REDUCED ORDER MODELING FOR TRANSPORT PHENOMENA BASED ON PROPER ORTHOGONAL DECOMPOSITION

A Thesis

by

TAO YUAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2003

Major Subject: Aerospace Engineering

REDUCED ORDER MODELING FOR TRANSPORT PHENOMENA BASED ON PROPER ORTHOGONAL DECOMPOSITION

A Thesis

by

TAO YUAN

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Paul G. A. Cizmas
(Chair of Committee)

Theofanis Strouboulis
(Member)

Ali Beskok
(Member)

Walter E. Haisler
(Head of Department)

December 2003

Major Subject: Aerospace Engineering

ABSTRACT

Reduced Order Modeling for Transport Phenomena
Based on Proper Orthogonal Decomposition. (December 2003)
Tao Yuan, B.E., Tsinghua University
Chair of Advisory Committee: Dr. Paul G. A. Cizmas

In this thesis, a reduced order model (ROM) based on the proper orthogonal decomposition (POD) for the transport phenomena in fluidized beds has been developed. The reduced order model is tested first on a gas-only flow. Two different strategies and implementations are described for this case. Next, a ROM for a two-dimensional gas-solids fluidized bed is presented. A ROM is developed for a range of diameters of the solids particles. The reconstructed solution is calculated and compared against the full order solution. The differences between the ROM and the full order solution are smaller than 3.2% if the diameters of the solids particles are in the range of diameters used for POD database generation. Otherwise, the errors increase up to 10% for the cases presented herein. The computational time of the ROM varied between 25% and 33% of the computational time of the full order solution. The computational speed-up depended on the complexity of the transport phenomena, ROM methodology and reconstruction error. In this thesis, we also investigated the accuracy of the reduced order model based on the POD. When analyzing the accuracy, we used two simple sets of governing partial differential equations: a non-homogeneous Burgers' equation and a system of two coupled Burgers' equations.

To Xiaoyan Zhu

ACKNOWLEDGMENTS

First and foremost I would like to thank my advisor, Dr. Paul Cizmas, not only for his keen insight to keep me on course, but also for his patience in dealing with my trial and errors. Next, I would like to thank my thesis committee, Dr. Theofanis Strouboulis and Dr. Ali Beskok. Also, I would like to thank my peers: Celerino Resendiz, Kyu-Sup Kim and Steve Chambers. Most importantly, I would like to thank my wife Xiaoyan Zhu for her backing and encouraging.

NOMENCLATURE

d_{ps} – Solid particle diameter

F_{gs} – Coefficient for the interphase force between gas and solid phases

g – Gravity acceleration

M – Number of snapshots

M_w – Average molecular weight of gas

N – Number of discrete spatial grid points

p – Pressure

R – Universal gas constant

Re – Reynolds number

T – Temperature

(u, v) – Components of velocity vector

$\vec{v}$ – Velocity vector

(x, y) – Cartesian coordinates

α – Time coefficients

ϵ – Volume fraction, error measurement

μ – Viscosity

ρ – Density

$\bar{\bar{\tau}}$ – Viscous stress tensor

ξ – Convection factor

Subscripts

g – Gas phase

s – Solid phase

Superscripts

$*$ – Tentative values

o – Old values

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

- A. Statement of the Problem 1
- B. Background 1
  - 1. Reduced Order Modeling 1
  - 2. Proper Orthogonal Decomposition 4
  - 3. Transport Phenomena in Fluidized Beds 6
- C. Outline of this Thesis 7

II PHYSICAL MODEL 9

- A. Governing Equations 9
- B. Boundary Conditions 10

III FULL NUMERICAL MODEL 11

- A. Discretization 11
- B. Discretized Governing Equations 12

IV METHODOLOGY FOR REDUCED ORDER MODELING BASED ON PROPER ORTHOGONAL DECOMPOSITION . . 19

- A. Database Generation 19
- B. Modal Decomposition 20
- C. Galerkin Projection 21
- D. Summary 23

V REDUCED ORDER MODELS BASED ON PROPER ORTHOGONAL DECOMPOSITION FOR TRANSPORT PHENOMENA 24

- A. Reduced order models based on proper orthogonal decomposition for gas-only flow phenomena 24
  - 1. ODExMFIX 24
  - 2. ODEx3 28
- B. Reduced order model based on proper orthogonal decomposition for gas-solid transport phenomena 34
- C. Summary 36

CHAPTER Page

VI ACCURACY OF REDUCED ORDER MODELS BASED ON PROPER ORTHOGONAL DECOMPOSITION 37

A. Non-homogeneous Burgers' Equation 37
  1. Database Generation 38
  2. Modal Decomposition 40
  3. Galerkin Projection 41
  4. Accuracy Analysis 42
    a. Accuracy of basis functions 43
    b. Accuracy of time coefficients 48
    c. Accuracy of reconstructed solution u 55
B. "Double" Burgers' Equations 57
  1. Database Generation 59
  2. Model Decomposition 61
  3. Galerkin Projection 61
  4. Accuracy Analysis 62
    a. Accuracy of basis functions 63
    b. Accuracy of time coefficients 64
    c. Accuracy of reconstructed dependent variables . . 67
C. Summary 69

VII RESULTS 70

A. Case I: compressible gas-only flow 70
B. Case II: gas-solid transport phenomena in a fluidized bed . 85
C. Summary 102

VIII CONCLUSIONS AND FUTURE WORK 103

A. Conclusions 103
B. Future Work 104

REFERENCES 105

APPENDIX A 108

APPENDIX B 111

APPENDIX C 112

APPENDIX D 113

Page

APPENDIX E 114

VITA 115

LIST OF TABLES

TABLE Page

I Reduced order models for gas-only flows 25

II Six sets of basis functions and corresponding databases 43

III Errors of ϕ_{i_A}, $i \in [0,3]$ 44

IV Errors of $\phi_{i_N}^{(\bar{\epsilon})}$ with respect to $\phi_{i_{analytical}}$, $i \in [0,3]$ 46

V Errors of $\phi_{i_N}^{(\bar{\epsilon})}$ with respect to ϕ_{i_A}, $i \in [0,3]$ 47

VI Cumulative energy for the Burgers' equation 47

VII Errors of $\alpha_{i_{analytical,ROM(3)}}$ 49

VIII Errors of $\alpha_{i_{analytical,ROM(2)}}$ 49

IX Errors of $\alpha_{i_{A,ROM(3)}}$ with respect to $\alpha_{i_{analytical}}$ 50

X Errors of $\alpha_{i_{A,ROM(2)}}$ with respect to $\alpha_{i_{analytical}}$ 50

XI Errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ with respect to $\alpha_{i_{analytical}}$, $i \in [0,3]$ 52

XII Errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ with respect to $\alpha_{i_{A,ROM}}$, $i \in [0,3]$ 55

XIII Errors of $u_{analytical,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$ 56

XIV Errors of $u_{A,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$ 56

XV Errors $\bar{\epsilon}_u$ of $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$ 57

XVI $\bar{\epsilon}_u$ and $\bar{\epsilon}_v$ 60

XVII Errors of $\phi^u_{i_A}$ and $\phi^v_{i_A}$ 63

XVIII Errors of $\phi^{u,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_A}$ and $\phi^{v,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_A}$ 63

TABLE Page

XIX Errors of time coefficients obtained from the POD-based ROM for "double" Burgers' equations 67

XX Errors of reconstructed $u_{N,ROM}^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}(0.5,t)$ and $v_{N,ROM}^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}(0.5,t)$ 67

XXI Parameters of case I 71

XXII Case I: errors of ODExMFIX at different values of μ_0 83

XXIII Parameters of case II 88

XXIV Case II: POD energy vs number of modes for p_g, u_g, v_g, u_s, and v_s . 90

XXV Case II: errors of the results of ODExS at reference condition 91

XXVI Case II: errors of the results of ODExS at off-reference conditions . . 100

LIST OF FIGURES

FIGURE Page

1 Typical behavior of a fluidized bed 7

2 Geometry and boundary conditions of a fluidized bed 10

3 Grid arrangement in MFIX 11

4 Control volume for mass balance 12

5 Control volume for momentum balance 14

6 $\phi_{i_{analytical}}$ and $\phi_{i_N}^{(\bar{\epsilon})}$ for the Burgers' equation 45

7 $\alpha_{i_{N,ROM}}^{(0.78)}(t)$ compared against $\alpha_{i_{analytical}}$, $i \in [1,3]$ for Burgers' equation 51

8 $\alpha_{i_{N,ROM}}^{(17.3)}(t)$ compared against $\alpha_{i_{analytical}}$, $i \in [1,3]$ for Burgers' equation 53

9 $\alpha_{i_{N,ROM}}^{(30.5)}(t)$ compared against $\alpha_{i_{analytical}}$, $i \in [1,3]$ for the Burgers' equation 54

10 Reconstructed $u_{i_{N,ROM}}^{(\bar{\epsilon})}(0.5,t)$ compared against analytical and numerical solutions of the Burgers' equation (6.1) 58

11 Basis functions for the double Burgers' equations 65

12 Time coefficients for the double Burgers' equations 66

13 $u_{N,ROM}^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}(0.5,t)$ and $v_{N,ROM}^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}(0.5,t)$ compared against analytical and numerical solutions of the PDEs 68

14 Case I: geometry, boundary conditions, and computational grid . . . 70

15 Case I: first six basis functions of u 72

16 Case I: first six basis functions of v 72

17 Case I: first six basis functions of p 72

FIGURE Page

18 Case I: cumulative energy 73

19 Case I: the first four time coefficients of v 74

20 Case I: the first four time coefficients of u 75

21 Case I: the first four time coefficients of p 76

22 Case I: α_1^p predicted by ODExMFIX using two sets of snapshots . . . 78

23 Case I: comparison of u between MFIX and ODExMFIX 79

24 Case I: comparison of v between MFIX and ODExMFIX 80

25 Case I: comparison of p between MFIX and ODExMFIX 81

26 Case I: ε_u and ε_v at different values of μ_0 84

27 Case I: field of u_g at different conditions 85

28 Case I: field of v_g at different conditions 86

29 Case II: geometry, boundary conditions, and computational grid . . . 87

30 Case II: first six basis functions of p_g 89

31 Case II: first six basis functions of u_g 89

32 Case II: first six basis functions of v_g 89

33 Case II: first six basis functions of u_s 90

34 Case II: first six basis functions of v_s 90

35 Case II: first four time coefficients of u_g obtained from ODExS 92

36 Case II: first four time coefficients of v_g obtained from ODExS 93

37 Case II: first four time coefficients of u_s obtained from ODExS 94

38 Case II: first four time coefficients of v_s obtained from ODExS 95

39 Case II: ϵ_g at $t = 1s$ 96

FIGURE Page

40 Case II: p_g at $t = 1s$ 96

41 Case II: u_g at $t = 1s$ 97

42 Case II: v_g at $t = 1s$ 97

43 Case II: u_s at $t = 1s$ 98

44 Case II: v_s at $t = 1s$ 98

45 Case II: errors of the results of ODExS at off-reference conditions . . 101

46 Case II: field of ϵ_g at $t = 1s$ with different D_p 102

CHAPTER I

INTRODUCTION

A. Statement of the Problem

Reduced order modeling based on the proper orthogonal decomposition (POD) is a conceptually novel and computationally efficient technique for computing unsteady transport phenomena. Compared to the full models which numerically solve the governing partial differential equations (PDEs) of the transport phenomena, POD-based reduced order models (ROMs) contain a smaller number of ordinary differential equations (ODEs). Consequently, the order reduction is achieved by (1) reducing the number of equations, and (2) replacing PDEs by ODEs. The focus of this research is to develop POD-based ROMs for the transport phenomena in fluidized beds and to investigate the accuracy of the POD-based ROMs.

B. Background

This section provides the background information of this study. The background information includes a literature review of the reduced order modeling, an outline of the POD technique, and a description of the gas-solid transport phenomena in fluidized beds.

1. Reduced Order Modeling

The goal of reduced order modeling is to replace the large number of governing PDEs by a smaller number of ODEs. Over the years, investigators have developed a number

---

The journal model is *Journal of Propulsion and Power*.

of techniques for constructing ROMs. Most of the previous work is concentrated on fluid-only flow phenomena. A review of the status of the reduced order modeling can be found in the article of Dowell *et al.*[1]

In structural dynamics, to solve the unsteady vibration problems, a conventional method is to construct ROMs using the eigenmodes of the structure as basis functions.[2] The same technique developed for structural dynamic problems has been applied to the transport phenomena. Florea *et al.*[3] have developed a ROM based on eigenmodes of an unsteady viscous flow in a compressor cascade. Thomas *et al.*[4] have constructed ROMs based on eigenmodes of flows about an isolated airfoil and an aeroelastic wing. Their studies are based on flow simulations in frequency domain using small perturbations. A static/dynamic correction technique has been implemented in their studies in order to improve the accuracy of the eigenmodes based ROMs. Romanowski *et al.*[5] have also constructed ROMs for the Euler equations based on fluid eigenmodes.

POD is an attractive alternative and/or complement to the use of eigenmodes in terms of computational cost and convenience.[1] POD is a procedure for extracting an optimal basis from an ensemble of signals.[6] POD is also called Karhunen-Loéve decomposition, singular value decomposition, principal components analysis, and singular systems analysis.

POD was first introduced to model coherent structures in turbulent flows by Lumley. Using the technique of POD, a series of snapshots obtained from experimental measurements and/or computational simulations, each at a different instant of time, are examined. These solution snapshots are used to form an eigenvalue problem that is solved to determine a set of optimal basis functions for representing the flow field. POD-based ROMs are generated by projecting the governing PDEs onto a space spanned by a small number of POD basis functions. Thus the flow field is described by a small number of ODEs. Background information on POD and POD-based reduced order modeling can be found in the review articles of Berkooz *et al.*[6] and Sirovich.[7]

POD-based ROMs for various systems have been constructed, *e.g.*, Burgers' model of turbulence,[8] Euler equations,[9] and Navier-Stokes equations.[10–14] POD-based ROMs have been developed in the time domain for a number of flows. Deane *et al.*[10] have applied the POD-based ROMs to two two-dimensional flow fields: flow in a periodically grooved channel and wake of an isolated circular cylinder. The short- and long-term accuracy of the POD-based ROMs have been studied through simulation, continuation and bifurcation analysis. Sahan *et al.*[11] have studied the POD-based ROMs applied to non-isothermal transitional grooved-channel flow. In their study, the POD-based ROMs have been derived for transitional flow and heat transfer. Ma *et al.*[12] have studied the POD-based ROM for simulating three-dimensional cylinder flow. Cazemier *et al.*[13] have investigated the POD-based ROMs for driven cavity flows. Rediniotis *et al.*[14] have applied the POD-based ROM to synthetic jets which are essential for flow control applications. Studies in the above references are based on low-speed flows.

POD-based ROMs have also been investigated in high-speed flows. For example, Lucia *et al.*[15] have shown that the POD-based ROM can accurately recreate a flow solution with strong shocks, given that the appropriate data is presented in the snapshots. POD has also been applied in the frequency domain. For example, Hall *et al.*[16] have generated a POD-based ROM for a small-disturbance unsteady two-dimensional inviscid flow about an isolated airfoil.

The application of POD/ROM to flow control, aeroelastic analysis, and iterative design is currently an active field of research. Romanowski[17] has applied the POD-based ROMs to aeroelastic analysis. Ravindran[18] has designed reduced order adaptive controllers for fluids. LeGresley *et al.*[9] have investigated airfoil design optimization using the POD-based ROMs.

2. Proper Orthogonal Decomposition

Suppose we have an ensemble of observations $\{u(x,t_i)\}$. These observations are assumed to form a linear infinite-dimensional Hilbert space L^2 on a spatial domain D.[19] From that ensemble of observations, POD extracts time-independent orthonormal basis functions $\{\phi_k(x)\}$ and time-dependent orthonormal time coefficients $\{\alpha_k(t_i)\}$, such that the reconstruction

$$u(x,t_i) = \sum_k \alpha_k(t_i)\phi_k(x) \tag{1.1}$$

is optimal in the sense that the average least-square truncation error

$$\varepsilon_m = \left\langle \left\| u(x,t_i) - \sum_{j=1}^{m} \alpha_j(t_i)\,\phi_j(x) \right\|^2 \right\rangle \tag{1.2}$$

is a minimum for any given number m of basis functions over all possible sets of basis functions.[19] Herein $||\cdot||$ denotes the L^2-norm given by

$$||f|| = (f,f)^{\frac{1}{2}},$$

where $(,)$ denotes the Euclidean inner product. $\langle\ \cdot\ \rangle$ denotes an ensemble average over a number of observations

$$\langle\ f\ \rangle = \frac{1}{N}\sum_{j=1}^{N} f(x,t_j).$$

The optimum condition specified by (1.2) is equivalent to finding functions ϕ that maximize the normalized averaged projection of u onto ϕ

$$\max_{\phi\in L^2(D)} \frac{\langle |(u,\phi)|^2\rangle}{||\phi||^2}, \tag{1.3}$$

where $|\cdot|$ denotes the modulus.[19]

The optimum condition specified by equation (1.3) reduces to[6]

$$\int_D \langle u(x)u^*(x')\rangle \phi(x')dx' = \lambda\phi(x), \tag{1.4}$$

The POD basis is therefore composed of the eigenfunctions $\{\phi_j\}$ of the integral equation (1.4). The kernel function of the integral equation (1.4) is the averaged autocorrelation function

$$\langle u(x)u^*(x')\rangle \equiv R(x,x').$$

In practice, the state of a numerical model is only available at discrete spatial grid points. Thus the observations in the ensemble are vectors instead of continuous functions. The autocorrelation function in the discrete case is replaced by the tensor product matrix[19]

$$R(x,x') = \frac{1}{M}\sum_{i=1}^{M} u(x,t_i)u^T(x',t_i), \tag{1.5}$$

where M is the number of observations contained in the ensemble.

The derivation of the integral equation (1.3) can be generalized to vector-valued functions such as the three-dimensional velocity fields $\mathbf{u}(\mathbf{x},t)$, where $\mathbf{u} = (u,v,w)$ and $\mathbf{x} = (x,y,z)$. In this case, $R(x,x')$ is replaced by

$$\mathbf{R}(\mathbf{x},\mathbf{x}') = \frac{1}{M}\sum_{i=1}^{M} \mathbf{u}(\mathbf{x},t_i)\mathbf{u}^T(\mathbf{x}',t_i). \tag{1.6}$$

The eigenfunctions $\phi_j(\mathbf{x})$ are also vector valued.

Off-Reference Condition

The POD basis functions are optimal at the reference condition. Herein the reference condition, also called design condition, represents the condition at which the basis functions are obtained. Take the flow field as an example, the POD basis functions are optimal at a special set of flow parameters (*e.g.*, the Reynolds number *Re*). When the flow parameters are not at their reference values, the POD basis functions obtained at the reference condition are no longer optimal. It is straightforward to generate the new set of optimal basis functions at the new condition. It is not computationally practical to provide the matrix R at each condition. There are two situations in which the POD basis functions can be used at off-reference conditions.[20]

First, the POD basis functions can be used at off-reference conditions, if the basis functions are sufficiently insensitive to the flow parameters. For example, the work of Sahan *et al.*[11] showed that the POD-based ROM constructed for simulating non-isothermal transitional grooved-channel flow at reference $Re = 430$ could successfully predict the flow field at Reynolds numbers in the range of $430 \leq Re \leq 1050$. One approach to extend the range of off-reference conditions in which the POD-based ROMs are valid is to generate databases by combining snapshots from different flow conditions (*e.g.*, the work of Ma *et al.*[12]).

Second, if the basis functions possess a property of universal similarity, the POD basis functions can be used at off-reference conditions. The work of Chambers *et al.*[8] explored this possibility by using the Burgers' model of turbulence. They showed that the POD basis functions in the inhomogeneous spatial variables were similar over a range of Reynolds numbers if they were scaled on outer variables. The work of Liu *et al.*[20] also provided experimental evidence of the basis functions similarity. Their results indicated that the POD basis functions of three-dimensional wall turbulence exhibit Reynolds number independence, when scaled properly on outer variables.

3. Transport Phenomena in Fluidized Beds

Fluidization is the phenomenon in which solid particles display fluid-like properties due to the flow of fluids.[21] Figure 1 demonstrates the typical behavior of a fluidized bed. The fluidized bed consists of a vessel containing solid particles and a bottom

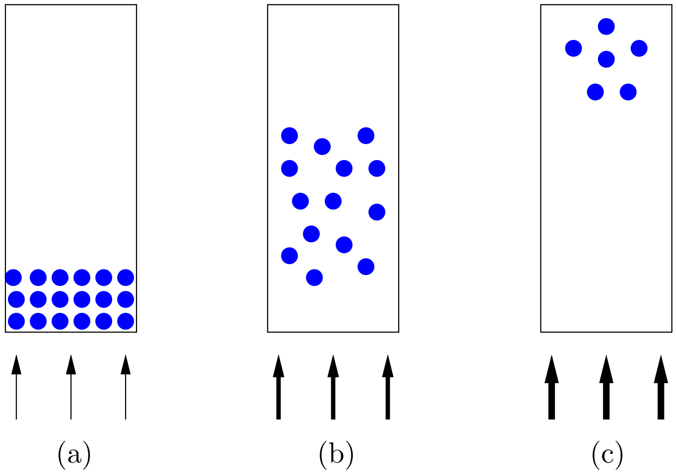


Fig. 1. Typical behavior of a fluidized bed

plate through which gas is injected. At low gas flow velocities as shown in Figure 1(a), the gas percolates through the void spaces between the solid particles and the solid remains a packed bed. When the gas velocity increases over a certain threshold, called the minimum fluidization velocity, the solid particles display fluid-like properties as show in Figure 1(b). This state is called fluidization. If the gas flow velocity is increased beyond the terminal velocity of the solid particles, the solid particles will be swept out of the bed as shown in Figure 1(c).[21]

C. Outline of this Thesis

Chapter II describes the transport equations and boundary conditions used to model the transport phenomena in fluidized beds. Chapter III presents the full numerical model used to simulate the transport equations. Chapter IV describes the general methodology used to construct POD-based ROMs. Chapter V presents the derivation of the POD-based ROMs for approximating the transport equations in fluidized beds. Chapter VI presents the analysis of the accuracy of the POD-based ROMs using two cases of Burgers' equations. Chapter VII presents the results of POD-based ROMs applied to two cases of transport phenomena in fluidized beds. The conclusions and future work are presented in Chapter VIII. Appendix A describes the constitutive models used to close the transport equations. Appendix B presents the algorithm for calculating the convection factors in the full numerical model. Appendixes C-E present samples of input files for the POD-based ROMs.

CHAPTER II

PHYSICAL MODEL

This chapter presents the physical model of the transport phenomena in fluidized beds. This chapter begins with the governing equations used to model the transport phenomena in fluidized beds. Next, the boundary conditions are described.

A. Governing Equations

Under isothermal conditions, the governing equations that model the gas-solid transport phenomena in fluidized beds are the mass and momentum balance equations given below:

- *Gas mass balance*

$$\frac{\partial \epsilon_g \rho_g}{\partial t} + \bigtriangledown \cdot (\epsilon_g \rho_g \vec{v}_g) = 0 \tag{2.1}$$

- *Solid mass balance*

$$\frac{\partial \epsilon_s \rho_s}{\partial t} + \bigtriangledown \cdot (\epsilon_s \rho_s \vec{v}_s) = 0 \tag{2.2}$$

- *Gas momentum balance*

$$\frac{\partial (\epsilon_g \rho_g \vec{v}_g)}{\partial t} + \bigtriangledown \cdot (\epsilon_g \rho_g \vec{v}_g \vec{v}_g) = -\epsilon_g \bigtriangledown p_g + \bigtriangledown \cdot \bar{\bar{\tau}}_g + \epsilon_g \rho_g \vec{g} + F_{gs}(\vec{v}_s - \vec{v}_g) \tag{2.3}$$

- *Solid momentum balance*

$$\frac{\partial (\epsilon_s \rho_s \vec{v}_s)}{\partial t} + \bigtriangledown \cdot (\epsilon_s \rho_s \vec{v}_s \vec{v}_s) = -\epsilon_s \bigtriangledown p_g - \bigtriangledown p_s + \bigtriangledown \cdot \bar{\bar{\tau}}_s + \epsilon_s \rho_s \vec{g} - F_{gs}(\vec{v}_s - \vec{v}_g) \tag{2.4}$$

where ϵ, ρ, and $\vec{v}$ denote the volume fraction, density, and velocity vector. The subscripts g and s denote the gas phase and solid phase, respectively. Expressions for the gas-phase viscous stress $\bar{\bar{\tau}}_g$, gas-solid drag F_{gs}, granular stress $\bar{\bar{\tau}}_s$, and solid pressure p_s are needed to close the governing equations. Constitutive models for these variables can be found in Appendix A and are also given in Syamlal *et al.*[22] and Syamlal.[23] The gas phase is modeled as a gas obeying the ideal gas law

$$\rho_g = \frac{p_g \mathcal{M}}{\mathcal{R} T_g} \tag{2.5}$$

or as an incompressible fluid with constant density. Herein $\mathcal{M}$, $\mathcal{R}$, and T_g denote the average molecular mass of gas, the universal gas constant, and the gas temperature, respectively.

B. Boundary Conditions

Figure 2 illustrates the geometry of a fluidized bed. The left and right boundaries are no-slip walls. At the bottom (inlet) of the bed, gas is injected with steady or unsteady, uniform or nonuniform velocities. At the top (outlet) of the bed, a constant gas pressure is specified.

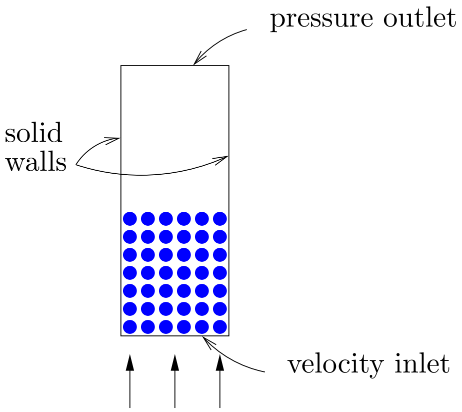


Fig. 2. Geometry and boundary conditions of a fluidized bed

CHAPTER III

FULL NUMERICAL MODEL

The full numerical model represents the traditional numerical model used to solve the transport equations given by equations (2.1)-(2.4). In this study, the numerical algorithm developed at the U.S. Department of Energy's National Energy Technology Laboratory (Syamlal *et al.*[22]) is used to solve the transport equations. The computer code, written in FORTRAN 90, is MFIX (Multiphase Flow with Interphase eXchanges). This chapter presents the discretization used in MFIX.

A. Discretization

MFIX uses a staggered grid arrangement as shown in Figure 3. Scalars are stored at the cell centers. Components of velocity vectors are stored at the cell faces. Equations for scalar variables are solved on the main grid. Equations for velocity components are solved on the staggered grids. If the velocity components and pressure are solved on the same grid, a checkboard pressure field could result. The staggered grid arrangement is used for preventing such unphysical solutions.[23] Using the staggered grid arrangement, MFIX uses three grids, which will be discribed in the following section, to solve a two-dimensional problem.

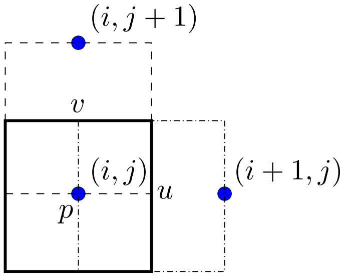


Fig. 3. Grid arrangement in MFIX

B. Discretized Governing Equations

This section describes the two-dimensional discretized governing equations in MFIX. MFIX uses the control volume method to discretize the governing equations.

Mass balance

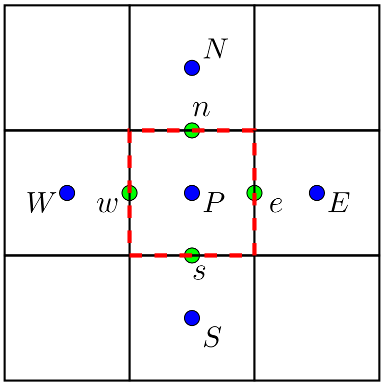


Fig. 4. Control volume for mass balance

For convenience, let us write the mass balance equations (2.1) and (2.2) as

$$\frac{\partial \epsilon_m \rho_m}{\partial t} + \bigtriangledown \cdot (\epsilon_m \rho_m \vec{v}_m) = 0, \tag{3.1}$$

where the subscript m indicates the phase (g or s). Figure 4 shows a control volume for the mass balance equations. P is the center of the control volume. E, W, N, and S represent the east, west, north, and south neighbor cells of the control volume. e, w, n, and s represent the east, west, north, and south faces of the control volume.

Volume fraction ϵ_m and density ρ_m are stored at the cell centers P, E, W, N, and S. In order to discretize the convection terms, volume fraction and density values at the cell faces e, w, n, and s must be evaluated. MFIX uses a convection weighting factor ξ to calculate the volume fraction and density at each face. For example,

$(\epsilon_m\rho_m)$ at the east face is calculated as[23]

$$(\epsilon_m\rho_m)_e = (\xi_m)_e(\epsilon_m\rho_m)_E + (1-(\xi_m)_e)(\epsilon_m\rho_m)_P = (\xi_m)_e(\epsilon_m\rho_m)_E + (\bar{\xi}_m)_e(\epsilon_m\rho_m)_P, \quad (3.2)$$

where $(\xi_m)_e$ is the convection weighting factor for $(\epsilon_m\rho_m)$ at the east face and $(\bar{\xi}_m)_e = 1-(\xi_m)_e$. The algorithm for calculating the convection weighting factor is presented in Appendix B.

Using the convection weighting factor, the mass balance equations are discretized as[23]

$$(a_m)_P(\epsilon_m\rho_m)_P = \sum_{nb}(a_m)_{nb}(\epsilon_m\rho_m)_{nb} + (b_m)_P, \quad (3.3)$$

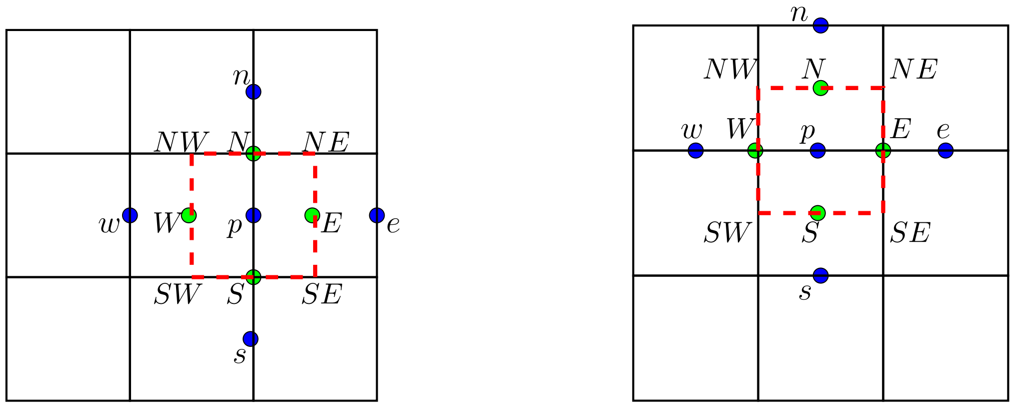
where the subscript nb represents E, W, N, and S. Herein $(a_m)_P$, $(a_m)_{nb}$, and $(b_m)_P$ are defined as

$$
\begin{aligned}
(a_m)_E &= -(\xi_m)_e(u_m)_eA_e, && (3.3a)\\
(a_m)_W &= (\bar{\xi}_m)_w(u_m)_wA_w, && (3.3b)\\
(a_m)_N &= -(\xi_m)_n(v_m)_nA_n, && (3.3c)\\
(a_m)_S &= (\bar{\xi}_m)_s(v_m)_sA_s, && (3.3d)\\
(a_m)_P &= \tfrac{\Delta V}{\Delta t} + ((u_m)_eA_e - (u_m)_wA_w + (v_m)_nA_n - (v_m)_sA_s) \\
&\quad + \textstyle\sum_{nb}(a_m)_{nb}, && (3.3e)\\
(b_m)_P &= (\epsilon_m\rho_m)_P^o\tfrac{\Delta V}{\Delta t}, && (3.3f)
\end{aligned}
$$

where A, ΔV, and Δt denote the face area, cell volume, and time step size, respectively. The superscript o denotes old (previous) time step values.

Momentum balance

Figures 5(a) and 5(b) show the control volumes used to discretize the x-momentum balance equation and y-momentum balance equation, respectively. In Figures 5(a)



(a) x-momentum balance

(b) y-momentum balance

Fig. 5. Control volume for momentum balance

and 5(b), p denotes the center of the control volume; e, w, n, and s represent the east, west, north, and south neighbor cells of the control volume; E, W, N, and S denote the east, west, north, and south faces of the control volume; NE, NW, SE, and SW denote the four corners of the control volume.

In MFIX, the gas and solid x-momentum equations are discretized as[23]

$$\begin{aligned}(a_m^u)_p(u_m)_p &= \sum_{nb}(a_m^u)_{nb}(u_m)_{nb} + (b_m^u)_p \\ &\quad -A_p(\epsilon_m)_p\left((p_g)_E - (p_g)_W\right) + \left(F_{gs}(u_l - u_m)_p\right)\Delta V, \end{aligned} \tag{3.4}$$

where m is used to indicate the phase (gas g or solid s). l denotes the phase other than m. $(a^u_m)_p$, $(a^u_m)_{nb}$, and $(b^u_m)_p$ are defined as

$$(a^u_m)_e = \tfrac{(\mu_m)_E A_E}{\Delta x} - (\xi^u_m)_e(\epsilon_m\rho_m)_e(u_m)_E A_E, \tag{3.4a}$$

$$(a^u_m)_w = \tfrac{(\mu_m)_W A_W}{\Delta x} + (\bar{\xi}^u_m)_w(\epsilon_m\rho_m)_w(u_m)_W A_W, \tag{3.4b}$$

$$(a^u_m)_n = \tfrac{(\mu_m)_N A_N}{\Delta y} - (\xi^u_m)_n(\epsilon_m\rho_m)_n(v_m)_N A_N, \tag{3.4c}$$

$$(a^u_m)_s = \tfrac{(\mu_m)_S A_S}{\Delta y} + (\bar{\xi}^u_m)_s(\epsilon_m\rho_m)_s(v_m)_S A_S, \tag{3.4d}$$

$$(a^u_m)_p = \textstyle\sum_{nb}(a^u_m)_{nb} + (a^u_m)^o_p, \tag{3.4e}$$

$$(a^u_m)^o_p = \tfrac{(\epsilon_m\rho_m)^o_p\Delta V}{\Delta t}, \tag{3.4f}$$

$$(b^u_m)_p = (a^u_m)^o_p(u_m)^o_p + S^u_m, \tag{3.4g}$$

$$\begin{aligned} S^u_m = {} & ((\lambda_m)_E tr(D_m)_E - (\lambda_m)_W tr(D_m)_W)A_p \\ & +(\mu_m)_E\tfrac{(u_m)_e-(u_m)_p}{\Delta x}A_E - (\mu_m)_W\tfrac{(u_m)_p-(u_m)_w}{\Delta x}A_W \\ & +(\mu_m)_N\tfrac{(v_m)_{NE}-(v_m)_{NW}}{\Delta x}A_N - (\mu_m)_S\tfrac{(v_m)_{SE}-(v_m)_{SW}}{\Delta x}A_S. \end{aligned} \tag{3.4h}$$

Similarly, the y-momentum equations are discretized as[23]

$$\begin{aligned} (a^v_m)_p(v_m)_p = {} & \sum_{nb}(a^v_m)_{nb}(v_m)_{nb} + (b^v_m)_p \\ & -A_p(\epsilon_m)_p\left((p_g)_N - (p_g)_S\right) + \left(F_{gs}(v_l - v_m)_p\right)\Delta V, \end{aligned} \tag{3.5}$$

where

$$(a^v_m)_e = \frac{(\mu_m)_E A_E}{\Delta x} - (\xi^v_m)_e(\epsilon_m\rho_m)_e(u_m)_E A_E, \tag{3.5a}$$

$$(a^v_m)_w = \frac{(\mu_m)_W A_W}{\Delta x} + (\bar{\xi}^v_m)_w(\epsilon_m\rho_m)_w(u_m)_W A_W, \tag{3.5b}$$

$$(a^v_m)_n = \frac{(\mu_m)_N A_N}{\Delta y} - (\xi^v_m)_n(\epsilon_m\rho_m)_n(v_m)_N A_N, \tag{3.5c}$$

$$(a^v_m)_s = \frac{(\mu_m)_S A_S}{\Delta y} + (\bar{\xi}^v_m)_s(\epsilon_m\rho_m)_s(v_m)_S A_S, \tag{3.5d}$$

$$(a^v_m)_p = \textstyle\sum_{nb}(a^v_m)_{nb} + (a^v_m)^o_p, \tag{3.5e}$$

$$(a^v_m)^o_p = \frac{(\epsilon_m\rho_m)^o_p\Delta V}{\Delta t}, \tag{3.5f}$$

$$(b^v_m)_p = (a^v_m)^o_p(v_m)^o_p + S^v_m - (\epsilon_m\rho_m)_p g\Delta V, \tag{3.5g}$$

$$\begin{aligned} S^v_m = {} & ((\lambda_m)_N tr(D_m)_N - (\lambda_m)_S tr(D_m)_S)A_p \\ & +(\mu_m)_E\frac{(u_m)_{NE}-(u_m)_{SE}}{\Delta y}A_E - (\mu_m)_W\frac{(u_m)_{NW}-(u_m)_{SW}}{\Delta y}A_W \\ & +(\mu_m)_N\frac{(v_m)_n-(v_m)_p}{\Delta y}A_N - (\mu_m)_S\frac{(v_m)_p-(v_m)_s}{\Delta y}A_S. \end{aligned} \tag{3.5h}$$

Gas pressure correction

An important step in the algorithm of MFIX is the discretization of a gas pressure correction equation. MFIX does not solve the gas mass balance equation. MFIX solves the gas pressure correction equation instead. The gas pressure correction equation is derived from the discretized gas mass balance equation and the discretized momentum balance equations. The gas pressure correction is solved to determine the gas pressure correction, p'_g. The control volume for the gas pressure correction equation is identical to the control volume used for discretizing the mass balance equations. The gas pressure correction equation can be written in the standard form[23]

$$a^p_P(p'_g)_P = \sum_{nb} a^p_{nb}(p'_g)_{nb} + b^p_P, \tag{3.6}$$

where

$$a_E^p = ((\epsilon_g\rho_g)_E\xi_e^p + (\epsilon_g\rho_g)_P\bar{\xi}_e^p)d_{ge}A_e, \tag{3.6a}$$

$$a_W^p = ((\epsilon_g\rho_g)_P\xi_w^p + (\epsilon_g\rho_g)_W\bar{\xi}_w^p)d_{we}A_w, \tag{3.6b}$$

$$a_N^p = ((\epsilon_g\rho_g)_N\xi_n^p + (\epsilon_g\rho_g)_P\bar{\xi}_n^p)d_{gn}A_n, \tag{3.6c}$$

$$a_S^p = ((\epsilon_g\rho_g)_P\xi_s^p + (\epsilon_g\rho_g)_S\bar{\xi}_s^p)d_{gs}A_s, \tag{3.6d}$$

$$a_P^p = \textstyle\sum_{nb} a_{nb}^p, \tag{3.6e}$$

$$\begin{aligned} b_P^p = {} & -\{\left(\tfrac{(\epsilon_g\rho_g)_P-(\epsilon_g\rho_g)_P^o}{\Delta t}\right)\Delta V \\ & + \left((\epsilon_g\rho_g)_E\xi_e^p + (\epsilon_g\rho_g)_P\bar{\xi}_e^p\right)(u_g^*)_eA_e \\ & - \left((\epsilon_g\rho_g)_P\xi_w^p + (\epsilon_g\rho_g)_W\bar{\xi}_w^p\right)(u_g^*)_wA_w \\ & + \left((\epsilon_g\rho_g)_N\xi_n^p + (\epsilon_g\rho_g)_P\bar{\xi}_n^p\right)(v_g^*)_nA_n \\ & - \left((\epsilon_g\rho_g)_P\xi_s^p + (\epsilon_g\rho_g)_S\bar{\xi}_s^p\right)(v_g^*)_sA_s\}. \end{aligned} \tag{3.6f}$$

Herein, the superscript $*$ indicates tentative velocities (*i.e.*, velocities before correction). The velocity corrections along the x-direction are given by[23]

$$(u_m)_p = (u_m^*)_p - d_{mp}((p_g')_E - (p_g')_W), \tag{3.7}$$

where

$$d_{gp} = \frac{A_p\left((\epsilon_g)_p + \frac{(\epsilon_s)_pF_{gs}\Delta V}{(a_s^u)_p+F_{gs}\Delta V}\right)}{(a_g^u)p + \frac{F_{gs}\Delta V(a_s^u)_p}{(a_s^u)_p+F_{gs}\Delta V}}, \tag{3.8}$$

$$d_{sp} = \frac{A_p\left((\epsilon_s)_p + \frac{(\epsilon_g)_pF_{gs}\Delta V}{(a_g^u)_p+F_{gs}\Delta V}\right)}{(a_s^u)p + \frac{F_{gs}\Delta V(a_g^u)_p}{(a_g^u)_p+F_{gs}\Delta V}}. \tag{3.9}$$

Note that in Equations (3.7), (3.8), and (3.9), p is the control volume center shown in Figure 5(a). Similarly, the velocity corrections along the y-direction are given by

$$(v_m)_p = (v_m^*)_p - d_{mp}((p_g')_N - (p_g')_S), \tag{3.10}$$

where p now is the control volume center shown in Figure 5(b).

Solid volume fraction correction

In order to successfully handle dense packing of solids, MFIX derives a solid volume fraction correction equation by including the effect of solid pressure in the discretized solid mass balance equation.[23] In the algorithm of MFIX, the solid volume fraction correction equation is solved instead of the solid mass balance equation. The solid volume fraction equation is written as[23]

$$a_P^{\epsilon}(\epsilon_s')_P = \sum_{nb} a_{nb}^{\epsilon}(\epsilon_s')_{nb} + b_P^{\epsilon}, \tag{3.11}$$

where

$$a_E^{\epsilon} = [(\epsilon_m\rho_m)_e^* e_e (K_s)_E - \xi_e^{\epsilon}(\rho_s)_E (u_s^*)_e]\, A_e, \tag{3.11a}$$

$$a_W^{\epsilon} = \left[(\epsilon_m\rho_m)_w^* e_w (K_s)_W + \bar{\xi}_w^{\epsilon}(\rho_s)_W (u_s^*)_w\right] A_w, \tag{3.11b}$$

$$a_N^{\epsilon} = [(\epsilon_m\rho_m)_n^* e_n (K_s)_N - \xi_n^{\epsilon}(\rho_s)_N (v_s^*)_n]\, A_n, \tag{3.11c}$$

$$a_S^{\epsilon} = \left[(\epsilon_m\rho_m)_s^* e_s (K_s)_S + \bar{\xi}_s^{\epsilon}(\rho_s)_S (v_s^*)_s\right] A_s, \tag{3.11d}$$

$$\begin{aligned} a_P^{\epsilon} = {} & (\rho_s)_P[\bar{\xi}_e^{\epsilon}(u_s^*)_e A_e - \bar{\xi}_w^{\epsilon}(u_s^*)_w A_w \\ & +\bar{\xi}_n^{\epsilon}(v_s^*)_n A_n - \bar{\xi}_s^{\epsilon}(v_s^*)_s A_s] \\ & +(K_s)_P[(\rho_s\epsilon_s^*)_e e_e A_e + (\rho_s\epsilon_s^*)_w e_w A_w \\ & +(\rho_s\epsilon_s^*)_n e_n A_n + (\rho_s\epsilon_s^*)_s e_s A_s] + (\rho_s)_P \tfrac{\Delta V}{\Delta t}, \end{aligned} \tag{3.11e}$$

$$\begin{aligned} b_P^{\epsilon} = {} & -(\rho_s\epsilon_s^*)_e (u_s^*)_e A_e + (\rho_s\epsilon_s^*)_w (u_s^*)_w A_w \\ & -(\rho_s\epsilon_s^*)_n (v_s^*)_n A_n + (\rho_s\epsilon_s^*)_s (v_s^*)_s A_s \\ & - [(\epsilon_s^*\rho_s)_P - (\epsilon_s\rho_s)_P^o]\, \tfrac{\Delta V}{\Delta t}. \end{aligned} \tag{3.11f}$$

Herein $K_s = \frac{\partial p_s}{\partial \epsilon_s}$.

CHAPTER IV

METHODOLOGY FOR REDUCED ORDER MODELING BASED ON PROPER ORTHOGONAL DECOMPOSITION

This chapter presents the general methodology for generating the POD-based ROMs. The methodology for generating the POD-based ROMs consists of three steps: (1) database generation; (2) modal decomposition; and (3) Galerkin projection. For convenience, let us use the following governing PDE to illustrate these three steps:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{D}(\mathbf{u}), \quad \text{in } \Omega \times (0, T], \tag{4.1}$$

where $\mathbf{u}(\mathbf{x}, t)$ is the state vector; Ω is the spatial domain; $(0, T]$ is the temporal domain. Equation (4.1) can represent the Burgers' equation, the Euler equations, the Navier-Stokes equations, or the transport equations (2.1)-(2.4). Additionally, proper boundary conditions and initial conditions must be specified.

A. Database Generation

The database is an ensemble of data that represent solutions of the governing equation (4.1). The database can be numerical solutions of (4.1), experimental measurements, or combination of numerical and experimental data. In this study, the database contains a number of snapshots, each at different momentum of time, obtained from numerical simulations of the governing equation (4.1). MFIX was used to generate the database for the transport equations (2.1)-(2.4) which model the transport phenomena in fluidized beds.

B. Modal Decomposition

Let us assume that a number of snapshots $\mathbf{u}(\mathbf{x}, t_i)$, $i \in [1, M]$ have been generated during the database generation step. Herein M is the total number of snapshots. In the modal decomposition step, POD is applied to the database to extract the basis functions of $\mathbf{u}$. First, $\mathbf{u}$ is decomposed into the mean $\bar{\mathbf{u}}(\mathbf{x})$ and the fluctuation $\mathbf{u}'(\mathbf{x}, t)$, *i.e.*,

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \mathbf{u}'(\mathbf{x}, t), \tag{4.2}$$

where

$$\bar{\mathbf{u}}(\mathbf{x}) = \frac{1}{T} \int_0^T \mathbf{u}(\mathbf{x}, t) dt = \frac{1}{M} \sum_{i=0}^{M} \mathbf{u}(\mathbf{x}, t_i)$$

and

$$\mathbf{u}'(\mathbf{x}, t_i) = \mathbf{u}(\mathbf{x}, t_i) - \bar{\mathbf{u}}(\mathbf{x}),\ i \in [1, M].$$

The tensor product matrix R is calculated as

$$R(\mathbf{x}, \mathbf{x}') = \frac{1}{M} \sum_{i=1}^{M} \mathbf{u}'(\mathbf{x}, t_i) \mathbf{u}'^T(\mathbf{x}', t_i).$$

The basis functions ϕ_k are the eigenvectors of the matrix $R(\mathbf{x}, \mathbf{x}')$. Using the basis functions, $\mathbf{u}(\mathbf{x}, t)$ is reconstructed as

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \sum_{j=1}^{M} \alpha_j(t) \phi_j(\mathbf{x}) = \sum_{j=0}^{M} \alpha_j(t) \phi_j(\mathbf{x}), \tag{4.3}$$

where the zeroth basis function $\phi_0(\mathbf{x})$ is the mean $\bar{\mathbf{u}}(\mathbf{x})$ and $\alpha_0(t) \equiv 1$.

Method of snapshots

A popular technique for finding eigenvalues and eigenvectors of Equation (1.6) is the method of snapshots proposed by Sirovich.[7] The method of snapshots is efficient when the resolution of the spatial domain N is higher than the number of snapshots M. The method of snapshots is based on the fact that the data vectors $\mathbf{u}_i$ and the eigenvectors ϕ_k are spanning the same linear space.[24] As a result, the eigenvectors can be written as a linear combination of the data vectors

$$\phi_k = \sum_{i=1}^{M} v_i^k \mathbf{u}_i, \quad k \in [1, M]. \tag{4.4}$$

If (4.4) is introduced in the eigenvalue problem $\mathbf{R}(\mathbf{x}, \mathbf{x}')\phi(\mathbf{x}) = \lambda\phi(\mathbf{x}')$ we obtain[7]

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}, \tag{4.5}$$

where $\mathbf{v}^k = (v_1^k, v_2^k, \ldots, v_M^k)$ is the kth eigenvector of (4.5); C is a symmetric $M \times M$ matrix defined by[7]

$$C_{ij} = \frac{1}{M}\left(\mathbf{u}'(\mathbf{x}, t_i),\ \mathbf{u}'(\mathbf{x}, t_j)\right). \tag{4.6}$$

Thus the eigenvectors of the $N \times N$ matrix $\mathbf{R}$ are calculated by computing the eigenvectors of the $M \times M$ matrix $\mathbf{C}$. In this study, a code due to Paul Cizmas and Antonio Palacios is used to perform POD using the method of snapshots.

C. Galerkin Projection

The eigenvalues are ordered such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M \geq 0$. The basis functions are also ordered according to their corresponding eigenvalues. If most of the energy is contained in the first m ($m < M$) POD modes, such that $\sum_{j=1}^{m} \lambda_j \simeq \sum_{j=1}^{M} \lambda_j$, it is reasonable to approximate $\mathbf{u}'(\mathbf{x}, t)$ using the first m POD modes:

$$\mathbf{u}(\mathbf{x}, t) \simeq \bar{\mathbf{u}}(\mathbf{x}) + \sum_{j=1}^{m} \alpha_j(t)\phi_j(\mathbf{x}) = \sum_{j=0}^{m} \alpha_j(t)\phi_j(\mathbf{x}). \tag{4.7}$$

Let us substitute the approximation of $\mathbf{u}(\mathbf{x}, t)$ given by equation (4.7) into the governing equation (4.1),

$$\sum_{j=1}^{m} \frac{d\alpha_j(t)}{dt}\phi_j(\mathbf{x}) = \mathbf{D}(\sum_{j=0}^{m} \alpha_j(t)\phi_j(\mathbf{x})). \tag{4.8}$$

When equation (4.8) is projected along the basis function, $\phi_k(\mathbf{x})$,

$$\left(\phi_k,\ \sum_{j=1}^{m}\frac{d\alpha_j(t)}{dt}\phi_j(\mathbf{x})\right) = \left(\phi_k,\ \mathbf{D}(\sum_{j=0}^{m}\alpha_j(t)\phi_j(\mathbf{x}))\right), \tag{4.9}$$

we obtain the ordinary differential equations,

$$\frac{d\alpha_k}{dt} = F_k(\alpha_1,\ldots,\alpha_m), \quad k \in [1,m], \tag{4.10}$$

where the unknowns are the time coefficients $\alpha_k(t)$, $k \in [1,m]$. When deriving equation (4.10) from equation (4.9), we have used the orthonormal property of the basis functions,

$$(\phi_k,\phi_j) = \delta_{kj} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}.$$

Order reduction has been achieved by (1) replacing the PDEs (4.1) by a system of ODEs (4.10), and (2) reducing the number of equations from N to m. The ODEs (4.10) can be integrated using appropriate ODE solvers, *e.g.*, the fourth-order Runge-Kutta method to predict the time history of α_j, $j \in [1,m]$. With the time coefficients obtained from the ODEs (4.10), $\mathbf{u}(\mathbf{x},t)$ can be reconstructed using the approximation (4.7). We can also obtain the values of α_j by directly projecting the database onto the jth basis function,

$$\alpha_j^{POD}(t_k) = (\phi_j(\mathbf{x}),\ \mathbf{u}'(\mathbf{x},t_k)), \quad j \in [1,m],\ k \in [1,M]. \tag{4.11}$$

α^{POD} can be used as reference to examine the accuracy of the POD-based ROM at the reference condition.

D. Summary

This chapter presented the general methodology for generating the POD-based ROMs. The governing equation (4.1) was used to illustrate this methodology. The POD-based ROM generated in this chapter consists of a system of ODEs.

CHAPTER V

REDUCED ORDER MODELS BASED ON PROPER ORTHOGONAL DECOMPOSITION FOR TRANSPORT PHENOMENA

This chapter describes the POD-based ROMs generated to approximate the transport equations (2.1)-(2.4). Two catalogs of POD-based ROMs have been generated. The first catalog includes two POD-based ROMs constructed for gas-only flow phenomena. The second catalog includes one POD-based ROM constructed for gas-solid transport phenomena. These POD-based ROMs are derived from the discretized governing equations described in Section III.B.

A. Reduced order models based on proper orthogonal decomposition for gas-only flow phenomena

For a two-dimensional gas-only flow problem, MFIX solves the gas x-momentum equation (3.4), the gas y-momentum equation (3.5), and the gas pressure correction equation (3.6). For gas-only flow problems, $\epsilon_g \equiv 1$, $\epsilon_s \equiv 0$, and $F_{gs} \equiv 0$. The dependent field variables are the gas pressure p_g and the gas velocities u_g and v_g. When describing the POD-based ROMs for gas-only flows, the subscript g is dropped for convenience. Table I lists the features of the two POD-based ROMs generated for gas-only flows.

1. ODExMFIX

ODExMFIX is a POD-based ROM generated to model gas-only flows. ODExMFIX is derived from the discretized momentum equations (3.4), (3.5) and the gas pressure correction equation (3.6). The discretized momentum equations and the gas pressure

Table I. Reduced order models for gas-only flows

Model	Governing Equations	Unknowns of the ODEs
ODExMFIX	x-momentum balance y-momentum balance pressure correction	α^u, α^v, α^p
ODEx3	x-momentum balance y-momentum balance mass balance	α^u, α^v, α^ρ

correction equation are rearranged as

$$a_p^u u_p - \sum_{nb} a_{nb}^u u_{nb} = b_p^u - \Delta y(p_E - p_W), \tag{5.1}$$

$$a_p^v v_p - \sum_{nb} a_{nb}^v v_{nb} = b_p^v - \Delta x(p_N - p_S), \tag{5.2}$$

$$a_P^p p'_P - \sum_{nb} a_{nb}^p p'_{nb} = b_P^p. \tag{5.3}$$

In ODExMFIX, p, u, and v are approximated using the POD basis functions as

$$u(\mathbf{x},t) = \bar{u}(\mathbf{x}) + u'(\mathbf{x},t) \cong \phi_0^u(\mathbf{x}) + \sum_{i=1}^{m^u} \alpha_i^u(t)\phi_i^u(\mathbf{x}), \tag{5.4}$$

$$v(\mathbf{x},t) = \bar{v}(\mathbf{x}) + v'(\mathbf{x},t) \cong \phi_0^v(\mathbf{x}) + \sum_{i=1}^{m^v} \alpha_i^v(t)\phi_i^v(\mathbf{x}), \tag{5.5}$$

$$p(\mathbf{x},t) \cong \phi_0^p(\mathbf{x}) + \sum_{i=1}^{m^p} \alpha_i^p(t)\phi_i^p(\mathbf{x}). \tag{5.6}$$

where m^u, m^v, and m^p are the number of POD modes used to approximate u, v, and p, respectively. The correction of the pressure, p', is approximated as

$$p'(\mathbf{x},t) \cong \sum_{i=1}^{m^p} (\alpha_i^p)'(t)\phi_i^p(\mathbf{x}). \tag{5.7}$$

Substituting the approximations of u, v, and p' given by (5.4), (5.5), and (5.7) into equations (5.1), (5.2), and (5.3), respectively, yields

$$a_p^u \sum_{i=0}^{m^u} \alpha_i^u \phi_i^u - \sum_{nb} a_{nb}^u \sum_{i=0}^{m^u} \alpha_i^u \phi_{i,nb}^u = b_p^u - \Delta y(p_E - p_W), \tag{5.8}$$

$$a_p^v \sum_{i=0}^{m^v} \alpha_i^v \phi_i^v - \sum_{nb} a_{nb}^v \sum_{i=0}^{m^v} \alpha_i^v \phi_{i,nb}^v = b_p^v - \Delta x(p_N - p_S), \tag{5.9}$$

$$a_P^p \sum_{i=1}^{m^p} (\alpha_i^p)' \phi_i^p - \sum_{nb} a_{nb}^p \sum_{i=1}^{m^p} (\alpha_i^p)' \phi_{i,nb}^p = b_P^p. \tag{5.10}$$

Projecting equations (5.8), (5.9), and (5.10) onto the basis functions ϕ_k^u, ϕ_k^v, and ϕ_k^p, respectively, generates three systems of linear equations:

$$\tilde{\mathcal{A}}^u \alpha^u = \tilde{\mathcal{B}}^u, \tag{5.11}$$

$$\tilde{\mathcal{A}}^v \alpha^v = \tilde{\mathcal{B}}^v, \tag{5.12}$$

$$\tilde{\mathcal{A}}^p \alpha^p = \tilde{\mathcal{B}}^p, \tag{5.13}$$

where

$$\tilde{\mathcal{A}}_{ij}^u = \left((a_p^u \phi_j^u - \sum_{nb} a_{nb}^u \phi_{j,nb}^u),\ \phi_i^u \right),$$

$$\tilde{\mathcal{B}}_i^u = \left([b_p^u - \Delta y(p_E - p_W) - (a_p^u \phi_0^u - \sum_{nb} a_{nb}^u \phi_{0,nb}^u)],\ \phi_i^u \right),$$

$$\tilde{\mathcal{A}}_{ij}^v = \left((a_p^v \phi_j^v - \sum_{nb} a_{nb}^v \phi_{j,nb}^v),\ \phi_i^v \right),$$

$$\tilde{\mathcal{B}}_i^v = \left([b_p^v - \Delta x(p_N - p_S) - (a_p^v \phi_0^v - \sum_{nb} a_{nb}^v \phi_{0,nb}^v)],\ \phi_i^v \right),$$

$$\tilde{\mathcal{A}}_{ij}^p = \left((a_p^p \phi_j^p - \sum_{nb} a_{nb}^p \phi_{j,nb}^p),\ \phi_i^p \right),$$

$$\tilde{\mathcal{B}}_i^p = \left(b_p^p,\ \phi_i^p \right).$$

Herein the dimensions of $\tilde{\mathcal{A}}^u$, $\tilde{\mathcal{A}}^v$ and $\tilde{\mathcal{A}}^p$ are $m^u \times m^u$, $m^v \times m^v$, and $m^p \times m^p$, respectively. The dimensions of $\tilde{\mathcal{B}}^u$, $\tilde{\mathcal{B}}^v$ and $\tilde{\mathcal{B}}^p$ are $m^u \times 1$, $m^v \times 1$, and $m^p \times 1$, respectively. These matrices are calculated using the field variables from the previous iteration. The systems of linear equations (5.11), (5.12), and (5.13) are solved using the LU decomposition method.

ODExMFIX uses an iterative algorithm which is similar to the algorithm used in MFIX. An outline of the solution algorithm in ODExMFIX is given below:

- Using the time coefficients from the previous iteration, reconstruct the field variables p, u and v. For compressible flows, calculate the density ρ using the ideal gas law.

- Solve the system of linear equations (5.11) and obtain the tentative values of $\alpha_i^u(t)$, $i \in [1, m^u]$. The values are called tentative values because they are calculated based on the previous pressure field and they will be corrected based on the pressure correction.

- Solve the system of linear equations (5.12) and obtain the tentative values of $\alpha_i^v(t)$, $i \in [1, m^v]$.

- Solve the system of linear equations (5.13) and obtain $(\alpha_i^p)'(t)$, $i \in [1, m^p]$.

- Correct the time coefficients of p, u and v.

- Check the convergence. If converged, advance to the next time step.

The input data for ODExMFIX are the basis functions of the velocities and the pressure. The solutions of ODExMFIX are α_i^u, $i \in [1, m^u]$, α_i^v, $i \in [1, m^v]$, and α_i^p, $i \in [1, m^p]$. An example of the input file of ODExMFIX is presented in Appendix C.

2. ODEx3

ODEx3 is a POD-based ROM generated to model compressible gas flows. ODEx3 is derived from the discretized gas mass equation (3.3) and the discretized gas momentum equations (3.4) and (3.5). For gas-only flows the dependent variables in MFIX are the pressure p and the velocities u and v. The density field is calculated from the pressure field using the ideal gas law (2.5). ρ is approximated using the basis functions ϕ_i^ρ, $i \in [1, m^\rho]$ as

$$\rho \cong \phi_0^\rho + \sum_{i=1}^{m^\rho} \alpha_i^\rho(t) \phi_i^\rho. \tag{5.14}$$

Consequently, the pressure p is approximated as

$$p \cong \frac{RT}{M_w} \left(\phi_0^\rho + \sum_{i=1}^{m^\rho} \alpha_i^\rho(t) \phi_i^\rho \right). \tag{5.15}$$

Using equations (3.3a)-(3.3f), the discretized gas mass equation (3.3) is rearranged as

$$\begin{aligned} \Delta V \cdot \frac{\rho_P - \rho_P^o}{\Delta t} &= -(\xi_e \rho_E + \bar{\xi}_e \rho_P) u_e \Delta y + (\xi_w \rho_P + \bar{\xi}_w \rho_W) u_w \Delta y \\ &\quad -(\xi_n \rho_N + \bar{\xi}_n \rho_P) v_n \Delta x + (\xi_s \rho_P + \bar{\xi}_s \rho_S) v_s \Delta x, \end{aligned} \tag{5.16}$$

where $\Delta y = A_e = A_w$ and $\Delta x = A_n = A_s$ for two-dimensional flows. Replacing $\frac{\rho_P - \rho_P^o}{\Delta t}$ by $\frac{\partial \rho}{\partial t}$ and substituting the approximations given by (5.4), (5.5), and (5.14) into equation (5.16), one obtains

$$\begin{aligned}\Delta V\sum_{i=1}^{m^\rho}\dot{\alpha}_i^\rho\phi_i^\rho \;=\; & -\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}(\xi_e\phi_{i,E}^\rho+\bar{\xi}_e\phi_i^\rho)\phi_j^u\cdot\alpha_i^\rho\alpha_j^u+\\ & +\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}(\xi_w\phi_i^\rho+\bar{\xi}_w\phi_{i,W}^\rho)\phi_{j,w}^u\cdot\alpha_i^\rho\alpha_j^u-\\ & -\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^v}(\xi_n\phi_{i,N}^\rho+\bar{\xi}_n\phi_i^\rho)\phi_j^v\cdot\alpha_i^\rho\alpha_j^v+\\ & +\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^v}(\xi_s\phi_i^\rho+\bar{\xi}_s\phi_{i,S}^\rho)\phi_{j,s}^v\cdot\alpha_i^\rho\alpha_j^v.\end{aligned}\tag{5.17}$$

Projecting equation (5.17) onto the basis functions ϕ_k^ρ, $k\in[1,m^\rho]$, generates m^ρ ODEs with the form of

$$\check{\mathcal{A}}_{kk}^\rho\dot{\alpha}_k^\rho=\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\check{\mathcal{F}}_{kij}^\rho\alpha_i^\rho\alpha_j^u+\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^v}\check{\mathcal{G}}_{kij}^\rho\alpha_i^\rho\alpha_j^v,\tag{5.18}$$

where

$$\check{\mathcal{A}}_{ij}^\rho=\delta_{ij}\cdot\Delta V,$$

$$\check{\mathcal{F}}_{kij}^\rho=-\left((\xi_e\phi_{i,E}^\rho+\bar{\xi}_e\phi_i^\rho)\phi_j^u,\ \phi_k^\rho\right)+\left((\xi_w\phi_i^\rho+\bar{\xi}_w\phi_{i,W}^\rho)\phi_{j,w}^u,\ \phi_k^\rho\right),$$

$$\check{\mathcal{G}}_{kij}^\rho=-\left((\xi_n\phi_{i,N}^\rho+\bar{\xi}_n\phi_i^\rho)\phi_j^v,\ \phi_k^\rho\right)+\left((\xi_s\phi_i^\rho+\bar{\xi}_s\phi_{i,S}^\rho)\phi_{j,s}^v,\ \phi_k^\rho\right).$$

For compressible gas flows, the discretized momentum equations are

$$a_p^u u_p=\sum_{nb}a_{nb}^u u_{nb}+b_p^u-\Delta y(p_E-p_W),\tag{5.19}$$

$$a_p^v v_p=\sum_{nb}a_{nb}^v v_{nb}+b_p^v-\Delta x(p_N-p_S).\tag{5.20}$$

Substituting equations (3.4e)-(3.4g) into equation (5.19) and substituting equations (3.5e)-(3.5g) into equation (5.20) yields

$$\left(\sum_{nb}a_{nb}^u+\frac{\rho_p^o\Delta V}{\Delta t}\right)u_p=\sum_{nb}a_{nb}^u u_{nb}+\frac{\rho_p^o\Delta V}{\Delta t}u_p^o+S^u-\Delta y(p_E-p_W),\tag{5.21}$$

$$\left(\sum_{nb} a^v_{nb} + \frac{\rho^o_p \Delta V}{\Delta t}\right) v_p = \sum_{nb} a^v_{nb} v_{nb} + \frac{\rho^o_p \Delta V}{\Delta t} v^o_p - \rho_p g \Delta V + S^v - \Delta x (p_N - p_S). \quad (5.22)$$

Equations (5.21) and (5.22) are rearranged as

$$\rho^o_p \Delta V \frac{u_p - u^o_p}{\Delta t} = \sum_{nb} a^u_{nb}(u_{nb} - u_p) + S^u - \Delta y (p_E - p_W), \quad (5.23)$$

$$\rho^o_p \Delta V \frac{v_p - v^o_p}{\Delta t} = \sum_{nb} a^v_{nb}(v_{nb} - v_p) - \rho_p g \Delta V + S^v - \Delta x (p_N - p_S). \quad (5.24)$$

Replacing $\frac{u_p - u^o_p}{\Delta t}$ and $\frac{v_p - v^o_p}{\Delta t}$ by $\frac{\partial u_p}{\partial t}$ and $\frac{\partial v_p}{\partial t}$, respectively, yields

$$\rho^o_p \Delta V \frac{\partial u_p}{\partial t} = \sum_{nb} a^u_{nb}(u_{nb} - u_p) + S^u - \Delta y (p_E - p_W), \quad (5.25)$$

$$\rho^o_p \Delta V \frac{\partial v_p}{\partial t} = \sum_{nb} a^v_{nb}(v_{nb} - v_p) - \rho_p g \Delta V + S^v - \Delta x (p_N - p_S). \quad (5.26)$$

Substituting the approximations of u, v, ρ and p given by equations (5.4), (5.5), (5.14), and (5.15) and the definitions of a^u_{nb} and a^v_{nb} given by equations (3.4a)-(3.4d) and equations (3.5a)-(3.5d) into equations (5.25) and (5.26), yields

$$\begin{aligned}
\rho_p^o \Delta V \cdot \sum_{i=1}^{m^u} \dot{\alpha}_i^u \phi_i^u \quad = \quad & -\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^u} \xi_E^u \phi_{i,e}^\rho \phi_{k,E}^u (\phi_{j,e}^u - \phi_j^u) \Delta y \alpha_i^\rho \alpha_j^u \alpha_k^u \\
& +\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^u} \bar{\xi}_W^u \phi_{i,w}^\rho \phi_{k,W}^u (\phi_{j,w}^u - \phi_j^u) \Delta y \alpha_i^\rho \alpha_j^u \alpha_k^u \\
& -\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^v} \xi_N^u \phi_{i,n}^\rho \phi_{k,N}^v (\phi_{j,n}^u - \phi_j^u) \Delta x \alpha_i^\rho \alpha_j^u \alpha_k^v \\
& +\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^v} \bar{\xi}_S^u \phi_{i,s}^\rho \phi_{k,S}^v (\phi_{j,s}^u - \phi_j^u) \Delta x \alpha_i^\rho \alpha_j^u \alpha_k^v \\
& +\sum_{i=0}^{m^u} \frac{\mu_E \Delta y}{\Delta x} (\phi_{i,e}^u - \phi_i^u) \alpha_i^u + \sum_{i=0}^{m^u} \frac{\mu_W \Delta y}{\Delta x} (\phi_{i,w}^u - \phi_i^u) \alpha_i^u \\
& +\sum_{i=0}^{m^u} \frac{\mu_N \Delta x}{\Delta y} (\phi_{i,n}^u - \phi_i^u) \alpha_i^u + \sum_{i=0}^{m^u} \frac{\mu_S \Delta x}{\Delta y} (\phi_{i,s}^u - \phi_i^u) \alpha_i^u \\
& +S^u - \Delta y \frac{RT}{M_w} \sum_{i=0}^{m^\rho} (\phi_{i,E}^\rho - \phi_{i,W}^\rho) \alpha_i^\rho, \qquad (5.27)
\end{aligned}$$

$$\begin{aligned}
\rho_p^o \Delta V \cdot \sum_{i=1}^{m^v} \dot{\alpha}_i^v \phi_i^v \;\; = \;\; & -\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^v} \xi_E^v \phi_{i,e}^\rho \phi_{j,E}^u (\phi_{k,e}^v - \phi_k^v) \Delta y \alpha_i^\rho \alpha_j^u \alpha_k^v \\
& +\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^v} \bar{\xi}_W^v \phi_{i,w}^\rho \phi_{j,W}^u (\phi_{k,w}^v - \phi_k^v) \Delta y \alpha_i^\rho \alpha_j^u \alpha_k^v \\
& -\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^v}\sum_{k=0}^{m^v} \xi_N^v \phi_{i,n}^\rho \phi_{j,N}^v (\phi_{k,n}^v - \phi_k^v) \Delta x \alpha_i^p \alpha_j^v \alpha_k^v \\
& +\sum_{i=0}^{m^\rho}\sum_{j=0}^{m^v}\sum_{k=0}^{m^v} \bar{\xi}_S^v \phi_{i,s}^\rho \phi_{j,S}^v (\phi_{k,s}^v - \phi_k^v) \Delta x \alpha_i^\rho \alpha_j^v \alpha_k^v \\
& +\sum_{i=0}^{m^v} \frac{\mu_E \Delta y}{\Delta x} (\phi_{i,e}^v - \phi_i^v) \alpha_i^v + \sum_{i=0}^{m^v} \frac{\mu_W \Delta y}{\Delta x} (\phi_{i,w}^v - \phi_i^v) \alpha_i^v \\
& +\sum_{i=0}^{m^v} \frac{\mu_N \Delta x}{\Delta y} (\phi_{i,n}^v - \phi_i^v) \alpha_i^v + \sum_{i=0}^{m^v} \frac{\mu_S \Delta x}{\Delta y} (\phi_{i,s}^v - \phi_i^v) \alpha_i^v \\
& +S^v - g \Delta V \sum_{i=0}^{m^\rho} \phi_i^\rho \alpha_i^\rho - \Delta x \frac{RT}{M_w} \sum_{i=0}^{m^\rho} (\phi_{i,N}^\rho - \phi_{i,S}^\rho) \alpha_i^\rho . \qquad (5.28)
\end{aligned}$$

Project (5.27) onto the basis function ϕ_l^u, $l \in [1, m^u]$ and obtain m^u ODEs

$$\begin{aligned}
\check{\mathcal{A}}_{ll}^u \dot{\alpha}_l^u \;\; = \;\; & \sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^u} \check{\mathcal{F}}_{lijk}^u \alpha_i^\rho \alpha_j^u \alpha_k^u + \sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^v} \check{\mathcal{G}}_{lijk}^u \alpha_i^\rho \alpha_j^u \alpha_k^v + \\
& + \sum_{i=0}^{m^u} \check{\mathcal{H}}_{li}^u \alpha_i^u + \sum_{i=0}^{m^\rho} \check{\mathcal{P}}_{li}^u \alpha_i^\rho + \check{\mathcal{S}}_l^u , \qquad (5.29)
\end{aligned}$$

where

$$\check{\mathcal{A}}_{ij}^u = \delta_{ij} \cdot \left(\rho_p^o \phi_j^u \Delta V ,\ \phi_i^u \right) ,$$

$$\check{\mathcal{F}}_{lijk}^u = \left([-\xi_E^u \phi_{i,e}^\rho \phi_{k,E}^u (\phi_{j,e}^u - \phi_j^u) \Delta y + \bar{\xi}_W^u \phi_{i,w}^\rho \phi_{k,W}^u (\phi_{j,w}^u - \phi_j^u) \Delta y],\ \phi_l^u \right) ,$$

$$\check{\mathcal{G}}_{lijk}^u = \left([-\xi_N^u \phi_{i,n}^\rho \phi_{k,N}^v (\phi_{j,n}^u - \phi_j^u) \Delta x + \bar{\xi}_S^u \phi_{i,s}^\rho \phi_{k,S}^v (\phi_{j,s}^u - \phi_j^u) \Delta x],\ \phi_l^u \right) ,$$

$$\begin{aligned}
\check{\mathcal{H}}_{li}^u \;\; = \;\; & \left(\frac{\mu_E \Delta y}{\Delta x} (\phi_{i,e}^u - \phi_i^u),\ \phi_l^u \right) + \left(\frac{\mu_W \Delta y}{\Delta x} (\phi_{i,w}^u - \phi_i^u),\ \phi_l^u \right) \\
& + \left(\frac{\mu_N \Delta x}{\Delta y} (\phi_{i,n}^u - \phi_i^u),\ \phi_l^u \right) + \left(\frac{\mu_S \Delta x}{\Delta y} (\phi_{i,s}^u - \phi_i^u),\ \phi_l^u \right) ,
\end{aligned}$$

$$\check{\mathcal{P}}^u_{li} = -\frac{RT}{M_w}\Delta y\left((\phi^\rho_{i,E} - \phi^\rho_{i,W}),\ \phi^u_l\right),$$

$$\check{\mathcal{S}}^u_l = \left(S^u,\ \phi^u_l\right).$$

Project (5.28) onto the basis function ϕ^v_l, $l \in [1, m^v]$ and obtain m^v ODEs

$$\begin{aligned}\check{\mathcal{A}}^v_{ll}\dot{\alpha}^v_l \quad = \quad & \sum_{i=0}^{m^\rho}\sum_{j=0}^{m^u}\sum_{k=0}^{m^v}\check{\mathcal{F}}^v_{lijk}\alpha^\rho_i\alpha^u_j\alpha^v_k + \sum_{i=0}^{m^\rho}\sum_{j=0}^{m^v}\sum_{k=0}^{m^v}\check{\mathcal{G}}^v_{lijk}\alpha^\rho_i\alpha^v_j\alpha^v_k + \\ & + \sum_{i=0}^{m^v}\check{\mathcal{H}}^v_{li}\alpha^v_i + \sum_{i=0}^{m^\rho}\check{\mathcal{P}}^v_{li}\alpha^\rho_i + \sum_{i=0}^{m^\rho}\check{\mathcal{M}}^v_{li}\alpha^\rho_i + \check{\mathcal{S}}^v_l,\end{aligned} \tag{5.30}$$

where

$$\check{\mathcal{A}}^v_{ij} = \delta_{ij}\cdot\left(\rho^o_p\phi^v_j\Delta V,\ \phi^v_i\right),$$

$$\check{\mathcal{F}}^v_{lijk} = \left([-\xi^v_E\phi^\rho_{i,e}\phi^u_{j,E}(\phi^v_{k,e} - \phi^v_k)\Delta y + \bar{\xi}^v_W\phi^\rho_{i,w}\phi^u_{j,W}(\phi^v_{k,w} - \phi^v_k)\Delta y],\ \phi^u_l\right),$$

$$\check{\mathcal{G}}^u_{lijk} = \left([-\xi^v_N\phi^\rho_{i,n}\phi^v_{j,N}(\phi^v_{k,n} - \phi^v_k)\Delta x + \bar{\xi}^v_S\phi^\rho_{i,s}\phi^v_{j,S}(\phi^v_{k,s} - \phi^v_k)\Delta x],\ \phi^v_l\right),$$

$$\begin{aligned}\check{\mathcal{H}}^u_{li} \quad = \quad & \left(\frac{\mu_E\Delta y}{\Delta x}(\phi^v_{i,e} - \phi^v_i),\ \phi^v_l\right) + \left(\frac{\mu_W\Delta y}{\Delta x}(\phi^v_{i,w} - \phi^v_i),\ \phi^v_l\right) \\ & + \left(\frac{\mu_N\Delta x}{\Delta y}(\phi^v_{i,n} - \phi^v_i),\ \phi^v_l\right) + \left(\frac{\mu_S\Delta x}{\Delta y}(\phi^v_{i,s} - \phi^v_i),\ \phi^v_l\right),\end{aligned}$$

$$\check{\mathcal{P}}^v_{li} = -\frac{RT}{M_w}\Delta x\left((\phi^\rho_{i,N} - \phi^\rho_{i,S}),\ \phi^v_l\right),$$

$$\check{\mathcal{M}}^v_{li} = -g\Delta V\left(\phi^\rho_i,\ \phi^v_l\right),$$

$$\check{\mathcal{S}}^v_l = \left(S^v,\ \phi^v_l\right).$$

ODEx3 consists of m^ρ ODEs (5.18), m^u ODEs (5.29), and m^v ODEs (5.30). The input data of ODEx3 are the basis functions of ρ, u, and v. The solutions of ODEx3 are α^ρ_i, $i \in [1, m^\rho]$, α^u_i, $i \in [1, m^u]$, and α^v_i, $i \in [1, m^v]$. An example of the input file of ODEx3 is presented in Appendix D.

B. Reduced order model based on proper orthogonal decomposition for gas-solid transport phenomena

For two-dimensional gas-solid transport phenomena, MFIX solves the discretized x-momentum equations of the gas-phase and the solid-phase (3.4), the discretized y-momentum equations of the gas-phase and the solid-phase (3.5), the gas pressure correction equation (3.6), and the solid volume fraction correction equation (3.11). The dependent field variables in MFIX are the gas pressure p_g, the void fraction ϵ_g, and the velocity components of the gas-phase and the solid-phase, u_g, v_g, u_s, and v_s.

Let us assume that the database containing a number of snapshots of p_g, ϵ_g, u_g, u_s, v_g, and v_s has been generated and the POD basis functions $\phi_i^{p_g}$, $\phi_i^{u_g}$, $\phi_i^{v_g}$, $\phi_i^{u_s}$, and $\phi_i^{v_s}$ have been extracted from this database. Herein the POD basis functions of ϵ_g are not computed, because in MFIX, an intermediate variable, the solid volume fraction ϵ_s is introduced and the solid volume fraction correction equation (3.11) is used to solve the corrections of ϵ_s. ϵ_g is computed from ϵ_s as

$$\epsilon_g = 1 - \epsilon_s. \tag{5.31}$$

In this thesis, the POD-based ROM generated to model the gas-solid transport phenomena is called ODExS. In ODExS, the solid volume fraction ϵ_s is also introduced as an intermediate variable. ODExS uses the same solid volume fraction correction equation as used in MFIX. Thus we call ODExS a *hybrid* model, because the order of the solid volume fraction correction equation has not been reduced in ODExS.

Another reason for keeping the solid volume fraction correction equation in ODExS is that void fraction ϵ_g should belong to $[0, 1]$. If ϵ_g was approximated using the POD basis functions of ϕ^{ϵ_g}, at some grid points, the reconstructed void fraction values were larger than 1 because of the approximation using POD basis functions.

Using the approach presented for constructing ODExMFIX, the discretized x-momentum equations, the discretized y-momentum equations, and the discretized gas pressure correction equation are projected onto the basis functions ϕ^{u_m}, ϕ^{v_m}, and ϕ^{p_g}, respectively. Five systems of linear algebraic equations are obtained:

$$\tilde{\mathcal{A}}^{u_m}\alpha^{u_m} = \tilde{\mathcal{B}}^{u_m}, \tag{5.32}$$

$$\tilde{\mathcal{A}}^{v_m}\alpha^{v_m} = \tilde{\mathcal{B}}^{v_m}, \tag{5.33}$$

$$\tilde{\mathcal{A}}^{p_g}\alpha^{p_g} = \tilde{\mathcal{B}}^{p_g}, \tag{5.34}$$

where m denotes the phase g or s. The definitions of $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ were presented in Section A.1.

The input data of ODExS consist of $\phi_i^{p_g}$, $\phi_i^{u_g}$, $\phi_i^{v_g}$, $\phi_i^{u_s}$, $\phi_i^{v_s}$, and the initial field of ϵ_g. The solutions of ODExS include $\alpha_i^{p_g}$, $\alpha_i^{u_g}$, $\alpha_i^{v_g}$, $\alpha_i^{u_s}$, $\alpha_i^{v_s}$, and ϵ_g. An outline of the solution algorithm in ODExS is described below:

- Using the time coefficients from the previous iteration, reconstruct the field variables p_g, u_g, v_g, u_s, and v_s. Calculate physical properties ρ_g and ρ_s. Calculate transport properties μ_g, μ_s, and F_{gs}.

- Solve the systems of linear algebraic equations (5.32) and obtain the tentative values of $\alpha_i^{u_g}(t)$, $i \in [1, m^{u_g}]$ and $\alpha_i^{u_s}(t)$, $i \in [1, m^{u_s}]$.

- Solve the systems of linear algebraic equations (5.33) and obtain the tentative values of $\alpha_i^{v_g}(t)$, $i \in [1, m^{v_g}]$ and $\alpha_i^{v_s}(t)$, $i \in [1, m^{v_s}]$.

- Solve the system of linear equations (5.34) and obtain $(\alpha_i^{p_g})'(t)$, $i \in [1, m^p]$.

- Correct $\alpha_i^{u_g}$, $\alpha_i^{u_s}$, $\alpha_i^{v_g}$, $\alpha_i^{v_s}$, and $\alpha_i^{p_g}$.

- Solve the solid volume fraction correction equation (3.11) and obtain $(\epsilon_s)'$. Correct ϵ_s, $\alpha_i^{u_s}$, and $\alpha_i^{v_s}$.
- Calculate the void fraction using equation (5.31).
- Check the convergence. If converged, advance to the next time step.

An example of the input file for ODExS is presented in Appendix E.

C. Summary

This chapter described the POD-based ROMs generated for the transport phenomena in fluidized beds. Two models for gas-only flows and one model for gas-solid transport phenomena have been presented.

CHAPTER VI

ACCURACY OF REDUCED ORDER MODELS BASED ON PROPER ORTHOGONAL DECOMPOSITION

The proper orthogonal decomposition extracts a set of orthonormal basis functions from a given ensemble of observations. The errors contained in these observations inevitably affect the accuracy of the POD-based ROMs. In this chapter, the accuracy of the POD-based ROMs is analyzed. In order to measure the accuracy, it is better to use some governing equations whose analytical solutions are known. Thus, in this chapter, the accuracy analysis is based on two simple sets of PDEs: a non-homogeneous Burgers' equation and a system of two coupled Burgers' equations. While the Burgers' equation is a significantly simplified model of the transport equations, it is suitable for investigating the properties of the POD-based ROMs applied to the transport phenomena.

A. Non-homogeneous Burgers' Equation

Consider the non-homogeneous Burgers' equation

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\cdot\frac{\partial u(x,t)}{\partial x} = f(x,t), \tag{6.1}$$

where $u(x,t)$ is the dependent variable and

$$\begin{aligned} f(x,t) \;=\; & 0.5\sin(\pi x)\sin(t) - 0.2\sin(2\pi x)\sin(2t) + 0.1\sin(5\pi x)\sin(5t) + \\ & + (x - 0.5\sin(\pi x)\cos(t) + 0.1\sin(2\pi x)\cos(2t) - 0.02\sin(5\pi x)\cos(5t)) * \\ & * (1 - 0.5\pi\cos(\pi x)\cos(t) + 0.2\pi\cos(2\pi x)\cos(2t) - 0.1\pi\cos(5\pi x)\cos(5t)). \end{aligned}$$

Equation (6.1) satisfies the following boundary conditions

$$u(0,t)=0,$$

$$u(1,t)=1$$

and the initial condition

$$u(x,0)=x-0.5\sin(\pi x)+0.1\sin(2\pi x)-0.02\sin(5\pi x).$$

The non-homogeneous term $f(x,t)$ has been chosen such that the analytical solution of equation (6.1) is

$$u_{analytic}(x,t) \;=\; x-0.5\sin(\pi x)\cos(t)+0.1\sin(2\pi x)\cos(2t)-0.02\sin(5\pi x)\cos(5t). \tag{6.2}$$

1. Database Generation

To generate a database for equation (6.1), we need to obtain a solution of equation (6.1). In general, an analytical solution may not be available for equation (6.1). For this reason, a numerical solution must be obtained.

To generate a numerical solution for equation (6.1), let us discretize the spatial domain $[0,1]$ using a mesh with 100 cells of constant length Δx. If the spatial derivative $\frac{\partial u}{\partial x}$ is approximated using a centered, second-order discretization, at each node i in the spatial domain, the PDE (6.1) can be converted into a pseudo-ODE

$$\frac{du_i}{dt}=-u_i\frac{u_{i+1}-u_{i-1}}{2\Delta x}+f(x_i,t), \quad i\in[1,N], \tag{6.3}$$

where $N=99$. By using this approach, the numerical solution of the PDE has been replaced by the numerical solution of a set of N first-order ODEs. Consequently, the order of that system is N.

The LSODI package due to Jeffrey F. Painter and Alan C. Hindmarsh is used to solve the system of ODEs (6.3). The system of ODEs (6.3) is integrated from $t = 0s$ to $t = 50s$. Snapshots are stored every $\Delta t = 0.1s$. Thus there are $M = 501$ snapshots in the numerical solution of PDE (6.1).

In order to analyze how the numerical errors contained in the databases influence the accuracy of the POD-based ROMs, the system of ODEs (6.3) has been solved at four different accuracy levels. In the LSODI package, the accuracy is controlled by specifying the relative tolerance parameter $rtol$ and/or the absolute tolerance parameter $atol$. The convergence criteria in the LSODI package is

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{u^{(j)}(x_i,t_k) - u^{(j-1)}(x_i,t_k)}{rtol \cdot |u^{(j)}(x_i,t_k)| + atol}\right)^2} < 1, \tag{6.4}$$

where $u^{(j)}(x_i, t_k)$ and $u^{(j-1)}(x_i, t_k)$ denote $u(x_i, t_k)$ at current iteration j and previous iteration $j-1$, respectively. Herein the absolute tolerance parameter $atol$ was zero for pure relative error control and the four $rtol$ values corresponding to the four accuracy levels were 0.01%, 0.3%, 5%, and 50%, respectively.

Because equation (6.4) and $rtol$ are not straightforward for measuring the numerical errors contained in the databases, let us define an error ϵ as

$$\epsilon = \frac{|\mathrm{LHS} - \mathrm{RHS}|}{\sqrt{\mathrm{LHS}^2 + \mathrm{RHS}^2}}, \tag{6.5}$$

where LHS and RHS are the left-hand-side and right-hand-side of equation (6.3). The errors corresponding to above four relative tolerance levels were 0.78%, 4.80%, 17.3% and 30.5%, respectively. The errors ϵ were averaged in space and time, and the spatial and temporal average error is $\bar{\epsilon}$. Herein, the time interval was $50s$. Let us use $DB_N^{(\bar{\epsilon})}$ to denote the database corresponding to the error of $\bar{\epsilon}$. $u_{N,PDE}^{(\bar{\epsilon})}(x, t_i)$ represents the ith snapshot in $DB_N^{(\bar{\epsilon})}$.

Additionally, because the analytical solution of the PDE (6.1) is known, a database called DB_A has been generated. The subscript A indicates that the snapshots in DB_A are calculated from the analytical solution (6.2). The kth component of the ith snapshot in DB_A is

$$u_A(x_k, t_i) = u_{analytical}(x_k, t_i) = u_{analytical}(k\Delta x, i\Delta t),$$

where $i \in [1, M]$ and $k \in [1, N]$.

2. Modal Decomposition

We decompose the dependent variable $u(x,t)$ into the mean $\bar{u}(x)$ and the fluctuation $u'(x,t)$:

$$u(x, t_i) = \bar{u}(x) + u'(x, t_i), \quad i \in [1, M].$$

POD is applied to the fluctuation $u'(x, t_i)$ to extract the basis functions.

There are two options for computing the basis functions. The first option is to use the method described in Section I.B.2 to directly extract the eigenfunctions of the tensor product matrix R. R is a $N \times N$ matrix given by

$$R_{ij} = \frac{1}{M} \sum_{k=1}^{M} u'(x_i, t_k) u'^T(x_j, t_k), \quad i, j \in [1, N].$$

Another option for computing the POD basis functions is to use the method of snapshots presented in Section IV.B. The kernel matrix C is computed as

$$C_{ij} = \frac{1}{M} \sum_{k=1}^{N} u'(x_k, t_i) u'^T(x_k, t_j), \quad i, j \in [1, M],$$

and C is a $M \times M$ matrix. In general, $N \gg M$, the method of snapshots is more computational efficient. Since in this case, $N = 99$ and $M = 501$, it is computationally more efficient to use the first option to compute the basis functions. Herein, we still used the method of snapshots. The PODDEC package due to Paul Cizmas and Antonio Palacios has been applied.

Let us denote the ith basis functions obtained from DB_A and $DB_N^{(\bar{\epsilon})}$ by ϕ_{i_A} and $\phi_{i_N}^{(\bar{\epsilon})}$, respectively. Note that the PODDEC package also produces a set of time coefficients $\alpha(t)$ by directly projecting the snapshots onto the basis functions. We use $\alpha_{i_{A,POD}}$ and $\alpha_{i_{N,POD}}^{(\bar{\epsilon})}$ to denote such time coefficients obtained from DB_A and $DB_N^{(\bar{\epsilon})}$, respectively.

3. Galerkin Projection

Using the basis functions, the dependent variable $u(x,t)$ is approximated using the first m basis functions:

$$u(x,t) = \bar{u}(x) + u'(x,t) \cong \bar{u}(x) + \sum_{i=1}^{m} \alpha_i(t)\phi_i(x) = \sum_{i=0}^{m} \alpha_i(t)\phi_i(x), \qquad (6.6)$$

where $\phi_0(x) = \bar{u}(x)$ and $\alpha_0(t) \equiv 1$. The total number of basis functions M is equal to the number of snapshots, *i.e.*, $M = 501$. Substituting the approximation of $u(x,t)$ given by (6.6) into the Burgers' equation (6.1), yields:

$$\sum_{i=1}^{m} \frac{d\alpha_i(t)}{dt}\phi_i(x) + \sum_{i=0}^{m}\sum_{j=0}^{m} \alpha_i(t)\alpha_j(t)\phi_i(x)\frac{d\phi_j(x)}{dx} = f(x,t), \qquad (6.7)$$

where $\frac{d\phi_j(x)}{dx}$ is calculated using the centered, second-order discretization. Projecting equation (6.7) onto the basis functions $\phi_k(x)$,

$$\sum_{i=1}^{m} \frac{d\alpha_i}{dt}\left(\phi_i,\ \phi_k\right) = -\sum_{i=0}^{m}\sum_{j=0}^{m} \alpha_i\alpha_j\left(\phi_i\frac{d\phi_j}{dx},\ \phi_k\right) + \left(f,\ \phi_k\right), \quad k \in [1,m], \qquad (6.8)$$

generates a system of first-order ODEs

$$\frac{d\alpha_k}{dt} = -\sum_{i=0}^{m}\sum_{j=0}^{m} \alpha_i\alpha_j\left(\phi_i\frac{d\phi_j}{dx},\ \phi_k\right) + \left(f,\ \phi_k\right), \quad k \in [1,m]. \qquad (6.9)$$

The unknowns of the ODEs (6.9) are the time coefficients $\alpha_i(t)$, $i \in [1, m]$. Order reduction is achieved if $m \ll N$. Herein the LSODI package is also used to solve the reduced order system of ODEs (6.9). After solving the time coefficients $\alpha(t)$ from the ODEs (6.9), $u(x, t)$ is reconstructed using the approximation given by (6.6).

Let us use $\alpha_{i_{A,ROM}}$ and $\alpha^{(\bar{\epsilon})}_{i_{N,ROM}}$ to denote the solutions of ODEs (6.9) using ϕ_{i_A} and $\phi^{(\bar{\epsilon})}_{i_N}$, respectively. The reconstructed u are represented by $u_{A,ROM}$ and $u^{(\bar{\epsilon})}_{N,ROM}$.

4. Accuracy Analysis

Because the analytical solution of the Burgers' equation (6.1) is given by (6.2), the analytical expressions of the basis functions and their corresponding time coefficients can be derived. These analytical expressions are used as references for the accuracy analyzing.

Here the analytical expressions of the basis functions and time coefficients are derived in the discretized spatial domain. The spatial domain was discretized into $N + 1$ uniform cells with $\Delta x = \frac{1}{N+1}$. The analytical expression of the zeroth basis function is found as

$$\phi_{0_{analytical}}(x) = \bar{u}_{analytical}(x) = \lim_{T \to \infty} \int_0^T u_{analytical}(x, t) = x. \tag{6.10}$$

Consequently,

$$u'_{analytical}(x, t) = -0.5 \sin(\pi x) \cos(t) + 0.1 \sin(2\pi x) \cos(2t) - 0.02 \sin(5\pi x) \cos(5t). \tag{6.11}$$

Note that

$$\sum_{n=1}^{N} \sin(i\pi n\Delta x) \sin(j\pi n\Delta x) = \begin{cases} 0 & \text{if } i \neq j \\ \frac{1}{2\Delta x} & \text{if } i = j \end{cases}, \tag{6.12}$$

thus $\{\pm\sqrt{2\Delta x} \sin(i\pi x)\}$ forms an orthonormal basis in the discretized spatial domain.

Herein, we chose the $\{-\sqrt{2\Delta x}\sin(i\pi x)\}$ to form the basis in order to compare to the numerical basis functions obtained from the PODDEC package. The expression (6.11) implies that $u'_{analytical}$ consists of three modes. The analytical expressions of these basis functions and their corresponding time coefficients are listed below:

$$\begin{aligned}
\phi 0_{analytical}(x) &= x, & \alpha 0_{analytical}(t) &= 1, \\
\phi 1_{analytical}(x) &= -\sqrt{2\Delta x}\sin(\pi x), & \alpha 1_{analytical}(t) &= 0.5\cos(t)/\sqrt{2\Delta x}, \\
\phi 2_{analytical}(x) &= -\sqrt{2\Delta x}\sin(2\pi x), & \alpha 2_{analytical}(t) &= -0.1\cos(2t)/\sqrt{2\Delta x}, \\
\phi 3_{analytical}(x) &= -\sqrt{2\Delta x}\sin(5\pi x), & \alpha 3_{analytical}(t) &= 0.02\cos(5t)/\sqrt{2\Delta x}.
\end{aligned} \tag{6.13}$$

a. Accuracy of basis functions

Using PODDEC, five sets of basis functions have been computed from DB_A and $DB_N^{(\bar{\epsilon})}$. Including the analytical basis functions given by (6.13), six sets of basis functions are compared in this section. Table II lists these six set of basis functions and their corresponding databases. In order to measure how well the basis functions

Table II. Six sets of basis functions and corresponding databases

Basis Function	Database
$\phi_{i_{analytical}}$	
ϕ_{i_A}	DB_A
$\phi_{i_N}^{(0.78)}$	$DB_N^{(0.78)}$
$\phi_{i_N}^{(4.80)}$	$DB_N^{(4.80)}$
$\phi_{i_N}^{(17.3)}$	$DB_N^{(17.3)}$
$\phi_{i_N}^{(30.5)}$	$DB_N^{(30.5)}$

agree with the analytical basis functions, we define an error of the basis function as

$$\bar{\epsilon}_{\phi_i} = \frac{\frac{1}{N}\sum_{k=1}^{N}\left|\phi_i(x_k) - \phi_{i_{analytical}}(x_k)\right|}{\left\|\phi_{i_{analytical}}\right\|_2} \times 100\%, \tag{6.14}$$

where N is the number of components of the basis vectors, which is 99 for the current case. $\bar{\epsilon}_{\phi_i}$ is normalized using the L_2-norm of $\phi_{i_{analytical}}$.

- *Accuracy of* ϕ_{i_A}

Table III lists the errors $\bar{\epsilon}_{\phi_i}$ of ϕ_{i_A} with respect to $\phi_{i_{analytical}}$. These errors are generated by the approximations made while solving the eigenvalue problem using the PODDEC package. Table III shows that ϕ_{i_A} agrees with the analytical basis functions very well. One can conclude that the errors due to the PODDEC package are negligible.

Table III. Errors of ϕ_{i_A}, $i \in [0,3]$

Basis Functions	ϕ_{0_A}	ϕ_{1_A}	ϕ_{2_A}	ϕ_{3_A}
Error	0.0178%	0.0118%	0.0137%	0.0094%

- *Accuracy of* $\phi_{i_N}^{(\bar{\epsilon})}$

Figure 6 shows the first four basis functions of $\phi_{i_N}^{(\bar{\epsilon})}$ compared against the analytical basis functions $\phi_{i_{analytical}}$. There is an excellent agreement between $\phi_{i_{analytical}}$ and $\phi_{i_N}^{(0.78)}$ corresponding to $\bar{\epsilon} = 0.78\%$. $\phi_{0_N}^{(4.80)}$, $\phi_{1_N}^{(4.80)}$, and $\phi_{2_N}^{(4.80)}$ corresponding to $\bar{\epsilon} = 4.80\%$ agree well with the analytical basis functions. Small differences are noticed between $\phi_{3_N}^{(4.80)}$ and $\phi_{3_{analytical}}$. When the numerical error $\bar{\epsilon}$ increases to 17.3%, $\phi_{0_N}^{(17.3)}$, $\phi_{1_N}^{(17.3)}$ and $\phi_{2_N}^{(17.3)}$ have slight oscillations around the analytical values and $\phi_{3_N}^{(17.3)}$ has obvious differences compared against $\phi_{3_{analytical}}$. For $\epsilon = 30.5\%$, large differences were

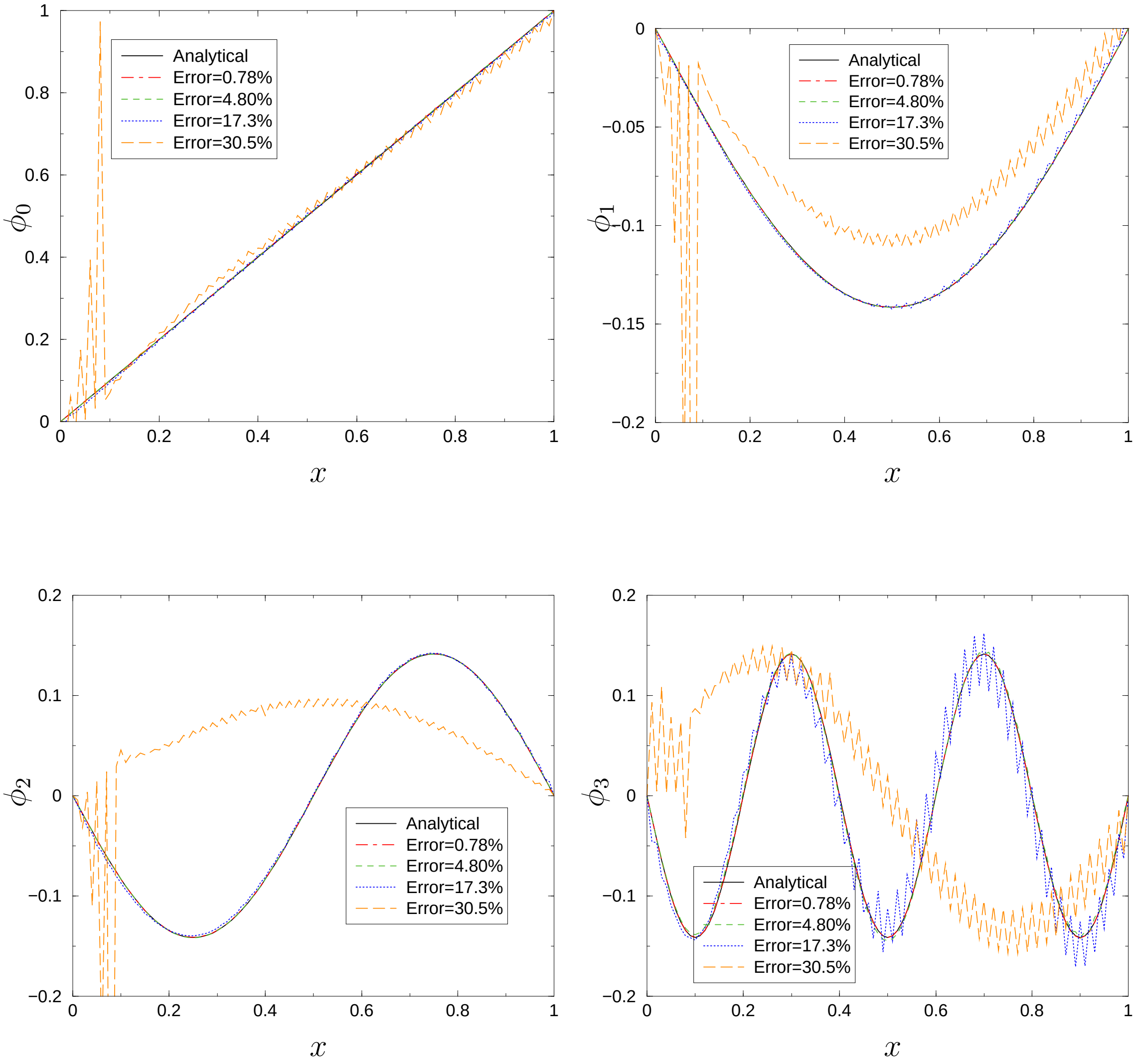


Fig. 6. $\phi_{i_{analytical}}$ and $\phi_{i_N}^{(\bar{\epsilon})}$ for the Burgers' equation

observed between $\phi_{i_N}^{(30.5)}$ and $\phi_{i_{analytical}}$. From the above comparisons, we can conclude that: (1)the accuracy of $\phi_{i_N}^{(\bar{\epsilon})}$ decrease as the numerical error $\bar{\epsilon}$ increases; and (2) basis functions that correspond to smaller eigenvalues are more sensitive to the numerical errors of the databases.

Table IV presents the errors of $\phi_{i_N}^{(\bar{\epsilon})}$ with respect to $\phi_{i_{analytical}}$. The errors listed in Table IV consist of the following two components: (1) numerical errors introduced by the eigenvalue problem solver of the PODDEC package, and (2) numerical errors due to the database $DB_N^{(\bar{\epsilon})}$. In order to analyze the contribution of the errors of the database $DB_N^{(\bar{\epsilon})}$, let us compute the errors of $\phi_{i_N}^{(\bar{\epsilon})}$ with respect to ϕ_{i_A}. Table V lists these errors. Table V proves that the accuracy of $\phi_{i_N}^{(\bar{\epsilon})}$ decreases as $\bar{\epsilon}$ increases.

Table IV. Errors of $\phi_{i_N}^{(\bar{\epsilon})}$ with respect to $\phi_{i_{analytical}}$, $i \in [0, 3]$

$\bar{\epsilon}$	ϕ_{0_N}	ϕ_{1_N}	ϕ_{2_N}	ϕ_{3_N}
0.78%	0.0180%	0.0125%	0.0147%	0.0217%
4.80%	0.0158%	0.0174%	0.0242%	0.2234%
17.3%	0.0741%	0.1376%	0.1818%	1.8900%
30.5%	5.0377%	3.2211%	10.3701%	11.1122%

- *Cumulative energy*

The effect of the numerical errors of the databases on the basis functions can also be observed from the cumulative energy spectrum. Table VI shows the cumulative energy captured by different number of modes used to approximate $u'(x,t)$. Note that $\bar{\epsilon} = 0.00\%$ represents the database of DB_A. For the analytical basis functions,

Table V. Errors of $\phi_{i_N}^{(\bar{\epsilon})}$ with respect to ϕ_{i_A}, $i \in [0,3]$

$\bar{\epsilon}$	ϕ_{0_N}	ϕ_{1_N}	ϕ_{2_N}	ϕ_{3_N}
0.78%	0.0003%	0.0009%	0.0019%	0.0210%
4.80%	0.0046%	0.0113%	0.0262%	0.2225%
17.3%	0.0752%	0.1325%	0.1711%	1.8903%
30.5%	4.9980%	3.2204%	10.3672%	11.1066%

Table VI. Cumulative energy for the Burgers' equation

$\bar{\epsilon}$	1 mode	2 modes	3 modes	4 modes
Analytical	96.003%	99.846%	100.00%	N/A
0.00%	96.005%	99.846%	100.00%	100.00%
0.78%	96.000%	99.845%	99.999%	99.999%
4.80%	95.975%	99.840%	99.997%	99.998%
17.3%	95.639%	99.626%	99.765%	99.855%
30.5%	49.860%	94.681%	97.025%	97.841%

the total energy E is defined as

$$E_{analytical} = \int_0^T \int_0^1 (u'_{analytical})^2 dxdt$$

and the relative energy captured by the kth mode is defined as

$$E_{k_{analytical}} = \frac{1}{E_{analytical}} \int_0^T \int_0^1 (\alpha_{k_{analytical}} \phi_{k_{analytical}})^2 dxdt.$$

In this analysis, the time interval is $T = 50$ seconds. For ϕ_{i_N} and ϕ_{i_A}, the total energy E is defined as the sum of all the eigenvalues[21]

$$E = \sum_{k=1}^{M} \lambda_k$$

and the relative energy captured by the kth mode is defined as

$$E_k = \frac{\lambda_k}{E}.$$

It can be seen from Table VI that as the error $\bar{\epsilon}$ increases, the cumulative energy decreases for the same number of modes. For the analytical solution, $\phi_{4_{analytical}}$ does not exist because $u'_{analytical}(x,t)$ consists of only three modes. When $\bar{\epsilon} > 0$, the energy captured by the fourth mode is not zero due to the numerical errors of the databases.

b. Accuracy of time coefficients

Because $\sum_{k=1}^{3} E_{k_{analytical}} = 100\%$, the number of equations is reduced from 99 to 3 by setting $m = 3$ in ODEs (6.9). The number of equations, however, can be reduced further to 2 by neglecting the third mode. The third mode can be neglected because it captures a relatively small portion of the total energy. As shown in Table VI, the third mode only covers about 0.15% of the total energy when $\bar{\epsilon} < 30.5\%$. Even for an error of 30.5%, the energy of the third mode is 2.3% of the total energy and it could be acceptable to neglect it. In this section, results for both $m = 3$ and $m = 2$ are analyzed. For convenience, ROM(3) and ROM(2) are used to denote the ROM with $m = 3$ and $m = 2$, respectively.

To measure the accuracy of the time coefficients obtained from the ROMs, the error of the time coefficients is defined as

$$\bar{\epsilon}_{\alpha_i} = \frac{\frac{1}{M}\sum_{k=1}^{M}\left|\alpha_i(t_k) - \alpha_{i_{analytical}}(t_k)\right|}{\sqrt{\sum_{k=1}^{M}\alpha_{i_{analytical}}(t_k)^2}} \times 100\%. \tag{6.15}$$

Note this error is defined with respect to the analytical time coefficients. In the following comparisons, some errors were calculated with respect to time coefficients other than the analytical time coefficients.

- *Accuracy of* $\alpha_{i_{analytical,ROM}}$

If the ROM (6.9) is generated using the analytical basis functions $\phi_{i_{analytical}}$, the time coefficients obtained from solving (6.9) are denoted by $\alpha_{i_{analytical,ROM}}$. Tables VII and VIII present the errors $\bar{\epsilon}_{\alpha_i}$ of $\alpha_{i_{analytical,ROM}}$ with respect to $\alpha_{i_{analytical}}$ for ROM(3) and ROM(2), respectively. In Table VII, the errors are generated by solving the ODEs (6.9). As seen from Table VII, the errors caused by solving the ODEs (6.9) are negligible. In Table VIII, the errors are generated by two sources: solving the ODEs (6.9) and neglecting the third mode. The errors caused by neglecting the third mode are also negligible.

Table VII. Errors of $\alpha_{i_{analytical,ROM(3)}}$

Time coefficients	$\alpha_{1_{analytical,ROM(3)}}$	$\alpha_{2_{analytical,ROM(3)}}$	$\alpha_{3_{analytical,ROM(3)}}$
Error	0.0006%	0.0037%	0.0179%

Table VIII. Errors of $\alpha_{i_{analytical,ROM(2)}}$

Time coefficients	$\alpha_{1_{analytical,ROM(2)}}$	$\alpha_{2_{analytical,ROM(2)}}$
Error	0.0174%	0.1753%

- *Accuracy of* $\alpha_{i_{A,ROM}}$

If the ROM is generated using the basis functions of ϕ_{i_A}, the time coefficients obtained from the ODEs (6.9) are called $\alpha_{i_{A,ROM}}$. Tables IX and X present the errors $\bar{\epsilon}_{\alpha_i}$ of $\alpha_{i_{A,ROM}}$ with respect to $\alpha_{i_{analytical}}$. Compared to the errors in Tables VII and VIII, the errors in Tables IX and X increased. The error increase is caused by the errors contained in the basis functions ϕ_{i_A}. The errors in ϕ_{i_A} are due to the numerical errors introduced by the eigenvalue problem solver of the PODDEC package.

Table IX. Errors of $\alpha_{i_{A,ROM(3)}}$ with respect to $\alpha_{i_{analytical}}$

Time coefficients	$\alpha_{1_{A,ROM(3)}}$	$\alpha_{2_{A,ROM(3)}}$	$\alpha_{3_{A,ROM(3)}}$
Error	0.0202%	0.0277%	0.0201%

Table X. Errors of $\alpha_{i_{A,ROM(2)}}$ with respect to $\alpha_{i_{analytical}}$

Time coefficients	$\alpha_{1_{A,ROM(2)}}$	$\alpha_{2_{A,ROM(2)}}$
Error	0.0224%	0.1775%

- *Accuracy of* $\alpha^{(\bar{\epsilon})}_{i_{N,ROM}}$

If the ROM is generated using the basis functions of $\phi^{(\bar{\epsilon})}_{i_N}$, the time coefficients obtained from the ODEs (6.9) are called $\alpha^{(\bar{\epsilon})}_{i_{N,ROM}}$. Figures 7, 8 and 9 show the results of $\alpha^{(\bar{\epsilon})}_{i_{N,ROM}}$ compared against the analytical time coefficients.

As shown in Figure 7, when the error $\bar{\epsilon}$ is as small as 0.78%, the time coefficients obtained from ROM(3) agree very well with the analytical time coefficients. ROM(2) shows slight errors in $\alpha^{(0.78)}_{2_{N,ROM}}(t)$. Similar results are obtained for the case with an

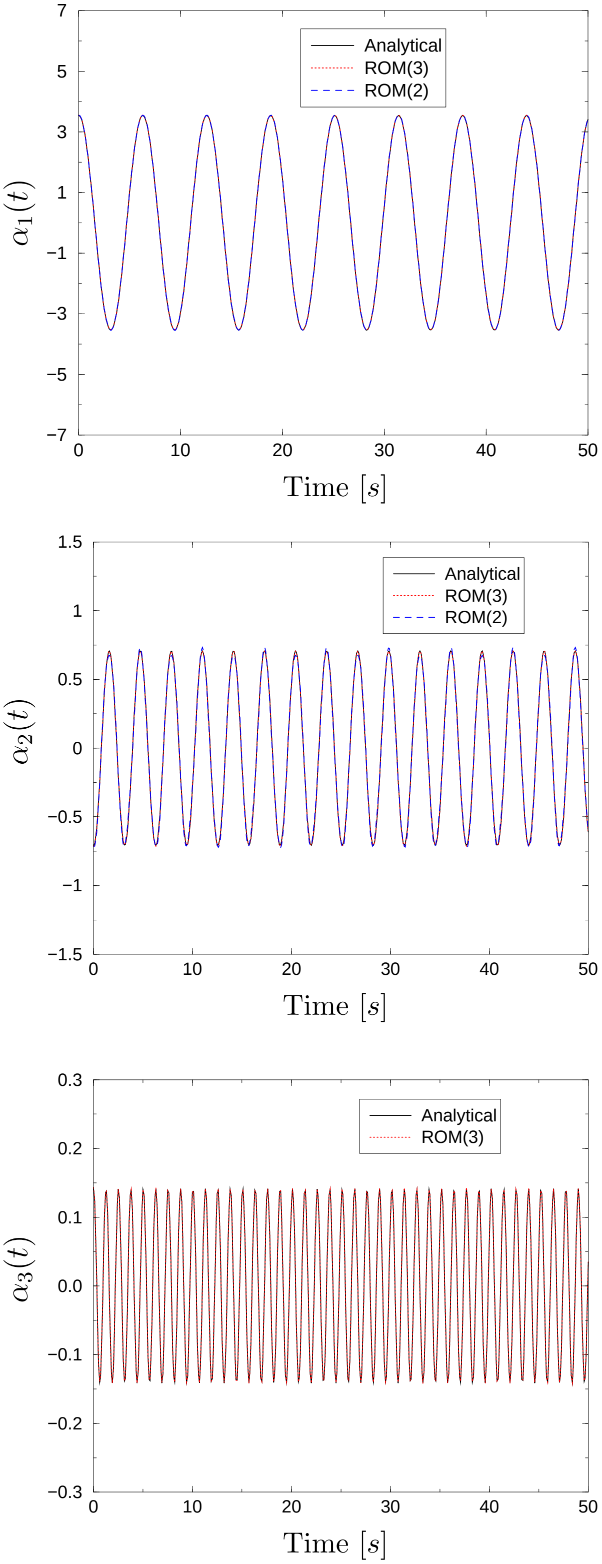


Fig. 7. $\alpha^{(0.78)}_{i_{N,ROM}}(t)$ compared against $\alpha_{i_{analytical}}$, $i \in [1,3]$ for Burgers' equation

error $\bar{\epsilon}$ of 4.8%. This proves that the ROMs provide a high accuracy approximation for the Burgers' equation with a dramatical reduction of order if a set of well-resolved basis functions is used. Figure 8 shows the results corresponding to $\bar{\epsilon} = 17.3\%$. Differences between the analytical time coefficients and time coefficients from ROMs are noticed, especially for the third time coefficient. The results corresponding to $\bar{\epsilon} = 30.5\%$ are shown in Figure 9. At this error level, the time coefficients obtained from ROMs show large errors compared against the analytical time coefficients.

Table XI presents the errors $\bar{\epsilon}_{\alpha_i}$ of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ with respect to $\alpha_{i_{analytical}}$. In Table XI, the errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM(3)}}$ consists of three components: (1) numerical errors due to the database $DB^{\bar{\epsilon}}_N$; (2) errors due to the eigenvalue problem solver of the PODDEC package; and (3) errors due to the ODE solver. The errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM(2)}}$ have an additional error source due to neglecting of the third mode. In order to analyze how the errors of the databases affect the accuracy, we calculated the errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ with respect to $\alpha_{i_{A,ROM}}$. Table XII presents these errors. Table XII shows that the accuracy of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ decreases as $\bar{\epsilon}$ increases. As $\bar{\epsilon}$ increases, the accuracy of the time coefficients corresponding to smaller eigenvalues decrease faster than the time coefficients corresponding to larger eigenvalues.

Table XI. Errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ with respect to $\alpha_{i_{analytical}}$, $i \in [0, 3]$

$\bar{\epsilon}$	$\alpha^{\bar{\epsilon}}_{1_{N,ROM(3)}}$	$\alpha^{\bar{\epsilon}}_{2_{N,ROM(3)}}$	$\alpha^{\bar{\epsilon}}_{3_{N,ROM(3)}}$	$\alpha^{\bar{\epsilon}}_{1_{N,ROM(2)}}$	$\alpha^{\bar{\epsilon}}_{2_{N,ROM(2)}}$
0.78%	0.0206%	0.0288%	0.0234%	0.0226%	0.1778%
4.80%	0.0181%	0.0473%	0.1478%	0.0221%	0.1797%
17.3%	0.0288%	0.2490%	1.6055%	0.0549%	0.3267%
30.5%	1.3119%	12.059%	18.270%	1.1303%	12.407%

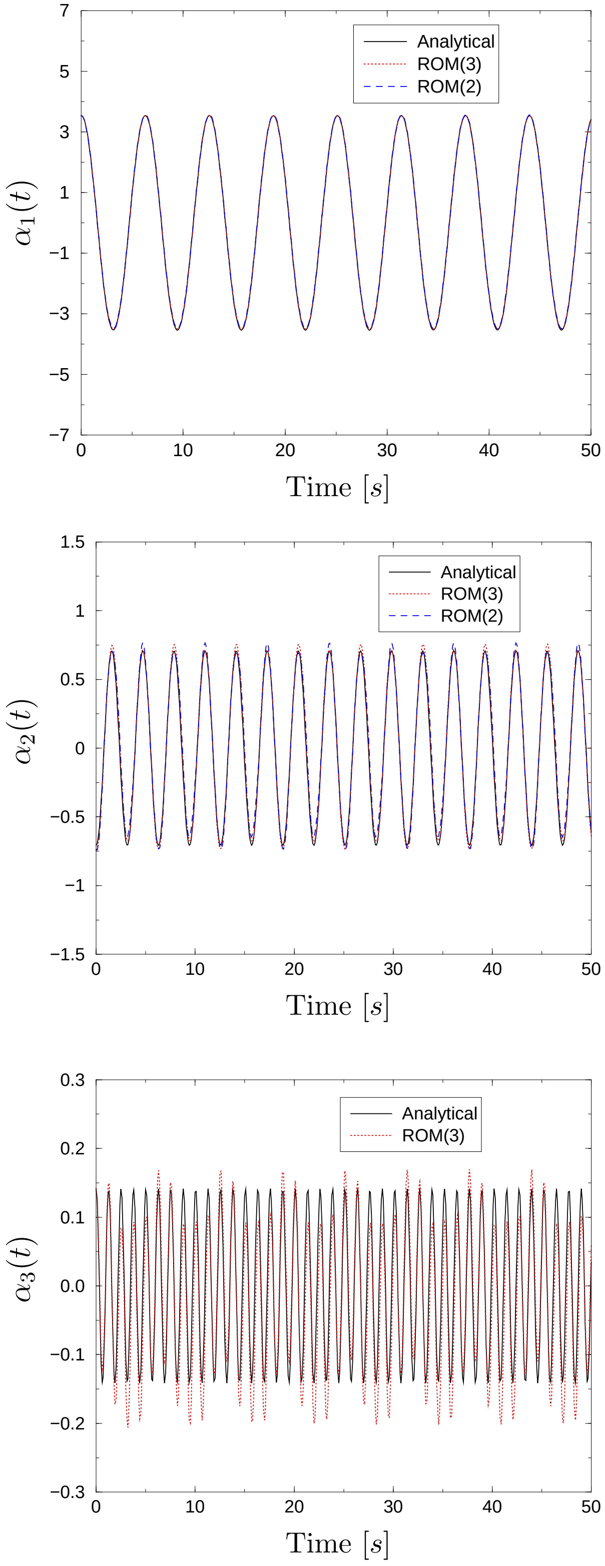


Fig. 8. $\alpha^{(17.3)}_{i_{N,ROM}}(t)$ compared against $\alpha_{i_{analytical}}$, $i \in [1,3]$ for Burgers' equation

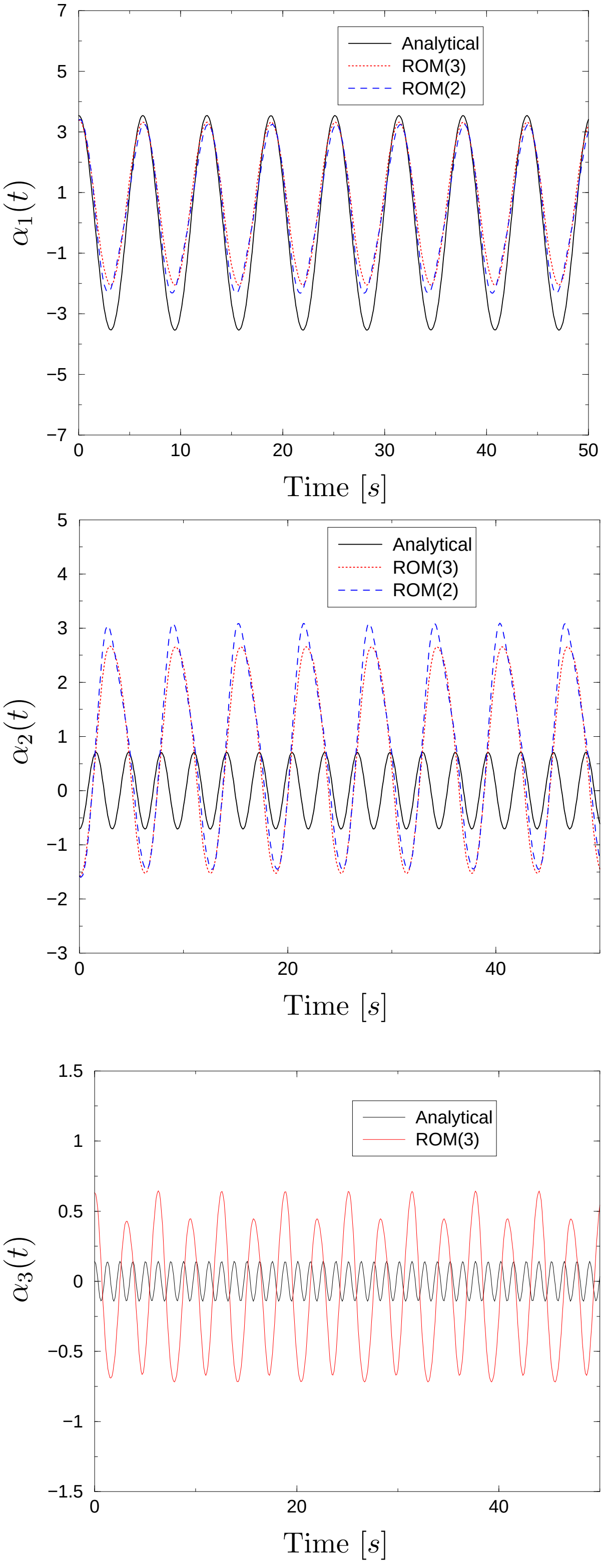


Fig. 9. $\alpha^{(30.5)}_{i_{N,ROM}}(t)$ compared against $\alpha_{i_{analytical}}$, $i \in [1, 3]$ for the Burgers' equation

Table XII. Errors of $\alpha^{\bar{\epsilon}}_{i_{N,ROM}}$ with respect to $\alpha_{i_{A,ROM}}$, $i \in [0,3]$

$\bar{\epsilon}$	$\alpha^{\bar{\epsilon}}_{1_{N,ROM(3)}}$	$\alpha^{\bar{\epsilon}}_{2_{N,ROM(3)}}$	$\alpha^{\bar{\epsilon}}_{3_{N,ROM(3)}}$	$\alpha^{\bar{\epsilon}}_{1_{N,ROM(2)}}$	$\alpha^{\bar{\epsilon}}_{2_{N,ROM(2)}}$
0.78%	0.0004%	0.0030%	0.0178%	0.0005%	0.0039%
4.80%	0.0037%	0.0231%	0.1536%	0.0025%	0.0228%
17.3%	0.0221%	0.2331%	1.6042%	0.0367%	0.2963%
30.5%	1.3005%	12.036%	18.260%	1.1234%	12.370%

c. Accuracy of reconstructed solution u

Using the time coefficients obtained by solving the ODEs (6.9), the dependent variable $u(x,t)$ is reconstructed using the approximation given by (6.6). In this section, the errors are calculated and compared for $u(x,t)$ at $x = 0.5$. Similar to (6.14) and (6.15), the error of the reconstructed $u(0,5,t)$ with respect to the analytical solution $u_{analytical}(0,5,t)$ is defined as

$$\bar{\epsilon}_u = \frac{\frac{1}{M}\sum_{k=1}^{M} |u(0.5,t_k) - u_{analytical}(0.5,t_k)|}{\|u_{analytical}(0.5,t)\|_2} \times 100\%. \tag{6.16}$$

- *Accuracy of reconstructed solution* $u_{analytical,ROM}(0.5,t)$

The solution $u_{analytical,ROM}(0.5,t)$ is reconstructed using $\alpha_{i_{analytical,ROM}}$ and $\phi_{i_{analytical}}$. Table XIII presents the errors $\bar{\epsilon}_u$ of $u_{analytical,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$. The error of $u_{analytical,ROM(3)}(0.5,t)$ is caused by solving the ODEs (6.9) and is only 0.0005%. This proves that the reconstructed solution $u_{analytical,ROM(3)}$ provides a very accurate approximation for the Burgers' equation if the analytical basis functions are used in the ROM(3). If the third mode is neglected, the error of $u_{analytical,ROM(2)}(0.5,t)$ is 0.0865% and still negligible. The contribution of the third mode to $u(x,t)$ is negligible compared to the first mode and the second mode because the third mode only captures 0.15% of the total energy (see Table VI).

Table XIII. Errors of $u_{analytical,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$

$u_{analytical,ROM}(0.5,t)$	$u_{analytical,ROM(3)}(0.5,t)$	$u_{analytical,ROM(2)}(0.5,t)$
Error $\bar{\epsilon}_u$	0.0005%	0.0865%

- *Accuracy of reconstructed solution* $u_{A,ROM}(0.5,t)$

The solution $u_{A,ROM}(0.5,t)$ is reconstructed using $\alpha_{i_{A,ROM}}$ and ϕ_{i_A}. Table XIV presents the errors $\bar{\epsilon}_u$ of $u_{A,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$. Compared to errors in Table XIII, the errors of Table XIV have one additional error source: the errors of eigenvalue problem solver of the PODDEC package. By comparing the values of Table XIII and Table XIV, one concludes that the errors introduced by the PODDEC package have almost no influence on the reconstructed solution $u(0,5,t)$.

Table XIV. Errors of $u_{A,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$

$u_{A,ROM}(0.5,t)$	$u_{A,ROM(3)}(0.5,t)$	$u_{A,ROM(2)}(0.5,t)$
Error $\bar{\epsilon}_u$	0.0005%	0.0865%

- *Accuracy of reconstructed solution* $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$

The solution $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$ is reconstructed using $\alpha^{(\bar{\epsilon})}_{i_{N,ROM}}$ and $\phi^{(\bar{\epsilon})}_{i_N}$. Table XV presents the errors $\bar{\epsilon}_u$ of $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$. Table XV shows that the errors of $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$ increase as $\bar{\epsilon}$ increases. These errors, however, are all smaller than 1%. Figure 10 shows $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$ compared against

Table XV. Errors $\bar{\epsilon}_u$ of $u^{(\bar{\epsilon})}_{N,ROM}(0.5,t)$ with respect to $u_{analytical}(0.5,t)$

$\bar{\epsilon}$	$u^{(\bar{\epsilon})}_{N,ROM(3)}(0.5,t)$	$u^{(\bar{\epsilon})}_{N,ROM(2)}(0.5,t)$
0.78%	0.0004%	0.0866%
4.80%	0.0058%	0.0867%
17.3%	0.0565%	0.0885%
30.5%	0.1210%	0.4108%

$u_{analytical}(0.5,t)$ and $u^{(\bar{\epsilon})}_{N,PDE}(0.5,t)$ at $\bar{\epsilon} = 0.78\%$, 17.3% and 30.5%, respectively. As seen in Figure 10, $u^{(\bar{\epsilon})}_{N,ROM(3)}(0.5,t)$ and $u^{(\bar{\epsilon})}_{N,ROM(2)}(0.5,t)$ agrees with the analytical solution very well when $\bar{\epsilon} = 0.78\%$ and 17.3%. When $\bar{\epsilon} = 30.5\%$, small differences between $u^{(30.5)}_{N,ROM}(0.5,t)$ and $u_{analytical}(0.5,t)$ are observed. Compared to the accuracy of the basis functions and the time coefficients, the accuracy of the reconstructed solution u is less sensitive to the numerical errors of the databases. This result is not surprising since the basis functions errors are compensated by the time coefficients α.

B. "Double" Burgers' Equations

Let us consider a case which consists of two non-homogeneous Burgers' equations. The Burgers' equations for the dependent variables $u(x,t)$ and $v(x,t)$ are given by

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = f(x,t), \tag{6.17}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} = g(x,t). \tag{6.18}$$

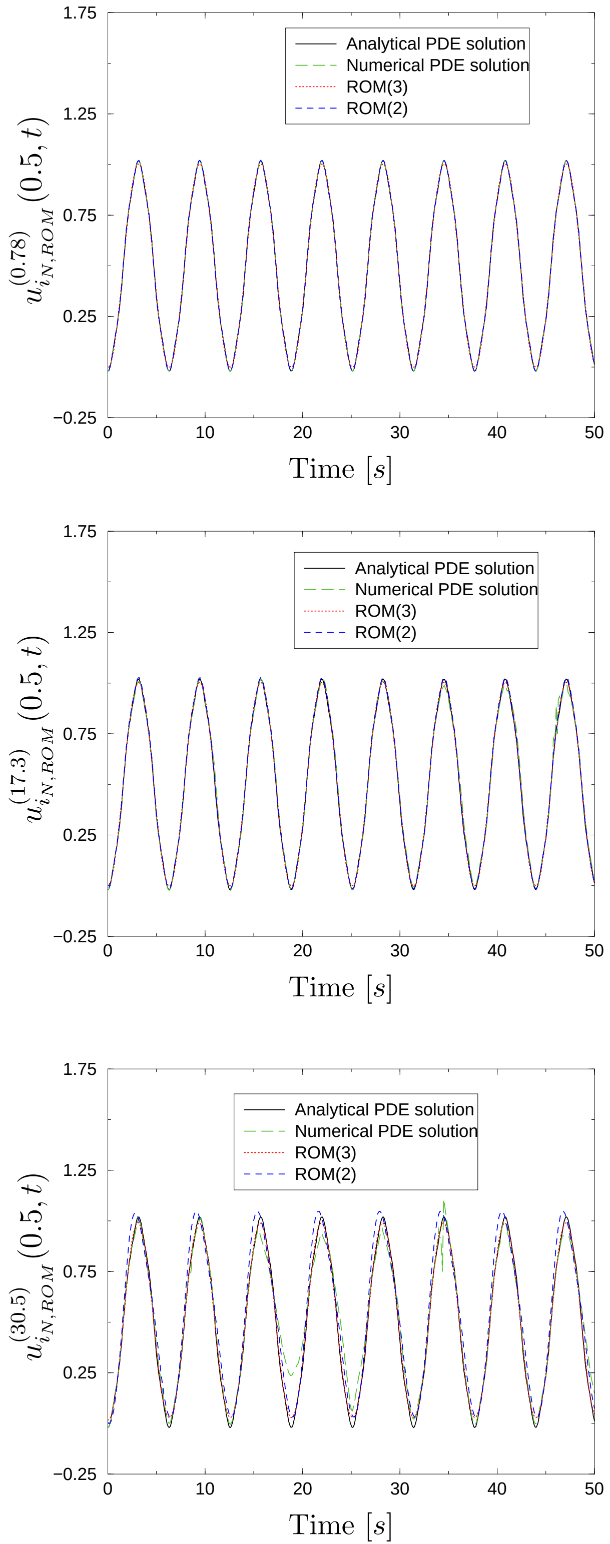


Fig. 10. Reconstructed $u^{(\bar{\epsilon})}_{i_{N,ROM}}(0.5,t)$ compared against analytical and numerical solutions of the Burgers' equation (6.1)

The right side term $f(x,t)$ and $g(x,t)$ are defined as

$$\begin{aligned} f(x,t) &= x\sin(t) + 0.5\sin(\pi x)\cos^2(t) - 0.02\sin(3\pi x)\cos(3t)\cos(t), \quad (6.19) \\ g(x,t) &= 0.5\sin(\pi x)\sin(t) - 0.06\sin(3\pi x)\sin(3t) + (1 - x\cos(t)) \cdot \\ &\quad \cdot (-0.5\pi\cos(\pi x)\cos(t) + 0.06\pi\cos(3\pi x)\cos(3t)). \quad (6.20) \end{aligned}$$

Thus the analytical solutions of equations (6.17) and (6.18) are

$$u_{analytical}(x,t) = 1 - x\cos(t), \quad (6.21)$$

$$v_{analytical}(x,t) = -0.5\sin(\pi x)\cos(t) + 0.02\sin(3\pi x)\cos(3t). \quad (6.22)$$

1. Database Generation

To obtain a numerical solution of the PDEs (6.17) and (6.18), let us discretize the spatial domain [0,1] using a mesh with 100 uniform cells and approximate the spatial derivative $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial x}$ using the centered, second-order discretization. Thus at each node i in the spatial domain, the PDEs (6.17) and (6.18) are converted into the pseudo-ODEs

$$\frac{du_i}{dt} = -v_i \frac{u_{i+1} - u_{i-1}}{2\Delta x} + f(x_i, t), \quad i \in [1, N] \quad (6.23)$$

and

$$\frac{dv_i}{dt} = -u_i \frac{v_{i+1} - v_{i-1}}{2\Delta x} + g(x_i, t), \quad i \in [1, N], \quad (6.24)$$

where $N = 99$. The numerical solution of the PDEs (6.17) and (6.18) has been replaced by the numerical solution of a set of $2 \times N$ first-order ODEs. Consequently, the order of that system is $2 \times N$. The ODEs (6.23) and (6.24) are integrated from $t = 0s$ to $t = 50s$ using the LSODI package. Snapshots are stored every $0.1s$. Thus there are 501 snapshots for u and 501 snapshots for v in the database.

For the “double” Burgers’ equations, two relative tolerance levels were used:

$rtol = 1 \times 10^{-3}$ and $rtol = 1 \times 10^{-4}$. For a relative tolerance parameter larger than $rtol = 1 \times 10^{-3}$, the LSODI package diverged before $t = 50s$. Similar to the error ϵ defined by equation (6.5), two errors ϵ_u and ϵ_v are defined for equation (6.23) and (6.24), respectively:

$$\epsilon_u = \frac{|\mathrm{LHS}_u - \mathrm{RHS}_u|}{\sqrt{\mathrm{LHS}_u^2 + \mathrm{RHS}_u^2}},$$
$$\epsilon_v = \frac{|\mathrm{LHS}_v - \mathrm{RHS}_v|}{\sqrt{\mathrm{LHS}_v^2 + \mathrm{RHS}_v^2}},$$

where LHS_u and RHS_u are the left-hand-side and right-hand-side of equation (6.23); LHS_v and RHS_v are the left-hand-side and right-hand-side of equation (6.24). Table XVI presents the average error $\bar{\epsilon}_u$ for equation (6.23) and $\bar{\epsilon}_v$ for equation (6.24). For convenience, let us use $DB_N^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}$ to denote the the database corresponding to the errors of $\bar{\epsilon}_u$ and $\bar{\epsilon}_v$. Additionally, using the analytical solutions (6.21) and (6.22), the database DB_A is generated. The snapshots in DB_A are calculated as

Table XVI. $\bar{\epsilon}_u$ and $\bar{\epsilon}_v$

$rtol$	$\bar{\epsilon}_u$	$\bar{\epsilon}_v$
1×10^{-4}	0.03%	0.7%
1×10^{-3}	0.1%	1.5%

$$u_A(x, t_k) = u_{analytical}(x, t_k) = u_{analytical}(x, k\Delta t), \quad k \in [1, M],$$

$$v_A(x, t_k) = v_{analytical}(x, t_k) = v_{analytical}(x, k\Delta t), \quad k \in [1, M],$$

where $M = 501$ and $\Delta t = 0.1s$.

2. Model Decomposition

Three databases DB_A, $DB_N^{(0.03;0.7)}$ and $DB_N^{(0.1;1.5)}$ have been generated. Each database consists of M snapshots $u(x,t_i)$, $i \in [1,M]$ and M snapshots $v(x,t_i)$, $i \in [1,M]$. We decompose $u(x,t_i)$ and $v(x,t_i)$ into the means and the fluctuations:

$$u(x,t_i) = \bar{u}(x) + u'(x,t_i), \quad i \in [1,M],$$

$$v(x,t_i) = \bar{v}(x) + v'(x,t_i), \quad i \in [1,M].$$

The method of snapshots was applied to $u'(x,t_i)$ and $v'(x,t_i)$ to extract the basis functions of ϕ^u and ϕ^v, respectively. The kernel matrices are computed as

$$C^u_{ij} = \frac{1}{M}\sum_{k=1}^{N} u'(x_k,t_i)u'^T(x_k,t_j), \quad i,j \in [1,M]$$

and

$$C^v_{ij} = \frac{1}{M}\sum_{k=1}^{N} v'(x_k,t_i)v'^T(x_k,t_j), \quad i,j \in [1,M].$$

3. Galerkin Projection

Using the basis functions ϕ^u_i and ϕ^v_i, the dependent variables $u(x,t)$ and $v(x,t)$ are approximated as

$$u(x,t) \cong \bar{u}(x) + \sum_{i=1}^{m^u} \alpha^u_i(t)\phi^u_i(x) = \sum_{i=0}^{m^u} \alpha^u_i(t)\phi^u_i(x), \tag{6.25}$$

$$v(x,t) \cong \bar{v}(x) + \sum_{i=1}^{m^v} \alpha^v_i(t)\phi^v_i(x) = \sum_{i=0}^{m^v} \alpha^v_i(t)\phi^v_i(x), \tag{6.26}$$

where m^u and m^v are the number of POD modes used to approximate $u'(x,t)$ and $v'(x,t)$, respectively.

Substituting the approximations of u and v given by (6.25) and (6.26) into the

PDEs (6.23) and (6.24), yields

$$\sum_{i=1}^{m^u} \frac{d\alpha_i^u}{dt}\phi_i^u = -\sum_{i=0}^{m^u}\sum_{j=0}^{m^v} \alpha_i^u \alpha_j^v \phi_j^v \frac{d\phi_i^u}{dx} + f, \tag{6.27}$$

$$\sum_{i=1}^{m^v} \frac{d\alpha_i^v}{dt}\phi_i^v = -\sum_{i=0}^{m^v}\sum_{j=0}^{m^u} \alpha_j^u \alpha_i^v \phi_j^u \frac{d\phi_i^v}{dx} + g, \tag{6.28}$$

where $\frac{d\phi}{dx}$ is computed using the centered, second-order discretization. Now, let us project equation (6.27) onto the basis function ϕ_k^u and equation (6.28) onto ϕ_k^v. We obtain the ODEs

$$\frac{d\alpha_k^u}{dt} = -\sum_{i=0}^{m^u}\sum_{j=0}^{m^v} \alpha_i^u \alpha_j^v \left(\phi_j^v \frac{d\phi_i^u}{dx},\ \phi_k^u\right) + \left(f,\ \phi_k^u\right), \quad k \in [1, m^u], \tag{6.29}$$

$$\frac{d\alpha_k^v}{dt} = -\sum_{j=0}^{m^u}\sum_{i=0}^{m^v} \alpha_j^u \alpha_i^v \left(\phi_j^u \frac{d\phi_i^v}{dx},\ \phi_k^v\right) + \left(g,\ \phi_k^v\right), \quad k \in [1, m^v]. \tag{6.30}$$

Thus, the POD-based ROM consists of m^u first-order ODEs given by (6.29) and m^v first-order ODEs given by (6.30). Consequently, the order of the POD-based ROM is $m^u + m^v$. The LSODI package is used to solve the POD-based ROM.

4. Accuracy Analysis

Because the analytical solutions of the PDEs (6.17) and (6.18) are given by (6.21) and (6.22), the analytical expressions of the basis functions and the time coefficients were derived using the approach described in Section A.4. The analytical basis functions and time coefficients for $u(x,t)$ are

$$\begin{aligned} &\phi^u_{0_{analytical}}(x) = 1, && \alpha^u_{0_{analytical}}(t) = 1, \\ &\phi^u_{1_{analytical}}(x) = -\sqrt{3\Delta x}x, && \alpha^u_{1_{analytical}}(t) = \cos(t)/\sqrt{3\Delta x}, \end{aligned} \tag{6.31}$$

and for $v(x,t)$ they are

$$\begin{aligned}
&\phi^v_{0_{exact}}(x) = 0, && \alpha^v_{0_{exact}}(t) = 1, \\
&\phi^v_{1_{analytical}}(x) = -\sqrt{2\Delta x}\sin(\pi x), && \alpha^v_{1_{analytical}}(t) = 0.5\cos(t)/\sqrt{2\Delta x}, \\
&\phi^v_{2_{analytical}}(x) = -\sqrt{2\Delta x}\sin(3\pi x), && \alpha^v_{2_{analytical}}(t) = -0.02\cos(3t)/\sqrt{2\Delta x}.
\end{aligned} \tag{6.32}$$

These analytical basis functions and time coefficients are used as references for the following accuracy analyses.

a. Accuracy of basis functions

Table XVII presents the errors ϵ_{ϕ_i} of ϕ_{i_A} with respect to $\phi_{i_{analytical}}$. Herein the error ϵ_{ϕ_i} is defined by equation (6.14). The errors of ϕ_{i_A} are generated by the PODDEC package and are negligible.

Table XVII. Errors of $\phi^u_{i_A}$ and $\phi^v_{i_A}$

Basis Functions	$\phi^u_{1_A}$	$\phi^v_{1_A}$	$\phi^v_{2_A}$
Error	0.0639%	0.0021%	0.0022%

Table XVIII presents the errors ϵ_{ϕ_i} of $\phi^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_N}$ with respect to $\phi_{i_{analytical}}$. Figure 11 shows the basis functions $\phi^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_N}$ compared against the analytical basis functions. The

Table XVIII. Errors of $\phi^{u,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_A}$ and $\phi^{v,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_A}$

$\bar{\epsilon}_u$	$\bar{\epsilon}_v$	$\phi^{u,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{1_N}$	$\phi^{v,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{1_N}$	$\phi^{v,(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{2_N}$
0.03%	0.7%	0.0639%	0.0051%	0.1976%
0.1%	1.5%	0.0681%	0.0338%	1.1576%

errors of $\phi_{i_N}^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}$ are generated by the PODDEC package and the numerical errors of the databases. As shown in Table XVIII and Figure 11, the accuracy of the basis functions decreases as the numerical errors of the databases increase. Basis functions which correspond to smaller eigenvalues are more sensitive to the numerical errors of the databases. Herein the second basis function of v is the most sensitive to the numerical errors of the databases.

b. Accuracy of time coefficients

The analytical solutions (6.21) and (6.22) imply that $u(x,t)$ and $v(x,t)$ consist of two modes and three modes, respectively. Herein the zeroth modes are counted. Thus when generating the POD-based ROMs, we choose $m^u = 1$ and $m^v = 2$. The number of equations is reduced from $2 \times N$ to $m^u + m^v$.

Four sets of basis functions are used to generate the POD-based ROMs. These four sets of basis functions are $\phi_{i_{analytical}}$, ϕ_{i_A}, $\phi_{i_N}^{(0.03;0.7)}$ and $\phi_{i_N}^{(0.1;1.5)}$. The time coefficients obtained from the POD-based ROM using these four sets of basis functions are denoted by $\alpha_{i_{analytical,ROM}}$, $\alpha_{i_{A,ROM}}$, $\alpha_{i_{N,ROM}}^{(0.03;0.7)}$ and $\alpha_{i_{N,ROM}}^{(0.1;1.5)}$, respectively. Table XIX presents the errors of the time coefficients obtained from the POD-based ROM with respect to the analytical time coefficients. Herein the error of the time coefficients is defined by equations (6.15). As shown in Table XIX, the errors of the time coefficients obtained from the POD-based ROM increase when the errors of the basis functions increases. Figure 12 shows the time coefficients of $\alpha_{i_{N,ROM}}^{(0.1;1.5)}$ and $\alpha_{i_{N,ROM}}^{(0.03;0.7)}$ compared against $a_{i_{analytical}}$. At the error level of $\bar{\epsilon}_u = 0.03\%$ and $\bar{\epsilon}_v = 0.7\%$, $\alpha_{1_{N,ROM}}^{u,(0.03;0.7)}$ and $\alpha_{1_{N,ROM}}^{v,(0.03;0.7)}$ agree very well with $\alpha_{1_{analytical}}^{u}$ and $\alpha_{1_{analytical}}^{v}$, respectively. Small differences between $\alpha_{2_{N,ROM}}^{v,(0.03;0.7)}$ and $\alpha_{2_{analytical}}^{v}$ are noticed. When the errors increase to $\bar{\epsilon}_u = 0.1\%$ and $\bar{\epsilon}_v = 1.5\%$, only $\alpha_{1_{N,ROM}}^{u,(0.1;1.5)}$ agrees very well with $\alpha_{1_{analytical}}^{u}$. There are small differences between $\alpha_{1_{N,ROM}}^{v,(0.1;1.5)}$ and $\alpha_{1_{analytical}}^{v}$. Large differences between $\alpha_{2_{N,ROM}}^{v,(0.1;1.5)}$

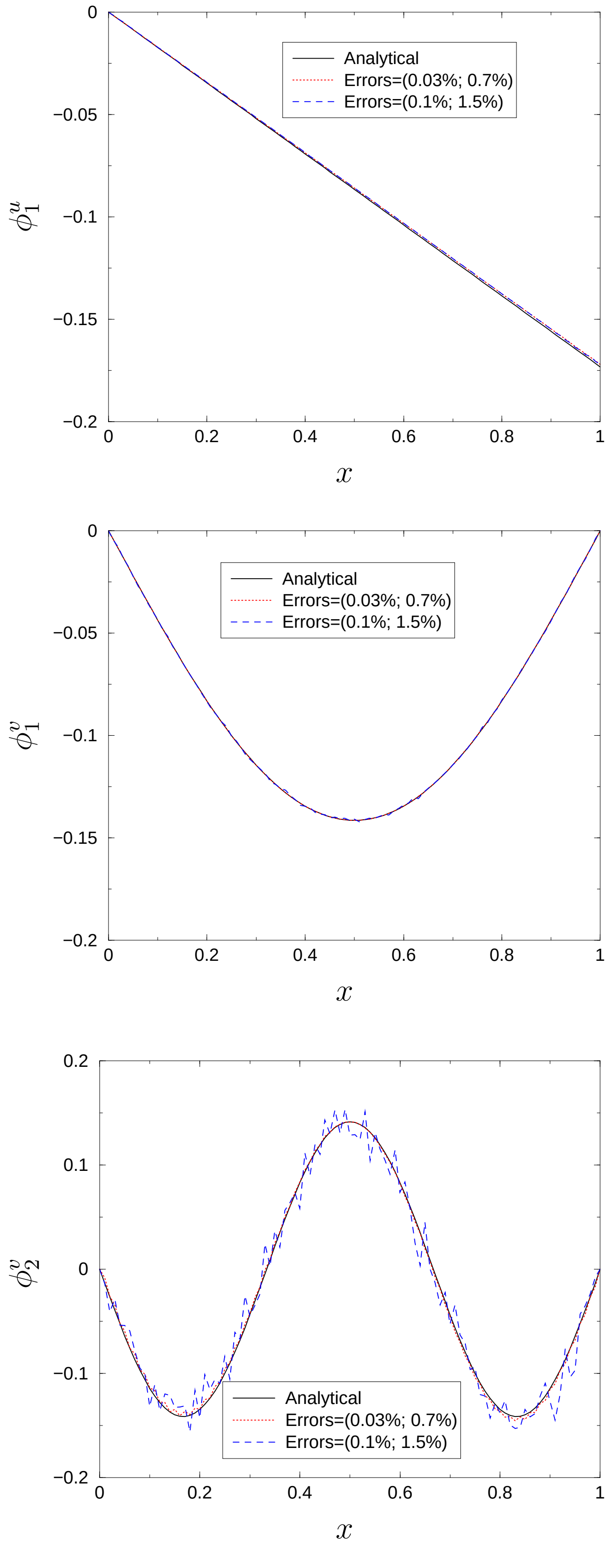


Fig. 11. Basis functions for the double Burgers' equations

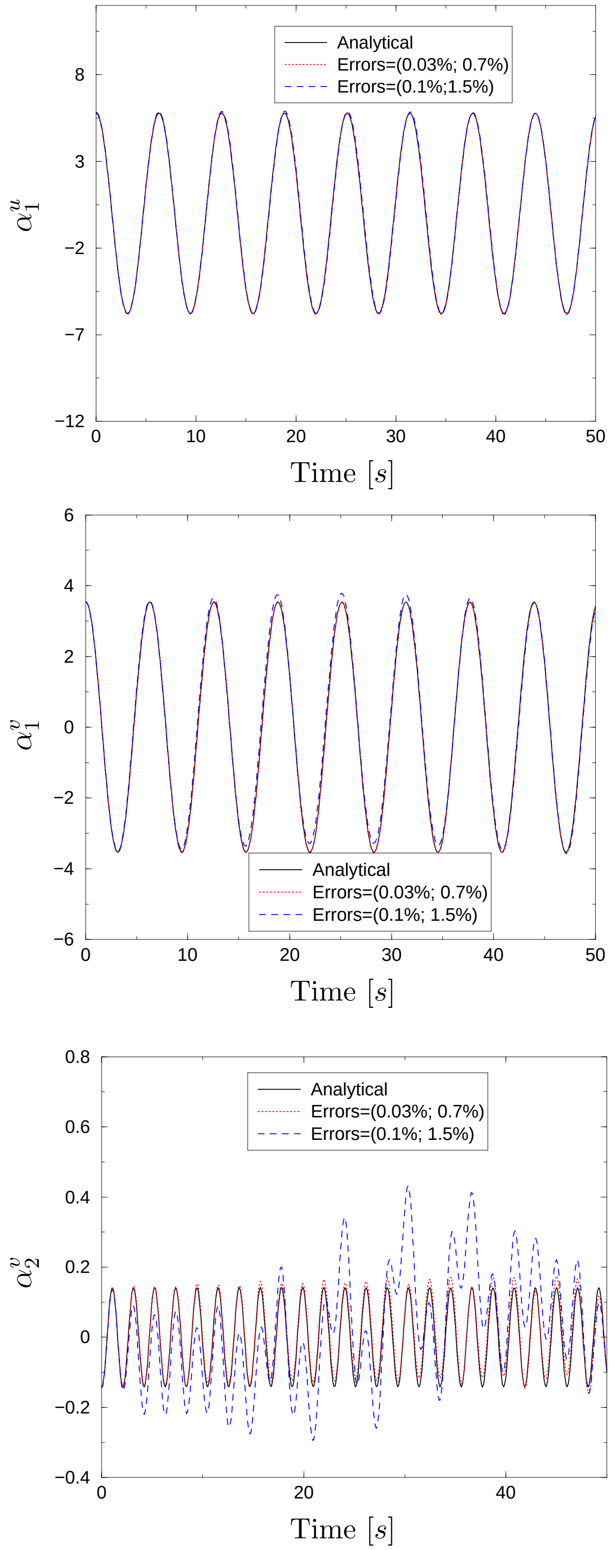


Fig. 12. Time coefficients for the double Burgers' equations

Table XIX. Errors of time coefficients obtained from the POD-based ROM for "double" Burgers' equations

Basis functions used in the ROM	$\alpha^u_{1_{ROM}}$	$\alpha^v_{1_{ROM}}$	$\alpha^v_{2_{ROM}}$
$\phi_{i_{analytical}}$	0.0028%	0.0043%	0.2258%
ϕ_{i_A}	0.0337%	0.0228%	0.2125%
$\phi^{(0.03;0.7)}_{i_N}$	0.0344%	0.0412%	0.6746%
$\phi^{(0.1;1.5)}_{i_N}$	0.0928%	0.2185%	4.6546%

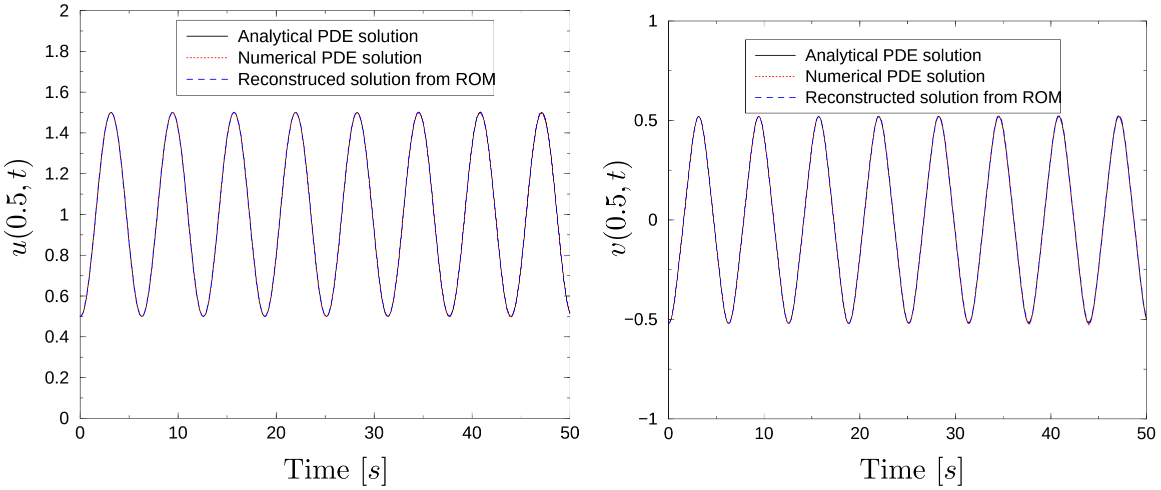
and $\alpha^v_{2_{analytical}}$ are observed. Thus the time coefficients which correspond to smaller eigenvalues are more sensitive to the errors of the basis functions.

c. Accuracy of reconstructed dependent variables

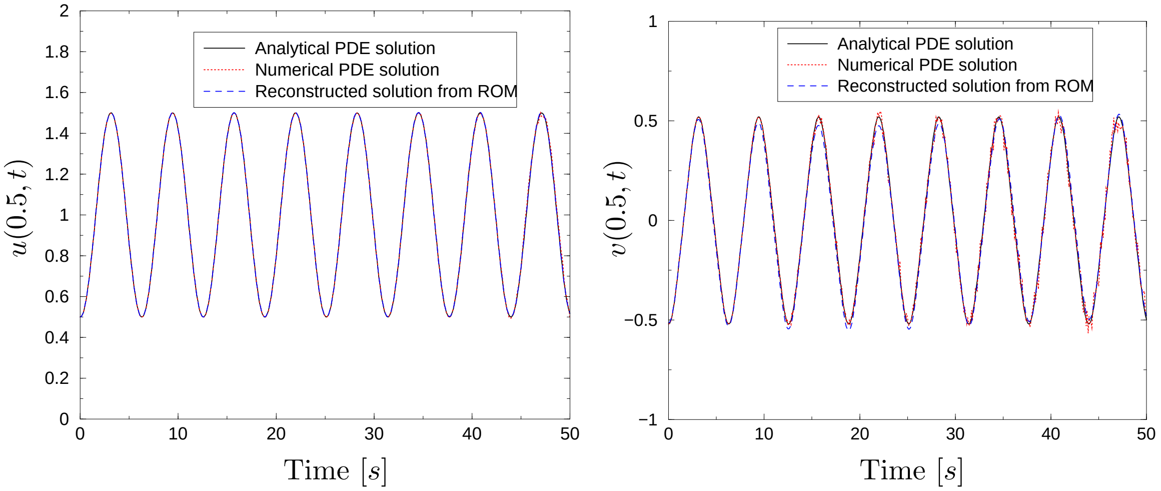
Let us reconstruct $u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ and $v^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ using $\alpha^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_{N,ROM}}$ and $\phi^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{i_N}$. Table XX lists the errors of $u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ and $v^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ with respect to the analytical solutions. Herein the error is defined by equation (6.16). Figure 13 shows $u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ and $v^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ compared against the analytical and numerical solutions of the PDEs (6.17) and (6.18). At both error levels, $u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ agree very well with the analytical solutions. $v^{(0.03;0.7)}_{N,ROM}(0.5,t)$ agree very well with the analytical solutions. Small differences are noticed between $v^{(0.1;1.5)}_{N,ROM}(0.5,t)$ and $v_{analytical}(0.5,t)$.

Table XX. Errors of reconstructed $u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ and $v^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$

$\bar{\epsilon}_u$	$\bar{\epsilon}_v$	$u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$	$v^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$
0.03%	0.7%	0.0052%	0.0445%
0.1%	1.5%	0.0279%	0.2626%



(a) $\bar{\epsilon}_u = 0.03\%$ and $\bar{\epsilon}_v = 0.7\%$



(b) $\bar{\epsilon}_u = 0.1\%$ and $\bar{\epsilon}_v = 1.5\%$

Fig. 13. $u^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ and $v^{(\bar{\epsilon}_u;\bar{\epsilon}_v)}_{N,ROM}(0.5,t)$ compared against analytical and numerical solutions of the PDEs

C. Summary

In this chapter, the POD-based ROMs have been constructed for two cases based on the Burgers' equation. The first case was an non-homogeneous Burgers' equation and the second case consists of two coupled non-homogeneous Burgers' equations. For both cases, the accuracy of the basis functions, the time coefficients obtained from the POD-based ROMs and the reconstructed dependent variables were analyzed. For the POD basis functions, the errors consist of two components: the errors caused by the PODDEC package (*i.e.*, the eigenvalue solver) and the errors of the databases. The errors caused by the PODDEC package are negligible. As the errors of the databases increase, the errors of the basis functions increase. The basis functions corresponding to smaller eigenvalues are more sensitive to the errors of the databases. For the time coefficients, the errors consist of four components: the errors of PODDEC package, the errors of the database, the errors of solving the POD-based ROMs, the errors of neglecting the POD modes corresponding to smaller eigenvalues. The errors caused by the ROMs solver, the PODDEC package and neglecting the POD modes corresponding to smaller eigenvalues are negligible. As the errors of the databases increase, the errors of the time coefficients increases. The time coefficients which correspond to smaller eigenvalues are more sensitive to the errors of the basis functions. For the reconstructed dependent variables, the errors are caused by all the sources that cause the errors of the basis functions and the time coefficients. Compared to the basis functions and the time coefficients, the errors of the reconstructed dependent variables are less sensitive to the errors of the databases.

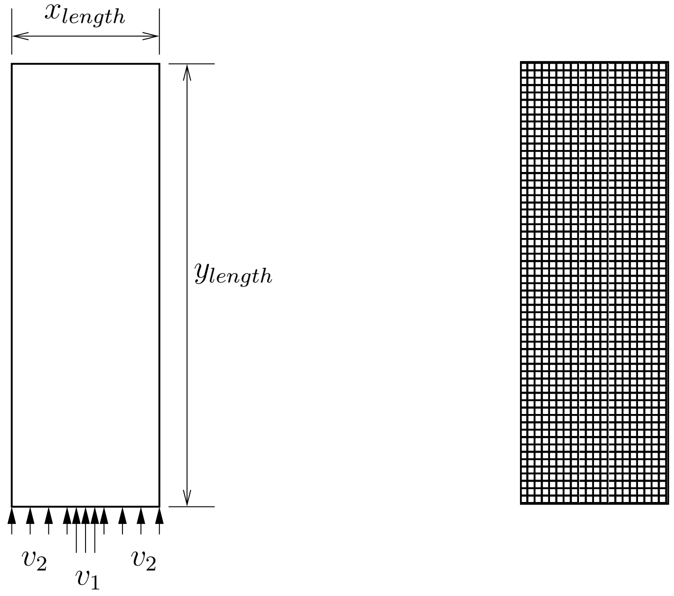
CHAPTER VII

RESULTS

This chapter presents the results of the POD-based ROMs applied to the transport phenomena in fluidized beds. Two cases are used to investigate the performance of these POD-based ROMs. The first case is a compressible gas-only flow problem. The second case consists of a gas phase and a solid phase.

A. Case I: compressible gas-only flow

The flow in case I is a compressible gas-only flow. The geometry and boundary conditions of case I are shown in Figure 14(a). Figure 14(b) shows the uniform computational grid used in case I. The parameters of case I are listed in Table XXI.



(a) Geometry and boundary conditions

(b) Computational grid

Fig. 14. Case I: geometry, boundary conditions, and computational grid

Note that MFIX uses CGS units and dimensional variables.

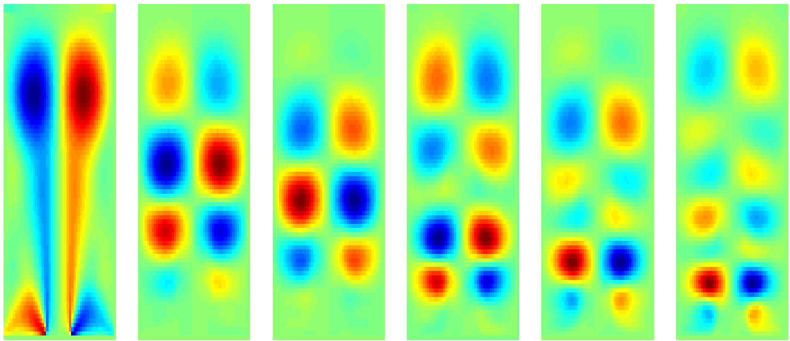
Table XXI. Parameters of case I

Parameter	Description	Value
x_{length}	Length of the domain in x-direction	$25.4cm$
y_{length}	Length of the domain in y-direction	$76.5cm$
$imax$	Number of cells in x-direction	50
$jmax$	Number of cells in y-direction	76
v_1, v_2	Gas inflow velocities	$12.6cm/s$, $1.0cm/s$
p_s	Static pressure at outlet	$1.01 \times 10^6 g/(cm \cdot s^2)$
T_0	Gas temperature	$297K$
μ_0	Gas viscosity	$1.8 \times 10^{-4} g/(cm \cdot s)$
t_{start}	Start time	$0s$
t_{stop}	Stop time	$13s$

The flow in case I was simulated from $t = 0s$ to $t = 13s$ using MFIX. Snapshots were stored every $0.05s$. Thus 260 snapshots were stored in the database generated by MFIX. POD basis functions of u, v, and p were calculated using the PODDEC package. Figures 15, 16, and 17 show the first six basis functions of u, v, and p, respectively. Figure 18 shows the cumulative energy retained by different number of POD modes.

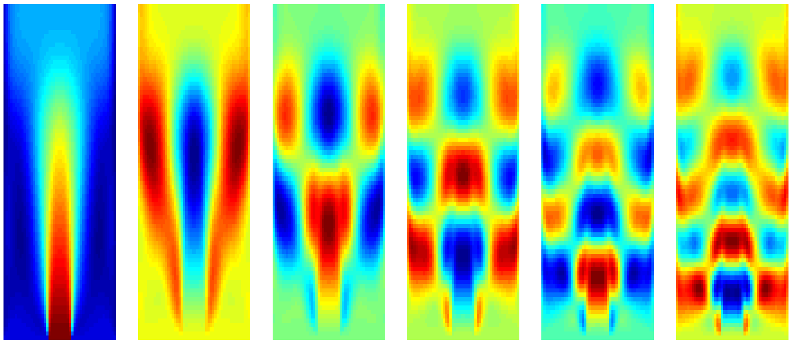
Results of POD-based ROMs at the reference condition

Let us apply ODExMFIX to the flow in case I at the reference condition. Figures 19, 20, and 21 show the first four time coefficients of u, v, and p obtained from ODExMFIX using different number of POD modes. In these figures, the reference



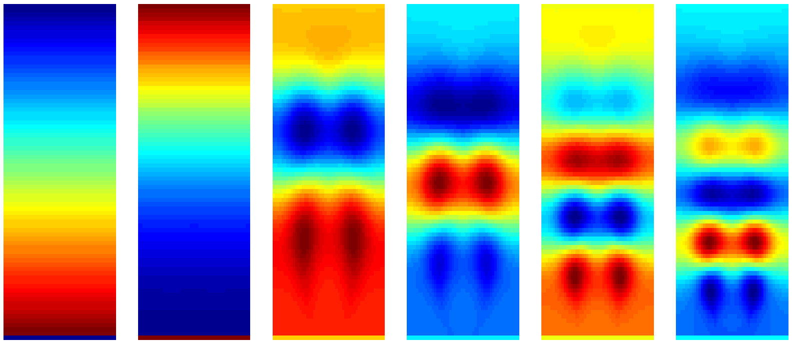
ϕ_0^u ϕ_1^u ϕ_2^u ϕ_3^u ϕ_4^u ϕ_5^u

Fig. 15. Case I: first six basis functions of u



ϕ_0^v ϕ_1^v ϕ_2^v ϕ_3^v ϕ_4^v ϕ_5^v

Fig. 16. Case I: first six basis functions of v



ϕ_0^p ϕ_1^p ϕ_2^p ϕ_3^p ϕ_4^p ϕ_5^p

Fig. 17. Case I: first six basis functions of p

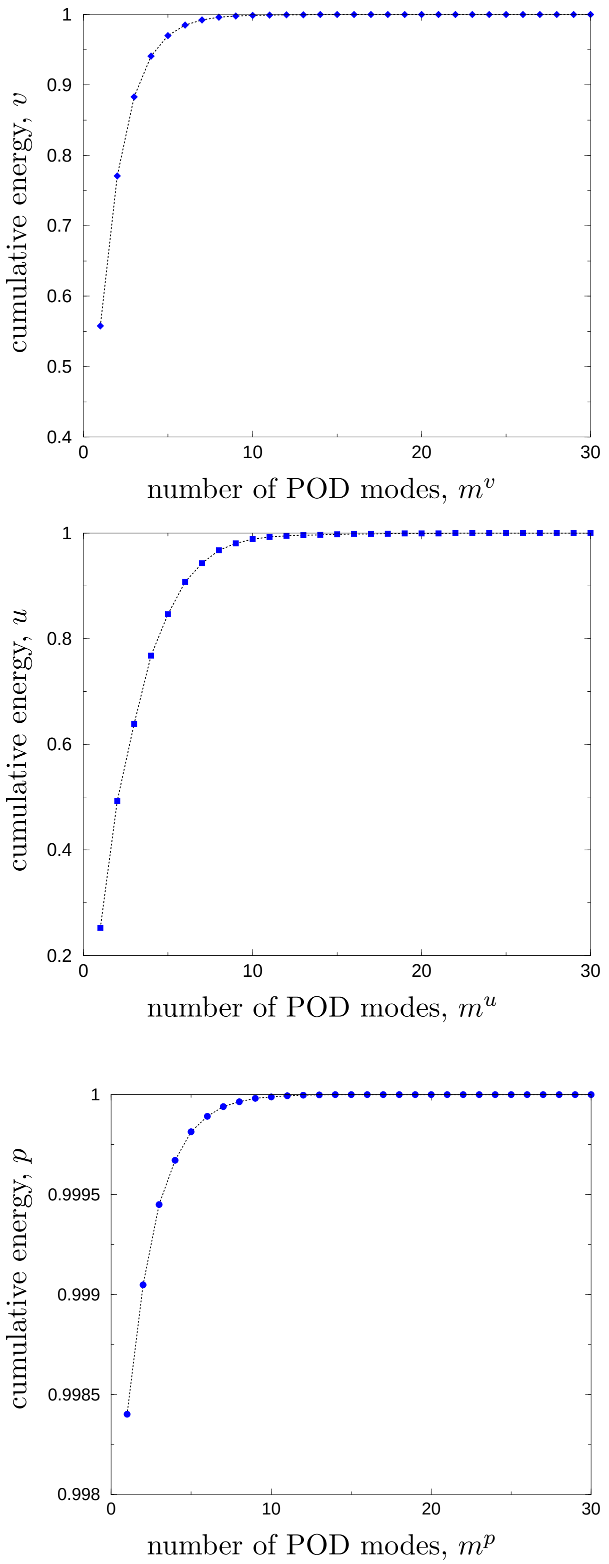


Fig. 18. Case I: cumulative energy

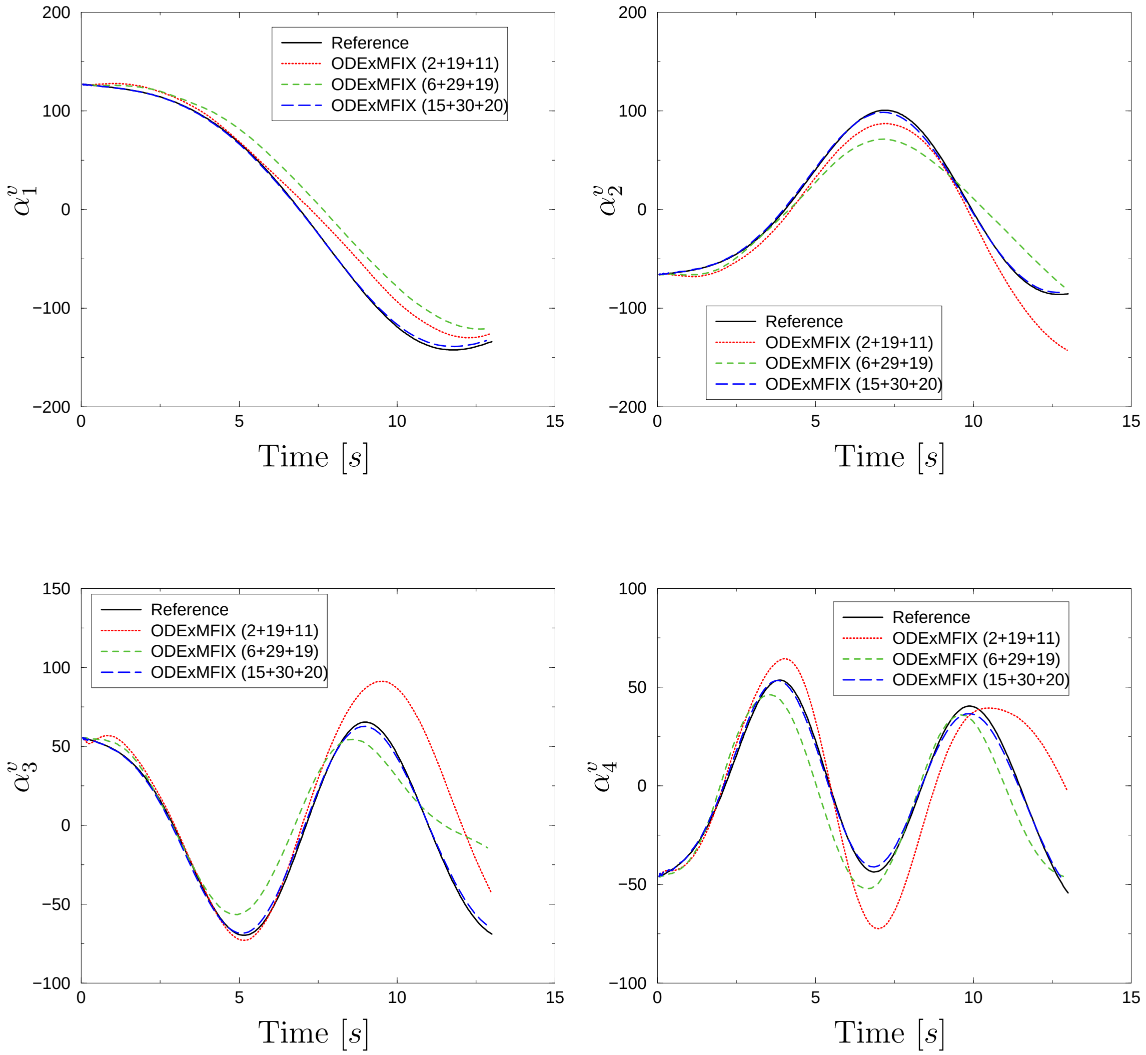


Fig. 19. Case I: the first four time coefficients of v

time coefficients were obtained by directly projecting the snapshots generated by MFIX onto the POD basis functions. ODExMFIX(2+19+11) denotes the model of ODExMFIX with $m^p = 2$, $m^u = 19$, and $m^v = 11$.

As shown in Figures 19 and 20, increasing the number of POD modes used by ODExMFIX significantly improved the accuracy of the time coefficients of u and v predicted by ODExMFIX. The time coefficients of u and v predicted by ODExMFIX(15+30+20) agreed very well with the reference time coefficients. By using these

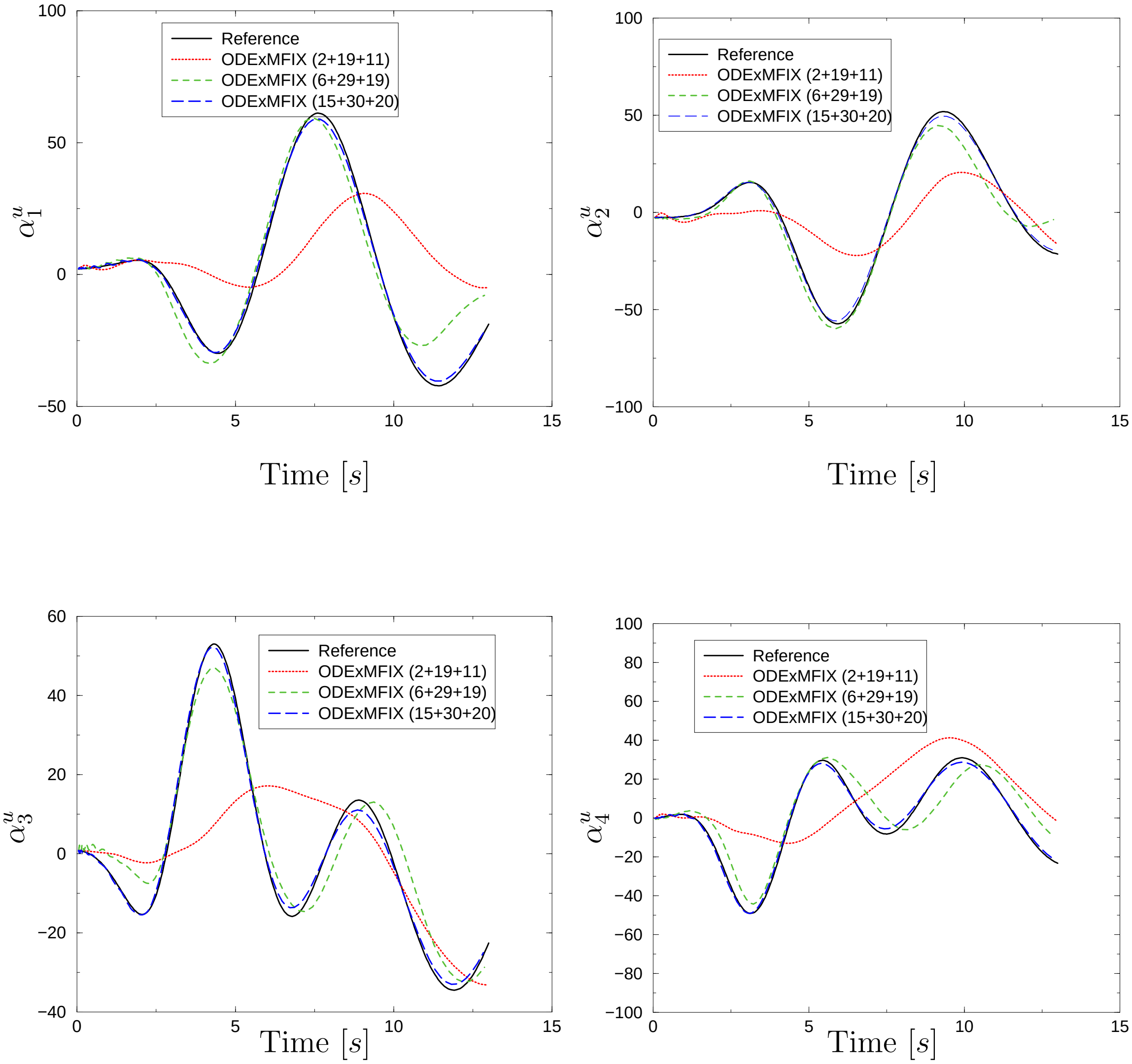


Fig. 20. Case I: the first four time coefficients of u

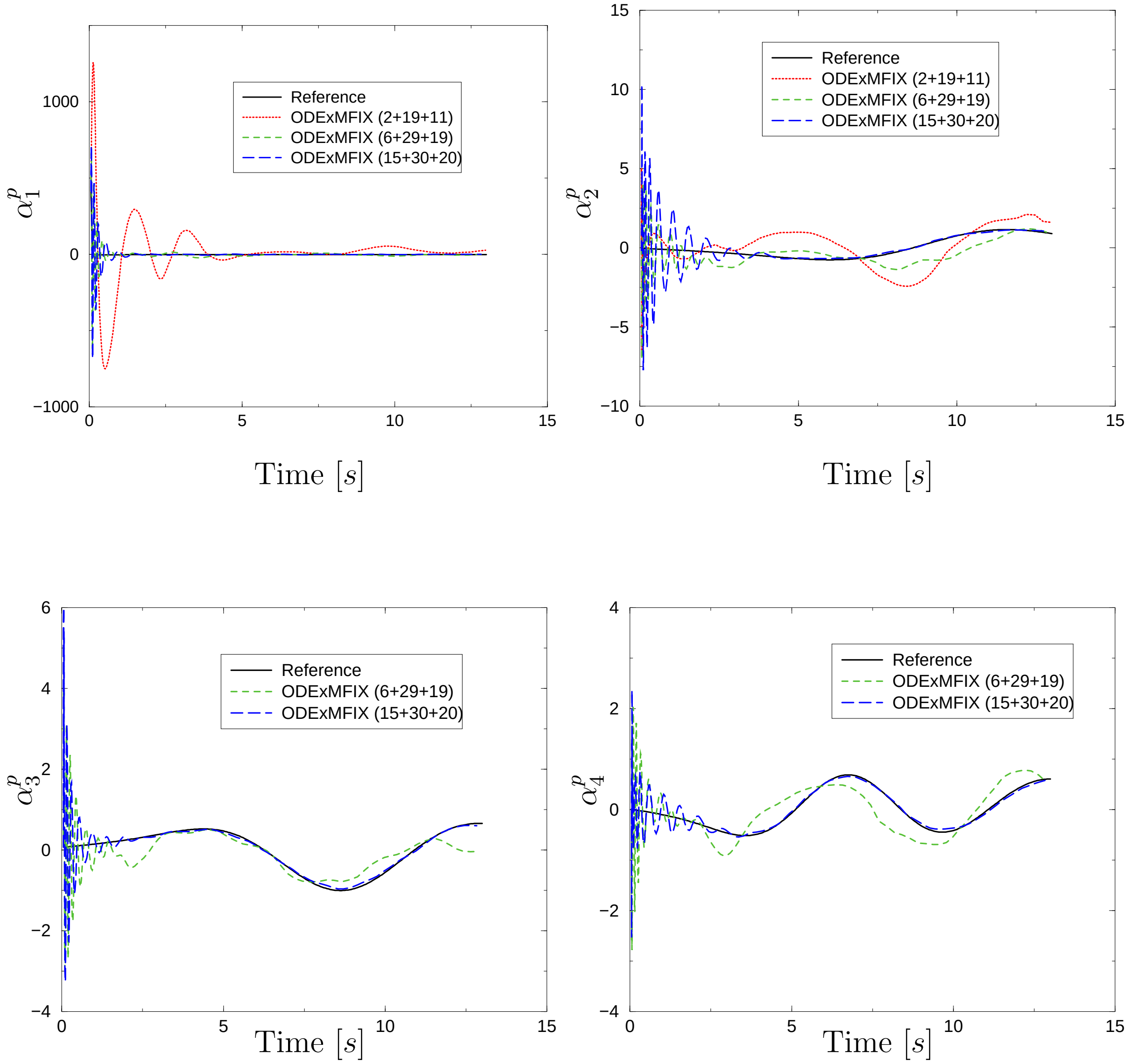


Fig. 21. Case I: the first four time coefficients of p

number of POD modes, about 99.99% of the total energy was retained by the POD modes used in ODExMFIX.

Figure 21 demonstrates that the time coefficients of p are more difficult to be captured by ODExMFIX. At the beginning period ($t \in [0s, 3s]$), ODExMFIX produced large oscillations in α^p. When $t > 3s$, the time coefficients obtained from ODExMFIX(15+30+20) agreed well with the reference time coefficients.

One possible reason that caused the large wiggles in α^p predicted by ODExMFIX is that the number of snapshots in the database is not enough. To prove that reason, we stored the snapshots every $0.01s$ instead of every $0.05s$. Thus the number of snapshots increased from 260 to 1195. Figure 22 shows α_1^p predicted by ODExMFIX(15+30+20) using these two sets of snapshots. Figure 22 shows that increasing the number of snapshots in the database did not improve the prediction of α_1^p using ODExMFIX.

The average magnitude of ϕ_0^p is 1010043.89 and the average magnitude of ϕ_1^p is 0.0143. The average magnitude of ϕ_1^p is 0.0000014% of the average magnitude of ϕ_0^p. Thus most of the spatial characteristics of the pressure field are captured by ϕ_0^p. Because of the large magnitude difference between ϕ_0^p and ϕ_1^p, ODExMFIX could not provide predictions of ϕ_i^p which agreed very well with the reference time coefficients of p.

Figures 23, 24, and 25 show the reconstructed flow fields compared against the results of MFIX at $t = 1.25s$, $t = 7.00s$, and $t = 13.00s$. As shown in Figures 23 and 24, the accuracy of the reconstructed u and v using time coefficients obtained from ODExMFIX increased as the number of POD modes used in ODExMFIX increased. For p, since almost all the spatial characteristics of the pressure field are captured by ϕ_0^p, all the reconstructed pressure fields are close to ϕ_0^p.

The computational cost for simulating the flow in case I using MFIX was 13187

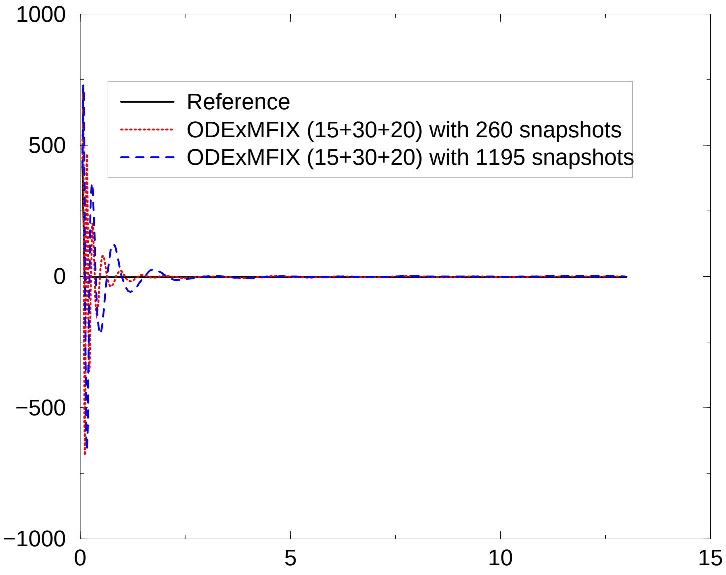
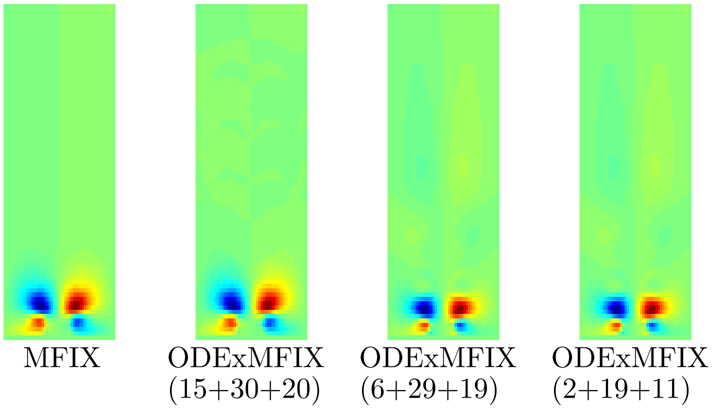
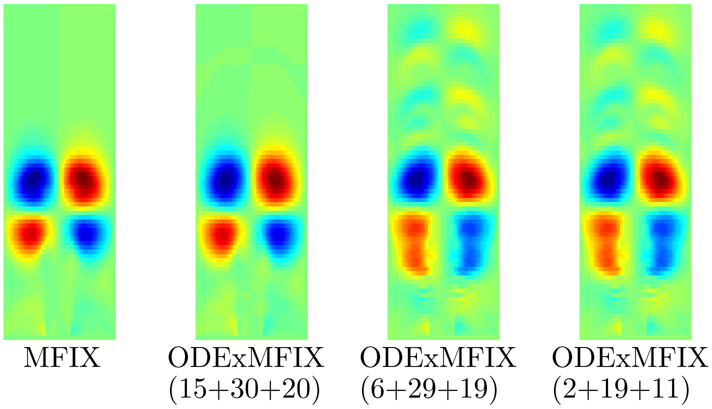


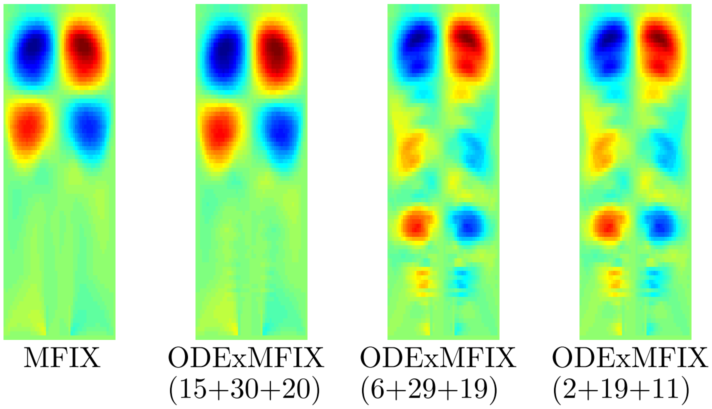
Fig. 22. Case I: α_1^p predicted by ODExMFIX using two sets of snapshots



(a) $t = 1.25s$

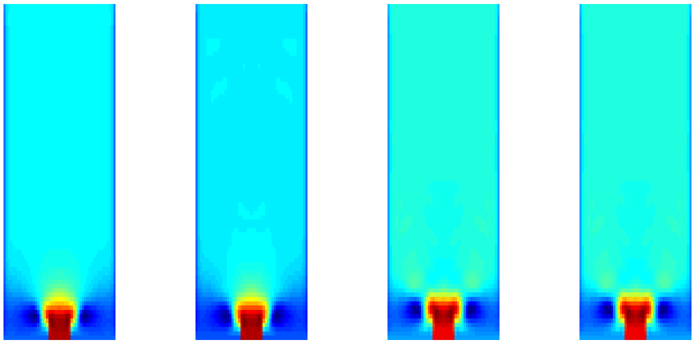


(b) $t = 7.00$



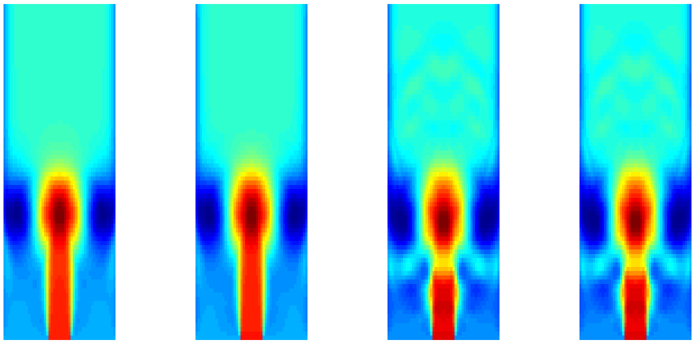
(c) $t = 13.00$

Fig. 23. Case I: comparison of u between MFIX and ODExMFIX



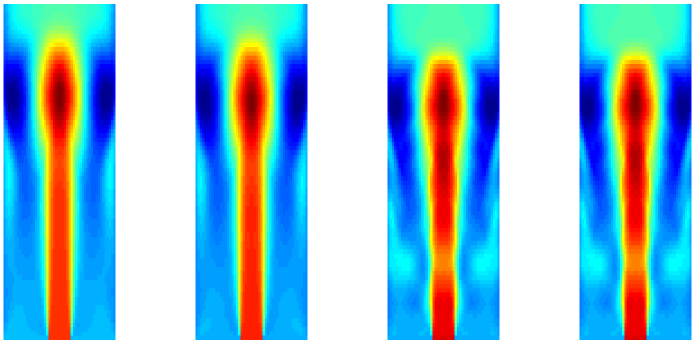
MFIX ODExMFIX (15+30+20) ODExMFIX (6+29+19) ODExMFIX (2+19+11)

(a) $t = 1.25s$



MFIX ODExMFIX (15+30+20) ODExMFIX (6+29+19) ODExMFIX (2+19+11)

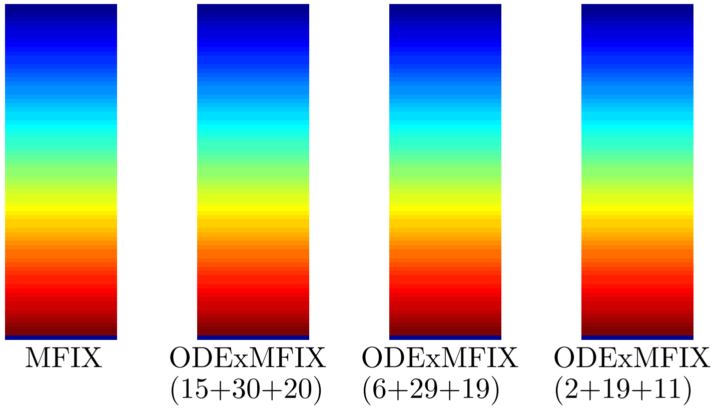
(b) $t = 7.00$



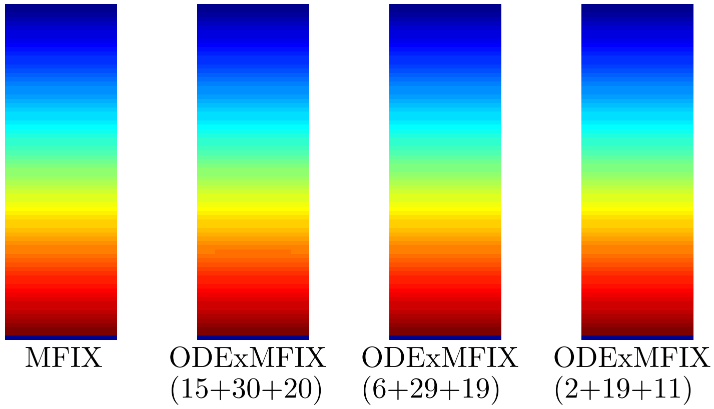
MFIX ODExMFIX (15+30+20) ODExMFIX (6+29+19) ODExMFIX (2+19+11)

(c) $t = 13.00$

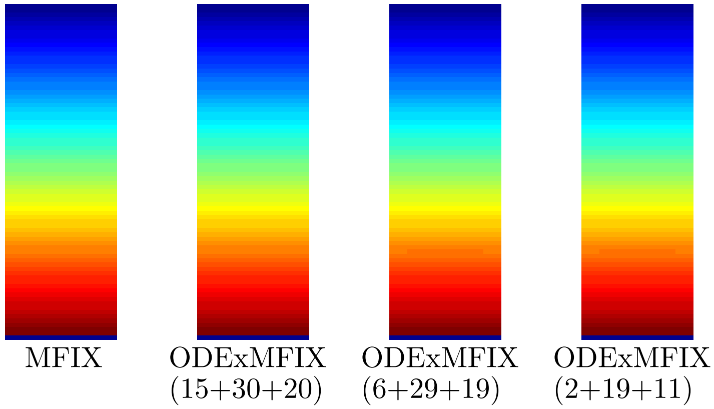
Fig. 24. Case I: comparison of v between MFIX and ODExMFIX



(a) $t = 1.25s$



(b) $t = 7.00$



(c) $t = 13.00$

Fig. 25. Case I: comparison of p between MFIX and ODExMFIX

seconds of CPU time. The computational cost of ODExMFIX(15+30+20) was 3352 seconds of CPU time. The computational cost of ODExMFIX(15+30+20) was 25.4% of the cost of MFIX. Computational cost reduction has been achieved by using ODExMFIX.

ODEx3 has also been applied to case I. ODEx3, however, failed to produce a converged result. In ODEx3, the unknowns are α^{ρ}, α^{u}, and α^{v}. The magnitudes of α_1^{ρ}, α_1^{u} and α_1^{v} are of the order of 10^{-10}, 10, and 100, respectively. Compared to u and v, the time coefficients of ρ are too small to be properly captured.

Results of POD-based ROMs at the off-reference conditions

Now, let us apply ODExMFIX at some off-reference conditions of case I. Herein we changed the gas viscosity μ_0 and used several different values of μ_0. ODExMFIX was generated using the POD basis functions obtained at $\mu_0 = 1.8\times10^{-4} g/(cm\cdot s)$. In order to measure the accuracy of ODExMFIX applied at the off-reference conditions, let us define an error ε_{var} as

$$\varepsilon_{var} = \frac{\sqrt{\sum_{i=1}^{N}(var_i^{MFIX} - var_i^{ROM})^2}}{\sum_{i=1}^{N}|var_i^{MFIX}|}, \tag{7.1}$$

where var represents the field variables p, u, or v. N is the total number of spatial grid points.

Table XXII lists the errors of ε_p, ε_u, and ε_v at eight different values of μ_0 including the reference value. In Table XXII, all the errors were calculated at $t = 13s$. As seen from Table XXII, the errors of pressure, ε_p, for all the eight conditions are very small, because the speed of the gas flow in case I is very low and the pressure variations are very small. Figure 26 shows the errors of ε_u and ε_v at different values of μ_0. As seen from Figure 26, at each condition, ε_v is always smaller than ε_u because the flow in case I mainly concentrates in $y-$direction and it is easier for ODExMFIX to capture

Table XXII. Case I: errors of ODExMFIX at different values of μ_0

μ_0 $[g/(cm \cdot s)]$	ε_p [%]	ε_u [%]	ε_v [%]
0.8×10^{-4}	3.60×10^{-8}	1.907	0.449
1.0×10^{-4}	3.72×10^{-8}	0.965	0.258
1.1×10^{-4}	3.82×10^{-8}	1.05	0.263
1.3×10^{-4}	2.04×10^{-8}	0.721	0.163
1.5×10^{-4}	5.99×10^{-9}	0.307	0.075
1.6×10^{-4}	7.71×10^{-9}	0.253	0.062
1.8×10^{-4}	1.10×10^{-8}	0.201	0.055
2.0×10^{-4}	8.36×10^{-9}	0.220	0.072
2.2×10^{-4}	1.21×10^{-8}	0.232	0.090
2.4×10^{-4}	2.21×10^{-8}	0.230	0.107
2.6×10^{-4}	1.54×10^{-8}	0.285	0.132
2.8×10^{-4}	3.32×10^{-8}	0.320	0.148
3.0×10^{-4}	3.07×10^{-8}	0.376	0.166

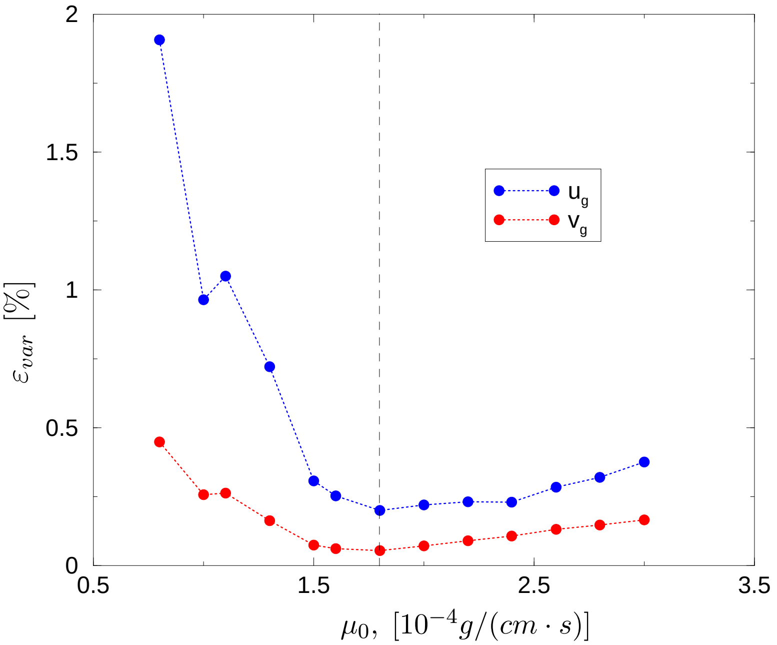
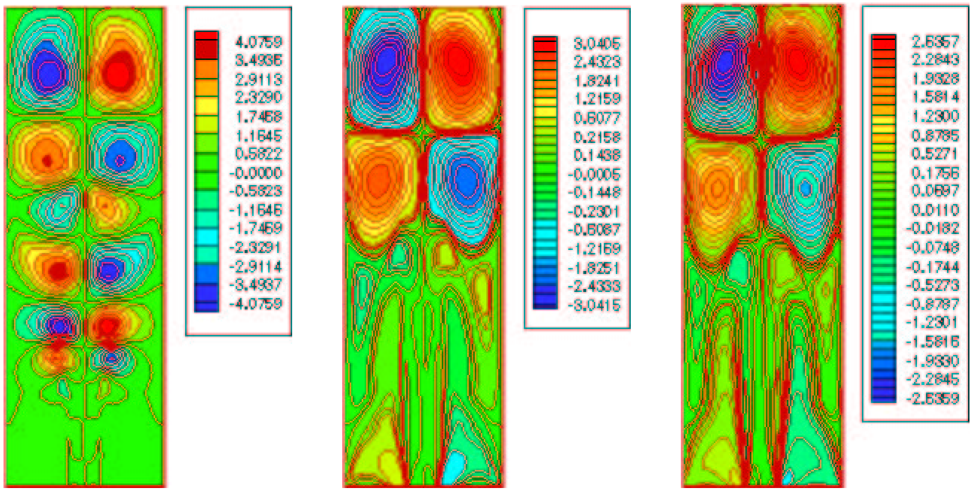


Fig. 26. Case I: ε_u and ε_v at different values of μ_0



(a) $\mu_0 = 0.8 \times 10^{-4} g/(cm \cdot g)$

(b) $\mu_0 = 1.8 \times 10^{-4} g/(cm \cdot g)$

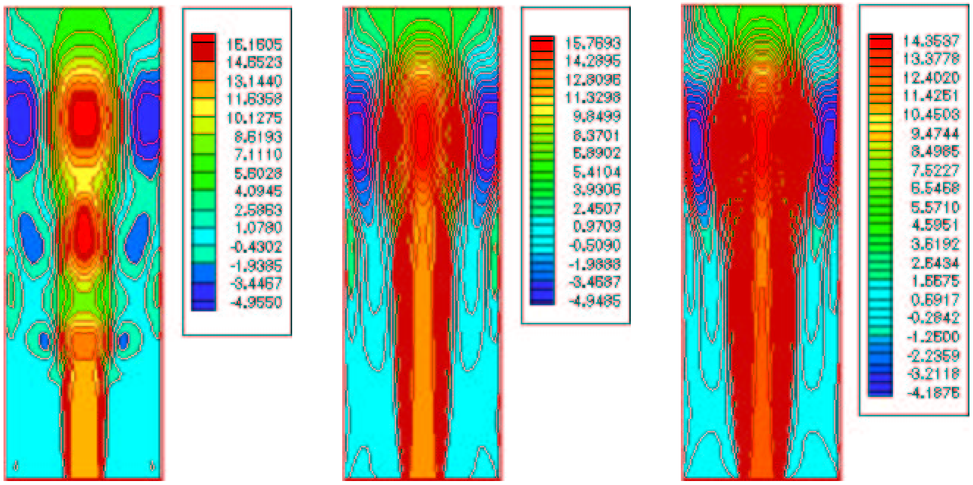
(c) $\mu_0 = 3.0 \times 10^{-4} g/(cm \cdot g)$

Fig. 27. Case I: field of u_g at different conditions

the details of v_g than the details of u_g. At the reference μ_0, both ε_v and ε_u have the minimum values. As μ_0 increased or decreased from the reference value, both ε_v and ε_u increased. ε_v and ε_u increased more rapidly when μ_0 decreased from the reference μ_0 than when μ_0 increased from the reference μ_0, because smaller μ_0 caused larger Reynolds number and more complicated flow fields. Figures 27 and 28 compare the flow fields at $\mu_0 = 0.8 \times 10^{-4} g/(cm \cdot g)$, $1.8 \times 10^{-4} g/(cm \cdot g)$, and $3.0 \times 10^{-4} g/(cm \cdot g)$.

B. Case II: gas-solid transport phenomena in a fluidized bed

Case II models the gas-solid transport phenomena in a fluidized bed. The geometry and boundary conditions of case II are shown in Figure 29(a). Figure 29(b) illustrates



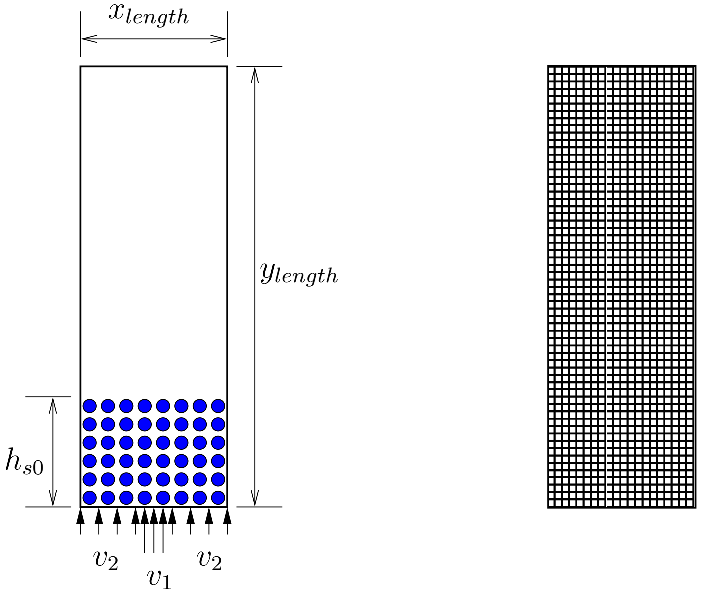
(a) $\mu_0 = 0.8 \times 10^{-4} g/(cm \cdot g)$

(b) $\mu_0 = 1.8 \times 10^{-4} g/(cm \cdot g)$

(c) $\mu_0 = 3.0 \times 10^{-4} g/(cm \cdot g)$

Fig. 28. Case I: field of v_g at different conditions

the uniform computational grid used in case II. Table XXIII lists the parameters of



(a) Geometry and boundary conditions

(b) Computational grid

Fig. 29. Case II: geometry, boundary conditions, and computational grid

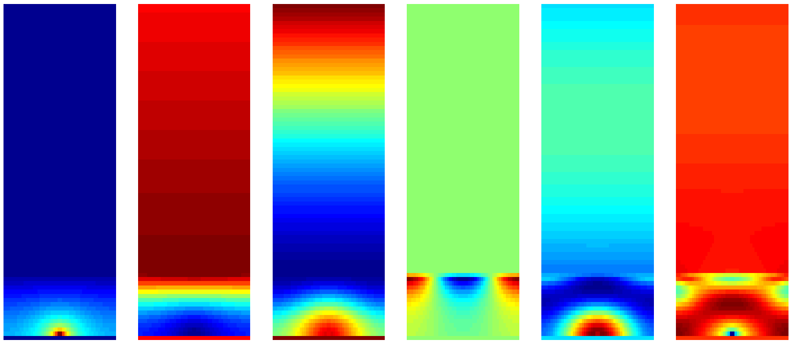
case II. In Table XXIII, ρ_{so}, D_p, h_{s0}, and ϵ_g^* denote the constant solid density, solid particle diameter, initial packed bed height, and packed bed void fraction, respectively.

MFIX simulated the transport phenomena in case II from $t = 0s$ to $t = 1s$. From $t = 0s$ to $t = 0.2s$, $v_1 = v_2 = 1cm/s$. At $t = 0.2s$, v_2 was increased to $120.0cm/s$. From $t = 0.2s$ to $t = 1.0s$, snapshots were stored every $0.0025s$, thus 320 snapshots were stored in the database. POD was applied to the database to calculate the POD basis functions. Figures 30-34 show the first six POD basis functions of p_g, u_g, v_g, u_s, and v_s, respectively. Table XXIV lists the number of POD modes needed to retain certain portions of the total energy.

Results of ODExS at the reference condition

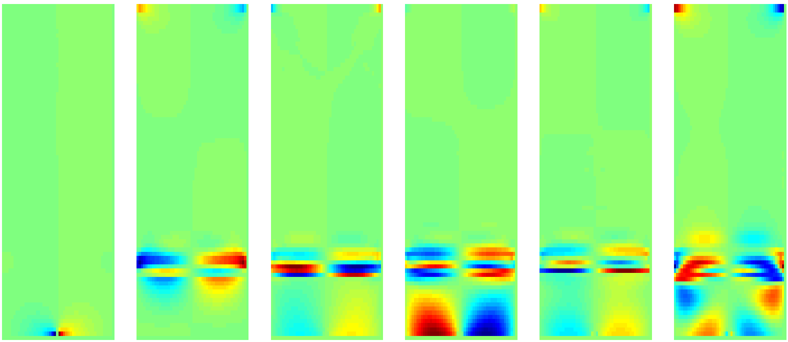
Table XXIII. Parameters of case II

Parameter	Value
x_{length}	$25.4cm$
y_{length}	$76.5cm$
$imax$	50
$jmax$	78
v_1	$120.0cm/s$
v_2	$1.0cm/s$
p_{g_s}	$1.01 \times 10^6 g/(cm \cdot s^2)$
T_{g0}	$297K$
μ_{g0}	$1.8 \times 10^{-4} g/(cm \cdot s)$
t_{start}	$0.2s$
t_{stop}	$1s$
Δt	$1 \times 10^{-4} s$
ρ_{so}	$1.0g/cm^3$
D_p	$0.05cm$
h_{s0}	$14.7cm$
ϵ_g^*	0.4



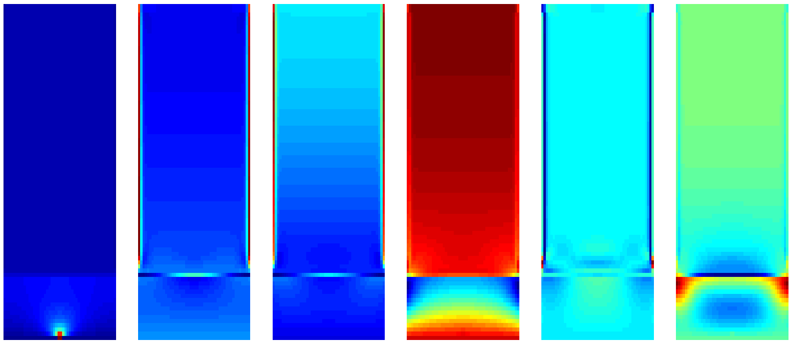
$\phi_0^{p_g}$ $\phi_1^{p_g}$ $\phi_2^{p_g}$ $\phi_3^{p_g}$ $\phi_4^{p_g}$ $\phi_5^{p_g}$

Fig. 30. Case II: first six basis functions of p_g



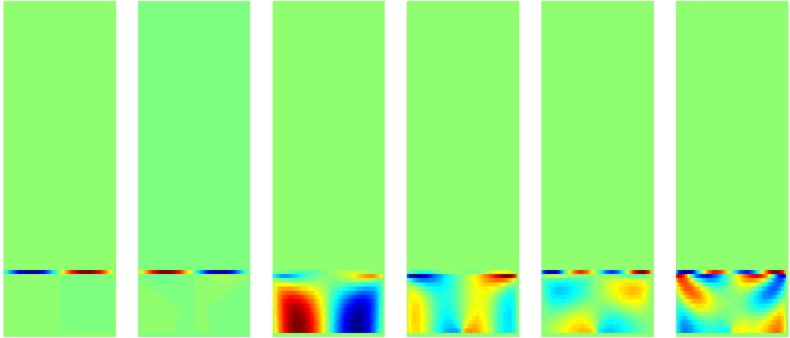
$\phi_0^{u_g}$ $\phi_1^{u_g}$ $\phi_2^{u_g}$ $\phi_3^{u_g}$ $\phi_4^{u_g}$ $\phi_5^{u_g}$

Fig. 31. Case II: first six basis functions of u_g



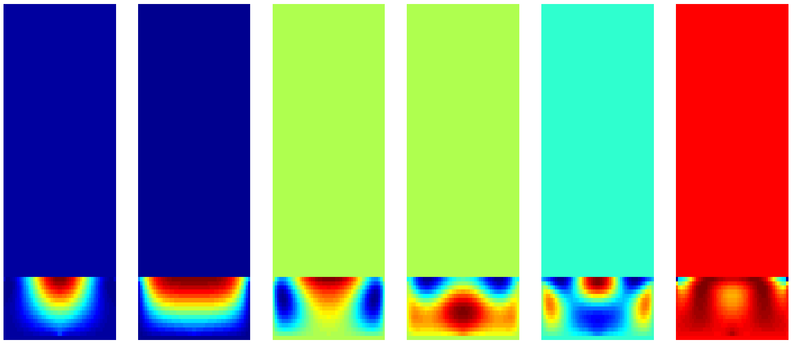
$\phi_0^{v_g}$ $\phi_1^{v_g}$ $\phi_2^{v_g}$ $\phi_3^{v_g}$ $\phi_4^{v_g}$ $\phi_5^{v_g}$

Fig. 32. Case II: first six basis functions of v_g



$\phi_0^{u_s}$ $\phi_1^{u_s}$ $\phi_2^{u_s}$ $\phi_3^{u_s}$ $\phi_4^{u_s}$ $\phi_5^{u_s}$

Fig. 33. Case II: first six basis functions of u_s



$\phi_0^{v_s}$ $\phi_1^{v_s}$ $\phi_2^{v_s}$ $\phi_3^{v_s}$ $\phi_4^{v_s}$ $\phi_5^{v_s}$

Fig. 34. Case II: first six basis functions of v_s

Table XXIV. Case II: POD energy vs number of modes for p_g, u_g, v_g, u_s, and v_s

POD	Number of modes				
energy	p_g	u_g	v_g	u_s	v_s
99%	2	4	3	4	2
99.9%	3	7	5	6	3
99.99%	4	11	7	7	5

Let us examine the results of ODExS applied to case II at the reference condition. Herein, we chose $m^{p_g} = 4$, $m^{u_g} = 11$, $m^{v_g} = 7$, $m^{u_s} = 7$, and $m^{v_s} = 5$ such that 99.99% of the total energy was retained by the POD modes used in ODExS. The computational cost of using MFIX to simulate the transport phenomena in case II was 23008 seconds of CPU time. The computational cost of using ODExS to simulate the transport phenomena in case II was 7595 seconds of CPU time. The cost of ODExS was 33.01% of the cost of MFIX.

Figures 35 and 36 show the first four time coefficients of u_g and v_g obtained from ODExS compared against the directly projecting results. Figures 37 and 38 show the first four time coefficients of u_s and v_s obtained from ODExS compared against the directly projecting results. Figures 39-44 show the field variables obtained from ODExS compared against the results of MFIX at $t = 1s$. Table XXV lists the errors ε_{var} of the field variables obtained from ODExS compared against the results of MFIX at $t = 1s$. The error ε_{var} was defined by Equation (7.1) and herein var represents ϵ_g, p_g, u_g, u_s, or v_s. As seen from Figures 39-44 and Table XXV, the results of ODExS agree very well with the results of MFIX at the reference condition.

Table XXV. Case II: errors of the results of ODExS at reference condition

Variable	Error, ε_{var} [%]
ϵ_g	0.0000356
p_g	0.000000578
u_g	0.01035
v_g	0.01026
u_s	0.505
v_s	0.765

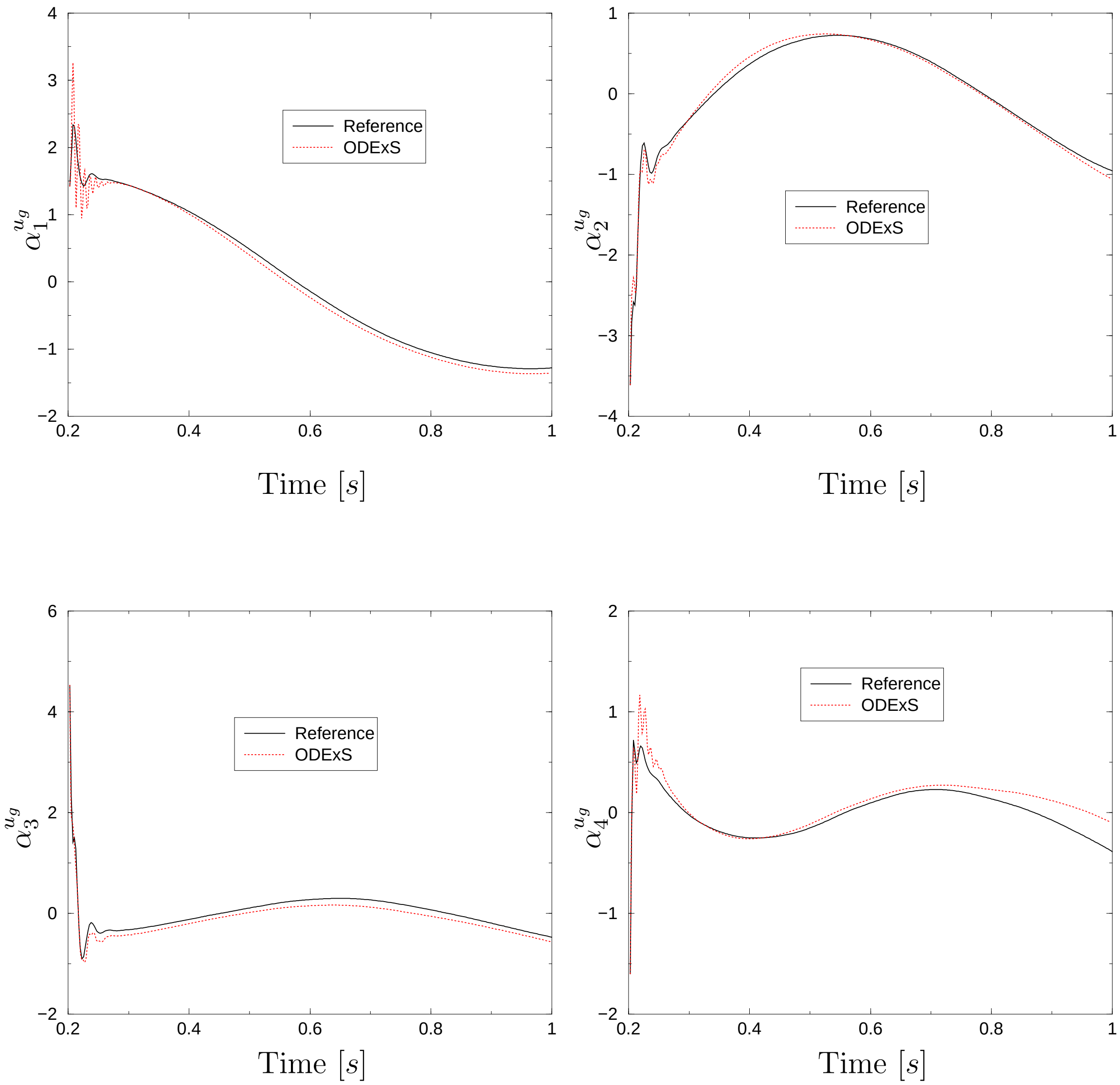


Fig. 35. Case II: first four time coefficients of u_g obtained from ODExS

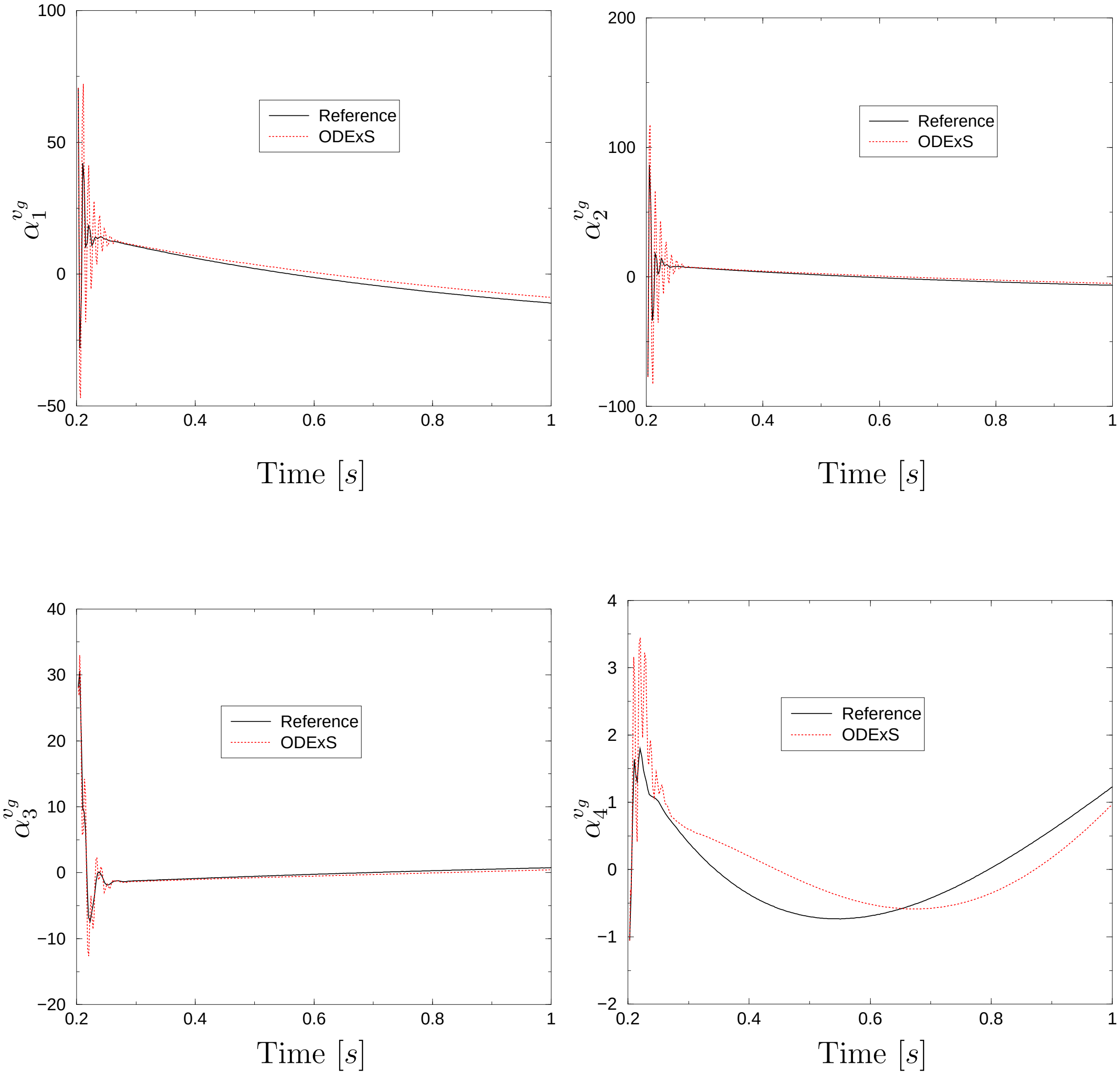


Fig. 36. Case II: first four time coefficients of v_g obtained from ODExS

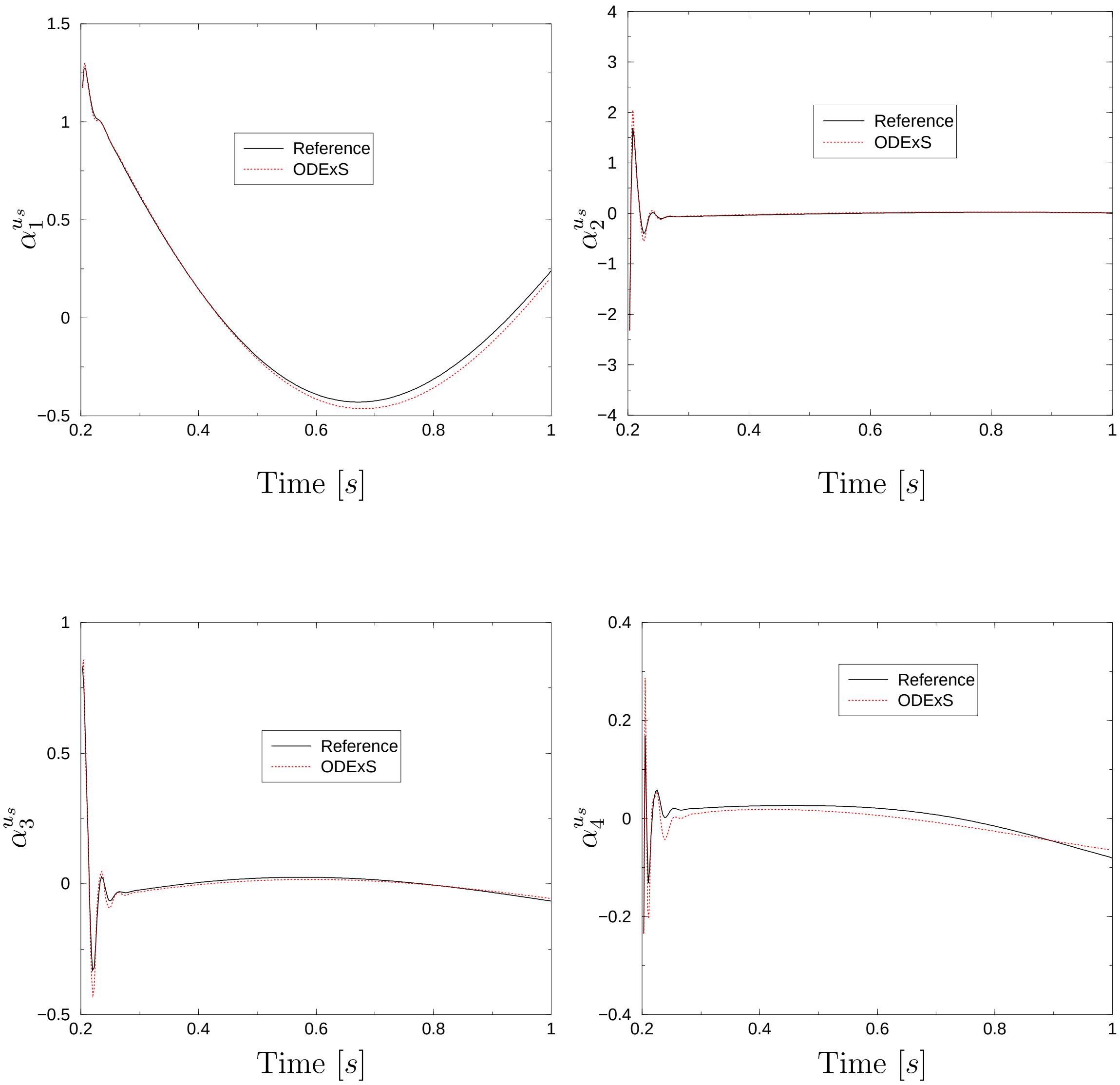


Fig. 37. Case II: first four time coefficients of u_s obtained from ODExS

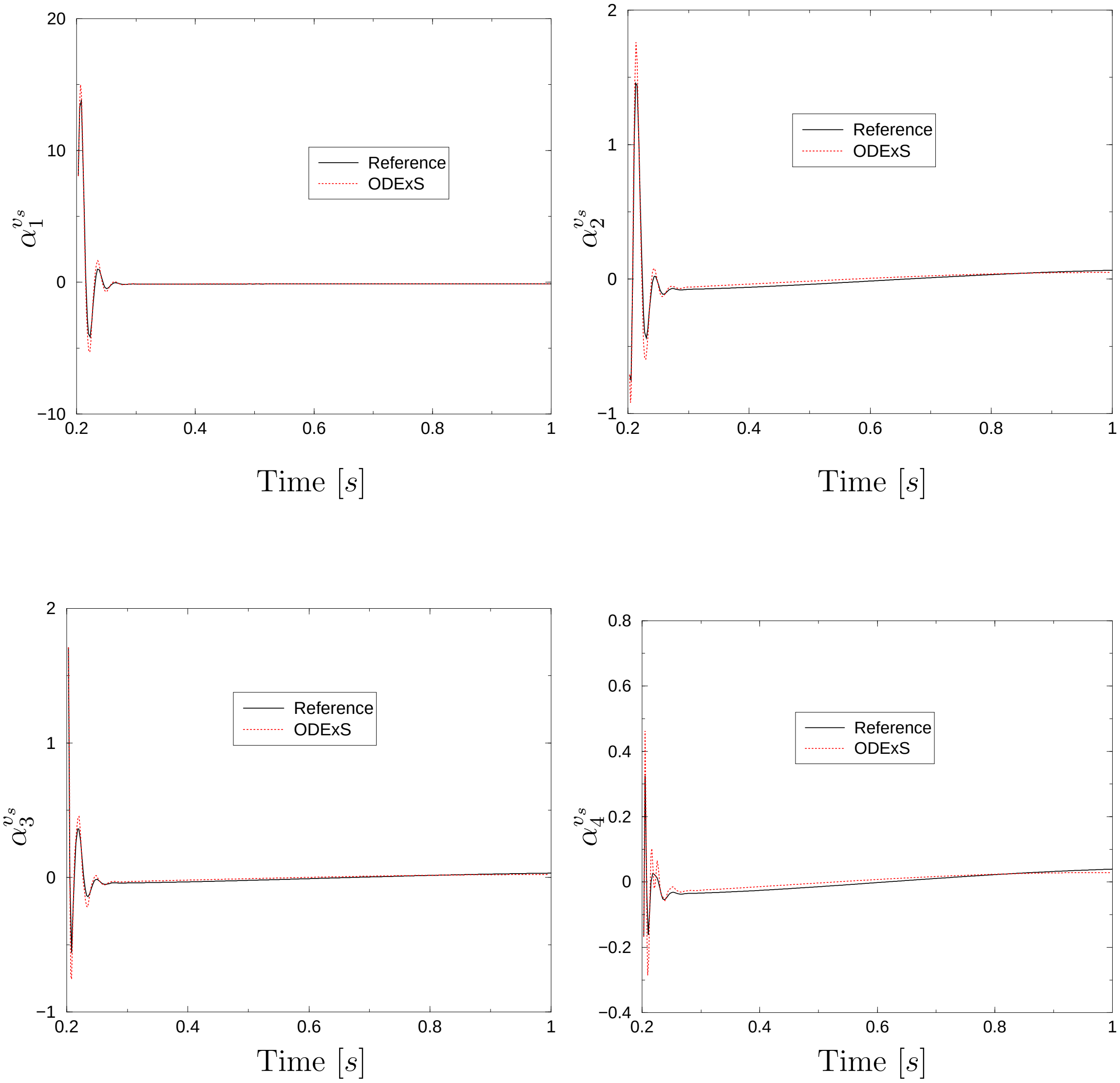
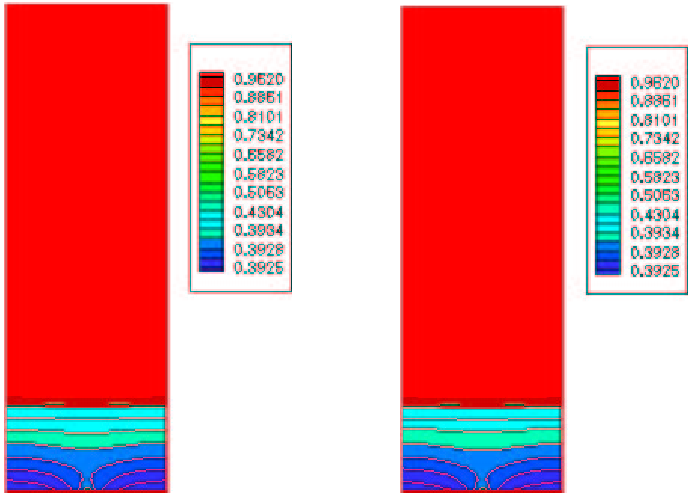
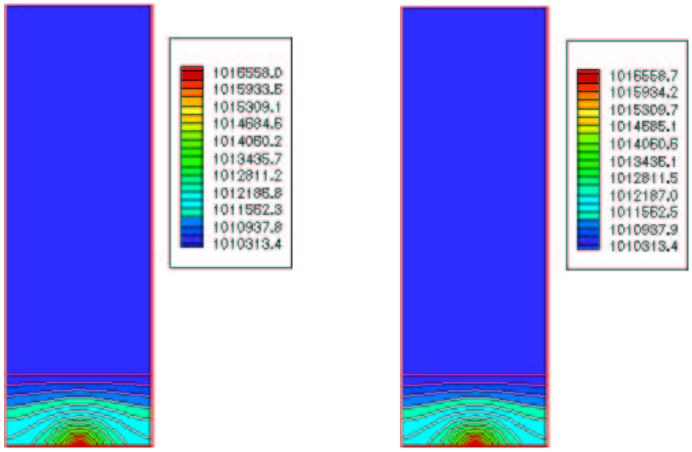


Fig. 38. Case II: first four time coefficients of v_s obtained from ODExS



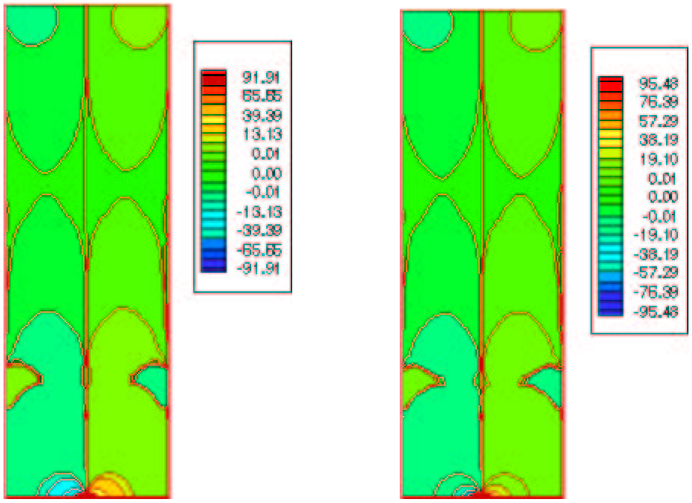
(a) MFIX (b) ODExS

Fig. 39. Case II: ϵ_g at $t = 1s$



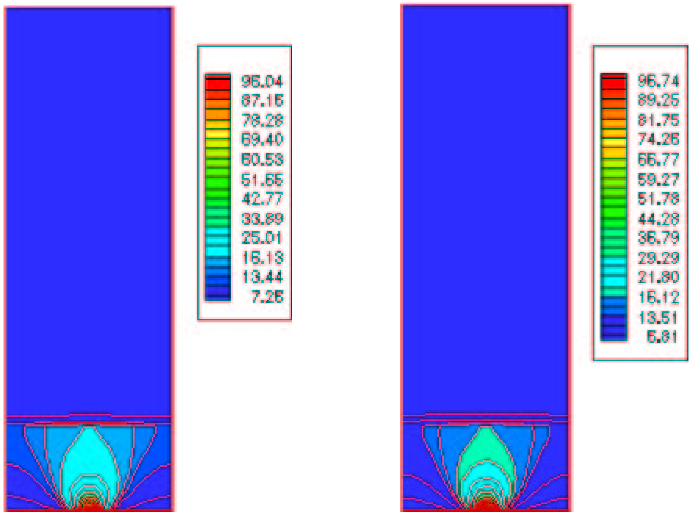
(a) MFIX (b) ODExS

Fig. 40. Case II: p_g at $t = 1s$



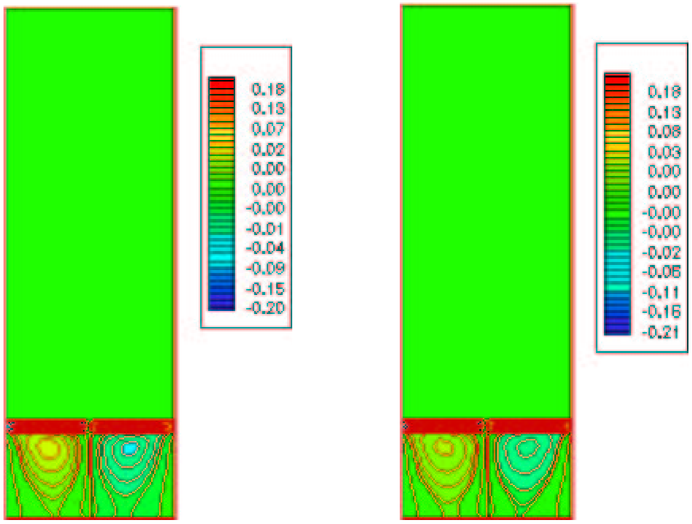
(a) MFIX (b) ODExS

Fig. 41. Case II: u_g at $t = 1s$



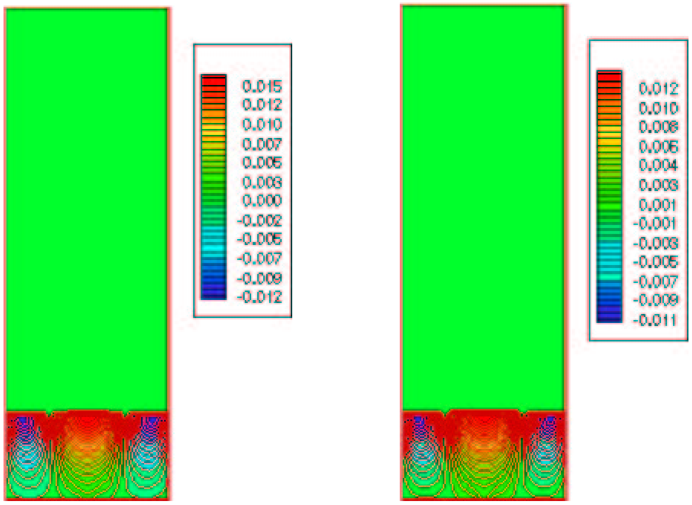
(a) MFIX (b) ODExS

Fig. 42. Case II: v_g at $t = 1s$



(a) MFIX (b) ODExS

Fig. 43. Case II: u_s at $t = 1s$



(a) MFIX (b) ODExS

Fig. 44. Case II: v_s at $t = 1s$

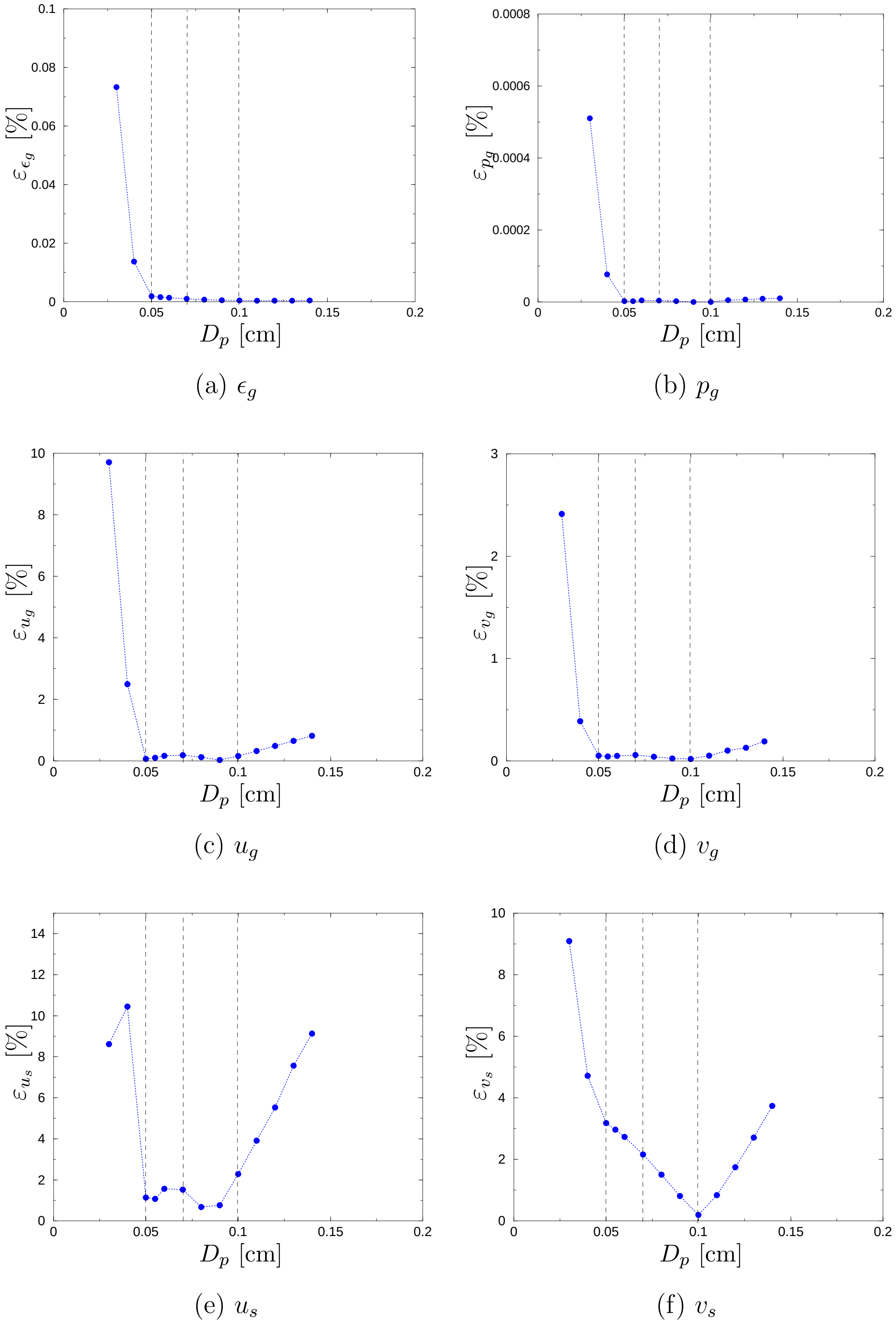
Results of ODExS at the off-reference conditions

When studying ODExS applied at the off-reference conditions, we generated the database by combining the numerical results of MFIX at three different values of solid particle diameters: $D_p = 0.05cm$, $D_p = 0.07cm$, and $D_p = 0.10cm$. At each condition, 320 snapshots were calculated using MFIX. The database was generated by combining the snapshots of these three conditions. Thus, there were 960 snapshots contained in the database. POD was applied to the database containing 960 snapshots to extract the POD basis functions. Using these POD basis functions, ODExS was generated and used to simulate the transport phenomena in case II at 13 different values of D_p. Herein, we chose $m^{p_g} = 2$, $m^{u_g} = 15$, $m^{v_g} = 11$, $m^{u_s} = 10$, and $m^{v_s} = 5$ such that 99.99% of the total energy were retained by the POD modes used in ODExS.

Table XXVI lists the errors of the field variables obtained from ODExS compared against the numerical results of MFIX at $t = 1s$. These errors are also shown in Figure 45. As seen from Table XXVI and Figure 45, ODExS generally provided good predictions of the field variables at $D_p \in [0.05cm, 0.10cm]$. When $D_p > 0.10cm$, the errors of ODExS increased as D_p increased. When $D_p < 0.05cm$, the errors of ODExS increased as D_p decreased. The errors of ODExS increased more rapidly as D_p decreased when $D_p < 0.05cm$ than as D_p increased when $D_p > 0.10cm$, because the physics of the transport phenomena has significantly changed when D_p decreased below $0.05cm$. Figure 46 shows the fields of ϵ_g at $t = 1s$ with $D_p = 0.03cm$, $D_p = 0.08cm$, and $D_p = 0.12cm$. At $D_p = 0.03cm$, some bubbles appeared and this phenomenon is called bubbling fluidization. ODExS generated in this section could not capture the very complicated features of the bubbling fluidization phenomenon.

Table XXVI. Case II: errors of the results of ODExS at off-reference conditions

D_p [cm]	ε_{ϵ_g} [%]	ε_{p_g} [%]	ε_{u_g} [%]	ε_{v_g} [%]	ε_{u_s} [%]	ε_{v_s} [%]
0.03	0.0733	0.00051	9.71	2.41	8.61	9.09
0.04	0.0137	0.000077	2.495	0.389	10.45	4.713
0.05	0.00186	0.0000022	0.0686	0.0517	1.08	3.18
0.055	0.00158	0.00000272	0.102	0.044	1.08	2.967
0.06	0.00136	0.0000045	0.166	0.0494	1.57	2.73
0.07	0.00100	0.0000044	0.191	0.0578	1.52	2.16
0.08	0.00073	0.0000024	0.122	0.0423	0.679	1.50
0.09	0.00054	0.00000027	0.0298	0.0235	0.766	0.807
0.10	0.00042	0.00000027	0.161	0.0191	2.29	0.194
0.11	0.000376	0.0000051	0.323	0.0524	3.917	0.841
0.12	0.000368	0.0000072	0.487	0.1005	5.53	1.75
0.13	0.000374	0.0000091	0.654	0.129	7.56	2.711
0.14	0.000451	0.0000107	0.819	0.191	9.128	3.736



(a) ϵ_g

(b) p_g

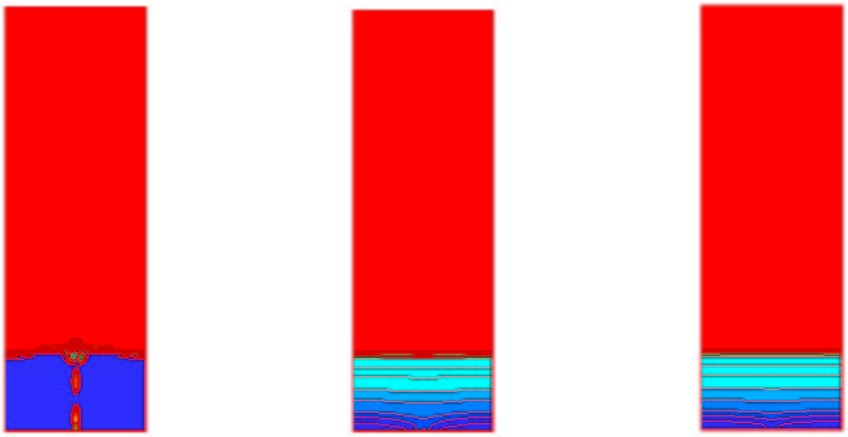
(c) u_g

(d) v_g

(e) u_s

(f) v_s

Fig. 45. Case II: errors of the results of ODExS at off-reference conditions



(a) $D_p = 0.03cm$ (b) $D_p = 0.08cm$ (c) $D_p = 0.12cm$

Fig. 46. Case II: field of ϵ_g at $t = 1s$ with different D_p

C. Summary

In this chapter, the POD-based ROMs have been applied to two cases of transport phenomena in fluidized beds. The first case modeled a compressible gas-only flow. The second case modeled a gas-solid interaction. In both cases, the POD-based ROMs have been applied to the reference conditions and the off-reference conditions.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

This chapter presents the conclusions derived from this work. In addition, recommendations for future work are presented.

A. Conclusions

In this thesis, POD-based ROMs have been applied for simulating the transport phenomena in fluidized beds. Two and one POD-based ROMs have been generated for gas-only flows and gas-solid transport phenomena, respectively. Results have proven that the POD-based ROMs are capable of capturing the details of the transport phenomena at both reference conditions and a range of off-reference conditions with large reductions of orders and computational costs.

In this thesis, we also analyzed the accuracy of the POD-based ROMs. The accuracy analysis was based on two sets of simple PDEs whose analytical solutions are known. Accuracy analysis demonstrated that the errors of the basis functions and time coefficients increase as the errors of the databases increase. The basis functions and time coefficients which correspond to smaller eigenvalues are more sensitive to the numerical errors of the databases. The errors of the reconstructed dependent variables also increase when the errors of the databases increase. The reconstructed dependent variables, however, are less sensitive to the errors of the databases compared against the basis functions and the time coefficients.

B. Future Work

In this thesis, the POD-based ROMs have been applied to the transport phenomena in fluidized beds. The geometry of the fluidized bed is simple. The challenge for the future is to extend the POD-based ROMs to complex geometry.

REFERENCES

[1] Dowell, E. H., Hall, K. C., Thomas, J. P., Florea, R., Epureanu, B. I., and Heeg, J., "Reduced Order Models in Unsteady Aerodynamics," AIAA Paper 99-1261, 1999.

[2] Florea, R, "Reduced Order Modeling and Eigenvalue Analysis of Unsteady Flows about Oscillating Airfoil and Turbomachinery Cascades," Ph.D. Dissertation, Duke University, Durham, North Carolina, 1996.

[3] Florea, R., Hall, K. C., and Cizmas, P. G. A., "Reduced-Order Modeling of Unsteady Viscous Flow in a Compressor Cascade," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 1039–1048.

[4] Thomas, J. P., Dowell, E. H., and Hall, K. C., "A Static/Dynamic Correction Approach for Reduced-Order Modeling of Unsteady Aerodynamics," AIAA Paper 2001-855, 2001.

[5] Romanowski, M. C., and Dowell, E. H., "Reduced Order Euler Equations for Unsteady Aerodynamic Flows: Numerical Techniques," AIAA Paper 96-0528, 1996.

[6] Berkooz, G., Holmes, P., and Lumley, J. L., "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 25, 1993, pp. 539–575.

[7] Sirovich, L., "Turbulence and the Dynamics of Coherent Structures Part:I-III," *Quarterly of Applied Mathematics*, Vol. XLV, No. 3, 1987, pp. 561–590.

[8] Chambers, D. H., Adrian, R. J., Moin, P., Stewart, D. S., and Sung, H. J., "Karhunen-Loéve Expansion of Burgers' Model of Turbulence," *Physics of Fluids*, Vol. 31, No. 9, 1988, pp. 2573–2582.

[9] LeGresley, P. A., and Alonso, J. J., "Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics," AIAA Paper 2001-0926,

2001.

[10]Deane, A. E., Kevrekidis, I. G., Karniadakis, G. E., and Orszag, S. A., "Low-Dimensional Models for Complex Geometry Flows: Application to Grooved Channels and Circular Cylinders," *Physics of Fluids*, Vol. 3, No. 10, 1991, pp. 2337–2354.

[11]Sahan, R. A., Liakopoulos, A., and Gunes, H., "Reduced Dynamical Models of Nonisothermal Transitional Grooved-Channel Flow," *Physics of Fluids*, Vol. 9, No. 3, 1997, pp. 551–565.

[12]Ma, X., and Karniadakis, G. E., "A Low-Dimensional Model for Simulating Three-Dimensional Cylinder Flow," *Journal of Fluid Mechanics*, Vol. 458, 2002, pp. 181–190.

[13]Cazemier, W., Verstappen, R. W. C. P., and Veldman, A. E. P., "Proper Orthogonal Decomposition and Low-Dimensional Models for Driven Cavity Flows," *Physics of Fluids*, Vol. 10, No. 7, 1998, pp. 1685–1699.

[14]Rediniotis, O. K., Ko, J., and Kurdila, A. J., "Reduced Order Nonlinear Navier-Stokes Models for Synthetic Jets," *Journal of Fluids Engineering*, Vol. 124, 2002, pp. 433-443.

[15]Lucia, D. J., King, P. I., Beran, P. S., and Oxley, M. E., "Reduced Order Modeling for a One-Dimensional Nozzle Flow with Moving Shocks," AIAA Paper 2001-2602, 2001.

[16]Hall, K. C., Thomas, J. P,. and Dowell, E. H., "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows," *AIAA Journal*, Vol. 38, No. 10, 2000, pp. 1853–1862.

[17]Romanowski, M. C., "Reduced Order Unsteady Aerodynamic and Aeroelastic Models using Karhunen-Loéve Eigenmodes," AIAA Paper 96-3981, 1996.

[18]Ravindran, S. S., "Reduced-Order Adaptive Controllers for Fluids using Proper

Orthogonal Decomposition," AIAA Paper 2001-0925, 2001.

[19]Cizmas, P. G. A., and Palacios, A., "Proper Orthogonal Decomposition of Turbine Rotor-Stator Interaction," *Journal of Propulsion and Power*, Vol. 19, No. 2, 2003, pp. 268–281.

[20]Liu, Z.-C., Adrian, R. J., and Hanratty, T. J., "Reynolds Number Similarity of Orthogonal Decomposition of the Outer Layer of Turbulent Wall Flow," *Physics of Fluids*, Vol. 6, No. 8, 1994, pp. 2815–2819.

[21]Cizmas, P. G., Palacios, A., O'Brien, T., and Syamlal, M., "Proper-Orthogonal Decomposition of Spatio-Temporal Patterns in Fluidized Beds," *Chemical Engineering Science*, Vol. 58, No. 19, 2003, pp. 4417-4427.

[22]Syamlal, M., Rogers, W., and O'Brien, T., "MFIX Documentation: Theory Guide," Technical Note DOE/METC-94/1004, U.S. Department of Energy, Morgantown, West Virginia, 1993.

[23]Syamlal, M., "MFIX Documentation: Numerical Technique," EG&G Technical Report DE-AC21-95MC31346, U.S. Department of Energy, Morgantown, West Virginia, 1998.

[24]Holmes, P. J., Lumley, J. L., Berkooz, G., Mattingly, J. C., and Wittenberg, R. W., "Low-Dimensional Models of Coherent Structures in Turbulence," *Physics Reports*, Vol. 287, 1997, pp. 337-384.

APPENDIX A

CONSTITUTIVE MODELS

Gas phase stress tensor

The gas viscous stress tensor $\bar{\bar{\tau}}_g$ is assumed to be of the Newtonian form

$$\bar{\bar{\tau}}_g = 2\mu_g \bar{\bar{D}}_g - \lambda_g \mathrm{tr}(\bar{\bar{D}}_g)\bar{\bar{I}},$$

where μ_g is the gas phase viscosity; $\lambda_g = -2/3\mu_g$; $\bar{\bar{I}}$ is an identity tensor; $\bar{\bar{D}}_g$ is the gas phase strain rate tensor, given by

$$\bar{\bar{D}}_g = \frac{1}{2}\left[\bigtriangledown\vec{v}_g + (\bigtriangledown\vec{v}_g)^T\right].$$

Solid phase stress tensor

MFIX uses the following model to compute the solid phase stress tensor

$$\bar{\bar{\tau}}_s = \begin{cases} \bar{\bar{\tau}}_s^{\mathcal{P}} & \text{if } \epsilon_g \le \epsilon_g^*\text{: Plastic Regime} \\ \bar{\bar{\tau}}_s^{\mathcal{V}} & \text{if } \epsilon_g > \epsilon_g^*\text{: Viscous Regime} \end{cases},$$

where ϵ_g^* is the packed-bed void fraction at which a granular flow regime transition is assumed to occur and ϵ_g^* is usually set to the void fraction at minimum fluidization.[22] The superscript $\mathcal{P}$ stands for plastic regime and $\mathcal{V}$ for viscous regime.

- *Plastic Regime*:

$$p_s^{\mathcal{P}} = 10^{25}(\epsilon_g^* - \epsilon_g)^{10}$$

$$\bar{\bar{\tau}}_s^{\mathcal{P}} = 2\mu_s^{\mathcal{P}}\bar{\bar{D}}_s$$

$$\mu_s^{\mathcal{P}} = \frac{p_s^{\mathcal{P}} \sin\phi}{2\sqrt{I_{2D_s}}}$$

Herein $\bar{\bar{D}}_s$ denotes the solid phase strain rate tensor. ϕ is the angle of internal friction. I_{2D_s} is the second invariant of the deviator of $\bar{\bar{D}}_s$:

$$I_{2D_s} = \frac{1}{6}\left[(D_{s11} - D_{s22})^2 + (D_{s22} - D_{s33})^2 + (D_{s33} - D_{s11})^2\right] + D_{s12}^2 + D_{s23}^2 + D_{s31}^2.$$

- *Viscous Regime*:

$$p_s^{\mathcal{V}} = K_{1s}\epsilon_s^2\Theta_s$$

$$\bar{\bar{\tau}}_s^{\mathcal{V}} = 2\mu_s^{\mathcal{V}}\bar{\bar{D}}_s + \lambda_s^{\mathcal{V}}\mathrm{tr}(\bar{\bar{D}}_s)\bar{\bar{I}}$$

$$\lambda_s^{\mathcal{V}} = K_{2s}\epsilon_s\sqrt{\Theta_s}$$

$$\mu_s^{\mathcal{V}} = K_{3s}\epsilon_s\sqrt{\Theta_s}$$

$$K_{1s} = 2(1 + e_s)\rho_s g_{0_s}$$

$$K_{2s} = 4d_{ps}\rho_s(1 + e_s)\epsilon_s g_{0_s}/(3\sqrt{\pi}) - \frac{2}{3}K_{3s}$$

$$K_{3s} = \frac{d_{ps}\rho_s\sqrt{\pi}}{6(3 - e_s)}\left[1 + 0.4(1 + e_s)(3e_s - 1)e_s g_{0_s}\right] + \frac{d_{ps}\rho_s 8\epsilon_s g_{0_s}(1 + e_s)}{10\sqrt{\pi}}$$

$$K_{4s} = \frac{12(1 - e_s^2)\rho_s g_{0_s}}{d_{ps}\sqrt{\pi}}$$

$$g_{0_s} = \frac{1}{1 - \epsilon_s} + 1.5\epsilon_s\left(\frac{1}{1 - \epsilon_s}\right)^2 + 0.5\epsilon_s^2\left(\frac{1}{1 - \epsilon_s}\right)^3$$

Herein e_s is the coefficient of restitution for particle-particle collisions. d_{ps} is the solid particle diameter. The granular temperature Θ_s is given by

$$\Theta_s = \left\{\frac{-K_{1s}\epsilon_s\mathrm{tr}(\bar{\bar{D}}_s) + \sqrt{K_{1s}^2\mathrm{tr}^2(\bar{\bar{D}}_s) + 4K_{4s}\epsilon_s\left[K_{2s}\mathrm{tr}^2(\bar{\bar{D}}_s) + 2K_{3s}\mathrm{tr}(\bar{\bar{D}}_s^2)\right]}}{2\epsilon_s K_{4s}}\right\}^2$$

Gas-solid momentum transfer

$$F_{gs} = \frac{3\rho_g \epsilon_s \epsilon_g}{4V_{rs}^2 d_{ps}} \left(0.63 + 4.8\sqrt{V_{rs}/Re_s}\right)^2 \left|\vec{v}_s - \vec{v}_g\right|$$

$$V_{rs} = 0.5\left(A - 0.06Re_s + \sqrt{(0.06Re_s)^2 + 0.12Re_s(2B - A) + A^2}\right)$$

$$A = \epsilon_g^{4.14}$$

$$B = \begin{cases} 0.8\epsilon_g^{1.28} & \text{if } \epsilon_g \leq 0.85 \\ \epsilon_g^{2.65} & \text{if } \epsilon_g > 0.85 \end{cases}$$

$$Re_s = \frac{d_{ps} \left|\vec{v}_s - \vec{v}_g\right| \rho_g}{\mu_g}$$

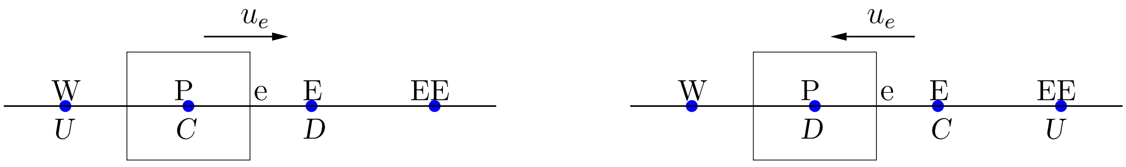
APPENDIX B

CONVECTION FACTOR OF SUPERBEE SCHEME

Let us use the control volume P shown in the following figure to illustrate the calculation of ξ_e, such that the field variable ϕ at the east face, ϕ_e is computed as

$$\phi_e = \xi_e \phi_E + (1 - \xi_e)\phi_P.$$

The following figure shows two cases according to the flow direction at the east face.



(a) $u_e \geq 0$ (b) $u_e < 0$

In this figure, C, U and D denote the central node, upwind node and downwind node, respectively. The calculation of the convection factor ξ_e using the superbee scheme is presented below:

$$\tilde{\phi}_C = \begin{cases} \frac{\phi_C - \phi_U}{\phi_D - \phi_U} & \text{if } \phi_C \neq \phi_D \\ 0 & \text{if } \phi_C = \phi_D \end{cases},$$

$$dwf = \begin{cases} \frac{1}{2}\max[0, \min(1, 2\theta), \min(2, \theta)],\ \theta = \frac{\tilde{\phi}_C}{1 - \tilde{\phi}_C} & \text{if } 0 \leq \tilde{\phi}_C < 1 \\ 1 & \text{if } \tilde{\phi}_C = 1 \\ 0 & \text{otherwise} \end{cases},$$

$$\xi_e = \begin{cases} dwf & \text{if } u_e \geq 0 \\ 1 - dwf & \text{if } u_e < 0 \end{cases}.$$

APPENDIX C

SAMPLE INPUT FILE FOR ODEXMFIX

```
#
# odexmfix.dat
# ODExMFIX input file for case I
# 1.1 Run control section
#
  TSTART  = 0.02
  TSTOP   = 5.0
  DT      = 1.D-3
  MAX_NIT = 20
  DT_MAX  = 1.D0
  DT_MIN  = 1.D-6
  DT_FAC  = 0.9D0
  TOL_RESID = 1.D-3
  TOL_DIVERGE = 1.D+2
#
# 1.2 Geometry and discretization section
#
  XLENGTH = 25.4D0     IMAX = 50
  YLENGTH = 76.5D0     JMAX = 76
  DISCRETIZE = 2
#
# 1.3 physical properties section
#
  MU_g0 = 1.8D-2
  MW_g0 = 29.D0
  T_g0  = 297.D0
#
# 1.4 pod section
#
   NP = 3
   NU = 7
   NV = 5
```

APPENDIX D

SAMPLE INPUT FILE FOR ODEX3

```
#
# ODEx3 input file for case II
#
# 1.1 Run control section
#
  TIME  = 0.2
  TSTOP = 13.0
  DT    = 1.D-5
  ATOL  = 1.D-3
#
# 1.2 Geometry and discretization section
#
  XLENGTH = 25.4D0     IMAX = 50
  YLENGTH = 76.5D0     JMAX = 76
  DISCRETIZE = 1
#
# 1.3 physical properties section
#
  MU_g0 = 1.8D-4
  MW_g0 = 29.D0
  T_g0  = 297.D0
#
# 1.4 pod section
#
   NR = 9
   NU = 11
   NV = 7
```

APPENDIX E

SAMPLE INPUT FILE FOR ODEXS

```
#
# Hybrid_puv input file for case II
#
# 1.1 Run control section
#
  TSTART  = 0.20
  TSTOP = 1.0
  DT    = 2.D-4
  MAX_NIT = 20
  DT_MAX  = 1.D0
  DT_MIN  = 1.D-6
  DT_FAC  = 0.9D0
  TOL_RESID = 1.D-3
  TOL_DIVERGE = 1.D1
#
# 1.2 Geometry and discretization section
#
  XLENGTH = 25.4D0     IMAX = 50
  YLENGTH = 76.5D0     JMAX = 78
  DISCRETIZE = 2
#
# 1.3 physical properties section
#
  MU_g0   = 1.8D-4
  MW_g0   = 29.D0
  T_g0    = 297.D0
  RO_s0   = 1.0
  D_p     = 0.05
  C_e     = 0.8
  Phi     = 30.0
  EP_star = 0.44
#
# 1.4 POD section
#
  nP_g    = 4
  nU_g    = 11
  nV_g    = 7
  nU_s    = 7
  nV_s    = 5
```

VITA

Tao Yuan, was born in May 1977 in Wuhan, Hubei Province, China. He received his Bachelor of Engineering degree in Thermal Engineering from the Tsinghua University, Beijing, China in July 2000.

He began his graduate study at Texas A&M University in September 2001 and received his Master of Science degree in Aerospace Engineering in December 2003. His research interests focused on the reduced order modeling for the transport phenomena.

Tao Yuan's permanent address is No.1 Yangluo Main Street, Xinzhou District, Wuhan, Hubei Province 430415, China.

The typist for this thesis was Tao Yuan.