the main chemical reactions in the format of mass action kinetics law:

$$\frac{du}{dt} = \Sigma \cdot \begin{pmatrix} k_1(t) & & 0 \\ & \ddots & \\ 0 & & k_R(t) \end{pmatrix} \cdot \rho(u) = f(u,t) \quad (6.3)$$

where  $\Sigma$  is the stoichiometric matrix,  $\rho(u)$  is the vector of reactant products, and  $k_i(t)$  is a time-dependent rate of a particular reaction  $i = 1, 2, \dots, R$ . For a different perspective, the equation (6.1) may be rewritten into the stoichiometric format

$$\frac{du}{dt} = \Sigma \cdot \nu(u,t) \quad (6.4)$$

where  $v_i = (v_1, v_2, \dots, v_R)$  is a vector of reaction velocities. The expression  $\rho(u)$  is quadratic in  $u$ , allowing simple, explicit evaluation of the right-side Jacobian:

$$J(u, t) = \Sigma \cdot \begin{pmatrix} k_1(t) & & 0 \\ & \ddots & \\ 0 & & k_R(t) \end{pmatrix} \cdot \frac{d\rho}{du}(u) \quad (6.5)$$

with sparsity of approximately  $2R/n^2$ . The system of ODEs adjoint to (6.3) can also be constructed explicitly. The Hessian of the right side is also available, and dependent only on time, in principle allowing second-order adjoint differentiation (see Section 4.2).

Formulating the dependence of the equation rates on time requires additional modeling assumptions, since the speed of chemical reaction may depend on temperature and exposure to sunlight [80]. In general, for an adequate modeling of the chemical process, the system (6.1) has to be coupled with the estimates of heat produced by each reaction, and an Arrhenius-type relationship between temperature and reactivity. In the problems of atmospheric chemistry, chemical heat production and transport may be ignored; then the only factor influencing the rate of chemical reactions is the sunlight intensity.

The package KPP is also capable of integrating the direct and the adjoint versions of the model. In our numerical experiments, we use preprocessing software only to generate the ODEs; integration of the full and the reduced models is performed using external solvers, as described in the following section.

## 6.2 MODEL INTEGRATION

According to our observations, the main complexity of the reaction-transport system (1.4) lies in the chemical reaction term, and the choices made in time integration of the discretized model are comparatively more important for simulation and reduction than the choice of the spatial discretization scheme. The errors and artifacts in the reduced description of transport are approximately the same for every solver: to our knowledge, there is no particular approach that shows distinct advantage. We describe the available ODE solvers in some detail, and then briefly remark on the solution of the PDE.

### **Time integration**

Empirically, the flaws introduced by an ODE solver into a representation of the full model become more prominent in the reduced model performance. While the full-model can usually be stably resolved by any generic Runge-Kutta scheme with an adaptive time step, the corresponding reduced problem may experience numerical blowup. As noted in Chapter 2, the generation of instability due to distortion of the phase portrait under projection may be unavoidable.

Our task is then to select a standard, validated solver for the full problem that will produce the most reliable snapshots, and perform well on a generic stiff problem, so that other reasons for instability are minimized. In principle, multiple solvers can be used, including the case where the snapshots are generated by one scheme, and the reduced model equations are solved by another. Whether the

sensitivity information of the full model is then still applicable to the reduced model is an unanswered question.

We refer to Sandu et. al. [100], [101], for an extensive list of the available solvers for stiff ODEs of atmospheric chemistry. Since a moderate error is almost always allowed in our applied problems, we are mainly interested in numerical stability and qualitative concerns such as positivity and mass conservation.

Since the right side of the ODE (6.3) is quadratic in  $u$ , it can be rewritten into the production-loss form

$$\frac{du}{dt} = f(u, t) = P(u, t) - L(u, t)u \quad (6.6)$$

with nonnegative production and loss terms  $P(u, t)$ ,  $L(u, t)$ . It is noted in [116] that the only available elementary integration method without step size restriction that preserves both positivity and mass of the model state  $u(t)$  is the implicit Euler scheme

$$u^{(i+1)} = u^{(i)} + \Delta t f(u^{(i+1)}, t_{i+1}) = \left( I + \Delta t L(u^{(i+1)}, t_{i+1}) \right)^{-1} \cdot \left( u^i + \Delta t P(u^{(i+1)}, t_{i+1}) \right) > 0 \quad (6.7)$$

with the time step  $\Delta t$ , with the mass conservation being a general property of Runge-Kutta type solvers.

The numerical scheme (6.7) is only the most basic option, as it experiences difficulties with stiff problems, is not very fast, and is sensitive to the errors in the derivative information when accelerated using Newton iterations. Beyond it, the choice is between a family of dedicated methods (two examples provided below) that make use of the production-loss format and the slow-fast behavior of the

chemical model, and the general purpose Rosenbrock methods that are widely used because of high computational efficiency.

A basic QSSA (quasi-steady state approximation) scheme is based on the assumption that the production and loss terms  $P, L$  vary only slightly over time. It is based on a formula that is exact if the terms are constant:

$$u^{(i+1)} = \exp(-\Delta t L(u^{(i+1)}, t_{i+1})) u^{(i)} + \left( I - \exp(-\Delta t L(u^{(i+1)}, t_{i+1})) \right) \cdot \left( L(u^{(i+1)}, t_{i+1}) \right)^{-1} \cdot P(u^{(i+1)}, t_{i+1}) \quad (6.8)$$

with additional approximations for the slow and the fast species:

$$\begin{aligned} \exp(-\Delta t L_f(u^{(i+1)}, t_{i+1})) &\approx 1 - \Delta t L_f(u^{(i+1)}, t_{i+1}) & \Delta t L_f(u^{(i+1)}, t_{i+1}) &\approx 0 \\ \exp(-\Delta t L_s(u^{(i+1)}, t_{i+1})) &\approx 0 & \Delta t L_s(u^{(i+1)}, t_{i+1}) &\gg 0 \end{aligned} \quad (6.9)$$

This scheme preserves positivity, is computationally efficient for problems with many fast and slow species, and has good stability properties. As formulated here, it does not preserve mass.

TWOSTEP, another scheme using the production-loss form of the model is based on a mass-conserving two-step backward differentiation formula

$$\begin{aligned} u^{(i+1)} &= \frac{4}{3}u^{(i)} - \frac{1}{3}u^{(i-1)} + \frac{2}{3}\Delta t f(u^{(i+1)}, t_{i+1}) = \\ &= \left( I + \frac{2}{3}\Delta t L(u^{(i+1)}, t_{i+1}) \right)^{-1} \cdot \left( \frac{4}{3}u^{(i)} - \frac{1}{3}u^{(i-1)} + \frac{2}{3}\Delta t P(u^{(i+1)}, t_{i+1}) \right) \end{aligned} \quad (6.10)$$

The approximation to the implicitly defined  $u^{(i+1)}$  is computed by applying Gauss-Seidel technique to the function

$$G(u) = \left( I + \frac{2}{3}\Delta t L(u, t) \right)^{-1} \cdot \left( \frac{4}{3}u - \frac{1}{3}u + \frac{2}{3}\Delta t P(u, t) \right) \quad (6.11)$$

resulting in a component-wise formula



$$u_l^{(i+1)} = G(u_1^{(i+1)}, u_2^{(i+1)}, \dots, u_{l-1}^{(i+1)}, u_l^{(i)}, u_{l+1}^{(i)}, \dots, u_n^{(i)}) \quad (6.12)$$

The method efficiently processes fast behavior in the model (exact implicit solution is achieved for the components  $u_l$  for which the production and loss terms  $(P)_l, (L)_l$  are constant). The method does not preserve positivity. For efficiency of the Gauss-Seidel procedure, it may be required that the model variables are not strongly coupled. This requirement rules out some heterogeneous chemistry problems, but fits well with our understanding of a successfully reduced model, where some correlating components have been eliminated, or lumped together.

The Rosenbrock methods are based on rewriting the system of ODEs (1.1) into an autonomous form by formally treating time in the right-hand side as a dependent variable:

$$\begin{aligned} \frac{du}{dt} &= f(u) \\ \frac{dt}{dt} &= 1 \end{aligned} \quad (6.13)$$

The idea is then to generalize on the linearly implicit approach to solving time-independent systems, and on the Newton's methods with an  $s$ -stage integration formula using first-order derivative information:

$$\begin{aligned} u^{(i+1)} &= u^{(i)} + \sum_{l=1}^s b_l k_l \\ k_l &= \Delta t f(u^{(i)} + \sum_{j=1}^{l-1} \alpha_{lj} k_j) + \Delta t J(u^{(i)} + \sum_{j=1}^l \beta_{lj} k_j) \end{aligned} \quad (6.14)$$

with the coefficients  $b, \alpha, \beta$  chosen for consistency and stability of the stiff ODE solution. For the time-dependent system, the expression (6.14) is modified to

$$k_i = \Delta t f(u^{(i)}) + \sum_{j=1}^{l-1} \alpha_{ij} k_{j,t_i} + \Delta t \sum_{j=1}^{l-1} \alpha_{ij} + \sum_{j=1}^l \beta_{ij} (\Delta t)^2 \frac{\partial f(u^{(i)}, t_i)}{\partial t} + \Delta t J(u^{(i)}, t_i) \sum_{j=1}^l \beta_{ij} k_j \quad (6.15)$$

The scheme is one-step, partially explicit, and available in many implemented forms, (for example, included in the Matlab initial value problems package). The positivity is usually not preserved, with an exception of the variant called ROS2:

$$\begin{aligned} u^{(i+1)} &= u^{(i)} + \frac{1}{2} k_1 + \frac{1}{2} k_2 \\ k_1 &= \Delta t f(u^{(i)}, t_i) + \left(1 + \frac{1}{\sqrt{2}}\right) \Delta t J(u^{(i)}, t_i) k_1 \\ k_2 &= \Delta t f(u^{(i)} + \Delta t k_1, t_i) - 2 \left(1 + \frac{1}{\sqrt{2}}\right) \Delta t J(u^{(i)}, t_i) k_1 + \left(1 + \frac{1}{\sqrt{2}}\right) \Delta t J(u^{(i)}, t_i) k_2 \end{aligned} \quad (6.16)$$

that preserves positivity when provided with a precise value of the Jacobian  $J(u, t)$ .

In our numerical experiments, we became aware of the difference between the performance of the solvers. However, since the understanding of the behavior of the reduced model is in many ways still basic, it is not clear how to tune some of the more complex methods to an *a priori* unknown behavior of the reduced model. We note that the possibility of a simple, un-tuned implementation for any problem is an important argument in favor of using the scheme. In addition, the use of any integration methods more complex than first-order, one-step schemes introduces a change in the derivative information that was not accounted for in the differentiation procedure used in Chapters 2, 4. For schemes like (6.8), (6.10), (6.16), the

expression for  $\frac{du}{dp}$  at  $t = t_{i+1}$  should match the corresponding derivative for the right-side expression, which is not enforced at all in the continuous adjoint differentiation. The discrete adjoint formulation (see [79], [97], [120]) resolves the problem by differentiating, in the reverse direction, the integration steps related to each other by a chain rule relationship

$$\frac{du^{(i+1)}}{dp} = \frac{du^{(i+1)}}{du^{(i)}} \cdot \frac{du^{(i)}}{du^{(i-1)}} \cdots \frac{du^{(2)}}{du^{(1)}} \cdot \frac{du^{(1)}}{dp} \quad (6.17)$$

We find, however, that the discrete adjoint differentiation approach is too computationally expensive for our goals of fast factor importance analysis and iterative optimization. A temporary solution is then to use a numerical scheme that is as close as possible to an unmodified Euler procedure, and has been observed to produce acceptable derivative information (as validated by the same derivatives being computed by finite difference methods). Based on the empirical evidence, we settle on standard second-order Rosenbrock solvers.

■

### **Discretization in space and operator splitting**

In the context of model reduction, the transport effects are of relatively lower importance. As we have not observed changes in numerical stability of transport equations under reduction, we suggest that the reduced model should be discretized by the same scheme that was used on the full model equations.

In most numerical experiments, we use a central difference discretization

$$\begin{aligned}\frac{\partial u(t)}{\partial x} &\approx \frac{u^{(i+1)}(t) - u^{(i-1)}(t)}{2\Delta x} \\ \frac{\partial^2 u(t)}{\partial x^2} &\approx \frac{u^{(i+1)}(t) - 2u^{(i)}(t) + u^{(i-1)}(t)}{(\Delta x)^2}\end{aligned}\tag{6.18}$$

We also suggest a popular third-order scheme with upwinding [116], known to better model the advection effects:

$$\frac{\partial u(t)}{\partial x} \approx \frac{-u^{(i-2)}(t) + 6u^{(i-1)}(t) - 3u^{(i)}(t) - 2u^{(i+1)}(t)}{6\Delta x}\tag{6.19}$$

For large problems, where the scheme and the step size chosen to best represent the advection effects may lead to incorrect representation of diffusion, we use the standard operator splitting of the advection-diffusion-reaction PDE (1.4) into three problems solved sequentially, with a smaller time step, on each time interval  $t_i \leq t \leq t_{i+1}$ , while using the results of the previous integration as the initial value for the next:

$$\frac{\partial u_A}{\partial t} = f_A(u_A) = -\nabla \cdot (w u_A)\tag{6.20}$$

$$\begin{aligned}\frac{\partial u_D}{\partial t} &= f_D(u_D) = \nabla \cdot (K \nabla u_D) \\ u_D(t_i) &= u_A(t_{i+1})\end{aligned}\tag{6.21}$$

$$\begin{aligned}\frac{\partial u_R}{\partial t} &= f_R(u_R) = f(u_R, t), \quad u_R(t_i) = u_D(t_{i+1}) \\ u_A(t_{i+1}) &= u_R(t_{i+1}), \quad u(t) := u_R(t)\end{aligned}\tag{6.22}$$

This scheme is appropriate for PDEs with continuous solutions (which adds some restrictions on the chemical reaction term). The accuracy of the scheme is improved

if the integration intervals are shifted so that the splitting is symmetric around the middle of each time interval:

$$\begin{aligned}
t_{i+1/2} &= \frac{t_i + t_{i+1}}{2}; \\
u_A(t_i) &= u(t_i): \quad \frac{\partial u_A}{\partial t} = f_A \quad t_i \leq t \leq t_{i+1/2}; \\
\frac{\partial u_D}{\partial t} &= f_D \quad t_i \leq t \leq t_{i+1/2}; \quad \frac{\partial u_R}{\partial t} = f_R \quad t_i \leq t \leq t_{i+1}; \\
\frac{\partial u_D}{\partial t} &= f_D \quad t_{i+1/2} \leq t \leq t_{i+1}; \quad \frac{\partial u_A}{\partial t} = f_A \quad t_{i+1/2} \leq t \leq t_{i+1}; \\
u(t_{i+1}) &= u_A(t_{i+1})
\end{aligned} \tag{6.23}$$

The internal time steps used in the scheme can be fairly large for the advection and diffusion equations (6.20), (6.21), and adaptively adjusted by an ODE solver for the reaction term (6.22).

In the cases where the adjoint model is built based on undiscretized PDE, we suggest splitting the adjoint operator (4.37) over exactly the same time intervals as the direct operator:

$$\begin{aligned}
\frac{\partial u_A^*}{\partial t} &= f_A^*(u_A^*) = \nabla \cdot (w u_A^*) \quad t_i \leq t \leq t_{i+1/2} \\
\frac{\partial u_D^*}{\partial t} &= f_D^*(u_D^*) = -\nabla \cdot (K \nabla u_D^*) \quad t_i \leq t \leq t_{i+1/2} \\
\frac{\partial u_R^*}{\partial t} &= f_R^*(u_R^*) = -\left(\frac{df}{du}\right)^T u_R^* - \frac{\partial g}{\partial u} \quad t_i \leq t \leq t_{i+1} \\
\frac{\partial u_D^*}{\partial t} &= f_D^* \quad t_{i+1/2} \leq t \leq t_{i+1}; \quad \frac{\partial u_A^*}{\partial t} = f_A^* \quad t_{i+1/2} \leq t \leq t_{i+1}
\end{aligned} \tag{6.24}$$

Note that in (6.24) the interaction of the system with the output function has been arbitrarily lumped together with the chemical interaction effects.

■

## CHAPTER 7

### EXAMPLES

In this chapter, we provide applied examples of how theoretical suggestions introduced in the thesis may be implemented; we are particularly interested in two central topics of simulation and optimization, presented in Chapters 2, 5. Most of the problems we solve are based on standard tests of numerical methods performance from the corresponding fields; Charney-DeVore and Lorenz models described in Sections 7.5, 7.6 present some research interest; our larger example SAPRC-99 discussed in Section 7.7 has industrial significance.

Due to varying features and complexity of our models of reaction and transport, direct comparison of the reduced model performance from problem to problem is not always meaningful. Comparing the reduced model with the full model applied to the same problem can be more instructive, but the specific compared characteristics depend on the context; we do not have an always relevant definition of the “quality of the reduced model”.

To introduce a measure of consistency, we establish informal quality thresholds for a successful numerical experiment. We shall try to achieve dimension reduction to at most 20% of the original problem dimension. The relative error in reproduction of a feature of interest is expected to be 10% or lower. The performance measurements are not meant to be systematic. Following the style of most of our reference material (for example, [38], [2], [6], [33], [45], [117]), we state that the main measure of the reduced model performance is a *qualitative*

reproduction of the correct solution. Because of this, the presentation of our results is mostly graphic, with brief comments on significant performance metrics (as full tables of error measurement are hard to analyse due to large dimension of models). For the examples where we discuss factor importance analysis, the experiment is considered successful if we are able to obtain factor importance information that is not available by inspection of the full model.

Our visualizations are usually in the form of superimposed graphs of the full and the reduced model solutions  $u$ ,  $\hat{u}$  (evolving in time, or observed at a particular time instance). Where appropriate we also plot the relative errors:

$$\theta(t) = \frac{\|u(t) - \hat{u}(t)\|_2}{\|\hat{u}(t)\|_2} \quad (7.1)$$

$$e_i(t) = \frac{u_i(t) - \hat{u}_i(t)}{|u_i(t)|} \quad (7.2)$$

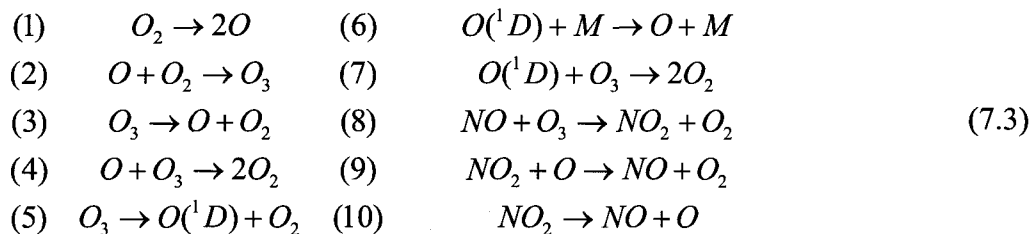
The computational expense of each problem is measured in seconds of Matlab runtime on an (average-performance) personal computer. The computational budget for the reduced model includes all the steps required to construct it, except maybe the generation of original snapshots. It also includes the computational cost of post-processing the answer using full model dynamics. The experiment is considered successful if, with the assistance of the reduced model, we manage to solve the problem faster (it is usually hard to predict by how much). We will routinely argue that the computational advantage grows with the increase in the dimension of the full model.

For larger examples, the interaction between model state components are fairly complex, with a tendency for the reduced ODE to become unstable over long integration intervals. In fact, numerical blowup due to instability is the most common scenario for failure of our methods. Since this type of flaw does not have a moderate form, we don't visualize it.

### 7.1 STRATOSPHERIC CHEMISTRY MECHANISM

In this section, we provide a very small example of a chemical kinetics model possessing such features of bigger models as stiffness, time-dependence of reaction rates, semi-periodic behavior of the solution. Due to highly correlated chemical processes, this model can be easily reduced.

We consider the following simple chemistry mechanism, classified as a 'Chapman-like model' [99] (typically used to predict the concentration of ozone in the stratosphere). It consists of 10 chemical reactions involving 7 species:



where the symbol  $M$  stands for the dense medium ('collision chaperoné') required for the chemical reactions. The reactions (1), (3), (5), (10) require an exposure to light, sometimes indicated by a symbol "+  $h\nu$ ".

The time evolution of the concentrations is described by a system of ODEs:



$$\begin{aligned}
\frac{d[O(^1D)]}{dt} &= k_5[O_3] - k_6[O(^1D)] \cdot [M] - k_7[O(^1D)] \cdot [O_3] \\
\frac{d[O]}{dt} &= 2k_1[O_2] - k_2[O] \cdot [O_2] + k_3[O_3] - k_4[O] \cdot [O_3] + \dots \\
&\dots + k_6[O(^1D)] \cdot [M] - k_9[O] \cdot [NO_2] + k_{10}[NO_2] \\
\frac{d[O_3]}{dt} &= k_2[O] \cdot [O_2] - k_3[O_3] - k_4[O] \cdot [O_3] - k_5[O_3] + \dots \\
&\dots - k_7[O(^1D)] \cdot [O_3] - k_8[O_3] \cdot [NO] \\
\frac{d[NO]}{dt} &= -k_8[O_3] \cdot [NO] + k_9[O] \cdot [NO_2] + k_{10}[NO_2] \\
\frac{d[NO_2]}{dt} &= k_8[O_3] \cdot [NO] - k_9[O] \cdot [NO_2] - k_{10}[NO_2]
\end{aligned} \tag{7.4}$$

where [...] denotes the concentration of the corresponding chemical specie. Typically, the concentrations in such models are measured in dimensionless units, such as “parts per billion”. In this model, the concentrations  $[O_2]$ ,  $[M]$  are kept fixed.

The rates of chemical reactions  $k_i$  are available from experimental data:

$$\begin{aligned}
k_1 &= 2.6 \cdot 10^{-10} \cdot SUN^3 & k_2 &= 8.0 \cdot 10^{-17} & k_3 &= 6.1 \cdot 10^{-4} \cdot SUN \\
k_4 &= 1.5 \cdot 10^{-15} & k_5 &= 1.0 \cdot 10^{-3} \cdot SUN^2 & k_6 &= 7.1 \cdot 10^{-11} \\
k_7 &= 1.2 \cdot 10^{-10} & k_8 &= 6.0 \cdot 10^{-15} & k_9 &= 1.0 \cdot 10^{-11} & k_{10} &= 1.2 \cdot 10^{-2} \cdot SUN
\end{aligned} \tag{7.5}$$

The time-dependent coefficient  $SUN$  is the normalized sunlight intensity, estimated by the expression

$$SUN = \frac{1 + \cos(\pi t')}{2}, \quad t' = \frac{1}{12} \left( 2 \frac{t}{3600} - 24 \right) \tag{7.6}$$

corresponding to a day with 12 hours of sunlight, the units of measurement are seconds.

To extract the correlations between model state components, we used 10–100 snapshots, distributed uniformly in time over a period of 1 day. The solution of the eigenvalue problem (2.20) was almost invariant to the choice of snapshots, resulting in the eigenvalues:

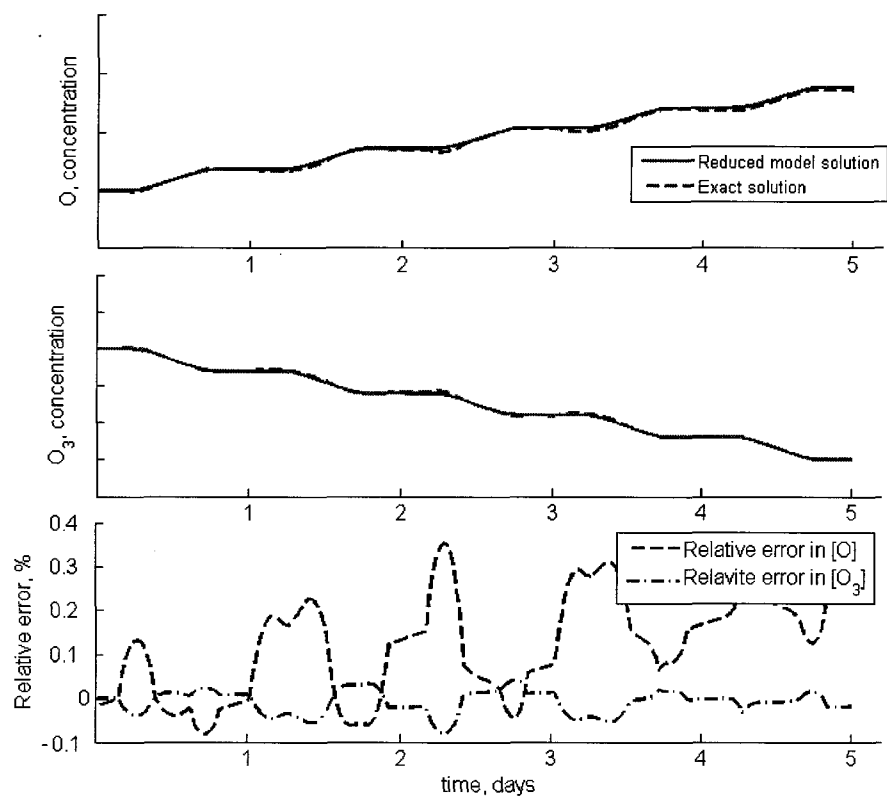
$$\lambda_1 \approx 2.67 \cdot 10^{-4}, \lambda_2 \approx 1.03 \cdot 10^{-10}, \lambda_3 \approx 4.09 \cdot 10^{-13}, \lambda_4, \lambda_5 \approx 0.$$

The standard POD-based reduction procedure was used to construct a model of dimension 2 by a change of coordinates and truncation of the 3 insignificant dimensions. In effect, the species  $[O(^1D)], [O], [O_3], [NO], [NO_2]$  are represented in new coordinates by the variables  $c_1, c_2, c_3, c_4, c_5$  with

$$\begin{aligned} c_1 &\approx -0.4[O(^1D)] - 0.59[O] + 0.07[O_3] - 0.56[NO] + 0.56[NO_2] \\ c_2 &\approx 0.87[O(^1D)] - 0.26[O] + 0.36[O_3] + 0.12[NO] + 0.12[NO_2] \\ c_3, c_4, c_5 &\approx 0 \end{aligned} \tag{7.7}$$

In this small example, the performance of the POD-reduced model is very good, even over a period of time in which no snapshots were taken. We have traced the evolution of concentrations of atomic oxygen  $[O]$  and ozone  $[O_3]$  over a period of 5 days, in the full and the reduced model representations. The reduced model reproduces the behavior of the full model with a maximal relative error of under 0.5% ; see Figure 7.1 for a visualization of results for a typical experiment.

We note that for this model the computational cost of solving the reduced model equations is not significantly lower than the corresponding cost for the full model; the comparison is inconclusive due to small scale, and an uncharacteristically dense full model Jacobian.



**Figure 7.1** Chapman-like mechanism, performance of the reduced model.

## 7.2 TEST OPTIMIZATION PROBLEM

In this section we provide a simple example of model-constrained optimization that uses model reduction. The problem has only the most basic components, and is intended for a reader not specifically interested in the models of atmospheric chemistry.

We define a multi-variable quadratic function  $\rho: R^n \rightarrow R$  by

$$\rho(x) = \sum_{i=1}^n (x_i - 1)(x_i - m_i) \quad (7.8)$$

where  $m_1, m_2, \dots$  are large constants. We choose a stiff low-rank matrix  $A \in R^{n \times n}$ ,

and use a corresponding linear test ODE as a constraint:

$$\begin{aligned} \frac{du}{dt} &= Au \\ u(0) &= p \\ x &= u(1) \end{aligned} \quad (7.9)$$

The task is to solve an initial conditions optimization problem:

$$\min_{p \in R^n} \rho(u) \quad (7.10)$$

The global minimum of (7.8) is

$$(x_{\min})_i = \frac{1}{2}(m_i + 1) \quad (7.11)$$

leading to an analytic solution of (7.10):

$$p_{\min} = \exp(-A) \cdot x_{\min} \quad (7.12)$$

where  $\exp(-A)$  denotes matrix exponentiation [42]. We set the initial guess to

$$p^{(0)} = (0, 0, \dots, 0)^T \quad (7.13)$$

This optimization problem can be solved by a gradient descent method, with a slightly better quality than the analytic solution (7.12), without numerical error in matrix exponentiation. We apply a standard steepest descent search. The gradient  $\nabla_p \rho$  is found by using a system of ODEs adjoint to (7.9):

$$\begin{aligned} \frac{du^*}{dt} &= -A^T u^* \\ u^*(1) &= \nabla_{u(T)} \rho \end{aligned} \quad (7.14)$$

with

$$\nabla_u \rho = (2u_1(T) - (m_1 + 1), 2u_2(T) - (m_2 + 1), \dots, 2u_2(T) - (m_2 + 1))^T \quad (7.15)$$

resulting in

$$\nabla_p \rho = u^*(0) \quad (7.16)$$

The length of the descent step  $a^{(k)}$  is defined by

$$\frac{d\rho(p - a^{(k)} \nabla_p \rho)}{da^{(k)}} = 0; \quad p = p^{(k)} \quad (7.17)$$

and is found by direct search, requiring a few integrations of (7.9) with initial conditions

$u(0) = p - a^{(k)} \nabla_p \rho$ ; see comments after (5.16). We stop the iterative search at a step such that the corresponding model state  $u(T)$  is within 0.1 of the value defined by (7.11).

To test the performance of reduction-based searches, we construct a reduced model of dimension  $k$  following the standard POD-based procedure. The state of the reduced model is described by the system

$$\begin{aligned}\frac{d\hat{u}}{dt} &= \Phi\Phi^T A\hat{u} \\ \hat{u}(0) &= \Phi\Phi^T (p - \mu) + \mu\end{aligned}\tag{7.18}$$

To find the coordinates  $q(t)$  in the representation  $\hat{u}(t) = \sum_i \hat{q}_i(t)\phi_i + \mu$  we solve the reduced ODEs

$$\begin{aligned}\frac{dq}{dt} &= \Phi^T A(\Phi q + \mu) \\ q(0) &= \Phi^T (p + \mu)\end{aligned}\tag{7.19}$$

The reduced adjoint system is written as

$$\begin{aligned}\frac{d\hat{u}^*}{dt} &= -A^T \Phi^T \Phi \hat{u}^* \\ \hat{u}^*(T) &= \Phi\Phi^T \nabla_{\hat{u}(T)} \rho\end{aligned}\tag{7.20}$$

To find the coordinates  $q^*(t)$  in the representation  $\hat{u}^*(t) = \sum_i \hat{q}_i^*(t)\phi_i$  we solve the system

$$\begin{aligned}\frac{dq^*}{dt} &= -\Phi^T A^T \Phi q^* \\ q^*(T) &= \Phi^T \nabla_{\hat{u}(T)} \rho\end{aligned}\tag{7.21}$$

The gradient of the output function is estimated as

$$\nabla_p \rho(\hat{u}) = \hat{u}^*(0)\tag{7.22}$$

The step length  $a$ , again, is found by direct search, requiring a few evaluations of (7.19) with the initial conditions  $q(0) = \Phi^T (p - a\nabla_p \rho(\hat{u}))$ .

The knowledge of the precise model state mean  $\mu$  is empirically not required for this problem, possibly due to low influence on the computation of the

adjoint derivative. The correct choice of  $\Phi$  (or, rather, of the set of snapshots on which the projection is based) is important. Following suggestions of Chapter 5, we perform several steps of the full model search and arrive at a value  $p = p^{(k_1)}$  for some small  $k_1$ . We then construct a set of snapshots by integrating (7.9) with initial conditions  $u(0) = p^{(k_1)}$ . In this setup, the obtained reduced model is consistently effective for the purposes of iterative optimization. In contrast, using the reduced model based on a set of snapshots generated from an *a priori* guess at the correct initial conditions often results in a search that converges very slowly, or to the incorrect answer.

In our numerical experiments, the matrix  $A$  and the constants  $m_i$  were generated randomly. We used the values  $-1000 < m_i < 1000$ ;  $\|A\|_2 \leq 1$ . The order of the reduced model was set at  $k \approx 0.15n$ . During the search, we switched to the reduced model at  $k_1 \leq 10$ . As a metric of performance, we measured the distance of the current value of  $x = u(T)$  from the minimizing value defined by (7.11):

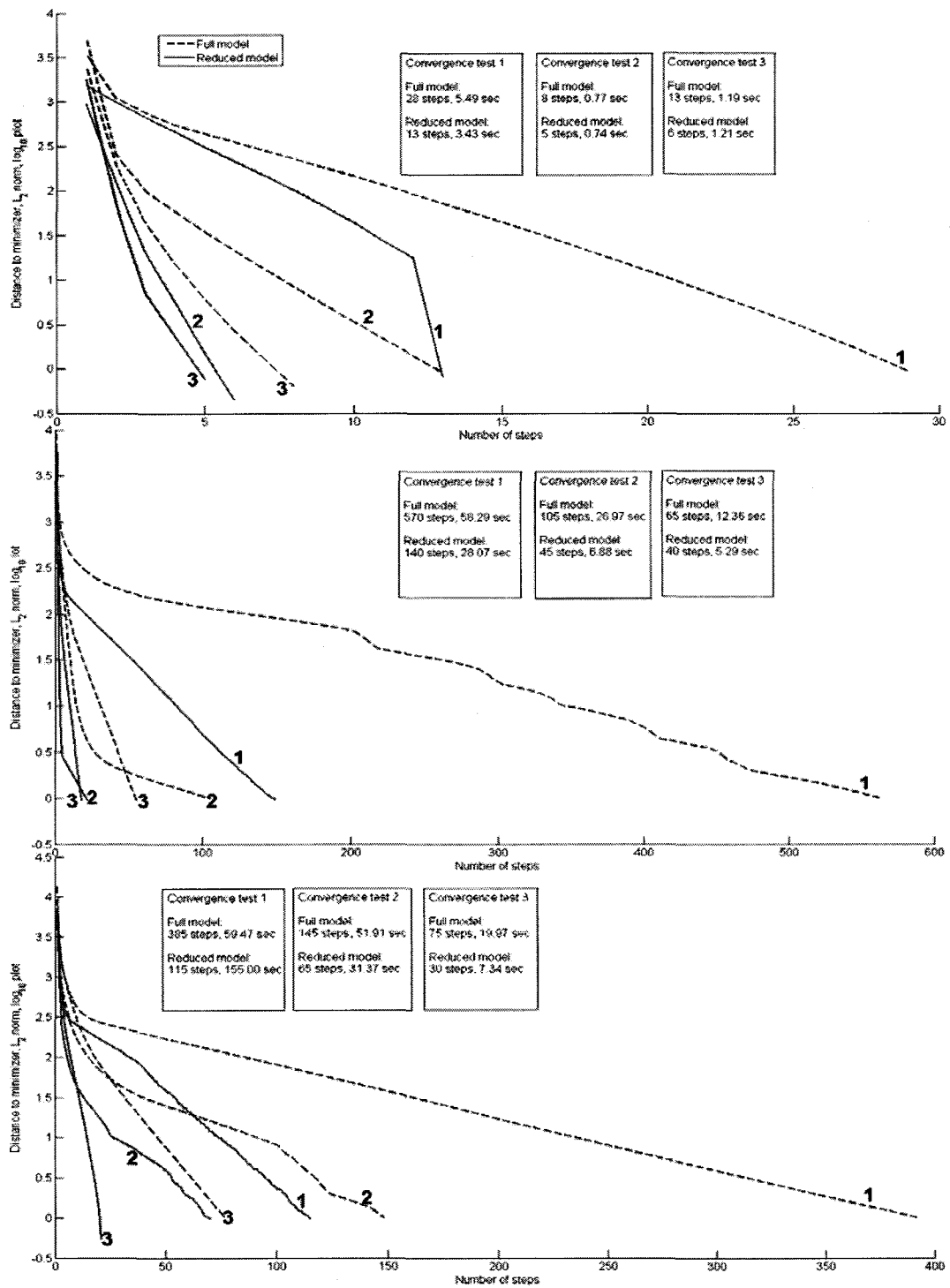
$$\left\| u(T; p^{(k)}) - \frac{1}{2}(m+1) \right\|_2 \quad (7.23)$$

The comparative performance of the full and the reduction-based searches is visualized in Figure 7.2. The three subplots show a total of 9 tests demonstrating the typical behavior of the search (in different randomized setups) for  $n = 10, 20, 30$  (from top to bottom). The number of steps required for convergence is shown on

one axis, the metric (7.23), in logarithmic scale, on the other. We also recorded the computational expense of each search.

The reduced model search consistently requires fewer steps, and integration of the ODE systems at each step is faster. Even with the additional computational expense associated to constructing the reduced model, the relative computational cost of the reduction-based search consistently decreases as the dimension of the problem grows. In the shown set of tests, the improvement in efficiency is from 80–100% to 30% of the full model search time. Since the considered problem is very basic, the obtained results should be viewed as a standard of an improvement in efficiency. As we will see in the following sections, comparable improvement in efficiency can be achieved for more sophisticated problems.





**Figure 7.2. Convergence of the full and the reduced optimization searches, problems of dimension 10, 20, 30 with random parameters.**

### 7.3 BRUSSELATOR

In this section, we apply model reduction to a commonly used test PDE model with ‘Brusselator’ dynamics [66], [134]. The goal is to observe how well the dynamics of the full model can be reproduced after very significant dimension reduction.

The model used here is a particular case of (1.4) in one spatial dimension. It includes diffusion and reaction of two chemical species:

$$\begin{aligned} \frac{\partial u}{\partial t} &= K \frac{\partial^2 u}{\partial x^2} + f(u) \quad t > 0, x \in (0,1) \\ f(u) &= \begin{pmatrix} a + u_1^2 u_2 - (b+1)u_1 \\ bu_1 - u_1^2 u_2 \end{pmatrix} \\ u(x,t) &= c \quad x = 0,1 \\ u(x,0) &= p \end{aligned} \tag{7.24}$$

discretized to a system of ODEs using central differences:

$$\begin{aligned} \frac{du_i}{dt} &= K \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} + a + u_i^2 u_{i+n/2} - (b+1)u_i \quad 1 < i < n/2 \\ \frac{du_i}{dt} &= K \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} + bu_{i-n/2} - u_{i-n/2}^2 u_i \quad n/2+1 < i < n \\ u_i(t) &:= u_1((i-1)\Delta x, t) \quad i \leq n/2 \\ u_i(t) &:= u_2((i-1)\Delta x, t) \quad n < i \leq n \end{aligned} \tag{7.25}$$

We use the reaction constants  $a=1, b=3, c=1$ , the diffusion coefficient  $K=0.001$ , and set the initial conditions to

$$p_i = 1 + \sin\left(\frac{2\pi i}{n/2+1}\right) \tag{7.26}$$

Arbitrarily, we set the full model dimension to  $n = 500$  (any sufficiently large value can be used in the experiment; for  $n < 50$  the computational benefits are questionable, since the integration of the reduced model is not significantly faster).

We generate a set of 25 snapshots, uniformly distributed on time interval  $0 \leq t \leq 10$ , and apply reduction. We use the standard eigenvalue energy criteria (2.33) to select the dimension of the reduced model. The rapid drop in the eigenvalue magnitudes is shown in Figure 7.3 (model dimensions  $n = 100, 500, 1000$  were used). According to the plot, only the first 20 eigenvectors are significant. We set the reduced model dimension to  $k = 15$ ; in practice, even lower dimension can be used. We show the distribution of relative error (7.1) in time for reduced models for different values of  $k$  in Figure 7.4. By inspection of the plot, the values close to  $k = 10$  result in very similar performance; this appears to be a good estimate for the true number of degrees of freedom of the model.

We note that a measurement of the overall error is a very general characteristic. For an effective judgement of the reduced model performance, we need to look at how well it preserves the qualitative behavior of the full model, over the whole solution, or for particular features of interest.

In Figure 7.4, we compare the full and the reduced representations of the time evolution of two species, the plots for each correspond to 5 fixed spatial locations  $u(x_i, t)$ ,  $x_i = 0.2, 0.4, \dots, 1$ . We observe that the quality of the reduced model decreases as diffusion effects propagate in time and space. Geometrically, we see a situation that is very typical in model reduction: the solution shape is almost

correct, but the timing of intervals of increase and decrease gets progressively worse. The most significant loss of quality in all components occurs on the time interval  $7.5 \leq t \leq 10$ ; the distribution of the component-wise relative error (7.2) over time shows a mean of 3% with a standard deviation of 9% (the distribution is constructed based on maximal component-wise errors for each time instance).

By inspection of the error magnitudes, we shall identify the behavior on the interval  $7.5 \leq t \leq 8.5$  as the feature of interest. Since the feature is clearly identified in time, a goal-oriented snapshot placement may improve its reproduction in the reduced model dynamics. Arbitrarily, we redistribute 25 snapshots so that 10 of them fall into the identified interval, the rest are placed uniformly. The maximal error distribution then improves to the mean value of 2%, with a standard deviation of 7%. Furthermore, the acceptable quality of the reduced model (mean error less than 10%) is also preserved over a longer time interval,  $0 \leq t \leq 30$ . We show the improved performance of the reduced model (for the second chemical specie, at 5 spatial locations) in Figure 7.6.

We will now test the effectiveness of using the reduced model in iterative optimization. We seek to recover the correct initial conditions (7.26) based on the observations of the model state at time  $T = 10$  by solving an optimization problem

$$\begin{aligned} \min_p \mathfrak{J}: \\ \mathfrak{J}(u) = \|u(10, x) - u_o(10, x)\|_2^2 \end{aligned} \tag{7.27}$$

We construct an iterative solution using gradient descent method, with the gradient found by adjoint differentiation of the PDE, as described in Section 4.2. The PDE (4.41) is discretized to the form

$$\begin{aligned} \frac{du_i^*}{dt} &= -K \frac{u_{i+1}^* - 2u_i^* + u_{i-1}^*}{(\Delta x)^2} - (2u_i u_{i+n/2} - (b+1)u_i^* - (b-2u_i)u_{i+n/2}^*) \quad 1 < i < n/2 \\ \frac{du_i}{dt} &= -K \frac{u_{i+1}^* - 2u_i^* + u_{i-1}^*}{(\Delta x)^2} - (u_{i-n/2}^2)u_i^* + (u_i^2)u_{i-n/2}^* \quad n/2+1 < i < n \\ u_i^* &= 0 \quad i=1, n/2, n/2+1, n \\ u_i(10) &= 2(u_i(10) - u_{O_i}(10)) \quad 1 \leq i \leq n \end{aligned} \quad (7.28)$$

resulting in the search step  $d^{(k)} = -a^{(k)} \nabla_p \mathfrak{J} = -a^{(k)} u^*(0, x)$ , with the step size  $a^{(k)}$  found by direct search. We note that only very limited performance of both full and reduction-based searches should be expected here, since an inverse problem to diffusion equations is not convex, and, more generally, not well-posed.

For this experiment, we set  $n = 50$ , started with an initial guess

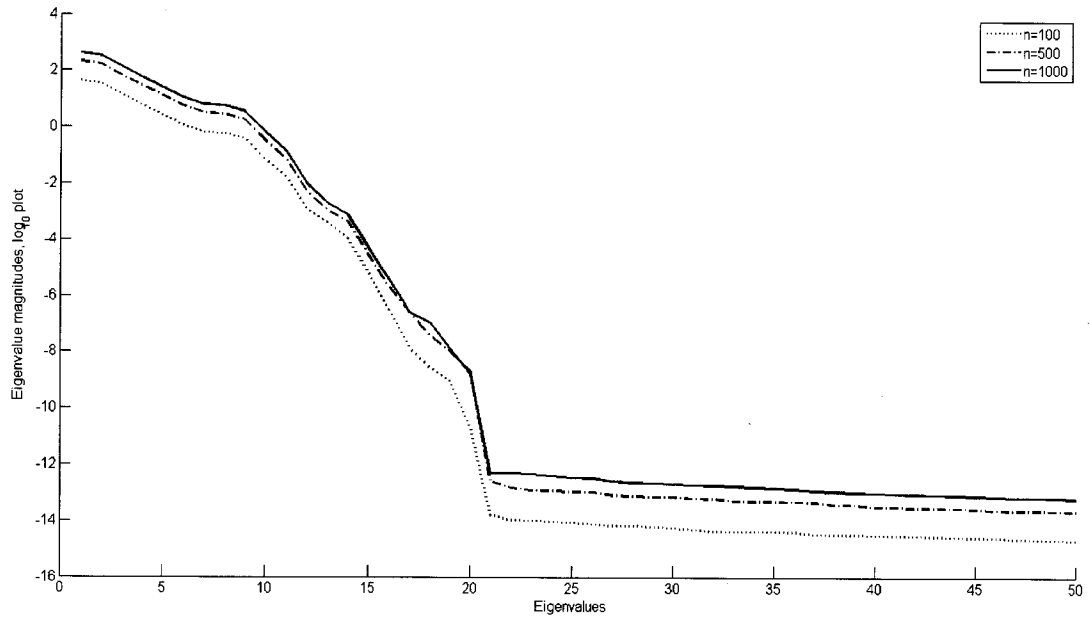
$$u_i(0) = p_i = 1 \quad (7.29)$$

and performed a search of 100 steps. We then used a reduced model of dimension  $k=15$ , obtained by an unmodified method of snapshots (for the purposes of optimization, the changes of snapshot placement, and weighting schemes did not result in a significant improvement of quality). As in the previous generic example (Section 7.2), each step of reduction-based search takes less computational time, and the search approaches the converged state in fewer steps.

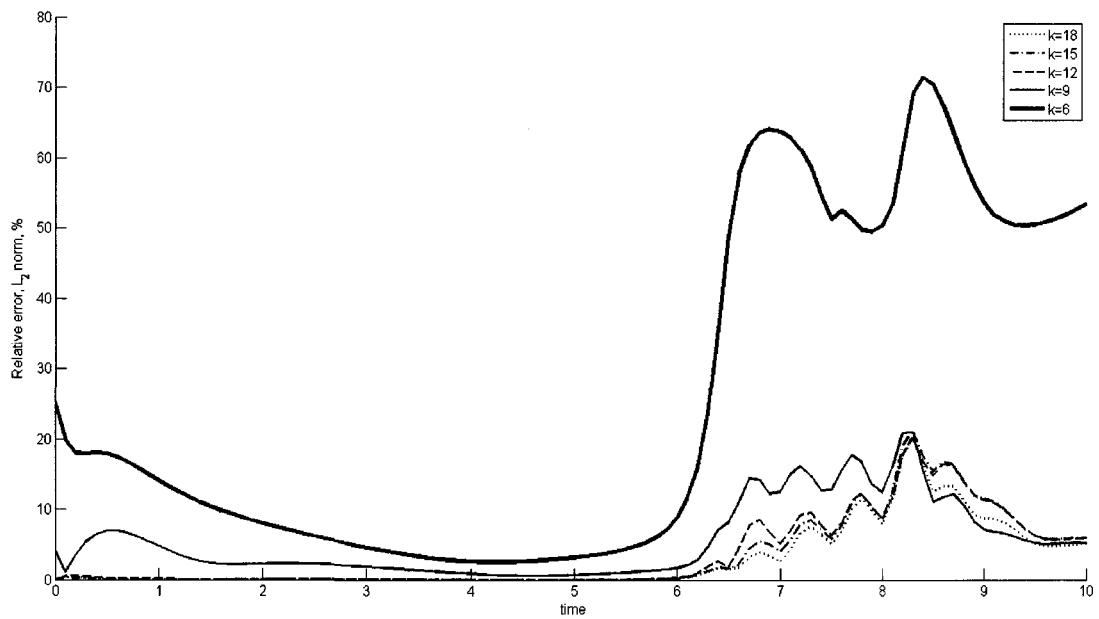
As we have expected, neither of the search results is very precise. In Figure 7.7, we show the obtained spatial distributions for the first chemical specie, i.e.

$u_1(0,x)$  of (7.24), and the corresponding  $\hat{u}_1(0,x)$ . We observe that the reduced model achieved a slightly better (though still inadequate) approximation to the correct initial conditions, and at approximately 50% lower computational cost. The numerical advantage will grow with the increase in the problem dimension  $n$  as both the direct and the adjoint ODEs can be integrated faster in the reduced form.

Besides the difficulties in solving the initial conditions optimization problems, the reaction-diffusion systems of the type considered in this section are among the easier subjects for reduction; the reduced model becomes difficult to tune for improved performance only if the reaction term is complex.

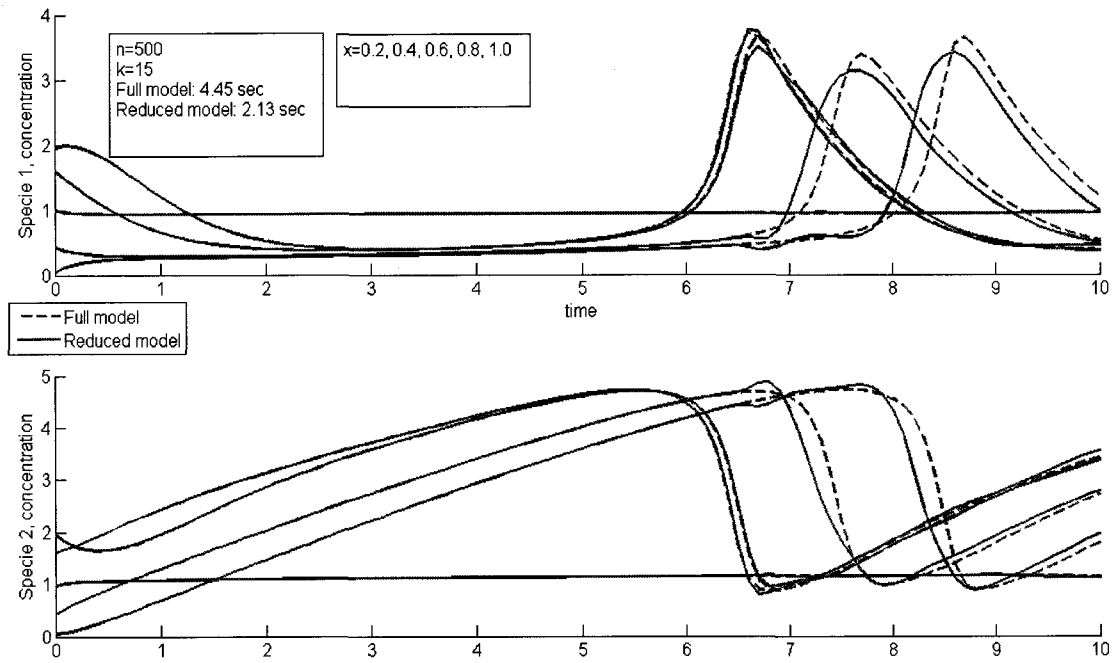


**Figure 7.3 First 50 eigenvalues of the “Brusselator” model covariance matrix.**

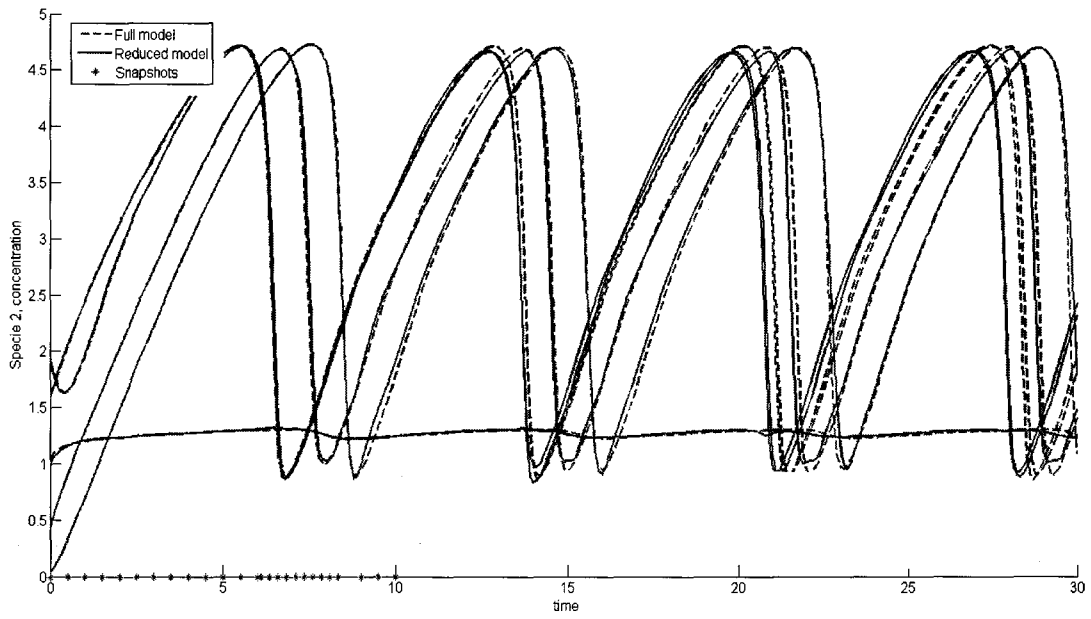


**Figure 7.4.** Relative error for the reduced version of the “Brusselator” model for different dimensions.

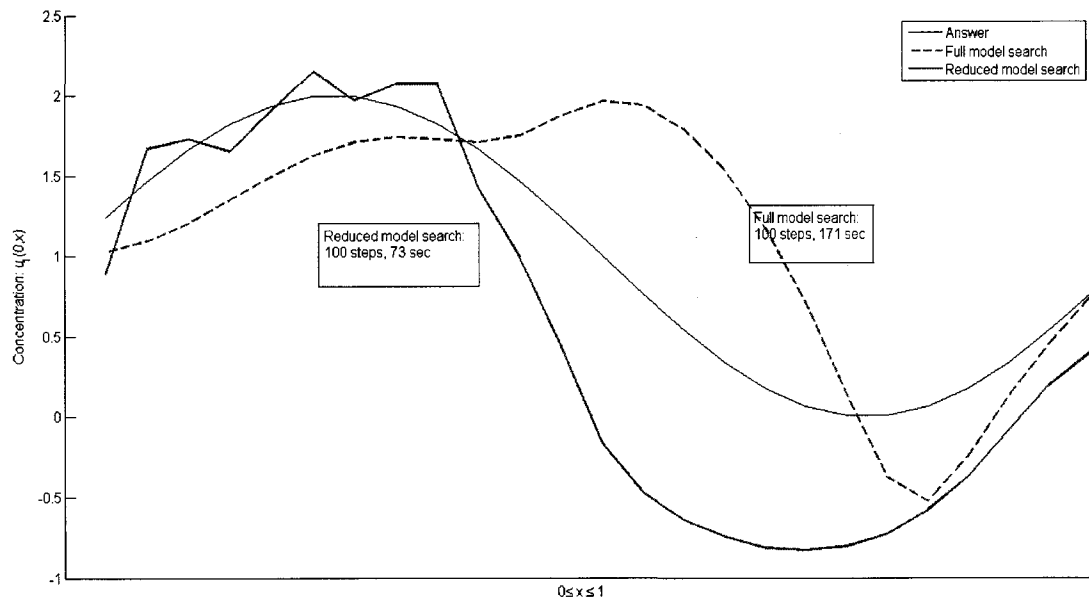




**Figure 7.5. Comparative performance of the full and reduced solutions for the “Brusselator” model.**



**Figure 7.6. Comparative performance of the full and reduced solutions for the “Brusselator” model; non-uniform placement of snapshots.**



**Figure 7.7. Comparative performance of the full and reduced solutions for the “Brusselator” model; iterative recovery of correct initial conditions.**

## 7.4 MOLENKAMP-CROWLEY PROBLEM

We shall now apply reduction to a Molenkamp-Crowley model [155]. The model is known to become computationally difficult over long periods of integration time, due to accumulating numerical error. We will tolerate the loss of quality in the discretized model, since the choice of a better integration scheme lies outside of the main interest of our study.

The model is described by a version of PDE (1.4) with advection transport effects and no reaction or diffusion. In 2 dimensions, the PDE is written as

$$\frac{du}{dt} + \nabla_x(wu) = \frac{du}{dt} + \frac{d(w_1u)}{dx_1} + \frac{d(w_2u)}{dx_2} = 0 \quad 0 \leq x_1, x_2 \leq 1 \quad (7.30)$$

We use the velocity field  $w = (w_1, w_2)$  that describes a unit speed counter-clockwise rotation of the initial distribution of concentrations around the point  $x_1 = x_2 = 0.5$ :

$$\begin{aligned} w_1(x_1, x_2) &= 2\pi(x_2 - 0.5) \\ w_2(x_1, x_2) &= -2\pi(x_1 - 0.5) \end{aligned} \quad (7.31)$$

Since the only structure in the model describes transport of the initial model state along the wind pattern, we expect that significant reduction is possible.

The initial conditions describe an exponential conic profile:

$$\begin{aligned} u(x_1, x_2, 0) &= M \exp\left(\frac{-1}{0.1 - (x_1 - m_1)^2 - (x_2 - m_2)^2}\right) \\ u(x_1, x_2, 0) &\approx 0 \quad (x_1 - 0.25)^2 + (x_2 - 0.25)^2 \geq 0.1 \end{aligned} \quad (7.32)$$

with the constants  $m_1, m_2$  defining the coordinates of the center of the cone, and the constant  $M$  adjusting the cone height. We use  $M = 2000$ ,  $m_1 = m_2 = 0.25$ .

The finite differences discretization on a square grid with  $\Delta x_1 = \Delta x_2 = 1/(n-1)$  results in  $n^2$  ODEs:

$$\begin{aligned} \frac{du_{i,j}}{dt} &= -\pi(j-(n-1)/2)(u_{i+1,j} - u_{i-1,j}) + \pi(i-(n-1)/2)(u_{i,j+1} - u_{i,j-1}) \quad 1 < i, j < n \\ \frac{du_{i,j}}{dt} &= 0 \quad i, j = 1, n \\ u_{i,j}(0) &= M \exp\left(-1/\left(0.1 - \left(\frac{i}{n-1} - 0.25\right)^2 - \left(\frac{j}{n-1} - 0.25\right)^2\right)\right) \end{aligned} \quad (7.33)$$

We use a set of 40 uniformly distributed snapshots on time interval  $0 \leq t \leq 1$  that corresponds to one full rotation of the initial profile. Since the exact solution of the model is periodic in time, we expect that this collection of snapshots is sufficient to capture the significant model dynamics over longer integration periods. The eigenvalues of the covariance matrix provide a tentative estimate of the degrees of freedom of the model. We show the distribution of eigenvalues for problems of dimension  $n^2 = 20^2, 30^2, 40^2, 50^2$  in Figure 7.8. We observe that only the first 18 eigenvalues have significant magnitudes.

For the dimension  $n^2 = 50^2$ , we performed reduction by an unmodified method of snapshots, with dimension  $k = 20$ , and integrated the full and the reduced equations over  $0 \leq t \leq 2$ , to simulate two full rotations of the initial profile. The final states  $u(2, x)$ ,  $\hat{u}(2, x)$  are shown in Figure 7.8.

Both the full and the reduced model solutions show significant deviation from the exact results of the initial profile rotation (precise position, obtained by geometric unit-speed rotation of the initial profile, is indicated by the grey circle).

The error introduced by the discretization (7.33) is evident over the whole domain (a ripple pattern is visible on the picture, errors of magnitude less than 1% of maximal cone height are not shown on the picture). If the goal of simulation is only to approximately locate the position of the cone, the reduced model produces a result of acceptable quality, and at a lower computational cost (65% improvement, taking into account reduced model construction time).

A higher quality of the reduced model solution can be achieved through the use of the exact model data in the snapshot set. Of course, a large set of such data may be unavailable in practice. For a small example of a realistic setup, we combined 20 snapshots from the solution of (7.30), and 20 snapshots from the exact model rotation. For the linear algebra operations required by the method of snapshots, it does not matter in which order the model states are taken. The final state  $\hat{u}(2, x)$  for the reduced model of dimension  $k = 20$  shown in Figure 7.9. We still observe the deformation of the profile, but the center of the cone is now placed better even in comparison with the full model solution. The ripples outside of the conic profile are significantly smaller (maximal error magnitude reduced from approximately 50% to 30% of the maximal cone height). Other simple tools for locating and improving the reproduction of features of interest, such as weighting and metric change, did not result in the significant improvement of the reduced model performance.

We shall now present an example of the use of the reduced model in optimization. For a test problem, we seek recover the correct initial conditions based

on a few instances of the model state  $u(t_i, x_1, x_2)$ , starting with a guess with no information about the correct profile shape (7.32):

$$p_{ij} = (u(0))_{ij} = 0.1 \quad (7.34)$$

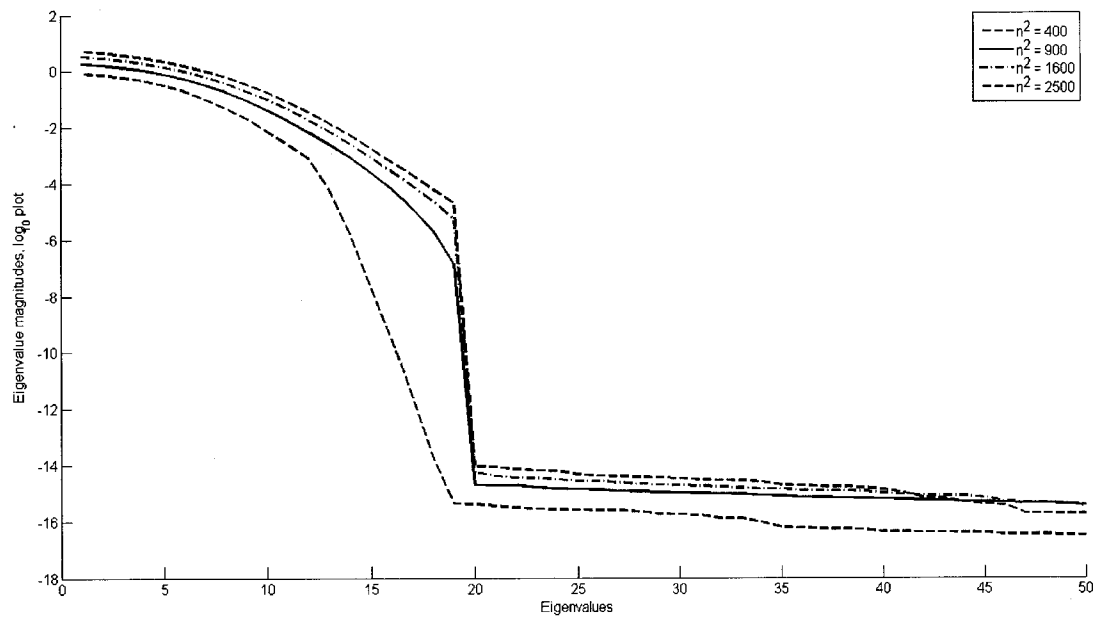
We used an output function that measures the quality of the reproduction of the exact state of the model on the interval  $t_i \leq t \leq t_{i_2}$  by comparing with the exact model state  $u_o$ :

$$\mathfrak{S} = \sum_{i=i_1}^{i_2} \int_{\Omega} \|u(t_i, x, y) - u_o(t_i, x, y)\|_2^2 dx dy \quad (7.35)$$

In our experiments,  $t_{i_1} = 1.8$ ,  $t_{i_2} = 2.0$ .

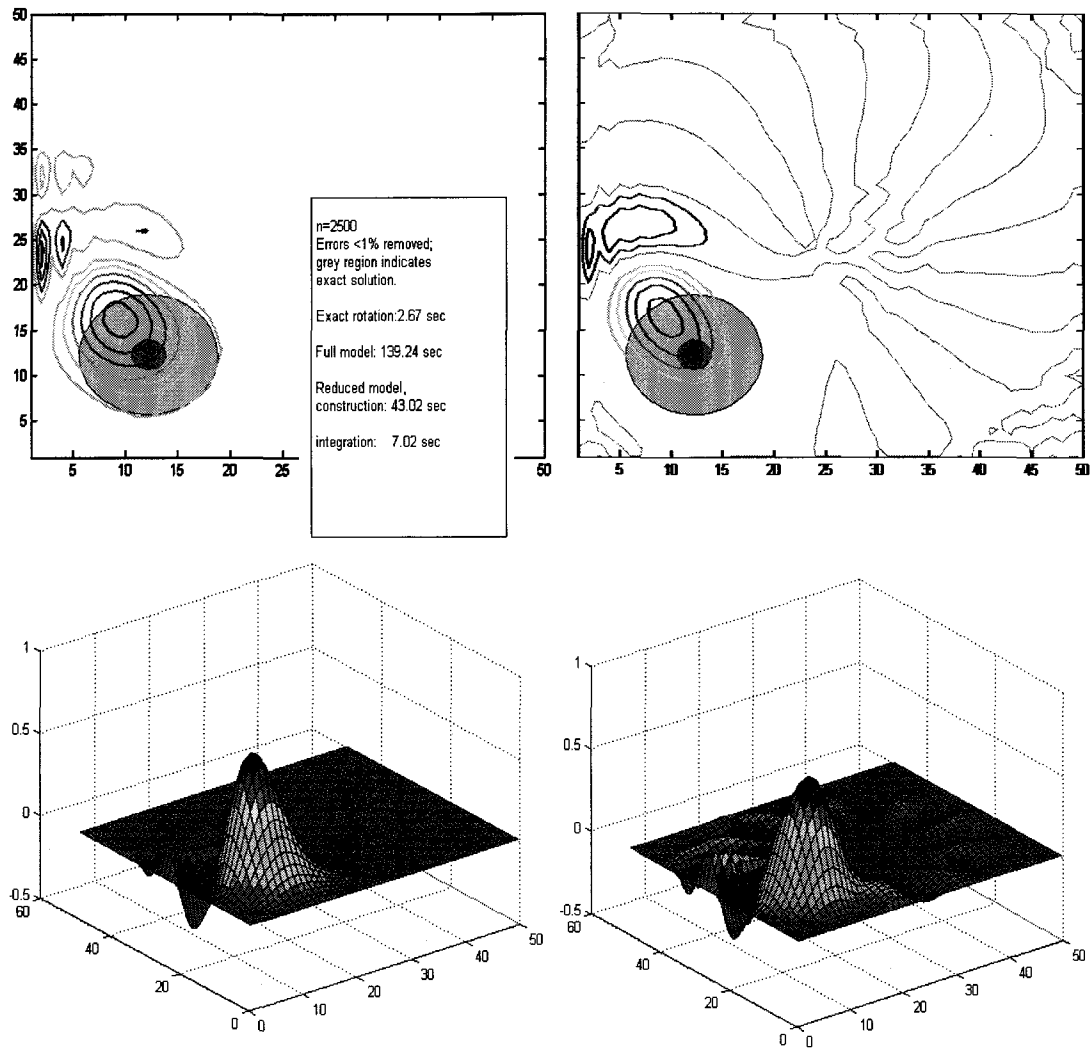
We then performed an iterative search using the full and the reduced versions of (7.33). The reduced model was constructed using a set of snapshots from one of the steps of the full model search. The derivative information required for the gradient descent method was found by adjoint differentiation of the discretized model. The comparison of performance after 10 search steps is shown in Figure 7.10. We observe that the reduced model locates the center with a slightly improved precision, at a significantly lower computational cost (75% improvement).

The considered problem is another example of relatively easy reduction of transport effects. Based on our experience, most of the development effort in reduction is associated with correct tuning of the reduction process to correctly represent the reaction term.

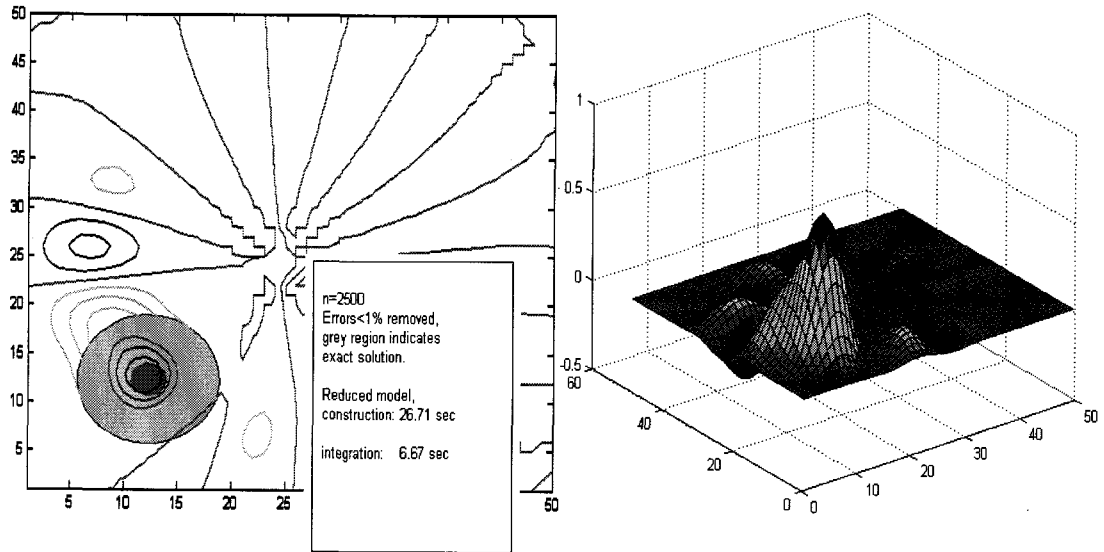


**Figure 7.7** First 50 eigenvalues of the Molenkamp-Crowley model covariance matrix.

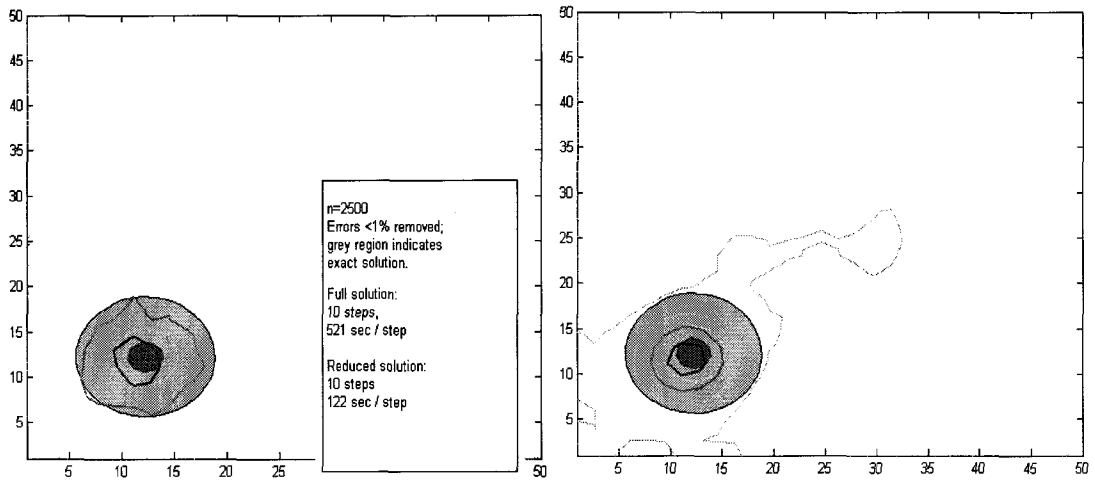




**Figure 7.8 Comparative performance of the full and reduced solutions for the Molenkamp-Crowley model.**



**Figure 7.9 Performance of a reduced solution for the Molenkamp-Crowley model: snapshots taken from the exact rotation of the profile**



**Figure 7.10 Comparative performance of the full and reduced solutions for the Molenkamp-Crowley model: iterative recovery of correct initial conditions, step 10.**

## 7.5 LORENZ MODEL

In this section we consider applying reduction to a model suggested by Lorenz et. al. [76]. It is based on a system of ODEs constructed in imitation of the more complex atmospheric behavior forecast models. We shall mainly use the model to illustrate our approach to sensitivity analysis of the reduction process.

The model equations are written as

$$\frac{du_i}{dt} = (u_{i+1} - u_{i-2})u_{i-1} - u_i + F, \quad 1 \leq i \leq n, \quad 0 < t \leq T \quad (7.36)$$

with an additional convention  $u_0 = u_n, u_{-1} = u_{n-1}, u_{n+1} = u_1$  required for interpreting the model state components  $u_i$  as values of some atmospheric physical quantity extending around a latitude circle. Unlike most of the other examples, the equations are not constructed based on a simplified version of conservation laws. Instead, generic algebraic terms in the right-hand side of the system are chosen so that the solution exhibits such typical characteristics of atmospheric models as semi-periodic advection and dissipation. By design, the importance of all state components is approximately equal, so making distinctions between the important and the negligible aspects of the model behavior is more difficult than in other examples.

We use the perturbed steady-state initial conditions

$$\begin{aligned} u_i(0) &= F, \quad i \neq n/2 \\ u_{n/2}(0) &= F + \Delta u_0 \end{aligned} \quad (7.37)$$

with  $n = 40$ ,  $\Delta u_0 = F/1000$ . The numerical stability and the effective complexity of the model (specifically, the propagation of consequences of the perturbation  $\Delta u_0$

over the latitude) depend on the value of the constant forcing term  $F$ . The author suggests the value  $F \approx 8/9$  as a threshold for the solution stability, and  $F = 4$  as a threshold of chaotic behavior. We use an intermediate value  $F = 2.5$ .

The simple structure of (7.36) suggests a possibility for some model reduction, for example, by lumping of the strongly correlated pairs  $u_i, u_{i+3}$  into one variable. Inspection of the covariance matrix (based on a uniformly distributed set of snapshots) also indicates the possibility of reduction; see Figure 7.11 for a visualization. While the magnitudes of the covariance matrix eigenvalues do not decrease as sharply as in some of the previous examples, the first 10–15 values consistently capture 99.99% of the eigenvalue energy. We set the reduced model dimension to  $k = 10$ .

The application of an unmodified reduction method with  $N = 20$  uniformly placed snapshots results in an unsatisfactory performance even on a small time interval,  $0 < t < 5$  (see Figure 7.12 for a distribution of values  $u_i(t), \hat{u}_i(t)$  for  $15 \leq i \leq 35$ , at time instances  $t = 0, 1, 2, 4, 5$ ). Using more snapshots results in a minor improvement in performance, but then on a slightly longer time interval  $0 < t < 10$  the reduced model deteriorates even further, to the point where even the state  $u(0)$  is not reproduced correctly. For such behavior, our current conclusion is that set of snapshots does not store the major features of the model, and, under reduction, amplifies negligible information.

The logical next step is to improve the performance of the reduced model using modified snapshot sets and weighting. Since the inspection of the model solution did not provide us with a useful guess of what is important, we shall now review some of the ways to obtain sensitivities and factor importance information.

Arbitrarily, we choose the reduced model state components  $\hat{u}_i : 25 < i \leq 35$  on the time interval  $4 \leq t \leq 5$  as a feature of interest. Direct amplification of corresponding snapshots and components (an event targeting, as described in Section 2.4) may be inapplicable here. As we will soon see from the factor importance analysis of the model, the chosen output is significantly entangled with the rest of the model evolution, and the parts of the snapshot information that fall into the feature of interest are not necessarily the most important for its reproduction in the reduced model solution.

To describe the relative importance of the model snapshots and snapshot components for the reduced model, we perform the calculations suggested in Section 2.2, systematically, with no additional assumptions (that would normally be made to avoid estimating importance of obviously negligible factors). We set

$$\hat{\mathfrak{S}} = \int_4^5 \hat{u}_i(t) dt \approx \Delta t \sum_{4 \leq t_j \leq 5} \hat{u}_i(t_j) \quad (7.38)$$

and compute the importance estimates

$$\hat{S}_j = \frac{d\hat{\mathfrak{S}}}{du_o(t_j)} \quad (7.39)$$

For every  $25 < i \leq 35$ ,  $t_1 \leq t_j \leq t_N$ , the expression (7.39) produces an output vector of length  $n = 40$ .

We visualize this description of first-order importance of the model state components in Figure 7.13. The plot shows the mean derivative magnitude, and the variability in the derivative values (sampled over time for state components, and over state components for the snapshots). This format of information presentation allows us to avoid making additional assumptions that a particular time instance, or a group of component states are more representative of the reduced model behavior.

Note that the numerical values obtained using (7.38) should be understood as sensitivities of the chosen output of interest with respect to numerical perturbations in the snapshot content. While this definition of importance of individual snapshots is not perfect, it is sufficient for our purposes.

Suppose that we decide to perform a computationally cheaper version of the same measurement, without model reduction. We can define

$$\mathfrak{S} = \int_4^5 u_i(t) dt \approx \Delta t \sum_{4 \leq t_i \leq 5} u_i(t_i) \quad (7.40)$$

and compute

$$S_j = \frac{d\mathfrak{S}}{du_o(t_j)} \quad (7.41)$$

with a restarted ODE differentiation, that is, obtaining  $u(t)$  by integration of (7.36) on the interval  $t_j < t \leq 5$  with initial conditions  $u(t_j) = u_o(t_j)$ . The results are visualized in Figure 7.14 (mean derivative magnitudes without variance are shown

on the bottom). We observe that a non-uniform structure of the reduced model sensitivity is not present in the full model measurements. Instead, the importance of the snapshot contents depends smoothly on the distance from the feature of interest (in time, and in model state space).

By comparing Figures 7.13 and 7.14 we also observe that the relative importance estimates (7.41) and (7.39) occasionally contradict each (for example, in components  $u_i: 25 \leq i \leq 40$ , snapshots  $u_o(t_i): i > 7, t_i > 1.4$ ). Our attempts at weighting based on the results of (7.40) have produced reduced models of very poor quality, even in comparison with un-weighted version. We conclude that for the current example the derivative information that does not take into account model reduction process is not efficient.

On the other hand, a version of the weighting based on the reduced model sensitivity has resulted in a small improvement in the reproduction of feature of interest. Our best observed performance is shown in Figure 7.15. For this version of the reduced model, we did not change the metric. We defined snapshot weights as the mean values of sensitivity (7.38):

$$w_j = \frac{1}{n} \sum_i \left| \frac{d\hat{\mathcal{S}}}{du_o(t_j)} \right|_i \quad (7.42)$$

In addition, as suggested in Chapter 2, we increased the density of snapshots over time intervals identified by inspection of factor importance as important. Specifically, the density of the snapshot set  $U_o$  was increased in the neighborhood of snapshots with numbers  $i = 1, 2, 3, 7, 8, 9, 14, 15, 16, 17$  that had a corresponding high



variation in snapshot importance, as shown in Figure 7.13. The corresponding time intervals are  $0 \leq t \leq 0.75$ ,  $1.75 \leq t \leq 2$ ,  $3.5 \leq t \leq 4.25$ .

We did not obtain new weights for this new collection of 30 non-uniformly distributed snapshots. Instead, the values obtained by (7.41) were treated as a set of estimates for the importance of time intervals: if several new observed system states fell into the interval  $t_{j-1} < t \leq t_j$ , they were assigned the same importance as the original snapshot  $u_o(t_j)$ .

For a quick comparison of performance of different reduced models, we can evaluate the difference  $|\mathfrak{S} - \hat{\mathfrak{S}}|$ . For our best model,  $|\mathfrak{S} - \hat{\mathfrak{S}}| \approx 2.29$ . For a comparison, a reduction with weighting (7.42) and a uniform snapshot placement produces a value of 3.48; a reduction with non-uniform snapshot placement and no weighting produced a value of 2.91; unmodified reduction shown in Figure 7.12 produced a value of 7.18.

Besides the first-order sensitivity estimates, the dependence of an output function (7.37) on features in the setup of reduction can also be characterized by a polynomial interpolation (see Section 2.2.2 for theoretical description). The idea [91], [92] is based on an argument that an *explicit, polynomial* estimation of the model response to the parameters of the reduction process is a more convenient tool for representing dependencies than either a table of instances of  $\mathfrak{S}$  for different reductions, or an explicit linear approximation based on derivatives. We note that using interpolating models results in a quality trade-off issue: while such

approximations can perform very well locally, there is no guarantee of global quality.

Since non-uniform placement of snapshots has proven relatively more important than other reduction modifications, we now show how to construct an interpolating model of the model response to snapshot placement. Suppose the number of the snapshots must remain fixed. We set

$$\hat{\mathfrak{S}}(\hat{u}) \approx \hat{\Gamma}(\Delta t_j) = \sum \chi \psi(\Delta t_j) \quad (7.43)$$

where the reduced model solution  $\hat{u}$  is based on the collection of snapshots

$$U_o = \{u_o(t_j + \Delta t_j)\}, \quad j = 1, 2, \dots, N \quad (7.44)$$

with the deviations from the regular snapshot placement  $\Delta t_j$  used as variables for the polynomial expansion  $\hat{\Gamma}$ . This choice of parameters does not influence the details of interpolation construction, and makes the regression equations (2.69) more numerically stable, since the values of all variables lie in approximately the same range. Informally, measuring the model response to  $\Delta t_j$  is expected to answer the question about the importance of the time interval in the neighborhood of  $t_j$  for the reproduction of feature of interest in the reduced model.

We use a full polynomial basis  $\Psi$  of order 2 on 20 variables. It includes 231 polynomials, requiring as many runs of the reduction process to find the expansion coefficients  $\chi$ . In principle, with the use of (allowably imprecise) derivative information as in Section 2.2, we can reduce the number of runs to  $231/21 = 11$ . The required expression



The coefficients in the expansion are almost invariant to the choice of the sample values of the variables ( $0 < \Delta t_j \leq \frac{1}{2N} = 0.025$ ) used for the regression, and so can be treated as empirically reliable local measurements of the importance and correlations of the snapshots displacements  $\Delta t_j$ .

Once the interpolating model is constructed, it can be used to estimate the range and sensitivities of the output quantity  $\hat{\mathfrak{S}}$  at the computational cost of a few direct evaluations of polynomials. The factor importance information represented by  $\hat{\Gamma}$  is not limited to derivatives; it is also possible to use it in statistical measurements such as (2.76), (2.77), with the corresponding quadratures computed much faster.

Arguably, this interpolation may be used also to optimize the output function, as in [88], with an important difference that polynomial interpolation is only locally valid, and only empirically reliable.

By comparing the linear and non-linear coefficients in the expansion parts (7.46), (7.47), we conclude that the information contained in  $\hat{\Gamma}$  is not limited to first-order linear approximations of the model response. It therefore may produce more efficient results than a linear importance estimate (7.41). In fact, for our current example, a weighting scheme

$$w_j = \left| \frac{d\hat{\Gamma}}{d\Delta t_j} \right| \tag{7.48}$$

provides an improvement in performance of the reduced model. The results (obtained with a uniform placement of snapshots) are shown in Figure 7.16; we measured  $|\mathfrak{J} - \hat{\mathfrak{J}}| \approx 2.20$ .

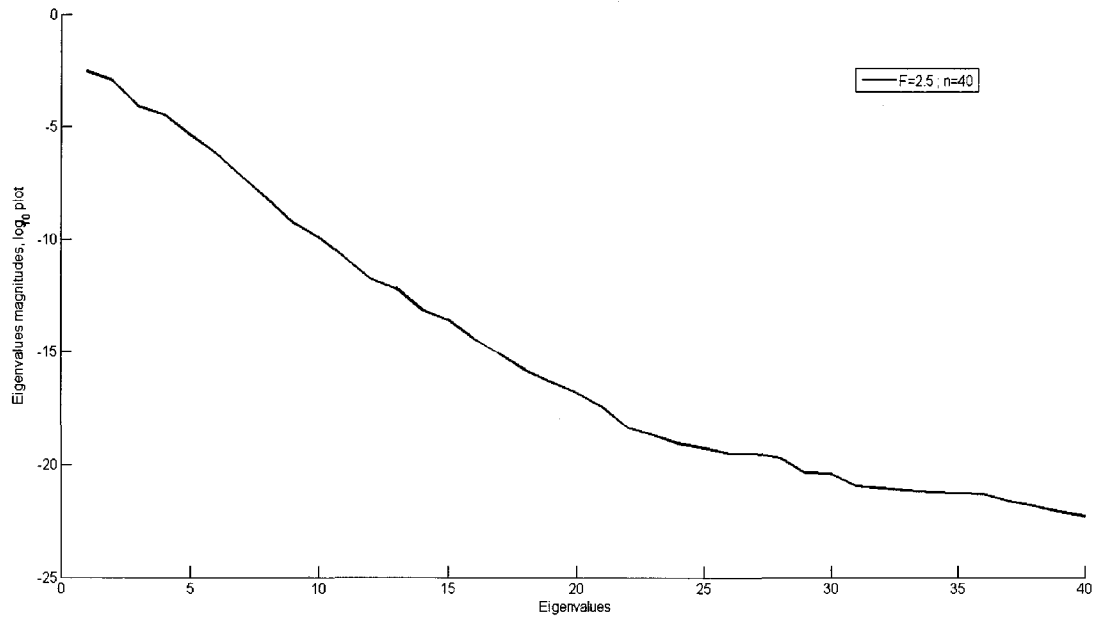
We conclude this section with a general remark on factor importance analysis in model reduction. Our use of sensitivity information is not fully automatic: as with this example, we often know what the important data components are, but deciding which modifications of the POD method to use takes several trials. The best combination of snapshot placement times, snapshot weights, and diagonal entries of the metric matrix can, in principle, be found as in [88], by solving an optimization problem (in this case, on 80 variables). In practice, the computational cost is too high to be justified by the expected improvement.

The implementation complexity for the procedures extracting required sensitivity information, differs from problem to problem. Since the features of interest, and the model reduction procedure details are still selected by inspection and subjective estimations of importance, there is currently no set of principles for comparison of usefulness of factor importance information.

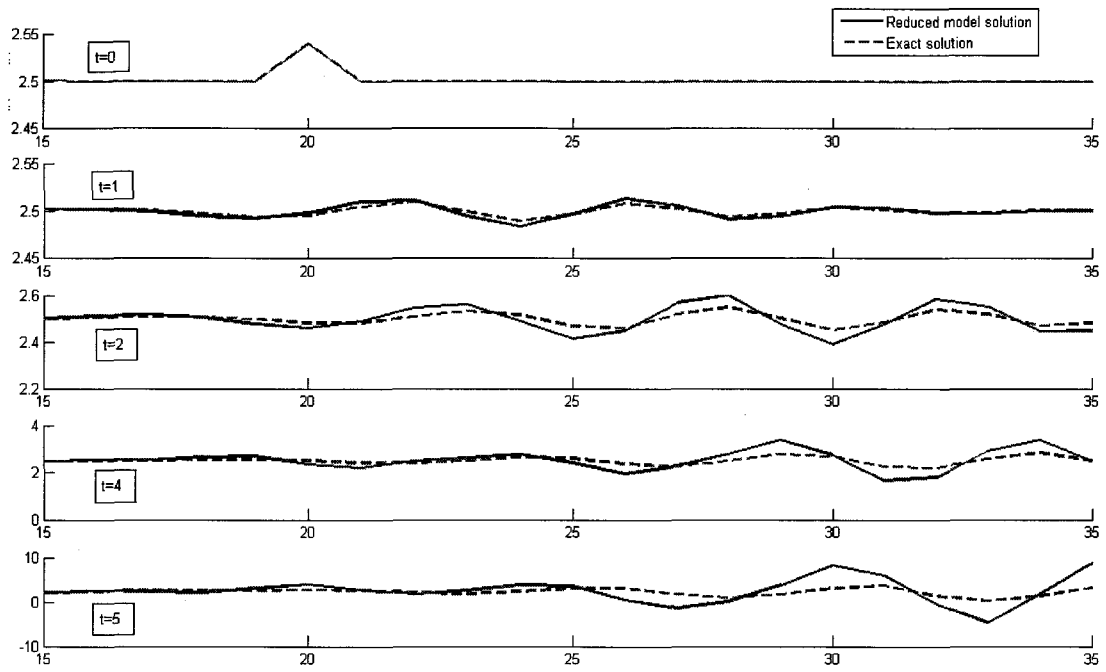
In our research, we observed that the suggested factor importance analysis (such as differentiation, polynomial approximation of sensitivity with respect to the model components and to the choices made in the reduction procedure) *reveals* structure not available by inspection of the reduced model state alone. At the current stage of development, model reduction is no longer a completely uncontrolled, trial-

and-error procedure. On the other hand, our analysis remains imprecise (due to subjective selection of what to observe), local, and *a posteriori*.

As noted previously, methods for formal *a priori* analysis of model reduction are not available available, and may not be because of fundamental reasons. In that case, the next stage of research is not so much an improvement of reduced model performance through even more advanced measurements, but a development of very fast sensitivity estimation procedures, and the use of reduced models in combination with advanced data assimilation techniques to correct the solution trajectories. Such techniques will benefit from the more basic material developed here.

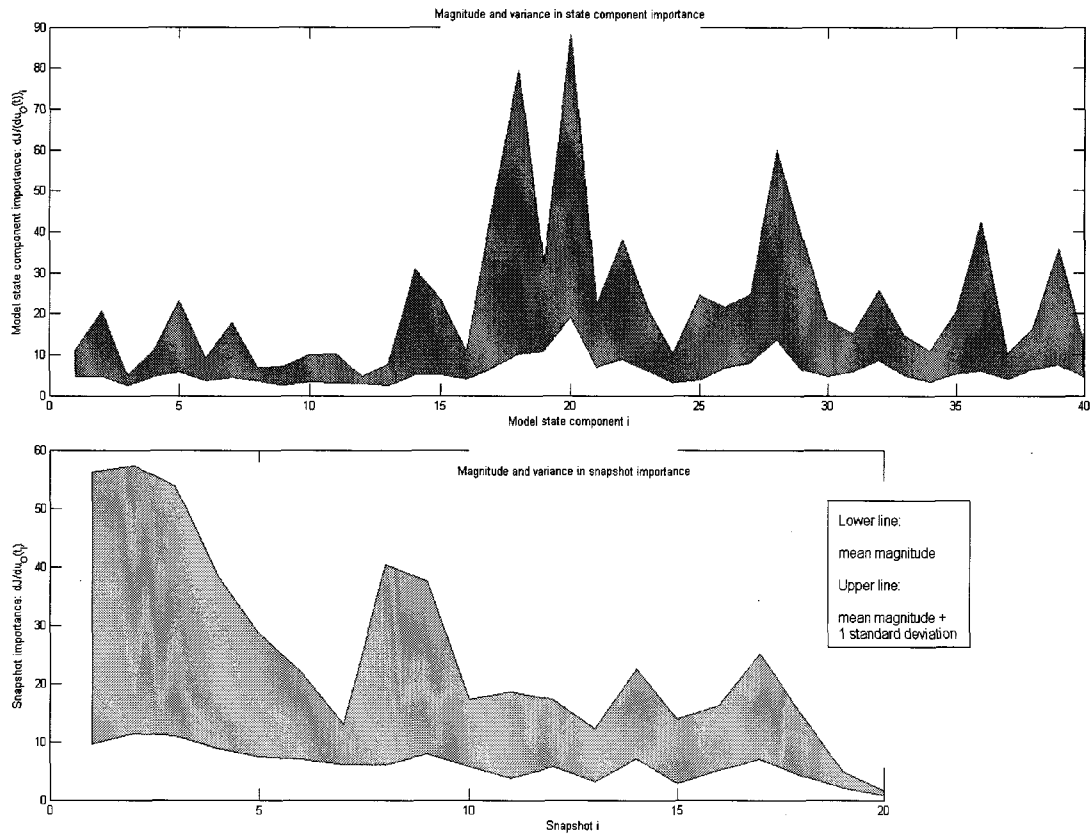


**Figure 7.11 First 40 eigenvalues of the Lorenz model covariance matrix.**

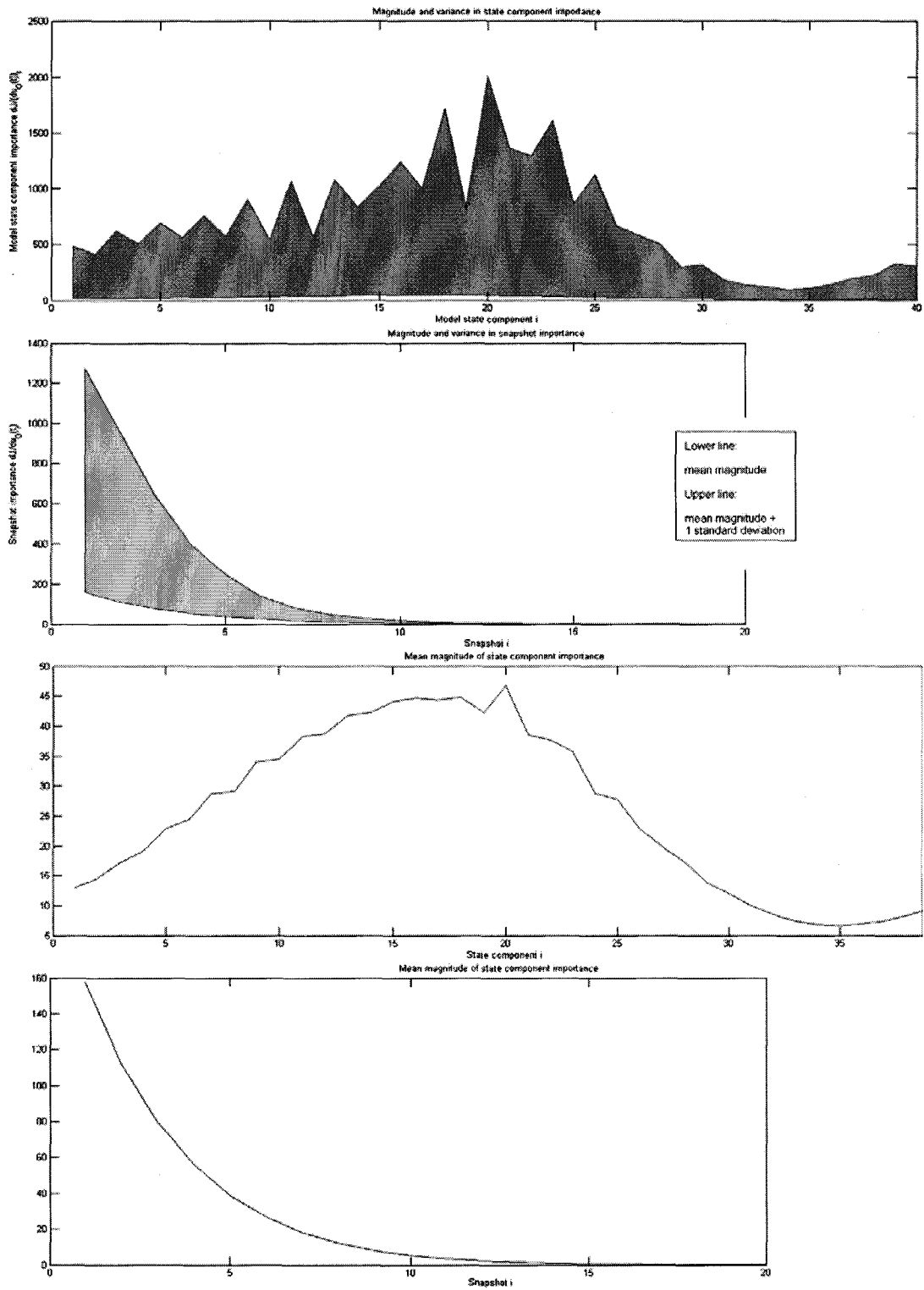


**Figure 7.12 Comparative performance of the full and reduced solutions for the Lorenz model.**

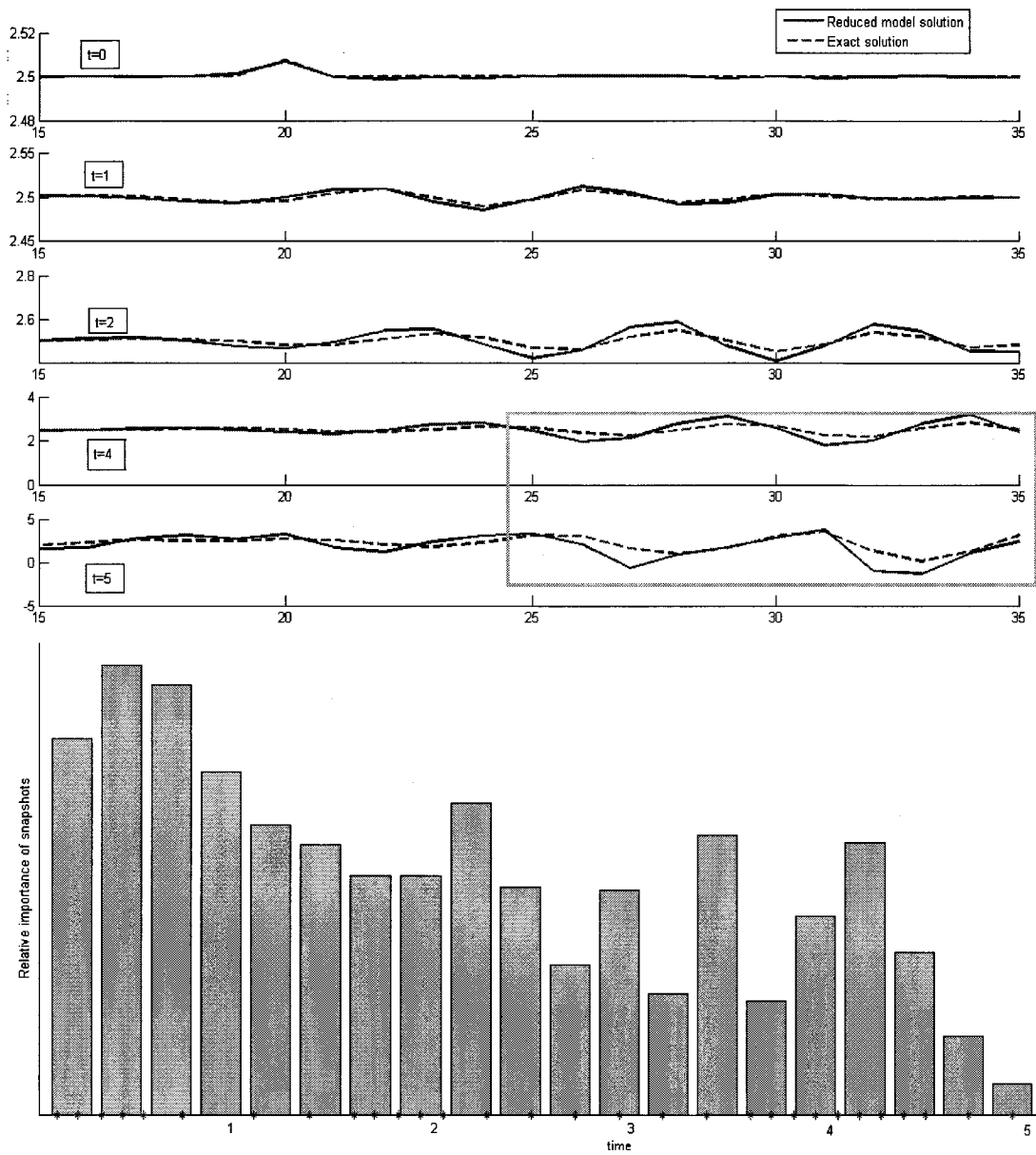




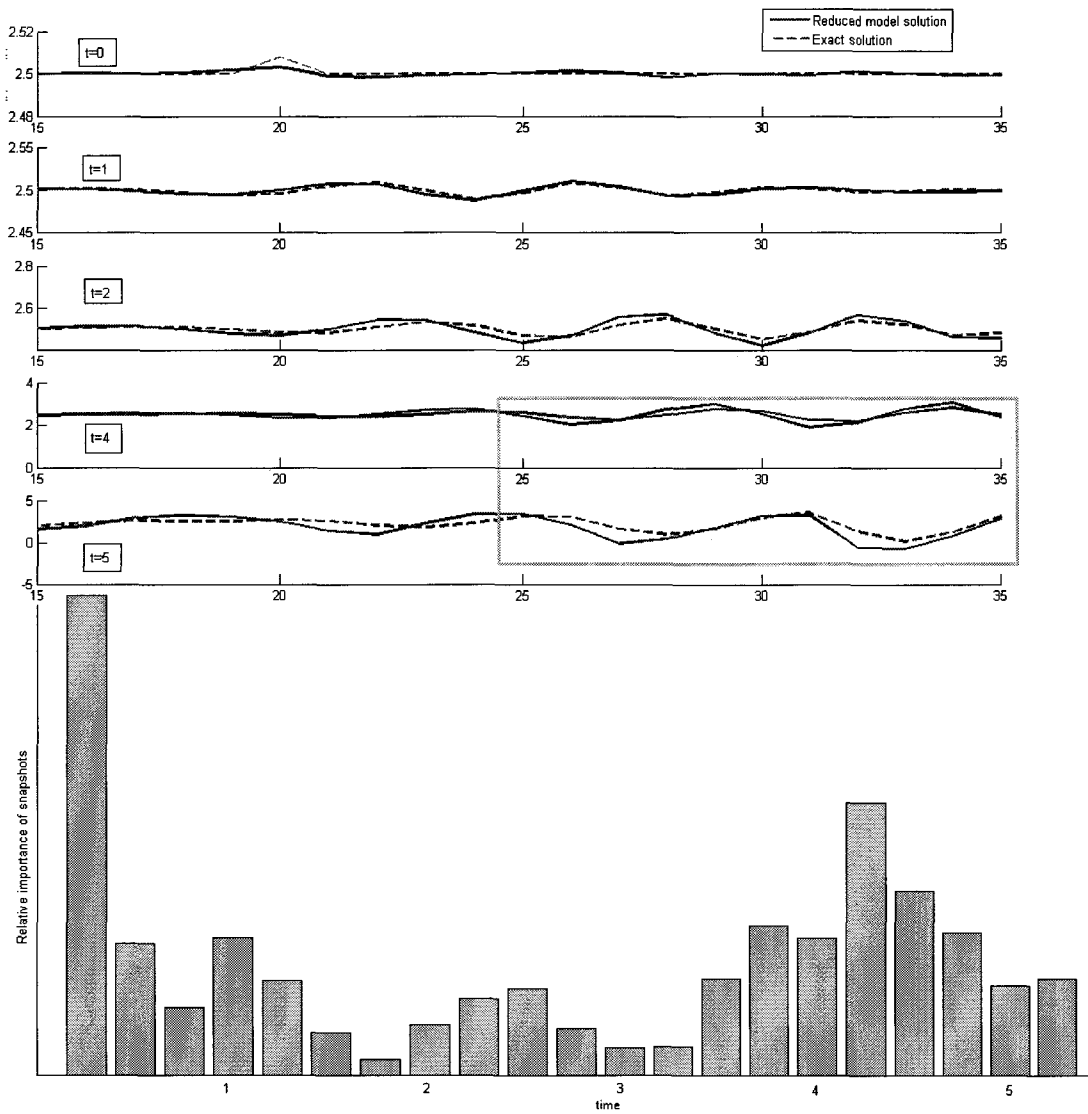
**Figure 7.13 Reduced Lorenz model, first-order sensitivity information**



**Figure 7.14 Full Lorenz model, first-order sensitivity information**



**Figure 7.15** Weighted reduction of the Lorenz model: performance of the model; snapshot weighting and placement scheme



**Figure 7.16** Weighted reduction of the Lorenz model: performance of the model; snapshot weighting chosen by polynomial interpolation

## 7.6 CHARNEY-DEVORE MODEL

We complete our overview of simple cases of SVD-based model reduction with a model suggested in [33] as an example of poor performance of the methods based on empirical orthogonal functions. We use a slightly modified form of the material used in the reference to give more substance to a general remark that not every kind of question can be answered with the help of model reduction.

The Charney-DeVore model used here is based on an advection-reaction system of ODEs, dependent on a number of parameters:

$$\begin{aligned}
 \frac{du_1}{dt} &= \gamma_1^* u_3 - C(u_1 - v_1) \\
 \frac{du_2}{dt} &= -(\alpha_1 u_1 - \beta_1) u_3 - C u_2 - \delta_1 u_4 u_6 \\
 \frac{du_3}{dt} &= (\alpha_1 u_1 - \beta_1) u_2 - \gamma_1 u_1 - C u_3 + \delta_1 u_4 u_5 \\
 \frac{du_4}{dt} &= \gamma_1^* u_3 - C(u_4 - v_4) + \varepsilon(u_2 u_6 - u_3 u_5) \\
 \frac{du_5}{dt} &= -(\alpha_2 u_1 - \beta_2) u_6 - C u_5 - \delta_2 u_4 u_3 \\
 \frac{du_6}{dt} &= (\alpha_2 u_1 - \beta_2) u_5 - \gamma_2 u_4 - C u_6 + \delta_2 u_4 u_2
 \end{aligned} \tag{7.49}$$

with the coefficients related to the physical quantities by:

$$\begin{aligned}
 \varepsilon &= \frac{16\sqrt{2}}{5\pi} & \alpha_i &= \frac{8\sqrt{2}}{\pi} \cdot \frac{i^2(i^2 + b^2 - 1)}{(4i^2 - 1)(i^2 + b^2)} \\
 \delta_i &= \frac{64\sqrt{2}}{15\pi} \cdot \frac{i^2 + b^2 - 1}{i^2 + b^2} & \beta_i &= \frac{\beta b^2}{i^2 + b^2} & i &= 1, 2 \\
 \gamma_i &= \frac{\gamma 4i}{4i^2 - 1} \cdot \frac{\sqrt{2}b}{\pi} & \gamma_i^* &= \frac{\gamma 4i^3}{4i^2 - 1} \cdot \frac{\sqrt{2}b}{\pi(i^2 + b^2)}
 \end{aligned} \tag{7.50}$$

The behavior of the physical model is determined by zonal flow forcing terms  $v_1, v_4$ , relaxation coefficient  $C$ , relative topographic height  $\gamma$ , and beta parameters  $b, \beta$ . The model is meant for long-term simulation of the flow, the unit of time is 1 day. The initial conditions are chosen arbitrarily, from the region

$$-1 \leq u_i(0) \leq 1, \quad i = 1, 2, \dots, 6 \quad (7.51)$$

The output of interest is the evolution of the components  $u_1, u_4$  that have a physical meaning of specie concentrations. We refer to the original source for more details on the meaning of the model.

An inspection of the model shows that at least for some parameter values, it may behave as expected of an advection-reaction model ODE with only a few degrees of freedom. For example, the setup

$$v_1 = 0.5, \quad v_4 = 0, \quad C = 0.05, \quad \beta = 0.25, \quad \gamma = 0.1, \quad b = 0 \quad (7.52)$$

allows construction of a reduced model with an unmodified method of snapshots. The first 3 eigenvalues of the covariance matrix consistently account for over 99.8% of the eigenvalue energy (2.33), leading to the estimate  $k = 3$  for the reduced model dimension.

The comparative performance of the full model and the reduced model obtained by an unmodified method of snapshots is shown in Figure 7.17. We did not include the initial integration period  $0 \leq t \leq 300$ , where the model exhibits fast transient behavior. During the transient stage the full and the reduced models may differ significantly, but the long-term behavior is reproduced correctly. If a more

precise reproduction of the full model behavior is required, the unreliable data of the fast transient period can be rejected and then replaced with the deduced values, as suggested in Section 2.4.3.

On the other hand, in the setup

$$v_1 = 0.95, \quad v_4 = -0.76095, \quad C = 0.1, \quad \beta = 1.25, \quad \gamma = 0.2, \quad b = 0.5 \quad (7.53)$$

the transient stage lasts for a long time, with the solution showing rapid transitions between two likely steady states (corresponding to distinct flow regimes of the physical model). The covariance matrix is very stiff, with the first 2 eigenvalues of the covariance matrix accounting for over 99.994% of the eigenvalue energy. The sensitivity properties of the model, however, prevent successful reduction.

In Figure 7.18 we show an example of two very different solution trajectories for the specie  $v_4$ , traced from the initial conditions taken within distance of 0.01 from each other in  $R^6$ . Multiple integrations of the model with varying initial conditions show that the trajectories of two types can be placed very closely, numerically dense with respect to each other for some time (note that the periods of rapid oscillations do not have to end at the same time for both trajectories).

We agree that the problem demonstrates some elementary flaws of our reduction methods, not related to the (largely preventable) instabilities of the reduced system over long integration intervals. This observation, however, points not so much to the poor understanding of SVD-based model reduction, as to the fact that it is possible to construct an example of the behavior that is inconsistent with the assumptions behind the method of snapshots.

Because of the closely placed distinct solution trajectories, such approaches as trial and error, or use of sensitivity information obtained from a small sample of trajectories, does not lead to an effective snapshot set that covers the whole integration period. The difficulty also applies to selecting the snapshot weights. If greater weights, or greater density of the snapshots is assigned to the region  $t > 1000$  where the solution trajectories are stable and lead to one of the two attractors, then only that attractor will be reproduced correctly. Experiments show that such reduced models do not exhibit transitions between regimes. On the other hand, a mixed set taken from stable trajectories leading to both attractors makes the reduced model conform to an average steady state that is never realized. Selective model reduction that, in principle, allows simultaneous use of several different sets of snapshots, and of several different optimal projections, does not apply here, since it requires associating projections with specific groups of components.

A partially acceptable solution is to take many snapshots from the fast transient period (possibly  $> 1000$  states), to increase the chances of capturing all the numerically reliable information. Since oscillations in the transient period are large in magnitude, in such collection of snapshots the points on the stable trajectory are mostly ignored. Our experiments show that such reduced models reproduce the full model behavior for some, but not for all initial conditions.

Rejection and subsequent recovery of the unreliable data is not applicable here, since immediately after the fast transient stage the model enters the steady state, where the evaluation of the model backwards in time is an ill-posed problem.

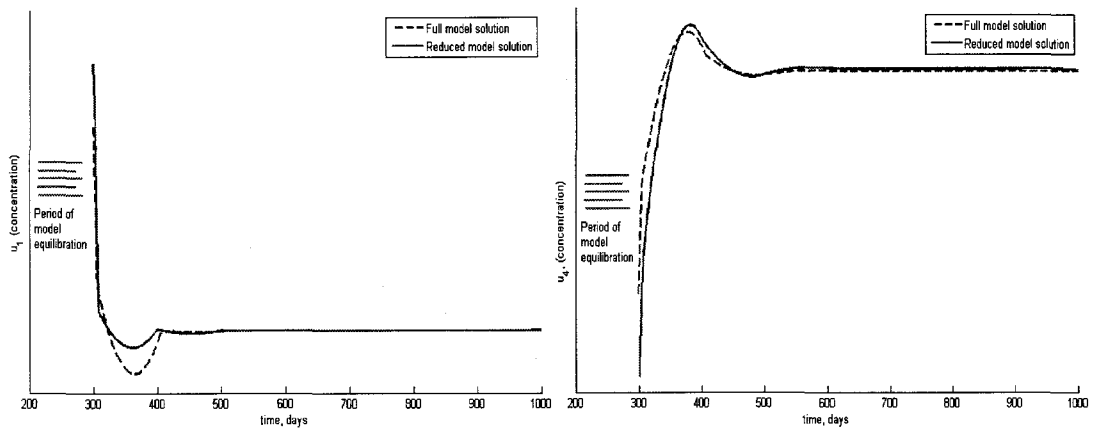


An original discussion of the model [33] suggested two tasks for validation of the reduced model. One is essentially the reproduction of the full model behavior during the fast transient period; the other is the convergence of the reduced model to a correct steady state, or the attractor in the region of the initial conditions. Our conclusion is that using SVD-based reduction methods alone, the second task cannot be consistently accomplished for both attractors at the same time.

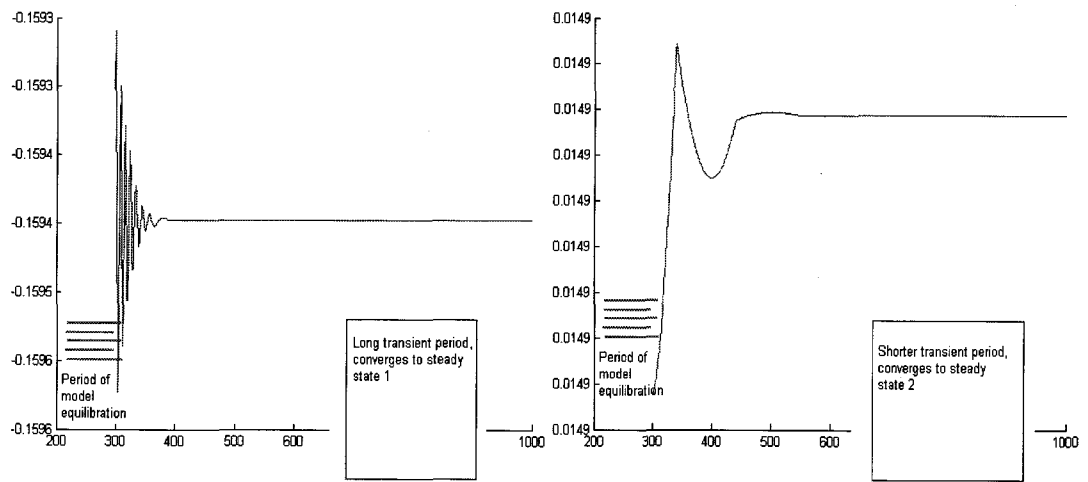
A correct representation of one attractor is possible to achieve using a metric based on the diagonal matrix that strengthens the representation of the variables of interest; for example,  $\Lambda_{1,1}, \Lambda_{4,4} = 1000$ , unit entries on the rest of the diagonal.

In Figure 7.19 we show a typical solution trajectory for the specie  $v_4$ , and also 2000 endpoints  $(v_1(T), v_4(T))$  resulting from the integration of the model over a long time interval,  $T = 4000$ . While one steady state was located correctly, about 50% of the integrations resulting from the uniform sampling of the initial conditions region (7.51) fell into a large, sharply defined attracting region, not obviously related to the full model dynamics.

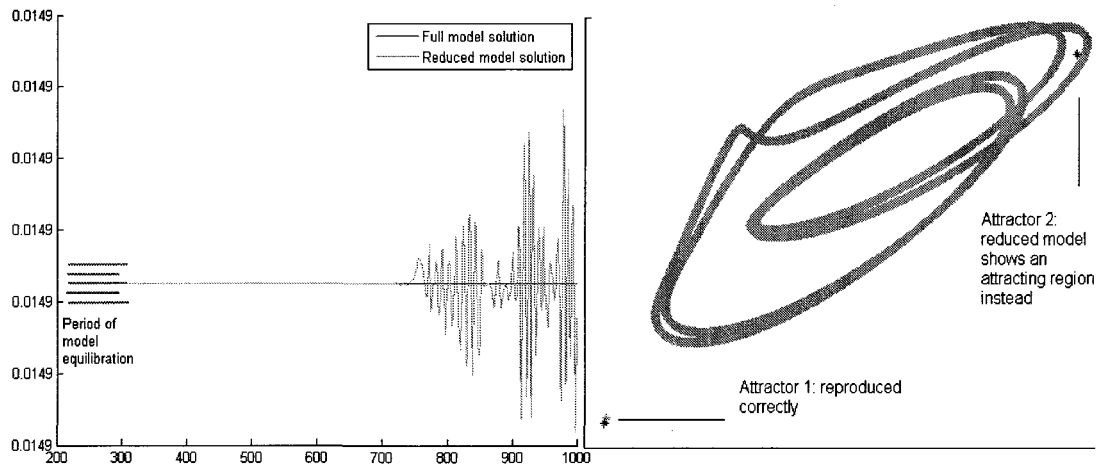
Given the difficulties with the construction of a reduced model that can effectively locate the attractor, we suggest that trying to reproduce more sophisticated features of the full model, such as chaotic shape of the attractor basin, or symmetries in the solution trajectories, is premature. In our applied work, we aim mainly for simulation of semi-periodic solutions, that are stable over long periods of time.



**Figure 7.17 Comparative performance of the full and reduced solutions for the Charney-DeVore model, stable setup: fast transient period not shown.**



**Figure 7.18** An example of the Charney-DeVore model solution trajectories: two distinct steady states



**Figure 7.19 Performance of the reduced solution in reproducing the chaotic features of the Charney-DeVore model: behavior during transient stage; placement of the attractors.**

## 7.7 SAPRC-99 MODEL

So far, we have shown how our suggestions on model reduction apply to simplified cases of large-dimensional reaction-transport models. Based on a set of test problems, we have explained how goal-oriented selection of weights, metric, and snapshot locations can be used to amplify the reproduction of features of interest in the reduced model performance. Our current understanding is that simple transport effects reduce nicely, and that once a reduced model is constructed, it is advantageous to use it to replace the full model in iterative searches. The main difficulty in implementing the proposed approaches lies in the complexity of a reaction term.

We shall now show that the developed material on reduction of ODEs of chemical kinetics also applies to models of industrial level complexity. We have identified such models as CBM-IV [128], SAPRC-99 [133], and GEOS-Chem [129], [130] as appropriate examples that have significance both for the atmospheric sciences (as major tools in air pollution prediction and control) and mathematics (as large-dimensional stiff ODEs). We chose SAPRC-99 as the central example; we find that other commonly used models in the field are either very similar structurally, or too large and computationally expensive to be within the scope of our work.

We shall now apply model reduction to this chemical mechanism. Since much of our work is novel (altogether, or for this class of problems), there is no good basis for the performance comparison. As we are interested both in the general

issues of implementing model reduction (such as correct use of sensitivity information to place snapshots), and the problem-specific properties (such as exploiting slow-fast behavior), we will report on both, in the same qualitative description style used in the text so far.

The main tasks are to demonstrate that a reduced model can efficiently reproduce the solution of the full model, and to provide a characterization of the behavior of the individual model state components (chemical species) in the context of model reduction.

#### 7.7.1 EXAMINATION OF THE CHEMICAL MECHANISM

At the time of our research (2006), the considered chemical model was the latest released update in the family of mechanisms designated SAPRC (named after Statewide Air Pollution Research Center, California), used to simulate the gas-phase atmospheric reactions of volatile organic compounds. It has been updated once, by a version SAPRC-07. The mechanism has been used in airshed models to determine absolute and relative ozone impacts of organic compounds emitted into the atmosphere, and for other control and research applications.

The mechanism has a complete form with over 400 and 550 inputs and outputs, correspondingly. A more convenient condensed form has 74 variables representing chemical species or groups of species, and 210 chemical reactions. We provide a partial description of model state components to chemical species in Table 8.1. For each specie or lumped variable, the table contains chemical notation

(grouped by classification), the corresponding variable number, an order of magnitude for the specie's average concentration, an order of magnitude for specie's average emission or deposit per second, and a proper name. We refer to the complete documentation [133] for more details.

As described in Section 6.1, the list of chemical equations, reactivity rates and emissions was automatically processed into an ODE, with the right-side function  $f(u,t)$  and its Jacobian  $J(u,t)$  recorded as Fortran procedures. We then converted the code to Matlab for convenience of integration with standard Rosenbrock ODE solvers (Section 6.2). Since some of the chemical reactions are very fast, the solutions are resolved up to the time step of one second.

The rates of some chemical reactions depend linearly on normalized sunlight intensity  $0 \leq SUN \leq 1$ , with an effective dependence on time modeled by a periodic expression similar to (7.4):

$$SUN(t) = \frac{1}{2} \left( 1 + \cos \left( \pi \left( \frac{2t_{local} - t_{sunset} - t_{sunrise}}{t_{sunset} - t_{sunrise}} \right)^2 \right) \right); \quad t_{local} \in [t_{sunrise}, t_{sunset}] \quad (7.54)$$

$$SUN(t) = 0 \quad t_{local} \notin [t_{sunrise}, t_{sunset}]$$

with the time variable converted from seconds from the start of the model to hours in the current day by

$$t_{local} = \frac{t}{3600} \quad \text{mod}_{24} \quad (7.55)$$

The local sunrise and sunset hours are fixed at the values

$$t_{sunrise} = 4.5, \quad t_{sunset} = 19.5 \quad (7.56)$$

placing the maximal sunlight intensity at  $t_{local} = 12$ .

Since ours is mainly a methodology study, we use the numerical code without modifications or correcting postprocessing, possibly resulting in some loss of chemical insight. We treat all the information included in the condensed model equations as completely reliable. In reality, there is some uncertainty associated with definitions of variables and the reactivity rates, and additional interpretation procedures are added onto the base mechanism. There is also uncertainty associated with initial conditions for the ODE; if they represent an unrealistic state of the system, the solution may exhibit atypical model behavior for a while. To resolve this last issue, we first integrate the model over a fixed time interval (on the order of one day), and use this *equilibrated* final state as more reliable initial conditions. We note that this is a standard practice for such models.

To assist model reduction, we shall now provide a characterization of the properties of the model that are available by inspection. We note that even with very few measurements used to characterize the behavior of each individual model state component, the resulting amount of data for 74 variables is almost too large for a reader to inspect and draw conclusions efficiently. We shall provide as much information as sufficient to improve the performance of reduced models. Additional factor importance information can be obtained (as shown in Section 2.2), but only for a small selection of model components of interest.

In Figure 7.20, we plot the solution of the full ODE (over an integration period of 72 hours, starting at  $t_{local} = 0$ ), for a selection of variables



( $i = 5, 13, 18, 25, 26, 42, 43, 57, 65, 69, 71, 73$ ). The individual species chosen for display demonstrate most of typical evolution patterns. We recorded the species behavior as it appeared in the solution of the ODE, without additional post-processing, or use of observational data. We note that the long-term behavior of some species (for example, unbounded growth in concentration) is unrealistic; the mathematical implementation of SAPRC-99 is effective only over time intervals on the order of 1 day.

At very low resolution in time, each trajectory appears to behave monotonously, leading to an informal expectation that the correlations between components observed in snapshots stay approximately the same as the model evolves. The solution resolved to hours shows a complicated (though non-chaotic, smooth) set of trajectories, with occasional change in individual concentrations up to an order of magnitude.

The periodic peaks in the evolution of concentration (present in some form for at least 70 components, clearly visible for variables 26, 57, 65, 73) are due to fast chemical reactions initialized by introduction of sunlight. The rapid production or consumption of each the specie continues until the model achieves an approximate balance in the quantities of reagents and continues to evolve more slowly. For 68 species, this results in a period of rapid change on the interval  $4.5 \leq t_{local} \leq 10$ , with a peak at approximately  $t_{local} = 7$  (exact placement depends on the specie). We

informally define the transient period of a specie evolution as an interval  $(t_{start}, t_{end})$  with an unusually fast change in concentration at the start and at the end:

$$t_{start} = \min(t_{local}), \quad t_{end} = \max(t_{local}): \quad \left| \frac{du_i(t_{local})}{dt} \right| \gg \frac{du_i(t)}{dt} \quad (7.57)$$

For practical purposes, we considered a derivative value *large* if it were a statistical outlier, i.e. placed at the distance of 3 standard deviations or more from the mean value. The expression (8.4) then describes multiple intervals, we took the longest.

To assist factor importance analysis, we provide measurements of individual species dynamics in Table 8.2. We record the magnitude of each component: at the start of the integration (after equilibration), average taken over a short time interval (1 day), average taken over a long time interval (10 days). We also record an approximate placement of the transient period, and the magnitude of the peak (as compared to the mean concentration of the specie).

Finally, we assign a number of informal labels describing features of specie evolution available by inspection. The label *fast* was assigned to 53 species with rapid concentration change during the transient period, with a peak magnitude close to or over 200% of the mean concentration. We denote all such species, in combination with corresponding transient time intervals, as the model's fast manifold (as defined in Chapter 2). We note that many species exhibited several periods of fast change, of secondary importance, due to much smaller amplitudes. For our purposes we attribute them, together with the rest of the model dynamics, to the *slow manifold*.

The label *smooth* was assigned to 12 species that did not exhibit a large amplitude during the transient period. At low resolution in time, such species appear to be unaffected by periodic events. The labels *increasing* (10 species) and *decreasing* (6 species) were assigned to the components that do not conserve the average concentration over time (change in magnitude is shown in columns 2-4 of Table 8.2). The presence of such species degrades the effectiveness of the model over longer integration intervals.

For an additional characterization of the model structure, we recall that the right-side expression  $f(u,t)$  is quadratic in  $u$  by design: (6.1), (6.2). The placement of the non-zero entries in the right-side Jacobian  $J(u,t)$  can be viewed as a summary of the direct reactions between the species. We visualize a  $74 \times 74$  sparsity pattern in Figure 7.21 (the denser region in the lower right corner does not correspond to any obvious chemical classification, the variables were automatically numbered for numerical efficiency [132]).

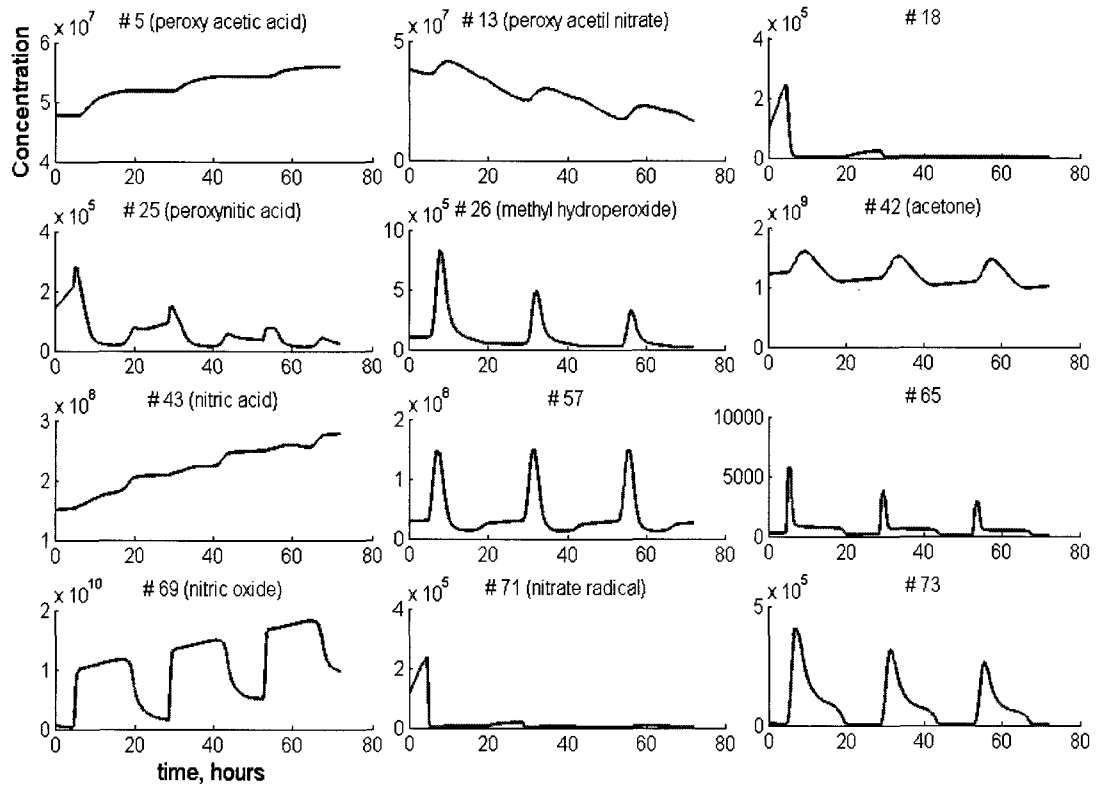
The sparsity of the Jacobian is approximately 15%. The rank of the matrix is also small (we observed 11–20 over the fast manifold, and 5–10 over the slow manifold). This leads to an informal conclusion that the effective number of degrees of freedom of the model is significantly less than the full dimension  $n = 74$ . Please note that such observations do not constitute a formal condition for model reduction (counter-examples are provided in Section 7.5 and 7.6 correspondingly).

While examination of the model structure may assist some of the steps of model reduction, we will mostly treat the system of ODEs as an (abstract)

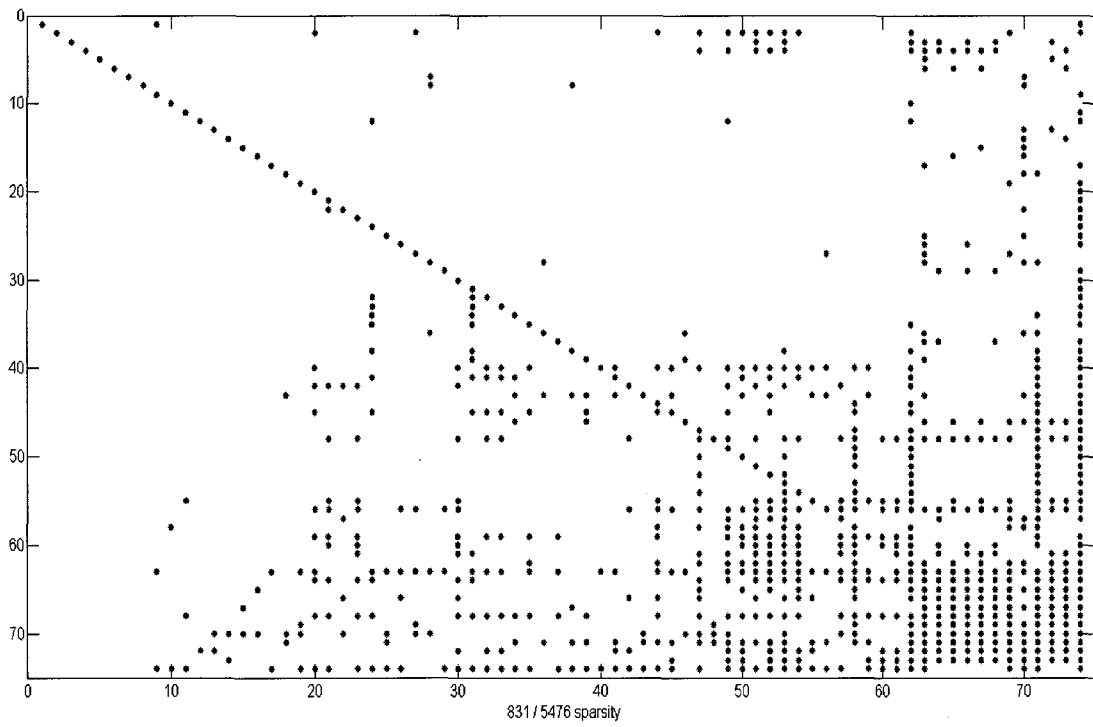
mathematical model. The choice of features of interest may be arbitrary. To provide some perspective on the applied meaning of our work, we will observe a selection of several ecologically significant species in the context of model reduction and factor importance analysis:

$$\{u_i: i = 62,69,70,43,56,13\} \quad (7.58)$$

This selection includes  $O_3$  (ozone),  $NO$  (nitric oxide),  $NO_2$  (nitrogen dioxide),  $HNO_3$  (nitric acid),  $HCHO$  (formaldehyde),  $PAN$  (peroxy acetyl nitrate); the list (7.58) was taken from the performance reviews of the updated version of SAPRC, available at [133].



**Figure 7.20 SAPRC99 solution, 72 hours**



**Figure 7.21 SAPRC99 right-side Jacobian sparsity pattern**

**Table 8.1 List of model species used in SAPRC-99 mechanism**

Type, notation.	Variable number	Concentration/ emission per second; log <sub>10</sub> order of magnitude	Description
<b>Constant species</b>			
O2, H2O, H2, M	N/A		Oxygen, water, hydrogen, generic medium ("air")
<b>Active inorganic species</b>			
O3	62	9	Ozone
NO	69	9	Nitric Oxide
NO2	70	9	Nitrogen Dioxide
NO3	71	3	Nitrate Radical
N2O5	18	3	Nitrogen Pentoxide
HONO	19	7 / 2	Nitrous Acid
HNO3	43	8	Nitric Acid
NHO4	25	4	Peroxynitric Acid
CO	40	10	Carbon Monoxide
SO2	9	9	Sulfur Dioxide
<b>Active radical species and operators</b>			
HO	74	7	Hydroxyl Radicals
HO2	63	6	Hydroperoxide Radicals
C O2	66	5	Methyl Peroxy Radicals
RO2_R	68	5	(operator) NO to NO2 conversion, HO2 formation
R2O2	48	5	(operator) NO to NO2 conversion, no HO2 formation
RO2_N	64	4	(operator) NO consumption, organic nitrate formation
CCO O2	72	5	Acetyl Peroxy Radicals
RCO O2	73	4	Peroxy Propionyl and higher acyl Radicals
BZCO O2	67	3	Peroxyacyl Radical from Aromatic Aldehydes
MA RCO3	65	2	Radicals from Acroleins Peroxyacyl
<b>Steady state radical species</b>			
O3P	58	7	Ground State Oxygen
O1D	10	-1	Excited State Oxygen
BZ O	46	4	Phenoxy Radicals
BZ(NO2) O	28	-11	Nito-substituted Phenoxy Radicals
HOCOO	27	0	Radical from Formaldehyde with HO2
<b>PAN and analogues</b>			
PAN	13	7	Peroxy Acetyl Nitrate
PAN2	14	6	PPN and higher alkyl PAN analogues
PBZN	15	5	PAN analogues from Aromatic Aldehydes
MA PAN	16	5	PAN analogues from Methacrolein
<b>Explicit and lumped molecule reactive organic product species</b>			
HCHO	56	8 / 3	Formaldehyde
CCHO	55	7 / 2	Acetaldehyde
RCHO	59	7 / 2	C3+ Aldehydes
ACET	42	9 / 4	Acetone
MEK	60	7 / 3	Ketones and other slow-reacting products

**Table 8.1 (continued):**

Type, notation.	Variable number	Concentration/ emission per second; log <sub>10</sub> order of magnitude	Description
MEOH	29	7 / 3	Methanol
COOH	26	4	Methyl Hydroperoxide
ROOH	37	5	Higher organic hydroperoxides
GLY	45	6 / 1	Glyoxal
MGLY	41	6 / 1	Methyl glyoxal
PHEN	39	6 / 2	Phenol
CRES	34	6 / 2	Cresols
NPHE	36	8	Nitrophenols
BALD	38	6 / 1	Aromatic Aldehydes
METHACRO	50	6 / 2	Methacrolein
MVK	54	5	Methyl Vinyl Ketone
ISOPROD	52	5 / 1	Isoprene products
<b>Lumped parameter products</b>			
PROD2	61	7 / 2	Ketones and other fast-reacting oxygenated products
RNO3	57	7	Organic Nitrates
<b>Uncharacterized reactive aromatic ring fragmentation products, RARFP</b>			
DCB1	35	6	Reactive aromatic ring fragmentation product, sorted by photolysis action
DCB2	32	5	
DCB3	33	5	
<b>Non-reacting species and low reactivity compounds</b>			
SULF	1	8	Sulfates
HCOOH	2	8	Formic Acid
CCO OH	3	7	Acetic Acid
RCO OH	4	7	Higher organic acids
CCO OOH	5	7	Peroxy Acetic Acid
RCO OOH	6	7	Higher organic peroxy acids
<b>Primary organics</b>			
ETHENE	44	7 / 3	Ethene
ISOPRENE	47	5 / 2	Isoprene
<b>Lumped parameter species</b>			
ALK1	11	8 / 3	Alkanes and non-aromatic compounds, sorted by reactivity
ALK2	20	8 / 3	
ALK3	21	9 / 4	
ALK4	30	8 / 4	
ALK5	23	8 / 3	
ARO1	31	7 / 3	Aromatics, sorted by reactivity
ARO2	24	7 / 3	
OLE1	51	7 / 3	Alkenes, sorted by reactivity
OLE2	52	5 / 1	
OLE3	53	7 / 3	
TRP1	49	6 / 2	Terpenes



**Table 8.2 Inspection of SAPRC-99 model dynamics**

#	Initial concentration	Average concentration (over 1 day)	Average concentration (over 10 days)	Transient phase; $t_{local}$ , hours.	Magnitude of the transient phase peak, % of 1-day mean	Labels
1	$4.33 \cdot 10^8$	$3.16 \cdot 10^8$	$2.04 \cdot 10^9$	N/A		<i>Smooth, increasing</i>
2	$2.30 \cdot 10^8$	$2.03 \cdot 10^8$	$4.65 \cdot 10^8$	8.5 – 9.8	113	<i>Smooth, increasing</i>
3	$7.15 \cdot 10^7$	$7.03 \cdot 10^7$	$7.51 \cdot 10^7$	6.8 – 8.8	101	<i>Smooth, increasing</i>
4	$2.38 \cdot 10^7$	$2.34 \cdot 10^7$	$2.49 \cdot 10^7$	6.5 – 8.5	101	<i>Smooth, increasing</i>
5	$5.17 \cdot 10^7$	$5.00 \cdot 10^7$	$5.69 \cdot 10^7$	6.9 – 8.9	103	<i>Smooth, increasing</i>
6	$1.57 \cdot 10^7$	$1.53 \cdot 10^7$	$1.71 \cdot 10^7$	6.6 – 8.6	102	<i>Smooth, increasing</i>
7	$2.33 \cdot 10^7$	$2.24 \cdot 10^7$	$2.70 \cdot 10^7$	2.2 – 4.6	104	<i>Smooth, increasing</i>
8	$5.23 \cdot 10^9$	$5.22 \cdot 10^9$	$5.28 \cdot 10^9$	3.9 – 7.0	100	<i>Smooth, increasing</i>
9	$4.66 \cdot 10^9$	$4.14 \cdot 10^9$	$8.15 \cdot 10^9$	N/A		<i>Smooth, increasing</i>
10	0	0.61	0.41	6.7 – 7.9	<b>261</b>	<i>Fast</i>
11	$2.76 \cdot 10^8$	$2.92 \cdot 10^8$	$2.63 \cdot 10^8$	N/A		
12	$2.09 \cdot 10^6$	$2.05 \cdot 10^6$	$1.95 \cdot 10^6$	5.1 – 7.7	<b>213</b>	<i>Fast</i>
13	$2.88 \cdot 10^7$	$3.64 \cdot 10^7$	$1.61 \cdot 10^7$	6.6 – 8.0	113	<i>Smooth, decreasing</i>
14	$4.08 \cdot 10^6$	$8.98 \cdot 10^6$	$4.23 \cdot 10^6$	5.8 – 7.2	<b>218</b>	<i>Fast</i>
15	$1.67 \cdot 10^5$	$2.91 \cdot 10^5$	$1.26 \cdot 10^5$	5.0 – 6.9	<b>199</b>	<i>Fast</i>
16	$1.98 \cdot 10^5$	$3.28 \cdot 10^5$	$1.17 \cdot 10^5$	4.7 – 5.3	<b>330</b>	<i>Fast</i>
17	$3.38 \cdot 10^5$	$7.43 \cdot 10^5$	$2.17 \cdot 10^5$	6.5 – 7.9	<b>261</b>	<i>Fast</i>
18	$1.69 \cdot 10^4$	$4.18 \cdot 10^4$	$4.94 \cdot 10^3$	4.6 – 5.5	<b>587</b>	<i>Fast, decreasing</i>
19	$2.10 \cdot 10^7$	$1.33 \cdot 10^7$	$2.68 \cdot 10^7$	5.0 – 6.1	158	<i>Increasing</i>
20	$1.60 \cdot 10^8$	$1.41 \cdot 10^8$	$1.39 \cdot 10^8$	8.3 – 10.0	<b>193</b>	<i>Fast</i>
21	$3.26 \cdot 10^8$	$2.38 \cdot 10^8$	$2.42 \cdot 10^8$	6.9 – 8.9	<b>245</b>	<i>Fast</i>
22	0.061	8.43	8.41	5.1 – 6.6	<b>400</b>	<i>Fast</i>
23	$1.73 \cdot 10^8$	$9.97 \cdot 10^7$	$1.04 \cdot 10^8$	5.6 – 7.2	<b>332</b>	<i>Fast</i>
24	$4.36 \cdot 10^7$	$2.23 \cdot 10^7$	$2.36 \cdot 10^7$	5.2 – 6.4	<b>390</b>	<i>Fast</i>
25	$7.47 \cdot 10^4$	$9.25 \cdot 10^4$	$2.96 \cdot 10^4$	4.7 – 6.5	<b>232</b>	<i>Fast, decreasing</i>
26	$4.78 \cdot 10^4$	$1.74 \cdot 10^5$	$4.99 \cdot 10^4$	6.3 – 9.2	<b>307</b>	<i>Fast</i>
27	0.045	2.18	1.07	5.6 – 8.6	<b>366</b>	<i>Fast</i>
28	0	$1.88 \cdot 10^{-11}$	$2.13 \cdot 10^{-11}$	4.5 – 4.9	<b>192</b>	<i>Fast</i>
29	$5.34 \cdot 10^7$	$4.866 \cdot 10^7$	$4.77 \cdot 10^7$	8.5 – 10.2	183	
30	$2.61 \cdot 10^8$	$1.69 \cdot 10^8$	$1.74 \cdot 10^8$	6.1 – 8.0	<b>286</b>	<i>Fast</i>
31	$7.03 \cdot 10^7$	$4.29 \cdot 10^7$	$4.45 \cdot 10^7$	5.8 – 7.6	<b>308</b>	<i>Fast</i>
32	$6.64 \cdot 10^5$	$5.049 \cdot 10^5$	$4.89 \cdot 10^5$	4.9 – 6.7	<b>245</b>	<i>Fast</i>
33	$4.99 \cdot 10^5$	$3.15 \cdot 10^5$	$2.94 \cdot 10^5$	4.9 – 6.5	<b>214</b>	<i>Fast</i>
34	$4.01 \cdot 10^6$	$2.48 \cdot 10^6$	$2.59 \cdot 10^6$	5.0 – 6.9	<b>288</b>	<i>Fast</i>
35	$3.51 \cdot 10^6$	$2.93 \cdot 10^6$	$2.92 \cdot 10^6$	5.0 – 6.7	<b>297</b>	<i>Fast</i>
36	$2.31 \cdot 10^8$	$2.04 \cdot 10^8$	$5.05 \cdot 10^8$	5.6 – 7.5	113	<i>Smooth, increasing</i>
37	$6.00 \cdot 10^4$	$2.75 \cdot 10^5$	$7.70 \cdot 10^4$	5.7 – 8.7	<b>296</b>	<i>Fast</i>
38	$1.74 \cdot 10^6$	$1.38 \cdot 10^6$	$1.37 \cdot 10^6$	5.0 – 7.5	<b>198</b>	<i>Fast</i>
39	$3.09 \cdot 10^6$	$1.63 \cdot 10^6$	$1.72 \cdot 10^6$	5.2 – 6.4	<b>370</b>	<i>Fast</i>
40	$1.66 \cdot 10^{10}$	$1.71 \cdot 10^{10}$	$1.72 \cdot 10^{10}$	6.7 – 8.5	112	
41	$3.83 \cdot 10^6$	$3.70 \cdot 10^6$	$3.69 \cdot 10^6$	5.1 – 7.3	<b>235</b>	<i>Fast</i>
42	$1.12 \cdot 10^9$	$1.27 \cdot 10^9$	$1.13 \cdot 10^9$	6.1 – 7.5	125	

**Table 8.2 (continued):**

#	Initial concentration	Average concentration (over 1 day)	Average concentration (over 10 days)	Transient phase; $t_{local}$ , hours.	Magnitude of the transient phase peak, % of 1-day mean	Labels
43	$2.07 \cdot 10^8$	$1.75 \cdot 10^8$	$2.95 \cdot 10^8$	17.7 – 19.3	117	<i>Smooth, increasing</i>
44	$1.07 \cdot 10^8$	$6.02 \cdot 10^7$	$6.30 \cdot 10^7$	5.5 – 7.1	337	<i>Fast</i>
45	$3.77 \cdot 10^6$	$4.26 \cdot 10^6$	$4.17 \cdot 10^6$	5.1 – 7.5	250	<i>Fast</i>
46	31.92	$3.72 \cdot 10^4$	$3.11 \cdot 10^4$	5.3 – 6.2	433	<i>Fast</i>
47	$1.72 \cdot 10^6$	$7.60 \cdot 10^5$	$8.20 \cdot 10^5$	4.7 – 5.3	474	<i>Fast</i>
48	$2.53 \cdot 10^4$	$6.27 \cdot 10^5$	$3.00 \cdot 10^5$	4.6 – 6.2	300	<i>Fast</i>
49	$3.37 \cdot 10^6$	$1.47 \cdot 10^6$	$1.60 \cdot 10^6$	4.7 – 5.3	464	<i>Fast</i>
50	$6.55 \cdot 10^6$	$3.12 \cdot 10^6$	$3.30 \cdot 10^6$	4.9 – 6.0	410	<i>Fast</i>
51	$5.00 \cdot 10^7$	$2.40 \cdot 10^7$	$2.54 \cdot 10^7$	4.9 – 6.0	413	<i>Fast</i>
52	$7.08 \cdot 10^5$	$3.40 \cdot 10^5$	$3.46 \cdot 10^5$	5.2 – 6.0	429	<i>Fast</i>
53	$3.45 \cdot 10^7$	$1.54 \cdot 10^7$	$1.65 \cdot 10^7$	4.7 – 5.5	453	<i>Fast</i>
54	$1.93 \cdot 10^5$	$9.73 \cdot 10^4$	$8.92 \cdot 10^4$	4.8 – 6.3	292	<i>Fast</i>
55	$7.71 \cdot 10^7$	$8.72 \cdot 10^7$	$8.49 \cdot 10^7$	5.3 – 8.3	208	<i>Fast</i>
56	$2.63 \cdot 10^8$	$2.81 \cdot 10^8$	$2.69 \cdot 10^8$	5.4 – 8.9	145	
57	$2.78 \cdot 10^7$	$3.74 \cdot 10^7$	$3.61 \cdot 10^7$	5.3 – 8.9	200	<i>Fast</i>
58	0	$8.03 \cdot 10^7$	$6.79 \cdot 10^7$	4.8 – 7.2	245	<i>Fast</i>
59	$3.93 \cdot 10^7$	$3.98 \cdot 10^7$	$3.90 \cdot 10^7$	4.9 – 7.6	237	<i>Fast</i>
60	$1.23 \cdot 10^8$	$1.75 \cdot 10^8$	$1.66 \cdot 10^8$	6.3 – 7.1	243	<i>Fast</i>
61	$2.51 \cdot 10^7$	$2.46 \cdot 10^7$	$2.47 \cdot 10^7$	4.8 – 8.13	158	
62	$3.37 \cdot 10^9$	$1.22 \cdot 10^{10}$	$6.77 \cdot 10^9$	N/A		<i>Decreasing</i>
63	$1.38 \cdot 10^5$	$5.31 \cdot 10^6$	$2.69 \cdot 10^6$	4.7 – 6.9	266	<i>Fast</i>
64	$8.13 \cdot 10^3$	$1.10 \cdot 10^5$	$5.24 \cdot 10^4$	4.6 – 6.1	382	<i>Fast</i>
65	159.01	777.66	353.78	4.5 – 6.2	264	<i>Fast, decreasing</i>
66	$3.11 \cdot 10^4$	$1.27 \cdot 10^6$	$6.24 \cdot 10^5$	5.5 – 7.0	332	<i>Fast</i>
67	62.88	$1.41 \cdot 10^3$	693.10	5.19 – 7.5	412	<i>Fast, decreasing</i>
68	$6.92 \cdot 10^4$	$1.04 \cdot 10^6$	$4.94 \cdot 10^5$	4.6 – 6.1	360	<i>Fast, decreasing</i>
69	$2.32 \cdot 10^9$	$7.36 \cdot 10^9$	$2.27 \cdot 10^{10}$	4.7 – 5.4	158	<i>Increasing</i>
70	$1.08 \cdot 10^{10}$	$4.15 \cdot 10^9$	$3.61 \cdot 10^9$	4.7 – 5.4	262	<i>Fast</i>
71	$1.56 \cdot 10^4$	$4.11 \cdot 10^4$	$6.83 \cdot 10^3$	4.5 – 4.8	569	<i>Fast, decreasing</i>
72	$9.72 \cdot 10^3$	$3.84 \cdot 10^5$	$1.86 \cdot 10^5$	5.5 – 7.0	349	<i>Fast</i>
73	$2.61 \cdot 10^3$	$9.90 \cdot 10^4$	$4.84 \cdot 10^4$	5.3 – 6.5	411	<i>Fast</i>
74	$1.37 \cdot 10^5$	$5.91 \cdot 10^7$	$6.10 \cdot 10^7$	17.5 – 18.9	250	<i>Fast</i>

### 7.7.2 REDUCTION OF CHEMICAL MODEL

In this section, we overview model reduction of the chemical mechanism. There are no particular expectations of performance beyond qualitative reproduction of the full model behavior in at least some components (and numerical stability of the solution: the reduction is not effective if integration of the reduced model equations fails).

We shall first use an unmodified method of snapshots, and then, based on the observed performance of the reduced model, apply some of the improvements suggested in Chapter 2. The understanding of the reduced model effectiveness is very general, more specific measurements will be used in the following section.

To decide on the time interval over which the snapshots should be taken, and on the dimension of the reduced model, we examine the covariance information. The distribution of eigenvalues is shown in Figure 7.22, for several different versions of the covariance matrix. Each matrix was built using the same equilibrated initial conditions (recorded in Table 8.2), and a uniform distribution of 40 snapshots over integration intervals of 6, 12, 30 hours and 5 days. We observe an extreme stiffness of the eigenvalue set: the ratio of the largest and the smallest eigenvalues is at approximately 30 orders of magnitude. Consistently, as few as 5 first eigenvalues capture 99.99% of the eigenvalue energy (and 20–25 eigenvalues capture 100% , up to machine precision).

At this point, we expect that the reduced model will be efficient on the integration interval of 12–24 hours. For this short integration interval, the dimension of the reduced model is estimated as  $5 \leq k \leq 15$ .

Longer integration periods produce curves that are close to power-law distribution (a straight line on a logarithmic scale graph), the eigenvalue distribution does not have a characteristic sharp decline that has indicated the empirically correct degree of freedom for the model in some of the previous examples. In addition, for long integration periods, the full model tends to become unrealistic, and the reduced model equations numerically unstable.

We note that there are indications that our setup for reduction is acceptable only as a first guess. For instance, for such a large dimension and stiffness, the error estimate (2.33) is only useful in relative terms (since, according to it, an alignment error of magnitude  $10^{10}$  may be declared acceptable). Also, a uniform placement of observations may be an ineffective way to extract covariance information; such a snapshot ensemble captures both the relevant correlations in the long-term evolution of the model, and the unreliable information from the transient periods. Our response to such remarks is that it is more efficient to adjust the reduced model setup after several attempts than to optimize it using almost absent *a priori* knowledge.

In our experiments, we used the reduction based on a uniform distribution of 15 snapshots over 20 hours, the reduced model dimension is  $k = 10$ . Average computational time for the reduced model was 1.77 seconds (including 1.60

seconds of integration time, and 0.17 seconds for linear algebra operations); compare with 80.25 seconds for integrating the full model. For the chosen integration period, the model is already sensitive to the details in reduction setup: for example, the use of a large collection of snapshots (so that more of them fall into transient periods) may result in numerical instability. We compare the full and the reduced model performance in Figure 7.23; the plots are for the species of interest defined in (7.58).

Suppose that we are allowed to modify the number and placement of snapshots. It is useful to think of the performance of the reduced model as a result of a trade-off between numerical stability and correctness of dynamics. More snapshots taken over a longer integration period contain more information about the full model, but increase the chance of numerical instability, amplify unreliable information, and provide too many points for the reduced model to conform to. On the other hand, too few snapshots may not contain enough information. Very short integration intervals will require integration restarts, with associated errors. A tradeoff between quality and computation time also takes place, but is less complicated to manipulate, since our only value of influence is the reduced model dimension  $k$ .

For a short integration interval, it is possible to achieve a good coincidence of the reduced and the full model behavior, using just the snapshot placement and weighting. An approximate understanding of the model dynamics and the importance of factors is sufficient. There are several options for setting up the

reduction. An improvement in the performance can be based on the understanding that not all the information contained in the snapshots is relevant or reliable. By observation of the model, we decide that this may be the case for the fast transient periods (a different analysis may identify another source of unreliable data, with the same processing steps as below).

The measurements recorded in Table 8.2 allows us to map the 53 *fast* transient intervals, and avoid them, or dampen their influence in estimating the covariance information. An approximate distribution of the fast manifold over time and model components is shown in Figure 7.24 (resolved in time up to 20 minutes of  $t_{local}$ ). The time intervals covered by the fast manifold are approximately  $5 \leq t_{local} \leq 11$ ,  $18 \leq t_{local} \leq 19$ . Unsystematic deletion of several snapshots from the indicated time intervals occasionally improves the reduced model performance. Removing all of them, however, results in a numerically stable reduced model that does not follow the correct trajectory; see Figure 7.25 for a typical performance (40 snapshots uniformly distributed over the allowed time intervals were used).

We explain this failure in the reduced model by ignoring too much of the relevant information. Without completely rejecting the idea that the data from the fast manifold is detrimental to the performance of the reduced model, we shall now apply the tools developed to suppress, rather than to completely exclude the data. The choice is between *event targeting* (Section 2.4.1) that allows amplification or damping of an arbitrary features of interest; and *selective model reduction* (Section

2.3) that projects a feature of interest using a different reduction from the rest of the model.

### Event targeting

The approach we call *event targeting* consists of applying distinct treatment either to all state components for selected time instances, or for all time instances for selected components. It applies best when the feature of interest is a single rectangular region in the snapshot ensemble (see Figure 2.1). A more complicated shape can be represented by a combination of overlapping rectangles (an “etch-a-sketch” drawing). The sequence of amplification and damping effects can be generated automatically if the expected benefits justify the effort (of solving, essentially, an optimal tiling problem).

We use a guided, non-optimal sequence to construct a diagonal metric  $\Lambda$  and a sequence of weights  $\{w\}$ :

$$w_i = 1 - \Delta_1 |M^F \cap \{(u_o(t_i))_j\}| + \Delta_1 |M^S \cap \{(u_o(t_i))_j\}|: \quad i = 1, 2, \dots, N; j = 1, 2, \dots, n \quad (7.59)$$

$$\Lambda_{j,j} = 1 - \Delta_2 |M^F \cap \{(u_o)_i\}| + \Delta_2 |M^S \cap \{(u_o)_i\}|: \quad i = 1, 2, \dots, N; j = 1, 2, \dots, n \quad (7.60)$$

where  $M^F, M^S$  are the fast and the slow manifolds, as defined in Chapter 2.

In the absence of information on the importance of individual components for the event representation, the expression  $|M \cap \dots|$  represents some measurement of the intersection of the snapshot with the event. For the discussed problem, we use a simple count of instance to determine weights and metric components. More specifically, in (7.59) we count how many model components are going through the

transient period during current time instance; in (7.60) we count how many snapshots fall into the transient period for this model component. The empirical coefficients are set to

$$\Delta_1 = 0.05, \quad \Delta_2 = 0.001 \quad (7.61)$$

resulting in a (normalized) distribution of 20 weights from 0.0213 to 0.0546 and 74 diagonal metric entries from 0.9873 to 1.000.

The modifications made to the default metric are somewhat weak; a higher value for  $\Delta_2$  may be more efficient. On the other hand, due to the presence of important components with of very small magnitude, the model is very sensitive to some changes to the metric, resulting in a risk of numerical instability.

The application of (7.59), (7.60) provides a uniform sweep of the snapshot set that dampens all information belonging to the fast manifold, and amplifies all information belonging to the slow manifold. Some elements receive contradictory treatment that cannot be completely compensated for (though additional sweeps with varying values of coefficients (7.61) provided small improvement in some experiments). Figure 7.26 shows an example of performance of the reduced model created with event targeting; note a clear improvement in comparison with results of an unmodified setup shown in Figure 7.23.

Some variations are possible for the event targeting approach, assuming either less or more detail in the description of the event. If only a small number of snapshots can be collected (for example, due to limitations on computational time), it is possible to place no snapshots outside of the feature of interest (no snapshots on



the fast manifold, in this case), and still achieve performance comparable to just shown. Some snapshots should be located on the boundary of the feature of interest (defined with reasonable precision), with the information contained in them receiving higher weight. Alternatively, instead of limiting the description of the fast manifold to just the boundaries, we decompose it into several features of different importance (such as “faster transient period”, “slower transient period”, “peak”), and dampen or amplify such features to a varying extent.

Geometrically, the event targeting approach projects all data into a reduced order space with the basis obtained using a modified procedure, in which some of the data in the set of snapshots is amplified or dampened. The obtained basis is optimal for the weighted criteria (2.98). The effective result is a combination of expected features (though not exactly a weighted average, since the operator of reduction is not linear).

The main weakness of the approach is the dependence on correct understanding of the feature of interest. We find that even structurally simple and small features of interest cannot be neatly amplified without some (trial-and-error) inspection of sensitivities and correlations with the rest of the model. For example, direct event targeting directed at the species of interest listed in (7.58) resulted in a generally worse performance in comparison with unmodified reduction setup. In the extreme case, if the applied importance analysis identifies the whole model as a feature of interest, the approach is not effective.

■

## Selective model reduction

*Selective model reduction* is a novel idea based on an understanding that model reduction is essentially projection. If the full model has features of interest that are best preserved under distinct projections, then a reduced order space should be defined using a combination of projections. We called the approach intrusive, because it attempts to modify the reduced model performance by means other than weighting, and the resulting subspace basis is not optimal in the sense of (2.21), or any obvious form of (2.98).

Since projections are defined in the model state space, without using any information about time, selective model reduction is a form of distinct treatment of the model state components. For completeness, we also suggested a form of selective model reduction by time interval, but it is not efficient for the current problem because of the computational expense associated to the required integration restarts (2.91) and patching by full model dynamics (2.92).

Selective model reduction is a way to reconcile the need to represent the fast transient manifold with the need to exclude it from the snapshot set. The additional computational expense consists of solving an extra eigenvalue problem (2.20).

We create two versions of the subspace basis. The first set of eigenvectors  $\Phi^I$  corresponds to the covariance matrix  $(U_o^I)(U_o^I)^T$  where  $U_o^I$  consists of uniformly based snapshots. The second set  $\Phi^{II}$  is based on the set  $U_o^{II}$  of snapshots uniformly placed on the intervals  $0 \leq t_{local} \leq 5$ ,  $10 \leq t_{local} \leq 18$ ,  $19 \leq t_{local} \leq 24$ . The

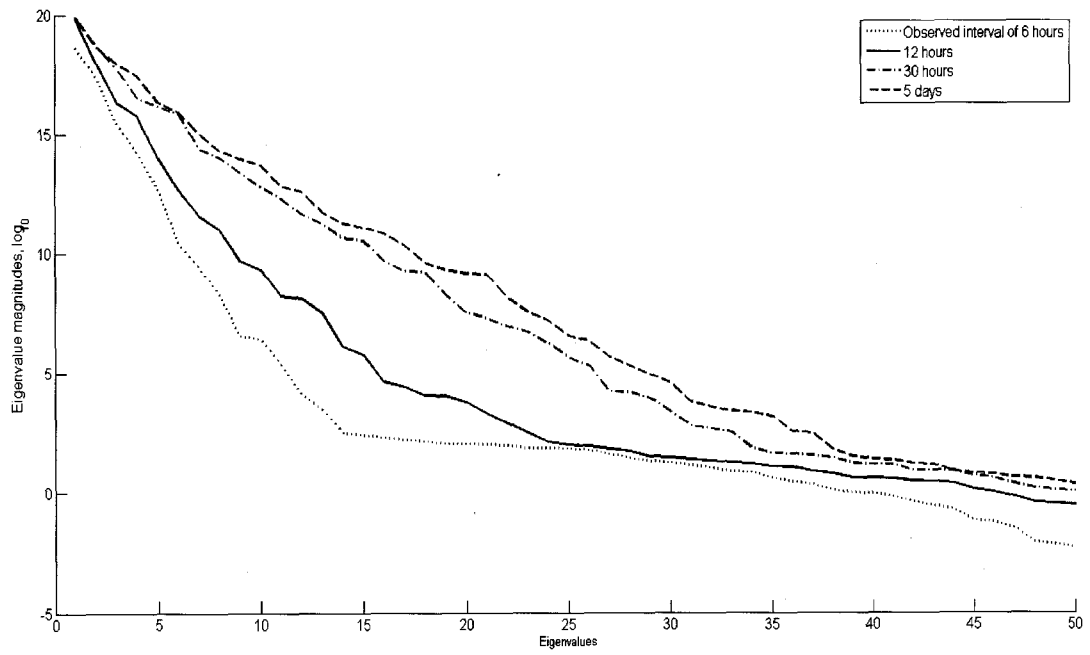
two versions of the eigenvalues and eigenvectors are slightly different (approximately 10% component-wise for the first 5 eigenvectors).

The choice of species for selective treatment is subjective; it depends on our correct understanding of the features of interest to be amplified. We apply the combined projection (2.87) as follows: a row for the matrix  $\Phi^C$  is taken from  $\Phi^I$  for the 21 species without the label *fast* in Table 8.2; from  $\Phi^{II}$  for the rest of the species. The resulting matrix is then normalized to  $\Phi^N$ .

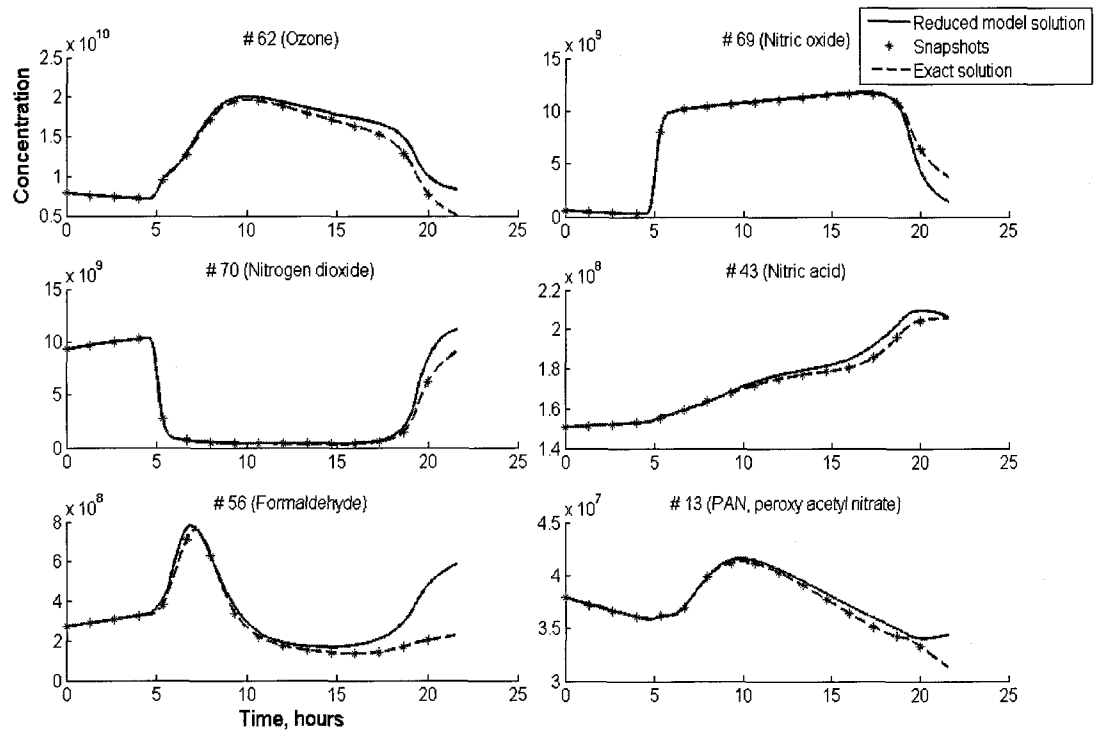
The snapshot sets  $U_o^I$ ,  $U_o^{II}$  were taken, correspondingly, from the unmodified setup (Figure 7.23), and the setup with omitted fast manifold (Figure 7.25). The resulting performance is visualized in Figure 7.27. We observe an improvement in comparison with an unmodified setup; the performance is slightly worse than in the slow manifold targeting setup. An additional attractive feature is that the obtained reduced model is relatively more stable, and can be integrated over longer time intervals. This latter observation gives us an important reason to dampen the influence of the fast manifold).

In our experience, selective model reduction works best when selective treatment is required for large parts of the model. In the scope of this section, we have not achieved direct amplification of arbitrary elements of the model behavior. The representation of features of interest such as (7.58) is easy to setup, but for the consistent, observed difference in performance, selective treatment should be also applied to the species that are strongly correlated to the ones listed in (7.58).

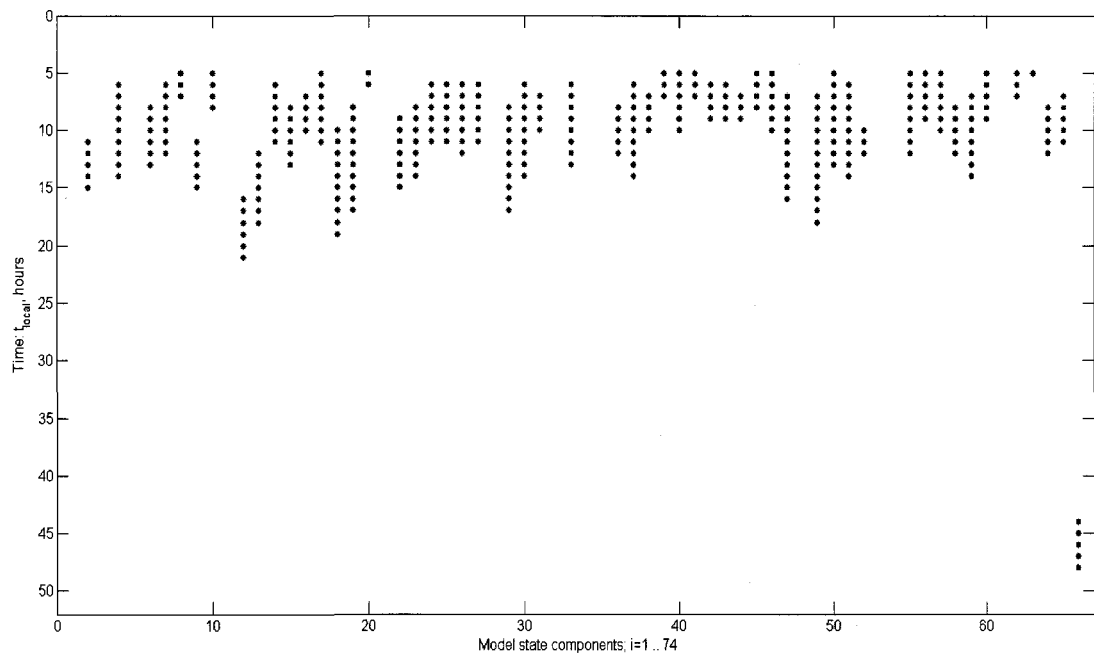
■



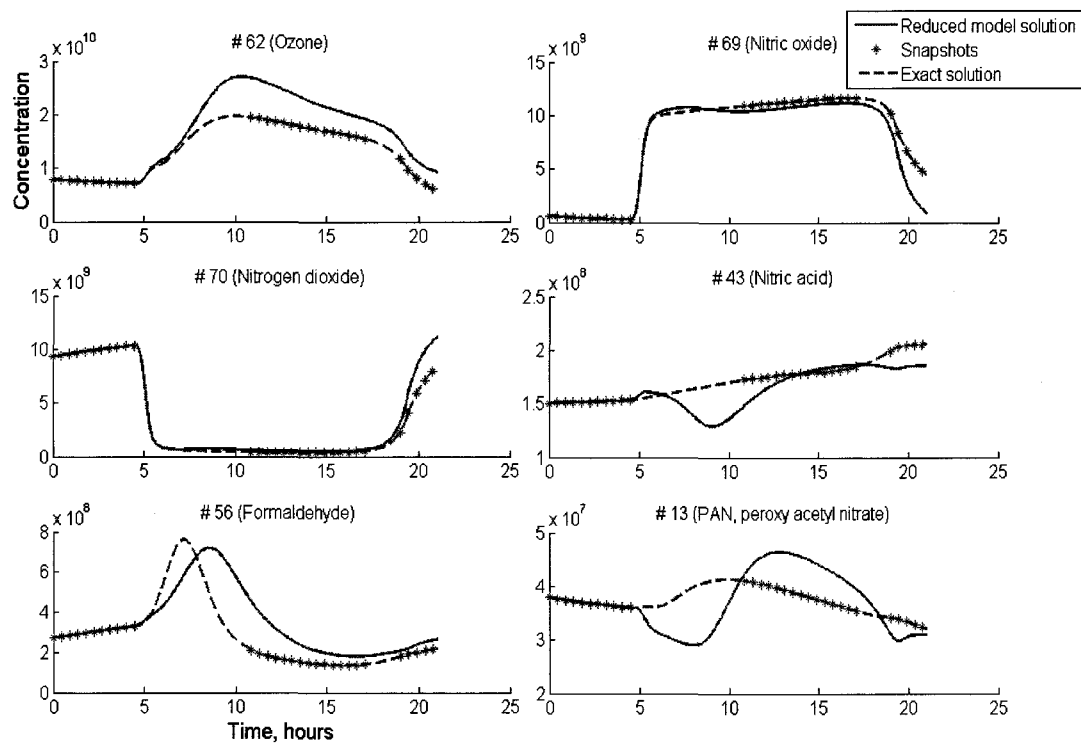
**Figure 7.22 First 50 eigenvalues of the SAPRC99 covariance matrix.**



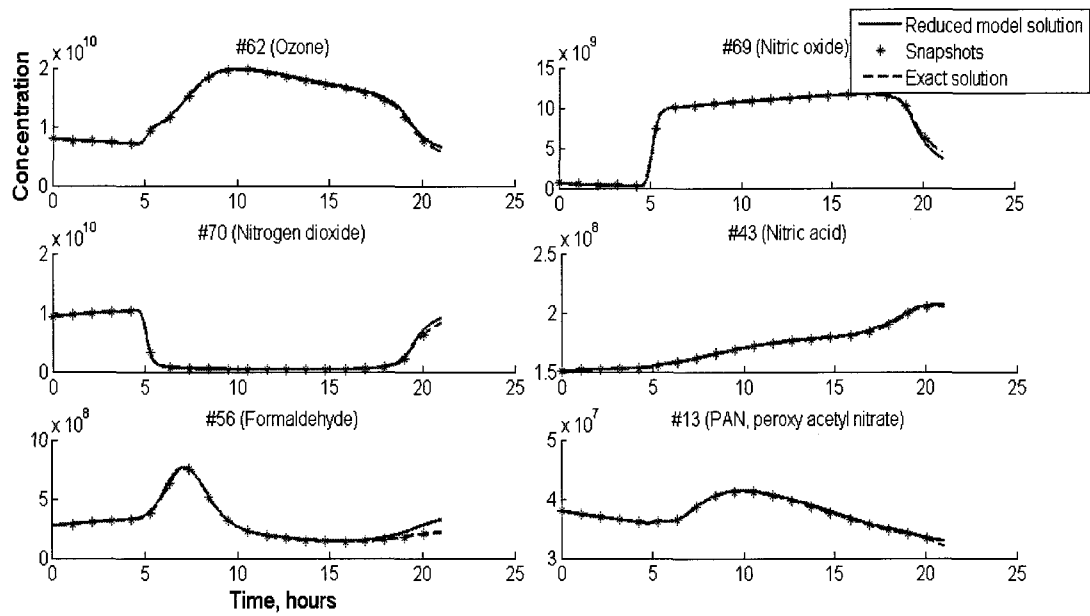
**Figure 7.23 Performance of the reduced model solution for SAPRC99 model: unmodified reduction setup.**



**Figure 7.24 Fast manifold of the SAPRC99 model: distribution over time and components.**

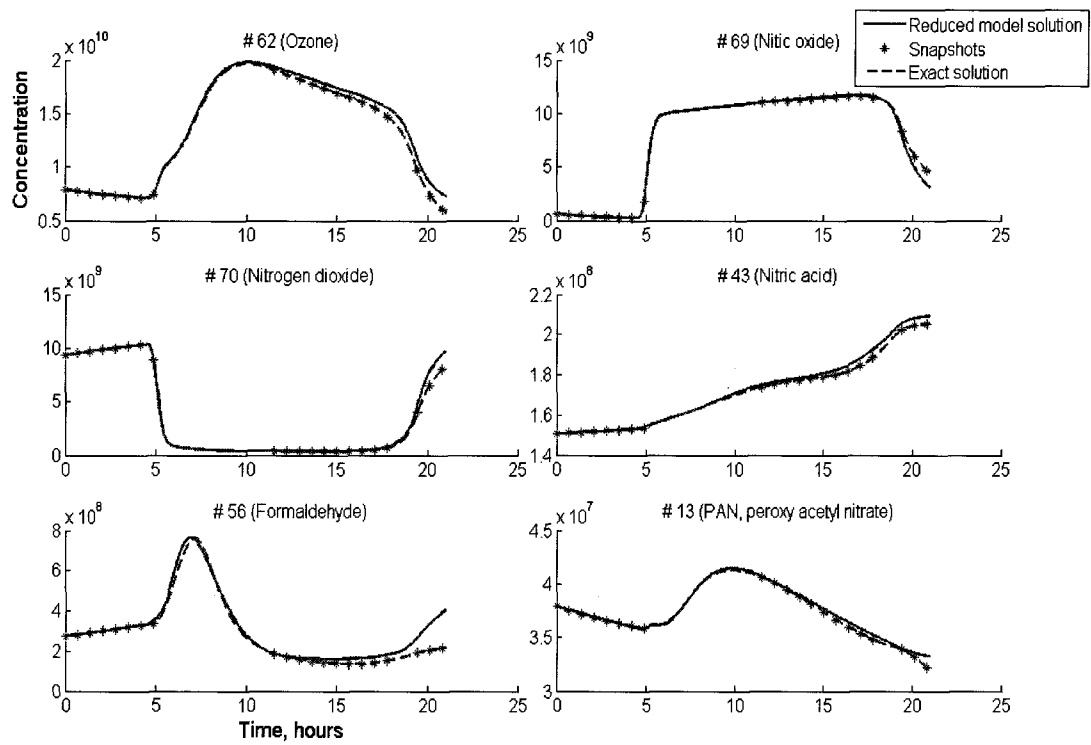


**Figure 7.25 Performance of the reduced model solution for SAPRC99 model: no snapshots during transient intervals**



**Figure 7.26 Performance of the reduced model solution for SAPRC99 model: slow manifold targeting**





**Figure 7.27 Performance of the reduced model solution for SAPRC99 model: selective model reduction.**

### 7.7.3 MEASUREMENT OF THE REDUCED MODEL PERFORMANCE

In this final section, we briefly discuss the quality of the a reduced model and collect a number of unsophisticated measurements that characterize the performance of two of our versions of the reduced SAPRC-99 (unmodified setup and slow manifold targeting setup). Formal analysis suggested in Chapters 2 and 3 addresses the issue, but the associated computational and code development cost can be very high. If the data is needed repeatedly, it should be obtained by inspection.

The quality of the reduced model should be understood as a very general concept. Summarizing the remarks in the previous material, we can say that it has four separate aspects. As a first definition, we used an unambiguous, but limited ‘correct reproduction of a feature of interest’. For some problems, that is all that is required to understand whether the reduced model accomplishes its goals. However, for large-dimensional problems in particular, we cannot readily choose an output function  $\mathfrak{F}$  to summarize all the model data.

Second, we compared the geometric shapes of the full and the reduced model solutions. This kind of description provides good understanding of how much the reduced model preserves the explicit full model dynamics, but is too informal to be used in the (eventual) automatic construction of reduced models.

Third, we looked at the evolution of relative error over time, and distribution of the component-wise error. We recommend looking at the distribution of error (7.1) for several different values of reduced model dimension  $k$ , to verify that the reduction setup uses an adequate dimension, and to see how the improvements on

the reduction process compare against switching to a reduced model of slightly higher dimension.

For the distribution of error (7.2) over components and time instances, the most effective metric is the variability of the error (we measure the standard deviation). A reduced model solution with an almost unchanging relative error reproduces the geometric shape of the full solution well. If that is the case, fairly large error magnitudes that are due either to bias, or to occasional (artifact) solution peaks may be acceptable.

The fourth and final characteristic of the reduced model is the numerical stability of integration. If the numerical solver cannot integrate the reduced model equations, the reduction setup needs to be changed. Increase in the number of snapshots, use of longer integration interval, and, paradoxically, increase in the dimension of the reduced space may lead to instability. The latter is due to an increased difficulty of error control in the solver that deals with a large-dimensional ODE with high sensitivity of solution to perturbations in data. Increase in dimension from minimal acceptable to intermediate values of  $k$  is counter-productive in any case: if the reduction is not significant, there is no computational advantage.

For every constructed reduced model, we would like to know if it is applicable multiple times (in an iterative process, or in a single task of simulation), for integration starting from initial conditions that are not exactly known *a priori*. If the data is needed repeatedly, we recommend trial integrations over a sample of points from the parameter space  $P$ .

We shall now apply the measurements to the reduced SAPRC-99 models; the goals are to give more substance to the conclusion that a model based on the event targeting approach is an improvement over the unmodified setup; and to show which individual chemical species are well reproduced in either version.

The measurements of a relative error (7.1) for the unmodified reduction setup are visualized in Figure 7.28. The quality of the reduced model grows slowly for increasing reduced space dimensions, starting with  $k = 10$ . There is no significant difference between models in the range  $12 \leq k \leq 25$ ; the maximal error is approximately 2.5%. We note that for the values  $25 \leq k \leq 65$  the reduced equations are effectively unstable. For reduced models of very low dimension, the performance is unacceptable, with numerical instability at  $k \leq 7$ .

A corresponding measurement for the event targeting setup is visualized in Figure 7.28. The same true number of degrees of freedom,  $k = 10$ , is observed. The maximal error has decreased to approximately 1%; the performance of the model with  $k = 8$  has become more acceptable. Our experiments also show that the numerical stability properties changed: now the effective range of dimensions is  $6 \leq k \leq 20$ .

To produce the next measurements, we simulate the possible range of initial conditions by introducing random perturbations to the value  $u(t_0)$  (equilibrated, recorded in Table 8.2):

$$(\hat{u}(t_0))_i = (u(t_0))_i(1 + \delta p_i / 100\%), \quad i = 1, 2, \dots, n \quad (7.62)$$

where  $\delta p$  is a vector of  $n$  random perturbation coefficients, with prescribed maximal length (in our experiments,  $\|\delta p\|_2 = 2, 2.5, 5\%$ ). We define a relative error for the component  $i$  at time  $t$  by

$$e_i(t) = \left( \frac{(u(t))_i - (\hat{u}(t))_i}{\mu_i} \right) \cdot 100\% \quad (7.63)$$

and observe the distributions of  $e_i$  over time (each recorded measurement is an average over the distribution observed with 100 randomly chosen values for  $\delta p$ ).

The results of measurements are recorded, for two reduction setups, in Tables 8.3, 8.4; extracts of significant table data are visualized in Figures 7.30, 7.31. In Table 8.3, we record the distribution of the observed relative error in species of interest (7.58); the metrics are the minimal and maximal values, mean, and standard deviation. The measurements were also taken for the initial conditions randomly perturbed with bounds of 1%, 2%, 5%, 10%. Judging from the error magnitudes, the reduced model shows adequate performance if the perturbation is bounded by approximately 2.5%, a larger deviation allowed in some components. The event targeting setup demonstrates a small advantage (becoming less significant as perturbation bound grows). The comparison of error standard deviation for different components is visualized in Figure 7.30. Observing the variation in error alone, we can state which species do not preserve the correct solution shape:  $i = 62, 69, 70$ .

We observe that the ability of a reduced model to capture the evolution of individual species appears to be an almost invariant characteristic, dependent more

on the specie itself than on the size of perturbation, or on the reduction setup. In other words, “a specie reproduced well by reduced model’ can be used as a label. We provide more complete information in Table 8.4 and Figure 7.31. In the table, we record the distribution of the error for all 74 species, for a 2.5% perturbation bound. In the plot, we visualize the error variance for all the species, and sort the species by reliability of reproduction. Note that the behavior of species

$$u_i: i = 18,33,37,41,45,47,49,52,53,54,60,65,67,71,73 \quad (7.64)$$

is particularly difficult to reproduce correctly.

**Table 8.3 Effectiveness of reduced model for perturbed initial conditions**

Maximal perturbation, %	#	Unmodified setup, relative error distribution, %				Slow manifold targeting setup, relative error distribution, %			
		Min	Max	Mean	St. dev.	Min	Max	Mean	St. dev.
0	62	1.50	6.60	3.32	1.28	1.20	4.60	2.39	<b>0.82</b>
	69	0.19	0.71	0.50	0.14	0.10	0.49	0.34	<b>0.10</b>
	70	2.28	28.52	6.55	5.62	1.79	19.76	4.67	<b>3.80</b>
	43	0.31	2.01	0.95	0.50	0.25	1.50	0.73	<b>0.38</b>
	56	4.72	36.01	12.57	7.03	3.59	24.78	9.09	<b>4.62</b>
	13	0.24	1.20	0.68	0.30	0.21	0.92	0.54	<b>0.22</b>
2	62	-9.98	47.81	0.62	9.69	-9.99	47.45	0.61	<b>9.66</b>
	69	-49.55	14.55	-6.41	14.26	-49.55	14.57	-6.39	<b>14.25</b>
	70	-47.60	11.43	1.31	<b>5.79</b>	-47.62	11.53	1.34	5.82
	43	-0.04	22.78	9.69	7.58	-0.05	22.35	9.60	<b>7.49</b>
	56	-42.79	47.52	-7.26	<b>23.93</b>	-44.56	46.88	-7.78	24.50
	13	-5.65	8.93	0.90	4.89	-5.66	8.90	0.85	<b>4.89</b>
5	62	-3.88	58.62	29.44	23.43	-3.82	57.38	28.17	<b>23.03</b>
	69	-21.86	677.9	99.08	221.03	-21.25	661.48	99.27	<b>217.76</b>
	70	-36.95	4.44	-23.81	17.24	-36.24	4.39	-23.54	<b>16.74</b>
	43	-0.01	14.94	5.66	4.59	0.01	14.39	5.45	<b>4.37</b>
	56	-16.32	70.36	7.10	22.88	-16.89	68.56	5.47	<b>22.41</b>
	13	-4.37	3.45	-0.44	2.29	-4.62	3.40	-0.45	<b>2.29</b>
10	62	-6.01	94.11	51.26	37.82	-5.67	89.02	46.87	<b>35.64</b>
	69	-35.29	1078	603.22	486.44	-34.24	1026.1	579.45	<b>462.10</b>
	70	-57.06	11.02	-32.11	28.24	-55.70	7.74	-31.96	<b>27.20</b>
	43	0.01	26.43	9.73	8.31	0.15	28.83	8.98	<b>7.44</b>
	56	-18.67	111.2	14.76	33.60	-20.80	111.67	10.85	<b>30.62</b>
	13	-4.93	5.77	-0.14	<b>4.36</b>	-5.78	5.39	-0.60	4.45

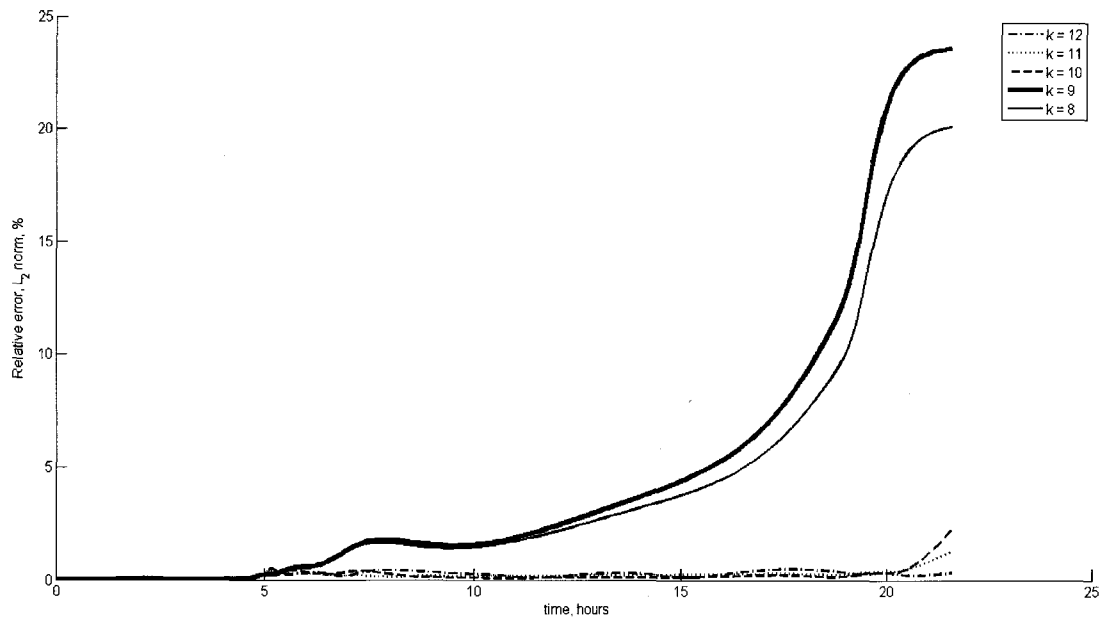
**Table 8.4 Effectiveness of reduced model: reproduction of individual species, 2.5 % perturbation**

#	Unmodified setup, relative error distribution, %				Slow manifold targeting setup, relative error distribution, %			
	Min	Max	Mean	St. dev.	Min	Max	Mean	St. dev.
1	-6.64	1.16	-1.54	2.04	-2.17	6.57	1.50	2.24
2	-1.60	-0.77	-1.28	0.19	-1.28	-0.26	-0.50	0.20
3	-1.17	-0.79	-1.03	0.09	-1.28	-1.02	-1.13	0.05
4	1.34	2.00	1.53	0.17	-0.09	0.39	0.24	0.11
5	-1.82	-1.29	-1.58	0.12	-0.23	0.11	-0.07	0.07
6	-0.23	0.43	0.03	0.15	-1.09	-0.65	-0.84	0.09
7	-2.07	5.57	0.51	1.84	-2.82	2.91	1.14	1.33
8	0.41	0.53	0.45	0.03	-0.15	-0.06	-0.09	0.02
9	-0.29	2.73	0.43	0.81	-0.58	1.98	1.23	0.60
10	-6.54	22.32	5.86	6.85	-16.82	9.46	-3.82	6.00
11	-0.66	13.95	3.06	4.12	-11.17	0.54	-2.36	3.01
12	-1.65	35.49	9.65	8.17	-24.43	16.17	-3.82	9.00
13	-0.73	2.33	0.43	0.77	-1.58	0.74	-0.15	0.57
14	-1.51	41.18	10.98	11.50	-34.00	4.90	-7.65	8.65
15	-0.74	29.47	8.35	7.77	-23.87	4.11	-5.70	5.90
16	-9.45	3.88	-1.45	3.39	-5.62	13.94	4.08	3.50
17	-4.14	45.39	16.50	12.45	-40.21	-1.21	-17.18	8.35
18	-7.30	48.76	14.60	14.86	-49.19	45.33	-9.51	16.75
19	-10.72	-1.77	-5.54	1.93	-6.46	10.61	5.21	2.89
20	-0.26	24.45	6.51	7.20	-21.56	-1.84	-6.85	4.97
21	-1.03	8.41	3.44	1.81	-6.50	0.69	-2.82	1.77
22	0.02	49.40	22.71	10.90	-46.11	-0.07	-28.32	8.51
23	-12.01	-0.50	-4.88	2.68	-4.32	14.20	6.30	4.12
24	-30.06	0.82	-10.44	6.22	-13.70	22.27	10.01	6.70
25	-15.83	2.32	-5.20	3.52	-3.18	13.35	7.09	2.67
26	-18.71	49.83	17.63	11.78	-48.52	0.04	-22.75	9.88
27	-0.01	45.57	15.48	10.62	-37.33	6.24	-17.79	9.81
28	-12.54	48.72	8.51	7.59	-21.53	48.14	-6.96	8.87
29	-0.27	25.56	6.68	7.51	-22.37	-0.21	-5.91	5.68
30	-6.59	0.56	-2.31	1.35	-3.24	7.50	3.44	2.42
31	-8.65	0.44	-3.17	1.87	-5.45	10.82	4.38	3.05
32	-42.96	48.62	-5.46	12.29	-45.08	48.69	10.25	9.95
33	-49.25	49.60	-14.54	14.76	-38.63	49.21	17.86	12.61
34	-49.40	9.86	-4.27	7.70	-38.48	45.22	5.68	11.46
35	-44.32	37.93	1.63	12.84	-33.60	49.29	3.74	9.53
36	0.51	3.20	1.22	0.64	-0.34	2.22	1.02	0.48
37	-0.03	49.84	13.07	10.64	-39.49	10.87	-14.80	10.94
38	-10.65	16.23	2.88	5.02	-9.22	29.63	3.83	9.44
39	-24.50	1.89	-8.38	5.20	-11.75	18.00	8.90	5.56
40	0.01	6.32	1.51	1.77	-4.74	0.01	-1.16	1.25
41	-17.22	48.23	14.28	13.39	-41.06	48.11	-5.49	20.49
42	-1.30	12.24	2.24	3.78	-9.91	0.69	-2.14	2.63
43	-0.73	3.02	0.22	0.89	-1.33	1.10	0.05	0.54
44	-13.13	-0.46	-5.10	2.90	-4.77	14.45	6.52	4.32

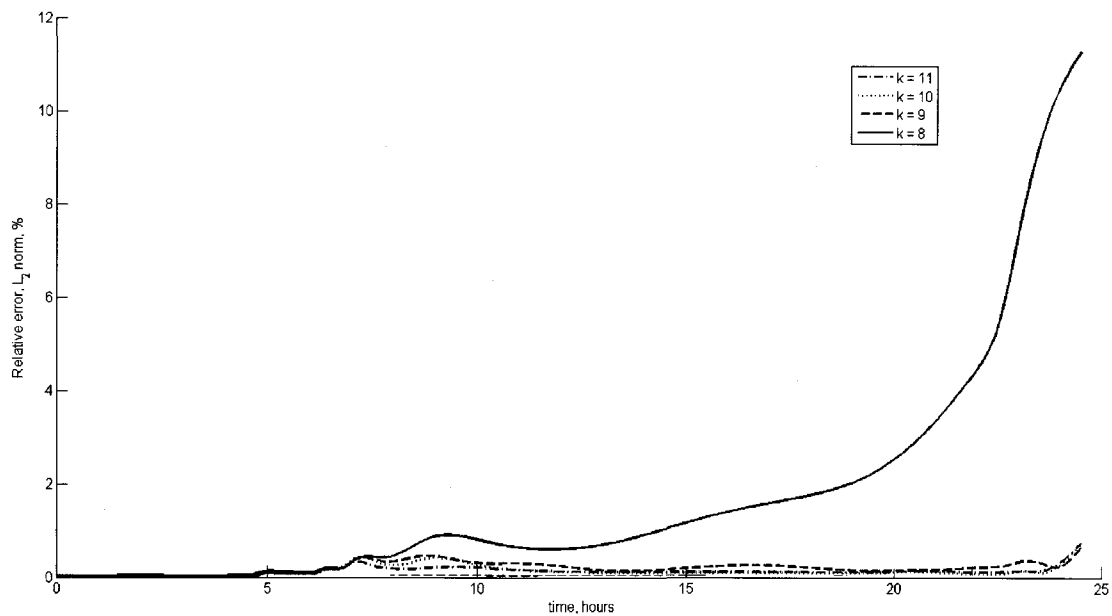


**Table 8.4 (continued):**

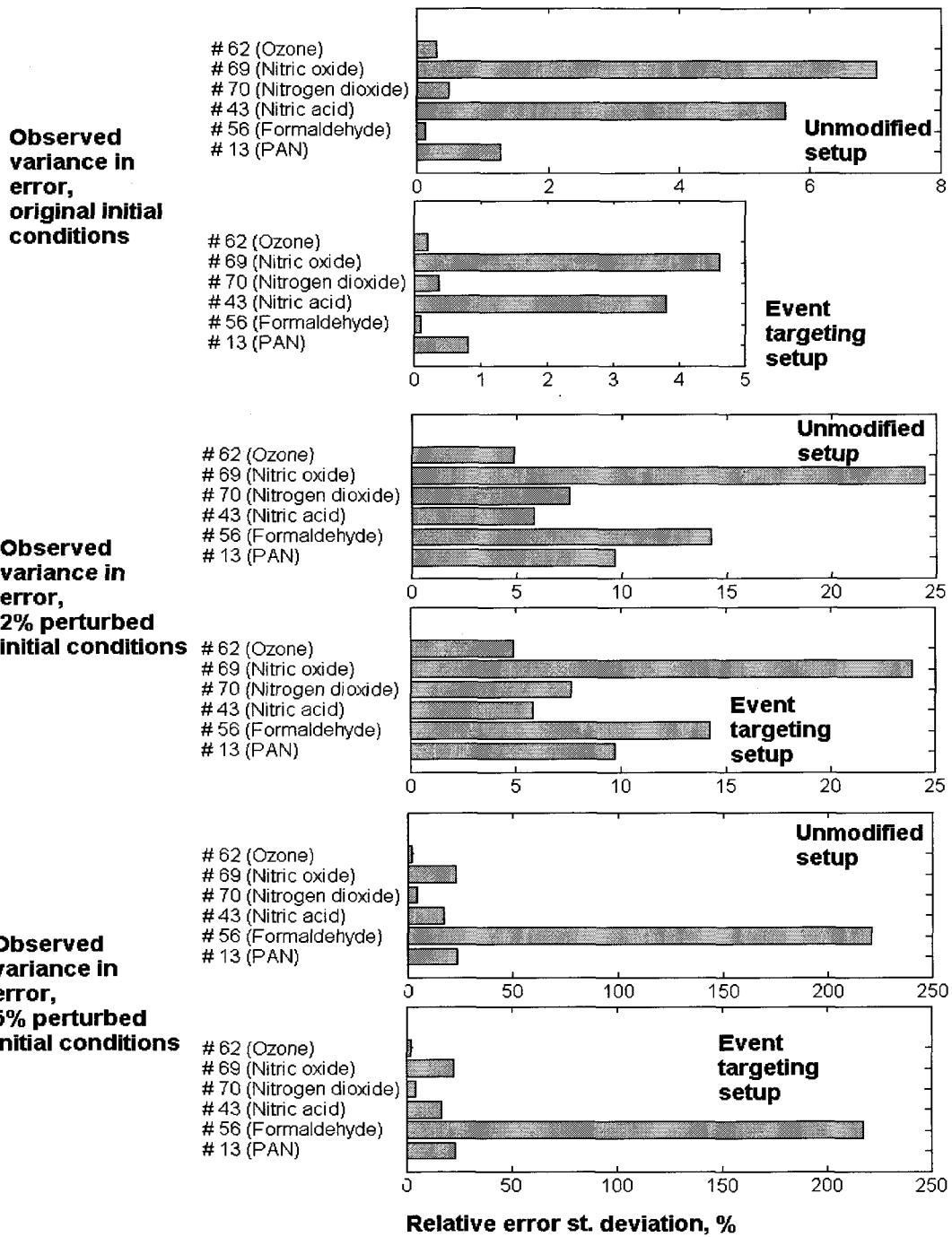
#	Unmodified setup, relative error distribution, %				Slow manifold targeting setup, relative error distribution, %			
	Min	Max	Mean	St. dev.	Min	Max	Mean	St. dev.
45	-0.55	48.21	14.06	11.20	-39.41	19.16	-8.11	12.66
46	-0.21	49.40	16.68	11.36	-38.50	19.13	-12.49	9.91
47	-9.64	49.86	15.99	15.29	-49.47	49.71	-15.03	12.15
48	-5.73	36.75	13.03	8.82	-32.14	0.66	-13.00	6.17
49	-13.28	49.24	14.56	14.68	-49.51	0	-16.32	11.58
50	-26.22	9.15	-6.85	5.53	-29.64	27.36	5.72	8.17
51	-28.48	5.91	-7.23	5.01	-27.80	24.90	5.84	7.86
52	-49.53	4.35	-18.96	13.34	-24.30	49.84	19.12	14.63
53	-29.57	49.45	11.61	13.27	-38.85	23.57	-10.88	11.65
54	-49.56	19.61	-20.34	15.51	-18.61	49.62	18.98	12.22
55	0.54	36.19	9.94	8.52	-26.82	6.57	-7.67	7.06
56	-0.29	19.54	6.15	4.90	-16.88	2.43	-6.42	3.91
57	-1.89	30.58	12.67	7.30	-28.57	1.12	-13.56	6.83
58	-3.85	15.14	4.86	4.89	-13.55	3.35	-3.94	4.01
59	-5.01	26.91	7.73	6.30	-19.63	15.23	-2.30	7.80
60	-43.25	49.71	16.66	13.52	-45.61	0.74	-15.77	10.26
61	-12.68	29.33	1.95	6.69	-8.13	20.47	3.61	5.69
62	-0.10	13.14	3.71	3.63	-11.76	0.38	-2.99	3.05
63	-1.06	30.60	11.04	7.62	-28.57	-2.16	-11.19	4.99
64	-17.68	25.47	10.84	6.52	-30.88	1.24	-11.80	4.58
65	-49.60	7.07	-8.57	13.31	-42.05	49.79	18.19	20.47
66	1.13	45.58	16.43	10.66	-41.49	-4.12	-16.88	7.33
67	-0.31	48.59	14.15	11.23	-46.20	26.91	-11.06	12.61
68	-15.83	32.70	12.17	8.10	-29.38	3.34	-11.91	5.80
69	-1.23	1.57	-0.08	0.70	-0.86	1.28	0.41	0.54
70	-0.61	2.58	0.54	0.69	-2.32	0.78	-0.70	0.56
71	-27.85	49.14	18.76	20.11	-48.91	32.10	-15.83	15.10
72	1.40	49.29	17.60	11.48	-44.69	-4.18	-17.85	8.05
73	-43.15	49.94	19.24	12.19	-49.84	1.41	-18.66	10.02
74	-2.75	3.44	1.03	1.49	-5.12	3.83	-2.07	1.86



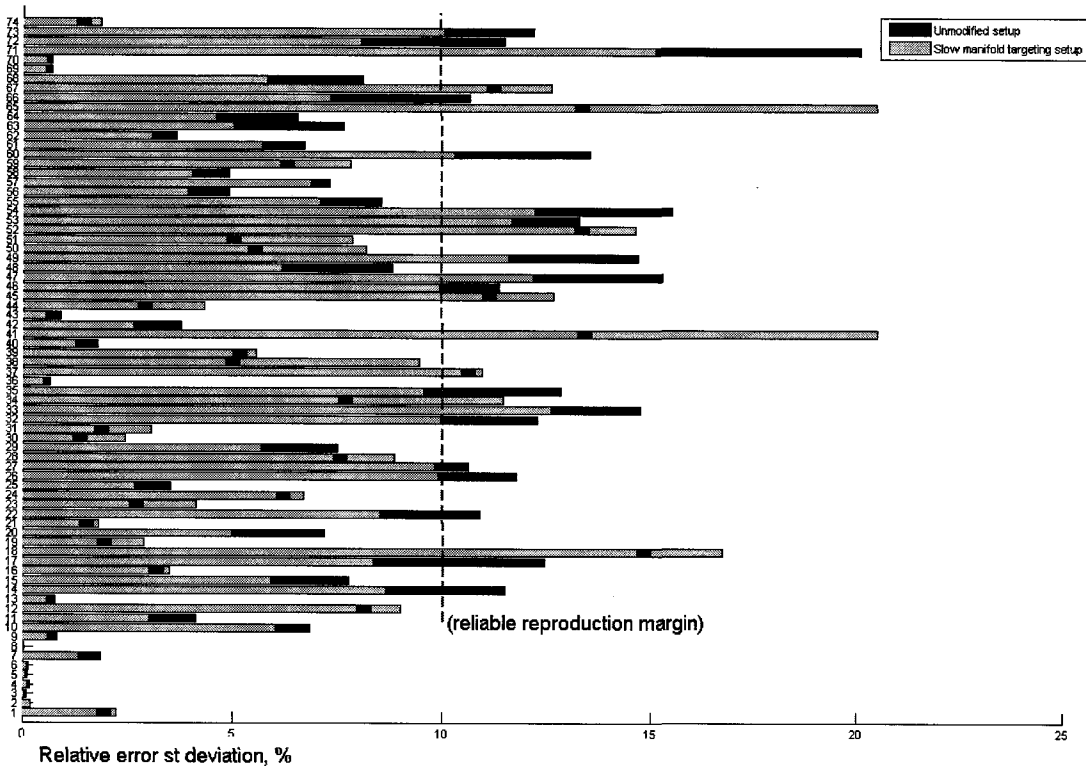
**Figure 7.28. Relative error for the reduced SAPRC-99 model for different dimensions.**



**Figure 7.29** Relative error for the reduced (slow manifold targeting) SAPRC-99 model for different dimensions.



**Figure 7.30 Effectiveness of reduced model for perturbed initial conditions: error variability in reproduction of species of interest.**



Chemical species, sorted by reliability of reproduction:

*(unreliably reproduced species in the end of the list)*

Unmodified setup:

8 3 5 6 4 2 36 70 69 13 9 43 30 74 40 21 7 31 19 1 23 44 16 25 62  
 42 11 58 56 51 38 39 50 24 59 64 61 10 20 57 29 28 63 34 15 68 12  
 55 48 27 37 66 22 45 67 46 72 14 26 73 32 17 35 53 65 52 41 60 49  
 33 18 47 54 71

Slow manifold

Targeting setup:

8 3 5 6 4 2 36 43 69 70 13 9 40 7 21 74 1 30 42 25 19 11 31 62 16  
 56 58 23 44 64 20 63 39 29 61 68 15 10 48 24 57 55 66 59 51 72 50  
 17 22 14 28 12 38 35 27 26 46 32 73 60 37 34 49 53 47 54 33 67 45  
 52 71 18 65 41

**Figure 7.31 Effectiveness of reduced model: error variability in reproduction of individual species; 2.5% perturbation.**

## CHAPTER 8

### CONCLUSIONS

In our work, we have explained how POD-based model reduction techniques can be enhanced for better performance of reduced models, and then used to replace the full models in the tasks of simulation, iterative optimization (and, by implication, control). The general goal was to investigate and validate POD-based reduction as a flexible and efficient choice, particularly appropriate for reaction-transport models of atmospheric chemistry. The specific goal was to demonstrate that the use of the reduced model can improve efficiency of iterative optimization procedures.

We can now identify the relationship of our work to other studies in the field. To modify the performance of the reduced model, we regularly used weighting of snapshots and the change of metric (dual weighting) [34], [35]; and also goal-oriented snapshot placement guided, in our case, by results of model factor importance analysis: [1], [68], [71], [88], [76]. We have shown how dual weighting can be used to target specific events in the model behavior. For the problems of iterative optimization, we used descent methods [10], [13], [37]; with adjoint differentiation of the model to obtain the gradients: [22], [24], [69].

We introduced new ideas of selective model reduction (taking into account slow-fast dynamics, or other chemical factor importance analysis considerations [80]); sensitivity information by interpolation [91], [92]; sensitivity information by differentiating the model reduction process (based on [80]); data rejection and

recovery in long-term integration (using POD-based reduction and Kriging): [5], [36], [44], [108]. We noted the availability of *a posteriori* error estimated for model reduction: [55], [58]; and also of conjugate gradient methods for iterative optimization [48].

Due to limitations on development effort, or inappropriateness for our specific tasks, we did not use Hessian-based model reduction: [8], [9]; formally optimal placement of snapshots [19]; *a priori* choice of metric: [33], [38]; POD-based reduction combined with such linear model reduction methods as balanced truncation, or empirical Grammians: [2], [3], [14], [52], [103]; preservation of symmetries in the reduced model [6]; additional empirical techniques such as acceleration of POD reduction [4]. In principle, this additional knowledge should not be rejected, and can lead to further improvements in the field of study.

We have illustrated our suggestions with models taken from multiple sources: [31], [66], [128], [133]; note in particular [76] and [134] for Lorenz model and SAPRC-99. In general, our work is applicable in many of the usual tasks associated with large models, including prediction of future behavior, recovery of true state of the system based on incomplete observations; inverse problems of simulation and control; data assimilation with filtering. The developed techniques will see further use in the area of uncertainty quantification for models dependent on large numbers of parameters: [91], [93].

In our study, the main example of the model was the advection-diffusion-reaction system, discretized to a sparse ODE. The covariance matrix of such model

had a characteristic distribution with only a few large eigenvalues. We used an understanding that capturing most of the eigenvalue energy is sufficient to reproduce most of the full model behavior. That is empirically true for systems dominated by an elliptic operator of the reaction-only system where parabolic operators of diffusion and advection transport produced a less rapidly decreasing sequence of eigenvalues, but had higher sparsity and lower importance. For a more complete understanding of model reduction, the studied model should be made more general, ideally, with a formally described relationship between the spectrum of the arbitrary differential operator and the performance of the reduced model.

We have reviewed one basic aspect of data assimilation: search for the solution of the initial conditions optimization problem. It turned out that the use of the reduced model is computationally efficient, provided the reduced model is constructed using snapshots that are sufficiently close to the optimizer (then the optimization problem is almost convex, and the reduced model evolution is almost equivalent to the full model evolution). We suggested a number of *a posteriori* measurements of the reduced model performance, but, again, it would be better to have a formal statement characterizing the problem, and the search step at which it is efficient to use the reduced model.

The question of preserving such properties of the full model as solution symmetries, positivity, conservation of physical quantities in the reduced model is largely unanswered (there are negative examples). We can define any such property as a feature of interest (or a combination of features of interest), and amplify its



presence in the reduced model. The procedure, however, is not automatic; it needs to be guided using results of computationally expensive factor importance analysis.

Overall, the main weakness of model reduction by projection is the lack of *a priori* quality estimation. For the more complex examples, the reduced model needs to be tuned (manually or partially manually, based on the problem-specific knowledge) for adequate performance. We have attempted to contain this weakness to only the process of constructing the reduced model. The presence of such difficulties as discretization errors, unreliable data, high sensitivity to small changes leading to incorrect factor importance analysis, numerical instabilities, high computational cost of linear algebra operations, etc, differs from problem to problem. Even though the previously available and the newly developed tools can deal with each difficulty separately, the model reduction may still be ineffective due to a combination of factors.

Once the reduced model is created, it can be used efficiently in many tasks that require multiple (forward and adjoint) evaluations of the full model, even for the full model factor importance analysis by large-scale sampling. Informally, we say that the *creation* of the reduced model is still mostly art, guided by trial-and-error and observational intuition (see [46]). The *use* of the reduced model is already closer to science. We would like to view the current work as a modest effort to improve on the current practices in reduction of nonlinear models and to shift the balance towards scientifically justified practice.

## REFERENCES

1. Alonso A.A., Keverkidis I.G., Banga J.R. and Frouzakis C.E. Optimal sensor location and reduced order observer design for distributed process systems, *Computers and Chemical Engineering*, Vol 28, 1, pp. 27-35(1), Jan 2004.
2. Antoulas A.C. *Approximation of large-scale dynamical systems*, SIAM, 2006.
3. Antoulas A.C. and Sorensen D.C. *Projection methods for balanced model reduction (Technical report)*, 1995.
4. Astrid P. *Reduction of process simulation models: a proper orthogonal decomposition approach (Thesis)*, Technische Universiteit Eindhoven, 2004.
5. Astrid P., Weiland S., Willcox K and Backx T. *Missing point estimation in models described by proper orthogonal decomposition (Proceedings)*, 43rd IEEE Conference on Decision and Control, Vol 5, 2004.
6. Aubry N., Lian W. and Titi E.S. *Preserving symmetries in the proper orthogonal decomposition*. *SIAM Journal of Scientific Computing*, 14, 2, pp. 483-505, Mar 1993.
7. Barrie L.A., Burrows J.P., Monks P., Nickovic S. and Borrel P. , *Chemical data assimilation for the observation of the Earth's atmosphere (Technical report)*, ACCENT, 2006.

8. Bashir O. Hessian-based model reduction with applications to initial-conditions inverse problems, (Thesis), Massachusetts Institute of Technology, 2007.
9. Bashir O., Willcox K., Ghattas O., Waanders B. and Hill J. Hessian-based model reduction for large-scale systems with initial condition inputs, *International Journal for Numerical Methods in Engineering*, 2007.
10. Bertsekas D.P. *Constrained optimization and Lagrange multiplier methods*, Academic Press, New York, 1992.
11. Bertsekas D.P. *Nonlinear programming*, Athena Scientific, 1995.
12. Bertsekas D. P. and Yu H. Projected equation methods for approximate solution of large linear systems, *Journal of Computational and Applied Mathematics* (submitted), 2007.
13. Biegler L.T., Ghattas O., Heinkenschloss M. and Waanders B. *Large-scale PDE-constrained optimization*, Springer, 2003.
14. Borggaard, J. Optimal reduced-order modeling for nonlinear distributed parameter systems (Proceedings), 2006 American Control Conference, 2006.
15. Borggaard J. and Burns J. A PDE sensitivity equation method for optimal aerodynamic design, *Journal of Computational Physics*, Vol 136 , 2 pp366-384, 1997.
16. Bryson A.E. and Yu-Chi H. *Applied optimal control*, John Wiley & Sons, 1979.

17. Bui T. T., Damodaran M. and Willcox K. Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics, Processings of the 15th AIAA Computational Fluid Dynamics Conference, pp 2003-4213, 2003.
18. Bui T. T., Damodaran M. and Willcox K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, AIAA Journal, Vol 42, 8 pp 1505-1516, 2004.
19. Bui T. T., Willcox K., Ghattas O. Waanders B. Goal-oriented, model-constrained optimization for reduction of large-scale systems, Journal of Computational Physics, Vol 224, 2, 2007.
20. Butcher J.C. Numerical methods for ordinary differential equations, John Wiley & Sons, 2007.
21. Cao Y, Jiang Z., Navon I.M and Zhedong L. A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition, International Journal of Numerical Methods in Fluids, vol. 53, 10, pp 1571-1583, 2007
22. Cao Y, Petzold L. and Radu S. Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solutions, SIAM Journal on Scientific Computing, 24(3), pp. 1076-1089, 2002.
23. Castaings W., Dartus D., LeDimet F.X. and Saulnier G.M. Sensitivity analysis and parameter estimation for the distributed modeling of

- infiltration excess overland flow, Hydrology and Earth System Sciences Discussions, Vol 4, 1, pp 363-405, 2007.
24. Celia M.A., Russell T.F., Herrera I., and Ewing R.E. An Eulerian-Lagrangian localized adjoint method for the advection-diffusion transport equation., Advances in Water Resources, 13, pp 187-206, 1990.
  25. Chai T. and Carmichael G. Four-dimensional data assimilation experiments with International Consortium of Atmospheric Research on transport and transformation ozone measurements. Journal of Geophysical Research 122, pp 15-33, 2007.
  26. Chai T., Carmichael G., Sandu A., Tang Y. and Daescu D.N. Chemical data assimilation of transport and chemical evolution over the Pacific (TRACE-P) aircraft measurements. Journal of Geophysical Research, Vol 111, 2006.
  27. Chai T., Carmichael G., Sandu A., Tang Y., Constantinescu E., Tianfeng C. and Daescu D.N. Predicting air quality: improvements through advanced methods to integrate models and measurements. Journal of Geophysical Research, Vol 227, pp. 3540-3571, 2008
  28. Chen Y. and McInroy J.E. Estimation of symmetric positive-definite matrices from imperfect measurements, IEEE Transactions on Automatic Control, Vol 47, 10, p1721, 2002.

29. Christensen E.A., Brons M. and Sorensen J.K. Evaluation of POD-based techniques applied to parameter dependent nonturbulent flows, *SIAM Journal of Scientific Computing*, 21(4), pp. 1419-1434, 2000.
30. Conn A.R., Gould N. and Toint P.L. Trust region methods (MPS/SIAM series on optimization), SIAM, 2002.
31. Conner G.R. and Grant C.P. Asymptotics of blow-up for a convection-diffusion equation with conservation, *Differential Integral Equations*, Vol 9 pp 719-728, 1996.
32. Covey C. and Wehner M.F. Precipitation-climate sensitivity to initial conditions in an atmospheric general circulation model (Technical report), Lawrence Livermore National Laboratory, 1997.
33. Crommelin D.T. and Majda A.J. Strategies for model reduction: comparing different optimal bases, *Journal of Atmospheric Science*, Vol 61, 17 pp 2206-2217, 2004.
34. Daescu D.N and Navon I.M. A dual-weighted approach to order reduction in 4D-variational data assimilation, accepted for publication in *Monthly Weather Review*, 2007.
35. Daescu D.N. and Navon I.M. Efficiency of a POD-based reduced second-order adjoint model in 4D-variational data assimilation, *International Journal of Numerical Methods in Fluids*, Vol 53, 6, pp 985–1004, 2007
36. Everson R. and Sirovich L. Karhunen-Loueve procedure for gappy data, The Rockefeller University, New York, 1994.

37. Fletcher R. Practical methods of optimization, John Wiley & Sons, 1981.
38. Fodor I.K. A survey of dimension reduction techniques (Technical report), Lawrence Livermore National Laboratory, 2002.
39. Gasinski L. and Papageorgiou N.S. Nonlinear analysis. Chapman & Hall, 2006.
40. Ghosh D., Avery P. and Farhat C. Uncertainty quantification of large-scale systems using domain decomposition. (Proceedings), 9th U.S. National Congress on Computational Mechanics. 2007.
41. Gianessi F. Constrained optimization and image space analysis, Springer, 2005
42. Golub G.H. Matrix Computations, John Hopkins Studies in the Mathematical Sciences, 1996.
43. Grepl M.A., Maday Y., Nguyen N.C. and Patera A.T. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, Mathematical Modeling and Numerical Analysis, Vol 41 ,3 ,pp 575-605, 2007.
44. Gunes H., Sirisup S. and Karniadakis G. E. Gappy data: to Krig or not to Krig? Journal of Computational Physics, Vol 212, 1, pp 358-382, 2005.
45. Gunzburger M.D. Reduced-order modeling, data compression and the design of experiments (Presentation), Second DOE Workshop of Multiscale Mathematics, 2004.

46. Gunzburger M.D. Perspectives in flow control and optimization, SIAM, 2002.
47. Hart D., Goodyer C.E., Berzins B., Jimak P.K. and Scales, L. Adjoint error estimation and spatial adaptivity for EHL-like models, IUTAM Symposium on Elastohydrodynamics and Micro-Elastohydrodynamics, pp. 47-58, 2006.
48. Hestenes M. Conjugate direction methods in optimization, Springer-Verlag, 1980
49. Hinze M. and Volkwein S. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control, Lecture Notes in Computational Science and Engineering, Vol 45, 2005.
50. Hinze M. and Volkwein S. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition, Computational Optimization and Applications, Vol 39, pp 319-345, 2007.
51. Hodel A.S., Tenison R.B. and Poola K. Numerical solution of large Lyapunov equations by approximate power iteration, Linear Algebra Applications, 236: pp. 205-230, 1996.
52. Hooimeijer A.A. Reduction of complex computational models (Thesis), Technische Universiteit Delft, 2001.



53. Homescu C and Navon I.M. Model reduction of large-scale systems: an overview of data assimilation techniques in meteorology, SIAM Conference on Control and Annual Meeting, 2001.
54. Homescu C., Petzold L. and Radu S. The effect of problem perturbations on nonlinear dynamical systems and their reduced order models, accepted for publication in SIAM Journal of Scientific Computing, 2007.
55. Homescu C., Petzold L. and Radu S. Error estimation for reduced order models of dynamical systems, SIAM Journal on Scientific Computing, 43: pp. 1693-1714, 2005.
56. Hundsdorfer W., Koren B., VanLoon M. and Verwer J.G. A positive finite-difference advection scheme, Journal of Computational Physics, Vol 117 , 1, pp 35-46, 1995.
57. Jacobi S.L.S., Kowalik J.S. and Pizzo J.T. Iterative methods for nonlinear optimization problems, Prentice-Hall, 1972.
58. Jeannerod C.P., Visconti J. Global error estimation for index-1 and index-2 DAEs. Numerical Algorithms, Vol 19, 1 pp 111-125, 1997.
59. Kalnay E. Atmospheric modeling, data assimilation and predictability, Cambridge, 2003.
60. Kenney C. and Laub A. Small-sample statistical condition estimates for general matrix functions, SIAM Journal of Scientific Computing, 15,1, pp. 36-61, Jan 1994.

61. Kim N. H. and Wang Haoyu. Adaptive reduction of random variables using global sensitivity in reliability-based optimization. *Int. J. Reliability and Safety*, Vol.1, 1-2, 2006.
62. Kiwiel K.C. and Murty K. Convergence of the steepest descent method for minimizing quasiconvex functions, *Journal of Optimization Theory and Applications (Technical note)* Vol 89, 1, pp 221-226, 1996.
63. Kragel B. Streamline diffusion POD models in optimization (Thesis), Universitat Trier, 2005.
64. Krizek M., Neittaanmaki P., Glowinski R. and Koroov S. Conjugate gradient algorithms and finite element methods, Springer, 2004.
65. Kunisch K and Volkwein S. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition, *Journal of Optimization Theory and Applications*, Vol 102 , 2 pp 345-371, 1999.
66. Kuznetsov S.P., Mosekilde E., Dewel G. and Borckmans P. Absolute and convective instabilities in a one-dimensional Brusselator flow model, *Journal of Chemical Physics*, Vol 106 , p 7609. 1997.
67. Lawless A.S., Nichols N.K., Boess C. and Bunse-Gerstner A. Using model reduction methods within incremental four-dimensional variational data assimilation, *Monthly Weather Review*, Vol 136, 4 pp 1511-1522, 2008
68. LeCadre J.P. and Ravazzola P. Model reduction and wideband analysis [underwater acoustic application], *International Conference on Acoustics, Speech and Signal PProcessing*, Vol 5, pp. 2427-2430, 1990.

69. LeDimet F.X. and Ngodock H.E. Sensitivity analysis in variational data assimilation, *Journal of the Meteorological Society of Japan*, 75(1B), pp. 245-255, 1997.
70. Lermusiaux P.F.J. and Robinson A.R. Data assimilation via error subspace statistical estimation, Part I: theory and schemes. *Monthly Weather 2 Review*, 127(7) pp. 1385-1407, 1999.
71. Leibfritz F. and Volkwein S. Numerical feedback controller design for PDE systems using model reduction: techniques and case studies. *Real-Time PDE-Constrained Optimization*, SIAM, 2006.
72. LeVeque R.J. *Numerical methods for conservation laws*. Birkhauser, 1992.
73. Li C.-K., Mathias R. Interlacing inequalities for totally nonnegative matrices, *Linear Algebra and its Applications*, Vol 341, 1-3, pp 35-44, 2002.
74. Llanos M.P. and Rossi J.D. Blow-up for a non-local diffusion problem with Neumann boundary conditions and a reaction term, *Nonlinear Analysis TM&A*, Vol 70, 4 pp 1629-1640, 2009.
75. Loeve M., Nostrand V. *Probability theory*, Princeton, N.J., 1964.
76. Lorenz E. and Emanuel K. Optimal sites for supplementary weather observations: simulation with a small model, *Journal of the Atmospheric Sciences*, Vol 55, 3, pp 399-414, 1998.

77. Luong B, Blum J. and Verron J. A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems*, Vol 14, 4, pp 979-997, 1997.
78. Matsuo, T. Understanding data assimilation: how observations and a model are weaved into analysis via statistics (Presentation), Symposium on Space Weather, the 84th AMS Annual meeting, 2004.
79. Malengier M. Application of the adjoint equation methods for parameter identification problems in nonlinear diffusion equations (Extended abstract), ICNAAM 2005.
80. Malinowski E.R. *Factor analysis in chemistry*, Wiley, 2002.
81. Neophytou M.K., Goussisa D.A., Van Loon M. and Mastorakos E. Reduced chemical mechanisms for atmospheric pollution using Computational Singular Perturbation analysis, *Atmospheric Environment* Vol 38 (22), pp 3661-3673, Jul 2004
82. Nguyen N.C. Reduced-basis approximations and a posteriori error bounds for nonaffine and nonlinear partial differential equations: application to inverse analysis (Thesis) Singapore-MIT Alliance, 2005.
83. Opmeer M.R., Wubs F.W. and Sorensen J.K. Evaluation of POD-based techniques applied to parameter dependent nonturbulent flows, *SIAM Journal of Scientific Computing*, 21(4) pp. 2469-2474, 2005.

84. Papadoupolo T. and Manolis Lourakis I.A. Estimating the Jacobian of the singular value decomposition: theory and applications, ECCV(1): pp.554-570, 2000.
85. Petzold L., Li S., Cao Y. and Serban R. Adaptive numerical methods for sensitivity analysis of differential-algebraic equations and partial differential equations, Computers & Chemical Engineering, Vol 30, 10-12, pp 1553-1559, 2006.
86. Petzold L. and Wenjie Z. Model reduction for chemical kinetics: an optimization approach.
87. Puel, J.P. A nonstandard approach to a data assimilation problem and Tychonov regularization revisited. SIAM Journal of Control and Optimization, Vol 48, 2, pp 1089-1111, 2009.
88. Ravindran S.S. Reduced-order adaptive controllers for fluid flows using POD, SIAM Journal on Scientific Computing, 15(4), pp. 457-478, Dec 2000.
89. Rathinam M. and Petzold L. A new look at proper orthogonal decomposition, SIAM Journal on Numerical Analysis, Vol 41 , 5, pp 1893-1925, 2003.
90. Rewiński M.J. and White J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol 22 , 2, 2003.

91. Roderick O., Anitescu M., Fischer M. and Won Sik. Y. Stochastic finite element approaches using derivative information for uncertainty quantification (Preprint) ANL/MCS 1551-1008, 2007.
92. Roderick O., Anitescu M. A note on sensitivity analysis of the multiphysics model (Technical report), Argonne National Laboratory, 2007.
93. Roderick O., Saftoiu I. Polynomial interpolation for predicting decisions and recovering missing data (Preprint) ANL/MCS ??? , 2007.
94. Rommes J. Methods for eigenvalue problems with applications in model order reductions (Thesis), Utrecht University, 2007.
95. Rowley C.V, Colonius T. and Murray R.M. Model reduction for compressible flows using POD and Galerkin projection, *Physica D*, 189(1-2), pp.115-129, 2004.
96. Sandu A., Daescu D.N, Carmichael R. and Chai T. Chemical data assimilation for the observation of the Earth's atmosphere.
97. Sandu A., Daescu D.N, Carmichael R. and Chai T. Adjoint sensitivity analysis of regional air quality models, *Journal of Computational Physics*, 204: pp. 222-252, 2005.
98. Sandu A., Constantinescu E.M., Carmichael R. and Chai T. Localized ensemble Kalman dynamic data assimilation for atmospheric chemistry, 7th International Conference on Computational Science (proceedings), 1, pp 1018-1025, 2007.

99. Sandu A. and Sander R. Technical note: simulating chemical systems in Fortran90 and Matlab with the kinetic preprocessor KPP-2.1, Atmos. Chem. Physics, 6, pp. 187-195, 2006.
100. Sandu A., Verver G., Blom J.G., Spee E.J. and Carmichael R. Benchmarking stiff ODE solvers for atmospheric chemistry problems I, implicit versus explicit, Atmospheric Environment, Vol 31, 19, pp 3151-3166, 1997.
101. Sandu A., Verver G., Blom J.G., Spee E.J. and Carmichael R. Benchmarking stiff ODE solvers for atmospheric chemistry problems II, Rosenbrock solvers, Atmospheric Environment, Vol 31, 20, pp 3459-3472, 1997.
102. Saltelli A., Tarantola S. Campolongo F. and Ratto M. Sensitivity analysis in Practice, John Wiley&Sons, 2004.
103. Sanjay L., Mardsen J.E. and Glavaski S. Empirical model reduction of controlled linear systems, Proceedings of the IFAC World Congress, pp. 473-478, 1999.
104. Schnabel R.B. and Toint Ph.L. Forcing sparsity by projecting with respect to a non-diagonally weighted Frobenius norm. Mathematical Programming, Vol 25,1 pp 125-129, 1983.
105. Sportisse B. and Djouad R. Use of proper orthogonal decomposition for the reduction of atmospheric chemical kinetics, Journal of Geophysical Research, Vol 112, 6, Mar 2007.

106. Sportisse B. and Quelo D. Data assimilation and inverse modeling of atmospheric chemistry, *Proceedings of Indian National Science Academy*, 69, 6. pp 661-668, 2003.
107. Sorensen D.C. Implicitly restarted Arnold/Lanczos methods for large-scale eigenvalue calculations (Technical report), 1995.
108. Stein L.M. Interpolation of spatial data; some theory for Kriging, Springer, 1999
109. Stoer J. and Bulirsch R. Introduction to numerical analysis, Springer, 2002.
110. Tabor M. Chaos and integrability in nonlinear dynamics: an introduction, Wiley, 1989.
111. Takei Y., Imai J. and Wada K. Opening door toward the 21st century. State space model identification using subspace extraction via Schur complement, *Transactions of the Institute of Electrical Engineers of Japan*, Vol 121-C, 1, pp 290-295, 2001.
112. Tsvetkova T.A. Construction of positive definite matrices which are near to a given one, *Ukrainian Mathematical Journal*, Vol 26 , 3 pp 348-348, 1974.
113. Ucinski D. Optimal measurement methods for distributed parameter system identification (Taylor and Francis systems and control book series), CRC, 2004.



114. Varga A. On stochastic balancing related model reduction, 39th IEEE Conference on Decision and Control (Proceedings), Vol 3, pp 2385-2390, 2000.
115. Vermeulen P.T., Heemink A.W. and Valstar J.R. Inverse modeling of groundwater flow using model reduction, Water Resources Research: 41(6), 2005.
116. Verwer J.G., Hausdorfer W. and Blom J.G. Numerical time integration for air pollution models, Report MAS-R9825, CWI, Amsterdam, 1997.
117. Volkwein S. Proper orthogonal decomposition: applications in optimization and control (Lecture Notes), CEA-EDF-INRIA Numerical Analysis Summer School, 2007.
118. Volkwein S. and Hepberger A. Impedance Identification by POD model reduction techniques, Automatisierungstechnik, 8, pp 437-446, 2007.
119. Wang K.Y., Lary D.J., Shallcross D.E., Hall S.M. and Pyle J.A. A review on the use of the adjoint method in four-dimensional atmospheric chemistry data assimilation, Meteorological Society, 127, pp. 2181-2205, 2001.
120. Wang Z., Navon I.M., Le Dimet F.X. and Zou X.L. The second order adjoint analysis: theory and applications, Meteorology and Atmospheric Physics, Vol 50, 1-3, pp. 3-20, Mar 1992.

121. Weideman J.A.C. The eigenvalues of Hermite and rational spectral differentiation matrices, *Numerische Mathematik*, Vol 61, 1 pp 409-432, 2005.
122. Willcox K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, *Proceedings of the 5th SMA Symposium*, Jan 2004.
123. Xia Y.S. and Wang J. On the stability of globally projected dynamical systems, *Journal of Optimization Theory and Applications*, Vol 106, 1, pp 129-150, 2000.
124. Xue Y., Cane M.A. and Zebiak S.E. Predictability of a coupled model of ENSO using singular vector analysis. part I: optimal growth in seasonal background and ENSO cycles, *Monthly Weather Review*, vol 125, 9 pp 2043-2056, 1997.
125. Yang C. and Petzold L. A posteriori error estimation and global error control for ordinary differential equations by the adjoint method, *SIAM Journal on Scientific Computing*, 26(2), pp. 359-374, 2004.
126. Zhano D. and Nagurney A. On the stability of Projected Dynamical Systems, *Journal of Optimization Theory and Applications*. Vol 85, 1 pp 97-124, 1995.
127. Zhedong L, Chen J., Zhu J. Wang R, and Navon I.M. An optimizing reduced order FDS for the tropical Pacific Ocean reduced gravity model,

International Journal for Numerical Methods in Fluids, Vol 55 , 2 pp 143-161, 2007.

128. Carbon Bond IV Photochemical Mechanism ,  
<http://airsite.unc.edu/soft/cb4/cb4main.html>
129. GEOS-Chem Model , <http://www.as.harvard.edu/chemistry/trop/geos>
130. General Circulation Models ,  
<http://www.giss.nasa.gov/research/modeling/gcms.html>
131. Software: Fast Chemical Solvers ,<http://people.cs.vt.edu/~asandu/>
132. KPP, the Kinetic Preprocessor ,  
<http://people.cs.vt.edu/~asandu/Software/Kpp/>
133. SAPRC Atmospheric Chemical Mechanisms,  
<http://www.engr.ucr.edu/~carter/SAPRC>
134. Test Set for IVP Solvers , <http://pitagora.dm.uniba.it/~testset/>