ABSTRACT

Title of Dissertation: APPLICATION OF REDUCED ORDER MODELING TECHNIQUES TO PROBLEMS IN HEAT CONDUCTION, ISOELECTRIC FOCUSING AND DIFFERENTIAL ALGEBRAIC EQUATIONS

Pramod P. Mathai, Doctor of Philosophy, 2008.

Dissertation directed by: Professor Benjamin Shapiro
Department of Aerospace Engineering

This thesis focuses on applying and augmenting 'Reduced Order Modeling' (ROM) techniques to large scale problems. ROM refers to the set of mathematical techniques that are used to reduce the computational expense of conventional modeling techniques, like finite element and finite difference methods, while minimizing the loss of accuracy that typically accompanies such a reduction.

The first problem that we address pertains to the prediction of the level of heat dissipation in electronic and MEMS devices. With the ever decreasing feature sizes in electronic devices, and the accompanied rise in Joule heating, the electronics industry has, since the 1990s, identified a clear need for computationally cheap heat transfer modeling techniques that can be incorporated along with the electronic design process. We demonstrate how one can create reduced order models for simulating heat conduction in individual components that constitute an idealized electronic device. The reduced order models are created using Krylov Subspace

Techniques (KST). We introduce a novel 'plug and play' approach, based on the small gain theorem in control theory, to interconnect these component reduced order models (according to the device architecture) to reliably and cheaply replicate whole device behavior. The final aim is to have this technique available commercially as a computationally cheap and reliable option that enables a designer to optimize for heat dissipation among competing VLSI architectures.

Another place where model reduction is crucial to better design is Isoelectric Focusing (IEF) - the second problem in this thesis - which is a popular technique that is used to separate minute amounts of proteins from the other constituents that are present in a typical biological tissue sample. Fundamental questions about how to design IEF experiments still remain because of the high dimensional and highly nonlinear nature of the differential equations that describe the IEF process as well as the uncertainty in the parameters of the differential equations. There is a clear need to design better experiments for IEF without the current overhead of expensive chemicals and labor. We show how with a simpler modeling of the underlying chemistry, we can still achieve the accuracy that has been achieved in existing literature for modeling small ranges of pH (hydrogen ion concentration) in IEF, but with far less computational time. We investigate a further reduction of time by modeling the IEF problem using the Proper Orthogonal Decomposition (POD) technique and show why POD may not be sufficient due to the underlying constraints.

The final problem that we address in this thesis addresses a certain class of dynamics with high stiffness - in particular, differential algebraic equations. With the help of simple examples, we show how the traditional POD procedure will fail to model certain high stiffness problems due to a particular behavior of the vector field which we will denote as *twist*. We further show how a novel augmentation to the traditional POD algorithm can model-reduce problems with *twist* in a computationally cheap manner without any additional data requirements.

APPLICATION OF REDUCED ORDER MODELING TECHNIQUES TO PROBLEMS IN HEAT CONDUCTION, ISOELECTRIC FOCUSING AND DIFFERENTIAL ALGEBRAIC EQUATIONS

by

Pramod P. Mathai

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Professor Benjamin Shapiro, Chair/Advisor
Professor Robert Sanner
Professor Christopher Cadou
Professor Sean Humbert
Professor Howard Elman, Dean's Representative.

Acknowledgements

I owe an unpayable debt to my parents for their selflessness and support over all these years. I owe much gratitude to my brother Pradeep for his advice and encouragement and also thank my sister-in-law Claudia for being such a wonderful addition to our family. I am grateful to my sister Preema for giving me a fresh perspective to life on so many occasions and for always being cheerful.

I thank my advisor Prof. Benjamin Shapiro for his guidance, enthusiasm, energy, and for giving me the freedom to work on different projects. I have observed and value the rare ability that he has in knowing when to deal with issues rather than the people behind the issues! I am also grateful to him and to Prof. Nuno Martins of the Electrical Engineering department at UMD for their advice and support on the gene regulation project. I thank my dissertation committee members - Prof. Howard Elman, Prof. Robert Sanner, Prof. Christopher Cadou, and Prof. Sean Humbert - for their suggestions and for their time.

I consider myself very fortunate in being able to add a great group of friends at every stage of my life. Their friendship and humor have moulded my life in ways for which I will be ever grateful.

Table of Contents

List of Tables

List of Figures

Abbreviations and Nomenclature

BT	Balanced Truncation
CA	Carrier ampholyte
FEM	Finite element model
FOM	Full order model
IEF	Isoelectric focusing
KST	Krylov Subspace Techniques
PDE	Partial differential equation
POD	Proper Orthogonal Decomposition
ROM	Reduced order model

Symbols	Description, with units in square brackets
$a_i(t)$	Coefficient of the basis vector ϕ_i in the reduced order model
C	Matrix that describes the interconnection architecture of a given device
C_p	Specific heat capacity [J/kg.K]
D_i	Diffusion coefficient of the i^{th} ampholyte [m^2/s]
D_H	Diffusion coefficient of hydrogen ion [m^2/s]
$E(x,t)$	Electric field intensity in the IEF channel at (x,t) [Volt/m]
E, A, B, D, P	Matrices used to describe a general state space system given by the equations $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = D^T x(t) + Pu(t)$
$\tilde{E}, \tilde{A}, \tilde{B}, \tilde{D}, \tilde{P}$	Reduced order state space matrices
$f_i(H)$	Titration curve for i^{th} ampholyte
$F(s)$	Full order transfer function of interconnected system (boards and components)
$\tilde{F}(s)$	Reduced order transfer function of interconnected system (boards and components)
$G(s)$	Transfer function of a general state space system where $Y(s) = G(s)U(s)$ and $Y(s)$, $U(s)$ are the Laplace transforms of $y(t)$ and $u(t)$ respectively
$\tilde{G}(s)$	Reduced order transfer function, where the full order transfer function is given by $G(s)$
$g_i(s)$	Full order transfer function that describes heat transfer in the i^{th} component
$\tilde{g}_i(s)$	Reduced order transfer function that describes heat transfer in the i^{th} component
$H(x,t)$	Hydrogen ion concentration at (x,t) [mol/l]
κ	Thermal conductivity [W/m.K]
$\mathbb{K}_j(A,B)$	j^{th} dimensional Krylov subspace corresponding to some matrices A and B
λ_{perc}	Percent of snapshot energy retained, in order to construct a reduced order model using traditional POD

l_o	Channel length in an IEF experiment
M_i	Mobility coefficient of the i^{th} ampholyte [m^2/V.s]
M_H	Mobility coefficient of hydrogen ion [m^2/V.s]
$m_j(\sigma^{(k)})$	j^{th} coefficient of the power series expansion of the full order transfer function $G(s)$, evaluated at $\sigma^{(k)}$
$\tilde{m}_j(\sigma^{(k)})$	j^{th} coefficient of the power series expansion of the reduced order transfer function $\tilde{G}(s)$, evaluated at $\sigma^{(k)}$
ϕ	Reduced order basis (this symbol is used in the context of a POD based reduced order model)
ϕ_i	Individual vectors in the reduced order basis ϕ
pI_i	Isoelectric point of the i^{th} ampholyte
pK_a	Acid dissociation constant for a donor group
pK_b	Base dissociation constant for a donor group
$P_j(x,t)$	Concentration of the j^{th} protein at (x,t) [mol/l]
$Q_i(x,t)$	Concentration of the i^{th} ampholyte at (x,t) [mol/l]
ρ	Density [kg/m^3]
$\sigma^{(k)}$	k^{th} interpolation point in KST
$\mathbb{S}$	Reduced order space
T_i	Absolute temperature at node i [K]
$x(t)$	State vector in a general state space system
$X(s)$	Laplace transform of a signal $X(t)$, where s is the frequency
x_f	Trajectory in full order space
x_r	Trajectory in reduced order space
$y(t)$	Output of a general state space system

Chapter 1

A Geometric Introduction to Model Reduction

1.1 What is model reduction?

Model reduction is defined as the process of describing and simulating the dynamics of a given physical problem in a minimal way. The implication is that there is always an inverse relationship between the accuracy and computational costs for modeling the problem and the choice of the model reduction technique is made by paying close attention to that relationship and the desired accuracy of the solution. The computational costs for any given model are specified by the required memory allocation as well as simulation time. A reduction in computational time and memory is achieved by reducing the number of *states* or *degrees of freedom* that is needed to model the physics.

The most popular computational techniques like finite element and finite difference methods, which are very accurate and have been widely applied across a range of physical problems, rely on breaking down the problem into computationally accessible sub-problems. The differential equation that typically describe the physics of the problem is broken down into a large number of states (from ≈14000 states for an unsteady CFD simulation of flow over an airfoil [119] to as high as 10^6 for complex problems like weather simulation [35]) with each state's behavior described by much simpler algebraic equation(s) that can then be accurately solved on a computer. Each of these simpler algebraic equations describe the evolution of a physical quantity on a single node (spatial location) using a few degrees of freedom per node.

There are many problems (describe later in Sec. 1.4) where such computational requirements (keeping track of millions of variables) are prohibitive and can in fact be done away with by considering only a few *dominant modes*, especially when one needs the value of only a specific set of node values. The modes of a physical problem are like the harmonics that describe the vibration of the skin of a drum. The most *dominant mode* would be the first harmonic, which is a good first approximation of the vibrational motion. The field of model reduction deals with the search for mathematical techniques that enable us to describe the *physics* of the problem with a few well chosen *modes*. For example, in the problem of turbulent flow, it was shown [99] how one could use a small number of dominant mode shapes to describe coherent structures for fluid vortices that are generated in turbulent flow.

To understand any model reduction technique, it is essential to understand the geometric picture behind the technique. One of the main mathematical concepts that lies behind all model reduction techniques is that of *projection*. In the next two sections, we motivate model reduction by linking it to the geometrical picture of projection. We then describe some of the vast number of problems where model reduction techniques have been applied. Finally, we end this chapter with a brief description of this thesis' contributions and its organization.

1.2 Static Reduced Order Modeling (ROM)

The three problems of pattern recognition (in static images), video encoding, and model reduction share the common aim of being able to minimally represent the ‘original data’. The ‘original data’ is different in these three cases and so is the idea of what the minimal representation is to be ultimately used for. Nevertheless, because of this common aim, the algorithms used for minimal representation in these three problems share common mathematical features, in particular, the idea of projection. For example, one of the popular methods of model reduction - proper orthogonal decomposition (POD) - is derived from the Karhunen Loeve technique in pattern recognition of static images. In this section and in the next, we will contrast the pattern recognition problem with the problem of model reduction in physics and video encoding.

Consider the following problem: How many dimensions (or degrees of freedom) are required to describe each of the 10 images of the ‘face of the bust’ in Fig. 1.1 ?

A simple way of storing each of the 10 images on a computer requires a vector of length equal to the number of pixels in each of those images. If the number of pixels in the first image equals 4096, then the computer considers the first image as ‘existing’ in 4096-dimensional space. For example, each of the entries in the 4096-dimensional vector that stores the first image records the gray-scale value of one pixel of that image. Hence, the complete set of 10 images can be represented as 10 different points in 4096-dimensional space, with each point corresponding to one of the 10 images.

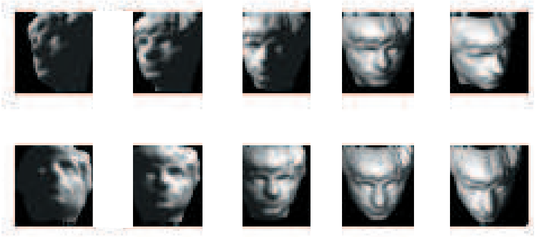


Figure 1.1: A set of 10 images of the face of a bust [108], each of them with a different pose or lighting direction. A human recognizes this face as 3-dimensional, but a simple way of storing these images in a computer involves a high dimensional vector, where each element of the vector records the gray-scale value of one pixel of an image.

It is natural to then ask whether one can make the computer understand that each of those images 'exists' in N-dimensional space, where $N \ll 4096$? It turns out that for this particular example, there is indeed a reduction algorithm (Isomap [108]), that can represent these 10 images in 3-dimensional space. Moreover, the three dimensions identified by the algorithm are 'up-down pose', 'left-right pose', and 'lighting direction' as shown in Fig.1.2 (where many more images of the face were used) unlike the 3 spatial dimensions that are apparent to a human observer. Each of the blue dots in Fig. 1.2 is a 3-dimensional representation of an image of the face with the 3 dimensions being 'up-down pose', 'left-right pose', and 'lighting direction'. A few of the faces are shown in Fig. 1.2 (which correspond to the blue dots that are circled in red) and the lighting direction for those faces are shown with the help of a knob below the face.

This is a standard problem in pattern recognition theory, which over here, we

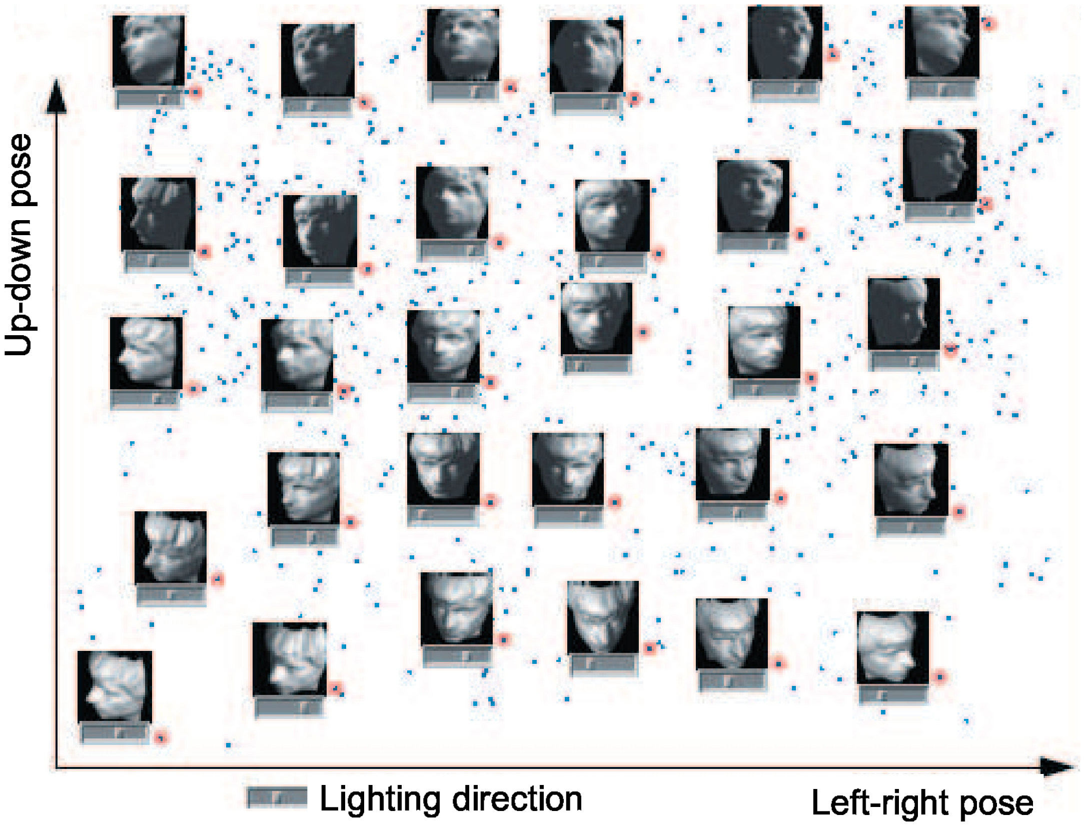


Figure 1.2: This figure is adapted from [108] and is a 3-dimensional representation of the images of the 'face of the bust' according as computed by the Isomap algorithm [108]. Each of the blue dots in the graph is an image of the face. The blue dots that have been circled in red, have the actual image shown next to the dot. The three dimensions are up-down pose, left-right pose and lighting direction. The axes for the up-down pose and left-right pose are shown, while the dimension for the lighting direction is shown by the knob that is below each image.

term as the ‘static model reduction problem’ - one searches for algorithms that can optimally represent a data set (we can now store the collection of 10 faces in the above example in just 3 instead of 4096 dimensions). One issue in static model reduction that is not apparent in the above example is the optimality of the representation. In the above example, a human observer would readily represent each face in 3-dimensions (x, y, and z dimension) and this representation is arguably optimal (a more optimal representation would have to go below 3 dimensions). One can argue that the algorithm's 3 dimensions are as optimal as the observer's. However, this *will not* necessarily hold in more complex problems. The observer's representation (the 3 spatial dimensions) is error free because each point on the face has a unique and fixed vector representation (upto Heisenberg's uncertainty principle!). This kind of error free representation will not hold in more complex problems - one hopes to reduce the error in the representation to the extent possible and some algorithms can even give apriori estimates for the errors (for example, the balanced truncation algorithm [67]). However, model reduction (whether it is done for static images or for physical problems) is always a game played in choosing between an optimal representation and an acceptable error, with the decision ultimately based on exactly what one intends to do with the reduced order model.

1.3 Dynamic Reduced Order Modeling

The next challenge is an optimal representation of the dynamics of some physical phenomenon. Suppose, one has a complex fluid phenomena for which one has available both, the governing equations, as well as a (computationally intensive) simulation. Two broad requirements that an engineer seeks are better physical insight into the problem and a faster simulation time.

The problem that we have here is that not only do we need to have an optimal (or compact) representation of the 'image' like we had in the static case, but we also have to 'track the trajectory' between the images. Suppose the full order model, which over here is the finite element model of the Navier-Stokes equation that describes the fluid phenomenon, exists in a high dimensional real space R^N (in this thesis we will only be concerned with real spaces). This means that if each of the dimensions represent an individual variable (say, velocity or density at a specific point on the grid), then the differential equation governing the dynamics of that problem (starting from a given initial state) can be assigned a unique trajectory T_f like the one shown in Fig.1.3 (where $N = 3$). Now, the reduced order modeling problem becomes one of 'shadowing' the trajectory T_f. Hence, if one wants to create a reduced order model in say, 2 dimensions, then we search for a 2-dimensional (linear) subspace, the 2-plane, on which one can shadow the original trajectory T_f with the requirement that the error between the original T_f and the reduced T_r is optimal in some sense. In principal, one could also search for a smaller nonlinear manifold in which one can 'shadow' the trajectory, but this thesis (and most of the existing literature for model reduction) focuses only subspaces which are linear by definition.

To put model reduction in perspective, it is interesting to compare the questions in image processing and model reduction. The dynamic model reduction prob-

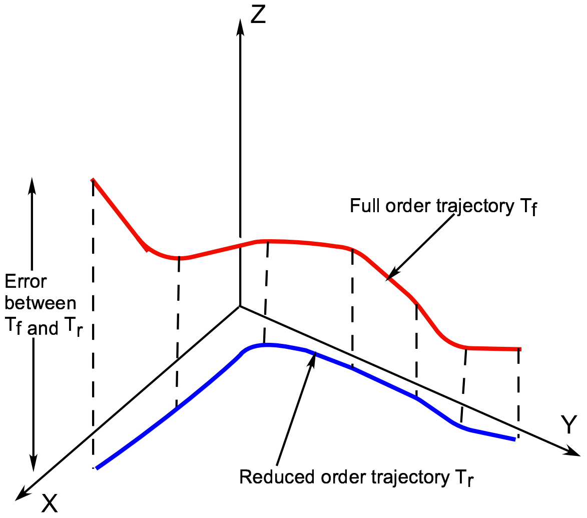


Figure 1.3: The full order trajectory T_f that exists in 3 dimensions is being 'shadowed' by the reduced order trajectory T_r in 2 dimensions. A reduced order modeling algorithm will need to compute the optimal subspace (shown in this figure as the X-Y plane) and the associated projection operator that minimizes the error between T_f and T_r.

lem has a much broader aim than the analogous problem in image processing - video compression (for example: MPEG compression). The aim of video compression is *efficiently archiving the details of that particular video* , with a very high degree of accuracy. The aim of dynamic reduced order modeling goes further - it is to capture the physics that caused the dynamics in the video with the help of a few sample videos, so as to be able to *efficiently predict the outcome of similar physics for a range of different initial conditions and/or parametric values.* So, while video compression deals with *accurate archiving*, model reduction deals with *archiving with the goal of capturing the underlying physics.* From hereon, we will drop the prefix 'dynamic' in 'dynamic model reduction' since we will always be interested in 'shadowing the dynamics' in this thesis.

1.4 Applications of Model Reduction

The need for reduced order modeling of any given physics will arguably always exist inspite of the availability of the state of the art computational hardware. Over the years, as computational power has increased, so has the need for modeling of even more ambitious problems, thus feeding the need for model reduction. We present a sample of the existing literature on reduced order modeling (ROM) techniques that shows the wide range of applications.

Two of the seminal papers in understanding macro-level properties of turbulent fluids were in the works of Sirovich [99] and Lumley [59]. They addressed the question of representing turbulent flows, that have chaotic attractors, with a relatively small number of 'eigenfunctions'. If this kind of representation is possible for turbulent flows, then the effect of parametric changes in flows or geometries on turbulent flows can be better understood. The essence of their work, for chaotic flows in confined geometries relies on the low-dimensional representation of the 'attractors' in turbulent flow. The entire turbulent flow can be regarded as the movement of a single point in infinite dimensional space. This point or *state* fully describes the flow at each instant of time. It was observed that for chaotic flows this point is drawn to a low-dimensional attracting set after an initial transient and that the dimension of this attracting set was low. Lumley and Sirovich proposed to sample this state at uncorrelated times and study the resultant ensemble so as to follow the state vector in the low-dimensional space of the attractor. They proposed that using a model reduction technique - proper orthogonal decomposition - one can construct the 'most likely' states for the flow. The perturbations in the flow can be represented in terms of these likely states or *eigenfunctions* and this representation can then be used for further parametric analysis.

Another need for model reduction is in the need for predicting tidal waves in coastal regions. The coast of Norway experiences frequent and dangerous tidal waves, which need to be predicted at least 6 hours in advance to evacuate the coastal population. The shallow water equations that are needed to predict the wave are modeled with 60,000 variables in the finite element formulation. Such a model takes at least 15 hours to compute and work on reducing this computational time has been done in [35] so that the results of the computational model can be used in time to warn and evacuate the affected population in case the tidal wave is dangerous. Similarly, work on using reduced order models for storm tracking in the Pacific North West has been described in [22]. The structural modules of the international space station require around 1000 states and in order to orient the space station in real time, reduced order controllers for performing the orientation of the space station are needed and have been described in [4]. Reduced order modeling of the millions of elements in the computational model of the Maxwell equations that describe the electrical behavior of an electronic chip has been widely researched including the first attempt by Pillage [76] (using Asymptotic Wave Transforms) and subsequently by others [64], [89], [46]. Balanced truncation - a model reduction technique - that uses ideas from control theory for reduced order modeling of linear systems was first proposed by Moore [67] and subsequently used by many others including [95], where it was used for control of forced response in bladed disks via mistuning. Arnoldi's original paper [6] on minimizing iterations for the matrix eigenvalue problem has subsequently been widely used in Krylov subspace model reduction, which will be discussed in detail later in this thesis. Krylov Subspace techniques have been used in interconnect modeling in chips [98], heat transfer modeling in chips [63] (which forms part of this thesis), in Pade based design for circuit analysis [23] and MEMS design [16]. Proper Orthogonal Decomposition, which will also be described later in this thesis, is based on the singular value decomposition theorem and was first proposed in probability theory by Loeve [56]. Since then, apart from Sirovich's work in turbulence, it has been widely used in modeling aeroelastic analysis of airfoils [86], feedback control of parabolic equations [7], modeling and control of thin film growth in HPCVD reactors [10], [43], [44] and finding stable conformations in protein folding [87].

In this thesis, we will focus on applying and augmenting model reduction techniques to problems in heat dissipation in electronic devices, separation of proteins and on a class of stiff problems that poses a problem for traditional model reduction techniques. We give a brief description of this thesis' contributions in the next section.

1.5 Contributions of this thesis

The need for Joule heat dissipation in electronic devices is well known especially as characteristic lengths in circuits reached less than 100 nm in the early part of this decade. Heat transfer design has become an integral part of chip design process, the design of small electronic devices like cell phones, MEMS devices, as well as in the design of larger scale problems like placement of blades in servers. The use of traditional computational techniques in heat transfer are either highly accurate, but computationally expensive (like finite element methods) or not very accurate at a fine scale, but much cheaper in terms of computational costs (like resistor-capacitance modeling of heat transfer). For the electronics industry, which innovates at the rate of a new generation device in every two years, there is a clear need for accurate, yet computationally cheap modeling techniques for heat transfer that can be incorporated along with the electronic design process. In this thesis, we show how to compactly model heat conduction in electronic devices using Krylov Subspace Techniques. We introduce a novel plug and play approach that would allow the designer to interconnect components for any given architecture (as needed by the VLSI design) in a cheap yet accurate manner. This kind of plug and play approach will enable the designer to test different placements of the components (within the constraints of the VLSI design process) and help him or her come up with an architecture that is best suited for efficient heat dissipation. This plug and play approach to understanding the effect of component architecture on heat dissipation due to conduction will be shown in chapters 2-5.

Another place where model reduction is crucial to better design is Isoelectric Focusing (IEF) - the second problem in this thesis - which is a popular technique that is used to separate minute amounts of proteins from the other constituents that are present in a typical biological tissue sample. This technique has been around for over forty years, but fundamental questions about how to design IEF experiments still remain because of the high dimensional and highly nonlinear nature of the differential equations that describe the IEF process as well as the uncertainty in the parameters of the differential equations. In this thesis, we show how with a simpler modeling of the underlying chemistry, we can still achieve the accuracy that has been achieved in existing literature for small ranges of pH (hydrogen ion concentration), but with less computational time. We investigate a further reduction of time by modeling the IEF problem using the Proper Orthogonal Decomposition (POD) technique and show why POD may not be sufficient due to the underlying constraints. Modeling IEF with simpler chemistry and investigating the application of POD to the IEF problem will be shown in chapters 6-8.

The final problem that we address in this thesis was inspired by our difficulty in modeling IEF with POD. However this is not about the IEF problem but addresses a certain class of dynamics with high stiffness - in particular, differential algebraic equations. We show how the traditional POD procedure will fail to model certain high stiffness problems due to a certain behavior of the vector field which we will denote as *twist*. We further show a novel augmentation to the POD procedure, which can model-reduce problems with *twist* in a computationally cheap manner without any additional data requirements. Augmenting POD to model-reduce problems which show *twist* will be shown in chapter 9.

Chapter 2

Heat Dissipation in Electronic Devices and the Need for Model Reduction

In this chapter, we begin by discussing how the increasing miniaturization in electronic devices has resulted in severe heat dissipation issues. We then discuss how the need for incorporating thermal design into the VLSI design of electronic devices results in either computationally expensive or inaccurate computational models. We focus on heat conduction and explain our novel approach to this problem - interconnecting component reduced order models to reproduce full device behavior in a way that is accurate, computationally cheap and also allows a ‘plug and play’ approach that is convenient for investigating the heat dissipation for various architectures.

2.1 Introduction

Thermal management and design in electronic and MEMS (Micro Electro Mechanical Systems) devices has assumed greater importance because of the trend towards higher power densities and the continuous miniaturization of electronic devices. A prediction by Gordon Moore in 1965 [68], about the number of transistors on an integrated circuit chip doubling every two years, has evolved from an observed trend to a stated goal of the semiconductor industry. The corresponding increase in heat dissipation per unit volume from a chip is shown in Fig. 2.1.

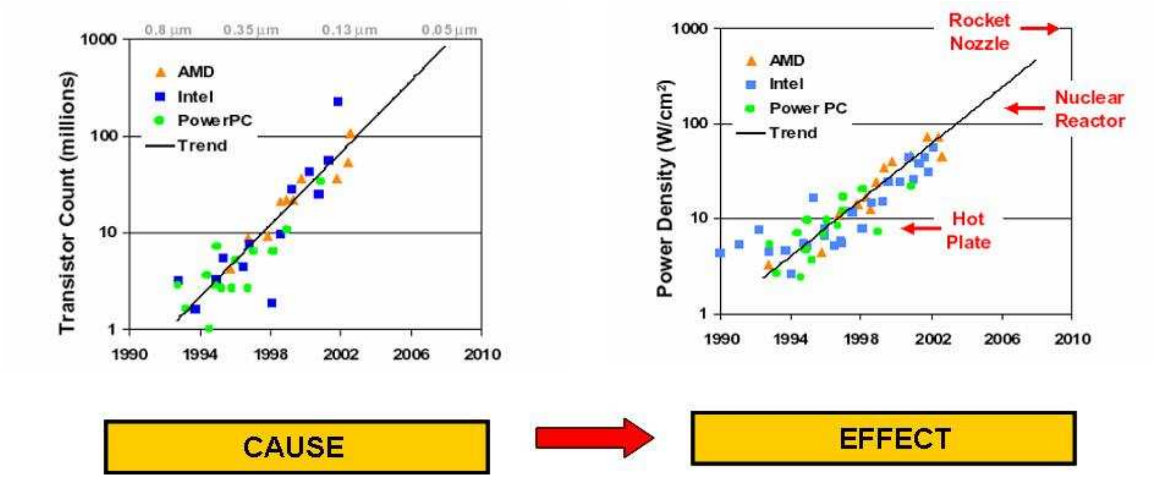


Figure 2.1: The trend in the exponential increase of transistors on a chip has resulted in the corresponding increase in heat dissipation. The extrapolated limit of the heat dissipation per unit area that is equivalent to a nuclear reactor has recently been reached in June 2008 at the IBM Zurich laboratory for a chip with 3-dimensional architecture. Figure courtesy Dr. Eric Pop at the University of Illinois, Urbana-Champaign. Data compiled by F. Labonte at Stanford University.

The extrapolated trend in Fig. 2.1 of the heat dissipation reaching the level of a nuclear reactor was reached in June 2008 at the IBM Zurich laboratory for a chip with 3-dimensional (stacked) architecture. With that kind of heat dissipation, the resultant temperature can easily surpass the melting point of the materials on the chip without proper cooling. However, long before such an extreme temperature is reached, individual transistors will stop functioning due to gate breakdown.

The switching of a transistor from an 'on' to 'off' state or vice versa generates heat due to the charging and discharging of the transistor capacitances and due to the flow of any (including leakage) current to the ground [49]. Part of the power dissipated varies linearly with the clock speed of the device. Hence, faster computers that have higher clock speeds imply higher power dissipation. In addition, there is Joule heat generated due to the many layers of interconnecting wires that transmit signals between different components in each device (the interconnects in the Pentium IV processor in 2001 had a total length of 2 km [4]).

There have been various strategies to actively cool chips, which is an entire research field in itself. This area of research focuses on the use of various kinds of heat pumps to circulate air or a liquid coolant throughout the CPU, for example, electroosmotically actuated micropump using DI water [42], valveless peizoelectrically actuated micropump with ethanol as a coolant [92], and flexural plate wave micropump using peizoelectrically actuation and fluorinert as a coolant [11]. In this thesis, we focus on another important strategy - designing of the various components in the VLSI architecture to achieve not only the traditional requirement of faster signal speed, *but also better heat dissipation into the environment.*

Today, it is common to find dual or four core chips even for desktops and laptops. Each core contains a central processing unit (CPU), which can process computations independent of other, spatially separated, cores. Thus, an appropriately parallelized code can be run faster on multi-core chips, and will moreover make a larger area available for heat dissipation. This is an example of designing different kinds of architectures for enabling better heat dissipation. However, this was not common till as recently as five years back. The introduction of multi-core chips, despite the prevalence of serial processing software, was precipitated in part by the performance of the Prescott series of chips in Intel's Netburst architecture in 2004, which failed due to the heat dissipation as clock speeds greater than 3.8 GHz were attempted. The VLSI on this chip was purportedly designed for 10 GHz, so this failure was a clear indication of bad thermal design rather than overclocking. Even today, most commercial software is not optimized for such multi-core architectures. Much of the software today is still written for serial, rather than parallel processing, which is a big concern for companies like Intel and Microsoft who have begun funding academic initiatives in 2008 for furthering the use of parallel processing (at the 'Universal Parallel Computing Research Centers (UPCRC)' across the United States).

Since 2004 there has been an even higher emphasis laid on designing VLSI architecture by simultaneously accounting for heat dissipation as well as signal speed for any given device architecture. The goal for industrial chip design has become the following -

Can chip designers, incorporate heat design into the VLSI design process without *slowing down the typical 2 year technology generation time-cycle at which new designs have been introduced since 1965?*

2.2 Examples of thermal design in electronic devices

In order to be able to create good thermal models of electronic devices, which are required to do system design, the designer has to have a thorough understanding of the heat transfer modes of the components of an electronic device. It is also necessary for the designer to be able to efficiently and accurately compute the impact of his or her design on the heat transfer properties of the device. A large volume of work exists with regards to understanding the physics of heat transfer in electronic enclosures. Yang [120] provides overviews of studies in the area of convection heat transfer. Moffat and Ortega [66] and Peterson and Ortega [75] have detailed discussions of applications involving natural convection in electronic enclosures. Larson and Viskanta [50] determined that radiation heat transfer was orders of magnitude larger than convection for 2 dimensional enclosures. Although convection and radiation form a very important part of the heat transfer processes in an electronic system, in this thesis we concentrate only on conduction. By itself, this is a very important problem (as discussed in [51]). It is hoped that the reduced order modeling ideas discussed in this thesis can be extended to convection and radiation.

Although we have given the physical example of individual chips in the above discussion, there is nothing sacred about this example. We could as well have chosen blades (single computers) in a server, components in a MEMS device, or any other example, where heat conduction poses a problem, where the physical length scale allows for modeling with the continuum approximation, and where efficiently designing the device architecture leads to better heat dissipation. In this work, we will denote all such physical examples with a single term: *components on a device.*

2.2.1 Heat conduction modeling on a device

The thermal properties of an electronic device are influenced by a range of parameters like chip positions, cooling channel shapes and their placement, and fan speed. An example of component layout in a device is shown in Fig.2.2.

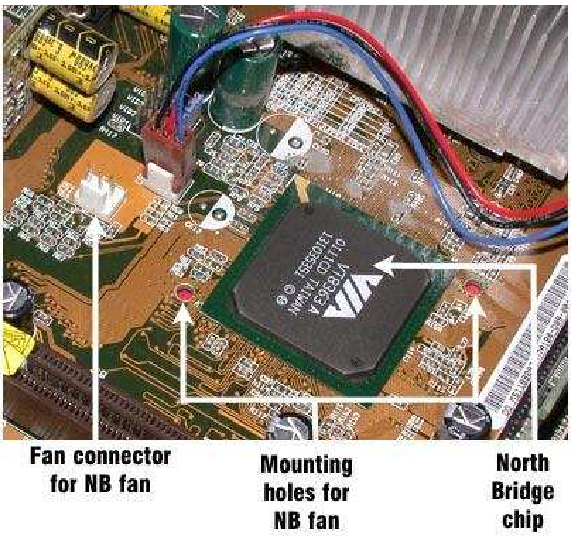
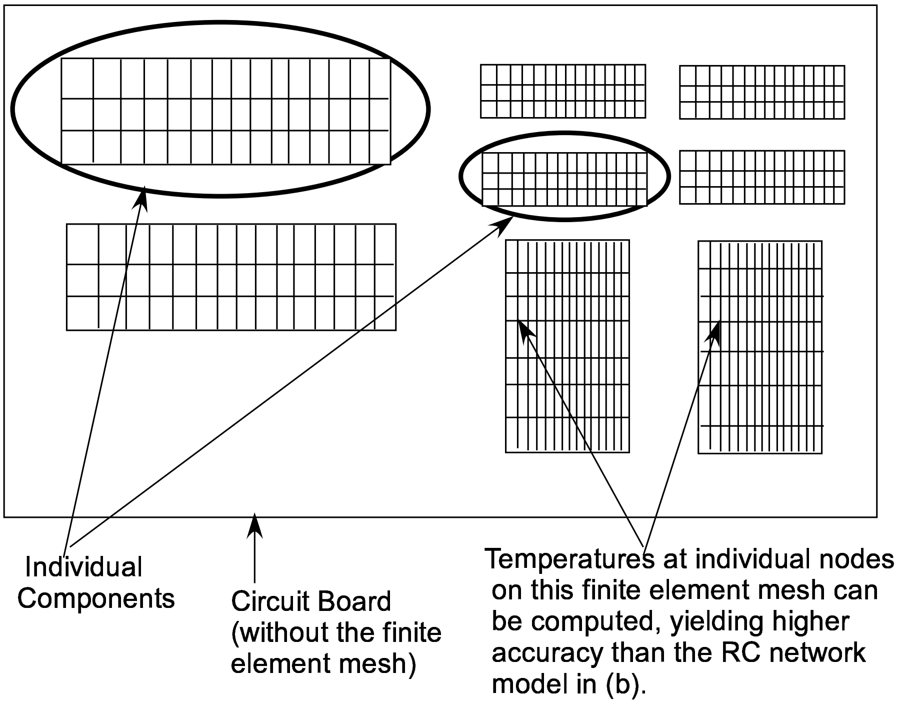


Figure 2.2: Layout of an electronic device (www.informit.com/content/images)

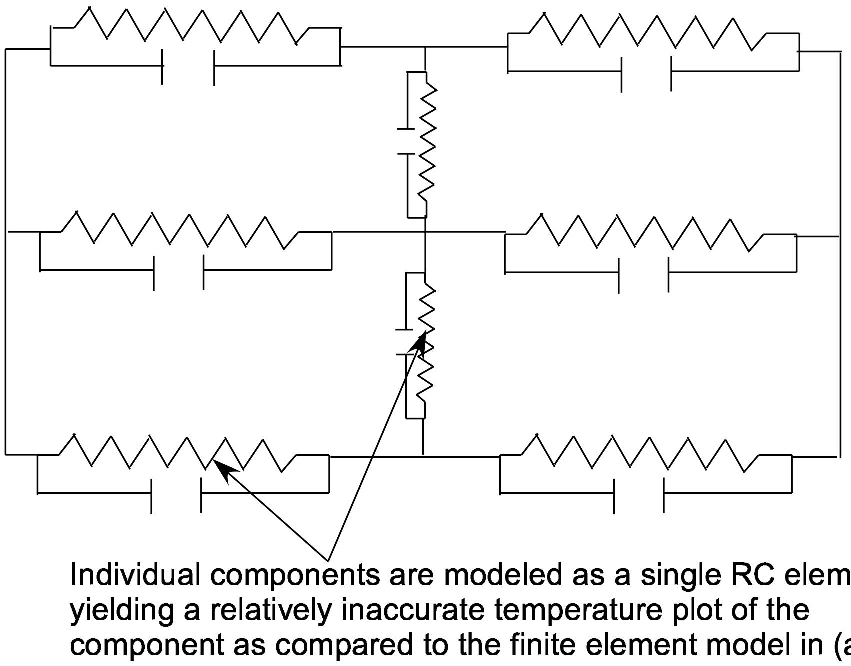
In order to design for the heat dissipation for the device, a designer must be able to search as much of the parameter space that influences the operating regime of the device as possible. Finite element or finite difference methods are common numerical procedures in studying heat transfer problems in electronic devices [17], [39].

Resistance models [90], eigenfunction related methods [27], [9] and the deconvolution method [107] have also been applied to the analysis of thermal networks. Such numerical schemes create a discretized model that is an approximation of the actual thermal problem. If the designer needs to guarantee a high level of accuracy for this large parameter space, the number of states in an FEM model has to be large. This implies increased costs in terms of memory requirements and computational time.

Thus, in order to efficiently incorporate the thermal design of chips in the VLSI design cycle, the computational costs for electro-thermal design of complex devices necessitate the development of compact models. Kreuger and Bar-Cohen [48] presented one of the earliest reduced order modeling efforts in which a chip package was modeled with a simplified resistor network that reduced the mesh size and the computational time involved. Lasance et al [117], formulated a simplified resistance network that was independent of boundary conditions, while in [12], a hierarchial reduced order modeling effort for chip packages was developed. While these approaches reduce the computational costs, they do not yield the detailed information that a designer needs for electro-thermal design of a complex device. For example, they do not help the designer in deciding the placement of the components on the board in Fig. 2.2. Fig. 2.3 below is a graphical comparison between different approaches to heat transfer design.


Individual Components
Circuit Board (without the finite element mesh)
Temperatures at individual nodes on this finite element mesh can be computed, yielding higher accuracy than the RC network model in (b).

(a) Typical full-order model of an electro-thermal system: Components are modeled with a finite element mesh. Large number of nodes implies good accuracy, but high computational costs.


Individual components are modeled as a single RC element, yielding a relatively inaccurate temperature plot of the component as compared to the finite element model in (a).

(b) A commonly used reduced-order model: Modeling the system as an RC network can minimize computation time, but the accuracy achieved may not be sufficient for complex circuits.

Figure 2.3: Typical methods that are used for modeling electro-thermal systems.

2.3 The ‘reduce-then-interconnect’ approach

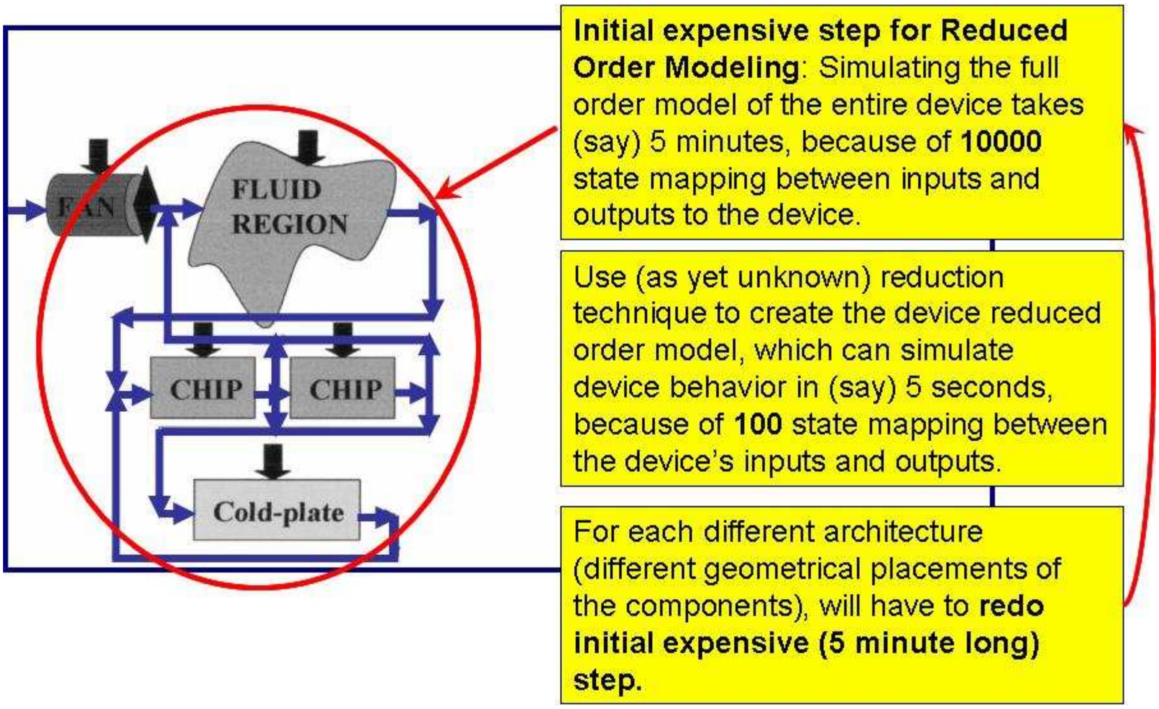
In [18], an Arnoldi-based reduced model of a thermal network has been proposed that gives very good results using a single reduced-order model for an entire device. For model reduction of electronic devices composed of many connected components, there are two broad approaches:

1. Connect the components and model-reduce the entire device OR
2. Model-reduce the components and then connect the individual reduced models to get a reduced-order model of the entire device.

We contend that the second option, if viable, is preferable to the designer. In the first option, a full (unreduced) finite element model must be created and reduced each time to create device models with different component architectures. In the second option, we have a library of reduced-order component models, and these components can be connected in different ways to get computationally cheap device models for different component architectures. We stress that heat flux that is exchanged between parts of different components can always be represented in the mathematical format that we will present in Chapter 4.

In this thesis we will focus on the reduce then connect method: we create reduced-order component models that are sufficiently rich to provide accurate results but are also sufficiently small to enable fast simulation of heat conduction in complex devices. A central requirement in our approach is that it should be possible to interconnect the reduced-order component models in a stable and accurate manner. In an application for controller reduction, Anderson[3] explains why a straightforward interconnection of reduced order component models can result in inaccurate or unstable behavior of the model of the entire device. We will formulate an algorithm to interconnect the reduced-order component models in such a way that the resulting device models for heat conduction are stable and accurate.

The amount of computational time that can be saved by using the reduce then connect paradigm can be demonstrated by a simple two dimensional example. Consider a device made up of five components as shown in Fig. 2.4.



Initial expensive step for Reduced Order Modeling: Simulating the full order model of the entire device takes (say) 5 minutes, because of **10000** state mapping between inputs and outputs to the device.

Use (as yet unknown) reduction technique to create the device reduced order model, which can simulate device behavior in (say) 5 seconds, because of **100** state mapping between the device's inputs and outputs.

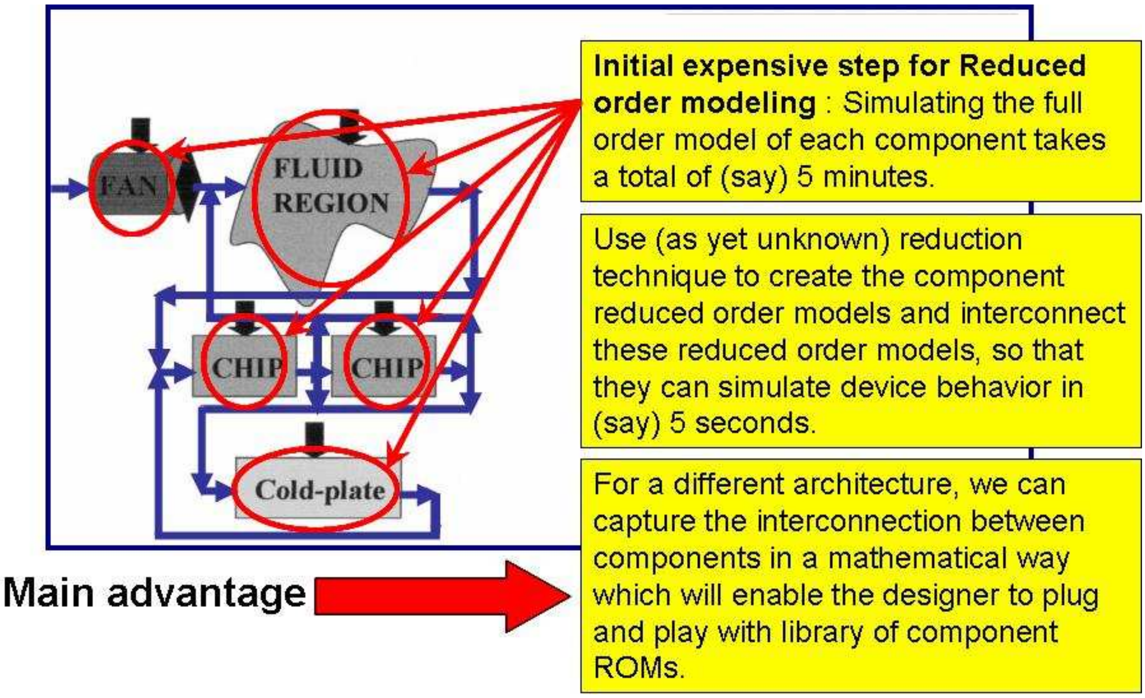
For each different architecture (different geometrical placements of the components), will have to **redo initial expensive (5 minute long) step.**

Figure 2.4: A single reduced order model of the entire device (all 5 components interconnected in a certain way) will take less simulation time than the full order model. However, computing reduced order models for each different architecture (different interconnections of the 5 components) will require an initial expensive computational step for simulating the full order model of that architecture. This will make the model reduction procedure expensive if one needs to evaluate many different architectures for an optimization run.

Suppose two of these components have external heating sources, and the device temperature is to be monitored at twenty junctions. If each of the five components is discretized with finite element methods in such a way that each component reduced order model has 2000 states, then the finite element model of the entire device will have 2000 × 5 = 10,000 states. Thus the mapping from the two external heating sources q1, q2 to the twenty junction temperatures $T_1, .., T_{20}$ will be a 800 state model. On a PC, such a model might take (say) 5 minutes of simulation time. Here, we should keep in mind that 5 minutes would then be the amount of time required to evaluate a *single* design in an optimization run (the total amount of time for the optimization would depend on the size of the entire design space).

Suppose that we apply model reduction with the connect then reduce paradigm. If this finite element description is reduced to 50 states by model reduction techniques, then the 10,000 state mapping from the heating sources q1 and q2 to the junction temperatures $T_1, .., T_{20}$ is replaced by a 50 state mapping that can be evaluated in (say) 5 seconds. Hence a simulation of the reduced order model will take 5 seconds for each architecture. However, if we want to examine a different device architecture, i.e., if we wanted to connect the 5 components in a different geometry, then we would have to *recreate* the reduced-order model of the device. Since the initial creation of the reduced-order model requires an evaluation of the original finite element model, it would take us 5 minutes to evaluate each new architecture.

Now, lets choose to use the 'reduce then connect' paradigm instead as shown in Fig. 2.5. Assume that each 2000 state component is replaced by a 10 state reduced-order model. Each model has inputs that will correspond to the fluxes and



Initial expensive step for Reduced order modeling : Simulating the full order model of each component takes a total of (say) 5 minutes.

Use (as yet unknown) reduction technique to create the component reduced order models and interconnect these reduced order models, so that they can simulate device behavior in (say) 5 seconds.

Main advantage

For a different architecture, we can capture the interconnection between components in a mathematical way which will enable the designer to plug and play with library of component ROMs.

Figure 2.5: If the same reduced order models of each of the 5 components can be interconnected for different architectures, the designer can adopt a plug and play approach for testing the whole-device heat dissipation. If viable, this approach will enable a simulation of each new architecture in 5 seconds, without the need of the initial expensive step of simulating the entire device. However, one needs to tune each component reduced order model to the dominant frequency range of the device (which varies with architecture only weakly) in order to make this approach possible. We show how one can make such a library of component reduced order models that can be used for different architectures.

temperatures that it will receive from the adjoining components and the models for the two active components also have an input that corresponds to each of their external heating fluxes. Likewise, each component has outputs that will supply its neighboring components with temperature and heating fluxes that they need as their boundary conditions. Now the five components that have 10 internal states each can be connected in different ways to achieve reduced-order models with different architectures. Each interconnected model will have 50 states and will evaluate in 5 seconds. In the reduce then connect paradigm, once a library of reduced-order models has been created, the designer can examine both different heating fluxes q1, q2 and different interconnection architectures, and can find the resulting temperatures at the twenty junctions in seconds.

Though the second approach is more useful, a simple interconnection of the reduced-order models can lead to stability problems as discussed in chapter 4 and in [3]. In chapter 4 we will show that if we intend to interconnect the model reduced subsystems, then we must make sure that the component reduced order models are accurate at the dominant frequencies of the interconnected system. We describe a novel control theoretic approach to interconnecting the reduced-order models of the components in a way that replicates the behavior of the entire device in a stable and accurate manner.

The next 3 chapters of this thesis are divided in the following manner. Chapter 3 discusses the structure of state space models that is derived from the FEM formulation of a conduction problem and also discusses reduced-order models. In particular, the Krylov based reduction algorithm that we have used for creating the reduced-order models of each component are discussed in Chapter 3 along with numerical examples. Chapter 4 discusses the approach that we use in interconnecting the various reduced-order models so that we can achieve accurate full system behavior and we demonstrate our approach with a numerical example. We conclude this portion with a discussion of our approach in Chapter 5. In the rest of this thesis, the terms ‘system’ and ‘device’ are used interchangeably, with the former being used in the mathematical portion of the text and the latter being used otherwise. We also interchangeably use the terms ‘sub-system’ and ‘component’, with the former being used in the mathematical portion of the text and the latter being used otherwise. The abbreviations FOM and ROM stand for Full Order Model and Reduced Order Model respectively.

Chapter 3

Model Reduction for Heat Conduction Using Krylov Subspace Techniques

A very popular and a highly accurate way of computing the heat conduction properties of an electronic device is by couching the partial differential equation for heat conduction in a variational form and computing the solution of the infinite dimensional partial differential equation (PDE) in a finite dimensional space. This is achieved by splitting the physical domain into a finite but very large number of smaller regions and the solution of the PDE in each small region is approximated by the solution of an algebraic equation that is solved on a computer using an appropriate solver. Such a solution of the PDE is termed as the finite element solution of the problem and is arguably the most accurate way to computationally obtain a solution of a given PDE (whose analytical solution is unknown).

In the heat conduction problem, if the heat conduction coefficient, density, and specific heat of the material does not vary appreciably with temperature, then the homogenous part of the partial differential equation for heat conduction is linear with respect to temperature. The finite element formulation of this linear dynamics can contain a large (in our examples, $O(10^3)$) number of degrees of freedom and is computationally expensive.

In control theory, such linear dynamics can be represented in what is known as the state space form in the time domain and in the transfer function form in the frequency domain. The state space form is equivalent to the finite element formulation. The state space form will have the same number of states as the total number of degrees of freedom in the finite element formulation. Once we have formulated the problem in the state space form, there are model reduction techniques like balanced truncation and Krylov subspace methods, that can considerably reduce the number of states (in our examples to $O(10)$) and hence reduce the computational costs, without losing much accuracy.

In this chapter, we we will define the transfer function for heat conduction of a single component in a device and show how one can construct a reduced order transfer function for that single component using concepts from Krylov subspace theory. We begin with an introduction to the concepts of state space systems and transfer functions.

3.1 Basics of control theory

In controls and dynamical systems theory, systems which can be described in the following form,

$$E\dot{x}(t) = Ax(t) + Bu(t). \tag{3.1}$$

$$y(t) = D^T x(t) + Pu(t). \tag{3.2}$$

with initial condition $x(t = 0) = x_0$ are termed as linear systems [72] (the notation Z^T stands for the transpose of the Z matrix). The vector $x(t)$ is called the state vector. This state vector $x(t)$ is chosen with the intent of providing a 'complete' description of the system, where the 'completeness' has to do with the *purpose* of the description. Eqns. 3.1 and 3.2 can be explained with the common example of simple harmonic motion of a swing in the park. For the purpose of providing a complete description of the motion of the swing at any given time t (under the simple harmonic assumption), the state vector $x(t)$ need only consist of two elements - the angular position $\theta(t)$ and the angular velocity $\dot{\theta}(t)$ [72]. If the motion of the swing is undamped, then Newton's first law and the simple harmonic assumption, provide the following governing equation for the motion of the swing

$$m\frac{d^2\theta(t)}{dt^2} = -k\theta(t) + F(t) \tag{3.3}$$

where m is the point mass of the spring, k is the spring constant of the swing, and $F(t)$ is the time-varying external force applied to the swing. We can write Eqn. 3.3 in the state space form of Eqn. 3.1 where the state space matrices are

$$\begin{aligned} x(t) &= [\theta(t)\ \ \dot{\theta}(t)]^T \\ u(t) &= [0\ \ F(t)]^T \\ A &= \begin{bmatrix} 1 & 0 \\ -k & 0 \end{bmatrix} \\ E &= \begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix} \\ B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3.4}$$

If one is interested in observing the angular velocity $\dot{\theta}(t)$, then the output $y(t)$ in Eqn. 3.1 is $y(t) = \dot{\theta}(t)$, the state space output matrices D^T and $P(t)$ are given by

$$
\begin{aligned}
D^T &= [1 \ \ 0] \\
P &= [0 \ \ 0] \qquad (3.5)
\end{aligned}
$$

Generalizing this, if $x(t)$ has dimension $M \times 1$, then the M elements of $x(t)$, should be chosen so that they 'completely' describe the system. If there are p inputs provided to the system, then $u(t)$ will have size $p \times 1$. The vector $y(t)$ describes the observation that one needs to make regarding the system at a given time t. This observation could consist of any linear combination of the elements of the state vector $x(t)$. In general, $y(t)$ can be of size $n \times 1$, if one needs to make n observations about the system. For the general case, especially when constraints are involved in the update of the state vector $x(t)$, the velocity vector $\dot{x}(t)$ has a multiplier E of size $M \times M$. In many cases, E is the identity matrix of size M, but for the theorems in this chapter, we will not assume that E is the identity. Given the sizes of the vectors, $x(t), u(t)$, and $y(t)$, the sizes of the matrices A, B, D^T, P, and E follows - A has size $M \times M$, B has size $M \times p$, D^T has size $n \times M$, P has size $r \times p$, and E has size $M \times M$.

3.1.1 Transfer function

The behavior of the output $y(t)$ in Eqn. 3.1 in response to the input $u(t)$ which varies sinusoidally at a certain frequency s, is very useful in analyzing the behavior of a linear system because the response to an arbitrary $u(t)$ can be analyzed in terms of a linear combination of the responses to the sinusoidally varying Fourier components of $u(t)$. The frequency response of the output $y(t)$ to a sinusoidal input $u(t)$ can be analyzed with the help of the Laplace transform. The Laplace transform for an arbitrary signal $X(t)$ is denoted as $X(s)$ and is defined by

$$X(s) = \mathcal{L}(X(t))(s) = \int_{0^-}^{\infty} e^{-st} X(t) dt \tag{3.6}$$

where $\mathcal{L}(.)$ denotes the Laplace transform and 0^- denotes the left limit (from $-\infty$) to 0. The state space equation given in Eqn. 3.1 can then be represented in the frequency domain with the help of the Laplace transform to give [72]

$$Y(s) = D^T(sE - A)^{-1}(Ex(0^-) + BU(s)) + PU(s) \tag{3.7}$$

where $Y(s)$ is the Laplace transform of $y(t)$ and $U(s)$ is the Laplace transform of $u(t)$ and $x(0^-)$ is the initial condition for $x(t)$. Hence we have

$$Y(s) = G(s)U(s) \tag{3.8}$$

where the function $G(s)$ that is defined by $G(s) = D^T(sE - A)^{-1}(Ex(0^{-1}) + B) + P$ is called the transfer function of the linear system given in Eqn. 3.1.

In the next section, we describe how one can construct the transfer function for describing heat conduction in a single component.

3.2 Transfer function for heat conduction

If each of the five components in Fig. 3.1, can be described with the help of linear system dynamics, then they will each have a *transfer function* $G_i(s)$ as shown on the right of Fig. 3.1. The physical interconnection between them (because of the heat fluxes exchanged between them) can be shown with the help of a block diagram as shown on the right of Fig. 3.1. In this chapter, we will only be concerned with creating a reduced order model of the transfer function of a single component (for example, $G_3(s)$ of the chip in Fig. 3.1). In the next chapter, we will show how the interconnection (due to the heat fluxes) between the components can be represented in a linear form. Fig. 3.1 is only for illustrative purposes - in this thesis we are only dealing with heat conduction, whereas the fan and fluid region in Fig. 3.1 will transfer heat via convection. The numerical examples in our work, will always deal with examples of heat conduction between solid components.

The partial differential equation describing heat conduction is given by

$$\nabla \cdot (\kappa \nabla T) + Q - \rho C_p \frac{\partial T}{\partial t} = 0. \tag{3.9}$$

T and Q denote the temperature (in K) and the heat input (in W/m^3) respectively. The thermal conductivity κ, density ρ, and the specific heat capacity C_p are assumed to be constant in our examples. In order to apply Krylov model reduction techniques to any given problem, it is first necessary to couch the physics of heat conduction in a state space format. In order to do this, we first create a finite element formulation of the conduction (diffusion) equation using the commercial software package FEMLAB [24]. The state space model of an isolated component can be extracted from this finite element model by linearizing the problem around a nominal temperature (we use FEMLAB to create the state space model from the finite element solution). The governing equations of the state space model are

$$E\dot{x}(t) = Ax(t) + Bu(t). \tag{3.10}$$

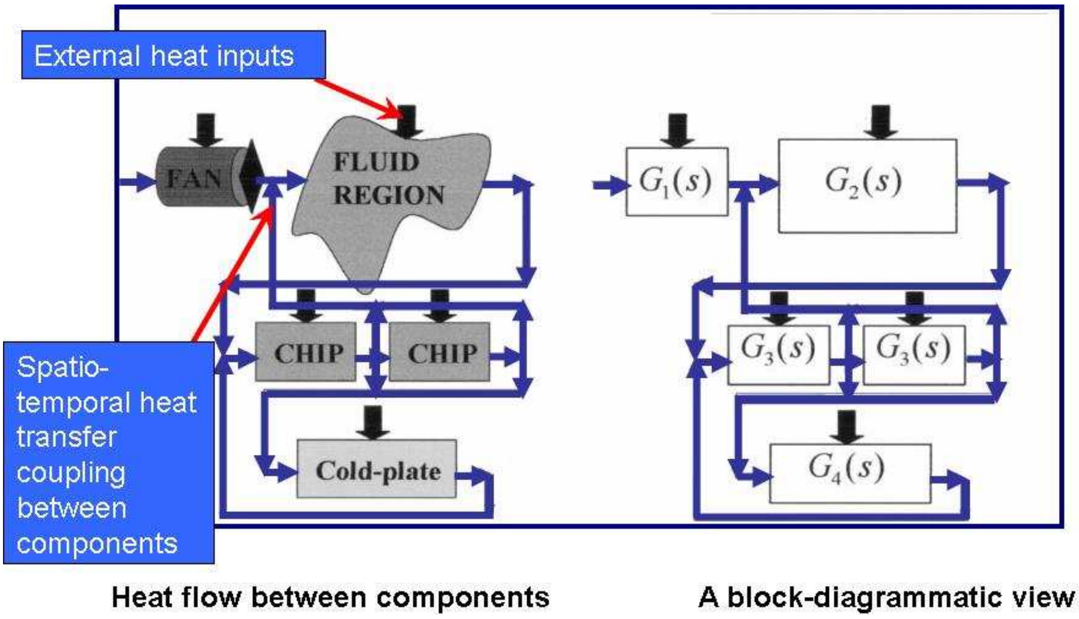

External heat inputs
FAN
FLUID REGION
CHIP
CHIP
Cold-plate
Spatio-temporal heat transfer coupling between components
$G_1(s)$
$G_2(s)$
$G_3(s)$
$G_3(s)$
$G_4(s)$
Heat flow between components
A block-diagrammatic view

Figure 3.1: Each component of the device on the left will have a transfer function that describes heat conduction in that component. The physical interconnection between the different components, because of the heat fluxes exchanged between them, can be represented with the help of a block diagram as shown on the right of the figure. In this chapter, we will only be concerned with creating a reduced order model of the transfer function of a single component, for example, $G_3(s)$ of the chip in this figure.

$$T(t) = D^T x(t). \tag{3.11}$$

in which $x(t)$ is the M x 1 vector of the discretized temperature modes at the various nodes of the mesh, and $u(t)$ is the p x 1 vector of external heat inputs to the system (power dissipated by the circuit, or flux received by the circuit). A and E are the constant M x M stiffness and mass matrices respectively. B is a constant M x p matrix that converts the external heat inputs $u(t)$ into an M x 1 input vector for the differential equation governing the state vector $x(t)$. For typical structures of electronic systems, M is usually very large. $T(t)$ represents a set of n junction temperatures at specified locations. D maps the M thermal modes to the n junction temperatures.

Assuming zero initial temperature offset $x(0) = 0$ (we can include an initial temperature in the formulation below but it does not change the basic results in what follows), and taking the Laplace transform of Eqn. (3.10), we get the frequency domain formulation of the state space model as

$$sEx(s) - Ax(s) = Bu(s) \tag{3.12}$$

in which $x(s)$ and $u(s)$ are the Laplace transforms of $x(t)$ and $u(t)$. Hence, the Laplace transform from the heat input $u(s)$ to the junction temperatures $T(s)$ is given by

$$T(s) = D^T(sE - A)^{-1}Bu(s) \tag{3.13}$$

The matrix $D^T(sE-A)^{-1}B$ is termed as the transfer function for heat conduction of the component and is denoted by the symbol $G(s) = D^T(sE - A)^{-1}B$. Eqn. 3.13 is the complete state space model of heat conduction in the component, which relates the heat inputs $u(s)$ and the junction temperatures $T(s)$ in the frequency domain. The frequency s is the speed of response of the system. The higher frequency modes of the system die out quickly, while the lower frequency modes are dominant and are mainly responsible for the long term response of the system.

3.3 Reduced order modeling of a single component

In the previous chapter, we discussed the potential utility of the plug and play approach of creating reduced-order models of individual components before interconnecting them to form a reduced-order model of the entire device. In this section, we show how the well known Krylov Subspace Technique [32] can be used for creating a reduced-order model of a single component of a device. This kind of reduced order modeling has been shown by other groups including [18] (where it was done for the entire device) and it forms the preliminary step for the interconnection idea that we have developed in this thesis.

The reduced-order model for a given component can be obtained by projecting the original linear dynamic system (3.10)-(3.11) onto a smaller state-space of dimension $m << M$, by means of an M x m projection matrix U. The geometrical picture behind this projection is the same as the one shown in Fig. 1.3 where we had the original FOM trajectory T_f in $M = 3$ being projected onto the ROM space with $m = 2$ dimensions to give the ROM trajectory T_r. Derivation of the matrix U is a critical step in any ROM technique and will be outlined later in this section, but for now we assume that U is known. The state variable $x(t)$ can then be written as

$$x(t) = U\tilde{x}(t). \tag{3.14}$$

where $\tilde{x}(t)$ is the state vector projected onto the reduced space. Here $x(t)$, of dimension M is large and $\tilde{x}(t)$, of dimension m, is small. Substituting Eqn. (3.14) into Eqn. (3.10) and multiplying by U^T, a reduced-order model of the form

$$\tilde{E}\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t). \tag{3.15}$$

$$T(t) = \tilde{D}^T\tilde{x}(t). \tag{3.16}$$

is obtained, in which $\tilde{A}, \tilde{E}, \tilde{B}$ and $\tilde{D}$ are given by

$$\tilde{A} = U^T AU,\ \tilde{E} = U^T EU,\ \tilde{B} = U^T B,\ \tilde{D} = U^T D. \tag{3.17}$$

Thus, the Laplace transform of the junction temperatures given by the reduced-order model of the component is

$$\tilde{T}(s) = \tilde{D}^T(s\tilde{E} - \tilde{A})^{-1}\tilde{B}u(s). \tag{3.18}$$

The projection matrix U must be chosen such that the input-output mapping of the reduced-order model approximates the input-output mapping of the unreduced model. We explain our choice of U in the next subsection.

3.3.1 Choice of reduction algorithm

Our choice of reduction algorithm for model-reducing each component was guided by our idea of interconnecting the reduced order models of components. We will describe the interconnection idea in detail in the next chapter. However, we will jump ahead and state the main criteria that any potential ROM approach needs to have, in order for us to be able to interconnect ROMs of different components. This criteria is *frequency weighting.*

Good frequency weighting in a ROM algorithm means that one should be able to make the reduced order model match the full order model in the frequency band that one desires. Suppose, that the component functions in the frequency range $10 - 100\ Hz$. A good reduced order model is one that matches the full order model's response in the frequency range $10 - 100\ Hz$, even if this means that the reduced order model is not as accurate as the full order model in the rest of the frequency spectrum. In the next chapter, we will show why one needs the individual component ROMs to be accurate at the device's natural frequency. Any candidate reduction technique should be amenable to frequency weighting so that individual component ROMs can be tuned to be accurate at the device's natural frequency.

Our first choice for a suitable ROM approach was balanced truncation (BT), which was first proposed by Moore [67]. The main attraction in BT is the optimality of the reduced order model - one can apriori prescribe the exact number of modes in the reduced order model so that the desired error bound between the FOM and ROM is satisfied. The idea behind this approach is that for state space systems, it is possible to choose a coordinate system for representing the state vector in which the states that cannot be observed in the system response are also simultaneously the states that cannot be controlled by the system's external inputs and vice versa. In control theory, such states are said to be simultaneously uncontrollable and unobservable (or minimally controllable and minimally observable). The transformation matrix U in Eqn. 3.17 is the one that equalizes, or *balances*, the controllability and observability grammians of the linear systems and thus enables one to order the states in decreasing order of simultaneous controllability and observability. A geometrical explanation of how BT 'balances' the ellipsoids that represent the controllability and observability grammians is given in [20]. Once this 'balancing' is achieved, the size of the model is reduced by choosing only those states that are simultaneously highly controllable and highly observable and discarding the rest of the states.

The extent of controllability and observability of the states are quantified by the Hankel singular values [67] of the system. The main advantage of BT is that an apriori error bound is available (based on the Hankel singular values) - i.e., one can choose the exact number of modes that satisfies a preset error bound between the ROM and the FOM transfer functions.

However, our main requirement for a model reduction algorithm is frequency weighting, which we found to be a computationally prohibitive step in BT. In our simulations, we found that the state of the art frequency weighting in BT, as performed by [118], adds considerably to the already prohibitive cost of balanced truncation which, even without frequency weighting, has $O(N^3)$ computational cost and $O(N^2)$ storage costs [4], where N is the total number of states in the full order model of the component (we have more than 2000 states per component in our models).

For our problem, we found that model reduction (which is basically, making an appropriate choice of U) can be cheaply and accurately achieved with the help of a Krylov Subspace Technique (KST) algorithm which is explained below. The main advantage of KST (compared to BT) is that it provides an intuitive and computationally cheap way to perform frequency-weighting, as we show in the next subsection.

3.3.2 Model reduction by projecting on Krylov subspaces

The basic idea behind KST is to match the ROM and FOM in the desired frequency range. This is achieved by creating a reduced-order model such that the first few terms in the power series expansion of the ROM transfer function around some frequency σ, matches the first few terms in the power series expansion of the original (full-order) transfer function. This would ensure that the spectral behavior of the ROM transfer function near σ matches that of the FOM transfer function upto some higher order term. As we will show in this section [32], this matching of the first few power series terms of the FOM and ROM can be simultaneously achieved around multiple frequencies σ_i.

In model reduction literature, the terms (or coefficients) in the power series expansion are also known as moments [32]. The moments are more rigorously defined as the value and subsequent derivatives of the transfer function (given in Eqn. (3.13)) at $s = \sigma$, where σ is any particular frequency.

From the expression of the transfer function of the full order model $G(s)$ that is given in Eqn. 3.13, we can expand $G(s)$ around any given $s = \sigma$ (provided that $(\sigma E - A)$ is nonsingular) in the following way:

$$\begin{aligned} G(s) &= D^T(sE-A)^{-1}B = -D^T(A-\sigma E-(s-\sigma)E)^{-1}B \\ &= -D^T((A-\sigma E)(I-(s-\sigma)(A-\sigma E)^{-1}E))^{-1}B \\ &= -D^T(I-(s-\sigma)(A-\sigma E)^{-1}E))^{-1}(A-\sigma E)^{-1}B \end{aligned} \tag{3.19}$$

With a power series expansion around σ, $G(s)$ can be written as:

$$\begin{aligned} G(s) &= \sum_{j=0}^{j=\infty} -(s-\sigma)^j D^T((A-\sigma E)^{-1}E)^j(A-\sigma E)^{-1}B \\ &= \sum_{j=0}^{j=\infty} (s-\sigma)^j m_j(\sigma) \end{aligned} \tag{3.20}$$

where the coefficient

$$m_j(\sigma) \equiv -D^T((A-\sigma E)^{-1}E)^j(A-\sigma E)^{-1}B \tag{3.21}$$

is called the j^{th} moment of $G(s)$ at σ. The term σ is commonly referred to as an *interpolation point.*

Analogously, for the reduced order transfer function $\tilde{G}(s)$ given in Eqn.3.18, the transfer function and the moments at σ can be written by using the reduced order state space matrices $\tilde{A}$, $\tilde{B}$, $\tilde{D}$, and $\tilde{E}$ in the following way:

$$\begin{aligned} \tilde{G}(s) &= \sum_{j=0}^{j=\infty} -(s-\sigma)^j \tilde{D}^T((\tilde{A}-\sigma\tilde{E})^{-1}\tilde{E})^j(\tilde{A}-\sigma\tilde{E})^{-1}\tilde{B} \\ &= \sum_{j=0}^{j=\infty} (s-\sigma)^j \tilde{m}_j(\sigma) \end{aligned} \tag{3.22}$$

where the coefficient

$$\tilde{m}_j(\sigma) \equiv -\tilde{D}^T((\tilde{A}-\sigma\tilde{E})^{-1}\tilde{E})^j(\tilde{A}-\sigma\tilde{E})^{-1}\tilde{B} \tag{3.23}$$

is called the j^{th} moment of $\tilde{G}(s)$ at σ.

The choice of σ depends on the relevant frequencies of interest in the physical problem. As shown in Fig. 3.2, there can be multiple interpolation points in a typical model reduction algorithm if there are multiple regions in the frequency spectrum in which we want the reduced-order model to match the full-order model. Hence KSTs ensure that once the relevant frequencies of interest are provided as inputs to the KST algorithm, the output is a reduced-order model whose first few terms (the number of terms is decided by the designer) in its power series expansions around those chosen frequencies matches those of the full-order model at and around those same frequencies [32]. Mathematically, if the original system has the transfer function $G(s)$ and the reduced-order model has the transfer function $\tilde{G}(s)$, then KSTs provide a transformation matrix U, that projects the original system to the reduced space in such a way that the first j moments of the original system $G(s)$ match the first j moments of the reduced-order model $\tilde{G}(s)$. Since only the first few terms of the power series expansion of $G(s)$ and $\tilde{G}(s)$ are supposed to match, the match will be better at those frequencies which are near the interpolation points and will differ at other frequencies (as shown in Fig. 3.2).

The transformation matrix U is to be chosen so that it projects the FOM onto a subspace $\mathbb{S}$ in the manner of Eqn. 3.17 so that the first, say J, moments of the FOM transfer function $G(s)$ about an interpolation point σ matches the first J moments of the ROM transfer function $\tilde{G}(s)$. We will show how one can choose U so that this moment matching happens simultaneously around multiple interpolation points σ^k. The proofs used here are the same as the ones given in [91] and [32]

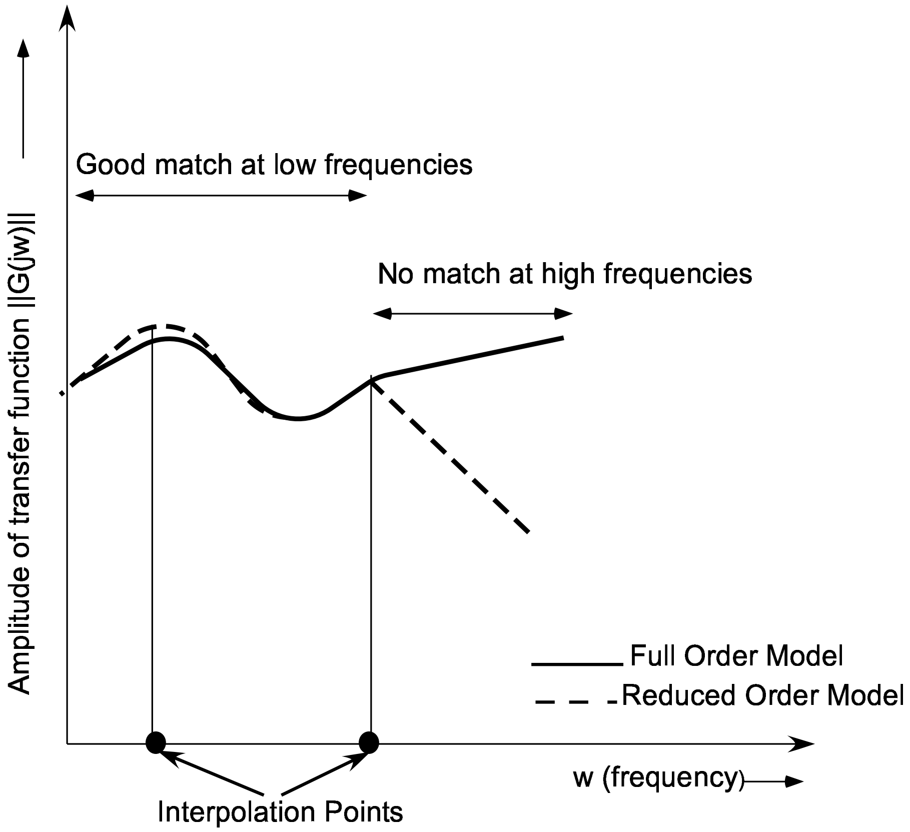

Amplitude of transfer function ||G(jw)||
Good match at low frequencies
No match at high frequencies
Full Order Model
Reduced Order Model
w (frequency)
Interpolation Points

Figure 3.2: Comparison of full-order model (FOM) and reduced-order model (ROM). They match at low frequencies because the interpolation points are in the low frequency range. At higher frequencies, the moments (the coefficients of the power series expansion of the transfer function) of the ROM and FOM are no longer equal and hence the behavior of the models diverge. This means that the ROM will match the FOM at low frequencies but not at higher frequencies.

except for some difference in notation.

First, we prove how this moment matching can be achieved between the FOM and ROM of a SISO (single input, single output) transfer function for a single interpolation point σ. Then we explain the generalization to the case of multiple $\sigma^{(k)}$, $k = 1, .., K$ and finally for MIMO (multiple input, multiple output) transfer functions with multiple $\sigma^{(k)}$, $k = 1, .., K$. For this, we first need to define a Krylov subspace.

Definition A j^{th} dimensional Krylov subspace $\mathbb{K}_j(R, f)$ corresponding to some matrix R and vector f is defined as

$$\mathbb{K}_j(R, f) = span(f, Rf, R^2 f, ..., R^{j-1} f) \tag{3.24}$$

where $span(v_1, v_2, ..., v_k)$ denotes the subspace spanned by the vectors $v_1, v_2, ..., v_k$.

The moment matching property between the FOM and ROM transfer functions of a SISO system is described in the following theorem.

Theorem 3.3.1 *For the full order SISO transfer function given by $G(s) = d^T(sE - A)^{-1}b$ (where b and d^T are vectors corresponding to the single input and output respectively), and the corresponding reduced order SISO transfer function given by $\tilde{G}(s) = \tilde{d}^T(s\tilde{E} - \tilde{A})^{-1}\tilde{b}$, we will have $m_j(\sigma) = \tilde{m}_j(\sigma) : j = 0, ..., J-1$ where $m_j(\sigma)$ and $m_j(\sigma)$ are the j^{th} moments of $G(s)$ and $\tilde{G}(s)$ respectively (as given in Eqns. 3.21 and 3.23), if $E, \tilde{E}, (\sigma E - A)$, and $(\sigma\tilde{E} - \tilde{A})$ are nonsingular and the transformation matrix U (that is used in Eqn. 3.17 to create $\tilde{G}(s)$ from $G(s)$) spans the Krylov subspace $\mathbb{K}_J((A - \sigma E)^{-1}E, (A - \sigma E)^{-1}b)$.*

Proof We first show how the zeroth moment matches for the hypothesis, i.e., $m_0(\sigma) = \tilde{m}_0(\sigma)$. Using Eqns. 3.23 and 3.17, the zeroth moment of $\tilde{G}(s)$ around σ is given by

$$\tilde{m}_0(\sigma) = -\tilde{d}^T(\tilde{A} - \sigma\tilde{E})^{-1}\tilde{b} = -d^T U(U^T AU - \sigma U^T EU)^{-1}U^T b \tag{3.25}$$

The vector $(A - \sigma E)^{-1}b$ is in the the Krylov subspace $\mathbb{K}_J$ and since U spans $\mathbb{K}_J$, there exists a vector r_0 such that

$$(A - \sigma E)^{-1}b = Ur_0 \tag{3.26}$$

Hence we have

$$\begin{aligned}(U^T(A - \sigma E)U)^{-1}U^T b &= (U^T(A - \sigma E)U)^{-1}U^T((A - \sigma E)(A - \sigma E)^{-1})b \\ &= (U^T(A - \sigma E)U)^{-1}U^T(A - \sigma E)Ur_0 = r_0 \end{aligned} \tag{3.27}$$

With this, we see that the zeroth reduced order moment $\tilde{m}_0(\sigma)$ equals the zeroth full order moment $m_0(\sigma)$ because $\tilde{m}_0(\sigma) = d^T Ur_0 = d^T(A - \sigma E)^{-1}b = m_0(\sigma)$.

For the next moment, the relation in Eqn. 3.27 can be used to conclude that

$$\begin{aligned}(U^T(A - \sigma E)U)^{-1}U^T EU(U^T(A - \sigma E)U)^{-1}U^T b &= (U^T(A - \sigma E)U)^{-1}U^T EUr_0 \\ &= (U^T(A - \sigma E)U)^{-1}U^T E(A - \sigma E)^{-1}b \end{aligned} \tag{3.28}$$

The vector $(A - \sigma E)^{-1}E(A - \sigma E)^{-1}b$ is also in the Krylov subspace $\mathbb{K}_J$. Hence there exists r_1 so that

$$(A - \sigma E)^{-1}E(A - \sigma E)^{-1}b = Ur_1 \tag{3.29}$$

Hence we have

$$\begin{aligned}&(U^T(A-\sigma E)U)^{-1}U^T((A-\sigma E)(A-\sigma E)^{-1})E(A-\sigma E)^{-1}b\\&\qquad=(U^T(A-\sigma E)U)^{-1}U^T(A-\sigma E)Ur_1=r_1\end{aligned} \tag{3.30}$$

With this, we can conclude that the first moment $\tilde{m}_1(\sigma)$ of the reduced order transfer function equals the first moment $m_1(\sigma)$ of the full order transfer function because

$$\begin{aligned}\tilde{m}_1(\sigma) &= d^TU(U^T(A-\sigma E)U)^{-1}U^TEU(U^T(A-\sigma E)U)^{-1}Ub\\&= d^TUr_1=d^T(A-\sigma E)^{-1}E(A-\sigma E)^{-1}b=m_1(\sigma)\end{aligned} \tag{3.31}$$

For the second moment, we use Eqns. 3.27 and 3.30 and knowing that $((A-\sigma E)^{-1}E)^2(A-\sigma E)^{-1}b$ can be written as the Ur_2 for some vector r_2 (since the vector $((A-\sigma E)^{-1}E)^2(A-\sigma E)^{-1}b$ lies in the Krylov subspace $\mathbb{K}_J$ which is spanned by U). By repeating these steps, this proof can continued for the first J moments to show that that $\tilde{m}_j = m_j\ \forall j=0,..,J-1$. □

In order to ensure that the first J_k moments of $\tilde{G}(s)$ around the k^{th} interpolation point $\sigma^{(k)}$ simultaneously match the corresponding moments of $G(s)$, for all the K interpolation points $\sigma^{(k)} : k=1,..,K$, we use the exact same proof as the above, with the requirement that the transformation matrix U simultaneously spans the K Krylov subspaces $\mathbb{K}_{J_k}((A-\sigma^{(k)}E)^{-1}E,(A-\sigma^{(k)}E)^{-1}b)$.

For MIMO systems, the j^{th} moment of the full order MIMO transfer function $G(s) = D^T(sE-A)^{-1}Bu(s)$ around the interpolation point $\sigma^{(k)}$ is given by the matrix function

$$m_j(\sigma^{(k)}) = D^T((A-\sigma^{(k)}E)^{-1}E)^{j-1}(A-\sigma^{(k)}E)^{-1}B \tag{3.32}$$

We note that the element in the (l_b, l_d) position of $m_j(\sigma^{(k)})$ is also the j^{th} moment of the SISO system that has state space matrices A, E, b, and d. Here, the vectors b and d are the l_b^{th} and l_d^{th} column vectors of the matrices B and D respectively. Hence, Theorem 3.3.1 can be applied to each of the elements of $m_j(\sigma^{(k)})$ to analogously prove the following moment matching theorem for MIMO systems [32], which we state here without further proof.

Theorem 3.3.2 *The first J_k moments $m_j(\sigma^{(k)}) : j = 0, ..., J_k - 1$ around each of the K interpolation points $\sigma^{(k)}$ $\forall k = 1, ..K$ of the full order MIMO transfer function given by $G(s) = D^T(sE - A)^{-1}B$ equals the corresponding J_k moments $\tilde{m}_j(\sigma^{(k)}) : j = 0, ..., J_k - 1$ of the reduced order MIMO transfer function given by $\tilde{G}(s) = \tilde{D}^T(s\tilde{E} - \tilde{A})^{-1}\tilde{B}$, if $E, \tilde{E}, (\sigma^{(k)}E - A)$, and $(\sigma^{(k)}\tilde{E} - \tilde{A})$ are nonsingular and the transformation matrix U (that is used in Eqn. 3.17 to create $\tilde{G}(s)$ from $G(s)$) simultaneously spans all the Krylov subspaces $\mathbb{K}_{J_k}((A - \sigma^{(k)}E)^{-1}E, (A - \sigma^{(k)}E)^{-1}B)$ $\forall k = 1, ..., K$.*

In particular, the KST algorithm we used in this paper is given in Algorithm 1.

3.3.3 Properties of ROMs created with Krylov Subspace Techniques

The transformation matrix U, is recursively built up by appending newly computed columns u_m, in each step of the loop. The newly added columns are orthonormalized with respect to the previously computed columns before appending them to the U matrix. Each subsequent column u_m (of the matrix U) contributes additional information about the system behavior in the reduced model. The inputs to this

Algorithm 1 Krylov Subspace Reduction via Arnoldi [32]

```
Initialize $m = 0$, $U = [\ ]$ ($U$ is initialized as an empty matrix)

for $k = 1$ to $K$ , ($K$ is the total number of interpolation points)

    for $j_k = 1$ to $J_k$ ($J_k$ is the number of moments to be matched at the $k^{th}$ interpolation point)

        if $j_k = 1$ ,
            $\tilde{u}_m = (\sigma^{(k)}E - A)^{-1}B$    ( $\sigma^{(k)}$ is the $k^{th}$ interpolation point.)
        else
            $\tilde{u}_m = (\sigma^{(k)}E - A)^{-1}E\tilde{u}_{m-1}$    ( $\tilde{u}_m$ is the $m^{th}$ column of $U$)
        end

        Orthonormalize $\tilde{u}_m$ with respect to all the previously
        computed columns of $U$ to get $u_m$
        $U = [U \ \ u_m]$
        $m = m + 1$
    end

end
```

algorithm are the full-order model, the choice of interpolation points $\sigma^{(k)}$, and the number of moments J_k to be matched at that interpolation point. The choice of the interpolation points will dictate the frequency range in which the reduced-order model is accurate.

It is a well known problem in Krylov subspace techniques that for MIMO systems in which we project only onto part of the controllability space (like in Algorithm 1), having multiple (≥ 8) inputs and multiple (≥ 3) moments matched at each interpolation point does not add new information to the transformation matrix U because of numerical limitations while computing successive columns u_m of U [32]. This is because limits to computational accuracy cause successive columns u_m to lie in the subspace of the previously computed columns of U, thus making U

less than full rank. In the next chapter, we will discuss how this affects our modeling of interconnected components, but we note here that in order to avoid having a less than full rank U matrix, it is better to have a smaller number of moments matched at many interpolation points than having many moments match at relatively fewer interpolation points [32]. This ensures that new information is always added to successive columns u_m and U is not rank deficient.

In many applications where a specific output or small set of outputs need to be monitored, a frequently used augmentation of Algorithm 1 is to incorporate information of the Krylov observable space given by $\mathbb{K}_J((A-\sigma^{(k)}E)^{-1}E,(A-\sigma^{(k)}E)^{-1}D^T)$ into the projection framework. In such a case, one uses a two-sided projection where,

$$\tilde{A} = V^T AU,\ \tilde{E} = V^T EU,\ \tilde{B} = U^T B,\ \tilde{D} = V^T D \tag{3.33}$$

is the reduced order state space matrix, with the matrix V constructed by applying Algorithm 1 to finding a basis for $\mathbb{K}_J((A-\sigma^{(k)}E)^{-1}E,(A-\sigma^{(k)}E)^{-1}D^T)$ and the matrix U is computed as in Algorithm 1. This can result in a smaller ROM than using a one-sided projection. However, we did not use the two-sided projection, because we are not concerned with any small set of node temperatures. Moreover, such a two-sided projection can many times result in an unstable ROM even when the original FOM is stable. Various algorithms, including the ones in [41] and [26] ensure stability in ROMs created by KSTs. We have observed that with the technique described in [41], where any unstable modes are discarded from the ROM, accurate ROMs can be achieved at the desired input frequencies.

The computational cost of this algorithm [32] comes from solving q linear systems of equations, with each system having dimension $N \times N$ where N is the size of the FOM state space matrix A and where

$$q = n \sum_{k=1}^{k=K} (j_k + 1) \tag{3.34}$$

where n is the size of the output vector, K is the total number of interpolation points, and j_k is the number of interpolation points matched at each interpolation point. This cost can be further reduced by exploiting the sparsity structure of the FOM state space matrices [4] which we have not implemented. There is no known global apriori error bound for Algorithm 1 for multiple interpolation points.

3.4 Numerical Examples

We demonstrate two examples for model reduction of the heat conduction problem for a single component. The Krylov subspace algorithm (Algorithm 1) was used to create the reduced-order models for the components shown in Fig.3.3. They demonstrate the efficacy of algorithm 1 in reducing the number of states without losing accuracy.

The partial differential equation describing heat conduction is

$$\nabla \cdot (\kappa \nabla T) + Q - \rho C_p \frac{\partial T}{\partial t} = 0. \tag{3.35}$$

T and Q denote the temperature (in K) and the heat input (in W/m^3) respectively. The thermal conductivity κ, density ρ, and the specific heat capacity C_p of each of the two systems are constant throughout the system and have the same values as silicon at 163 $W/m.K$, 2330 kg/m^3, and 703 $J/kg.K$ respectively.

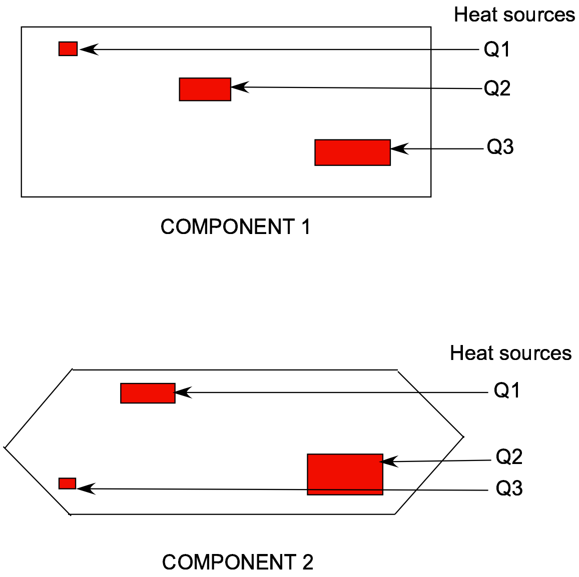


Figure 3.3: Component geometries and heat sources for the reduced-order modeling results shown in Fig.3.4 and Fig.3.5. For Components 1 and 2, $Q1 = 1 * 10^6\ W/m^3$, $Q2 = 3 * 10^6\ W/m^3$, $Q3 = 2.5 * 10^6\ W/m^3$.

In Fig.3.4, we show full-order and reduced-order temperature profiles for 3 heat sources (values mentioned in the Fig.3.3) on a single rectangular component. The four boundaries of the component are maintained at $T = 300\ K$. The plots show the absolute temperature plot of the component above the nominal temperature of $T = 300\ K$ at the end of 3 s. They were computed by calculating the temperature rises for equally spaced mesh points of the component and interpolating for the values between these points. The reduced-order model has 30 states and took 0.01 s to compute the temperature response on a computer running 64-bit MATLAB on a 2.3 GHz AMD Opteron processor. The full-order model has 1243 states and took over 13 s to compute the temperature response.

In Fig.3.5, we show temperature profiles for an arbitrarily shaped component

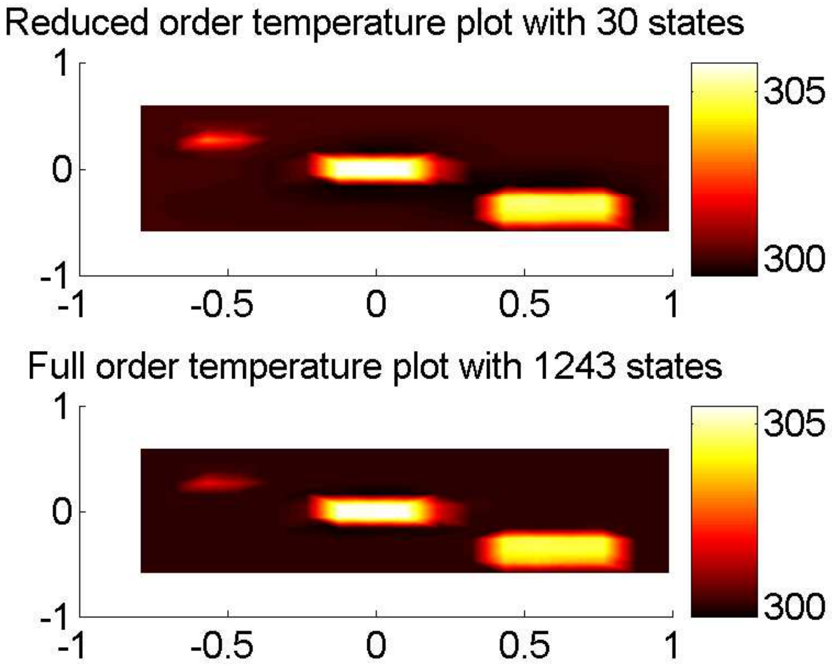


Figure 3.4: Comparison of reduced and full-order temperature plots of the top (rectangular) component with the heat sources shown in Fig.3.3. The temperature profiles are shown for the final time $t = 3s$.

having the same physical properties (silicon) as those in the above example. The shape was chosen to demonstrate that there is no restriction on the component geometry when we create its reduced-order model. The boundaries of the component are thermally insulated. The value of the heat inputs are shown in the figure. The reduced-order model has 30 states and took 0.01 s to compute the temperature response. The full-order model has 1803 states and took over 49 s to compute the temperature response.

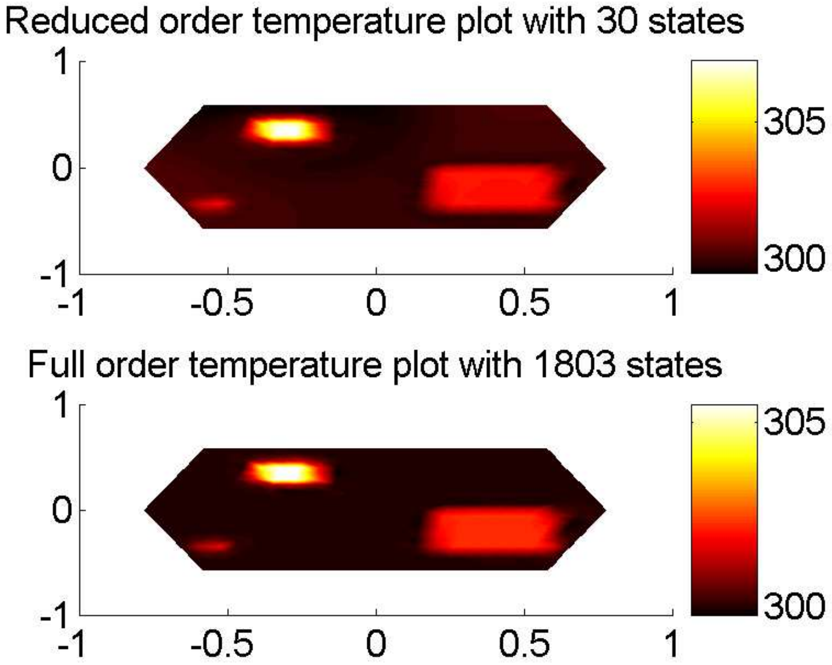


Figure 3.5: Comparison of reduced and full-order temperature plots of the bottom component in Fig.3.3. The temperature profiles are shown for the final time $t = 3$ s.

For each of the two components shown in Fig.3.3, the number of states were reduced from a full-order description of more than 1000 states to a reduced-order description of just 30 states. We computed the errors as the percentage difference between the final and reduced model's temperatures. We defined the error as the percentage difference between the temperatures computed by the FOM and ROM at a given point. The maximum error in temperature profiles was found to be below 1% for both the examples mentioned above.

Chapter 4

Construction of the Device ROM by Interconnecting the Component ROMs

In the previous chapter, we have demonstrated a Krylov subspace technique for creating a reduced-order model for a single component. In this chapter, we show how one can interconnect the component ROMs to form an accurate ROM for the entire device. We split this chapter into four parts - modeling of the heat flux in a control-theoretic format, showing how errors can come about if the errors in the component ROMs amplify at the full device dominant frequency, showing how the small gain theorem can be applied to rectify the above error, and demonstrating our 'reduce-then-interconnect' approach with a numerical example.

We assume the following notation from hereon: the transfer function $g_j(s)$ of component j represents the full-order (unreduced) function that relates its inputs (heating from neighboring components and heat source in component j) to its outputs (temperatures at specific locations on component j). The transfer function $\tilde{g}_j(s)$ is the corresponding reduced-order transfer function. The inputs of both, the reduced and unreduced models, are the same. They represent the exact same heat inputs - due to heat flux from the adjacent components as well as from their own internal heat sources. The outputs, for the reduced as well as the unreduced models, are exactly the same too. It is only the internal mapping between the inputs and outputs of the full (unreduced) model of a component that changes in definition (and size) when compared to the internal mapping of the reduced model of the component (see Eqns. 3.13 and 3.18). To summarize, $g_1(s), g_2(s), ..., g_N(s)$ are the complete (unreduced) transfer functions of the N components (subsystems) of a device (system) and $\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)$ are the reduced transfer functions of the same N subsystems.

4.1 Modeling heat flux in a control theoretical format

As shown in Fig. 4.1 heat flux is exchanged between the component and board only through the top and bottom solder regions. These solder regions are idealized as rectangles in our work, but they can be of any shape without affecting our results.

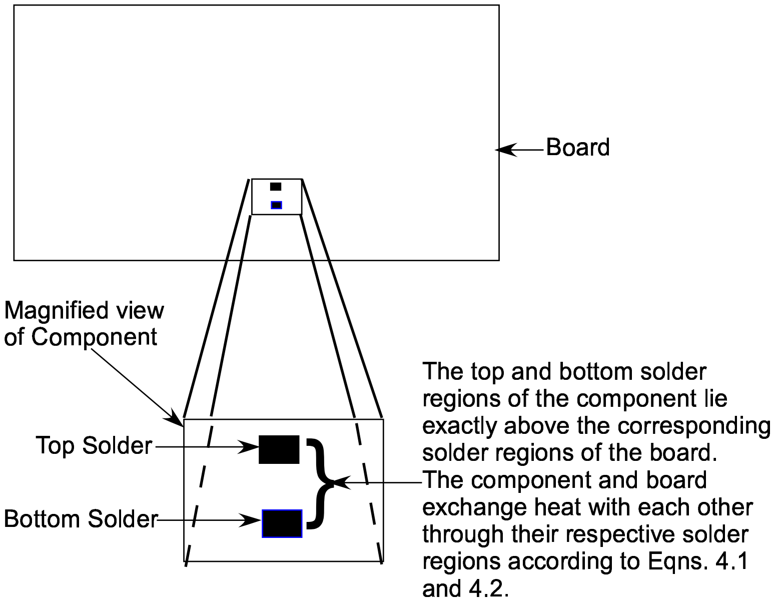

Board
Magnified view of Component
Top Solder
Bottom Solder
The top and bottom solder regions of the component lie exactly above the corresponding solder regions of the board. The component and board exchange heat with each other through their respective solder regions according to Eqns. 4.1 and 4.2.

Figure 4.1: Heat flux between the component and the board is exchanged through the top and bottom solder regions.

The exchange of heat between the component and the board can be represented with the help of the following equations:

$$\begin{aligned} \dot{x}_{Board} &= A_{Board}x_{Board} + B_{Board}[f_{Board}^{Top} \quad f_{Board}^{Bottom}]^T \\ T_{BoardSolder} &= D_{BoardSolder}^T x_{Board} \end{aligned} \tag{4.1}$$

where A_{Board}, B_{Board}, and x_{Board} are the board state space matrices and state vector respectively. The vector $T_{BoardSolder}$ is given by $T_{BoardSolder} = [T_{BoardSolder}^{Top} T_{BoardSolder}^{Bottom}]$ where $T_{BoardSolder}^{Top}$ and $T_{BoardSolder}^{Bottom}$ are the average temperature of the grid points that surround the top and bottom 'solder' regions of the board. The matrix $D_{BoardSolder}^T$ is the corresponding output matrix. The fluxes f_{Board}^{Top} and f_{Board}^{Bottom} are the heat fluxes into the board through the top and bottom solder regions.

For the component, we have

$$\begin{aligned} \dot{x}_{Comp} &= A_{Comp}x_{Comp} + B_{Comp}[f_{Comp}^{Top} \quad f_{Comp}^{Bottom} \quad q]^T \\ T_{CompSolder} &= D_{CompSolder}^T x_{Comp} \end{aligned} \tag{4.2}$$

where A_{Comp}, B_{Comp}, and x_{Comp} are the component state space matrices and state vector respectively. The vector $T_{CompSolder}$ is given by $T_{CompSolder} = [T_{CompSolder}^{Top} T_{CompSolder}^{Bottom}]$ where $T_{CompSolder}^{Top}$ and $T_{CompSolder}^{Bottom}$ are the average temperature of the grid points that surround the top and bottom 'solder' regions of the component. The matrix $D_{CompSolder}^T$ is the corresponding output matrix. The fluxes f_{Comp}^{Top} and f_{Comp}^{Bottom} are the heat fluxes into the component through the top and bottom solder regions and q is the internal heat source in the component.

The relation between the fluxes exchanged through the top and bottom solder regions of the board and component are given by the following equation

$$\begin{aligned} f_{Comp}^{top} &= -\frac{k}{\Delta}(T_{CompSolder}^{top} - T_{BoardSolder}^{top}) \\ f_{Comp}^{bottom} &= -\frac{k}{\Delta}(T_{CompSolder}^{bottom} - T_{BoardSolder}^{bottom}) \end{aligned} \tag{4.3}$$

where k and Δ are the heat conduction coefficient and thickness of a solder connection between the board and component.

This kind of interconnection between the temperatures at grid points on the board and component will yield a linear model if k is assumed constant with temperature. It is possible to represent the above interconnection due to heat flux in a state space format with the help of a connection matrix C which has the following structure. Let α be a vector containing a concatenated list of all possible inputs from all subsystems and β be a concatenated list of all possible outputs from all subsystems. Then the connection matrix C depicts all the interconnections between the various subsystems. The matrix C has element $C_{pq} = 1$ if the q^{th} output β_q is connected to the p^{th} input α_p and $C_{pq} = 0$ otherwise. Physically, this means (referring to Figs. 4.2 and 4.3 below) that if the second output of component A which is (for example) output 2 in the concatenated list of outputs β, is connected to the first input of component B which is (for example) input 3 in the concatenated list of inputs α, then $C_{32} = 1$. If they were not connected then $C_{32} = 0$.

Since the matrix C is defined solely on the basis of the interconnection of the inputs and outputs of the state space model (whether reduced or unreduced) the matrix C remains exactly the same whether we are dealing with the unreduced model or the reduced model. Referring to Figs. 4.2 and 4.3, we can see that component A has

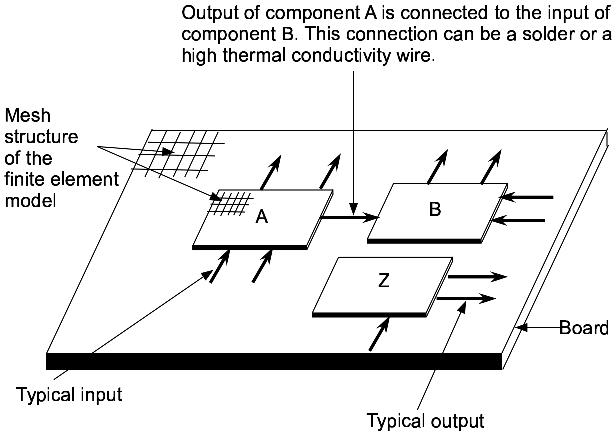

Output of component A is connected to the input of component B. This connection can be a solder or a high thermal conductivity wire.
Mesh structure of the finite element model
A
B
Z
Board
Typical input
Typical output

Figure 4.2: This figure shows a physical interconnection structure between components A,B, and Z. One of the outputs of component A is connected to an input of component B. The components can also be connected to the board, but this connection is not shown here in order to avoid clutter. The input/output numbering is shown in Fig. 4.3, and the corresponding C matrix value is described below.

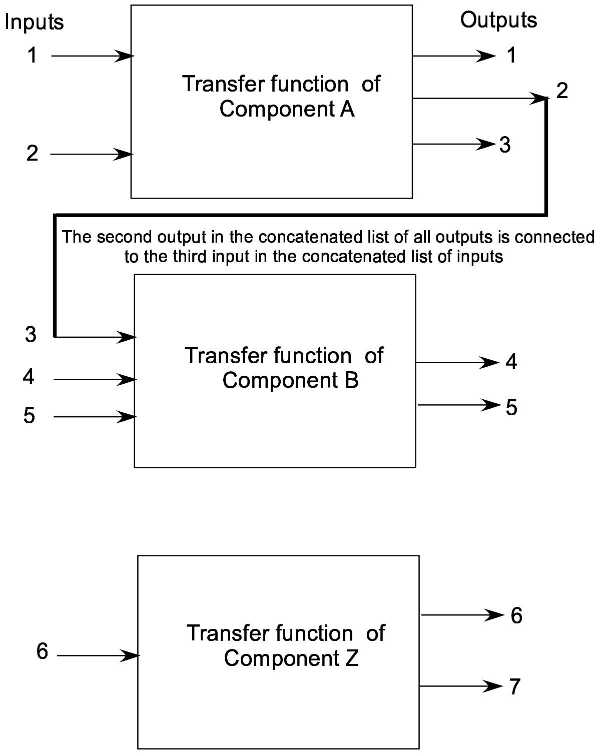

Inputs
Outputs
1
2
Transfer function of Component A
1
2
3
The second output in the concatenated list of all outputs is connected to the third input in the concatenated list of inputs
3
4
5
Transfer function of Component B
4
5
6
Transfer function of Component Z
6
7

Figure 4.3: In the transfer function format, this figure shows that an output of A, the second in the concatenated list of outputs (of all components) is connected to an input of B, the third in the concatenated list of inputs. Hence the connection matrix C has entry C_{32}= 1. If they were not connected then entry C_{32} would be 0.

2 inputs and 3 outputs, component B has 3 inputs and 2 outputs, and component Z has 1 input and 2 outputs. Hence the concatenated list of outputs of all components has 7 elements and the concatenated list of inputs of all components has 6 elements. Hence the interconnection matrix C is a 6×7 matrix. To avoid clutter in Figs. 4.2 and 4.3 we have reduced the number of inputs, outputs, and interconnections between the components and we have also not explicitly shown the fact that each of the components actually exchange heat with (and hence is connected to) the board. In Figs. 4.2 and 4.3, we have shown a connection between components A and B - an output of A (the second in the concatenated list of outputs of all components) is connected to an input of B (the third in the concatenated list of inputs of all components). Thus, element C_{32} of matrix C has value 1 and all the other elements of matrix C have value 0. Thus if we wish to represent the interconnection between the components A,B, and Z in the form of the interconnection matrix C for Fig. 4.3, then

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

It is straightforward to show that this interconnection yields the complete system transfer function $F(s)$ between β (concatenated list of outputs of all components) and α (concatenated list of inputs of all components) as

$$F(s) = G(s)[I - CG(s)]^{-1}. \tag{4.4}$$

where

$$G(s) = \text{diag}[g_1(s), g_2(s), ..., g_N(s)]. \tag{4.5}$$

with $g_1(s), g_2(s), ..., g_N(s)$ being the transfer functions of the N subsystems that constitute the entire subsystem, $G(s)$ being the transfer function made by appending the transfer functions $g_1(s), g_2(s), ..., g_N(s)$ block-diagonally, and $\beta(s) = F(s)\alpha(s)$. Now, if we connect the N reduced subsystems $\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)$ in the same manner we get

$$\tilde{F}(s) = \tilde{G}(s)[I - C\tilde{G}(s)]^{-1}. \tag{4.6}$$

where

$$\tilde{G}(s) = \text{diag}[\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)]. \tag{4.7}$$

and $\beta(s) = \tilde{F}(s)\alpha(s)$.

In the next section we explain why the errors in the component ROMs when interconnected to get the ROM of the device in the form of Eqn. 4.6, can amplify at the whole device dominant frequency. We also show how to place an upper bound on the device error by decomposing the error in the entire device into a sum of weighted component errors using the triangle law. This decomposition and the small gain theorem will enable us to formulate an algorithm that allows us to tune the component ROMs to be accurate at the device dominant frequency so that the error in the ROM of the entire device can be made as small as desired.

4.2 Amplification of component ROM errors after interconnection

Even if the reduced and unreduced models of the individual components match very well i.e., $g_j(s) \approx \tilde{g}_j(s)$ for the operating frequency range of that component, there is still no guarantee that the connected unreduced and reduced models ($F(s)$ and $\tilde{F}(s)$ respectively in Eqns. 4.4 and 4.6 above) of the entire device will match. The reason is that if the errors between $g_j(s)$ and $\tilde{g}_j(s)$ are large enough at the dominant frequencies of the interconnected system, then these errors will multiply (in the feedback sense) in Eqn. 4.6 and the interconnected system of reduced-order models might be inaccurate and even unstable. Hence, if we intend to interconnect the reduced subsystems, then we must make sure that the reduced subsystems are accurate at the dominant frequencies of the interconnected system. The question of how accurate the reduced subsystems must be can, in fact, be specified in the form of a theorem.

This theorem relies largely on an application of the small gain theorem that can be found in a standard controls textbook [122]. Fig. 4.4 aids in getting a better understanding of the basic idea behind the small gain theorem.

Suppose we have a device that is made up of two components that are connected in a cyclical fashion, i.e., each of them has exactly one input and exactly one output, with the output of the first component being connected to the input of the second component and vice versa. Suppose the first component amplifies it's input signal 10 times in its output and the second component diminishes its input signal by 5 times in its output. Then we can see that the composite device (made up of the

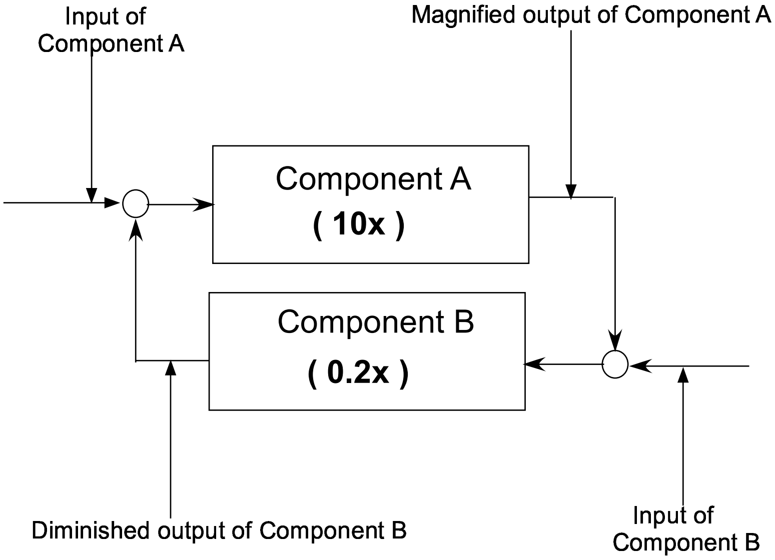


Figure 4.4: Components A and B are connected in a cyclical fashion. Component A magnifies its input by 10 times in its output. Component B diminishes its input by 5 times in its output.

two components) doubles any signal in a single cycle of the signal (a cycle is defined as a passage of the signal exactly once through both the components). Thus, the composite device made up of these two components is an inherently unstable system because of the net magnification of any signal that passes through it. The small gain theorem extends the above example to a device made up of multiple components that are connected to each other in an arbitrary way. It turns out that the question of stability of the device can in fact be answered for an input signal of any particular frequency. The small gain theorem explicitly states a condition (based on the transfer function of the individual components, the input signal frequency, and the interconnection matrix C) that decides whether or not the device is stable [122].

The theorem given below extends a previously derived result [3] (which itself makes use of the small gain theorem). The proof of the theorem is the same as given in [96].

Theorem 4.2.1 *Assume the subsystems $g_1(s), g_2(s), ..., g_N(s)$ are stable, that their full-order interconnection $F(s) = G(s)[I - CG(s)]^{-1}$ is stable, and the model reduced subsystems $\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)$ are also stable (the components in our model do have stable full-order and reduced-order transfer functions and the interconnection of full-order transfer functions is also stable). Then the connected system composed of reduced sub-systems, $\tilde{F}(s) = \tilde{G}(s)[I - C\tilde{G}(s)]^{-1}$ is guaranteed to be stable if*

$$\|[I - CG(s)]^{-1}CE(s)\|_\infty = \sup_{\Omega} \|[I - CG(i\omega)]\|_2 < 1. \tag{4.8}$$

where the supremum is over all frequencies Ω and $E(s) = G(s) - \tilde{G}(s)$, where $G(s)$ and $\tilde{G}(s)$ are defined as diagonal block matrices of subsystem and full-order and reduced-order models respectively (as defined in Eqns. 4.5 and 4.7 respectively).

Proof We can rewrite the ROM transfer function of the entire device $\tilde{F}(s)$ in the following way

$$\begin{aligned}
\tilde{F}(s) &= \tilde{G}(s)(I - C\tilde{G}(s))^{-1} \\
&= \tilde{G}(s)\frac{Adj(I - C\tilde{G}(s))}{det(I - C\tilde{G}(s))} \\
&= \tilde{G}(s)\frac{Adj(I - CG(s) - CE(s))}{det(I - CG(s) - CE(s))}
\end{aligned} \tag{4.9}$$

Since all the component ROMs $\tilde{g}_i(s)$ are assumed stable, we have that $\tilde{G}(s) =$

$diag(\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s))$ is stable. Likewise, the adjoint matrix $Adj(I - C\tilde{G}(s))$ is composed of an addition and multiplication of stable transfer functions and is hence stable. Hence $\tilde{F}(s)$ is stable *iff* $det(I - CG(s) - CE(s))$ has no roots in the closed right half plane (RHP).

But $I - CG(s) - CE(s) = (I - CG(s))(I - CG(s)^{-1}CE(s))$ and $det(AB) = (det(A))(det(B))$, so it is enough to show that $det(I - CG(s) - CE(s)) = det((I - CG(s))) det((I - CG(s)^{-1}CE(s)))$ has no roots in the RHP.

We know that $det((I-CG(s)))$ has no roots in the RHP by the assumption that $F(s) = G(s)(I-CG(s))^{-1}$ is stable. We also know that $det(I-P(s))$ has no roots in the RHP by the small gain theorem because $||P(s)||_\infty = ||(I-CG(s))^{-1}CE(s)||_\infty < 1$.

The relevant statement of the small gain theorem is that $det(I - P(s))$ has no roots in the RHP *iff* $inf(\underline{\sigma}(I - P(s))) \neq 0$ for all $s \in RHP$ (where $\underline{\sigma}(X)$ is the minimum singular value of the matrix X), but the infimum is bounded below by $1 - ||P(s)||_\infty \geq 0$ and so it cannot be zero in the RHP. □

4.2.1 Mitigation of the device ROM error by frequency weighting of component ROMs

We are basically concerned about reducing the infinity norm of the error between the full-order and reduced-order system, i.e. we intend to minimize $||\Xi(s)||_\infty = ||F(s) - \tilde{F}(s)||_\infty$. Now, it can be shown by extending results in [3] that up to a first order approximation

$$\begin{aligned} \|\Xi(s)\|_\infty &\approx \|U(s)E(s)V(s)\|_\infty, \\ U(s) &\approx \tilde{U}(s),\ and \\ V(s) &\approx \tilde{V}(s). \end{aligned} \tag{4.10}$$

where

$U(s) = G(s)[I - CG(s)]^{-1}C + I$, $V(s) = [I - CG(s)]^{-1}$;

$\tilde{U}(s) = \tilde{G}(s)[I - C\tilde{G}(s)]^{-1}C + I$, $\tilde{V}(s) = [I - C\tilde{G}(s)]^{-1}$ and

$E(s) = G(s) - \tilde{G}(s)$.

The first order approximation in Eqn. 4.10 can be shown by realizing the equivalence of either of the 2 substitutions: either we replace $\tilde{G}(s)$ by $G(s) - E(s)$ or we replace $G(s)$ by $\tilde{G}(s)+E(s)$. In the first case, we get that (we omit the frequency dependence to avoid clutter)

$$\begin{aligned} \Xi &= G(I - CG)^{-1} - \tilde{G}(I - C\tilde{G})^{-1} \\ &= G(I - CG)^{-1} - (G - E)(I - CG - E)^{-1} \\ &= G(I - CG)^{-1} - (G - E)(I + (I - CG)CE)^{-1}(I - CG)^{-1} \end{aligned} \tag{4.11}$$

If the components are reduced so that the quantity $\Delta = (I - CG)CE$ is kept small, then we have that $(I + \Delta)^{-1} = I - \Delta + O(||\Delta||^2)$. Hence we have that

$$\Xi = (G(I - CG)^{-1}C)E(I - CG)^{-1} + O(||\Delta||^2) \tag{4.12}$$

If instead, we had used the substitution, $G = \tilde{G} + E$, we would similarly find that

$$\Xi = (\tilde{G}(I - C\tilde{G})^{-1}C)E(I - C\tilde{G})^{-1} + O(||\Delta||^2) \tag{4.13}$$

Either way, the sub-system errors $E(s)$ are, to the first order in $E(s)$, being amplified by the entire system frequency response $(I - CG)^{-1}$

$$\begin{aligned} \Xi(s) &\approx (G(s)(I - CG(s))^{-1}C)E(s)(I - CG(s))^{-1} \\ &\approx (\tilde{G}(s)(I - C\tilde{G}(s))^{-1}C)E(s)(I - C\tilde{G}(s))^{-1} \end{aligned} \tag{4.14}$$

With the help of this first order approximation, we can use the triangle inequality to show that

$$\|\Xi(s)\|_\infty \leq \sum_{p=1}^{p=N} \|U_{Ip}(s)\epsilon_p(s)V_{pJ}(s)\|_\infty. \tag{4.15}$$

where $U_{Ip}(s)$ and $V_{Ip}(s)$ denotes the Ip^{th} block of $U(s)$ and $V(s)$ respectively, and $\epsilon_p(s) = g_p(s) - \tilde{g}_p(s)$.

Thus any model reduction effort that hopes to capture the entire interconnected device behavior will have to keep the individual component errors small at the natural frequencies of the entire device. Furthermore, the amount of component to device error amplification is adequately captured by a reduced order estimate of the entire device dynamics $(I - C\tilde{G}(s))^{-1}$. It is not necessary to know the full order estimate $(I - CG(s))^{-1}$ exactly in order to know the frequencies at which the component ROMs are to be minimized as computing the dominant frequency range from the FOM can be time consuming with respect to the rest of the reduced order modeling process (since the precise picture of the dominant frequency range would involve computing the eigenvalues of $(I - C\tilde{G}(s))^{-1}$) .

We can use the intuition from the physics of the heat conduction problem to decrease the time taken for the construction of the component reduced order models. During heat dissipation, the dominant poles of the system are negative and close to zero. Hence, the behavior of the full-order system is mainly determined by these dominant poles and we do not need to evaluate the full-order system's behavior in the entire frequency range. We only need to compute the dominant natural frequencies of the entire system and ensure that the reduced-order models of the subsystems are accurate in an appropriate dominant frequency range Δ (say, the low frequency range which is spanned by the two most dominant poles of the system as illustrated schematically in Fig. 3.2).

Thus, we have to model reduce the sub-systems $g_k(s)$ in such a way so as to keep $\|U_{Ik}(s)\epsilon_k(s)V_{kJ}(s)\|_\Delta$ (the error in the dominant frequency range Δ) small for all k, where $U(s) = G(s)[I - CG(s)]^{-1}C + I$, $V(s) = [I - CG(s)]^{-1}$, and $\epsilon_k(s) = g_k(s) - \tilde{g}_k(s)$. Since $U(s) \approx \tilde{U}(s)$ and $V(s) \approx \tilde{V}(s)$, we can instead minimize $\|\tilde{U}_{Ik}(s)\epsilon_k(s)\tilde{V}_{kJ}(s)\|_\Delta$. In order to do this, we have to compute the entire system's frequency response, but this needs to be done only once (even a rough estimate of the frequency response of the entire system is enough). In fact, if the designer has knowledge about the dominant frequencies for a particular interconnection and if the arrangement of the components is not drastically changed for the next design iteration, then the dominant frequency range of the new interconnected system will be approximately the same as that of the previous system. In that case, the same set of reduced-order models of the components of the system may be used and interconnected with the new interconnection matrix (since the reduced-order models have already been computed in such a way that their interconnection yields an accurate behavior in the dominant frequency range). In our trials with device architectures that were not drastically different from each other we did observe a good agreement between the reduced and unreduced models of the device using just a single library of reduced-order models. In the industry, the placement of components on a board are primarily determined on the basis of VLSI design and hence the range of architectures that a heat transfer specialist can optimize over can be expected to be similar between different design iterations.

In the model reduction literature, ensuring that the reduced-order models are accurate at particular frequencies is termed as ‘frequency weighted model reduction’. According to Krylov based frequency weighted model reduction theory [32], a rule of thumb to ensure frequency weighting is to choose the interpolation points to be logarithmically spaced in the dominant frequency range.

In summation, the algorithm that we used to perform model reduction of the individual subsystems and interconnect the reduced-order models, which is based on Theorem 4.2.1 (that was originally proved by Anderson [3] in a different context) is given in algorithm 2.

In brief, the above algorithm first estimates the dominant frequency range of the complete system, computes initial (unweighted) reduced-order models for each component and then iteratively refines these reduced-order models by solving the minimization problem mentioned in step 4 of the algorithm. In each iteration of the algorithm, we add interpolation points for each component’s reduced-order model (as mentioned before, one interpolation point in each logarithmic decade of the desired frequency range). A stopping criterion for the reduced-order model can be applied by requiring that the total error between the full-order and reduced-order system is less than a desired value. In our simulations, we needed 4 iterations for

Algorithm 2 Creation and stable interconnection of reduced-order models

1. Estimate the dominant natural frequencies of the complete (unreduced) system and choose the dominant frequency range Δ of the system based on the spacing between the eigenvalues of the system. Choose appropriate interpolation points based on Δ.

2. Using the Krylov reduction algorithm 1, compute unweighted reduced-order models for each of the N subsystems. Call these initial reduced-order models as $\tilde{g}_1^0(s), \tilde{g}_2^0(s), ..., \tilde{g}_N^0(s)$.

3. Estimate $\tilde{U}(s)$ and $\tilde{V}(s)$ (refer to Eqns. 4.10 and 4.15), the model reduction projection matrices, based on $\tilde{g}_1^0(s), \tilde{g}_2^0(s), ..., \tilde{g}_N^0(s)$, namely $\tilde{U}^0(s) = \tilde{G}^0(s)[I - C\tilde{G}^0(s)]^{-1} + I$ and $\tilde{V}^0(s) = \tilde{G}^0(s)[I - C\tilde{G}^0(s)]^{-1}$.

4. Find $\tilde{g}_1^1(s), \tilde{g}_2^1(s), ..., \tilde{g}_N^1(s)$ by solving the frequency weighted Krylov subspace problem: min $\|\tilde{U}_{Ik}^0 \epsilon_k(s) \tilde{V}_{kJ}^0\|_\Delta$ where the index k runs from 1 to N and $\epsilon_k(s)$ is the error transfer function between $g_k(s)$ and $\tilde{g}_k(s)$. Frequency weighting in the Krylov subspace method is done by choosing appropriate interpolation points to lie in the required frequency range. The initial choice of interpolation points is arbitrarily chosen in the dominant frequency range, but we have to vary the choice of interpolation points in the dominant frequency range so that the minimum of $\|\tilde{U}_{Ik}^0 \epsilon_k(s) \tilde{V}_{kJ}^0\|_\Delta$ is reached for all components k.

5. Repeat steps 3 and 4 until an acceptable set of reduced subsystems (that reduces the error between the interconnected FOM and ROM below a desired value) is found.

satisfactory results.

In the next section, we apply the proposed frequency weighted model reduction of the sub-systems and their interconnection mentioned above to a numerical example in which 20 components are connected to and hence exchanges heat flux with a board.

4.3 Numerical example of heat transfer between 20 reduced order components to a board.

We now demonstrate an example for model reduction done on a system of 20 components connected to a board. The reduced-order models for the complete system was done using Algorithm 1 (for creating ROMs of individual components) and Algorithm 2 (for creating a stable and accurate interconnection of the component ROMs). For our example, we have modeled a board with 20 components connected to it as shown in Fig.4.5 below.

There are 20 components on the board (shown above in Fig.4.5). Each of the components exchanges heat with the board and the only mode of heat transfer is conduction. Each of the components are joined to the board by two solder connections as shown in Fig.4.5 inset above. These solder connections were a cluster of 5-6 finite element nodes near the top and bottom edges of each component. They model a conducting surface (like thermal paste) that might be typically sandwiched between the component and the board, but for this example we will term this cluster as a “solder” connection. For an actual component, the number of nodes in the

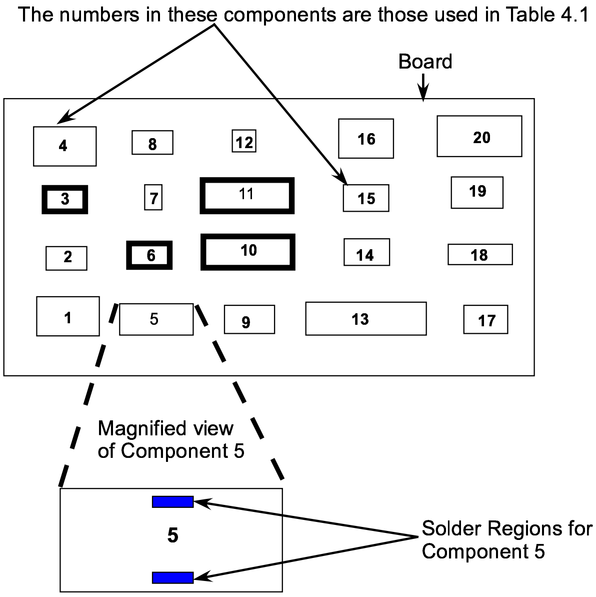

The numbers in these components are those used in Table 4.1
Board
4
8
12
16
20
3
7
11
15
19
2
6
10
14
18
1
5
9
13
17
Magnified view of Component 5
5
Solder Regions for Component 5

Figure 4.5: Arrangement of the 20 components on the board. The components are numbered, and the corresponding step heat inputs applied to the components are mentioned in Table 4.1 below. The bold boundaries on components 3,6,10, and 11 denote that each of those components have all 4 of their boundaries kept at a constant temperature of 300 K.

cluster could be increased or decreased as per the situation. This decision should be made while building up the library of component ROMs (and before interconnecting the components). Each cluster in the solder region of the component exchanges heat with the below cluster in the solder region of the board. All the nodes in each solder cluster of a component were modeled so as to exchange the same heat flux with the solder cluster of the adjacent component. This cluster is an idealization of a thermal contact between a component and the board. The heat flux for an entire cluster (solder contact) is modeled as just one input in the state space format and this kind of contact between the board and the component was captured as an input-output relationship with the connection matrix C explained in Eqn. 4.4. This kind of interconnection between the components and the board reduced the effective number of inputs to just two for each component in addition to the source that generates heat in the component (which amounts to 3 inputs per component).

Each of the idealized components as well as the board were modeled as being made of silicon and having constant material properties that were independent of temperature. The physical properties of all the components and the board are the same as silicon (as mentioned for the examples in Fig. 3.3). Components 3,6,10 and 11 (labeled in Fig.4.5) have each of their 4 boundaries at a constant temperature of 300 K. The boundaries of the rest of the 16 components as well as the board, are thermally insulated. For ease of notation, the step (heat) inputs that have been applied to the components (labeled in Fig.4.5) have been denoted as follows : $q_i = u_i \cdot \mathbf{1}(t - a_i)$ which denotes that $q_i = 0$ for $t \leq a_i$, and $q_i = u_i$ for $t > a_i$. The heat inputs applied all 20 components are listed in Table 4.1. In Fig. 4.6, a figurative explanation of the symbols u_i and a_i (used in Table 4.1) is shown for the particular example of the step heat input applied to Component 1.

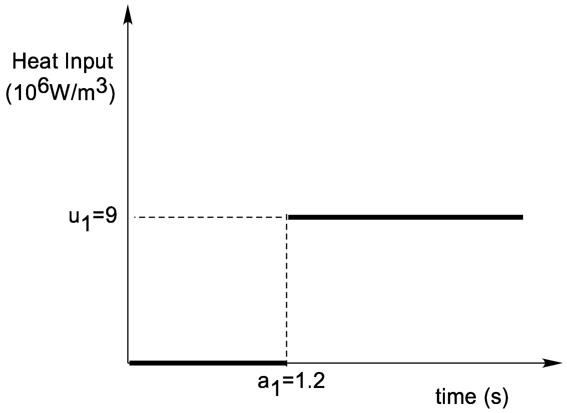


Figure 4.6: This figure shows the step heat input applied to Component 1. Table 4.1 gives the values of the step heat inputs for all 20 components.

The interconnections between the components (subsystems) were modeled in the following way. The value of the system matrices A,B,E, and D for each of the subsystems was individually extracted from FEMLAB (a finite element solver). Within FEMLAB, a linear time-dependent solver was used for simulating this problem. Each of the components was discretized with mesh sizes of the order of around 1000 nodes per component. We extracted complete state space models of each of the components with the inputs being the heat fluxes into each component. The state space matrices were obtained by linearizing the finite element model around a nominal temperature of 300 K. Though we created a reduced-order model for each component, we did not create a reduced-order model for the board. The reason is that when we initially connected the reduced-order model of the board to the reduced-order model of the components, the whole system's temperature response showed an increased amount of error (around 7%) in certain areas on the board. We consider this to be a result of an inherent numerical limitation in the Krylov subspace method (Algorithm 1), which to the best of our knowledge has not yet been resolved. A good explanation of this limitation can be found in [32]. Basically, in each inner loop of Algorithm 1, we add more information about the full-order model into the transformation matrix U, by appending new columns u_m to U. Now, if there are many inputs in the state space model (i.e. a B is a "fat" matrix, like in the state space model of the board in our numerical example, which has 40 inputs, 2 from each component connected to it), the new columns (u_m) that are added to U, in successive iterations of the Krylov subspace algorithm (Algorithm 1) can, because of limitations in computational accuracy, lie in the subspace of the previously computed columns of U. This makes the matrix U have less than full rank. When such a (less

Table 4.1: Heat Inputs applied to the 20 components

Component	u_i (10^6 W/m^3)	a_i (s)
1	9	1.2
2	2	1.5
3	7	2.2
4	6.4	0.9
5	8.8	1.3
6	4.3	1.3
7	8.1	1.4
8	5.6	2.7
9	7.7	0.4
10	5.0	2.2
11	8.4	2.4
12	7.7	0.8
13	4.9	0.9
14	8.5	0.3
15	6.2	0.9
16	8.9	0.4
17	9.7	0.04
18	3.6	0.1
19	9.1	0.4
20	7.0	0.6

than full rank) transformation matrix is used for reduced-order modeling, it results in inaccurate reduced-order models.

Hence, we chose to connect the full-order model of the board to the reduced-order model of the components and avoid the above problem. We would like to stress that using a full-order model of the board has a one time, fixed computational cost. It is basically the total number of components on the board that influences the total number of states (and hence, computational cost) in the interconnected system reduced model. Specifically, in our example the full-order model of the board has 1429 states and the full-order model of each of the 20 components has around 1000 states and thus the full-order model of the interconnected system has 23109 states. The reduced-order model of each of the 20 components has 30 states. Thus even though, we connect the reduced-order models of the components to the full-order model of the board, the reduced-order model of the interconnected system has only 2029 states (20 x 30 + 1429). As the number of components on the board increases, the difference between the number of states in the FOM and ROM of the interconnected system will roughly scale by the total number of components.

We connected the state space systems of the components on MATLAB (with the connection matrix C) and computed the dominant frequency range of the entire system. The dominant eigenvalues were clustered in the $0 - 10$ rad/s range (this is step 1 of Algorithm 2). We chose to have 3 interpolation points at 0.1 rad/s, 2 rad/s and 5 rad/s as the initial guesses for the interpolation points (with 1 moment to be matched at each interpolation point) and allowed them to vary in the dominant frequency range as mentioned in Algorithm 2.

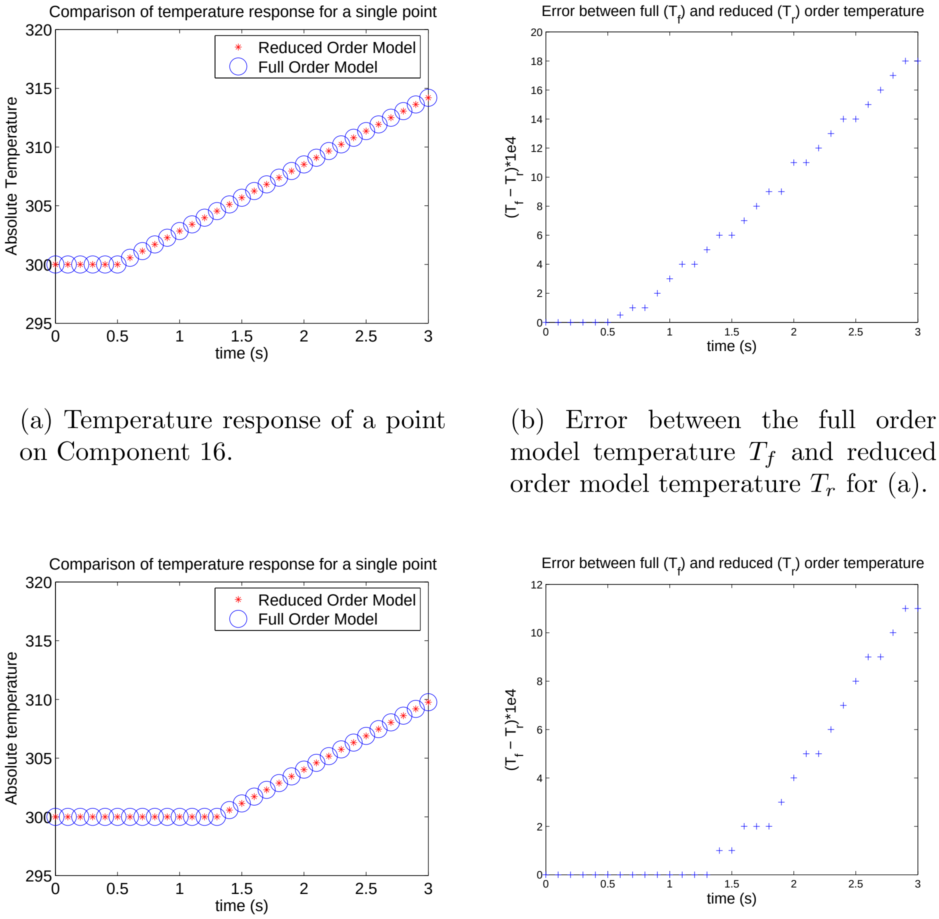
In our entire simulation, we required 4 iterations of Algorithm 2 which took close to 40 minutes on a 2.3 GHz AMD Opteron processor (with 64 bit MATLAB). The final interpolation points were chosen as $\sigma_1 = 1\ rad/s$, $\sigma_2 = 10^{-1}\ rad/s$, $\sigma_3 = 10^{-2}\ rad/s$, $\sigma_4 = 10^{-3}\ rad/s$, and $\sigma_5 = 10^{-4}\ rad/s$. After the reduced-order models were computed, it took less than 5 minutes to connect the reduced-order models of the 20 components (and the full-order model of the board) to form reduced-order models of the entire system.

The resulting temperature response at 4 points of the system are shown in Figs. 4.7 and 4.8 below. Each plot shows results for the full-order as well as the reduced-order model. The plots for the FOM and ROM response are indistinguishable because the errors are very small.

In Figs. 4.9 and 4.10, we have provided a plot of the temperature profile for the entire system (components connected to the board). Fig.4.9(a) and Fig.4.9(b) show the reduced-order and full-order plots of the components respectively. Figs.4.10(a) and Fig.4.10(b) show the reduced-order and full-order plots of the board respectively. The plots of the temperature profile for the 20 components are shown separately (below) the plot of the temperature profile for the board, because the temperature rise in the components is higher than that of the board.

The full-order model computations are very expensive, especially in terms of memory involved. For the reduced-order models we required 340 MB memory. This amount of memory usage is largely due to the *full-order model of the board*, which is being connected to the reduced-order models of the 20 components. As the number of components on the board increase, the (fixed) memory requirements for accomodating the full order model of the board will be much less relative to the memory requirements of the components in the device ROM. For creating the interconnected system model with the full-order systems, we had to make use of the sparse structure of the system matrices (see Eqn.4.6 and Eqn.4.7), which are mostly block diagonal except for the few off block-diagonal terms corresponding to the solder connections. The plot was created by computing the temperature for a few evenly spaced points with the respective (full or reduced) order models in MATLAB, and interpolating for the values in between.

For each of the components, we can see that most of the component is at roughly the same temperature except for the two solder regions near the top and bottom of the component where the component exchanges heat flux with the component. The solder regions are cooler than the rest of the component because the heat flux flows out of the component and into the board. In the temperature plots of the board, one can see the regions that are below the component have a higher temperature then the surrounding. For the initial set of 3 interpolation points (before Algorithm 2 was applied for interconnecting the board to the components), the reduced-order plots of the board temperature did not match that of the full-order plot of the board. However, one can see that in Figs.4.9 and 4.10 (after Algorithm 2 was used, and the number of interpolation points for the component ROMs was increased to 5), there is a very accurate match between the contours of the hot spots on the board on the full-order model and the contours on the reduced-order model. We computed the errors as the percentage difference between the full and reduced model's temperatures. The maximum error of the temperature response on



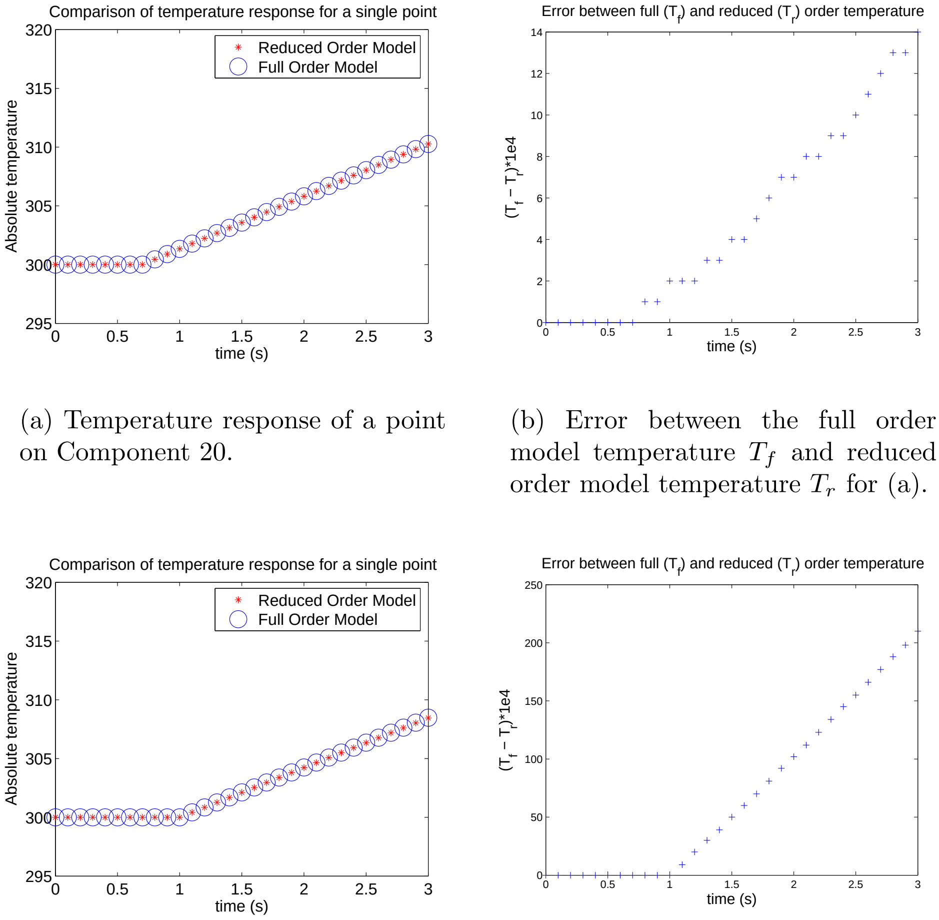
(a) Temperature response of a point on Component 16.

(b) Error between the full order model temperature T_f and reduced order model temperature T_r for (a).

(c) Temperature response of a point on Component 1.

(d) Error between the full order model temperature T_f and reduced order model temperature T_r for (c).

Figure 4.7: Comparison between the full system and reduced system temperature response of two different points on the device from $t = 0$s until $t = 3$ s. The corresponding error between the full and reduced order model temperature response is magnified by a factor of 10^4 and is shown in (b) and (d) respectively.



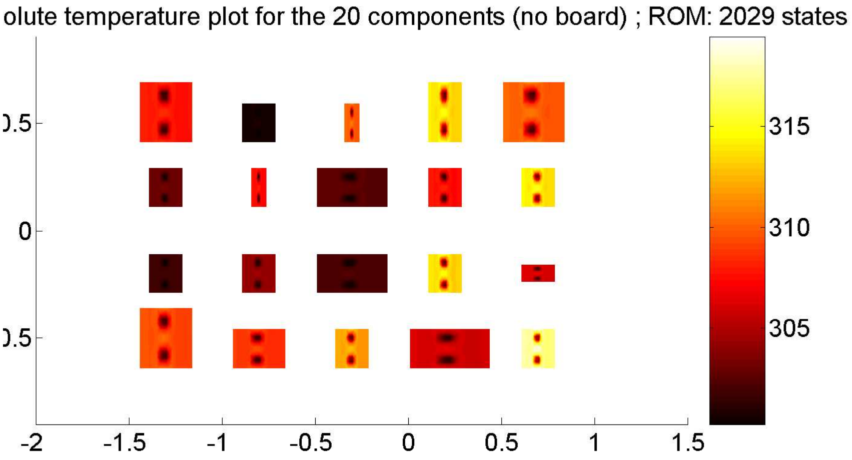
(a) Temperature response of a point on Component 20.

(b) Error between the full order model temperature T_f and reduced order model temperature T_r for (a).

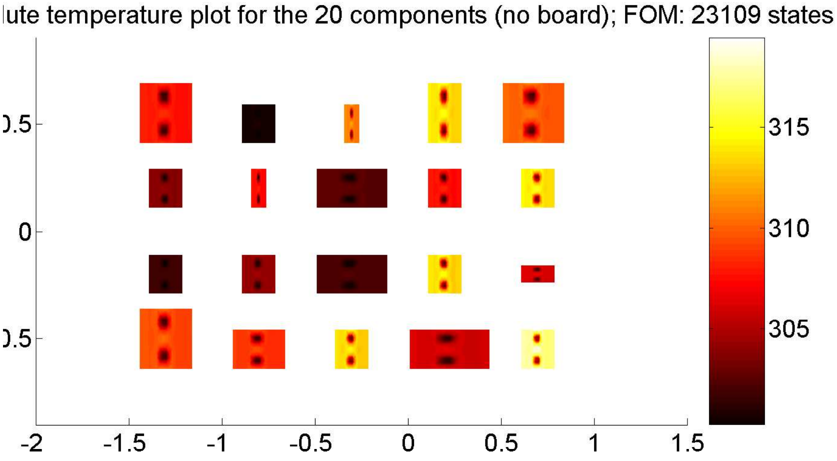
(c) Temperature response of a point on Component 4.

(d) Error between the full order model temperature T_f and reduced order model temperature T_r for (c).

Figure 4.8: Comparison between the full system and reduced system temperature response of two different points on the device from $t = 0$s until $t = 3$ s. The corresponding error between the full and reduced order model temperature response is magnified by a factor of 10^4 and is shown in (b) and (d) respectively.

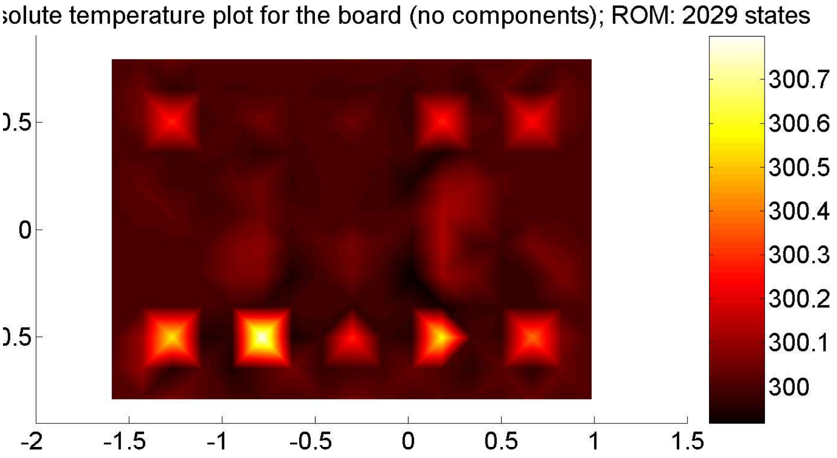


(a) Absolute temperature plot of the reduced-order model of the components. The two "cold" spots on each component correspond to the solder region of the component. The two solder regions of each component do not have any direct heat source like the rest of the component (the component heat source values are mentioned in Table 4.1), which is the reason why it is at a lower temperature than the rest of the component.

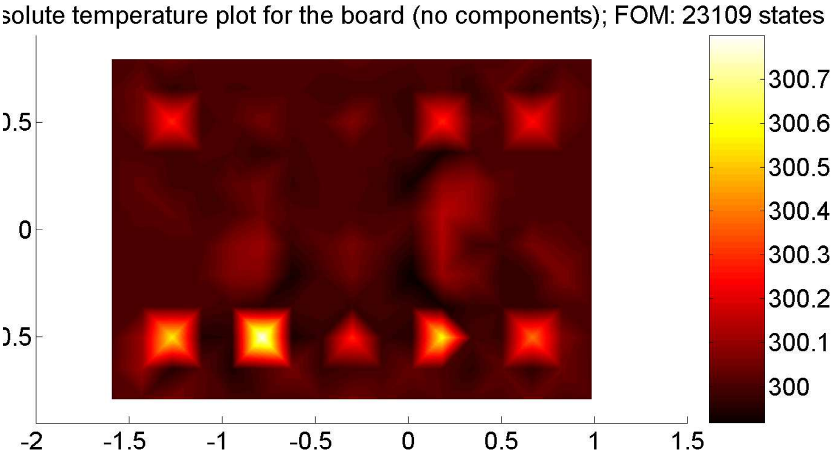


(b) Absolute temperature plot of the full-order model of the components.

Figure 4.9: Comparison between the full system and reduced system temperature profiles of the components at the end of $3s$.The components are shown separately from the board because the temperature rise in the components are much larger than on the board. The complete system FOM (board + component) has 23109 states, while the ROM has 2029 states.



(a) Absolute temperature plot of the reduced-order model of the board.



(b) Absolute temperature plot of the full-order model of the board.

Figure 4.10: Comparison between the full system and reduced system temperature profiles of the components at the end of $3s$. The components are shown separately from the board because the temperature rise in the components are much larger than on the board.

various parts of the system was less than 1%. The number of states for the entire system was reduced from an original of around 23109 to 2029. In both, the full-order and the reduced-order system, the board contributes 1429 elements. In the reduced-order system, each of the components only have 30 states as compared to more than 1000 states in the full-order model. Though there is a fixed cost due to the includion of the full order model of the board, we can see that as the number of components on the board increase, the savings in computational time and memory are very significant in the reduced-order model of the system when compared to the full-order model.

Table 4.2 presents a comparison of reduced order modeling techniques that have been used for thermal simulation of microelectronic components. Each of the other three modeling approaches - resistor modeling, Pade approximation, and Arnoldi-based (single ROM for the entire device) - were created with a specific need. The resistor modeling approach is intuitive, but the relative lack of feature resolution of this approach can result in larger errors (as high as 7 % in [90]). While the other two approaches - Pade [52] and Arnoldi-based [18] - have a very high accuracy, they both rely on modeling the entire device with a single reduced order model. In our simulations, the initial iterations - which consists of computing the frequency weighted reduced-order model for all the components - took 40 minutes. The connection of the reduced-order models of the components took less than 5 minutes, and after that, computing the temperature responses took 55 seconds. Hence, the temperature distribution for new system architectures can be computed in 5 min 55 seconds. For the connection structure in our problem, the error was less than 1% as compared to the full-order system. The main advantage of our models is that we can interconnect many reduced-order models instead of having to compute a reduced-order model of the entire system for each different component layout.

Table 4.2: Comparison of Model Reduction Techniques

Modeling method	**FOM/ROM number of elements (physical problem)**	**ROM run time**	**Error (compared to)**	**Recreate device ROM from ground up?**
Resistor modeling [**90**]	ROM:2171 (9 components; 12 varying heat sources)	25 min	less than 7%(ANSYS)	YES
Pade approximation [**52**]	ROM:2375 (Voltage Regulator)	240 s (as compared to 3110 s for FOM)	Almost exact (FOM)	YES
Arnoldi based single ROM [**18**]	FOM:30000 ROM: 85 (17 components on a board)	10 min	less than 1%(FOM)	YES
Algorithm 2 in this chapter	FOM:23109 ROM:2029 (20 components on a board)	less than 1 min after the initial interconnection of the system	less than 1%(FOM)	NO

Chapter 5

Discussion and Future Work for Interconnection of Reduced Order Models

We have demonstrated the use of an efficient Krylov subspace method to create reduced-order models of different components of a thermal conduction network of an electronic device. We have also described a novel method of assembling various such reduced-order models in order to accurately compute the entire system behavior. The size of the system in the numerical example was reduced from 23109 states to approximately 2029 states, which reduced the time for the simulation to less than 1 minute (once the initial iterations for computing the frequency weighted reduced-order models of the components have been completed). A designer can use the method that we propose in this paper to create a library of reduced-order component models and then connect them together to achieve reduced-order system models.

The mathematical format for interconnection that we presented in this paper can be effectively applied to model the kind of physical interconnection between components that we have shown in Sec. 4.3, where the the heat conduction coefficient does not change appreciably with temperature. One of the future extensions of the work presented here would be to account for variability in the heat conduction coefficient. Such kind of 'model reduction with parametric variation' has been done using multi-parameter moment matching using Krylov subspace techniques for the problem of accurate parametric reduction for addressing the variability of integrated circuit interconnect performance [54].

In a finite element package like ANSYS or FEMLAB, the kind of reduced-order results described in this paper can be offered as an option to the designer. Once the designer has modeled a particular component of the device, he or she can store the reduced-order description of the component in a library along with those of other components. The designer can then connect that library of components to create cheap but accurate system models. The designer would find it convenient if there was a GUI incorporated in the software that would help him or her connect the (heat) inputs of a particular component to those of another component, instead of having to manually compute the interconnection matrix C. When faced with a new interconnection structure, the designer should first check whether the existing library contains component reduced-order models that satisfy the stopping criterion in algorithm 2. If it does meet the criterion, then the designer can use the same library, else he or she can use the component ROMs in the existing library as a starting guess and use Algorithm 2 to compute new component reduced-order models.

In Algorithm 2, one also needs to have an approximate idea of the dominant frequency range of the system model for different interconnections. One can potentially use a result from linear algebra to do this cheaply. Briefly, the idea is that if one has access to the dominant eigenvalues of the component reduced-order models that are being interconnected, the system's state space matrix A_{system} (which has only got a few *known* off block-diagonal terms corresponding to the entries due to the solder joints) can be treated as a perturbation of a matrix that is formed by block-diagonally appending the component matrices, A_i. Since we only need a rough approximation of the dominant frequency range, this can be computed by applying Gershgorin's theorem [30] to the block-diagonal matrices ($A_i, i = 1$ to N) in order to get the range in which the dominant eigenvalues will lie. Even a rough approximation of this range will suffice, because when creating ROMs of the component, it is enough to add interpolation points that are logarithmically spaced in the required frequency range.

In our simulations, we have modified the interconnection structure slightly and observed that one can still use the same library with accurate results, but the following problems is still open: in an optimization run that searches for maximally heat dissipative architecture for the device, by how much can we modify the interconnection structure (which decides the architecture) without having to also modify the library of reduced-order component models that we use to create the reduced order model of the device?

The final aim is to couple these kinds of conduction ROMS to convection and radiation ROMs (convection has been model-reduced using proper orthogonal decomposition in many applications including flows past cavities [88] and identifying coherent structures in turbulence [99]) and provide a complete design capability for a heat transfer specialist using a similar 'reduce-then-interconnect' approach. We think that the answers to the above two questions - allowable variation in interconnection and ROMS for convection and radiation - are important topics for future research in order that designers be able to apply model reduction techniques to practical problems.

Chapter 6

Introduction to Isoelectric focusing: Definitions and Physics

For every kind of analysis of biological material, be it genomic, proteomic, or glycomic, the base material (DNA, proteins, or carbohydrates respectively) is always found in an impure state when it is extracted from the tissue. One needs to be able to separate this base material from the other impurities in order to be able to perform any kind of further analysis. In this part of the thesis, we will concentrate on one such technique - the isoelectric focusing (IEF) process - which is a popular technique that is used to separate proteins from the other constituents of the extracted material. In this chapter, we will describe the IEF process and make a few assumptions to keep this discussion at the level of an overview of the IEF physics. In the next chapter, we will justify those assumptions in detail, describe the governing equations for IEF, and our simulations of IEF physics.

6.1 Separation Techniques

A complete separation of a mixture of chemical constituents can be represented [28] by

$$(a + b + c + d + ...) \longrightarrow (a) + (b) + (c) + (d) + ... \tag{6.1}$$

where the parenthesis represent different regions of space and the letters a, b, c, d,... represent the individual constituents occupying those regions. A group of constituents that are originally intermixed, are forced into different spatial locations by the process of *separation.*

There are about 20 basic techniques of separation [29]. Basic techniques are either named after the underlying physical phenomena - adsorption, crystallization, ion exchange, diffusion etc. - or a distinct form of operation - chromatography, distillation, dialysis, field flow fractionation etc. Sometimes, different separation techniques are used simultaneously for achieving separation. In Fig. 6.1 [36], we have an example of a '2-dimensional' separation technique in which 3 different proteins are separated along two orthogonal axes. One of the axes uses IEF for separation and the other uses capillary electrophoresis for separating the proteins.

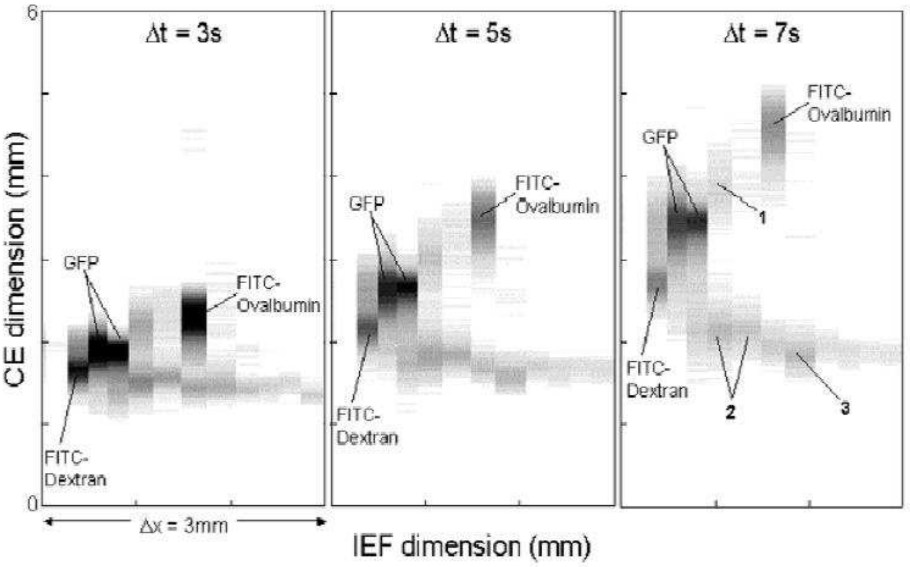


Figure 6.1: Separation of three different proteins - GFP, FTC-Ovalbumin, FITC-Dextran - at three time intervals as reported in [36]. The proteins are separated along two orthogonal axes, which use IEF and capillary electrophoresis respectively as the separation techniques.

The separation technique investigated in this thesis is isoelectric focusing (IEF), which is used to isolate proteins in a mixture. The physics involved is that of electrophoresis which is the movement of charged particles in a heterogenous fluid under the influence of an electric field.

6.2 Separation of proteins

Proteins are large organic compounds made of amino acids arranged in a linear chain and joined by peptide bonds (bond formed when the carboxyl group of one molecule reacts with the amino group of another molecule [1]). They are considered as the building blocks of nature [1], and participate in every process within cells. Proteins are involved in varied functions like catalyzing biochemical reactions in metabolism, forming a scaffolding that maintains cell structure, cell signaling, digestion etc.

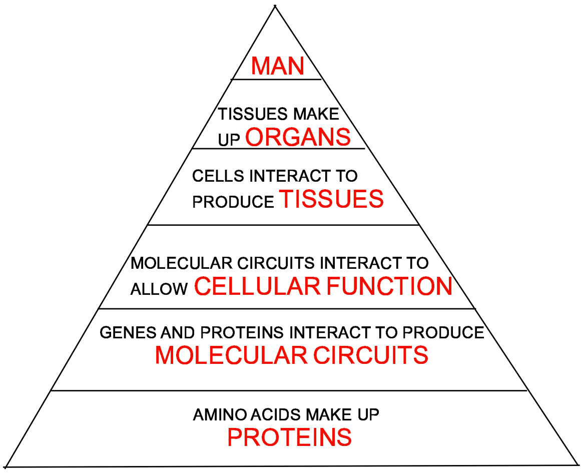

MAN
TISSUES MAKE UP ORGANS
CELLS INTERACT TO PRODUCE TISSUES
MOLECULAR CIRCUITS INTERACT TO ALLOW CELLULAR FUNCTION
GENES AND PROTEINS INTERACT TO PRODUCE MOLECULAR CIRCUITS
AMINO ACIDS MAKE UP PROTEINS

Figure 6.2: Proteins are building blocks of nature.

The process of extraction of proteins from cells begins with cell lysis, in which a cell's membrane is disrupted and its internal contents released into a solution known as crude lysate. The resulting mixture is purified in a centrifuge which fractionates the various cellular components into fractions containing soluble proteins, membrane lipids, cellular organelles and nucleic acids. The proteins formed as a result are salted out (taking advantage of the amino acid structure). After the salting out process, we get a solution that primarily contains a mixture of different proteins.

This is where the separation techniques prove useful. In order to to understand any significant properties of proteins, for example, how structural changes of an individual protein correlates with a particular phenotypic behavior of the organism, this mixture of proteins need to be separated in the manner of Eqn. 6.1. These are achieved by separation techniques that differentiate among proteins because of the differences in their physical properties like molecular weight, net charge, and binding affinity.

6.3 History of Separation Processes

Two well understood physical properties that are used for separating mixtures of fluids or particles are separations based on *charge* and *mass*. The earliest documented example of the first kind (i.e., based on electrokinetic forces) was discovered by Reuss [80] in 1809. He discovered the phenomenon of electroosmosis, which is the motion of polar liquids in an electric field, when he found water moving through sand particles under the influence of an external electric field. Tiselius [109] was the first to monitor the movement of protein molecules in an electric field which was termed as moving boundary electrophoresis (MBE). The principle of MBE was that when the light refracted by regions of the liquid medium in which the proteins were dispersed was analyzed in a schlieren optics device, one could differentiate between different regions of the separated protein-liquid mixture because each of the regions had a different refractive index. This technique enabled the first accurate measurement of protein mobilities [57]. In zone electrophoresis (ZE), the proteins are completely separated from each other (unlike in MBE where they overlap). This was achieved by carrying out the electrophoresis on a 'support medium' and not in a liquid. Paper was the first such 'support medium' that was used [47]. However, the high content of carboxylic groups in paper produced severe streaking of the proteins across the paper during separation. In 1950, Gordon et al. [30] introduced the technique of electrophoresis in an agar gel. This process achieved popularity when the first '2-dimensional' separation of biological fluids like sera was achieved in the gel [31]. Here '2-dimensional' separation means that, orthogonal to a first dimension electrophoretic step, an immuno-detection step based on simple diffusion was activated by placing proper antisera (blood serum containing a mixture of antibodies, produced by the immune system for the same antigen) across the length where electrophoresis took place. Smithies [100] was the first to report the excellent resolving power of starch (specifically, potato starch) based gels for detecting haptioglobins in sera, but their use waned due to the opaqueness of this gel. Electrophoresis in thin layers of silica gels was first reported by Honnegar [38] in 1961. It was around this time that the first separation of proteins using IEF was reported. IEF exploited the ‘amphoteric’ properties of proteins. We will explain more about this property over the next few sections.

With the proliferation of separation techniques, many ‘2-dimensional’ separations based on orthogonal coupling of any two techniques were devised (as shown in Fig. 6.1). Some of the other separation techniques that were used in conjunction with the ones that have been mentioned above were SDS-PAGE (based purely on mobility difference due to mass) [25], gel-chromatography [77] and isotachophoresis [34].

Whether used purely by itself, or in combination with an orthogonal technique (especially SDS-PAGE), IEF attained immense popularity, partly due to its use of a unique property of proteins that distinguishes itself from many other impurities that are typically found in biological samples. This property, termed ‘amphotericity’, and the way it is exploited in IEF is described next.

6.4 The IEF separation process

A few definitions [28] are in order before an explanation of IEF.

- **Amphoteric molecules:** Molecules that interact with both acids and bases are called amphoteric molecules. Eg: water, amino acids.
- **Electrolytes:** Substances that dissociate into free ions, when dissolved in a solvent, are called electrolytes. Eg: acids, bases, salts.
- **Ampholytes:** Substances that are both amphoteric as well as electrolytic in nature are called ampholytes. Eg: proteins, peptides.

- **pH:** $pH = -log_{10}([H+])$, where [H+] denotes the local concentration of hydrogen ions in mol/litre. pH always lies in the range $0 < pH < 14$. Acids have $pH < 7$, and bases have $pH > 7$.

IEF is a separation technique based on electrophoresis that is used to isolate ampholytes (like proteins and peptides) from the mixture in which they occur. The basic experimental setup is shown in Fig. 6.3.

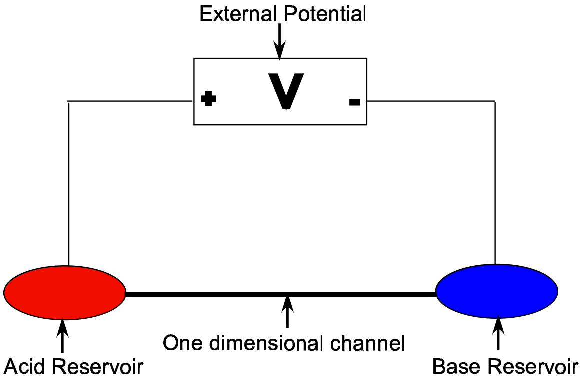


Figure 6.3: IEF experimental setup.

From Fig. 6.3, we can see that when the external potential is switched on, there will be an electric field vector at each point in the channel. Assume for the moment that the electric field vector at each point in the channel is constant. The channel is filled with some gel-like medium (explained in detail later in Sec. 7.2) and it connects the acid and base reservoir. Further assume, that this medium establishes a stable pH gradient along the channel (there must be a pH gradient because the left (acidic) reservoir has a lower pH than the right (basic) reservoir). Now, consider the behavior of a single protein molecule. Since this protein molecule is an ampholyte, it will dissociate into ions due to its electrolytic tendency. Due to its amphoteric tendency, the ampholyte molecule will simultaneously react chemically with the hydrogen (H+) as well as hydroxyl (OH-) ions that it finds in its vicinity. The protein molecule is an amino acid, so it will either gain or lose charge (by way of reacting with H+ or OH- ions) according to the surrounding pH. Thus, each protein molecule can have a net positive or negative charge on it. As long as the protein molecule has a net charge, it will drift along the channel because of the electric field at each point along the channel, and continuously gain or lose charge to the surrounding medium according to a chemical law that is governed by what is known as its **titration curve** (explained later in Sec. 7.1 and in the next chapter). The forces acting on a single ampholyte molecule are shown in Fig. 6.4.

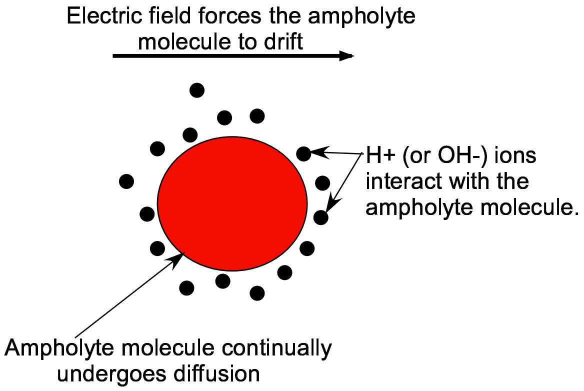

Electric field forces the ampholyte molecule to drift
H+ (or OH-) ions interact with the ampholyte molecule.
Ampholyte molecule continually undergoes diffusion

Figure 6.4: Forces on Ampholyte molecule.

At some point in time, due to the nature of the titration curve, the molecule might find itself at a point in the channel where it *does not have any charge on it.* At this point, which is termed as the **isoelectric point** of that ampholyte, the molecule will stop having a convective velocity, but it will still have motion due to diffusion. Until the molecule reached near its isoelectric point, the diffusive forces acting on it were negligible as compared to the convective forces that were acting on it.

This random diffusive velocity is very small, but it might take the molecule away from its isoelectric point. However, as soon as the molecule moves away from its isoelectric point, it will gain or lose charge (in accordance with its titration curve and the local pH) and will gain a drift velocity due to the electric field that will *focus* it back to its isoelectric point. Thus all the molecules of a particular ampholyte are continuously focused back to their isoelectric point.

The assumptions made in the above argument (which will be justified in the coming sections) for the focusing behavior of proteins were:

- A constant electric field in the channel.
- A stable pH gradient is established in the channel due to the gel-like medium in the channel.
- Each ampholyte in the channel (including the required proteins) is continually focused to its isoelectric point (pI) along the channel.

Different ampholytes have different pIs and will hence focus at different regions along the channel, thus yielding the needed separation. Once this focusing has been completed, each protein can be eluted separately from the channel. Though an experimental research topic in its own right, in this thesis we are not concerned with the process of eluting the ampholytes.

6.5 Advantages of IEF as a separation process

The main advantage of IEF as compared to other separation processes is that it is based on a unique property of proteins (or peptides) - that it is an ampholyte. In the previous section, it was shown how this unique property is exploited for the purpose of separation. IEF has been used to separate proteins with a pI difference of as low as 0.005 pH units [111]. This unique property has enabled the use of IEF in conjunction with other separation process that are based purely on charge or mass of the protein molecule. For example, this kind of '2-dimensional' separation can be done by firstly, separating proteins according to their pI (by IEF) in a slab of gel and then, separating those proteins that find themselves close to each other even after IEF (because of very close isolectric points), by gel electrophoresis (separation based on difference in molecular weights of the protein molecules) along a perpendicular direction in the gel. This kind of 2-D separation has been used to separate proteins, hundreds at a time from biological fluids and tissues [2]. IEF is also used in monitoring the purification of proteins, evaluating the stability of proteins and in proteomics.

In the next chapter, we will begin by explaining the two main concepts that justify the assumptions mentioned in Sec. 6.4:

- The **titration curve** for each ampholyte - which will help explain the isoelectric point and its relation to the ampholytes' focusing behavior.
- The **carrier ampholytes** - which will help explain the stability of the pH gradient in the channel that is essential for good focusing.

We will then discuss the need for a computational model for a better design of the IEF experiment. We look at the governing equations for IEF and show how our approach to modeling the chemical reaction between the ampholytes and the hydrogen ions results in reduced simulation times.

Chapter 7

Governing Equations and Full Order Model Simulations for IEF

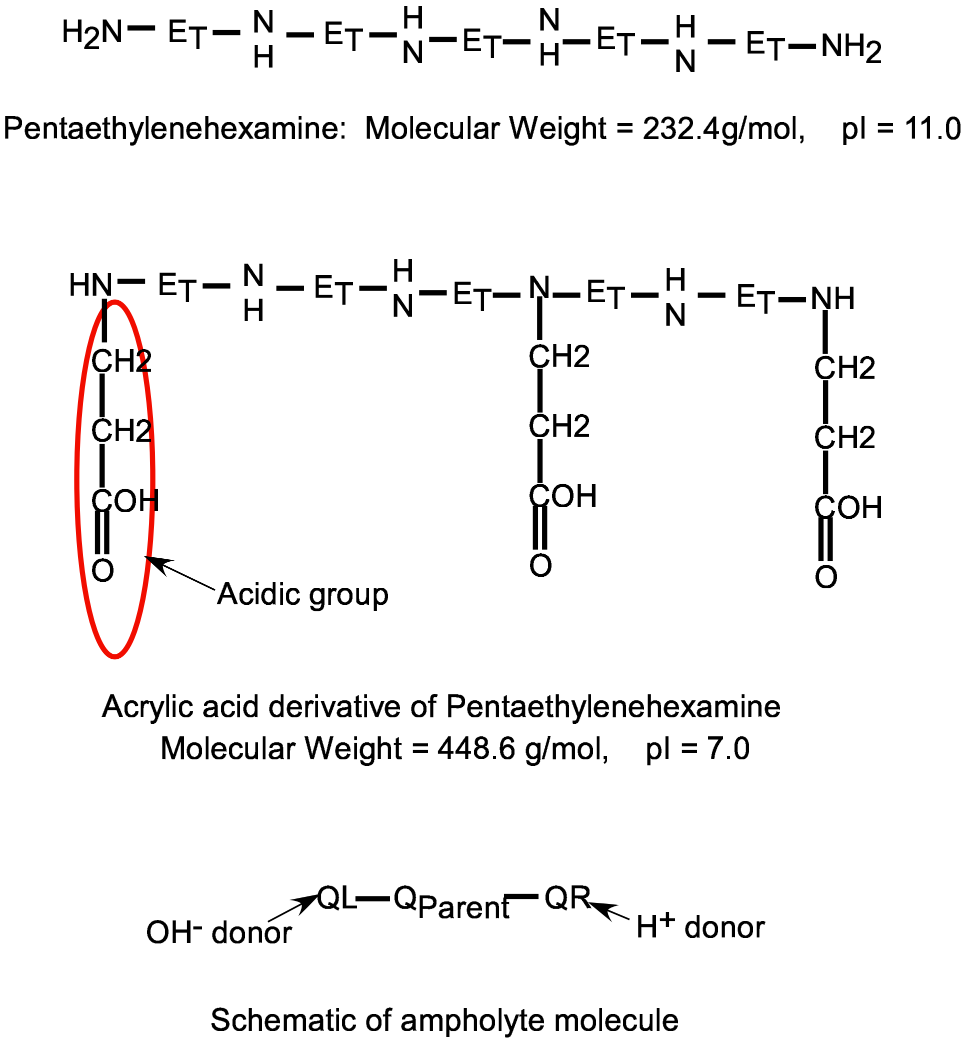
In this chapter, we will first discuss two main concepts - the **titration curve** and **carrier ampholytes** - that explain the chemistry of the IEF process and justify the assumptions made in the previous chapter. We will then present an overview of the experimental and computational results that are currently available. After that we will derive the governing equations for IEF in Sec. 7.5 and our fast dynamics assumption that enables us to perform quicker simulations than the ones in the existing literature. We will then present the simulations of the finite element simulations of our model which we run using the COMSOL finite element modeling software. We note at the outset that even though our model in this chapter provides a much faster simulation time than the existing literature, there is still a need for even cheaper simulations. Our simulations in this chapter will provide the data for a further reduction in model size that we attempted with proper orthogonal decomposition. The further reduction in model size will be explained in the next chapter in which we will denote the model and the finite element simulations of this chapter as the *full order model for IEF simulations.*

7.1 Titration curves

Titration curves explain the dynamics of the chemical reaction between ampholytes and the surrounding H+ and OH- ions. Mathematically, the titration curve gives the net charge per ampholyte molecule at a given pH. In this thesis this net charge per ampholyte molecule is denoted as $f(H)$ (or $f(pH)$, though $f(H)$ is used in this thesis). Three examples of ampholyte molecules are given in Fig. 7.1.

In what follows, we non-dimensionalize the chemical reaction by denoting the non-dimensionalized hydrogen ion concentration $[H]$ as $[\hat{H}] = 10^{-7}\frac{mol}{litre}[\hat{H}]$. The non-dimensionalized hydroxyl ion concentration $[\hat{OH}]$ is given by $[\hat{OH}] = \frac{1}{[\hat{H}]}$ (since $[\hat{H}][\hat{OH}] = 10^{-14}\frac{mol^2}{litre^2}$). The equation for the titration curve $f(H)$ can be determined from the rate constants of the chemical reaction between the ampholyte molecule and the surrounding H+ and OH- ions in the following way.

Let the ampholyte have a concentration $[Q]$ at the instant that it is introduced into the buffered solution. We assume equilibrium chemistry for the dissociation reaction of the ampholyte, which means that for the purposes of determining the migration of the ampholyte molecules, we do not need to account for the fact that there is a finite amount of time that is needed for the ampholyte molecule to dissociate (during which the convective forces on the ampholyte molecule might vary by a small quantity). We can then use the Henderson Hasselbalch equation [53] for weak acids to determine the average net charge on the ampholyte molecule at a particular pH. Assume that at equilibrium, a certain fraction A of the ampholyte


H2N— ET— N H — ET— H N — ET— N H — ET— H N — ET—NH2
HN— ET— N H — ET— H N — ET— N— ET— H N — ET—NH
CH2
CH2
COH
O
Acidic group
OH- donor
QL—QParent—QR
H+ donor

Pentaethylenehexamine: Molecular Weight = 232.4g/mol, pI = 11.0

Acrylic acid derivative of Pentaethylenehexamine
Molecular Weight = 448.6 g/mol, pI = 7.0

Schematic of ampholyte molecule

Figure 7.1: The schematic of an ampholyte molecule is shown at the bottom of the figure. The molecule at the top is the parent ampholyte molecule Pentaethylenehexamine, and the molecule below it is derived from the former by adding acidic groups to it [85].

molecules has dissociated into its acidic group in the following way:

$$QR + H_2O \overset{K_R}{\rightleftharpoons} H_3O^+ + QR^- \tag{7.1}$$

where QR is the acidic portion of the ampholoyte molecule that dissociates into QR^- when it donates a hydrogen ion. Hence we have (for weakly buffered solutions), $[QR] = [Q](1 - A)$, $[QR^-] = [Q]A$ and K_R is the equilibrium constant for the reaction.

Assume that at equilibrium, a certain fraction B of the ampholyte molecules has dissociated into its basic group in the following way:

$$QL + H_2O \overset{K_L}{\rightleftharpoons} OH^- + QL^+ \tag{7.2}$$

where QL is the basic portion of the ampholoyte molecule that dissociates into QL^+ when it donates a hydroxyl ion. Hence we have (for weakly buffered solutions), $[QL] = [Q](1 - B)$, $[QL^+] = [Q]B$ and K_L is the equilibrium constant for the reaction. Then we have that

$$[H^+][QR^-] = [H^+][QR]A = K_R[QR] = K_R[Q](1 - A). \tag{7.3}$$

and

$$[OH^-][QL^+] = [OH^-][QL]B = K_L[QL] = K_L[Q](1 - A). \tag{7.4}$$

From hereon, we drop the square brackets that denote the concentration for a given species. For example, instead of the $[H]$, we will denote the hydrogen ion concentration as H (or $\hat{H}$ in the non-dimensional case). The net average charge per ampholyte molecule $f(\hat{H})$, is given by $f(\hat{H}) = Q(B - A)$, which from Eqns. 7.3 and 7.4 can be written as (since $[OH] = \frac{1}{H}$)

$$f(\hat{H}) = Q(B - A) = Q(\frac{K_L \hat{H}}{K_L \hat{H} + 1} - \frac{K_R}{K_R \hat{H} + 1}) \tag{7.5}$$

The net charge $f(\hat{H})$ is zero at the **isoelectric point**. Denote the hydrogen ion concentration at the isoelectric point as IEH. We get IEH by setting the value of $f(\hat{H})$ in Eqn. 7.5 to zero. Hence we have that $[IEH] = (\frac{K_R}{K_L})^{1/2}$. If we denote the quantity $(K_R K_L)^{1/2}$ as q, then we can rewrite the titration curve equation Eqn. 7.5 as

$$f(\hat{H}) = \frac{q}{q + \frac{IEH}{\hat{H}}} - \frac{q}{q + \frac{\hat{H}}{IEH}} \tag{7.6}$$

Consider an ampholyte molecule that has an isoelectric point pI = 8. Then the net charge per unit ampholyte molecule is given by the titration curve in Fig. 7.2 (where the net charge is plotted against pH instead of $\hat{H}$).

From Fig. 7.2, we can see that the net charge on the ampholyte molecule, when it finds itself in a region in the channel wih pH = 8 is zero. This pH is called the **isoelectric point** or **pI** of that particular ampholyte. This fits in with the definition of the isoelectric point mentioned in Sec. 6.4 where we stated that the ampholyte molecule focuses at its pI because at that point there is no net charge on the molecule and hence it does not experience any convective force due to the electric field. When the ampholyte molecule finds itself in a region that has $pH < pI$, which is in the left portion of the titration curve in Fig. 7.2), the net charge on the

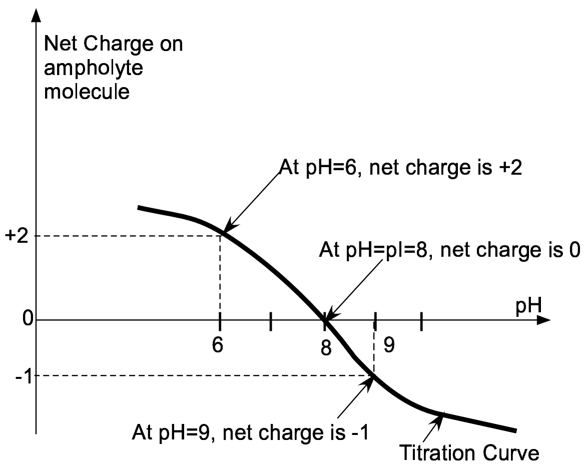


Figure 7.2: Example of Titration Curve.

ampholyte molecule is positive. Hence it experiences a convective force to the right (since the electric field points to the right), towards its isoelectric point. When the ampholyte molecule finds itself in a region that has $pH > pI$, which is in the right portion of the titration curve in Fig. 7.2), the net charge on the ampholyte molecule is negative. Hence it experiences a convective force to the left, again, towards its isoelectric point. In this way, the amphoteric property of proteins is exploited in IEF to isolate the proteins by focusing it at its isoelectric point.

7.2 Carrier Ampholytes

In Sec. 6.4, one of the assumptions made in order to explain the focusing process in IEF was that a stable pH gradient is established in the channel, which is maintained by the 'medium' present in the channel. This 'medium' is described in this section.

Beginning in 1961, Svensson [104], [105], [106] was the first to report the possibility of separating amphoteric molecules (like proteins) if a stable pH gradient could be established across two electrodes in an appropriate medium. However, he lacked the technique to 'carry' such a stable pH gradient across a wide pH range. His student Vesterberg was the first to chemically construct such a medium [115], [116]. He made certain oligoamines react with a $\alpha - \beta$ acrylic acid to simultaneously generate a range of amphoteric molecules whose pIs were closely spaced and spanned a pH range that was wide enough for a viable separation of many different kinds of proteins that were typically found in a biological sample. This 'medium' is then, a collection of many (> 100) ampholytes [81] whose isoelectric points (pI), uniformly span the pH range between the left (acidic) reservoir and the right (basic) reservoir. These ampholyte molecules are called *carrier* ampholytes because they maintain a stable pH gradient in the channel (they 'carry' the pH gradient).

Each of these carrier ampholytes (CAs) are present in much higher concentration (in the order of 1000 times greater) than the proteins in the mixture. These carrier ampholytes are commercially available for any given pH range, and we use these carrier ampholytes to separate the proteins that we desire. For example, if we need to separate 2 proteins P1 and P2 which have pIs 6.50 and 6.86 respectively, we can purchase a mixture of carrier ampholytes with their pI's in the range $6 < pH < 8$ and use them to separate the 2 proteins. These ampholytes when introduced into the CA mixture at the beginning of the IEF process, seek out each of their pIs along the channel, to create a staircase-like pH gradient as shown in Fig. 7.3.

The 'steps' in the staircase-like pH gradient are created because each of the

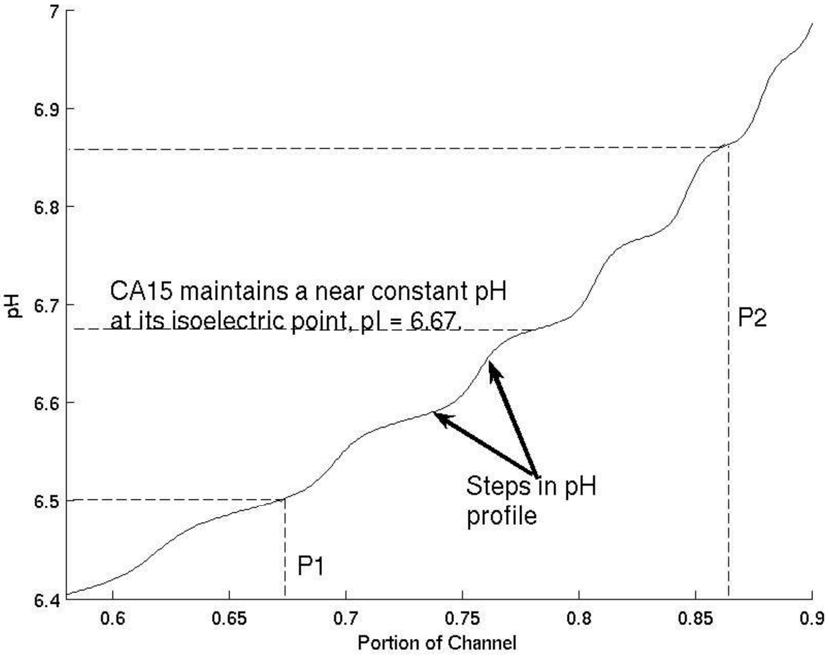


Figure 7.3: The pH profile for isoelectric focusing with 30 ampholytes is said to have a staircase-like gradient as shown in this illustration. Proteins P1 and P2 focus at the regions of the channel that have pH values that match their respective isoelectric points (pI). In the figure we show how the 15^{th} carrier ampholyte CA15 helps maintain a near constant pH equal to its pI (=6.67) due to its buffering properties.

CAs focus at their respective pIs. When a particular CA focuses in a particular region, it results in that region having a near constant pH around where that CA focused, and which is equal to that particular CA's pI. This corresponds to the flat portion of one step shown in Fig 7.3. The width of each step depends on the mobility and diffusivity coefficients of that particular CA [81]. For example, if in Fig. 7.3 the acid reservoir has pH = 6, and the basic reservoir has pH = 8, and CA15 has pI = 6.67, then at the end of the focusing process, CA15 maintains a region of near constant pH (= pI = 6.67) in the region around where it focuses.

Now, the protein mixture is introduced into the channel filled with the CAs. Since the proteins have a much lower concentration than the CAs, the CAs act like buffers that maintain a step like pH gradient, while the proteins disturb the pH gradient in the focusing process. Since the proteins as well as the CAs have both got the ampholytic tendency, the proteins can be simultaneously introduced along with the CAs, and both, the proteins and the CAs, undergo simultaneous focusing in the channel. The order of introduction of the proteins and the CAs does not change the basic physics in any way. For a different initial condition, the transient behavior will change compared to the first case (when the protein mixture is introduced after the CAs). However, the final positions of the CAs as well as the proteins after focusing, as well as the buffering nature of the CAs remain the same.

From the point of view of an individual protein molecule, one can imagine its basic tendency as 'always seek the isoelectric point (pI)'. That being the case, even if the pH gradient is changing in the channel (due to the CAs being focused), the protein molecule will keep gaining/losing charge and moving towards its isoelectric point. Since the CAs have much higher concentration than the proteins, the pH gradient created by the CAs is stable with respect to the (minor) changes in pH caused by the focusing of the proteins.

One can see that a large number of carrier ampholytes corresponds to a large number of steps in the pH gradient. Hence the term 'staircase-like pH gradient' is often used in literature [81]. The formation of this staircase-like pH gradient is used as one of the tests of a computational model.

7.3 The assumption of constant electric field in the channel

One of the assumptions that we have made in modeling IEF is that the electric field in the channel is constant throughout the focusing process. In reality, there is a sharp drop in electric field intensity as soon as the voltage is switched on and then a relatively constant electric field is observed for the rest of the focusing process. The inclusion of varying electric fields will involve an additional Poisson equation for electric field intensity that will increase computational time because of the involvement of all charged species in the resultant governing equations. Here, we explain our reasoning behind the constant electric field assumption.

The electric current in the channel is monitored with the help of an ampmeter in the external circuit that is shown in Fig. 7.4. It has been reported [103] that in order to avoid Joule heating in the channel, the best experimental results are obtained when the voltage is increased in a few steps. One will typically observe an initial electric current which shows a sharp drop and then remain at a constant level till the next step up in the voltage is applied as shown in Fig. 7.4.

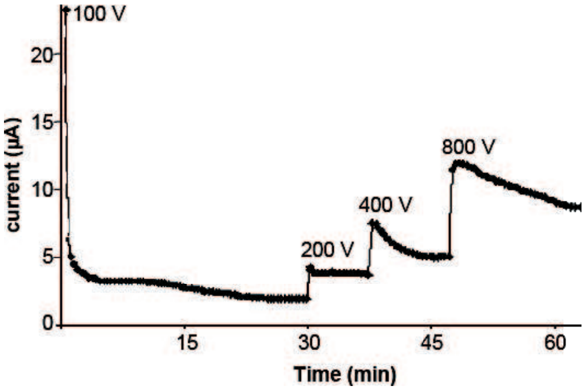


Figure 7.4: Electric field stabilization at each step in the stepped-voltage experiment

The physics behind the quick drop of the current in the channel to a near constant level at a given constant voltage is the following - the initial large current is due to all the carrier ampholytes moving towards their respective isoelectric points. There is a contribution due to the movement of the proteins too, but it is negligible due their small concentrations relative to the CAs. As the CA molecules focus around their pIs, they lose almost all their charge and from then on, only gain/lose charge due to their (relatively small) diffusive movement away from their pIs. However, any deviation away from their pI will result in them getting focused back to the pI by the electric field due to the charge they will pick up by reacting with hydrogen or hydroxyl ions. Hence, when the current in the external circuit shows a steep drop, one can conclude that most of the CA molecules are near their pIs. For the purposes of our modeling, we will hence assume that it is enough to be able to predict the IEF characteristics like the ampholyte concentration, focusing time and the pH gradient with the assumption of a constant electric field.

7.4 Previous experimental and theoretical techniques used to understand IEF

There are only two tools that have been widely used for monitoring IEF experiments: optical imaging of dyes in the channel and monitoring the electric current in the channel. Since accurate measurement of quantities like the step width of all steps in staircase-like pH gradient is not possible with only the above two tools, experimental techniques have a limited scope in understanding the dynamics of the focusing process and moreover, they are both expensive and time consuming.

Optical imaging is a widely used technique ([61], [60]) in which a dye marker (which has a known pI), is introduced in along with the CAs and the proteins in the channel. We then know that the region where this dye focuses has a pH equal to the pI of that dye. If one uses many dyes with known pIs, and captures the movement of each of those dyes in that channel, then one can plot the approximate formation of the pH gradient in time. For a more accurate graph of the pH in the channel, one needs to use a proportionately bigger number of dyes (with different pIs) and carefully track the movement of each dye in the entire channel, which is a labor intensive process.

However, there is much more detail about the focusing process, like the widths of the individual regions where each ampholyte focuses and the concentration of the ampholytes in each peak that can neither be measured by monitoring the electric current [103] nor by using dyes (since the dyes would need to have the same diffusivity and mobility as the ampholytes).

The experimental techniques that rely on monitoring the electric current in the channel, yield a rough measure of the pH gradient formation and the time till focusing. Recognizing the limitations of such experimental techniques, many research groups have applied numerical techniques towards understanding the focusing process. Exact electrochemical data like diffusivity, mobility, and titration curves that are needed to simulate IEF are not known even to the manufacturers of the CAs because all the CA molecules are synthesized together using variations of Vesterberg's original synthesis technique and chemists have not been able to isolate every ($\approx$ 1000) type of CA molecule from such mixtures in order to measure their individual chemical properties. Nonetheless, non-dimensional results have been used to gain an understanding of the IEF process. Palusinski et al. [73] had one of the first theoretical explanations of the way in which the IEF parameters affect the focusing process. In [110] and [69] a finite difference analysis of the IEF equations for 5 ampholytes showed a match of some qualitative features of the results to experimental data. In [71] a finite element simulation of the IEF equations with 150 ampholytes showed the formation of the step-like pH gradient. However this simulation took almost 40 hours for a single run (for a pH range of 3-10). In [111] and [61] the protein electrochemical parameters completely determined the width of the focusing region and their results matched those given by a detailed optical imaging experiment. In [60] the sensitivity of changes in the pH gradient due to changes in the initial condition of the experiment wee explained.

The main argument for better modeling and computational techniques for understanding IEF is that even though the chemical techniques for designing better CAs have advanced in the past 40 years, it is not clear what exact molecular properties to aim for in the chemical design. For example, the question of how sensitive the focusing process is to the sharpness of the titration curves, is not something that can be cheaply answered by experiments or by the present modeling/computational techniques. Due to the approximations that we make in our modeling approach, we will show how it is able to predict focusing for small pH gradients in a much shorter simulation time than existing approaches. We contend that our approach can be further used as a guide for designing ampholyte electrochemical properties for experiments that use larger pH gradients. Our modeling technique is explained in the next section

7.5 Governing Equations of IEF

In this section, we derive the governing partial differential equation that describes the focusing of ampholytes in a one dimensional channel. Except for more recent explorations of IEF in 2-dimensional geometries (including [102] and some work which we have presented elsewhere [62]), all prior computational treatments of IEF make use of one dimensional geometries - i.e., the ampholytes, hydrogen and hydroxyl ions, all diffuse and convect in one dimension. This is reasonable because the channels used for IEF are typically $\approx 30mm$ long and have an internal diameter of $0.2mm$ [103]. We note however that the underlying physics does not change in

2 dimensions and our modeling of the physics can be used to answer questions in 2D IEF like for example, how the use of tapered channels can be used to create a steeper pH gradient in the thinner sections of a channel and hence separate proteins which have isoelectric points that are very close to each other.

From hereon, all the modeling that is described is for IEF in 1 dimension (the x-axis). The unsteady convection-diffusion equation describing the behavior of the ampholyte species Q_i is given by

$$\frac{\partial(Q_i(x,t))}{\partial t} = D_i \cdot \frac{\partial(Q_i(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_i \cdot E \cdot f_i(H(x,t)) \cdot Q_i(x,t)) \tag{7.7}$$

The diffusive flux is modeled as $-D_i \frac{\partial Q(x,t)}{\partial x}$ using Fick's law of diffusion [21]. The diffusion constant D_i is assumed as constant because we assume that the focusing process is isothermal [111]. The convective flux which is given by

$$F_{conv}(Q_i(x,t)) = M_i E(x,t) Q_i(x,t) f_i(H(x,t)) \tag{7.8}$$

can be understood by accounting for the physics of each molecule's motion. The electric field $E(x,t)$ makes the charged ampholytes drift. The net charge on each ampholyte molecule on average at a given pH is given by $f_i(H(x,t))$ as given by the titration curve. Hence the net charge for the ampholyte concentration $Q_i(x,t)$ is $Q_i(x,t)f_i(H(x,t))$. The constitutive relation used here (analogous to the ones for mass, heat and momentum) is that the flux is proportional to the force. The proportionality constant M_i is called the mobility coefficient. We will discuss the relation between M_i and D_i later in this chapter.

The electric field $E(x,t)$ will be assumed as constant in space and time, for reasons mentioned before. From hereon, we denote it as E. The error in focusing time due to this assumption is small, because the spatio-temporal variation is significant only in the small time interval at the beginning of the focusing process.

Finally, the divergence of the diffusive and convective flux is equated with the time rate of change of concentration of the ampholyte to give Eqn 7.7.

Apart from the carrier ampholytes, each of the protein species are also modeled in exactly the same manner (since, they too, are ampholytes) with their corresponding physical parameters D_j, M_j, and $f_j(H(x,t))$. The governing equations for a protein species P_j are

$$\frac{\partial(P_j(x,t))}{\partial t} = D_j \cdot \frac{\partial(P_i(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_j \cdot E \cdot f_j(H(x,t)) \cdot P_j(x,t)) \tag{7.9}$$

To summarize, for N ampholytes and M proteins, the convection diffusion equations that govern them are

$$\begin{aligned}
\frac{\partial(Q_i(x,t))}{\partial t} &= D_i \cdot \frac{\partial(Q_i(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_i \cdot E \cdot f_i(H(x,t)) \cdot Q_i(x,t)); \quad i = 1 : N \\
\frac{\partial(P_j(x,t))}{\partial t} &= D_j \cdot \frac{\partial(P_i(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_j \cdot E \cdot f_j(H(x,t)) \cdot P_j(x,t)); \quad j = 1 : M
\end{aligned} \tag{7.10}$$

One could write similar equations for the hydrogen ion dynamics, because they too are ampholytic. Since the charge on each hydrogen ion is exactly +1, we would have the following equation

$$\frac{\partial(H(x,t))}{\partial t} = D_H \cdot \frac{\partial(H(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_H \cdot E \cdot H(x,t)) \tag{7.11}$$

However, with our assumption of equilibrium chemistry due to the fast reaction time of the hydrogen ion and the ampholytes and the chemical reactions given in Eqns. 7.3 and 7.4, we choose to model only this fast dynamics in the following way. Since the i^{th} ampholyte has concentration $Q_i(x,t)$ at (x,t), the hydrogen ions consumed by it since the initial instant is given by $\Delta H_i(x,t) = Q_i(x,t) f_i(H(x,t))$. Let $H_o(x)$ be the initial concentration of hydrogen ions at the initial time. Then we have the total hydrogen ion consumption by all the ampholytes as

$$H(x,t) = H_0(x) - \sum_{i=1}^{i=N} Q_i(x,t) f_i(H(x,t)) - \sum_{j=1}^{i=M} P_j(x,t) f_j(H(x,t)) \tag{7.12}$$

With this the complete set of equation that govern the ampholytes, proteins and hydrogen ions concentration are gven by

$$\begin{aligned}
\frac{\partial(Q_i(x,t))}{\partial t} &= D_i \cdot \frac{\partial(Q_i(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_i \cdot E \cdot f_i(H(x,t)) \cdot Q_i(x,t)); \quad i = 1 : N \\
\frac{\partial(P_j(x,t))}{\partial t} &= D_j \cdot \frac{\partial(P_i(x,t))}{\partial x^2} - \frac{\partial}{\partial x}(M_j \cdot E \cdot f_j(H(x,t)) \cdot P_j(x,t)); \quad j = 1 : M \\
H(x,t) &= H_0(x) - \sum_{i=1}^{i=N} Q_i(x,t) f_i(H(x,t)) - \sum_{j=1}^{i=M} P_j(x,t) f_j(H(x,t)) \qquad (7.13)
\end{aligned}$$

In [111], the boundary conditions for the ampholytes at the end of the channels have been set as impermeable to ampholytes and proteins. However, in actual IEF experiments, it has been noted by [111] (and others like [61]), that the ampholytes' and proteins' flux into the reservoirs is proportional to their concentration at the ends of the channel (which is very small compared to the peak concentration of the ampholyte at the isoelectric point). However, this proportionality constant, i.e., the permeability is not well understood by the IEF modeling community. We have instead chosen to model the boundary condition as Dirichlet type with a small constant ampholyte (or protein) concentration at the edges of the channel. We have set this small value the same as that set at the initial condition (which is very small as compared to the maximum concentration at the focused peak of the ampholyte), which also helps in satisfying the hydrogen ion constraint that we have defined in Eqn.7.13.

7.5.1 Titration Curve value for CA molecules

Here, we compute the titration curve parameter q for CA molecules with good buffering capacities - i.e., CA molecules that can maintain a stable pH value in the part of the channel that is near the ampholyte's isoelectric point, inspite of (small) changes in ampholyte concentration due to the chemical reaction between the ampholyte and the hydrogen ion.

The acid dissociation constant pK_a [28] for a hydrogen ion donor group is defined as $pK_a = -log_{10}K$, where $K = \frac{[H+][A-]}{[HA]}$, is the equilibrium constant of the dissociation reaction of the acid HA. It is well known [81] that the difference between the pK_a values of the acidic and basic groups (QR and QL) of the CA molecules lie between 2.0 and 2.5. Higher buffering capacities for the ampholyte molecule are desirable and they are achieved with a lower value of pK_a. We assume that all CA molecules have $pK_a = 2$ (again, this can be changed with better knowledge of individual CA molecule properties) and term such ampholytes as 'good' ampholytes.

Now the base dissociation constant pK_b for any group is defined as $pK_b = 14 - pK_a$ [28]. It is defined analogously as compared to pK_a, except with the hydroxyl instead of hydrogen ion. Since the equilibrium constant K_L was defined for the hydroxyl ion reaction, from the above, for a good ampholyte, we should have

$$pK_R - (14 - pK_L) = 2 \tag{7.14}$$

where $pK_R = -log_{10}K_R$ and $pK_L = -log_{10}K_L$. We have normalized the hydrogen ion concentration in the following way $\hat{H} = \frac{H}{10^{-7}}$. With this, we get the non-dimensionalized equilibrium constants as $\hat{K_R} = \frac{K_R}{10^{-7}}$ and $\hat{K_L} = \frac{K_L}{10^{-7}}$. Using the values of $\hat{K_R}$, $\hat{K_R}$, Eqn. 7.14 and some algebra, we get that

$$q = (\hat{K_R}\hat{K_L})^{0.5} = 10^{-(\frac{p\hat{K}_R + p\hat{K}_L}{2})} = 0.1 \tag{7.15}$$

This is the value of q for a good CA molecule in our model for the titration curve. We can analogously compute the value q for CA species that have sub-optimal buffering, i.e., higher values of the difference between the pK_a values of the QR and QL groups.

7.6 Non-Dimensionalization and Simulation Results

As explained before, most of the parameters in the IEF physics, including mobility, diffusivity, and the titration curve parameters, and even the individual concentrations of each ampholyte molecule in the experimental setup are only known approximately. Hence, our aim is to understand the effect of scaling the convective and diffusive forces, relative to appropriate units. In this section we will show how we non-dimensionalized Eqn. 7.13. Once again, all the non-dimensionalized parameters have a 'hat' symbol over them. For example, the parameter D_H when non-dimensionalized, will be represented by the symbol $\hat{D_H}$. In the next section, we will show the results for our simulation of the IEF process with 100 ampholytes.

We begin by noting that the values for the mobility coefficients M_H and M_Q that have been widely used in simulations (including [61], [103]), are $M_H = 36.2 \times 10^{-8} \frac{m^2}{V.s}$ and $M_Q = 3 \times 10^{-8} \frac{m^2}{V.s}$. The value of M_Q is the average mobility coefficient for all ampholytes, but in our simulations, we will be using the same value of M_Q for all ampholytes. The reason for this is that very little is known about the exact distribution of the carrier ampholyte molecules in terms of their molecular structure [111]. The molecular structure is needed for knowing the molecular weights and ultimately D_Q and M_Q. Some information about the molecular structure is known in the following sense - chemists can know for sure that 3 or 4 specific ampholyte molecules are definitely present among the hundreds that are synthesized in the laboratory [85]. It is also known that ampholyte molecular weights increase with decreasing pIs, because of the addition of acidic groups as shown in Fig. 7.1. In a conversation with Amgen - one of the manufacturers of carrier ampholytes - the author confirmed that even they do not have any more information about the individual ampholyte molecular properties. This lack of information about the ampholyte molecules sounds dire, but we note that even with such 'average molecule' information we can get results that agree with experiment (as have other research groups [111]) and we also note that such information (whenever it is eventually known) is easy to incorporate in our modeling.

We model the governing equation for the hydrogen ions as constraints in Eqns. 7.13, and hence the mobility and diffusive parameters D_H and M_H, are not present in the governing equations 7.13. However, we will use them to non-dimensionalize the other parameters and the length and time scales. From the definition of the mobility coefficient and the Einstein-Nernst relation [78], we have the following relation between M_H and D_H

$$\frac{M_H}{D_H} = \frac{zF}{RT} \approx 38V^{-1} \tag{7.16}$$

where $z = +1$ is the charge on the hydrogen ion, $F = 9.65 \times 10^4 \frac{C}{mol}$ is the Faraday constant, $R = 8.31 \frac{J}{K.mol}$ is the universal gas constant and $T = 300K$ is room temperature (IEF experiments are conducted at room temperature). From this we get that $D_H \approx 10^{-8} \frac{m^2}{s}$. We define the non-dimensional parameters for length x and time t as

$$\begin{aligned} x &= l_0 \hat{x} \\ t &= t_0 \hat{t} \end{aligned} \tag{7.17}$$

where l_0 is the length scale and t_0 is the time scale. The scales l_0 and t_0 are chosen in the following way. When we divide Eqn. 7.7 (where we replace the symbols D_i, M_i and Q_i by D_Q, M_Q, and Q for clarity) by $D_H t_0$ and multiply it by l_0^2, we get

$$\left(\frac{l_0^2}{D_H t_0}\right) \frac{\partial Q(x,t)}{\partial t} = \left(\frac{D_Q}{D_H}\right) \frac{\partial^2 Q(x,t)}{\partial \hat{x}^2} - \left(\frac{l_0 M_Q E}{D_H}\right) \frac{\partial (Q(x,t) f(H(x,t)))}{\partial \hat{x}} \tag{7.18}$$

We choose $l_0 = 3 \times 10^{-2}m$, which is a typical channel length in IEF experiments, and $t_0 = 9 \times 10^4 s$ to get $\frac{l_0^2}{D_H t_0} = 1$. Next, we estimate $\hat{D_Q} = \frac{D_Q}{D_H}$ in the following way. From Stokes' drag for a low Reynold's number flow and from Einstein's relation for Brownian motion [78], we have the following relation between the diffusion coefficient of a spherical particle and its diameter

$$D \approx \frac{kT}{\mu d} \tag{7.19}$$

where μ is the viscosity of the liquid, d_Q is the diameter of particle. Hence we have

$$\frac{D_Q}{D_H} \approx \frac{d_H}{d_Q} \tag{7.20}$$

The above equation is approximate because the Stokes drag is for spherical particles, while the ampholyte molecule and the hydrogen ion are not spherical. For a hydrogen ion, which typically exists as a hydronium ion (H_3O+) surrounded by 6 water molecules in a solvation shell around it [121] we estimate $d_H \approx 1\ nm$. Now protein molecules can vary in size between $1-10\ nm$ [1], which we also approximate as the average size of the carrier ampholyte molecule. We take the more conservative estimate of $d_Q = d_P = 10\ nm$. We note that in our simulations, the higher the value of d_Q, the longer it takes for our simulations to converge, because the high Peclet number of our problem (which is the ratio of the convective to diffusive forces) turns out to be inversely proportional to d_Q and hence increases with decreasing d_Q. Hence, with respect to our model, we are in a sense choosing the worst case scenario by choosing the higher value of d_Q. This makes the non-dimensional value of the diffusion coefficient $\hat{D_Q}$ as

$$\hat{D_Q} = \frac{D_Q}{D_H} \approx \frac{d_H}{d_Q} \approx 0.1 \tag{7.21}$$

Next, we compute the value of the constant $\frac{l_0 M_Q E}{D_H}$. As mentioned before, E is considered a constant along the channel. In typical experiments, the electric field strength across the channel is around $3 \times 10^4 \frac{V}{m}$ [111]. We rewrite the ratio $\frac{M_Q}{D_H}$ as $\frac{M_Q}{D_H} = \frac{M_Q}{D_Q}\frac{D_Q}{D_H} \approx 0.1 \times \frac{M_Q}{D_Q}$.

From this, and from the Einstein-Nernst relation in Eqn. 7.16 (when rewritten for ampholyte molecules) will give the value of the ratio $\frac{M_Q}{D_Q} \approx 40V^{-1}$. This is because we assign the conservative value of 1 for the average charge number z for an ampholyte molecule. This value is conservative because after the small initial time at the start of the focusing process, most of the ampholyte in the channel is near its isoelectric point. The titration curve would then imply that the average charge number z is closer to 0 than 1. Once again, this higher value of the average charge number for the ampholyte molecule should increase the Peclet number which is supposed to slow the convergence of the simulations. In future simulations, once the behavior of the charge number of the CAs is better known to experimentalists, the amount of dissociation of the ampholyte molecule into the Q_R and Q_L and the valency of the dissociated ions will be better known (as compared to our assumption that the ampholyte molecule dissociates into monovalent groups). At that point, our model can be modified by the Linderstrom-Land approximation which states that a z-valent ion behaves like a monovalent ion with z-fold concentration [112].

Thus, we get that $\frac{M_Q}{D_H} \approx 0.1 \times 40 = 4V^{-1}$. We define the non-dimensional parameter $\hat{M_Q} = \frac{l_0 M_Q E_0}{D_H}$, where E_0 is the electric field scale and $E = E_0\hat{E}$. If we set $E_0 = 3 \times 10^4 \frac{V}{m}$ (so $\hat{E} = 1$), we get the value $\hat{M_Q} = 10^{-2} \times 4 \times 3 \times 10^4 = 1200$. We now have all the parametric values for simulating IEF and present the results of our simulation in the next section.

7.7 Simulation Results for IEF with 100 Carrier Ampholytes

We present here, the results of a simulation for the focusing of 100 ampholytes in a 3cm long channel (non-dimensional length $\hat{l} = 1$) with the acidic reservoir at pH=6.0, and the basic reservoir at pH=8.0. The electric field strength is $3 \times 10^4 \frac{V}{m}$, which corresponds to the non-dimensional $\hat{E} = 1$. The 100 ampholytes have equally spaced isoelectric points pI in the above pH range, with the i^{th} ampholyte having $pI_i = pH_L + \frac{(pH_R - pH_L)}{N+1} i = 6 + \frac{2}{101} i$. These are the same conditions that are simulated in [112] and we will compare our simulations to their results. The main difference in our modeling approach as compared to [112] lies in our modeling of the titration curves and the hydrogen ion governing equations. In [112] each titration of an ampholyte molecule is represented with the help of differential equations (from the rate equations of the respective chemical reactions) which take a small, but finite, amount of time as compared to the instantaneous titration reaction as represented in our governing (algebraic) equation for the hydrogen ions. The other approximations we make (which are not made in [112]) are constant electric fields and Dirichlet boundary conditions. Our simulations were run on the finite element software COMSOL running on a Linux Platform on a 2.3 GHz dual Opteron processor.

The non-dimensional initial conditions for all ampholytes is uniform with $\hat{Q}_i(x,0) = \hat{Q}_0 = 1$. There is one protein species with $pI = 7.0$ and uniform initial condition $\hat{P}(x,0) = \hat{P}_0 = \frac{\hat{Q}_0}{133}$ (equal to the ratio in [112], which lies in the 80-240 fold decrease in protein to ampholyte concentration for actual IEF experiments [103]). As explained before we have Dirichlet boundary conditions for both ampholytes and proteins set at Q_0 and P_0 respectively. The initial condition for hydrogen ions need to satisfy the constraint in Eqn. 7.13, and are set in the following way. When there are no ampholytes in the channel and we only have the acid and base in the respective reservoirs at the ends of the channel, we expect a linear distribution of hydrogen ions across the channel due to diffusion. We denote this distribution by $\hat{H}_0(x)$, and have $\hat{H}_0(x) = \hat{H}_L + \frac{\hat{H}_R - \hat{H}_L}{L}x$ where $\hat{H}_L$ and $\hat{H}_R$ are got from pH_L, pH_R and the constant $Co = 10^{-7}$ as described in Sec. 7.1. Now, when we introduce the ampholytes and proteins in the channel, but *do not yet switch on the electric field*, we can expect the distribution of the hydrogen ions to be modified due to the chemical reaction between the ampholytes and the hydrogen ions. This new hydrogen ion distribution which we call $HoBC$ (the 'BC' stands for 'before current'), is given by

$$HoBC(x) = H_0 - \sum_{i=1}^{i=N} Q_i(x,0) f_i(H_0(x)) \sum_{j=1}^{j=M} P_j(x,0) f_j(H_0(x)). \tag{7.22}$$

It is $HoBC$ that is the initial condition for the constraint in Eqn. 7.13. At time t_0, we switch on the constant electric field $\hat{E} = 1$, and run the simulation till $\hat{T} = 0.1$, which corresponds to an experimental focusing time of 15 minutes.

Focusing is said to be achieved when the required protein (or any other ampholyte) first attains a quasi-steady focused peak at some point in the channel. In [112] the simulations show that the carrier ampholyte with $pI = 7$, getting focused at 14 minutes. In our simulations, for the ampholyte with $pI = 7$, we see that the peak of the ampholytes starts stabilizing between $t = 800s$ and $t = 900s$. This is shown in Fig. 7.5.

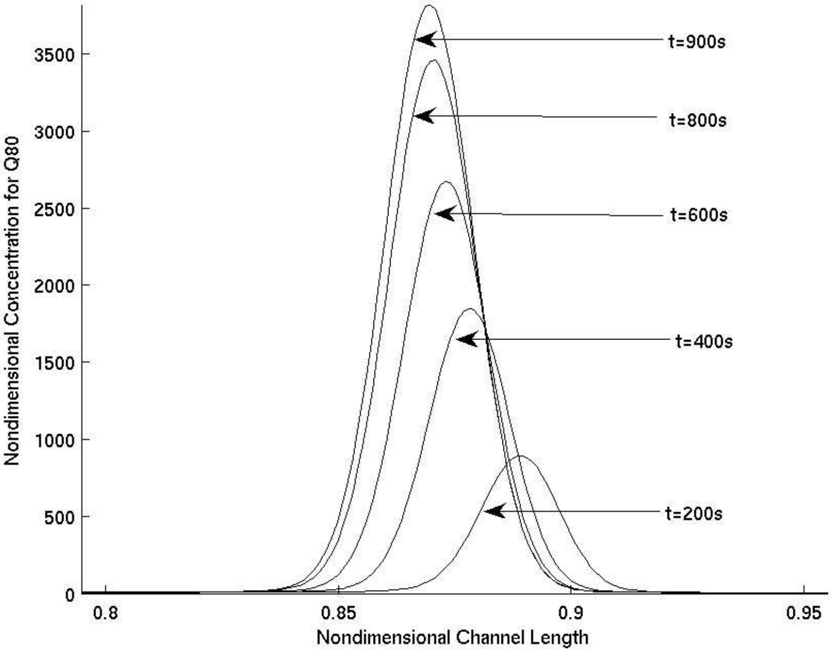


Figure 7.5: The focused peak for the ampholyte Q50 with pI=7 begins stabilizing between 800-900s

The focusing of the protein with pI=7, is shown in Fig. 7.6, with the focusing time being the same as for the ampholyte with pI=7.0. This is to be expected because we have assigned the same mobility, diffusivity and titration curve parameters to this protein as the carrier ampholytes.

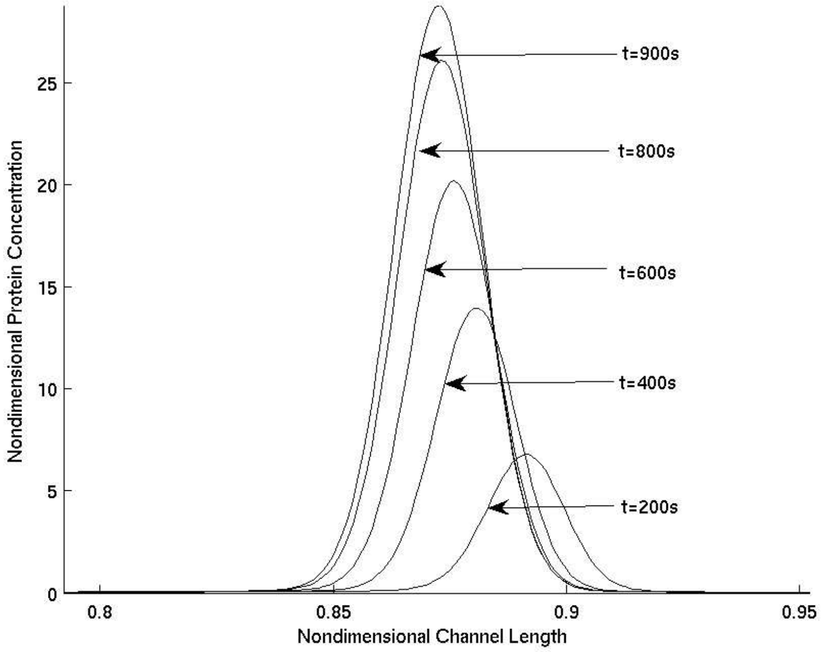


Figure 7.6: The focused peak for the protein with pI=7 begins stabilizing between 800-900s

The focused peaks of 4 of the carrier ampholytes are shown in Fig. 7.7 and the resultant (experimentally expected) step-ladder pH graph in Fig. 7.8.

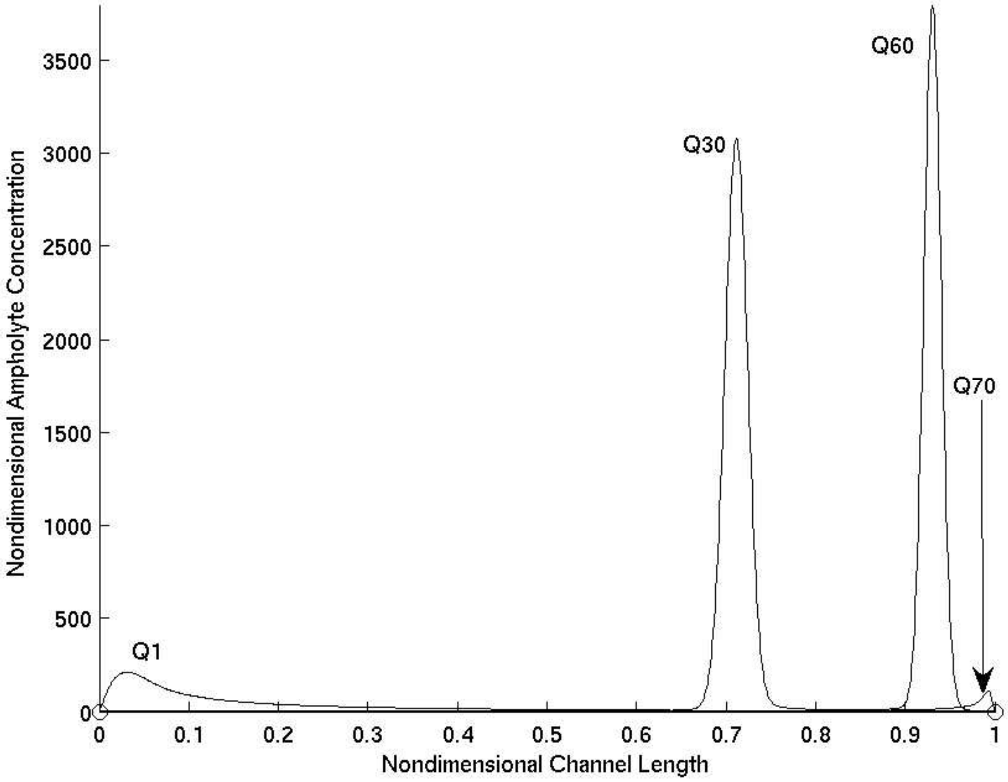


Figure 7.7: The concentrations of the 4 ampholytes Q1, Q30, Q60 and Q70 at t=15 minutes. The ampholyte Q70 (and higher) have begun drifting out of the channel by this time.

The pH gradient towards the basic end of the channel (near pH=8) has begun degrading because the carrier ampholytes with isoelectric points in that range have begun washing out of the channel. This kind of degradation of the pH gradient at the cathodic (basic) end of the channel is also observed experimentally and is termed as *cathodic drift* [81]. To this day, this is a big obstacle in the successful

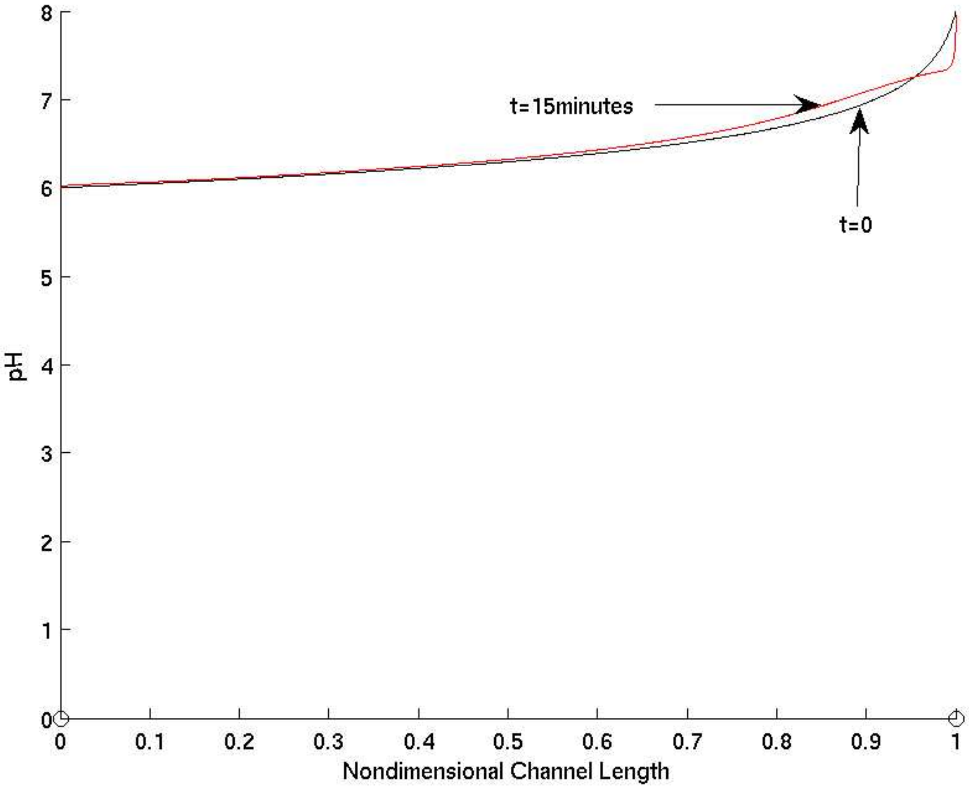


Figure 7.8: pH step ladder graph at the initial time and at t=15 minutes. The pH gradient at the basic end of the graph begins degrading by the end of the simulation because the carrier ampholytes in that region have begun washing out of the channel.

focusing of the ampholytes with pIs greater than 7 [83].

The causes of cathodic drift have been listed as including electroosmosis and electrophoretic flux [70], but it is not yet well understood (the cathodic drift time predicted by electroosmosis and electrophoretic flux exceed the experimental cathodic drift time in [112]). In our simulations, cathodic drift can be seen by the stretching out of the steps' plateau in the pH gradient near the acidic end, and shrinking near the basic end of the channel. Our simulations predict a faster occurrence of cathodic drift as compared to [112] because we have modeled the boundary conditions as Dirichlet-type. This allows for a greater ampholyte flux through the ends of the channel, than if the ends were semipermeable like in an actual experiment.

The experimental focusing time for the same conditions as in our simulations, was reported by Thormann et al. [112] as 'similar' to their computational results. They (and other research groups including [61]) have reported that exact experimental verification of focusing time is difficult without further advances in accurately imaging the movement of dyes in the channel.

As noted before, the main advantage of computer simulations is in being able to design better IEF experiments by, say, understanding the sensitivity of focusing behavior to CA molecule parameters like titration curve and mobility coefficient. This will ultimately help chemists in designing better CA molecules that can focus proteins with closely spaced pIs without overlap between each other. For such optimization problems, one needs to have modeling approaches that are fast even at the expense of omitting some detail in the physics (like we have in the hydrogen constraint and assumptions of equilibrium chemistry). Our simulations were run on the COMSOL software that ran on a 2.3 GHz dual Opteron processor (using Linux). For our simulations, with the experimental focusing time of 15 minutes, our computer simulation takes close to 25 minutes, whereas other groups have reported simulation times [111], [61] varying between 20 hours for the pH range 5-8 and 40 hours for the pH range 3-10. Our simulations for the pH range 5-8 matched the focusing time for the ampholyte pI=7 given in [111], but our cathodic drift caused greater degradation of the pH gradient near pH=8 than [111]. However, we believe that on the basis of the simulation time taken for the 5-8 case (20 hours at least), we can conclude the following:- With the assumptions that we have made for modeling IEF, and at the expense of faster cathodic drift for the ampholytes that are nearer to the basic end of the channel, our model of the IEF process predicts focusing times (especially for ampholytes that are further away from the basic end) with a much lower simulation time. Hence, our modeling approach is better suited for time-intensive optimization runs of the CA molecule chemical design (for example, sensitivity of focusing time to titration curves) or for examining the effects of changing other parameters of the IEF experimental setup (like channel length or applied voltage).

Due to the fast simulation time (25 minutes) of our approach, the way in which we have modeled IEF in this chapter, already qualifies as a 'reduction technique'. However, many of the current experiments in IEF have thousands of carrier ampholytes over wider pH ranges. Moreover, the electric field strengths are double or triple than the ones which we have used in our simulations (or the comparable experiments in [112]), which means the ratio of convective to diffusive forces - the Peclet number - is double or triple in value as compared to our simulations. The number of elements of the stiffness matrices in the finite element formulation (ignoring finer grids that would be needed for higher Peclet numbers), will vary at least as the square of the number of ampholytes. Collectively, the larger number of ampholytes, higher Peclet number and higher pH gradient will make the computational time prohibitive for more complex simulations of IEF, especially when they need to be incorporated in optimization runs. The peak memory requirements for our simulation was around 1.4GB which again, would only increase in more intensive simulations.

We note that the simulation time and memory allocation that quantify our simulation performance are all still restricted to 1-dimensional channel geometries. Experiments for 2D IEF, in which 2-dimensional channel geometries were used for focusing of ampholytes with closer pIs [19], will also require simulations for better design of channel geometries. Even for as few as 5 ampholytes, we observed [62] that 2-dimensional (tapered) channel geometries require over 10 hours of simulation time (in part this was due to the increased number of grid points required at the inflection points in the 2-dimensional geometry). There is clearly a further need for model reduction of the IEF problem, which we investigated with the use of proper orthogonal decomposition. We present this in the next chapter.

Chapter 8

Model Reduction for Nonlinear Dynamics - Proper Orthogonal Decomposition

In the previous chapter we saw how even for the case of *just* 100 ampholytes in a relatively small pH range of 6-8, the finite element simulation took close to a half hour and over 1.4 GB of memory. Actual IEF experiments can have a few thousand ampholytes and a pH range of 3-10 [61]. The amount of time needed for a single such simulation can run into days with orders of magnitude increase in the amount of memory (in another group's work [111] 40 hours were needed for simulating a 120 ampholyte IEF experiment with pH range 3-10). This is notwithstanding the simulation time needed for more complex channel geometries like 2D which are currently being investigated [97]. Clearly, this kind of computational power is not available to every biochemist who wants to design better experiments, by say, designing CA molecules with sharper titration curves that can separate proteins having only a small difference in their isoelectric points. If the correct mathematical requirements for sharper titration curves are known, then the CA molecules can be better designed. The above reasons necessitate the need for reduced order modeling of the IEF problem.

The IEF problem has a nonlinear constraint that governs the hydrogen ion evolution as well as nonlinearities in the convective portion of the differential equations that govern each ampholyte. The projection based methods that are successful for model reduction are techniques like balanced truncation, Krylov subspace techniques and Pade approximants. Most of these techniques are suited only for full order models with linear dynamics. Modifications of these have been used with some success for weakly nonlinear problems [33] that use the Jacobians of the nonlinear functions to attain a linearized FOM to which KSTs are applied. However, such techniques have additional costs associated with them, depending on the kind of modifications that are required, that make them impractical for the reduction of most nonlinear problems.

We have investigated the use of POD for model reduction of the IEF problem. POD was first suggested as a model reduction tool in CFD by Lumley in 1970 [59]. It was developed in probability theory in 1978 by Loeve [56]. Interest in it as a tool in model reduction greatly increased after its use in modeling coherent structures by Sirovich [99] in 1987 and then later in modeling the turbulent boundary layer in [8] and for modeling compressible flows in [114]. POD is a systematic model reduction technique that is popularly used for nonlinear problems. As explained later in Sec.8.1, POD is not dependent on the full order model being linear. Moreover, using POD, one can make a clear, physical choice of the mode shapes and include them in the ROM subspace on the basis of the strength of their individual contribution to the evolution of the dynamics.

Even though the use of POD did not succeed in model-reducing the IEF problem it motivated us to look for shortcomings of the traditional POD procedure when applied to stiff problems. This resulted in our observation of a particular kind of shortcoming (termed *twist*) which we explain in the next chapter, where we also show how this shortcoming can be successfully resolved by augmenting the traditional POD algorithm.

In this chapter, we first give a detailed explanation of the main theorem in POD and the properties of the POD reduced order model. We then discuss Galerkin projection, and how it can be used to project the full order dynamics on the reduced subspace of the POD modes. Finally, we explain our investigation of applying the POD procedure for reducing the IEF problem and why we believe this leads to an ill-conditioned problem.

8.1 Main theorem in POD

The geometrical question that is answered by POD can be stated with the help of Fig. 8.1. The *reduced order* curve x_r is the projection of the full order curve x on the subspace $\mathbb{S}$. The full order curve x evolves according to the full order dynamics given by $\dot{x} = f_{FOM}(x)$ (where $f_{FOM}(.)$ is the full order vector field) and the reduced order curve x_r evolves according to the reduced order dynamics given by $\dot{x_r} = f_{ROM}(x_r)$ (where $f_{ROM}(.)$ is the reduced order vector field). Placing this in context of the turbulence example that we described in Sec. 1.4, if x describes the trajectory of the infinite dimensional vector that describes the state of the chaotic turbulent flow, then x_r describes the projection of x onto the low dimensional chaotic attractor. The question answered by POD is:

Given the full order curve x that evolves in a high dimensional space (for example,

$\mathbb{R}^3$ in Fig. 8.1 and $\mathbb{R}^{2618}$ in our IEF problem, as explained later), what is the subspace of a smaller, fixed dimension (a 2-dimension plane in Fig. 8.1), and the associated reduced order dynamics given by $\dot{x_r} = f_{ROM}(x_r)$, which minimizes the induced 2-norm error $||x - x_r||_2$ which is given by

$$||x - x_r||_2^2 = \int_{t_i}^{t_f} ||x(t) - x_r(t)||_2^2 dt \tag{8.1}$$

where the initial time t_i and final time t_f denote the starting and ending instant of the evolution of the full and reduced order trajectory.

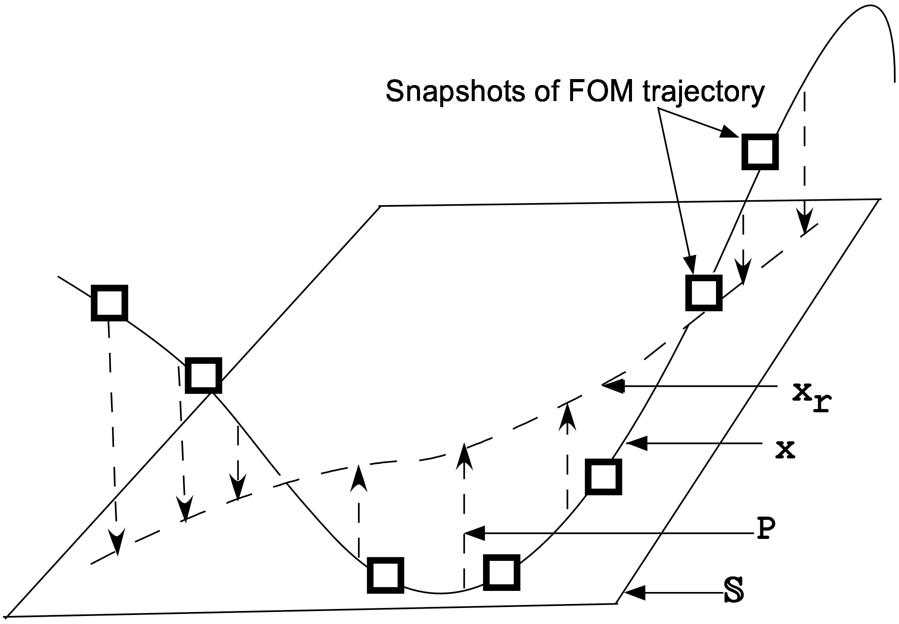


Figure 8.1: The full order curve x evolves in a high dimension space (say, $\mathbb{R}^3$, in this illustration). A projection $\mathbb{P}$ of the curve x onto a smaller dimensional space $\mathbb{S}$, gives the *reduced order* curve x_r. The optimal subspace $\mathbb{S}$ (as per the criterion in Eqn. 8.1), is generated by POD. The squares denote 'snapshots' of the FOM trajectory, i.e., values of the values of the state vector in the FOM, which will be used to create the optimal subspace $\mathbb{S}$ via POD.

In Fig. 8.1, the concepts of projection and higher dimensional space have been depicted in terms of the familiar space $\mathbb{R}^3$. However, the main theorem of POD, and the properties of the reduced order models, hold true for appropriate extensions in more general function spaces [37]. In the explanation of the main properties of POD given below, we will restrict ourselves to the familiar space $\mathbb{R}^N$, but we note that an analogous extension to more general function spaces is well understood [37].

The POD technique uses data - either from experiments or from computer simulations - in constructing smaller dimensional subspaces on which one can project the full order dynamics. Such data sets are commonly called *snapshots* in POD literature [99] (as shown in Fig. 8.1). These snapshots are elements of $\mathbb{R}^N$. For example, for the example of applying POD in order to understand the structure of the chaotic attractor of the turbulent flow, one would first create a full order finite element formulation model of the flow. This model will have 3 variables (from the velocity) plus one variable (from the temperature) for each of the grid points, for a total of 4 variables for each grid point. Thus, for M grid points in the simulation, this would mean that the snapshot at a particular time t, would lie in $\mathbb{R}^{4M}$.

For a general problem, let $U(x,t)$ denote the full order data in $\mathbb{R}^N$ (where N is the total number of degrees of freedom in a finite element formulation and could hence be $\approx 10^3$ or higher) which one needs to approximate in a smaller dimensional subspace $\mathbb{S}$. The size of N would be typically decided by the number of grid points and the number of independent variables as explained in the turbulence chaotic attractor example above. Let (x,y) denote the inner product of two vectors in $\mathbb{R}^N$ is given by

$$(x, y) = \sum_{i=1}^{i=N} x_i \cdot y_i \tag{8.2}$$

where x_i and y_i are individual elements of the vectors x and y respectively. The main idea in POD is to find a subspace of a small (and fixed) dimension m, whose orthonormal basis vectors form the basis of the subspace ϕ, so that the following maximum is reached:

$$\underset{(\eta,\eta)=1}{max} \langle |(U^{'}, \eta)| \rangle^2 = \langle |(U^{'}, \phi)| \rangle^2 \tag{8.3}$$

where $(U^{'}, \eta)$ denotes the inner product of the basis η with $U(x, t)$, and $\langle z \rangle$ denotes the time average of the quantity z. The optimal subspace ϕ (of dimension m) that is given by POD, gives the least error on average, as compared to any other subspace of the same (or lower) dimension.

Any time average (instead of the simple linear time average) or inner product (instead of the canonical one given in Eqn. 8.2) can be used. The only requirement of the time average $\langle . \rangle$ is that it commutes with the inner product $(.)$. In most applications, including in this thesis, the time average $\langle . \rangle$ is typically chosen to be a (weighted) arithmetic mean, which does commute with the inner product given in Eqn. 8.2.

In order to solve the constrained optimization problem in Eqn. 8.3, we can use the following Lagrangian function,

$$\mathcal{L}(\eta, \lambda) = \langle |(U^{'}, \eta)|^2 \rangle - \lambda((\eta, \eta) - 1) \tag{8.4}$$

where λ is the Lagrange multiplier for the constraint on the normality of the basis function η. The optimal basis ϕ which maximizes the cost function in Eqn. 8.3 will satisfy the following condition for all basis variations ζ:

$$\frac{d}{d\delta}\mathcal{L}(\phi + \delta\zeta, \lambda)|_{\delta=0} = 0 \tag{8.5}$$

Differentiating with respect to δ and setting the result to zero, and using the fact that the time average commutes with the inner product, we get that the optimal solution ϕ must solve the following eigenvalue problem:

$$\mathcal{R}\phi = \lambda\phi \tag{8.6}$$

where the kernel $\mathcal{R}$, is given by:

$$\mathcal{R} = \langle |(U^{'}U)| \rangle \tag{8.7}$$

In a practical application of POD, the kernel $\mathcal{R}$ that is based on the continuous solution $U(x,t)$ in Eqn. 8.7, has to be approximated with a finite number of snapshots of the solution. Suppose there are k snapshots of the solution, that are available either through a time series simulation or from an experiment. We denote each snapshot as $U_i, i = 1,..,k$, with $U_i \in \mathbb{R}^N$. One can approximate the kernel with the outer product of the snapshots in the following way:

$$\mathcal{R} = \frac{1}{k}\sum_{i=1}^{i=k} U_i(x)U_i(x)^{'} \tag{8.8}$$

With this approximation (based on the finite number of snapshots $U_i, i =$

$1, .., k$ instead of the continuous $U(x,t)$), the optimal basis ϕ are the eigenvectors of $\mathcal{R} = \frac{1}{k}\sum_{i=1}^{i=k} U_i(x)U_i(x)'$. It can be shown that the maximum described in Eqn. 8.3, is attained by the basis ϕ. This is expressed in the following theorem [37] which is valid for the infinite dimensional case (with continuous $U(x,t)$), but which we present here for the practical case of the finite snapshot set.

Theorem 8.1.1 *Let* $U = (U_i : U_i \in \mathbb{R}^N, i = 1, .., k)$ *be an ensemble of* k *snapshots of some process, and let* $\mathcal{R}$ *be the covariance matrix of the snapshots as defined in Eqn. 8.8. Let* λ_j *be the ordered eigenvalues of* $\mathcal{R}$*, with* $\lambda_1 \geq \lambda_2 ... \geq \lambda_k \geq 0$*. Let* $\rho_{\mathbb{S}} U$ *be the projection of* U *onto some* $m(\leq k)$ *dimensional subspace* $\mathbb{S}$*. Then the minimum value of the projection error* $||U - \rho_{\mathbb{S}} U||$ *over all* m *dimensional subspaces* $\mathbb{S}$ *is given by* $\sum_{j=m+1}^{j=k} \lambda_j$*. In addition, the basis of the minimizing subspace* $\mathcal{S}$ *is given by the span of the eigenvectors* $\phi_1, .., \phi_m$ *corresponding to the eigenvalues* $\lambda_1, ..., \lambda_m$*.*

$\mathcal{R}$ is symmetric positive semi-definite and hence we are guaranteed real non-negative eigenvalues λ_j. Hence in order to have a maximum 2-norm error of say 1% , we set the value $\lambda_{perc} = 99\%$ and choose the smallest value m, and the subspace $\mathcal{S}$ spanned by the corresponding eigenvectors $\phi_1, .., \phi_m$ such that the following condition is satisfied

$$\frac{\sum_{j=1}^{j=m} \lambda_j}{\sum_{j=1}^{j=k} \lambda_j} \geq \lambda_{perc} = 0.99. \tag{8.9}$$

8.2 Cheap computation of the optimal basis: The method of snapshots [99]

The matrix of the eigenvalue problem in Eqns. 8.6 and 8.8 is of size $N \times N$, where N is the size of the snapshot vector U_i. N can be very large depending on the number of grid points in the problem. We have 238 grid points in our IEF simulation and we perform the simulation for 10 ampholyte species and hydrogen. Hence, we have a total of $N = 238 \times 10 + 238 = 2618$ degrees of freedom. In order to construct the kernel $\mathcal{R}$ in Eqn. 8.8, one needs to compute outer products of k vectors with themselves, with each vector of size N. If each element of a single outer product matrix takes $O(1)$ computational time (this is the time taken for computing the inner product of two vectors of size N), then since there are N^2 entries in each outer product and k outer products, the total computational cost for constructing $\mathcal{R}$ is $O(kN^2)$. In many problems, a small number of snapshots can be a good representative set of the full order dynamics. In such a case (when the number of snapshots k is much lesser than N), instead of the expensive *direct* method of constructing the kernel $\mathcal{R}$ which would take $O(kN^2)$ steps, one can make use of the following trick, called the *method of snapshots* (first proposed by Sirovich [99]), for computing ϕ.

From the definition of the kernel $\mathcal{R}$ (Eqn. 8.8), we can see that the range of $\mathcal{R}$ is contained in the span of the snapshots U_i. For the case of the kernel being chosen as a weighted arithmetic mean of the snapshots, we have:

$$\mathcal{R} = \sum_{i=1}^{i=k} \alpha_i U_i(x) U_i(x)^{'} \tag{8.10}$$

with weights $\alpha_i, i = 1, .., k$ satisfying $\sum_{i=1}^{i=k} \alpha_i = 1$. We can see that for any vector z,

$$\begin{aligned} \mathcal{R}z = (\sum_{i=1}^{i=k} \alpha_i U_i U_i^{'}) z &= \sum_{i=1}^{i=k} \alpha_i U_i (U_i^{'} z) \\ &= \sum_{i=1}^{i=k} c_i U_i \end{aligned} \tag{8.11}$$

where the scalar $c_i = \alpha_i (U_i^{'} z)$. Since $\mathcal{R}\phi = \lambda\phi$, we can see that the optimal basis ϕ lies in the snapshots U_i. Hence, there exists scalars $b_{l,j}$ such that any of the basis vectors ϕ_l can be written as:

$$\phi_l = \sum_{j=1}^{j=k} b_{l,j} U_j \tag{8.12}$$

Hence the eigenvalue problem $\mathcal{R}\phi_l = \lambda_l \phi_l$ simplifies to:

$$\begin{aligned} \mathcal{R} \sum_{j=1}^{j=k} b_j U_j &= \lambda_l \sum_{j=1}^{j=k} b_{l,j} U_j \\ \sum_{i=1}^{i=k} \alpha_i U_i U_i^{'} \sum_{j=1}^{j=k} b_{l,j} U_j &= \lambda_l \sum_{j=1}^{j=k} b_{l,j} U_j \\ \sum_{i=1}^{i=k} \alpha_i U_i (\sum_{j=1}^{j=k} U_i^{'} U_j b_{l,j}) &= \lambda_l \sum_{i=1}^{i=k} b_{l,i} U_i \end{aligned} \tag{8.13}$$

Since the above relation holds for each of the snapshots U_i, we see that it is enough to solve the following eigenvalue problem:

$$Vb = \lambda b \tag{8.14}$$

where $b = [b_1 b_2 ... b_k]'$ and V is a $k \times k$ matrix with each entry $V_{ij} = \alpha_i (U_i' U_j)$.

The k eigenvalues $\lambda_1, .., \lambda_k$ of the kernel V are exactly the same as the k largest eigenvalues of the kernel $\mathcal{R}$, with the rest of the $N - k$ eigenvalues of $\mathcal{R}$ equal to zero since $\mathcal{R}$ has rank k. For a given eigenvalue λ_l of V and the corresponding eigenvector $b_l = [b_{l,1} b_{l,2} ... b_{l,k}]'$, we can use the snapshots U_j to construct the required basis vectors ϕ_l using Eqn. 8.12. Since constructing V has a cost of order $O(Nk^2)$ (since we compute N outer products of vectors, with each outer product matrix of size $k \times k$), this is much cheaper than constructing $\mathcal{R}$ (which has a cost of order $O(kN^2)$), when $k << N$ (number of snapshots much lesser than the dimension of the full order space), which is frequently the case in POD applications.

8.3 Galerkin projection

The final tool that one needs in order to perform model reduction via POD is an appropriate projection technique. In this section, we describe the Galerkin projection technique and the way it is used for projecting the dynamics of the full order dynamics (a large finite dimensional space in the case of finite element models), onto a lower dimensional space, where one tracks the evolution of a finite set of ODEs. The explanation of Galerkin projection in terms of a general Hilbert space (instead of the special case $\mathbb{R}^N$) is given below. $\mathbb{R}^N$ is a Hilbert space with inner product defined as in Eqn. 8.2 so the results in this section follow exactly for real spaces by replacing $\mathbb{H}$ with $\mathbb{R}^N$. The geometric picture for the Galerkin projection is shown in Fig. 8.1 itself.

A Hilbert space $\mathbb{H}$ is a space of all vector valued functions which are smooth (belong to C^{∞}, i.e. the function and all its derivatives are continuous) and has the following norm (inner product) for any two elements u and v which belong to $\mathbb{H}$.

$$(u, v) = \int\limits_{\Omega} u(x) \cdot v(x) dV \tag{8.15}$$

Dynamical systems which evolve on a Hilbert space $\mathbb{H}$, can be described in the following form:

$$\frac{du}{dt} = X(u) \tag{8.16}$$

where $X(.)$ represents a nonlinear operator that may involve spatial derivatives and/or integrals. Here $u(t) \in \mathbb{H}$, and $X(u)$ is a vector field on $\mathbb{H}$. Any model reduction technique aims to 'shadow' the full order dynamics given by Eqn. 8.16 on a reduced subspace in the following way:

$$\frac{dv}{dt} = X_{\mathbb{S}}(v) \tag{8.17}$$

where $X_{\mathbb{S}}(v)$ is a vector field on a subspace $\mathbb{S} \in \mathbb{H}$, and $v(t) \in \mathbb{S}$. The Galerkin projection operator $\mathbb{P}$ which achieves the projection $\mathbb{P} : \mathbb{H} \rightarrow \mathbb{S}$, is described by the basis of the subspace $\mathbb{S}$. Let $\phi_1, .., \phi_n$ form an orthonormal basis of $\mathbb{S}$. Then we have $\mathbb{P} = \Phi \cdot \Phi^T$, where $\Phi = [\phi_1 \phi_2 ..... \phi_n]$. The vector field $X_{\mathbb{S}}$ is given by

$$X_{\mathbb{S}}(v) = \mathbb{P} \cdot X(v) \tag{8.18}$$

When applied to POD, we track the evolution of the full order PDE vector field X with ODEs on the reduced subspace $\mathbb{S}$, on which the vector field is given by $X_{\mathbb{S}}$. This is achieved in the following manner. Eqn. 8.17 is written in the basis of the reduced subspace's coordinates. In our case, this basis is found from the POD process. Hence, we have

$$v(t) = \sum_{k=1}^{k=n} a_k(t) \cdot \phi_k \tag{8.19}$$

Using Eqns. 8.17 and 8.19, and taking inner products with each of the basis vectors ϕ_k, we get

$$\sum_{j=1}^{j=n} \frac{d(a_j)}{dt} \cdot (\phi_j, \phi_k) = (X_{\mathbb{S}}(v(t)), \phi_k) \quad k = 1, ..., n \tag{8.20}$$

Since the basis vectors are orthonormal, we have

$$\frac{da_k}{dt} = (X_{\mathbb{S}}(v(t)), \phi_k) \quad k = 1, ..., n \tag{8.21}$$

Hence, we track the evolution of the original full order Eqn. 8.16, with the n ODEs given by Eqn. 8.21.

We see then, the need for the snapshots of the dynamical system to be well chosen to represent the range of all possible dynamical behavior that one wishes to capture in the reduced order model. If the behavior of the system is not represented in the set of snapshots (for example, the chaotic attractor of the turbulence dynamics at a particular Reynolds number), then one should not expect to see that behavior reproduced in the reduced order model. Many times, the strategy that is adopted is taking snapshots at every Δt time step in the evolution of the full order dynamics with the value of Δt set to a small enough value so that no appreciable change occurs in the dynamics of the full order model within that time interval.

The Galerkin projection technique is independent of the subspace $\mathbb{S}$. It is the main theorem of POD (Sec. 8.1), which provides the appropriate subspace where this reduction is optimal. We now have the tools for the complete POD algorithm, which we will term as the *Traditional* POD algorithm (as originally used by Sirovich for the problem of turbulence in fluid flow) to differentiate it from the *Augmented* POD algorithm which we will describe in the next chapter. We state the traditional POD algorithm in Algorithm 3 and in the next section, we discuss our application of this algorithm to the IEF problem.

Algorithm 3 Traditional POD algorithm

1. Generate k snapshots $U_i \in \mathbb{R}^N : i = 1, .., k$ that span the time interval and parametric range of interest.

2. Set the value of the 'energy retained' parameter λ_{perc}

3. Compute the POD basis ϕ using the method of snapshots described in Eqns. 8.13 and 8.14.

4. Arrange the k eigenvectors of the POD basis ϕ according to the decreasing values of their respective eigenvectors. Choose the first m of these basis vectors so that the condition given in Eqn. 8.9 is satisfied. Denote this set of m largest basis vectors as the reduced order subspace $\mathbb{S}$.

5. Project the full order dynamics onto $\mathbb{S}$ by the Galerkin projection method using Eqn. 8.21 to get the reduced order model.

8.4 POD implementation for the IEF problem

We investigated the use of POD on a smaller IEF problem than the one shown in the previous section. The main difference was that we reduced the number of ampholytes to $N = 10$ and we maintained the pH range between the acid and base reservoir at $pH_L(acid) = 6$ and $pH_R(base) = 8$. The reduced number of ampholytes results in a system of 10 PDEs coupled with the constraint for the hydrogen ion. The reduced pH range resulted in a less nonlinear variation of the hydroigen ion constraint. The isoelectric points were linearly spaced in the range $pH_L - pH_R$.

For this problem, we found that most of the ampholytes retain their focused peaks till time $t = 0.1$ before washing out of the channel. The plots for all the ampholytes and hydrogen at $t = 0.1$ are shown in Fig. 8.2. Most of the ampholytes begin to concentrate near their pIs by time $t = 0.02$ as shown in Fig. 8.3. For this reason, we decided to create a ROM for the time period $0.02 < t < 0.1$.

We created separate sets of modes for each of the ampholytes and for hydrogen. One way to compute the hydrogen ion evolution in the ROM is to use a nonlinear constraint solver to solve for the hydrogen constraint equation. However, this took much more time than computing the full order model. Hence, we needed to create POD modes for the hydrogen ion constraint equation. Using the POD algorithm 3, we used the 9 snapshots of the ampholyte and hydrogen at times $t = 0.02, 0.03, .., 0.1$. We observed that the underlying shape of most of the ampholyte focused peaks do not change between consecutive snapshots and providing more closely shaped snapshots do not ultimately provide any more (or better) mode shapes.

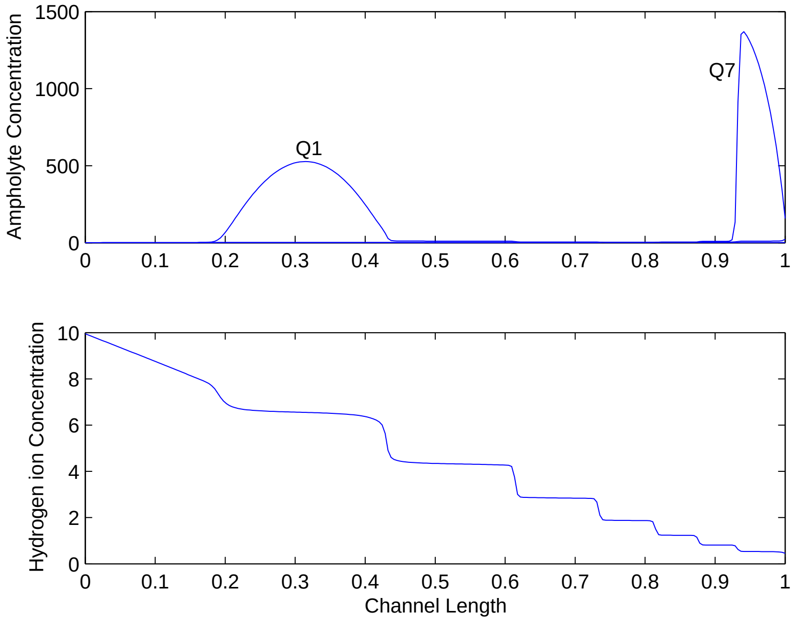


Figure 8.2: Plots for non-dimensionalized concentrations of ampholytes Q1 and Q7 and $\hat{H}$ at $t = 0.1$. The hydrogen ion concentration is non-dimensionalized as described in the previous chapter - $\hat{H} = \frac{H}{10^{-7}}$. Ampholytes Q8, Q9, and Q10 have begun washing out of the channel and their concentrations in the channel are much lower than the concentrations of the other ampholytes.

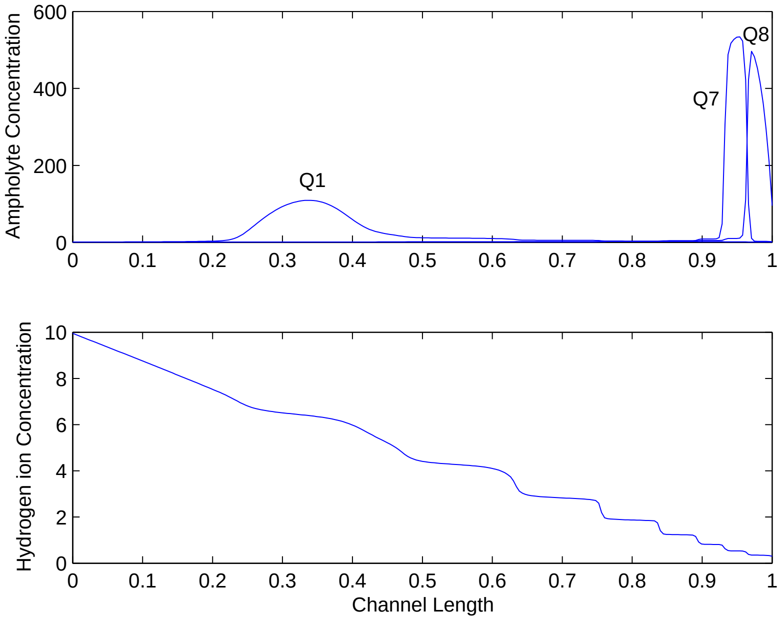


Figure 8.3: Plots for ampholytes Q1, Q7, Q8 and $\hat{H}$ at $t = 0.02$. Ampholytes Q9, and Q10 have begun washing out of the channel and their concentrations in the channel are much lower than the concentrations of the other ampholytes.

For better accuracy of the reduced order model, and also to enable the Dirichlet boundary conditions for the ampholytes to be automatically satisfied in the ROM, we created the ROM for only the perturbations in the ampholytes' behavior. We did this by subtracting the mean of the snapshots of a particular ampholyte from *each of the snaps* of that ampholyte to get the 'perturbed snaps'. The POD algorithm was applied to these 'perturbed snaps'. By setting the retained energy parameter λ_{perc} to $\lambda_{perc} = 99\%$ for each of the ampholytes snapshots, we get the following differential algebraic equation (DAE) :

$$\begin{aligned}\frac{da_i^k(t)}{dt} &= F_i(t) = (A_i(x,t), \phi_i^k(x)) \\ 0 &= G(t) = (B(x,t), \psi_H^j(x)) \qquad (8.22)\end{aligned}$$

where (x,t) is the spatio-temporal location of the concerned variable, $\phi_i^k(x)$ is the k^{th} mode of the i^{th} ampholyte Q_i and $a_i^k(t)$ is the coefficient multiplying that ampholyte mode. $\psi_H^j(x)$ is the j^{th} mode of the hydrogen ion. $(A_i(x,t), \phi_i^k)$ and $(B(x,t), \psi_H^j(x))$ are the inner products of $A_i(x,t)$ and $G(t)$ with the respective ampholyte and hydrogen ion modes where $A_i(x,t)$ and $G(t)$ are given by

$$\begin{aligned}A_i(x,t) &= D_i(\sum_{k=1}^{k=p_i} a_i^k(t)\frac{d^2\phi_i^k(x)}{dx^2}) - M_iE(\frac{d}{dx}(\sum_{k=1}^{k=p_i} f_i(\sum_{j=1}^{j=q} a_H^j(t)\psi_H^j(x))a_i^k(t)\phi_i^k(x))) \\ B(x,t) &= H_o(x) - \sum_{j=1}^{j=q} a_H^j(t)\psi_H^j(x) - \sum_{i=1}^{N}(\sum_{k=1}^{k=p_i} a_i^k(t)\phi_i^k(x))f_i(\sum_{j=1}^{j=q} a_H^j(t)\psi_H^j(x))\end{aligned}$$

where $a_H^j(t)$ is the mode coefficient that gives the time evolution of the hydrogen ion mode $\psi_H^j(x)$. The i^{th} ampholyte Q_i has a total of p_i POD modes. The hydrogen ions has a total of q POD modes. N is the total number of ampholytes. D_i, M_i and $f_i(.)$ are the diffusion coefficient, mobility coefficient, and the titration curve of the ampholyte Q_i and E is the constant electric field.

The ampholyte mode shapes for one such ampholyte $Q4$ is shown in Fig. 8.4. The hydrogen ion mode shape is shown in Fig. 8.5.

For each of the 10 ampholytes, we get an average of close to 4 modes, for a total of 39 modes. For the hydrogen ion, we get 5 modes. Thus, the resultant DAE is of the form shown in Eqn. 8.22 with a total of 39 DEs and 5 AEs, with each DE

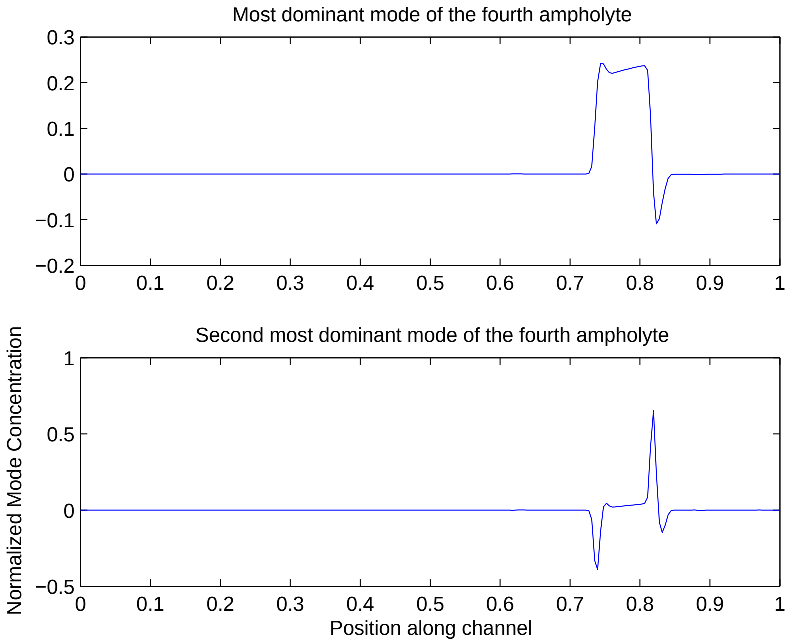


Figure 8.4: Q4 ampholyte mode shape

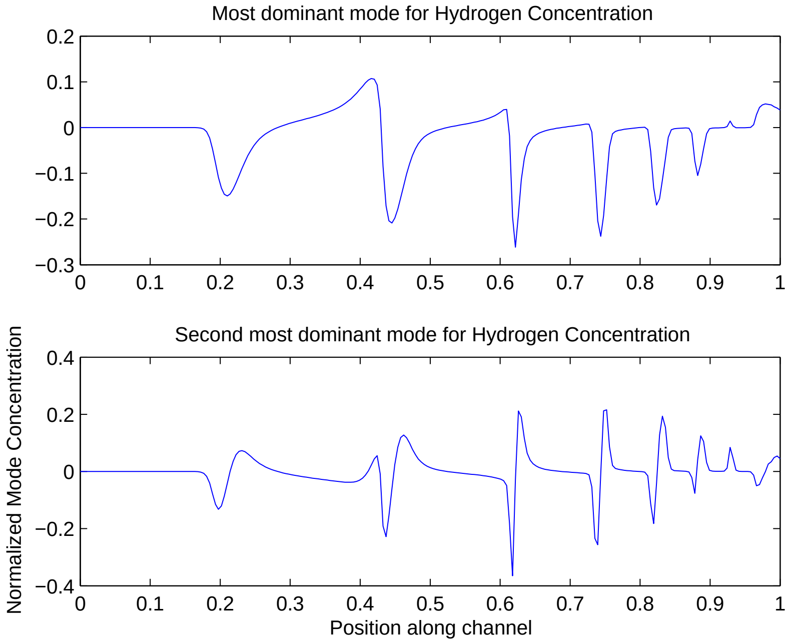


Figure 8.5: Q4 ampholyte mode shape

(or AE) as the governing equation for the corresponding mode.

We analytically compute the Jacobian, which we denote in the following way.

$$J(a_Q, a_H) = \begin{pmatrix} \frac{\partial F}{\partial a_Q} & \frac{\partial F}{\partial a_H} \\ \\ \frac{\partial G}{\partial a_Q} & \frac{\partial G}{\partial a_H} \end{pmatrix}.$$

where the time and space dependence of the variables are suppressed to avoid clutter. F is the ampholyte vector field projected on the ampholyte's POD modes, G is the projection of the hydrogen ion's constraint on the hydrogen ion's POD modes, a_Q is the vector of mode coefficients of all the ampholytes' POD modes, and a_H is the vector of mode coefficients of the hydrogen ion. In what follows, we explain how the weaker POD modes for the hydrogen ion constraint are needed to retain the accuracy of the ROM, but their retention causes the condition number of the constraint's Jacobian to worsen and prevents the ROM-DAE from evolving beyond a very small time step.

8.4.1 Need for a balance between ROM simulation error and the condition number of the constraint Jacobian $\frac{\partial G}{\partial a_H}$

The solvability of a DAE is determined by whether or not the matrix $\frac{\partial G}{\partial a_H}$ is singular [13]. The DAE is said to have index=1 if we can construct a differential equation (as opposed to the available algebraic equation) for the dependent variable in the constraint equation by differentiating the constraint once (with respect to the independent variable) and using the nonsingularity of $\frac{\partial G}{\partial a_H}$ (along with the implicit value theorem). If the minimum number of times that the constraint has to be differentiated in order to construct a differential equation for the dependent variable in the constraint is n, then the DAE is said to have index n. Solvers for DAEs with index=1 are available, but as the index of the DAE increases to 2 or more, the solvers become less reliable.

From the analytical Jacobian that we computed, we checked that constraint Jacobian given by the matrix $\frac{\partial G}{\partial a_H}$ is non-singular and hence the DAE 8.22 has index=1 at the initial condition. Since the algebraic constraint is essentially a conservation law for the hydrogen ions, it is automatically satisfied at the initial time.

For our problem, we have relied on the MATLAB stiff solvers ode15s, ode23t, ode23s, and ode23tb. The results stated in this section were achieved by using ode15s but similar results were attained by using the other solvers. The condition number for the DAE Jacobian $J(a_Q, a_H)$ is 10^7, but problems with higher condition numbers have been solved, in particular for problems dealing with combustion [79].

However, the condition number of just the constraint's Jacobian $\frac{\partial G}{\partial a_H}$ has to be maintained at a low value throughout the simulation. This follows from the nature of the focusing process. At the sharp ends of each step in the step ladder-like hydrogen ion concentration profile, any error in the hydrogen ion concentration at time t_i will translate into an error in the concentration profiles of the two ampholytes that border that sharp end of the step. Such an error in the ampholyte concentration at the sharp end of the step at time t_{i+1} will then cause a larger error in the hydrogen ion concentrations at subsequent time steps. Hence, if the constraint equation for the hydrogen ions is to be strictly satisfied (within a given error tolerance), then it is crucial that the condition number of constraint's Jacobian $\frac{\partial G}{\partial a_H}$ is low, because it is the constraint's Jacobian that determines the error in the DAE solver [13].

In our problem, the large condition number of $\frac{\partial G}{\partial a_H}$ prevents the solver from proceeding beyond a small initial time interval. With the POD mode coefficients set at the values attained by projecting the 9 snaps at $t = 0.02, 0.03, .., 0.1$, we found that the condition number of $\frac{\partial G}{\partial a_H}$ increases to as high as 630 as shown in Fig. 8.6.

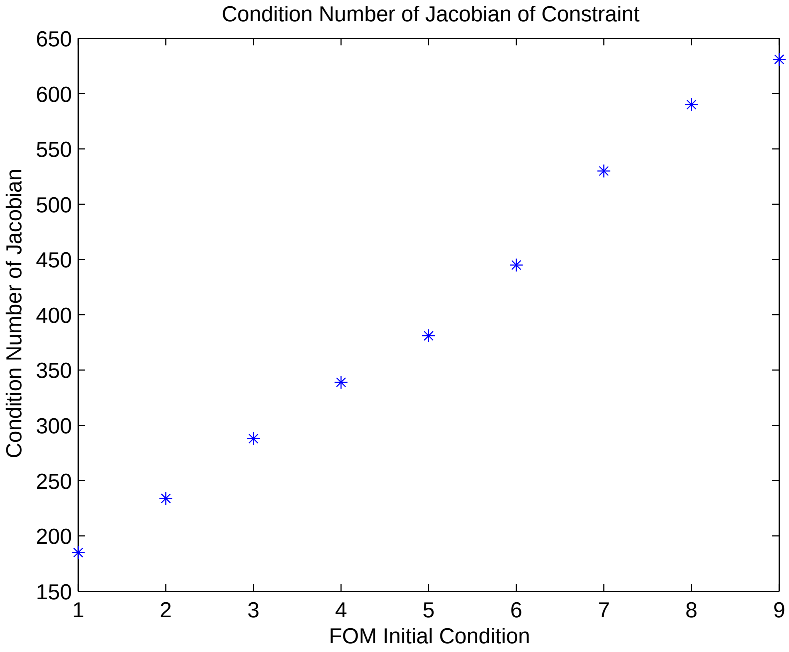


Figure 8.6: Condition number of $\frac{\partial G}{\partial a_H}$ at the 9 snaps at $t = 0.02, 0.03, .., 0.1$. On the x-axis, the FOM initial condition 1 corresponds to the snap at $t = 0.02$, FOM initial condition number 2 corresponds to the snap at $t = 0.03$, and so on.

The errors e_i for each component y_i of the dependent variable in the Matlab ODE solvers are made to satisfy [94]

$$e_i \leq Relerror \cdot |y_i| + Abserror_i \tag{8.23}$$

where $Relerror$ is termed as the relative error for all components and $Abserror_i$ is the absolute error tolerance for each component. For relative error tolerances that vary in the range $10^{-9} < Relerror < 10^{-3}$ and for absolute tolerances that vary in the range $10^{-7} < Abserror_i < 10^{-3}$, we found that the ROM DAE does not proceed beyond $\Delta t = 10^{-3}$, i.e., until $t = 0.021$.

The increase in the condition number is caused by the nature of the IEF physics. The concentration of the hydrogen ions decreases in the IEF simulation because the hydrogen ion profile across the channel becomes more step-ladder like as shown in Fig.8.2 with sharper steps as time progresses because of the sharper concentration of the ampholytes. The weaker POD modes for hydrogen ions contribute lesser as the simulation progresses causing the condition number of $\frac{\partial G}{\partial a_H}$ to increase.

Hence, we have opposing interests at work - we need the weaker modes to retain the accuracy and satisfy the constraint, but their retention soon causes the condition number of $\frac{\partial G}{\partial a_H}$ to worsen and prevents the ROM-DAE from evolving beyond a very small time step. We need to find a balance between retaining the weaker modes, until their effect on the ROM simulation error is negligible, and discarding them after a small time step so that they do not cause the ill-conditioning of the constraint matrix $\frac{\partial G}{\partial a_H}$.

One of the ways in which we attempted to overcome this ill-conditioning was by weighting the hydrogen modes. The idea was that one could weight the weaker hydrogen modes in such a way that after a small time interval, their contribution to the ampholyte concentration would be less than a small constant. For example, we experimented with exponentially decreasing (with time) weights attached to the weaker modes, which we could remove from the original list of ROM equations, once their contribution was lower than a small preset value. This would prevent the ill-conditioning of $\frac{\partial G}{\partial a_H}$ without loss of too much accuracy.

However, this weighting needs to be systematic and logically extendible to the case of many more ampholytes. The constraint modes which drop off from the list of ROM equations as a result of the weighting are those that primarily contribute to the edges of the steps in the pH gradient. For a fixed value of N (the total number of ampholytes), we could device an ad-hoc way of weighting the constraint modes so that a balance between simulation error and ill-conditioning could be made. If the weaker constraint modes are dropped off from the list of ROM equations at an earlier than recommended time step, the ROM simulations will no longer be accurate and the errors will propagate into further time steps. For the general case of varying N, we could not devise a systematic way to attach weights to the weaker constraint modes so that the balance between accuracy and ill-conditioning is optimal. Hence, we could not extend this approach to the general case of an arbitrary number of ampholytes.

A different approach for reduced order modeling of the IEF equations, that could be tried is the 'equation-free' approach as described by Kevrekidis et. al [45]. This approach relies on using a reduced order model in conjunction with the full order model, in order to compute the dynamics that evolves at the different time-scales in a multi-scale problem. In the 'equation-free' approach, the faster hydrogen ion dynamics would be computed using the original PDE, while the slower ampholyte evolution would be computed using POD or any other appropriate reduction technique. The computational treatment of different time scales with separate numerical techniques has led to a reduction in overall computation time for various multi-scale problems like protein-folding and catalytic surface reactions as noted in [45] and can arguably be applied to the IEF problem as well. However, this approach is a subject for future research and has not been tested in our work.

While considering the impact of stiffness, especially in DAEs, on the traditional POD algorithm, we observed that a certain kind of dynamics would not be reducible through the traditional POD process. In the next chapter, we present this shortcoming and a novel augmentation of the POD algorithm that overcomes this shortcoming.

Chapter 9

A Shortcoming in Applying POD to DAEs, and a Computationally Cheap Resolution

9.1 Introduction to the shortcoming

POD is arguably the most important systematic technique that is known for model reduction of problems with nonlinear dynamics. Modifications to POD are an area of ongoing research because the traditional methods of applying POD have been found wanting in special classes of dynamics [79]. Here we present a shortcoming to applying *traditional* POD to an important class of dynamics - problems with high stiffness, in particular, DAEs. In this chapter, we denote *traditional* POD by the reduction procedure described in the previous chapter (Algorithm 3)- where the modes were chosen solely on the basis of energy of the snapshots of the solution of the original (full order) problem.

One example where a need for a reduced order model for a stiff problem might occur is in designing a controller for a chemical reaction. The chemical reactions are frequently modeled with algebraic equations for the equilibrium chemistry. Hence, the dynamics of the chemical plant will be described by DAEs. Now, when a controller is needed to say, regulate the input of a certain catalyst, one would need to design a 'computational model' of the entire plant, which will be used in conjunction with chemical sensors to regulate the catalyst's input. To be able to do this in real time, this 'model' should be computationally cheap (as well as accurate) - thus necessitating the need for a reduced order model of the chemical plant.

We began considering this class of problems while thinking about model reduction of the IEF problem, whose governing equations are stiff because of the presence of the reactive chemistry of the ampholytes and hydrogen ions, which we model as a constraint. The over-arching problem is stiffness: reduced order modeling of stiff systems is a difficult problem (for example, in problems regarding combustion [58] and chemical engineering [93]). However, the shortcoming in POD that we show in this chapter- which we will term as *twist* from hereon - does not appear explicitly in the IEF problem. With the help of examples, we show how *twist* can happen in some stiff problems and describe how we can augment the traditional POD procedure and fix this issue. However, we emphasize that the solution to the *twist* issue *does not solve the IEF model reduction problem* (which is an area of future research).

The term *twist* refers to something that occurs in the phase-plot geometry and will be motivated and defined more precisely with numerical examples in the later sections, but we provide a brief explanation here. When performing POD, one expects the ROM subspace to be large enough so that the the ROM trajectory can shadow the FOM trajectory in the ROM space. However, as in all FOM/ROM examples, the FOM trajectory will have components that evolve in subspaces that are not included in the ROM subspace. The assumption is that these neglected subspaces are not important for the shadowing of the FOM trajectory in the ROM space. However, as we show in some stiff problems, when the evolution of the FOM trajectories happens in subspaces that are *entirely outside* the ROM space, the ROM space can be rendered incapable of providing a large enough ‘playground’ for the ROM trajectory to shadow the FOM trajectory. We term such an evolution of the FOM trajectory *entirely outside* the ROM space as *twist*.

This can also be described with the following analogue. Suppose one watches a movie and is then assigned the task of describing the movie’s plot to some one who hasn’t watched it. Any twist in the movie’s plot must be explained more clearly than the rest of the movie, and the failure to do so will leave the second person unable to understand the rest of the description of the movie. The twist in the movie’s plot is similar to the twist in the phase-plot of the dynamics, the narration of the plot is analogous to the shadowing of the FOM trajectory (the whole movie) in the ROM subspace, and the failure to understand the rest of the movie is analogous to the failure of the ROM trajectory to shadow the FOM trajectory till the final time step of the trajectory’s evolution (end of the movie). The key is to augment the ROM space with all the subspaces in which *twist* can happen so that the ROM space is large enough to enable the ROM trajectory to shadow the FOM trajectory. Such an augmentation is necessary, because, as we will show, the traditional POD algorithm is incapable of finding the ROM space where *twist* occurs.

With the help of examples, we show how the presence of stiffness makes it *more likely* that traditional POD cannot be applied. We also show how twist can be addressed by a computationally cheap augmentation of the traditional POD program.

9.2 Examples that show 'twist'

Traditional POD chooses the ROM subspace based solely on the energy criterion. In this section, we will show examples of dynamics where traditional POD does not work due to the occurrence of twist in FOMs which have constraints. In the next section, we will show how one can augment POD to successfully resolve this issue. All the equations in these examples will be in cylindrical coordinates.

9.2.1 Cylinder example with constraint

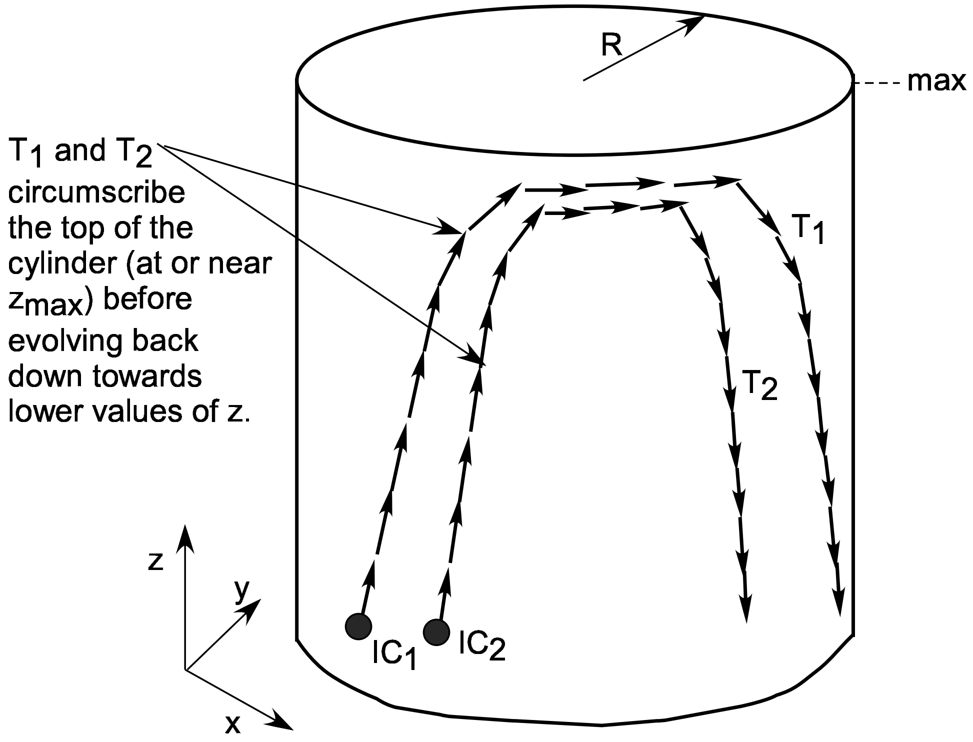
Eqn. 9.1 is an example of dynamics in which twist is observed. The differential equation in Eqn. 9.1 evolves on the surface of the cylinder with radius R as shown in Fig. 9.1. The trajectories first evolve along increasing values of z, and then circumscribe the cylinder at or very near the top of the cylinder ($z = max$), before evolving back down towards decreasing values of z.

The equations (in cylindrical coordinates) describing the evolution of the trajectory on the surface of the cylinder shown in Fig. 9.1 are:

$$\begin{aligned} \dot{\theta} &= exp(\alpha(z - max)) \\ \dot{z} &= (z - max)(\theta - \Theta) \\ 0 &= r^2(cos^2(\theta) + \frac{sin^2(\theta)}{1-\epsilon}) - R^2. \end{aligned} \tag{9.1}$$

The parameters for the above equation were set as given in Table 9.1

The slenderness of the cylinder is shown in the side-by-side view of the cylinder and its magnified view in Fig. 9.2. In order to show the FOM and ROM trajectories



R
max
T1 and T2 circumscribe the top of the cylinder (at or near zmax) before evolving back down towards lower values of z.
T1
T2
z
y
x
IC1
IC2

Figure 9.1: The cylinder is a two-dimensional manifold on which the differential algebraic equations given in Eqn. 9.1 evolve for the given time interval. This illustration (not to scale) shows the parameters that describe the cylinder and also shows two trajectories T_1 and T_2, that evolve according to Eqn. 9.1 with initial conditions IC_1 and IC_2 respectively. The arrows along T_1 and T_2 represent the direction of the vector field. The trajectories first evolve along increasing values of z, and then circumscribe the cylinder at or very near the top of the cylinder (max), before evolving back down towards decreasing values of z.

Table 9.1: Parameter Values for Eqn. 9.1

α	max	R	ϵ	Θ
10	10	10^{-3}	0	$\frac{\pi}{2}$

that evolve on the surface of the cylinder, we will show the magnified view of the cylinder from hereon, with the scales labeled on the axes. We will continue showing the magnified view in the next example in subsection 9.2.2 where the cross section of the cylinder is an ellipse.

The evolution of the trajectory starting at the initial condition $[0, 0, 10^{-3}]$ is shown in Fig. 9.3 and Fig. 9.4 on the time interval $[0, 10]$. One can see that the trajectory peaks at the top of the cylinder, then evolves along the top surface of the cylinder (the trajectory circumscribes the top of the cylinder along increasing values of θ) before tracing a path along the wall of the cylinder towards decreasing values of z.

For computing the POD basis for this FOM, we used 51 snapshots that were equally spaced in the time interval [0, 10] with the retained energy parameter λ_{perc} set at $\lambda_{perc} = 95\%$. The mode energies for the first 3 POD modes are given in Fig. 9.5.

From Fig. 9.5 and with the value of λ_{perc} set at $\lambda_{perc} = 95\%$, the POD basis (that was computed with the traditional POD algorithm) consists of only the following vector: $[-.1711, -.9853, -10^{-4}]$, which in Cartesian coordinates is

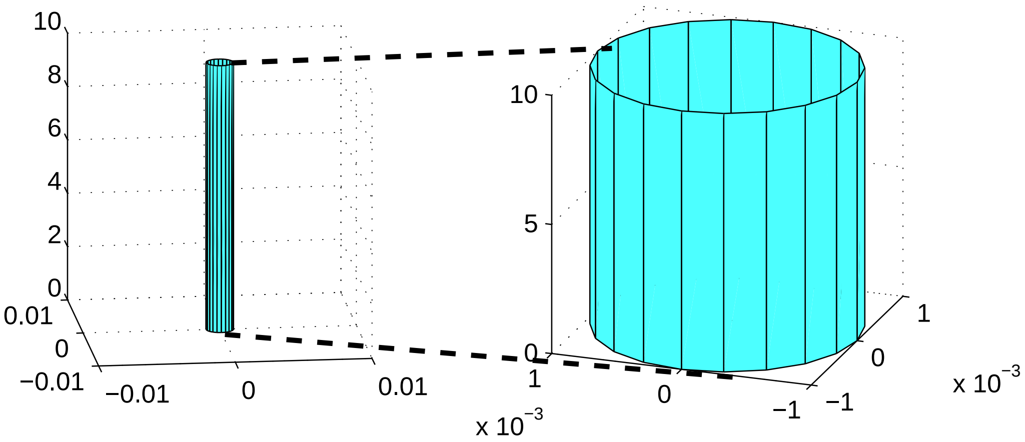


Figure 9.2: The cylinder described in Table 9.1 is slender with the slenderness ratio $\frac{max}{R} = 10^4$, where max is the height of the cylinder and R is its radius of cross-section. The magnified view of the cylinder is on the right and it is this visual scale that we will use to show the trajectories in the rest of this chapter.

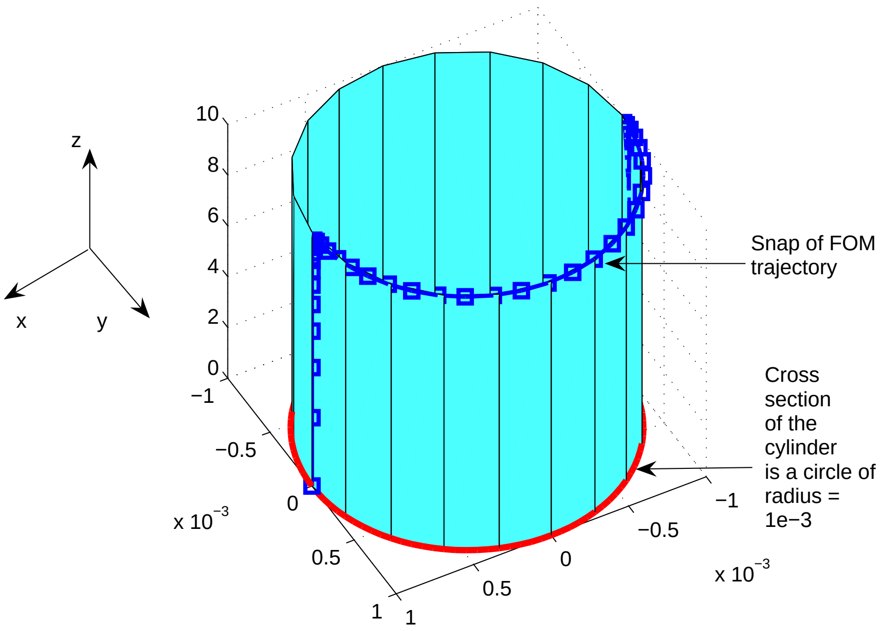


Figure 9.3: FOM trajectory evolving on the surface of a cylinder (of circular cross section) according to Eqn. 9.1 with parameter values mentioned in Table 9.1. Each snapshot of the trajectory is marked by a square. In this view, we see the trajectory rising up to the top of the cylinder and evolving along the top of the cylinder.

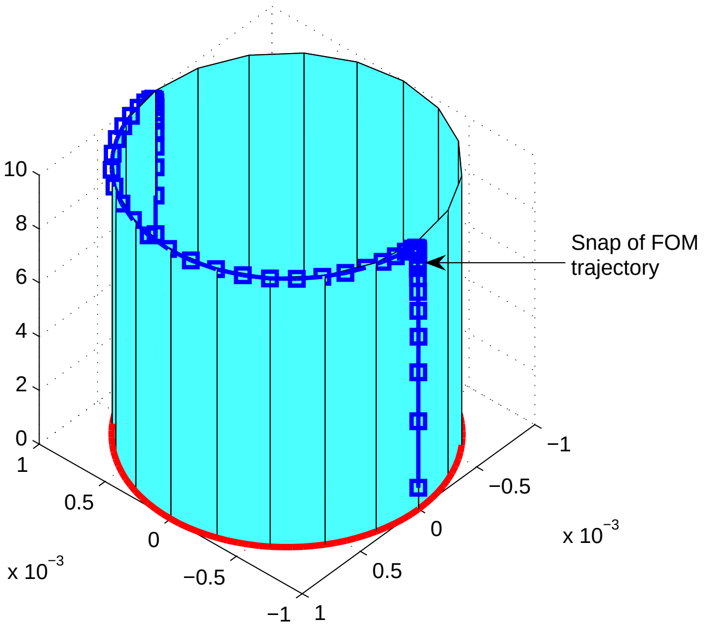


Figure 9.4: FOM trajectory evolving on the surface of a cylinder (of circular cross section) according to Eqn. 9.1 with parameter values mentioned in Table 9.1. In this view, we see the FOM trajectory evolving back down to lower values of z.

$[-9.8 \times 10^{-5}, 1.7 \times 10^{-5} - .99]$, which is a vector that is very slightly tilted away from the z-axis. This is understandable because the radius of the cylinder is very small and the FOM trajectory's energy is almost completely determined by it's projection on the z-axis. We note that inspite of the 'respectable' value of λ_{perc}, the ROM basis consists of only a single vector. In fact, if we had allowed the trajectory to evolve further (this would be along the negative z-axis) and included snapshots further along that trajectory, then even if we had chosen a much higher value of λ_{perc}, we would still only pick up the single vector in the POD basis. None of the dynamics in the $x - y$ plane would be picked up by this basis.

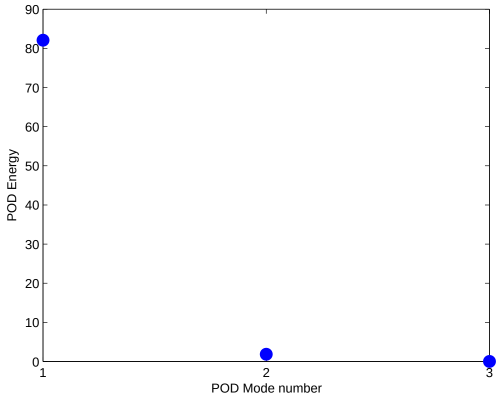


Figure 9.5: The energies of the first 3 POD modes are shown. The first POD mode has over 97 % of the total POD mode energy.

The FOM state vector can be projected onto the POD basis ϕ, to get the following ROM representation.

$$a(t) = c(t)\phi \tag{9.2}$$

where $a(t) = [\theta(t), z(t), r(t)]$ is the FOM state vector, ϕ is the POD basis which in this case is just a single vector given by $\phi = [\phi_1, \phi_2, \phi_3]$, and $c(t)$ is a scalar which describes the ROM's evolution along the POD basis. In traditional POD, to formulate the equation that describes the evolution of the ROM, we substitute Eqn. 9.2 in Eqn. 9.1, and pre-multiply both sides of the resulting equation by ϕ^T to get

$$M_{ROM}c(t) = f_{ROM}(c(t)) \tag{9.3}$$

where $M_{ROM} = \phi^T M_{FOM}\phi$, $M_{FOM} = diag([1, 1, 0])$, and $f_{ROM}(c(t)) = \phi^T f$ where $f(t) = [f_1(t), f_2(t), f_3(t)]^T$ is given by

$$\begin{aligned} f_1(t) &= exp(\alpha(c(t)\phi_2 - max)) \\ f_2(t) &= (c(t)\phi_2 - max)(c(t)\phi_1 - \Theta) \\ f_3(t) &= (c(t)\phi_3)^2(cos^2(c(t)\phi_1) + \frac{sin^2(c(t)\phi_1)}{1-\epsilon}) - R^2. \end{aligned} \tag{9.4}$$

The ROM given by Eqns. 9.2, 9.3, and 9.4 only evolves until the top of the cylinder. This corresponds to the top most point of the FOM's trajectory as shown in Figs. 9.6 and 9.7.

One can see that at the top most point of the trajectory, we will have $f_{ROM} = \phi^T f = 0$, i.e., the field f as defined in Eqn. 9.4 is orthogonal to (or *twists* out of) the POD-subspace ϕ at the top most point of the trajectory. This is the point at which *twist* occurs, and since the ROM vector field vanishes at this point, the ROM trajectory is unable to shadow the FOM trajectory beyond this point. So, after it finishes evolving along the top most part of its trajectory (the top of the cylinder), the FOM trajectory evolves back down along the wall of the cylinder, but the ROM trajectory can not shadow the trajectory back down towards lower values of z.

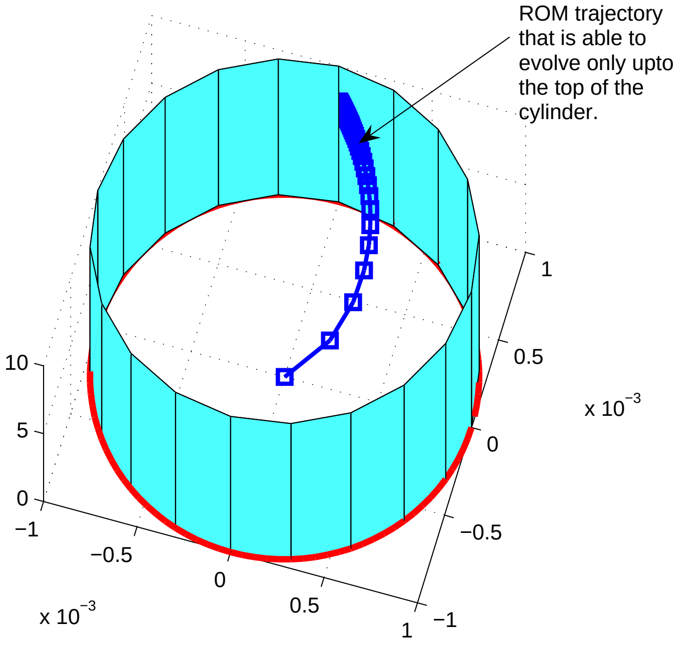


Figure 9.6: ROM trajectory is only able to evolve uptil the top of the cylinder (see Fig. 9.7 for another view of this figure). The squares show the snapshots of the ROM trajectory, which is mapped back onto the FOM space (in this case back onto $\mathbb{R}^3$). Successive snapshots of the ROM trajectory in this figure are uniformly separated in time. We can see that the ROM vector field vanishes (equals zero) at the top of the cylinder (since the successive squares of the ROM trajectory get closer to each other in space, although they are equally separated in time). This prevents the ROM trajectory from shadowing the FOM trajectory (shown in Fig. 9.4) back down to lower values of z.

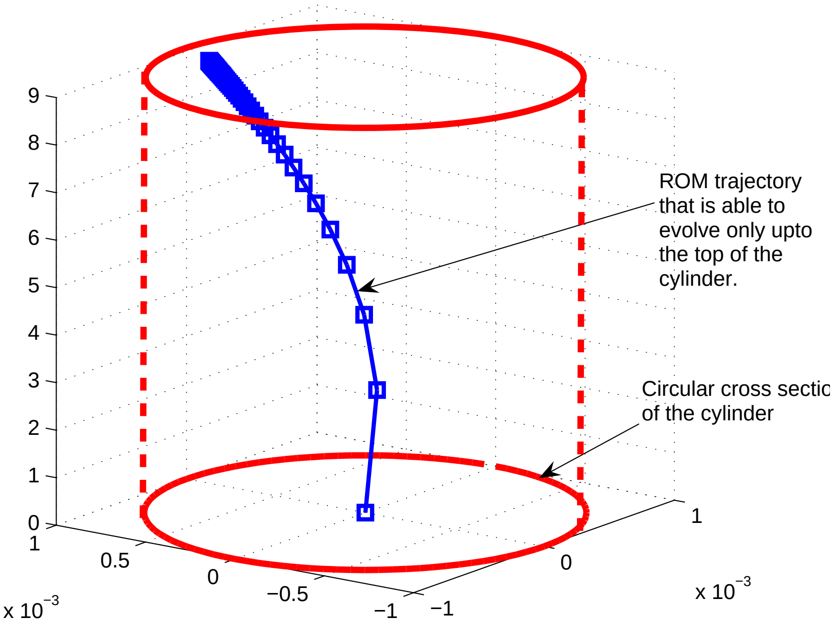


Figure 9.7: Side views of the ROM trajectory that is only able to evolve uptil the top of the cylinder. We can see that the ROM vector field vanishes (equals zero) at the top of the cylinder (since the successive squares of the ROM trajectory get closer to each other in space, although they are equally separated in time). The circular cross section and the outline of the cylinder is shown here.

With the above motivating example, we can give the following definition of *twist* :

Definition *Twist* is said to occur when the ROM vector field vanishes ($f_{ROM} = \phi^T f = 0$) even if the FOM field does not, i.e., the field f as defined in Eqn. 9.4 is orthogonal to the POD-subspace ϕ.

9.2.2 Elliptical example with constraint

In this example, we include multiple trajectories (corresponding to different initial conditions) and we see that in spite of including many trajectories and including the snaps along all of them, twist still persists. We retain the same value of $\lambda_{perc} = 95\%$, but we choose the constraint manifold as the cylinder with an elliptical cross section. As we proceed in this example, it will become clear that depending on the initial conditions, a higher eccentricity in the ellipse can force the occurence of twist for even higher values of λ_{perc} (with a higher value of λ_{perc} one should ideally expect a more accurate ROM). In the next section, we will show how one can augment POD to avoid the occurence of twist.

Apart from a higher eccentricity of the FOM equations are the same as in Eqn. 9.1, with the parameters set as in Table 9.2.

Table 9.2: Parameter Values for Eqn. 9.1

α	max	R	ϵ	Θ
$-.03$	10	10^{-3}	0.9	$\frac{\pi}{2}$

We chose to include 51 equally spaced (in time) snapshots for each of the 4 trajectories, corresponding to the 4 initial conditions as given in Table 9.3.

Table 9.3: Four Initial conditions for FOM with constraint manifold as cylinder with an elliptical cross section

	IC-1	IC-2	IC-3	IC-4
θ	.5341	.5655	.5969	.6283
z	0	0	0	0
r	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}

Two views of the FOM plot corresponding to the initial condition $[.6283, 0, 5 \times 10^{-4}]$ are shown in Figs. 9.8 and 9.9. The POD basis, in polar coordinates, is the vector $[-.14, .99, 0]$, which in Cartesian coordinates is just the z-axis $[0, 0, 1]$.

Following the same general procedure as in Eqns. 9.2, 9.3 and, 9.4, we can construct the ROM model for this example by projecting the FOM Eqn. 9.1 (with the parameters as given in Table 9.2) onto the POD basis $\phi = [-.14, .99, 0]$. The ROM evolves till $t = t_{final}$, but it stalls at the particular point on the ROM subspace where the projected vector field vanishes because once again, we see that at this point, $f_{ROM} = \phi^T f = 0$, i.e., the field f as defined in Eqn. 9.4 (with parameters set for this example as in Table 9.2) is orthogonal to (or *twists* out of) the POD-subspace ϕ at the top most point of the trajectory. Hence, the ROM trajectory shown in Fig. 9.10 can only shadow the FOM trajectory along the z-axis and hence, cannot shadow it beyond the top most point (the largest z-axis value) of the FOM

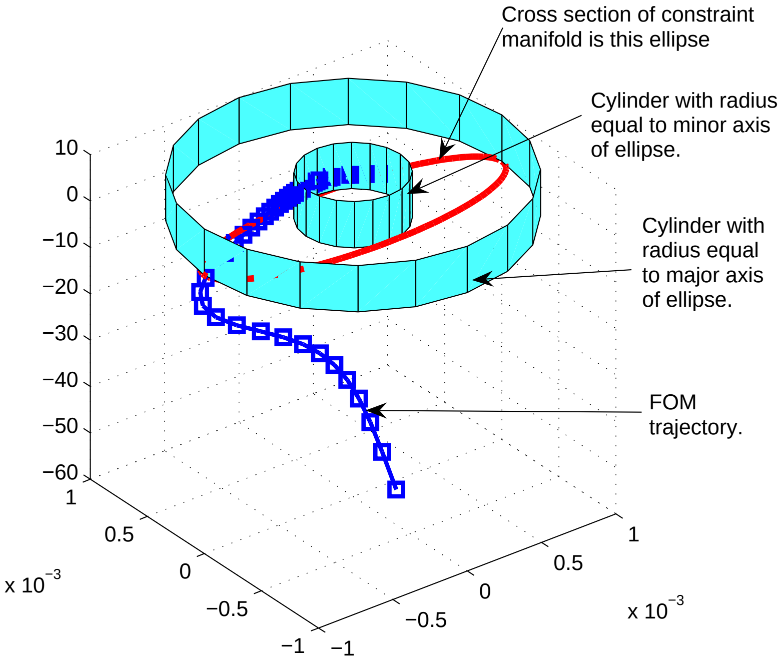


Figure 9.8: First view of the FOM trajectory evolving on the surface of a cylinder, with an elliptical cross section. The eccentricity of the ellipse is 0.9. The initial condition of the trajectory in polar coordinates is $[.6283, 0, 5 \times 10^{-4}]$.

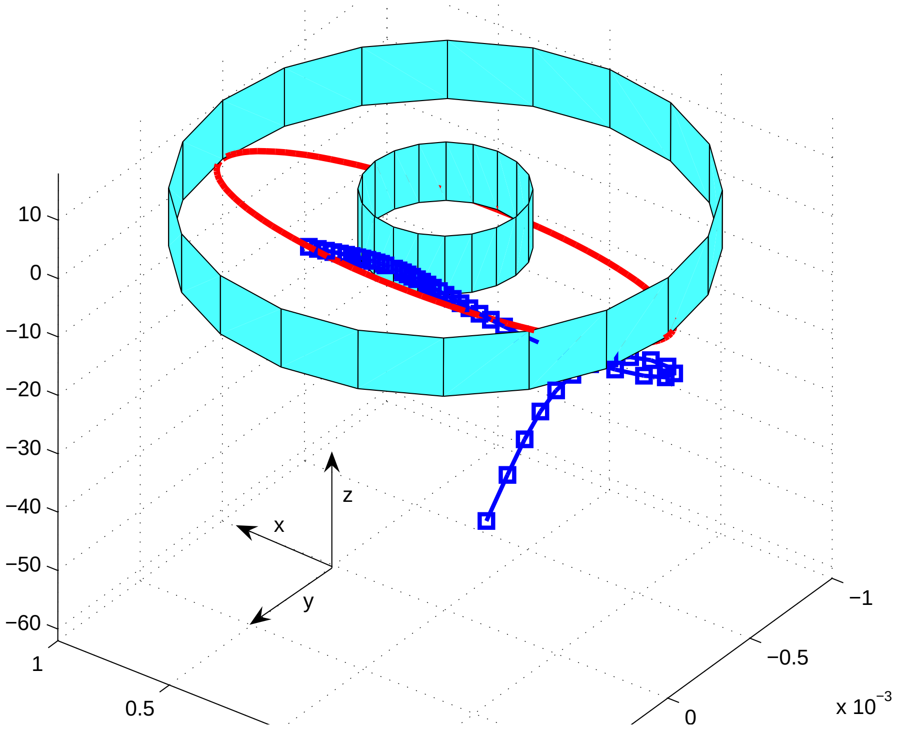


Figure 9.9: Second view of the FOM trajectory of Fig. 9.8.

trajectory.

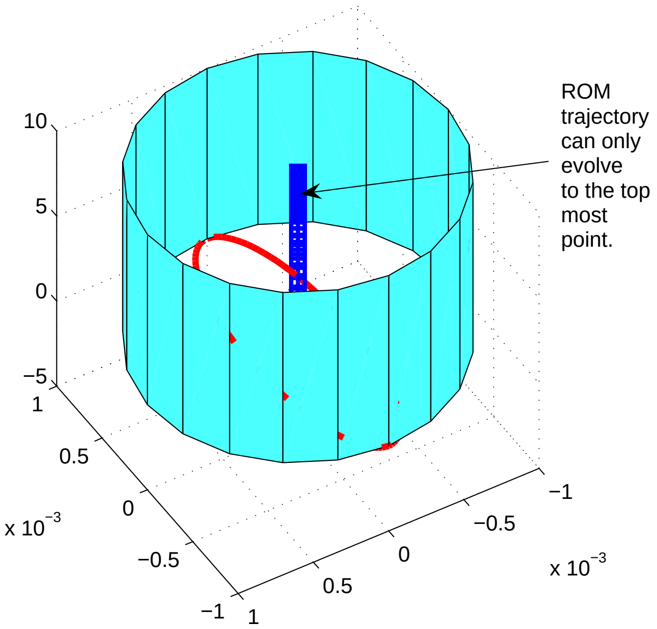


Figure 9.10: The ROM trajectory for the cylindrical constraint with elliptical cross section is only able to evolve uptil the top of the cylinder where the ROM vector field vanishes.

It is partly a matter of chance that using the traditional POD program with a given value of λ_{perc} is able to create a 'big enough ROM playground', i.e., an ROM subspace in which the FOM trajectories can be shadowed. The chances of producing a large enough playground are bettered if the problem has dynamics with a low stiffness. This is because the FOM trajectories would then be able to 'spread out' further, because there is no constraint binding it to any surface, and hence the POD energy in the snaps will show a wider spread across the dimensions in the FOM space. If this spread is not large enough, then as we have shown, it is possible that some of the dimensions into which the FOM trajectory evolves, will not contain enough energy (as defined in the POD sense) to be included in the reduced POD basis and hence cause twist. We note that this can occur (as in our examples) even if the FOM trajectory spends a long enough time evolving solely along the neglected POD dimensions (not having the FOM trajectory evolve for a sufficient amount of time in the neglected, but crucial, POD dimensions was recognized early on [99] as a deficiency in POD based reduction). All the snapshots in the FOM trajectories in both examples are equally spaced in time, and we see that a majority of them are present in the top most part of the cylinder (near $z = max$), which means that the FOM trajectories spend a longer time at or near the top of the trajectories than along the side walls of the cylinder. We know of no general result for arbitrarily high dimensional dynamics that shows how to choose a 'high enough' value of λ_{perc} to ensure that *twist* never occurs. Moreover, a 'high enough' value of λ_{perc}, if carelessly set, will lead to an unnecessarily large ROM space, which would lead to very little reduction in computation time.

9.3 Augmenting the traditional POD basis

Twist occurs because the ROM vector field needs to evolve in a subspace that the traditional POD procedure has neglected. There is no known general way to apriori know this subspace. For a typical higher dimensional FOM, where for example, 5 ROM modes out of a total of 1000 FOM modes have been chosen as part of the traditional POD basis, based purely on the retained energy λ_{perc}, one cannot know which subset of the neglected 995 modes are necessary to avoid twist. As far as we know, the spread of energy among the POD modes is problem specific. If we did apriori know the set of those state vector snaps near which twist occurred, then we could assign an appropriately higher weight to those snaps before the traditional POD procedure or we could set λ_{perc} to an appropriately high value. But as it stands, the one characteristic that has been observed for problems that can be successfully modeled with POD or any other ROM procedure, is that the modes' energies fall roughly exponentially [37] or even more strongly in stiff problems, as shown in Fig. 9.11.

Since twist occurs when the vector field is ignored in stiff problems, the solution to avoiding twist would lie in augmenting the traditional POD basis with the dominant part of the POD basis of the vector field. However, if we visualize the vector field in Fig. 9.12, we can also expect a large component of the vector field's POD energy λ_{perc} to lie along the traditional POD basis. This is because the 2-norms of the vector field snaps are dominated by the part of the trajectory that crawls along the walls of the cylinder and we will yet again be ignoring the part where the twist occurs - the top portion of the cylinder, where the 2-norm of the vector field is small because of the slow dynamics. In the subspaces of the FOM space were the norm of the vector field is large, *it is more likely* that the snapshots of the trajectories themselves will also have a large norm and contribute to the POD energy of the Traditional POD algorithm 3. Hence, it is *more likely* that the ROM space created by the Traditional POD algorithm with a lower value of λ_{perc} will end up including the subspace in which the vector field norm is large. One should focus

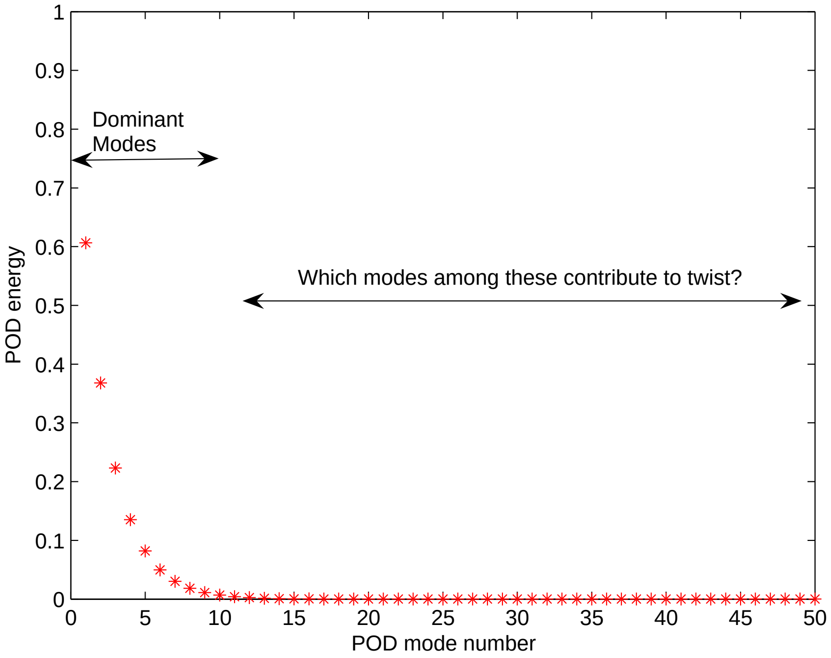


Figure 9.11: This illustration depicts the observation that the POD modes' energy tend to drop off exponentially in many physical problems. The dominant modes are included in the reduced order model. In general, it is not possible to apriori know which of the neglected modes need to be included in the ROM subspace in order to eliminate *twist*. If a large value of λ_{perc} (which determines the number of retained modes in the ROM space) is set arbitrarily, will lead to an unnecessarily large ROM space, which would lead to very little reduction in computation time

instead on the region of the FOM space were the norm of the vector field is smaller.

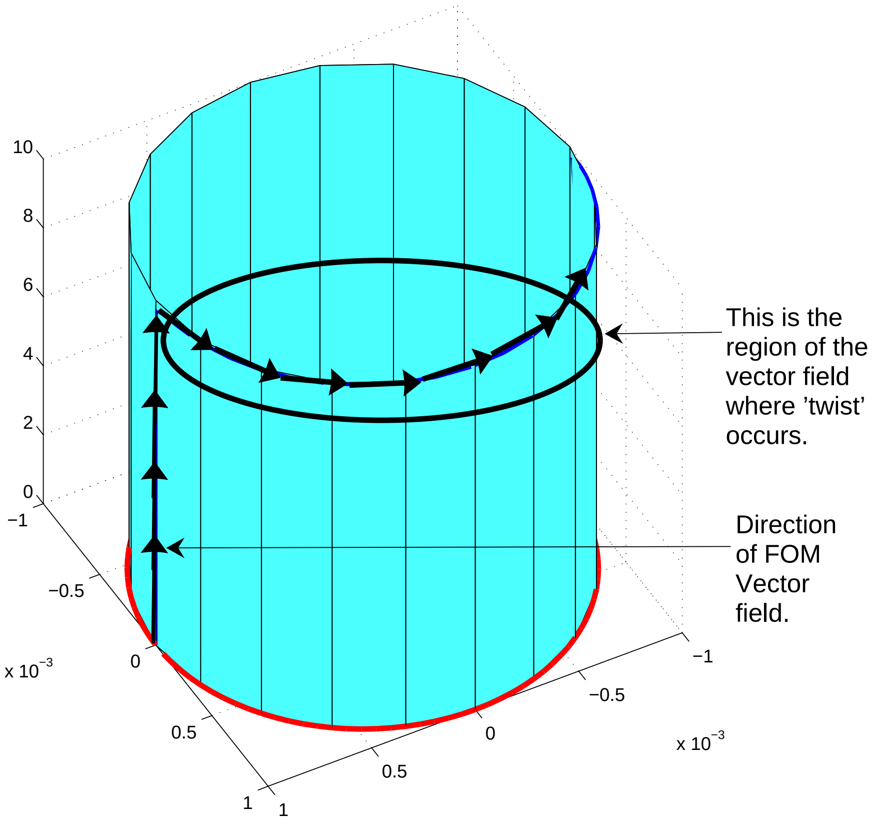


Figure 9.12: The portion of the FOM subspace due to which *twist* occurs is the $X-Y$ plane. The energy of the trajectory in the $X-Y$ plane has a negligible contribution to the retained energy λ_{perc} in the traditional POD procedure. However, the vector field in this portion of the subspace twists out of the POD space computed by the traditional POD algorithm.

We should concentrate on computing the POD basis of only the *small-norm* portion of the vector field, which is the subspace that is more likely to be excluded from the ROM space and is hence the region where the FOM trajectory is more likely to twist out of the ROM space. By linearity, the POD basis of the *small-norm* portion of the vector field can also be expected to have a subset of its basis vectors very close to the traditional POD basis, but it will also have the crucial neglected basis vectors because of which the twist occurred. We consider only the lowest $\eta\%$ (by 2-norm) of the vector field snaps for further augmenting the POD basis. The value of the η is to be set by the user, but it should be low enough so that it mainly concentrates on the slower portion of the dynamics - where the twist is more likely to occur. For example, one can choose η by saying that the L_1 norm of the chosen fraction of the vector field snaps should be lower than some preset fraction of the L_1 norm of the entire set of vector field snaps, where the fraction is set by some characteristic number of the underlying physics. For example, in convection-diffusion problems which can have very different time scales, there will could be a convection dominated part of the dynamics and a diffusion dominated portion. This competition between convection and diffusion is captured by the global Peclet number Pe. In such a case, one can set η by requiring that we should consider all those vector field snaps whose 2-norms are less than $1/Pe$ of the entire set of snaps.

The portion of the POD basis of the *small-norm* portion of the vector field which are 'close' to the traditional POD basis can be eliminated by a cheap merging step. The ideas of this section results in the following algorithm that seeks to eliminate twist by augmenting the traditional POD basis with the POD basis of the *small-norm* portion of the vector field.

In the next section, we will show how even using fairly conservative values for λ_{perc}^{State}, λ_{perc}^{Field}, η, and λ_{perc}^{Merge} we can augment the POD basis of the example shown in subsection 9.2.2 and achieve a successful reduced order model.

Algorithm 4 Augmented Basis algorithm to eliminate twist

```
1. Perform traditional POD with the FOM state vector snaps and retain those modes which together contain $\lambda_{perc}^{State}$ of the energy. Denote these modes as $Basis_1$.

2. Estimate the FOM vector field snaps from the finite differences of the FOM state vector snaps in the following way: If $(U_i^j : i = 1,..,N)$ are the $N$ snapshots of the FOM state vectors for a trajectory corresponding to the $j^{th}$ initial condition, then the $N-1$ vector field snaps $(V_i^j : i = 1,..,N-1)$ for the $j^{th}$ initial condition are given by $V_i^j = U_{i+1}^j - U_i^j$.

3. Sort the vector field snaps in ascending order of their 2-norms.

4. Perform POD on the lowest (by 2-norm) $\eta$ % of the vector field snaps and retain those modes which together contain $\lambda_{perc}^{Field}$. Denote these modes as $Basis_2$.

5. Merge $Basis_1$ and $Basis_2$, by eliminating those basis vectors in $Basis_2$ that are 'close' to $Basis_1$ by a very cheap POD step that uses the vectors in $Basis_1$ and $Basis_2$ as 'snapshots' retains those modes which together contain $\lambda_{perc}^{Merge}$ of the union of $Basis_1$ and $Basis_2$.
```

9.4 Implementing the Augmented Basis algorithm on a numerical example

In subsection 9.2.2, for the problem of creating a reduced order model for the case of the full order model having a cylindrical constraint with elliptical cross section, we showed how the traditional POD basis failed to shadow the 3-D FOM trajectory in the 1-D ROM space (Fig. 9.10).

We now apply the Augmented Basis algorithm of the previous section to the above example. As mentioned before, with $\lambda_{perc} = 95\%$ we get the POD basis $Basis_1 = [-.14, 99, 0]$. For each of the 4 initial conditions stated in Table 9.3, we compute the 50 vector field snaps from the differences of the 51 state vector snaps that are aready available to us and stack them in the ascending order of their 2-

norms. In Fig. 9.13 we show the vector field snaps for one of the initial conditions, which we separate into two groups - the vector field snaps that are circled are those those which have lower 2-norms on average than the ones that are not circled.

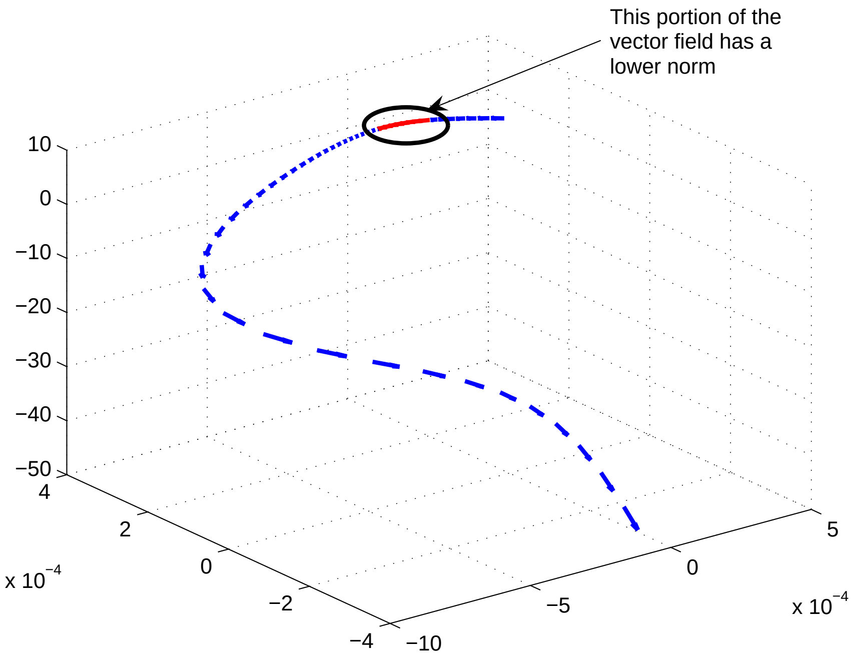


Figure 9.13: The vector field snaps that are circled are roughly where *twist* occurs. This set of vector field snaps have a lower 2-norm on average than the rest of the vector field snaps (that are not circled). It is only the vector field snaps that are circled that have to be considered for augmenting the POD basis to avoid *twist*. This set of snaps should be separated out of the rest of the vector field snaps with an appropriate choice of the parameter η.

In this example, we set $\eta = 10\%$. We found that a POD procedure on this low-norm fraction of the vector field snaps yielded the 2-D POD basis $Basis_2 = [v_1; v_2]$, with $v_1 = [-.998, -.069, 0]$ and $v_2 = [-.069, -.998, -10^{-4}]$, even with a λ_{perc} as low as 81%. It is clear that we get this 2-D basis (and hence an eventually successful

ROM) because we are focusing on the crucial parts of the vector field where the twist happened. Now we see that $Basis_1$ (the POD basis got from traditional POD applied to the trajectory snapshots) is very close to $Basis_2$ (the POD basis of the low-norm vector field) and hence we should do an additional, but very cheap POD step, in order to eliminate the redundancy. The final 2-D augmented basis $AugBasis = [w_1; w_2]$ with $w_1 = [-.1361, -.99, 0]$ and $w_2 = [.99, -.1361, 0]$ captures the energy of the 3 'snapshots' of $Basis_1$ and $Basis_2$ for any value of λ_{perc}^{Merge} in the range $67\% < \lambda_{perc}^{Merge} < 99\%$. This additional 'merging' step should be expected because the fraction η is heuristically set (with a nod to the underlying physics) and will hence tend to pick a part of the traditional POD basis ($Basis_1$) or basis-vectors that are very close to it as a part of vector field POD basis ($Basis_2$). The resulting 2-D ROM equations are given by the previously stated set of equations Eqns. 9.2, and 9.3, except that we now have the previously 1-dimensional basis ϕ replaced with the 2-dimensional augmented basis $AugBasis = [w_1; w_2]$ and the scalar $c(t)$ in Eqns. 9.2, and 9.3 is now a 2-vector with $c(t) = [c_1(t), c_2(t)]$. The results of the successful ROM with the sugmented basis are as follows.

Both the vectors of the augmented basis in the cylindrical coordinates have non-zero $\theta - z$ components, but have a zero radial (r) component. However, the main source of the any error between the FOM and any ROM for this example lies in the (in)ability of the ROM to 'shadow' the evolution of the FOM trajectory along the z-axis. We see that the augmented ROM is able to shadow the z-axis evolution of the FOM trajectory very well, all the way till the final time step of the FOM trajectory, as shown in Fig. 9.15.

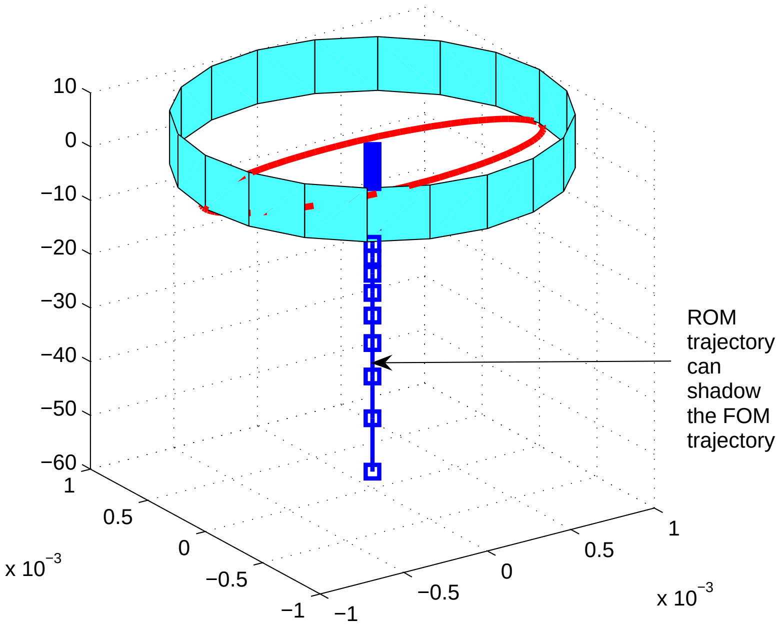


Figure 9.14: Successful shadowing of the FOM trajectory (in 3 dimensions) in the augmented ROM basis (in 2 dimensions). The ROM trajectory can shadow the FOM trajectory to the top of the cylinder *and* back down (to the negative values of z.)

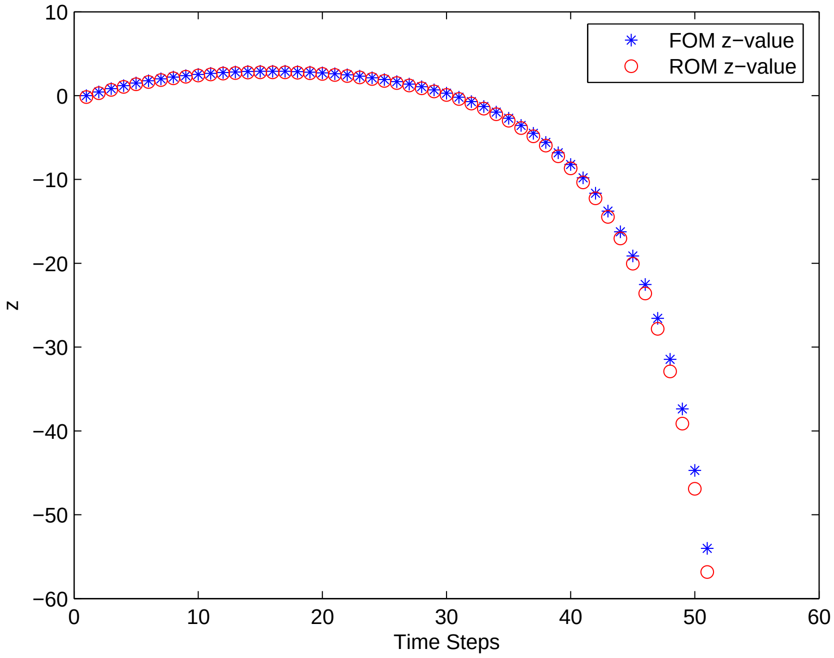


Figure 9.15: The error between the z-axis values of the FOM and augmented ROM is low.

A comparison of the times taken for the FOM and ROM procedures is given in Table 9.4.

Table 9.4: Simulation time for the Augmented POD algorithm

FOM time for 1 initial condition	$1.9s$
ROM time for 1 initial condition	$0.71s$
Traditional POD basis computation time	$6.17 \times 10^{-2}s$
Augmented POD basis computation time	$6.39 \times 10^{-2}s$

9.5 Discussion

The phenomenon of *twist* can be expected to occur in highly stiff problems, as shown in the simple examples in this chapter. The geometry of the constraint manifold, together with the dynamics of the FOM trajectory can force the traditional POD algorithm (which does not focus on the vector field) to pick an insufficiently large ROM space that does not allow the ROM trajectory to successfully shadow the FOM trajectory for the entire time interval of the trajectory's evolution. In chemical engineering and biochemistry, the presence of reactive equations which could be modeled algebraically is one example of such dynamics. In control theory, when such stiff dynamics need to be cheaply modeled as a 'plant' in the control loop, one may need to account for *twist.* We show how one can account for twist by augmenting the traditional POD algorithm. One advantage of the augmented

(AugBasis) algorithm is that no exact knowledge of time scale variation or the shape of the constraint manifold is needed, which would have been critical for setting λ_{perc} in the traditional POD procedure. Another advantage is that the extra computation is cheap and no additional data is required apart from what is already available for the traditional POD technique (since the vector field is estimated from the state vector snaps).

Bibliography

[1] B. Alberts, D. Bray, J. Lewis, M.Raff, K. Roberts & J.D. Watson, "Molecular Biology of the Cell," Garland Publishing Inc., New York & London, 1994.

[2] N.G. Anderson & N.L. Anderson, "Analytical Biochemistry," 1978, 85, p.331.

[3] B.D.O. Anderson, "Controller reduction- concepts and approaches," IEEE Transactions in Automatic Control, vo.l34, Aug 1989, pp.802-812.

[4] A.C. Antoulas, "Application of Large-Scale Dynamical Systems," SIAM Press, 2005, Chapter 9.

[5] T. Apostol, "Calculus Vol 1: One-Variable Calculus with an Introduction to Linear Algebra," John Wiley and Sons, 1967, Second Edition, Chapter 3.

[6] W.E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenproblem," Quart. Appl. Math., (9), pp.17-29, 1951.

[7] J.A. Atwell, B.B. King, "Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations," Math. Comput. Modeling, (33), pp.1-19, 2001.

[8] N. Aubrey, P. Holmes, J.L. Luley, E. Stone, "The dynamics of a coherent structure in the wall region of a turbulent boundary layer," J. Fluid. Mech., 192, pp.115-173, 1988.

[9] W. Batty, C.E. Christofferson, S. David, A.J. Panks, R.G. Johnson, C.M. Snowden and M.B. Steer, "Steady state and transient electro-thermal simulation of power devices and circuits based on fully physical thermal model," in Proc. 6th THERMINIC Workshop, Sept.24-27, 2000, pp.125-130.

[10] S.C. Beeler, G.M. Kepler, H.T. Tran, H.T. Banks, "Reduced order modeling and control of thin film growth in an HPCVD reactor," Technical report CRSC-00-33, Center for Research in Scientific Computation, North Carolina State University, 1999.

[11] J. P. Black and R. M. White, "Microfluidic applications of ultrasonic flexural plate waves," Proc. Transducers 99 Conf., pp. 11341136.

[12] D.S. Boyalakuntla and J.Y. Murthy, "COBRA-Based compact models for simulation of electronic chip packages," Proc. Interpack., July 2001, IPACK2001-15 534.

[13] K.E. Brenan, S.L. Campbell, & L.R. Petzold, “Numerical solution of initial-value problems in differential-algebraic equations,” North-Holland, 1989, Chapter 2.

[14] H-T Chang, Y-F Huang, S-H Chiou, T-C Chiu, M-M Hsieh, “Advanced Capillary and Microchip Electrophoresic Techniques for proteomics,” Current Proteomics, 2004, vol.1, p.325-347.

[15] E. Chiprout, “Model reduction for large circuit simulation”, Institute for Mathematics and its Applications (IMA), Univ. of Minnesota, Sept. 1997.

[16] J.V. Clark, D. Bindell, W.Kao, E. Zhu, A. Kuo, N. Zhou, J. Nie, J. Demmel. Z. Bai, S. Govindjee, K.S.J. Pister, M. Gu, A. Agogino, “Addressing the needs of complex MEMS design,” Proc. MEMS 2002, Las Vegas, 2002.

[17] M. Clemens, E. Gjonaj, P.Pinder and T. Weiland, “Self-consistent simulations of transient heating effects in electrical devices using the Finite Integration Technique,” IEEE Trans. Magnetics, pt.1, vol.37, pp.3375-3379, Sept 2001.

[18] L. Codecasa, D. D’Amore and P. Maffezzoni, “An Arnoldi based thermal network reduction method for electro-thermal analysis,” IEEE Transactions on Components and Packaging Technologies, Vol.26, No.1, March 2003, pp.186 - 192.

[19] H. Cui, K. Horiuchi, P. Dutta, and C. F. Ivory “Multistage Isoelectric Focusing in a Polymeric Microfluidic Chip,” Analytical Chemistry, 77 (24), 2005. pp. 7878 -7886.

[20] G.E. Dullerud and F. Paganini, “A course in Robust Control Theory,” Springer 2000.

[21] L.C.Evans, “Partial Differential Equations,” American Mathematical Society, Providence, 1998.

[22] B.F. Farrell, P.J. Ioannou, “State estimation using a reduced order Kalman filter,” Journal Atmospheric Sciences, (56), pp.3666-3680, 2001.

[23] P. Feldmann, R.W. Freund, “Efficient linear circuit analysis by Pade approcimation via the Lanczos process,” IEEE. Trans. CAD, Integrates Circuits and Systems, 14, pp.639-649, 1995.

[24] FEMLAB Reference Manual 2002.

[25] K.A. Ferguson, “Starch gel electrophoresis application to the classification of pituitary proteins and polypeptides,” Metabolism, 13, 1964, pp. 9851002.

[26] R.W. Freund, “Passive Reduced Order Modeling via Krylov Subspace Methods,” Proc. 2000 International IEEE Symposium on Computer Aided Control System Design, Anchorage, Alaska, Sept. 2000, pp.261-266.

[27] Y.C. Gerstenmaier and G. Wachutka, “Time dependent temperature fields calculated using eigenfunctions and eigenvalues of the heat conduction equation,” in Proc. 6th THERMINIC Workshop, Sept.24-27, 2000, pp.55-61.

[28] J.C.Giddings, in I.M. Kolthoff & P.J. Elving, Eds. “Treatise on Analytical Chemistry, Part 1,” Wiley, New York, 1982, Chapter 3.

[29] J.C.Giddings, “Unified Separation Science,” John Wiley & Sons, 1991, Chapter 1.

[30] A. H. Gordon, B. Keil, K. Sebesta, “Electrophoresis of Proteins in Agar Jelly,” Nature 164, pp.498-499, 17 September 1949.

[31] P. Grabar, C.A. Williams, “A method permitting the combined study of the electrophoretic and immunochemical properties of a mixture of proteins: application to blood serum (Article in French),” Biochim. Biophys. Acta. 10., 1953, pp.193-194.

[32] E.J. Grimme, “Krylov projection methods for model reduction,” Ph.D. Thesis, ECE Dept., U. of Illinois, Urbana-Champaign, (1997).

[33] P.K. Gunupudi, M.S. Nakhla, “Model-reduction of nonlinear circuits using Krylov-space techniques,” Proc. 36th ACM/IEEE Conf. on Design automation, 1999, pp.13-16.

[34] H. Haglund, in D.Glick (Ed.) “Methods of Biochemical Analysis,” Wiley Interscience, New York, Vol.19, pp. 1-104.

[35] A.W. Heemink, M. Verlaan, A.J. Segers, “Variance reduced ensemble Kalman filter,” Report 00-03, Department of Applied Mathematical Analysis, Delft University of Technology, The Netherlands, 2000.

[36] A.E. Herr, J.I. Molho, K.A. Drouvalakis, J.C. Mikkelsen, P.J. Utz, J.G. Santiago, T.W. Kenny, “On-chip Coupling of Isoelectric Focusing and Free Solution Electrophoresis fo Multidimensional Separations,” Analytical Chemistry, 75, 2003, pp.1180-1187.

[37] P. Holmes, J.L. Lumley, & G. Berkooz, “Turbulence, coherent structures, dynamical systems and symmetry,” Cambridge University Press, 1996, Chapter 3.

[38] C.G. Honnegar, “Thin-film ionophoresis and thin-film ionophoresis-chromatography,” Chim. Acta, 44, 1961, pp.173.

[39] J.T. Hsu and L. Vu-Quoc, “A rational formulation of thermal circuit models for electro-thermal simulation - Part 1: Finite Element Method,” IEEE Trans. Circuits Syst., vol.43, pp.721-732, pp.721-732, Sept. 1996.

[40] F.P. Incropera and D.P. Dewitt, “Introduction to Heat Transfer,” Wiley, 4th Edition, 2001.

[41] I.M. Jaimoukha, E. M. Kasenally, “Implicitly Restarted Krylov Subspace Methods for Stable Partial Realizations,” SIAM Journal Matrix Anal. Appl., Vol.18, July 1997, pp.633-652.

[42] L. Jiang, J. Mikkelsen, J. M. Koo, D. Huber, S. Yao, L. Zhang, P. Zhou, J. G. Maveety, R. Prasher, J. G. Santiago, T. W. Kenny, and K. E. Goodson, “Closed-loop electroosmotic microchannel cooling system for VLSI circuits,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 25, no. 3, Sept. 2002, pp. 347355.

[43] G.M. Kepler, H.T. Tran, H.T. Banks, “Reduced order model compensator control of species transport in a CVD reactor,” Technical report CRSC-99-15, Center for Research in Scientific Computation, North Carolina State University, 1999.

[44] G.M. Kepler, H.T. Tran, H.T. Banks, “Compensator control for chemica vapor deposition film growth using reduced order design models,” Technical report CRSC-99-41, Center for Research in Scientific Computation, North Carolina State University, 1999.

[45] I.G. Kevrekidis, C.W. Gear, G. Hummer, “Equation-Free: The Computer-aided analysis of complex multiscale systems,” AIChE Journal, Vol. 50, 7, 2004, pp.1346-1355.

[46] S.W. Kim, B.D.O. Anderson, A.G. Madievski, “Error bound for transfer function order reduction using frequency weighted balanced truncation,” Systems Control Lett., (24), pp.183-192, 2005.

[47] , D. von Klobusitzky, P. Konig, “Biochemical study of snake venom from the genus Bothrops,” Arch. Exptl. Pathol. Pharmakol. Naunyn-Schmiedeberg 192, 1939, pp.271

[48] W. Kreuger and A. Bar-Cohen, “Thermal characterization of a PLCC-expanded Rjc methodology,” IEEE Trans. Comp.,Hybrids Manufact. Technol.,vol.15, pp.691-698, May 1992.

[49] S. Krishnan, S.V. Garimella, G.M. Chrysler, R.M. Mahajan, “Towards a thermal Moore’s law,” IEEE Transactions on Advanced Packaging, Vol. 30, 3, 2007, pp.462-474.

[50] D.W.Larson and R. Viskanta, “Transient combined free air convection and radiation in a rectangular enclosure,” J.Fluid Mech., Vol.78, 1976, pp.65-85.

[51] C.J.M. Lasance, “ The conceivable accuracy of experimental and numerical thermal analysis of electronic systems,” IEEE Transactions on Components and Packaging Technologies, Vol.25, No.3, Sep 2002, pp.366-382.

[52] S.S. Lee and D.J. Allstot, “Electrothermal simulation of Integrated Circuits,” IEEE J. Solid-State Circuits, Vol.28, No.12, Dec. 1993, pp. 1283-1293.

[53] R. de Levie, “The HendersonHasselbalch Equation: Its History and Limitations,” J. Chem. Educ., 2003, 80: 146.

[54] P. Li, F. Liu, X. Li, L. T. Pileggi, S R. Nassif “Modeling Interconnect Variability Using Efficient Parametric Model Order Reduction,” Proceedings of the conference on Design, Automation and Test in Europe, 2, 2005, pp. 958-963.

[55] Y.C.Liang, W.Z. Lin, H.P. Lee, S.P. Kim, K.H.Lee and H.Sun, “Proper orthogonal decomposition and its applications - Part 2: Model Reduction for MEMS Dynamical Systems,” Journal of Sound and Vibration (2002), 256(3), pp. 515-532.

[56] M. Loeve, “Probability Theory II,” No. 46 in Graduate Texts in Mathematics. Springer-Verlag, fourth Edn, 1978.

[57] L.G. Longsworth in M.Bier (Ed.), “Electrophoresis, Theory, Methodology and Applications,” Vol. 1, Academic Press, New York, 1959.

[58] T. Lu, C.K. Law, “A directed relation graph method for mechanism reduction,” Proc. Comb. Inst., 30, pp.1333-1341, 2005.

[59] J.L. Lumley, “Stochastic tools in turbulence,” Academic Press, 1970.

[60] K. Macounova, C.R. Cabrera, M.R. Holl & P. Yager, “Generation of natural pH gradients in microfluidic channels for use in isoelectric focusing,” Analytical Chemistry, 2000, vol.72, p.3745-3751.

[61] Q. Mao,, J. Pawliszyn, W. Thormann, “Dynamics of capillary isoelectric focusing in the absence of fluid flow: High resolution computer simulation and experimental validation with whole column optical imaging,” Analytical Chemistry, 2000, vol.72, p.5493-5502.

[62] P.Mathai, B.Shapiro, D.DeVoe, S.Sivanesan, “ Modeling and Simulation of Ampholyte Behavior in 2-dimensional Isoelectric Focusing,” 2005 ASME International Mechanical Engineering Congress on Exposition, Orlando, Florida, Nov 2005.

[63] P.Mathai, B.Shapiro, “Interconnection of Subsystem reduced-order models in the electrothermal analysis of large systems,” IEEE Transactions on Components and Packaging Technologies, [see also IEEE Transactions on Components, Packaging and Manufacturing Technology, Part A: Packaging Technologies], 30(2), pp.317-329, June 2007.

[64] N.P. van der Meijs, “Model reduction for VLSI physical verification,” Technical Report, Department of ITS/EE, Delft University of Technology, The Netherlands, 2000.

[65] I. Mezic and S. Narayan, “Overview of some theoretical and experimental results on modeling and control of shear flows,” Proc. Of 39th IEEE Conference on Decision and Control, Sydney, Australia, Dec.2000, pp.1709-1715.

[66] R.J. Moffat and A. Ortega, “Direct air cooling of electronic components,” Advances in Thermal modeling of thermal Components and Systems, Vol.1, A.Bar-Cohen and A.D.Krauss, Eds, Hemisphere New York, pp.129-282.

[67] B.C. Moore, “Principal component analysis in linear systems: controllability, observability and model reduction”, IEEE Transactions in Automatic Control, AC-26, pp.17-32, 1981.

[68] G. Moore, “Cramming more components onto integrated circuits,” Electronics, Vol.38, 8, 19 April 1965.

[69] R. Mosher, D. Dewey, W. Thormann, D.A. Saville, & M.Brier, “Computer simulation and experimental validation of the electrophoretic behavior of proteins,” Analytical Chemistry, 1989, vol. 61, p.362-366.

[70] R. Mosher, W. Thormann, “Experimental and Theoretical dynamics of isoelectric focusing: IV: Cathodic, Anodic, and symmetrical drifts of the pH gradient,” Electrophoresis, 1990, vol.11, p.717-723.

[71] R. Mosher & W. Thormann, “High resolution Computer simulation of the dynamics of isoelectric focusing using carrier ampholytes,” Electrophoresis, 2002, vol. 23, p.1803-1814.

[72] K. Ogata, “Modern Control Engineering,” Prentice Hall India, 3rd Edition, 1998.

[73] O.A.Palusinski, A. Graham, R.A. Mosher & M. Brier, “Theory of Electrophoretic Separations”: Part 2: Construction of a Numerical Simulation & its applications,” AIChe Journal, Feb. 1986, Vol. 32, No.2, p.215-223.

[74] E. Papanicolau and S. Gopalakrishna, “Natural convection in shallow, horizontal air layers encountered in electronic cooling,” ASME Journal of Electronic Packaging, Vol.117, 1993, pp. 307-316.

[75] G.P. Peterson and A. at particulaOrtega, “Thermal control of electronic equipment and devices,” Advances in Heat Transfer, Vol.20, J.P. Hartnett and T.F. Irvine JR., Eds, Academic Press San Diego, 1990, pp.181-314.

[76] L.T. Pillage, R.A. Rohrer, “Asymptotic waveform evaluation for timing analysis,” IEEE. Trans. CAD, (9), pp.352-366, 1990.

[77] J. Porath, B. Gelotte, P. Flodin, “A Method for concentrating Solutes of High Molecular Weight,” Nature, 188, 1960, pp.493-494.

[78] R.F. Probstein, “Physicochemical Hydrodynamics,” Wiley Interscience, New York, 2nd Ed, 1994.

[79] M. Rathinam, L.R. Petzold, “A new look at Proper Orthogonal Decomposition,” SIAM Journal of Numerical Analysis, Vol.41, No.5, pp.1893-1925, 2003.

[80] F.F. Reuss, “Charged Induced Flow (Original article in Russian),” Memoires de la Societe Imperiale des Naturalistes de Moskou 2, pp.327-344, 1809.

[81] P.G. Righetti, “Isoelectric focusing: Theory, Methodology and Applications,” Elsevier Biomedical, 1983, Chapter 1.

[82] P.G. Righetti, “Electrophoresis: The march of pennies, the march of dimes,” Journal of Chromatography A, 1079, 2005, pp.24-40.

[83] P.G. Righetti, C. Simo, R. Sebastiano, A. Citterio, "Carrier ampholytes for IEF, on their fortieth anniversary (1967-2007), brought to trial: The verdict," Electrophoresis, 28, pp.3799-3810, 2007.

[84] H. Rilbe, "pH and Buffer theory - A New Approach," Wiley Series in Solution Chemistry, 1996, Chapter 1.

[85] Personal Communication with Dr.Scott Rodkey, Department of Pathology, University of Texas-Houston.

[86] M.C. Romanowski, E.H. Dowell, "Using eigenmodes to form an efficient Euler based unsteady aerodynamics analysis," Proceedings of Symposium on Aeroelasticity and Fluid Structure Interaction Problems: American Society of Mechanical Engineers, AD. Vol.44, pp.147-160, 1994.

[87] T.D. Romo, J.D. Clarage, D.C. Sorensen, G.N, Phillips Jr., "Automatic identification of discrete substates in proteins: Singular Value Decomposition analysis of time-averaged crystallographic refinements, " Proteins Structure Function Genetics, (22), pp.311-321, 1995.

[88] C. W. Rowley, T. Colonius, and R. M. Murray, "POD based models of self sustained oscillations in the flow past an open cavity," AIAA, Ed., 2000.

[89] A.E. Ruehli, A. Cangellaris, "Progress in the methodologies for the electrical modeling of interconnects and electronic packages," Proc. IEEE, (89), pp.740-771, 2001.

[90] M.-N.Sabry, et al., "Realistic and Efficient Simulation of Electro-thermal Effects in VLSI Circuits," IEEE Trans. VLSI Systems, Vol. 5, No. 3, 1997, pp. 283-289.

[91] B. Salimbahrami and B. Lohmann,"Krylov Subspace Methods in Linear Model Order Reduction: Introduction and Invariance properties," Technical Report, Institut of Automation, University of Bremen, August 2002.

[92] C. G. J. Schabmueller, M. Koch, A. G. Evans, A. Brunnschweiler, and M. Kraft, "Design and fabrication of a self-aligning gas/liquid micropump," Proc. SPIE, vol. 4177, 2000, pp. 282290.

[93] D.A. Schwer, P. Lu, W.H. Green Jr., "An adaptive chemistry approach to modeling complex kinetics in reacting flows," Combustion and Flame, 133, pp.451-465, 2003.

[94] L.F. Shampine, M.W. Reichelt, "The Matlab ODE Suite". This paper can be accessed at the following Mathworks website: www.mathworks.com/access/helpdesk/help/pdf_ doc/otherdocs/ode_ suite.pdf

[95] B.Shapiro, "Passive control of flutter and forced response in bladed disks via mistuning," PhD Thesis, Caltech 1999.

[96] B. Shapiro, "Frequency of weighted sub-system model truncation to minimize systems level model reduction errors," Technical Report, Department of Aeronautics, University of Maryland, College Park, 2001.

[97] J. Shim, P. Dutta, C.F. Ivory, "Modeling and simulation of IEF in 2-D microgeometries," Electrophoresis, Feb 2007, 28: 4. pp.572-586.

[98] L.M. Silveira, M. Kamon, I. Elfadel, J.White, "A coordinate transformed Arnoldi algorithm for generating guaranteed stable reduced order models of RLC circuits," Computer Methods in Applied Mechanics and Engineering, 169(3-4), pp.377-389, February 1999.

[99] L. Sirovich, "Turbulence and dynamics of coherent structures, parts I - III," Quarterly of Applied Mathematics, vol. 45, pp. 561-590, 1987.

[100] O.Smithies, "Zone electrophoresis in starch gels: group variations in the serum proteins of normal human adults," Biochem. J., 61, 1955, pp.629-641.

[101] M. Spivak, "Calculus on Manifolds," W.A. Benjamin Inc., 1965.

[102] M. Stasna, K.Sleis, "Two-dimensional gel isoelectric focusing," Electrophoresis, Vol. 26(18), pp. 3586-3591, Septr 2005.

[103] A.V. Stoyanov, C. Das, C.K. Fredrickson, Z.H. Fan, "Conductivity properties of carrier ampholytes pH gradients in isoelectric focusing," Electrophoresis, 2005, vol.26, p.473-479.

[104] H. Svansson,"Isoelectric fractionation, analysis, and characterization of ampholytes in natural pH gradients .1. Differential equations of solute concentrations at a steady state and its solution for simple cases," Acta. Chem. Scand. 15, 1961, pp.325-341.

[105] H. Svansson, "Isoelectric Fractionation, Analysis, and Characterization of Ampholytes in Natural pH Gradients. II. Buffering Capacity and Conductance of Isoionic Ampholytes," Acta. Chem. Scand. 16, 1962, pp.456-466.

[106] H. Svansson, “Isoelectric fractionation, analysis, and characterization of ampholytes in natural pH gradients. III. Description of apparatus for electrolysis in columns stabilized by density gradients and direct determination of isoelectric points,” Archives of Biochemistry and Biophysics, Suppl.1, 1962,pp.132-138.

[107] V. Szekely, “Identification of RC networks by deconvolution : Chances and Limits,” IEEE Trans. CAS 1, vol. 45, Mar.1998, pp.244-258.

[108] J.B. Tenenbaum, V. deSilva and J.C. Langford, “A Global Geometric Framework for Nonlinearity Dimensionality Reduction,” Science, Vol.290(5500), 2000, pp. 2319-2323.

[109] A. Tiselius, “A new apparatus for electrophoretic analysis of colloidal mixtures,” Trans. Faraday, Soc. 33, 1937, pp.524-531.

[110] W. Thormann, R.A. Mosher, M. Brier, “Experimental & Theoretical dynamics of iseolectric focusing: Elucidation of a general separation mechanism,” Journal of Chromatography, 1986, vol. 351, p.17-29.

[111] W. Thormann, T.Huang, J. Pawliszyn & R.A. Mosher, “High-resolution computer simulation of the dynamics of isoelectric focusing of proteins,” Electrophoresis, 2004, vol.25, p.324-337.

[112] W. Thormann, & R.A. Mosher, “High-resolution computer simulation of the dynamics of isoelectric focusing using carrier ampholytes: : Focusing with concurrent electrophoretic mobilization is an isotachophoretic process,” Electrophoresis, 2006, vol.27, p.968-983.

[113] L.N. Trefethen, D.Bau III, “Numerical Linear Algebra,” Society of Industrial and Applied Mathematics (SIAM), 1997.

[114] L. Ukeiley, J. Seiner, N. Sinha, S. Arunajatesan, “Low dimensional description of cavity flows,” Bull. Amer. Phys. Soc., 45(9), p.138, 2000.

[115] O. Vesterberg,“Synthesis of carrier ampholytes for isoelectric focusing,” Acta Chem. Scand., 1969, pp.2653-2666.

[116] O. Vesterberg, in W.B. Jakoby (Ed.) “Methods of Enzymology,” Academic Press, New York, Vol.22, 1971, pp.389-412.

[117] C.J.M. Lasance, H. Vinke, and H. Rosten, “Thermal characterization of electronic devices with boundary condition independent compact models,” IEEE Trans. Comp.,Packag., Manufact. Technol., vol.18, no.4, pp.723-731, 1995.

[118] G. Wang, V. Sreeram, W.Q. Liu,“A new frequency-weighted balanced truncation method and an error bound,” IEEE Transactions on Automatic Control, Vol. 44, 9, 1999, pp.1734-1737.

[119] K. Willcox, A. Megretski, “Fourier series for accurate, stable, reduced order models in large-scale linear applications,” SIAM Journal for Scientific Computing, Vol. 26, No. 3, pp. 944-962, 2005.

[120] K.T. Yang, “Natural convection in enclosures,” in Handbook of Single-Phase Convective Heat Transfer, S.Kakac, R.K.Shah, and W.Aung, Eds, New York : John Wiley and Sons, 1988, Chapter 13.

[121] A.A. Zavitsas, “Properties of water solutions of electrolytes and nonelectrolyes,” J. Phys. Chem. B, (105), pp.7805-7815, 2001.

[122] K. Zhou, J.C. Doyle, K. Glover, “Robust and Optimal Control,” Prentice Hall, Upper Saddle, New Jersey, 1995.