

Reduction of process simulation models : a proper orthogonal decomposition approach

Citation for published version (APA):

Astrid, P. (2004). *Reduction of process simulation models : a proper orthogonal decomposition approach*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven.
<https://doi.org/10.6100/IR581728>

DOI:

[10.6100/IR581728](https://doi.org/10.6100/IR581728)

Document status and date:

Published: 01/01/2004

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

***Reduction of Process Simulation Models:
a proper orthogonal decomposition
approach***

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus, prof.dr. R.A. van Santen,
voor een commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen op
dinsdag 30 november 2004 om 16.00 uur

door

Patricia Astrid

geboren te Bandung, Indonesië

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. A.C.P.M. Backx

en

prof.dr.ir. R.G.C. Beerkens

Copromotor:

dr. S. Weiland

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Astrid, Patricia

Reduction of process simulation models:

a proper orthogonal decomposition approach / by

Patricia Astrid. - Eindhoven : Technische Universiteit Eindhoven, 2004.

Proefschrift. -ISBN 90-386-1653-8

NUR 919

Trefw.: orthogonale functies / decompositiemethoden / partiele
differentiaalvergelijkingen / procesregeling.

Subject headings: reduced order system / singular value decomposition /
partial differential equations / process control.

*To my mother Mangiring Gaberia Siburian
and my father Manahan Baha Radja Soadun Tampubolon*

Eerste promotor: prof. dr. ir. A.C.P.M. Backx

Tweede promotor: prof. dr. ir. R.G.C. Beerkens

Copromotor: dr. S. Weiland

Kerncommissie:

prof. ir. O.H. Bosgra

prof. dr. ir. L.P.H. de Goey

Promotiecommissie:

prof. dr. ir. P.P.J. van den Bosch

prof. dr. rer. nat. S. Volkwein

dr. J.M.A. Scherpen

dr. K.E. Willcox

This PhD-study is a part of EET-REGLA project “*Development of a method for the control of glass melt quality in industrial glass furnaces*” and supported with a grant (EET K99002) of the Dutch Programme EET (Economy, Ecology, Technology), a joint initiative of the Ministry of Economic Affairs, Ministry of Education, Culture and Sciences and Ministry of Housing, Spatial Planning and the Environment. The programme is carried out by the EET Programme Office, a partnership of Senter and Novem.

Summary

Physical processes described by partial differential equations (PDE's) are usually simulated by discretizing the spatial and the temporal domain of the variables (temperature, velocity). In this way, numerical approximations of the dynamic behavior of these processes are obtained. As a general rule, the finer the discretization, the more accurate the numerical solution of the PDE's will be. However, a fine discretization leads to a large number of equations which need to be solved simultaneously at every time step. Hence, the model complexity increases with increasing requirements on model accuracy.

The objective of this PhD thesis is to develop generic methods to reduce the complexity of a system of PDE's to not more than 100 equations. In doing so, the reduced model should maintain a maximum level of accuracy, while for simulation purposes it is desired that the computational speed of the reduced order model is at least 50 times faster¹ than real-time. The last requirement is relevant for the synthesis of (real-time) dynamic optimization. In this thesis, mainly models for heat conduction of conductive and convective processes are considered, with a focus on applications in glass furnaces.

A technique based on the orthogonal decomposition of a collection of measurements of physical quantities (such as temperature) in position and time (signals) is used to reduce the complexity of models. Following ideas from Fourier series expansions, signals are represented as series of orthonormal functions. These so-called basis functions approximate the spatial distribution of the signal while the coefficients of the basis functions represent the time-varying dynamics. The basis functions are derived from measured or simulated data and are physically relevant. The reduced order model is obtained by applying a Galerkin projection of the equations of the original model onto the space spanned by a finite set of well selected basis functions.

In this PhD thesis, this technique is applied and implemented to heat conduction models and a Computational Fluid Dynamics model of an industrial glass melt feeder. From all applications described in this PhD thesis, the reduced order model consists of less than 1% of the number of equations of the original model. The examples show that with such a reduction a maximum error of

¹This factor is based on experience for achieving high performance (real-time) dynamic optimization.

1% in the variation of the physical variables (signals) is achievable.

The low complexity of the reduced order model enables the design and synthesis of optimal model-based controllers. Using the reduced order models, various linear controllers have been synthesized to optimally control both linear and nonlinear heat conduction processes. The controller minimizes a quadratic criterion function in deviation between the desired and actual state and the input variables of the system. The application of such a controller to heat conduction processes shows that the tracking of an arbitrary desired temperature profile can be achieved in an optimal manner.

Despite the drastic reduction of the number of equations, the computational gain for the reduction of nonlinear processes is low. To reduce the computation time, a method of 'missing point estimation' (MPE) is proposed in this PhD thesis and combined with the POD reduction technique. The reduced model is then based on a selected set of points in the spatial domain.

On the basis of two selection criteria, an ordering of the relevant points in the spatial domain is proposed. The most relevant points are selected for describing the process dynamics, while the dynamical features of the process in the remaining points are estimated. In combination with the reduction technique of orthogonal basis functions, this leads to a reduction of computation time. For nonlinear heat conduction model of a heated plate, a reduced order model that is 100 times faster than real-time can be achieved.

In particular, these reduction and acceleration techniques (POD and MPE) are applied to simulate a transition of operating point in a glass melt feeder. This transition concerns a color change of glass in a glass melt feeder and leads to a drastic change of many physical quantities in the glass melt. For this transition, an optimal set of basis functions is determined from simulation data. A reduced order model for the temperature distribution in the feeder has been constructed from the nonlinear model by applying the POD and MPE technique. It is shown that a reduced model of order 18 attains a resolution with a maximum error of 1% in the variations of the physical quantities, while achieving a computational speed that is about 8 times faster than real-time. The attained accuracy and acceleration are adequate for the anticipation of the process dynamics and for process monitoring. However, the computational speed is not sufficient for on-line control design. If, apart from temperature, also the dynamics of the velocity and pressure field of the nonlinear model are incorporated in the reduction procedure, then a computational speed of about 30 times faster than real-time is feasible. For control design, the desired computational speed is minimally 50 times faster than real-time. The reduc-

tion techniques presented in this thesis therefore have sufficient potential and perspective to enable model-based control system design in the near future, for example by further improvement of the MPE method, or by improving the speed of convergence in the numerical simulations. Further enhancement of the computational gain can be achieved by exploiting the advantages of parallel computing. Parallel computing is already enabled in the new generation software for glass furnaces but it is not used in this PhD thesis.

Samenvatting

Fysische processen waarvan het gedrag beschreven wordt door partiële differentiaalvergelijkingen (PDV's) worden doorgaans gesimuleerd door zowel het spatiële (plaats) als temporele (tijd) domein van de variabelen (temperatuur, snelheden) te discretiseren. Zodoende worden numerieke benaderingen verkregen van het dynamische gedrag van dergelijke systemen. Als regel geldt dat naarmate de discretisatie fijner is, de benadering van de oplossingen van de PDV's beter zal zijn. Echter, een fijne discretisatie heeft een groot aantal vergelijkingen tot gevolg. Deze dienen simultaan in de tijd opgelost te worden. De complexiteit zal toenemen, naarmate een grotere nauwkeurigheid is vereist.

Het doel van dit proefschrift is om generieke methoden te ontwerpen om de complexiteit van een dergelijk door PDV's beschreven model te reduceren tot hoogstens een 100-tal vergelijkingen met maximaal behoud van nauwkeurigheid en zodanig dat voor simulatiedoeleinden een rekentijd van minimaal 50 keer² sneller dan 'real-time' verkregen wordt. De laatste eis is relevant voor de synthese van (real-time) dynamische optimalisatie. In dit proefschrift worden hiertoe hoofdzakelijk modellen van warmteoverdracht zoals warmtegeleiding, conductieve en convectieve processen beschouwd.

Een techniek gebaseerd op orthogonale decompositie van een verzameling van metingen van fysische grootheden (zoals temperatuur) in tijd en plaats (signalen) wordt gebruikt om de complexiteit van modellen te reduceren. Hierin, worden signalen gerepresenteerd door reeksen van orthonormale functies naar voorbeeld van Fourier reeks expansies. Deze zogenaamde basisfuncties benaderen de spatiële distributie van het signaal terwijl de coëfficiënten het tijdsafhankelijke gedrag representeren. De basisfuncties worden afgeleid uit gemeten of experimentele data en zijn daarmee fysisch relevant. Het gereduceerde model wordt verkregen door het procesgedrag van het oorspronkelijke (complexe) systeem te projecteren op de geselecteerde basisfuncties volgens een Galerkin projectie.

Deze techniek is in dit proefschrift op modellen voor warmtegeleidingsprocessen en een Computational Fluid Dynamics (CFD) model van een industrieel glas smelt kanaal (*glass melt feeder*) toegepast en geïmplementeerd in de CFD

²De factor 50 is een op ervaring gebaseerde grens voor goed presterende 'real-time' dynamische optimalisatie.

software omgeving. Uit alle toepassingen die in dit proefschrift zijn beschreven bestaat het gereduceerde model uit slechts 1% van het aantal vergelijkingen van het oorspronkelijke systeem. Voor de in dit proefschrift beschreven voorbeelden wordt een fout verkregen van maximaal 1% van de variaties in de fysische grootheden (signalen).

De complexiteit van het gereduceerde model is dermate laag dat daardoor het ontwerp van optimale modelgebaseerde regelaars mogelijk is. Met behulp van de verkregen gereduceerde modellen zijn optimale lineaire regelaars ontworpen voor lineaire en niet-lineaire warmtegeleidingsprocessen. Deze regelaar minimaliseert een kwadratische functie in de afwijking tussen gewenste en werkelijke toestand en de ingangsvariabelen van het systeem. In de toepassing van een dergelijke regelaar op warmtegeleidingsprocessen is aangetoond dat een willekeurig gewenst temperatuurprofiel realiseerbaar is.

Ondanks de grote reductie van het aantal vergelijkingen, is voor niet-lineaire modellen de winst in rekentijd echter nog gering. Om de rekentijd te verminderen is in dit proefschrift een methode van ‘schatting voor ontbrekende data’ (missing point estimation of MPE) beschreven en gecombineerd met de reductie techniek van orthogonale decomposities. Het gereduceerde model wordt daarin gebaseerd op een selectie van punten in het spatiële domein.

Op grond van een tweetal selectie criteria is een ordening voorgesteld van relevantie van punten in het spatiele domein. De meest relevante punten worden geselecteerd voor de beschrijving van de procesdynamica, terwijl met behulp van schattingen de dynamica in de overige punten wordt beschreven. In combinatie met de reductietechniek van orthogonale basis functies, leidt dit zowel tot een reductie van het aantal vergelijkingen (99% reductie) als een vermindering van de rekentijd (tot een factor 100 ten opzichte van het originele model in het geval van niet-lineaire warmtegeleiding process van een verwarmde plaat).

In het bijzonder zijn deze technieken toegepast voor het simuleren van de omschakeling van werkpunt in een glas-smelt oven. Deze omschakeling betreft een kleurverandering van glas in een glas smelt kanaal. Een omkleuring leidt in dynamisch opzicht tot een drastische wijziging van een groot aantal fysische grootheden in een glas smelt kanaal. Voor dit niet-lineaire proces zijn optimale basisfuncties bepaald en is een gereduceerd model afgeleid op grond van projecties van basisfuncties en de MPE techniek. Aangetoond is dat voor de beschrijving van de temperatuur in de glasoven met een eenvoudig model van 18 basisfuncties een maximale fout van 1% van de variaties in de fysische grootheden haalbaar is terwijl de rekentijd ongeveer 8 keer sneller is dan real-time. Deze nauwkeurigheid en versnellingsfactor zijn voldoende voor

de anticipatie van het procesgedrag en voor proces monitoring. Echter nog niet voor (on-line) regelaarontwerp. Indien naast temperatuur eveneens de dynamica van het snelheidsveld en de drukprofielen wordt meegenomen in de model reductie procedure, dan is een rekentijd van ongeveer 30 keer sneller dan real-time haalbaar. Voor regelaarontwerp zal de gewenste snelheid ten opzichte van het echte proces minimaal 50 keer sneller moeten zijn dan het echte proces. De reductietechnieken die in dit proefschrift zijn beschreven bieden daarmee voldoende perspectief om model gebaseerde regelaarsynthese in de nabije toekomst mogelijk te maken; bijvoorbeeld door verdere aanpassingen in de MPE procedure of door een snellere convergentie in de numerieke simulaties. Verdere verhoging van simulatiesnelheid van het gereduceerde model ten opzichte van het echte proces kan bereikt worden door toepassing van parallelle computer processoren. Parallelle berekeningen worden nu al uitgevoerd bij de nieuwste generatie programmatuur voor het simuleren van processen in glasovens, maar zijn niet toegepast in dit proefschrift.

Ringkasan

Proses-proses fisika yang digambarkan melalui persamaan differensial parsial (PDE) disimulasikan dengan diskritisasi domain waktu dan domain ruang. Secara umum, semakin tinggi level diskritisasi, semakin akurat solusi PDE tersebut. Di lain pihak, akibat diskritisasi ini, jumlah persamaan yang harus diselesaikan secara simultan pun meningkat drastis.

Tujuan dari disertasi doctoral ini adalah pengembangan metoda generik untuk mengurangi tingkat kompleksitas model numerik yang dihasilkan melalui diskretisasi PDE. Model yang dihasilkan disebut sebagai model tereduksi. Model yang dihasilkan harus mengandung sedikit mungkin persamaan (maksimum dalam orde 100 persamaan) namun tetap mengandung tingkat keakurasian yang tinggi. Dalam disertasi ini, model-model yang digunakan terutama adalah model perpindahan panas.

Dalam disertasi ini, teknik yang digunakan didasarkan pada dekomposisi orthogonal dari sinyal yang bergantung pada tempat dan waktu. Sinyal tersebut direpresentasikan sebagai deret dari sekumpulan fungsi orthonormal. Fungsi-fungsi yang dinamakan fungsi-fungsi basis ini menggambarkan distribusi spasial dari sinyal sementara koefisien dari fungsi-fungsi tersebut menggambarkan dinamika sinyal terhadap waktu. Fungsi-fungsi basis ini diturunkan dari data pengukuran atau data simulasi dan memenuhi kriteria optimal. Model tereduksi didapat melalui proyeksi model asal pada fungsi-fungsi basis melalui proyeksi Galerkin.

Teknik ini diaplikasikan untuk model-model perpindahan panas melalui konduksi dan juga model dari glass feeder. Untuk aplikasi pada glass feeder, teknik ini diimplementasikan pada perangkat lunak untuk Computational Fluid Dynamics. Dari semua aplikasi yang dijabarkan di disertasi ini, model tereduksi hanya mengandung 1% dari jumlah persamaan model asal. Selain itu, model tereduksi juga memiliki tingkat keakurasian yang cukup tinggi, di mana penyimpangan maksimum hanya 1% dari besarnya variabel proses yang diamati.

Kompleksitas dari model tereduksi cukup rendah sehingga rancangan pengendali (*control design*) dimungkinkan. Berdasarkan model tereduksi, pengendali dengan tipe optimal linear quadratic regulator (LQR) dirancang untuk perpindahan panas secara konduksi. Pengendali tersebut meminimalkan sebuah fungsi kuadratik yang didefinisikan sebagai fungsi penyimpangan dari set point

dan penggunaan energi. Dalam disertasi ini ditunjukkan bahwa profil temperatur yang diinginkan dapat dicapai secara optimal.

Meskipun model tereduksi mengandung jumlah persamaan yang jauh lebih sedikit dibandingkan model asal, waktu simulasi untuk model nonlinear tidak menurun secara drastis. Untuk meningkatkan kecepatan simulasi, dalam disertasi ini digunakan teknik "Missing Point Estimation (MPE)". Model tereduksi dibangun berdasarkan seleksi dari beberapa sel diskrit dalam domain ruang.

Sel diskrit yang digunakan dipilih berdasarkan dua kriteria pemilihan. Sel diskrit yang relevan dipilih berdasarkan deskripsi dari dinamika proses, dan sel diskrit yang tidak terpilih diestimasi. Kombinasi dari teknik MPE dengan model reduksi berdasarkan dekomposisi orthogonal menghasilkan model tereduksi yang jauh lebih cepat daripada model tereduksi yang dihasilkan melalui cara yang konvensional (hingga lebih dari 100 kali lebih cepat daripada model asal untuk model nonlinear dari perpindahan panas secara konduksi).

Dalam disertasi ini teknik-teknik ini digunakan juga untuk mensimulasikan *job change* di dalam feeder gelas, yang merupakan bagian dari tungku gelas. *Job change* yang dideskripsikan di disertasi ini adalah perubahan warna pada gelas dalam feeder gelas. Perubahan warna membawa perubahan drastis pada dinamika proses karena beberapa parameter fisik seperti konstanta perpindahan panas juga berubah drastis. Untuk proses ini fungsi-fungsi basis yang optimal ditentukan dan model tereduksi diturunkan melalui teknik dekomposisi orthogonal dan juga MPE.

Hasil simulasi menunjukkan bahwa model dengan 18 persamaan dapat menggambarkan proses perubahan warna secara cukup akurat. Model tereduksi yang dihasilkan sekitar 7-8 kali lebih cepat dari real time. Hal ini dikarenakan dalam disertasi ini variabel proses yang diperhitungkan hanya temperatur. Waktu simulasi ini cukup memadai untuk analisa proses, on-line tuning danantisipasi namun tidak cukup memadai untuk perancangan pengendali secara online. Jika variabel proses yang lain seperti kecepatan fluida dan tekanan juga direduksi, maka waktu simulasi yang dicapai bisa mencapai 30 kali lebih cepat dari real time. Teknik-teknik yang dideskripsikan dalam disertasi ini dapat juga diterapkan pada variabel proses yang lain. Pengurangan waktu simulasi dapat pula dicapai melalui aplikasi perhitungan paralel (*parallel computing*) di mana parameter model dapat dihitung secara paralel oleh beberapa prosesor komputer. Hal ini tidak diaplikasikan selama penelitian S3 ini, namun fitur perhitungan paralel sudah tersedia dalam perangkat lunak yang digunakan untuk simulasi proses.

Contents

| | |
|---|-----------|
| Summary | 1 |
| Samenvatting | 5 |
| Ringkasan | 9 |
| 1 Introduction | 15 |
| 1.1 Application of computational models in process industries . . . | 15 |
| 1.2 Derivation of simpler computation models | 18 |
| 1.3 Problem formulation | 19 |
| 1.4 Conceptual introduction to proper orthogonal decomposition . | 21 |
| 1.5 Objective and contribution of this thesis | 27 |
| 1.6 Thesis outline | 28 |
| 2 Proper Orthogonal Decomposition | 31 |
| 2.1 Computational models | 31 |
| 2.2 The model reduction problem | 35 |
| 2.3 Introduction to orthonormal basis | 38 |
| 2.4 Model reduction by proper orthogonal decomposition | 44 |
| 2.4.1 Time invariant case | 45 |
| 2.4.2 Time variant case | 48 |
| 2.4.3 POD basis problem | 51 |

| | | |
|----------|---|-----------|
| 2.4.4 | POD basis $\{\varphi_i\}$ in a general Hilbert space | 52 |
| 2.4.5 | Reduction by POD basis | 60 |
| 2.5 | Application of POD to wave propagation equation | 62 |
| 2.6 | Galerkin projection | 69 |
| 2.7 | Summary | 75 |
| 3 | Application of POD to Heat Conduction Models | 77 |
| 3.1 | Introduction | 77 |
| 3.2 | Computational Fluid Dynamics models | 79 |
| 3.3 | One dimensional heat conduction model | 85 |
| 3.3.1 | Introduction | 85 |
| 3.3.2 | Discretization of one dimensional heat conduction model | 87 |
| 3.4 | Reduced modeling of one dimensional heat conduction model . | 90 |
| 3.4.1 | Derivation of POD-based reduced model for one dimensional heat conduction problem | 92 |
| 3.4.2 | Heating at the left end | 93 |
| 3.4.3 | Heating in the middle | 96 |
| 3.4.4 | Development and validation of the reduced order model | 99 |
| 3.5 | Two dimensional heat conduction model | 110 |
| 3.6 | Controller design | 115 |
| 3.6.1 | Control objective | 117 |
| 3.6.2 | The LQR Controller | 119 |
| 3.7 | Application of the LQR controller | 121 |
| 3.8 | Summary | 127 |

| | | |
|----------|--|------------|
| 4 | Acceleration of Model Reduction for Nonlinear Models | 129 |
| 4.1 | Introduction | 129 |
| 4.2 | Introduction to missing point estimation | 133 |
| 4.3 | Selection of the important grid points | 136 |
| 4.3.1 | Point selection criterion 1 | 136 |
| 4.3.2 | Point selection criterion 2 | 139 |
| 4.4 | Nonlinear heat conduction model | 141 |
| 4.5 | POD-MPE model | 145 |
| 4.6 | Point selection in two dimensional heat conduction model . . . | 149 |
| 4.7 | Controller design | 153 |
| 4.8 | Summary | 159 |
| | | |
| 5 | Reduced Order Modeling of a Glass Melt Feeder | 161 |
| 5.1 | Introduction to glass furnace operation | 161 |
| 5.2 | The feeder model | 166 |
| 5.3 | Snapshot collection and POD reduction | 172 |
| 5.3.1 | Determination of POD basis $\{\varphi_i\}$ | 174 |
| 5.3.2 | Construction of the reduced feeder model | 175 |
| 5.4 | Validation of the reduced order model | 176 |
| 5.5 | Application of MPE to the glass melt feeder | 180 |
| 5.5.1 | Application of MPE based on selection criterion 1 . . . | 182 |
| 5.5.2 | The Shannon Entropy | 186 |
| 5.5.3 | Optimization of point selection | 189 |

| | | |
|----------|---|------------|
| 5.5.4 | Application of MPE based on selection criterion 2 | 192 |
| 5.5.5 | Comparison between selection criterion-1 and selection criterion-2 | 197 |
| 5.6 | Simulation of the glass colour change in the feeder | 198 |
| 5.6.1 | Collection of colour change simulation data | 199 |
| 5.6.2 | Acceleration of the model simulating colour change in feeder | 205 |
| 5.6.3 | Performance of MPE-based reduced models | 212 |
| 5.7 | Summary and conclusions | 215 |
| 6 | Conclusions and Recommendations | 217 |
| 6.1 | POD as a model reduction method | 217 |
| 6.2 | Acceleration of POD model reduction method | 218 |
| 6.3 | Acceleration of CFD models for glass processing simulations . . | 220 |
| 6.4 | Applicability of the developed reduced order modeling techniques | 221 |
| 6.5 | Recommendations for future research | 221 |
| A | The Method of Snapshot | 227 |
| B | LQR design based on nonzero reference signals | 229 |
| | Nomenclature | 235 |
| | Bibliography | 239 |
| | Acknowledgement | 247 |
| | Curriculum vitae | 251 |

Introduction

| | | | |
|-----|---|-----|--|
| 1.1 | Application of computational models in process industries | 1.4 | Conceptual introduction to proper orthogonal decomposition |
| 1.2 | Derivation of simpler computation models | 1.5 | Objective and contribution of this thesis |
| 1.3 | Problem formulation | 1.6 | Thesis outline |

1.1 Application of computational models in process industries

Mass, momentum, and energy transfer are the main physical phenomena simulated in many computational models. The time-dependent characteristics of these phenomena determine the state of the process variables such as temperature, flow fields, and chemical concentrations. In chemical reactors for example, temperature, flow and concentration are very important process variables which determine the quality of the end-product. The quality of the end product can be enhanced by manipulating the relevant process variables in the chemical reactors.

Typically, mass, momentum and energy transfer are described by a set of partial differential equations, defined over a specified spatial domain (such as three dimensional reactors, distillation columns, tubes) and over a specified time period.

One of the prominent modeling tools for transport phenomena in fluids is the Computational Fluid Dynamics (CFD). In CFD, the transport phenomena are described by discretizing the spatial (position) and temporal domain (time) of the governing equations. The CFD models provide the simulation data of the process variables at every discretized spatial domain and at every time step.

The availability of the simulation models will give better insight in the process in many ways. For example, simulations can be used for prediction of future process conditions, process monitoring, and for understanding of the real process itself. The simulation models can also be used as a tool to improve the existing engineering designs. An example of the application of process simulation models in industry is for instance the application of CFD models in glass furnaces [8],[9], [17],[18]. A sketch of glass furnace is depicted in Figure 1.1.

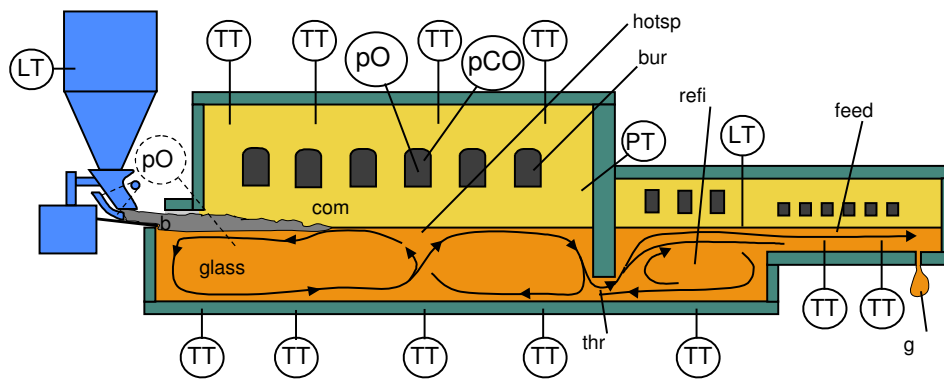


Figure 1.1: A schematic view of glass furnace, the spatial domain for the CFD models. Abbreviations in the figure refer to different parts of the furnace: combustion chamber (com), burners (bur), refiner (refi), feeder (feed), throat (thr). The sensors installed are termocouples (TT) and pressure sensor of gases (pO for oxygen, pCO for carbonmonoxide). The glass product flows through a dosing mechanism at the end of the furnace, the product is called gob (g)

The modeling of heat and mass transfer in the glass furnaces by Computational Fluid Dynamics has advanced since the last two decades of the 20th century. In the glass industries, the CFD models have been used as a tool for [9]:

- improvement of glass furnace design [18]
- optimization of process operation parameters [18]
- understanding of the physics and chemistry of glass melting and combustion [9]
- process monitoring

- control of glass-melting operations [8],[18]

The use of CFD models for improving process design and optimization is also applicable to other industries, such as oil and aviation industries. CFD models allow engineers to improve the existing design, understand the influence of physical parameters on the product quality and eventually to control the processes.

If the CFD models are representative enough for a real process, then the model can be used as a base-model for online control design. The process can be controlled to reach a desired condition by optimally controlling the required amount of energy, costs, and labor needed to manufacture a product.

In model based control, the simulation models are used to extract the process states to the controller modules. Sometimes, the simulation models are referred to as "soft-sensors" since they provide the data of the process states without performing measurements by physical sensors. If the process is operating in volatile conditions, such as extremely high temperature, the information from physical sensors is limited due to the restrictive durability of the sensors. In this situation, it is beneficial to have simulation models which can provide information about the process states which are not measured by the physical sensors.

Additionally, the simulation models may also predict future process conditions provided that the simulation models are faster than real time. In addition, such models can be used as base models for Model Predictive Control (MPC), which has been applied extensively in process industry [8], [31]. The knowledge of the desired condition in the future and the predicted conditions of the process will help the controller to optimize the energy inputs or manipulating variables to reach a desired process condition.

Although various controllers have been successfully implemented in the process industry, the application of model-based control based on CFD models remains very restrictive. Control designs for such processes are usually based upon a simple model derived from the input-output data generated by the CFD models. For control, models faster than real time are required so that the controller can anticipate future conditions. Fast models are not only useful for control system design. Fast models can also be used for online tuning of the process parameters, for better anticipations of the process conditions, and many others. For example, for most applications in Model Predictive Control, the model needs to be at least 50 times faster than real time to implement such a controller.

Despite the fact that CFD models have been proven to be reliable representations of various chemical processes and fluid flows, such models are generally too slow and too large to be useful for online control system designs and real time prediction.

In CFD models, the relevant spatial domain is discretized into into very small grid cells. The dimension of the grid cells is small enough so that the numerical models can approximate the continuous dynamics. In general, the number of grid cells used in CFD models range from $10^3 - 10^8$ grid cells. For every grid cell, the governing mass, momentum, and energy equations are assumed to hold. Hence, for every partial differential equation, a set of discretized equations equal to the number of grid cells has to be solved. The CFD models then typically consist of $10^3 - 10^8$ equations that need to be solved at every time sample. These types of models are called *high order* or *large scale* models with “order” referring to the number of equations to be solved at every time step.

CFD models generally are very slow and numerically intractable. In addition to the large dimension, CFD models also have to take nonlinearities into account which may be present in the models due to the variable-dependent parameters such as temperature-dependent density and temperature dependent viscosity. Consideration of the nonlinearities and the fact that the mass, momentum, and energy balances are coupled, substantially add the complexity of CFD models.

It would be beneficial to have a simpler model (in the sense that it comprises less number of equations) which approximate the original CFD models and provides the estimates of the current and future process variables in a fast and reliable manner. It is also desirable to derive such models through an automated and integrated procedure, so that without the requirement of physical insights, a simpler model can be derived from the high-fidelity and high resolution CFD models. Furthermore, it is also desired that the simpler models still take the original physical relationship into account as well.

The topic of this thesis is how to obtain such simplified models.

1.2 Derivation of simpler computation models

There exist many ways to derive a simple model from a complex one. Derivation of simple models in general means derivation of models which comprise

less number of equations and which are numerically fast for computations or simulations.

The simple models can be derived based on physical insights or based on the study of data collected from simulations or experiments. Hence there are usually two approaches for the derivation of simpler models:

- **Physical-insight based approach**

Using physical insight, an initially complex model can be transformed into a simpler one by considering its physical phenomena. For example, the CFD model of a glass furnace can be replaced by a model which assumes that the glass furnace is a series of continuous-stirred tank chemical reactors (CSTR) [30]. This assumption leads to a simpler model since the original CFD model can be approximated by a smaller number of ordinary differential equations. The disadvantage of this approach is that detailed physical insight is required. If the process is substantially changed, then the physical insights may become obsolete.

- **Black-box modeling approach**

With the advent of system identification techniques, such as subspace identification and neural networks, empirical models can be derived from the input output data. The number of actuators (energy sources, electrodes, or stirrers) and the number of outputs of interest are generally limited (less than 20). System identification techniques can be used to derive a model based on a set of input-output data. The advantage of this approach is that, there is no detailed physical insight required. However, the physical interpretation of the original model is often lost.

1.3 Problem formulation

The main advantage of the physical-insight based approach is that the physical relationship is taken into account in the resulting models. If there are changes in the physical parameters, then they can be related directly to the models.

The main advantage of the black-box modeling is the generic characteristic of the approach. The approach remains applicable as long as input-output data is provided and there is no need to conduct detailed and time-consuming studies for the modeling of such processes.

It would be desirable to combine the advantages of the physical-insight and

the black-box modeling approaches in the derivation of a simpler model. This is the main motivation of this PhD study.

Given a model described by discretized partial differential equations, find a simpler model which meets the following characteristics:

1. the model should be derived from a generic and repeatable procedure such that there is no need to study the physical relationships in detail;
2. the model should take the original physical relationships into account;
3. the model should have significantly lower number of equations, less than 100 equations is the target;
4. the model should retain sufficient accuracy level (less than 10% of the variations of process variables);
5. the model should compute faster than the original model that it is feasible to perform online prediction, online tuning and online control design. A factor 50 faster than real time is the target;
6. the model should have a structure which enables controller synthesis.

In this PhD study, it is desired that the simple models have substantially lower number of equations than the given (original) model. Hence, the problem of finding a simple model becomes the problems of finding a *reduced order model*.

In the system theory literature, reduced order models can be derived by various methods. One of the most popular reduction methods include the method by balancing [45],[62], which was introduced by Moore [45] in 1981 and extended for nonlinear systems by Scherpen [62]. In the balanced model reduction, a given model is reduced by balancing the contribution from the outputs of interest and the other variables (usually referred to as *states* in the control engineering literature). The computation of balanced model reduction is demanding for nonlinear cases [62], and therefore the application domain is restricted to linear and small scale nonlinear models. The optimal Hankel norm approximation used in the balancing method has been introduced by Adamjan, Arov and Krein in [1] and has been developed in a state space setting by Glover in [24]. This method produces explicit bounds on the Hankel and

H_∞ norm of the error system defined as the difference between the original and reduced order system. However, from a computational point of view, the method is numerically intractable for systems whose state dimension exceeds the order of 100.

The method of Proper Orthogonal Decomposition, which is also known as the Karhunen-Loève decomposition, was introduced independently by Karhunen [32], Loève[40], Pougachev[56], and Obukhov[49]. The method is data based and well suited for the reduction of large scale systems. The derivation of reduced order models by POD is numerically tractable for very large scale systems. The method is applied mainly to analyze the statistical properties of large scale data [41], [43] and has been employed in nonlinear PDE-based models [29], [60], [61]. Since for the applications in this PhD thesis are PDE-based models which generate large scale data, the implemented approach in this thesis for deriving a reduced order model is the Proper Orthogonal Decomposition (POD) approach. The next section will give an introduction to the main concepts and features of this approach.

1.4 Conceptual introduction to proper orthogonal decomposition

The method of Proper Orthogonal Decomposition is based on patterns generated by the simulation data or the experiments. This data is collected by excitations of the physical process through manipulations of process variables, external inputs and disturbances, such as variations of the input energy, chemical compositions, influx rate, and many others.

Since the excitation of a physical process is rarely random, the collected data usually shows regularities, a certain pattern that keeps repeating itself from one time period to the next time period.

An illustrative example is given as follows: Consider a one dimensional slab of length $L=10$ cm. There are three actuators in the slab: at the left end, the middle, and at the right end. The schematic view of the slab is given in Figure 1.2. Suppose the slab is heated at the left end and suppose the temperature distribution along the slab is initially uniform at 200°C . Then the temperature at the left end is increased to 400°C and kept at this level during a certain time period. Suppose that in the numerical model, the one dimensional slab is divided into finite number grid cells, for example into 400

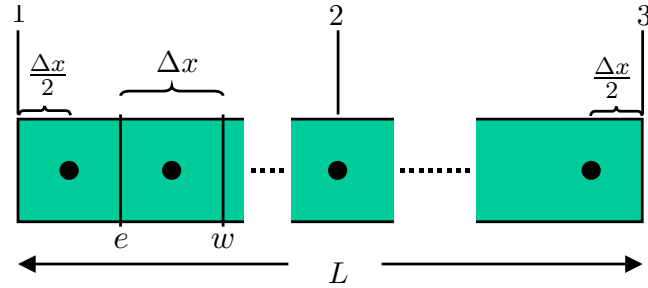


Figure 1.2: One dimensional slab with three actuators

grid cells, hence $\Delta X = 0.025$ cm. The numerical model thus comprises 400 equations describing the temperature of every grid point. From the numerical model, temperature data can be collected.

The temperature distribution along the slab when it is heated at the left end is shown in Figure 1.3 at different time step k . It can be observed that the

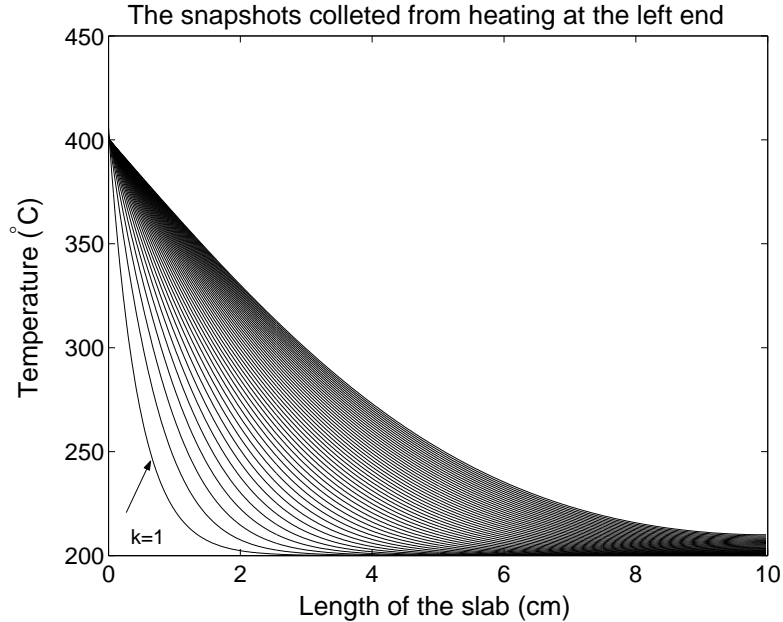


Figure 1.3: The snapshots collected from heating at the left end

temperature distribution as a result of the excitation signal at the left end shows a main repeated pattern. The main repeated pattern of heating at the

left end is shown in Figure 1.4.

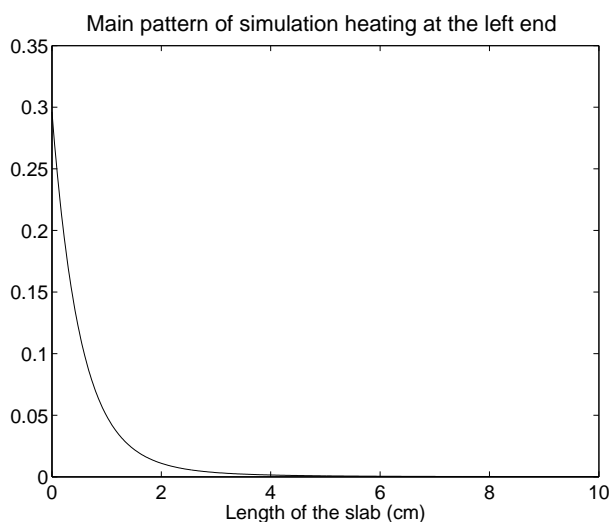


Figure 1.4: The main pattern of the temperature distribution along the slab when it is heated at the left end

Similarly, if the slab is heated at the middle, the collected temperature data also shows a repeating pattern. The temperature data is shown in Figure 1.5 and the main pattern is shown in Figure 1.6.

The example shows that even though the slab is discretized into a large number of grid cells along the spatial domain, the spatial dynamics (such as the temperature distribution along the slab at every time step) shows regularity or a repeating pattern.

Although the example given is very simple, in many more rigorous models, regular patterns are often apparent from the collected simulation data.

This is the basic idea behind Proper Orthogonal Decomposition. If the original model is governed by partial differential equations, then the spatial domain is discretized into a high number of grid cells to approximate the continuous spatial domain as good as possible. As a result, the model comprises a high number of equations since the governing PDE's need to hold for *every grid cell*.

On the other hand, the spatial dynamics or the distribution of a physical variable such as temperature along the slab shows repeating patterns. The

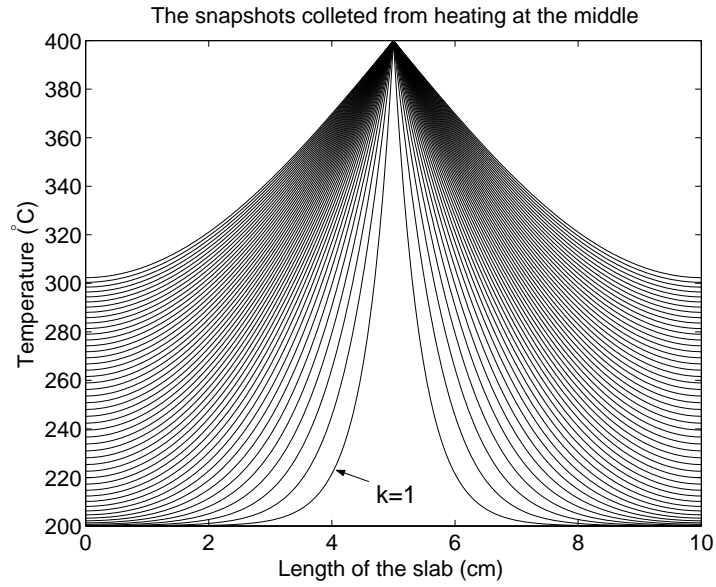


Figure 1.5: The snapshots from the experiment of heating at the middle

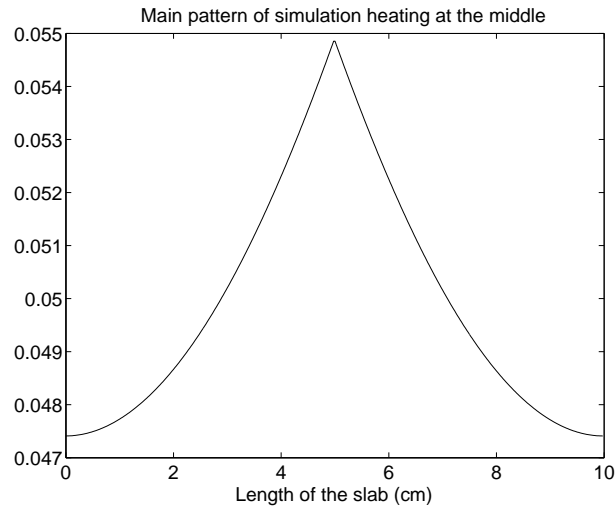


Figure 1.6: The main pattern of the temperature distribution along the slab when it is heated at the middle

number of the apparent patterns is also generally very small compared to the number of grids in the spatial discretization.

This leads to the idea that the distribution of a physical variable, such as tem-

perature at every time step and every location can be expressed, or accurately approximated as functions of the patterns.

Suppose the values of the temperature distribution along the slab at every time step $\mathbf{T}(t)$ can be expressed as the linear combination of K patterns:

$$\mathbf{T}(t) = a_1(t)\varphi_1 + a_2(t)\varphi_2 + \cdots + a_K(t)\varphi_K \quad (1.1)$$

where $\mathbf{T}(t)$ is the vector of the variables over the whole spatial domain and at time step t . Hence $\mathbf{T}(t)$ is a vector with $K = 400$ elements as the spatial domain of the slab is divided into 400 grid cells. The patterns are denoted by $\{\varphi_i\}_{i=1}^K$.

In mathematical terminology, the patterns $\{\varphi_i\}_{i=1}^K$ are called the *basis functions* or the *modes*. A physical variable such as temperature is thus expressed or *decomposed* into a linear combination of spatial patterns. The patterns or basis functions are independent of each other. In mathematical terms it means that the basis functions are *orthogonal* to each other.

Associated with each pattern, there is a time-varying coefficient which varies according to the temperature distribution at time t . The patterns, however, do not depend on time. Hence, if the model of the time varying coefficients is available, we can express or approximate the original variable based on (1.1).

The idea of expanding physical quantities in expansions of the form (1.1) is certainly not new. In fact, it dates back to the work of Joseph Fourier in his memoir *On the Propagation of Heat in Solid Bodies* written in 1805. There he proposed to expand an arbitrary function in a series of trigonometric basis functions. The well known Fourier series are examples of (1.1), where the patterns consist of trigonometric functions. The main difference, however, the Fourier patterns or Fourier basis functions are *not derived* from the data. These trigonometric functions are aimed to approximate any *arbitrary* functions. In POD, the basis functions are derived from the data.

Suppose the number of patterns can be *reduced* only to n patterns such that $\mathbf{T}(t)$ can be expressed as a linear combination of n patterns:

$$\mathbf{T}(t) \approx a_1(t)\varphi_1 + a_2(t)\varphi_2 + \cdots + a_n(t)\varphi_n \quad (1.2)$$

where n is substantially smaller than K in (1.1). If the process variables can be expressed as a linear combination of very few patterns, then an approximate model of the process variable can be derived by building a model for the first n time varying coefficients.

This is the essence of the model reduction by Proper Orthogonal Decomposition. Starting from a rigorous model, simulation data is collected to generate patterns which reflect the main dynamics of the data. Then, the number of patterns are *reduced* such that the effects of the neglected patterns are minimized.

In the decomposition (1.1) and (1.2), the patterns $\{\varphi_i\}$ are ordered according to their relevance to $\{\mathbf{T}(k)\}$. Hence the first pattern is always the main pattern of the data, the second pattern is less dominant than the first one and also independent of the previous one and so on. Therefore the method is called *Proper Orthogonal Decomposition* as it decomposes the original data into a set of independent (orthogonal) patterns by ordering the level of importance of each pattern.

The models of the evolution of the time varying coefficients are the reduced order models. If there are only n patterns used to approximate the original variable, then only n time-varying coefficients have to be found. This is a drastic reduction if n is very small compared to the original number of grid cells.

The models of time varying coefficients can be built by replacing the original variables in the original equations by the linear combinations of the patterns or basis functions in (1.2). Once the original variable in the original models has been replaced by (1.2), the original model is transformed into functions of the time varying coefficients $\{a_i(t)\}_{i=1}^n$. The explicit model of the time-varying coefficients, or the *reduced order model* can be constructed once the original models become the functions of the time varying coefficients only.

Hence, in this approach, there is a mixture of physical insight and black-box modeling approaches in deriving a simpler model from a rigorous one. The patterns are generated from the collected simulation data or experiments. Further, the model of the time varying coefficients of each pattern is derived from the original model which constitute the physical relationships. The simplification here is the reduction of the number of patterns. As a result, the number of time varying coefficients will decrease and only very few time varying coefficients have to be found. Consequently, a model with a lower number of equations than the original model can be constructed.

1.5 Objective and contribution of this thesis

The main goal is to provide a fast and reliable model reduction scheme for process simulation models, in particular Computational Fluid Dynamics models. It is desired that the resulting model reduction scheme will yield a model which can be readily integrated with control system design and optimization techniques.

To meet the goal, the following objectives of this thesis are defined as follows:

1. Investigation of Proper Orthogonal Decomposition as a reduced order modeling tool for various processes.
2. Implementation of reduced order modeling algorithms in the CFD simulating software of a glass melting furnace.
3. Integration of the reduced order models with the control system designs for a number of case studies.
4. Acceleration of reduced order modeling techniques for nonlinear cases

Proper Orthogonal Decomposition has been used extensively as a reduced order modeling technique [71],[29], [34] in fluid dynamics. The applications are still restricted within academic research so that the tool remains relatively unknown to the CFD developers. Though in the control community model reduction is a relatively established research field, the applications of the tools in commercial software remain restrictive.

Since 1980s, the CFD-based glass processing simulation software has become a reliable tool for estimating the process variables in a glass furnace. Despite the reliability of the glass simulating software, it is still too slow for online process monitoring, prediction and control system design.

The black-box approach has been used to accelerate the simulating software, but this approach is very ad-hoc. During this PhD study, for the first time, an integrated model reduction scheme is built for a glass simulation software. As a main feature, a reduced order model based on a solid theoretical approach can be built directly from the original CFD models to allow fast simulations, online and real time predictions.

This thesis provides an extensive guideline of how to implement POD method on the existing software. In this thesis, it is shown explicitly how to calculate a

reduced order model from a model discretized by a CFD discretization scheme. It will help CFD developers in the future to implement the algorithm in the computational software.

Although reduced order models of very low order can be derived for nonlinear systems, the resulting reduced order models are usually not significantly faster than the original models. This is due to the fact that the nonlinear reduced order models are built from the original models.

If the original models are time-varying because the physical parameters are variable dependent, the nonlinear models will also be time varying. In this thesis, an acceleration scheme is introduced by building the reduced order model from a subset of the original equations. This acceleration scheme is based on a method first introduced in [22] as a scheme to estimate the missing data from a static image. In this thesis, the method is extended for dynamical systems. The investigation is among the first in model reduction field which considers an acceleration of model reduction computation by building the reduced order model only by a partial set of the original equations. Chapter 4 gives the detailed description of this acceleration method. In Chapter 4, integration of the reduced order model obtained by the acceleration technique is obtained for a nonlinear heat conduction model. It is shown that the original process can be controlled by using the reduced order model as the base model. The implementation of controller design based on the reduced CFD models will serve as a good basis in the future for the integration of commercial CFD softwares with the controller design.

For a highly non-linear change of operating point in a glass melt feeder, by only reducing the temperature, an acceleration factor of 8 times faster than real time is achieved. Applications of the techniques to other process variables such as velocity fields in the future will achieve a computational gain of about 25-30 faster than real time. Note that the development processor technology is following the Moore's Law [46] that the capacity of computer processors is doubled every year. Hence, implementation of the techniques presented in this PhD thesis will enable real-time model based control of nonlinear processes in the near future.

1.6 Thesis outline

The thesis is organized as follows.

Chapter 2 gives the mathematical foundations for Proper Orthogonal Decomposition. This chapter is mathematically formal, the definitions of basis functions, linear spaces, and model reduction problems are treated from mathematical perspectives here. As illustrations, a step-by-step procedure of reduced order modeling of wave propagation equation is given.

Chapter 3 discusses the application of POD to PDE-based models discretized by the Finite Volume Method, which is one of the most popular discretization techniques implemented in commercial CFD softwares. In this chapter, one dimensional and two dimensional heat conduction models are reduced. A controller is also built based on the reduced order model of the two dimensional heat conduction model.

Chapter 4 extends the discussion of model reduction to the simulation of nonlinear PDE-based models. As an example, a nonlinear heat conduction model is used. In this chapter, the problems encountered with the simulation time of nonlinear models are discussed. This chapter presents an acceleration technique of the nonlinear reduced order modeling. The acceleration technique is developed from the construction of reduced order modeling based on partial set of original equations only.

In Chapter 5, reduced order modeling strategy for a glass feeder is presented. The theories discussed in previous chapters, including the acceleration technique are applied here to a glass feeder model. The reduced order modeling algorithms are implemented on a CFD simulating software of glass melt furnace.

Finally, in Chapter 6, conclusions and recommendations for future research are given.

Proper Orthogonal Decomposition

| | |
|--|--|
| 2.1 Computational models 2.2 The model reduction problem 2.3 Introduction to orthonormal basis 2.4 Model reduction by proper orthogonal decomposition | 2.5 Application of POD to wave propagation equation 2.6 Galerkin projection 2.7 Summary |
|--|--|

2.1 Computational models

The vast development of computing resources and modeling techniques enables a reliable modeling of most physical processes such as laminar fluid flows, heat exchanges, electromagnetic coupling phenomena in semiconductors and many others. The governing equations of such models are usually differential equations, either as partial differential equations or (PDE's, where signals depend on more than one independent variables such as time and space) or ordinary differential equations (ODE's, where signals depend on one independent variable only).

Input/output variables of an ODE based model are considered univariate signals ¹ $\mathbf{y} : \mathbb{T} \rightarrow \mathbb{R}^{n_y}$, $\mathbf{u} : \mathbb{T} \rightarrow \mathbb{R}^{n_u}$ where \mathbf{y} is the output signal and \mathbf{u} is the input signal. Here $\mathbb{T} \subseteq \mathbb{R}$ is the set of the independent variable (usually time) and \mathbb{R} is the set of dependent variables, in this thesis the set of real numbers.

The ODE based models for univariate signals are typically represented as:

$$f(\mathbf{y}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}, \mathbf{u}, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(n)}) = 0 \quad (2.1)$$

where f is an arbitrary, sufficiently smooth function and

$$\mathbf{y}^{(i)} = \frac{d^i \mathbf{y}}{dt^i}, \quad \mathbf{u}^{(i)} = \frac{d^i \mathbf{u}}{dt^i}$$

¹univariate: dependent on a single independent variable only

is the i -th derivative of \mathbf{y} and \mathbf{u} . The equation (2.1) is understood to be satisfied for all time $t \in \mathbb{T}$. The input signal \mathbf{u} may be comprised of manipulating variables, noise, reference signals, or disturbances.

In control engineering, it is common to describe the model (2.1) into an equivalent *state space form*:²

$$\begin{aligned}\dot{\mathbf{z}} &= f(\mathbf{z}, \mathbf{u}) \\ \mathbf{y} &= g(\mathbf{z}, \mathbf{u})\end{aligned}\tag{2.2}$$

where \mathbf{z} is the state vector of the systems and \mathbf{y} is the output vector.

The transformation of an ODE based model to an equivalent state space system can be illustrated in the following example:

Example 2.1.1 Consider a mechanical system of a car suspension. The car suspension system comprises the car mass m , the suspension spring with spring constant κ and a damper with a damping constant b . The external input \mathbf{u} is the force applied to the mechanical system. The output of interest is the displacement \mathbf{y} of the body. The time set is $\mathbb{T} = \mathbb{R}^+$. The mathematical relation between \mathbf{u} and \mathbf{y} is described by the second-order ODE:

$$m \frac{d^2 \mathbf{y}}{dt^2} + b \frac{d\mathbf{y}}{dt} + \kappa \mathbf{y} = \mathbf{u}\tag{2.3}$$

In the form of (2.1), this reads

$$f(\mathbf{y}, \mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{u}) = m\mathbf{y}^{(2)} + b\mathbf{y}^{(1)} + \kappa\mathbf{y} - \mathbf{u}$$

The model in (2.3) can be transformed into a *state space model* by considering the displacement \mathbf{y} and the velocity of the displacement $\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt}$ as state variables. A state vector \mathbf{z} is:

$$\mathbf{z} = \begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \end{pmatrix}$$

The model given by (2.3) can be written in state space form as:

$$\begin{aligned}\dot{\mathbf{z}} &= \begin{pmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \end{pmatrix} = \begin{pmatrix} \mathbf{z}_2 \\ -\frac{1}{m}(b\mathbf{z}_2 + \kappa\mathbf{z}_1 - \mathbf{u}) \end{pmatrix} \\ &= \mathbf{f}(\mathbf{z}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{z}_1 = g(\mathbf{z}, \mathbf{u})\end{aligned}$$

In the case of partial differential equations where inputs and outputs are multivariate signals³, $\mathbf{u} : \mathbb{T} \times \mathbb{X} \mapsto \mathbb{R}^{n_u}$, $\mathbf{y} : \mathbb{T} \times \mathbb{X} \mapsto \mathbb{R}^{n_y}$ where \mathbb{T} denotes

²In many control literature, the states are denoted as \mathbf{x} . Since in this thesis \mathbf{x} denote the spatial coordinates, the state notation is changed into \mathbf{z} to avoid confusion

³multivariate: dependent on more than one independent variables

the set of time as before and \mathbb{X} denotes a spatial domain. Typically for one dimensional space (e.g a slab): $\mathbb{X} \subseteq \mathbb{R}$, for a two dimensional spatial space (e.g a plate) $\mathbb{X} \subseteq \mathbb{R}^2$ and $\mathbb{X} \subseteq \mathbb{R}^3$ in a three dimensional case (e.g a cube).

We call the system and n -D system if $\mathbb{T} \times \mathbb{X} \subseteq \mathbb{R}^n$.

The general form of a PDE-based model can be written analogously to (2.1) as:

$$\mathbf{f}(\mathbf{y}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}, \mathbf{u}, \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(n)}) = 0 \quad (2.4)$$

where for $\mathbb{X} \subseteq \mathbb{R}^q$, the superscript (i) refers now to the multi index $(i) = (i_0, \dots, i_q)$ and

$$\mathbf{y}^{(i)} = \frac{\partial^{i_0 + \dots + i_q} \mathbf{y}}{\partial t^{i_0} \partial x_1^{i_1} \dots \partial x_q^{i_q}}, \quad \mathbf{u}^{(i)} = \frac{\partial^{i_0 + \dots + i_q} \mathbf{y}}{\partial t^{i_0} \partial x_1^{i_1} \dots \partial x_q^{i_q}}$$

In (2.4), there are $q + 1$ independent variables and (2.4) expresses the relation between any finite set of partial derivatives of \mathbf{y} and \mathbf{u} .

Example 2.1.2 Consider a temperature profile T defined over a one dimensional spatial domain $\mathbb{X} = [0, 1]$ and time domain $\mathbb{T} = [0, t_f]$. The one dimensional heat equation is given by:

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \quad (2.5)$$

According to (2.4), there are 2 independent variables, i.e., a time and a spatial coordinate. In this case, $q = 1$ and (2.4) involves two differentiation terms. Let $T^{(1)}$ be the differentiation term with respect to time in (2.5). Let $T^{(2)}$ be the differentiation term with respect to space. That is, with the multi-index notation:

$$T^{(1)} = T^{(1,0)} = \frac{\partial T}{\partial t}, \quad i = (i_0, i_1) = (1, 0) \quad (2.6)$$

$$T^{(2)} = T^{(0,2)} = \frac{\partial^2 T}{\partial x^2}, \quad i = (i_0, i_1) = (0, 2) \quad (2.7)$$

Then (2.5) is equivalent to the (2.4) :

$$f(T^{(1)}, T^{(2)}) = T^{(1)} - \kappa T^{(2)} = 0$$

Since models described by partial differential equations of the form (2.4) do not allow a state space representation (2.2) with finite dimensional state space, they are said to be infinite dimensional or distributed parameter system, see e.g Curtain and Zwart[21].

For computation tractability, the time domain \mathbb{T} and spatial domain \mathbb{X} of the PDE-based models are usually discretised into a discrete time domain \mathbb{T}_d and a discrete spatial domain \mathbb{X}_d . Here, the discrete time and discrete spatial domains are subsets of the continuous time and continuous spatial domain: ⁴.

$$\begin{aligned}\mathbb{T}_d &\subseteq \mathbb{Z}; & \mathbb{T}_d &\subseteq \mathbb{T} \\ \mathbb{X}_d &\subseteq \mathbb{Z}; & \mathbb{X}_d &\subseteq \mathbb{X}\end{aligned}$$

Moreover if \mathbb{T} and \mathbb{X} are bounded sets then such discretisation can be taken to be of finite cardinality such that:

$$\begin{aligned}\mathbb{T}_d &= \{t_1, \dots, t_N\}; & t_i &\in \mathbb{T} \\ \mathbb{X}_d &= \{x_1, \dots, x_K\}; & x_i &\in \mathbb{X}\end{aligned}$$

with $N = \text{card}(\mathbb{T}_d)$ and $K = \text{card}(\mathbb{X}_d)$ are both finite.

The discretised solutions of (2.4) are discrete multivariate signals $T_d : \mathbb{X}_d \times \mathbb{T}_d \rightarrow \mathbb{R}^q$. Suppose $\text{card}(\mathbb{X}_d) < \infty$. Then for each $k \in \mathbb{T}_d(k)$, we can define the vector \mathbf{T}_d as the *collection* of discretised solutions (2.4) over the spatial domain \mathbb{X}_d at the k -th time step as:

$$\mathbf{T}_d(k) = \text{col}_{x \in \mathbb{X}_d} T_d(x, k)$$

where col denotes “stacking subsequent entries in a column vector”.

Suppose that in discrete form, the original PDE model can be written as:

$$\mathbf{T}_d(k+1) = f_d(\mathbf{T}_d(k), \mathbf{u}(k)). \quad (2.8)$$

For convenience, since we are dealing mainly with discrete-time models in this thesis, the subscript d is dropped in the forthcoming discussions.

Many models described by partial differential equations require only few equations. The coupled mass and momentum balance equations [13] for fluid flows only consists of equations for velocity components in different different directions (three equations for 3 dimensional coordinate system) and the mass balance equation. To approximate the continuous solutions, both the spatial and time domains are discretised into a large number of elements (or “grid cells”). This leads to high order discrete models despite the small number of PDE’s in the original model because the PDE’s hold in every grid cell of

⁴the continuous spaces are discretised and represented as vectors. Dimension of time and spatial domains refer to the length of the vectors

the discretised domain. Fine discretisation⁵ of the time domain is usually not problematic because this simply means that the equations have to be solved at a smaller sampling time. On the other hand, fine discretisation of the spatial domain is problematic since at every time step, a high number of equations has to be solved simultaneously.

The state space models as in (2.1) and (2.8) are referred in control engineering terminology as *high order* or *large scale* when $\mathbf{T}(k) \in \mathbb{R}^K$ with $K > 200$.

2.2 The model reduction problem

The large scale models require a lot of computational effort. While many advances have been implemented in the numerical modeling of physical processes to enhance the compatibility of the real process with the simulated one, the models remain computationally intensive. This problem motivates the research of model reduction of computational models, especially those governed by PDE's.

The model reduction problem can be formalized as follows:

- Given a model class (ODE, PDE, discrete, continuous) \mathcal{M} .
- Given a complexity function $c : \mathcal{M} \rightarrow \mathbb{R}^+$. A model $M_1 \in \mathcal{M}$ is considered to be "simpler" than a model $M_2 \in \mathcal{M}$ if:

$$c(M_1) \leq c(M_2)$$

or the complexity of M_1 is less than the complexity of M_2 . A complexity function can be defined as the number of equations in \mathcal{M} .

- Given a misfit function: $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$ with the property:

$$d(M, M) = 0$$

$$d(M_1, M_2) = d(M_2, M_1)$$

where $M, M_1, M_2 \in \mathcal{M}$. The misfit function should be thought as a measure of distance between two models M_1, M_2 in \mathcal{M} . The first property of the misfit function states that the misfit between a model with itself is zero. The second property states that the misfit between two models is a symmetric function

⁵fine discretisation: division of continuous spatial domain into very small grid cells

With these notions we can formalize two model reduction problems

Problem 2.2.1 Given $M \in \mathcal{M}$ of complexity $c_M = c(M)$, find the reduced model $M_r \in \mathcal{M}$ of complexity $c_{M_r} = c(M_r) < c_M$ such that the misfit $d(M, M_r)$ is minimized.

Problem 2.2.1 states that a reduced order model is a simpler model (based on the defined notion of complexity) which minimizes a defined misfit function. In this problem, we try to minimize the misfit by defining first the maximum allowable complexity of the reduced model. Alternatively, we can also first define the misfit threshold and find a simpler model such that the complexity of the reduced model is minimized. This is formulated in Problem 2.2.2

Problem 2.2.2 Given $M \in \mathcal{M}$ and $d_o \geq 0$, find M_r which minimizes the complexity $c(M_r)$ subject to

$$d(M, M_r) \leq d_o$$

To clarify the problem formulations, consider the following example:

Example 2.2.3 Model reduction problem of a discrete state space model

- \mathcal{M} is a set of models that allow a representation of the form:

$$\begin{aligned} \mathbf{z}(k+1) &= A\mathbf{z}(k) + B\mathbf{u}(k) \\ \mathbf{y}(k) &= C\mathbf{z}(k) + D\mathbf{u}(k) \end{aligned} \quad (2.9)$$

- $M = (A, B, C, D) \in \mathcal{M}$ has the *complexity*

$$c(M) = \dim \mathbf{z}$$

in (2.9).

- Define the misfit function $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$ as:

$$d(M_1, M_2) = \| G_1 - G_2 \|$$

where the transfer function $G_i = C_i(Is - A_i)^{-1}B_i + D_i$, $i = 1, 2$ and $\| \cdot \|$ is a norm of a rational transfer function⁶. For example, the norm can be taken as the H_∞, H_2, L_1, L_2 or Hankel norm [62].

⁶refer to basic control theory textbooks for the formulation of a transfer function, for example Ogata[50]

The model reduction problem can be cast into the formulation of 2.2.1: Given an original model $M_1 \in \mathcal{M}$, find a reduced model of the form $M_2 \in \mathcal{M}$ (2.9) with complexity $c(M_2) \ll c(M_1)$ such that

$$d(M_1, M_2) = \| G_1 - G_2 \|$$

is minimized.

There are many ways to achieve a reduced order model which meets the specifications as in Problem 2.2.1 or Problem 2.2.2. An option is to rely on a-priori knowledge. In the case of a set of ODEs for example to throw away the variables which are known to be almost static: $\dot{\mathbf{z}} \approx 0$. The other alternative is to re-model the complex physical phenomena which require large scale models into an integrated set of simple models. This approach has been conducted in [30], where a large scale model of a glass furnace is represented as a series of continuous stirred tank reactors.

The simplifications transform the original large scale model into a smaller number of ODE's. This approach requires a tedious study of the original model and depends on good physical insight. It is difficult to automatize such procedure. Once a new model is given, a tedious study has to be conducted to assess the possibility of changing the model into a simpler one.

The other alternative is to use *black-box modeling*. Instead of modeling the whole state \mathbf{z} , one identifies the relationship between the inputs and the outputs empirically by system identification techniques for a fixed state dimension or a fixed complexity level. The disadvantage of this alternative is the loss of all physical interpretations of the original model.

The specifications of the desired reduced order models in our case are:

1. Generic, repeatable for other physical processes or engineering design. A generic procedure means that detailed physical insight is not required when deriving a reduced order model.
2. Physically related: the original physical relationship needs to be preserved.
3. Computationally attractive.

From the desired specifications of the reduced order model, proper orthogonal decomposition (POD) is used as the model reduction technique in this the-

sis. The method uses the data to project the high dimensional ⁷ simulation or experimental data onto a set of the appropriately chosen basis functions. Subsequently, the basis functions are combined with the original equations to obtain a reduced order model.

This chapter is organized as follows. First, as the method of proper orthogonal decomposition uses orthonormal basis functions to approximate signals, an introduction to the concept of orthonormal basis (section 2.3) is presented. This part describes the mathematical formalities required to introduce orthonormal bases for a specified data set. The introduction to proper orthogonal decomposition (section 2.4) is accompanied by illustrative examples to clarify the mathematical derivations. The introductory section to Proper Orthogonal Decomposition is concluded by an application of POD to data collected from numerical simulation of a wave propagation equation (section 2.5). The comparison of POD to other approximation techniques as Fourier series is shown in section 2.5. Sections 2.3 and 2.4 are mathematically formal. The algorithm of POD basis selection is given at the end of section 2.4. The procedure to build reduced order models using the Galerkin projection is presented in section 2.6. The chapter is concluded with a summary.

2.3 Introduction to orthonormal basis

In this section, the mathematical background of orthonormal basis functions is introduced. The notion of basis function will be used frequently in the forthcoming discussions. Basis functions are closely related to the concept of linear spaces and therefore it is essential to introduce a number of algebraic notions from the theory of linear spaces.

Definition 2.3.1 A linear vector space \mathcal{X} over the real scalar field \mathbb{R} is a nonempty set \mathcal{X} with an addition mapping: $(f, g) \rightarrow f + g$ from $\mathcal{X} \times \mathcal{X}$ to \mathcal{X} and a scalar multiplication mapping $(\alpha, f) \rightarrow \alpha f$ from $\mathbb{R} \times \mathcal{X}$ to \mathcal{X} such that

⁷high dimensional here refers to the dimension of the data vector, for example more than 200 data elements at every time step

the following axioms are satisfied:

$$\begin{aligned}(f + g)(x) &= f(x) + g(x) \\ (f + g)(x) + w(x) &= f(x) + (g(x) + w(x)) \\ (\alpha f)(x) &= \alpha f(x) \\ (\alpha + \beta)f(x) &= \alpha(f(x)) + \beta(f(x)) \\ (\alpha(f(x) + g(x))) &= \alpha(f(x)) + \alpha(g(x))\end{aligned}$$

Definition 2.3.2 A **linear combination** of vectors $\varphi_1, \dots, \varphi_n$ of a linear vector space \mathcal{X} is an expression of the form:

$$\alpha_1\varphi_1 + \alpha_2\varphi_2 + \dots + \alpha_n\varphi_n$$

where $\alpha_i \in \mathbb{R}$.

Definition 2.3.3 For any nonempty subset Φ of a linear vector space \mathcal{X} , the set of all linear combinations is called the *span* of Φ and is denoted by

$$\text{span}\Phi$$

Definition 2.3.4 A set $\{\varphi_i\}_{i=1}^n$ is **linearly independent** if

$$\sum_{i=1}^n \alpha_i \varphi_i = \alpha_1 \varphi_1 + \alpha_2 \varphi_2 + \dots + \alpha_n \varphi_n = 0$$

if and only if $\alpha_i = 0$ for all $i = 1, \dots, n$.

Definition 2.3.5 If the linear vector space \mathcal{X} is the span of a finite set of linearly independent vectors $\{\varphi_i\}_{i=1}^n$, then the set $\{\varphi_i\}_{i=1}^n$ is called a **basis** for \mathcal{X} . In that case, $\mathcal{X} = \text{span}(\varphi_1, \dots, \varphi_n)$ and we write

$$n = \dim \mathcal{X}$$

If there exists no finite set of vectors such that $\mathcal{X} = \text{span}\Phi$ then \mathcal{X} is *infinite dimensional*.

Definition 2.3.6 A **norm** is a nonnegative function on a linear vector space, $\| \cdot \|: \mathcal{X} \rightarrow \mathbb{R}^+ = [0, \infty)$ such that:

- $\| f \| = 0$ if and only if $f = 0$

- $\|f + g\| \leq \|f\| + \|g\|$ for all $f, g \in \mathcal{X}$
- $\|\alpha f\| = |\alpha| \|f\|$ for all $f \in \mathcal{X}$ and $\alpha \in \mathbb{R}$

Definition 2.3.7 A **normed linear space** is a linear vector space \mathcal{X} equipped with a norm $\|\cdot\|_{\mathcal{X}}$ and denoted by $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$.

Definition 2.3.8 An element $f \in \mathcal{X}$ is a **closure point** of $\mathcal{V} \subset \mathcal{X}$ if $\forall \epsilon > 0$ there exists $v \in \mathcal{V}$ such that:

$$\|v - f\| < \epsilon$$

The closure, $\text{closure}(\mathcal{V})$ is the set of all closure points of \mathcal{V} .

Definition 2.3.9 A subset \mathcal{V} of a normed linear space is **dense** in \mathcal{X} if its closure is equal to \mathcal{X} . This means, every element f of \mathcal{X} may be approximated as closely as desired by some elements v of \mathcal{V} , in the sense that for all $f \in \mathcal{X}, \epsilon > 0$ there exists $v \in \mathcal{V}$ such that $\|f - v\| < \epsilon$.

Definition 2.3.10 A normed linear space $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ is **separable** if it contains a dense subset that is countable. An example of a countable set is for example the set of integers and an example of an uncountable set is the set of real number.

Definition 2.3.11 An **inner product** on a linear vector space \mathcal{X} defined over the real field \mathbb{R} is a map $(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that for $f, g \in \mathcal{X}$ and $\alpha, \beta \in \mathbb{R}$ it holds that

- $(f, g) = (g, f)$
- $(f, f) \geq 0$ and $(f, f) = 0$ if and only if $f = 0$.
- $(\alpha f + \beta g, h) = \alpha(f, h) + \beta(g, h)$

Definition 2.3.12 An **inner product space** is a linear vector space equipped with an inner product (\cdot, \cdot) and denoted by $(\mathcal{X}, (\cdot, \cdot))$.

Definition 2.3.13 Every inner product space $(\mathcal{X}, (\cdot, \cdot))$ defines a normed space $(\mathcal{X}, \|\cdot\|)$ by introducing the norm

$$\|f\| = \sqrt{(f, f)}.$$

This norm is referred to as the **induced norm** of $(\mathcal{X}, (\cdot, \cdot))$.

Definition 2.3.14 Two elements φ_i and φ_j in the inner product vector space \mathcal{X} are said to be **orthogonal** if

$$(\varphi_i, \varphi_j) = 0.$$

The orthogonal complement of a subspace $\Phi \subseteq \mathcal{X}$ consists of all vectors that are orthogonal to the vectors in Φ and is denoted by Φ^\perp . Hence,

$$\Phi^\perp = \{x \in \mathcal{X} \mid (x, y) = 0 \quad \forall y \in \Phi\}.$$

In general, $(\Phi^\perp)^\perp \supseteq \Phi$ but if \mathcal{X} is finite dimensional, then

$$(\Phi^\perp)^\perp = \Phi.$$

Definition 2.3.15 A sequence $\{f_n\}_{n=1}^\infty$ in a normed linear space $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ is a **Cauchy sequence** if

$$\|f_n - f_m\|_{\mathcal{X}} \rightarrow 0, \quad \text{as } n, m \rightarrow \infty$$

Example 2.3.16 Let \mathcal{X} be the space of continuous functions on $[0, 1]$ with the norm $\|f\| = (\int_0^1 |f(t)|^2 dt)^{1/2}$ is finite. For $n \in \mathbb{Z}^+$ define

$$f_n(t) := \begin{cases} 0 & \text{for } 0 \leq t \leq \frac{1}{2} - \frac{1}{n} \\ \frac{nt}{2} - \frac{n}{4} + \frac{1}{2} & \text{for } \frac{1}{2} - \frac{1}{n} \leq t \leq \frac{1}{2} + \frac{1}{n} \\ 1 & \text{for } \frac{1}{2} + \frac{1}{n} \leq t \leq 1 \end{cases}$$

Then clearly $f_n \in \mathcal{X}$ for all n .

To check whether $\{f_n\}$ is a Cauchy sequence, calculate:

$$\begin{aligned} \|f_m - f_n\|^2 &= \int_0^1 |f_m(t) - f_n(t)|^2 dt \\ &= \int_{\frac{1}{2} - \frac{1}{m}}^{\frac{1}{2} - \frac{1}{n}} \left(\frac{mt}{2} - \frac{m}{4} + \frac{1}{2}\right)^2 dt + \int_{\frac{1}{2} - \frac{1}{n}}^{\frac{1}{2} + \frac{1}{n}} \left(\frac{mt}{2} - \frac{m}{4} - \frac{nt}{2} + \frac{n}{4}\right)^2 dt + \\ &\quad \int_{\frac{1}{2} + \frac{1}{m}}^{\frac{1}{2} + \frac{1}{n}} \left(\frac{mt}{2} - \frac{m}{4} - \frac{1}{2}\right)^2 dt \\ &= \frac{1}{6} \left(\frac{m}{n^2} - 2\frac{1}{n} + \frac{1}{m}\right) \leq \frac{1}{6m} - \frac{1}{6n} \end{aligned}$$

Clearly the sequence $\{f_n\}$ is Cauchy because $\|f_n - f_m\|^2 \rightarrow 0$ as $m, n \rightarrow \infty$.

Definition 2.3.17 A normed linear space \mathcal{X} is **complete** if every Cauchy sequence has a limit in \mathcal{X} .

In the example of Definition 2.3.15, the pointwise limit $f_{\text{lim}} := \lim_{n \rightarrow \infty} f_n(t)$ with $t \in [0, 1]$ satisfies:

$$f_{\text{lim}}(t) = \begin{cases} 0 & \text{for } 0 \leq t < \frac{1}{2} \\ 1 & \text{for } \frac{1}{2} < t \leq 1 \end{cases}$$

Since $f_{\text{lim}}(t)$ is discontinuous at $t = \frac{1}{2}$, the limit of the Cauchy sequence is not in \mathcal{X} .

Definition 2.3.18 A Hilbert space is an inner product space which is complete.

An example of a separable Hilbert space $\mathcal{X} = L_2(\mathbb{X}, \mathbb{R})$ is a collection of square-integrable functions, such as temperature profiles in Example 2.5.

Let \mathbb{X} be a spatial domain. If the mapping $T : \mathbb{X} \rightarrow \mathbb{R}$ describes the temperature $T(x)$ at position $x \in \mathbb{X}$, then $\mathcal{X} = \{T : \mathbb{X} \rightarrow \mathbb{R} \mid T \text{ continuous}\}$ denotes a collection of such mappings which has the structure of a linear space by setting

$$\begin{aligned} (T_1 + T_2)(x) &:= T_1(x) + T_2(x) \\ (\alpha T_1)(x) &:= \alpha T_1(x) \end{aligned}$$

Then, for $T \in \mathcal{X}$, define the L_2 norm for \mathcal{X} :

$$\|T\|_2 = \left\{ \int_{\mathbb{X}} |T(x)|^2 dx \right\}^{1/2}$$

and suppose that \mathcal{X} consists of all continuous mappings $T : \mathbb{X} \rightarrow \mathbb{R}$ for which $\|T\|_2 < \infty$.

For $T_1, T_2 \in \mathcal{X}$ the inner product (T_1, T_2) is defined as,

$$(T_1, T_2) := \int_{\mathbb{X}} T_1(x) T_2(x) dx$$

The norm $\|\cdot\|_2$ is the **induced norm** of the inner product space $(\mathcal{X}, (\cdot, \cdot))$ and $(\mathcal{X}, (\cdot, \cdot))$ is a separable **Hilbert space**. This Hilbert space is usually denoted by $L_2(\mathbb{X}, \mathbb{R})$.

Definition 2.3.19 Let $\mathbb{I} \subseteq \mathbb{Z}$ be a finite or infinite index set⁸. A set $\{\varphi_i\}_{i \in \mathbb{I}}$ is an **orthonormal set** if:

$$(\varphi_i, \varphi_j) = \delta_{ij} := \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{if } i \neq j. \end{cases}$$

⁸an example of a finite index set is the set of positive integers: $\mathbb{I} = \mathbb{Z}_+ = \{1, 2, \dots, n\}$

for all $i, j \in \mathbb{I}$.

Definition 2.3.20 Let \mathcal{Y} be a subset of a normed space \mathcal{X} . A point $x \in \mathcal{X}$ is a **closure point** of \mathcal{Y} if for all $\epsilon > 0$ there exists $y \in \mathcal{Y}$ such that $\|x - y\| < \epsilon$. The closure of \mathcal{Y} , denoted by

$$\text{closure}(\mathcal{Y})$$

is the set of all closure points in \mathcal{Y} .

Definition 2.3.21 Let the orthonormal set $\{\varphi_i\}_{i \in \mathbb{I}}$ is a countable subset of $(\mathcal{X}, (\cdot, \cdot))$. The set of $\{\varphi_i\}_{i \in \mathbb{I}}$ is **maximal** if the closure of the span of $\{\varphi_i\}$ is \mathcal{X} , that is

$$\text{closure}(\text{span}\{\varphi_i, i \in \mathbb{I}\}) = \mathcal{X}$$

Definition 2.3.22 An orthonormal set $\{\varphi_i\}_{i \in \mathbb{I}}$ is an **orthonormal basis** of a separable Hilbert space $(\mathcal{X}, (\cdot, \cdot))$ if it is a maximal orthonormal set.

The following theorem is the key result of this section and represents our prime interest in orthonormal bases. For the proof, see [21]:

Theorem 2.3.23 *If $(\mathcal{X}, (\cdot, \cdot))$ is a separable Hilbert space with orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$, any element $f \in \mathcal{X}$ can be written as:*

$$f = \sum_{i \in \mathbb{I}} (f, \varphi_i) \varphi_i$$

The real number (f, φ_i) are referred to as the **Fourier coefficients** of f in the basis $\{\varphi_i\}_{i \in \mathbb{I}}$.

The result stated in Theorem 2.3.23 is a very important basis for model reduction by proper orthogonal decomposition. First, it states that any $f \in \mathcal{X}$ can be expanded in terms of an orthonormal basis of \mathcal{X} if f is an element of a separable Hilbert space. Second, it provides explicit expressions for the Fourier coefficients and most important one, it allows optimal projections of f on the subspace of \mathcal{X} in a straightforward way as we will show next section.

In the next section, Theorem 2.3.23 is used as the key tool for optimal data reduction. As a physical motivation for the definitions and results of this section, suppose that the data is a collection of physical variables such as temperatures on a specified spatial domain \mathbb{X} (e.g a furnace).

In physical systems, the temperature profiles represent real and continuous functions with $T(x)$ denoting the temperature at location $x \in \mathbb{X}$. Hence T is supposed to be an element of a set \mathcal{X} of a continuous functions on \mathbb{X} .

In order to allow expansions of temperature data in terms of orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$, the space \mathcal{X} has to be:

- a linear space with addition and scaling such that expressions of the form $T_1 + T_2 \in \mathcal{X}$ and $\alpha_1 T_1 + T_2 \in \mathcal{X}$ make sense inside \mathcal{X} .
- an inner product space. The space \mathcal{X} has to be equipped with the inner product $(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ to allow the concept of orthonormality and more particularly, the concept of distance or error between the temperatures.
- Hilbert and separable to have the existence and the uniqueness of expansion as in Theorem 2.3.23 for any orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$.

In conclusion, these are structural properties so that we can proceed to the data reduction problem.

2.4 Model reduction by proper orthogonal decomposition

Proper Orthogonal Decomposition (POD), has been renowned as an efficient data reduction method [4],[71]. POD is also known with other names, such as Karhunen-Loève decomposition [29],[23], [34] and Principle Component Analysis (PCA) [74]. POD is a method of finding *compact representations* of an ensemble of data in the form of a set of countable, orthonormal basis functions $\{\varphi_i\}_{i \in \mathbb{I}}$ where \mathbb{I} is a countable index set.

The method of proper orthogonal decomposition is closely related to the theory of Hilbert space. If the data or functions to be approximated lie in the Hilbert space, then expansions in terms of orthonormal basis functions are possible as stated in Theorem 2.3.23.

This section is divided into two subsections: the first subsection introduces the solution of the model reduction problems (Problem 2.2.1 and Problem 2.2.2) for the time invariant case and the second subsection introduces the solution of these problems for the time varying case.

2.4.1 Time invariant case

Let \mathcal{X} be a Hilbert space and let $\{\varphi_i\}_{i \in \mathbb{I}}$ be an orthonormal basis of \mathcal{X} . Partition this space into two sets: $\mathcal{X}' = \text{span}\{\varphi_i\}_{i \in \mathbb{I}'}$ and $\mathcal{X}'' = \text{span}\{\varphi_i\}_{i \in \mathbb{I}''}$ where \mathbb{I}' and \mathbb{I}'' are complementary. Then every element $\mathbf{T} \in \mathcal{X}$ can be written as:

$$\mathbf{T} = \sum_{i \in \mathbb{I}'} (\mathbf{T}, \varphi_i) \varphi_i + \sum_{i \in \mathbb{I}''} (\mathbf{T}, \varphi_i) \varphi_i = \mathbf{T}' + \mathbf{T}'' \quad (2.10)$$

with $\mathbf{T}' = \sum_{i \in \mathbb{I}'} (\mathbf{T}, \varphi_i) \varphi_i$ and $\mathbf{T}'' = \sum_{i \in \mathbb{I}''} (\mathbf{T}, \varphi_i) \varphi_i$. Let $a_i = (\mathbf{T}, \varphi_i)$ denote the Fourier coefficients of \mathbf{T} with $i \in \mathbb{I}$.

The decomposition (2.10) has the following properties:

1. $\mathbf{T}' \in \mathcal{X}'$; $\mathbf{T}'' \in \mathcal{X}''$
2. $(\mathbf{T}', \mathbf{T}'') = 0$
3. $\|\mathbf{T} - \mathbf{T}'\|^2 = \|\mathbf{T}''\|^2 = \sum_{i \in \mathbb{I}''} a_i^2$
4. $\|\mathbf{T} - \mathbf{T}''\|^2 = \|\mathbf{T}'\|^2 = \sum_{i \in \mathbb{I}'} a_i^2$
5. $\mathbf{T}' = \arg \min_{\mathbf{T}_o \in \mathcal{X}'} \|\mathbf{T} - \mathbf{T}_o\|$; $\mathbf{T}'' = \arg \min_{\mathbf{T}_o \in \mathcal{X}''} \|\mathbf{T} - \mathbf{T}_o\|$

The last property ⁹ implies that \mathbf{T}' is the best approximation of \mathbf{T} in \mathcal{X}' and \mathbf{T}'' is the best approximation of \mathbf{T} in \mathcal{X}'' .

We can derive an expression for the squared error between \mathbf{T} and \mathbf{T}' as:

$$\|\mathbf{T} - \mathbf{T}'\|^2 = \|\mathbf{T}''\|^2 \quad (2.11)$$

$$= \sum_{i \in \mathbb{I}''} (\mathbf{T}, \varphi_i)^2 \quad (2.12)$$

$$= \min_{\mathbf{T}_o \in \mathcal{X}'} \|\mathbf{T} - \mathbf{T}_o\|^2 \quad (2.13)$$

The squared error $\sum_{i \in \mathbb{I}''} (\mathbf{T}, \varphi_i)^2$ is the *minimum* error since $\mathcal{X}' \subset \mathcal{X}$ is *orthogonal* to $\mathcal{X}'' \subset \mathcal{X}$. The length of an orthogonal projection is minimal, therefore the squared error is minimal.

The properties (3-5) are crucial for solving the model reduction problems for functions or data in the Hilbert space.

⁹the notation $\arg \min$ or $\arg \max$ stands for the minimum argument or the maximum argument with argument is the input to a function, for example x is the argument of $f(x)$

Example 2.4.1 Example of Model Reduction (Problem 2.2.1)

Let \mathcal{X} be a separable Hilbert space with orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$. Consider \mathcal{X} as a *model class*. Define the complexity function $c: \mathcal{X} \rightarrow \mathbb{Z}_+ \cup \{\infty\}$ as the number of nonzero Fourier coefficients in the (2.10). That is,

$$c(\mathbf{T}) := \text{card}(\mathbf{T}, \varphi_i) \quad a_i = (\mathbf{T}, \varphi_i) \neq 0; \quad i \in \mathbb{I} \quad (2.14)$$

Define the *misfit* $d: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ by:

$$d(\mathbf{T}, \mathbf{T}') := \|\mathbf{T} - \mathbf{T}'\| \quad (2.15)$$

where $\|\cdot\|$ is the norm induced by the inner product space $(\mathcal{X}, (\cdot, \cdot))$.

The model reduction problem as in Problem 2.2.1 then reads:

Given $\mathbf{T} \in \mathcal{X}$ of complexity $c_{\mathbf{T}} = c(\mathbf{T})$. Find $\mathbf{T}' \in \mathcal{X}$ of complexity $c(\mathbf{T}') \leq c'$ such that $d(\mathbf{T}, \mathbf{T}')$ is minimal.

Using the property (3) in the decomposition of functions in Hilbert spaces, this problem is equivalent to finding the index set $\mathbb{I}' \subseteq \mathbb{I}$ of cardinality c' such that $\sum_{i \in \mathbb{I}'} a_i^2$ is minimal. In other words, one needs to select c' such that the sums of projections of \mathbf{T} onto the complementary basis $\{\varphi_i\}_{i \in \mathbb{I}'}$ in \mathcal{X}' is minimal.

Note that the solution of this problem is basis dependent. That is, a different orthonormal basis will give different values of $\sum_{i \in \mathbb{I}'} a_i^2$. This means, different basis functions will give different quality of the reduced order model for the same complexity function.

An algorithm for solving the problem of Example 2.4.1 for *finite dimensional* Hilbert spaces \mathcal{X} is as follows:

Let $K = \dim(\mathcal{X})$, then $\mathbf{T} = \sum_{i=1}^K a_i \varphi_i$ and $a_i = (\mathbf{T}, \varphi_i)$. The Fourier coefficients (a_1, \dots, a_K) can be ordered such that:

$$a_{i_1}^2 \geq a_{i_2}^2 \geq \dots \geq a_{i_{c'}}^2 \geq \dots \geq a_{i_K}^2 \geq 0$$

Then:

$$\mathbf{T}' := \sum_{j=1}^{c'} a_{i_j} \varphi_{i_j}$$

has complexity $c(\mathbf{T}') = c'$ and has the minimal misfit $d(\mathbf{T}, \mathbf{T}') = \sum_{j=c'+1}^K a_{i_j}^2$ in the class of all elements $\mathbf{T}_0 \in \mathcal{X}$ of complexities at most c' . In other words, \mathbf{T}' is the solution of this optimal model reduction problem.

Example 2.4.2 Example of Model Reduction Problem 2.2.2

Given the same model class as in Example 2.4.1 where $\mathbf{T} \in \mathcal{X}$ and $d_0 \geq 0$, find $\mathbf{T}' \in \mathcal{X}$ such that $d(\mathbf{T}, \mathbf{T}') \leq d_0$ and $c' = c(\mathbf{T}')$ is minimal among all $\mathbf{T}' \in \mathcal{X}$ for which $d(\mathbf{T}, \mathbf{T}') \leq d_0$.

The solution for the case where $\dim \mathcal{X} = K$ is given as follows: Given $\mathbf{T} \in \mathcal{X}$ and the threshold of the misfit d_0 , then the optimal approximation of \mathbf{T} with minimal complexity $c' = c(\mathbf{T}')$ is:

$$\mathbf{T}' = \sum_{j=1}^{c'} a_{i_j} \varphi_{i_j}$$

where c' is the minimal integer for which $\sum_{j=1}^{c'} a_{i_j}^2 \geq d_0$

The solutions of Example 2.4.1 and Example 2.4.2 depend on the choice of basis. For example, consider the Hilbert space $L_2(\mathbb{X}, \mathbb{R})$ of square integrable functions on the spatial domain $\mathbb{X} = [0, 1]$. A possible choice of standard orthonormal bases for $\mathbb{X} = L_2[0, 1]$ is for example the Fourier modes.

A function f in $L_2(\mathbb{X}, \mathbb{R})$ can thus be expanded in Fourier modes as:

- Fourier mode expansion

$$f(x) = \frac{1}{2}a_0 + \sum_{i=1}^{\infty} a_i \sqrt{2} \cos(\pi i x) + \sum_{i=1}^{\infty} b_i \sqrt{2} \sin(\pi i x)$$

Plots of the first three Fourier modes are given in Figure 2.1.

For one nonzero data point (e.g vector of data at one time step) $\mathbf{T} \in \mathcal{X}$, the trivial choice for the best basis would be any orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$ of \mathcal{X} where $\varphi_1 = \frac{\mathbf{T}}{\|\mathbf{T}\|}$. In this case, $\mathbf{T} = \|\mathbf{T}\| \varphi_1$ and $c(\mathbf{T}) = 1$ which is obviously of minimal complexity among all possible bases of \mathcal{X} .

The problem of finding the best basis for multiple data points, $\{\mathbf{T}(t)\}_{t \in \mathbb{T}} \in \mathcal{X}$ belonging to an arbitrary time set \mathbb{T} is less trivial and will be discussed in the next section.

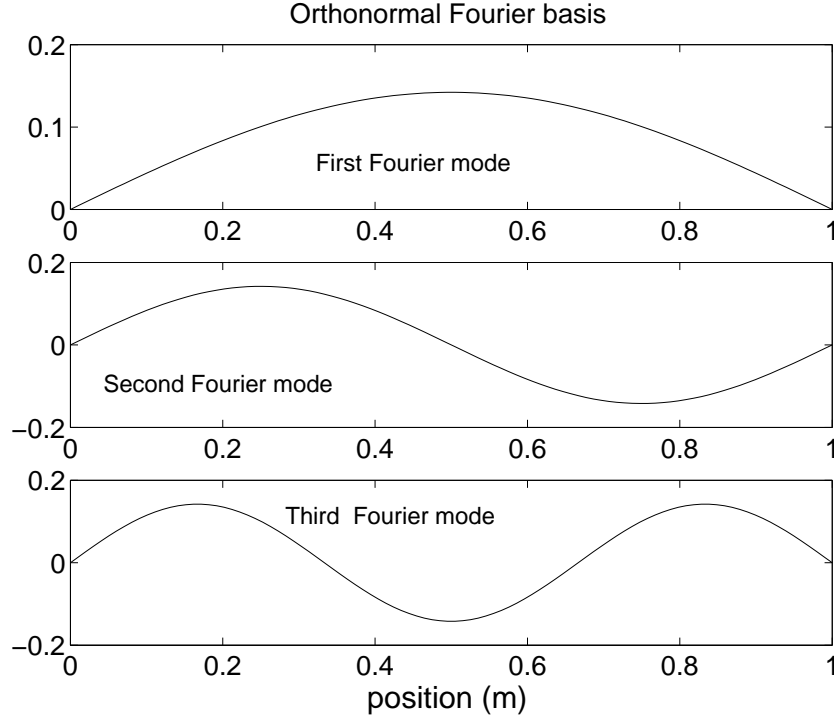


Figure 2.1: The first three Fourier basis functions, a function T can be approximated by $\sum_{i=1}^n a_i \varphi_i$ where φ_i are the basis functions

2.4.2 Time variant case

Let $\mathbb{T} \subseteq \mathbb{R}$ be a time set. Consider for any $t \in \mathbb{T}$ the function $T(\cdot, t)$ which maps $x \in \mathbb{X}$ to \mathbb{R} as an element of a Hilbert space \mathcal{X} with orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$. Based on Theorem 2.3.23, the expansion of $T(x, t)$ is given by:

$$T(x, t) = \sum_{i \in \mathbb{I}} a_i(t) \varphi_i(x) \quad (2.16)$$

where for all $t \in \mathbb{T}$,

$$a_i(t) = (T(\cdot, t), \varphi_i)$$

are the time dependent Fourier coefficients of $T(\cdot, t)$ in the basis $\{\varphi_i\}_{i \in \mathbb{I}}$. If the Hilbert space is infinite dimensional, the index set \mathbb{I} will be, in general, an infinite set and the expansion in (2.16) is an infinite one.

Let

$$\mathbf{T}_{\text{snap}} = \{T(\cdot, t) \in \mathcal{X} \mid t \in \mathbb{T}\} \quad (2.17)$$

be the collection of time dependent measurement, observation, or simulation data. The notation $T(., t)$ means the collection of T over all $x \in \mathbb{X}$. For notational convenience,

$$\mathbf{T}(t) = T(., t)$$

is defined as the vector of the solutions at time t .

Problem 2.4.3 Model reduction for time-varying data in the setting of Problem 2.4.1

Given \mathbf{T}_{snap} and the basis $\{\varphi_i\}_{i \in \mathbb{I}}$ of \mathcal{X} a complexity $c' < c(\mathbf{T}_{\text{snap}})$, where

$$c(\mathbf{T}_{\text{snap}}) = \max_{t \in \mathbb{T}} \text{card}\{(\mathbf{T}(t), \varphi_i) \neq 0; i \in \mathbb{I}\}$$

Find $\mathbf{T}_{\text{snap}}^! = \{\mathbf{T}(t)\}_{t \in \mathbb{T}}$ such that the misfit

$$d(\mathbf{T}_{\text{snap}}, \mathbf{T}_{\text{snap}}^!)$$

is minimal.

The solution for time varying data is as follows:

If the averages of Fourier coefficients $\{a_{i_j}(t)\}_{i \in \mathbb{I}}$ are ordered such that:

$$\langle a_{i_1}^2(t) \rangle \geq \langle a_{i_2}^2(t) \rangle \geq \dots \geq \langle a_{i_c}^2(t) \rangle \geq \dots$$

Then $\mathbf{T}_{\text{snap}} = \{\mathbf{T}^!(t) \mid t \in \mathbb{T}\}$ is optimal because

$$d(\mathbf{T}_{\text{snap}}, \mathbf{T}_{\text{snap}}^!) = \sum_{j=c'+1}^K a_{i_j}^2$$

is minimal.

The most optimal approximation of \mathbf{T}_{snap} in the first c' basis functions will be:

$$\mathbf{T}_{\text{snap}} = \sum_{j=1}^{c'} a_{i_j}(t) \varphi_i$$

The averaging operation $\langle \cdot \rangle$ is defined for $f : \mathbb{T} \rightarrow \mathbb{R}$ as:

\mathbb{T} **discrete, finite** $\text{card } \mathbb{T} = N$

$$\langle f \rangle = \frac{1}{N} \sum_{t \in \mathbb{T}} f(t)$$

\mathbb{T} **discrete, infinite**

$$\langle f \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N f(t)$$

\mathbb{T} **continuous, bounded**

$$\langle f \rangle = \frac{1}{b-a} \int_a^b f(t) dt$$

\mathbb{T} **continuous, unbounded**

$$\langle f \rangle = \lim_{b \rightarrow \infty} \frac{1}{b} \int_0^b f(t) dt$$

Similarly, the model reduction problem for time varying data can also be cast into the form of Problem 2.4.2:

Problem 2.4.4 Given \mathbf{T}_{snap} , the basis $\{\varphi_i\}_{i \in \mathbb{I}}$ of \mathcal{X} , and the misfit

$$d(\mathbf{T}_{\text{snap}}, \mathbf{T}'_{\text{snap}}) = \langle d(\mathbf{T}(t), d(\mathbf{T}'(t))) \rangle \leq d_0,$$

find $\mathbf{T}'_{\text{snap}} = \{\mathbf{T}'(t)\}_{t \in \mathbb{T}}$ such that the complexity $c(\mathbf{T}'_{\text{snap}})$ is minimal.

The solution of this problem is:

If the averages of Fourier coefficients $\{a_i(t)\}_{i \in \mathbb{I}}$ are ordered such that:

$$\langle a_1^2(t) \rangle \geq \langle a_2^2(t) \rangle \geq \dots \geq \langle a_{c'}^2(t) \rangle \geq \dots \langle a_K^2(t) \rangle$$

then

$$\mathbf{T}'_{\text{snap}} \sum_{i=1}^{c'} a_i(t) \varphi_i$$

is the optimal solution with c' is the minimal complexity.

The solutions for the model reduction problems are thus the ordering of the Fourier coefficient averages. If the basis functions $\{\varphi_i\}$ are chosen such that the corresponding Fourier coefficients are ordered:

$$\langle a_1(t) \rangle \geq \langle a_2(t) \rangle \geq \dots \geq \langle a_K(t) \rangle$$

then any truncation up to the first c' basis functions will solve both Problem 2.4.1 and Problem 2.4.2. It will be shown that POD basis functions have the characteristics needed to solve both problems.

2.4.3 POD basis problem

The POD basis problem is formulated as:

Problem 2.4.5 Given observations \mathbf{T}_{snap} of elements of a separable Hilbert space \mathcal{X} (finite or infinite dimensional), find an orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$ such that the averages of i -th Fourier coefficient $a_i(t) = (\mathbf{T}(t), \varphi_i)$ are ordered. That is:

$$\langle (\mathbf{T}(t), \varphi_1)^2 \rangle \geq \langle (\mathbf{T}(t), \varphi_2)^2 \rangle \geq \dots$$

or in terms of the Fourier coefficients $a_i(t) = (\mathbf{T}(t), \varphi_i)$ such that

$$\langle a_1^2(t) \rangle \geq \langle a_2^2(t) \rangle \geq \dots$$

The resulting basis $\{\varphi_i\}$ is the POD basis or often cited as the POD basis functions ¹⁰.

Note that the POD basis functions with the property of ordered Fourier coefficient averages solves both Problem 2.4.1 and Problem 2.4.2. It solves Problem 2.4.1 because the minimal misfit will be given by $\sum_{j=1}^{c'} a_j(t)$. It solves Problem 2.4.2 because the first c' POD basis functions will give the required misfit with the minimal complexity.

In other kinds of basis functions, there is no automatic ordering of the basis functions that re-ordering is required before the solutions of Problem 2.4.1 and Problem 2.4.2 can be found.

The following discussions give a step by step derivation of a POD basis $\{\varphi_i\}_{i=1}^n$.

¹⁰often also cited as POD basis vectors because the basis is represented as a set of vectors in discrete case

2.4.4 POD basis $\{\varphi_i\}$ in a general Hilbert space

Let \mathbb{X} be the spatial domain and \mathbb{T} be the time domain. Let \mathcal{X} be a Hilbert space of mappings from $\mathbb{X} \rightarrow \mathbb{R}$. Note that in the general case, there is no distinction between the finite and infinite cases. Thus the spatial and the time domain may not be bounded and/or finite dimensional. Suppose that $\{\varphi_i\}_{i \in \mathbb{I}}$ is an orthonormal basis of \mathcal{X} . An element $T \in \mathcal{X}$ can then be represented as:

$$T = \sum_{i \in \mathbb{I}} a_i \varphi_i \quad (2.18)$$

where $\{a_i\}_{i \in \mathbb{I}}$ are the Fourier coefficients of T in this basis.

Suppose there is a countable (either finite or infinite) number of functions $\mathbf{T}(t) \in \mathcal{X}$ with $t \in \mathbb{T}$ with \mathbb{T} a countable set. We can think of these collections of functions as signals at different time instants. Note that $\mathbf{T}(t)$ is a function from $\mathbb{X} \mapsto \mathbb{R}$. Hence to express the dependence on \mathbb{X} we will write $T(x, t)$. That is, T is viewed as a mapping from the Cartesian product $\mathbb{X} \times \mathbb{T}$ to the reals.

Let \mathbf{T}_{snap} be equal to the collection $\{\mathbf{T}(t)\}_{t \in \mathbb{T}}$, as defined in (2.17).

The first POD basis function is the function φ_1 which *maximizes* the averaged projection of the elements of \mathbf{T}_{snap} onto φ_1 [34]. The problem of maximizing the projections of the member of the ensemble on average can be formulated as:

$$\varphi_1 = \arg \max_{\varphi_1 \in \mathcal{X}} \frac{\langle \mathbf{T}(t), \varphi_1 \rangle^2}{\|\varphi_1\|^2} \quad (2.19)$$

Example 2.4.6 The formulation stated in (2.19) can be illustrated in a simple example in the finite dimensional case. Consider a heated slab defined over a spatial domain $\mathbb{X} = [0, L]$ where L is the length of the slab. The slab is divided into 2 grid cells (see Figure 2.2) and the temperature profiles $T(x, t)$ are collected for 2 time steps:

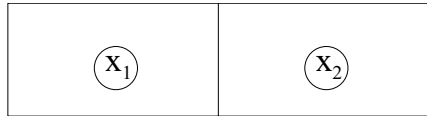


Figure 2.2: A slab divided into 2 grid cells, temperature data is collected for 2 time steps for these grid cells

The observed data \mathbf{T}_{snap} in this case (finite dimensional) is a 2×2 matrix with number of rows equals the number of grid cells and the number of columns equal to the number of time steps. Suppose the temperature data is

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \quad \mathbf{T}(2)) = \begin{pmatrix} T(x_1, t_1) & T(x_1, t_2) \\ T(x_2, t_1) & T(x_2, t_2) \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$$

The data can be plotted in a two dimensional plot where x_1 refers to the first grid cell and x_2 to the second grid cell.

The length of the projectio of a given vector $\mathbf{x} \in \mathbb{R}^2$ onto a vector $\mathbf{v} \in \mathbb{R}^2$ is calculated as the inner product between the two vectors. If $\mathbf{x}, \mathbf{v} \in \mathbb{R}^2$, the inner product can be solved using geometry:

$$a = (\mathbf{x}, \mathbf{v}) = \|\mathbf{x}\| \|\mathbf{v}\| |\cos \theta| \quad (2.20)$$

with θ the angle formed by \mathbf{x} and \mathbf{v} .

When ranging over all $\mathbf{v} \in \mathbb{R}^2$ with $\|\mathbf{v}\| = 1$, a in (2.20) is maximized if \mathbf{v} is *aligned* with the \mathbf{x} . In this case $\mathbf{v} = \mathbf{x} / \|\mathbf{x}\|$ is *optimal* in the sense that a is maximal. The illustration is given in Figure 2.3

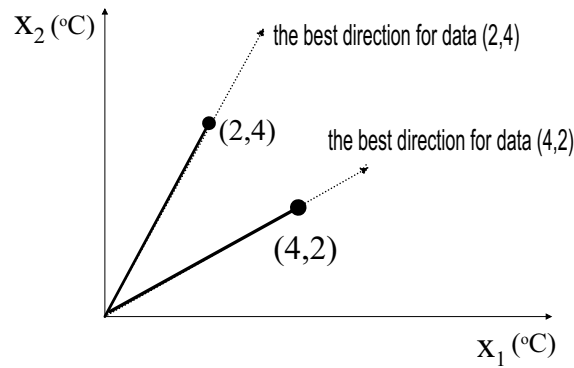


Figure 2.3: The best directions are the ones parallel to the data themselves, here the data is the temperature data of a slab divided into 2 grid cells, with x_1 as the data of the first grid cell and x_2 is the second one

If there are more than one data point such as in the time varying case, then the best direction for the ensemble of the data is the vector \mathbf{v} which maximizes the projection of the data on *on average*. Hence, the best POD basis element would be the one which maximizes the mean of the individual projections of the data and has length one. This is the interpretation of (2.19).

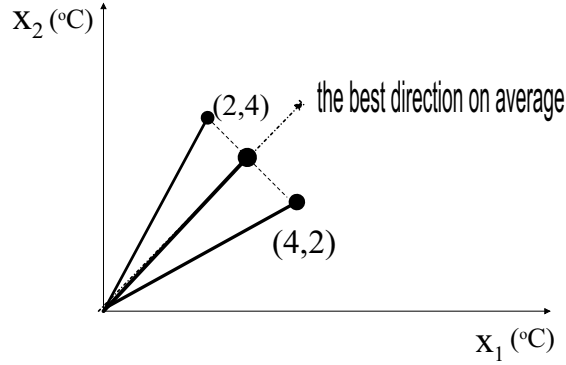


Figure 2.4: The best POD basis is the one which is equal to the averaged best directions of the data and normalized with length 1

The problem stated in (2.19) can be solved by means of a calculus of variations. Introduce the Lagrangian $J(\varphi_1)$ with Lagrange multiplier λ_1 .

$$J(\varphi_1) = \langle |(\mathbf{T}(t), \varphi_1)|^2 \rangle_{\mathbb{T}} - \lambda_1 (\|\varphi_1\|_{\mathcal{X}}^2 - 1) \quad (2.21)$$

To find the extremum of (2.21), we use the variational argument and replace φ_1 in (2.21) by:

$$\varphi_1 + \delta\psi, \quad \delta \in \mathbb{R}; \psi \in \mathcal{X}$$

where ψ is an arbitrary function in the Hilbert space \mathcal{X} .

A necessary condition for φ_1 being an extremum of (2.21) is that:

$$\frac{d}{d\delta} J(\varphi_1 + \delta\psi) |_{\delta=0} = 0 \quad (2.22)$$

Expanding (2.22) using (2.21) results in:

$$\begin{aligned} \frac{d}{d\delta} J(\varphi_1 + \delta\psi) |_{\delta=0} &= \frac{d}{d\delta} \langle |(\mathbf{T}(t), \varphi_1)|^2 \rangle_{\mathbb{T}} - \lambda (\|\varphi_1\|_{\mathcal{X}}^2 - 1) \\ &= \frac{d}{d\delta} [\langle (\mathbf{T}(t), \varphi_1 + \delta\psi)_{\mathcal{X}} (\mathbf{T}(t), \varphi_1 + \delta\psi)_{\mathcal{X}} \rangle_{\mathbb{T}}] \\ &\quad - \frac{d}{d\delta} [\lambda (\varphi_1 + \delta\psi, \varphi_1 + \delta\psi)_{\mathcal{X}}] |_{\delta=0} \end{aligned} \quad (2.23)$$

$$(2.24)$$

where $(\cdot, \cdot)_{\mathcal{X}}$ denotes the inner product operation in the Hilbert space \mathcal{X} . The

terms in (2.23) can be further simplified to:

$$\begin{aligned}
& \frac{d}{d\delta} J(\varphi_1 + \delta\psi) |_{\delta=0} = -\frac{d}{d\delta} [\lambda((\varphi, \varphi) + (\varphi, \delta\psi) + (\delta\psi, \varphi) + (\delta\psi, \delta\psi))] |_{\delta=0} \\
& + \frac{d}{d\delta} [\langle (\mathbf{T}(t), \varphi_1) + (\mathbf{T}(t), \delta\psi) \rangle \langle (\mathbf{T}(t), \varphi_1) + (\mathbf{T}(t), \delta\psi) \rangle] |_{\delta=0} \\
& = \frac{d}{d\delta} \langle (\mathbf{T}(t), \varphi_1)(\mathbf{T}(t), \varphi_1) + (\mathbf{T}(t), \varphi_1)(\mathbf{T}(t), \delta\psi) + (\mathbf{T}(t), \delta\psi)(\mathbf{T}(t), \varphi_1) \rangle |_{\delta=0} \\
& - \frac{d}{d\delta} \lambda((\varphi_1, \varphi_1) + (\varphi_1, \delta\psi) + (\delta\psi, \varphi_1) + (\delta\psi, \delta\psi)) |_{\delta=0}
\end{aligned}$$

By eliminating the terms which are not functions of δ such as $\langle \mathbf{T}(t), \varphi_1 \rangle$ and (φ_1, φ_1) and rearranging, we arrive at:

$$\frac{d}{d\delta} J(\varphi_1 + \delta\psi) |_{\delta=0} = 2\langle (\mathbf{T}(t), \varphi)(\mathbf{T}(t), \psi) \rangle - 2\lambda(\varphi_1, \psi) \quad (2.25)$$

Setting this to zero yields

$$\langle (\mathbf{T}(t), \varphi_1)(\mathbf{T}(t), \psi) \rangle - \lambda(\varphi_1, \psi) = 0 \quad (2.26)$$

Using commutativity¹¹ of $\langle \cdot \rangle_{\mathbb{T}}$ and (\cdot, \cdot) , we find:

$$\langle \langle (\mathbf{T}(t), \varphi_1)_{\mathcal{X}} \mathbf{T}(t) \rangle - \lambda\varphi_1, \psi \rangle = 0 \quad (2.27)$$

Because (2.27) has to be equal to zero for any arbitrary function ψ , then

$$\langle (\mathbf{T}(t), \varphi_1)_{\mathcal{X}} \mathbf{T}(t) \rangle - \lambda_1\varphi_1 = 0$$

is a necessary condition for φ_i to satisfy (2.19).

Now define an operator $C : \mathcal{X} \mapsto \mathcal{X}$ by setting

$$C(\varphi) = \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle \quad (2.28)$$

Then (2.27) reads:

$$C(\varphi_1) = \lambda\varphi_1 \quad (2.29)$$

That is, φ_1 is the eigenfunction of the operator C .

The second POD basis function is found by adding the restriction to the Lagrangian that the second POD basis function must be orthogonal to the first one:

$$J_2(\mu, \lambda_2, \varphi_2) = (\varphi_2, \mathbf{C}\varphi_2) - \lambda_2 [(\varphi_2, \varphi_2) - 1] - \mu(\varphi_1, \varphi_2) \quad (2.30)$$

¹¹commutativity of these two operators means $\langle (A, B) \rangle = \langle (A), B \rangle$

The next POD basis functions are found by finding an extremum to the Lagrangian with the restriction that they have to be orthogonal to the already calculated POD basis. In this way, all POD basis functions are independent. If the operator $C(\varphi)$ in the eigenvalue problem (2.29) is self-adjoint, then the eigenvectors are all orthogonal and the eigenvalues $\lambda_i \geq 0$. We will see in the following lemma that indeed $C\varphi_i$ is self-adjoint.

Lemma 2.4.7 *Let $C : \mathcal{X} \rightarrow \mathcal{X}$ be defined as:*

$$C(\varphi) := \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle_{\mathbb{T}} \quad (2.31)$$

If $\langle \cdot \rangle_{\mathbb{T}}$ and (\cdot, \cdot) commute, then $C : \mathcal{X} \mapsto \mathcal{X}$ defined by (2.31) is self-adjoint, i.e., $C = C^$.*

The adjoint of C is the operator $C^ : \mathcal{X} \mapsto \mathcal{X}$ with the property that $\forall \varphi, \psi \in \mathcal{X}$:*

$$(C(\varphi), \psi) = (\varphi, C^*(\psi))$$

Proof 2.4.8 *Let $\varphi, \psi, \in \mathcal{X}$. Then:*

$$\begin{aligned} (C(\varphi), \psi) &= (\langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle_{\mathbb{T}}, \psi) \\ &= \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t), \psi \rangle_{\mathbb{T}} \\ &= \langle (\mathbf{T}(t), \varphi) (\mathbf{T}(t), \psi) \rangle_{\mathbb{T}} \\ &= \langle (\varphi, (\mathbf{T}(t), \psi) \mathbf{T}(t)) \rangle_{\mathbb{T}} \\ &= (\varphi, \langle (\mathbf{T}(t), \psi) \mathbf{T}(t) \rangle_{\mathbb{T}}) \\ &= (\varphi, C(\psi)) \end{aligned}$$

Hence $C^* = C$.

Since $C(\cdot)$ is a self adjoint (or a symmetric operator), it follows immediately that all POD basis functions are **solutions** to the eigenvalue problem as in (2.29) and (2.30). This leads to the following key theorem:

Theorem 2.4.9 *Given observations \mathbf{T}_{snap} of elements of a separable Hilbert space \mathcal{X} (finite or infinite dimensional). Assume (\cdot, \cdot) and $\langle \cdot \rangle_{\mathbb{T}}$ commute. Then a basis $\{\varphi_i\}_{i \in \mathbb{I}}$ is a solution of the POD basis problem formulated in Problem 2.4.5 if and only if $\{\varphi_i\}$ are the eigenfunctions of the operator C defined in (2.28) with $\|\varphi_i\| = 1$ and the eigenvalues (spectral values) of C satisfy:*

$$\lambda_1 \geq \lambda_2 \geq \dots$$

Each of the POD basis function φ_i solves the eigenvalue problem:

$$\begin{aligned}\langle (\mathbf{T}(t), \varphi_i)_{\mathcal{X}} \mathbf{T}(t) \rangle &= \lambda_i \varphi_i \\ C(\varphi_i) &= \lambda_i \varphi_i\end{aligned}\quad (2.32)$$

such that the following properties are satisfied:

$$\langle a_1(t) \rangle \geq \langle a_2(t) \rangle \geq \dots$$

where

$$a_i(t) = (\mathbf{T}(t), \varphi_i)$$

The expression in (2.32) is thus the expression which solves the POD basis problem 2.4.5. It is valid for a general Hilbert space, either finite or infinite dimensional and it is valid for every inner product provided that the inner product operation commutes with the averaging operation.

For different cases of the dimensionality of the spatial domain \mathbb{X} and the time domain \mathbb{T} we can work out (2.32) as:

- \mathbb{T} discrete and finite and \mathbb{X} finite.

This is the most practical case, because continuous spatial domains are discretised in the numerical models and data is collected for a number finite time samples. Without loss of generality, we assume that $\mathcal{X} = \mathbb{R}^K$ with K is the number of the discretisation points (or cells) of the spatial domain.

In the finite dimensional case, we can represent the collection of the data \mathbf{T}_{snap} as a matrix of dimension $K \times N$:

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \quad \mathbf{T}(2) \quad \dots \quad \mathbf{T}(N)) \quad (2.33)$$

This matrix \mathbf{T}_{snap} is commonly referred as the *snapshot matrix* because we literally collect an ensemble of "snapshots" or data at various time instants in \mathbf{T}_{snap} .

In finite dimensional case, we can define \mathbf{C} in (2.28) as:

$$\begin{aligned}\mathbf{C}(\varphi_i) &= \langle (\mathbf{T}(t), \varphi_i) \mathbf{T}(t) \rangle = \frac{1}{N} \mathbf{T}_{\text{snap}}^\top \varphi_i \mathbf{T}_{\text{snap}} \\ &= \frac{1}{N} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top \varphi_i = \lambda_i \varphi_i \\ &= \mathbf{C} \varphi_i = \lambda_i \varphi_i\end{aligned}$$

where $\mathbf{T}_{\text{snap}}^\top$ is the transpose of the matrix \mathbf{T}_{snap} and $\mathbf{C} = \frac{1}{N} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top$ is the correlation matrix.

Using Theorem 2.4.9, the eigenvalue problem can be solved simultaneously for K eigenvalues, this results in:

$$\mathbf{C}\Phi = \Phi\Lambda$$

where $\Phi \in \mathbb{R}^{K \times N}$, $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_N)$. The diagonal matrix $\Lambda \in \mathbb{R}^{N \times N}$ has the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_N$ in the diagonal components. That is,

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$$

Note that eigenvalues are not necessarily distinct. The unique property of real symmetric matrices is that its eigenvalues are real and its eigenvectors are orthogonal, which means that the eigenvectors are independent even though the eigenvalues may be the same.

The eigenvalues are ordered as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_K \geq 0$. The ordering occurs as the first one maximizing the mean squared projection of the data without orthogonality constraint and the next ones maximizing the mean squared projection with the orthogonality constraint.

The procedure of obtaining the POD basis for the finite dimensional case of \mathbb{T} and \mathbb{X} can be presented in an algorithmic setting:

Algorithm 2.4.10 Algorithm for solving the POD basis problem in finite dimensional Hilbert space $\mathcal{X} = \mathbb{R}^K$.

1. Input: the data \mathbf{T}_{snap} as a matrix:

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \ \mathbf{T}(2) \ \dots \ \mathbf{T}(N))$$

2. Define correlation matrix \mathbf{C} as:

$$\mathbf{C} = \frac{1}{N} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top$$

3. Compute the eigenvalue decomposition ¹²:

$$\mathbf{C}\Phi = \Phi\Lambda$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N) \geq 0$ and Φ is an orthogonal matrix. N is the number of nonzero eigenvalues, $N \geq K$.

¹²eigenvalues routines are standard routines in computational packages such as MATLAB, LAPACK

4. Output: Let φ_i be the i -th column of Φ . Then $\{\varphi_i\}_{i=1}^K$ of \mathcal{X} that solves the POD basis problem.

- \mathbb{T} is infinite, \mathbb{X} is finite

If \mathbb{T} is infinite (infinite samples), then $\mathbf{T}_{\text{snap}} = \{\mathbf{T}(t) \in \mathcal{X}; \quad t \in \mathbb{T} \text{ is an infinite collection. We can rewrite (2.28) as:}$

$$\begin{aligned} C(\varphi) &= \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{T}_{\text{snap}}^\top \varphi \mathbf{T}_{\text{snap}} \\ &= \lim_{N \rightarrow \infty} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top \varphi \end{aligned}$$

The correlation matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$ is:

$$\mathbf{C} := \lim_{N \rightarrow \infty} \frac{1}{N} \sum \mathbf{T}(t) \mathbf{T}^\top(t) \quad (2.34)$$

Then Algorithm 2.4.10 yields a solution to the POD basis problem in this case, provided \mathbf{C} in step 2 is replaced by (2.34).

- \mathbb{T} is bounded and continuous, \mathbb{X} is finite

If \mathbb{T} is continuous and bounded (e.g in the interval $[0, \infty]$), then

$$\begin{aligned} C(\varphi) &= \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \frac{1}{\tau \rightarrow \infty} \int_0^\tau \mathbf{T}^\top(t) \Phi \mathbf{T}(t) dt \\ &= \frac{1}{\tau \rightarrow \infty} \int_0^\tau \mathbf{T}^\top(t) \mathbf{T}(t) dt \varphi \end{aligned}$$

The correlation matrix is defined as:

$$\mathbf{C} := \frac{1}{\tau} \int_0^\tau \mathbf{T}^\top(t) \mathbf{T}(t) dt \quad (2.35)$$

Algorithm 2.4.10 yields the solution by replacing \mathbf{C} in step (2) by (2.35).

- \mathbb{T} unbounded and continuous, \mathbb{X} is finite

If \mathbb{T} is unbounded and continuous (e.g in the interval $[0, \infty]$), then (2.28) is rewritten as:

$$\begin{aligned} C(\varphi) &= \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau \mathbf{T}_{\text{snap}}^\top \varphi_i \mathbf{T}_{\text{snap}} dt \\ &= \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top \varphi_i dt \end{aligned}$$

The covariance matrix \mathbf{C} in step (2) of Algorithm 2.4.10 becomes:

$$\mathbf{C} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top \quad (2.36)$$

For the case where \mathbb{X} is infinite dimensional, it is not possible to form a matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$ as in the case of finite dimensional. The inner product $(\mathbf{T}(t), \varphi_i)$ in (2.28) cannot thus be expressed as a vector multiplication. This term has to be let as it is. So for all the case of infinite dimensional \mathbb{X} :

$$(\mathbf{T}(t), \varphi_i) \mathbf{T}(t)$$

cannot be further simplified.

Accordingly, for different cases of \mathbb{T} , the eigenvalue problem (2.28) can be worked out as:

- \mathbb{T} finite and discrete, \mathbb{X} is infinite dimensional

$$C(\varphi) \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \frac{1}{N} (\mathbf{T}(t), \varphi) \mathbf{T}(t)$$

- \mathbb{T} infinite and discrete, \mathbb{X} is infinite dimensional

$$C(\varphi) = \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} (\mathbf{T}(t), \varphi) \mathbf{T}(t)$$

- \mathbb{T} continuous and bounded, \mathbb{X} is infinite dimensional

$$C(\varphi) = \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \frac{1}{b-a} \int_a^b (\mathbf{T}(t), \varphi) \mathbf{T}(t) dt$$

- \mathbb{T} continuous and unbounded, \mathbb{X} is infinite dimensional

$$C(\varphi) = \langle (\mathbf{T}(t), \varphi) \mathbf{T}(t) \rangle = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau (\mathbf{T}(t), \varphi) \mathbf{T}(t) dt$$

The most encountered problem is finding POD basis for a finite, discrete samples of data in \mathbf{T}_{snap} . The other cases are presented here for completeness purposes of the discussion.

2.4.5 Reduction by POD basis

The reduction idea in POD basis is, the function $T : \mathbb{X} \times \mathbb{T} \mapsto \mathbb{R} \in \mathcal{X}$ where \mathcal{X} is a Hilbert space can be represented by n -th order expansion in the POD

basis. Denote the value of T for $x \in \mathbb{X}$ and $t \in \mathbb{T}$ as $T(x, t)$ and the expansion of $T(x, t)$ by n POD basis functions as $T_n(x, t)$.

$$T_n(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x)$$

In order to have $T_n(x, t)$ approximates $T(x, t)$ well, n POD basis functions which correlate better to $T(x, t)$ must be chosen. The correlation between a POD basis function φ_i and the data is appealed in the corresponding eigenvalue. If the corresponding eigenvalue λ_i is large, then the POD basis function φ_i will approximate the data better than the POD basis function with small eigenvalues. Thus, the larger the eigenvalue, the more correlation exists between the data and the corresponding eigenvector or POD basis function φ_1 .

Note that if the eigenvalue problem is solved simultaneously as in the finite dimensional case of both \mathbb{T} and \mathbb{X} , the eigenvalues are ordered: $\lambda_1 \geq \lambda_2 \dots \lambda_N \geq 0$. The first POD basis function is therefore the most important element of the POD basis because the first eigenvalue is the maximum one and the last POD basis function is the least important element. This leads to a straightforward criterion for n -th order approximation by POD basis.

Define the n -th correlation level by: ((2.37)).

$$P_n = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^N \lambda_i}; \quad n = 1, \dots, N \quad (2.37)$$

The positivity of eigenvalues implies that

$$0 < P_1 \leq P_2 \leq \dots P_N = 1$$

The quantity (2.37) is used to determine the truncation degree of the POD basis function. The number of POD basis elements can be chosen such that the fraction of the first eigenvalues in (2.37) are large enough to capture the information in the data. Ad-hoc criterion commonly applied is n has to be determined for $P = 0.99$, Lumley, et.al [29].

If n is small and yet $P \approx 1$, then with small number of POD basis function we can approximate the original data very well. In turn, we can build a reduced order model consist of n number of equations only. The approximation of the data by POD basis is presented in section 2.5 while the construction of the reduced order model is discussed in section 2.6.

2.5 Application of POD to wave propagation equation

In the previous section, the derivation of a POD basis for both finite and infinite dimensional Hilbert spaces is given. In practice, we use numerical (discrete) models to simulate continuous physical phenomena or collect finite data samples from measurements. This section illustrates the application of POD to a numerical model of a wave propagation and compares the approximation of the solutions by the POD and Fourier basis. The POD and Fourier bases are considered as candidate bases for describing the solutions of the wave propagation equation.

Example 2.5.1 The undamped one dimensional wave propagation for a rod or string of unit length on a spatial domain $\mathbb{X} = [0, 1]$ is governed by the following PDE:

$$\frac{\partial^2 T}{\partial t^2} = \kappa^2 \frac{\partial^2 T}{\partial x^2} \quad (2.38)$$

where $\kappa > 0$.

The solutions $T : \mathbb{X} \times \mathbb{T} \rightarrow \mathbb{R}$ are assumed to belong to the Hilbert space $L_2(\mathbb{X}, \mathbb{R})$. Hence the displacement at time t $\mathbf{T}(t) = T(\cdot, t) \in L_2(\mathbb{X}, \mathbb{R})$ or the displacement are square integrable functions for all $t \geq 0$.

The initial conditions are given by:

$$T(x, 0) = f(x), \quad \frac{\partial T}{\partial t}(x, 0) = g(x).$$

where f and g are the given real-valued functions on $\mathbb{X} = [0, 1]$.

The boundary conditions for the wave equation are:

$$T(0, t) = 0, \quad T(1, t) = 0 \quad (2.39)$$

for all $t \geq 0$.

As the solutions are mappings $T : \mathbb{X} \times \mathbb{T} \rightarrow \mathbb{R}$ that are assumed to satisfy the boundary conditions given by (2.39), we define the Hilbert space:

$$\mathcal{X} = \{\mathbf{T}(t) \in L_2(\mathbb{X}, \mathbb{R}) \mid T(0, t) = 0, \quad T(1, t) = 0\}$$

and assume that *all* solutions $T(x, t)$ in (2.38) belong to \mathcal{X} .

We will derive and compare two orthonormal bases for \mathcal{X} and we will investigate the effect of the projections of the solutions of (2.38) to the subspace of \mathcal{X} spanned by a finite number of basis functions of \mathcal{X} .

Candidate orthonormal bases of \mathcal{X} are for example the set $\{\varphi_i\}_{i=1}^{\infty}$ with

$$\{\varphi_i(x)\} = \{\sqrt{2} \sin(i\pi x)\}$$

The boundary conditions imply that $\varphi_i(0) = 0$ and $\varphi_i(1) = 0$. The basis fulfills the boundary condition (2.39) and is, in fact, an orthonormal basis for \mathcal{X} . This basis is the Fourier basis.

Consequently, the solution $T(x, t)$ of (2.38) and (2.39) can be represented as:

$$T(x, t) = \sum_{i=1}^{\infty} a_i(t) \varphi_i(x).$$

The approximation of $T(x, t)$ is the partial sum

$$T_n(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x). \quad (2.40)$$

If $g(x) = 0$, then one can easily verify that the Fourier coefficients $a_i(t)$ in (2.40) are the solutions of the second order ODE:

$$\ddot{a}_i = -\omega_i^2 a_i, \quad \omega_i = \kappa^2 i\pi \quad (2.41)$$

with the initial conditions:

$$a_i(0) = (f, \varphi_i) = \int_0^1 f(x) \varphi_i(x) dx \quad (2.42)$$

$$\dot{a}_i(0) = (g, \varphi_i) = \int_0^1 g(x) \varphi_i(x) dx = 0 \quad (2.43)$$

Suppose that the initial wave profile $f(x)$ for the 1 meter rod is as given in Figure 2.5. The initial velocity is set to zero, i.e $g(x) = 0$. Hence the solutions of (2.38) and (2.39) are expressed in the Fourier basis are given by:

$$T(x, t) = \sum_{i=1}^1 2 \int_0^{\infty} f(x') \sin(i\pi x') dx' \sin(\kappa^2 i\pi t) \sin(\kappa^2 i\pi x) \quad (2.44)$$

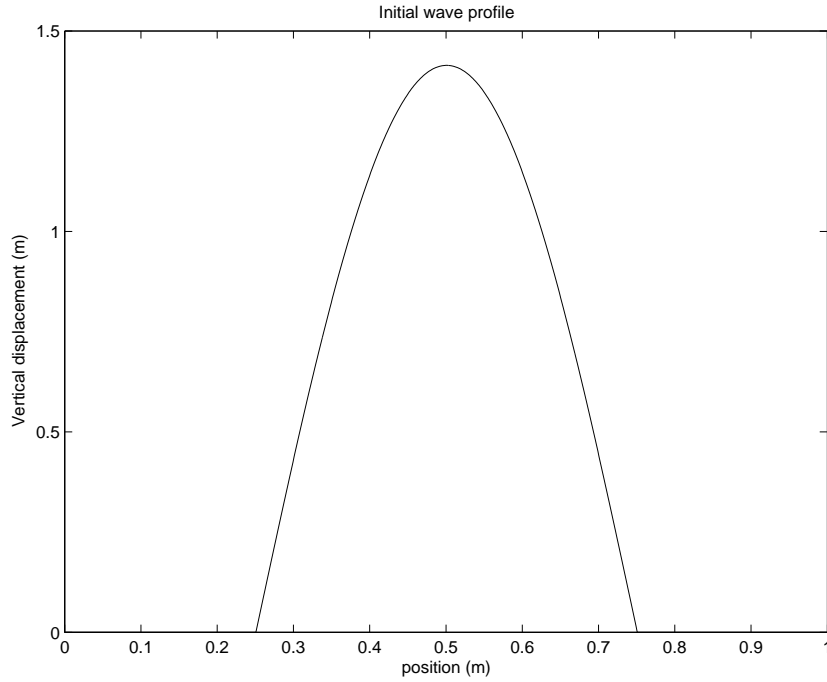


Figure 2.5: The initial profile of wave equation

The numerical model for solving (2.38) is obtained by discretizing the spatial domain \mathbb{X} into 599 grid cells, so $\mathbf{T}(t) = \text{col}_{x \in \mathbb{X}}\{T(x, t)\} \in \mathbb{R}^{599}$ and the time set \mathbb{T} is discretised into 100 equidistant time steps between $t = 0$ and $t = 1$ second.

The POD basis for the wave equations with initial condition as depicted in Figure 2.5 is obtained after collecting data of wave profiles $T(x, t)$ during 100 time steps. Hence $\mathbf{T}_{\text{snap}} = \{\mathbf{T}(\cdot, t_k) \mid k = 1, \dots, 100\}$ with $0 \leq t_1 \leq t_2 \leq \dots \leq t_{100} = 1$. The matrix \mathbf{T}_{snap} has dimension 599×100 .

Algorithm 2.4.10 is applied and results in:

1. collect the simulation data during 100 time steps in a snapshot matrix \mathbf{T}_{snap} :

$$\mathbf{T}_{\text{snap}} = \begin{pmatrix} T(1, 1) & T(1, 2) & \dots & T(1, 599) \\ T(2, 1) & T(2, 2) & \dots & T(2, 599) \\ \dots & \dots & \dots & \dots \\ T(100, 1) & T(100, 2) & \dots & T(100, 599) \end{pmatrix}$$

where $T(x, t)$ denotes the wave profile at $x \in \mathbb{X}$ and at time step $t \in \mathbb{T}$.

2. Calculate the correlation matrix $\mathbf{C} = \frac{1}{100} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top$
3. Conduct the eigenvalue decomposition for \mathbf{C} :

$$\mathbf{C}\Phi = \Phi\Lambda$$

where $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_{599})$; $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{100})$.

The maximum number of eigenvalues for the data of 599 grid cells collected at 100 time steps is 100 eigenvalues¹³. The logarithmic plot of the eigenvalues $\lambda_1, \dots, \lambda_{100}$ is given in Figure 2.6. This plot is commonly referred in the literature as the *eigenspectrum* of C .

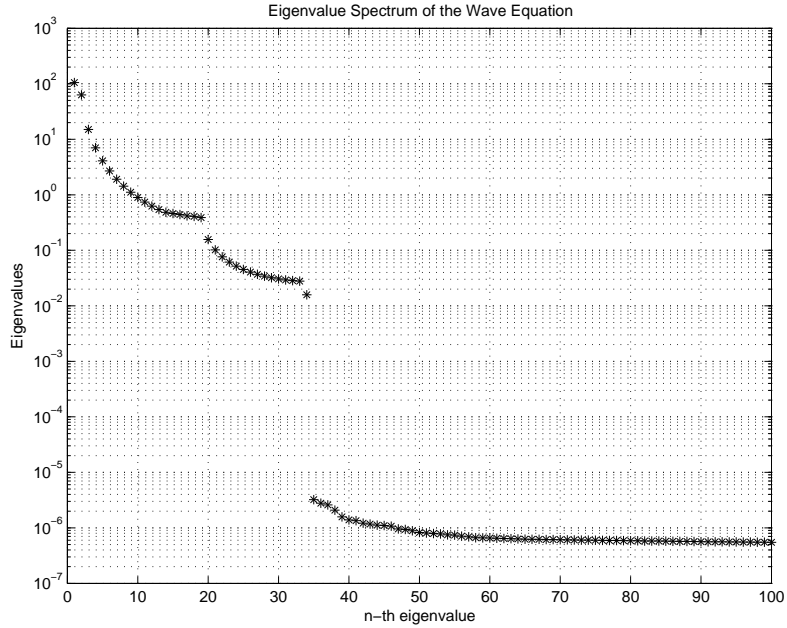


Figure 2.6: The eigenvalue spectrum of the wave propagation data, ordered from the largest to the minimum one

Figure 2.6 shows that there is a sharp drop in the eigenvalue plot after the 35-th eigenvalue. The existence of a sharp drop in the eigenvalue plot means that

¹³the maximum rank of a real matrix which belong to $\mathbb{R}^{N_R \times N_C} = \min(N_R, N_C)$

the original data (which is collected for 100 time steps) is highly correlated with the first 35 eigenvector directions. If n eigenvalues are significantly larger than the others, then we can approximate the original solutions by n POD basis functions only which will result in small deviations.

The first two POD basis functions φ_1 and φ_2 correspond to the two largest eigenvalues λ_1 and λ_2 and are plotted in Figure 2.7. The first POD basis function is equal to the mean of the wave profile data (it is the best direction for all data on average) and the second POD basis is orthogonal to the first one and maximizes the average of the data with the orthogonality restriction.

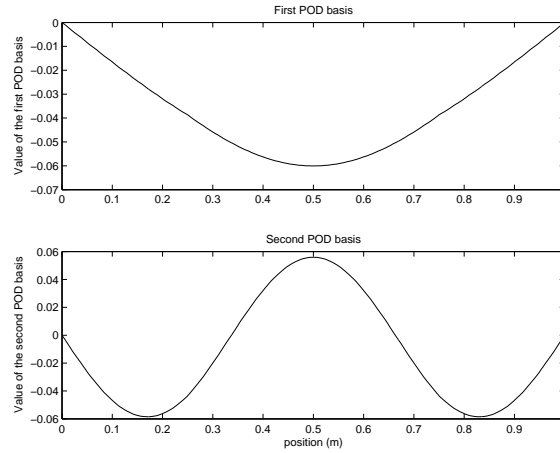


Figure 2.7: The POD bases φ_1 (top) and φ_2 (bottom) used to approximate the wave propagation, the wave profile by the partial sum $T(x, t) \approx T_2^{POD}(x, t) = \sum_{i=1}^2 a_i(t)\varphi_i(x)$

The original solutions (2.44) is compared to the approximation by two Fourier modes and two POD modes as formulated in (2.40) with $n = 2$. The comparisons of the solutions at the 5th, 60th, and 100th time step are shown in Figure 2.8 to Figure 2.10, respectively. From the depicted results, it is obvious that the first two Fourier modes poorly approximate the original solutions. The two Fourier basis function or modes used to approximate the wave propagation are:

$$\varphi_1(x) = \sqrt{2} \sin \pi x, \quad \varphi_2(x) = \sqrt{2} \sin 2\pi x$$

The plots of the two Fourier modes are shown in Figure 2.11. The two Fourier

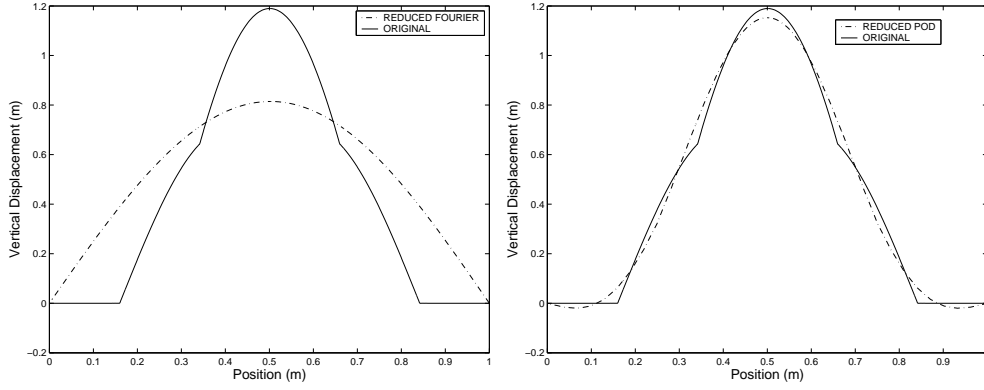


Figure 2.8: The comparison between the analytical solution and the second order approximations using Fourier (T_2^F) (left) and the POD basis (T_2^{POD}) (right) at the 5th time step

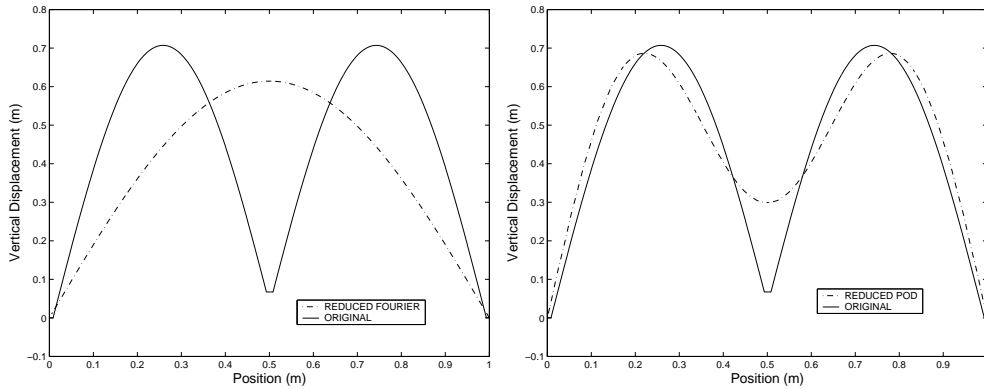


Figure 2.9: The comparison between the analytical solution and the second order approximations using Fourier (T_2^F) (left) and the POD basis (T_2^{POD}) (right) at the 60th time step

modes failed to capture the variations in the original data because these Fourier modes do not resemble the data adequately. See for example Figure 2.10. The combination of the first two Fourier modes fail to capture the constant value at $x \leq 0.25$ and $x \geq 0.75$.

In contrast, the second order POD expansion leads to a better approximation of the original solutions. There are some discrepancies with the original model because we can still take more than two POD basis, but the reconstructed snapshots data by two POD bases follow the original wave propagation better

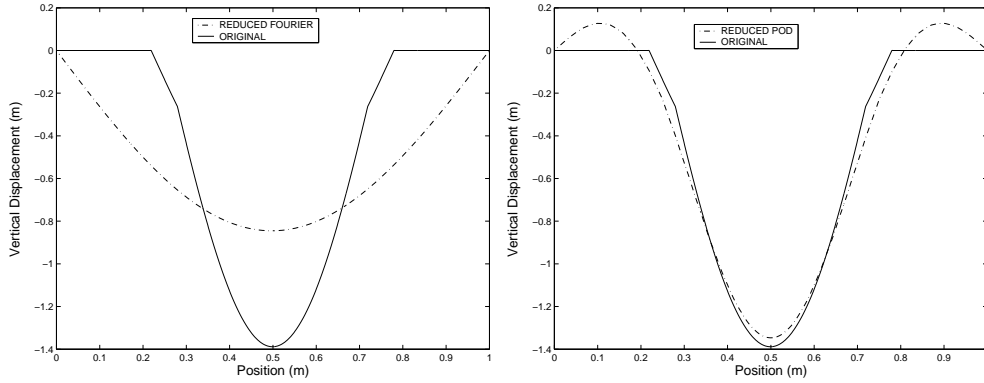


Figure 2.10: The comparison between the analytical solution and the second order approximations using Fourier (T_2^F (left)) and the POD basis (T_2^{POD} (right)) at the 60th time step

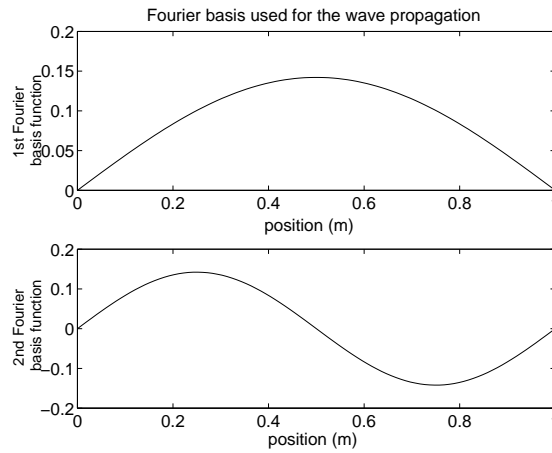


Figure 2.11: The first two Fourier basis functions used to construct the reduced order model

than the reduced order model by Fourier basis.

The first POD basis function is the same as the first Fourier modes, only with reversed direction. The second POD basis function has a different shape than the first Fourier modes and apparently accounts for better approximation of the wave propagation. Hence, POD bases are the optimal orthonormal basis for these solutions of the wave propagation equation.

If we return to the initial discussion of how POD basis differ from other orthonormal basis, it is clear that POD bases are better in approximating a set of data because they are derived directly from the data, while the other orthonormal bases are defined without any relation with the data. This property brings advantages and disadvantages of POD basis.

Indeed, the POD basis can better approximate data from which they are generated than other orthonormal bases. The basis functions therefore reflect the relevant dynamics of the data, provided that these dynamics are captured in the data. However, the validity of approximations defined by the partial sums T_n^{POD} is limited to how well the first n basis functions represent the dynamics of the system.

2.6 Galerkin projection

In the previous section, the procedure for obtaining a POD basis $\{\varphi_i(x)\}$ from simulation data was discussed. The original variables $T(x, t)$ is approximated by $T_n(x, t)$, which are linear combinations of the first n POD basis functions:

$$T(x, t) \approx T_n(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x) \quad x \in \mathbb{X}, t \in \mathbb{T}$$

The objective of this section is to discuss how a reduced order model is constructed using POD basis functions $\varphi_i(x)$, not merely reconstruction by POD basis as presented in the previous section. The reduced order model is the model which describes the evolution of POD basis coefficients $\{a_i(t)\}_{i=1}^n(t)$, $t \in \mathbb{T}$, where $n \ll K$ with K the order of the original model.

Consider a PDE for a function $T : \mathbb{X} \times \mathbb{T} \rightarrow \mathbb{R}$ with $D(\cdot)$ an operator that involves only the spatial derivatives and $M(\cdot)$ is a polynomial operator in temporal derivatives. That is, T is a solution of the equation:

$$M(T) = D(T) \tag{2.45}$$

where $M(\cdot)$ is given by:

$$M = M_0 + M_1 \frac{\partial}{\partial t} + \cdots + M_N \frac{\partial^N}{\partial t^N}$$

and $M_i \in \mathbb{R}$, $i = 0, \dots, N$ and $D(\cdot)$ is an operator that does not involve the temporal derivatives.

Define a residual function for (2.45):

$$R(T) = M(T) - D(T) \quad (2.46)$$

Example 2.6.1 Consider the one dimensional heat equation:

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \quad (2.47)$$

This is of the form (2.45) with

$$D(\cdot) = \kappa \frac{\partial^2}{\partial x^2}, \quad M(\cdot) = \frac{\partial}{\partial t}$$

The residual $R(T)$ is:

$$R(T) = \frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial x^2}$$

Note that T satisfies (4.1) if and only if the residual is zero. Furthermore, it is important to observe that the residual is a function on $\mathbb{X} \times \mathbb{T}$.

Suppose that we have derived a POD basis $\{\varphi_i(x)\}_{i=1}^{\infty}$ for the solutions $T(\cdot, t)$. If we replace the solutions $T(x, t)$ of (4.1) by a *truncated* linear combinations of the first n POD basis functions, i.e.,

$$T_n(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x)$$

then the residual will be

$$R(T_n) = M(T_n) - D(T_n)$$

Since in general $T \neq T_n$, we will have $R(T_n) \neq 0$.

For reduced order modeling, we would like to satisfy the original equations as good as possible. This means that for physical systems, we would like the variables approximated by the POD basis to still follow the governing equations. It implies that the POD basis coefficients $\{a_i(t)\}_{i=1}^n$ have to be determined such that the original governing equations are fulfilled as closely as possible, so the residual $R(T_n)$ viewed as a function on $\mathbb{X} \times \mathbb{T}$ has to be minimized.

The Galerkin projection is defined as:

Definition 2.6.2 Given an orthonormal basis $\{\varphi_i\}_{i=1}^{\infty}$ of a Hilbert space \mathcal{X} , a residual operator R of the model $M(T) = D(T)$. Galerkin projection amounts to setting

$$(\varphi_i, R(T_n(\cdot, t))) = 0; \quad i = 1, \dots, n$$

The Galerkin projection as in Definition 2.6.2 states that the projection of the residual to the span of the first n basis functions vanish. This means that the residual $R(T_n)$ is not correlated to $\{\varphi_i\}_{i=1}^n$ at all. It can be easily seen that for example if we would like to reconstruct the residual $R(T_n)$ using the first n basis functions, then this is equivalent to:

$$R(T_n)(\cdot, t) \approx \tilde{R}(T_n)(\cdot, t) = \sum_{i=1}^n (\varphi_i, R(T_n(\cdot, t))) \varphi_i = 0$$

Thus the approximation of the residual in the span of the first n basis function is zero. The orthogonality of the residual $R(T_n)$ to the span of the basis functions $\{\varphi_i\}_{i=1}^n$ implies that the residual is minimal.

Example 2.6.3 In the previous section we have worked out the derivation of POD basis $\{\varphi_i\}_{i=1}^n$ from a set of simulation data for the wave equation. Herewith the derivation of the reduced order model of the associated POD coefficients for the wave equation is presented. By POD, the solution $T(x, t)$ is expanded in orthonormal basis functions $\varphi_i(x)$.

The wave propagation equation is given by:

$$\frac{\partial^2 T}{\partial t^2} = \kappa^2 \frac{\partial^2 T}{\partial x^2}$$

In the form of (4.1), the operator

$$D(\cdot) = \kappa^2 \frac{\partial^2}{\partial x^2}$$

and the operator

$$M(\cdot) = \frac{\partial^2}{\partial t^2}.$$

In reduced order modeling, the number of basis functions are truncated to n , so that the solutions are approximated by:

$$T_n(x, t) \approx \sum_{i=1}^n a_i(t) \varphi_i(x) \tag{2.48}$$

The residual $R(T_n)$ of the wave equation is given by:

$$R(T_n) = \frac{\partial^2 T_n}{\partial t^2} - \kappa^2 \frac{\partial T_n}{\partial x^2}$$

To find the model of the POD coefficients $\{a_i(t)\}_{i=1}^n$, Galerkin projection of the residual onto the basis functions $\varphi_i(x)$ with $i = 1, \dots, n$ is conducted:

$$(R(T_n), \varphi_i(x)) = 0, \quad i = 1, \dots, n$$

or equivalently

$$\left(\varphi_j(x), \sum_{i=1}^n \ddot{a}_i(t) \varphi_i(x) \right) - \left(\varphi_j(x), \kappa^2 \sum_{i=1}^n a_i(t) \frac{\partial^2 \varphi_i(x)}{\partial x^2} \right) = 0 \quad (2.49)$$

Due to the orthonormality of $\varphi_i(x)$ and linearity of the differential operator $M(\cdot)$, the terms on the left hand side of (2.49) vanish if $i \neq j$.

Hence, the final n -th order reduced model of the one-dimensional undamped wave propagation is:

$$\ddot{a}_j(t) = \left(\varphi_j(x), \kappa^2 \sum_{i=1}^n a_i(t) \frac{\partial^2 \varphi_i(x)}{\partial x^2} \right) \quad (2.50)$$

In the previous example, we generate POD basis $\{\varphi_i\}_{i=1}^{100}$ based on simulation data with the initial wave profile as depicted in Figure 2.5. We would like to use the POD basis derived in that procedure to construct a reduced order model. This reduced order model is then used to simulate the wave propagation excited by other initial condition. For this purpose, we took 3 POD basis functions corresponding to three largest eigenvalues with the eigenvalues as plotted in Figure 2.6.

The set of equations we need to solve to find $a_j(t)$, ($j = 1, 2, 3$) is:

$$\ddot{a}_j = \left(\varphi_j(x), \kappa^2 \sum_{i=1}^3 a_i(t) \frac{\partial^2 \varphi_i(x)}{\partial x^2} \right); \quad j = 1, 2, 3 \quad (2.51)$$

$$(2.52)$$

The wave profile at every time step $\mathbf{T}(t)$ is obtained by discretisation of the spatial domain \mathbb{X} into 599 grid cells. For every *grid cell*, the wave equation

(2.38) has to hold, so the order of the original model $c_T = 599$. The reduced order model (2.51) comprise three ODE's, hence the complexity function $c_T = 3$ for the reduced order model. The misfit function $d_T = R(T) = 0$ in the original model and in the reduced order model $d_T = R(T_3)$ as in (2.46) with $n = 3$ which is minimized.

The **initial conditions** are given as

$$T(x, 0) = \sin \pi x + \sin 5\pi x = f(x) \quad (2.53)$$

$$\frac{\partial T}{\partial t}(x, 0) = 0 = g(x) \quad (2.54)$$

The initial condition $T(x, 0)$ is plotted in Figure 2.12.

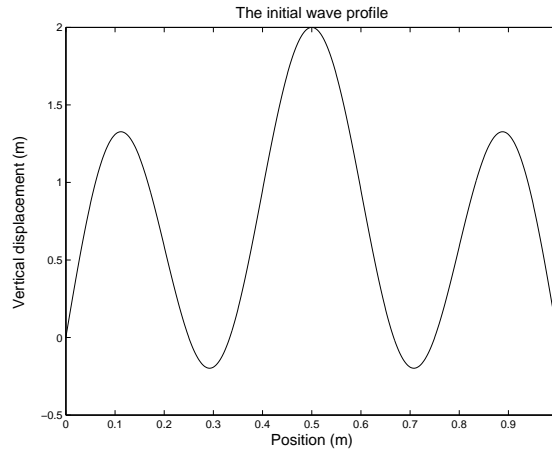


Figure 2.12: The initial wave profile

The corresponding initial conditions for the reduced order model of $\{a_i(0)\}_{i=1}^3$ are obtained by performing the Galerking projection of (2.53) onto the POD basis $\{\varphi_i\}_{i=1}^3$:

$$a_i(0) = (\varphi_i, f) \quad (2.55)$$

$$\dot{a}_i(t)(0) = (\varphi_i, g) \quad (2.56)$$

The reduced order modeling (2.51) with the initial conditions (2.55) uniquely define $\{a_i(t)\}_{i=1}^3$. Note that the POD basis used in (2.51) was derived from simulation data with **different initial condition**.

The comparisons between the original wave profile $T(x, t)$ and the wave profile constructed from reduced order model $T_n(x, t)$ are shown in Figure 2.13 and

Figure 2.14 at the 10th, 30th, 60th, and 100th time step. The reduced model can still follow the original dynamics quite well despite the fact that the POD basis used in the reduced order model was derived from other simulation data.

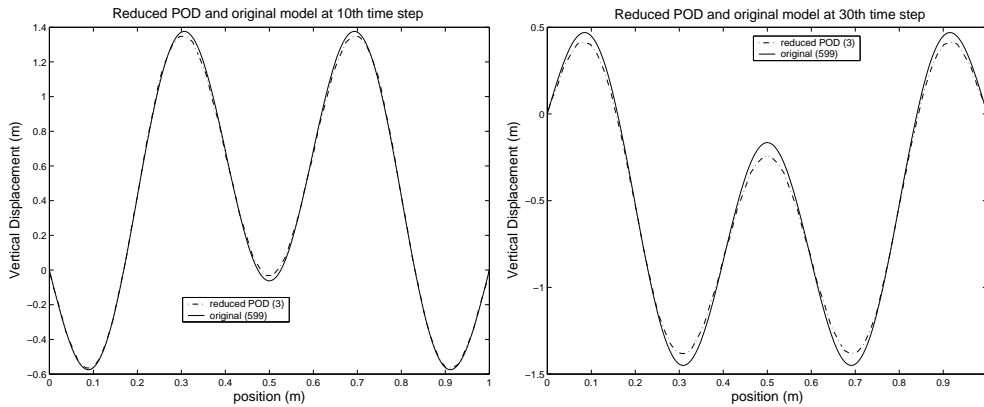


Figure 2.13: The comparisons between reduced and original model at 10th and 30th time step, initial condition is given in Figure 2.12. The POD basis $\{\varphi_i\}$ used in the reduced order model is derived from other simulation data

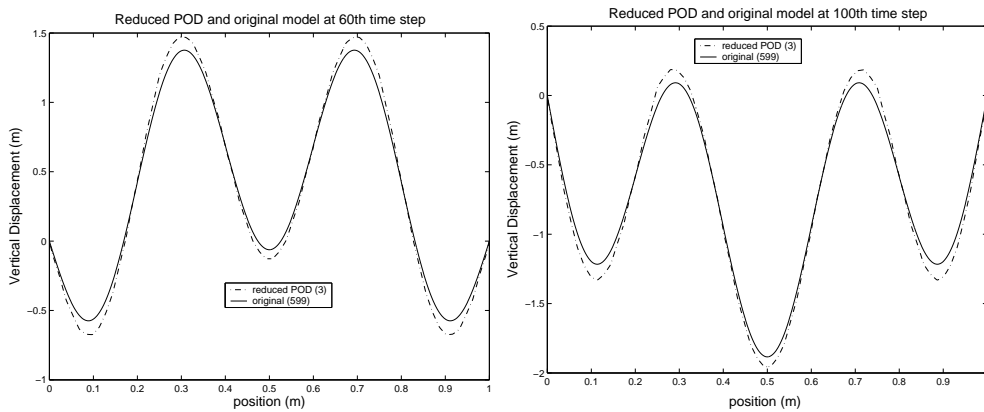


Figure 2.14: The comparisons between reduced and original model at 60th and 100th time step, initial condition is given in Figure 2.12. The POD basis $\{\varphi_i\}$ used in the reduced order model is derived from other simulation data

The reduced order model is obtained by combining the original equations with the POD bases and incorporating the boundary conditions. In black-box modeling, the original equations are considered unknown and the relationship between the input and outputs of interest are determined empirically from the study of the input-output data. This approach for example has been done in [31], where the model of the POD coefficients $a_i(t)$ are derived by an identification technique instead derived directly from the original equations.

In this thesis, the reduced order model is built by projecting **original equations** onto the POD basis. Parameter changes, input changes, and all other changes are thus taken into account by the reduced order model.

2.7 Summary

This chapter gives the introduction of Proper Orthogonal Decomposition as a tool for model reduction. The model reduction problem is introduced in section 2.2. Starting from an original model which belong to a specified model class (ODE, PDE, state space model, etc), we would like to find an a model in the same model class which is less complex than the original one and meet a specified misfit criterion.

The idea to derive a reduced order model originates from the possibility of representing a variable in a suitable, data based coordinate system consisting of orthonormal basis functions such as Fourier, Laguerre or Legendre basis functions. In section 2.3, an introduction to orthonormal basis function is given to provide a mathematical insight of how this approximation is allowed since not all functions can be approximated by expansion of orthonormal basis function.

The method of proper orthogonal decomposition is formalized for general case in section 2.4. Algorithm 2.4.10 gives a step by step procedure of obtaining POD basis for data collected from simulation or experiments.

We demonstrated the usage of POD and its reduction properties in a model of a one-dimensional undamped wave propagation equation in section 2.5. In this section, reconstructions of the wave profiles by POD basis are compared to the reconstruction by Fourier basis. It is shown that POD basis gives less deviation with respect to the original model in comparison with the Fourier basis.

The reduced order model can be found by projecting the original equations onto n POD basis functions as discussed in section 2.6. The number of POD basis function is much smaller than the original dimension of the original model. This results in a reduced order model of lower dimension. As an example, a reduced order model of the wave propagation is constructed by applying the Galerkin projection. To validate the reduced order model, a set of different initial conditions are given. The initial conditions are different than the initial conditions used to derive the POD basis. Since the reduced order model is taking the original equations into account, it is shown that the reduced order model can predict the original wave profiles quite accurately.

Application of POD to Heat Conduction Models

| | | | |
|-----|---|-----|---------------------------------------|
| 3.1 | Introduction | 3.5 | Two dimensional heat conduction model |
| 3.2 | Computational Fluid Dynamics models | 3.6 | Controller design |
| 3.3 | One dimensional heat conduction model | 3.7 | Application of the LQR controller |
| 3.4 | Reduced modeling of one dimensional heat conduction model | 3.8 | Summary |

This chapter is an extended version from the papers [4], [27]

3.1 Introduction

In Chapter 2, the proper orthogonal decomposition (POD) method has been presented as a model reduction technique. The method is based on the approximation of massive data by a limited number of POD basis functions. These POD basis functions can be considered as the *patterns* of the data, derived from the simulation data (see Algorithm 2.4.10 in Chapter 2 for step-by-step procedure for calculating the POD basis functions).

In Chapter 2, the POD technique has been applied to a wave propagation model. The model of the wave propagation equation is a PDE-based model and the simulation model is obtained by discretization of the spatial domain into a high number of grid cells. As a result, in the original model, a high number of equations has to be solved to obtain the solutions. It is shown in Chapter 2 that the high dimensional simulation data can be approximated very well by a very small number of POD basis functions.

A time varying coefficient is assigned to each POD basis function. The reduced

order model is the model for the time varying coefficients. Since the number of POD basis functions is very small compared to the number of grid cells, the reduced order model only comprises as many equations as the number of POD basis functions. Once the POD basis functions have been calculated, the reduced order model can be obtained by performing a Galerkin projection (Section 2.6) of the original equations onto the selected POD basis function. Hence, POD is a model reduction technique which combines physical insight approach (by projecting the *original equations* onto the basis functions) and the black-box model approach (by deriving the POD basis functions from the simulation data set).

In this chapter, we will focus on PDE based models discretized by the Finite Volume approach. The Finite Volume approach is a very popular discretization technique applied in the Computational Fluid Dynamics (CFD) packages. Although POD has been a very popular reduction method in the academic community, its implementation on commercial simulation software is barely known. This might be because most of the technical presentations in the academic papers rarely show the step-by-step procedures of building a reduced order model by POD from the discretized models. This chapter will give an the implementation of the POD method on models discretized by the Finite Volume Method [70, 54].

Heat conduction models in one and two dimensional solid plates will be used as illustrations in this chapter. Although these models do not show the heat transfer in fluids as in the case of CFD models, the implementation of the POD method in these models can also be conducted in CFD models as both are discretized in the same manner. Furthermore, heat conduction models are easier to understand and will give better insight of the POD method itself. The objective of this chapter is to provide a thorough guideline of the POD implementation in models discretized by the Finite Volume Method. In this chapter, the resulting reduced order model will also be used to control the temperature distribution of a solid plate optimally.

This chapter is organized as follows. First, an overview of CFD models will be given in Section 3.2. Subsequently, the Finite Volume Method as the discretization technique of CFD models will be applied to a one dimensional heat equation in Section 3.3. Reduced order modeling of a one dimensional heat conduction is presented in Section 3.4. The two dimensional heat conduction model is discussed in Section 3.5 and the controller design based on the reduced order model is given in Section 3.7.

3.2 Computational Fluid Dynamics models

Computational Fluid Dynamics (CFD) is among the most prominent modeling techniques for flows and heat transfer processes in fluids. In CFD, mass, momentum, and energy balance equations defined over a spatial domain are solved [13]. The balance equations are in the form of PDE's, and consist of

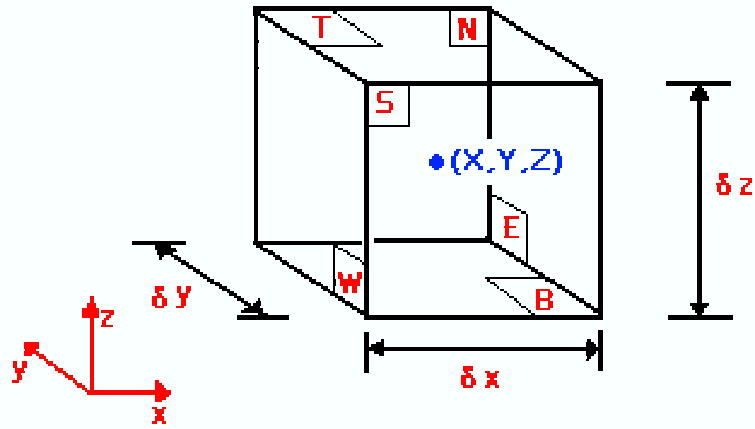


Figure 3.1: A grid cell in three dimensional Cartesian coordinate system. The coordinate system is given by (x, y, z) coordinate. The capitals E, W, T, B, N, S refer to the eastern, western, top, bottom, north, and south faces. The grid cell size is given by $\delta_x \times \delta_y \times \delta_z$. The sizes are usually very small so that the discretized model can follow the changes of the temperature and velocity fields

- The mass balance equation:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}) \quad (3.1)$$

- The momentum balance equation:

$$\frac{\partial}{\partial t} \rho \mathbf{v} = -[\nabla(\rho \mathbf{v} \mathbf{v})] - \nabla p - [\nabla(\mathbf{w})] + \rho \mathbf{g} \quad (3.2)$$

- The energy balance equation:

$$\frac{\partial \rho c_p T}{\partial t} = -\nabla(\rho c_p T \mathbf{v}) + \nabla(\kappa \text{grad } T) + q \quad (3.3)$$

where $\mathbf{v} \in \mathbb{R}^3$ is the vector of the fluid velocity in the x, y, z direction, \mathbf{w} is the vector of the shear stresses on the fluid, ρ is the fluid density, p is the pressure, ∇p is the vector of the gradient of the pressure, and q is the external energy sources. In the Cartesian coordinate system, the components of ∇p is $\partial p/\partial x$, $\partial p/\partial y$, and $\partial p/\partial z$. The shear stress vector \mathbf{w} has nine components: $\tau_{xx}, \tau_{xy}, \tau_{xz}, \tau_{yy}, \tau_{yx}, \tau_{yz}, \tau_{zz}, \tau_{zy}, \tau_{zx}$ [13]. The velocity components in the x, y, z directions are usually denoted by (v_x, v_y, v_z) , respectively.

If the density ρ and the viscosity μ are constant, the set of the balance and the momentum equations are referred to as the Navier-Stokes equations. The divergence operator $\nabla(\cdot)$ for the Cartesian coordinate system is:

$$\nabla(\cdot) = \frac{\partial}{\partial x}(\cdot) + \frac{\partial}{\partial y}(\cdot) + \frac{\partial}{\partial z}(\cdot)$$

It should be noted that $[\nabla \cdot \rho \mathbf{v} \mathbf{v}]$ and $[\nabla \cdot \mathbf{w}]$ are not simple divergences because of the tensorial nature of $\rho \mathbf{v} \mathbf{v}$ and \mathbf{w} . For further details, see [13].

It should be pointed out that the balance equations also hold for other coordinate systems, such as cylindrical and radial coordinates. Details of mass, momentum, and energy conservation equations for other coordinate systems can be found in Bird, et.al [13].

In CFD models, these balance equations are discretized by discretizing both the spatial domain \mathbb{X} and the time domain \mathbb{T} . A grid cell in a rectangular spatial domain \mathbb{X} , together with the direction of the Cartesian coordinate system is given in Figure 3.1.

Some popular discretization methods applied in CFD are for example the Finite Element Method and the Finite Volume Method. In this thesis, we focus on the discretisations by the Finite Volume Method. In this method, the balance equations are discretised over a finite volume cell and a specified time step.

CFD modeling software is used extensively in chemical, mechanical, oil refinery, aviation industries as well as in meteorology and oceanography. Some well-known CFD commercial packages are FLUENT, CD-ADAPCO, CFX. Starting in the 1980s, the Netherlands Institute of Applied Physics (TNO-TPD) has developed the Glass Tank Model (GTM) for CFD simulations of glass processing.

CFD packages generally consist of three main modules:

1. Pre-processor modules.

In the pre-processor modules, the following details are defined:

- the specifications of the spatial domain \mathbb{X} such as dimensions and shapes, including its discrete grid divisions. Advanced CFD packages usually have a default mesh generator.
- the equations to be solved, the standard equations are Navier-Stokes equations and the energy conservation equations.
- the boundary conditions.
- the specifications of the different phases in the spatial domain (gas, liquid, solid, mixed phases).
- the physical parameters such as density ρ , viscosity μ and heat conductivity κ , often strongly temperature dependent.
- specifications of energy sources which influence the energy and flow distribution such as the local electrical energy sources.

2. Solver modules.

In this module, the following steps are performed:

- discretization of the governing equations.
- online update of the time varying physical parameters if necessary.
- iterative calculation of the discretized equation solutions.

3. Post processor module

In this module, the results of the simulations for instance the temperature in each grid point at all time steps are presented through graphical interfaces (1D,2D, or 3D plots) and exported as data files.

Similar to other computational softwares, the efficiency of CFD models depends on many factors such as memory allocations, numerical methods applied in the solver module, and the data transfer. The scope of this thesis is focused on minimizing or reducing the computational effort required to solve the discretized equations. Other issues such as efficient memory allocations, optimal gridding of the spatial domains, subroutines structures which also affect the performance of a computational software are beyond the scope of this thesis.

CFD models often provide representative simulation models of the real systems and play an important roles in the automatization of process industries. Nowadays, operators and engineers are able to obtain the accurate predictions

of the changes without conducting experiments on the real system [9, 17, 18, 8]. Often, the real system is operating in very volatile conditions such as high operating temperatures and only a limited number of sensors can be installed in the system. The availability of reliable CFD models enables operators and engineers to approximate or estimate the process variables not measured by sensors. The available estimations of the data help them to analyze the plant safely and efficiently since there is no need to install sensors in very extreme operating conditions to access the unmeasured variables.

On the other hand, CFD models often require a lot of computational time. Depending on the complexities of the models, it is generally quite difficult to obtain fast simulation models. CFD models are adequate as off-line simulation tools, but not fast enough to be used as prediction modules or to be used as a basis for control design. One of the contributing factors in the intractability of CFD models is the fact that the mesh generators divide the spatial domain into many grid cells in order to approximate the continuous behavior. The number of grid cells in CFD models are typically in the range 10^3 to 10^8 grid cells [71], [16]. Hence for every time step, the CFD software has to find 10^3 to 10^8 solutions simultaneously.

It is therefore impossible to use CFD models for model based control system design. Even though the CFD models provide a representative model of the real system, the models cannot be used to enhance the product quality, to minimize the consumption of energy inputs, or to improve the economic operation of the plant by means of controllers. Suppose a real valued variable $T(x, t)$ is defined over a discrete spatial domain \mathbb{X} consisting of K grid points as a discretized solution of a PDE. For fixed $t \in \mathbb{T}$, let the vector $\mathbf{T}(t)$ comprise all K discrete spatial solutions over \mathbb{X} at time $t \in \mathbb{T}$. For example, $T(x, t) \in \mathbb{R}$ can be a temperature distribution in a chemical reactor \mathbb{X} and solution of the energy balance (3.3). Consider further $\mathbf{T}_{\text{ref}}(t)$ as the reference or desired temperature distribution at time t , and let \mathbf{u} be the adjustable inputs such as fuel or electrical inputs.

Suppose the discretised PDE has the following structure ¹:

$$\mathbf{A}(k)\mathbf{T}(k+1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (3.4)$$

where $\mathbf{T}(k+1)$ is the vector of the unknowns at time $k+1$, $\mathbf{u}(k)$ is the vector of the external input parameters such as manipulating variables or measurable disturbances. The matrices \mathbf{A} , \mathbf{A}_0 , \mathbf{B} vary at every time step if the original model is non-linear. These matrices do not depend on time if the

¹models discretised implicitly are usually represented as $Ax = b$

model is linear time invariant and in linear case we denote them as A, A_0, B , respectively.

In optimal model based control, one would like to manipulate the inputs \mathbf{u} in (3.4) so as to minimize the error between the given reference signal \mathbf{T}_{ref} and the actual signals \mathbf{T} . The error $\mathbf{T} - \mathbf{T}_{\text{ref}}$ is defined in a suitable norm over a specified time period.

In the case of temperature control, the control objective is to find the minimal input energy needed to reach the desired temperature distribution over a certain period of time. If predictions of the temperature are available, for example for the next one or two hours ($N = 1\text{hr}$), this knowledge can be used by the controller to track the desired temperature optimally.

Mathematically, the optimization procedure to find the optimal input signals \mathbf{u}_{opt} amounts to solving the following problem:

$$\mathbf{u}_{\text{opt}} = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{t=0}^{N_{\text{opt}}} \|\mathbf{T}_{\text{ref}}(t) - \mathbf{T}(t)\| \quad (3.5)$$

where \mathcal{U} is the collection of feasible inputs, that is the collection of input signals whose variations, amplitudes, and other characteristics remain within allowable constraints. The formulation (3.5) states that the optimal input \mathbf{u}_{opt} is the optimal input signal that minimizes the error $\mathbf{T}_{\text{ref}} - \mathbf{T}$ in a suitable norm over a control horizon of N_{opt} samples. For example, the norm $\|\cdot\|$ can be an L_2 norm:

$$\|\mathbf{T}_{\text{ref}}(t) - \mathbf{T}(t)\|^2 = \sum_{x \in \mathbb{X}} |T_{\text{ref}}(x, t) - T(x, t)|^2$$

in discrete case.

Hence to determine the optimal control input $\mathbf{u}_{\text{opt}}(t)$, a prediction of the process variables for N future time steps has to be compared with future reference signals $\mathbf{T}_{\text{ref}}(t)$.

If the simulation model requires a lot of computing effort, the length of the prediction N is limited. The optimization problem is also difficult to be solved if the reference signals and the process variables are of high dimension, i.e. $\mathbf{T}_{\text{ref}}(t) \in \mathbb{R}^K$, with $K > 200$. This is usually the case for discrete PDE based models since the spatial domain \mathbb{X} is discretized into more than hundreds of grid cells. Current optimization algorithms work well with at most 100 – 200 variables, which is far below the average discretization level of CFD models.

Another alternative to find the optimal control input \mathbf{u}_{opt} in (3.5) is by choosing a subset of the spatial domain $\mathbb{X}_0 \subseteq \mathbb{X}$ instead of the whole spatial domain \mathbb{X} . In practice, process variables at specific locations are more important than the others. The domain \mathbb{X}_0 can be thought of as the collection of sensor locations or outputs of interest. Let $\tilde{T} : \mathbb{X}_0 \times \mathbb{T} \rightarrow \mathbb{R}$ be the measured process variables in \mathbb{X} , i.e. $\tilde{T}(x, t)$ denote the value at location $x \in \mathbb{X}_0$ and $t \in \mathbb{T}$. Then let the collection of sensor measurements or output parameters at time t as $\mathbf{y}(t) = \tilde{\mathbf{T}}(t) = \tilde{T}(\cdot, t)$. Similarly, let $\mathbf{y}_{\text{ref}} = \tilde{\mathbf{T}}_{\text{ref}}(t)$ where $\tilde{\mathbf{T}}_{\text{ref}}(t)$ is the vector of points $T_{\text{ref}}(x, t)$ with $x \in \mathbb{X}_0$.

In this case, the optimal control problem becomes:

$$\mathbf{u}_{\text{opt}} = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{t=0}^N \|\mathbf{y}_{\text{ref}}(t) - \mathbf{y}(t)\|$$

where as before

$$\|\mathbf{y}_{\text{ref}}(t) - \mathbf{y}(t)\|^2 = \sum_{x \in \mathbb{X}_0} |\tilde{T}_{\text{ref}}(x, t) - \tilde{T}(x, t)|^2 \quad (3.6)$$

Obviously the measured variables $\tilde{T}(x, t)$ is related to $T(x, t)$ and it is difficult to infer $T(x, t)$ from $\tilde{T}(x, t)$ as $T(x, t)$ has also to be calculated. The optimization of (3.6) is still difficult when a large scale model is used, even with limited number of outputs \mathbf{y} to be optimized.

In Chapter 2, we have presented POD as model reduction technique. In POD, the process variables of interest are approximated as the expansions of the POD basis functions $\{\varphi_i\}_{i \in \mathbb{I}}$.

If the process variables $\mathbf{T}(t)$ are expanded in n POD basis functions as:

$$\mathbf{T}(t) = \Phi \mathbf{a}(t)$$

where $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_n)$ and $\mathbf{a}(t) = \text{col}(a_1(t), a_2(t), \dots, a_n(t))$, then the optimization problem (3.5) can be written as:

$$\mathbf{u}_{\text{opt}} = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{t \in \mathbb{T}} \|\mathbf{a}_{\text{ref}}(t) - \mathbf{a}(t)\| \quad (3.7)$$

where $\mathbf{a}_{\text{ref}}(t) = \Phi^\top(t) \mathbf{T}_{\text{ref}}$ and $\mathbf{a}(t) = \Phi^\top \mathbf{T}(t)$ is calculated from the reduced order model. Note that instead of optimizing the error between the whole field $\mathbf{T}(t)$, the new optimization problem is set as the minimization of the error

between the desired POD coefficients and the predicted POD coefficients. If the number of POD coefficients n is very low, $\Phi \mathbf{a}(t)$ is an approximation of $\mathbf{T}(t)$ and the optimization problem in the reduced order model is more feasible to solve than the original one (3.5).

The following section discusses the application of POD on the one dimensional heat conduction model discretized by the Finite Volume Method.

3.3 One dimensional heat conduction model

3.3.1 Introduction

In this section, a reduced order model of a one dimensional heat conduction process is presented in details. The governing equation of the heat conduction process is discretized by the Finite Volume Method, which is one of the mostly applied discretization methods in CFD commercial packages. This section will give an overview of the discretization scheme and the construction of a reduced order model by Proper Orthogonal Decomposition.

Consider a slab of length $L = 0.1$ m. At specific locations (denoted by numbers $\{1, 2, 3\}$ in Figure 3.2), heat sources are installed. A sketch of the slab is depicted in Figure 3.2. The heat transfer by conduction along the slab is

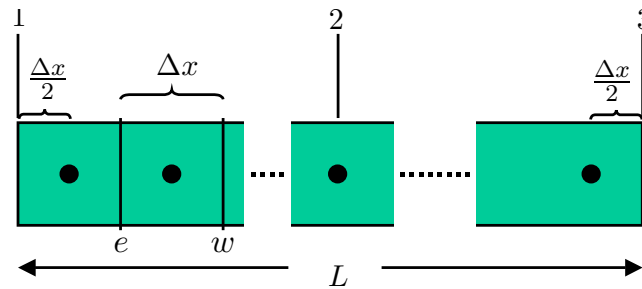


Figure 3.2: One Dimensional Slab or the spatial domain \mathbb{X} for (3.8)

given by the partial differential equation [13]

$$\rho c_p \frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \quad (3.8)$$

The physical parameters are $L = 0.1$ m, thermal conductivity is taken to be

a constant, $\kappa = 100 \text{ W}\cdot\text{m}^{-1}\text{K}^{-1}$, and the multiplication of density ρ and heat capacity c_p is defined as $\rho c_p = 10 \times 10^6 \text{ J}/\text{m}^3\text{K}$.

The temperature T is a function defined over the one-dimensional slab on the spatial domain $\mathbb{X} = [0, L] = [0, 0.1]$ m and over a time interval $\mathbb{T} = [0, t_f]$. For every location in the slab $x \in \mathbb{X}$ and time $t \in \mathbb{T}$ we can associate a temperature $T(x, t)$. The temperature T is therefore a function $T : \mathbb{X} \times \mathbb{T} \rightarrow \mathbb{R}$.

Here, the continuous spatial domain $\mathbb{X} = [0, 0.1]$ is divided into 400 grid cells of equal length. The length of the grid cell is denoted as $\Delta X = 2.5 \times 10^{-4}$ m, see Figure 3.2.

There are 400 grid points associated with the grid cells. The grid points are defined at the following locations:

$$\mathbb{X}_d = \{x_1, \dots, x_{400}\} = \left\{ \frac{\Delta X}{2}, \frac{3\Delta X}{2}, \dots, \frac{799\Delta X}{2} \right\}$$

The one-dimensional heat conduction model is simulated for a period of 240 seconds: $\mathbb{T} = [0, 240]$. The time domain \mathbb{T} is discretized into 120 time steps:

$$\mathbb{T}_d = \{t_1, \dots, t_{120}\} = \{2, \dots, 240\}$$

For notational convenience, since discrete models are calculated, we drop the subscript d from the discretised spatial domain \mathbb{X}_d and the discretised time domain \mathbb{T}_d .

Thus both the discrete spatial and time domain will be denoted as $\mathbb{X} = \{x_1, x_2, \dots, x_{400}\}$ and $\mathbb{T} = \{t_1, t_2, \dots, t_{120}\}$, respectively.

Let $x_{u_1}, x_{u_2}, x_{u_p} \in \mathbb{X}$ define the *locations* of the locations of p actuators. Subsequently set the actuators $u_j(t)$ as the temperatures at the positions x_{u_j} :

$$u_j(t) := T(x_{u_j}, t) \quad j = 1, \dots, p \quad (3.9)$$

The input signals at every time t , $\{u_j(t)\}_{j=1}^p$ can be collected into a vector $\mathbf{u}(t) \in \mathbb{R}^p$:

$$\mathbf{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \dots \\ u_p(t) \end{pmatrix} = \text{col}\{u_i(t)\}_{i=1}^p \quad (3.10)$$

3.3.2 Discretization of one dimensional heat conduction model

The equation (3.8) is discretised by the Finite Volume Method [70],[54] by integrating (3.8) over a specified grid cell volume ΔV and over a specified time step.

For the grid volume ΔV and from t to $t + \Delta t$ we obtain:

$$\int_t^{t+\Delta t} \int_{\Delta V} \rho c_p \frac{\partial T}{\partial t} dV dt = \int_t^{t+\Delta t} \int_{\Delta V} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) dV dt \quad (3.11)$$

The grid volume $\Delta V = \Delta X \times A$ where ΔX is the length of the grid cell and $A = \Delta Y \times \Delta Z$ is the cross-sectional area of the grid cell. For the slab as defined in Figure 3.2, the cross sectional area is 1 cm^2 .

By integrating the right hand side of (3.11) first with respect to ΔV and exchanging the integration order of the term in the left hand side, (3.11) can be rewritten as:

$$\int_w^e \int_t^{t+\Delta t} \rho c_p \frac{\partial T}{\partial t} dt dV = \int_t^{t+\Delta t} \left(\kappa A \frac{\partial T}{\partial x} \right)_e - \left(\kappa A \frac{\partial T}{\partial x} \right)_w dt \quad (3.12)$$

where e and w in (3.12) denote the eastern and western neighboring faces, respectively.

In the Finite Volume Method, the variable at a central grid node is assumed to *prevail over the whole grid cell* [70]. Based on this assumption, the left hand side of (3.12) is written as:

$$\int_w^e \int_t^{t+\Delta t} \rho c_p \frac{\partial T}{\partial t} dt dV = \rho c_p (T(x_P, t_j + \Delta t) - T(x_P, t_j)) \Delta V \quad (3.13)$$

where $T(x_P, t_j)$ denotes the temperature at location x_P and time t_j .

A central differencing scheme ([70],[54]) is applied to the right hand side of (3.12):

$$\begin{aligned} \rho c_p (T(x_P, t_j + \Delta t) - T(x_P, t_j)) \Delta V = & \int_t^{t+\Delta t} \left(\kappa A \frac{T(x_{P+1}, t) - T(x_P, t)}{x_{P+1} - x_P} \right) dt \\ & - \int_t^{t+\Delta t} \left(\kappa A \frac{T(x_{P-1}, t) - T(x_P, t)}{x_{P-1} - x_P} \right) dt \end{aligned} \quad (3.14)$$

The temperatures $T(x_{P+1}, t)$ and $T(x_{P-1}, t)$ are the temperatures at the eastern and western neighboring cells of x_P at time t .

Using the mean-value theorem, the integrands on the right hand side of (3.14) can be calculated as:

$$I_T = \int_t^{t+\Delta t} T(x_P, t) dt = [\theta T(x_P, t_{j+1}) + (1 - \theta)T(x_P, t)] \quad (3.15)$$

for some values of θ between 0 and 1.

Depending on the value of θ , we can discretize a PDE implicitly or explicitly with time, see Table 3.1.

Table 3.1: The value of θ in (3.15) and the discretization scheme

| θ | Discretization Scheme |
|------------------|-----------------------|
| 0 | explicit |
| $0 < \theta < 1$ | semi-implicit |
| $\theta = 1$ | fully-implicit |

CFD models may be obtained by various discretization schemes. Throughout this thesis, we consider a fully implicit discretization scheme. This scheme is often preferred over the explicit scheme because the scheme is recognized to be stable and robust regardless of the size of the time step Δt [70].

By implementing the fully implicit discretization, (3.14) can be written as:

$$\rho c_p \left(\frac{T(x_P, t + \Delta t) - T(x_P, t)}{\Delta t} \right) \Delta X = \left(\kappa \frac{T(x_{P+1}, t + \Delta t) - T(x_P, t + \Delta t)}{x_{P+1} - x_P} \right) - \left(\kappa \frac{T(x_P, t + \Delta t) - T(x_{P-1}, t + \Delta t)}{x_P - x_{P-1}} \right) \quad (3.16)$$

Rearrangement of (3.16) results in:

$$\underbrace{\left(\rho c_p \frac{\Delta x}{\Delta t} + \frac{\kappa}{x_{P+1} - x_P} + \frac{\kappa}{x_P - x_{P-1}} \right)}_{a_P} T(x_P, t + \Delta t) = \underbrace{\rho c_p \frac{\Delta x}{\Delta t}}_{a_P^0} T(x_P, t) + \underbrace{\frac{\kappa}{x_{P+1} - x_P}}_{a_E} T(x_{P+1}, t + \Delta t) + \underbrace{\frac{\kappa}{x_P - x_{P-1}}}_{a_W} T(x_{P-1}, t + \Delta t) \quad (3.17)$$

Rewriting (3.17) in terms of coefficients yields:

$$a_P T(x_P, t + \Delta t) = a_P^0 T(x_P, t) + a_E T(x_{P+1}, t + \Delta t) + a_W T(x_{P-1}, t + \Delta t) \quad (3.18)$$

where E and W denote the eastern and western neighboring cells.

The discrete model (3.18) can be cast into the standard one dimensional CFD-Finite Volume model for every $T(x_P, t + \Delta t)$ [70] as:

$$\begin{aligned} a_P T(x_P, t + \Delta t) &= a_W T(x_W, t) + a_E T(x_E, t) + a_P^0 T(x_P, t) + S_u \mathbf{u}(t) \\ a_P &= a_W + a_E + a_P^0 - S_u \end{aligned} \quad (3.19)$$

where a_W and a_P are the contributions from the western and eastern neighboring grid cells, a_P^0 is the contribution from $T(x_P, t)$. The term S_u denotes the contributions from the neighboring cells if these are the boundary points or the actuators.

Equation (3.19) is general for every grid point. To solve (3.19) model simultaneously for the whole spatial domain \mathbb{X} , we define a vector $\mathbf{T}(t) \in \mathbb{R}^{400}$ as:

$$\mathbf{T}(t) = \begin{pmatrix} T(x_1, t) \\ \dots \\ T(x_{400}, t) \end{pmatrix}.$$

For every P -th grid cell, we can write an expression as in (3.19). We can thus expand (3.19) for the whole spatial domain and obtain the recursion:

$$A\mathbf{T}(t + \Delta t) = A_0\mathbf{T}(t) + B\mathbf{u}(t) \quad (3.20)$$

where A is a tridiagonal matrix with the coefficients a_P in the main diagonal, and the coefficients a_W and a_E in the lower and upper diagonal:

$$A = \begin{pmatrix} a_P & -a_E & 0 & 0 \\ -a_W & a_P & a_E & \dots \\ 0 & -a_W & -a_P & a_E & 0 \\ \dots & \ddots & & & \end{pmatrix} \quad (3.21)$$

The matrix $B \in \mathbb{R}^{400 \times p}$ is a collection of the row vectors $S_u \in \mathbb{R}^{1 \times p}$ where p is the number of actuators:

$$B = \begin{pmatrix} S_{u_1} \\ S_{u_2} \\ \dots \\ S_{u_{400}} \end{pmatrix}$$

For ease of notation, we replace the time t and $t + \Delta t$ in (3.20) with time step k and time step $k + 1$ respectively. The final discrete CFD model for the one dimensional heat equation is thus:

$$A\mathbf{T}(k + 1) = A_0\mathbf{T}(k) + B\mathbf{u}(k) \quad (3.22)$$

Some numerical methods commonly applied in CFD packages to find the unknowns $\mathbf{T}(k + 1)$ in (3.22) are the LU-decomposition, Stone's Implicit LU Decomposition, Tri-Diagonal Matrix Algorithm (TDMA). The matrices A and A_0 are sparse (there are many zeros in the matrix coefficients) and most of the numerical solutions employed to solve (3.22) exploit the sparsity of the matrices to obtain the solutions efficiently.

If the investigated problem is nonlinear, for example with a temperature-dependent heat conductivity, then a similar derivation shows that the coefficients in (3.19) become time dependent. In that case, the model (3.22) becomes:

$$\mathbf{A}(k)\mathbf{T}(k + 1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}(k)\mathbf{u}(k)$$

and the solver module of the CFD packages also updates the coefficients in the matrices in (3.22). In this chapter, we focus on linear CFD model first, where all physical constants are not temperature dependent.

3.4 *Reduced modeling of one dimensional heat conduction model*

Previously, we discussed the derivation of a discrete CFD Finite Volume model from a PDE. The derivation is based on a linear one dimensional heat conduction model. The same principles can be applied to more complex systems. After integration over every grid cell and specified time step, discretized mass, momentum, and energy conservation equations in CFD models will have the general form as in (3.22).

In this thesis, we use Proper Orthogonal Decomposition (POD) as the model reduction method. As described in Chapter 2, if the function $T : \mathbb{X} \times \mathbb{T} \rightarrow \mathbb{R}$ is an element in the Hilbert space, we can expand $T(x, t)$ with $x \in \mathbb{X}$ and $t \in \mathbb{T}$ as:

$$T(x, t) = \sum_{i \in \mathbb{I}} a_i(t) \varphi_i(x) \quad (3.23)$$

where $\{\varphi_i\}_{i \in \mathbb{I}}$ is the POD basis and \mathbb{I} is the countable index set, which may run until infinity. For discrete models, \mathbb{X} and \mathbb{T} are discrete with finite cardinality, hence the index set \mathbb{I} is also finite. The POD basis functions $\varphi_i(x)$ are often also referred to as POD basis vectors if \mathbb{X} has finite cardinality.

In the reduced order models, we approximate $T(x, t)$ by the expansion up to the n -th POD basis function:

$$T(x, t) \approx T_n(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x) \quad (3.24)$$

We then replace $T(x, t)$ in the original equations by $T_n(x, t)$ and perform the Galerkin projection (section 2.6) to obtain a reduced order model which gives the dynamics of $\{a_i(t)\}_{i=1}^n$.

There exist several ways of deriving a reduced order model for $a_i(t)$:

1. Perform the Galerkin projection of the POD basis functions $\{\varphi_i\}_{i \in \mathbb{I}}$ directly onto the discrete model, e.g discrete heat conduction model (3.22)
2. Perform the Galerkin projection on the continuous model, derive $a_i(t)$ as a continuous ordinary differential equation (ODE) model and simulate the ODE based model by a numerical ODE solver. For example this approach is conducted in the reduced order model of the wave propagation equation (Section 2.5).
3. Derive a black-box model based on system identification techniques to derive the models of $\{a_i(t)\}$. This approach is presented in [31].

Since the POD basis is an orthonormal basis (refer to section 2.4 for more details), the POD coefficients $\{a_i(t)\}$ in finite dimensional case can be expressed as:

$$a_i(t) = \varphi_i^\top \mathbf{T}(t)$$

where $\mathbf{T}(t) \in \mathbb{R}^K$ is the collection of data at K points at the time t .

In the black box modeling as proposed in [31], the collections of the POD coefficients $\{a_i(t)\}$ can be treated as "outputs" and the input signals which excite the dynamics of $\mathbf{T}(t)$ are considered as inputs. The model for $\{a_i(t)\}$ is then derived by system identification techniques based on the input-output relationships between the POD coefficients and the input signals.

The third approach can be used as an alternative if there is considerable lack of knowledge about the original equations. The disadvantage of this approach is that the dynamics of $\{a_i(t)\}$ is not derived from the original equations which represent the physical behavior of the system. The second approach is only feasible when the original model is simple, for example if the analytical solutions of the original equations are available. For nonlinear PDE's, especially when flow fields have to be solved, there exist no analytical solutions. Some more advanced discretization and numerical schemes have to be employed as well to solve such systems. In this thesis, we have access to the discretized equations and we reduce the CFD model by conducting a Galerkin projection of the POD basis functions directly on the discrete CFD models.

3.4.1 Derivation of POD-based reduced model for one dimensional heat conduction problem

The discrete heat conduction model has the general recursive equations as:

$$A\mathbf{T}(k+1) = A_0\mathbf{T}(k) + B\mathbf{u}(k) \quad (3.25)$$

where $\mathbf{T}(k) \in \mathbb{R}^K$ with K is the number of grid points, $A \in \mathbb{R}^{K \times K}$, $A_0 \in \mathbb{R}^{K \times K}$, and $B \in \mathbb{R}^{K \times p}$ where p is the number of actuators.

We can approximate the solution of (3.22) at $x \in \mathbb{X}$ and at the k -th time step as in (3.24). We can thus approximate $\mathbf{T}(k)$ by $\mathbf{T}_n(k)$ as:

$$\mathbf{T}(k) \approx \mathbf{T}_n = \Phi\mathbf{a}(k) \quad (3.26)$$

where $\mathbf{a}(k) = \text{col}\{a_i(k)\}_{i=1}^n$ is the vector of POD coefficients at time step k . The POD basis is collected in the matrix $\Phi \in \mathbb{R}^{K \times n}$ where K is the number of the grid points and n the number of POD basis functions. The POD basis is given as:

$$\Phi = (\varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_n).$$

Replacing $\mathbf{T}(k+1)$ by $\mathbf{T}_n(k+1)$ in (3.25) results in the following model:

$$A\Phi\mathbf{a}(k+1) = A_0\Phi\mathbf{a}(k) + B\mathbf{u}(k) \quad (3.27)$$

The reduced order CFD model is obtained by performing a Galerkin projection of the (3.27) on the POD basis Φ :

$$\underbrace{\Phi^T A \Phi}_{A_r} \mathbf{a}(k+1) = \underbrace{\Phi^T A_0 \Phi}_{A_{0_r}} \mathbf{a}(k) + \underbrace{\Phi^T B}_{B_r} \mathbf{u}(k)$$

$$A_r \mathbf{a}(k+1) = A_{0_r} \mathbf{a}(k+1) + B_r \mathbf{u}(k) \quad (3.28)$$

The equation (3.28) is the reduced order CFD model. The original model has dimension K , thus $A, A_0 \in \mathbb{R}^{K \times K}$, $B \in \mathbb{R}^K \times p$ with K is the number of grid cells used to find a solution and p the number of actuators. The reduced order has dimension $n \ll K$, thus $A_r, A_{0,r} \in \mathbb{R}^{n \times n}$, $B_r \in \mathbb{R}^{n \times p}$. The original model has sparse matrix structures in A and A_0 , while the reduced order model has dense matrix structures in A_r and $A_{0,r}$.

The POD technique will be applied to the one dimensional heat conduction model with the spatial domain as depicted in Figure 3.2 for two cases first:

- Heating the slab at the left end
The actuator for heating at the left end of the slab is denoted by $u_1(t)$:

$$u_1(t) = T(x_{u_1}, t) = T(0, t)$$

- Heating the slab in the middle
The actuator for heating at the middle is denoted by $u_2(t)$:

$$u_2(t) = T(x_{u_2}, t) = T(0.05, t)$$

The initial temperature condition at the grid points is uniform at 200°C:

$$T(x, 0) = 200, x \in \mathbb{X}. \quad (3.29)$$

3.4.2 Heating at the left end

The initial temperature is uniform along the slab \mathbb{X} at 200°C and the slab is heated at the left end, so $u_1(t) = T(0, t)$. The slab is heated at constant temperature of 400°C at the left end. The number of actuator in this case is $p = 1$.

The boundary conditions for (3.8) then read:

$$T(x, 0) = 200, \quad x \in \mathbb{X}, \quad T(x_{u_1}, t) = T(0, t) = 400, \quad t \in \mathbb{T} \quad (3.30)$$

The right-end of the slab *in this experiment* is insulated:

$$\frac{\partial T}{\partial x}(0.1, t) = 0; \quad t \in \mathbb{T} \quad (3.31)$$

The one-dimensional heat conduction model (3.8) with boundary condition (3.30),(3.31) is discretized. The discrete model has the form as in (3.22). The model (3.22) is simulated for 120 time steps of 2 seconds each.

Algorithm 2.4.10 is applied to derive the POD basis from the simulation data of the one dimensional heat conduction case with heating at the left end. The procedure is as follows:

1. Calculate and collect the simulation data in a matrix \mathbf{T}_{snap} :

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \quad \mathbf{T}(2) \quad \dots \quad \mathbf{T}(120))$$

where the vector $\mathbf{T}(t) \in \mathbb{R}^{400}$ is the collection of the temperature data (snapshots) for the whole spatial domain \mathbb{X} at time t . The snapshots collected from heating at the left end are depicted in Figure 3.3. The slab is subject to temperature increase of 400°C at the left end. The temperatures at the left part of the slab increase fast.

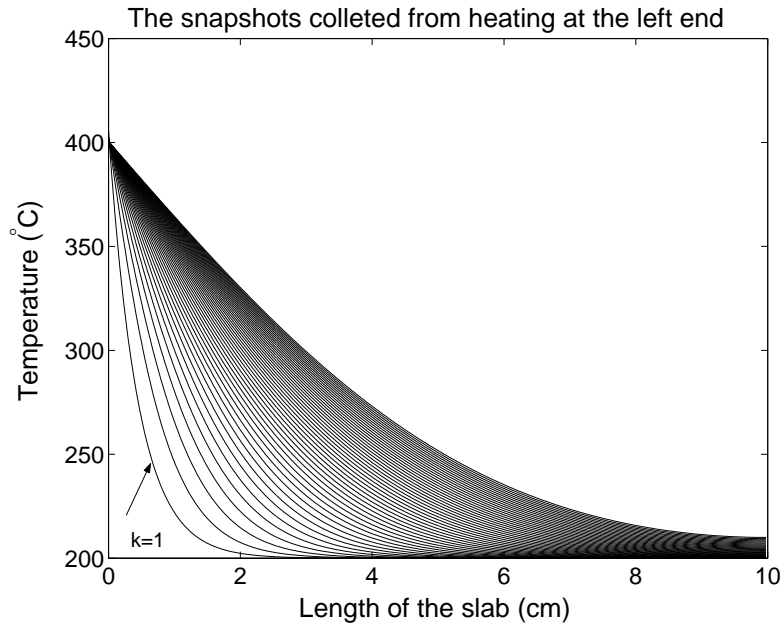


Figure 3.3: The snapshots of temperatures in the grid cells collected from heating at the left end

2. Construct the symmetric correlation matrix $\mathbf{C} \in \mathbb{R}^{400 \times 400}$ (see Chapter

2)

$$\mathbf{C} = \frac{1}{120} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top$$

3. Conduct the eigenvalue decomposition for \mathbf{C} :

$$\mathbf{C}\Phi = \Phi\Lambda$$

where

$$\Phi = (\varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_{400})$$

is orthogonal and

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{120})$$

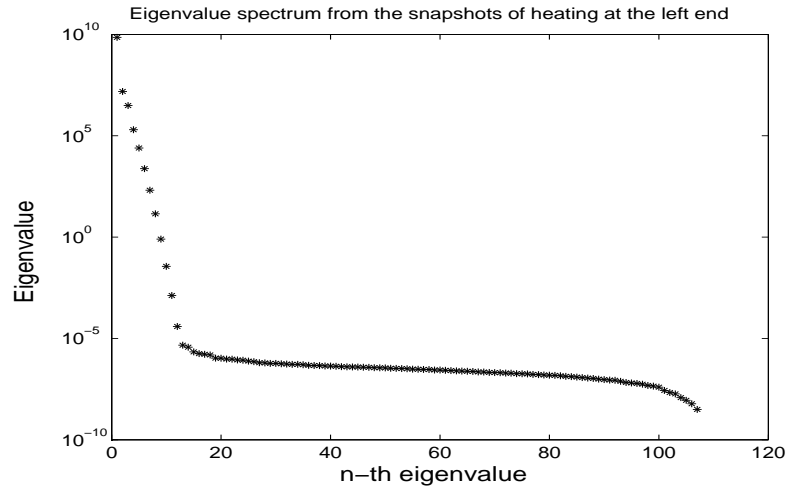
The eigenvalues are ordered: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{120}$.The plot of the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{120}$ is shown in Figure 3.4. The plot as presented Figure 3.4 is also known as the *eigenvalue spectrum*.

Figure 3.4: The plot of the eigenvalue resulted from the eigenvalue decomposition of the correlation matrix \mathbf{C}

4. Truncate the POD basis functions which correspond to the first 6 eigenvalues $\{\lambda_i\}_{i=1}^6$. Hence, set

$$\Phi = (\varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_6)$$

The choice of truncation degree $n = 6$ is based on the value of P_n defined by:

$$P_n = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^{120} \lambda_i}$$

The closer P_n to 1, the better the approximation of \mathbf{T}_{snap} will be. The logarithmic plot of $1 - P_n$ for $n = 1, \dots, 34$ is given in Figure 3.5. For $n \geq 6$ is taken, $\log(1 - P_n)$ is in the order of 10^{-8} , so P_n is close enough to 1. The first POD basis function φ_1 corresponds to the largest eigenvalue

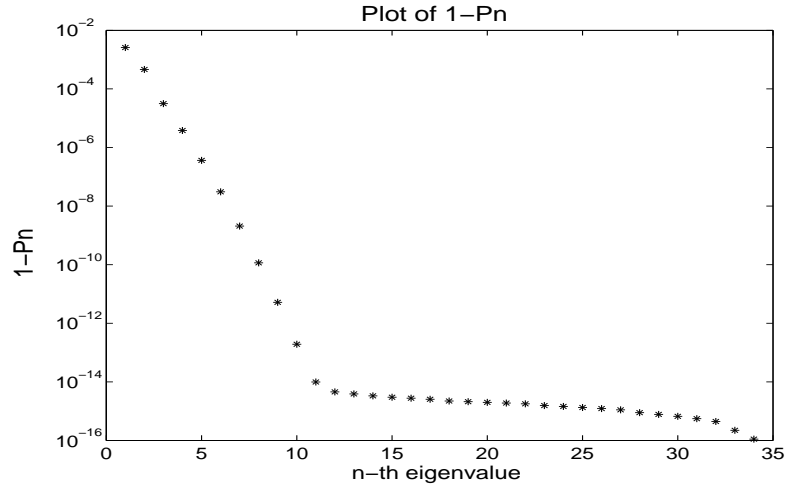


Figure 3.5: The logarithmic plot of $1 - P_n$, with n is the number of eigenvalues taken

λ_1 and reveals the main pattern of the collected snapshots. The first POD basis function φ_1 is depicted in Figure 3.6. The collected snapshots are shown in Figure 3.3. By comparing Figure 3.3 and the resulted first POD basis function in Figure 3.6, it is obvious that φ_1 reveals the most dominant spatial dynamics of \mathbf{T}_{snap} .

3.4.3 Heating in the middle

As in the previous case, the initial temperature is uniform along the slab \mathbb{X} at 200° . The slab is now heated by the actuator in the middle only: $u_2(t) = T(0.05, t)$. The number of actuator in this experiment is also one: $p = 1$.

The boundary condition for (3.8) reads:

$$T(x, 0) = 200, \quad x \in \mathbb{X}, \quad T(x_{u_2}, 0) = T\left(\frac{L}{2}, t\right) = 400, \quad t \in \mathbb{T} \quad (3.32)$$

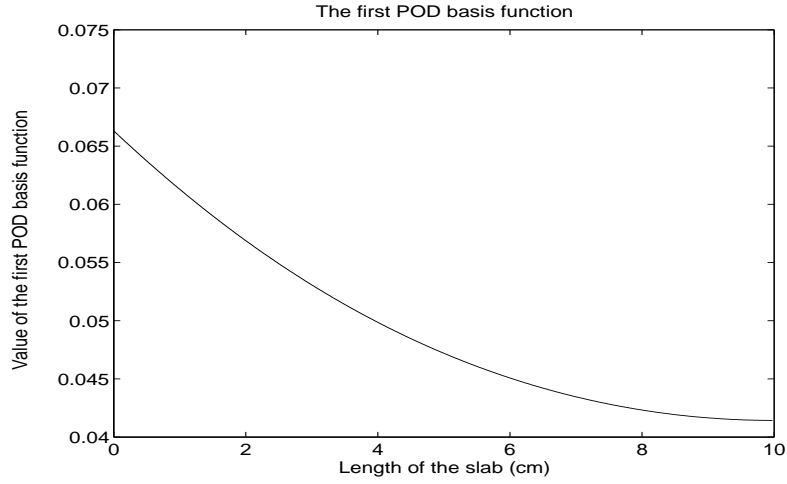


Figure 3.6: The first POD basis function which corresponds to the largest eigenvalue derived from the snapshots generated by heating at the left end

In this experiment, the left end and the right end of the slab are insulated:

$$\frac{\partial T}{\partial x}(0, t) = \frac{\partial T}{\partial x}(0.1, t) = 0, \quad t \in \mathbb{T} \quad (3.33)$$

With the boundary condition as in (3.32), (3.33), the discretized model by setting the boundary temperatures $T(0, t)$ and $T(0.1, t)$ as:

$$T(0, t) = T(x_1, t) \quad T(0.1, t) = T(x_{400}, t)$$

since x_1 and x_{400} are the closest grid point locations to the $x = 0$ and $x = 0.1$.

The one-dimensional heat conduction model (3.8) with boundary conditions (3.31) and (3.32) is discretized and simulated for 120 time steps. The discrete model is constructed by employing the Finite Volume method.

Algorithm 2.4.10 is again implemented to derive the POD basis $\{\varphi_i\}$ for this experiment. The temperature data for 120 time steps are collected in $\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \ \mathbf{T}(2) \ \dots \ \mathbf{T}(120))$. Figure 3.7 shows the collected snapshots for this case. Figure 3.8 shows the eigenvalue spectrum (plot of eigenvalue) resulting from the eigenvalue decomposition of \mathbf{C} . The plot in Figure 3.8 is only for the nonzero eigenvalues. The logarithmic plot of $1 - P_n$ with P_n as formulated in (2.37) is shown in Figure 3.4.3. As in the case of heating at the left end, the truncation degree in the case of heating at the middle is taken to be $n = 6$. This corresponds to $\log(1 - P_n)$ in the order of 10^{-10} . The first POD basis

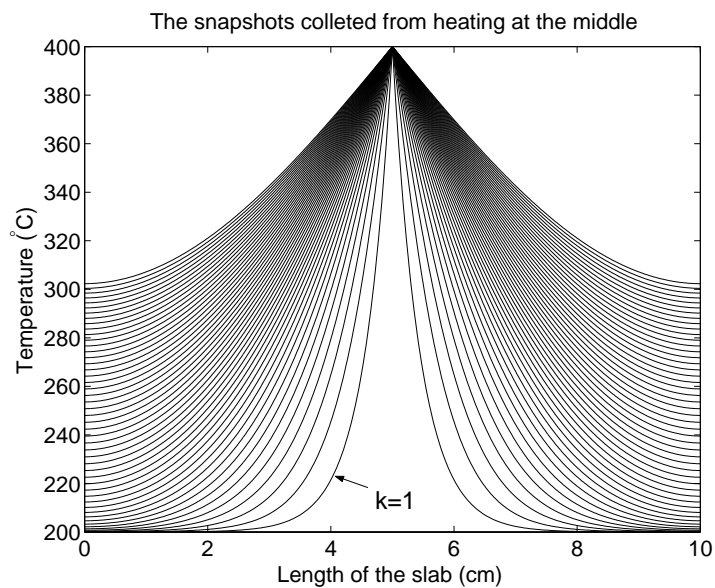


Figure 3.7: The snapshots from the experiment of heating at the middle

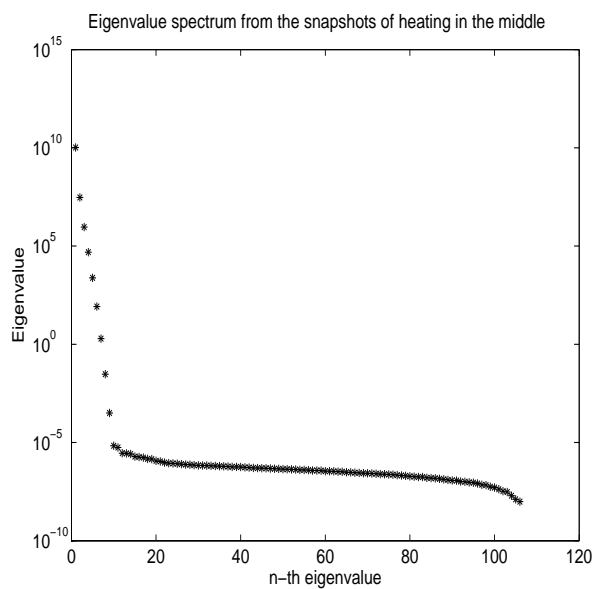


Figure 3.8: The plot of the nonzero eigenvalues resulted from the eigenvalue decomposition of the snapshot data collected from heating at the middle

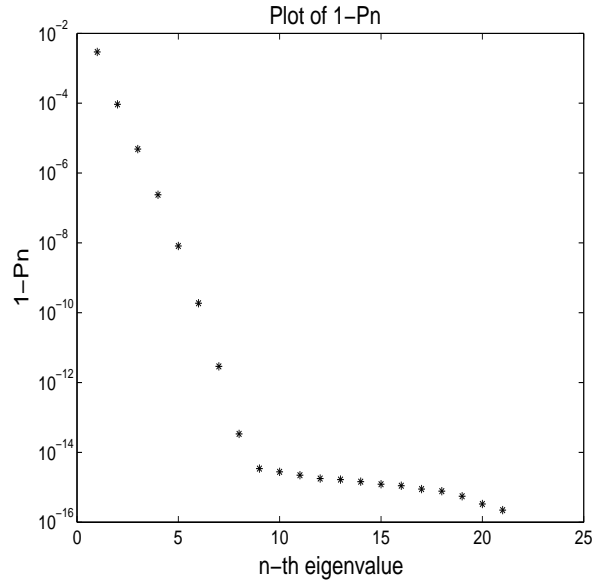


Figure 3.9: The plot of $1 - P_n$ where n is the number of eigenvalues. The closer P_n to 1, the better the approximation by POD basis Φ will be

function also reveals the most dominant spatial characteristic of the data in \mathbf{T}_{snap} . The first POD basis function derived from heating at the middle is shown in Figure 3.10.

Comparison of Figure 3.10 with Figure 3.7 clearly shows how the main patterns of the snapshots in Figure 3.7 is captured in Figure 3.10.

3.4.4 Development and validation of the reduced order model

For each heating experiment, a POD basis of 6 basis functions has been derived. Denote the collection of POD basis functions derived from the first experiment as the matrix Φ_1 and the POD basis functions derived from the second experiment as the matrix $\Phi_2 \in \mathbb{R}^{400 \times 6}$. By applying Galerkin projection on these two bases, we obtain two reduced order models.

In line with (3.28), this leads to two reduced order models for heating at the left end. One reduced model M_1 is constructed from POD basis Φ_1 from the same experiment while the other one M_2 is derived by the POD basis Φ_2 . Both M_1 and M_2 belong to the same model class \mathcal{M} : discrete implicit state

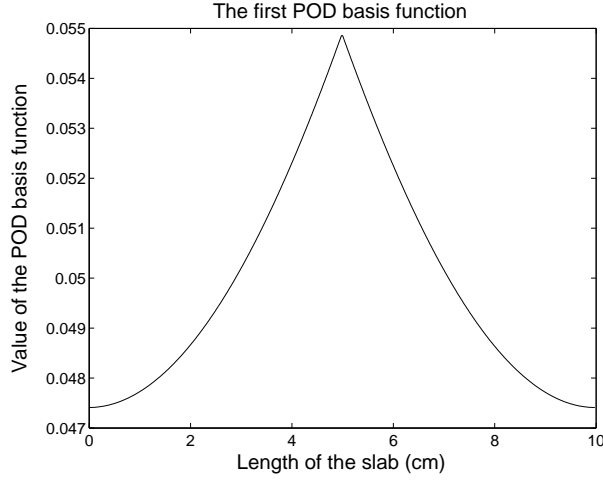


Figure 3.10: The first POD basis function derived from the snapshots obtained by heating the slab at the middle. The first POD basis function clearly shows the main patterns of the dynamics

space models. Specifically:

- Reduced model M_1 :

$$\Phi_1^\top A \Phi_1 \mathbf{a}_1(k+1) = \Phi_1^\top A_0 \mathbf{a}_1(k) + \Phi_1^\top B u_1(k)$$

- Reduced model M_2 :

$$\Phi_2^\top A \Phi_2 \mathbf{a}_2(k+1) = \Phi_2^\top A_0 \mathbf{a}_2(k) + \Phi_2^\top B u_1(k)$$

The same boundary conditions are applied, so $\mathbf{T}(0) = 200^\circ$ and $u_1(k) = 400^\circ C$ for all time. From the initial boundary condition, for both reduced model M_1 and M_2 , the initial POD coefficients can be derived:

$$\mathbf{a}_1(0) = \Phi_1^\top \mathbf{T}(0), \quad \mathbf{a}_2(0) = \Phi_2^\top \mathbf{T}(0) \quad (3.34)$$

Both reduced order models M_1 and M_2 are simulated with initial conditions as defined in (3.34). The POD coefficients $\mathbf{a}_1(k)$ and $\mathbf{a}_2(k)$ at every k -th time step are calculated by the reduced order models M_1 and M_2 .

The temperature fields reconstructed by the reduced order models are given by:

$$\mathbf{T}_{n_1}(k) = \Phi_1^\top \mathbf{a}_1(k) \quad (3.35)$$

$$\mathbf{T}_{n_2}(k) = \Phi_2^\top \mathbf{a}_2(k) \quad (3.36)$$

where \mathbf{T}_{n_1} and \mathbf{T}_{n_2} denote the temperatures reconstructed by the reduced order model M_1 and M_2 , respectively.

Figure 3.11 shows the comparison between the temperatures reconstructed by the reduced order model M_1 and M_2 and the original one.

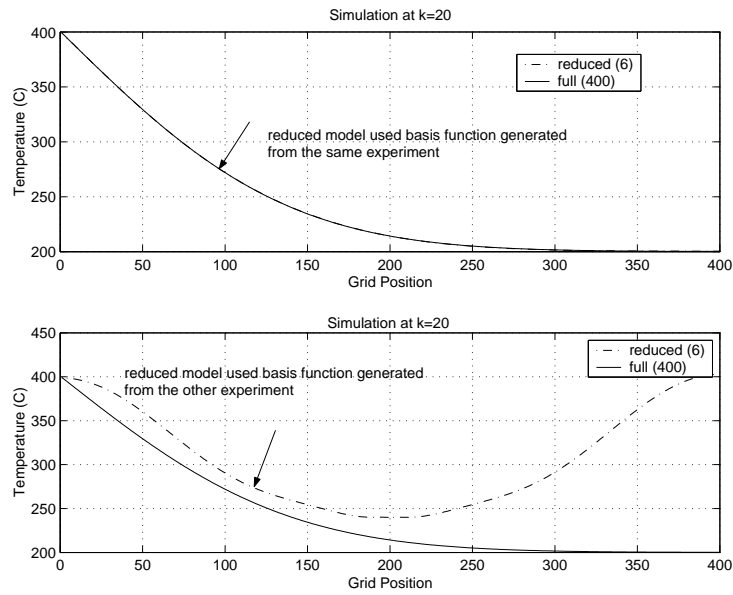


Figure 3.11: The comparison between the reduced order model and the original model describing heating at the left end. Reduced order model M_1 and the original one is compared in the top part. The reduced order model M_2 is compared with the original one at the bottom figure. Both reduced order models are built from 6 POD basis functions

Figure 3.11 shows that the reduced order model M_1 is in very good agreement with the original model because the POD basis Φ_1 used in the reduced order model was derived from the same condition as the validation model.

The bottom figure of Figure 3.11 shows that the reduced order model M_2 fails to approximate the original model well because the POD basis was derived for a completely different case. The reduced order model at the bottom of Figure 3.11 was derived from the simulation data heating at the middle. The POD basis Φ_2 only reveals the dynamics of the spatial variations when the slab is heated at the middle.

By comparing the first POD basis function from the first experiment (Figure

3.6) and the first POD basis function from the second experiment (Figure 3.10), it is obvious that these two experiments are completely different from each other. The two experiments were conducted by excitations at different locations. We conclude that Φ_1 cannot approximate data from the second experiment and Φ_2 cannot approximate data from the first experiment as well.

It is therefore important to excite all actuators located in different places and include these excitations in the snapshot matrix \mathbf{T}_{snap} . A practical approach would be to use the knowledge of the typical excitation signals which are implemented in the real systems and collect snapshots based on these input signals. This pragmatic approach is taken because the POD basis is derived from data. A reduced order model constructed from a POD basis would be reliable when it is used to simulate the conditions in the same operating range as the one used to generate the snapshot data.

In the following situations, snapshots are collected for this heat conduction processes for three situations: individual actuator excitations and the combinations. The model is linear that superposition principle apply: the response due to the excitation of two actuators equal the sum of the response of individual actuators. Snapshots are collected for 240 seconds or 120 time steps ($\Delta t = 2$ s) for each experiment.

The same step-by-step procedure for deriving POD basis is conducted:

1. Collection of snapshots from four experiments in $\mathbf{T}^{\text{snap}} \in \mathbb{R}^{400 \times 480}$:

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}_{\text{snap}_1} \quad \mathbf{T}_{\text{snap}_2} \quad \mathbf{T}_{\text{snap}_3})$$

where $\mathbf{T}_{\text{snap}_j}$ indicates the snapshot matrix from the j -th experiment: heating at the left end, middle and right end

2. Perform eigenvalue decomposition of the correlation matrix $\mathbf{C} = \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$:

$$\mathbf{C}\Phi = \Phi\Lambda$$

where Φ is the POD basis with POD basis vectors φ_i in its columns. The matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{400})$ with $\lambda_1 \leq \dots \leq \lambda_{400}$. The eigenvalue spectrum of the first 50 eigenvalues is shown in Figure 3.12

3. The truncation up to the first n POD basis function is based on the value of $1 - P_n$ where P_n is formulated as:

$$P_n = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^{720} \lambda_i}$$

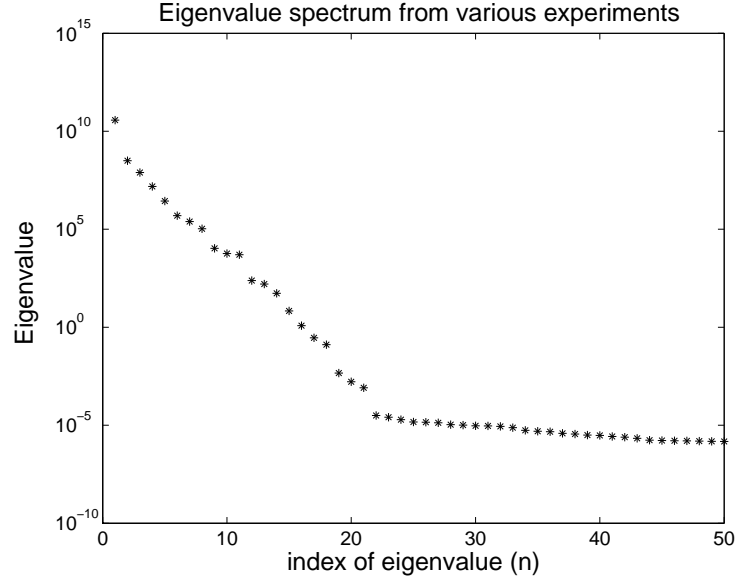


Figure 3.12: The eigenvalue spectrum obtained from snapshots data constructed from various experiments

The logarithmic plot of $1 - P_n$ is shown in Figure 3.13. Ten POD basis functions are chosen to construct the reduced order model. Ten POD basis functions correspond to $(1 - P_{10}) \approx 10^{-7}$. The POD basis functions corresponding to the first four largest eigenvalues are plotted in Figure 3.14. The first POD basis function reveals the main pattern or main behavior of the four experiments.

4. Construction of the reduced order model

The POD basis Φ derived from various experiments is validated by projecting the POD basis Φ onto the model describing simultaneous heating at the left and the right end. The reduced order model has the form:

$$\Phi^\top A \Phi \mathbf{a}(k+1) = \Phi^\top A_0 \Phi \mathbf{a}(k) + \Phi^\top B \mathbf{u}(k) \quad (3.37)$$

where A, A_0, B are the matrices obtained from discretizing (3.8) by the Finite Volume Method.

The input vector $\mathbf{u}(k) \in \mathbb{R}^2 = \begin{pmatrix} u_1(k) \\ u_2(k) \end{pmatrix}$ where:

$$u_1(k) = T(0, k); \quad u_2(k) = T(0.1, k)$$

The temperatures $T(0, k)$ and $T(0.1, k)$ is the temperature at $x = 0$ m and $x = 0.1$ m.

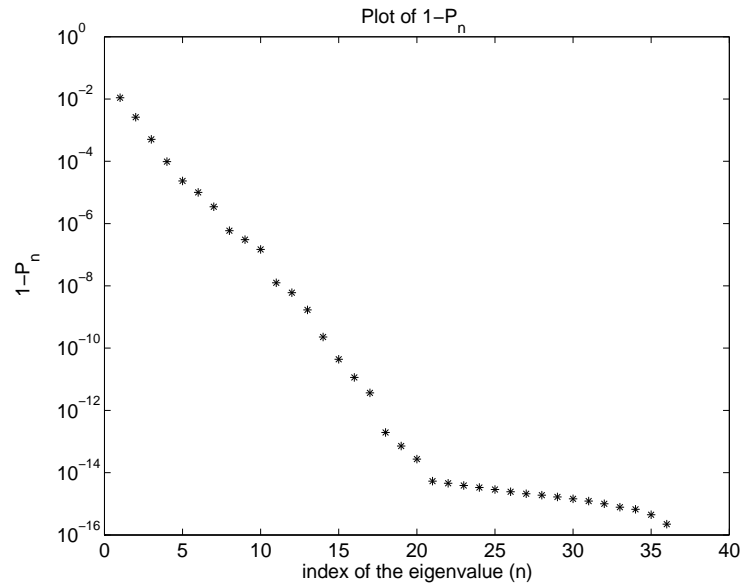


Figure 3.13: The plot of $1 - P_n$, the smaller $1 - P_n$ is, n is the number of eigenvalues taken. The closer P_n to 1, the better the approximation by n POD basis functions will be

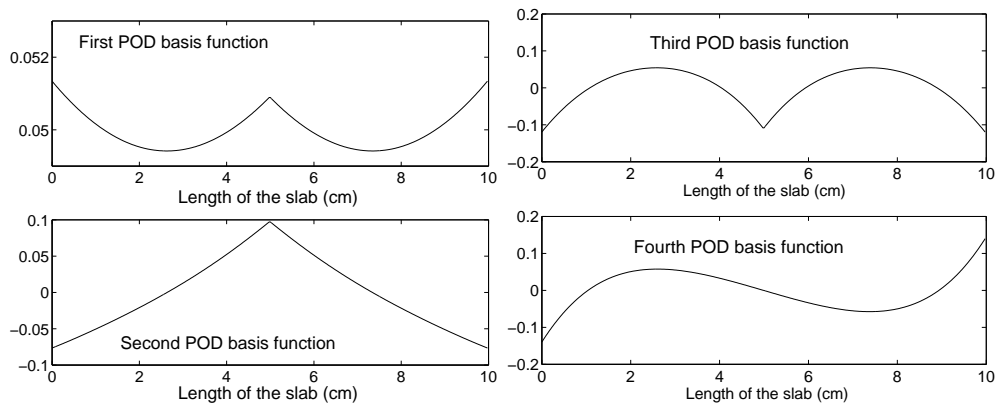


Figure 3.14: The plot of the first four POD basis functions correspond to the four largest eigenvalues

The comparisons between the reduced order model of order 10 and the original model of order 400 are shown in Figure 3.15 and Figure 3.16. The comparisons are shown for the temperature profile at time step $k = 20$ or at $t = 40$ seconds and at time step $k = 60$ or at $t = 2$ minutes. It can be seen that the reduced

order model follows the dynamics of the original model quite well in these two time instants.

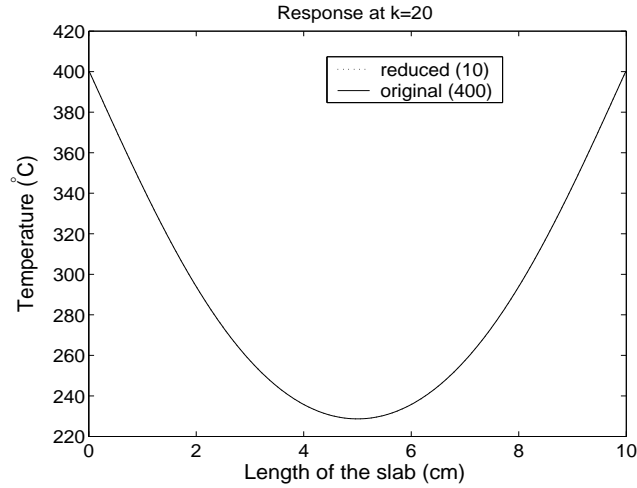


Figure 3.15: The reduced and the original model at timestep $k = 60$

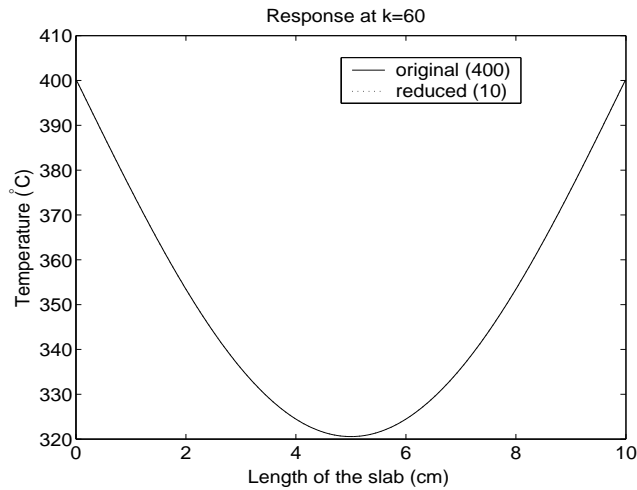


Figure 3.16: The reduced and the original model at timestep $k = 60$

The plot of the average absolute error $\bar{\epsilon}$ for this validation data is shown in Figure 3.17. The absolute error average $\bar{\epsilon}$ is calculated as:

$$\bar{\epsilon} = \frac{1}{200} \sum_{k=1}^{200} | \mathbf{T}(k) - \mathbf{T}_{10}(k) | \quad (3.38)$$

where $\mathbf{T}_{10}(k)$ is the approximate temperature obtained from the reduced order model constructed by first 10 POD basis functions at time step k . From

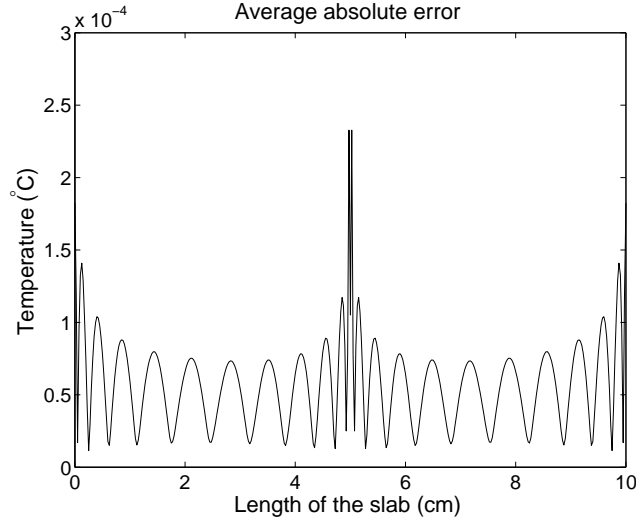


Figure 3.17: The plot of mean absolute error as defined in (3.38)

Figure 3.17, it is concluded that the reduced order model simulates the original dynamics very closely, with only maximum average absolute error of $3 \times 10^{-4} \text{ } ^\circ\text{C}$.

As discussed in Chapter 2, if the snapshot data is approximated by the POD basis functions $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_{10})$, then POD coefficients are ordered according to:

$$\langle a_1^2(k) \rangle \geq \langle a_2^2(k) \rangle \geq \dots \geq \langle a_{10}^2(k) \rangle \quad (3.39)$$

where $\langle a_i(k) \rangle$ is the mean of the i -th POD coefficient defined in this case:

$$\langle a_i(k) \rangle = \frac{1}{360} \sum_{k=1}^{360} a_i(k)$$

If the POD basis is used to validate another situation which is not the exactly the same as the condition used to generate \mathbf{T}_{snap} , the ordering in (3.39) can be different.

This fact is also shown in this validation example. Figure 3.18 shows the comparison between the first five averaged values of POD coefficients $\langle a_i(k) \rangle_{i=1}^5$ in the approximation of the snapshot data and the POD coefficients obtained

from the calculation of the reduced order model when simulating another condition.

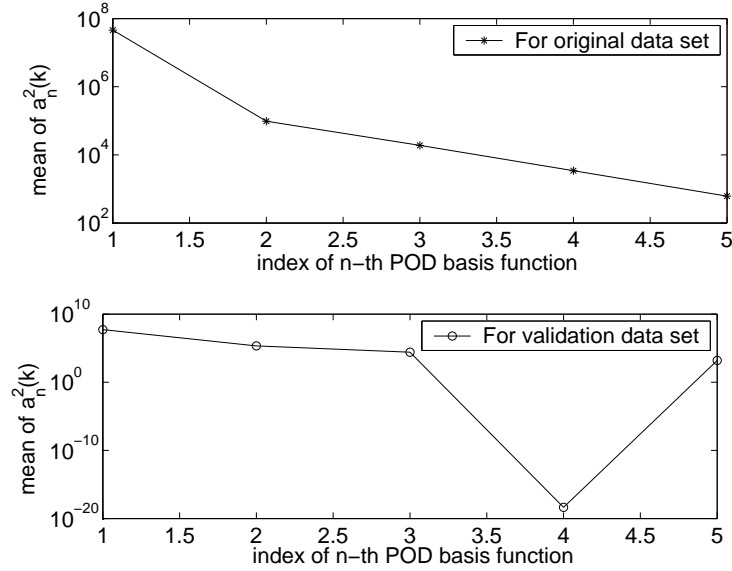


Figure 3.18: The comparisons of $\langle a_i(k) \rangle_{i=1}^5$. The validation data is different than the snapshot data, therefore the ordering of the POD coefficients is also different

The ordering is different in the validation data set as now $\langle a_4(k) \rangle < \langle a_5(k) \rangle$. However, since the new condition is still within the operating range of the data collected in \mathbf{T}_{snap} , the POD basis functions used to construct the reduced order model are still valid.

Even though in the original data set there are no simultaneous excitations of two actuators, the simultaneous excitations can be described by the individual excitations of the actuators at the left and the right end which are already captured in \mathbf{T}_{snap} . For all experiments, the initial conditions are uniform temperature distribution at 200°C subject to the actuator's excitations at 400°C .

Until now the validation of the reduced order models for different excitation locations has been discussed. As we have seen, different configurations will give different temperature distributions along the slab or different spatial dynamics. By spatial dynamics we mean the change of the variable *distribution* along the spatial domain. The spatial dynamics is distinguished from the *temporal*

dynamics. Temporal dynamics is the change of each process state (such as each grid cells) with respect to time.

The POD basis approximates the spatial dynamics and the POD coefficients approximate the temporal dynamics. We have seen how crucial it is to take the excitations at various possible locations into account . If the excitations at various locations are not incorporated in the snapshot data, then the new spatial dynamics will not be approximated well by the POD basis.

The next question is, what the effect of the variations of the input signals with respect to time will be. In other words, do variations of the excitation signal with respect to time affect the distribution of the process variable (the spatial dynamics) such as temperature in the spatial domain \mathbb{X} .

In the following example, the effect of the different time-varying excitation signals will be investigated.

Example 3.4.1 The POD basis is derived by step excitation signals (at different locations) collected in \mathbf{T}_{snap} . We will validate the reduced order model using sinusoidal excitation signals for the actuators at the left and the right end. The sinusoidal variation is shown in Figure 3.19.

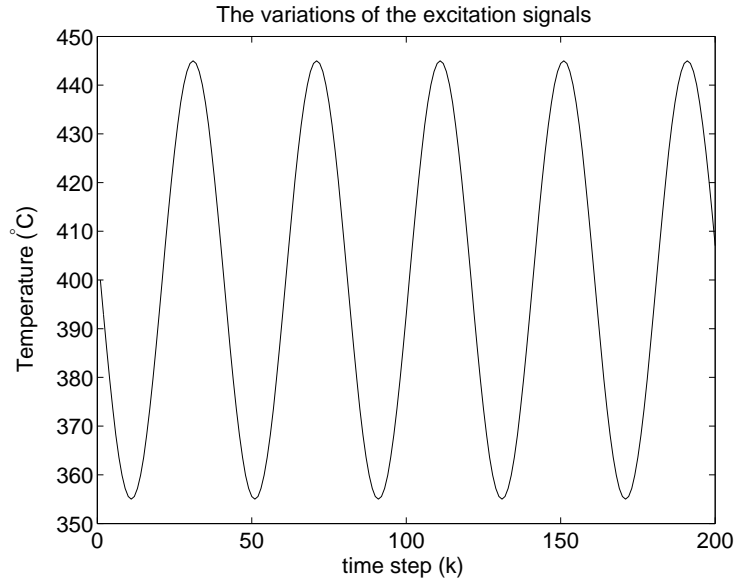


Figure 3.19: The sinusoidal excitation signal, applied at the left and the right end actuator of the slab

The comparison between the original and the reduced order model in describing the condition of heating at both the left and right end with sinusoidal excitation signals are shown in Figure 3.20.

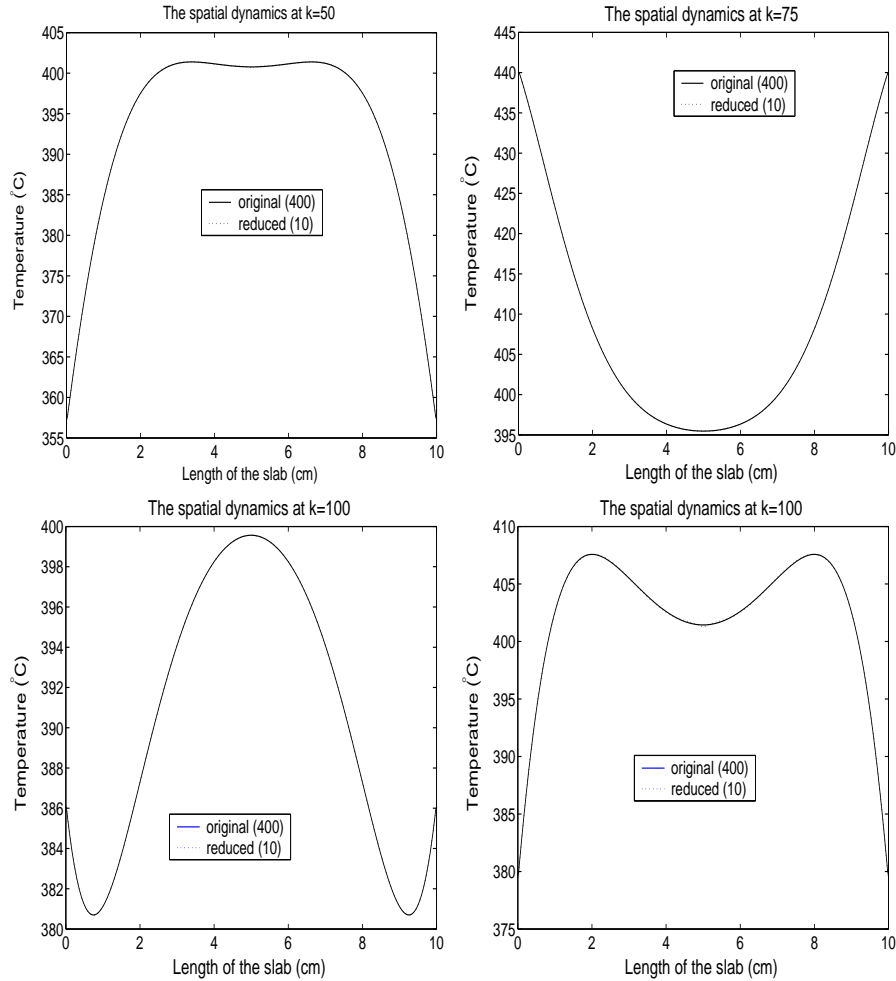


Figure 3.20: The comparisons between the original and the reduced order model, sinusoidal excitation signals are applied

From Figure 3.20, it is obvious that the reduced order model can still follow the original dynamics very well despite the fact that the slab is excited by different type of excitation signals. As obvious from Figure 3.20, the temperature distribution is not affected substantially by the different variations of the excitation signals with respect to time. The excitation signals are varying in sinusoids as shown in Figure 3.19, but the temperature distribution along the

slab at each time instant remains basically the same as in the case when the excitation signals are constant step signals.

From the example, it can be concluded that for the case of a linear heat conduction process, variations in the temporal dynamics are not as crucial as variations in the spatial boundary conditions. The POD basis will be derived from the distribution of the process variables along the spatial domain. Therefore it is more important to incorporate the variations of the spatial boundary conditions (such as excitations at different *locations*) rather than to vary the excitation signals with respect to time.

In the case of nonlinear PDE's, such as a temperature-dependent heat conductivity $\kappa(T)$, then the procedure of collecting snapshot data is more ad-hoc. Spatial variations will interact with the temporal variations as well, that the time-varying behavior of each actuator will affect the distribution of the process variable in the spatial domain. The variations of the excitation signals with respect to time will affect the variations of physical parameters such as heat conductivity with time. Different heat conductivity distribution will give different temperature distributions along the spatial domain. Thus in nonlinear case, there is no guarantee that the change of the temporal dynamics will not affect the quality of the reduced order model substantially.

As mentioned in [67], it will be very difficult to determine the dependency of nonlinear PDE solutions to the physical parameters. The snapshot data for nonlinear model is thus at best chosen to be the data generated from typical operating condition of the system.

3.5 Two dimensional heat conduction model

This section discusses the reduced order modeling of a two dimensional heat conduction model. The sketch of a two dimensional plate is given in Figure 3.21. The dimension of the plate is 0.3m \times 0.4m in length (x) and height (y), respectively. The defines a spatial domain is thus $\mathbb{X} = [0, 0.3] \times [0, 0.4]$. The temperature along the north side is kept at 100° C and there are four actuators at the west side which gives incoming heat fluxes of 500 (W/m²). The eastern and the southern boundaries are insulated.

Thermal conductivity of the plate material is given as $\kappa = 1000$ W/mK. The thickness of the plate is 1m. The computational domain is divided into 44 grid cells along the y direction and 33 grid cells along the x direction. Thus

in total, there are 1452 grid points. The grid size in the horizontal direction x and vertical direction y is $\Delta X = \Delta Y = 0.0091\text{m}$. The time step Δt is equal to 1 second.

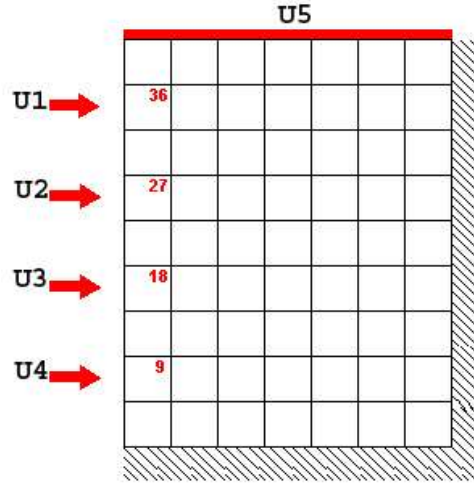


Figure 3.21: Sketch of the Heated Plate with constant temperature on the north and four inputs on the west side. The inputs are denoted by U_1, U_2, U_3, U_4, U_5

The heat conduction model of this heated plate is given by:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) \quad (3.40)$$

There are incoming heat fluxes at the west boundary side denoted by U_1 to U_4 , see Figure 3.21. At the southern, eastern and western boundary sides where the actuators U_1 to U_4 are not placed, the plate is insulated:

In the numerical simulation, (3.40) is discretised by employing the Finite Volume Method [70], where (3.40) is integrated over a unit volume $\Delta V = \Delta x \times \Delta y \times 1\text{cm}$.

$$\int_t^{t+\Delta t} \int_{\Delta V} \rho c_p \frac{\partial T}{\partial t} dV dt = \int_t^{t+\Delta t} \int_{\Delta V} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) dV dt + \int_t^{t+\Delta t} \int_{\Delta V} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) dV dt \quad (3.41)$$

The same discretization procedure is applied as in section 3.3.2. The integration of (3.40) for a specific g -th grid point located at (x_P, y_P) results in (3.42).

$$\begin{aligned} a_g T((x_P, y_P), k+1) &= a_g^0 T((x_P, y_P), k) + a_W T((x_{P-1}, y_P), k+1) \\ &+ a_E T((x_{P+1}, y_P), k+1) + a_N T((x_P, y_{P+1}), k+1) \\ &+ a_S T((x_P, y_{P-1}), k+1) + S_g \mathbf{u}(k) \end{aligned} \quad (3.42)$$

The term $T((x_P, y_P), k)$ denotes the temperature at a specific coordinate (x_P, y_P) at time step k and $T((x_{P-1}, y_P), k)$, $T((x_{P+1}, y_P), k)$, $T((x_P, y_{P+1}), k)$, $T((x_P, y_{P-1}), k)$ denote the temperatures of the west, east, north, and south neighboring grid points, respectively. The temperature at the current time step and at a specific grid point is denoted by $T(x_P, y_P, k)$.

The term $S_g = (S_{u_1}(k) \dots S_{u_5}(k))$ incorporates the contribution from the inputs. In this example, the inputs are defined at the north and the western boundaries. The input vector is $\mathbf{u}(k) = \text{col}(u_1(k), u_2(k), \dots, u_5(k))$.

Calculating (3.42) for all grid points and collecting all grid points into a vector

$$\mathbf{T}(k) = \begin{pmatrix} T((x_1, y_1), k) \\ T(x_2, y_2), k \\ \dots \\ T(x_{33}, y_{44}), k \end{pmatrix}$$

and setting

$$B(k) = \begin{pmatrix} S_1 \\ S_2 \\ \dots \\ S_{1452} \end{pmatrix}$$

will give the recursive linear system of equations:

$$A\mathbf{T}(k+1) = A_0\mathbf{T}(k) + B\mathbf{u}(k) \quad (3.43)$$

The matrix $A \in \mathbb{R}^{1452 \times 1452}$ in (3.43) is a penta-diagonal matrix with the structure:

$$A = \begin{pmatrix} a_g & -a_E & 0 & \dots & -a_S & -a_N & 0 & \dots \\ -a_W & -a_g & -a_E & 0 & \dots & 0 & -a_S - a_N & 0 & \dots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ \ddots & \ddots & \ddots & \dots & -a_W & -a_N & 0 & a_g & \end{pmatrix} \quad (3.44)$$

and the matrix $A_0 \in \mathbb{R}^{1452 \times 1452}$ is a diagonal matrix with a_g^0 at every grid point on its diagonal.

The original model is simulated until steady state subject to the change in north temperature boundary to 70°C and step changes to 100kW/m^2 in the west fluxes. The initial condition is $T(x, y, 0) = 0$ for the whole plate.

Algorithm 2.4.10 is implemented to find the POD basis functions. The simulation results during 1000 time steps are collected into

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \quad \mathbf{T}(2) \quad \dots \quad \mathbf{T}(1000)).$$

The eigenvalue spectrum of $\mathbf{C} = \frac{1}{1000} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$ is depicted in Figure 3.24. To

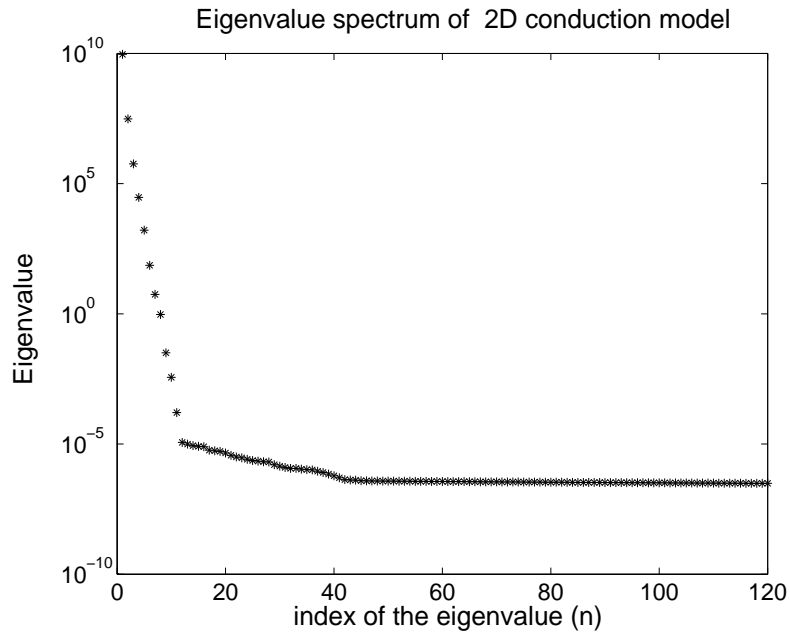


Figure 3.22: Eigenvalue spectrum of the two dimensional heat conduction

construct the reduced order model, only five POD basis functions are taken, hence $\Phi = (\varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_5)$. The first five POD basis associated with the first five largest eigenvalues are shown in Figure 3.23 and Figure 3.24.

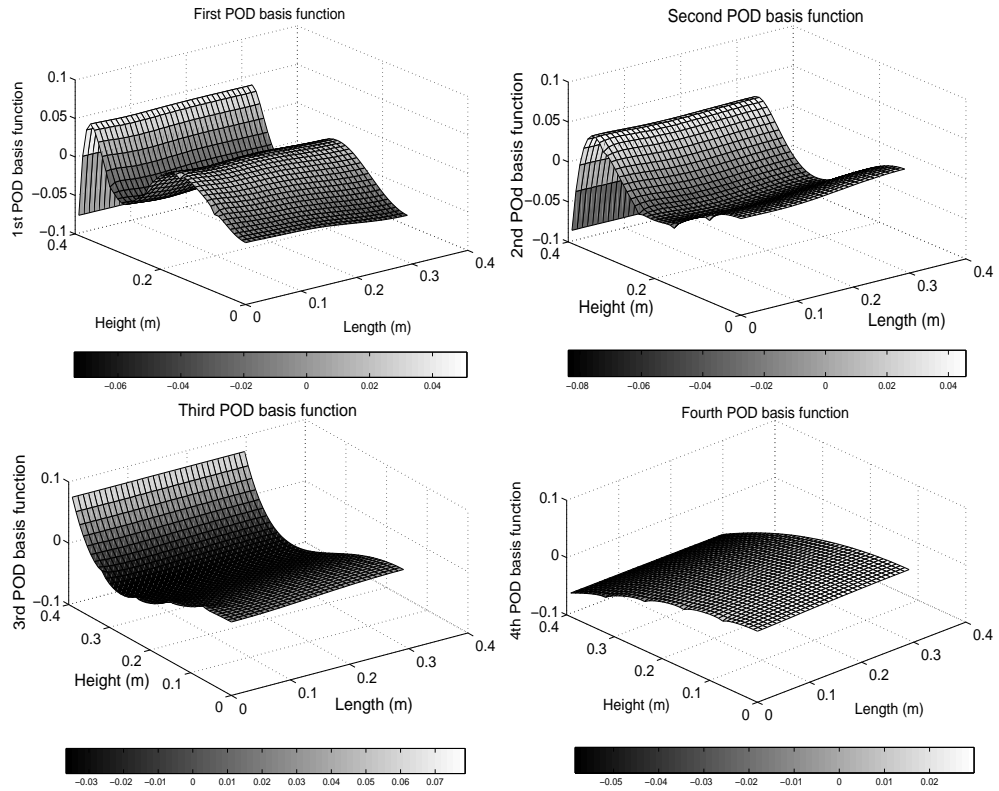


Figure 3.23: The first to fourth POD basis functions

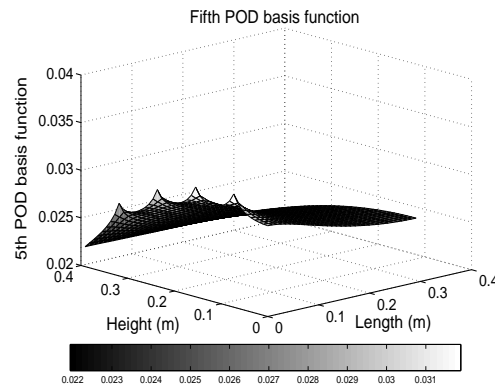


Figure 3.24: The fifth POD basis function

In the reduced order model, the temperature data at every time step $\mathbf{T}(k)$ is approximated by:

$$\mathbf{T}(k) \approx \mathbf{T}_n(k) := \Phi \mathbf{a}(k)$$

where $n = 5$ and $\mathbf{a}(k) = \text{col}(a_1(k), \dots, a_5(k))$. Replacing $\mathbf{T}(k)$ by $\mathbf{T}_n(k)$ in (3.43) and applying the Galerkin projection of (3.43) onto Φ , we obtain a reduced order model of the form:

$$\begin{aligned} \underbrace{\Phi^\top A \Phi}_{A_r} \mathbf{a}(k+1) &= \underbrace{\Phi^\top A_0 \Phi}_{A_0} \mathbf{a}(k) + \underbrace{\Phi^\top B}_{B_r} \mathbf{u}(k) \\ A_r \mathbf{a}(k+1) &= A_0 \mathbf{a}(k) + B_r \mathbf{u}(k) \end{aligned} \quad (3.45)$$

The reduced order model as in (3.45) is then validated by using the same initial conditions and the excitation signals as in the conditions used to derive the snapshot data in \mathbf{T}_{snap} . Hence, (3.45) is simulated by the initial condition with $\mathbf{a}(0)$ as:

$$\mathbf{a}(0) = \Phi^\top \mathbf{T}(0) = \Phi^\top \mathbf{0} = \mathbf{0}$$

The same conditions are validated because the reduced order model is primarily going to be used to control the process at the same operating point. The reduced order model is also simulated for 1000 time steps.

The mean of the absolute deviation $\bar{\epsilon}$ between the original model and the reduced order model using five POD basis functions ($n = 5$) is shown in Figure. 3.25. The time-averaged absolute error is defined as:

$$\bar{\epsilon} = \frac{1}{1000} \sum_{k=1}^{1000} | \mathbf{T}(k) - \mathbf{T}_n(k) | \quad (3.46)$$

The plot of $\bar{\epsilon}$ ((3.46)) is given in Figure 3.25. It is shown that the reduced order model can capture the dynamics of the original model quite accurately, with highest deviation is less than 0.08°C for temperature variations of about 100°C . The reduced order model is simulated by solving five equations in (3.45) while the original model solves 1452 equations. So with less than 0.5% of the original order, the reduced order model can simulate the original process very well.

3.6 Controller design

Since the reduced order model is representative enough for the original process, a controller can be designed based on the reduced order model. The control objective of this case is to reach the desired temperature distribution along the plate optimally. The original model has 1452 states, while the reduced order model has only 5 states. Consequently, the reduced order model is much more feasible to be used in the control and optimization algorithm.

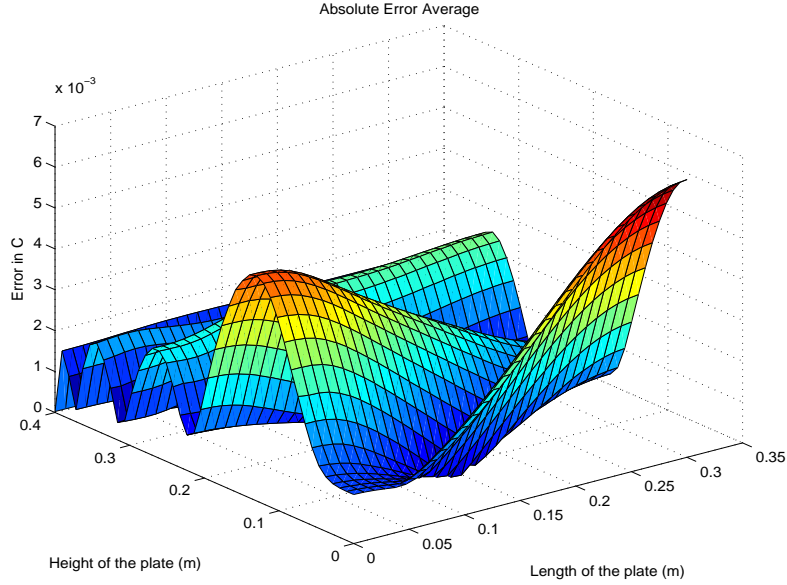


Figure 3.25: The spatial distribution of the time-averaged absolute error ϵ as defined in (3.46)

The reduced order model is given by (3.45). Assuming that A_r in (3.45) is invertible, (3.45) can be written in explicit form as

$$\mathbf{a}(k+1) = \underbrace{A_r^{-1}A_{0r}}_{A_s} \mathbf{a}(k) + \underbrace{A_r^{-1}B_r}_{B_s} \mathbf{u}(k) = A_s \mathbf{a}(k) + B_s \mathbf{u}(k) \quad (3.47)$$

where $A_s = A_r^{-1}A_{0r} \in \mathbb{R}^{5 \times 5}$ and $B_s = A_r^{-1}B_r \in \mathbb{R}^{5 \times 5}$.

The measurement points can be considered as outputs of interest. In this example the measurements are temperatures at several locations. The output vector \mathbf{y}_m is defined as:

$$\mathbf{y}_m(k) = C\mathbf{T}(k) \quad (3.48)$$

where $C \in \mathbb{R}^{n_y \times 1452}$ is a matrix that selects the measured temperatures. The number of measurement points is denoted by n_y . For the two dimensional heat conduction model, there are six measurement points chosen. The locations of the six measurement points are shown in Figure 3.26.

Since we use the reduced order model, $\mathbf{y}_m(k)$ can be transformed to a function of $\mathbf{a}(k)$ since $\mathbf{T}(k)$ is approximated as $\mathbf{T}_n(k) = \Phi\mathbf{a}(k)$ in the reduced order model. Hence,

$$\mathbf{y}_m(k) \approx C\Phi\mathbf{a}(k) = C_s\mathbf{a}(k) \quad (3.49)$$

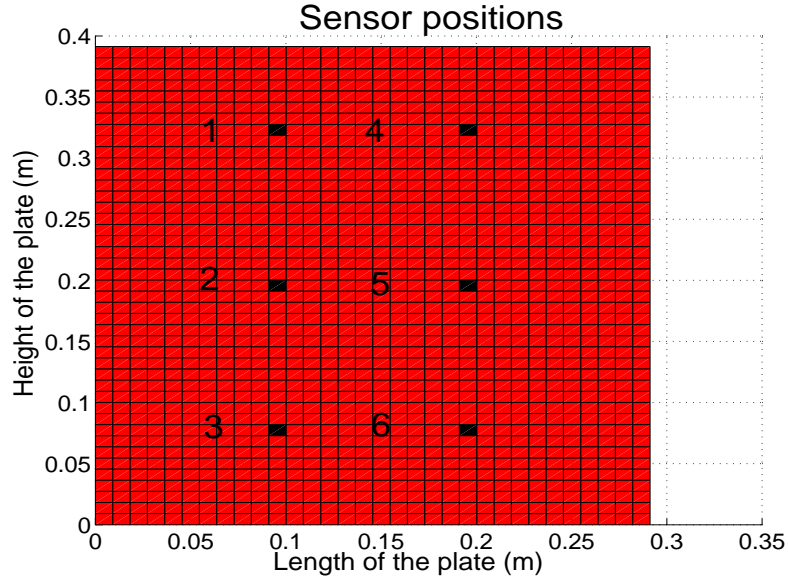


Figure 3.26: The locations of the measurement points, denoted by numbers 1 to 6

where $C_s = C\Phi$.

The equations (3.47) and (3.49) form together the *state space model*:

$$\begin{aligned}\mathbf{z}(k+1) &= A_s \mathbf{z}(k) + B_s \mathbf{u}(k) \\ \mathbf{y}(k) &= C_s \mathbf{z}(k)\end{aligned}\tag{3.50}$$

where $\mathbf{z}(k) = \text{col}(a_1(k), a_2(k), \dots, a_5(k))$ is the *state* of the systems. The matrices $A_s \in \mathbb{R}^{5 \times 5}$, $B_s \in \mathbb{R}^{5 \times 5}$, $C_s \in \mathbb{R}^{6 \times 5}$ are called *state space matrices*.

Based on the state space model (3.50), a controller can be designed. Control design is not feasible for the original model because the state vector $\mathbf{z}(k)$ will have dimension of 1452. Control algorithms will calculate the required input signals needed to reach the control objective based on the state space matrices A_s , B_s and C_s of the reduced order model. If the dimensions of these matrices are large (e.g larger than 200), then the calculation will be tedious.

3.6.1 Control objective

In many applications, it is desired to steer the temperature to a specific desired setpoint by applying suitable control actions. This means that one wishes to

minimize the tracking error, obtained as the difference between the desired and achieved temperature. For notational convenience, the spatial notation is slightly abused. The spatial coordinate pair (x, y) is shortened as x so that in the forthcoming discussion $T(x, y, k)$ is written as $T(x, k)$. Let \mathcal{U} be the collection of all sequences $\mathbf{u}(k) \in \mathbb{R}^p$ with $k = 0, \dots, N_{\text{opt}}$.

Translated mathematically, the control objective or the control problem is formulated as follows:

Problem 3.6.1 Given the desired temperature $T_{\text{ref}}(x, k)$ with $x \in \mathbb{X}$ and $k = 0, \dots, N_{\text{opt}}$, find a control signal $\mathbf{u}(k) \in \mathcal{U}$ such that the squared L_2 norm of the tracking error

$$\sum_{k=0}^{N_{\text{opt}}} \sum_{x \in \mathbb{X}} \| T_{\text{ref}}(x, k) - T(x, k) \|^2$$

is minimized over all feasible input signals $\mathbf{u} \in \mathcal{U}$. Thus \mathbf{u} is determined as:

$$\mathbf{u}(k) = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{k=0}^{N_{\text{opt}}} \sum_{x \in \mathbb{X}} \| T_{\text{ref}}(x, k) - T(x, k) \|^2. \quad (3.51)$$

The prediction N_{opt} is the length of the prediction horizon over which the control signals are going to be optimized. For optimization over a long horizon or large N_{opt} , a fast model is required because N_{opt} predictions of $T(x, k)$ have to be provided.

The control problem 3.6.1 is defined for the full order model. Since the order of the discretised original model is very high, i.e.1452, it is not feasible to incorporate the temperature at the whole plate into the control and optimization problem. To be able to incorporate the whole temperature distribution, the control objective has to be re-formulated based on the reduced model.

In the reduced model, the temperature distribution is expressed as a spectral expansion of a set of POD basis functions:

$$\mathbf{T}(k) \approx \mathbf{T}_n(k) = \Phi \mathbf{a}(k) \quad (3.52)$$

where $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_n)$, $k = 0, 1, \dots, N_{\text{opt}}$.

Similarly, the reference temperature profile \mathbf{T}_{ref} can also be approximated as a spectral expansion of n POD basis functions:

$$\mathbf{T}_{\text{ref}}(k) \approx \mathbf{T}_{\text{ref}_n}(k) = \Phi \mathbf{a}_{\text{ref}}(k) \quad (3.53)$$

Since the POD basis functions are orthonormal vectors, $\Phi^\top \Phi$ equals identity matrix, so $\mathbf{a}(k)$ and \mathbf{a}_{ref} can be written as:

$$\begin{aligned}\mathbf{a}(k)\mathbf{z}(k) &= \Phi^\top \mathbf{T}_n(k) \\ \mathbf{a}_{\text{ref}}(k) = \mathbf{r}(k) &= \Phi^\top \mathbf{T}_{\text{ref}_n}(k)\end{aligned}$$

To control the full order model, the original control problem has to be reformulated in terms of the reduced order model. In the next section, the design of a Linear Quadratic Regulator (LQR) will be given.

3.6.2 The LQR Controller

There are various optimal controllers which can be applied to solve the optimization problem 3.6.1. The type of controller which is going to be implemented in this case is the Linear Quadratic Regulator (LQR) controller [50].

Problem 3.6.1 can be also reformulated for the *reduced order model* and an LQR control objective as follows:

Problem 3.6.2 Control problem of the reduced order model:

Given the reduced model in state space form,

$$\mathbf{z}(k+1) = A_s \mathbf{z}(k) + B_s \mathbf{u}(k) \quad (3.54)$$

$$\mathbf{y}(k) = C_s \mathbf{z}(k) \quad (3.55)$$

the POD basis $\Phi = (\varphi_1 \ \dots \ \varphi_n)$ and let $\mathbf{r}(k) = \text{col}(\mathbf{r}_1(k), \dots, \mathbf{r}_n(k))$ and $r_i := (\mathbf{T}_{\text{ref}}(k), \varphi_i)$. Find $\mathbf{u} \in \mathcal{U}$ such that:

$$\begin{aligned}J(\mathbf{z}_0, \mathbf{u}) &= \underbrace{\sum_{k=0}^{N_{\text{opt}}-1} \left[(\mathbf{r}(k) - \mathbf{z}(k))^\top Q (\mathbf{r}(k) - \mathbf{z}(k)) \right]}_{\text{deviation from the desired POD coefficients}} \\ &+ \underbrace{\sum_{k=0}^{N_{\text{opt}}-1} \left[\mathbf{u}^\top(k) R \mathbf{u}(k) \right]}_{\text{cost of input energy}} + \underbrace{\left[\mathbf{r}(N_{\text{opt}}) - \mathbf{z}(N_{\text{opt}}) \right]^\top E \left[\mathbf{r}(N_{\text{opt}}) - \mathbf{z}(N_{\text{opt}}) \right]}_{\text{deviation at the last prediction step}}\end{aligned} \quad (3.56)$$

is minimized over $\mathbf{u} \in \mathcal{U}$. Here, $\mathbf{z}(k)$ is the solution of (3.50) with initial condition $\mathbf{z}(0) = \mathbf{z}_0$, $Q = Q^\top \geq 0$, $R = R^\top > 0$, $E = E^\top \geq 0$ are the weighting matrices for the states, input, and states at last prediction step, respectively.

In problem 3.6.2, we will assume that the control $\mathbf{u}(k)$ may depend on the state $\mathbf{z}(k)$ and the reference trajectory \mathbf{r}_{ref} . Stated otherwise, $\mathbf{u}(k)$ will causally depend on the state \mathbf{z} of (3.50) and the remaining reference trajectory \mathbf{r} . Thus \mathcal{U} comprises the control signals $\mathbf{u}(k)$ of the form:

$$\mathbf{u}(k) = f(\mathbf{z}(k), \mathbf{r}_{\text{ref}}(k), \mathbf{r}_{\text{ref}}(k+1), \dots, \mathbf{r}_{\text{ref}}(N_{\text{opt}}))$$

In the classic LQR design, the optimal control signal $\mathbf{u}(k)$ is only a function of the state $\mathbf{z}(k)$ while $\mathbf{r} = 0$.

To solve (3.56), we consider the solution $\mathbf{P}(k) \in \mathbb{R}^{n \times n}$ of the Algebraic Riccati Equation:

$$\begin{aligned} \mathbf{P}(k) = & A_s^\top \mathbf{P}(k+1) A_s + Q \\ & - A_s^\top \mathbf{P}(k+1) B_s (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{P}(k+1) A_s \end{aligned} \quad (3.57)$$

which is solved backwards in time for $k = N_{\text{opt}}$ until $k = 0$ with initial condition $P(N_{\text{opt}}) = E$. The equation in (3.57) is known as the Algebraic Riccati Equation (ARE).

To incorporate the reference signals in the optimization algorithm, define $\mathbf{v}(k) \in \mathbb{R}^{n_z}$ with n be the number of states be the unique solution of

$$\mathbf{v}(k) = \left(A_s^\top - A_s^\top \mathbf{P}(k+1) B_s (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \right) \mathbf{v}(k+1) - Q \mathbf{r}(k) \quad (3.58)$$

Similar to the computation of (3.57), $\mathbf{v}(k)$ is computed backward from $k = N_{\text{opt}}$ to $k = 0$ with $\mathbf{v}(N_{\text{opt}}) = 0$.

Theorem 3.6.3 *Let $\mathbf{P}(k)$ and $\mathbf{v}(k)$ be as formulated in (3.57) and (3.58). Then the optimal control input that solves Problem 3.6.2 is given by:*

$$\mathbf{u}^*(k) = - \underbrace{F_u(k) \mathbf{z}(k)}_{\text{function of states}} - \underbrace{G_u(k) \mathbf{v}(k+1)}_{\text{function of reference signals}}$$

where:

$$F_u(k) = -(R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{P}(k+1) A_s \quad (3.59)$$

$$G_u(k) = (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{v}(k+1) \quad (3.60)$$

The proof of Theorem 3.6.3 can be found in Appendix B. Note that the optimal control input \mathbf{u}^* is now a function of both the state $\mathbf{z}(k)$ and the reference signals since $\mathbf{v}(k+1)$ is a function of the reference signals.

3.7 Application of the LQR controller

The LQR controller as discussed in the previous section is applied to the full model of the heated plate. The control design is based on the reduced model which only has 5 states instead of 1452 as in the original model. That is, we take $n = 5$ in Problem 3.6.2.

The reference trajectory \mathbf{T}_{ref} is chosen to be the steady state temperature distribution in the open loop case when the plate is heated from zero temperature distribution by constant heat fluxes on the west side of 1000 kW/m^2 and a constant north temperature of 70° C .

The desired trajectory \mathbf{T}_{ref} in terms of the POD basis coefficients is calculated as:

$$\mathbf{r} = \Phi^\top \mathbf{T}_{\text{ref}} \quad (3.61)$$

In this example, the desired temperature distribution \mathbf{T}_{ref} is the same for all time steps, so \mathbf{r} is also constant.

The weighting matrices Q and R for the states and the inputs are chosen to be:

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} 10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 10^{-8} & 0 & 0 \\ 0 & 0 & 0 & 10^{-8} & 0 \\ 0 & 0 & 0 & 0 & 10 \end{pmatrix}$$

The prediction horizon N_{opt} is equal to 1000. The symmetric matrix $E = \mathbf{P}(k)$ is the solution of steady state Riccati Equation:

$$E = A_s^\top E A_s + Q - A_s^\top E B_s \left(R + B_s^\top E B_s \right)^{-1} B_s^\top E A_s$$

This choice of E is the solution $\mathbf{P}(k)$ of (3.57) by setting $\mathbf{P}(k) = \mathbf{P}(k+1)$ when $k = N_{\text{opt}}$.

In this example, it is considered important to track all desired POD coefficients \mathbf{r} as close as possible, therefore the weighting Q equals identity. The heated plate warms up quite slowly at the top part. Therefore the north temperature (u_5) is weighted more than the other control signals. The western heat fluxes are weighted the same.

Despite the dramatic reduction of model order, on which the controller is based, the controller performs very well. In Figure 3.27, it can be seen that the deviation from the desired temperature distribution is very small.

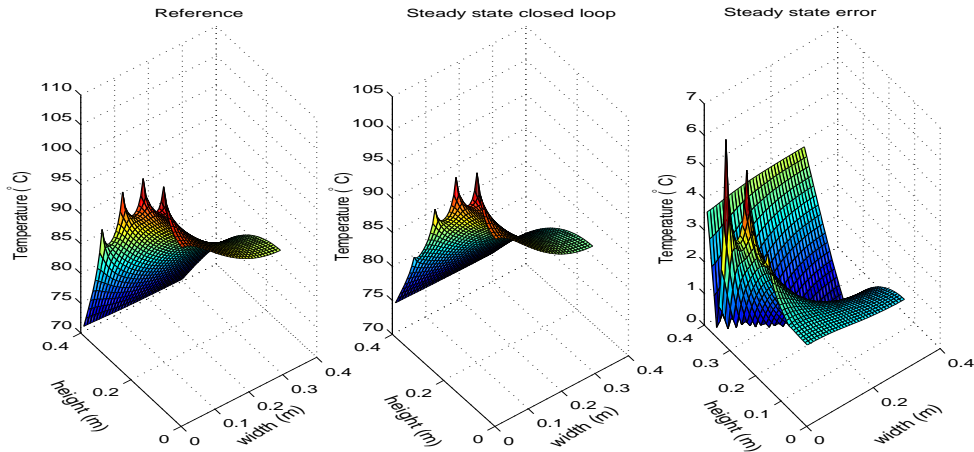


Figure 3.27: The desired temperature profile (left), the steady state temperature profile obtained by implementation of the controller (middle), the steady state error (right)

The maximum deviation is 4°C for temperature variations of about 75°C during the simulation. The deviation plotted in (3.27) is a function of the amplitude difference, while the LQR controller as formulated in problem 3.6.2 minimized the quadratic deviation. The maximum deviations occur at the temperature near the north boundary, have slow dynamics compared to the other regions of the plates. Therefore, the dynamics captured by the basis vectors used to construct the reduced order model is less accurate in this area than elsewhere. Due to the insulations, heat is preserved more in the southern area because there is no heat exchanged with the surroundings. The dynamics of this area then dominates the dynamics of the whole system because there are more temperature changes.

The optimality of the closed loop system can be seen from the plots of the actuator signals in Figure 3.28 to Figure 3.30.

All actuators try to reach the desired reference signal from zero temperature as quickly as possible. Therefore there is a significant raise of heat fluxes in the beginning of controller's operation. It is interesting to observe from Figure 3.29 that for two actuators: u_3 and u_4 (the two heat fluxes from the west side)

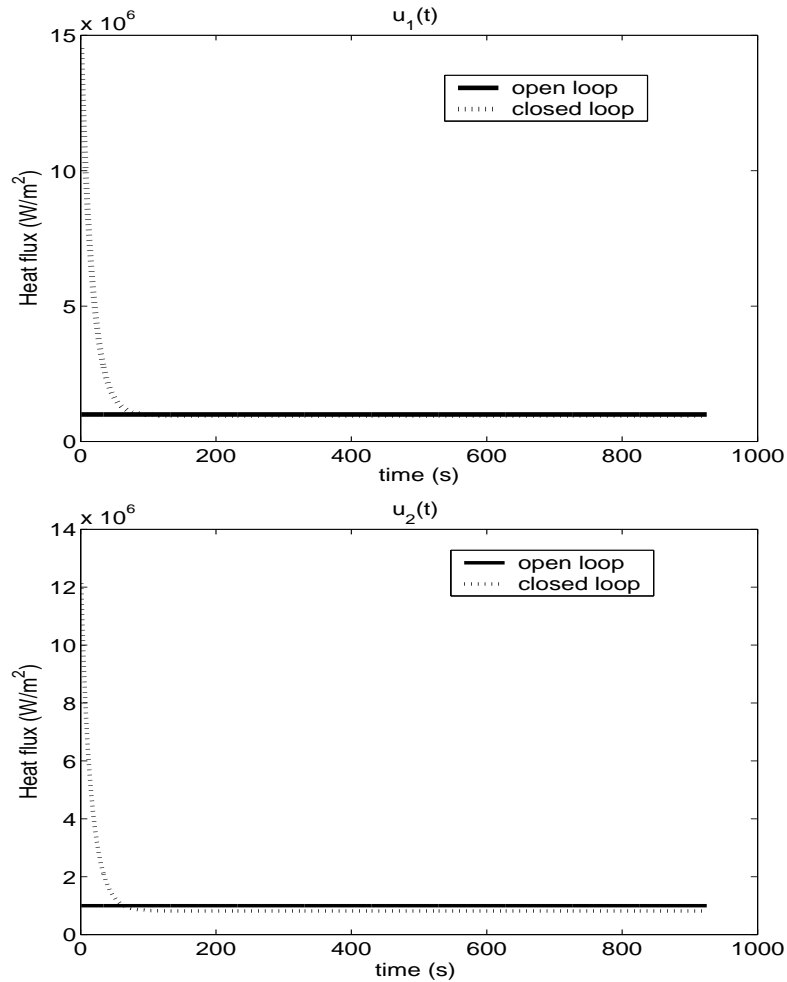


Figure 3.28: The closed loop and open loop responses of u_1 and $u_2(t)$. Initially, the heat fluxes increase to reach the desired temperature as fast as possible. The settling heat fluxes are about the same as the open loop inputs

the settling heat fluxes are lower than in the open loop excitations for the same time period. These actuators compensate the increase of energy demand in the transient time in order to reach the desired setpoint as by minimizing the energy supply in the remaining period.

This case study shows that by using the reduced order model for optimal control design (LQR control in this case), we can control a high order complex system described by PDE's. It also shows that by using an optimal controller,

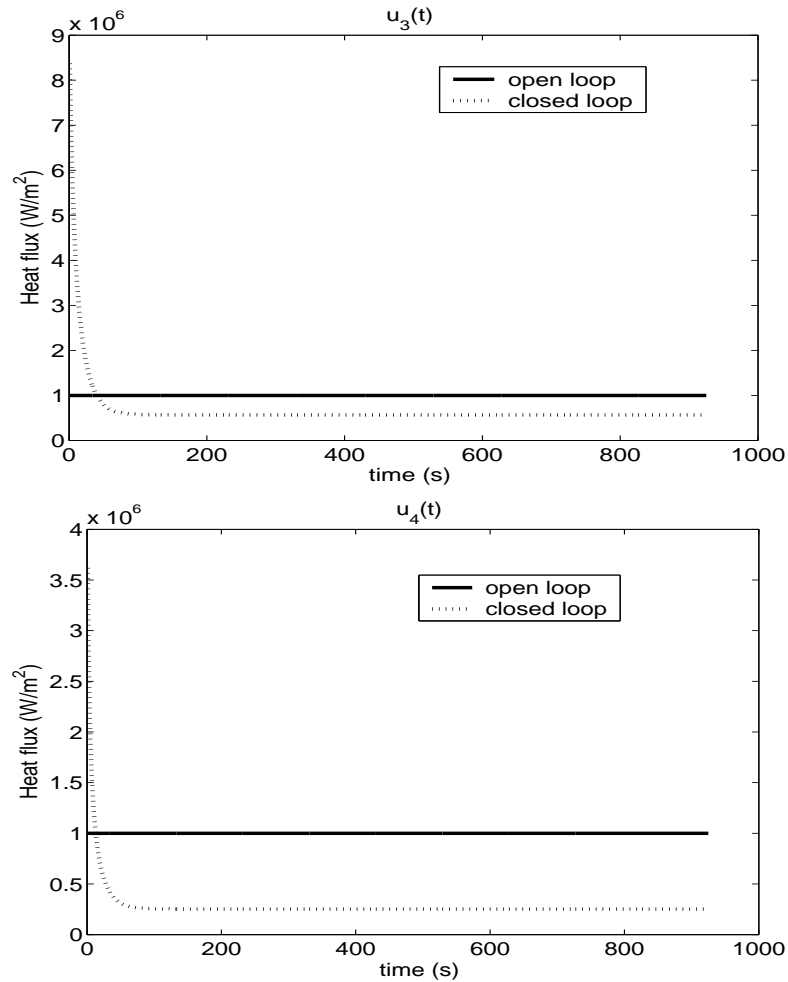


Figure 3.29: The closed loop and open loop responses of u_3 and u_4 . Initially, the heat fluxes increase to reach the desired temperature as fast as possible. The settling heat fluxes are lower in these two inputs

the amount of energy consumption needed to reach a desired temperature distribution can be managed efficiently.

Energy management is indeed a very important issue for industries: how to meet process specifications by managing energy consumption as efficiently as possible. Model based control is often infeasible due to the high complexity of the model. This study shows that the availability of reduced order model enables the synthesis of optimal controllers.

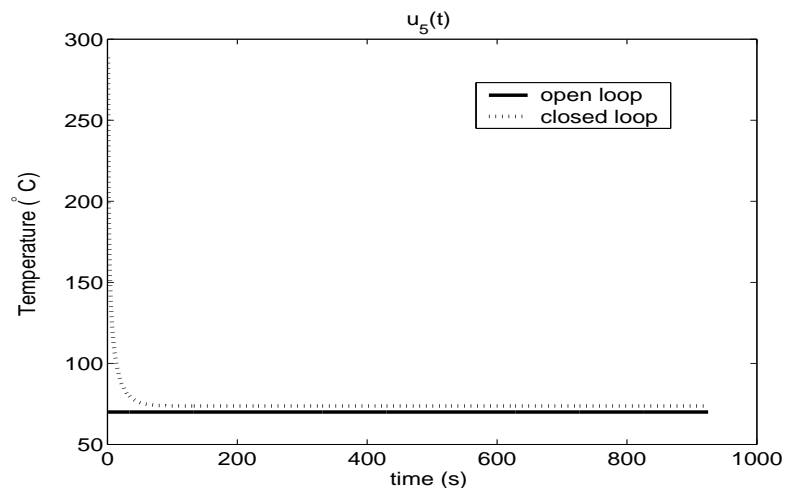


Figure 3.30: The closed loop and open loop responses of u_5 or the north temperature. Initially, the temperature increases to reach the desired temperature as fast as possible. The settling north temperature about the same as the open loop input

The responses of the six measurement points as depicted in Figure 3.26 are shown in Figure 3.31 to Figure 3.33. The comparisons between the open loop

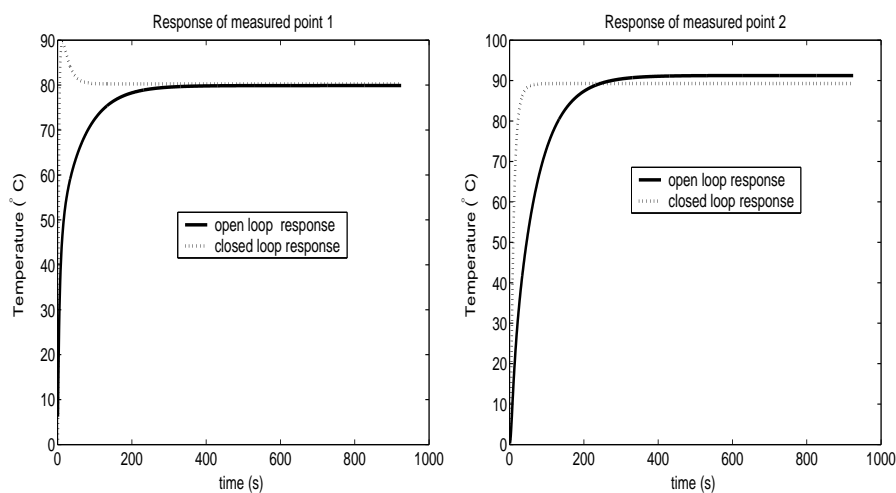


Figure 3.31: The open loop (solid) and closed loop responses (dotted) of measured point 1 and measured point 2

and closed loop responses of the measured points show that the closed loop

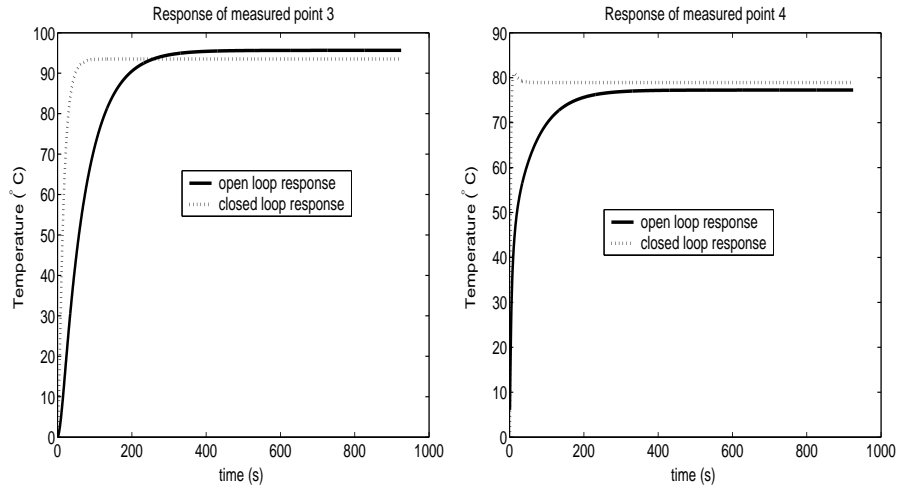


Figure 3.32: The open loop (solid) and closed loop responses (dotted) of measured point 3 and measured point 4

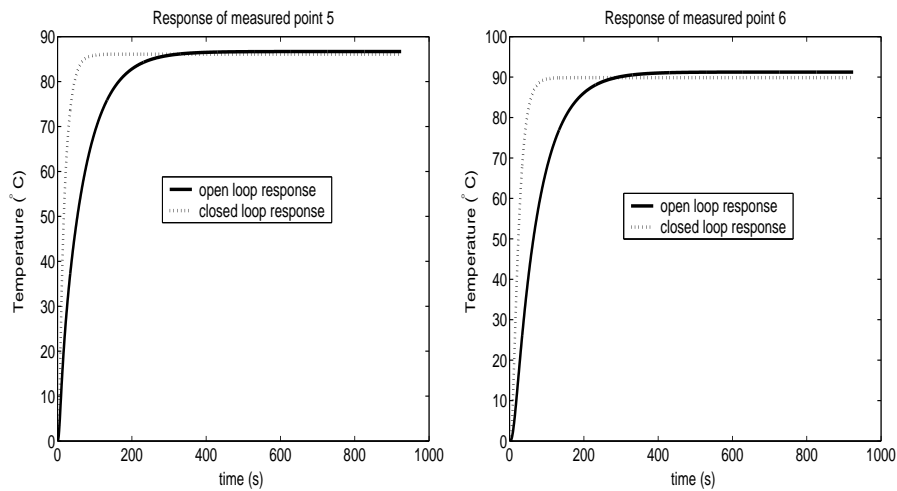


Figure 3.33: The open loop (solid) and closed loop responses (dotted) of measured point 5 and measured point 6

responses reach steady state faster than the open loop responses because the actuators anticipate the desired reference signals.

In this case, there are no constraints on the input signals. In principle, many controller types can be implemented by using the reduced order model as the base model. One can implement the Model Predictive Controller (MPC) to

handle constraints in input signals, variations of the states, and many others. Examples of MPC implementations on reduced order models can be found in [31] and [4].

3.8 Summary

In this chapter, the method of proper orthogonal decomposition has been used as a tool for reduced order modeling of heat conduction models. The heat conduction models are discretized by the finite volume method, which is used in many Computational Fluid Dynamics (CFD) packages. An introduction to CFD models is given in section 3.2. The discretization scheme is presented in detail for the case of a one dimensional heat conduction model in section 3.3.2. In section 3.3, it is shown that the spatial variations are very crucial in the snapshot data since the POD basis will capture the spatial dynamics. It is also shown that in the case of linear heat conduction processes, the variations of the excitation signals with respect to time do not substantially change the spatial dynamics. As a result, the reduced order model can still follow the original dynamics very well under different types of temporal excitations.

In both models, the POD method is shown to be a very effective reduction method. The resulting reduced order model for the two dimensional heat conduction process is used in section 3.7 as the base model for control system design. Since the reduced order model has a significantly lower dimension than the original model (less than 1% of the original model order), it is feasible to integrate it with the controller. The controller can successfully track the reference temperature distribution defined for the whole spatial domain. It is shown in section 3.7 that with an optimal control design, the input energy can be managed such that the reference trajectory can be reached optimally.

In conclusion, this chapter gives a detailed overview of how to derive a reduced order model from PDE's discretized by the Finite Volume Method and how to build a controller based on the reduced order model. The procedures of obtaining the reduced order model are summarized as follows:

1. Calculate the variables of interest (e.g temperature) for a specified time period using the original model, which is assumed to be a representative approximation of the original system. Use the typical variations of excitation signals. If there are different spatial boundary conditions for different cases which are going to be considered, then incorporate the

variations of the spatial boundary conditions in the snapshot matrix.

2. Derive the POD basis functions using Algorithm 2.4.10
3. Construct the reduced order model by performing Galerkin projection of the original discretized model onto the POD basis functions
4. Validate the reduced order model for different conditions by comparing the results of the original model and the reduced order model
5. If the validation results are satisfactory, design an optimal controller by translating the control objective as functions of the original process variables into functions of the POD coefficients. Besides the LQR controller implemented in this chapter, many other controller types, such as Model Predictive Controller, can be applied if the reduced order model is given in state space form. Software packages such as MATLAB provide a wide range of options for controller design including MPC.

Acceleration of Model Reduction for Nonlinear Models

| | | | |
|-----|---|-----|--|
| 4.1 | Introduction | 4.5 | POD-MPE model |
| 4.2 | Introduction to missing point estimation | 4.6 | Point selection in two dimensional heat conduction model |
| 4.3 | Selection of the important grid points | 4.7 | Controller design |
| 4.4 | Nonlinear heat conduction model | 4.8 | Summary |

This chapter is an extended version from the paper [3]

4.1 Introduction

Proper Orthogonal Decomposition (POD) or also known as Karhunen-Loève Expansion is one of the most prominent model reduction techniques for large scale models, [6], [34],[25]. The method is based on the use of data, either from experiments or from a rigorous simulation model which represent the typical operating conditions of the systems.

The data of process variables are collected and the main spatial dynamics are extracted from the data through a decomposition in basis functions. The main dynamics are then represented by a set of optimal, orthonormal basis functions. Typically, the number of basis functions that are used in the reduced order model is very few compared to the original order of the model. Nevertheless, it can capture the original dynamics within reasonable accuracy.

This method is therefore very useful for data obtained from numerical simulations of PDE-based models, which generally result in high order systems due to the fine spatial discretisation.

A continuous PDE-based model for a process variable $T : \mathbb{X} \times \mathbb{T} \rightarrow \mathbb{R}$ where \mathbb{T} is the time domain and \mathbb{X} is the spatial domain may be written as:

$$L(T) = D(T) \quad (4.1)$$

where $L(\cdot)$ is a polynomial differentiation operator:

$$L = L_0 + L_1 \frac{\partial}{\partial t} + \cdots + L_m \frac{\partial^m}{\partial t^m}$$

and $L_i \in \mathbb{R}$, $i = 0, \dots, m$ and $D(\cdot)$ is an operator that does not involve the temporal derivatives. The function $D(T)$ in general constitutes non linear functions of T , which may involve spatial derivatives of T and other external nonlinear or linear functions.

The residual $R(x, t)$ is defined as:

$$R(x, t) = L(T(x, t)) - D(T(x, t)) \quad (4.2)$$

In Chapter 2 we derived conditions under which T can be expressed as a Fourier expansion[29], i.e. the function T must belong to a separable Hilbert space \mathcal{X} so that it admits an expansion:

$$T(x, t) = \sum_{i=1}^{\infty} a_i(t) \varphi_i(x), \quad x \in \mathbb{X}, t \in \mathbb{T} \quad (4.3)$$

where the functions $\{\varphi_i(x)\}$ define an orthonormal basis for the Hilbert space \mathcal{X} .

The truncated expansion of $T(x, t)$ to the n -th order is given by:

$$T_n(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x), \quad x \in \mathbb{X}, t \in \mathbb{T} \quad (4.4)$$

The POD method requires that the Galerkin projection of the residual $R(T_n)$ on the space spanned by $\{\varphi_i\}_{i=1}^n$ vanishes. That is,

$$(R(T(x, t)), \varphi_i(x)) = 0, \quad i = 1, \dots, n \quad (4.5)$$

where (\cdot, \cdot) denotes the inner product.

Since the POD basis is orthonormal, using the linearity of the inner product, we obtain:

$$(L(T_n), \varphi_i) = \left(L \left(\sum_{j=1}^n a_j(t) \varphi_j \right), \varphi_i \right) = L(a_i(t))$$

From (4.5), the reduced order model is derived as an n -th order ODE which solves the unknown POD basis coefficients $a_i(t)$:

$$L(a_i(t)) = \left(D \left(\sum_{j=1}^n a_j(t) \varphi_j(x) \right), \varphi_i(x) \right) \quad (4.6)$$

In practice, the computational domain $\mathbb{X} \times \mathbb{T}$ is discretised as $\hat{\mathbb{X}} \times \hat{\mathbb{T}}$, with $\hat{\mathbb{X}} \subseteq \mathbb{X}$ and $\hat{\mathbb{T}} \subseteq \mathbb{T}$, both of finite cardinality. The resulting dimension of $\hat{\mathbb{X}}$, K , is typically high. Numerically, (4.1) is solved as a discrete K -th order model with the solution vector $\mathbf{T}(k) \in \mathbb{R}^K$ at every time step k .

Various reduced order modeling techniques such as balanced truncation, proper orthogonal decomposition attempt to find a suitable basis $\{\varphi_i\}_{i=1}^n$ so that the reduced order model can still follow the dynamics of the original model with very limited number of basis functions.

The main objective of model reduction is to obtain a “simpler” model which described the original model by less number of equations. In Chapter 2, a “simpler” model is considered as a model which is less complex than the original one. In Chapter 2, the complexity function has been defined as the number of nonzero Fourier coefficients [21] $\{a_i(t)\}$ in the expansion (4.3). The definition of a complexity function as the number of Fourier coefficients is used when deriving a POD reduced model.

In general, in the original model, the complexity function c_T may be equal to infinity as in (4.3). In the reduced order model, the complexity function c_{T_n} is limited to n . Since $c_{T_n} \ll c_T$, the reduced order model is considered simpler than the original model.

On the other hand, this notion of complexity does not relate to the *computational time*. Even though the reduced order model has less number of variables to be solved, the computational time may not decrease dramatically. In particular, if the original model is nonlinear, it is very difficult to decrease the computational time. Linear models, on the other hand, achieve satisfactory computational gain such that the CPU time needed to solve the reduced order model is very small compared to the original CPU time [31].

This can be explained as follows: For linear time invariant (LTI) systems, the Galerkin projection is conducted on linear time invariant functions as well. The functions $L(T)$ and $D(T)$ are not updated and the resulting reduced order model is fixed. In the case of nonlinear systems, these functions are

updated if they contain dependencies on T .

This is the typical case of Computational Fluid Dynamics (CFD) models as the density, heat conductivity, heat capacity are also temperature dependent. The discretization of the nonlinear PDE's results in linear time varying (LTV) systems. The reduced order model cannot only be calculated just once as in the case of LTI models, but has to be constantly updated because the model parameters are also changing. This is very time consuming so that the CPU time of the calculation by the reduced order model does not differ significantly from the CPU time needed by the original model.

Some work has been done to improve numerical efficiency for linear-time-varying systems of modest scale (of dimension $\approx 10^2$) by employing several numerical techniques needed to simulate the reduced model [19]. Finding the solution of the POD coefficient $\{a_i(t)\}_{i=1}^n$ when a reduced model has been constructed is not tedious, but *updating parameters* of the reduced models is quite cumbersome.

In this chapter, we propose a practical solution to overcome this problem by conducting Galerkin projection on equations describing the dynamics of several points in the spatial domain instead of the equations of all grid points. It is necessary to identify which points in the spatial domain significantly contribute to the dynamics of the system. If there is only a limited number of points in the spatial domain which characterize the global dynamics, then only parameters of these points have to be updated. If only a partial set of model parameters has to be updated, the computational speed of the reduced order model will increase.

We show that by conducting such a procedure, the expected computational saving can be achieved. The procedure is applied to a linear time varying (LTV) system. The LTV system itself is obtained from a discretisation of a 2D nonlinear heat conduction problem. The reduced order model, which is considerably faster than the original one is also combined with a controller to control a nonlinear heat conduction process.

This chapter is organized as follows. First, the technique used to estimate the states of the system based on limited information is discussed, then the discretised 2D nonlinear heat conduction model is presented. Subsequently, the application of the acceleration technique to the 2D nonlinear heat conduction model is shown and finally, the control design based on the reduced order model is presented. The last section covers the main conclusion of this chapter.

4.2 Introduction to missing point estimation

Normally, POD basis coefficients $\{a_i(t)\}_{i=1}^n$ are found by projecting the POD basis $\Phi = \{\varphi_i\}_{i=1}^n$ onto the set of all equations of the original model or onto the complete set of the original data. Alternatively, POD basis coefficients can also be found on the basis of incomplete data, or incomplete equations. The objective of this section is to formalize a theoretical basis for such approach. The approach is named here as the Missing Point Estimation (MPE) since here the original data is assumed to be incomplete and the incomplete or missing data is estimated by the approximated POD basis coefficients.

Consider a subset \mathbb{X}_0 of the spatial domain \mathbb{X} , thus $\mathbb{X}_0 \subset \mathbb{X}$. This subset is also referred to as "a mask", a terminology usually used in image processing theory [34].

Suppose that the data set $T(x, t)$ is obtained (or measured) with $x \in \mathbb{X}, t \in \mathbb{T}$ for both \mathbb{X} and \mathbb{T} finite sets of cardinality K and N . Let $\mathbf{T}(t) = \text{col}_{x \in \mathbb{X}} T(x, t)$ and define a snapshot matrix \mathbf{T}_{snap} :

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \quad \dots \quad \mathbf{T}(N))$$

Assume that $T(\cdot, t)$ belongs to a separable Hilbert space \mathcal{X} , which is equipped with an ℓ_2 inner product (refer to Chapter 2 for more details).

Since $T(\cdot, t)$ is in the Hilbert space, the original, complete data set can be described by its POD expansion as:

$$\mathbf{T}(t) = \sum_{i=1}^{\infty} a_i(t) \varphi_i \quad (4.7)$$

The truncated expansion of (4.7) is:

$$\mathbf{T}_n(t) = \sum_{i=1}^n a_i(t) \varphi_i \quad (4.8)$$

The original function T is defined for the spatial domain \mathbb{X} and the time domain \mathbb{T} . If the spatial domain is restricted to \mathbb{X}_0 , then we can define T defined *only* at \mathbb{X}_0 as \tilde{T} . That is, let $\tilde{T} = T|_{\mathbb{X}_0 \times \mathbb{T}}$ be the mapping $\tilde{T} : \mathbb{X}_0 \times \mathbb{T} \rightarrow \mathbb{R}$. The function \tilde{T} is referred to as the *restriction* of T since it is defined for a *restricted* spatial domain \mathbb{X}_0 .

Similarly, let $\tilde{\varphi}_i$ be the restriction of φ_i to \mathbb{X}_0 . If the original POD basis function φ_i is defined for the whole spatial domain \mathbb{X} , then $\tilde{\varphi}_i$ is only defined

for \mathbb{X}_0 . The elements of $\tilde{\varphi}_i$ are the elements of the original POD basis function φ_i at the location $x \in \mathbb{X}_0$. We also denote $\tilde{\Phi} = (\tilde{\varphi}_1 \ \dots \ \tilde{\varphi}_n)$ as the basis for \tilde{T} in \mathbb{X}_0 . However, $\tilde{\Phi}$ is not an orthonormal nor an optimal basis.

Since the original function T can be approximated by the POD basis functions $\{\varphi_i\}_{i=1}^n$, $\tilde{\mathbf{T}}$ can also be approximated by $\tilde{\varphi}_i$ as:

$$\tilde{\mathbf{T}}_n(t) = \sum_{i=1}^n a_i(t) \tilde{\varphi}_i \quad (4.9)$$

To explain these notations, the following example is given.

Example 4.2.1 Suppose the spatial domain \mathbb{X} is divided into 5 grid cells whose locations are denoted as x_1, \dots, x_5 . Thus, $\text{card}(\mathbb{X}) = 5$. The temperature data at every time step is denoted as

$$\mathbf{T}(t) = \begin{pmatrix} T_1(x_1, t) \\ T_2(x_2, t) \\ T_3(x_3, t) \\ T_4(x_4, t) \\ T_5(x_5, t) \end{pmatrix}$$

Hence $\mathbf{T}(t) \in \mathbb{R}^5$.

Suppose $T_2(t)$ and $T_4(t)$ represent the measured temperature data. These two temperatures can be collected in a vector

$$\tilde{\mathbf{T}}(t) = \begin{pmatrix} T_2(x_2, t) \\ T_4(x_4, t) \end{pmatrix}$$

In this case, $\tilde{\mathbf{T}} \in \mathbb{R}^2$, so $G = 2 = \text{card}(\mathbb{X}_0)$.

$\mathbf{T}(t)$ is a collection of functions in a Hilbert space \mathbb{R}^5 , (refer to section 2.3) so that $\mathbf{T}(t)$ can be expanded by the POD basis functions.

$$\mathbf{T}(t) = \sum_{i=1}^5 a_i(t) \varphi_i \quad (4.10)$$

The POD basis functions are vectors in \mathbb{R}^5 , thus every POD basis function $\varphi_i \in \mathbb{R}^5$. Denote $\varphi_i(r)$ as the r -th entry of φ_i .

Then $\tilde{\varphi}_i = \text{col}(\varphi_i(2), \varphi_i(4))$. Hence, the measurement data $\tilde{\mathbf{T}}$ can be written as:

$$\tilde{\mathbf{T}}(t) = \begin{pmatrix} T_2(t) \\ T_4(t) \end{pmatrix} = \sum_{i=1}^5 a_i(t) \begin{pmatrix} \varphi_i(2) \\ \varphi_i(4) \end{pmatrix} = \sum_{i=1}^5 a_i(t) \tilde{\varphi}_i \quad (4.11)$$

If the original data is approximated by 2 POD basis functions as:

$$\mathbf{T}_n(t) = \mathbf{T}_2(t) = \sum_{i=1}^2 a_i(t) \varphi_i$$

then the corresponding two measured temperature data can be approximated as:

$$\tilde{\mathbf{T}}_n(t) = \tilde{\mathbf{T}}_2(t) = \sum_{i=1}^2 a_i(t) \tilde{\varphi}_i$$

As the POD basis is orthonormal, then the POD coefficients $\{a_i(t)\}_{i=1}^n$ can be calculated from the knowledge of the complete temperature data $\mathbf{T}(t)$ as $(\varphi_i, \mathbf{T}(t))$. The arising question is, is it possible to *estimate* the POD coefficients $\{a_i(t)\}$ from the restricted observations $\tilde{\mathbf{T}}$ defined over \mathbb{X}_0 ? In this case, the estimations of the POD coefficients by $\tilde{\mathbf{T}}$ are denoted by $\{\tilde{a}_i(t)\}$.

Given the orthonormal basis $\{\varphi_i\}_{i \in \mathbb{I}}$ for the process variable \mathbf{T} defined over the spatial domain \mathbb{X} and a measurement $\tilde{T}(x, t)$ defined over the mask \mathbb{X}_0 , our objective is to estimate the $\{\tilde{a}_i(t)\}_{i=1}^n$ in (4.12)

$$\tilde{T}_n(x, t) = \sum_{i=1}^n \tilde{a}_i(t) \tilde{\varphi}_i(x), \quad x \in \mathbb{X}_0 \quad (4.12)$$

such that the least square error

$$E(t) = \|\tilde{T}(x, t) - \tilde{T}_n(x, t)\|_{\mathbb{X}_0}^2 \quad (4.13)$$

is minimized.

The interpretations of (4.13) and (4.12) is that the coefficients $\{\tilde{a}_i\}_{i \in \mathbb{I}}$ are estimated such that the difference between the available data $\tilde{\mathbf{T}}(t)$ and its truncated expansion $\tilde{\mathbf{T}}_n(t)$ is minimized.

The optimal coefficients $\tilde{a}_i^*(t)$ in the criterion (4.13) satisfy the linear system of equations

$$\sum_{i=1}^n \tilde{a}_i^*(t) (\tilde{\varphi}_i(x), \tilde{\varphi}_j(x))_{\mathcal{X}_0} = \left(\tilde{T}(x, t), \tilde{\varphi}_j(x) \right)_{\mathcal{X}_0}, \quad j = 1, \dots, n \quad (4.14)$$

which can be written as

$$M\tilde{\mathbf{a}}(t) = \mathbf{f}(t) \quad (4.15)$$

where

$$M_{ij} = (\tilde{\varphi}_i(x), \tilde{\varphi}_j(x))_{\mathcal{X}_0}$$

and the j -th row of $\mathbf{f} \in \mathbb{R}^n$ as

$$f_j(t) = \left(\tilde{T}(x, t), \tilde{\varphi}_j(x) \right)_{\mathcal{X}_0} \quad (4.16)$$

Thus, with the knowledge of limited information, we can estimate the basis coefficients $\{\tilde{a}_i\}_{i \in \mathbb{I}}$.

Given the estimates of the coefficients $\{\tilde{a}_i^*(t)\}_{i \in \mathbb{I}}$ from solving (4.14) and the *complete* set POD basis functions $\{\varphi_i\}_{i \in \mathbb{I}}$, we can estimate $\mathbf{T}(t)$ as:

$$\mathbf{T}(t) \approx \hat{\mathbf{T}}(t) = \sum_{i=1}^n \tilde{a}_i^*(t) \varphi_i \quad (4.17)$$

Thus, $\tilde{a}_i(t)$ are calculated from the data defined on \mathbb{X}_0 . Using the complete POD basis, we can reconstruct the complete data $\mathbf{T}(t)$ and also estimate the data at the points outside \mathbb{X}_0 .

The quality of the POD basis coefficients indeed depends on how the subset \mathbb{X}_0 is selected. In the next section, we will discuss selection criteria for \mathbb{X}_0 . This is analogous to the problem of finding a set of sensor locations such that the measurement points will represent the dynamics of the original system accurately.

4.3 Selection of the important grid points

4.3.1 Point selection criterion 1

The problem of point selection is equivalent to selecting \mathbb{X}_0 so that the estimated POD coefficients $\{\tilde{a}_i(t)\}_{i=1}^n$ based on the measurement $\tilde{T} = T|_{\mathbb{X}_0 \times \mathbb{T}}$ provides a good estimate of T through (4.17).

Finding \mathbb{X}_0 is of evident interest for the characterization of suitable sensor locations by which the system dynamics can be recovered. The purpose of this section is to propose a selection criterion for such mask.

Let a POD basis be defined on \mathbb{X} and let the i -th basis function $\varphi_i(x)$ be represented as $\varphi_i(x) = (\varphi_i, x)$ with $\varphi_i \in \mathbb{R}^K$, $K = \dim(\mathbb{X})$ and (\cdot, \cdot) the standard (no weighting) inner product on \mathbb{R}^K . Let

$$\Phi = (\varphi_1 \quad \dots \quad \varphi_n) \quad (4.18)$$

represent the first n POD basis functions, $n \leq K$. In section 4.2, we also have introduced $\tilde{\varphi}_i$ as the restrictions of φ_i to a subset of the spatial domain $\mathbb{X}_0 \subset \mathbb{X}$.

Similarly, define $\tilde{\Phi}$ as:

$$\tilde{\Phi} = (\tilde{\varphi}_1 \quad \dots \quad \tilde{\varphi}_n) \quad (4.19)$$

If there are G measurement points, i.e. $G = \text{card } \mathbb{X}_0$, then $\tilde{\Phi} \in \mathbb{R}^{G \times n}$.

Suppose that $\mathbf{a}_{\text{org}}(t) \in \mathbb{R}^n$ denotes the vector of the n POD coefficients obtained by calculating $\{a_i(t)\}_{i=1}^n$ from a complete data. Then

$$\mathbf{T}_n(t) := \Phi \mathbf{a}_{\text{org}}(t)$$

and the observation $\tilde{\mathbf{T}}(t)$ is then approximated as

$$\tilde{\mathbf{T}}_n(t) = \tilde{\Phi} \mathbf{a}_{\text{org}}(t) \quad (4.20)$$

The formulation in (4.15) states that the estimates of the POD coefficients $\tilde{\mathbf{a}}$ obtained from $\tilde{\mathbf{T}}(t)$ satisfy

$$M \tilde{\mathbf{a}}(t) = \mathbf{f}(t)$$

where

$$M = \tilde{\Phi}^\top \tilde{\Phi} \quad (4.21)$$

and $\tilde{\Phi}$ is as defined in (4.19).

The vector $\mathbf{f} \in \mathbb{R}^n$ is defined as :

$$\mathbf{f} = \tilde{\Phi}^\top \tilde{\mathbf{T}}(t)$$

Since $\tilde{\mathbf{T}}$ can be approximated by $\tilde{\mathbf{T}}_n(t)$, then if only the truncated POD expansions of measured data are available:

$$f_j(t) = \left(\tilde{\varphi}_j, \tilde{\Phi} \mathbf{a}_{\text{org}} \right)$$

The estimates of the POD coefficients $\tilde{\mathbf{a}}$ based on the observation or incomplete data \tilde{T} can be written as:

$$\tilde{\Phi}^T \tilde{\Phi} \tilde{\mathbf{a}} = \tilde{\Phi}^T \tilde{\Phi} \mathbf{a}_{\text{org}} \quad (4.22)$$

From (4.22), we know that the requirement $\tilde{\mathbf{a}} \approx \mathbf{a}_{\text{org}}$ is equivalent to requiring

$$\tilde{\Phi}^T \tilde{\Phi} \approx I \quad (4.23)$$

The relation expressed in (4.23) implies that the closer $\tilde{\Phi}^T \tilde{\Phi}$ to the identity matrix, then the better the estimation of the original POD coefficients will be. So we must choose \mathbb{X}_0 such that $\tilde{\Phi}^T \tilde{\Phi}$ defined over \mathbb{X}_0 is close to the identity matrix.

Thus with $M \in \mathbb{R}^{n \times n}$ defined as in (4.15) we are interested in minimizing the criterion

$$e_{\mathbb{X}_0} = \| M - I \|^2 \quad (4.24)$$

in some matrix norm $\| \cdot \|$. In particular we consider the following norm for a square matrix X

$$\| X \|^2 := \sum_{i=1}^n \sum_{j=1}^n |X_{ij}|^2$$

where i and j represent the row and column indices of X , X_{ij} means the element of X at the i -th row and j -th column.

Note that $e_{\mathbb{X}_0} = 0$ if $\mathbb{X}_0 = \mathbb{X}$. Further, \mathbb{X}'_0 is considered a better mask than \mathbb{X}''_0 if

$$e_{\mathbb{X}'_0} \leq e_{\mathbb{X}''_0} \quad (4.25)$$

and

$$\dim \mathbb{X}'_0 = \dim \mathbb{X}''_0 \quad (4.26)$$

The quantity thus $e_{\mathbb{X}_0}$ determines whether a particular choice of the subset \mathbb{X}'_0 will be better than \mathbb{X}''_0 . Suppose \mathbb{X} consists of K grid points where the locations of the grid points can be denoted as $\{x_k\}_{k=1}^K$. The case when \mathbb{X}_0 consists of only one point located at x_k , $\mathbb{X}_0 = x_k$ is of specific interest.

Given the basis $\{\varphi_i\}_{i=1}^n$ and $\dim \mathbb{X}_0 = 1$, the criterion (4.24) then enables an ordering of all points in \mathbb{X} . For every grid point $x_k \in \mathbb{X}$, we calculate $e_{\mathbb{X}_0} = e_{x_k}$.

After this calculation is performed for every point in \mathbb{X} , let us re-index the points in \mathbb{X} as x_{k_1}, \dots, x_{k_K} such that

$$e_{x_{k_1}} \leq e_{x_{k_2}} \leq \dots \leq e_{x_{k_K}}$$

where k_1, k_2, \dots, k_K is the re-ordered index.

The point with minimum e_{x_k} , that is the one located at x_{k_1} will be considered as the *most important* point while the point with largest e_{x_k} or the one located at x_{k_K} is considered as the *least important* point. The measurement data $\tilde{\mathbf{T}}$ can thus be composed of the points with G smallest e_{x_k} .

Note that this is not the most optimal ordering, this is the ordering we assume to be effective to distinguish the dominant states of the original model.

Choosing only the first few points of the ordered e_{x_k} may result in a bad condition number of the matrix $M = \tilde{\Phi}^T \tilde{\Phi}$. The condition number Θ of a nonsingular matrix M is defined as:

$$\Theta(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)} \quad (4.27)$$

where $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$ are the maximum and the minimum eigenvalues of M , respectively.

The condition number is important when we use M as an operator to solve a set of unknowns z , such as:

$$Mz = b \quad (4.28)$$

In (4.28), the solution z is *unique* if $\Theta(M)$ is close to 1 (100 is usually the threshold [34]). If $\Theta(M)$ is close to 1, then we consider M *well-conditioned*. If the maximum eigenvalue of M is very large and the minimum eigenvalue of M is very small (≈ 0), then M is composed of almost linear vectors. A numerical calculation to solve (4.28) will then result in multiple solutions of z . This is an unwanted case as we would like to obtain a unique solution of z . If $\Theta(M)$ is large, we consider M *ill-conditioned* or singular. The best condition number $\Theta(M) = 1$, which means that all eigenvalues of M are the same and z can be determined uniquely.

Hence the number of points in mask \mathbb{X}_0 has to be chosen such that the condition number M in (4.15) is close enough to 1. Note that $M = \tilde{\Phi}^T \tilde{\Phi}$ only depends on the chosen POD basis and the mask \mathbb{X}_0 .

4.3.2 Point selection criterion 2

The first selection criterion is based on the deviation of $\tilde{\Phi}^T \tilde{\Phi}$ from the identity matrix. Since the POD basis Φ consists of orthonormal basis functions, $\Phi^T \Phi = I$. Hence, the first criterion is in principle based on $\tilde{\Phi}^T \tilde{\Phi}$ from $\Phi^T \Phi$. The

selection criterion 1 is based on the assumption that all POD basis are of *equal* importance. There is no weighting in the POD basis functions applied.

In this section, we proposed a criterion based on the snapshot data $\mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$ where $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{K \times N}$.

Define the matrix $J \in \mathbb{R}^{N \times N}$ as:

$$J = \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}} \quad (4.29)$$

The (i, j) th entry of J is given by:

$$J_{ij} := \sum_{k=1}^K T(x_k, t_i) T(x_k, t_j) \quad (4.30)$$

Since $\mathbf{T}(t) = \Phi \mathbf{a}(t)$ with Φ the POD basis matrix of dimension $K \times K$, we can decompose J as:

$$J = \tilde{J} + \hat{J}$$

where the (i, j) -th entry of \tilde{J} and \hat{J} are given by:

$$\tilde{J}_{ij} = \mathbf{a}_n^\top(t_i) \Phi_n^\top \Phi_n \mathbf{a}_n(t_j); \quad \hat{J}_{ij} = \mathbf{a}_t^\top(t_i) \Phi_t^\top \Phi_t \mathbf{a}_t(t_j)$$

where Φ and $\mathbf{a}(t)$ are decomposed as $\Phi = [\Phi_n \ \Phi_t]$ and $\mathbf{a}(t) = \text{col}(\mathbf{a}_n(t), \mathbf{a}_t(t))$ with $\Phi_n \in \mathbb{R}^{K \times n}$ having n columns and $\mathbf{a}_n(t)$ having n entries.

Let $\tilde{\Phi}_k \in \mathbb{R}^{1 \times n}$ be the k -th row of Φ_n . Then \tilde{J}_{ij} can be expanded as

$$\begin{aligned} \tilde{J}_{ij} &= \mathbf{a}_n^\top(t_i) \tilde{\Phi}_1^\top \tilde{\Phi}_1 \mathbf{a}_n(t_j) + \cdots + \mathbf{a}_n^\top(t_i) \tilde{\Phi}_K^\top \tilde{\Phi}_K \mathbf{a}_n(t_j) \\ &= \sum_{k=1}^K \mathbf{a}_n^\top(t_i) \tilde{\Phi}_k^\top \tilde{\Phi}_k \mathbf{a}_n(t_j) \end{aligned} \quad (4.31)$$

where each term in the right hand side of (4.31) denotes the contribution of one point $x_k \in \mathbb{X}$ to \tilde{J} .

Define, for each point $x_k \in \mathbb{X}$, the $L \times L$ matrix $E(x_k)$ whose (i, j) -th entry is

$$E_{ij}(x_k) := \tilde{J}_{ij} - \mathbf{a}_n^\top(t_i) \tilde{\Phi}_k^\top \tilde{\Phi}_k \mathbf{a}_n(t_j). \quad (4.32)$$

Then, for $k = 1, \dots, K$, define e_{x_k} by setting:

$$e_{x_k} := \| E(x_k) \| \quad (4.33)$$

where the norm $\|E(x_k)\|$ is defined as

$$\|E(x_k)\| = \sum_{i=1}^N \sum_{j=1}^N E_{ij}(x_k)^2$$

with $E_{ij}(x_k)$ is the element of the matrix $E(x_k)$ at the i -th row and j -th column.

The point with the lowest e_{x_k} is the most representative point because it can approximate \tilde{J} better than the other points. Let us re-index the points in \mathbb{X} as x_{k_1}, \dots, x_{k_K} such that

$$e_{x_{k_1}} \leq e_{x_{k_2}} \leq \dots \leq e_{x_{k_K}}$$

After e_{x_k} has been ordered in ascending order, the number of points has to be chosen until the condition number of $M = \tilde{\Phi}^\top \tilde{\Phi}$ is close to 1. Condition number of 100 is usually the threshold [35]. Otherwise, the solution of (4.15) is not unique.

4.4 Nonlinear heat conduction model

In this chapter, only the implementation of selection criterion 1 is going to be shown for a nonlinear, two dimensional heat conduction model. The spatial domain \mathbb{X} for the nonlinear heat conduction model of a thin plate is depicted in Figure 4.1. The thickness of the plate is 1 cm.

The nonlinear heat conduction model is given by:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) \quad (4.34)$$

where $T(x, t)$ is the temperature distribution of the plate, κ is the heat conductivity, and the actuators are defined as the incoming heat fluxes from the west boundary and the north temperature boundary.

In (4.34), the heat conductivity κ is temperature dependent and formulated as:

$$\kappa = 1.059e^{(-9.8852T)} + 9.867 \times 10^{-3}T^3 \quad (4.35)$$

The model is thus originally a nonlinear PDE-based model since the heat conductivity is temperature dependent. To simulate (4.34), the computational

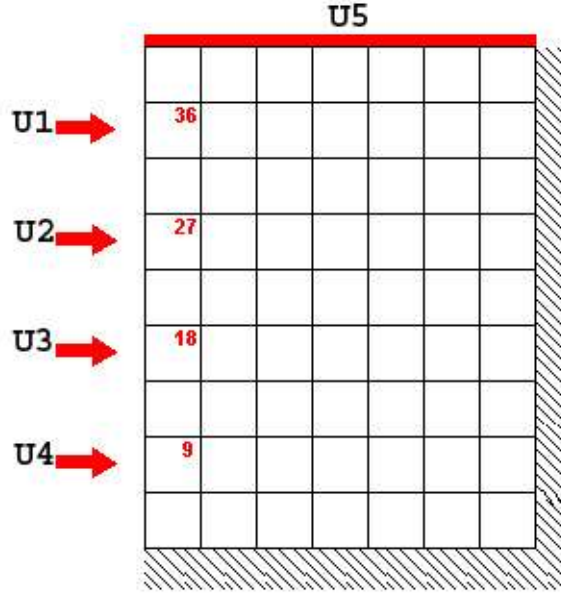


Figure 4.1: Computational Domain \mathbb{X} of the Heat Conduction Model. The plate is divided into 33 grid cells in the horizontal x direction and 44 grid cells. There are five manipulating variables, u_1 to u_5 are the incoming heat fluxes from the west side and u_5 is the north temperature

domain \mathbb{X} is discretised into 44 orthogonal grid cells in the y (vertical) direction and 33 orthogonal grid cells in the x (horizontal) direction. The grid dimension is $\Delta X = 0.091$ m and $\Delta Y = 0.091$ m.

The method used to discretise (4.34) is Computational Fluid Dynamics Finite Volume Method [70], [54].

In Finite Volume Method, (4.34) is integrated for every grid cell over a specified time horizon and a finite grid cell volume $\Delta V = \Delta x \times \Delta y \times 1$ cm. The algorithm is implemented in MATLAB, but generally CFD commercial packages which use Finite Volume also implement the same code. For algorithm details please refer to [70],[54].

$$\int_t^{t+\Delta t} \int_{\Delta V} \rho c_p \frac{\partial T}{\partial t} dV dt = \int_t^{t+\Delta t} \int_{\Delta V} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) dV dt + \int_t^{t+\Delta t} \int_{\Delta V} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) dV dt \quad (4.36)$$

Let $P = (x_l, y_m)$ be a position of a grid point. Let $T_P(k)$ denote the temper-

ature at time step k at the position P .

Let also the neighboring positions

$$N = (x_l, y_{m+1}), \quad E = (x_{l+1}, y_m), \quad S = (x_l, y_{m-1}), \quad W = (x_{l-1}, y_m)$$

denote the grid points on the north, east, south, west neighboring points of P .

If the temperature at specific grid point and at the future time step is denoted by $T_P(k+1)$, then based on (4.36), the discretised equation can be written as:

$$\begin{aligned} \rho c_p \frac{\Delta x \Delta y}{\Delta t} T_P(k+1) &= \rho c_p \frac{\Delta x \Delta y}{\Delta t} T_P(k+1) \\ &+ \frac{k_E(k+1)AT_E(k+1) - k_P(k+1)AT_P(k+1)}{\delta_{PE}} \\ &+ \frac{k_P(k+1)AT_P(k+1) - k_W(k+1)T_W(k+1)}{\delta_{PW}} \\ &+ \frac{k_N(k+1)AT_N(k+1) - k_P(k+1)AT_P(k+1)}{\delta_{PN}} \\ &+ \frac{k_P(k+1)AT_P(k+1) - k_S(k+1)AT_S(k+1)}{\delta_{PS}} \\ &+ S_u(k)\Delta V \mathbf{u}(k) \end{aligned} \quad (4.37)$$

where E, W, N, S denote the eastern, western, northern, and southern neighboring grid cells and $\delta_{PE}, \delta_{PW}, \delta_{PN}, \delta_{PS}$ are the distances from a particular grid cell to its eastern, western, northern, and southern grid cells, respectively. For every grid point, S_u is a vector of dimension n_u with n_u as the number of actuators. The actuators are here the western boundary heat fluxes and the north temperature boundary.

Recursive formulation of the discretised equation is shown in (4.38), where all the unknown temperatures of the whole grid cells at time step $k+1$ are collected within $\mathbf{T}(k+1) \in \mathbb{R}^{1452}$ and the coefficients are also functions of the temperature itself due to the dependency of the conductivity constant. If the time step is chosen sufficiently small, the conductivity constant at specified grid point $k_P(k+1)$ doesn't differ too much from $k_P(k)$, and we can use conductivity constant at the current time step to evaluate the temperature distribution at the future time step. In this example, the time step Δt is chosen to be 25 seconds. It is assumed that this time step is small enough to assume that the heat conductivity is constant at the current time step and can be used to evaluate the temperature in the future time step.

With this assumption, (4.37) can be solved simultaneously by representing the

temperatures as a vector

$$\mathbf{T}(k) = \begin{pmatrix} T_1(k) \\ T_2(k) \\ \dots \\ T_{1452}(k) \end{pmatrix}$$

The solutions $\mathbf{T}(k+1)$ is found by solving (4.38):

$$\mathbf{A}(k)\mathbf{T}(k+1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4.38)$$

where $\mathbf{A}(k) \in \mathbb{R}^{K \times K}$ is a penta diagonal matrix with the central, northern, southern, western, and eastern coefficients in the associated diagonal part.

$$\mathbf{A}(k) = \begin{pmatrix} a_P(k) & -a_E(k) & 0 & \dots & -a_N(k) & -a_S(k) & 0 & \dots \\ -a_W(k) & a_P(k) & -a_E(k) & 0 & \dots & -a_N(k) & -a_S(k) & \ddots \\ \dots & \dots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \dots & \dots & \ddots & \ddots & \ddots & \ddots & \ddots & a_P(k) \end{pmatrix}$$

The matrix $\mathbf{A}_0(k) \in \mathbb{R}^{1452 \times 1452}$ is a diagonal matrix with a_P^0 in its diagonal part. The matrix $\mathbf{B}(k) \in \mathbb{R}^{1452 \times n_u}$ is a matrix for input signals with n_u is the number of inputs.

The matrix $\mathbf{B}(k)$ comprises the S -term of every grid point as in (4.37):

$$\mathbf{B}(k) = \begin{pmatrix} S_{u_1} \\ S_{u_2} \\ \dots \\ S_{u_{1452}} \end{pmatrix}$$

where S_j denotes the S -term from j -th grid point.

The input signals of all actuators are collected in a vector $\mathbf{u} \in \mathbb{R}^5$:

$$\mathbf{u} = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \dots \\ u_5(t) \end{pmatrix}$$

Equation (4.38) is the numerical model for the nonlinear 2D heat conduction model and represents a discrete Linear Time Varying (LTV) system. The

matrices are written in bold symbols as the coefficients become functions of the temperature. At every time step, the matrices are updated to accommodate the temperature dependencies of the heat conductivity. The calculation as in (4.38) is the calculation solved by CFD modeling packages which employ the Finite Volume Method for nonlinear cases. Note that aside from finding the solutions, the model also has to be updated and the update also contributes to the overall CPU time needed for finding the solutions.

4.5 POD-MPE model

The recursive equation (4.38) is the basis for the numerical model. From the simulations of (4.38) a reduced order model will be derived. The simulation is conducted for 300 time steps with the excitations of north temperature at 70°C and step changes of the western heat fluxes to 100kW/m^2 . Initially at $t = 0$, the temperature of the plate is uniform at 0°C . Snapshots are collected in a matrix $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{1452 \times 400}$.

Algorithm 2.4.10 is conducted to derive the POD basis functions $\{\varphi_i\}_{i=1}^n$. The simulation data for 300 time steps is collected into $\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \ \dots \ \mathbf{T}(300))$. The POD basis functions are the eigenvectors of $\mathbf{C} = \frac{1}{300} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$.

The eigenvalue spectrum of \mathbf{C} is shown in Figure 4.2.

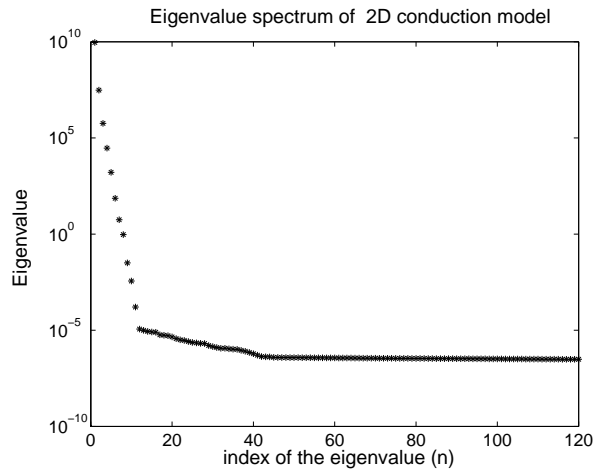


Figure 4.2: Eigenvalue spectrum obtained from the set of snapshots collected from the nonlinear heat conduction simulation

Six POD basis functions corresponding to 6 largest eigenvalues are chosen, thus $\Phi = (\varphi_1 \dots \varphi_6)$. Beyond 156th eigenvalue, the eigenvalues are zero. This corresponds to

$$1 - P_6 = 1 - \frac{\sum_{i=1}^6 \lambda_i}{\sum_{i=1}^{156} \lambda_i} = 7.8 * 10^{-9}$$

Since P_6 is quite close to 1, six POD basis functions are considered adequate to construct the reduced order model. Hence the POD basis Φ now comprises six basis functions:

The POD-based reduced order model is derived as the projection of (4.38) onto a set of orthonormal basis functions Φ . The original variable $\mathbf{T}(k)$ is approximated by:

$$\mathbf{T}_6(k) = \sum_{i=1}^6 a_i(k) \varphi_i$$

The original model are given by :

$$\mathbf{A}(k)\mathbf{T}(k+1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}\mathbf{u}(k) \quad (4.39)$$

Substituting $\mathbf{T}(k+1)$ by \mathbf{T}_6 and projecting Φ onto (4.39) yields:

$$\underbrace{\Phi^T \mathbf{A}(k) \Phi}_{\mathbf{A}_r(k)} \mathbf{a}(k+1) = \underbrace{\Phi^T \mathbf{A}_0(k) \Phi}_{\mathbf{A}_{0r}(k)} \mathbf{a}(k) + \underbrace{\Phi^T \mathbf{B}}_{\mathbf{B}_r(k)} \mathbf{u}(k) \quad (4.40)$$

$$\mathbf{A}_r(k) \mathbf{a}(k+1) = \mathbf{A}_{0r}(k) \mathbf{a}(k) + \mathbf{B}_r(k) \mathbf{u}(k) \quad (4.41)$$

where the matrices \mathbf{A} , \mathbf{A}_0 , \mathbf{B} are also functions of the temperature \mathbf{T} .

From (4.40), the reduced order model needs to be updated at every time step and in fact still a function of the whole temperature field, as $\mathbf{A}_r(k)$ and $\mathbf{B}_r(k)$ are temperature-dependent. The reduced matrices can also only be formed once the original matrices are available. This process is generally quite expensive, updating process for the original matrices may consume 50% or more of the CPU time necessary to obtain the solutions. The reduction of dimension saves the computing time needed to compute the inverse of the CFD matrices, however if the updating process is still a bottleneck, drastic decrement in the dimension will not influence the overall computing performance.

To accelerate the computation, the Missing Point Estimation is implemented. Suppose a subset of points from $\mathbf{T}(k) \in \mathbb{R}^K$ has been chosen as $\tilde{\mathbf{T}} \in \mathbb{R}^G$, where

$G < K$. In turn, this subset can also be approximated by the expansion in POD basis functions:

$$\tilde{\mathbf{T}}(k) \approx \sum_{i=1}^n \tilde{a}_i(k) \tilde{\varphi}_i(k)$$

Note that the POD basis functions used to approximate $\tilde{\mathbf{T}}(k)$ are also subsets of the original, complete POD basis functions φ_i .

Derivation of the reduced order model based on $\tilde{\mathbf{T}}(k)$ follows analogously as in the case of complete data. First write the original model in terms of $\tilde{\mathbf{T}}(k)$:

$$\begin{aligned} \tilde{\mathbf{A}}_P(k) \tilde{\mathbf{T}}(k+1) = & \tilde{\mathbf{A}}_0(k) \tilde{\mathbf{T}}(k) + \tilde{\mathbf{A}}_W(k+1) \tilde{\mathbf{T}}_W(k+1) + \tilde{\mathbf{A}}_E(k+1) \tilde{\mathbf{T}}_E(k+1) \\ & + \tilde{\mathbf{A}}(k)_N \tilde{\mathbf{T}}(k+1)_N + \tilde{\mathbf{A}}_S(k) \tilde{\mathbf{T}}_S(k+1) + \tilde{\mathbf{B}}\mathbf{u}(k) \end{aligned}$$

where $\tilde{\mathbf{T}}_W, \tilde{\mathbf{T}}_E, \tilde{\mathbf{T}}_N, \tilde{\mathbf{T}}_S$ are the corresponding western, eastern, northern, and southern grid points of $\tilde{\mathbf{T}}$. The matrices $\tilde{\mathbf{A}}_P, \tilde{\mathbf{A}}_W, \tilde{\mathbf{A}}_E, \tilde{\mathbf{A}}_N, \tilde{\mathbf{A}}_S$ are all diagonal matrices of dimension $G \times G$ with the collections of a_P, a_W, a_E, a_N, a_S in its diagonal part, respectively.

The neighboring points can also be approximated by the POD expansions:

$$\tilde{\mathbf{T}}_W(k+1) = \tilde{\Phi}_W \tilde{\mathbf{a}}(k+1) \quad (4.42)$$

$$\tilde{\mathbf{T}}_E(k+1) = \tilde{\Phi}_E \tilde{\mathbf{a}}(k+1) \quad (4.43)$$

$$\tilde{\mathbf{T}}_N(k+1) = \tilde{\Phi}_N \tilde{\mathbf{a}}(k+1) \quad (4.44)$$

$$\tilde{\mathbf{T}}_S(k+1) = \tilde{\Phi}_S \tilde{\mathbf{a}}(k+1) \quad (4.45)$$

where $\tilde{\Phi}_W, \tilde{\Phi}_E, \tilde{\Phi}_N, \tilde{\Phi}_S$ are the subsets of POD basis Φ corresponding to the elements of $\tilde{\mathbf{T}}_W, \tilde{\mathbf{T}}_E, \tilde{\mathbf{T}}_N, \tilde{\mathbf{T}}_S$, respectively.

The POD coefficients based on the incomplete data can be found by solving (4.15):

$$M \tilde{\mathbf{a}}(k+1) = \tilde{\Phi}^\top f$$

where $M = \tilde{\Phi}^\top \tilde{\Phi}$.

The equation 4.15 can also be implemented in (4.42) by pre-multiplying both sides of (4.42) by $\tilde{\Phi}^\top$. The variable f in (4.15) is the right-hand side of (4.42). The pre-multiplication yields:

$$\begin{aligned} \tilde{\Phi}^\top \tilde{\mathbf{A}}_P(k) \tilde{\mathbf{T}}(k+1) = & \tilde{\Phi}^\top \tilde{\mathbf{A}}_0(k) \tilde{\Phi}^\top \tilde{\mathbf{T}}(k) + \tilde{\Phi}^\top \tilde{\mathbf{A}}_W(k+1) \tilde{\mathbf{T}}_W(k+1) \\ & + \tilde{\Phi}^\top \tilde{\mathbf{A}}_E(k+1) \tilde{\mathbf{T}}_E(k+1) + \tilde{\Phi}^\top \tilde{\mathbf{A}}(k)_N \tilde{\mathbf{T}}(k+1)_N + \tilde{\Phi}^\top \tilde{\mathbf{A}}_S(k) \tilde{\mathbf{T}}_S(k+1) \\ & + \tilde{\Phi}^\top \tilde{\mathbf{B}}\mathbf{u}(k) \end{aligned}$$

Replacing the term $\tilde{\mathbf{T}}(k+1)$ and its neighboring counterparts with the POD expansions yields

$$\begin{aligned} \tilde{\Phi}^\top \tilde{\mathbf{A}}_P(k) \tilde{\Phi} \mathbf{a}(k+1) &= \tilde{\Phi}^\top \tilde{\mathbf{A}}_0(k) \tilde{\Phi} \mathbf{a}(k) + \tilde{\Phi}^\top \tilde{\mathbf{A}}_W(k+1) \tilde{\Phi}_W \mathbf{a}(k+1) \\ &+ \tilde{\Phi}^\top \tilde{\mathbf{A}}_E(k+1) \tilde{\Phi}_E \tilde{\mathbf{a}}(k+1) + \tilde{\Phi}^\top \tilde{\mathbf{A}}(k)_N \tilde{\Phi}_N \tilde{\mathbf{a}}(k+1) + \tilde{\Phi}^\top \tilde{\mathbf{A}}_S(k) \tilde{\Phi}_S \tilde{\mathbf{a}}(k+1) \\ &+ \tilde{\Phi}^\top \tilde{\mathbf{B}} \mathbf{u}(k) \end{aligned}$$

Collect the terms with $\tilde{\mathbf{a}}(k+1)$ on the left hand side and the following reduced order model is obtained:

$$\tilde{\Phi}^\top \left(\tilde{\mathbf{A}}(k) \right) \tilde{\mathbf{a}}(k+1) = \tilde{\Phi}^\top \tilde{\mathbf{A}}_0(k) \tilde{\Phi} \mathbf{a}(k) + \tilde{\Phi}^\top \tilde{\mathbf{B}} \mathbf{u}(k) \quad (4.46)$$

where $\tilde{\mathbf{A}}(k) \in \mathbb{R}^{G \times n}$ is formulated as:

$$\tilde{\mathbf{A}}(k) = \tilde{\mathbf{A}}_P(k) \tilde{\Phi} - \tilde{\mathbf{A}}_W(k+1) \tilde{\Phi}_W - \tilde{\mathbf{A}}_E(k+1) \tilde{\Phi}_E - \tilde{\mathbf{A}}(k)_N \tilde{\Phi}_N \tilde{\mathbf{A}}_S(k) \tilde{\Phi}_S$$

The final reduced order model based on POD-MPE is written in compact form as:

$$\begin{aligned} \underbrace{\tilde{\Phi}^\top \left(\tilde{\mathbf{A}}(k) \right)}_{\tilde{\mathbf{A}}_r(k)} \tilde{\mathbf{a}}(k+1) &= \underbrace{\tilde{\Phi}^\top \tilde{\mathbf{A}}_0(k) \tilde{\Phi}}_{\tilde{\mathbf{A}}_{0r}(k)} \mathbf{a}(k) + \underbrace{\tilde{\Phi}^\top \tilde{\mathbf{B}}}_{\tilde{\mathbf{B}}_r(k)} \mathbf{u}(k) \\ \tilde{\mathbf{A}}_r(k) \tilde{\mathbf{a}}(k+1) &= \tilde{\mathbf{A}}_{0r}(k) \mathbf{a}(k) + \tilde{\mathbf{B}}_r(k) \mathbf{u}(k) \end{aligned} \quad (4.47)$$

where $\tilde{\mathbf{A}}_r(k) \in \mathbb{R}^{n \times n}$, $\tilde{\mathbf{A}}_{0r} \in \mathbb{R}^{n \times n}$, $\tilde{\mathbf{B}} \in \mathbb{R}^{n \times n_u}$. Note that these matrices are constructed from the CFD equations of G points, not K points, with G is the number of points selected to be the mask $\tilde{\mathbf{T}}(k)$. In the end n POD coefficients have to be estimated, equivalent to the case of the original POD model, but the reduced order model is built based on less number of points. Hence, only model parameters such as the coefficients of $a_P, a_W, a_E, a_N, a_S, a_P^0$ at the G points have to be updated.

Once $\tilde{\mathbf{a}}(k+1)$ has been found, the other points which are not in the mask can be calculated using the complete POD basis functions as:

$$\hat{\mathbf{T}}(k+1) = \hat{\Phi} \tilde{\mathbf{a}}(k+1) \quad (4.48)$$

where $\hat{\mathbf{T}}(k+1) \in \mathbb{R}^{K-G}$ refers to the points which are not selected in the mask $\tilde{\mathbf{T}}$.

In this section, an POD-MPE reduced order model has been derived. The reduced model is constructed based on a partial set of the original data. The objective is to accelerate the computational process of reduced order models in the case of constantly updated model parameters. Varying model parameter is the characteristic of numerical CFD models when the physical parameters are functions of the process variables themselves.

4.6 Point selection in two dimensional heat conduction model

In section 4.3.1 and section 4.3.2, two selection criteria of the representative points in $\tilde{\mathbf{T}}$ have been proposed. For this chapter, only criterion 1 will be used. In Chapter 5, both criteria will be used.

Aside from finding the representative points, it is important to note that the POD-MPE reduced order model (4.47) is a dynamic model. It is important to accommodate the appropriate boundary conditions similar as in the case of the conventional POD based reduced order model. Hence, before any of the criteria of point selections are imposed, there are some obligatory points which have to be taken so that the changing boundary conditions such as the changing heat fluxes and the north temperature will be incorporated.

It is imperative to include points adjacent to control inputs to accommodate changes in manipulated variables. In this case, the control inputs are western and northern boundaries. Further, there is boundary condition applied to the isolated eastern and southern boundaries, and the information of the zero temperature gradient across these boundaries will be lost if the points adjacent to them are excluded. Hence, in the first step, we take points which have direct connection to the boundary conditions and control inputs. Figure 4.3 depicts grid points that belong to this category marked by the bold frame. There are 150 points which are adjacent to the boundary condition. The rest of the points are selected based on criterion 1 (section 4.3.1).

For every remaining point, $e_{\mathbb{X}_0}$ (4.24) is calculated by setting $\mathbb{X}_0 = x_k$, hence \mathbb{X}_0 consists of only one point. Thus $e_{\mathbb{X}_0} = e_{x_k}$. After e_{x_k} has been calculated for every remaining point, then the quantity is ordered from small to large. Small $e_{\mathbb{X}_0}$ indicates that the point is important while large $e_{\mathbb{X}_0}$ indicates that the point is not important. Figure 4.4 shows the plot of ordered $e_{\mathbb{X}_0}$ from small to large.

The number of points have to be chosen such that the condition number of

$$M = \tilde{\Phi}^T \tilde{\Phi}$$

is small (below 100). The condition number Θ is calculated as:

$$\Theta = \frac{\max \lambda(M)}{\min \lambda(M)}$$

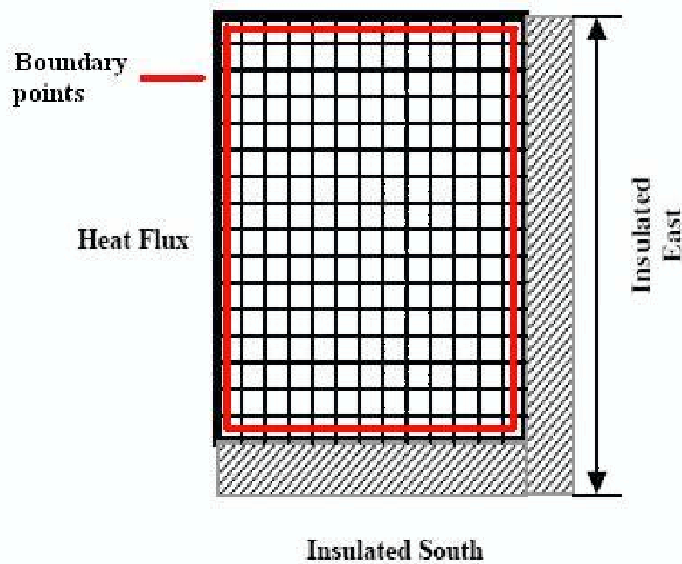


Figure 4.3: Selected boundary points, marked by the bold frame

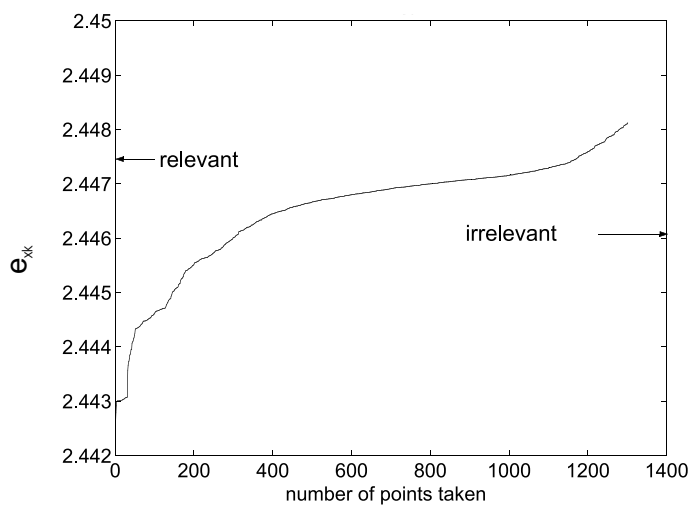


Figure 4.4: Ordered $e_{x_0} = e_{x_k}$ (4.24) for the remaining points outside the 150 obligatory points

A low condition number indicates that the points chosen are relatively independent of each other.

Figure 4.5 shows the resulted condition number of M comprises 150 obligatory points and plus the extra points selected from the ordering of $e_{\mathbb{X}_0}$. From Figure 4.5, the condition number goes asymptotically to 1 after 800 extra points have been taken. This is quite a lot since we will only halve the updating process. The highest condition number is in the order of 20, and this is actually quite good since it is well below 100.

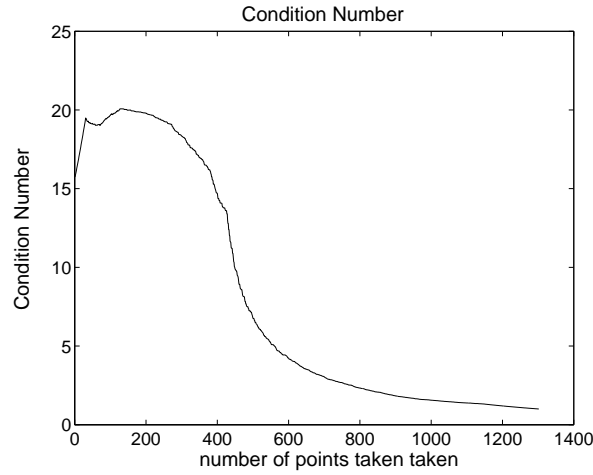


Figure 4.5: The condition number of M constructed from the 150 obligatory points and the extra points taken based on the criterion of $e_{\mathbb{X}_0}$. The x-axis shows the number of extra points taken

To decide how many number of extra points aside from the obligatory points are taken, some experiments are done by taking the first 50, 100, 150, 200, 250, and 300 extra points, thus sampling between 0 to 300 extra points.

Figure 4.6 shows the plot of the maximum absolute error between the simulation of POD-MPE model constructed from partial data set and the original model. The POD-MPE model is simulated by employing the same initial conditions and boundary conditions when snapshots data is generated.

The maximum error is calculated as:

$$\epsilon_{\max} = \max \| \mathbf{T}(k) - \Phi \tilde{\mathbf{a}}(k) \|, \quad k = 1, \dots, 300$$

The error plot as in Figure 4.6 drops at the 200-th extra points. The maximum

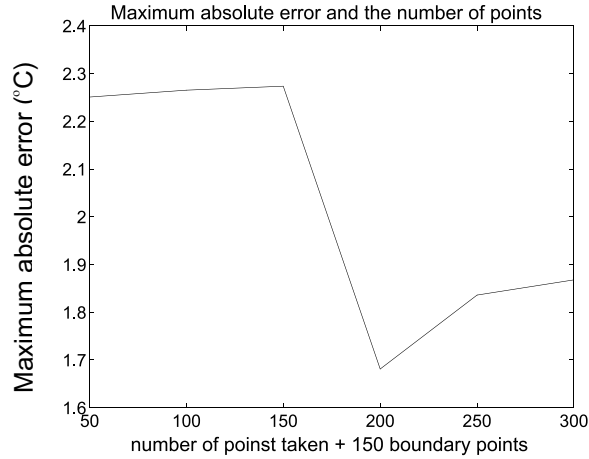


Figure 4.6: The maximum absolute error from the original model. The x-axis shows the number of extra points used aside from the 150 obligatory points to construct the POD-MPE model

error is quite small as the temperature range of the plate lies between 0 to 100°C. It is observed that between 50 extra points and 200 points the error only differs for about 0.5°C. Plot of the CPU time in Figure 4.7 shows the CPU time needed to solve the POD-MPE model by taking the obligatory and the extra points. The CPU time is computed for the whole 300 time steps. Comparing Figure 4.7 and Figure 4.6, 50 extra points are taken to construct the POD-MPE model. Hence, in total, the reduced order model is constructed by 200 points.

Table 4.1 enlists the comparisons of the simulation time in second and the maximum absolute error between the POD-MPE model, the conventional POD model and the original model.

Table 4.1: Comparison between POD and POD-MPE models

| Model Type | Maximum Absolute Error | CPU time (s) |
|-------------|------------------------|--------------|
| Original | 0° C | 7200 |
| POD | 0.35° C | 600 |
| POD-MPE-200 | 2.25° C | 42 |

The original model is slightly faster than the real time, since for 300 time steps with $\Delta t = 25$ s, it simulates in 7200 seconds. The conventional POD model built by constantly updating the reduced order model is 12.5 times faster than real time but more acceleration factor is required to be able to apply optimal

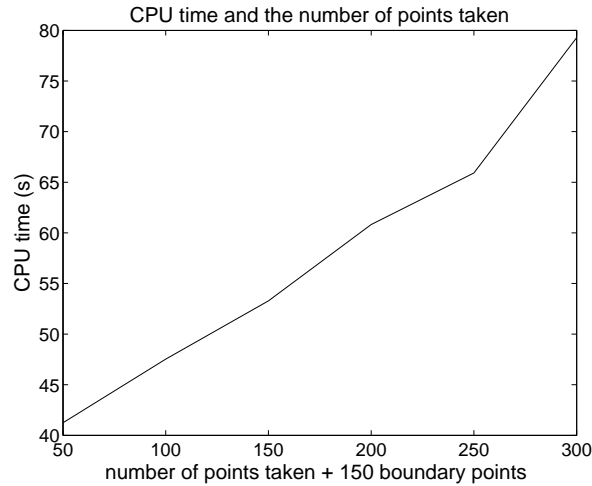


Figure 4.7: The CPU time (in seconds) of the time needed to solve the POD-MPE model. The x-axis shows the number of extra points used aside from the 150 obligatory points to construct the POD-MPE model

control design with long prediction horizon. The POD-MPE model is more than 175 times faster than the real time, it still incorporates the nonlinearities but only at selected points and give reasonable deviation from the original model.

The plots of the most dominant basis vector (the one correspond to the largest singular value) of the complete data and the incomplete data are shown in Figure 4.8. At the locations where the points are not selected, the basis vector component is zero. The time varying deviation from original model with conventional POD model and the POD-MPE is shown in Figure 4.9 and 4.3. The maximum temperature deviation when we only use limited data is quite reasonable, accounts to about 2°C, which is about 1% of the plate temperature range which lies in the 0°C to 100°C. Indeed the approximation by classic POD is more superior, but this is compensated by a lot more computing effort.

4.7 Controller design

With the available fast model, a model based controller can be designed. The control objective is to achieve a desired temperature distribution \mathbf{T}_{ref} optimally. The optimal control problem is cast as:

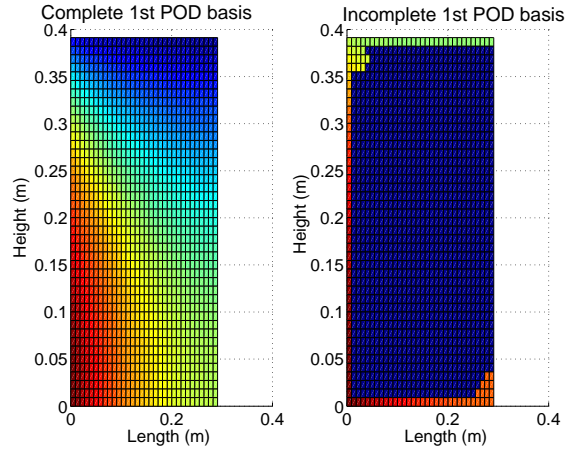


Figure 4.8: Left: most dominant basis vector with complete data, right: most dominant basis vector with incomplete data

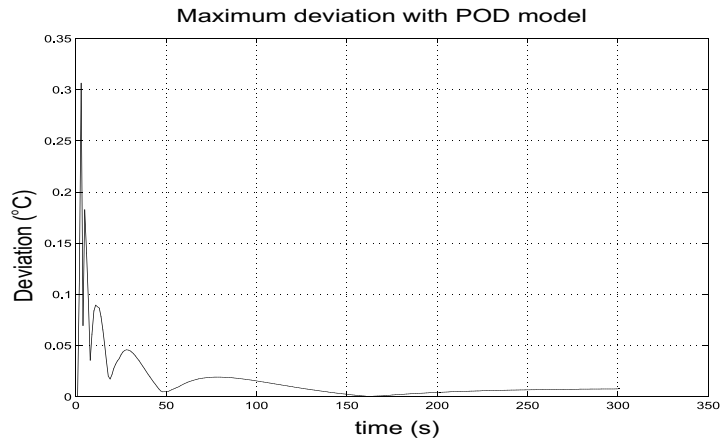


Figure 4.9: Maximum deviation of every time step of reduced order model by classic POD

Given the temperature $T_{\text{ref}}(x, t)$, with $x \in \mathbb{X}$ and $t \in \mathbb{T}$, find a control input $\mathbf{u}(t) \in \mathcal{U}$ where \mathcal{U} is the class of allowable input signal such that the squared L_2 -norm of the tracking error

$$\sum_{k=0}^{N_{\text{opt}}} \sum_{\mathbb{X}} \| T_{\text{ref}}(x, k) - T(x, k) \|^2$$

is minimized where is N_{opt} the length of optimization horizon and \mathbb{X} is the spatial domain.

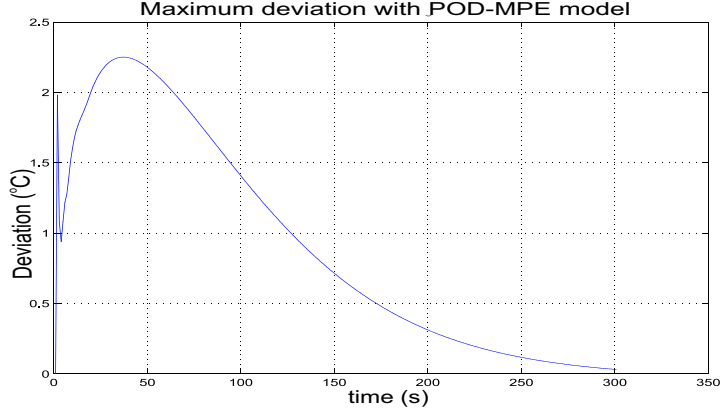


Figure 4.10: Maximum deviation of every time step of reduced order model by POD-MPE

The optimal input signal is thus $\mathbf{u}(t)$ which solves the following optimization problem:

$$\mathbf{u}(t) = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{k=0}^{N_{\text{opt}}} \sum_{\mathbb{X}} \| T_{\text{ref}}(x, k) - T(x, k) \|^2$$

The optimization problem is difficult to solve if \mathbf{T}_{ref} is high dimensional, that is $\dim(\mathbf{T}) > 200$. This difficulty is solved by introducing the reduced order model. In reduced order model, \mathbf{T}_{ref} can also be expanded by the n -th order POD expansion. In this example, we have chosen six POD basis functions, so $n = 6$.

$$\mathbf{T}_{\text{ref}}(k) \approx \sum_{i=1}^6 a_i(t) \varphi_i = \Phi \mathbf{a}_{\text{ref}}(k) \quad (4.49)$$

As the POD basis functions are orthonormal, $\Phi^{\top} \Phi$ equals identity matrix, so \mathbf{T}_{ref} can be translated as the desired POD coefficients

$$\mathbf{a}_{\text{ref}}(k) = \Phi^{\top} \mathbf{T}_{\text{ref}}(k) \quad (4.50)$$

With \mathbf{a}_{ref} defined as in (4.50), the control objective for the reduced order model becomes:

Given the desired POD coefficients $\mathbf{a}_{\text{ref}}(k)$, find a control input

$u(t) \in \mathcal{U}$ such that the squared L2-norm of the tracking error

$$\sum_{k=0}^{N_{\text{opt}}} e(k) = \| \mathbf{a}_{\text{ref}}(t) - \mathbf{a}(t) \|^2$$

is minimized.

The optimal control problem is modified then for $\mathbf{u}(t)$ as:

$$\mathbf{u}(k) = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{k=0}^{N_{\text{opt}}} \| \mathbf{a}_{\text{ref}}(t) - \mathbf{a}(k) \|^2$$

For model based control design, a fast model is necessary so that the predictions of the future process variables can be incorporated in the optimization module. The fast model is provided by the POD-MPE based model which provides the estimates of the POD coefficients based on the incomplete data. The optimal control problem is defined as:

$$\mathbf{u}(t) = \arg \min_{\mathbf{u} \in \mathcal{U}} \sum_{k=0}^{N_{\text{opt}}} \| \mathbf{a}_{\text{ref}}(k) - \tilde{\mathbf{a}}(k) \|^2$$

Consider again the POD-MPE model equation (4.47), the explicit state-space model can be derived from (4.47):

$$\tilde{\mathbf{A}}_r(k) \tilde{\mathbf{a}}(k+1) = \tilde{\mathbf{A}}_{0r}(k) \tilde{\mathbf{a}}(k) + \tilde{\mathbf{B}}_r(k) \mathbf{u}(k) \quad (4.51)$$

$$\tilde{\mathbf{a}}(k+1) = \underbrace{\tilde{\mathbf{A}}_r^{-1}(k) \tilde{\mathbf{A}}_{0r}(k)}_{\tilde{\mathbf{A}}_s(k)} \tilde{\mathbf{a}}(k) + \underbrace{\tilde{\mathbf{A}}_r^{-1}(k) \tilde{\mathbf{B}}_r(k)}_{\tilde{\mathbf{B}}_s(k)} \mathbf{u}(k) \quad (4.52)$$

$$\tilde{\mathbf{a}}(k+1) = \mathbf{A}_s(k) \tilde{\mathbf{a}}(k) + \mathbf{B}_s(k) \mathbf{u}(k) \quad (4.53)$$

There are six measurement points in the heated plate, all measurement points are collected in $\mathbf{y} \in \mathbb{R}^6$. The equation for \mathbf{y} is:

$$\begin{aligned} \mathbf{y}(k) &= C\mathbf{T}(k) = \underbrace{C\Phi}_{C_s} \tilde{\mathbf{a}}(k) \\ &= C_s \tilde{\mathbf{a}}(k) \end{aligned}$$

where $C \in \mathbb{R}^{1452}$ is a row vector with 1 at the locations of the measurement points and zeros otherwise.

The locations six measurement points for the two dimensional plate is shown in Figure 4.11.

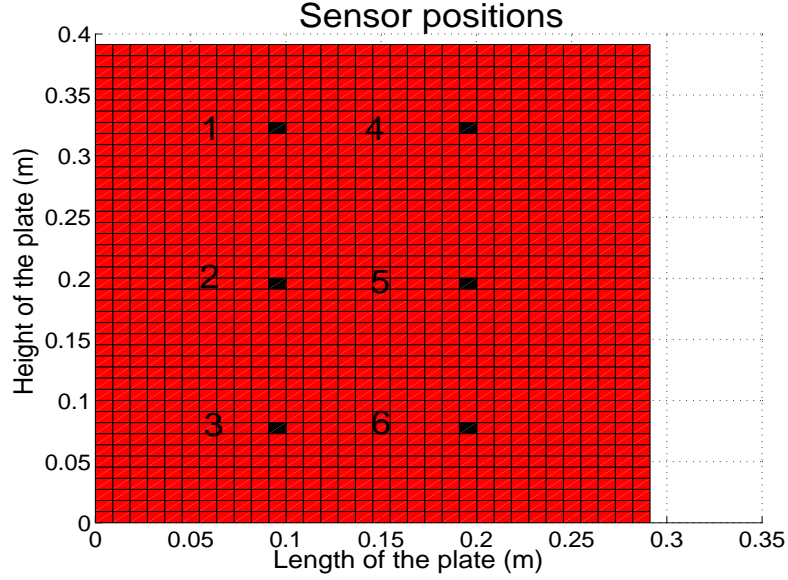


Figure 4.11: The locations of the measurement points, denoted by numbers 1 to 6

Consider the POD coefficients $\tilde{\mathbf{a}}(k)$ as the *states* of the system, hence $\mathbf{z}(k) = \tilde{\mathbf{a}}(k)$.

The state space model for (4.47) has the form:

$$\mathbf{z}(k+1) = \mathbf{A}_s(k)\mathbf{z}(k) + \mathbf{B}_s(k)\mathbf{u}(k) \quad (4.54)$$

$$\mathbf{y}(k+1) = C_s\mathbf{z}(k) \quad (4.55)$$

The implemented controller for this system is an Linear Quadratic Regulator with nonzero reference signals. The modified classic LQR objective function to be minimized then reads:

$$\begin{aligned} J(\mathbf{z}_0, \mathbf{u}) &= \sum_{k=0}^{N_{\text{opt}}-1} [(\mathbf{a}_{\text{ref}}(k) - \mathbf{z}(k))^T Q (\mathbf{a}_{\text{ref}}(k) - \mathbf{z}(k))] \\ &+ \mathbf{u}^T(k) R \mathbf{u}(k) \\ &+ \mathbf{z}(N_e)^T E [\mathbf{a}_{\text{ref}}(N_{\text{opt}}) - \mathbf{z}(N_{\text{opt}})] \end{aligned} \quad (4.56)$$

where N_{opt} is the length of the prediction horizon. In this example, $N_{\text{opt}} = 150$.

The optimal control input \mathbf{u}^* has to be found such that $J(\mathbf{z}_0, \mathbf{u}^*) \leq J(\mathbf{z}_0, \mathbf{u}) \forall \mathbf{u} \in \mathcal{U}$. The objective function J_o in (4.56) is a function of both the actual

POD coefficients $\mathbf{z}(k)$ and the desired POD coefficients \mathbf{a}_{ref} . Thus the optimal control input \mathbf{u}^* must also be a function of the actual POD coefficients (or the state) and the desired POD coefficients.

Such input, for an LTI system is found to be (the proof can be found in Appendix B)[28]

$$\mathbf{u}^* = - \underbrace{\mathbf{F}_z(k)\mathbf{z}(k)}_{\text{state feedback}} - \underbrace{\mathbf{F}_v(k)\mathbf{v}(k+1)}_{\text{function of reference signals}} \quad (4.57)$$

where

$$\begin{aligned} \mathbf{F}_z(k) &= \left(R + \mathbf{B}_s(k)^\top \mathbf{P}(k+1) \mathbf{B}_s(k) \right)^{-1} \mathbf{B}_s^\top(k) \mathbf{P}(k+1) \mathbf{A}_s(k) \\ \mathbf{F}_v(k) &= \left(R + \mathbf{B}_s^\top(k) \mathbf{P}(k+1) \mathbf{B}_s(k) \right)^{-1} \mathbf{B}_s^\top(k) \end{aligned}$$

The details about the \mathbf{P} and \mathbf{v} can be found in Appendix B. Since the state space matrices in this case are time varying due to the temperature dependent heat conductivity and the variations depend on the input signal \mathbf{u} , the designed controller is a *suboptimal* LQR controller. The control input $\mathbf{u}(k)$, is defined exactly as in (4.57), but $\mathbf{P}(k+1)$ and $\mathbf{v}(k+1)$ are *recalculated* using the state space matrices $\mathbf{A}_s(k)$ and $\mathbf{B}_s(k)$ at every time step.

The weighting matrices for the states $Q \in \mathbb{R}^{6 \times 6}$ and for the inputs $R \in \mathbb{R}^{6 \times 6}$ are defined as:

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} 10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 10^{-8} & 0 & 0 \\ 0 & 0 & 0 & 10^{-8} & 0 \\ 0 & 0 & 0 & 0 & 10 \end{pmatrix}$$

The steady state response with the LQR controller can be seen in Figure 4.12. The steady state deviation from the desired temperature is still reasonable which accounts for maximum 6°C for temperature variations of about 137°C in the plate and considering the fact that the model used as the reference for the controller is based on 200 data points only. This shows the capability of POD-MPE reduced model, which is computed very fast, to control a full order model with reasonable deviation.

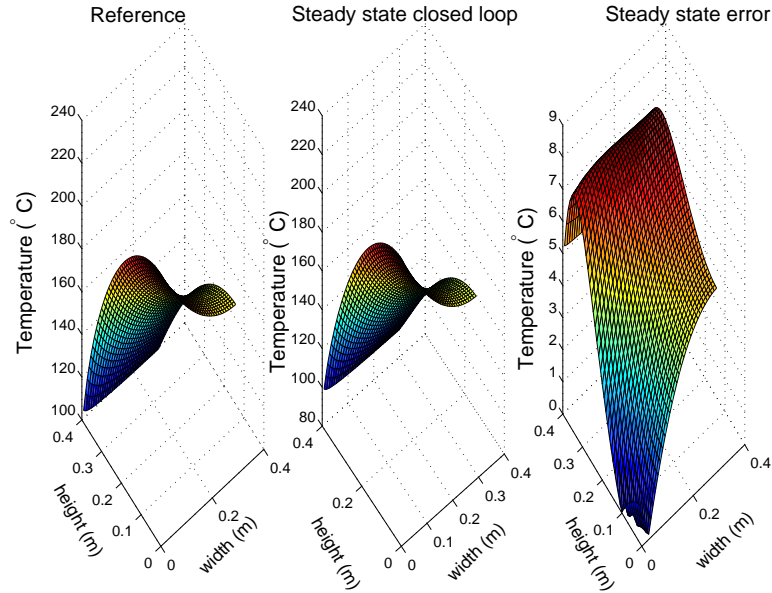


Figure 4.12: The desired temperature distribution (left), the temperature distribution of the plate controlled by the LQR controller (middle) and the steady state error (right)

4.8 Summary

Reduced order modeling by POD yields a model which comprises much less number of equations than the original model. However, it does not mean that the reduced order model will be significantly faster than the original model. The POD based reduced model is in principle built by combining the original model with the POD basis functions. In the case of nonlinear CFD models, the parameters of the original model are constantly updated and therefore the construction of the reduced order model is also still time-consuming.

An approach referred as the Missing Point Estimation is introduced in section 4.2. In MPE, the POD coefficients can also be estimated from an incomplete data of process variables. The original MPE approach which was introduced by Everson and Sirovich [22, 34] to approximate a static image is then expanded in this thesis for dynamic simulation in section 4.5 to accelerate the model reduction computation. Two point selection criteria have also been proposed as tools to find the representative grid points which predominate the dynamics of other grid points.

The MPE method is applied to a nonlinear heat conduction model, where the heat conductivity is temperature dependent. The resulted numerical model is of the type Linear Time Varying (LTV) system. It is shown in section 4.6 that the POD-MPE reduced order model can follow the dynamics of the original model reliably. The original model has 1452 states, the reduced model is built based on the information of 200 states only. A time varying LQR controller has also been designed with the POD-MPE reduced order model as the base model since it is faster than real time.

The POD-MPE reduced model is shown to be fast and reliable enough to be used as a base model for the controller design. This approach is a very promising approach to be elaborated further in the research of model reduction since most of the time, reduced order modeling of nonlinear or varying parameter models is still numerically unattractive.

Reduced Order Modeling of a Glass Melt Feeder

| | |
|---|---|
| 5.1 Introduction to glass furnace operation 5.2 The feeder model 5.3 Snapshot collection and POD reduction 5.4 Validation of the reduced order model | 5.5 Application of MPE to the glass melt feeder 5.6 Simulation of the glass colour change in the feeder 5.7 Summary and conclusions |
|---|---|

This chapter is an extended version from the papers [5],[6], [7]

5.1 Introduction to glass furnace operation

Since the second half of the nineteenth century, most glasses have been molten in continuously operating glass furnaces. In general, there are three main parts of the furnace, i.e the melting tank, the working end, and the feeder section or the foreheath section. A schematic figure of a glass furnace is given in Figure 5.1.

1. Melting tank

In this part, the mixture of raw material components, which form glass is molten. Here, the mixture is referred to as "glass forming batch". The batch is charged from the doghouse to the melting tank.

The glass forming batch is mainly heated by radiative heat, transferred from the flames and from the superstructure in the combustion chamber above the batch blanket [68].

During heating of the glass forming batch, up to the temperatures of $\pm 1750 - 1900$ K [68], a complex process of chemical reactions takes

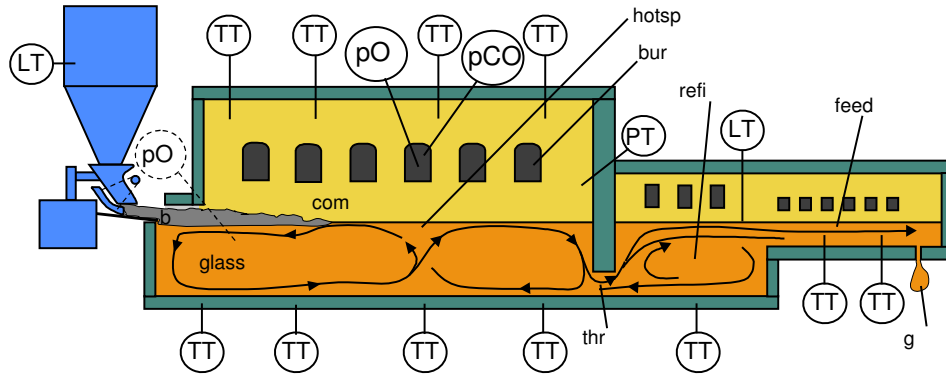


Figure 5.1: A sketch of glass furnace. Abbreviations in the figure refer to different parts of the furnace: combustion chamber (com), burners (bur), refiner (refi), feeder (feed), throat (thr). The sensors installed are thermocouples (TT) and pressure sensors of gases (pO for oxygen, pCO for carbon monoxide). The glass melt flows through a dosing mechanism at the end of the furnace, the droplet of molten glass is called gob (g)

place, which transforms the solid raw materials into a liquid melt phase with gaseous inclusions and dissolved gases. After a liquid melt phase is formed in the batch blanket, these melt phases enter the bulk flow of the glass melt in the tank. The melt follows several flow patterns through the melting tank towards the throat. The typical flow patterns result from the forced and free convection. The convection occurs due to the imposed pull rate, i.e. the amount of glass melt per period of time withdrawn from the furnace for producing glass products. The free convective flow is caused by the density gradients in the glass melt as a consequence of the existing temperature gradient or bubbling within the glass melt.

Apart from melting of the batch, the glass melting tank is also used to homogenize, fine, and refine the glass melt. Freshly formed glass melt still contains dissolved gases and gas bubbles. During the fining process, bubbles (mainly containing CO_2 , N_2 , O_2 , SO_2 , H_2O , etc) and a part of the dissolved gases are removed. At relatively high temperature, fining agents such as sodium sulfate and antimony oxide decompose and produce fining gases SO_2 and/or O_2 . The fining gases diffuse into the existing gas bubbles, the bubbles expand, rise (bubble ascension velocity is proportional to the radius of the bubbles), and escape towards the

surface of the glass melt. The fining gases will also strip other gases from the glass melt. In the refining process, remaining small bubbles, containing gases, which dissolve chemically during glass melt cooling, are reabsorbed in the glass melt. After these refining processes, the glass melt enters the working end via the throat of the melting tank. Part of the refining process may still take place in the working end.

2. Working end

The working end is a section between the glass melting tank and the feeder canals. The molten glass is brought into this section to be thermally conditioned and occasionally to be further homogenized or refined before brought to the feeder canals.

3. Feeder

The glass melt feeder is located between the refiner and the glass exit point or the spout. The basic geometry of a glass feeder is an almost rectangular shallow canal. Often the outlet passage is skewed, that it is narrower than the inlet part. In the glass melt feeder, the temperature of the glass melt is controlled before it is brought to the spout. In the feeder, the glass melt is slowly cooled down to maintain uniform temperature distribution at the exit. A well controlled temperature distribution is required for the forming process of the glass products.

Full automatization of glass making process in glass factories is still in a developing stage and full understanding of all relevant processes in glass furnaces has still not been achieved. In 1980s, The Netherlands Institute of Applied Physics of TNO developed a glass process simulator (mathematical model of the glass melt process), which is known under the name "TNO-Glass Tank Model" (GTM) software.

The glass simulating software packages are used to simulate various processes taking place in the furnace [52], [47]. The simulation results enable glass industries to evaluate their furnace designs, to study the influence of the physical parameters on the product quality, and to understand the physics and the chemistry of the glass melting and combustion processes ([9],[10],[17],[18]).

Since the operating temperatures of glass furnaces are quite high, not many sensors can be installed in the furnaces. Information from the existing physical sensors such as redox sensors, thermocouples, and pressure measurements is limited. Therefore it is difficult to estimate the ongoing process and to tune the process either online or off-line. The glass process simulator helps the operators and engineers to monitor the variables of the ongoing process

(flows, temperatures, chemical conversion rate, residence time) to estimate the relevant variables (temperature, pressure, chemical composition) inaccessible by physical sensors and use the estimation for trouble shooting or for analysis of the furnace performance.

The glass melt process simulator (GTM) is a Computational Fluid Dynamics model, discretised by CFD Finite Volume Method [70],[54]. The models for glass furnaces are described by a set of nonlinear partial differential equations (PDE) with boundary conditions, based on the mass, momentum and energy balances. Additionally, a number of non-linear functions of independent scalars are also incorporated into the model to describe the dependent properties like viscosity, heat conductivity, radiative property and density of the glass melt.

As demonstrated in the previous chapters, discretization of PDE-based models lead to high-order models. The glass furnace is usually discretised into 10^3 to 10^8 grid cells. The computational model of a glass furnace is complex as most of the physical parameters such as viscosities and conductivities are temperature dependent. The model takes considerable computational effort before it can provide reasonably accurate and consistent estimates of the the process states.

The simulation models provides fairly well estimations of the temperature, velocities, and chemical conversions. Some experiments have been conducted to validate the results of simulation models [2],[47]. This estimations of the process variables are needed by a controller to enhance the performance of the furnace such as optimizing the use of energy consumption because the data from the physical sensors is very limited. For good anticipation, the model must provide predictions of the process states faster than real time. Depending on the characteristics of the process, the model may have to be 50-100 times faster than real time. It is infeasible to achieve a good anticipatory action when the controller obtains information from a simulation model which is not fast enough.

In previous chapters, we have demonstrated the use of the POD method as a model reduction technique for process simulation models governed by PDE's. Every solution of PDE-based models is a function of a spatial position $x \in \mathbb{X}$ and time $t \in \mathbb{T}$, where \mathbb{X} and \mathbb{T} are the spatial domain (e.g a furnace) and time domain. So, in general (under some mathematical restrictions as discussed in section 2.4), we can represent the relevant variable such as temperature,

denoted by $T(x, t)$ as:

$$T(x, t) = \sum_{i=1}^n a_i(t) \varphi_i(x) \quad (5.1)$$

Assume that the spatial domain \mathbb{X} has K grid points and by stacking $T(x, t)$ over all spatial domain at a particular time step, define

$$\mathbf{T}(t) := \operatorname{col}_{x \in \mathbb{X}} T(x, t)$$

as the solution vector of the variables at time t . The functions $\varphi_i(x)$ are the so-called POD basis functions and they are derived from the simulation data of $T(x, t)$ for a specified time period¹. The POD basis functions are kept fixed during the simulations. The basis functions are determined from the results of the furnace modeling obtained by imposing a set of typical input signal excitations and boundary conditions used in the operation of real furnaces. The time varying coefficients $\{a_i(t)\}_{i=1}^n$ are the POD basis coefficients which describe the time-varying dynamics of the system.

The reduced order model, as explained in Section 2.6 is obtained by projecting (5.1) onto the space spanned by the first n dominant basis functions. In this way, we obtain a model for the POD basis coefficients $\{a_i(t)\}_{i=1}^n$. Since the number of the POD basis functions n is very low compared to the original discretization level of the model K , in the reduced order model we only need to find n time-varying POD basis coefficients.

In this chapter, the results of reduced order modeling of a glass melt feeder is presented. In contrast to the glass melting tank, the feeder modeling is less complex and can be easily validated since the operating temperature range of the feeder is accessible by the thermocouples. The operating temperature range of the glass melt feeder is very narrow (1450 – 1500 K) in comparison to the glass melting tank, which may have operating temperature range from room temperature to 1900 K. Modeling of the flows and temperature distribution of the melting tank is more complex because of the existence of different phases: solid (batch blanket), liquid (glass melt), and gas (bubbles). Therefore modeling of the glass melt feeder is considered in this study. The model of the glass melt feeder is focused on the processes taking places in the melted glass and not including the combustion chamber above the feeder channels.

This chapter is organized as follows. First the governing equations of the feeder simulation model are given. Then the simulations of the temperature in the

¹POD basis functions are also called POD basis vectors in discrete case as the basis functions are discretized into K elements

feeder and the application of POD technique are presented. Further, the acceleration techniques of reduced order modeling as applied and demonstrated in Chapter 4 for nonlinear models are also applied to the feeder simulation model. The acceleration technique is based on the estimation of the POD coefficients $\{a_i(t)\}_{i=1}^n$ based on the data (e.g temperature) obtained only at a selected number of locations in the feeder. The chapter is concluded with a summary.

5.2 The feeder model

A glass melt feeder is the section of a container, fiber, or TV glass furnace, which is located between the refiner and the glass melt exit point (spout). The feeder is fed by incoming glass melt from the working end, measured in tons/day and given as the *pull rate*. In the entrance part of the feeder, the glass is cooled very slowly and under very strict conditions to maintain small temperature gradients across the height and the width of the feeder. In the end-part of the feeder, the glass is not cooled anymore, but the temperature differences of the glass are kept to a minimum. For the control of the glass product quality, it is extremely important to precisely regulate the temperature within the feeder because non-uniform distribution will produce defect glass products such as irregular shapes or cracks [11].

A schematic view of a glass melt feeder channel is given in Figure 5.2:

Figure 5.2 shows the control mechanisms of a glass feeder as well. Above the glass melt, the temperature along the furnace roof or also called *crown*, is set to control the heat transfer by radiation. The surface temperature is controlled by some PID controllers [50] as shown in Figure 5.2. The PID controllers receive feedback from some thermocouple measurements at the glass surfaces (denoted by TC_1 to TC_4) in Figure 5.2. There are also some measurements in the glass melt themselves, denoted by GT_1 to GT_4 . The measurements are fed back to the PID controllers to obtain the desired temperatures GT_{1d} to GT_{4d} at the measurement points. In industrial applications, the parameters of the controllers are fixed by the manufacturers of the control systems. The parameters of the PID controllers are defined from the desired specifications of the controlled responses. For example, how fast should the temperature rise, what is the allowable temperature overshoot, and how big the steady state error should be. These parameters can be determined based on the transient responses of the measurement points and no detailed mathematical modeling

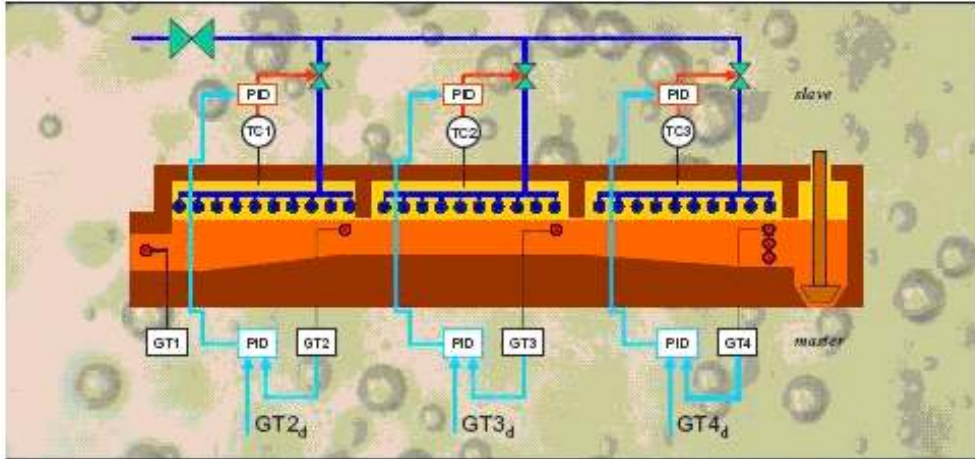


Figure 5.2: Schematic view of a glass feeder, the glass melt is entering the feeder from the left side and at the right end is discharged as glass gob to the forming machine

is required.

To improve the temperature and composition uniformity of the melt, a stirrer is sometimes used in the glass melt. Heat transfer by conduction is also taking place in the glass melt and between the glass melt and the feeder walls. Disturbances in the feeder may occur from the variation of pull rates or the temperature variations of the incoming glass melt.

Figure 3.21 shows a schematic view of a discretized spatial domain \mathbb{X} of a feeder. For demonstration purposes, here the geometry is a simplified one compared to the geometry of industrial feeders. It has dimensions of $8.5\text{m} \times 0.55\text{m} \times 2\text{m}$ in length, height, and width, respectively. This defines a Cartesian spatial volume $\mathbb{X} = [0, 8.5] \times [0, 0.55] \times [0, 2]$ whose coordinates are denoted by x (length), y (height) and z (width). The glass melt is in the layer $0.34\text{m} \leq y \leq 0.55\text{m}$. The nominal pull rate for this feeder is 80 tons per day.

In general, the glass melt flow in the feeder can be considered as an incompressible and laminar flow. The governing equations for the feeder are Navier-Stokes equations that describe the velocity field (v_x, v_y, v_z) in the x, y and z direction respectively as well as the pressure field p and the energy equations for the temperature field T [13]. The Navier-Stokes equations are solved for the glass media only, while the energy equations are solved for heat transfer in the glass melt media, through the feeder walls, and heat transfer from the

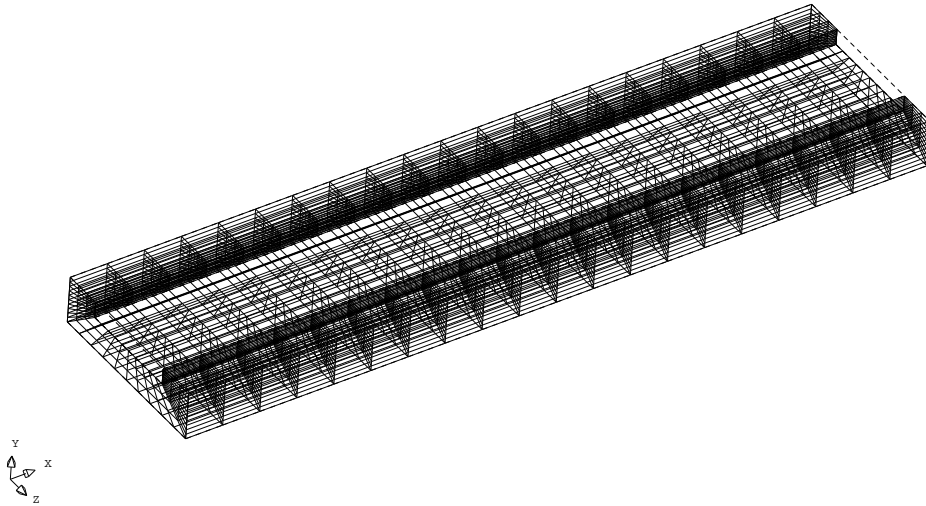


Figure 5.3: Geometry and grid cells of the feeder channel. The Cartesian coordinate orientation is denoted by the x for length, y for height, and z for width. The entrance of the feeder (which is connected to the working end) starts from the left part and the outlet of the feeder/the spout is on the rightmost part

crown to the melt surface.

The governing balance equations for the feeder are [70]:

Conservation equation of the glass melt:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v_x)}{\partial x} + \frac{\partial(\rho v_y)}{\partial y} + \frac{\partial(\rho v_z)}{\partial z} = 0 \quad (5.2)$$

For the momentum equations, define a substantive derivative $\frac{D\psi}{Dt}$ as the functions of the derivative of ψ with respect to time and the velocity field \mathbf{v} :

$$\frac{D\psi}{Dt} = \frac{\partial \psi}{\partial t} + v_x \frac{\partial \psi}{\partial x} + v_y \frac{\partial \psi}{\partial y} + v_z \frac{\partial \psi}{\partial z} = \frac{\partial \psi}{\partial t} + \mathbf{v} \cdot \text{grad } \psi \quad (5.3)$$

Momentum in x-direction of glass[70]

$$\frac{D\rho v_x}{Dt} = \frac{\partial(-p + \tau_{xx})}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{M_x} \quad (5.4)$$

where $\tau_{xx}, \tau_{yz}, \tau_{zx}$ are the viscous stresses. The suffices i and j in τ_{ij} indicate that the stress component acts in the j -direction on the surface normal to the

i -direction. The term S_{M_x} indicate the body forces in the x -direction such as the centrifugal force, and the electromagnetic force.

Momentum in y -direction[70]

$$\frac{D\rho v_y}{Dt} = \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial(-p + \tau_{yy})}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + S_{M_y} \quad (5.5)$$

The body force due to the gravity will be incorporated in the term S_{M_y} in y -direction as $-\rho g$.

Momentum in z -direction[70]

$$\frac{D\rho v_z}{Dt} = \frac{\partial(\tau_{xz})}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial(-p + \tau_{zz})}{\partial z} + S_{M_{zz}} \quad (5.6)$$

Energy Equation, solved for the refractory walls and glass melt[70]

$$\frac{D\rho E}{Dt} = -\nabla(p\mathbf{v}) + \frac{\partial v_x \tau_{xx}}{\partial x} + \frac{\partial v_x \tau_{yx}}{\partial y} + \frac{\partial v_x \tau_{zx}}{\partial z} \quad (5.7)$$

$$+ \frac{\partial v_y \tau_{xy}}{\partial x} + \frac{\partial v_y \tau_{yy}}{\partial y} + \frac{\partial v_y \tau_{zy}}{\partial z} \quad (5.8)$$

$$+ \frac{\partial v_z \tau_{xz}}{\partial x} + \frac{\partial v_z \tau_{yz}}{\partial y} + \frac{\partial v_z \tau_{zz}}{\partial z} + \nabla(\kappa \text{ grad } T) + S_E \quad (5.9)$$

where E is the sum of the internal/thermal energy, kinetic energy, and gravitational potential energy. The notation S_E in the energy equation refers to the contribution from external sources, such as heaters, electrical boostings, or the incoming heat fluxes from the combustion chamber. If the internal (thermal energy) per unit mass equals $c_p T$ where c_p is the heat capacity and the kinetic energy is $\frac{1}{2}(v_x^2 + v_y^2 + v_z^2)$, then the energy equation can be written as:

$$\begin{aligned} \frac{D\rho c_p T}{Dt} &= -\nabla(p\mathbf{v}) + \frac{\partial v_x \tau_{xx}}{\partial x} + \frac{\partial v_x \tau_{yx}}{\partial y} + \frac{\partial v_x \tau_{zx}}{\partial z} + \frac{\partial v_y \tau_{xy}}{\partial x} \\ &+ \frac{\partial v_y \tau_{yy}}{\partial y} + \frac{\partial v_y \tau_{zy}}{\partial z} + \frac{\partial v_z \tau_{xz}}{\partial x} \\ &+ \frac{\partial v_z \tau_{yz}}{\partial y} + \frac{\partial v_z \tau_{zz}}{\partial z} + \nabla(\kappa \text{ grad } T) + S_E - \frac{D\rho \frac{1}{2}(v_x^2 + v_y^2 + v_z^2)}{Dt} \end{aligned} \quad (5.10)$$

Most physical parameters of the glass melt are functions of temperature. In this case we list the temperature-dependent parameters of the green container glass (the temperature is in Kelvin):

- Density ρ (kg/m³), linear fitting of density taken from [66]

$$\rho(T) = 2540 - 0.14T$$

- Viscosity μ (Ns/m²) [11]

$$\mu(T) = 10^{-2.592} + \frac{4242.904}{T - 541.8413}$$

The viscosity relation is known as the *VFT relation*

- Specific heat c_p (J/kgK) [11]

$$c_p(T) = 1221 + 0.0956T$$

- Thermal conductivity κ (W/m.K) [66]

$$\kappa(T) = 0.527 + 0.001T + 2.67 \times 10^9 T^3$$

The heat conductivity is calculated using the Rosseland Mean Absorption approach. This approximation is valid if the distance over which the absorption takes place is sufficiently small compared to the glass depth. It is assumed that this requirement is satisfied in the glass feeder [51].

The radiative heat transfer mainly takes place between the combustion chamber and the glass melt [59]. The convective heat transfer from the flame to the glass melt can often be neglected in the simulation model. The heat flux into the glass melt q depends on various fluxes:

- Flux from the glass melt to the combustion chamber

$$F_{\text{out}} = \varepsilon_g \sigma T_s^4 + (1 - \varepsilon_g) F_{\text{in}}$$

- Flux from the combustion chamber into the glass bath

$$F_{\text{in}} = \varepsilon_f \sigma T_f^4 + (1 - \varepsilon_f) M_{\text{out}}$$

- Flux from the crown to the combustion chamber

$$M_{\text{out}} = \varepsilon_c \sigma T_c^4 + (1 - \varepsilon_c) M_{\text{in}}$$

- Flux from the combustion chamber to the crown

$$M_{\text{in}} = \varepsilon_f \sigma T_f^4 + (1 - \varepsilon_f) F_{\text{out}}$$

where $\varepsilon_g, \varepsilon_f, \varepsilon_c$ are the emission coefficients of the glass melt, flame, and the crown, respectively. Here the emission coefficients are assumed to be independent of the wave length. The Stefan Boltzmann constant is 5.67×10^{-8} W/m²K⁴. The temperatures T_s, T_f, T_c are the temperatures of the glass melt surface, the flame, and the crown in Kelvin.

The heat flux $q(x)$ into the glass melt at the position x in the furnace, measured in W/m² is given by [9]:

$$q(x) = \frac{\alpha_A \sigma (T_f^4(x) - T_s^4(x)) + \alpha_B \sigma (T_c^4(x) - T_s^4(x))}{\alpha_C} \quad (5.11)$$

where

$$\begin{aligned} \alpha_A &= \varepsilon_g \varepsilon_f (2 - \varepsilon_f - \varepsilon_c + \varepsilon_c \varepsilon_f) \\ \alpha_B &= \varepsilon_g \varepsilon_c (1 - \varepsilon_f) \\ \alpha_C &= 1 - (1 - \varepsilon_f)^2 (1 - \varepsilon_c) (1 - \varepsilon_g) \end{aligned}$$

In the glass melt feeder, the variables to be solved are the velocity components at the three dimensional directions v_x, v_y, v_z , the pressure field p and the temperature field T . These variables are not solved simultaneously by the simulation model. Starting from the initial values and estimated fields, the variables are solved consecutively in the order v_x - v_y - v_z - p - T until convergence of all conservation equations is achieved. Refer to [54] and [70] for details of the algorithm.

For every variable (v_x, v_y, v_z, p, T) and for every time step k , the following Linear Time Varying (LTV) model (refer to [54] [70] for details):

$$\mathbf{A}(k)\mathbf{T}(k+1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (5.12)$$

where $\mathbf{T}(k) \in \mathbb{R}^K$ is the K -th dimensional vector comprising the variable we would like to solve defined over the whole spatial domain \mathbb{X} and time domain \mathbb{T} . The input vector $\mathbf{u}(k)$ comprises the terms of the external sources such as the crown temperature, electrical boostings, heaters and the terms where boundary changes (such as inlet and outlet temperatures) are imposed.

Note that the CFD matrices $\mathbf{A}, \mathbf{A}_0, \mathbf{B}$ are now time-varying because the coefficients in the CFD matrices have to be updated when the elements of vector $\mathbf{T}(k)$ are changing. The equation (5.12) is solved for every variable. Since the variables are dependent on each other, such as the dependencies of temperature on the velocities and the dependencies of physical parameters on some of

the variables, the equations are solved several times within one time step until the conservation equations converge and the imposed boundary conditions are fulfilled.

To solve the equations numerically over the spatial domain \mathbb{X} and a finite time domain \mathbb{T} , the feeder is divided into 7128 grid points. Some grid points act as boundary points, where the Dirichlet or Neumann boundary conditions are imposed. These boundary points belong to the input terms $\mathbf{u}(k)$ in (5.12), and they do not belong to the variables to be solved $\mathbf{T}(k)$. The number of non-boundary points are 3800, therefore $\mathbf{T}(k) \in \mathbb{R}^{3800}$.

In the next section, we will discuss the derivation of POD basis $\{\varphi_i\}$ based on the simulation results from several excitations imposed on a glass melt feeder.

5.3 Snapshot collection and POD reduction

The glass melt feeder is excited by varying the pull rate between 65 tons/day and 110 tons/day. The nominal distribution of the crown temperature is depicted in Figure 5.5. The simulation is run for a process time of 480 minutes, with a sampling interval $\Delta t = 1$ minute. The crown is divided into four zones,

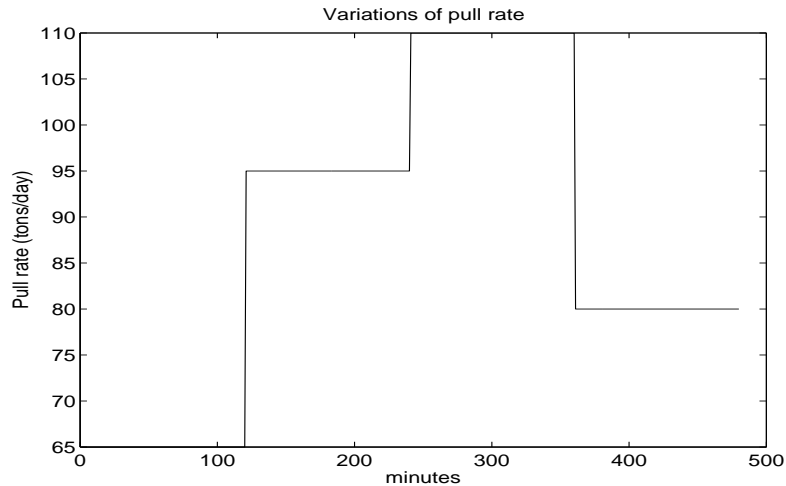


Figure 5.4: Pull Variations

from the inlet to the outlet of the feeder. In the simulation, the nominal temperature distribution of each zone is varied from the nominal temperature

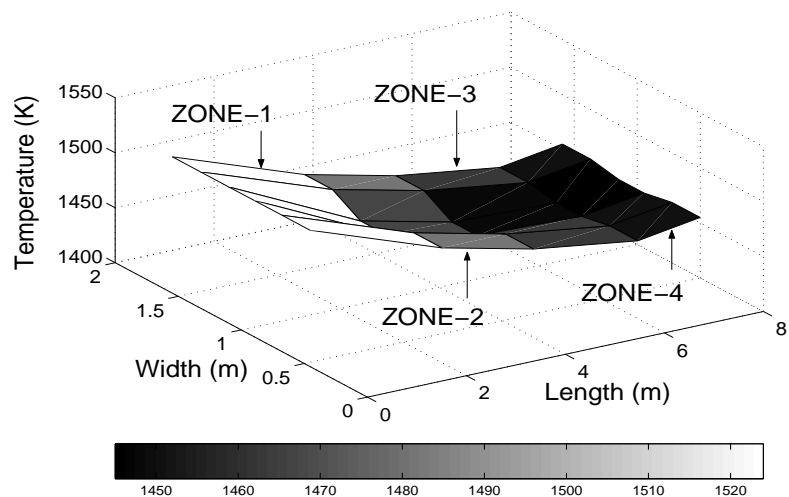


Figure 5.5: The nominal crown temperature profile, divided into four zones

as shown in Figure 5.6.

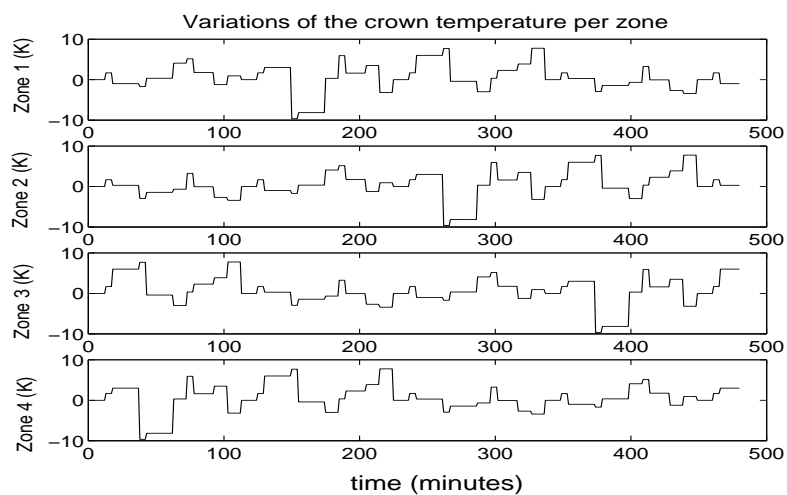


Figure 5.6: The variation of the temperature distribution of each crown zone during the simulation. The variations are from the nominal temperature distribution as shown in Figure 5.5

5.3.1 Determination of POD basis $\{\varphi_i\}$

We assume that the data of the simulations belong to a separable Hilbert space (Chapter 2) so that every variable such as the temperature $T(x, y, z, t)$ can be approximated by its truncated expansion $T_n(x, y, z, t)$ in n orthonormal basis functions as:

$$T_n(x, y, z, t) = \sum_{i=1}^n a_i(t) \varphi_i(x, y, z) \quad (5.13)$$

where $\{\varphi_i\}_{i=1}^n$ is the orthonormal basis. If the spatial domain \mathbb{X} is divided into K grid cells, then the collection of the field at time t for the whole spatial domain can be collected in a vector $\mathbf{T}(t) \in \mathbb{R}^K$. In this case, the basis functions are also vectors with length K .

For notational convenience, we would slightly abuse the notation for spatial coordinate. The spatial coordinate (x, y, z) is replaced by x in the forthcoming discussion. So $T(x, y, z, t)$ will be denoted further as $T(x, t)$.

Since the spatial domain \mathbb{X} and the time domain \mathbb{T} are discrete and finite, $T(x, t)$ is an element of a finite dimensional Hilbert space (refer to Section 2.4). As in the previous chapters, we denote further $\mathbf{T}(t)$ as the collection of the simulation data $T(x, t)$ at a specified time t for the whole spatial domain \mathbb{X} . The simulation data for the whole simulation is then collected in $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{3800 \times 480}$ where

$$\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \quad \mathbf{T}(2) \quad \dots \quad \mathbf{T}(480))$$

Algorithm 2.4.10 is then implemented to find the POD basis $\{\varphi_i\}$. The POD basis functions are the eigenvectors of the correlation matrix $\mathbf{C} = \frac{1}{480} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$.

The eigenvalue spectrum ² of \mathbf{C} is shown in Figure 5.7. The number of the POD basis functions in the reduced order model is determined from the criterion in (2.37):

$$P_n = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^N \lambda_i}$$

where N is the maximum number of the nonzero eigenvalues. If P_n is close to 1, then the better the approximation of the snapshot data by n POD basis functions will be. Figure 5.8 shows the logarithmic plot $1 - P_n$ for $n = 1, \dots, 50$. We truncate the POD basis functions up to $n = 18$. In the original model, we have to solve 3800 unknowns while in the reduced order model, we only have to find 18 POD basis coefficients $\{a_i(t)\}_{i=1}^{18}$.

²plot of the eigenvalues

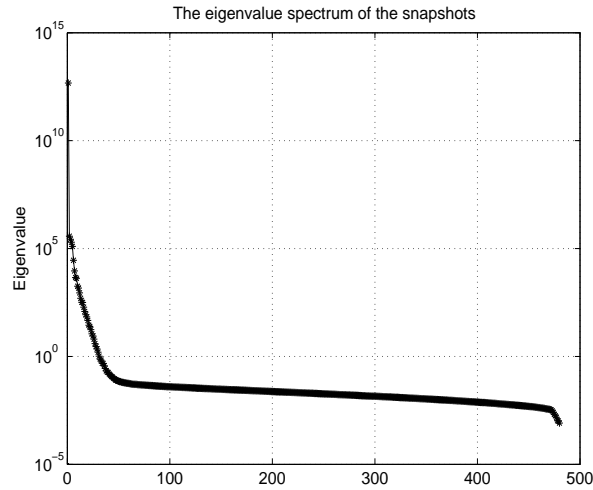


Figure 5.7: The eigenvalue spectrum from the eigenvalue decomposition of the simulation data collected from the feeder, the eigenvalues are ordered: $\lambda_1 \geq \lambda_2 \cdots \geq 0$

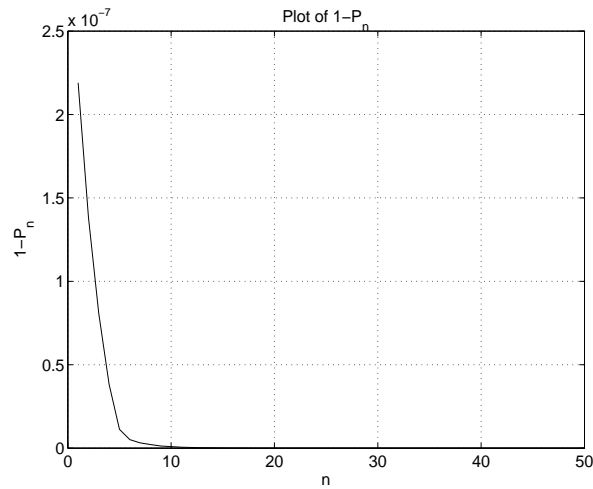


Figure 5.8: The logarithmic plot $1 - P_n$ which is used to determine the truncation degree of the POD basis functions

5.3.2 Construction of the reduced feeder model

For the temperature field of the feeder, initially we solve the following equations:

$$\mathbf{A}(k)\mathbf{T}(k+1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (5.14)$$

where $\mathbf{u}(k)$ is the vector comprising the inputs at time step k , such as the temperature of each zone of the crown.

We have derived 18 POD basis functions or vectors from the collection of simulation data matrix $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{3800 \times 480}$. The POD basis vectors are collected in a matrix $\Phi \in \mathbb{R}^{3800 \times 18}$:

$$\Phi = (\varphi_1 \quad \dots \quad \varphi_{18})$$

The reduced order model for the temperature field is constructed by replacing $\mathbf{T}(k)$ by the expansion in the n POD basis functions:

$$\mathbf{T}(k) = \sum_{i=1}^{18} a_i(k) \varphi_i = \Phi \mathbf{a}(k)$$

where $\mathbf{a}(k) = \text{col}(a_1(k), a_2(k), \dots, a_{18}(k))$ and projecting Φ onto (5.14):

$$\Phi^\top \mathbf{A}(k) \Phi \mathbf{a}(k+1) = \Phi^\top \mathbf{A}_0(k) \Phi \mathbf{a}(k) + \Phi^\top \mathbf{B}(k) \mathbf{u}(k) \quad (5.15)$$

Note that the reduced model for the temperature field now has 18 unknowns, while the original CFD model for temperature field has 3800 unknowns.

The governing equations of CFD models are often a set of nonlinear PDE's. For such systems, knowledge of typical excitation signals encountered during the operation will be beneficial because choosing the most effective excitation signal for nonlinear systems is still an open problem. Once we shift to another operating point, there is no guarantee that the reduced order model gives a reliable approximation of new dynamics. The best way is to collect simulation data based on several typical operating points. Industrial processes usually have limited operating range and most process controllers are also working to improve the performance of a plant/system only in a specified operating range.

5.4 Validation of the reduced order model

To validate the reduced order model, a random excitation signal is imposed on the feeder. The random excitation signals applied to the four zones of the crown temperature are shown in Figure 5.9. The pull rate is kept at its nominal value, 80 tons/day. These excitation signals are different than those applied to derive the POD basis. The objective is to observe the ability of the reduced order model to estimate the temperature when the excitation signals

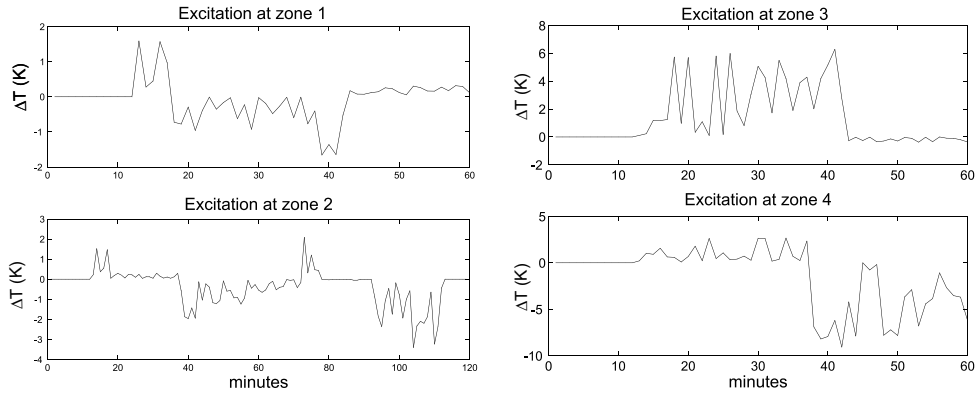


Figure 5.9: Random excitation signals at the four zones of the feeder

are varying differently. The variations are kept within the normal operating range of the feeder.

Figure 5.11 and Figure 5.13 show the results of the reduced temperature field in comparison with the original temperature field. The results are plotted for several measurement locations. The measurement locations at the glass surface and the glass outlet are shown in Figure 5.10 and Figure 5.12.

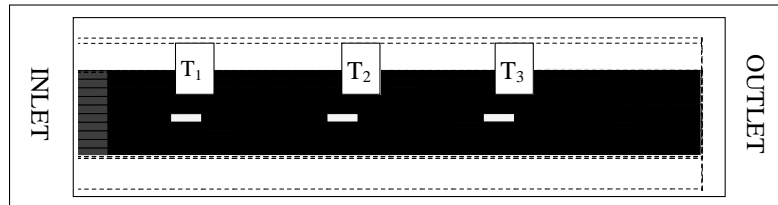


Figure 5.10: The measurement locations at the glass surface, denoted from the left to the right as T_1, T_2, T_3 . The inlet of the feeder is at the leftmost part and the outlet at the rightmost part

The original temperature field is solved by $K = 3800$ equations and the reduced temperature field is obtained by applying the POD technique and taking $n = 18$ dominant modes which correspond to 18 largest eigenvalues of the temperature snapshots.

From Figure 5.11 and Figure 5.13, it can be observed that the reduced order model approximates the temperature field of the original model closely. The plot of the absolute average error for the grid cells of glass melt is given in

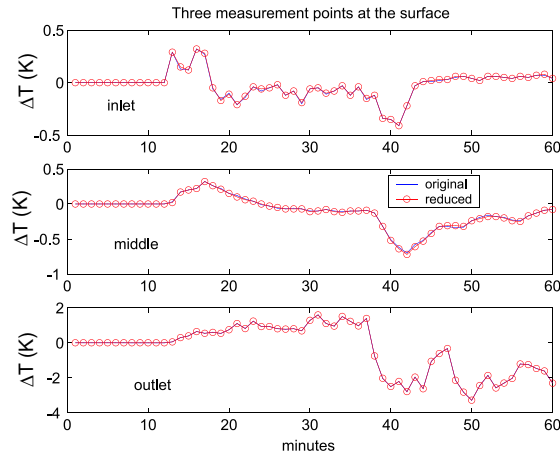


Figure 5.11: Comparison between the temperature field calculated by the reduced order model and the original model at the surface measurement points (see Figure 5.10). The top figure refers to T_1 , middle to T_2 and bottom to T_3

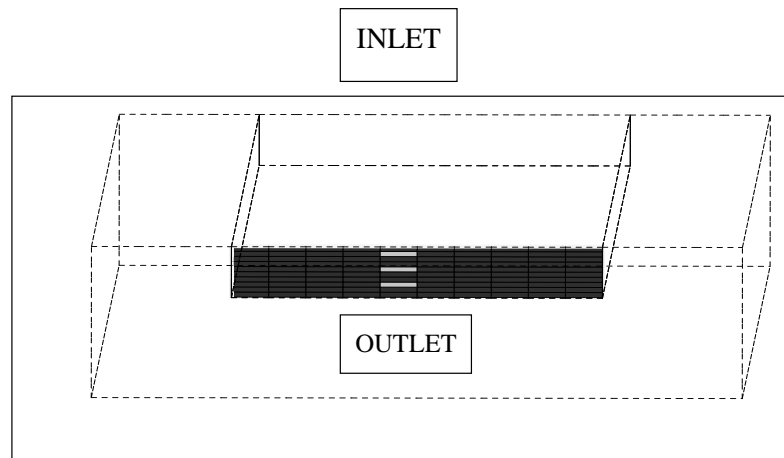


Figure 5.12: The measurement locations at the glass outlet, shown by the white grid cells. The sensor positions from the top to the bottom are denoted as T_4, T_5, T_6

Figure 5.14

The maximum averaged absolute error is very small, it accounts about $0.04^\circ K$ maximum. The averaged absolute error between the original temperature $T(x, k)$ and the temperature calculated by the reduced order model $T_n(x, k)$

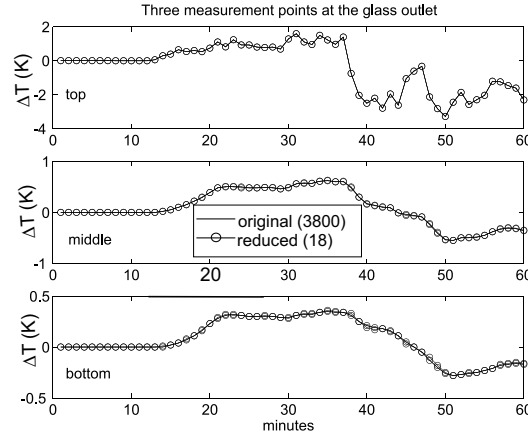


Figure 5.13: Comparison between the temperature field calculated by the reduced order model and the original model at the outlet measurement points (see Figure 5.12)

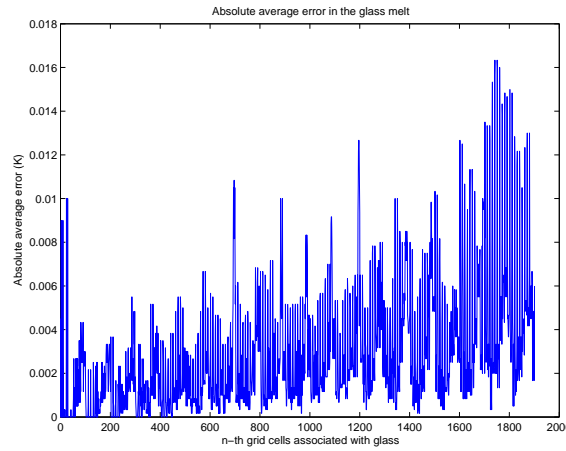


Figure 5.14: The average of the absolute error for the grid cells of glass melt

is:

$$\bar{\epsilon}(x) = \frac{1}{N_s} \sum_{k=1}^{N_s} |T_n(x, k) - T(x, k)| \quad (5.16)$$

where N_s is the number of time samples, $T_n(x, k)$ is the temperature field at the location \bar{x} obtained from the reduced order model, $T(x, k)$ is the temperature field obtained from the original model.

Even though the number equations to be solved in the reduced model is only 0.005% of the original number of equations, the average absolute deviation is less than 0.05% of the difference between the minimal and the maximal temperature in the glass melt, which lies in the range of 30 K.

5.5 Application of MPE to the glass melt feeder

The drastic reduction in the model order is expected to drastically reduce the computational load of the reduced order model. The order of the original model is more than 200 times smaller than the reduced order model, but the computational time needed to solve the temperature field is only enhanced by factor 2.

The model reduction computational time for time-varying and non-linear models such as CFD models is not dramatically decreased though the number of POD basis functions is very small. This is due to the fact that the reduced order model is obtained by projecting the *full* model onto the low number of POD basis functions.

It implies that to obtain a reduced order model, the parameters of the *full order model* have to be obtained first. The original CFD model has the following form:

$$\mathbf{A}(k)\mathbf{T}(k+1) = \mathbf{A}_0(k)\mathbf{T}(k) + \mathbf{B}\mathbf{u}(k)$$

Note that the CFD matrices $\mathbf{A}(k)$, $\mathbf{A}_0(k)$, $\mathbf{B}(k)$ are time-varying. The coefficients of these matrices are constantly updated to accommodate the varying physical parameters such as the density, viscosity, and the heat conductivity.

The *reduced* CFD model is obtained as follows:

$$\Phi^\top \mathbf{A}(k)\Phi \mathbf{a}(k+1) = \Phi^\top \mathbf{A}_0(k)\Phi \mathbf{a}(k) + \Phi^\top \mathbf{B}(k)\mathbf{u}(k) \quad (5.17)$$

Hence to obtain the reduced order model, $\mathbf{A}(k)$, $\mathbf{A}_0(k)$, $\mathbf{B}(k)$ have to be computed first. This is a very expensive process. For large scale systems this is currently the main bottleneck for performing fast simulations and on-line control system design.

In Chapter 4, the reduced order modeling by the MPE method is proposed to accelerate the reduced model computation. In MPE method, the models of the POD coefficients are derived from a part of the original equations. The part of the original equations correspond to a part of the selected locations in

spatial domain \mathbb{X} . The collection of the selected locations in \mathbb{X} is denoted in Chapter 4 as \mathbb{X}_0 .

If the original temperature field T is defined for the whole spatial domain \mathbb{X} , then the temperature field at \mathbb{X}_0 is denoted as \tilde{T} . Suppose \mathbb{X} is discretized into K grid points and \mathbb{X}_0 comprises of G grid points, $G < K$. Denote the discrete temperature data at k -th time step for \mathbb{X} as $\mathbf{T}(k)$ and the discrete temperature data for \mathbb{X}_0 as $\tilde{\mathbf{T}}(k) \in \mathbb{R}^G$. If $\mathbf{T}(k)$ can be approximated by the POD basis functions $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_n)$, then $\tilde{\mathbf{T}}$ can also be approximated as:

$$\tilde{\mathbf{T}}(k) \approx \tilde{\Phi} \mathbf{a}(k) \quad (5.18)$$

where the elements of $\tilde{\Phi} = (\tilde{\varphi}_1 \ \tilde{\varphi}_2 \ \dots \ \tilde{\varphi}_n)$ are the elements of the original POD basis Φ corresponding to the locations $x \in \mathbb{X}_0$. The basis $\tilde{\Phi}$ is also basis for \mathbb{X}_0 , but unlike the original POD basis, it is not orthonormal nor an optimal basis.

The POD coefficients $\mathbf{a}(k)$ are originally calculated by projecting the POD basis functions $\Phi \in \mathbb{R}^{K \times n}$ onto K equations describing the dynamics of $\mathbf{T}(k)$. In MPE method, the POD coefficients $\mathbf{a}(k)$ are *estimated* by projecting $\tilde{\Phi} \in \mathbb{R}^{G \times n}$ onto G equations describing the dynamics of $\tilde{\mathbf{T}}(k)$. As a result, only parameters of G equations need to be calculated in contrast to the original reduced model where the parameters of K equations have to be calculated. The resulting POD coefficients obtained by the MPE method are denoted as $\tilde{\mathbf{a}}$. Once $\tilde{\mathbf{a}}$ obtained, the temperature field can be reconstructed using the *full* POD basis as:

$$\hat{T} = \Phi \tilde{\mathbf{a}}$$

The locations in \mathbb{X}_0 have to be chosen in such a way the estimated POD coefficients $\tilde{\mathbf{a}}$ is close to the POD coefficients calculated from projections onto the full model. In section 4.3.1 and section 4.3.2, there are two criteria proposed to select the subset \mathbb{X}_0 .

The first selection criterion (Criterion 1) is based on the approximation of $\Phi^\top \Phi$ by $\tilde{\Phi}^\top \tilde{\Phi}$. The original, complete POD basis functions are orthonormal that $\Phi^\top \Phi$ equals an identity matrix. The closer $\tilde{\Phi}^\top \tilde{\Phi}$ to identity matrix with $\tilde{\Phi}$ with $\tilde{\Phi}$ are the elements of the original POD basis Φ corresponding to the locations $x \in \mathbb{X}_0$, then the more representative \mathbb{X}_0 is.

In the second selection criterion, the POD basis functions constructed from the subset \mathbb{X}_0 are weighted by the POD coefficients. The POD coefficients are obtained by the projection of Φ onto the snapshot matrix \mathbf{T}_{snap} . In

the approximation by POD basis functions, the process variables such as the temperature data at every time step are described as the linear combination of the POD basis functions weighted by the time varying coefficients at every time step ((5.1)). In criterion 1, the POD basis functions $\tilde{\Phi}$ are not weighted by the POD coefficients, thus there is no direct relation to the approximation of the process variables.

Both Criterion 1 and Criterion 2 are going to be applied to the glass melt feeder models. The theoretical details of both criteria can be found in section 4.3.1 and section 4.3.2.

5.5.1 Application of MPE based on selection criterion 1

For the feeder, we have selected $n = 18$ basis functions, and the number of grid points, K , involved in the CFD calculation (5.12) is 3800. The symmetry along the width of the feeder channel is taken into account in both the original and the reduced order, so $\Phi \in \mathbb{R}^{1900 \times 18}$ in (5.15).

The POD model is 2.26 times faster than the original model. To improve the computational gain of the reduced order model, we apply the method of Missing Point Estimation (MPE) as discussed in Chapter 4.

Using MPE, we have the following reduced-order model to derive $\mathbf{a}(k)$ and to solve (refer to section 4.5 for details of derivation):

$$\underbrace{\tilde{\Phi}^T \tilde{\mathbf{A}}(k) \tilde{\Phi}_{\text{nb}}}_{\mathbf{A}_r} \tilde{\mathbf{a}}(k+1) = \underbrace{\tilde{\Phi}^T \tilde{\mathbf{A}}_0(k) \tilde{\Phi}}_{\tilde{\mathbf{A}}_{0r}} \tilde{\mathbf{a}}(k) + \underbrace{\tilde{\Phi}^T \tilde{\mathbf{B}}(k)}_{\tilde{\mathbf{B}}_r} \mathbf{u}(k) \quad (5.19)$$

$$\mathbf{A}_r \tilde{\mathbf{a}}(k+1) = \tilde{\mathbf{A}}_{0r}(k) \tilde{\mathbf{a}}(k) + \tilde{\mathbf{B}}_r(k) \mathbf{u}(k) \quad (5.20)$$

Note that $\tilde{\Phi}_{\text{nb}}$ is the POD basis corresponding to the locations $x \in \mathbb{X}_0$ as well as its neighboring points whose locations do not belong to \mathbb{X}_0 . For details, refer to section 4.5. The resulting reduced order model is called the POD-MPE model.

In the POD-MPE reduced order models, boundary conditions still have to be satisfied. The crown temperature defines a set of Dirichlet boundary conditions, and these have to be incorporated in the POD-MPE models as well. In this way, the POD-MPE model can update the information of the boundary condition continuously. To incorporate the boundary conditions, all points which are adjacent to the boundary cells are included in \mathbb{X}_0 . There are 265 points in the feeder model which are adjacent to the boundary cells where

crown temperature, inlet temperature, inlet velocity are defined. These points are considered as "obligatory points".

The remaining points are selected from the calculation of the deviation $\tilde{\Phi}\tilde{\Phi}^\top$ from $\Phi^\top\Phi$ (see section 4.3.1). Since $\tilde{\Phi}$ comprises the elements of the POD basis Φ on \mathbb{X}_0 , $\tilde{\Phi}$ depends on the choice of \mathbb{X}_0 . Thus, $\tilde{\Phi}$ can be written as function of \mathbb{X}_0 :

$$\begin{aligned}\tilde{\Phi} &= \tilde{\Phi}(\mathbb{X}_0) = \left(\varphi_{1|\mathbb{X}_0} \quad \dots \quad \varphi_{n|\mathbb{X}_0} \right) \\ &= (\tilde{\varphi}_1 \quad \dots \quad \tilde{\varphi}_n)\end{aligned}$$

To select the points, \mathbb{X}_0 is chosen to be a location of *one grid point* only. Hence $\tilde{\Phi} = \tilde{\Phi}(\mathbb{X}_0)$, $\tilde{\Phi} \in \mathbb{R}^{1 \times n}$.

The deviation of $\tilde{\Phi}^\top\tilde{\Phi}$ formed from only one grid point at location $\mathbb{X}_0 = x_k$ is calculated as:

$$e_{x_k} = \| E_{\mathbb{X}_0} \| = \| \tilde{\Phi}(x_k)^\top \tilde{\Phi}(x_k) - \Phi^\top \Phi \| \quad (5.21)$$

where

$$\| X \| = \sum_{i=1}^n \sum_{j=1}^n X_{ij}^2 \quad (5.22)$$

with X_{ij} is the element of a square matrix $X \in \mathbb{R}^{n \times n}$ at i -th row and j -th column. As $\mathbb{X}_0 = x_k$, then $\tilde{\Phi}(\mathbb{X}_0) = \tilde{\Phi}(x_k)$.

Thus, we calculate (5.21) for all remaining candidate points. In this example, since 265 boundary points have been taken, there are remaining 1635 points to be chosen. The locations of the remaining points are denoted as x_1, \dots, x_{1635} . After (5.21) is calculated for every point, e_{x_k} values are then ordered such that:

$$e_{x_{k_1}} \leq e_{x_{k_2}} \leq \dots \leq e_{x_{k_{1635}}} \quad (5.23)$$

with $k_1, k_2, \dots, k_{1635}$ are the ordered index.

The point at location x_{k_1} is considered the most important point while points at location $x_{k_{1635}}$ is the least important point. Figure 5.15 shows the plot of the ordered e_{x_k} .

From Figure 5.15, there are points which are more representative than the others, although in this example the difference is small. However, in order to select \mathbb{X}_0 , a starting point is needed and the point with minimum e_{x_k} is considered in this example as the best candidate point. It is desired to have sufficient points in \mathbb{X}_0 such that the condition number of $\tilde{\Phi}^\top(\mathbb{X}_0)\tilde{\Phi}(\mathbb{X}_0)$ will be

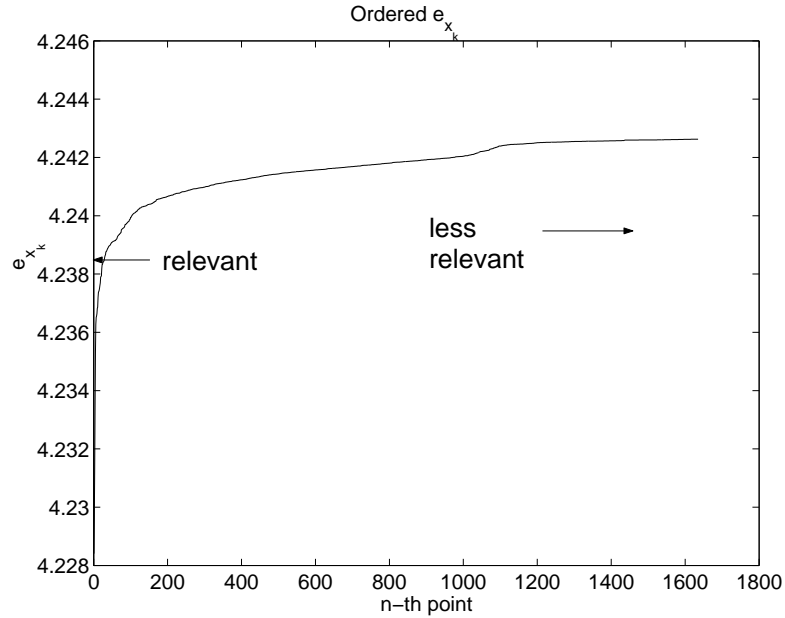


Figure 5.15: Plot of the ordered e_{x_k} , with e_{x_k} calculated as in (5.21)

close to the condition number of $\Phi^\top \tilde{\Phi}$. Since the POD basis is an orthonormal basis, $\Phi^\top \Phi$ equals identity and the condition number of $\Phi^\top \Phi$ equals 1.

The condition number is important for the estimation of the POD coefficients. If the condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$ is higher than 100 ([35],[34]), then there exist linear dependencies of the points in \mathbb{X}_0 . It means the equations in (5.19) are dependent on each other and there exists many possible solutions of $\mathbf{a}(k)$.

To check the condition number, the following procedure is conducted:

Procedure 5.5.1 Given \mathbb{X}_0 as the locations of the 265 points adjacent to the boundary points where the changing boundary conditions are defined.

1. From all candidate points $x_k \in \mathbb{X} \setminus \mathbb{X}_0 := \{x \in \mathbb{X} \mid x \notin \mathbb{X}_0\}$, determine x_{k_1} as the location of the point with minimum e_{x_k} . Add x_k to \mathbb{X}_0 .
2. Construct $\tilde{\Phi}(\mathbb{X}_0)$
3. Calculate the condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$.

4. Repeat step 1 to 3 for each addition of $x_{k_2}, x_{k_3}, \dots, x_{k_{1635}}$ where $e_{x_{k_2}} \leq e_{x_{k_3}} \leq \dots, e_{x_{k_{1635}}}$.

For each choice of \mathbb{X}_0 , a condition number is calculated. The plot of the condition number is depicted in Figure 5.16. The condition number plot has already

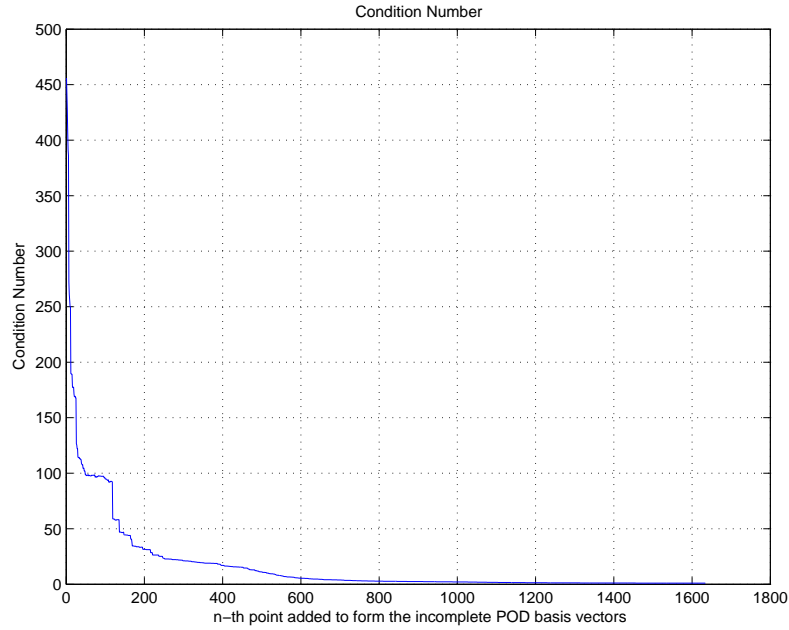


Figure 5.16: Plot of the condition number of $\tilde{\Phi}^T \tilde{\Phi}$

flattened after 1000th extra point has been added to the 265 obligatory points, thus with \mathbb{X}_0 consists of 1265 points, the condition number of $\tilde{\Phi}(\mathbb{X}_0)^T \tilde{\Phi}(\mathbb{X}_0)$ is close to the condition number of $\Phi^T \Phi$.

Since we selected 1000 points and there are 265 points for the boundary conditions, $\tilde{\Phi} \in \mathbb{R}^{1265 \times 18}$. The CFD matrices which have to be convolved by the POD basis vectors also only have 1265 rows. By comparing (5.19) and (5.12) which is solved by taking 1900 points in the symmetric case, it is clear that (5.19) is more attractive by computational point of view because we could obtain the estimates of the POD basis coefficients with a cheaper procedure.

Figure 5.17 depicts the selected points at the glass domain at the glass melt 10 cm below the glass surface. There are 960 points taken in the glass domain. The fact that there are more points selected in the glass melt can be

understood, because the main dynamics is excited in the glass domain.

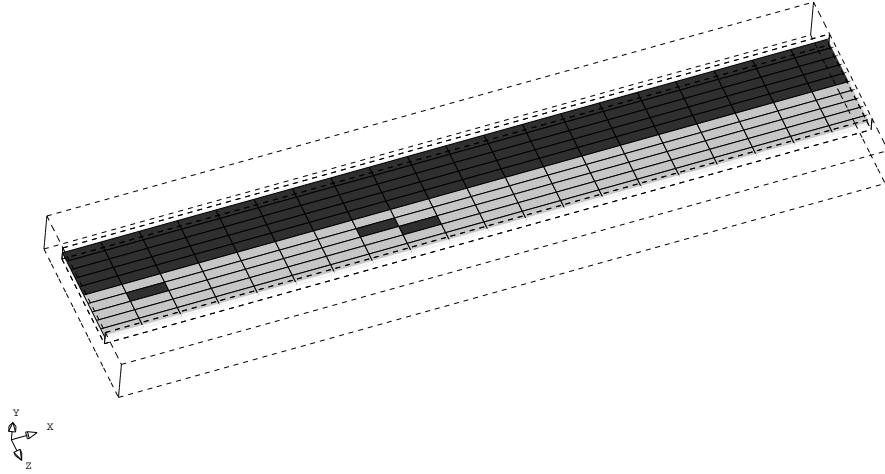


Figure 5.17: Parts of the selected 1265 points at the cross section of the glass melt (shown by the grey grid cells) 10 cm below the glass surface

5.5.2 The Shannon Entropy

As an alternative to the condition number, the so-called *Shannon Entropy* can also be used to determine the number of points. The Shannon Entropy equation is used in the information theory to determine the minimum compression rate of a signal which guarantees that the essential information of the uncompressed signal still exists in the compressed signal.

Analogously, in the case of POD-MPE reduced order models, we would like to select \mathbb{X}_0 in such a way so that $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$ approximates $\Phi^\top \Phi$ well.

The Shannon Entropy equation is given as:

$$H = - \sum_{i=1}^N P_i \ln P_i \quad (5.24)$$

where $\sum_{i=1}^N P_i = 1$. The quantity P_i in (5.24) defines a probability function. For example given a 3 letters $\{a, b, c\}$ and a sequence composed from these letters as $\{a, b, b, b, c\}$. For each letter, the probability of $P_i, i = 1, 2, 3$ can be

calculated as how often a particular letter is found in the sequence divided by the number of elements in the sequence.

The same concept can also be applied to the POD-MPE models. Consider the eigenvectors of $\tilde{\Phi}^\top \tilde{\Phi}$ as the patterns of $\tilde{\Phi}^\top \tilde{\Phi}$, hence the eigenvectors are considered as the letters in the previous example.

The probability P_i of each eigenvector of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$ is formulated as:

$$P_i(\mathbb{X}_0) = \frac{\lambda_i(\mathbb{X}_0)}{\sum_{i=1}^{18} \lambda_i(\mathbb{X}_0)} \quad (5.25)$$

where λ_i is the i -th eigenvalue of $\tilde{\Phi}^\top \tilde{\Phi}$. The probability P_i is denoted as function of the restricted domain $\mathbb{X}_0 \subset \mathbb{X}$ since different choice of \mathbb{X}_0 will lead to different $\tilde{\Phi}^\top \tilde{\Phi}$. Similarly, let H also become a function of \mathbb{X}_0 .

In case of no missing data or $\mathbb{X}_0 = \mathbb{X}$, $\Phi^\top \Phi$ equals an identity matrix. The eigenvalues of an identity matrix are all equal to 1, hence P_i in that case is given by:

$$P_i(\mathbb{X}_0) = P_i(\mathbb{X}) = \frac{1}{18}$$

and

$$H(\mathbb{X}_0) = H(\mathbb{X}) = \ln 18$$

The goal is to choose \mathbb{X}_0 so that $H(\mathbb{X}_0) \approx H(\mathbb{X})$. Similar to the calculation of the condition number (Procedure 5.5.1), $H(\mathbb{X}_0)$ is calculated with $P_i(\mathbb{X}_0)$ as defined in (5.25) and \mathbb{X}_0 is formed from the 265 boundary points and x_{k_1} as the location of the point with minimum e_{x_k} ((5.21)). Then x_{k_2} is added to \mathbb{X}_0 , calculate $H(\mathbb{X}_0)$ for this configuration. Add a point one by one based on the ordering of e_{x_k} , calculate $H(\mathbb{X}_0)$ for each configuration. Continue the procedure until $x_{k_{1635}}$ is added to \mathbb{X}_0 .

Figure 5.18 shows the plot of the Shannon's Entropy, where $H(\mathbb{X}_0)$ is plotted by adding one point to the 265 boundary points until the last candidate point (based on the ordering of e_{x_k}) is added. After 1000 points have been added to the 265 obligatory points, $H(\mathbb{X}_0) \approx H(\mathbb{X})$ or $H(\mathbb{X}_0) \approx \ln 18$. This can be interpreted as the truncation degree because adding points beyond 1000 points add very little change to $H(\mathbb{X}_0)$.

The POD-MPE model is then validated by varying the crown temperature of each zone as shown in Figure 5.19. The deviation of the POD-MPE model from the original model is negligible. The maximum average error is 0.04°C. The acceleration factor of the POD-MPE model with 1265 points is 3.35 times

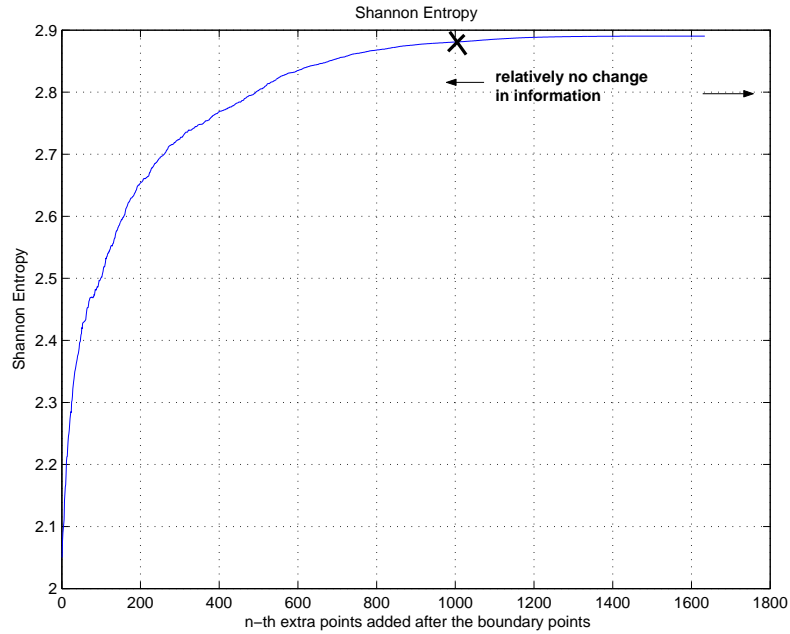


Figure 5.18: Plot of Shannon's Entropy for $\tilde{\Phi}^T \tilde{\Phi}$

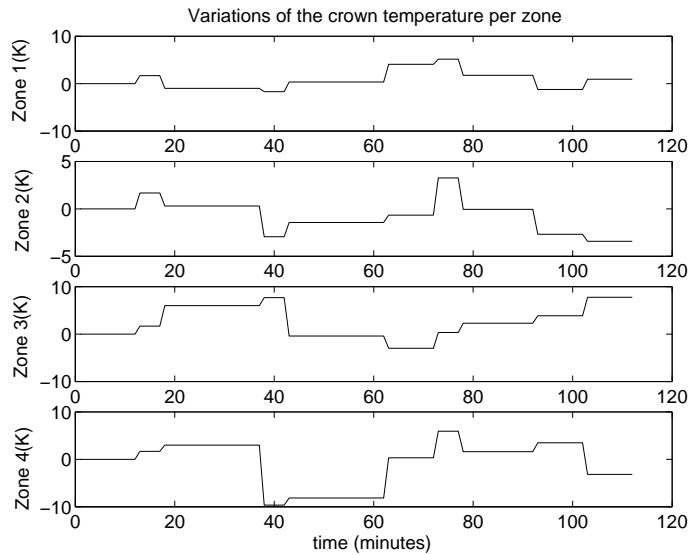


Figure 5.19: The crown temperature variations of each zone

faster than the original model. The ordering strategy of the points does not yield quite a low number of selected points. We still have to take a lot of

points before a good condition number is reached. The procedure of selecting \mathbb{X}_0 will be optimized by the so-called *greedy algorithm* [72] in the next section.

5.5.3 Optimization of point selection

The procedure of point selection presented in the previous section may lead to a high number of points before a low condition number (or alternatively $H(\mathbb{X}_0) \approx H(\mathbb{X})$ from the Shannon entropy) could be reached. To reduce the number of points required to have a low condition number, we implement the greedy algorithm [72]:

Algorithm 5.5.2 The greedy algorithm

1. Determine G as the number of points to be selected in \mathbb{X}_0 .
2. Determine \mathbb{X}_0 as the locations of the 265 boundary points and x_{k_1} as the point for which $e_{x_{k_1}}$ is minimum.
3. Construct $\tilde{\Phi}(\mathbb{X}_0)$ as the basis for \mathbb{X}_0 corresponding to the POD basis elements Φ at the locations of \mathbb{X}_0 and calculate the condition number of $\tilde{\Phi}^\top \tilde{\Phi}$.
4. Add the point to \mathbb{X}_0 which gives the minimal condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi} \mathbb{X}_0$.
5. Repeat step 4 until \mathbb{X}_0 consists of the locations of G points.

The greedy algorithm is implemented for the glass melt feeder, and 300 extra points are chosen. With the 265 boundary points, we obtained in total 565 points.

The total selected points in the glass domain are 386 points, which is 40% of the total selected points. The selected points in the glass domain 10cm below the glass melt surface is given in Figure 5.20.

Figure 5.21 shows the comparisons between the original (symmetrical) model, the POD-MPE model with 1265 points and the POD-MPE based model with 565 points. The comparisons are conducted for the measurement points T_1 to T_3 at the surface and T_4 to T_6 at the outlet.

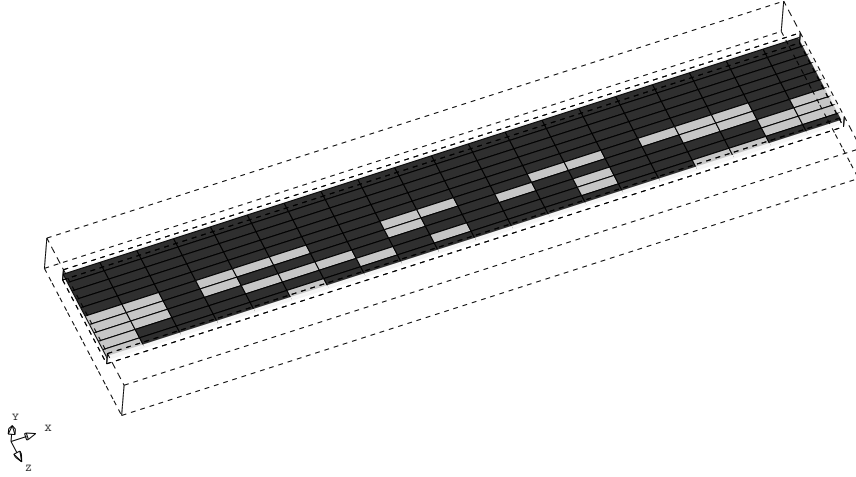


Figure 5.20: Parts of the selected 565 points at the cross section of the glass melt located 10 cm below the glass melt surface

Table 5.1 tabulates the comparisons between POD, POD-MPE with 1265 points and POD-MPE with 565 points. The table enlists the maximum average absolute error (the average absolute error is calculated by (5.16)) and the computational gain. The computational gain is computed as the CPU time required to update the model parameters and to solve the temperature field. The computational gain of POD-MPE with 565 points corresponds to 7.5 times faster than real time. In this section, all POD bases are considered

Table 5.1: Comparison between POD and POD-MPE models

| Model Type | Maximum Absolute Average Error (maximum of (5.16) for all $x \in \mathbb{X}$) | Computational Gain |
|--------------|---|--------------------|
| POD | 0.007° C | 226% |
| POD-MPE-1265 | 0.007° C | 335% |
| POD-MPE-565 | 0.012° C | 620% |

of equal importance. It would be interesting to take into account the relation of POD bases with data in the point selection. The POD basis functions are weighted by the POD coefficients when approximating the snapshot data. The point selection criterion 2 is based on the weighting of the POD basis functions with the POD coefficients and the application of this criterion 2 will be presented in the next section.

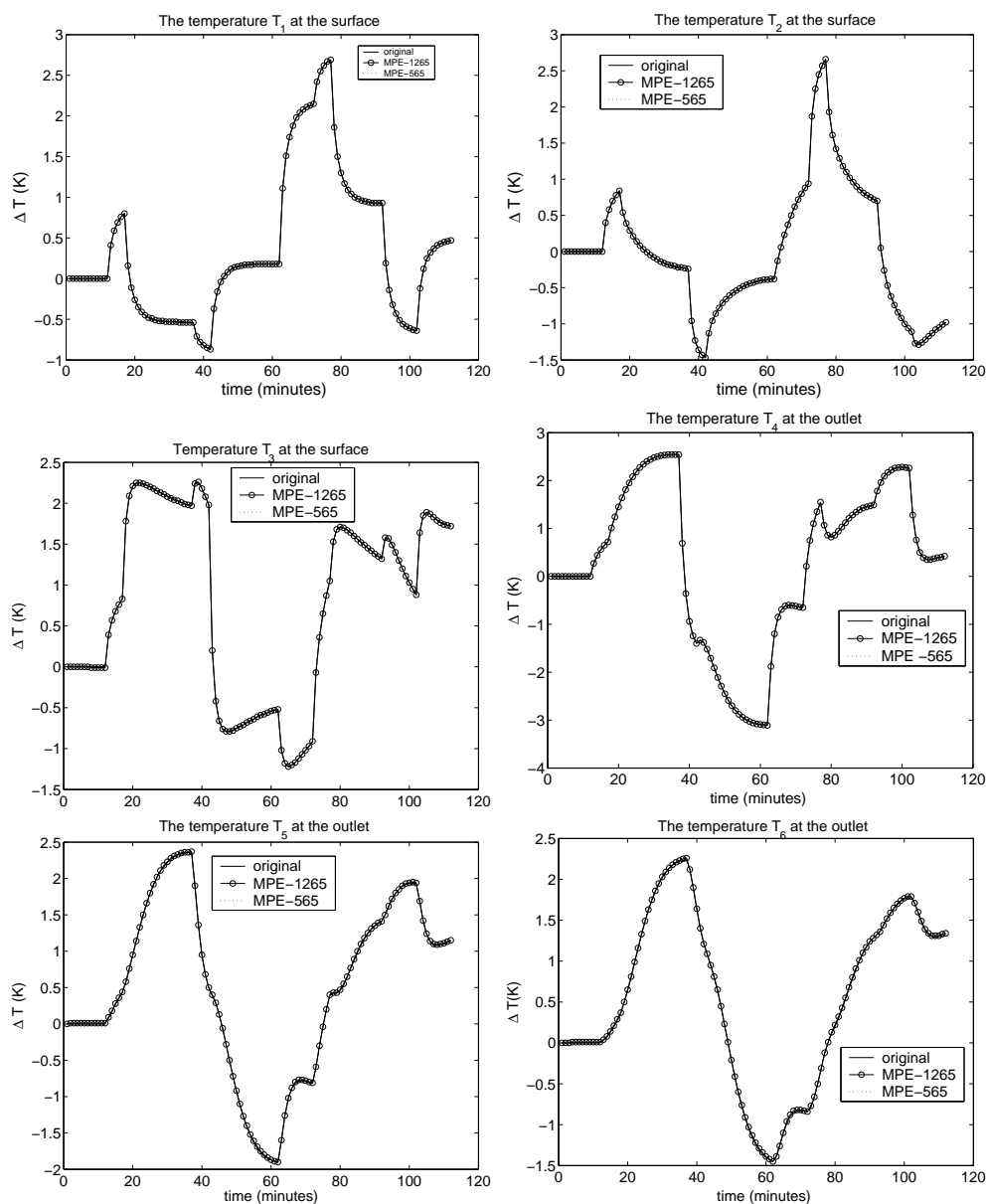


Figure 5.21: The temperatures modeled by the original and POD-MPE models at the measurement points T_1 to T_3 (see its configuration in Figure 5.10) and the temperatures at the measurement points T_4 to T_6 (see its configuration in Figure 5.12)

5.5.4 Application of MPE based on selection criterion 2

The point selection by criterion 2 is similar to point selection in criterion 1, but in criterion 2, there is weighting by the POD coefficients (see section 4.3.1). Suppose the POD basis Φ comprises of n POD basis functions is derived from the snapshot data $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{K \times L} = (\mathbf{T}(1) \ \mathbf{T}(2) \ \dots \ \mathbf{T}(L))$.

We derive a matrix \tilde{J} whose (i, j) -th entry is given by:

$$\tilde{J}_{ij} = \mathbf{a}^\top(t_i) \Phi^\top \Phi \mathbf{a}(t_j) \quad (5.26)$$

where

$$\mathbf{a}(t_j) = \Phi^\top \mathbf{T}(t_j)$$

There is $\mathbf{a}(t_i)$ in (5.26) to indicate the different time step. The matrix \tilde{J} is an approximation to $\mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$ where $\mathbf{T}(t_j)$ in every column of \mathbf{T}_{snap} is replaced by (5.1).

In criterion 2, we define for each point located at $x_k \in \mathbb{X}$, an $L \times L$ matrix $E_J(x_k)$ whose (i, j) -th entry is given by:

$$E_{J_{ij}}(x_k) = \tilde{J}_{ij} - \mathbf{a}^\top(t_i) \tilde{\Phi}(x_k)^\top \tilde{\Phi}(x_k) \mathbf{a}(t_j)$$

where the elements of $\tilde{\Phi}(x_k) \in \mathbb{R}^{1 \times n}$ are the elements of Φ correspond to every location $x_k \in \mathbb{X}$.

For every grid point located in \mathbb{X} , we calculate

$$e_{x_k} := \| E_J(x_k) \| \quad (5.27)$$

where the norm $\| \cdot \|$ is as defined in (5.22).

The point with large e_{x_k} is considered not important and the point with small e_{x_k} is considered important. The points are then re-index as $x_{k_1}, x_{k_2}, \dots, x_{k_K}$ by ordering e_{x_k} such that $e_{x_{k_1}} \leq e_{x_{k_2}} \leq \dots \leq e_{x_{k_K}}$.

The plot of the ordered e_{x_k} is depicted in Figure 5.22.

Note that the difference between the smallest e_{x_k} and the largest e_{x_k} in Figure 5.22 is relatively small, but to determine \mathbb{X}_0 , a starting point is needed. The points should be taken so that the condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$ is close to the condition number of $\Phi^\top \Phi$. The condition number is calculated by implementing Procedure 5.5.1. The plot of the condition number is shown in Figure 5.23.

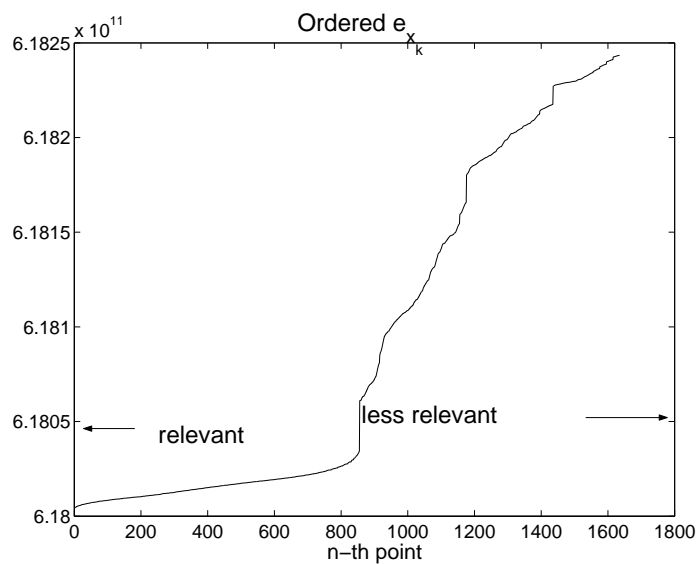


Figure 5.22: Plot of e_{x_k} , to the left correspond to relevant points, to the right to irrelevant points

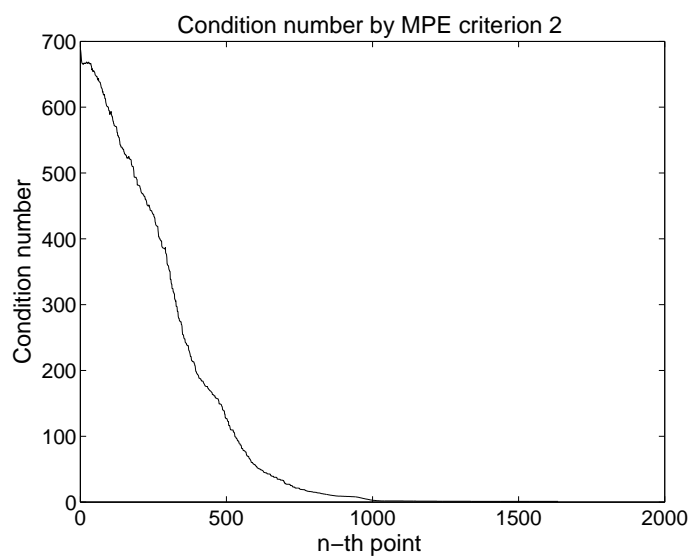


Figure 5.23: Plot of the condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$. The restricted domain \mathbb{X}_0 is constructed from the locations of the 265 obligatory points and the locations of points added to \mathbb{X}_0 according to Procedure 5.5.1

In Figure 5.23, it is shown that the condition number of $\tilde{\Phi}^\top \tilde{\Phi}$ beyond adding 1000 points to the 265 points does not change significantly. In total, there are 1265 points in \mathbb{X}_0 chosen to construct the MPE-based reduced model. The condition number of $\tilde{\Phi}^\top \tilde{\Phi}$ formed by 1265 points is equal to 2.8327. A part of the selected 1265 points are shown in Figure 5.24 at the cross section 10 cm below the glass surface. The acceleration factor with 1265 points is

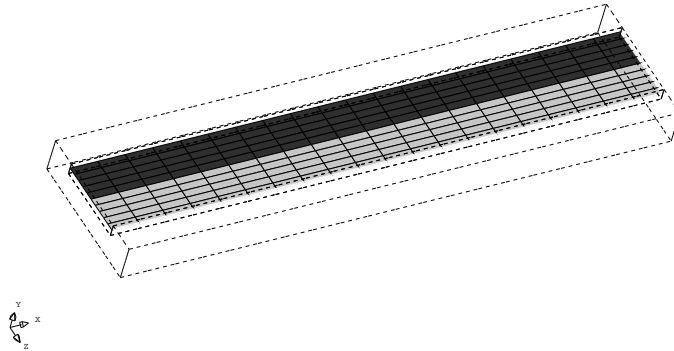


Figure 5.24: Part of the selected 1265 points (shown by the grey grid cells) depicted for the glass melt 10 cm below the glass surface

still not too satisfactory either, the POD-MPE model is only faster 3.35 times than the original model. To optimize the point selection so that not all points which are located closely together are chosen, the greedy algorithm (Algorithm 5.5.2) is applied. We first compose \mathbb{X}_0 from the locations of the 265 obligatory points and the location x_{k_1} with the smallest value of e_{x_k} in (5.27). The second point is chosen to be the one which gives the best condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}$ when added to the already chosen point. This is continued until in total (including the 265 boundary condition), 565 points are taken. The final condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$ with 565 points is 3.18.

Since we selected 565 points, $\tilde{\Phi} \in \mathbb{R}^{565 \times 18}$ and the CFD matrices are also adjusted accordingly so that only parameters of 565 equations are calculated. The chosen points in the glass domain and in the walls are shown in Figure 5.25. The plots are for a half of the plane because we consider symmetry along the feeder width. By comparing Figure 5.25 and Figure 5.24, it is clear that the implementation of greedy algorithm results in a selection of points which are not colocated together. The candidate points are added to \mathbb{X}_0 which gives the best condition number of $\tilde{\Phi}^\top \tilde{\Phi}$ when combined with the already chosen points. This implies that the added point is independent of the already chosen points. Since in CFD models every grid point is a function of its neigh-

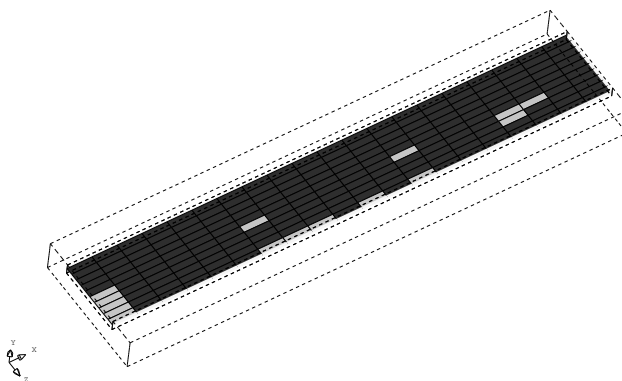


Figure 5.25: Chosen points (shown by the grey grid cells) at the glass melt 10 cm below the glass surface

boring points, the requirement implies that the independent points are not located in the close neighbourhood.

Similar to the case of POD-MPE model by criterion 1, the validation of POD-MPE models based on the selection criterion 2 is also conducted by varying the crown temperature as shown in Figure 5.19. Figure 5.26 show the comparisons between the original (symmetrical) model, the POD-MPE model of Criterion 2 with 1265 points and the POD-MPE model optimized by the greedy algorithm with 565 points. The comparisons are conducted for the measurement points T_1 to T_3 at the surface and T_4 to T_6 at the outlet.

Since the number of points taken in POD-MPE model derived from points selected by Criterion 2 is also the same as in the case of the number of points selected by Criterion 1, the acceleration factor is also the same. The maximum absolute error average (the maximum of (5.16) over all $x \in \mathbb{X}$) is also in the same range. The computational gain of the POD-MPE model with 565 points corresponds to 7.5 times faster than real time. In the computation, about 120 iterations have to be performed for each time step because the temperature field also has to converge such that the mass, momentum, and energy balances are satisfied. If the number of iterations is reduced, then the deviation from the original temperature field will be larger. If the maximum absolute error is allowed to be larger for example to 0.1°C , then the number of iterations can be reduced to about 50 times iterations for each time step. The convergence criterion applied in the simulating software needs to be investigated further as

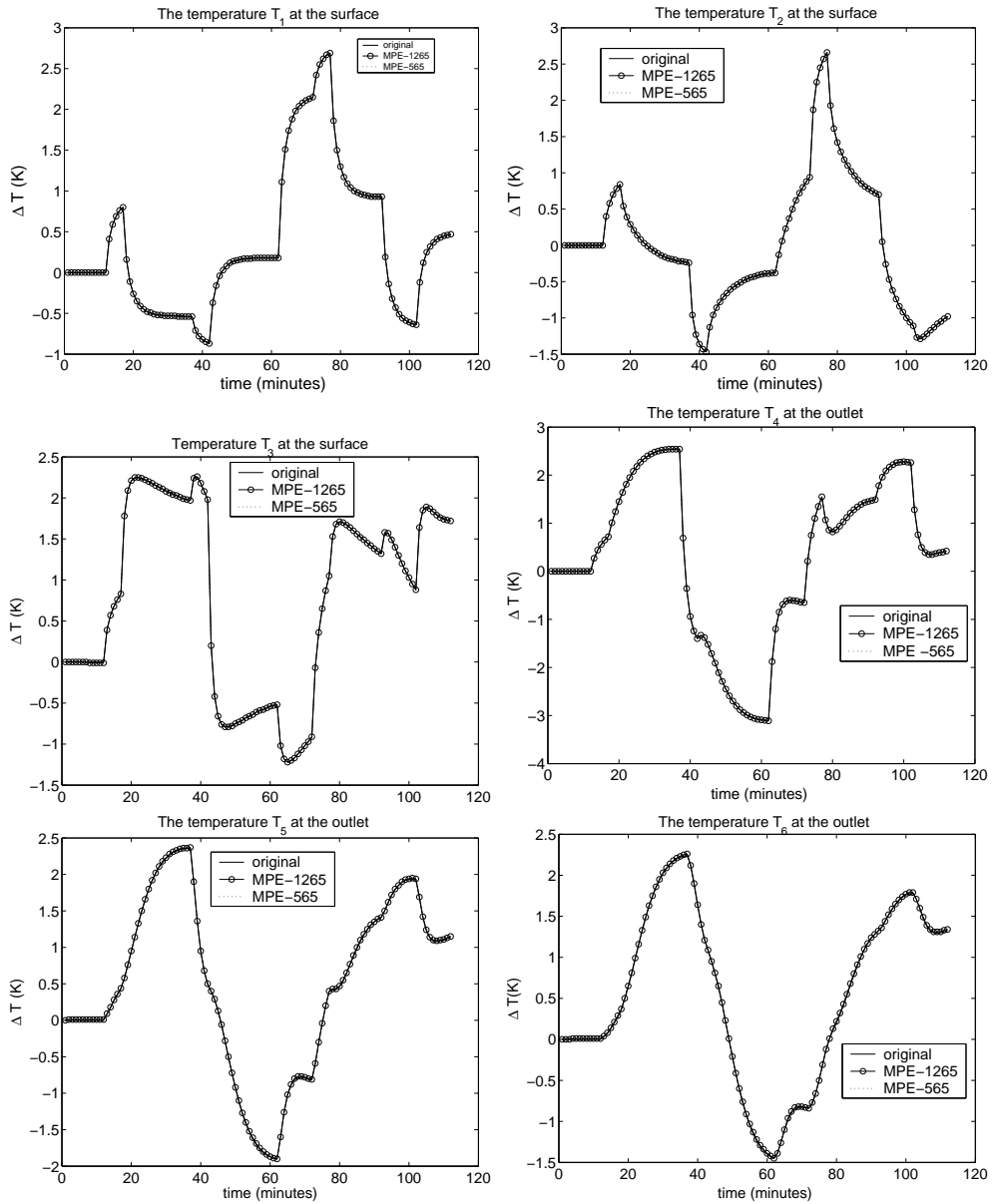


Figure 5.26: The temperatures changes (ΔT is the temperature change from the initial condition) modeled by the original and POD-MPE models Criterion 2 at the measurement points T_1 to T_3 (see its configuration in Figure 5.10) and at the measurement points T_4 to T_6 (see its configuration in Figure 5.12)

Table 5.2: Comparison between POD and MPE models

| Model Type | Maximum Absolute Average Error (Maximum of (5.16) over all $x \in \mathbb{X}$) | Computational Gain |
|--------------|--|--------------------|
| POD | 0.007° C | 226% |
| POD-MPE-1265 | 0.007° C | 335% |
| POD-MPE-565 | 0.012° C | 620% |

well so that the optimal number of iterations can be reduced.

5.5.5 Comparison between selection criterion-1 and selection criterion-2

The MPE selection criterion 1 is based on the deviation of $\tilde{\Phi}^\top \tilde{\Phi}$ from $\Phi^\top \Phi$. In Criterion 2, the point selection is based on the weighting of the POD basis by the POD coefficients. In short, MPE-criterion 1 (without the implementation of the greedy algorithm) groups points which have similar variations in the POD basis functions.

In MPE-criterion 2, the basis functions are weighted by the POD coefficients. Since the temperature data at every time step is approximated by the POD basis functions weighted by the POD coefficients at every time step ((5.1)), this criterion is related to the temperature data. As a result, the MPE-criterion 2 tends to group the points which have the same variations of the temperature data and in the glass feeder, points which are located closed to each other tend to have the same temperature range as well.

Due to the proximities of the selected points by MPE-criterion 2, the condition number of $\tilde{\Phi}^\top \tilde{\Phi}$ from the points selected by the MPE-criterion 2 also decreases slower than the points selected by MPE-criterion 1. Figure 5.27 shows the plots of the condition number (obtained by conducting the procedure 5.5.1 for points selected by MPE-criterion 1 and MPE-criterion 2).

By comparing Figure 5.24 and Figure 5.17, it is also obvious that the selection criterion-2 groups the points which are closely located to each other and the distribution of the selected points is more spread out in the points selected by criterion 1 than the points selected by criterion 2. As a result, the condition number of $\tilde{\Phi}(\mathbb{X}_0)^\top \tilde{\Phi}(\mathbb{X}_0)$ decreases faster by criterion 1 than by criterion 2 since the selected points are more independent of each other due to the more

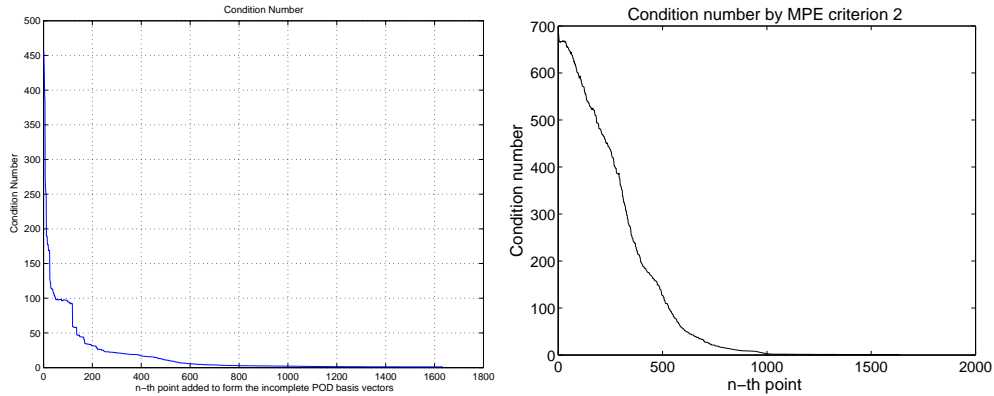


Figure 5.27: Plot of condition number of $\tilde{\Phi}^T \tilde{\Phi}$ by criterion 1 (left) and criterion 2 (right)

distant locations.

5.6 Simulation of the glass colour change in the feeder

A glass melt feeder is not restricted to produce only one specific type of glass. The same glass feeder is used to produce glass of different colours and different weights. The change of product type is normally called a *job change*. In this section, reduced order modeling of the temperature changes during colour change in a glass melt feeder is discussed. Initially, the feeder contains green container glass. A new glass product with different colour will be produced by replacing the green container glass by the flint (uncoloured) container glass melt.

The physical properties [59] of the green and flint glasses are enlisted as follows: Properties of the flint container glass melt which differ significantly from the green container glass are the heat conductivities and the absorption coefficients. Since the green container glass has a darker colour than the flint container glass, it absorbs more light than the flint container glass. For the same temperature range, the green container glass also has a lower effective conductivity than the flint container glass. Darker colour preserves more thermal energy in the material.

If a new glass product is replacing the existing glass, then initially the con-

Table 5.3: Physical parameters for the green and flint container glass nekr (T is the temperature in Kelvin)

| Parameters | Green Container |
|---|---|
| Density [66] ρ (kg/m ³) | $2540 - 0.14T$ |
| Viscosity [11] μ (Ns/m ²) | $10^{-2.592} + \frac{4242.904}{T-541.8413}$ |
| Specific heat c_p (J/kg.K) | $1222 + 0.0957T$ |
| Heat conductivity [11] κ (W/mK) | $0.527 + 0.001T + 1.8 \times 10^{-9}T^3$ |
| Absorption coefficient [11] | 367.859 |
| Surface emissivity [26] | 0.89 |

Table 5.4: Physical parameters of the flint container glass (T is the temperature in Kelvin)

| Parameters | Flint Container |
|---|---|
| Density [66] ρ (kg/m ³) | $2536 - 0.14T$ |
| Viscosity [11] μ (Ns/m ²) | $10^{-2.490} + \frac{4094.950}{T-553.2733}$ |
| Specific heat c_p (J/kg.K) | $1220 + 0.0957T$ |
| Heat conductivity [11] κ (W/mK) | $0.527 + 0.001T + 2.54 \times 10^{-8}T^3$ |
| Absorption coefficient [11] | 26.029 |
| Surface emissivity [26] | 0.89 |

centration of the new glass is zero for the whole feeder except at the feeder entrance. As the new glass melt disperses into the existing glass, the concentration of the new glass will increase. The physical properties such as heat conductivity will change as well as function of the concentration of both the new and the old glass.

5.6.1 Collection of colour change simulation data

The colour change will significantly change some physical properties of the glass melt. In the case of the green container glass melt replaced by the flint container glass melt, the heat conductivity will change by a factor ± 8 . The reduced order model also has to take into account these significant changes. If the reduced order model is derived from simulation data where the colour change from green to flint is not simulated, then the reduced order model cannot be expected to be able to simulate the colour change phenomena. Note that in general colour changes from green to other colour than flint will result in completely different temperature distribution. Reverse colour change such as from flint to green will also change the temperature distribution. Thus,

every possible colour change process in the feeder should be also incorporated in the simulation data.

In this example, reduced order modeling of the temperature field during the colour change simulation is presented. The feeder has the same geometry and the grid cell configuration as in previous examples. The green container glass melt in the feeder is assumed to be initially under the steady state condition with constant pull rate of 80 tons/day and the nominal crown temperature distribution as depicted in Figure 5.5. The crown temperature of each zone is varied as shown in Figure 5.28.

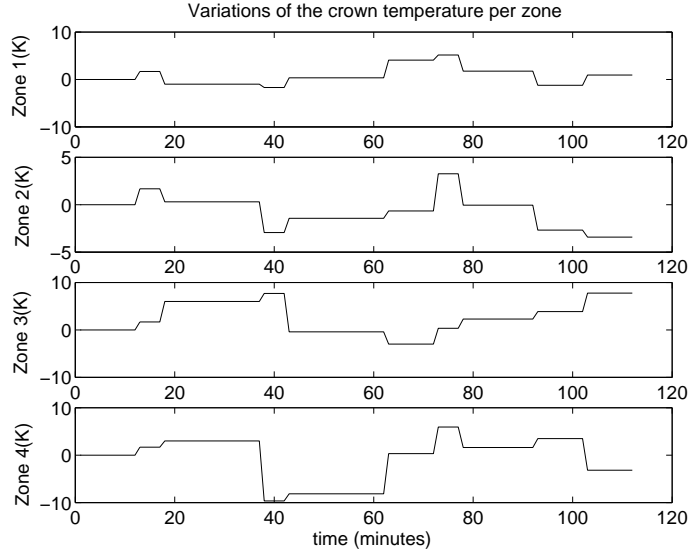


Figure 5.28: The crown temperature variations of each zone

The green container glass melt is then replaced by the flint container glass melt at $t = 0$ and the simulation runs for 112 minutes with sampling time of 1 minute. To derive the reduced order model of the temperature field, 112 temperature data collected in the snapshot matrix $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{3800 \times 112}$ where $\mathbf{T}_{\text{snap}} = (\mathbf{T}(1) \ \mathbf{T}(2) \ \dots \ \mathbf{T}(112))$.

The POD basis functions Φ are the eigenvectors of $\mathbf{C} = \frac{1}{112} \mathbf{T}_{\text{snap}}^T \mathbf{T}_{\text{snap}}$. The eigenvalue spectrum of \mathbf{C} is shown in Figure 5.29

For the colour change simulation, 18 POD basis functions, correspond to 18 largest eigenvalues are taken. This corresponds to $1 - P_n = 1 - P_{18} \approx 10^{-10}$.

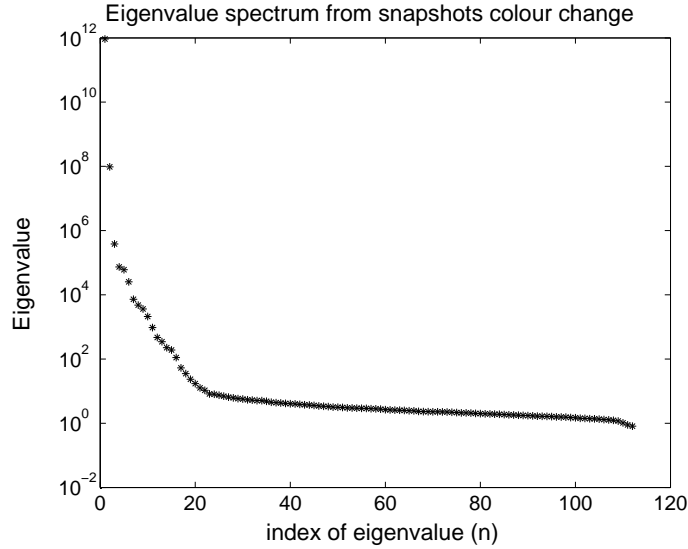


Figure 5.29: The eigenvalue spectrum from the eigenvalue decomposition of the colour change simulation

The reduced order model is obtained by employing Galerkin projection of the 18 POD basis functions $\Phi = (\varphi_1 \ \varphi_2 \ \dots \ \varphi_{18})$ onto the original model describing the temperature distribution.

Figure 5.30 shows the comparison between the results of the reduced order model and the original model for the measured temperature profiles at the glass melt surface (configuration of the measurements is shown in Figure 5.10) and at the outlet (configuration of the measurements is shown in Figure 5.12). The conditions simulated are the same as in the snapshot generation. From Figure 5.30, it is clear that the reduced order model can capture the dynamics of the original model quite well.

The plot of the average absolute error of every grid point $\bar{\epsilon}(x)$ at location $x \in \mathbb{X}$ is given in Figure 5.31. The average absolute error $\bar{\epsilon}(x)$ is defined in (5.16).

The highest $\bar{\epsilon}(x)$ is about 0.08 K, observed in the glass melt. The temperature variations in the glass melt during the simulation is about 20 K. Hence, the deviation of the reduced model from the original model accounts for less than 0.5% of the temperature changes. The changes of the calculated heat conductivities due to the mixing of green container glass with flint container glass are shown in Figure 5.32. The operating temperature of the glass melt feeder

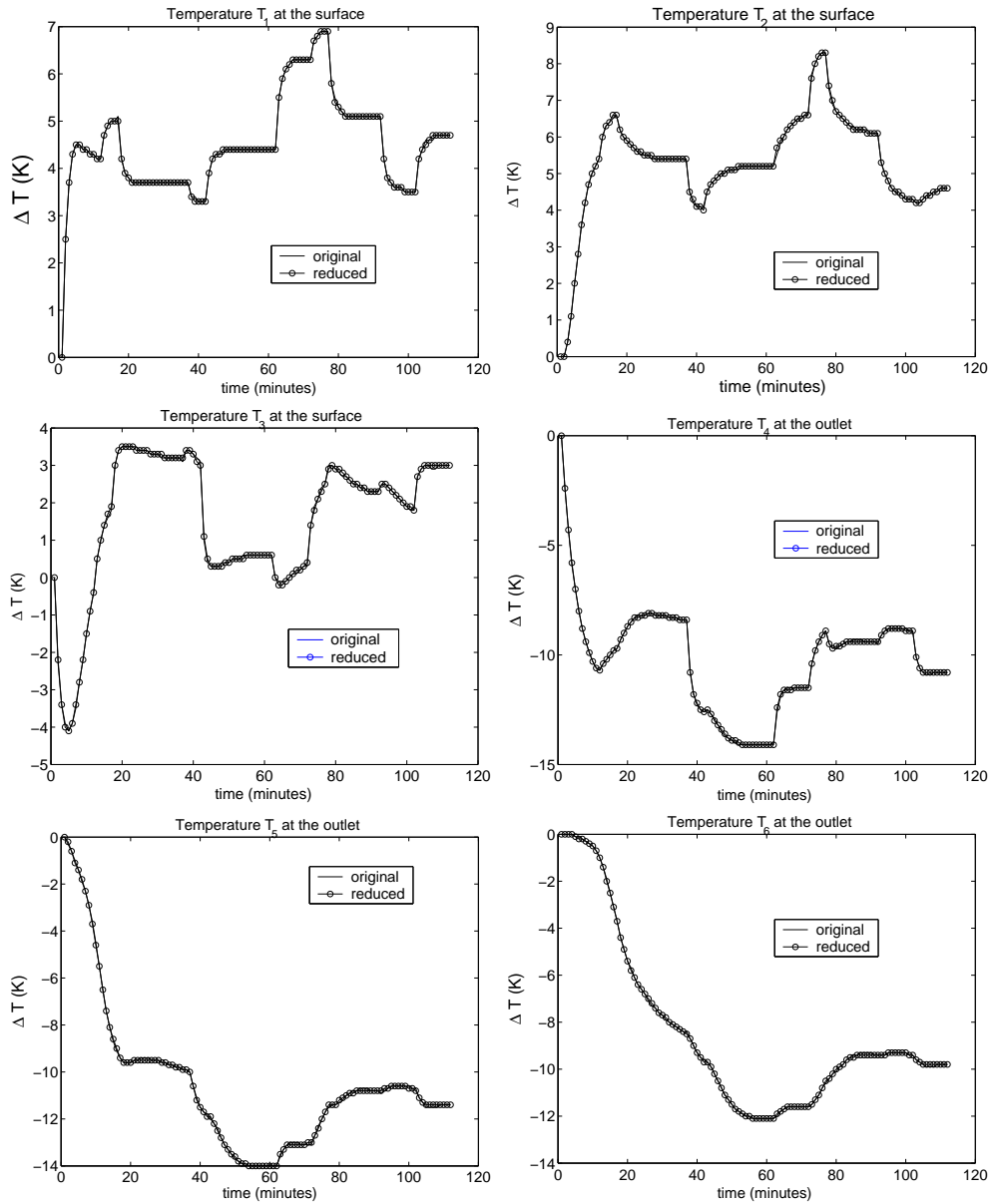


Figure 5.30: Reduced and original temperature profiles during colour change process at the measurement points T_1 to T_6 . The points T_1 to T_3 are the measurement points at the surface (Figure 5.10) and the points T_4 to T_5 are the measurement points at the outlet (Figure 5.12)

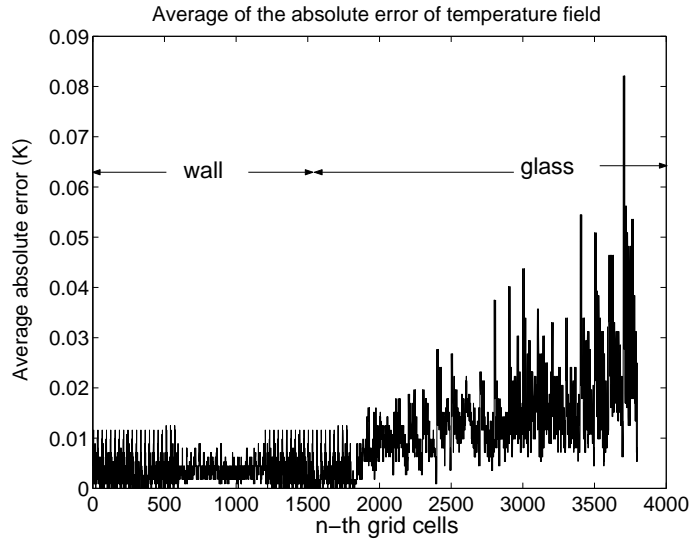


Figure 5.31: The plot of the average absolute error for both grid cells of the walls and the glass media

is around 1480 K and the effective heat conductivity of the green container glass is varying around 7 W/m.K. For this operating temperature, the flint container glass melt has an effective heat conductivity of around 84 W/m.K. Figure 5.32 shows how the heat conductivity changes from around 7 W/m.K to heat conductivity of the flint glass melt. The heat conductivity of every measurement location as plotted in Figure 5.32 is calculated from the temperature calculated by the original and the reduced order model. It is clear from Figure 5.32 that the reduced order model can follow the drastic change of the heat conductivity distribution.

In the simulation, the crown temperature is also varied. At the measurement points in the glass surface (T_1 to T_3), the effect of crown temperature is more evident. Although the heat conductivity also changes in the surface, the heat transfer from the crown temperature is more dominant than the temperature does not drop dramatically. At the measurement point outlet bottom (T_6), the effect of crown temperature is small and the increase of the heat conductivity makes the temperature drop quite significantly at this point.

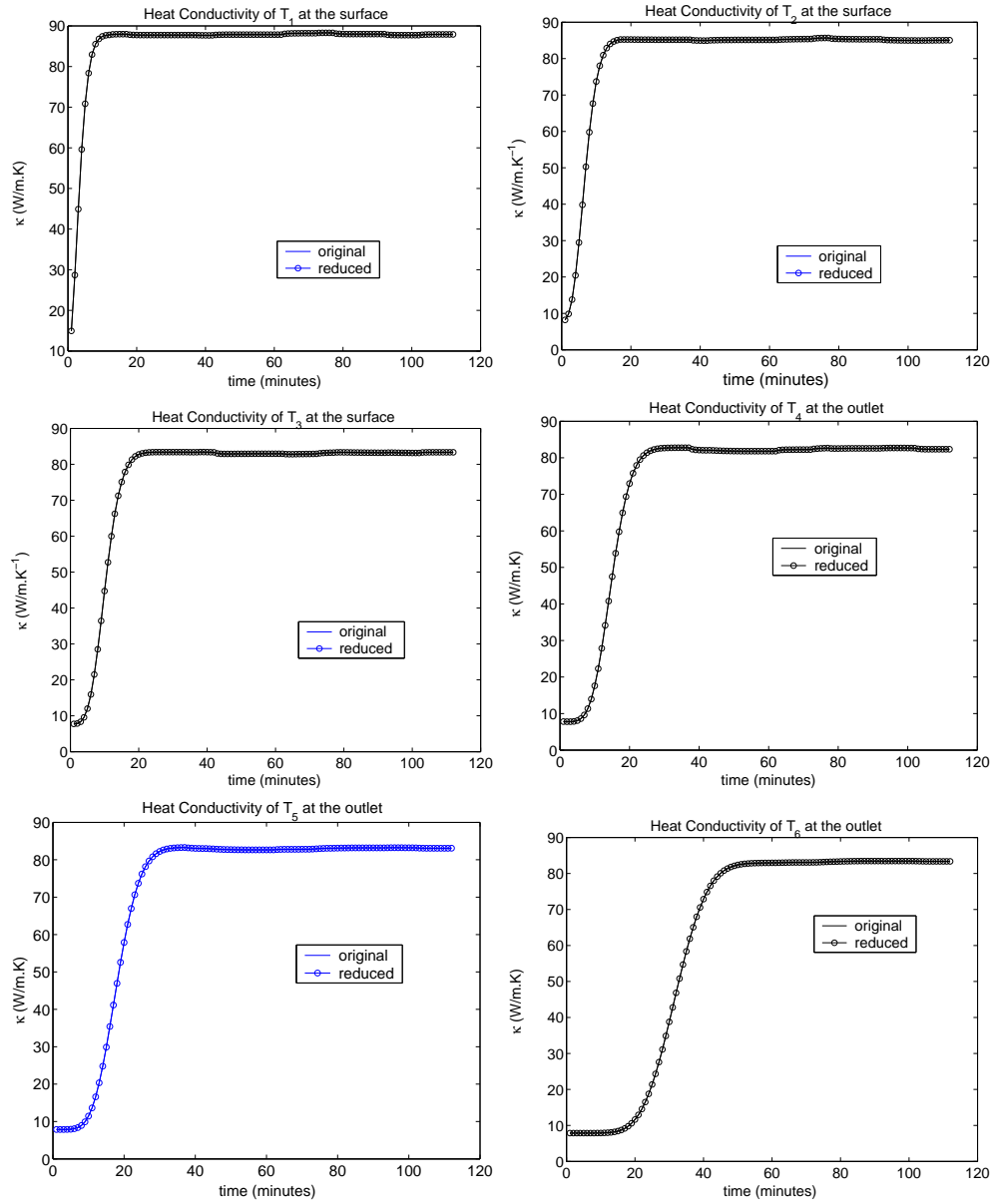


Figure 5.32: Heat conductivity calculated by the temperature distribution from the original and the reduced models at the measurement points T_1 to T_6 . T_1 to T_3 are the measurement points at the surface (shown in Figure 5.10) and T_4 to T_6 are the measurement points at the outlet (shown in Figure 5.12)

5.6.2 Acceleration of the model simulating colour change in feeder

The reduced order model of the temperature changes during the colour change process is 2.26 times faster than the original model. The reduced order model saves the computational effort needed to find the solution of the discretized equations. However, since the original model is a set of nonlinear PDE's, the parameters of the discretized equations also have to be updated. To construct a faster reduced order model, MPE (Missing Point Estimation) is applied again in this case.

Before the selection of points are made, similar to the previous cases, the grid (obligatory) points which are adjacent to the boundary cells defining the crown temperature, inlet temperature and pull rate are taken first. The remaining points are taken based on the MPE criteria as proposed in section 4.3.1 and section 4.3.2. The feeder model is symmetrical along the width (z -direction), therefore from 3800 points, only 1900 points are considered. There are 265 obligatory points, so the remaining points have to be chosen from the remaining 1635 points.

Application of the MPE selection criterion 1

In this approach (refer to section 5.5 for previous examples and section 4.5 for theoretical background), the deviation $\tilde{\Phi}^\top \tilde{\Phi}$ from the complete POD basis inner product $\Phi^\top \Phi$ is considered.

The ordered e_{x_k} as calculated in (5.21) is shown in Figure 5.33.

Based on Figure 5.33, the mask \mathbb{X}_0 is constructed from the locations of the 265 boundary points and the extra points with smallest e_{x_k} . The number of extra points is chosen such that the condition number of $\tilde{\Phi}^\top \tilde{\Phi}$ constructed by the points located in \mathbb{X}_0 is close to the condition number of $\Phi^\top \Phi$. Figure 5.34 shows the condition number calculated according to the Procedure 5.5.1. The condition number plot shows that there is no real asymptotic decrease of the condition number until the extra 400th point where the condition number is about 7. The plot of the condition number flattens between 400 and 1000 points. There are extra 400 points taken based on this criterion. In total, together with the 265 boundary points, there are 665 points in \mathbb{X}_0 .

The chosen points based on this criterion is shown in Figure 5.35 for a cross section of the glass melt.

The number of points, 665 points can be further reduced by implementing the

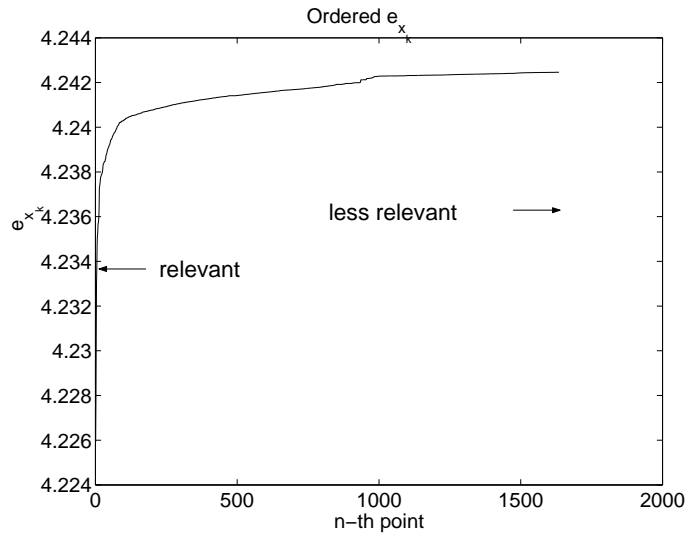


Figure 5.33: The plot of the ordered e_{x_k} as calculated in (5.21) for colour change simulation

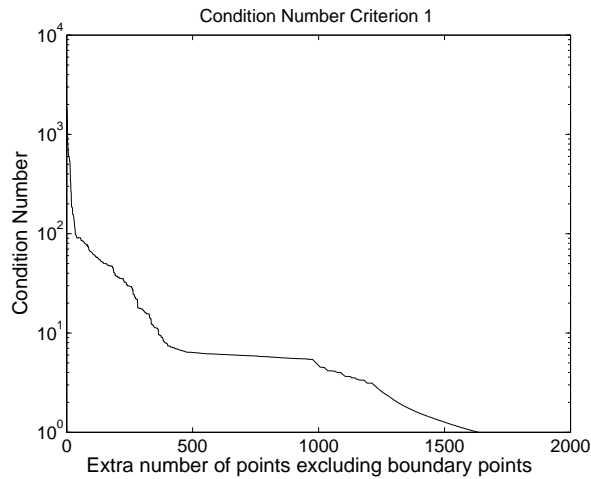


Figure 5.34: The plot of the condition number of $\tilde{\Phi}\tilde{\Phi}$ with $\tilde{\Phi}$ is the restriction of POD basis Φ at \mathbb{X}_0 . The restricted domain \mathbb{X}_0 is constructed from the procedure 5.5.1

greedy algorithm (Algorithm 5.5.2).

After implementation of the greedy algorithm, 200 extra points are selected and combined with the 265 boundary points. In total, there are 465 points

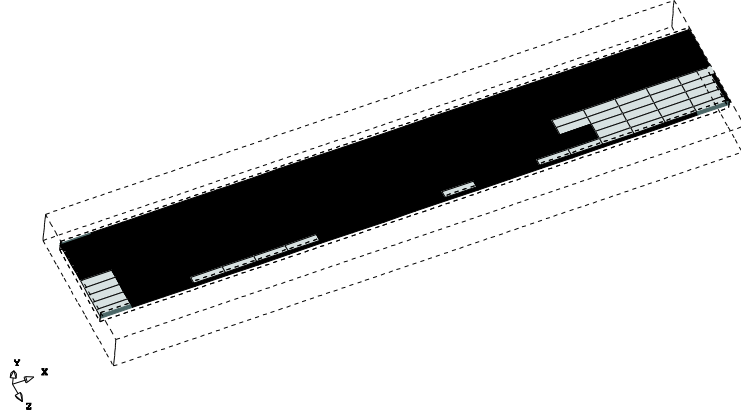


Figure 5.35: The selected points (shown by the grey grid cells) based on MPE-Criterion 1 in the glass melt 10 cm below the glass surface.

taken for the POD-MPE reduced order models. The corresponding condition number of $\tilde{\Phi}^\top \tilde{\Phi}$ is 4.637.

The comparisons between the original model, POD model with 18 basis functions, POD-MPE with 665 points and 18 basis functions, and POD-MPE with 465 points and 18 basis functions are depicted in Figure 5.36. For all models, symmetry is considered.

Table 5.5 shows the maximum error average (calculated by (5.31)) and the resulting computational gain with respect to the computing time of the original model. From the results tabulated in table 5.5, it is clear the POD-MPE

Table 5.5: Comparison between POD and POD-MPE models

| Model Type | Maximum Average Absolute Error (Maximum of (5.16) over all $x \in \mathbb{X}$) | Computational Gain |
|-------------|--|--------------------|
| POD | 0.081° C | 226% |
| POD-MPE-665 | 0.082° C | 527% |
| POD-MPE-465 | 0.13° C | 754% |

model can still follow the dynamics of the original model very well. The resulting reduced order model with MPE based on 465 points is 7.54 times faster than the original model and this corresponds to 8.5 times faster than

the real time because only *temperature* is reduced. This acceleration factor is not adequate for the application of model based predictive controllers (MPC) which require a model which is more than 50 times faster than real time. If other process variables such as velocities are also reduced, then computational gain of about 30 times faster than real time is feasible.

To enhance the computational gain with respect to real time without reducing other process variables, the model can be simulated in parallel processors that the calculation of the model parameters and the construction of the reduced matrices can be divided into several processors. For control design, it is desirable to have a low order model because it is going to be more tractable for the optimization modules. Even though the original model is computed fast, but if the state dimension is large, then solving the optimization problem will still be computationally demanding.

Application of MPE Criterion 2

As in the case of selection criterion 2, from the 1635 remaining points, the quantity e_{x_k} is calculated from (5.27) for each point and ordered. The plot of e_{x_k} is shown in Figure 5.37. There is a steep change starting from the 1400th point, implying that beyond 1400th points, the remaining points are no longer important according to selection criterion 2.

The condition number of $\tilde{\Phi}^T \tilde{\Phi}$ constructed from the 265 boundary points and the extra points based on the order of e_{x_k} is shown in Figure 5.38

Compared to the condition number resulted from the ordering of criterion 1 (Figure 5.34), the condition number of the $\tilde{\Phi}^T \tilde{\Phi}$ based on criterion 2 is decreasing slower. Based on Figure 5.38, 1200 extra points are taken. In total, there are 1465 points selected and this corresponds to a condition number of $\tilde{\Phi}^T \tilde{\Phi}$ equals 24.2553.

To compare the application of criterion 1 and criterion 2 in the colour change simulation, compare Figure 5.39 and Figure 5.35. Figure 5.39 shows the points selected by taking the first 400 points with the smallest e_{x_k} calculated by (5.27) and the obligatory points while Figure 5.35 shows the points selected by taking the first 400 points with the smallest e_{x_k} calculated by (5.21) and the obligatory points. It is clear the points chosen based on the selection criterion 2 are located very closely together. Since they are located very close to each other, they are dependent on each other and this leads to strong linear dependency when they are combined together to form \mathbb{X}_0 .

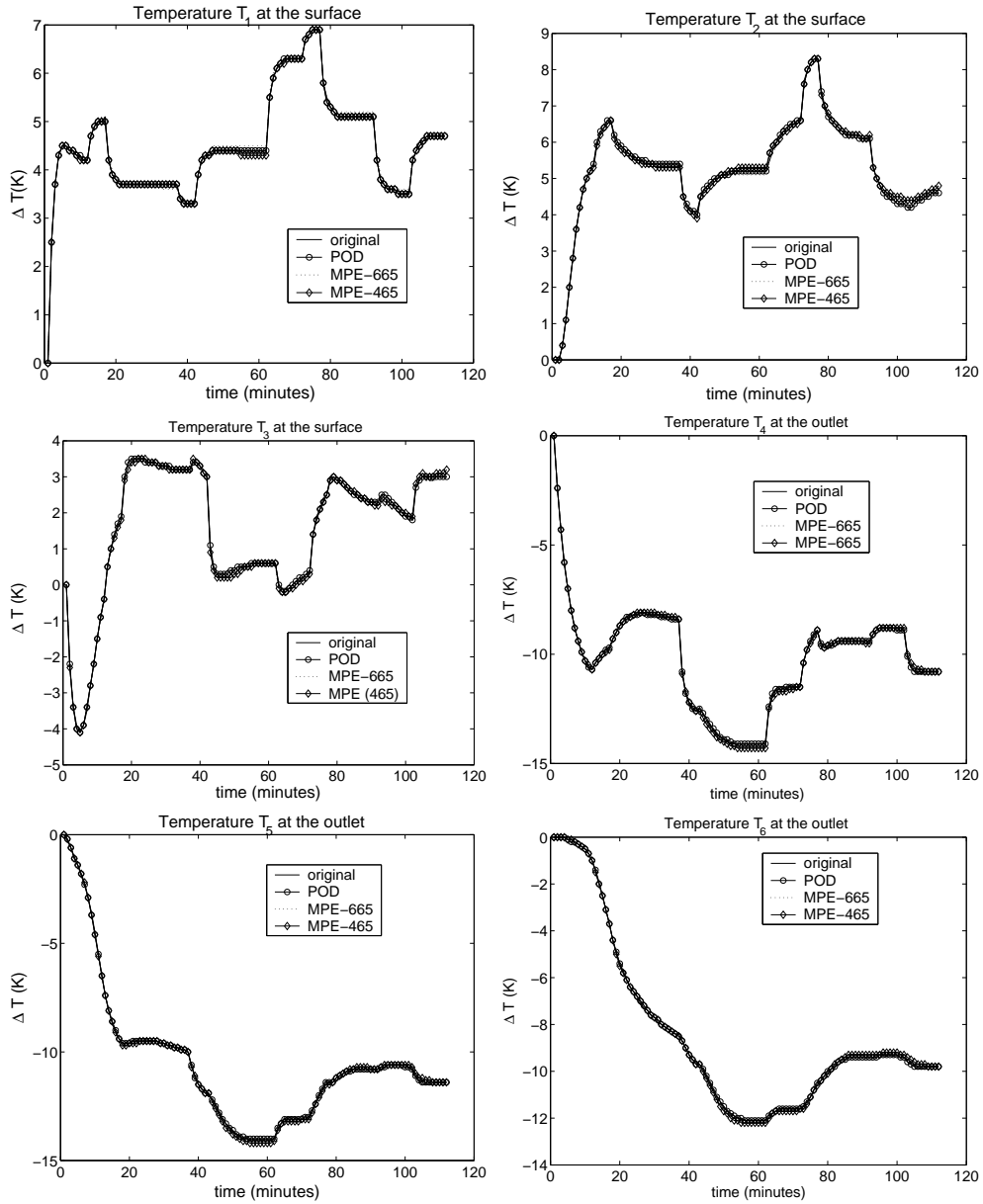


Figure 5.36: Reduced and original temperature profiles at the measured points T_1 to T_6 during the colour change simulation. The reduced order models are the POD, the POD-MPE with 665 points and the POD-MPE with 465 points. The POD-MPE models are built from points selected by Criterion 1

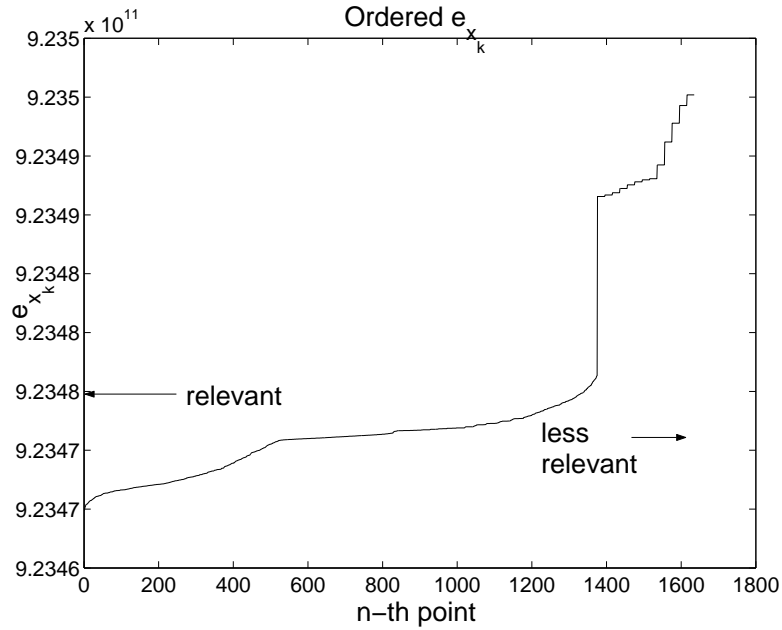


Figure 5.37: The plot of the ordered e_{x_k} based on the criterion 2 of point selection. The quantity e_{x_k} is calculated for each of the remaining 1635 points (5.27)

To reduce the number of points required to construct the POD-MPE reduced order model, the greedy algorithm is applied again. The first mask \mathbb{X}_0 is constructed from the location of the 265 boundary points and the first point with minimum e_{x_k} calculated by (5.27). The next point is added to \mathbb{X}_0 which gives the best condition number of $\tilde{\Phi}^T \tilde{\Phi}$ when combined with the already chosen \mathbb{X}_0 .

From the implementation of the greedy algorithm (Algorithm 5.5.2), 200 extra points are taken. This corresponds to condition number of $\tilde{\Phi}^T \tilde{\Phi}$ of 4.5306.

Table 5.6 enlists the maximum of average absolute error and the computational gain obtained by employing the various reduced models. The maximum of the average absolute error of POD-MPE models constructed from 1465 points are worse than the POD-MPE models constructed from the 665 points with selection criterion 1 since the condition number $\tilde{\Phi}^T \tilde{\Phi}$ for 1465 points based on selection criterion 2 is higher than the condition number of selection criterion 1. As a result, the estimation of the temperature field by selection criterion 2 is worse than the estimation based on less points by selection criterion 1 but

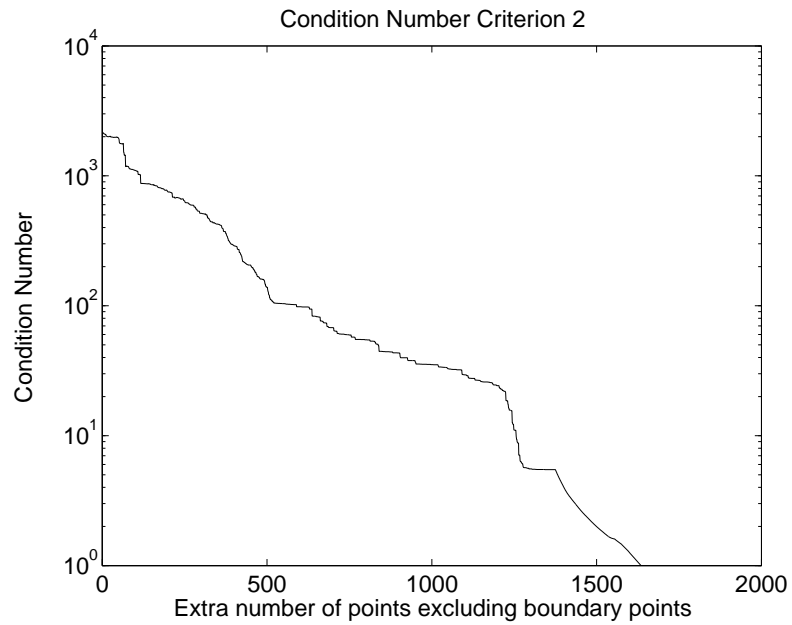


Figure 5.38: The plot of the condition number constructed from 265 boundary points and the extra points based on the ordering of E_J

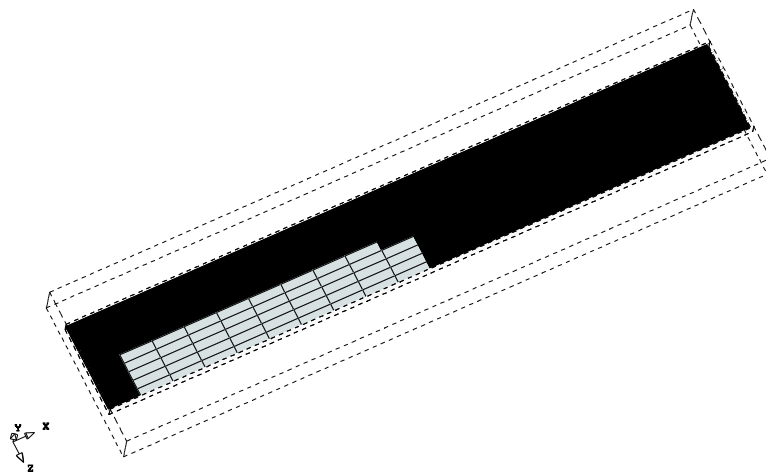


Figure 5.39: Some of the 665 chosen points (shown by the grey grid cells) based on Criterion 2 at the cross section of the glass melt 10 cm below the glass surface

Table 5.6: Comparison between POD and POD-MPE models

| Model Type | Maximum Average of Absolute Error (Maximum of (5.16) over all $x \in \mathbb{X}$) | Computational Gain |
|--------------|---|--------------------|
| POD | 0.081° C | 226% |
| POD-MPE-1465 | 0.1750° C | 289% |
| POD-MPE-465 | 0.1866° C | 754% |

with better condition number.

Figure 5.40 shows the comparisons of the temperature fields obtained by the original, POD, POD-MPE selection criterion 2 with 1465 points and POD-MPE selection criterion 2 with 465 points. In conclusion, selection criterion-2 is more prone to linear dependency compared to the selection criterion-1. This is due to the fact that the selection in criterion 2 is related to the approximation of the temperature data. As a result, only the points varying in the same range will be chosen and normally these points are closely located together.

5.6.3 Performance of MPE-based reduced models

In contrast to other numerical acceleration techniques [16], where the acceleration is focused only on how to solve a high-dimensional discretized equations, MPE enables characterization of the important points. In turn, it enables building an approximating models based on the governing equations of the several points only.

The acceleration factor of the POD-MPE reduced models with respect to real time depends on the structure of the original model. For example, the solver module which comprises the subroutines calculating the model parameters and the subroutines solving the discretized equations consumes 75% of the total computing time. If the total computational time is 2 times faster than real time, then manipulation of the solver modules by MPE method can only lead to a reduced order model which is maximum 8 times faster than real time.

To enhance the computational gain of the reduced order model, the computational gain of the original model has to be enhanced as well. Currently many computational packages also have features for parallel programming, where calculation of model parameters can be conducted in parallel.

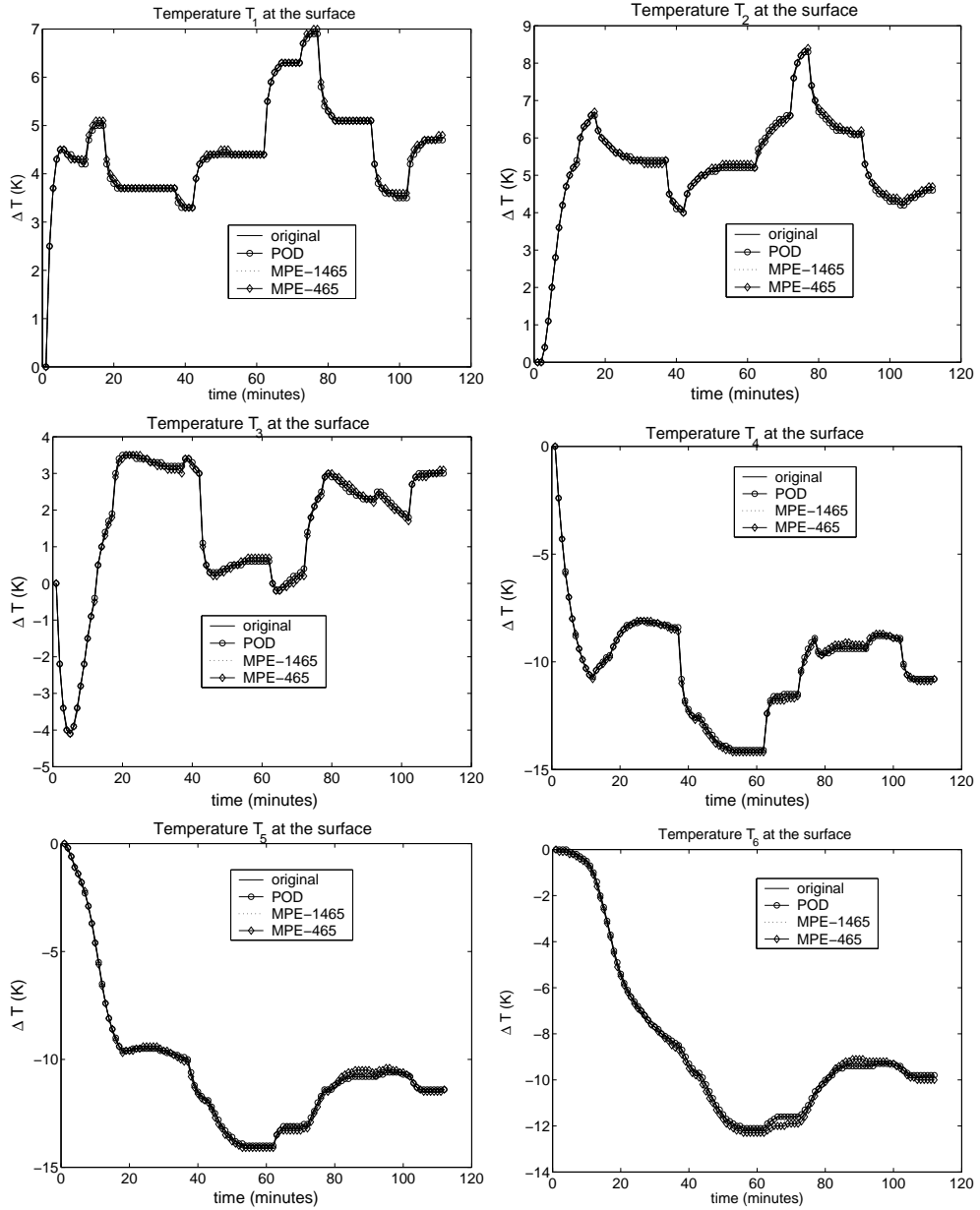


Figure 5.40: Reduced and original temperature profiles at measured points T_1 to T_6 . Reduced order models are the POD, the POD-MPE with 665 points and POD-MPE with 465 points. The points are selected by MPE Criterion 2

If many sequential computing procedures can be replaced by parallel computing, then there will be significant increase in the computational gain. The performance of a computing software will also be affected by other factors such as efficient data transfer between subroutines, efficient allocations of computer memory, and efficient data storage. A more detailed investigation on the divisions of the computational processes in the simulation model is required in order to assess the feasibility of using POD-MPE reduced order models for model based control.

From the case studies discussed in this chapter, the final POD-MPE model is 7-8 times faster than real time because only the temperature is reduced. In general, this acceleration factor is adequate for process monitoring but not sufficient enough for the application of Model Based Predictive Controller which normally require a model which is at least 50. Provided the reduction techniques are also applied to velocity fields, then acceleration factor of about 25-30 times faster than real time is attainable. Since the capacity of computer processors are doubled every year, application of Model Based Predictive Controller with nonlinear models as the base for glass furnaces is feasible in the near future.

The POD-MPE models are still taking the changes of the physical properties into account and faster than original model. The controller designs are usually based on the linear models because linear models are very fast since there are no updates of model parameters. On the other hand, the prediction provided by the linear model is usually not reliable when severe nonlinearity exists, for example during glass colour change. In the controller design, a correction factor is usually added to the linear model to take into account the discrepancies between the original model and the estimation from the linear model. The POD-MPE models are faster than the original model and still take the nonlinearities into account. These models can provide a correction factor in the prediction of the linear model.

The correction term $F_{\text{cor}}(t)$ is introduced as function of the discrepancy between the variable \mathbf{T}_{org} calculated by the original model and \mathbf{T}_{lin} calculated by the linear model:

$$F_{\text{cor}}(t) = f(\mathbf{T}_{\text{org}}(t) - \mathbf{T}_{\text{lin}}(t)) \quad (5.28)$$

The simplest function is for instance by introducing a proportional constant $F_{\text{cor}} = K(\mathbf{T}_{\text{org}}(t) - \mathbf{T}_{\text{lin}}(t))$. Alternatively, the correction factors can also be added to handle the discrepancies between the predictions of the POD-MPE models and the linear models. In this way, although the POD-MPE

reduced model may not be fast enough for controller's design, it can improve the prediction quality of the linear model.

5.7 Summary and conclusions

This chapter discusses the reduced order modeling of a glass melt feeder for estimations of the temperature distribution in the glass melt and in the refractory walls. The introduction to glass melt processes in section 5.1. The original model of glass melt feeder canal is given in section 5.2. Proper Orthogonal Decomposition is also applied to original the feeder model and the results are presented in Section 5.4.

Since the nonlinear reduced order model is still computationally expensive, we propose a Missing Point Estimation approach, in which we estimate the dynamics of the reduced states based on the partial equations of the original model. Two selection criteria have been proposed in Chapter 4 (Section 4.3.1) and (Section 4.3.2). The comparisons of the results of both criteria show that criterion 1 yields a set of points which are more independent than criterion 2. It is shown that by estimating POD coefficients based on partial knowledge of the original data, we can still follow the original dynamics reliably. In this chapter, the number of points selected by both criteria is further reduced by implementing the *greedy algorithm*.

In conclusion, the step-by-step procedure of deriving reduced order models for CFD models are as follows:

1. Derive the the governing PDE's of the CFD models and the boundary conditions, defined over a spatial domain \mathbb{X} and over a specific time period;
2. Discretize the PDE's (CFD software produces the discretized equations automatically);
3. Run simulations by implementation of the typical input variations, usually available from the measurement data of the real plant operations;
4. Collect the snapshots of the variables from the original model we would like to approximate by the reduced order model, for example temperature data;

5. Derive the POD basis functions from the snapshot data by implementing Algorithm 2.4.10. If the number of grid cells is larger than the number of time samples, it is recommended to implement the method of snapshots (Appendix A);
6. Derive the model of the POD basis coefficients by implementing the Galerkin projection on the discretized equations (see section 5.3.2);
7. To accelerate the model reduction computations in the case of time varying parameters or nonlinear PDE's, select the points which dominantly characterize the dynamics of the process. Selection can be made based on MPE-criterion 1 (section 5.5.1) or MPE-criterion 2(section 5.5.4). The number of points should be chosen such that the condition number of $\tilde{\Phi}^T \tilde{\Phi}$ constructed from the limited number of points is well below 100.
8. To accommodate the changes of input parameters $\mathbf{u}(k)$, it is essential to include the grid points which are located adjacent to the location of the actuators.
9. If the number of points which have to be taken is still too large or there are too many selected grid points located closed to each other, employ the *greedy algorithm* to optimize the point selection
10. Derive the model of the POD coefficients by conducting the Galerkin projection only on the equations of the selected points.
11. The reduced order model describing the dynamics of the POD coefficients is ready

Conclusions and Recommendations

| | |
|---|--|
| 6.1 POD as a model reduction method | 6.4 Applicability of the developed reduced order modeling techniques |
| 6.2 Acceleration of POD model reduction method | 6.5 Recommendations for future research |
| 6.3 Acceleration of CFD models for glass processing simulations | |

6.1 *POD as a model reduction method*

In this thesis, the method of proper orthogonal decomposition (POD) has been proposed as a model reduction technique for high order models obtained from the discretization of PDE's. Such models are developed in systems for which mass, momentum, and heat transfer processes are simulated. In this thesis, the solutions of PDE's are functions of both spatial and temporal coordinates. Usually, the main bottleneck for fast simulation by PDE-based models is the fine discretization of the spatial domains.

The POD method is characterized by the property to describe the spatial distribution of the relevant physical variables in terms of a set of orthonormal basis functions. These basis functions are selected from observed data and are optimal in a well-defined sense (section 2.4). In the numerical models, the spatial domain is discretized into a high number of grid cells, while in POD models, the spatial distributions are described by the first few and most relevant POD basis functions. The time-dependent characteristics of the variables are given by the time varying coefficients of the POD basis functions. The model of the time varying coefficients is referred to as *the reduced order model*, and is obtained by projecting the POD basis functions onto the original governing equations. Throughout the results presented in this thesis, it is shown that with very few POD basis functions (less than 1% of the num-

ber of grid cells), the temporal and spatial dynamics of the variables can be approximated very well.

The POD basis functions are optimal for the process variable data from which the POD basis functions are derived. This is demonstrated in Chapter 2 when the approximation of a wave propagation model by POD basis functions is compared to the approximation by classical Fourier modes. The POD basis functions are shown to approximate the simulation data better than the Fourier modes.

Since the POD basis functions are derived from the simulation or measurement data of the process variables, the validity of the reduced order model will also depend on the quality of the data set. If the data set sufficiently represents the typical variations of the system, then the reduced order model will be valid for simulations within this operating range. In Chapter 3, it is shown for the one-dimensional heat conduction model that it is very essential to excite all relevant spatial dynamics¹ because the information about the spatial dynamics is incorporated in the POD basis functions. On the other hand, in Chapter 3 we have shown that for diffusive processes, the choice of the time-varying dynamics of the excitation signals does not really affect the quality of the reduced order model.

The reduced order model can be used as a base model for controller design. In Chapter 3, a Linear Quadratic Regulator (LQR) controller has been designed for a two-dimensional heat conduction model on the basis of the reduced order model. Even though the original model is a linear one, it is very difficult to solve the optimization problem with as the model is high order. The control and optimization problem becomes very tractable if the model can be reduced based on a few number of POD basis functions inferred from the open loop data. It is shown in Chapter 3 that the desired temperature distribution can be controlled using the reduced order model as the base-model for the controller.

6.2 Acceleration of POD model reduction method

Although a very small number of POD basis functions can approximate the original variable distribution in the spatial domain, the computational load of the resulting reduced order model is still quite high. In complex nonlinear

¹variations of the variable distribution in the spatial domain, e.g variations of the temperature field in the spatial domain

cases, the reduced order model is often only 2-3 times faster than the original model. This is due to the fact that the reduced order model is built by using the original model and the original model is a nonlinear one. As a result, the parameters of the original models have to be calculated before the reduced order model can be constructed. This is the main bottleneck.

The resulting reduced model usually has significantly lower dimension than the original model that preservation of the sparsity structure is not really an issue here. The matrices in the original CFD models have sparse structures but the dimensions are high that the solutions must be found through longer iterative procedures. Although the matrices of the reduced models have dense structures, the dimensions are much lower that the computational issue is more in the construction of the reduced model rather than in solving a small scale dense system.

To improve the computational efficiency of the reduced order model, the method of missing point estimation (MPE) is proposed. The method enhances the computational effort needed to update the parameters of the reduced order models. The MPE method is based on the estimation of POD coefficients from partial observation or from information of a selected number of positions in the spatial domain [34, 15, 72]. In this thesis, this method has been extended to dynamical systems, where the model of the POD coefficients is derived from a limited number of equations describing the dynamics of a limited number of grid points.

The selection of the grid points has been made in such a way so that the estimations of the POD coefficients are still close to the calculated POD coefficients derived the "conventional" POD method. Two criteria have been proposed to select the points. From the comparisons between the selection criterion 1 and the selection criterion 2 in Chapter 5, the points selected by the selection criterion 2 is more closely located together. This is not efficient since the dynamics of the variables located at the selected grid points are dependent on each other. Therefore from the results of the comparisons, it is better to use the selection criterion 1. The selection of points based on both criteria can be further optimized by applying the greedy algorithm [72] as described in Chapter 5.

Both model reduction methods based on the conventional POD and the combination with MPE are applied to a nonlinear heat conduction model and the resulting reduced order model attains satisfactory computational gain as described in Chapter 4. The reduced order model by POD-MPE is 150 times faster than the original model, while the classic POD model is only about 10

times faster than the original one. The POD-MPE reduced order model is very accurate, the deviation from the original model is also very low. Further, in Chapter 4, the POD-MPE reduced order model is also used as a base model for LQR controller design. Despite the fact that the POD-MPE model is only an approximation of the POD model, the temperature distribution of the plate can be controlled. Hence, POD-MPE is a promising model reduction technique which can lead to a much faster prediction than the conventional model reduction method. The resulting model can also be used as a basis for control and optimization modules.

6.3 Acceleration of CFD models for glass processing simulations

The method of POD-MPE is also implemented on a glass simulation software. The CFD model with the reduction of the *temperature field* of the glass feeder is about 7-8 times faster than real time. This is adequate for process monitoring and online tuning. On the other hand, this acceleration factor is not sufficient for control system design where a computational speed of at least 50 times faster than real time is desired. In the glass simulating software, the model is more complex because more equations (calculation of the convection terms, calculation of material properties, and interfaces with other subroutines for data transfer) contribute to the computational effort. If the reduction techniques implemented in this PhD thesis are extended to include process variables other than temperature (such as the velocity and pressure fields), then a further acceleration can be achieved with a factor that will be more or less proportional to the number of variables. If the velocity fields and pressure are included in the reduction procedure, then an acceleration factor of about 25-30 times faster than real time is feasible. Note that the acceleration factor also depends on the quality of the computing mainframe. Following Moore's Law [46], the capacity of computing processors is doubled every year. Hence, in a near future, design of model predictive control based on nonlinear reduced order models is certainly feasible.

6.4 Applicability of the developed reduced order modeling techniques

In this PhD thesis, the reduced order modeling techniques are applied mainly for heat transfer processes. However, this does not imply that the techniques developed in this thesis are restricted to the domain of heat transfer processes only. The POD and MPE techniques can be applied to reduce models in other application domains such as large-scale electronic systems, aerodynamics and large-scale ODE based systems typically found in mechanical applications. The effectiveness of reduced order modeling developed by the proper orthogonal decomposition approach depends on the smoothness of the data collected from the measurements or simulations. The case studies applied in this PhD thesis all show smooth dynamics, where discontinuities do not occur. If the investigated system exhibits discontinuous dynamics, then the reduced order modeling based on this approach does not work well since it is difficult to obtain a POD basis numerically from the data with discontinuous dynamics.

6.5 Recommendations for future research

The following are recommendations for future research:

1. Application of the POD-MPE model in the controller designs
For feedforward controllers which require fast and long predictions, linear models are still preferable as the linear models are fast and reliable for a certain operating point. If there exists a severe nonlinearity in the ongoing process such as a colour change of glass in glass furnaces, then linear models are not adequate. The POD-MPE model takes the nonlinearities into account and is faster than the original model. In this PhD thesis, the POD-MPE model is applied to reduce the temperature model. If the POD-MPE models are also applied for the velocity fields as well, then acceleration factor of 25-30 times faster than real time is attainable. In that case, development of model-based predictive controller for POD-MPE models will be feasible in the near future.

Alternatively, the POD-MPE model can be used as a representation of the original model even though it is not as fast as the linear model. The information from the POD-MPE model can be used by the controller to improve the prediction of the linear models since predictions provided by the linear model may not be adequate if severe nonlinearities exist.

For example, by introducing a correction term F_{cor} ((5.28)) in the state space equation of the linear model as a function of the discrepancies between the prediction of the POD-MPE models and the linear models.

2. Application of parallel computing

The performance of POD-MPE model is related to the original model. To further assess the benefits of POD-MPE method, there should be clear indications on the the computational burden of the pre-processor, solver, and the post-processor modules. For example, if the solver module consumes 90% of the total computational time, the maximum computational gain which can be attained by applying reduced order model is 10. The computational gain of the reduced models with respect to real time will depend on the computational gain of the original model with respect to real time. If the original model is 10 times faster than real time and the POD-MPE model is 5 times faster than the original model, then this is already adequate for the application of the Model based Predictive Controller.

Application of parallel computing will significantly improve the computational tractability of both the original and the reduced models as the update of the matrix parameters can be divided into several processors. Many computational software packages, including the new generation of the glass simulating software GTM^X developed by TNO-TPD in the Netherlands have parallel computing features. The application of parallel computing will both accelerate the computational time of the original and the reduced order model with respect to real time.

3. Reduced order modeling of multivariable signals

In this thesis, the reduced order modeling is conducted for the temperature field. In the future, it will be desirable to also reduce multivariable signals such as the velocity vector in a 3 dimensional Cartesian coordinate system. In [6], the 3-dimensional velocity field is reduced by individually treating every component of the velocity vector as a variable such as temperature. The POD basis functions are then computed for each velocity component separately and a separate reduced order model is constructed for each component. This is conducted because in the numerical models, the components of the velocity vector are solved sequentially and there are different models built for each component.

The componentwise treatment of multivariable signals may not be optimal. To handle multivariable signal, mathematical grounds of orthonormal basis functions such as the concept of Hilbert space for multivariable signals and the definition of inner product have to be investigated. Since

the components of multivariable signals co-exist and interact with each other, it is desirable to have a set of *multivariable* POD basis functions as well.

Reduced modeling of multivariable signals will also enhance the computational gain of the reduced order model significantly as the reduced order model immediately solves the whole components of the multivariable signal instead solving them sequentially. On the other hand, the algorithm employed currently in the computational software which solves the velocity components separately has to be modified so that the discretized equations governing the multivariable signals such as the velocity components can be solved simultaneously.

4. Robustness analysis of the reduced order models

Throughout this thesis, it is assumed that the reduced order model is valid (the maximum deviation from the original dynamics is below 10%) within the operating range of the collected snapshot data. In the future, it will be beneficial to have a more precise detail about the validity range of the reduced order model in terms of the physical parameters. For example, it will be important to know whether the reduced order model will still be a valid model if the parameters such as heat conductivity or viscosity are varying between specified minimum and maximum values. In nonlinear PDE's, changes of the physical parameters results in the change of the spatial distribution that the POD basis functions may not be able to approximate the new spatial distribution anymore. If the limitation of the reduced order model can be determined in terms of the physical parameter variations, then it will help engineers and operators in the deciding whether a new reduced order model has to be made or whether the existing models are already adequate.

As a first step, uncertainties can be introduced as time varying disturbances which affect the estimations of the time-varying POD coefficients. In control engineering, several structures of time-varying uncertainties of a state space model can already be analyzed that the concepts can also be applied to reduced order model.

5. Stability analysis of the MPE-POD reduced order models

There is no guarantee [57] that if the original model is internally stable, the reduced order model will also be stable. Internal stability here means that given an initial non-equilibrium condition, the system will settle down to the equilibrium condition in the absence of the external inputs. In [57], an analysis of the stability guarantee for POD based reduced order models is given for a class of nonlinear ODE models. In this thesis,

we propose an MPE-POD reduced order model, where the reduced order model is derived from a part of the original states. Since the MPE-POD based models are also used for predictions and controller design, it will be desirable to have a guarantee that the MPE-POD model will also preserve the stability of the full-order model.

6. Use of adaptive POD basis functions

The POD basis functions are guaranteed to be valid only for the snapshot data (from which the basis functions are derived). On the other hand, the reduced order model built from the POD basis functions is also expected to be able to simulate other conditions than the ones collected in the snapshot data. Adaptive POD basis functions are proposed in [34], where the POD basis functions are updated by a neural network algorithm. Alternatively, new POD basis functions can also be constructed from knowledge of several points in the spatial domain [34, 15]. If the POD basis functions can be updated from the knowledge of several measurement data or from the original simulations, the validity range of the reduced order model will also be broadened.

7. Combination of MPE with the Empirical Balancing approach

In POD, all states of the system are considered equally important. In the balanced model reduction technique [62, 39], there is weighting between the states of the system and the desired outputs (usually sensor or locations which considered more important than the others). The computation of the basis functions of the balanced model reduction technique was demanding, but Lall et.al [39] proposed an empirical balancing approach which computed the basis functions in a similar way as the computation of the POD basis functions. The MPE approach can also be implemented for other kinds of basis functions. The application of MPE with basis functions from the empirical balancing approach can improve estimation on the locations which are more important than the others. This is applicable for example in the case of glass melt feeder, where the temperature at the outlet points are more important than the temperature at other locations.

8. Analysis of the spatial structures of the POD basis functions

The POD basis functions are calculated by stacking all the snapshot data of every time sample in one dimensional vector. This results in high dimensional data although the spatial domain is only divided into less than 100 parts in every direction. A possible Cartesian structure of the spatial coordinates is lost in this way. In some cases, the variations of a variable may also only vary in one direction. This may indicate that

the spatial variations may also be decomposed as functions of each coordinate direction. As a result, it is possible that the POD basis functions can also be decomposed as individual basis functions in each coordinate direction. For example in the three dimensional Cartesian coordinate, each POD basis function φ_i may be able to be described as function of POD basis function in each direction, thus $\varphi_i = f(\varphi_i(x), \varphi_i(y), \varphi_i(z))$. Since each direction is discretized into a much lower number of divisions than the overall number of grid cells, the decomposition of POD basis functions into individual basis functions in each coordinate direction may lead to a more manageable data dimension.

A

The Method of Snapshot

A POD basis is obtained by an eigenvalue decomposition of the correlation matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$ where K is the number of grid points. The correlation matrix is built from the snapshot data $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{K \times N}$ where N is the number of snapshots by setting $\mathbf{C} = \frac{1}{N} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$.

Solving an eigenvalue problem is numerically intensive. Usually, the number of snapshots is less than the number of grid points. If $N < K$, Sirovich [29] proposed *the method of snapshot* to find the POD basis based on the eigenvalue decomposition of an $N \times N$ correlation matrix instead of $K \times K$. The method is named the method of snapshot to refer to the number of snapshots (not number of grid points) solved in the eigenvalue problem.

Suppose $N < K$. The step-by-step POD basis derivation by the method of snapshot is as follows:

1. Construct the temporal correlation matrix based on the snapshot data:

$$\mathbf{C}_N = \frac{1}{N} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$$

2. Conduct an eigenvalue decomposition of \mathbf{C}_N :

$$\mathbf{C}_N \Psi_i = \lambda_i \Psi_i, \quad i = 1, \dots, n$$

where $\lambda_1, \dots, \lambda_n$ are the n largest eigenvalues of \mathbf{C}_N . Note that if \mathbf{T}_{snap} has dimension of $K \times N$ with N is the smallest dimension, then the rank of \mathbf{T}_{snap} is maximal N . The number of nonzero eigenvalues is less or equal to the maximum rank.

3. The eigenvalues of the temporal correlation matrix \mathbf{C}_N are equal to the eigenvalues of the $\mathbf{C} := \frac{1}{N} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}}$, so truncation can also be based on Λ obtained by eigenvalue decomposition of \mathbf{C}_N . Decide about the truncation level from

$$P_n = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^N \lambda_i}$$

4. Let Ψ_1, \dots, Ψ_n denote the eigenvectors of \mathbf{C}_N corresponding to the n largest eigenvalues, and compute, for $i = 1, \dots, n$:

$$\varphi_i := \frac{1}{\|\mathbf{T}_{\text{snap}}\Psi_i\|} \mathbf{T}_{\text{snap}}\Psi_i$$

Then $\{\varphi_i\}_{i=1}^n$ is a POD basis.

Proof A.0.1 Obviously $\|\varphi_i\| = 1$. Furthermore, we have

$$\frac{1}{N} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}} \Psi_i = \lambda_i \Psi_i$$

Premultiplying with \mathbf{T}_{snap} yields

$$\begin{aligned} \frac{1}{N} \mathbf{T}_{\text{snap}} \mathbf{T}_{\text{snap}}^\top \mathbf{T}_{\text{snap}} \Psi_i &= \mathbf{C} \mathbf{T}_{\text{snap}} \Psi_i \\ &= \|\mathbf{T}_{\text{snap}} \Psi_i\| \mathbf{C} \varphi_i \\ &= \lambda_i \mathbf{T}_{\text{snap}} \Psi_i = \lambda_i \varphi_i \|\mathbf{T}_{\text{snap}} \Psi_i\| \end{aligned}$$

Hence, $\mathbf{C} \varphi_i = \lambda_i \varphi_i$ so that φ_i is an eigenvector of \mathbf{C} with eigenvalue λ_i .

Remark

If the smallest dimension of \mathbf{T}_{snap} is higher than 10^3 , then computation of eigenvalue decomposition is numerically quite tedious. Software package such as MATLAB offers the possibility of finding the first n largest eigenvalues iteratively instead calculating all eigenvalues.

B

LQR design based on nonzero reference signals

The linear quadratic regulator (LQR) is an optimal control strategy for a given linear and usually time-invariant dynamical system. The control strategy is designed so as to minimize a given quadratic function in the state and the input of the system. Translated properly, this strategy achieves an optimal tracking of a given reference trajectory with minimal control (or steering) energy.

The LQR controller is designed for a system that is represented in state space form by the equations:

$$\mathbf{z}(k+1) = A_s \mathbf{z}(k) + B_s \mathbf{u}(k), \quad \mathbf{z}(0) = \mathbf{z}_0 \quad (\text{B.1})$$

$$\mathbf{y}(k+1) = C_s \mathbf{z}(k) \quad (\text{B.2})$$

where $\mathbf{z} \in \mathbb{R}^n$ is the state of the system (such as the POD coefficients), $A_s \in \mathbb{R}^{n \times n}$, $B_s \in \mathbb{R}^{n \times n_u}$, $C_s \in \mathbb{R}^{n_y \times n}$ are the state space matrices and $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of the inputs or manipulating variables. The outputs $\mathbf{y} \in \mathbb{R}^{n_y}$ can be the outputs of interest or the measurement points.

Assume that a desired reference trajectory $\mathbf{r}(k)$ is given for $k = 0, \dots, N_{\text{opt}}$, where \mathbf{r} has the same dimension as \mathbf{z} .

Consider the cost function which has to be minimized:

$$\begin{aligned} J(\mathbf{z}_0, \mathbf{u}) = & \underbrace{\sum_{k=0}^{N_{\text{opt}}-1} \left[(\mathbf{r}(k) - \mathbf{z}(k))^{\top} Q (\mathbf{r}(k) - \mathbf{z}(k)) \right]}_{\text{deviation from references}} + \underbrace{\sum_{k=0}^{N_{\text{opt}}-1} \left[\mathbf{u}^{\top}(k) R \mathbf{u}(k) \right]}_{\text{cost of input energy}} \\ & + \underbrace{\mathbf{z}^{\top}(N_{\text{opt}}) E \mathbf{z}(N_{\text{opt}})}_{\text{deviation at the last prediction step}} \quad (\text{B.3}) \end{aligned}$$

where $Q \in \mathbb{R}^{n \times n}$ is the weighting matrix for the states, $R \in \mathbb{R}^{n_u \times n_u}$ is the weighting matrix for the input signals and $E \in \mathbb{R}^{n \times n}$ is the weighting matrix at the end of the prediction horizon N_{opt} . We assume that $Q = Q^{\top} \geq 0$, $R = R^{\top} > 0$ and $E = E^{\top} \geq 0$.

Control problem:

The controls \mathbf{u} are of state feedback type. That is, a control $\mathbf{u}(k)$ with $k = 0, \dots, N_{\text{opt}}$ is *feasible* if there exist functions f_k such that

$$\mathbf{u}(k) = f_k(\mathbf{z}(k), \mathbf{r}(k), \mathbf{r}(k+1), \dots, \mathbf{r}(N_{\text{opt}}))$$

Let \mathcal{U} denote the set of all such controls.

Given the state space model as in (B.1), the cost function as in (B.3), the reference trajectory \mathbf{r} , and the class of feasible control, \mathcal{U} , find $\mathbf{u}^* \in \mathcal{U}$ such that

$$J(\mathbf{z}_0, \mathbf{u}^*) \leq J(\mathbf{z}_0, \mathbf{u}) \quad \forall \mathbf{u} \in \mathcal{U} \quad (\text{B.4})$$

The optimal control input \mathbf{u}^* will be a function of the reference \mathbf{r} and the state \mathbf{z} . To solve the minimization problem, we introduce two mappings \mathbf{P} and \mathbf{v} . \mathbf{P} is a mapping from the time set $\mathbb{T} := \{0, 1, \dots, N_{\text{opt}}\}$ to the set of real valued symmetric matrices of dimension $n \times n$. Hence $\mathbf{P} : \mathbb{T} \rightarrow \mathbb{R}^{n \times n}$. Similarly, \mathbf{v} is a mapping from the time set \mathbb{T} to \mathbb{R}^n . These mappings are defined recursively by the solutions of (B.5) and (B.6).

$$\mathbf{P}(k) = A_s^\top \mathbf{P}(k+1) A_s + Q - A_s^\top \mathbf{P}(k+1) B_s (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{P}(k+1) A_s \quad (\text{B.5})$$

To calculate $\mathbf{P}(k)$, define \mathbf{P} at the end of the optimization horizon N_{opt} as $\mathbf{P}(N_{\text{opt}}) = E$ where $E \in \mathbb{R}^{n \times n}$ and calculate (B.5) backward from N_{opt} to k .

Let also $\mathbf{v} : \mathbb{T} \rightarrow \mathbb{R}^n$ be the unique solution of

$$\mathbf{v}(k) = \left(A_s^\top - A_s^\top \mathbf{P}(k+1) B_s (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \right) \mathbf{v}(k+1) - Q \mathbf{r}(k) \quad (\text{B.6})$$

In this case, $\mathbf{v}(N_{\text{opt}}) = 0$.

To find the optimal control input \mathbf{u}^* which minimizes (B.4), define $V(k)$ as a function of $\mathbf{P}(k)$ and $\mathbf{v}(k)$:

$$V(k) := \mathbf{z}^\top(k) \mathbf{P}(k) \mathbf{z}(k) + 2\mathbf{v}^\top(k) \mathbf{z}(k)$$

Then find $V(k+1) - V(k)$:

$$\begin{aligned} V(k+1) - V(k) &= \mathbf{z}(k+1)^\top (k+1) \mathbf{P}(k+1) \mathbf{z}(k+1) + 2\mathbf{v}^\top(k+1) \mathbf{z}(k+1) \\ &\quad - \mathbf{z}^\top(k) \mathbf{P}(k) \mathbf{z}(k) - 2\mathbf{v}^\top(k) \mathbf{z}(k) \end{aligned} \quad (\text{B.7})$$

Substituting the state equation (B.1) into (B.7) results in:

$$\begin{aligned} V(k+1) - V(k) &= (A_s \mathbf{z}(k) + B_s \mathbf{u}(k))^\top \mathbf{P}(k+1) (A_s \mathbf{z}(k) + B_s \mathbf{u}(k)) \\ &+ 2\mathbf{v}^\top(k+1) (A_s \mathbf{z}(k) + B_s \mathbf{u}(k)) - \mathbf{z}^\top(k) \mathbf{P}(k) \mathbf{z}(k) \\ &- 2\mathbf{v}^\top \mathbf{z}(k) \end{aligned} \quad (\text{B.8})$$

Opening the terms in brackets in (B.8) yields:

$$\begin{aligned} V(k+1) - V(k) &= \mathbf{z}(k)^\top A_s^\top \mathbf{P}(k+1) A_s \mathbf{z}(k) + \mathbf{u}(k)^\top B_s^\top \mathbf{P}(k+1) B_s \mathbf{u}(k) \\ &+ \mathbf{u}^\top(k) B_s \mathbf{P}(k+1) A_s \mathbf{z}(k) + \mathbf{z}^\top(k) A_s^\top \mathbf{P}(k+1) B_s \mathbf{u}(k) \\ &+ 2\mathbf{v}^\top(k+1) A_s \mathbf{z}(k) + 2\mathbf{v}^\top(k+1) B_s \mathbf{u}(k) \\ &- \mathbf{z}^\top(k) \mathbf{P}(k) \mathbf{z}(k) - 2\mathbf{v}^\top(k) \mathbf{z}(k) \end{aligned} \quad (\text{B.9})$$

Note that:

$$2\mathbf{v}^\top(k+1) A_s \mathbf{z}(k) = \mathbf{v}^\top(k+1) A_s \mathbf{z}(k) + \mathbf{z}^\top(k) A_s^\top \mathbf{v}(k+1) \quad (\text{B.10})$$

$$2\mathbf{v}^\top(k+1) B_s \mathbf{u}(k) = \mathbf{v}^\top(k+1) B_s \mathbf{u}(k) + \mathbf{u}^\top(k) B_s^\top \mathbf{v}(k+1) \quad (\text{B.11})$$

Substituting (B.5) in (B.9) and considering (B.10) and (B.11) results in:

$$\begin{aligned} V(k+1) - V(k) &= -\mathbf{z}^\top(k) Q \mathbf{z}(k) \\ &+ \mathbf{z}^\top(k) A_s^\top \mathbf{P}(k+1) B_s (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} \dots \\ &\quad B_s^\top \mathbf{P}(k+1) A_s \mathbf{z}(k) + \mathbf{u}^\top(k) B_s^\top \mathbf{P}(k+1) B_s \mathbf{u}(k) \\ &+ \mathbf{u}^\top(k) B_s^\top P(k+1) A_s \mathbf{z}(k) + \mathbf{z}^\top(k) A_s^\top \mathbf{P}(k+1) B_s \mathbf{u}(k) \\ &+ \mathbf{v}^\top(k+1) A_s \mathbf{z}(k) + \mathbf{z}^\top(k) A_s^\top \mathbf{v}(k+1) \\ &+ \mathbf{v}^\top(k+1) B_s \mathbf{u}(k) + \mathbf{u}^\top(k) B_s^\top \mathbf{v}(k+1) \\ &- 2\mathbf{v}^\top(k) \mathbf{z}(k) \end{aligned} \quad (\text{B.12})$$

Introduce a notation for weighted quadratic terms:

$$\| \mathbf{d} \|_M^2 = \mathbf{d}^\top M \mathbf{d}$$

The expression in (B.12) can be rewritten using the notation of quadratic terms as:

$$\begin{aligned} V(k+1) - V(k) &= \| \mathbf{u}(k) + (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{P}(k+1) A_s \mathbf{z}(k) \\ &+ (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{v}(k+1) \|_{R+B_s^\top \mathbf{P}(k+1) B_s}^2 \\ &- (\mathbf{r}(k) - \mathbf{z}(k))^\top Q (\mathbf{r}(k) - \mathbf{z}(k)) - \mathbf{u}^\top(k) R \mathbf{u}(k) \\ &- \mathbf{v}^\top(k+1) B_s (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{v}(k+1) \\ &+ X(k) \mathbf{z}(k) + \mathbf{z}^\top(k) Y(k) + \mathbf{r}^\top(k) Q \mathbf{r}(k) \end{aligned}$$

where

$$\begin{aligned}
X(k) &= \mathbf{v}^\top(k+1) \left(A_s - B_s(R + B_s^\top P(k+1)B_s)^{-1} B_s^\top P(k+1)A_s \right) \\
&\quad - \mathbf{v}^\top(k) - \mathbf{r}^\top(k)Q = 0 \\
Y(k) &= \left(A_s^\top - A_s^\top \mathbf{P}(k+1)B_s(R + B_s \mathbf{P}(k+1)B_s^\top)^{-1} B_s^\top \right) \mathbf{v}(k+1) \\
&\quad - \mathbf{v}(k) - Q\mathbf{r}(k) = 0
\end{aligned} \tag{B.13}$$

By application of (B.6), it is obvious that both $X(k)$ and $Y(k)$ in (B.13) are zero.

Introduce

$$\begin{aligned}
F_z(k) &:= (R + B_s^\top \mathbf{P}B_s)^{-1} B_s^\top \mathbf{P}(k+1)A_s \\
F_v(k) &:= (R + B_s^\top \mathbf{P}B_s)^{-1} B_s^\top
\end{aligned}$$

The term $V(k+1) - V(k)$ can then be written as:

$$\begin{aligned}
V(k+1) - V(k) &= \left\| \mathbf{u}(k) + F_z(k)\mathbf{z}(k) + F_v(k)\mathbf{v}(k+1) \right\|_{R+B_s^\top \mathbf{P}(k+1)B_s}^2 \\
&\quad - (\mathbf{r}(k) - \mathbf{z}(k))^\top Q(\mathbf{r}(k) - \mathbf{z}(k)) - \mathbf{u}^\top(k)R\mathbf{u}(k) \\
&\quad - \mathbf{v}^\top(k+1)B_s(R + B_s \mathbf{P}(k+1)B_s^\top)^{-1} B_s^\top \mathbf{v}(k+1) \\
&\quad + \mathbf{r}^\top(k)Q\mathbf{r}(k)
\end{aligned} \tag{B.14}$$

Using (B.14), the cost function can now be rewritten as

$$\begin{aligned}
J(\mathbf{z}_0, \mathbf{u}) &= \sum_{k=0}^{N_{\text{opt}}-1} [V(k) - V(k+1)] \\
&\quad + \sum_{k=0}^{N_{\text{opt}}-1} \left\| \mathbf{u}(k) + F_z(k)\mathbf{z}(k) + F_v(k)\mathbf{v}(k+1) \right\|_{R+B_s^\top \mathbf{P}(k+1)B_s}^2 \\
&\quad - \sum_{k=0}^{N_{\text{opt}}-1} \left\| B_s \mathbf{v}(k+1) \right\|_{(R+B_s \mathbf{P}(k+1)B_s^\top)^{-1}}^2 \\
&\quad + \sum_{k=0}^{N_{\text{opt}}-1} \left\| \mathbf{r}(k) \right\|_Q^2 + \mathbf{z}^\top(N_{\text{opt}})E\mathbf{z}(N_{\text{opt}})
\end{aligned}$$

Use the initial conditions of the recursions of \mathbf{P} and \mathbf{v} in (B.5) and (B.6), together with the definition of $V(k)$ to infer that $\sum_{i=0}^{N_{\text{opt}}-1} [V(k) - V(k+1)] = V(0) - V(N_{\text{opt}}) = V(0) - \mathbf{z}^\top(N_{\text{opt}})E\mathbf{z}(N_{\text{opt}})$ where we used that $\mathbf{v}(N_{\text{opt}}) = 0$.

Hence, we obtain the cost for any control \mathbf{u} :

$$\begin{aligned} J(\mathbf{z}_0, \mathbf{u}) &= \mathbf{z}_0^\top \mathbf{P}(0) \mathbf{z}_0 + \mathbf{v}^\top(0) \mathbf{z}_0 \\ &+ \sum_{k=0}^{N_{\text{opt}}-1} \|\mathbf{u}(k) + F_z(k) \mathbf{z}(k) + F_v(k) \mathbf{v}(k+1)\|_{R+B_s^\top \mathbf{P}(k+1) B_s}^2 \\ &- \sum_{k=0}^{N_{\text{opt}}-1} \|B_s \mathbf{v}(k+1)\|_{(R+B_s^\top \mathbf{P}(k+1) B_s)^{-1}}^2 + \sum_{k=0}^{N_{\text{opt}}-1} \|\mathbf{r}(k)\|_Q^2 \end{aligned}$$

where the first two right hand terms and the last right hand side term are independent from \mathbf{u} .

The fact that the first two right hand side terms are independent from \mathbf{u} implies that the minimum of $J(\mathbf{z}_0, u)$ is obtained when

$$\sum_{k=0}^{N_{\text{opt}}-1} \|\mathbf{u}(k) + F_z(k) \mathbf{z}(k) + F_v(k) \mathbf{v}(k+1)\|_{R+B_s^\top \mathbf{P}(k+1) B_s}^2 = 0$$

is minimal.

So the control input $\mathbf{u}(k)$ which minimizes (B.4) is $\mathbf{u}^*(k)$ defined as:

$$\mathbf{u}^*(k) = - \underbrace{F_z(k) \mathbf{z}(k)}_{\text{state feedback}} - \underbrace{F_v(k) \mathbf{v}(k+1)}_{\text{function of reference}} \quad (\text{B.15})$$

The design of LQR control design can be summarized in the following theorem:

Theorem B.0.2 *Given the system (B.1), the control objective (B.4), (B.5) and (B.6) together with the matrix sequences*

$$\begin{aligned} F_z(k) &:= (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \mathbf{P}(k+1) A_s \\ F_v(k) &:= (R + B_s^\top \mathbf{P}(k+1) B_s)^{-1} B_s^\top \end{aligned}$$

Then the optimal control $\mathbf{u} \in \mathcal{U}$ that minimizes (B.4) is given by

$$\mathbf{u}(k)^* := -F_z(k) \mathbf{z}(k) - F_v(k) \mathbf{v}(k+1)$$

Remark

The LQR controller designed here is originally defined for Linear Time Invariant systems. The discretization of nonlinear PDE's usually leads to Linear Time Varying systems. In that case, the state space matrices are time varying with varying inputs \mathbf{u} . For such systems, a *suboptimal* LQR control can be

designed. The suboptimal control is designed for LTV systems based on the assumption that at every time step, an LTI system is given. Within one time step, the recursive equation (B.5) is calculated by calculating E with the state space matrix $\mathbf{A}_s(k)$ and $\mathbf{B}_s(k)$. Hence, the initial value for solving (B.5) backwards will be different for each time step. Similarly, the recursive equation (B.6) is also calculated for every time step using the actual state space matrix $\mathbf{A}_s(k)$ and $\mathbf{B}_s(k)$. The control input $\mathbf{u}(k)$, is defined exactly as in Theorem B.0.2, but $\mathbf{P}(k+1)$ and $\mathbf{v}(k+1)$ are recalculated at every time step.

Nomenclature

Acronyms

| | |
|-----|---------------------------------|
| ARE | Algebraic Riccati Equation |
| LQR | Linear Quadratic Regulator |
| LTi | Linear Time Invariant |
| LTV | Linear Time Varying |
| MPC | Model Predictive Control |
| MPE | Missing Point Estimation |
| PDE | Partial Differential Equation |
| POD | Proper Orthogonal Decomposition |

Greek symbols

| | |
|-------------------|---|
| Λ | a diagonal matrix with eigenvalues in its diagonal part |
| λ | eigenvalue |
| Φ | a matrix with POD basis elements $\{\varphi\}_i$ in its columns |
| $\tilde{\Phi}$ | The POD basis comprises POD basis functions correspond to the selected points |
| $\tilde{\varphi}$ | The POD basis functions whose elements correspond to the selected points only |
| φ | an orthonormal basis function, e.g POD basis function or Fourier basis function |

Latin Symbols

| | |
|----------------------|--|
| $\tilde{\mathbf{a}}$ | the POD coefficients obtained from MPE |
| \mathbf{a} | the vector of POD coefficients |
| $\tilde{\mathbf{T}}$ | the collection of the selected data in POD-MPE |
| \mathbf{C} | the correlation matrix |

| | |
|----------------------------|--|
| \mathbf{P} | the solution of the Riccati Equation |
| \mathbf{T} | vector of states/variables in the computation of original model |
| \mathbf{T}_n | the variables of the original models approximated by the POD expansion |
| \mathbf{T}_{ref} | the desired distribution of a process variable, e.g temperature |
| \mathbf{T}_{snap} | snapshot collection, in finite dimensional case snapshot matrix |
| \mathbf{u} | vector of inputs /manipulating variables |
| \mathbf{y} | the vector of the outputs of interest, e.g measurement points |
| \mathbf{z} | the vector of the states |
| A_s | square state space matrix for the states, dimension is the same as the number of states |
| B_s | state space matrix for input signals, dimension is the same as the number of states times the number of inputs |
| C_s | state space matrix for the outputs, dimension is the same as the number of outputs times the number of states |
| k | the k -th time step |
| n | the number of POD basis elements in the reduced order model |
| x | the spatial location |

Sets and set operations

| | |
|----------------|--------------------------------------|
| \forall | for all |
| \in | an element of |
| \notin | not an element of |
| \mathbb{I} | index set, e.g the positive integers |
| \mathbb{R} | the set (collection) of real number |
| \mathbb{R}^+ | set of positive real number |
| \mathbb{T} | time domain |
| \mathbb{X} | the spatial domain |

| | |
|----------------|---|
| \mathbb{X}_0 | the subset of \mathbb{X} which comprises the locations of the selected points for MPE approach |
| \mathbb{Z} | the set (collection) of integers |
| \mathcal{M} | Set of model class |
| \mathcal{U} | the set of all admissible control signals, e.g control signals that does not violate the physical constraints defined |
| \subset | subset of |
| \subseteq | subset to equal |
| \supset | supset of |
| \supseteq | supset to equal |

Functions and operators

| | |
|-------------------------|---|
| (\cdot, \cdot) | inner product operation |
| arg | the argument of a function, as x is the argument of $f(x)$ |
| card | the maximum number of nonzero Fourier coefficients or the number of elements in a set |
| col | stacking a set of elements into a column vector |
| dim | dimension, e.g length of a vector |
| $\langle \cdot \rangle$ | averaging operation |
| max | the maximum of |
| min | the minimum of |
| ∇ | the divergence operator |
| span | the set of all linear combinations |

Physical parameters

| | |
|----------|--------------------------------|
| κ | heat conductivity [W/m.K] |
| μ | viscosity [Ns/m ²] |
| ρ | density [kg/m ³] |

| | |
|----------|---|
| σ | the Stefan Boltzmann constant [$\text{W}/\text{m}^2\text{K}^4$] |
| c_p | specific heat capacity [$\text{J}/\text{kg}\cdot\text{K}$] |

Subscripts

| | |
|--------|-------------------------------------|
| B | the bottom neighboring grid point |
| E | the eastern neighboring grid point |
| i, j | integer valued indeces |
| N | the northern neighboring grid point |
| P | the central grid point |
| r | reduced |
| S | the southern neighboring grid point |
| T | the top neighboring grid point |
| W | the western neighboring grid point |
| opt | optimal |
| ref | reference |
| snap | snapshot |

Bibliography

- [1] V.M. Adamjan, D.Z. Arov, and M.G. Krein. Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur–Takagi problem. *Math USSR Sbornik*, 15:31–73, 1971.
- [2] J. GeBlein A.M. Lankhorst, B.D. Paarhuis. Experimental validation of the TNO forehearth model. In *7th International Seminar on Mathematical Modeling and Advanced Numerical Methods in Furnace Design and Operation*, Velke Karlovice, June 5-6 2003.
- [3] P. Astrid. Fast large scale model reduction technique for large scale LTV systems. In *Proceedings of the American Control Conference*, Boston, June 2004.
- [4] P. Astrid, L. Huisman, S. Weiland, and A.C.P.M. Backx. Reduction and predictive control design for a computational fluid dynamics model. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pages 3378 – 3383, Las Vegas, December 2002.
- [5] P. Astrid, S. Weiland, K.E. Willcox, and A.C.P.M. Backx. Missing point estimation in models described by proper orthogonal decomposition. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahama, December 2004.
- [6] P. Astrid, S. Weiland, and A. Twerda. Reduced order modeling of an industrial glass feeder model. In *Proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, August 2003.
- [7] P. Astrid, S. Weiland, and K.E. Willcox. On the acceleration of the POD-based model reduction technique. In *Proceedings of the 16th International Symposium on Mathematical Theory of Network and Systems*, Leuven, July 2004.
- [8] A.C.P.M. Backx. *Model-Based Glass Melter Control*, chapter 2, pages 137–155. In *Mathematical Simulation in Glass Technology*, eds: H.Loch and D.Krause. Springer, Berlin, 2002.
- [9] R.G.C. Beerkens. *Modeling of the Melting Process in Industrial Glass Furnaces*, chapter 2, pages 17–73. In *Mathematical Simulation in Glass Technology*, H.Loch and D.Krause (eds). Springer, Berlin, 2002.

-
- [10] R.G.C. Beerkens, T. van der Heijden, and H.P.H. Muijsenberg. Possibilities of glass tank modeling for the prediction of the quality of melting processes. *Ceram.Eng.Sci.Proc.*, 14(3-4):139–160, 1993.
- [11] R.G.C Beerkens, H.de Waal, and F.Simonis. *Handbook for Glass Technologist*. TNO-TPD Glass Technology, Eindhoven, 1997.
- [12] D.J. Bergman. Nonlinear behavior and $1/f$ noise near a conductivity threshold: Effects of local microgeometry. *Phys. Rev. B*, 39(7):4598, March 1989.
- [13] B.B. Bird, W.E. Stewart, and E.N. Lightfoot. *Transport Phenomena*. John Wiley and Sons, New York, 1960.
- [14] P.N Bloassey and J.L Lumley. Reduced order modeling and control of near wall turbulent flow. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 2851–2856, 1999.
- [15] T. Bui-Thanh, M. Damodaran, and K.E. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.
- [16] G.P. Boerstoel C. Vuik, A. Saghir. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. *International Journal for Numerical Methods in Fluids*, 33.
- [17] M.G. Carvalho, J.Wang, and M.Nogueira. Investigation of glass melting and fining processes by means of comprehensive mathematical model. *Ceramic Transactions*, 82:143–152, 1997.
- [18] M.G. Carvalho, N. Speranskaia, J. Wang, and M. Nogueira. Modeling of glass melting furnaces: applications to control, design and operation optimization. *Ceramic Transactions*, 82:109–135, 1997.
- [19] Y. Chahlaoui and P. van Dooren. Estimating grammians of large scale time varying system. In *Proceedings of the 15th Triennial IFAC World Congress*, Barcelona, July 2002.
- [20] B. Cockburn, G.E Karniadakis, and C.Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, 2000.
- [21] R. Curtain and H. Zwart. *An Introduction to Linear Infinite-Dimensional Systems Theory*. Springer Verlag, New York, 1995.
- [22] R. Everson and L. Sirovich. The Karhunen-Loève procedure for gappy data. *Journal Opt. Soc. Am.*, 12:1657–1664, 1995.

-
- [23] S. Glavaski, J.E. Marsden, and R.M. Murray. Model reduction, centering and the Karhunen-Loève expansion. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 2, pages 2071–2076, Tampa, Florida, December 1998.
- [24] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their l^∞ -error bounds. *International Journal of Control*, 39:1115–1193, 1984.
- [25] W.R. Graham, J. Peraire, and K.Y. Tang. Optimal control of vortex shedding using low order models. Part II - model based control. *Journal Opt. Soc. Am.*, 44:973–990, 1999.
- [26] R. Günther and J. Currie. *Glass-melting tank furnaces*. Sheffield: Society of glass technology, reprint edition, 1980.
- [27] M. Hazenberg. On low order modeling using proper orthogonal decomposition. Master's thesis, Eindhoven University of Technology, Electrical Engineering Department, July 2002.
- [28] M. Hazenberg, P. Astrid, and S. Weiland. Low order modeling and optimal control design of a heated plate. In *Proceedings of the 5th European Control Conference*, Cambridge, September 2003.
- [29] P. Holmes, Lumley, and G.Berkooz. *Turbulence, Coherence Structure, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 1996.
- [30] L. Huisman. Estimation of process variables in a glass melting furnace. In *Proceedings of the 15th Triennial IFAC World Congress*, Barcelona, July 2002.
- [31] L. Huisman and S. Weiland. Identification and model predictive control of an industrial glass feeder. In *Proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, August 2003.
- [32] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae, Ser.A1*, 34, 1946.
- [33] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, Oxford, 1999.
- [34] M. Kirby. *Geometric Data Analysis, An Emprical Approach to Dimensionality Reduction and the Study of Patterns*. John Wiley and Sons.Inc, New York, 2001.

-
- [35] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, Inc, New York, 1993.
- [36] P. Krysl, S. Lall, and J. E. Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for Numerical Methods in Engineering*, 51(4):479–504.
- [37] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40:492–515.
- [38] K. Kunisch, S. Volkwein, and L. Xie. HJB-POD based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems*, to appear.
- [39] S. Lall, J.E. Marsden, and S. Glavaski. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal on Robust and Nonlinear Control*, 12(5):519–535.
- [40] M. Loève. Fonctions aléatoire de second ordre. *Comptes Rendus Acad. Sci. Paris*, 220, 1945.
- [41] E.N. Lorenz. *Empirical orthogonal functions and statistical weather prediction*. In Statistical Forecasting Project. MIT Press, Cambridge, MA, 1956.
- [42] L.Sirovich. *Empirical eigenfunctions and low dimensional systems*, pages 139–163. In *New Perspectives in Turbulence*, L.Sirovic (eds). Springer-Verlag, 1991.
- [43] J.L. Lumley. *The structure of inhomogeneous turbulence*, pages 166–178. In *Atmospheric Turbulence and Wave Propagation*, Yaglom and V.I. Tatarski (eds). Nauka, Moscow, 1967.
- [44] H.V Ly and H.T Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33.
- [45] B. Moore. Principle component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.
- [46] G.E. Moore. Cramping more components onto integrated circuits. *Electronics*, 38(8), 1965.

-
- [47] H.P.H. Muysenberg, F. Simonis, R. van Roosmalen, and M. van Doom. Verification of 3d mathematical simulation with measured temperature profiles during furnace operation. In *4th International Conference on Advances in fusion and processing of Glass*, 1995.
- [48] D.J. Newman and G.E. Karniadakis. Low-dimensional modeling of flow-induced vibrations via proper orthogonal decomposition. In *21st Symposium on Naval Hydrodynamics*, Norway, 1996.
- [49] A.M. Obukhov. Statistical description of continuous fields. *Trudy Geophys. Int. Aked. Nauk. SSSR*, 24:3–42, 1954.
- [50] K. Ogata. *Modern Control Engineering*. Prentice Hall, India, second edition, 1995.
- [51] O.M.G.C. op den Camp and V.O. Aume. Advanced forehearth design by means of mathematical modeling. In *The 18th International Congress on Glass*, San Fransisco, July 5-10 1999.
- [52] O.M.G.C op den Camp and O.S Verheijen. Application of glass process simulation (GPS) in furnace operation. In *7th International Seminar on Mathematical Modeling and Advanced Numerical Methods in Furnace Design and Operation*, Velke Karlovice, June 5-6 2003.
- [53] H.M Park and M.W Lee. Control of Navier Stokes equations by means of mode reduction. *International Journal for Numerical Methods in Fluids*, 33.
- [54] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, London, 1980.
- [55] C.L. Philips and J.M. Parr. *Signals, systems, and transforms*. Prentice Hall, New Jersey, 1995.
- [56] V.S. Pougachev. General theory of the correlations of random functions. *Izv. Akad. Nauk. SSSR. Math. Ser.*, 17:401–2, 1953.
- [57] S. Prajna. POD model reduction with stability guarantee. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 5, pages 5254 – 5258, Maui, December 2003.
- [58] TNO Institute of Applied Physics. *User's Guide: The Glass Tank Model version s12-00*. TNO Institute of Applied Physics, Delft, 2000.

-
- [59] TNO Institute of Applied Physics. *X-stream : User Manual, Programmer Manual, Installation Manual, version 2.2*. TNO Institute of Applied Physics, Delft, 2003.
- [60] S.S Ravindran. Reduced-order adaptive controllers for fluid flows using POD. *Journal of Scientific Computing*, 15(4):457–478, 2000.
- [61] S.S Ravindran and K. Ito. A reduced order method for simulating and control of fluid flows. *Journal of Computational Physics*, 143(2):403–425, 1998.
- [62] J.M.A. Scherpen. *Balancing for Nonlinear Systems*. PhD dissertation, University of Twente, Department of Applied Mathematics, March 1994.
- [63] W.E. Schiesser and C.A. Silebi. *Computational Transport Phenomena, numerical methods for the solution of transport problems*. Cambridge University Press, Cambridge, 1997.
- [64] P. Simons, A. Twerda, R. Verweij, J. Wang, F. Simonis, A.M. Lankhorst, and B.D. Paarhuis. X-stream: the most comprehensive glass furnace simulation tool. In *Glass Odyssey, 6th ESG Conference*, Montpellier, July 2-6 2002.
- [65] L. Sirovich, B.W. Knight, and J.D Rodriguez. Optimal low-dimensional dynamical approximations. *Q.Appl.Math.*, 48:535–548, 1990.
- [66] J. Stanek. *Electric Melting of Glass*. Elsevier, Amsterdam, 1977.
- [67] S. Strogatz. *Nonlinear Dynamics and Chaos*. Westview Press, 2000.
- [68] O.S Verheijen. *Thermal and Chemical Behaviour of Glass Forming Batches*. PhD dissertation, Eindhoven University of Technology, Department of Chemical Engineering, 2003.
- [69] O.S. Verheijen and O.M.G.C. op den Camp. Advanced operation support system for redox control. In *Advances in fusion and processing of Glass conference*, Rochester, July 27-30 2003.
- [70] H.K. Versteeg and W.K. Malalasekera. *An Introduction to Computational Fluid Dynamics, The Finite Volume Method*. Pearson Prentice Hall, Essex, 1995.
- [71] K.E. Willcox. *Reduced Order Aerodynamics Models for Aeroelastic Control of Turbomachines*. PhD dissertation, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, February 2000.

-
- [72] K.E. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. In *34th AIAA Fluid Dynamics Conference and Exhibit*, Portland, 2004.
- [73] K.E. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.
- [74] Yeung and Ruzzo. Principle component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–74, 2001.

Acknowledgement

As a four-year-old, I proudly carried my schoolbag, which my father brought from Holland, to start my very first school day in life at the Yahya Kindergartenschool in Bandung. Nothing passed my mind that day that this school journey would end up 24 years later in the Netherlands.

I hereby acknowledge those who have helped me all the way through this last step. I thank my first promoter Prof. Ton Backx for all his guidance, support and most of all his constant optimism during these four years. My second promoter, Prof. Ruud Beerkens, has given me more than academic supervision with his invaluable expertise in glass process technology. I thank him for his total commitment in supervising me and for being a very efficient and reliable supervisor whenever I need anything. I'm indebted to my daily supervisor Dr. Siep Weiland who is always available for discussions and who minds taking care of my thesis formalities such as the ISBN number, committed all the way to correct my scribbled notes and from whom I learn a great deal of mathematical perspectives.

The visit I spent at MIT during the fall semester 2003 has been a great experience, thanks to Dr. Karen Willcox. She guided me just like one of her regular graduate students and had been personally a very attentive person. I would like to thank her as well for proofreadings the conference papers and the last version of the thesis draft.

Special thanks to Prof. Okko Bosgra, apart from being my committee member, he was the one who initiatively sent out my resumes to all control groups in Holland more than 4 years ago. This PhD work would not be possible without his initiative. I am indebted to Prof. Philip de Goey who has agreed at the last minute to be in my core committee and who has proofread my thesis draft in a very short time. I would like to convey my gratitude to Dr. Jacquelin Scherpen and Prof. Stefan Volkwein who are willing to spare their time to be in my PhD committee.

I really appreciate the help from my 'computing'-supervisor Dr. Aris Twerda at TNO Delft. It was amazing to see how things turned out with his assistance considering my illiteracy of Fortran programming when I started. I would like to thank Philip Simons, who had spent regular time with me to discuss the spaghetti code. Dr. Olaf op den Camp has certainly been a good project

leader and his pragmatism helped me a lot. Andries Habraken helped me out by providing a marvellous test case and his cheerfulness made things just go easier. Even though we met only twice a year, Sven Kahl certainly has been a great support for this project.

From inner-REGLA-circle, I thank Leo Huisman for the discussions we had in work and in faith. I'm grateful to Dr. John van der Schaaf who willingly supervised me informally for the first 2.5 years. I would like to acknowledge Martijn Hazenberg, who has built the nicest test case with his MSc thesis and his "hoe gaat ie" e-mails after his graduation.

Many thanks to Prof. Paul van den Bosch as the CS group leader for giving me so much support such as the laptop, the reference for NWO grant, and the compliments after presentations. Udo Bartzke has been a cheerful computer administrator and always went easy on everything I asked, vielen Dank. I'm grateful to Barbara Cornelissen for bearing with my abrupt administrative presence and the last minute rush with my Form 2. Thanks to Dr. Maurice Heemels and Dr. Ad van den Boom as well who gave me the chance to explore my-in-Dutch teaching capabilities.

I will always remember the nice time I had with my fellow PhD students. To Aleksandar, thanks for sharing with me the crazy days of being PhD students especially in bridging the gap between the 'east' and 'west' wing of our group. To Andrei thanks for the funny stories, thanks to John Kessels for doing a superb job when he took over my task as the lunch seminar organizer. I also thank Mircea for some funny conversations we had at "Bos-Bisch" and other occasions. I wish good luck for Bart, John, Mircea, Mark, Marteen, Andrej, Nielis, Heico, Michiel, in their own PhD endeavors. Of course I won't forget my promo-overleg team: Ronald Schimmel, Arjen Steiner, Oscar Verheijen, Jan Hermans who patiently listened to my math stuff presentation though I know it was the least they would like to listen to on Friday afternoons.

My thanks also go to Devi Putra and his wife Dina with their beautiful children who give me a piece of home back in Helmond. Ich möchte mich bedanken für Ihre Unterstützung bei: Uta Riedel, Marion und Joachim Reinhold in Hamburg, Renate und Manfred Ackerman in Gernsheim. Ich werde niemals vergessen dass Ihr mir sehr geholfen habt während meinem Aufenthalt in Deutschland.

I have tremendous support from my family to finish this degree although when being told about my PhD topic, my parents commented, "Are you sure you have to reduce thousands of equations to about 10 ? That's impossible !" I

am grateful to my father, who is always proud of me and to my sister and my brother with whom I spent great 20 years together in our home in Bandung.

I especially thank my mother for her dedication to me. She was my first teacher and had brought me up in a 'genderless' manner that I never feel limited by my womanhood. I hope this PhD thesis shows her that everything she has gone through is not in vain.

Last but not least, many thanks to someone special, Saoer for the company, love and encouragement. I really appreciate his progressive mind even though at some points it meant me being in another continent. Those long hours on the telephone, care and visits keep me grounded in the midst of difficult times I often had.

In the end, thank you Lord for the goodness I experience every single day.

Eindhoven, October 2004

Curriculum vitae



Patricia Astrid was born on May 1st, 1976 in Bandung, West Java, Indonesia. Physics was her favorite subject at school and after completing high school at St. Aloysius in Bandung, she went to study Applied Physics at Bandung Institute of Technology (ITB), Indonesia in August 1994. During her third year at college, she became interested in control engineering and elaborated this topic further in her BSc thesis. She obtained her BSc degree (*cum-laude*) in Applied Physics in April 1998.

She was working part time during her final year and full time after she obtained her degree at the Center for Research on Energy, ITB. She then obtained a scholarship in September 1998 from the University of Groningen in the Netherlands to pursue an MSc study in Chemical Engineering. She spent one year in Germany during her second year MSc study for internship at Merck KGaA in Gernsheim and MSc.thesis at the Institute of Control Engineering, Technical University Hamburg-Harburg. She obtained the MSc degree from the Department of Chemical Engineering, University of Groningen, the Netherlands in June 2000.

She returned to the Netherlands in September 2000 to commence her PhD study at the Department of Electrical Engineering, Eindhoven University of Technology. During the fall semester 2003/2004, she was a visiting PhD student at the Aerospace Computational Design Laboratory, Massachusetts Institute of Technology, Cambridge, USA.