ADVANCES IN REDUCED-ORDER MODELING BASED ON PROPER ORTHOGONAL DECOMPOSITION FOR SINGLE AND TWO-PHASE FLOWS

A Thesis

by

RAYMOND LEE FONTENOT

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 2010

Major Subject: Aerospace Engineering

ADVANCES IN REDUCED-ORDER MODELING BASED ON PROPER ORTHOGONAL DECOMPOSITION FOR SINGLE AND TWO-PHASE FLOWS

A Thesis

by

RAYMOND LEE FONTENOT

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 2010

Major Subject: Aerospace Engineering

ABSTRACT

Advances in Reduced-Order Modeling Based on

Proper Orthogonal Decomposition

for Single and Two-Phase Flows. (December 2010)

Raymond Lee Fontenot, B.S., McNeese State University

Chair of Advisory Committee: Dr. Paul Cizmas

This thesis presents advances in reduced-order modeling based on proper orthogonal decomposition (POD) for single and two-phase flows. Reduced-order models (ROMs) are generated for two-phase gas-solid flows. A multiphase numerical flow solver, MFIX, is used to generate a database of solution snapshots for proper orthogonal decomposition. Time-independent basis functions are extracted using POD from the data, and the governing equations of the MFIX are projected onto the basis functions to generate the multiphase POD-based ROMs. Reduced-order models are constructed to simulate multiphase two-dimensional non-isothermal flow and isothermal flow particle kinetics and three-dimensional isothermal flow. These reduced-order models are applied to three reference cases. The results of this investigation show that the two-dimensional reduced-order models are capable of producing qualitatively accurate results with less than 5% error with at least an order of magnitude reduction of computational costs. The three-dimensional ROM shows improvements in computational costs.

This thesis also presents an algorithm based on mathematical morphology used to extract discontinuities present in quasi-steady and unsteady flows for POD basis augmentation. Both MFIX and a Reynolds Average Navier-Stokes (RANS) flow solver, UNS3D, are used to generate solution databases for feature extraction. The algorithm is applied to bubbling fluidized beds, transonic airfoils, and turbomachinery seals. The results of this investigation show that all of the important features are extracted without loss in accuracy.

To My Family

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Cizmas, for his patience and advice through the course of my research. Next, I would like to thank my thesis committee, Dr. Othon Rediniotis and Dr. Andrew Duggleby. Also, I would like to thank my fellow graduate students, past and present, for allowing me to bounce ideas off them, regardless of how much sense they made. Finally, I would like to thank my parents and brothers and sisters for their encouraging words and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

NOMENCLATURE

Acronyms

ASF	Alternating Sequenential Filter
MEDts	Morphological Edge Detection algoritm for structured domains
MEDuns	Morphological Edge Detection algoritm for unstructured domains
MFIX	Multiphase Flow with Interphase Exchanges (software)
ODE	Ordinary Differential Equations
ODEti	POD-Based ROM for energy equation
ODEg	POD-Based ROM for granular energy equation
ODEV	POD-Base ROM for 3D isothermal fluidized beds
PDE	Parital Differential Equations
POD	Proper Orthogonal Decomposition
PODDEC	Proper Orthogonal Decomposition (software)
RANS	Reynolds Averaged Navier-Stokes equations
ROM	Reduced-Order Model
SST	Shear-Stress Transport turbulence model
UNS3D	Unstructed 3D Flow Solver (software)

Roman Letters

a, A	Face area
avg_h	Harmonic average
b	Structuring element
be	Blur-erosion operator
c_p	Specific heat at constant pressure

c_v	Specific heat at constant volume
d	Dilation morphological operator
D	Blurring element displacement
D_{nb}	Control volume diffusion
$\mathbf{D}$	Total derivative
db	Dilation-blur operator
d_{ps}	Solid particle diameter
e	Erosion morphological operator
e	Internal energy
E	Total energy per unit mass
f	Image
f_1	Shear model blending function
F_{gs}	Gas-solid particle drag force
g	Gravitational acceleration
g_o	Radial distribution function
H	Total enthalpy per unit mass
h	Internal enthalpy
Ind	Edge strength index
(i, j, k)	Discrete coordinates
J	Isoparametric mapping Jacobian matrix
J_s	Collisional dissipation
(l, m)	Structuring element local coordinates
k	Thermal conductivity

k	Turbulent kinetic energy
K	Diffusivity coefficient
L_{rs}	Reduced stencil size
m	Number of POD modes
M	Number of snapshots
M_w	Average molecular weight of gas
$\vec{n}$	Normal vector
N	Number of discrete spatial grid points
N, S, E, W, T, B, P	Control volume centers
n, s, e, w, t, b, p	Control volume faces
p, P	Pressure
$\vec{p}, \vec{q}$	Face diagonals
q	Generic field variable
$\vec{q}$	Conductive heat flux
q_h	Heat source
R	Universal gas constant
$R(x, x')$	Auto-correlation function
$\mathbf{R}$	Tensor product matrix
Re	Reynolds number
S	Surface area
S_{ij}	Strain rate tensor
t	Time
T	Temperature

$thres$	Image threshold operator
$tr(D)$	Trace of stress tensor
(u, v, w)	Components of velocity vector
v_{slip}	Particle slip velocity
$\vec{v}$	Velocity vector
V	Volume
(x, y, z)	Cartesian coordinates

Greek Letters

α	POD time coefficients
β	Discountinuity POD time coefficients
γ	Ratio of specific heats
γ_l	Gas-solids heat transfer coefficient
δ_{ij}	Kronecker delta
ε_g	Void fraction
(ξ, η)	Isoparametric mapping coordinates
ξ	Convection weighing factor
η	Function of restitution coefficient
κ_s	Granular conductivity
ϕ	POD basis functions
ψ	Discontinuity POD basis functions
Ψ	Isoparametric mapping shape functions
$\prod$	Exchange of granular energy
μ	Viscosity

μ_L	Molecular viscosity
μ_T	Turbulent viscosity
ω	Turbulent specific dissipation
ρ	Density
λ	Eigenvalue
λ_s	Second coefficient of solids viscosity
Θ	Granular energy
τ	Shear and normal stress
$\bar{\bar{\tau}}_g$	Gas phase viscous stress
$\bar{\bar{\tau}}_s$	Granular phase stress
σ	Normal stress

Subscript

g	Gas phase
i	Counting index
s	Solid phase
l, m	Gas or Solid phase
0	Average
nb	Neighbor control volume
N, S, E, W, T, B, P	Control volume centers
n, s, e, w, t, b, p	Control volume faces

Superscript

T	Transpose
o	Value from previous iteration

CHAPTER I

INTRODUCTION

A. Statement of Problem

Modeling of fluid dynamic processes is a critical engineering design tool for complex machines, reactors, and pipe networks which depends on robust computational fluid dynamic models. These models are extremely important, but they are computationally expensive and often very time consuming for modeling most practical problems. For design applications where repetitive simulations are needed, the use of full-order models becomes inefficient. Reduced-order models (ROMs) based on proper orthogonal decomposition (POD) are a computationally efficient alternative with quick turnaround times for results. ROMs based on POD achieve their computational efficiency by (i) replacing the partial differential equations (PDEs) that describe the flows of interest with ordinary differential equations (ODEs) and (ii) reducing the number of equations solved.

This research is a continuation of previous work by Tao Yuan and Brian Richardson who developed POD-based reduced-order models to describe isothermal and non-isothermal transport phenomena in fluidized beds [1, 2]. The focus of this research is (i) to improve the POD-based ROM for the scalar energy equation, (ii) develop and implement a POD-based ROM for the computation of the granular energy equation, (iii) extend the current POD-based ROM to three dimensions, and (iv) develop and apply a novel method based on mathematical morphology to extract transient and periodic features for basis augmentation of POD-based ROMs.

This thesis follows the style of the Journal of Computational Physics.

B. Background

This section provides a literature review of reduced-order modeling based on POD and of mathematical morphology for feature detection. The use of each method in this thesis is also discussed.

1. Reduced-Order Modeling Based on Proper Orthogonal Decomposition

Reduced-order modeling is a means of replacing an infinite-dimensional dynamical system, consisting of governing PDEs, with a dynamical system of ODEs of small dimension. Several techniques have been developed in previous work for constructing ROMs in a variety of fields. A full review of ROMs for flow phenomena can be found in the article of Lucia *et al.* [3].

Eigenmodes representative of the flow, with a technique similar to that used in structural dynamics, are used to generate ROMs. A ROM for the unsteady viscous flow in compressor cascades was developed by Florea *et al.* [4]. Thomas *et al.* [5] used the eigenmodes of flows about an isolated airfoil and aeroelastic wing to generate a ROM. Finally, Romanowski *et al.* [6] constructed ROMs for the Euler equations based on the representative eigenmodes.

An attractive alternative to using eigenmodes as a basis for ROMs is proper orthogonal decomposition (POD). POD extracts an optimal basis from an ensemble of shapshots for a modal decomposition [7]. POD is also known as the Karhunen-Loéve decomposition, singular value decomposition, singular systems analysis, and principal components analysis. POD uses a time series of snapshots from experiments and/or computational models. These snapshots are then used to assemble an autocorrelation which in turn is solved to determine a representative set of optimal basis functions for the flow field. ROMs based on POD are generated by projecting of the governing equations PDEs onto a smaller space spanned by POD basis functions. This allows the fluid flow to be represented by a small number of ODEs. More in-depth discussions on POD-based reduced-order models can be found in reviews by Sirovich [8] and Berkooz *et al.* [7].

Lumley was the first to apply POD to fluid dynamic problems [7]. He used POD as a means to extract the coherent structures from turbulent flows to study and characterize them [7]. POD has been used to extract spatial and temporal patterns in fluidized beds and determine the feasibility of ROMs for two-phase gas-solids flows [9]. POD was also used to determine the feasibility of constructing POD-based ROMs for turbomachinery [10]. In both of these cases, it was shown that using POD to generate a reduced-order model was feasible.

Reduced-order models based on POD have been constructed for several different system of equations, including the Burgers [1], Euler [11], and Navier-Stokes [12, 13] equations. These ROMs have modeled several different types of fluid dynamic problems. Deane *et al.* [14] applied POD-based ROMs to flows in periodically grooved channels as well as to the wake of an isolated circular cylinder. They accurately predicted limit-cycle behavior for these flows using their ROMs [14]. Hall *et al.* [15] generated POD-based ROMs to analyze and predict two-dimensional aeroelastic phenomena on a NACA 64A010 airfoil. Hall *et al.*'s model predicted flutter boundaries for this airfoil reasonably well for a wide range of Mach numbers [15]. Rediniotis *et al.* [16] used POD-based ROMs as a means to control synthetic jets. Yuan [1] applied POD-based ROMs to isothermal fluidized bed reactors, and Richardson [2] extended the POD-based ROM to non-isothermal fluidized bed reactors.

POD has also been used to model flows with moving boundaries and discontinuities. Pettit and Beran [17] created POD-based ROMs for a moving boundary and stationary bump in supersonic flows for the discrete Euler equations. They showed that POD-based ROMs were able to completely capture the dynamics of the flow as long as the shock located on the leading edge of the moving panel remained attached [17]. ROMs based on POD for quasi-steady and unsteady shock motion were studied heavily by Lucia [18]. Lucia applied POD-based ROMs to three types of flow geometries: a one-dimensional nozzle, a blunt body, and a transonic panel. For the one-dimensional nozzle Lucia was able to accurately model the shock to within one grid point [18]. For the blunt-body and transonic panel, Lucia also achieved good results using POD-based ROMs. However, both cases used the full-order model to solve the region of the flow where the shock was located, requiring an a priori knowledge of the shock location. POD was only applied to regions where the dynamics of the flow were not significant [18].

This thesis explores two avenues for POD-based ROMs. The thesis explores extending POD-based ROMs to two-dimensional isothermal flow particle kinetics and three-dimensional isothermal flows in fluidized beds. This builds on the work performed by Yuan [1] and Richardson [2] for POD-based ROMs of isothermal and non-isothermal two-dimensional fluidized beds, respectively. The thesis also explores POD-based ROMs for flows with moving discontinuities. Specifically, the thesis explores detecting and extracting moving discontinuities where their location is not well known, unlike the flows investigated by Lucia [18] and Pettit and Beran [17]. Once extracted, it is expected that these features can be used to augment the POD basis functions. However, the use of the extracted features for POD basis augmentation is not explored in this thesis. The method used to extract the moving discontinuities for basis augmentation is mathematical morphology, which is discussed in the next section.

2. Morphological Feature Detection

Mathematical morphology is a mathematical theory used primarily as a technique for classifying edges and features in images. Matheron and Serra are attributed with developing the theory in the 1960's at the École des Mines de Paris, France [19]. Matheron and Serra first applied morphology to binary images of iron ore deposits as a means of determining the ore's porosity [20]. Morphology has since been extended to more complex images (grayscale and color or multiscale) as well as complex topological spaces [21]. Morphology has also been used for single and multidimensional signal analysis [22]. Morphology is infinite dimensional [20], and the algorithms based upon it are easily extensible from lower to higher dimensions [23] For a more thorough discussion on mathematical morphology and its applications, books by Serra [24], Soille [20], and Dougherty and Lotufo [25] are recommended.

Morphological edge detection is only one of several types of image processing techniques used to extract edges. Some examples of non-morphological edge detection algorithms include Sobel, Canny, and Prewitt, among others [26]. Morphological based edge detection algorithms are generally better at detecting edges in non-noisy and noisy images and are computationally more efficient than their non-morphological counterparts [27, 28]. Computational efficiency is a by-product of morphology's algebraic basis; other methods are generally differentially based, requiring expensive computation of gradients [27].

Several morphological edge detection algorithms exist. Examples of these include dilation residue, erosion residue, blur-minimum [26], color gradient (multiscale versions of the previous morphological edge detection algorithms) [29], and ASF (alternating sequential filter, based on the erosion residue technique using alternating morphological noise suppression) [27]. Dilation and erosion residue are the simplest means of morphological edge detection [26]. These use the differences between the maximum and minimum of the images, respectively, to detect edges. These perform marginally well for simple grayscale images [26]. Dilation and erosion residue can be combined to detect edges for slight improvement of noisy images when compared to the separate methods [26]. Lee *et al.* [26] developed the blur-minimum method to handle noisy images and detect step type edges. This operator uses blurring, a non-morphological operation, to reduce noise before morphological operations are applied [26]. It performs significantly better for noisy images than most of its morphological edge detection counterparts and non-morphological edge detection algorithms as well [26, 27, 29]. The blur-minimum morphological edge detection method is a popular means of edge detection for gray and multiscale images because of its performance. Due to its popularity and performance with noisy images, the blur-minimum technique was selected as the morphological feature detection algorithm used to extract features for POD basis augmentation in this study.

C. Summary of Work

This thesis presents several advances in reduced-order modeling based on proper orthogonal decomposition for single and multiphase flows. The existing POD-based ROM developed by Richardson [2] was improved for the scalar energy equation in multiphase flows. A ROM based on POD was developed and implemented into the current multiphase ROM for the granular energy equation. The ROM was additionally extended to three-dimensional fluidized beds. The POD algorithm was applied to the multiphase governing PDEs. These reduced-order models were developed into numerical algorithms that were used to model two- and three-dimensional fluidized beds. The derivation of the algorithms as well as the results from the application of the codes are presented in the thesis.

This thesis also presents an algorithm based on mathematical morphology for extraction of transient and periodic features present in both single and multiphase flows as a means of augmenting POD basis functions. Morphological feature detection, using the blur-minimization technique, is applied to three different types of flows that exhibit moving discontinuities: bubbling fluidized beds, cavity flows, and transonic airfoils. The derivation of the algorithm for morphological feature detection is presented for highly structured spatial domains, as is the extension to general spatial domains. Results from both morphological feature detection algorithms are presented in this thesis.

D. Original Contributions of Work

The reduced-order model based on proper orthogonal decomposition for multiphase flows is a continuation of previous work. The author improved the ROM based on POD for the scalar energy equation of multiphase flow. The ROM was extended by applying POD to the granular energy equation. The ROM was further extended to model three-dimensional isothermal multiphase flows. The generated numerical models were used to simulate fluidized beds and investigate their speed-up factor and accuracy. The investigation included the simulation of a non-bubbling fluidized bed. Several different input combinations of basis functions were tested to model the flow. The thesis presents POD basis functions that were the optimal combination of best accuracy and best reduction of computational speed in comparison with the full-order.

The author also developed an algorithm as a means of extracting periodic and transient flow features for future augmentation of the POD basis functions. The algorithm, based on mathematical morphology, is a novel approach for extracting flow features. This algorithm was applied to single and multiphase flows that exhibit these quasi-steady and unsteady features which can present modeling problems when applying POD. The results of the extraction algorithm are compared to the results from the full-order models to determine if the feature was accurately depicted. The feasibility of using mathematical morphology for augmentation of POD basis functions is determined.

E. Outline of Thesis

Chapter II describes the theory of POD and the general scheme used to generate reduced-order models based on POD. Chapter III describes mathematical morphology and outlines the algorithms used to extract moving discontinuities for highly structured and general spatial domains. Chapter IV presents the physical models for single and multiphase flows. Chapter V presents the single and multiphase full-order models used in this study. Chapter VI presents the derivation of the POD-based ROMs to the scalar energy equation, granular energy equation, and three-dimensional isothermal multiphase fluidized beds. The application and results of the POD-based ROMs are presented in Chapter VII. Chapter VII also presents the application and results of morphological feature detection to flows with moving discontinuities. Conclusions and future work are presented in Chapter VIII. Appendix A presents constitutive models for multiphase flows. Appendix B presents an example of the morphological feature detection algorithm as applied to a one-dimensional signal. Appendices C-E present sample input files for the POD-based ROMs. Appendices F-H present sample input files for the morphological feature detection algorithm.

CHAPTER II

REDUCED-ORDER MODELS BASED ON PROPER ORTHOGONAL DECOMPOSITION

A. Introduction

The purpose of this chapter is to present the theory and methodology of reduced-order modeling based on proper orthogonal decomposition. The theory of proper orthogonal decomposition is presented. The general scheme for developing and generating ROMs based on POD is also presented.

B. Proper Orthogonal Decomposition

The purpose of this section is to present the mathematical theory of proper orthogonal decomposition. POD is a method for obtaining the optimal basis set from a group of observations, $u(x,t)$. POD extracts key spatial features from physical systems with spatial and temporal characteristics [30]. From the set of observations $u(x,t_i)$, POD extracts time-independent orthonormal basis functions $\Phi_k(x)$ and time-dependent orthogonal coefficients $\alpha_k(t_i)$ where the reconstruction

$$u(x,t_i) = \sum_{k=1}^{M} \alpha_k(t_i)\Phi_k(x) \qquad i = 1,\ldots,M \tag{2.1}$$

is optimal when the average least-square truncation error

$$\varepsilon_m = \langle \| u(x,t_i) - \sum_{k=1}^{M} \alpha_k(t_i)\Phi_k(x) \|^2 \rangle \tag{2.2}$$

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

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

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

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

The optimal condition given by (2.2) is equivalent to finding the functions Φ that maximize the normalized averaged projection of u onto Φ

$$\max_{\Phi \in L^2} \frac{\langle |(u, \Phi)|^2 \rangle}{\|\Phi\|^2}, \tag{2.3}$$

where $| \cdot |$ denotes the modulus. The optimum condition reduces to [30]

$$\int_D \langle u(x) u^*(x') \rangle \Phi(x') dx' \; = \lambda \Phi(x), \tag{2.4}$$

which is a homogeneous Fredholm integral of the second kind [31].

Consequently, the optimal basis functions Φ_k, or the POD basis functions, are the eigenfunctions of (2.4), whose kernel function is the auto-correlation function

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

For the finite-dimensional case, the auto-correlation function $R(x, x')$ is replaced by the tensor product matrix

$$\mathbf{R}(x, x') = \frac{1}{M} \sum_{i=1}^{M} u(x, t_i) u^T(x', t_i), \tag{2.5}$$

where M is the number of observations.

The eigenfunctions $\Phi_k(x)$ are vector-valued functions and have the same dimension M as the observations of u. It can be shown that the eigenvectors of $\mathbf{R}$ are the eigenfunctions (POD basis functions) $\Phi_k(x)$ [30]. The derivation of the integral equation (2.3) is also generalized to the vector-valued functions such as the three-dimensional velocity field $\mathbf{u}(\mathbf{x}, t)$, where $\mathbf{u}$ is composed of (u, v, w) and $\mathbf{x}$ is the Cartesian coordinate system in three dimensions, (x, y, z). $\mathbf{R}(\mathbf{x}, \mathbf{x}')$ can then be written as:

$$\mathbf{R}(\mathbf{x}, \mathbf{x}') = \frac{1}{M} \sum_{i=1}^{M} \mathbf{u}(\mathbf{x}, t_i) \mathbf{u}^T(\mathbf{x}', t_i). \tag{2.6}$$

C. General Scheme for Reduced-Order Modeling Based on Proper Orthogonal Decomposition

Reduced-order models utilizing proper orthogonal decomposition require three critical steps for successful model development and deployment: (i) generation of the database for POD, (ii) modal decomposition of the database, and (iii) Galerkin projection of the basis functions onto the governing equations. For discussion purposes, the following PDE will be used:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{D}(\mathbf{u}) \tag{2.7}$$

where $\mathbf{u}(\mathbf{x}, t)$ is a state vector. Ω and $(0, T]$ are the spatial and temporal domains respectively. $\mathbf{D}(\mathbf{u})$ is the total derivative of $\mathbf{u}$ [1].

1. Database Generation

The database is a collection of solution data gathered from a governing equation or set of governing equations or experiments. In this example, a database set representative of the solutions of (2.7) is assumed to have been collected. The database is either collected from experimental data, numerical simulation, or a combination of both.

The database can consist of a wide variation of several different physical quantities, such as pressure, density, velocity, temperature, etc., for the same temporal domain sample. This is a means of augmenting the solution database which lends to more flexibility of the ROM.

2. Modal Decomposition

The resultant output of the database generation is a set of snapshots representative of $\mathbf{u}(\mathbf{x}, t_i)$, $i \in [1, M]$, where M is the total number of snapshots collected in the database set. It will be assumed that $\mathbf{u}$ can be decomposed into $\bar{\mathbf{u}}(\mathbf{x})$, the time averaged mean, and $\mathbf{u}'(\mathbf{x}, t)$, the time dependent fluctuation. The basis functions ϕ_j are the eigenvectors of the matrix $R(\mathbf{x}, \mathbf{x}')$. With the basis functions, $\mathbf{u}(\mathbf{x}, t)$ is reconstructed as:

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \sum_{j=1}^{M} \alpha_j(t)\phi_j(\mathbf{x}) = \sum_{j=0}^{M} \alpha_j(t)\phi_j(\mathbf{x}) \tag{2.8}$$

where $\bar{\mathbf{u}}(\mathbf{x})$ can be represented as:

$$\bar{\mathbf{u}}(\mathbf{x}) = \phi_0(\mathbf{x})\alpha_0(t) \tag{2.9}$$

with the time coefficient $\alpha_0 \equiv 1$ allowing (2.8) to be simplified.

The solution database forms an eigenvalue problem that must be solved to extract the basis functions $\phi(\mathbf{x})$ and the time coefficients $\alpha(t)$. A popular technique for solving the eigenvalue problem is the method of snapshots developed by Sirovich [8]. The method of snapshots is an efficient method if the resolution of the spatial domain N is greater than that of the total number of snapshots M. This method is based on the principle that the vectors $\mathbf{u}_i$ and the eigenvectors ϕ_k both span the same linear space [30]. This property allows the eigenvectors to be written as a linear combination of the data vectors,

$$\phi_k = \sum_{i=1}^{M} v_i^k \mathbf{u_i}, \qquad k \in [1, M]. \tag{2.10}$$

Introducing (2.10) into the eigenvalue problem

$$\mathbf{R}(\mathbf{x}, \mathbf{x}')\phi(\mathbf{x}) = \lambda\phi(\mathbf{x}'), \tag{2.11}$$

simplifies the eigenproblem to [8]:

$$\mathbf{Cv} = \lambda\mathbf{v}\ . \tag{2.12}$$

In (2.12), $v^k = (v_1^k, v_2^k, ..., v_M^k)$ is the k^{th} eigenvector. $\mathbf{C}$ is a symmetric $M \times M$ matrix defined as [8]:

$$C_{ij} = \frac{1}{M}(\mathbf{u}'(\mathbf{x}, t_i), \mathbf{u}'(\mathbf{x}, t_j)). \tag{2.13}$$

The eigenvectors of tensor product matrix $\mathbf{R}$ in (2.11), given by (2.5), are calculated by computing the eigenvectors of the much smaller matrix $\mathbf{C}$. This study used a code called PODDEC developed by Paul Cizmas and Antonio Palacios based on the method of snapshots to extract the basis functions and time coefficients from the generated database.

3. Galerkin Projection

The final stage of generating reduced-order models based on proper orthogonal decomposition is the use Galerkin projection to project the basis functions extracted by the method of snapshots onto the governing equations. First, the eigenvalues are ordered by decreasing magnitude such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M \geq 0$. This in turn orders the eigenvectors, or basis functions, ϕ according to their eigenvalues. It is assumed that the majority of the energy is contained in the first m POD modes, where

$m < M$, such that:

$$\sum_{j=1}^{m} \lambda_j \simeq \sum_{j=1}^{M} \lambda_j. \tag{2.14}$$

Therefore, it is reasonable to approximate $\mathbf{u}(\mathbf{x}, t)$ using the first POD modes:

$$\mathbf{u}(\mathbf{x}, t) \simeq \sum_{j=0}^{m} \alpha_j(t)\phi_j(\mathbf{x}). \tag{2.15}$$

Next, (2.15) is substituted into the example PDE (2.7) to yield:

$$\sum_{j=1}^{m} \frac{d\alpha_j(t)}{dt}\phi_j(\mathbf{x}) = \mathbf{D}\left(\sum_{j=0}^{m} \alpha_j(t)\phi_j(\mathbf{x})\right). \tag{2.16}$$

Using Galerkin projection, the basis functions $\phi_k(\mathbf{x})$ are projected onto (2.16) to give:

$$\left(\phi_k, \sum_{j=1}^{m} \frac{d\alpha_j(t)}{dt}\phi_j(\mathbf{x})\right) = \left(\phi_k, \mathbf{D}\left(\sum_{j=0}^{m} \alpha_j(t)\phi_j(\mathbf{x})\right)\right) \tag{2.17}$$

This transforms the governing equations from PDEs to ODEs where the only unknowns are the time coefficients $\alpha_k(t), \quad k \in [1, m]$:

$$\frac{d\alpha_k}{dt} = F_k(\alpha_1, \alpha_2, ..., \alpha_m), \qquad k \in [1, m]. \tag{2.18}$$

In the derivation of (2.18) from (2.17), the orthogonality of the basis functions is used:

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

where δ_{kj} is the Kronecker delta.

By utilizing POD, the PDEs of (2.7) were replaced by the ODEs of (2.18). The total number of equations were also reduced from N spatial points to m number of modes. These two results of POD generate a much simpler system to solve. The linear ODEs of (2.18) can easily be solved using the appropriate ODE solvers to predict the time coefficients α_j. With the time coefficients from (2.18), $\mathbf{u}(\mathbf{x}, t)$ can be reconstructed using the POD approximation given by Equation (2.15). The time coefficients α_j are also obtained from directly projecting the solution database onto the basis function ϕ_j:

$$\alpha_j^{POD}(t_k) = (\phi_j(\mathbf{x}), \mathbf{u}(\mathbf{x}, t_k)) \qquad j \in [1, m], \quad k \in [1, M]. \tag{2.19}$$

The time coefficients obtained from (2.19) using PODDEC are considered the "exact" solution of the time coefficients when compared to the time coefficients obtained from the ROMs. This "exactness" is only true insofar as the time coefficients from the ROM and full-order model are generated for the same reference conditions.

D. Summary

This chapter presented the theory of proper orthogonal decomposition. The general scheme generating reduced-order models based on POD was also presented. The scheme was illustrated with an example PDE (2.7). It was shown that the governing PDEs were replaced with ODEs, and the total number of equations was reduced from N spatial points to m number of modes. PODDEC is used to extract both the time coefficients, α_k, and basis functions, ϕ_k, using the method of snapshots. The time coefficients and basis functions are used to approximate the field variables. The governing PDEs is projected onto the basis functions to generate the new system of reduced ODEs. The next chapter presents morphological feature detection used to extract moving discontinuities for basis augmentation of reduced-order models based on POD.

CHAPTER III

MORPHOLOGICAL FEATURE DETECTION

A. Introduction

The purpose of this chapter is to present both the motivation for augmenting POD basis functions obtained from flows with moving discontinuities and the algorithm for locating and extracting the discontinuities. The beginning of the chapter will present the motivation and discussion of using POD for flows with moving discontinuities. The bulk of the chapter though will present the algorithm based on mathematical morphology used to locate the discontinuities in fluid flows. First, the theory and development of mathematical morphology is presented. Non-morphological operations required for feature detection are presented next. The algorithm based on the blur-minimization feature detection technique proposed by Lee *et al.* [26] is presented for highly structured computational domains. The end of the chapter will also discuss the extension of the algorithm to general computational domains.

B. Proper Orthogonal Decomposition of Flows with Moving Discontinuities

Proper orthogonal decomposition, as discussed earlier, is a method of approximating a spatially and temporally dependent variable, such as velocity, with time-dependent time coefficients and spatial-dependent basis functions as given by (2.8). This approximation is satisfied so long as the dynamics of the problem are fully captured within the database. For fluid dynamic problems without moving discontinuities this is generally true. However, for flows that exhibit moving discontinuities the POD approximation is not sufficient. Lucia [18] showed that POD was capable of modeling these types of flows with moving discontinuities, but he had to decompose the solution domain into separate regions. Lucia modeled the shocked region with POD when the location of the shock was known and with the full-order model when its location was not known a priori [18]. For the case with known shocks, Lucia increased the number of basis functions used to approximate each variable and was able to accurately model the shock [18].

The work of Lucia [18] may not be the most efficient or accurate method for modeling moving discontinuities. Decomposing the domain into separate regions limits the predictive capabilities of the reduced-order model. For example, in the case of multiphase flows the moving discontinuities are bubbles generated from inlet velocities above the particle fluidization limit. This phenomena will be discussed in more detail later in this thesis. Richardson [2] generated a POD-based ROM for multiphase isothermal flow. The void fraction, ε_g, fully captures the bubbles as it is a measure of the amount of gas in a finite volume. A plot of the void fraction for selected time instances from the full-order model, MFIX, of a bubbling fluidized bed is shown in Figure 1. Table I [32] presents the flow conditions used to generate the solution. Notice that in this situation the bubbles can occur within the entire bed region, which is roughly half of the entire physical domain. This makes domain decomposition impractical.

Table I: Parameters of an isothermal bubbling fluidized bed.

Parameter	Description	Units	
x_{length}	Length of the domain in x-direction	cm	39.37
y_{length}	Length of the domain in y-direction	cm	58.44
i_{max}	Number of cells in x-direction	-	108
j_{max}	Number of cells in y-direction	-	124
v_1	Gas jet velocity	cm/s	355.0
v_2	Gas distributor velocity	cm/s	28.4
p_{g_s}	Static pressure at outlet	g/(cm/s^2)	1.06×10^6
T_{g0}	Gas temperature	K	297
μ_{g0}	Gas viscosity	g/(cm/s)	1.8×10^{-4}
t_{start}	Start time	s	0.0
t_{stop}	Stop time	s	1.0
Δt	Initial time step	s	1.0×10^{-4}
ρ_s	Particle density	g/cm^3	2.61
D_p	Particle diameter	cm	0.08
h_{s0}	Initial height of packed bed	cm	29.2
ϵ_g^*	Initial void fraction of packed bed	-	0.4

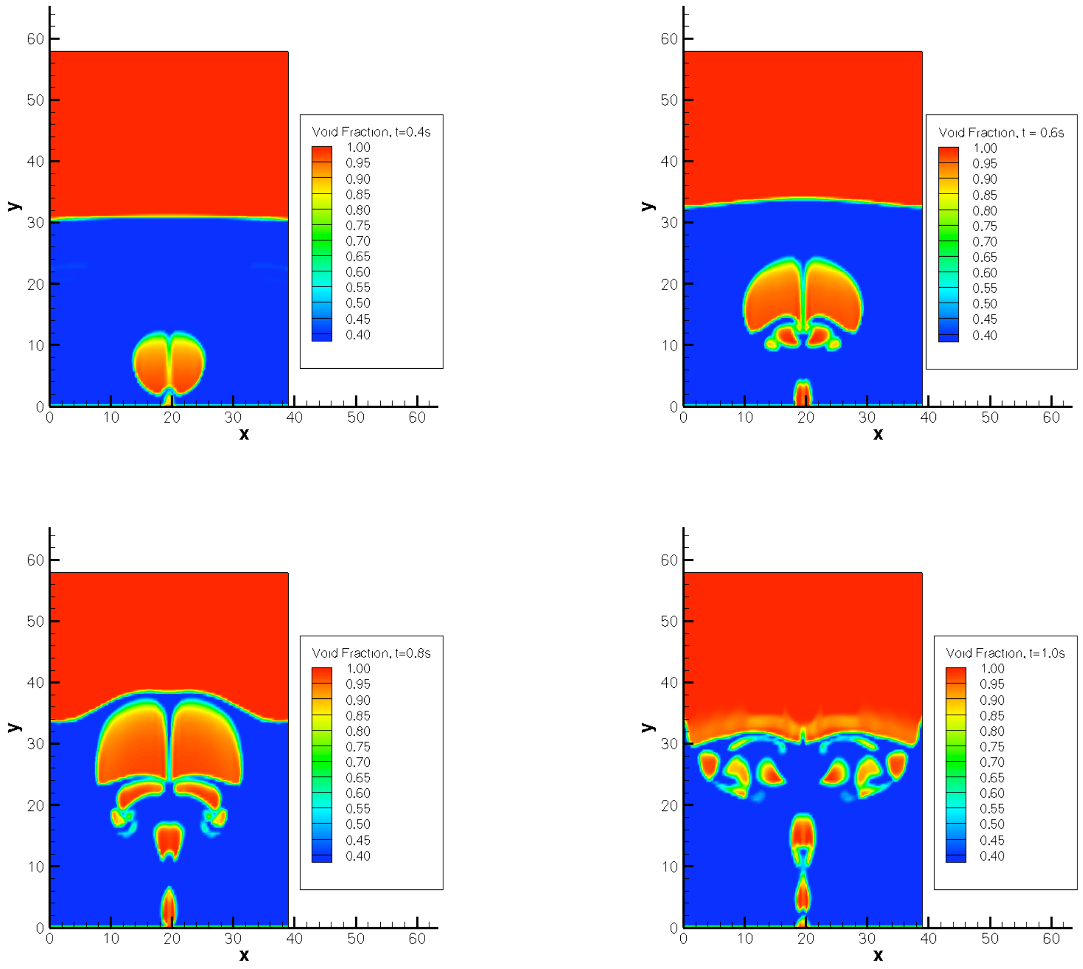


Figure 1: Contour plots of the void fraction at $t = 0.4$, 0.6, 0.8, and 1.0 seconds using MFIX to depict possible bubble locations.

Even if domain decomposition is possible, a simple increase in the number of POD modes used to approximate the solution is not guaranteed to accurately model the moving discontinuity. If the same fluidized bed is used and the void fraction is approximated in the usual way with POD time coefficients and basis functions (see (B)),

$$\varepsilon_g = \sum_{j=0}^{M} \alpha^{\varepsilon_g}(t_j)\phi_j^{\varepsilon_g}, \tag{3.1}$$

it can easily be shown that no ensemble of basis functions and time coefficients are sufficient to correctly model the flow. This is illustrated in Figure 2, where the maximum ensemble of basis functions is 320 for this particular case. Note the whited-out areas on the plots. These areas are unphysical regions where the void fraction is above 1 (fully gaseous) or below ≈ 0.39 (maximum solid packing limit). This is a classic example of Gibbs phenomena [33] observed on the approximated the solids correction equation (see the MFIX Numerics Guide for details [34]), which is becomes constrained ODE after the POD approxmation of the void fraction. This illustrates that no simple addition of basis functions or time coefficients can approximate the void fraction correctly. Therefore, a new method must be developed in order to model moving discontinuities this complex flow with POD.

A proposed method by Brenner *et al.* [35] to is to add a "discontinuity" basis function to the POD approximation. This discontinuity basis function would completely and accurately depict the moving discontinuity such that the ROM would apply to the entire flow field without any special treatment or prior knowledge of the shocked region needed. A possible form of the POD approximation would be:

$$\mathbf{u}(\mathbf{x}, t) = \sum_{j=0}^{m} \alpha_j^u(t)\phi(\mathbf{x}) + \sum_{k=0}^{n} \beta_k^u(t)\psi(\mathbf{x}), \tag{3.2}$$

where $\beta_k(t)$ and $\psi(\mathbf{x})$ are the discontinuity time coefficients and basis functions respectively. n would be the number of discontinuity modes required to model the shock. The discontinuity database would be collected from the full-order snapshots. This could be performed using several algorithms, with image processing techniques the most promising of these. Mathematical morphology, used as a solution post-

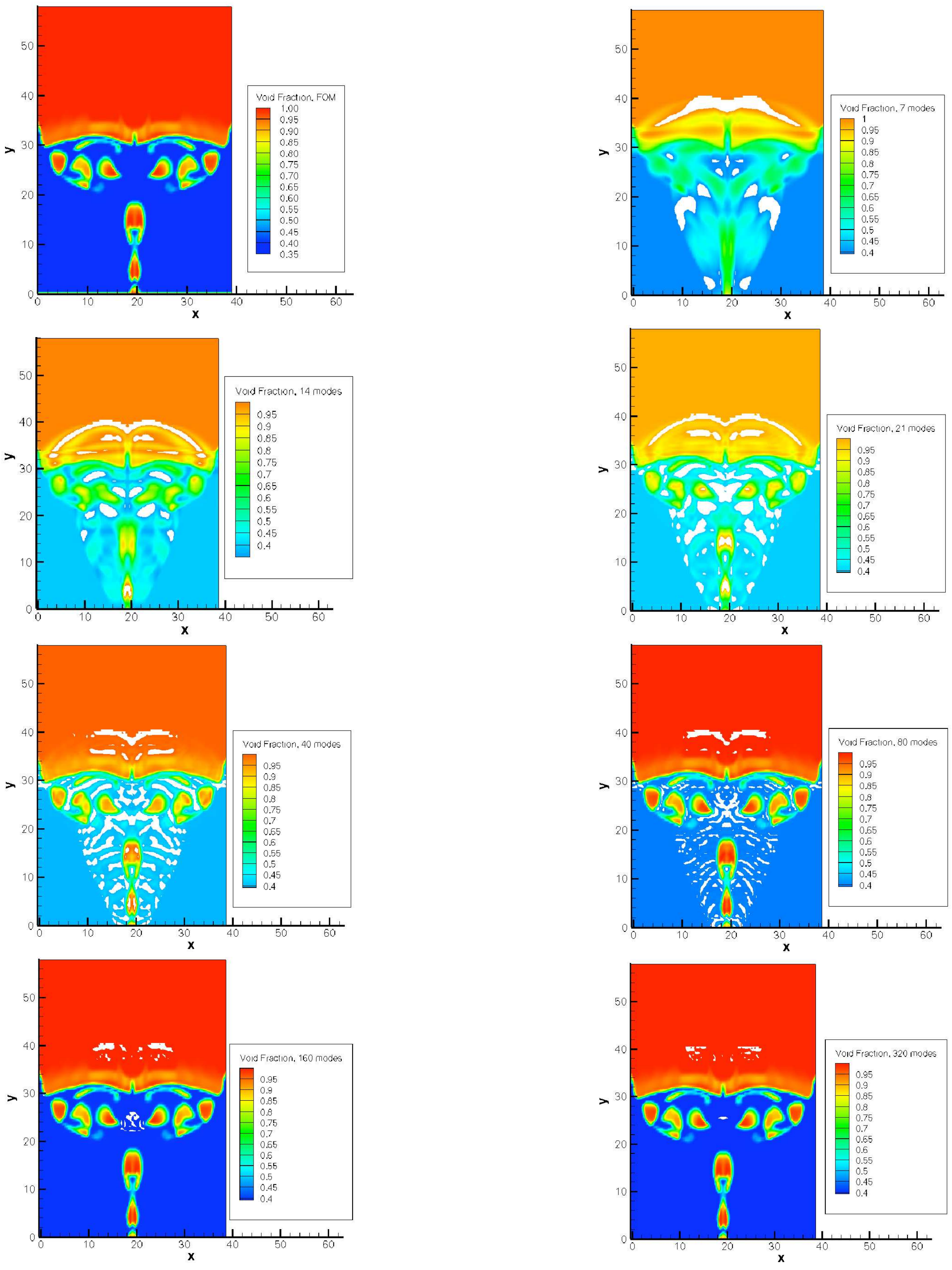


Figure 2: Contour plots of the POD reconstruction of the void fraction at $t = 1.0$ second for 7, 14, 21, 40, 80, 160, and 320 mode reconstructions compared to the full-order model.

processing technique, is the most promising of these algorithms. The next section will discuss the theory of mathematical morphology and the algorithms used to collect the discontinuity database.

C. Mathematical Morphology

This section presents mathematical morphology as a means of augmenting the POD basis functions. The history and theory of mathematical morphology is presented first. The algorithms developed for detecting and extracting moving discontinuities based on the blur-minimization technique [26] for highly-structured and general computational domains are presented.

1. Theory

Mathematical Morphology was originally developed as an imaging classification technique by Georges Matheron and Jean Serra at the École des Mines de Paris, France in the 1960's [20]. Serra had to describe and classify iron ore deposits from the iron mines of Lorraine, France [19]. At the same time Matheron was investigating porous media for a relationship between their geometry and their permeabilities [36]. Later, Serra and Matheron formalized mathematical morphology in terms of two basic operations, erosion and dilation, as well as operations derived from the two basic operations [19].

Mathematical morphology, herein referred to as morphology, is based heavily upon set theory, integral geometry, and lattice algebra [36]. Morphology is also infinite dimensional [20]. Morphology's basis on set theory makes the operations nonlinear [19]. This nonlinearity is what allows morphology to be applied to image analysis, which is the primary application of the theory [19, 20]. In fact, Serra and Matheron's investigation was of binary images describing the ore and the pores (1 representing the ore and 0 for its pores) [20]. This combination of objects, and in the case of gray or color scale images many more object overlays, requires the usage of Boolean algebra of set operations, which is also nonlinear.

As mentioned earlier, morphology can be expressed by two basic operations and various combinations of these to generate more complex operations. Morphology attempts to extract various features of interest located within the entire spatial domain, which can be a single or multidimensional image of any type (binary, grayscale, or color/multiscale) [35]. The discussion is limited herein to the two-dimensional case, but all morphological operations are easily extended to the three-dimensional case [23].

Feature extraction is performed with the usage of so-called structuring elements. These structuring elements are used to probe the image for the features of interest. There is no limit as to the shape or size a structuring element can take. The structuring elements are designed to mimic the feature of interest, such as rods for edges, circles for holes, or a spiral for spiral-type galaxies [20]. Structuring elements can also be adaptive, with varying size and shape based on the local qualities of the image [37]. Combinations of structuring elements can also be used to generate larger structuring elements [20]. Structuring elements are easily extended to higher dimensions as well. For example, amoeba adaptive elements have been successfully implemented for use in two and three dimension images [37].

The structuring element, denoted as b, is defined locally throughout the image, f, with coordinates (l, m), which define the local spatial extrema a set of elements occupies in the global coordinate system (i, j). For example, if a structuring element is a five node vertical rod-type element with a center defined as $(0, 0)$ in local coordinates, the element's coordinates in terms of the global coordinates would be: $(i, j - 2 \times m)$, $(i, j - m)$, (i, j), $(i, j + m)$, and $(i, j + 2 \times m)$. b initially assumes the values at these coordinates in f. The structuring elements are required to be continuous at the boundaries:

$$\left.\frac{\partial b}{\partial \mathbf{x}}\right|_{\delta\Omega} = 0, \tag{3.3}$$

such that no false edges are created at the boundaries due to the operations performed there.

Morphology is based upon two basic operations known as erosion and dilation. Each of these operations can be thought of as shifts of an image f. For the image point $f(i,j)$ on the structuring element $b(l,m)$, the image is shifted along the structuring element. If the structuring element is the vertical rod used in the previous paragraph, the image would be shifted up and down vertically along the structuring element. After these shifts, the maximum and minimum is computed at each point on the element. The maximum of these shifts, known as dilation, can be expressed as:

$$d(f(i,j)) = \max(f(i-l, j-m) + b(k,m)). \tag{3.4}$$

Erosion is the minimum of these shifts, given as:

$$e(f(i,j)) = \min(f(i+l, j+m) - b(k,m)). \tag{3.5}$$

Combinations of these two operations can lead to more sophisticated operations, such as opening and closing. The use of erosion and dilation can also be improved through the addition of non-morphological techniques. The next section will discuss augmenting the basic morphological operations with non-morphological operands.

2. Extensions For Feature Detection

The purpose of this section is to detail the non-morphological operations required for sufficient feature detection. While erosion (3.5) and dilation (3.4) are generally sufficient to perform higher-order operations and techniques, they alone cannot detect and extract certain types of edges, namely step-type edges [26]. In order to detect these types of edges, blurring must be employed. Blurring allows for the image points around a step edge to be a local maximum or minimum. Blurring can be defined as:

$$blur(b) = \frac{1}{L_{rs}} \sum_{\vec{P}=\vec{R}-D}^{\vec{R}+D} b(\vec{P}). \tag{3.6}$$

In (3.6), $\vec{R}$ represents the current coordinate on the stencil (l, m). L_{rs} is the reduced stencil size or length that blurring occurs over on the structuring element. D is the blurring displacement, defined as

$$D = \frac{L_{rs} - 1}{2}.$$

The reduced stencil length L_{rs} is set according to the size of the structuring element. Lee *et al.* [26] determined that $L_{rs} = 3$ was a sufficient size for blurring on a five element structuring element. The size of the reduced element must be at least two elements less than the size of the structuring element. This requirement is such that the edges are not over distorted by the blurring operation. If the structuring element is a vertical five node rod element, the blurring operation at the center node would include nodes $(l, m-1)$, (l, m) and $(l, m+1)$ in local coordinates. Blurring also has the added benefit of noise reduction by diffusing noise along the structuring element.

In addition to blurring, the algorithm requires two additional operations. These operations, referred as blur/erosion (be) and dilation/blur (db), act to eliminate obvious non-edges [26]. They are applied to the structuring elements after the image has been initially blurred with (3.6) and then dilated and eroded with Equations (3.4) and (3.5). These two operators can be defined as:

$$be(b) = blur(b) - e(blur(b)), \tag{3.7}$$

$$db(b) = d(blur(b) - blur(b). \tag{3.8}$$

In order to actually locate edges or other features, an edge strength index is required. To obtain this index, the maximums of the blur/erosion and dilation/blur operations (Equations (3.7) and (3.8) respectively) over the local structuring elements is required. For an ensemble of structuring elements, these maximums are defined as:

$$be_{max}(b(l,m)) = \max(be_1(b(l,m)), be_2(b(l,m)), ..., be_n(b(l,m))), \tag{3.9}$$

$$db_{max}(b(l,m)) = \max(db_1(b(l,m)), db_2(b(l,m)), ..., db_n(b(l,m))), \tag{3.10}$$

where n is the index of the last structuring element describing the local grid. Once these local maximums have been determined, the edge strength index can be computed at the image point $f(i,j)$ [26]:

$$Ind(f(i,j)) = \min(db_{max}(b(l,m)), be_{max}(b(l,m)), 0). \tag{3.11}$$

Logically, the higher the edge strength index is, the more likely an edge is located at the image point.

The edge strength index given by (3.11) marks the full edge, which can include more than the relevant number of points. This over emphasis of an edge is due to the previous operations acting on a non-binary image. An edge is defined as no more than two points for any image class [35]. Therefore, for non-binary images thresholding must be applied to extract edges and other features of interest. Thresholding is an image segmentation technique which will allow the edge definition to be satisfied. In this thesis global thresholding was investigated. For global thresholding, a maximum and minimum value α and β respectively, are set as limits. All values above or below these values are zeroed out, while those image points that lie between these thresholding limits are set equal to one. Global thresholding can therefore be thought of as a binary fix to the image. Global thresholding is formally defined as:

$$thres(f(i,j)) = \begin{cases} 1 & \text{if } \alpha \leq f(i,j) \leq \beta \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

The thresholding limits in (3.12) are image specific and are determined numerical experimentation.

3. Algorithm for Feature Detection

This section details the algorithm developed herein to extract moving discontinuities present in CFD solution databases. Based upon Lee *et al.*'s blur-minimization algorithm [26], the algorithm is extended for images of CFD solutions. The algorithm is presented below.

- Read in the images to be checked for discontinuities.
- Define the structuring elements. For this thesis, four rod-type structuring elements of five-unit lengths are used. The structuring elements are oriented in the following orientations: horizontal, vertical, and diagonal (45^o and 135^o with respect to the horizontal). This differs from Lee *et al.*'s algorithm [26] as they described four additional elements, but these were found to over-exaggerate the edges and were therefore not included in this algorithm.
- Blur the image using (3.6) with $L_{rs} = 3$ for the structuring elements.
- Compute the erosion and dilation of the blurred structuring elements using (3.4) and (3.5).

- Compute the combined difference operands with (3.7) and (3.8).
- Compute the maximums of the blur/erosion and dilation/blur operators with (3.9) and (3.10).
- Determine the edge strength given by (3.11).
- Apply global thresholding to the images with (3.12). This is added to the algorithm of [26] in order to satisfy the edge width constraint for grayscale images.

The algorithm, known as MEDts or Morphological Edge Detection for totally structured grids, solves for the edge strength and location for highly structured grids with nearly uniform grid spacing. It requires the input grid and flow field information. The field variable upon which the algorithm is applied is dependent upon what features are of interest. For example in two phase bubbling flows, the void fraction (ε_g) is the field variable of interest as it clearly defines where a bubble is within a fluidized bed. A void fraction of 1.0 indicates a fully gaseous region while a void fraction of less than ~ 0.9 indicates a region associated within the bed. More options for features of interest will be discussed later in the results section of this thesis. A sample input file for MEDts is given in Appendix F. Appendix B also presents an example application of the algorithm to a one-dimensional signal.

4. Application to General Domain Structures

This section presents the extension of the previous algorithm and mathematical morphology to image domains of general structures. In general, computational domains for fluid dynamic simulations are not highly structured. They may be structured with widely varying cell size, hybrid with structured and unstructured regions, or fully unstructured. For these types of computational domains, the algorithm presented in the previous section fails. It assumes that the domain is a highly structured, or pixelated, domain in which all neighbor elements are well known and distributed linearly throughout the image. Therefore, the image points must be related to one another and the structuring elements must be explicitly defined.

To relate all of the image points of f to one another for a general domain, only the connectivities of the cells to nodes are required. From these connectivities all inter-grid relations can be determined. This new algorithm requires the following relationships be known about the domain: cell face to cell edge, cell edge to edge face, cell edge to node, node to node, node to cell edge, and node to cell face. These dependencies are required to determine the value of each point on the structuring elements.

The structuring elements take on the same orientations as with the previous algorithm, but now the size and values at each location of the structuring elements must be determined. To set the size of the structuring element at each image point, the distance between it and the closest image neighbor is determined. This distance is used to limit the size of the structuring elements. With general domains, the structuring elements are allowed to vary from point to point as it is not known a priori the distance between grid points. By limiting the size to the minimum distance between neighbor nodes, the chance of false feature detection is minimized. Large structuring elements, *ie.* those not defined based on the minimum distance, could stretch across features not locally defined. However, the minimum distance may not be enough to satisfy the minimization of false features. In some cases, an upper limit on the size of the structuring element must be applied to force the structuring elements to remain within the local vicinity of the image point. This "local" area can generally be defined as no more than two generations of cells out from an image point. The inverse is also possible where the structuring element does not reach out far enough from the image point. This requires a lower limit on the size of the structuring element to force coverage of the local area. Both of these situations are covered in this thesis. Regardless of how the distance between nodes on the structuring element is defined, it is either is added or subtracted from the actual coordinates of the current image point in the appropriate directions for the structuring element to be fully defined in the spatial domain.

With the structuring elements defined in the spatial domain, the values that the structuring element assumes at each node must be determined. To compute the value at each node location, isoparametric mappings are utilized [38]. This requires knowing where in space, or on which face, the node is located. An exact location is required as the computation of the isoparametric shape functions is geometrically-dependent. This will be discussed more later. Determination of the structuring element node location is rather simple, as each connected face to the central node is searched, and each subsequent generation of nodes out to four generations is searched as well. An example of the possible searches for structuring element definition is shown in Figure 3. In Figure 3, the center node is red, with each generation of nodes marked with a different color. Each subsequent generation of faces searched are denoted by a lighter shade of gray. The search is carried out four generations of nodes to assure that the entire local area has been searched. A node is located in space when:

- A node on the structuring element is also a node on a face. Collocated structuring element nodes assume the exact value of the image and no approximation is required.

- A node is located on the edge of a face. Linear interpolation of the structuring element is all that is required to determine the value for the node.

- The ratio of sum of the areas generated by the addition of the node onto the face and the actual face area is equal approximately equal to one. This is expressed as:

$$\frac{\sum_{i=1}^{N} a_N}{a_{face}} \approx 1, \tag{3.13}$$

  where N is the number of faces generated by adding the point to the face. $N = 3$ for triangular faces, and $N = 4$ for quadrilateral faces. a_n is the area of each generated face while a_{face} is the actual area of the face. The areas are computed by the cross product:

$$a = \frac{1}{2}(\vec{p} \times \vec{q}),$$

  where $\vec{p}$ and $\vec{q}$ are the diagonals of a quadrilateral or the vertices of a triangle.

The structuring elements are still required to be continuous at the boundaries as given by (3.3).

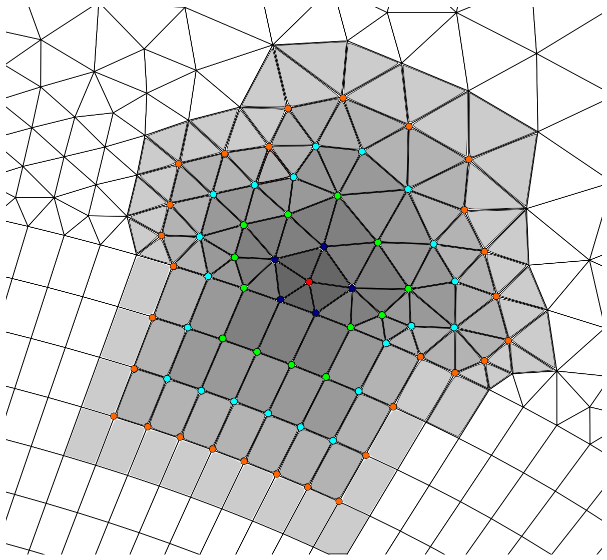


Figure 3: Example of generational search path for structuring element definition.

As of now the location of the structuring elements and the exact location in space of their nodes is known. When the nodes are located on faces of the domain, isoparametric mappings are utilized to determine the physical values of the variables at the node. Isoparametric mappings use shape functions to determine the value a point located in the domain of the face assumes. These shape functions can be thought of as transformations between the actual face and a master element [38]. This transformation ensures that the field variable vectors across the face are continuous as the master element is required to be continuous [38]. Two types of master elements were employed in this thesis, three-node triangular and four-node quadrilateral elements. These elements, along with the mappings from the original faces to the master elements, are shown in Figures 4 and 5. The shape functions for a three-node triangular element are given as:

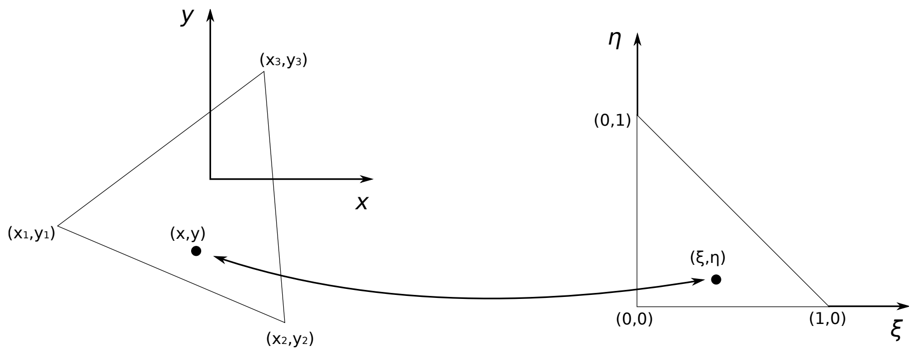


Figure 4: Isoparametric mapping from a general triangle to the master element.

$$\Psi_1(\xi,\eta) = 1 - \xi - \eta \tag{3.14}$$

$$\Psi_2(\xi,\eta) = \xi \tag{3.15}$$

$$\Psi_3(\xi,\eta) = \eta, \tag{3.16}$$

where ξ and η are the coordinates in the master element domain.

For a four-node quadrilateral element, the shape functions are defined as:

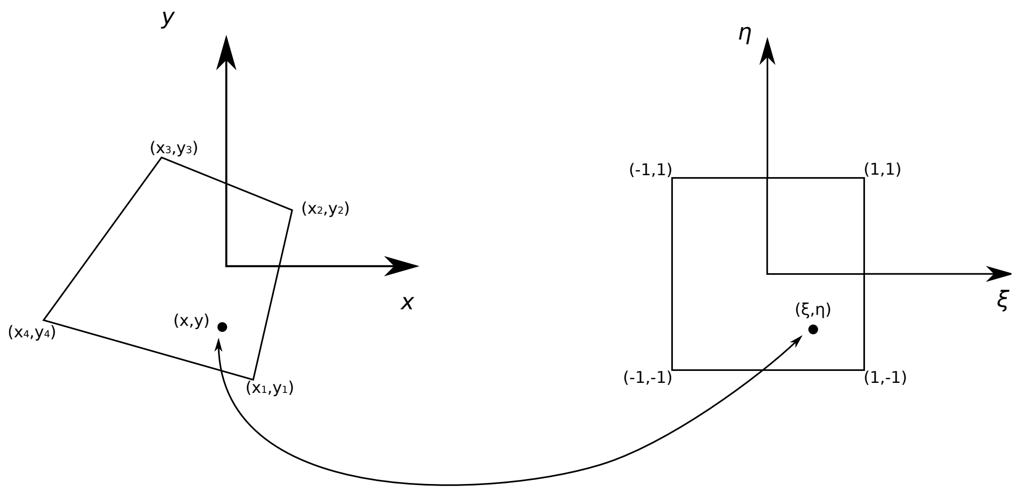


Figure 5: Isoparametric mapping from a general quadrilateral to the master element.

$$\Psi_1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta) \tag{3.17}$$

$$\Psi_2(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta) \tag{3.18}$$

$$\Psi_3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta) \tag{3.19}$$

$$\Psi_4(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta). \tag{3.20}$$

(3.14) - (3.16) and (3.17) - (3.20) can be related to the spatial domain by the following equations:

$$x = \sum_{i=1}^{N} x_i \Psi_i \tag{3.21}$$

$$y = \sum_{i=1}^{N} y_i \Psi_i \tag{3.22}$$

where x and y are the coordinates of the structuring element point and x_i and y_i are the coordinates of the points that define the face. To determine the shape functions, we substitute their definitions into (3.21) and (3.22) and solve for ξ and η. For the purpose of this thesis, the shape functions for the quadrilateral element (Equations (3.17) - (3.20)) are substituted into (3.21) and (3.22). The methodology is similar for triangular elements. The substitution yields:

$$\begin{aligned}4x-(x_1+x_2+x_3+x_4)=&\xi(-x_1+x_2+x_3-x_4)+\eta(-x_1-x_2+x_3+x_4)\\&+\xi\eta(x_1-x_2+x_3-x_4)\end{aligned}\tag{3.23}$$

$$\begin{aligned}4y-(y_1+y_2+y_3+y_4)=&\xi(-y_1+y_2+y_3-y_4)+\eta(-y_1-y_2+y_3+y_4)\\&+\xi\eta(y_1-y_2+y_3-y_4)\end{aligned}\tag{3.24}$$

Equations (3.23) and (3.24) are nonlinear but easily solved using the Multivariate Newton-Raphson (MNR) method. The system for the MNR is:

$$\delta\mathbf{X}=[J]^{-1}\mathbf{R},\tag{3.25}$$

where

$$\delta\mathbf{X}=\begin{Bmatrix}\delta\xi\\ \delta\eta\end{Bmatrix},$$

$$\mathbf{R}=\begin{Bmatrix}F_1(\xi,\eta)\\ F_2(\xi,\eta)\end{Bmatrix},$$

$$[J]^{-1}=\frac{1}{|J|}\begin{bmatrix}\frac{\partial F_2(\xi,\eta)}{\partial\eta} & -\frac{\partial F_1(\xi,\eta)}{\partial\eta}\\ -\frac{\partial F_2(\xi,\eta)}{\partial\xi} & \frac{\partial F_1(\xi,\eta)}{\partial\xi}\end{bmatrix}$$

$$|J|=\frac{\partial F_1(\xi,\eta)}{\partial\xi}\frac{\partial F_2(\xi,\eta)}{\partial\eta}-\frac{\partial F_2(\xi,\eta)}{\partial\xi}\frac{\partial F_1(\xi,\eta)}{\partial\eta}.$$

$[J]^{-1}$ is the inverse Jacobian matrix, $|J|$ is the determinant of the Jacobian matrix,

$\delta\mathbf{X}$ is the solution vector, and $\mathbf{R}$ is the equation vector where $F_1(\xi, \eta)$ and $F_2(\xi, \eta)$ are (3.23) and (3.24) for the above system of equations. (3.25) is iterated until convergence is reached, that is when the change in $\delta\mathbf{X}$ is reasonably small. ξ and η are computed from:

$$\xi^k = \xi^{k-1} + \delta\xi \tag{3.26}$$

$$\eta^k = \eta^{k-1} + \delta\eta, \ , \tag{3.27}$$

where k is the current iteration. ξ and η are also limited $\leq |1|$. This is a consequence of the mapping to the master element. This limitation also requires that the transformed points lie exactly within the face, otherwise the transformation will not be invertible, that is the determinant of the Jacobian is negative [39]. Another consequence of incorrectly determining the location of a point is over- or under-predicting its value. Since no information on the surrounding grid is known during the mapping process, there is no guarantee that the solution is smooth outside of the element. Once ξ and η are known, the shape functions (Equations (3.14) - (3.16) and Equations (3.17) - (3.20)) can be determined. These shape function are used to compute the value at the structuring element node. The value of the field variable at the structuring element node is given by:

$$q = \sum_{i=1}^{N} q_i \Psi_i, \tag{3.28}$$

where q is the field variable at the node, q_i is the field variable at the corners of the face, Ψ_i are the shape functions, and N is the number of edges for the face.

The algorithm for general domains MEDuns, or Morphological Edge Detection for unstructured domains, solves for the edge strength and location for general domains. It requires the input grid and flow field information. As before with MEDts, MEDuns can detect features present within any field variable. The structure of the algorithm is the same as with MEDts, with the exception of the definition of the structuring elements. Sample input files for MEDuns are given in Appendices G and H.

D. Summary

This chapter presented the motivation and methodology for feature extraction as a means of augmenting POD basis functions used to generate reduced-order models of flows with moving discontinuities. Previous work on ROM generation with POD for flows with moving discontinuities was discussed. Bubbling in fluidized beds was given as the initial motivation for developing the feature extraction algorithm. The feature extraction algorithm based on mathematical morphology was presented. The algorithm and the use of mathematical morphology was extended to unstructured spatial domains via isoparametric mappings. The following chapter presents the full-order models used in this thesis.

CHAPTER IV

PHYSICAL MODELS

A. Introduction

This chapter presents the physical models that describe single and multiphase flows. The first section presents the physical model used for single phase flows. The second section presents the multiphase physical model.

B. Single Phase Physical Model

This section presents the physical model used herein to simulate single phase flows. The single phase flows investigated are modeled using the unsteady, compressible, and viscous Navier-Stokes equations for mass, momentum, and energy. The mass conservation equation is expressed as [40]:

$$\frac{\partial}{\partial t}\int\limits_V \rho \cdot dV + \oint\limits_S \rho \cdot (\vec{v} \cdot \vec{n}) \cdot dS = 0. \tag{4.1}$$

V, S, and $\vec{n}$ are the cell control volume, area of the cell faces, and unit normal vector to the cell faces, respectively. ρ and $\vec{v}$ are the flow density and velocity, respectively. The first term of (4.1) is the time rate of change of total mass within the cell volume. The second term represents the mass flow through the cell faces.

The conservation of momentum equation can be written as [40]:

$$\frac{\partial}{\partial t}\int\limits_V \rho \cdot \vec{v} \cdot dV + \oint\limits_S \rho \cdot \vec{v} \cdot (\vec{v} \cdot \vec{n}) \cdot dS = \int\limits_V \rho \cdot \vec{g} \cdot dV + \oint\limits_S (-p \cdot \vec{n} + \tau \cdot \vec{n}) dS. \tag{4.2}$$

$\vec{g}$, p, and τ are the gravity force on a cell volume, pressure imposed on the cell faces, and the normal and shear stresses acting on the fluid, respectively. The first term of (4.2) represents the temporal variation of momentum. The second term is the convective term describing the momentum transfer across a boundary of a cell volume. The third and forth terms in (4.2) represent the body and surface forces, respectively.

The conservation of energy equation is expressed as [40]:

$$\frac{\partial}{\partial t}\int_V \rho \cdot E \cdot dV + \oint_S \rho \cdot E \cdot (\vec{v} \cdot \vec{n}) \cdot dS = \int_V (\rho \cdot \vec{g} \cdot \vec{v} + q_h) \cdot dV + \oint_S k \cdot (\nabla T \cdot \vec{n}) \cdot dS - \oint_S p \cdot (\vec{v} \cdot \vec{n}) \cdot dS + \oint_S (\tau \cdot \vec{n}) \cdot \vec{n} \cdot dS. \quad (4.3)$$

E and q_h are the total energy per unit mass and heat source, respectively. The first term of (4.3) is the time rate of change of energy per unit volume. The second term is the convective term describing energy transfer across a cell face. The third term is the heat sources and work done on the cell volume. The fourth term is the dissipative term describing the diffusion of heat due to thermal conduction. The final two terms represent the work done on the element by pressure and shearing forces.

The total energy per unit mass E in (4.3) is given as:

$$E = e + \frac{|\vec{v}|^2}{2}, \quad (4.4)$$

where e is the internal energy and $\vec{v}$ is the velocity vector. The total enthalpy per unit mass H can be expressed in terms of E by:

$$H = h + \frac{|\vec{v}|^2}{2} = E + \frac{p}{\rho}, \quad (4.5)$$

where h, p, and ρ is the internal enthalpy, pressure, and density, respectively. e and

h can be expressed in terms of temperature and specific heats:

$$e = c_v \cdot T, \tag{4.6}$$

$$h = c_p \cdot T, \tag{4.7}$$

where T is the temperature and c_v and c_p are the specific heats at constant volume and constant pressure respectively.

With (4.5), (4.3) can be expressed in terms of the enthalpy H:

$$\begin{aligned}\frac{\partial}{\partial t}\int_V \rho \cdot E \cdot dV + \oint_S \rho \cdot H \cdot (\vec{v} \cdot \vec{n}) \cdot dS = \int_V (\rho \cdot \vec{g} \cdot \vec{v} + q_h) \cdot dV \\ + \oint_S k \cdot (\nabla T \cdot \vec{n}) \cdot dS + \oint_S (\tau \cdot \vec{n}) \cdot \vec{n} \cdot dS.\end{aligned} \tag{4.8}$$

The media is assumed to a an ideal gas that obeys the ideal gas law:

$$P = \rho \cdot R \cdot T, \tag{4.9}$$

where R is the gas constant. R can be related to the specific heats:

$$R = c_p - c_v.$$

The ratio of specific heats is $\gamma = \frac{c_p}{c_v}$. The pressure p can also be expressed in terms of the total energy E:

$$p = (\gamma - 1)\rho \left[E - \frac{|\vec{v}|^2}{2}\right]. \tag{4.10}$$

Turbulence effects are modeled using the two-equation eddy-viscosity Shear Stress Transport (SST) model of Menter [41]. This model introduces two additional transport equations: turbulent kinetic energy and specific dissipation of turbulence. The turbulent eddy viscosity μ_T is calculated as the ratio between the turbulent kinetic energy and turbulent specific dissipation:

$$\mu_T = \rho \cdot \frac{k}{\omega}, \tag{4.11}$$

where k and ω are the turbulent kinetic energy and turbulent specific dissipation, respectively.

Turbulent kinetic energy transport can be expressed as [41]:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_j}(\rho v_j k) = \frac{\partial}{\partial x_j}\left[(\mu_L + \sigma_k \mu_T) \cdot \frac{\partial k}{\partial x_j}\right] + \tau_{ij} \cdot S_{ij} - \beta^* \rho \omega k. \tag{4.12}$$

In (4.12) μ_L is the molecular viscosity. τ_{ij} and S_{ij} are the turbulent stresses and strain-rate tensor, respectively. β^* is a turbulent model parameter. The first term of (4.12) represents the time rate of change of turbulent kinetic energy in the cell volume. The second term is the convective term representing the transfer of turbulent kinetic energy across a cell face. The third term is the conservative turbulent kinetic energy diffusion. The last two terms in (4.12) represent the production and dissipation of turbulent kinetic energy, respectively.

The specific dissipation of turbulence transport equation is given as:

$$\begin{aligned}\frac{\partial \rho \omega}{\partial t} + \frac{\partial}{\partial x_j}(\rho v_j \omega) =& \frac{\partial}{\partial x_j}\left[(\mu_L + \sigma_w \mu_T) \cdot \frac{\partial \omega}{\partial x_j}\right] + \frac{C_w \rho}{\mu_T} \tau_{ij} \cdot S_{ij} \\ & - \beta \rho \omega^2 + 2(1 - f_1) \cdot \frac{\rho \sigma_{w2}}{w} \cdot \frac{\partial k}{\partial x_j} \cdot \frac{\partial \omega}{\partial x_j},\end{aligned} \tag{4.13}$$

where C_w and β are turbulent model parameters. f_1 is the shear model blending function. It blends the model coefficients of the $k - \omega$ model in the boundary layer with the model coefficients of the transformed $k - \epsilon$ model in the free-shear layer and freestream. The first term of (4.13) represents the time rate of change of specific dissipation of turbulence of the control volume. The second term is the convective term representing the transfer of specific dissipation of turbulence across a boundary of the control volume. The third term is the conservative specific dissipation of turbulence diffusion. The fourth and fifth term of (4.13) represent the production and dissipation of specific turbulent diffusion, respectively. The last term of (4.13) represents the turbulent cross diffusion.

The blending function f_1 is given as [41]:

$$f_1 = \tanh(arg_1^4), \tag{4.14}$$

where arg_1 is given by [41]:

$$arg_1 = \min\left[\max\left(\frac{\sqrt{k}}{0.09\omega d}, \frac{500\mu_L}{\rho\omega d^2}\right), \frac{4\rho\sigma k}{CD_{kw}d^2}\right]. \tag{4.15}$$

In (4.15) d is the distance from the cell to the nearest wall. CD_{kw} is the positive part of the cross diffusion term in (4.13) and is given as [41]:

$$CD_{kw} = \max\left(2 \cdot \frac{\rho\sigma_{w2}}{w} \cdot \frac{\partial k}{\partial x_j} \cdot \frac{\partial \omega}{\partial x_j}, 10^{-20}\right). \tag{4.16}$$

The turbulent model constants are calculated from the blending function (4.14) and the blending equation, given as [41]:

$$\phi = f_1 \cdot \phi_q + (1 - f_1) \cdot \phi_2, \tag{4.17}$$

where ϕ is the blending parameter. ϕ is representative of σ_k, σ_ω, β, and C_w. The coefficients for the $k - \omega$ model are: $\sigma_{k_1} = 0.85$, $\sigma_{\omega_1} = 0.5$, $\beta_1 = 0.075$, and $C_{w_1} = 0.533$ [42]. The coefficients for the $k - \epsilon$ model are given as: $\sigma_{k_2} = 1.0$, $\sigma_{\omega_2} = 0.856$, $\beta_2 = 0.0828$, and $C_{w_2} = 0.44$ [42].

C. Multiphase Physical Model

This section presents the physical model for transport phenomena in fluidized beds. Transport phenomena in fluidized beds are described first. The governing equations for multiphase transport phenomena are described next. A description of the boundary conditions for isothermal and non-isothermal multiphase flows follows.

1. Transport Phenomena in Fluidized Beds

Figure 6 shows the typical behavior displayed in a fluidized bed. The fluidized bed is typically contained within a vessel, such as a reactor. This vessel contains solid particles resting over a plate through which a gas is injected. At low inlet gas velocities, as shown in Figure 6a [9], the gas fills the voids present in the solid particle bed while the bed remains packed. As the velocity is increased to a critical value, known as minimum fluidization, the solid particles will display fluid-like properties as shown in Figure 6b [9]. A continued rise in the injected gas velocity can produce bubbling, inducing mixing within the bed [32]. If the gas velocity is allowed to continue to increase, particle terminal velocity is eventually reached. In this case, the particles will be taken out of the bed as shown in Figure 6c [9].

2. Governing Equations

The mass and momentum equations for non-reacting non-isothermal transport phenomena in fluidized beds are presented below.

- Gas Continuity Equation

$$\frac{\partial \varepsilon_g \rho_g}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \vec{u_g}) = 0 \tag{4.18}$$

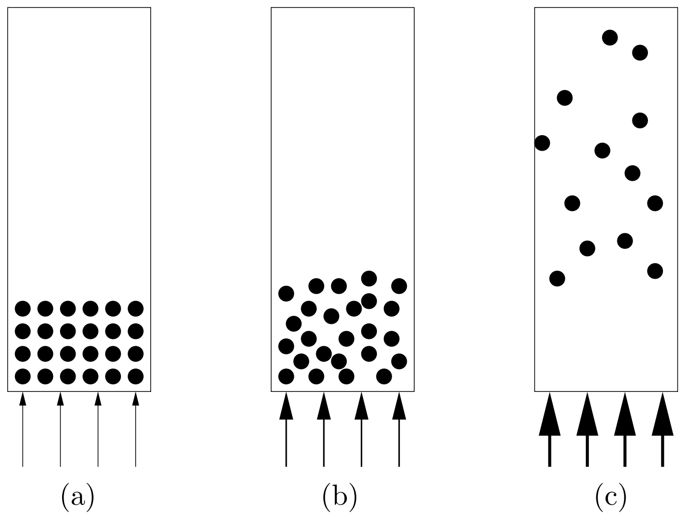


Figure 6: Typical behavior of a fluidized bed.

- Solid Continuity Equation

$$\frac{\partial \varepsilon_s \rho_s}{\partial t} + \nabla \cdot (\varepsilon_s \rho_s \vec{u_s}) = 0 \tag{4.19}$$

- Gas Momentum Equation

$$\frac{\partial \varepsilon_g \rho_g \vec{u_g}}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \vec{u_g} \vec{u_g}) = -\varepsilon_g \nabla p_g - \nabla \cdot \bar{\bar{\tau}}_g + \varepsilon_g \rho_g \vec{g} + F_{gs}(\vec{u_s} - \vec{u_g}) \tag{4.20}$$

- Solid Momentum Equation

$$\frac{\partial \varepsilon_s \rho_s \vec{u_s}}{\partial t} + \nabla \cdot (\varepsilon_s \rho_s \vec{u_s} \vec{u_s}) = -\varepsilon_s \nabla p_g - \nabla p_s - \nabla \cdot \bar{\bar{\tau}}_s + \varepsilon_s \rho_s \vec{g} - F_{gs}(\vec{u_s} - \vec{u_g}) \tag{4.21}$$

In the mass and momentum equations, ε, ρ and $\vec{u}$ represent the void fraction, density, and velocity vector, respectively. Subscripts g and s represent the gas and solid phases, respectively. The momentum equations are closed by the gas-phase viscous stress $\bar{\bar{\tau}}_g$, solid-phase granular stress $\bar{\bar{\tau}}_s$, solid pressure p_s, and gas-solid drag force F_{gs}. The constitutive models for these terms are given in Appendix A, with a more thorough discussion given by Syamlal [34] and Syamlal *et al.* [43].

The gas and solid phase scalar energy balances describing the respective phase temperature fields for non-reacting fluidized beds, neglecting radiation, are given by:

- Gas Energy Equation

$$\varepsilon_g \rho_g C_{pg} \left[\frac{\partial T_g}{\partial t} + \vec{u_g} \cdot \nabla T_g \right] = -\nabla \vec{q_g} + \gamma_g (T_s - T_g) \tag{4.22}$$

- Solids Energy Equation

$$\varepsilon_s \rho_s C_{ps} \left[\frac{\partial T_s}{\partial t} + \vec{u_s} \cdot \nabla T_s \right] = -\nabla \vec{q_s} + \gamma_s (T_s - T_g) \tag{4.23}$$

where $\vec{q_l}$ and γ_l represent the conductive heat flux and the gas-solids heat transfer coefficient, respectively. T and C_{pl} represent the temperature and specific heat at constant pressure of either phase, respectively. The subscript l designates either gas or solid phase in the previous terms. The gas phase is modeled as a gas obeying the ideal gas law:

$$\rho_g = \frac{p_g M}{R T_g} \tag{4.24}$$

where R and M are the universal gas constant and the molecular mass of the gas, respectively. The specific heats C_{ps} and C_{pg} are modeled as those for ash and air, respectively.

Granular energy, described by kinetic theory, is defined as the specific kinetic energy of the random fluctuating component of the velocity of the solids phase, U_s [43]:

$$\frac{3}{2}\Theta = \frac{1}{2}\langle U_s^2 \rangle. \tag{4.25}$$

The granular energy balance equation for a single solids phase in a fluidized bed is given by:

$$\frac{3}{2}\varepsilon_s \rho_s \left[\frac{\partial \Theta}{\partial t} + \vec{u_s} \cdot \nabla \Theta \right] = \nabla(\kappa_s \nabla \Theta) + \bar{\bar{\tau}}_s \nabla \vec{u_s} + \prod_s -\varepsilon_s \rho_s J_s \tag{4.26}$$

where Θ is the granular energy. $\nabla(\kappa_s \nabla\Theta)$ represents the diffusive flux of granular energy. $\varepsilon_s \rho_s J_s$ is the rate of granular energy dissipation, with J_s representing the collisional dissipation between particles. $\prod_s$ is the exchange of granular energy between gas and solid phases. The constitutive equations for the granular energy equation is given in Appendix A, with a more thorough discussion given by Syamlal *et al.* [43] and Benyahia *et al.* [44].

3. Boundary Conditions

The basic geometry of a 2-D fluidized bed is given in Figure 7. The lower region of the reactor or vessel is packed with solid particles. The solid particles are initially modeled at rest. The bottom of the reactor has a plate, known as a distributor, with multiple slots or holes through which gas is injected. The gas injection is modeled either as a uniform or non-uniform distribution. The non-uniform injection is generally comprised of a uniform flow in the outer regions of the bed and a central jet at a higher velocity. For non-isothermal flows, the injected gas is modeled with a higher temperature than the gas present in the bed. The walls are modeled as no-slip walls. The top of the vessel is modeled as a constant pressure boundary.

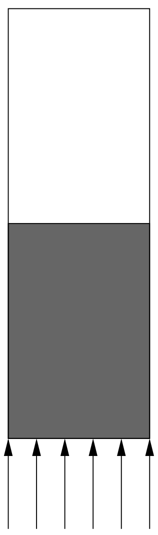


Figure 7: Basic geometry and boundary conditions.

D. Summary

This chapter presented governing PDEs for the physical models used to simulate single and multiphase flows. The rudimentary boundary conditions for multiphase flows were also presented. Transport phenomena in fluidized beds was also described. The next chapter will present the full-order numerical models for single and multiphase flows.

CHAPTER V

FULL-ORDER MODELS

A. Introduction

This chapter presents the full-order numerical models used in the simulations and as the basis for the reduced-order models. The first section presents the single phase model. The second section presents the multiphase model.

B. Single Phase Model

This section presents the full-order model used herein to simulate single phase flows. The single phase numerical model solves the compressible, viscous, and unsteady Reynolds-Averaged Navier-Stokes (RANS) equations (Equations (4.1) - (4.3), with averaging applied). Turbulence is modeled using the two equation $k-\omega$ Shear-Stress Transport (SST) turbulence model proposed by Menter [41] (Equations (4.12) and (4.13)). Both the RANS and SST equations are solved using an explicit time marching scheme with implicit residual smoothing. The numerical model, known as UNS3D (Unstructured 3-Dimensional Flow Solver), was developed by Han and Cizmas [45]. UNS3D is written in FORTRAN 90 with support for 32- and 64-bit processors on Linux/Unix platforms. UNS3D is discretized in three dimensions for Cartesian coordinates. UNS3D can support structured, unstructured, or hybrid computational domains. UNS3D is advanced in time using a Runge-Kutta time marching scheme. The solution output of UNS3D consists of time-dependent density, pressure, velocities, temperature, viscosity, and turbulent parameters. With UNS3D only used for simulation purposes and not used to derive a POD-based ROM, a full discussion of the discretization schemes and boundary condition implementations are not provided here. For a more thorough discussion, the dissertations of Kim [46] and Gargoloff [42] as well as the paper by Han and Cizmas [45] are recommended.

C. Multiphase Model

This section presents the full-order model used to simulate multiphase transport phenomena in fluidized beds. The multiphase numerical model solves the governing PDEs (4.18) - (4.26) using a time marching, fully implicit scheme. The algorithm, known as MFIX (Multiphase Flow with Interphase eXchanges), was developed at the U.S. Department of Energy's National Energy Technology Laboratory [43]. MFIX is written in free-format FORTRAN 77 and has support for 32- and 64-bit processors on Linux/Unix platforms. MFIX can be discretized in two or three dimensions for Cartesian or cylindrical coordinate uniform or non-uniform grids. MFIX can enforce free-, partial-, or no-slip conditions at the wall. It can also enforce adiabatic or isothermal walls for heat transfer purposes. MFIX solves numerically solves (4.18) - (4.26). The solution output of MFIX consists of time-dependent pressure, gas and solid velocities, densities, and temperatures, void fraction, and granular energy. The following sections discuss the spatial discretization and the discretization of the energy, granular energy, and z-component of velocity, which are relevant to this study.

1. Discretization of the Spatial Domain

This section describes the discretization of the spatial domain for MFIX. MFIX uses two types of grids to discretize the spatial domain, one for scalars and one for velocities. The arrangement of the spatial domains is shown in Figure 8 for two dimensions. The arrangement of the spatial domains is shown in Figure 9 for three dimensions.

Scalars are stored at the cell centers. Velocity vectors are stored at the cell faces. MFIX uses two different grids to avoid unphysical solutions in the velocity and pressure fields; if the velocity and pressure were solved on the same grid, a checkerboard pattern could develop in the pressure field [34].

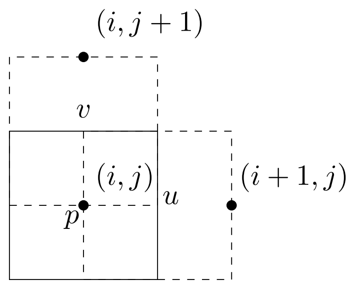


Figure 8: Two-dimensional grid arrangement in MFIX.

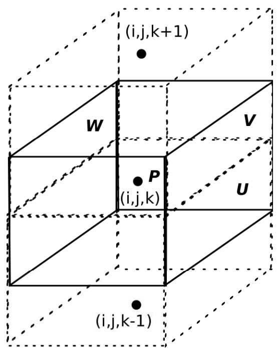


Figure 9: Three-dimensional grid arrangement in MFIX.

Figure 10 shows a two-dimensional control volume used for the scalar transport equations. Point P is the center of the control volume. Points N, S, E, and W are the centers of the north, south, east, and west neighbor control volumes. n, s, e, and

w are the north, south, east, and west faces of the control volume. All scalar values are stored at the centers of the control volumes.

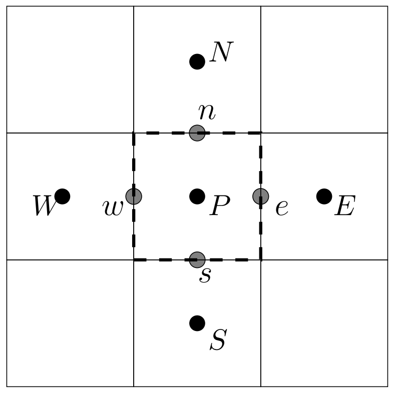
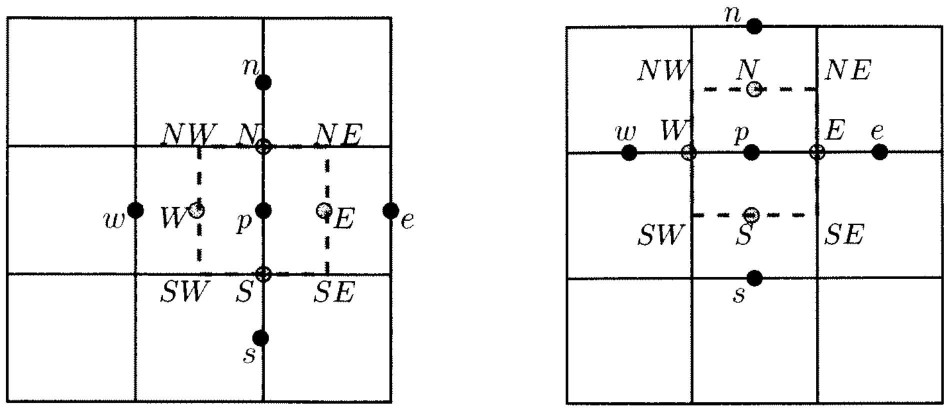


Figure 10: Control volume for scalar equations.

Figures 11(a) and 11(b) show the two-dimensional staggered grid arrangements used to solve the x- and y-momentum equations respectively. p is the center of the control volume. n, s, e, and w are the centers of the north, south, east, and west neighbor control volumes. N, S, E, and W are the north, south, east, and west faces of the control volume. NE, NW, SE, and SW denote the northeast, northwest, southeast, and southwest corners of the control volume. The velocities are stored at the cell faces. The staggered grid arrangement for the z-momentum is arranged in a similar manner to the staggered grid of Figure 9 as well as Figures 11(a) and 11(b).



(a) x-momentum balance (b) y-momentum balance

Figure 11: Control volume for momentum balance.

2. Discretization of the Governing Equations

This section describes the discretization of the energy and granular energy balance equations in two dimensions and the z-momentum balance equation in three dimensions in MFIX. The granular energy equation is also discretized in three dimensions to show the effect of the third dimension on the scalar equations. The remaining discretized forms of the governing equations are given in the MFIX Documentation Numerical Technique [34]. These equations are not discussed here as they were previously developed into reduced-order models by Yuan [1] and Richardson [2]. Though the energy equation was previously developed into a reduced-order model by Richardson [2], it is reviewed here as its implementation as a reduced-order model was improved in this study. All equations are discretized using the control volume method.

The energy equation is discretized as follows [34]:

$$a_P(T_m)_p = \sum_{nb} a_{nb}(T_m)_{nb} + b_P. \tag{5.1}$$

T represents the temperature for m phase (gas or solid). p and nb represent the contribution of the center and neighbor cell centers, respectively. The coefficients of the energy equation, a and b, are given for non-reacting flows, with no radiation effects, below.

$$a_E^{T_m} = D_e - \frac{(\xi^{T_m})_e}{2}(\varepsilon_m\rho_m)_E((C_{p,m})_P + (C_{p,m})_E)(u_m)_e A_E \tag{5.2}$$

$$a_W^{T_m} = D_w + \frac{(\xi^{T_m})_w}{2}(\varepsilon_m\rho_m)_W((C_{p,m})_P + (C_{p,m})_W)(u_m)_w A_W \tag{5.3}$$

$$a_N^{T_m} = D_n - \frac{(\xi^{T_m})_n}{2}(\varepsilon_m\rho_m)_N((C_{p,m})_P + (C_{p,m})_N)(v_m)_n A_N \tag{5.4}$$

$$a_S^{T_m} = D_s - \frac{(\xi^{T_m})_s}{2}(\varepsilon_m\rho_m)_S((C_{p,m})_P + (C_{p,m})_S)(v_m)_s A_S \tag{5.5}$$

$$a_P^{T_m} = -\left(\sum_{nb} a_{nb}^{T_m} + \frac{(\varepsilon_m\rho_m)^o C_{p,m}\Delta V}{\Delta t}\right) \tag{5.6}$$

$$b_P^{T_m} = -\left(\frac{(\varepsilon_m\rho_m)^o C_{p,m}\Delta V}{\Delta t}(T_P^o)_m\right) \tag{5.7}$$

In (5.2) - (5.5), D_{nb}, where nb represents either the north, south, east, or west face (denoted by n, s, e,and w respectively), is the diffusion. The diffusion is modeled as:

$$D_{nb} = \frac{(avg_h(K_m))_{nb}A_{nb}}{\Delta x_{nb}} \tag{5.8}$$

The flux term in the diffusion, $avg_h(K_m)$, is the harmonic average derived by Syamlal [34, p. 17, Eq. 3.6-3.7]. K_m is the thermal conductivity for either phase. N, S, E, and W represent the cell centers of the north, south, east, and west neighboring cells respectively. ρ_m, $C_{p,m}$, ε_m, and ξ are the density, specific heat at constant pressure, void fraction or solid packing fraction, and convection weighing factor of the m phase respectively. u_m and v_m are the phase velocities in the x- and y-direction. A and ΔV are the cell area and the cell volume respectively. ^{o} denotes a variable at the previous time step. Note that (5.6) and (5.7) differ from those presented by Richardson [2], as Richardson did not include the void fraction from the previous time step as required.

The granular energy equation is a scalar transport equation similar to the energy equation. It is discretized as follows [34]:

$$a_P(\Theta)_p = \sum_{nb} a_{nb}(\Theta)_{nb} + b_P. \tag{5.9}$$

As before, p and nb represent the contributions from the center and neighbor cells respectively. Θ is the granular energy. a and b, the coefficients of the discretized equation in two dimensions are given below:

$$a_E^\Theta = D_e - \frac{3(\xi^\Theta)_e}{2}(\varepsilon_s\rho_s)_E(u_s)_e A_E \tag{5.10}$$

$$a_W^\Theta = D_w + \frac{3(\xi^\Theta)_w}{2}(\varepsilon_s\rho_s)_W(u_s)_w A_W \tag{5.11}$$

$$a_N^\Theta = D_n - \frac{3(\xi^\Theta)_n}{2}(\varepsilon_s\rho_s)_N(v_s)_n A_N \tag{5.12}$$

$$a_S^\Theta = D_s - \frac{3(\xi^\Theta)_s}{2}(\varepsilon_s\rho_s)_S(v_s)_s A_S \tag{5.13}$$

$$a_P^\Theta = -\left(\sum_{nb} a_{nb}^\Theta + \frac{3(\varepsilon_s\rho_s)^o\Delta V}{2\Delta t}\right) + SRC_L\Delta V \tag{5.14}$$

$$b_P^{T_m} = -\left(\frac{(\varepsilon_s\rho_s)^o\Delta V}{\Delta t}\Theta^o\right) + SRC_R\Delta V \tag{5.15}$$

$$SRC_L = \frac{48}{\sqrt{\pi}}\eta(1-\eta)(\varepsilon_s\rho_s)\varepsilon_s g_o\frac{\sqrt{\Theta}}{d} + P_s\frac{tr(D_s)}{\Theta} + \frac{-\lambda_s tr(D_s)^2}{\Theta} + 3F_{gs} \tag{5.16}$$

$$SRC_R = \lambda_s tr(D_s)^2 + 2\mu_s + tr(D_s^2) + P_s(-tr(D_s) + \frac{81\varepsilon_s(\mu_g v_{slip})^2}{g_o d_{ps}^3\rho_s(\pi\Theta)^{1/2}} \tag{5.17}$$

$$v_{slip} = \sqrt{(u_s-u_g)^2 + (v_s-v_g)^2} \tag{5.18}$$

For three dimensions, two coefficients are required for the top and bottom neighbors, represented by t and b. The other scalar variables require these coefficients as well and have the same form. These coefficients are given here:

$$a_T^{\Theta} = D_t - \frac{3(\xi^{\Theta})_t}{2}(\varepsilon_s \rho_s)(w_s)_e A_T \tag{5.19}$$

$$a_B^{\Theta} = D_b + \frac{3(\xi^{\Theta})_b}{2}(\varepsilon_s \rho_s)(w_s)_b A_B \tag{5.20}$$

The equation for v_{slip} is also modified to include the third dimension:

$$v_{slip} = \sqrt{(u_s - u_g)^2 + (v_s - v_g)^2 + (w_s - w_g)^2}.$$

In (5.10) - (5.13), (5.19), and (5.20), D_{nb}, where nb represents either the north, south, east, west, top, or bottom faces (denoted by n, s, e, w, t, and b respectively), is the diffusion. The diffusion is modeled as:

$$D_{nb} = \frac{(avg_h(\kappa_s))_{nb} A_{nb}}{\Delta x_{nb}} \tag{5.21}$$

The flux term in the diffusion, $avg_h(\kappa_s)$, is the harmonic average derived by Syamlal [34, p. 17, Eq. 3.6-3.7]. κ_s is the granular conductivity. N, S, E, W, T, and B represent the north, south, east, west, top, and bottom neighboring cell centers respectively. ρ_s, ε_s, and ξ are the density, solid packing fraction, and convection weighing factor of the solid phase respectively. u_s and v_s are the solid phase velocities in the x- and y-direction. A and ΔV are the cell area and the cell volume respectively. ^{o} denotes a variable at the previous time step.

SRC_L and SRC_R (Equations (5.16) and (5.17)) are the source terms representing particle-particle and gas-particle interaction and granular energy dissipation. All variables that compromise the source terms are taken at the center. g_o, v_{slip}, and η is the radial contact distribution function, particle slip velocity, and restitution function respectively. μ_m is the viscosity of either phase. P_s and λ_s are the solid phase pressure and the second coefficient of viscosity respectively. d_{ps} is the particle diameter. $tr(D_s)$ and $tr(D_s^2)$ are the traces of the stress tensor and square of the stress tensor, respectively. F_{gs} is the drag force on the particles. The constitutive equations for the granular energy are given in Appendix B.

The momentum equations are solved on the staggered grid shown in Figure 11. The discretized gas and solid z-momentum equation is given here [34]:

$$a_P(w_m)_p = \sum_{nb} a_{nb}(w_m)_{nb} + b_P - A_p(\varepsilon_m)_p((P_g)_T - (P_g)_B) + F_{gs}(w_s - w_g)\Delta V. \quad (5.22)$$

p and nb represent the contributions from the center and neighbor cells. T and B represent the top and bottom neighbor cell respectively. m designates the gas and solid phase. a and b are the coefficients of the discretized momentum equation. w_m is the velocity in the z-direction for either phase. ε_m is the void fraction or solid packing fraction depending on the phase. P_g is the gas pressure. F_{gs} is the drag force on the particles. ΔV is the cell volume. The coefficients of the discretized z-momentum equation are given as:

$$a_E^{w_m} = D_E - \xi_e^{w_m}(\varepsilon_m\rho_m)_E(u_m)_eA_e \quad (5.23)$$

$$a_W^{w_m} = D_w + \xi_w^{w_m}(\varepsilon_m\rho_m)_W(u_m)_wA_w \quad (5.24)$$

$$a_N^{w_m} = D_N - \xi_n^{w_m}(\varepsilon_m\rho_m)_N(v_m)_nA_n \quad (5.25)$$

$$a_S^{w_m} = D_S + \xi_s^{w_m}(\varepsilon_m\rho_m)_S(v_m)_sA_s \quad (5.26)$$

$$a_T^{w_m} = D_T - \xi_t^{w_m}(\varepsilon_m\rho_m)_T(w_m)_tA_t \quad (5.27)$$

$$a_B^{w_m} = D_B + \xi_b^{w_m}(\varepsilon_m\rho_m)_B(w_m)_bA_b \quad (5.28)$$

$$a_P^{w_m} = \sum_{nb} a_{nb}^{w_m} + \frac{(\varepsilon_m\rho_m)^o}{\Delta t}\Delta V \quad (5.29)$$

$$b_P^{w_m} = \frac{(\varepsilon_m \rho_m)^o}{\Delta t} \Delta V w_m^o. \tag{5.30}$$

In (5.23) - (5.28), D_{nb} represents the diffusion. nb is the index representing the north, south, east, west, top, and bottom neighbor cells (N, S, E, W, T, and B respectively). The diffusion is expressed as:

$$D_{nb} = \frac{(avg_h(\mu_m))_{nb} A_{nb}}{\Delta x_{nb}} \tag{5.31}$$

The flux term in the diffusion, $avg_h(\mu_m)$, is the harmonic average derived by Syamlal [34, p. 17, Eq. 3.6-3.7]. μ_m is the gas or solid viscosity. n, s, e, w, t, and b represent the north, south, east, west, top, and bottom faces respectively. ρ_m, ε_m, and ξ are the density, void fraction or solid packing fraction, and convection weighing factor of the gas or solid phase respectively. u_m, v_m, and w_m are the gas or solid phase velocities in the x-, y-, or z-direction.

D. Summary

This chapter presented the full-order models for single (UNS3D) and multiphase flows (MFIX). The spatial discretization was presented for both models. The energy, granular energy, and z-momentum equations of MFIX were discretized. In the next chapter, the multiphase equations will be implemented into a reduced-order model based on proper-orthogonal decomposition. From now on, MFIX will be the basis of comparison for the multiphase POD-based ROM results.

CHAPTER VI

REDUCED-ORDER MODELS BASED ON PROPER ORTHOGONAL DECOMPOSITION FOR TRANSPORT PHENOMENA IN FLUIDIZED BEDS

A. Introduction

The purpose of this chapter is to present the derivations and techniques required to generate the POD-based ROMs discussed herein. The ROMs will be referred to by the following names to ease the discussion: ODEti for the improved energy equation ROM, ODEg for the granular energy equation ROM, and ODEV for the full 3D isothermal ROM. These ROMs were generated from the discretized equations for multiphase flow given in the previous chapter. They will each be further discussed in the following sections.

B. Reduced-Order Model Based on Proper-Orthogonal Decomposition for the Scalar Energy Equation

This section presents the improved reduced-order model for the scalar energy equation. A ROM for the scalar energy equation was first presented by Richardson [2]. Upon implementing the other ROMs, it was discovered that ODEt [2] was not correctly implemented, which will be outlined later. Therefore, the derivation of the ROM for the scalar energy equations will be rederived here. The same unknowns given by Richardson [2], plus the void fraction ε_g are required to solve the scalar energy equation. The basis functions and time coefficients of these variables were extracted to generate the ROM.

The scalar energy equation can be written as [34, p. 54]:

$$a_P(T_l)_p = \sum_{nb} a_{nb}(T_l)_{nb} + b_P, \tag{6.1}$$

where l denotes the phase either gas g or solid s. (5.6) and (5.7) are next substituted into (6.1):

$$-\left(\sum_{nb} a_{nb}^{T_m} + \frac{(\varepsilon_l\rho_l)^o C_{p,l}\Delta V}{\Delta t}\right)(T_l)_P = \sum_{nb} a_{nb}(T_l)_{nb} - \left(\frac{(\varepsilon_l\rho_l)^o C_{p,l}\Delta V}{\Delta t}(T_l^o)_P\right). \tag{6.2}$$

(6.2) can be rearranged as:

$$-(\varepsilon_l\rho_l)^o C_{p,l}\Delta V \frac{(T_l)_P - (T_l^o)_P}{\Delta t} = \sum_{nb} a_{nb}((T_l)_P + (T_l)_{nb}). \tag{6.3}$$

By replacing $\frac{(T_l)_P-(T_l^o)_P}{\Delta t}$ with $\frac{\partial (T_l)_p}{\partial t}$, (6.3) can be written as:

$$-(\varepsilon_l\rho_l)^o C_{p,l}\frac{\partial (T_l)_p}{\partial t}\Delta V = \sum_{nb} a_{nb}((T_l)_P + (T_l)_{nb}). \tag{6.4}$$

T_l, the phase temperature, is then approximated using the POD basis functions and time coefficients with the reconstruction equation (B):

$$T_l(\mathbf{x},t) \simeq \phi_O^{T_l}(\mathbf{x}) + \sum_{i=1}^{m^{T_l}} \alpha_i^{T_l}(t)\phi_i^{T_l}(\mathbf{x}). \tag{6.5}$$

m is the number of modes used to approximate T_l. Substituting (6.5) into (6.4) yields:

$$-(\varepsilon_l\rho_l)^o C_{p,l}\Delta V\left(\sum_{i=1}^{m^{T_l}} \dot{\alpha}_i^{T_l}\phi_i^{T_l}\right) = \sum_{nb} a_{nb}\sum_{i=0}^{m^{T_l}} \alpha_i^{T_l}((\phi_i^{T_l})_P + (\phi_i^{T_l})_{nb}). \tag{6.6}$$

With (5.2) - (5.5), (6.6) becomes:

$$
\begin{aligned}
&-(\varepsilon_l\rho_l)^o C_{p,l}\Delta V\sum_{i=0}^{m^{T_l}}\dot{\alpha}_i^{T_l}\phi_i^{T_l} = \\
&\quad -\sum_{i=0}^{m^{T_l}}\sum_{j=0}^{m^{u_l}}\frac{(\xi^{T_l})_e}{2}(\varepsilon_l\rho_l)_E((C_{p,l})_P+(C_{p,l})_E)\Delta y(\phi_j^{u_l})_E((\phi_i^{T_l})_E+(\phi_i^{T_l})_P)\alpha_i^{T_l}\alpha_i^{u_l}+ \\
&\quad \sum_{i=0}^{m^{T_l}}\sum_{j=0}^{m^{u_l}}\frac{(\xi^{T_l})_w}{2}(\varepsilon_l\rho_l)_W((C_{p,l})_P+(C_{p,l})_W)\Delta y(\phi_j^{u_l})_W((\phi_i^{T_l})_W+(\phi_i^{T_l})_P)\alpha_i^{T_l}\alpha_i^{u_l}- \\
&\quad \sum_{i=0}^{m^{T_l}}\sum_{j=0}^{m^{u_l}}\frac{(\xi^{T_l})_n}{2}(\varepsilon_l\rho_l)_N((C_{p,l})_P+(C_{p,l})_N)\Delta x(\phi_j^{v_l})_N((\phi_i^{T_l})_N+(\phi_i^{T_l})_P)\alpha_i^{T_l}\alpha_i^{v_l}+ \\
&\quad \sum_{i=0}^{m^{T_l}}\sum_{j=0}^{m^{u_l}}\frac{(\xi^{T_l})_s}{2}(\varepsilon_l\rho_l)_S((C_{p,l})_P+(C_{p,l})_S)\Delta x(\phi_j^{v_l})_S((\phi_i^{T_l})_S+(\phi_i^{T_l})_P)\alpha_i^{T_l}\alpha_i^{v_l}+ \\
&\quad \sum_{i=0}^{m^{T_l}}\frac{avg_h((K_l)_E)\Delta y}{\Delta x}((\phi_i^{T_l})_E+(\phi_i^{T_l})_P)\alpha_i^{T_l}+ \\
&\quad \sum_{i=0}^{m^{T_l}}\frac{avg_h((K_l)_W)\Delta y}{\Delta x}((\phi_i^{T_l})_W+(\phi_i^{T_l})_P)\alpha_i^{T_l}+ \\
&\quad \sum_{i=0}^{m^{T_l}}\frac{avg_h((K_l)_N)\Delta x}{\Delta y}((\phi_i^{T_l})_N+(\phi_i^{T_l})_P)\alpha_i^{T_l}+ \\
&\quad \sum_{i=0}^{m^{T_l}}\frac{avg_h((K_l)_S)\Delta x}{\Delta y}((\phi_i^{T_l})_S+(\phi_i^{T_l})_P)\alpha_i^{T_l}. \qquad (6.7)
\end{aligned}
$$

In (6.7) the area terms A_E and A_w and A_N and A_S reduce to Δy and Δx, respectively, for two dimensional flows. $avg_h(K_{nb})$ is the harmonic average of the thermal conductivity given by Syamlal [34, p. 17, Eq. 3.6-3.7]. (6.7) is known as the discretized finite volume scalar energy equation approximated with proper orthogonal decomposition. (6.7) is projected onto the basis functions $\phi_k^{T_l}$, where $k \in [1, m^{T_l}]$:

$$
\begin{aligned}
&-(\varepsilon_l\rho_l)^o \Delta V \sum_{i=0}^{m^{T_l}} \dot{\alpha}_i^{T_l}(\phi_i^{T_l}, \phi_k^{T_l}) = \\
&\quad -\sum_{i=0}^{m^{T_l}} \sum_{j=0}^{m^{u_l}} \frac{(\xi^{T_l})_e}{2} (\varepsilon_l\rho_l)_E ((C_{p,l})_P + (C_{p,l})_E) \Delta y (\phi_j^{u_l})_E (((\phi_i^{T_l})_E + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} \alpha_i^{u_l} + \\
&\quad \sum_{i=0}^{m^{T_l}} \sum_{j=0}^{m^{u_l}} \frac{(\xi^{T_l})_w}{2} (\varepsilon_l\rho_l)_W ((C_{p,l})_P + (C_{p,l})_W) \Delta y (\phi_j^{u_l})_W (((\phi_i^{T_l})_W + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} \alpha_i^{u_l} - \\
&\quad \sum_{i=0}^{m^{T_l}} \sum_{j=0}^{m^{u_l}} \frac{(\xi^{T_l})_n}{2} (\varepsilon_l\rho_l)_N ((C_{p,l})_P + (C_{p,l})_N) \Delta x (\phi_j^{v_l})_N (((\phi_i^{T_l})_N + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} \alpha_i^{v_l} + \\
&\quad \sum_{i=0}^{m^{T_l}} \sum_{j=0}^{m^{u_l}} \frac{(\xi^{T_l})_s}{2} (\varepsilon_l\rho_l)_S ((C_{p,l})_P + (C_{p,l})_S) \Delta x (\phi_j^{v_l})_S (((\phi_i^{T_l})_S + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} \alpha_i^{v_l} + \\
&\quad \sum_{i=0}^{m^{T_l}} \frac{avg_h((K_l)_E)\Delta y}{\Delta x} (((\phi_i^{T_l})_E + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} + \\
&\quad \sum_{i=0}^{m^{T_l}} \frac{avg_h((K_l)_W)\Delta y}{\Delta x} (((\phi_i^{T_l})_W + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} + \\
&\quad \sum_{i=0}^{m^{T_l}} \frac{avg_h((K_l)_N)\Delta x}{\Delta y} (((\phi_i^{T_l})_N + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l} + \\
&\quad \sum_{i=0}^{m^{T_l}} \frac{avg_h((K_l)_S)\Delta x}{\Delta y} (((\phi_i^{T_l})_S + (\phi_i^{T_l})_P), \phi_k^{T_l}) \alpha_i^{T_l}.
\end{aligned}
\tag{6.8}
$$

(6.8) can be rewritten as:

$$
\check{A}_{ij}^{T_l} \dot{\alpha}_k^{T_l} = \sum_{i=0}^{m^{T_l}} \sum_{j=0}^{m^{u_l}} \check{F}_{kij}^{T_l} \alpha_i^{T_l} \alpha_i^{u_l} + \sum_{i=0}^{m^{T_l}} \sum_{j=0}^{m^{v_l}} \check{G}_{kij}^{T_l} \alpha_i^{T_l} \alpha_j^{v_l} + \sum_{i=0}^{m^{T_l}} \check{H}_{ki}^{T_l} \alpha_i^{T_l} \tag{6.9}
$$

where

$$
\check{A}_{ij}^{T_l} = \delta_{ij} \cdot ((\varepsilon_l\rho_l)^o \Delta V \phi_i^{T_l}, \phi_k^{T_l})
$$

$$\check{F}^{T_l}_{kij} = -\frac{(\xi^{T_l})_e}{2}(\varepsilon_l\rho_l)_E((C_{p,l})_P + (C_{p,l})_E)\Delta y(\phi^{u_l}_j)_E(((\phi^{T_l}_i)_E + (\phi^{T_l}_i)_P), \phi^{T_l}_k) +$$
$$\frac{(\xi^{T_l})_w}{2}(\varepsilon_l\rho_l)_W((C_{p,l})_P + (C_{p,l})_W)\Delta y(\phi^{u_l}_j)_W(((\phi^{T_l}_i)_W + (\phi^{T_l}_i)_P), \phi^{T_l}_k),$$

$$\check{G}^{T_l}_{kij} = -\frac{(\xi^{T_l})_e}{2}(\varepsilon_l\rho_l)_N((C_{p,l})_P + (C_{p,l})_N)\Delta x(\phi^{v_l}_j)_N(((\phi^{T_l}_i)_N + (\phi^{T_l}_i)_P), \phi^{T_l}_k) +$$
$$\frac{(\xi^{T_l})_S}{2}(\varepsilon_l\rho_l)_S((C_{p,l})_P + (C_{p,l})_S)\Delta x(\phi^{v_l}_j)_S(((\phi^{T_l}_i)_S + (\phi^{T_l}_i)_P), \phi^{T_l}_k),$$

$$\check{H}^{T_L}_{ki} = \frac{avg_h((K_l)_E)\Delta y}{\Delta x}(((\phi^{T_l}_i)_E + (\phi^{T_l}_i)_P), \phi^{T_l}_k) + \frac{avg_h((K_l)_W)\Delta y}{\Delta x}(((\phi^{T_l}_i)_W + (\phi^{T_l}_i)_P), \phi^{T_l}_k) +$$
$$\frac{avg_h((K_l)_N)\Delta x}{\Delta y}(((\phi^{T_l}_i)_N + (\phi^{T_l}_i)_P), \phi^{T_l}_k) + \frac{avg_h((K_l)_S)\Delta x}{\Delta y}(((\phi^{T_l}_i)_S + (\phi^{T_l}_i)_P), \phi^{T_l}_k).$$

(6.9) has been reduced from the a large number of equations to a set of m^{T_l} ODEs. It can be written in a simpler form:

$$\tilde{A}^{T_l}\alpha^{T_l} = \tilde{B}^{T_l}, \tag{6.10}$$

where

$$\tilde{A}^{T_l} = \left(a^{T_l}_p\phi^{T_l}_j - \sum_{nb} a^{T_l}_{nb}(\phi_j)_{nb}), \phi^{T_l}_k\right),$$
$$\bar{B}^{T_l} = (b^{T_l}_p, \phi^{T_l}_k).$$

The dimensions of $\tilde{A}$ and $\tilde{B}$ are $m^{T_l} \times m^{T_l}$ and $m^{T_l} \times 1$, respectively. ODEti uses a similar iterative, time marching algorithm to that of MFIX. The solution algorithm for ODEti is outlined below:

- With the time coefficients read-in from the input file or from a previous iteration and the basis functions generated from PODDEC, the field variables, u_g, u_s, v_g, v_s, P_g, ε_g, T_g, and T_s, are reconstructed.
- The fluid and solid physical properties not defined in the input are computed.

Using the ideal gas law, the gas density ρ_g is computed. The gas-solid drag F_{gs} is computed. The solid viscosity μ_s and temperature dependent gas thermal conductivity k_g are computed.

- The uncorrected velocity field time coefficients $\alpha_i^{u_g}(t)$, $i \in [1, m^{u_g}]$, $\alpha_i^{u_s}(t)$, $i \in [1, m^{u_s}]$, $\alpha_i^{v_g}(t)$, $i \in [1, m^{v_g}]$, and $\alpha_i^{v_s}(t)$, $i \in [1, m^{v_s}]$, defined by the linear system of equations [47][p. 248, Eq. 30-31] are solved. m denotes the number of modes used for each variable. The time coefficients are called "uncorrected" as they are computed using the previous pressure field and void fraction. The gas and solid velocities will be corrected using the gas pressure correction and solids volume fraction correction equations, respectively.

- The linear system of equations describing the time coefficients of the gas pressure correction equation, $\alpha_i^{p_g'}(t)$, $i \in [1, m^{p_g}]$, [47, p. 249, Eq. 32] is solved.

- The gas pressure and velocities' time coefficients are corrected using the time coefficients computed from the gas pressure correction equation.

- The linear system of equations describing the time coefficients of the solid volume fraction correction equation $\alpha_i^{\varepsilon_s'}(t)$, $i \in [1, m^{\varepsilon_s}]$ [2].

- The time coefficients of the void fraction and the solid velocities are corrected using the time coefficients computed from the solid volume fraction correction equation.

- The linear system of equations for gas and solid temperature time coefficients given by (6.10) are solved yielding $\alpha_i^{T_g}(t)$, $i \in [1, m^{T_g}]$ and $\alpha_i^{T_s}(t)$, $i \in [1, m^{T_s}]$.

- Check convergence of the time coefficients computed from the linear systems. If the solution has converged, ODEti is advanced forward in time to the next time step. Otherwise, the solutions are iterated until convergence is reached.

ODEti requires the basis functions extracted from the eight field variable databases generated by MFIX. ODEti solves for the time coefficients of these field variables: u_g, u_s, v_g, v_s, P_g, ε_g, T_g, and T_s. A sample input file for ODEti is given in Appendix C. Table II and III show the subroutines modified from ODEt created by Richardson [2]. These subroutines were modified to correct the errors present in ODEt.

Table II: Most important routines modified or added for ODEti.

Subroutine	Modification
adjust_eps.f	Fixed so only used with 3D computations.
calc_ab_t_g/s.f	Corrected boundary references.
calc_mu_s.f	Corrected calculation of solid viscosity, μ_s to match Summary of MFIX Equations [44].
calc_k_g/s.f	Modified for calculation of temperature dependent thermal conductivities.
calc_kcp.f	Corrected logic.
calc_resid_s1.f	Corrected calculation of residual.
calc_trd.f	Calculate trace of strain tensors only where fluids/solids are present.
conv_dif_phi.f	Corrected boundary references.
conv_dif_u_g/s.f	Added diffusion term for south face.
conv_rop.f	New subroutine used to compute the face value of density for convective flux calculation.

Table III: Additional routines modified or added for ODEti.

Subroutine	Modification
correct_1.f	Removed correction of solid quantities.
check_convergence.f	Added temperature residuals to total residual, allowing temperature to correctly converge.
iterate.f	Modified to follow computations of field variables in the same order as MFIX. Update properties after each field variable is computed. Added calls to new subroutines used to compute fluxes.
mflux.f	New subroutine used to compute corrected fluxes
partial_elim_s.f	Corrected boundary references.
set_outflow.f	Correctly set outlet flow conditions for pressure outflow.
set_index.f	Corrected definition of cell, boundary, and neighbor indexes.
source_phi.f	Corrected boundary references.
solve_energy_eq.f	Corrected boundary references.

C. Reduced-Order Model Based on Proper-Orthogonal Decomposition for the Granular Energy Equation

This section presents the development of the reduced-order model for the granular energy equation. ODEg is a POD-based ROM generated to model isothermal flow with granular energy in a fluidized bed. ODEg is derived from the discretized 2-D momentum equations, gas pressure correction equation, solids volume fraction equation, and granular energy equation used in MFIX [34]. There are a total of seven unknowns per cell: the gas and solid velocities u_g, u_s, v_g, and v_s, the gas pressure P_g, void fraction ε_g, and granular energy Θ. The basis functions and time coefficients of these variables are required to generate the ROM for the granular energy equation. The ROMs for the x- and y-momentum equations and pressure correction equation were presented by Yuan [1] and the solid volume fraction equation by Richardson [2].

The granular energy equation can be written as:

$$a_P(\Theta)_p = \sum_{nb} a_{nb}(\Theta)_{nb} + b_P, \tag{6.11}$$

(5.14) and (5.15) are next substituted into (6.11) to yield:

$$\begin{aligned}-\left(\sum_{nb} a_{nb}^{\Theta} + \frac{3(\varepsilon_s\rho_s)^o\Delta V}{2\Delta t}\right)(\Theta)_P = &\sum_{nb} a_{nb}(\Theta)_{nb} - \\ &\left(\frac{3(\varepsilon_s\rho_s)^o\Delta V}{2\Delta t}(\Theta^o)_P\right) + \Delta S\Delta V,\end{aligned} \tag{6.12}$$

where ΔS is the difference between the right and left source terms in (5.14) and (5.15). (6.12) can be rearranged as:

$$\begin{aligned}-\frac{3}{2}(\varepsilon_s\rho_s)^o\Delta V\frac{(\Theta)_P - (\Theta^o)_P}{\Delta t} = \\ \sum_{nb} a_{nb}((\Theta)_P + (\Theta)_{nb}) + \Delta S\Delta V.\end{aligned} \tag{6.13}$$

By replacing $\frac{(\Theta)_P - (\Theta^o)_P}{\Delta t}$ with $\frac{\partial(\Theta)_P}{\partial t}$, (6.13) can be written as:

$$-\frac{3}{2}(\varepsilon_s \rho_m)^o \frac{\partial(\Theta)_P}{\partial t}\Delta V = \sum_{nb} a_{nb}((\Theta)_P + (\Theta)_{nb}) + \Delta S \Delta V. \tag{6.14}$$

Θ is then approximated using the POD basis functions and time coefficients with the reconstruction equation (B):

$$\Theta(\mathbf{x}, t) \simeq \phi_O^\Theta(\mathbf{x}) + \sum_{i=1}^{m^\Theta} \alpha_i^\Theta(t) \phi_i^\Theta(\mathbf{x}). \tag{6.15}$$

m is the number of modes used to approximate Θ. Substituting (6.15) into (6.14) yields:

$$-\frac{3}{2}(\varepsilon_s \rho_s)^o \Delta V \left(\sum_{i=1}^{m^\Theta} \dot{\alpha}_i^\Theta \phi_i^\Theta \right) = \sum_{nb} a_{nb} \sum_{i=0}^{m^\Theta} \alpha_i^\Theta ((\phi_i^\Theta)_P + (\phi_i^\Theta)_{nb}) \Delta S \Delta V. \tag{6.16}$$

With (5.10) - (5.13), (6.16) becomes:

$$
\begin{aligned}
-\frac{3}{2}(\varepsilon_s\rho_s)^o\Delta V\sum_{i=0}^{m^\Theta}\dot{\alpha}_i^\Theta\phi_i^\Theta = \\
&-\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{u_s}}\frac{3(\xi^\Theta)_e}{2}(\varepsilon_s\rho_s)_E\Delta y(\phi_j^{u_s})_E((\phi_i^\Theta)_E+(\phi_i^\Theta)_P)\alpha_i^\Theta\alpha_i^{u_s}+ \\
&\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{u_s}}\frac{3(\xi^\Theta)_w}{2}(\varepsilon_s\rho_s)_W\Delta y(\phi_j^{u_s})_W((\phi_i^\Theta)_W+(\phi_i^\Theta)_P)\alpha_i^\Theta\alpha_i^{u_s}- \\
&\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{v_s}}\frac{3(\xi^\Theta)_n}{2}(\varepsilon_s\rho_s)_N\Delta x(\phi_j^{v_s})_N((\phi_i^\Theta)_N+(\phi_i^\Theta)_P)\alpha_i^\Theta\alpha_i^{v_s}+ \\
&\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{v_s}}\frac{3(\xi^\Theta)_s}{2}(\varepsilon_s\rho_s)_S\Delta x(\phi_j^{v_s})_S((\phi_i^\Theta)_S+(\phi_i^\Theta)_P)\alpha_i^\Theta\alpha_i^{v_s}+ \\
&\sum_{i=0}^{m^\Theta}\frac{3avg_h((\kappa_s)_E)\Delta y}{2\Delta x}((\phi_i^\Theta)_E+(\phi_i^\Theta)_P)\alpha_i^\Theta+ \\
&\sum_{i=0}^{m^\Theta}\frac{3avg_h((\kappa_s)_W)\Delta y}{2\Delta x}((\phi_i^\Theta)_W+(\phi_i^\Theta)_P)\alpha_i^\Theta+ \\
&\sum_{i=0}^{m^\Theta}\frac{3avg_h((\kappa_s)_N)\Delta x}{2\Delta y}((\phi_i^\Theta)_N+(\phi_i^\Theta)_P)\alpha_i^\Theta+ \\
&\sum_{i=0}^{m^\Theta}\frac{3avg_h((\kappa_s)_S)\Delta x}{2\Delta y}((\phi_i^\Theta)_S+(\phi_i^\Theta)_P)\alpha_i^\Theta+\Delta S\Delta V. \qquad (6.17)
\end{aligned}
$$

In (6.17) the area terms A_E and A_w and A_N and A_S reduce to Δy and Δx, respectively, for two dimensional flows. $avg_h((\kappa_s)_{nb})$ is the harmonic average of the granular conductivity given by Syamlal [34, p. 17, Eq. 3.6-3.7]. (6.17) is known as the discretized finite volume granular energy equation approximated with proper orthogonal decomposition. (6.17) is projected onto the basis functions ϕ_k^Θ, where $k \in [1, m^\Theta]$:

$$
\begin{aligned}
-\frac{3}{2}(\varepsilon_s\rho_s)^o\Delta V \sum_{i=0}^{m^\Theta} \dot{\alpha}_i^\Theta(\phi_i^\Theta, \phi_k^\Theta) = \\
&-\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{u_s}} \frac{3(\xi^\Theta)_e}{2}(\varepsilon_s\rho_s)_E\Delta y(\phi_j^{u_s})_E(((\phi_i^\Theta)_E + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta\alpha_i^{u_s} + \\
&\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{u_s}} \frac{3(\xi^\Theta)_w}{2}(\varepsilon_s\rho_s)_W\Delta y(\phi_j^{u_s})_W(((\phi_i^\Theta)_W + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta\alpha_i^{u_s} - \\
&\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{v_s}} \frac{3(\xi^\Theta)_n}{2}(\varepsilon_s\rho_s)_N\Delta x(\phi_j^{v_s})_N(((\phi_i^\Theta)_N + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta\alpha_i^{v_s} + \\
&\sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{v_s}} \frac{3(\xi^\Theta)_s}{2}(\varepsilon_s\rho_s)_S\Delta x(\phi_j^{v_s})_S(((\phi_i^\Theta)_S + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta\alpha_i^{v_s} + \\
&\sum_{i=0}^{m^\Theta} \frac{3avg_h((\kappa_s)_E)\Delta y}{2\Delta x}(((\phi_i^\Theta)_E + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta + \\
&\sum_{i=0}^{m^\Theta} \frac{3avg_h((\kappa_s)_W)\Delta y}{2\Delta x}(((\phi_i^\Theta)_W + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta + \\
&\sum_{i=0}^{m^\Theta} \frac{3avg_h((\kappa_s)_N)\Delta x}{2\Delta y}(((\phi_i^\Theta)_N + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta + \\
&\sum_{i=0}^{m^\Theta} \frac{3avg_h((\kappa_s)_S)\Delta x}{2\Delta y}(((\phi_i^\Theta)_S + (\phi_i^\Theta)_P), \phi_k^\Theta)\alpha_i^\Theta + (\Delta S\Delta V, \phi_k^\Theta). \qquad (6.18)
\end{aligned}
$$

(6.18) can be rewritten as:

$$\check{A}_{ij}^\Theta\dot{\alpha}_k^\Theta = \sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{u_s}} \check{F}_{kij}^\Theta\alpha_i^\Theta\alpha_i^{u_s} + \sum_{i=0}^{m^\Theta}\sum_{j=0}^{m^{v_s}} \check{G}_{kij}^\Theta\alpha_i^\Theta\alpha_j^{v_s} + \sum_{i=0}^{m^\Theta} \check{H}_{ki}^\Theta\alpha_i^\Theta + \check{S}^\Theta \qquad (6.19)$$

where

$$\check{A}_{ij}^\Theta = \delta_{ij} \cdot (\frac{3}{2}(\varepsilon_s\rho_s)^o\Delta V\phi_i^\Theta, \phi_k^\Theta)$$

$$\check{F}^{\Theta}_{kij} = -\frac{3(\xi^{\Theta})_e}{2}(\varepsilon_s\rho_s)_E\Delta y(\phi^{u_s}_j)_E(((\phi^{\Theta}_i)_E + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k)+$$
$$\frac{3(\xi^{\Theta})_w}{2}(\varepsilon_s\rho_s)_W\Delta y(\phi^{u_s}_j)_W(((\phi^{\Theta}_i)_W + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k),$$

$$\check{G}^{\Theta}_{kij} = -\frac{3(\xi^{\Theta})_e}{2}(\varepsilon_s\rho_s)_N\Delta x(\phi^{v_s}_j)_N(((\phi^{\Theta}_i)_N + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k)+$$
$$\frac{3(\xi^{\Theta})_S}{2}(\varepsilon_s\rho_s)_S\Delta x(\phi^{v_s}_j)_S(((\phi^{\Theta}_i)_S + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k),$$

$$\check{H}^{\Theta}_{ki} = \frac{3avg_h((\kappa_s)_E)\Delta y}{2\Delta x}(((\phi^{\Theta}_i)_E + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k) + \frac{3avg_h((\kappa_s)_W)\Delta y}{2\Delta x}(((\phi^{\Theta}_i)_W + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k)+$$
$$\frac{3avg_h((\kappa_s)_N)\Delta x}{2\Delta y}(((\phi^{\Theta}_i)_N + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k) + \frac{3avg_h((\kappa)_s)_S)\Delta x}{2\Delta y}(((\phi^{\Theta}_i)_S + (\phi^{\Theta}_i)_P), \phi^{\Theta}_k),$$

$$\check{S}^{\Theta} = (\Delta S \Delta V, \phi^{\Theta}_k)$$

(6.19) has been reduced from the a large number of equations to a set of m^{Θ} ODEs. It can be written in a simpler form:

$$\tilde{A}^{\Theta}\alpha^{\Theta} = \tilde{B}^{\Theta}, \tag{6.20}$$

where

$$\tilde{A}^{\Theta} = \left((a^{\Theta}_p\phi^{\Theta}_j - \sum_{nb} a^{\Theta}_{nb}(\phi_j)_{nb}), \phi^{\Theta}_k\right),$$
$$\bar{B}^{\Theta} = (b^{\Theta}_p, \phi^{\Theta}_k).$$

The dimensions of $\tilde{A}$ and $\tilde{B}$ are $m^{\Theta} \times m^{\Theta}$ and $m^{\Theta} \times 1$ respectively. ODEg uses a similar iterative, time marching algorithm to that of MFIX implemented similarly to ODEti. The solution algorithm for ODEg is outlined below:

- With the time coefficients read-in from the input file or from a previous iteration and the basis functions generated from PODDEC, the field variables, u_g, u_s, v_g, v_s, P_g, ε_g, and Θ, are reconstructed.

- The fluid and solid physical properties not defined in the input are computed. Using the ideal gas law, the gas density ρ_g is computed. The gas-solid drag F_{gs} is computed. The solid viscosity μ_s and granular conductivity κ_s are calculated.

- The uncorrected velocity field time coefficients $\alpha_i^{u_g}(t)$, $i \in [1, m^{u_g}]$, $\alpha_i^{u_s}(t)$, $i \in [1, m^{u_s}]$, $\alpha_i^{v_g}(t)$, $i \in [1, m^{v_g}]$, and $\alpha_i^{v_s}(t)$, $i \in [1, m^{v_s}]$, defined by the linear system of equations [47, p. 248, Eq. 30-31] are solved. m denotes the number of modes used for each variable. The time coefficients are called "uncorrected" as they are computed using the previous pressure field and void fraction. The gas and solid velocities will be corrected using the gas pressure correction and solids volume fraction correction equations, respectively.

- The linear system of equations describing the time coefficients of the gas pressure correction equation, $\alpha_i^{p_g'}(t)$, $i \in [1, m^{p_g}]$, [34, p. 249, Eq. 32] is solved.

- The gas pressure and velocities' time coefficients are corrected using the time coefficients computed from the gas pressure correction equation.

- The linear system of equations describing the time coefficients of the solid volume fraction correction equation $\alpha_i^{\varepsilon_s'}(t)$, $i \in [1, m^{\varepsilon_s}]$ [2].

- The time coefficients of the void fraction and the solid velocities are corrected using the time coefficients computed from the solid volume fraction correction equation.

- The linear system of equations for the time coefficients of granular energy given by (6.20) is solved for $\alpha_i^{\Theta}(t)$, $i \in [1, m^{\Theta}]$.

- Convergence of the time coefficients computed from the linear systems is checked. If the solution has converged, ODEg is advanced forward in time to the next time step. Otherwise, the solutions are iterated until convergence is reached.

ODEg requires the basis functions extracted from the seven field variable databases generated by MFIX. ODEg solves for the time coefficients of these field variables: u_g, u_s, v_g, v_s, P_g, ε_g, and Θ. A sample input file for ODEg is given in Appendix D. Table IV shows the subroutines created for ODEg. These subroutines are important because they perform the projection of the governing PDEs onto the granular energy basis functions, solve the linear system of equations for the time coefficients of Θ, and reconstruct the solution field of Θ using the time coefficients α^{Θ}. Table V shows the list of the most important subroutines modified for ODEg. These modifications were implemented using MFIX version 2005-4 and generally included adding terms that included the granular energy. Table VI shows the list of support subroutines modified for ODEg. These support subroutines do not directly interact with the solution subroutines but are used to read-in variables or declare new ones.

Table IV: Most important routines created for ODEg.

Subroutine	New Subroutine Description
calc_ab_theta_m.f	Projects the granular energy equation system onto the basis functions extracted from the granular energy snapshots
theta_phi_prod.f	Calculates the constant basis function products outside the iteration loop
reconstruct_theta_m.f	Reconstructs the granular energy from the basis functions and time coefficients

Table V: Most important routines modified for ODEg.

Subroutine	Modification
allocate_arrays.f	Defined dynamic allocation size for the basis function arrays, time coefficients, constant basis function products and all new granular energy equation parameters not defined in ODEx.
adjust_theta.f	Added to ODEg, no modification from MFIX.
calc_mu_s.f	Added subroutine to compute viscosity related terms for non-algebraic granular energy computations
check_convergence.f	Added granular energy to total residual.
iterate.f	Added call line for solve_granular_energy.
reset_new.f	Added granular energy to be updated before every iteration.
source_granular_energy.f	Implemented 2-D version of routine
solve_granular_energy.f	Modified the call line for the linear equation solver and calc_resid_s to solve for the time coefficients of the granular energy, α^{Θ}. Added the call line for the new calc_ab_theta_m routine.
time_march.f	Added lines to open and write out computed time coefficients to the output files. Added call statements for reconstruct_theta_m routine.
update_old.f	Added granular energy to be updated after every iteration

Table VI: Support routines modified for ODEg.

Subroutine	Modification
basis_mod.f	Added granular energy basis function definitions
get_data.f	Added statement to print error flag for undefined granular energy input parameters
namelist.inc	Added number of modes for granular energy parameters. Option to solve granular energy also added.
param_mod.f	Added definition for nTheta_m, the integer number of modes used for granular energy reconstruction
read_basis.f	Modified routine to read the granular energy basis functions from the PODDEC generated basis function files
set_ic.f	Modified routine to read the first time coefficients for granular energy from the PODDEC generated time coefficient files to generate an initial field solution
time_coe_mod.f	Added definition for α^{Θ} the granular energy time coefficient

D. Reduced-Order Model Based on Proper-Orthogonal Decomposition for Three-Dimensional Isothermal Multiphase Flows

This section presents the reduced-order model for three-dimensional isothermal multiphase flows. This model incorporates the z-momentum equation and modifies the existing ROMs to include the influence from the top and bottom cell neighbors. The unknowns are the phase velocities u_g,u_s,v_g,v_s,w_g, and w_s, gas pressure P_g, and void fraction ε_g. The basis functions and time coefficients of these variables are collected to generate the reduced-order model.

The z-momentum equation can be written as:

$$a_p(w_l)_P = \sum_{nb} a_{nb}(w_l)_{nb} + b_P - A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B) + F_{gs}(w_s - w_g).\Delta V, \quad (6.21)$$

where l denotes the phase, either gas g or solid s. (5.29) and (5.30) are substituted into (6.21) to yield:

$$\left(\sum_{nb} a_{nb}^{w_l} + \frac{(\varepsilon_l\rho_l)^o}{\Delta t}\Delta V\right)(w_l)_P = \sum_{nb} a_{nb}^{w_l}(w_l)_{nb} + \frac{(\varepsilon_l\rho_l)^o}{\Delta t}\Delta V(w_l^o)_p - A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B) + F_{gs}(w_s - w_g)\Delta V. \quad (6.22)$$

(6.22) can be rearranged to:

$$(\varepsilon_l\rho_l)^o\Delta V\frac{(w_l - w_l^o)_p}{\Delta t} = \sum_{nb} a_{nb}((w_l)_{nb} - (w_l)_p) - A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B) + F_{gs}(w_s - w_g)\Delta V. \quad (6.23)$$

$\frac{(w_l - w_l^o)_p}{\Delta t}$ in (6.23) can be replaced with $\frac{\partial (w_l)_p}{\partial t}$. (6.23) can be rewritten as:

$$(\varepsilon_l \rho_l)^o \Delta V \frac{\partial (w_l)_p}{\partial t}_p = \sum_{nb} a_{nb}((w_l)_{nb} - (w_l)_p) - \\ A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B) + F_{gs}(w_s - w_g)\Delta V. \tag{6.24}$$

w_l is then approximated using the POD basis functions and time coefficients with the reconstruction equation (B):

$$w_l(\mathbf{x}, t) \simeq \phi_o^{w_l}(\mathbf{x}) + \sum_{i=1}^{m^{w_l}} \alpha_i^{w_l}(t)\phi_i^{w_l}(\mathbf{x}). \tag{6.25}$$

m is the number of modes used to approximate w_l. Substituting (6.25) into (6.24) yields:

$$(\varepsilon_l \rho_l)^o \Delta V \sum_{i=1}^{m^{w_l}} \dot{\alpha}_i^{w_l} \phi_i^{w_l} = \sum_{nb} a_{nb} \sum_{i=0}^{m^{w_l}} ((\phi_i^{w_l})_{nb} + (\phi_i^{w_l}))_p) - \\ A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B) + \\ F_{gs}(\sum_{i=0}^{m^{w_s}} \alpha_i^{w_s}(\phi_i^{w_s})_p - \sum_{i=0}^{m^{w_g}} \alpha_i^{w_g}(\phi_i^{w_g})_p)\Delta V \tag{6.26}$$

With (5.23) - (5.28), (6.26) becomes:

$$
\begin{aligned}
&(\varepsilon_l\rho_l)^o\Delta V\sum_{i=1}^{m^{w_l}}\dot{\alpha}_i^{w_l}\phi_i^{w_l} = \\
&-\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}}\xi_e^{w_l}(\varepsilon_l\rho_l)_E A_e(\phi_j^{u_l})_e(\phi_j^{w_l})_e((\phi_i^{w_l})_p-(\phi_i^{w_l}))_e)\alpha_i^{w_l}\alpha_j^{u_l}+ \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}}\xi_w^{w_l}(\varepsilon_l\rho_l)_W A_w(\phi_j^{u_l})_w(\phi_j^{w_l})_w((\phi_i^{w_l})_p-(\phi_i^{w_l}))_w)\alpha_i^{w_l}\alpha_j^{u_l}- \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}}\xi_n^{w_l}(\varepsilon_l\rho_l)_N A_n(\phi_j^{v_l})_n(\phi_j^{w_l})_n((\phi_i^{w_l})_p-(\phi_i^{w_l}))_n)\alpha_i^{w_l}\alpha_j^{v_l}+ \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}}\xi_s^{w_l}(\varepsilon_l\rho_l)_S A_s(\phi_j^{v_l})_s(\phi_j^{w_l})_S s((\phi_i^{w_l})_p-(\phi_i^{w_l}))_s)\alpha_i^{w_l}\alpha_j^{v_l}- \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}}\xi_t^{w_l}(\varepsilon_l\rho_l)_T A_t(\phi_j^{w_l})_t(\phi_j^{w_l})_t((\phi_i^{w_l})_p-(\phi_i^{w_l}))_t)\alpha_i^{w_l}\alpha_j^{w_l}+ \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}}\xi_b^{w_l}(\varepsilon_l\rho_l)_B A_b(\phi_j^{w_l})_b(\phi_j^{w_l})_b((\phi_i^{w_l})_p-(\phi_i^{w_l}))_b)\alpha_i^{w_l}\alpha_j^{w_l}- \\
&\sum_{i=0}^{m^{w_l}}\frac{avg_h((\mu_l)_E)A_e}{\Delta x}((\phi_i^{w_l})_p-(\phi_i^{w_l})_e)\alpha_i^{w_l}+\sum_{i=0}^{m^{w_l}}\frac{avg_h(\mu_l)_W)A_w}{\Delta x}((\phi_i^{w_l})_p-(\phi_i^{w_l})_w)\alpha_i^{w_l}+ \\
&\sum_{i=0}^{m^{w_l}}\frac{avg_h((\mu_l)_N)A_n}{\Delta y}((\phi_i^{w_l})_p-(\phi_i^{w_l})_n)\alpha_i^{w_l}+\sum_{i=0}^{m^{w_l}}\frac{avg_h((\mu_l)_S)A_s}{\Delta y}((\phi_i^{w_l})_p-(\phi_i^{w_l})_s)\alpha_i^{w_l}+ \\
&\sum_{i=0}^{m^{w_l}}\frac{avg_h((\mu_l)_T)A_t}{\Delta z}((\phi_i^{w_l})_p-(\phi_i^{w_l})_t)\alpha_i^{w_l}+\sum_{i=0}^{m^{w_l}}\frac{avg_h((\mu_l)_B)A_b}{\Delta z}((\phi_i^{w_l})_p-(\phi_i^{w_l})_b)\alpha_i^{w_l}+ \\
&A_p(\varepsilon_l)_p((P_g)_T-(P_g)_B)+F_{gs}(\sum_{i=0}^{m^{w_s}}\alpha_i^{w_s}(\phi_i^{w_s})_p-\sum_{i=0}^{m^{w_g}}\alpha_i^{w_g}(\phi_i^{w_g})_p)\Delta V. \qquad (6.27)
\end{aligned}
$$

$avg_h((\mu_l)_{nb})$ is the harmonic average of the viscosity given by Syamlal [34, p. 17, Eq. 3.6-3.7]. (6.27) is known as the discretized finite volume z-momentum equation approximated with proper orthogonal decomposition. (6.27) is projected onto the basis functions $\phi_k^{w_l}$, where $k \in [1, m^{w_l}]$:

$$
\begin{aligned}
&(\varepsilon_l\rho_l)^o \Delta V \sum_{i=1}^{m^{w_l}} \dot{\alpha}_i^{w_l}(\phi_i^{w_l}, \phi_k^{w_l}) = \\
&-\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \xi_e^{w_l}(\varepsilon_l\rho_l)_E A_e(\phi_j^{u_l})_e(\phi_j^{w_l})_e(((\phi_i^{w_l})_p - (\phi_i^{w_l}))_e), \phi_k^{w_l})\alpha_i^{w_l}\alpha_j^{u_l} + \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \xi_w^{w_l}(\varepsilon_l\rho_l)_W A_w(\phi_j^{u_l})_w(\phi_j^{w_l})_w(((\phi_i^{w_l})_p - (\phi_i^{w_l}))_w), \phi_k^{w_l})\alpha_i^{w_l}\alpha_j^{u_l} - \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \xi_n^{w_l}(\varepsilon_l\rho_l)_N A_n(\phi_j^{v_l})_n(\phi_j^{w_l})_n(((\phi_i^{w_l})_p - (\phi_i^{w_l}))_n), \phi_k^{w_l})\alpha_i^{w_l}\alpha_j^{v_l} + \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \xi_s^{w_l}(\varepsilon_l\rho_l)_S A_s(\phi_j^{v_l})_s(\phi_j^{w_l})_s(((\phi_i^{w_l})_p - (\phi_i^{w_l}))_s), \phi_k^{w_l})\alpha_i^{w_l}\alpha_j^{v_l} - \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \xi_t^{w_l}(\varepsilon_l\rho_l)_T A_t(\phi_j^{w_l})_t(\phi_j^{w_l})_t(((\phi_i^{w_l})_p - (\phi_i^{w_l}))_t), \phi_k^{w_l})\alpha_i^{w_l}\alpha_j^{w_l} + \\
&\sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \xi_b^{w_l}(\varepsilon_l\rho_l)_B A_b(\phi_j^{w_l})_b(\phi_j^{w_l})_b(((\phi_i^{w_l})_p - (\phi_i^{w_l}))_b), \phi_k^{w_l})\alpha_i^{w_l}\alpha_j^{w_l} - \\
&\sum_{i=0}^{m^{w_l}} \frac{avg_h((\mu_l)_E)A_e}{\Delta x}(((\phi_i^{w_l})_p - (\phi_i^{w_l})_e), \phi_k^{w_l})\alpha_i^{w_l} + \\
&\sum_{i=0}^{m^{w_l}} \frac{avg_h((\mu_l)_W)A_w}{\Delta x}(((\phi_i^{w_l})_p - (\phi_i^{w_l})_w), \phi_k^{w_l})\alpha_i^{w_l} + \\
&\sum_{i=0}^{m^{w_l}} \frac{avg_h((\mu_l)_N)A_n}{\Delta y}(((\phi_i^{w_l})_p - (\phi_i^{w_l})_n), \phi_k^{w_l})\alpha_i^{w_l} + \\
&\sum_{i=0}^{m^{w_l}} \frac{avg_h((\mu_l)_S)A_s}{\Delta y}(((\phi_i^{w_l})_p - (\phi_i^{w_l})_s), \phi_k^{w_l})\alpha_i^{w_l} + \\
&\sum_{i=0}^{m^{w_l}} \frac{avg_h((\mu_l)_T)A_t}{\Delta z}(((\phi_i^{w_l})_p - (\phi_i^{w_l})_t), \phi_k^{w_l})\alpha_i^{w_l} + \\
&\sum_{i=0}^{m^{w_l}} \frac{avg_h((\mu_l)_B)A_b}{\Delta z}(((\phi_i^{w_l})_p - (\phi_i^{w_l})_b), \phi_k^{w_l})\alpha_i^{w_l} + \\
&(A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B), \phi_k^{w_l}) + \\
&(F_{gs}(\sum_{i=0}^{m^{w_s}} \alpha_i^{w_s}(\phi_i^{w_s})_p - \sum_{i=0}^{m^{w_g}} \alpha_i^{w_g}(\phi_i^{w_g})_p)\Delta V, \phi_k^{w_l}).
\end{aligned}
\tag{6.28}
$$

(6.28) can be rewritten as:

$$\begin{aligned}\check{A}^{w_l}_{ij}\dot{\alpha}^{w_l}_k =& \sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{u_l}} \check{F}^{w_l}_{kij}\alpha^{w_l}_i\alpha^{u_l}_i + \sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{v_l}} \check{G}^{w_l}_{kij}\alpha^{w_l}_i\alpha^{v_l}_j + \\ & \sum_{i=0}^{m^{w_l}}\sum_{j=0}^{m^{w_l}} \check{H}^{w_l}_{kij}\alpha^{w_l}_i\alpha^{w_l}_j \sum_{i=0}^{m^{w_l}} \check{I}^{w_l}_{ki}\alpha^{w_l}_i + \check{S}^{w_l}\end{aligned} \tag{6.29}$$

where

$$\check{A}^{w_l}_{ij} = \delta_{ij}\cdot((\varepsilon_l\rho_l)^o\Delta V\phi^{w_l}_i, \phi^{w_l}_k)$$

$$\begin{aligned}\check{F}^{w_l}_{kij} =& -(\xi^{w_l})_e(\varepsilon_l\rho_l)_E A_e(\phi^{u_l}_j)_e(((\phi^{w_l}_i)_p - (\phi^{w_l}_i)_e), \phi^{w_l}_k) + \\ & (\xi^{w_l})_w(\varepsilon_l\rho_l)_W A_w(\phi^{u_l}_j)_w(((\phi^{w_l}_i)_p - (\phi^{w_l}_i)_w), \phi^{w_l}_k),\end{aligned}$$

$$\begin{aligned}\check{G}^{w_l}_{kij} =& -(\xi^{w_l})_n(\varepsilon_l\rho_l)_N A_n(\phi^{u_l}_j)_n(((\phi^{w_l}_i)_p - (\phi^{w_l}_i)_n), \phi^{w_l}_k) + \\ & (\xi^{w_l})_s(\varepsilon_l\rho_l)_S A_s(\phi^{u_l}_j)_s(((\phi^{w_l}_i)_p - (\phi^{w_l}_i)_s), \phi^{w_l}_k),\end{aligned}$$

$$\begin{aligned}\check{H}^{w_l}_{kij} =& -(\xi^{w_l})_t(\varepsilon_l\rho_l)_T A_t(\phi^{u_l}_j)_t(((\phi^{w_l}_i)_p - (\phi^{w_l}_i)_t), \phi^{w_l}_k) + \\ & (\xi^{w_l})_b(\varepsilon_l\rho_l)_B A_b(\phi^{u_l}_j)_b(((\phi^{w_l}_i)_p - (\phi^{w_l}_i)_b), \phi^{w_l}_k),\end{aligned}$$

$$
\begin{aligned}
\check{I}^{w_l}_{ki} =& \frac{avg_h((\mu_l)_E)A_e}{\Delta x}(((\phi^{w_l}_i)_p - (\phi^{w_l}_i))_e), \phi^{w_l}_k) + \\
& \frac{avg_h((\mu_l)_W)A_w}{\Delta x}(((\phi^{w_l}_i)_p - (\phi^{w_l}_i))_w), \phi^{w_l}_k) + \\
& \frac{avg_h((\mu_l)_N)A_n}{\Delta y}(((\phi^{w_l}_i)_p - (\phi^{w_l}_i))_n), \phi^{w_l}_k) + \\
& \frac{avg_h((\mu_l)_S)A_s}{\Delta y}(((\phi^{w_l}_i)_p - (\phi^{w_l}_i))_s), \phi^{w_l}_k) + \\
& \frac{avg_h((\mu_l)_T)A_t}{\Delta z}(((\phi^{w_l}_i)_p - (\phi^{w_l}_i))_t), \phi^{w_l}_k) + \\
& \frac{avg_h((\mu_l)_B)A_b}{\Delta z}(((\phi^{w_l}_i)_p - (\phi^{w_l}_i))_b), \phi^{w_l}_k),
\end{aligned}
$$

$$
\check{S}^{w_l} = (A_p(\varepsilon_l)_p((P_g)_T - (P_g)_B) + F_{gs}(\sum_{i=0}^{m^{w_s}} \alpha^{w_s}_i(\phi^{w_s}_i)_p - \sum_{i=0}^{m^{w_g}} \alpha^{w_g}_i(\phi^{w_g}_i)_p)\Delta V, \phi^{w_l}_k).
$$

(6.29) has been reduced from a large number of equations to a set of m^{w_l} ODEs. It can be written in a simpler form:

$$
\tilde{A}^{w_l}\alpha^{w_l} = \tilde{B}^{w_l}, \tag{6.30}
$$

where

$$
\tilde{A}^{w_l} = \left((a^{w_l}_p \phi^{w_l}_j - \sum_{nb} a^{w_l}_{nb}(\phi_j)_{nb}), \phi^{w_l}_k \right),
$$

$$
\bar{B}^{w_l} = (b^{w_l}_p + A_P((P_g)_T - (P_g)_B) + F_{gs}(\sum_{i=0}^{m^{w_s}} \alpha^{w_s}_i(\phi^{w_s}_i)_p - \sum_{i=0}^{m^{w_g}} \alpha^{w_g}_i(\phi^{w_g}_i)_p)\Delta V, \phi^{w_l}_k).
$$

The dimensions of $\tilde{A}$ and $\tilde{B}$ are $m^{w_l} \times m^{w_l}$ and $m^{w_l} \times 1$ respectively. ODEV uses a similar iterative, time marching algorithm to that of MFIX implemented similarly to ODEti and ODEg. The solution algorithm for ODEV is outlined below:

- With the time coefficients read-in from the input file or from a previous iteration and the basis functions generated from PODDEC, the field variables, u_g, u_s, v_g, v_s, w_g, w_s, P_g, and ε_g, are reconstructed.

- The fluid and solid physical properties not defined in the input are computed.

Using the ideal gas law, the gas density ρ_g is computed. The gas-solid drag F_{gs} is computed. The solid viscosity μ_s is calculated.

- The uncorrected velocity field time coefficients $\alpha_i^{u_g}(t)$, $i \in [1, m^{u_g}]$, $\alpha_i^{u_s}(t)$, $i \in [1, m^{u_s}]$, $\alpha_i^{v_g}(t)$, $i \in [1, m^{v_g}]$, and $\alpha_i^{v_s}(t)$, $i \in [1, m^{v_s}]$, defined by the linear system of equations [47, p. 248, Eq. 30-31], and (6.30) for the uncorrected z-direction velocity time coefficients $\alpha_i^{w_g}(t)$, $i \in [1, m^{w_g}]$ and $\alpha_i^{w_s}(t)$, $i \in [1, m^{w_s}]$, are solved. m denotes the number of modes used for each variable. The time coefficients are called "uncorrected" as they are computed using the previous pressure field and void fraction. The gas and solid velocities will be corrected using the gas pressure correction and solids volume fraction correction equations, respectively.

- The linear system of equations describing the time coefficients of the gas pressure correction equation, $\alpha_i^{p_g'}(t)$, $i \in [1, m^{p_g}]$, [47, p. 249, Eq. 32] is solved.

- The gas pressure and velocities' time coefficients are corrected using the time coefficients computed from the gas pressure correction equation.

- The linear system of equations describing the time coefficients of the solid volume fraction correction equation $\alpha_i^{\varepsilon_s'}(t)$, $i \in [1, m^{\varepsilon_s}]$ [2].

- The time coefficients of the void fraction and the solid velocities are corrected using the time coefficients computed from the solid volume fraction correction equation.

- Convergence of the time coefficients computed from the linear systems is checked. If the solution has converged, ODEV is advanced forward in time to the next time step. Otherwise, the solutions are iterated until convergence is reached.

ODEV requires the basis functions extracted from the eight field variable databases generated by MFIX. ODEV solves for the time coefficients of these field variables: u_g, u_s, v_g, v_s, w_g, w_s, P_g, and ε_g. A sample input file for ODEV is given in Appendix E. Table VII shows the subroutines created for ODEV. These subroutines are important because they perform the projection of the governing PDEs onto the z-velocity phase velocity basis functions, solve the linear system of equations for the time coefficients of w_l, and reconstruct the solution field of w_l using the time coefficients α^{w_l}. A table is not provided for the subroutines modified from the previous ROMs to implement ODEV as all of the subroutines had to be modified to incorporate the third dimension.

Table VII: Most important routines created for ODEV.

Subroutine	New Subroutine Description
calc_ab_w_g/s.f	Projects the z-momentum equation system onto the basis functions extracted from the z-direction velocity snapshots
w_phi_prod.f	Calculates the constant basis function products outside the iteration loop
reconstruct_w_g/s.f	Reconstructs the z-direction velocity from the basis functions and time coefficients
source_w_g/s.f	Calculates source terms for z-momentum equations
conv_dif_w_g/s.f	Calculates convection/diffusion terms for z-momentum equations

E. Summary

The chapter also presented three reduced-order models based on POD for multiphase flows in fluidized beds. The first ROM, ODEti, models non-isothermal flow in fluidized beds. The second ROM, ODEg, models isothermal flow with granular energy in fluidized beds. The third ROM, ODEV, simulates three-dimensional isothermal flow in fluidized beds.

The discretized energy, granular energy, and z-momentum equations were projected onto the basis functions ϕ^{T_l}, ϕ^{Θ}, and ϕ^{w_l} respectively. Three systems of equations were obtained:

$$\tilde{A}^{T_l}\alpha^{T_l} = \tilde{B}^{T_l}, \tag{6.31}$$

$$\tilde{A}^{\Theta}\alpha^{\Theta} = \tilde{B}^{\Theta}, \tag{6.32}$$

$$\tilde{A}^{w_l}\alpha^{w_l} = \tilde{B}^{w_l}, \tag{6.33}$$

The algorithm for the POD-based ROMs were presented. The subroutines that were created or modified were listed for the energy, granular energy, and three-dimensional isothermal ROMs. Results for the POD-based ROMs are presented in the next chapter.

CHAPTER VII

RESULTS

A. Introduction

This chapter presents the results of the POD-based ROMs and the morphological feature detection algorithm. The chapter presents the results of POD-based ROMs for three cases of two-phase flows: non-isothermal flow, isothermal flow with granular energy, and three-dimensional isothermal flow. The chapter also presents the results of the morphological feature detection algorithm for three different flows that exhibit moving discontinuities: isothermal bubbling fluidized beds, transonic supercritical airfoils, and cavity flows in turbomachinery seals.

B. POD-Based Reduced-Order Models for Two-Phase Flows

This section presents results for three cases: (1) non-isothermal flow, (2) isothermal flow with granular energy, and (3) three-dimensional isothermal flow. None of the cases shown use the acceleration techniques presented by Richardson [2].

1. Case I: Non-Isothermal Flow

Case I is a two-phase model for non-isothermal flow. The model consists of one solids and one gas phase. The geometry, boundary conditions, and computational domain of case I are shown in Figure 12. The computational domain is a uniform rectangular grid. As described previously on page 45 air enters through the lower boundary of the bed. A central jet with a velocity $v_1 = 12.6$ cm/s is surrounded on both sides by a distributor with a velocity $v_2 = 1.0$ cm/s. The central jet is injected at a temperature of 450 K, while the distributor flow is injected at a lower temperature of 297 K. The central jet is approximately 1-cm wide. The particle diameter is constant throughout the bed, and each phase (gas and solids) is modeled as a non-reacting single species. The model parameters for this case are listed in Table VIII.

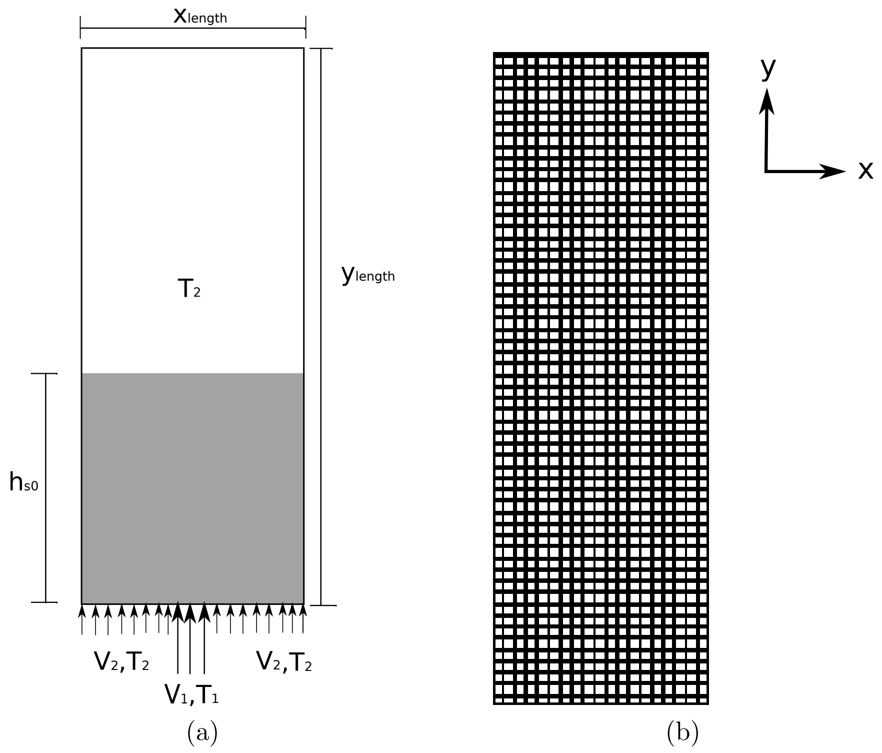


Figure 12: Case I: (a) geometry, boundary conditions, and (b) computational grid.

POD is applied to the database of snapshots computed by MFIX for eight field variables modeled using the parameters of Table VIII. The first six POD basis functions of ε_g, p_g, u_g, u_s, v_g, v_s, T_g, and T_s are shown in Figures 13 - 20, respectively. The modes are normalized by maximum values. ϕ_0 is the basis function for the average mode.

Tables IX and X show the percentage of energy captured by each mode and the cumulative energy captured by the sum over the modes for each field variable.

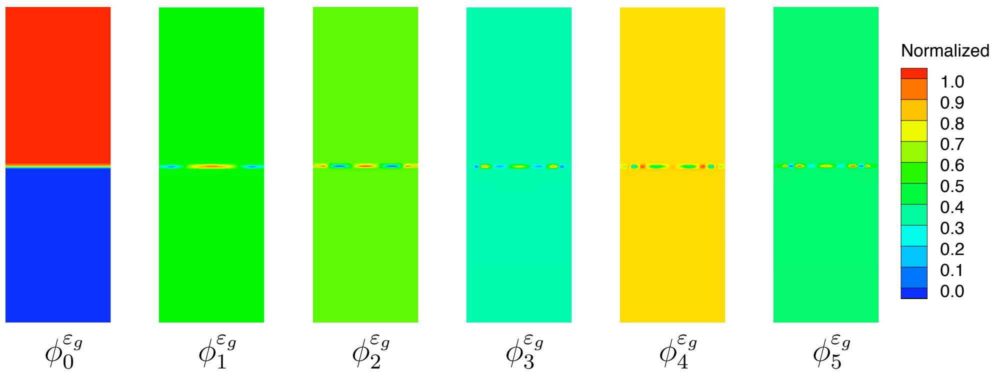


Figure 13: Case I: first six basis functions of void fraction, ε_g.

Table VIII: Parameters of case I.

Parameter	Description	Units	
x_{length}	Length of the domain in x-direction	cm	25.4
y_{length}	Length of the domain in y-direction	cm	76.5
i_{max}	Number of cells in x-direction	-	108
j_{max}	Number of cells in y-direction	-	124
v_1, v_2	Gas jet velocities	cm/s	12.6
v_2	Gas distributor velocity	cm/s	1.0
p_{g_s}	Static pressure at outlet	g/(cm/s^2)	1.01×10^6
T_{g_1}	Gas jet temperature	K	450
T_{g_2}	Gas distributor temperature	K	297
μ_{g_0}	Gas viscosity	g/(cm/s)	1.8×10^{-4}
t_{start}	Start time	s	0.0
t_{stop}	Stop time	s	1.0
Δt	Initial time step	s	1.0×10^{-4}
ρ_s	Particle density	g/cm^3	1.0
D_p	Particle diameter	cm	0.05
h_{s0}	Initial height of packed bed	cm	38.25
k_g	Gas phase thermal conductivity	J/(cm K s)	1.0×10^{-10}
k_s	Solids phase thermal conductivity	J/(cm K s)	1.505×10^{-4}
C_{pg}	Gas phase specific heat	cal/(g K)	0.25
C_{ps}	Solids phase specific heat	cal/(g K)	0.310713
ϵ_g^*	Initial void fraction of packed bed	-	0.4

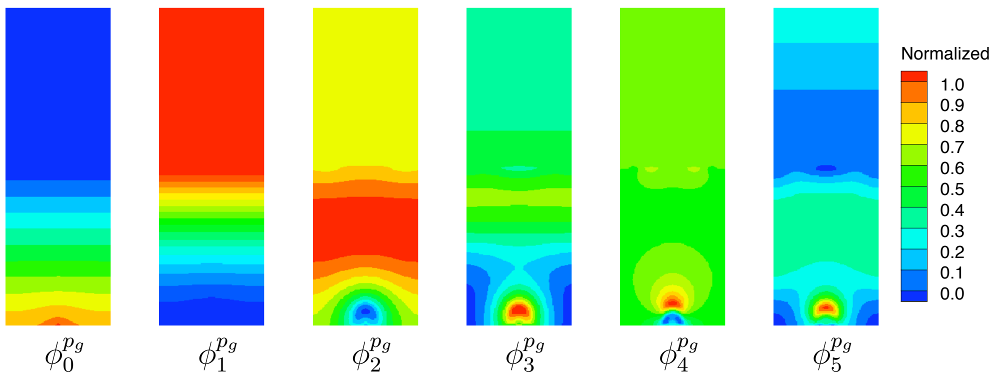


Figure 14: Case I: first six basis functions of gas pressure, p_g.

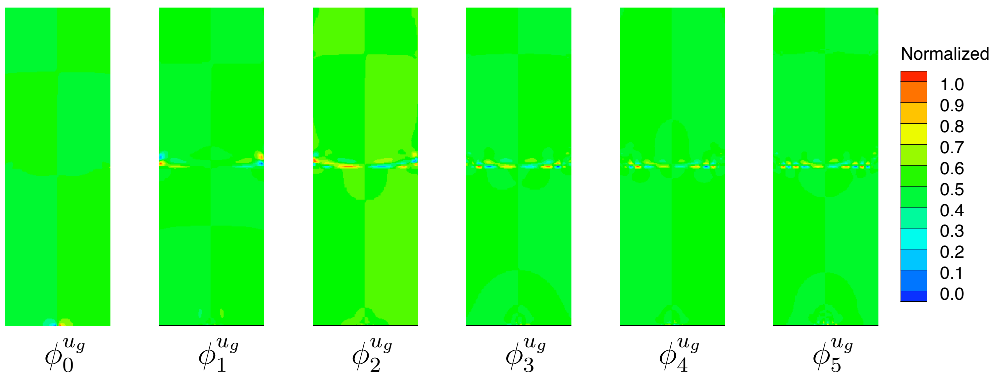


Figure 15: Case I: first six basis functions of gas velocity, u_g.

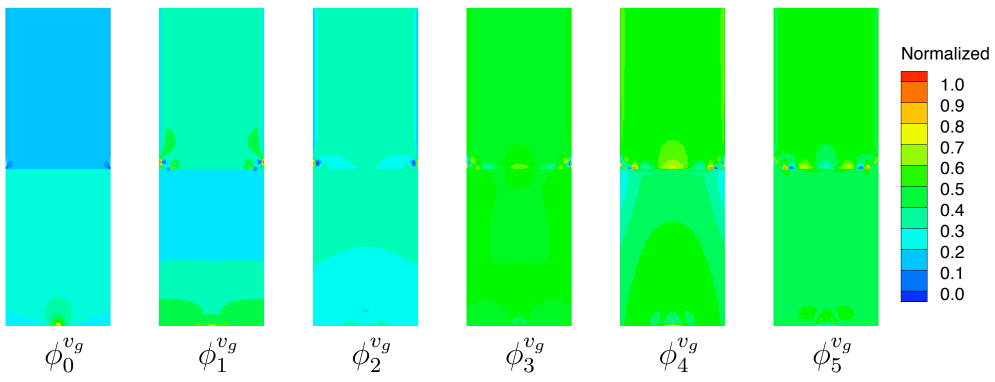


Figure 16: Case I: first six basis functions of gas velocity, v_g.

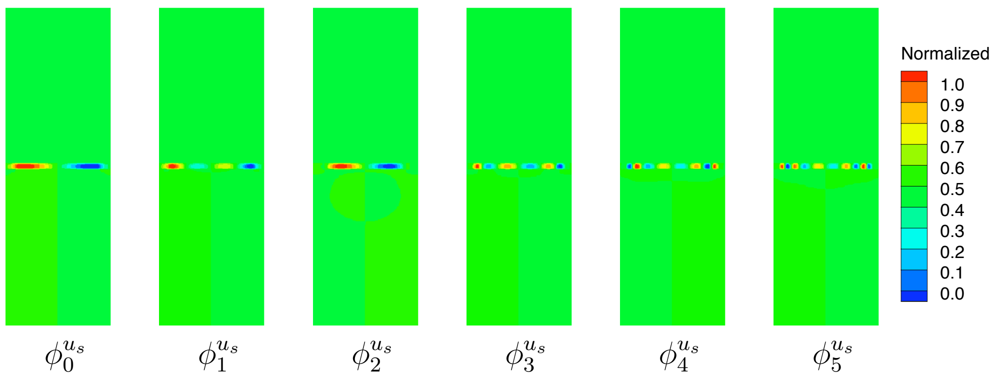


Figure 17: Case I: first six basis functions of solids velocity, u_s.

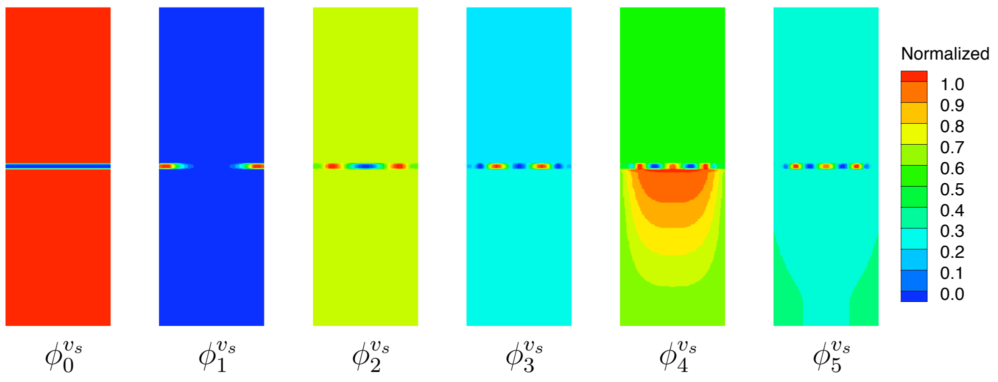


Figure 18: Case I: first six basis functions of solids velocity, v_s.

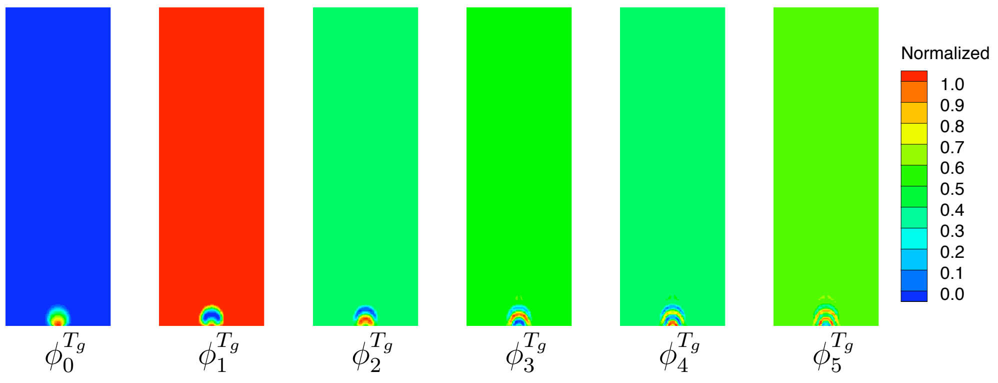


Figure 19: Case I: first six basis functions of gas temperature, T_g.

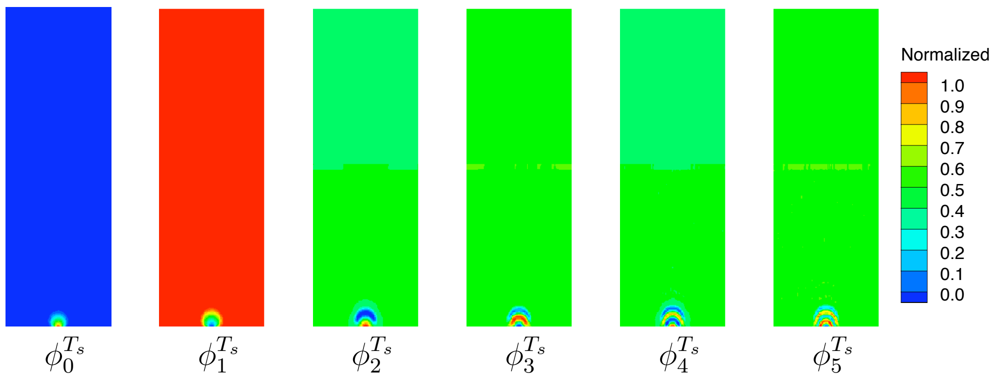


Figure 20: Case I: first six basis functions of solids temperature, T_s.

Table IX: Energy variation for the gas pressure, velocities, and temperature of case I.

Number of Modes	p_g		u_g		v_g		T_g	
	Energy	Total	Energy	Total	Energy	Total	Energy	Total
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1	99.647	99.647	72.554	72.554	69.776	69.776	74.704	74.704
2	0.324	99.971	18.085	90.604	26.723	96.499	15.879	90.583
3	0.024	99.996	4.518	95.159	1.589	98.088	5.446	96.030
4	0.002	99.999	2.119	97.277	0.942	99.408	2.207	98.024
5	0.002	99.999	1.029	98.307	0.377	99.613	0.961	99.199
6	0.000	99.999	0.589	98.895	0.205	99.725	0.432	99.631
7	0.000	99.999	0.424	99.320	0.112	99.798	0.198	99.829
8	0.000	99.999	0.223	99.543	0.073	99.855	0.086	99.916

Table X: Energy variation for the void fraction and solids velocities and temperature of case I.

Number	ε_g		u_s		v_s		T_s	
of Modes	Energy	Total	Energy	Total	Energy	Total	Energy	Total
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1	66.100	66.100	53.276	53.276	73.644	73.644	97.362	97.362
2	23.714	89.814	29.241	82.518	18.500	92.144	2.264	99.627
3	5.294	95.108	10.670	93.189	5.297	97.442	0.290	99.916
4	2.606	97.771	3.837	97.026	1.453	98.895	0.060	99.977
5	1.300	99.014	1.680	98.706	0.749	99.644	0.015	99.992
6	0.552	99.567	0.700	99.407	0.213	99.856	0.004	99.997
7	0.239	99.806	0.293	99.701	0.066	99.923	0.001	99.999
8	0.104	99.910	0.122	99.823	0.037	99.604	0.000	99.999

The number of modes used and the energy spectrum captured by the chosen number of modes for each of the eight field variables is shown in Table XI.

The number of modes was chosen based on the best compromise between accuracy and computational speed-up of the reduced-order model. The mode set selected for this case, shown in XII, was similiar to that of Richardson [2]. However, the number of modes used for T_g and u_g were increased to improve modeling of the gas temperature. The modes selected were determined through numerical experimentation. Using the modes in Table XII, it is possible to demonstrate that the improved POD-based ROM is more computationally efficient than the full-order model. The

Table XI: Cumulative energy captured by the chosen number of modes for case I.

Variable	Number of Modes	Symbol	Cumulative Energy [%]
p_g	2	N_{p_g}	99.971
u_g	5	N_{u_g}	98.307
v_g	5	N_{v_g}	99.613
T_g	12	N_{T_g}	99.994
u_s	8	N_{u_s}	99.823
v_s	6	N_{v_s}	99.856
T_s	3	N_{T_s}	99.916
ε_g	7	N_{ε_g}	99.806

computational times of each is summarized in Table XII. The computation time required for MFIX was 10493 seconds, and for ODEti it was 1286 seconds. The POD-based ROM ODEti, without Richardson's acceleration techniques [2], was 8.16 times faster than the full-order model, MFIX.

One way to assess the accuracy of the ROM is to compare the time coefficients computed by the ROM to those obtained by directly projecting the database of the snapshots onto the basis functions. The time coefficients obtained via direct projection (from (2.19)) gives the "exact" solution of the time coefficients. Comparisons of the two time coefficients for p_g between ODEti and MFIX are given in Figure 21. The first two time coefficients for u_g are compared for ODEti and MFIX in Figure 22. The first four time coefficients of ε_g, u_s, v_g, and v_s are compared for ODEti and MFIX and are shown in Figures 23 - 26. These time coefficients tend to match well with the

Table XII: Summary of case I CPU times for MFIX and ODEti.

Code	N_{p_g}	N_{u_g}	N_{v_g}	N_{T_g}	N_{u_s}	N_{v_s}	N_{T_s}	N_{ε_g}	t_{CPU} [s]
MFIX	-	-	-	-	-	-	-	-	10493
ODEti	2	5	5	12	8	6	3	7	1286

"exact" time coefficients. Some variation of the time coefficients is expected due to the combined effect of the approximations of each of the field variables.

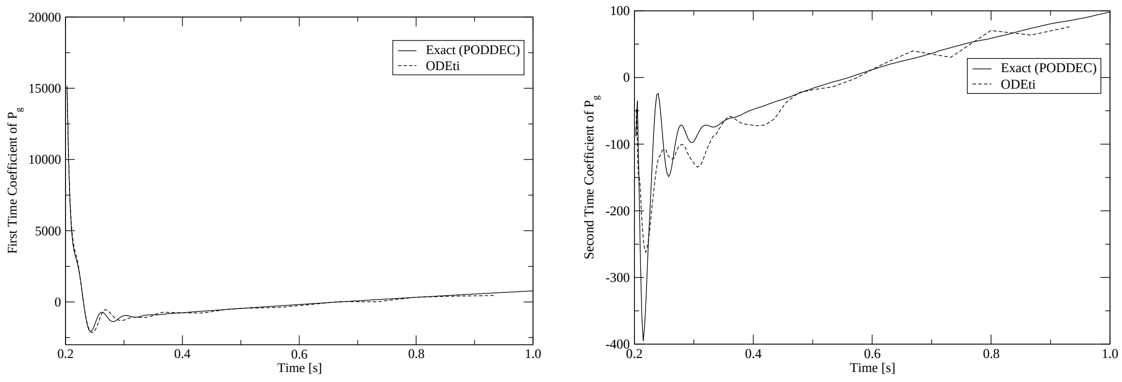


Figure 21: Case I: The time coefficients of gas pressure, p_g, using ODEti and direct projection.

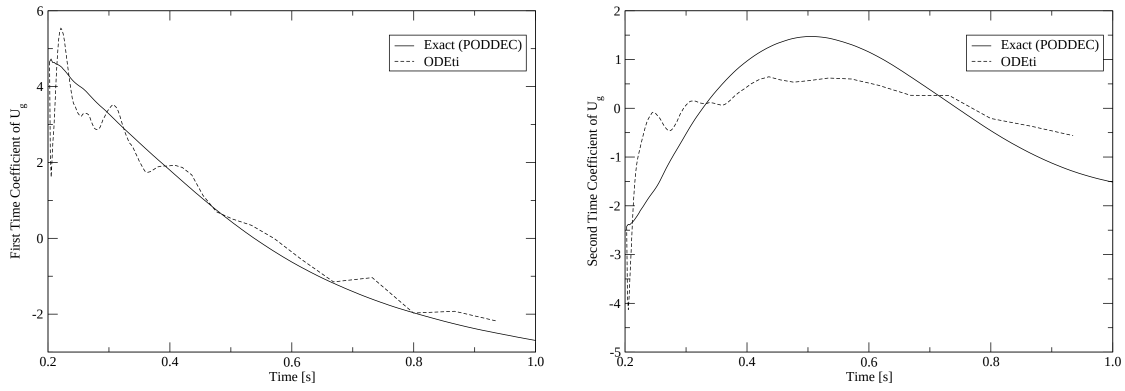


Figure 22: Case I: The first two time coefficients of gas velocity, u_g, using ODEti and direct projection.

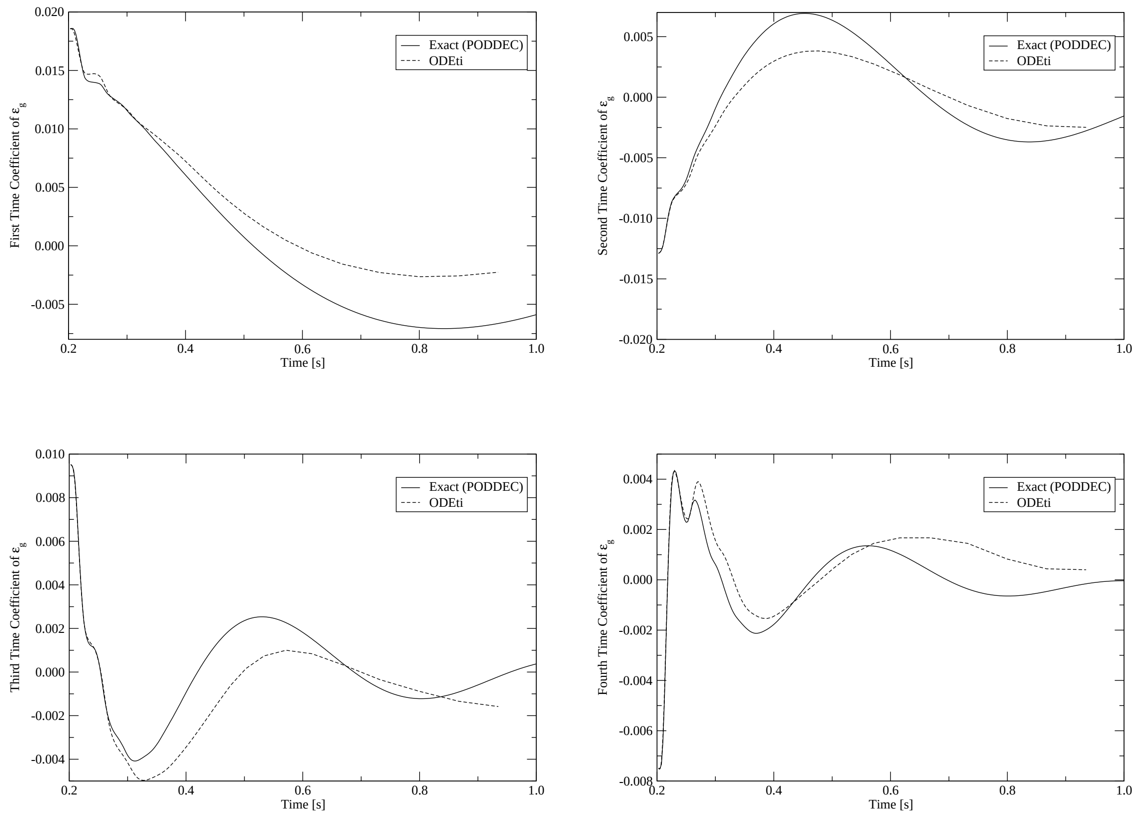


Figure 23: Case I: The first four time coefficients of void fraction,ε_g, using ODEti and direct projection.

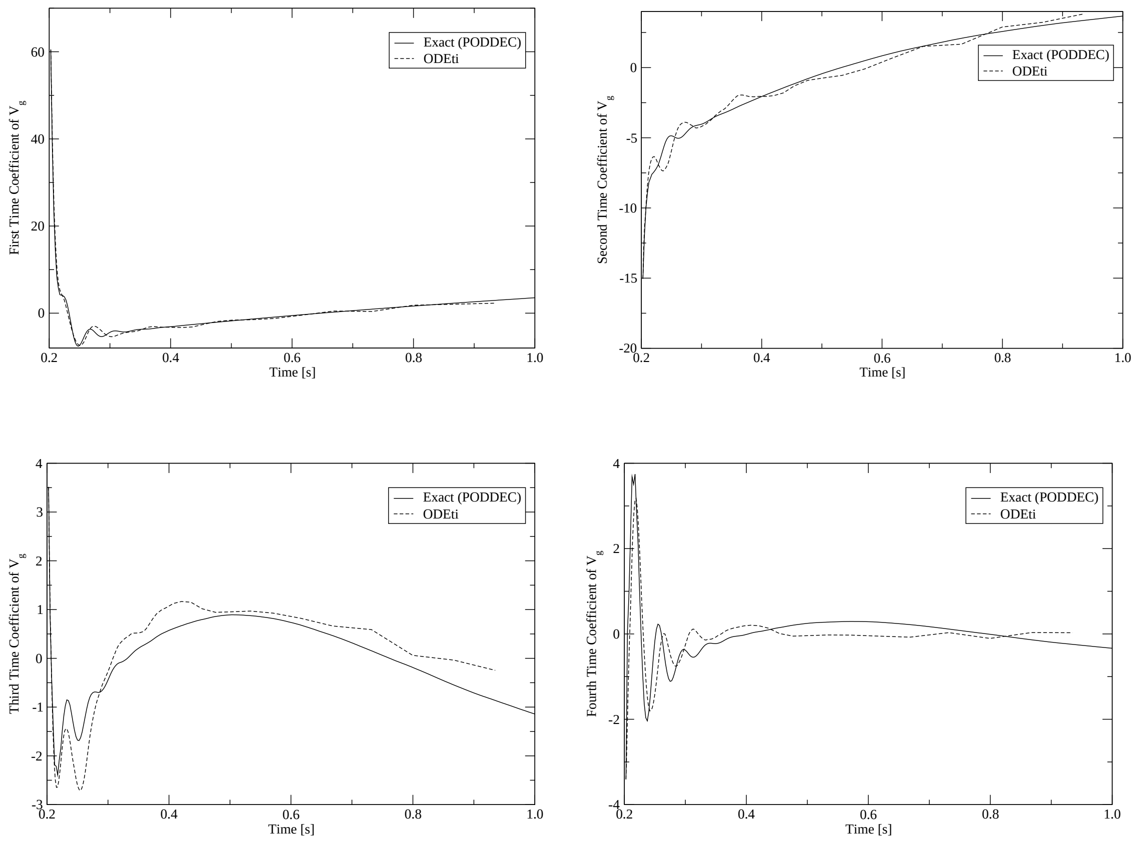


Figure 24: Case I: The first four time coefficients of gas velocity, v_g, using ODEti and direct projection.

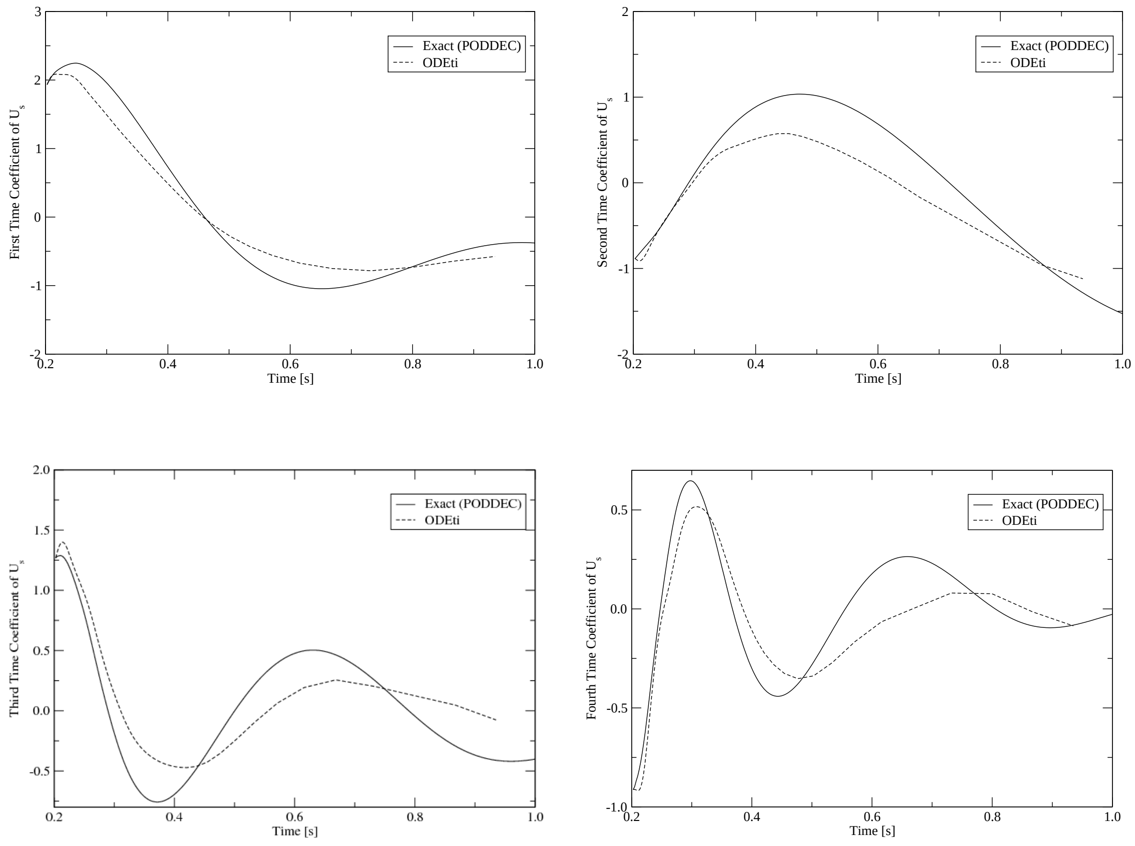


Figure 25: Case I: The first four time coefficients of solids velocity, u_s, using ODEti and direct projection.

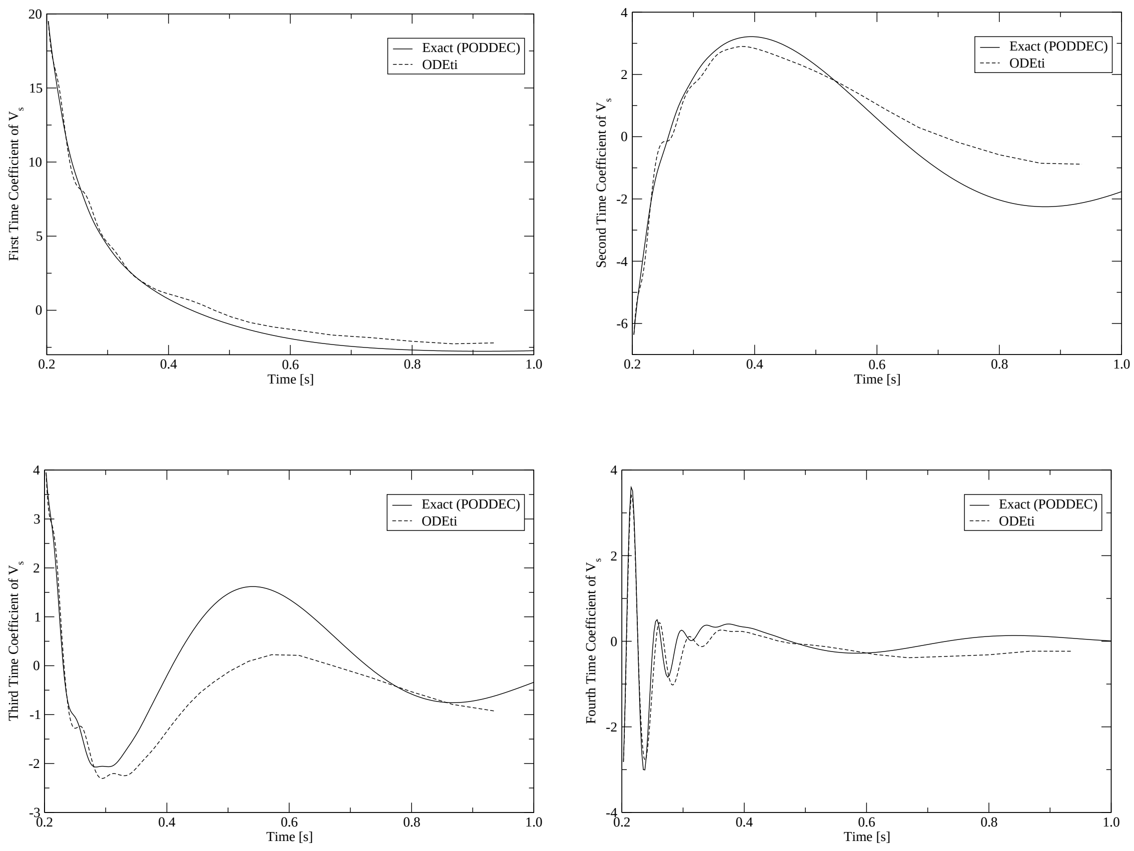


Figure 26: Case I: The first four time coefficients of solids velocity, v_s, using ODEti and direct projection.

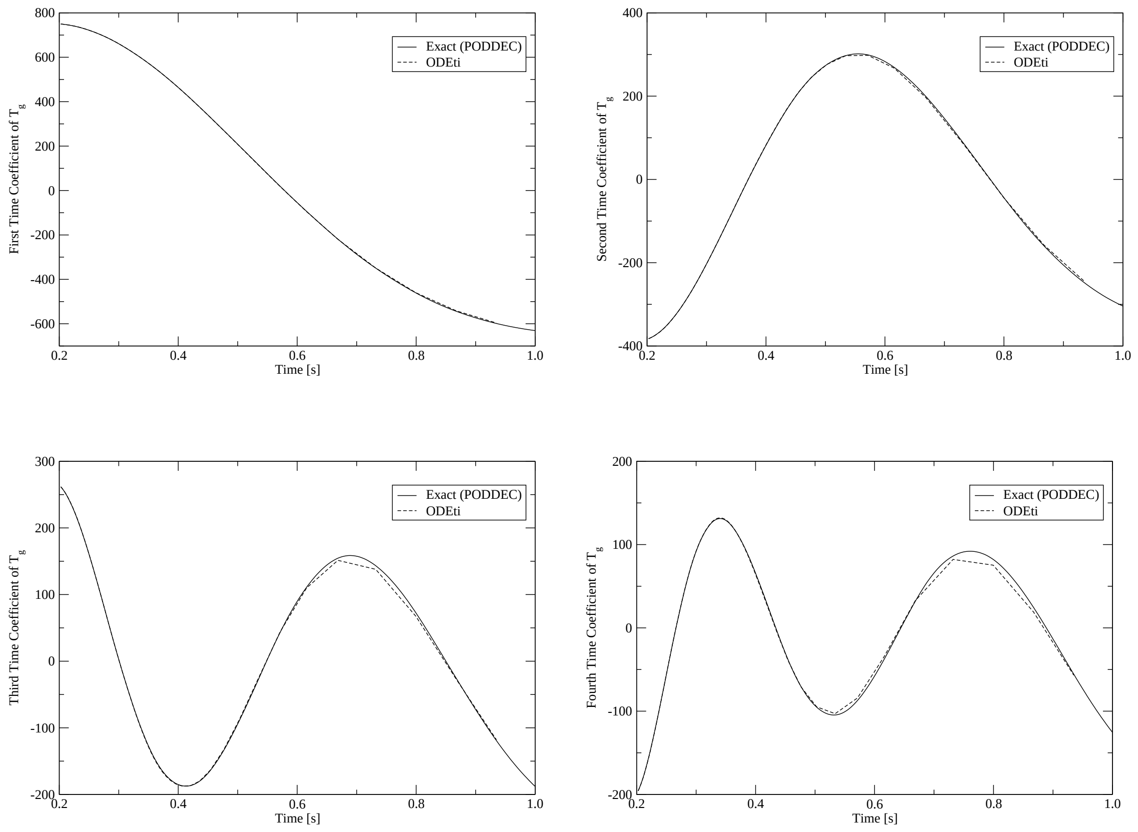


Figure 27: Case I: The first four time coefficients of gas temperature, T_g, using ODEti and direct projection.

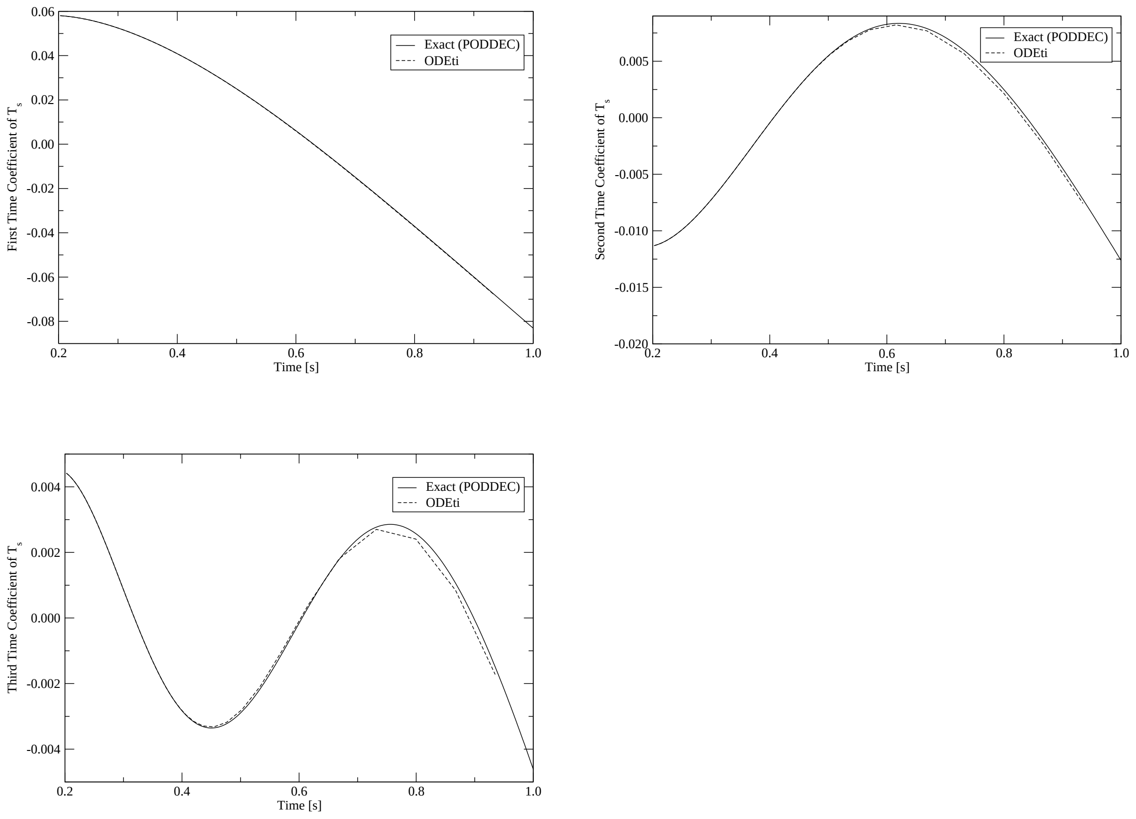


Figure 28: Case I: The three time coefficients of solids temperature, T_s, using ODEti and direct projection.

The time coefficients of T_g and T_s are compared against the "exact" solution obtained by projection (from (2.19)). The time coefficients from ODEti are almost exactly the same as those from MFIX in Figures 27 and 28.

Contour plots at $t = 1.0$ s were generated for each of the field variables at the end of the integration period. The numerical results obtained at the end of the integration period tend to have the largest error, as these errors are accumulated during time integration. A comparison of the results between MFIX and ODEti for the pressure, void fraction, gas and solids velocities, and gas and solids temperature is shown in Figures 29 - 32.

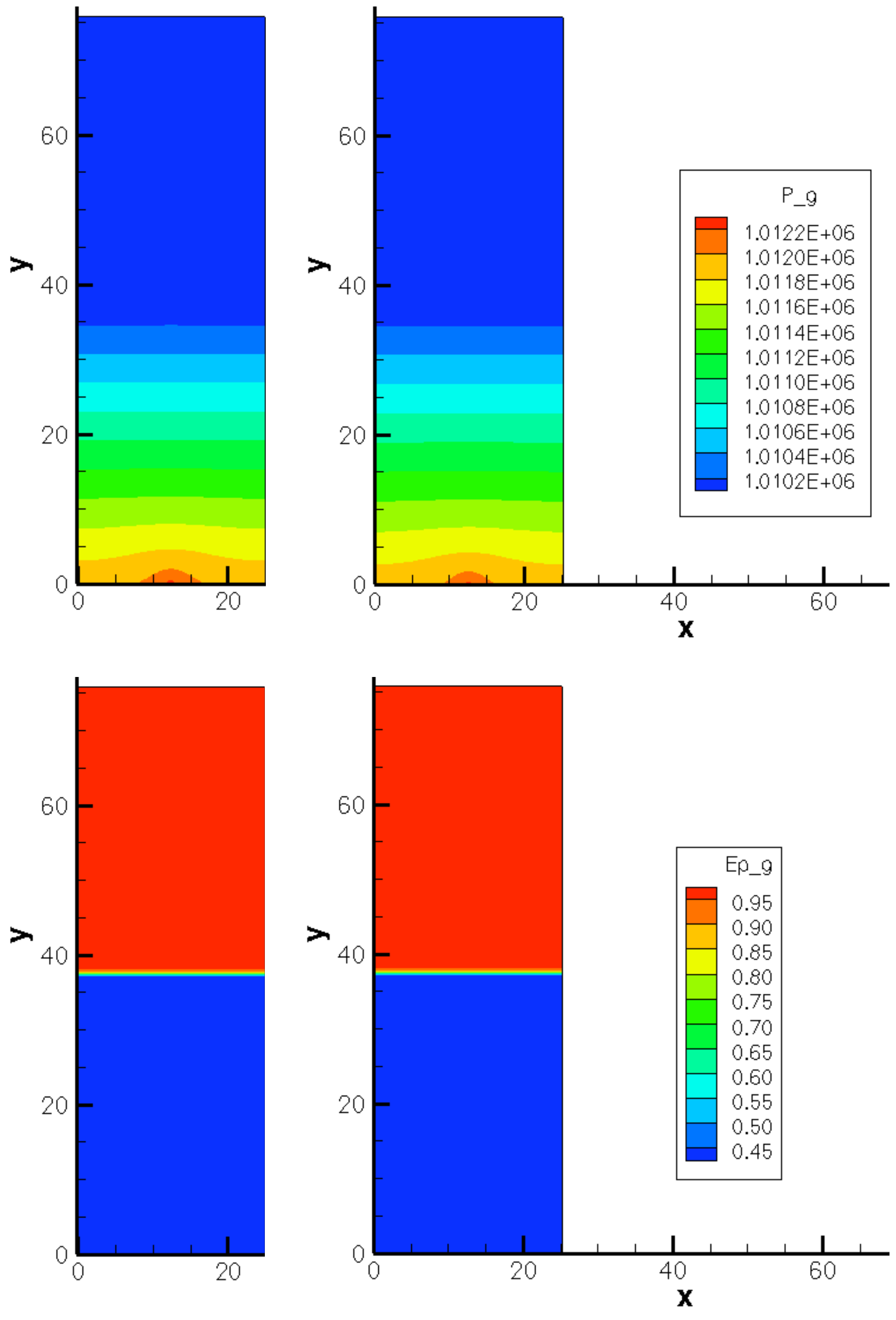
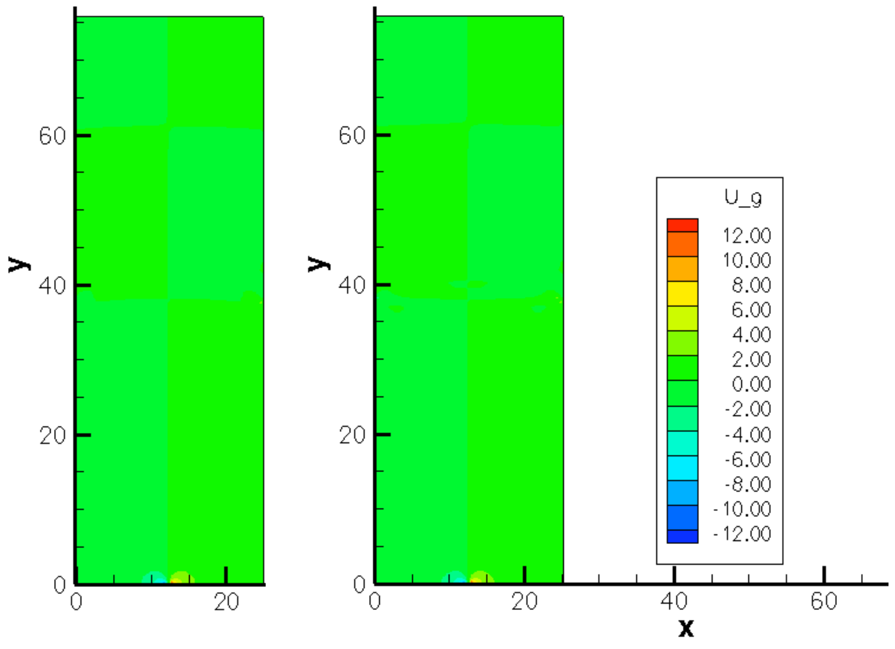


Figure 29: Case I contour plots at $t = 1.0$ s using MFIX (right) and ODEti (left): gas pressure, p_g, (top) and void fraction, ε_g (bottom).



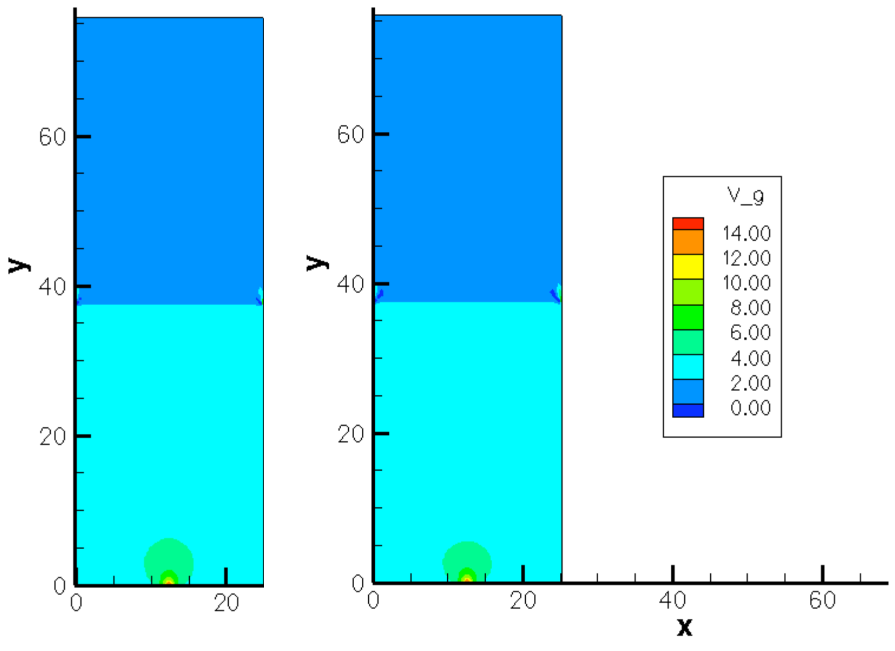


Figure 30: Case I contour plots at t = 1.0 s using MFIX (right) and ODEti (left): gas velocity, u_g, (top) and gas velocity, v_g (bottom).

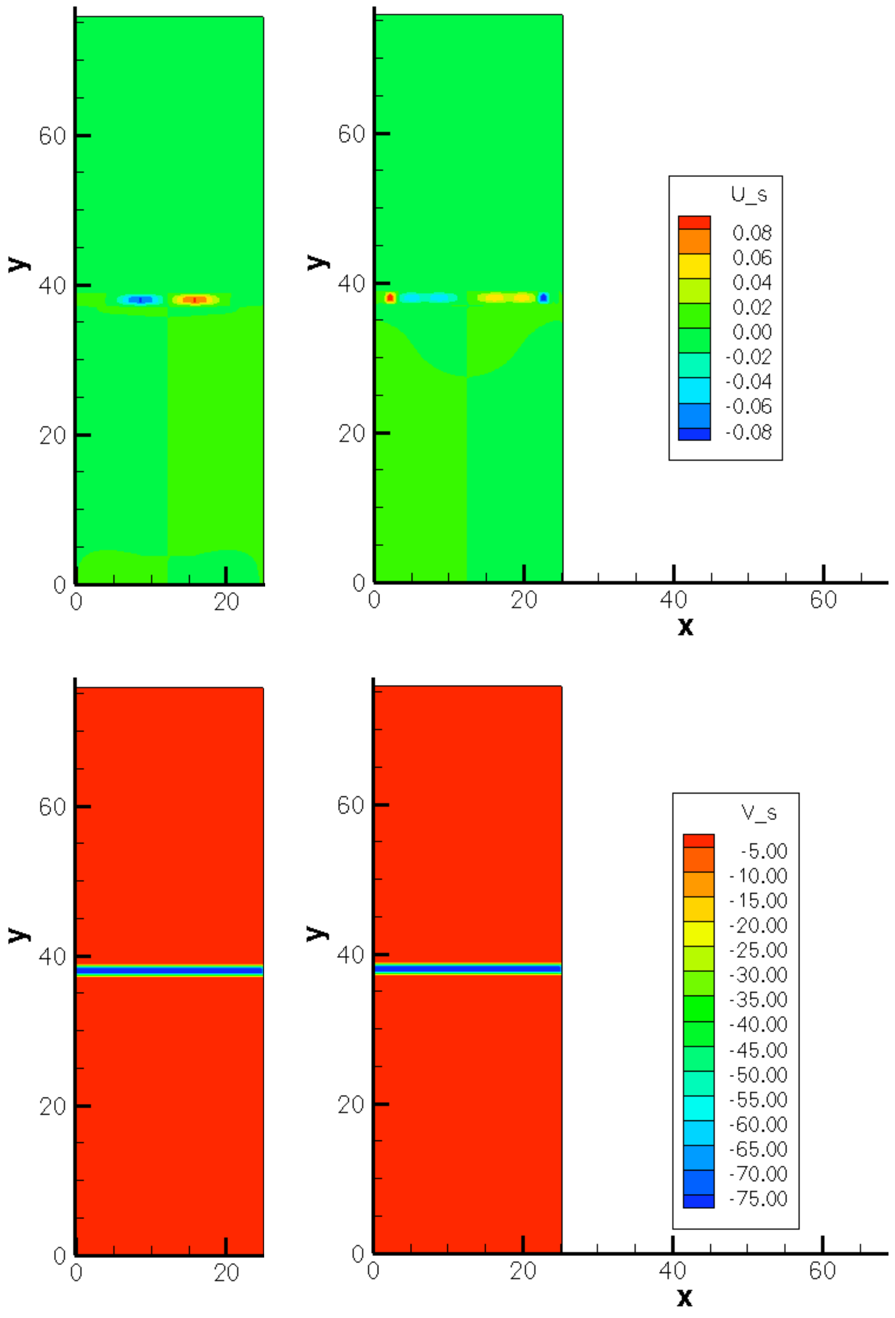


Figure 31: Case I contour plots at t = 1.0 s using MFIX (right) and ODEti (left): solids velocity, u_s, (top) and solids velocity, v_s (bottom).

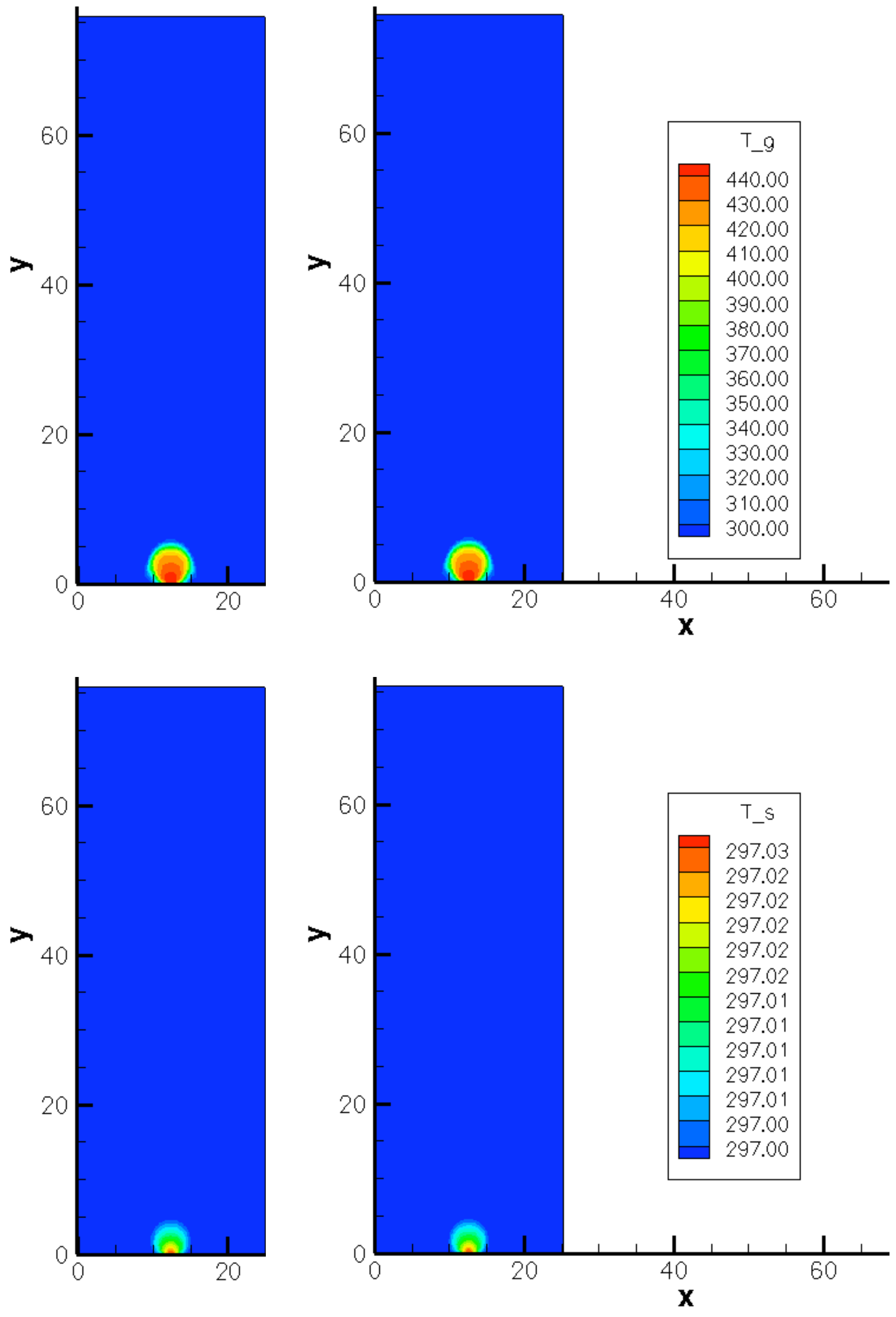


Figure 32: Case I contour plots at t = 1.0 s using MFIX (right) and ODEti (left): gas temperature, T_g, (top) and solids temperature, T_s (bottom).

The contour plots show generally good agreement between ODEti and MFIX for p_g, ε_g, u_g, v_g, and v_s. However, u_s from ODEti does not match well with u_s from MFIX at the top of the bed, where the majority of the particle dynamics present in the bed takes place. This is arises from the unphysical nature of the solids velocity fields computed from MFIX rather than problems approximating the solids velocity fields computed in ODEti. This is easily seen if the solids velocity fields computed by Richardson [2, p. 63] for isothermal flow are compared to those shown in Figure 31 for the same reference condition (except for the increased jet inlet temperature used here). They should be nearly identical; however, both fields share none of the same features. Regardless of the solids velocity fields, the mode set chosen is able to represent the other isothermal variables accurately.

The contour plots of T_g and T_s of ODEti match almost exactly with the temperature fields computed in MFIX. The temperature fields are zoomed for ODEti and MFIX in Figure 33. For the mode set given in Table XII, ODEti achieves a one order of magnitude computational speed-up factor over MFIX and is capable of calculating qualitatively accurate results.

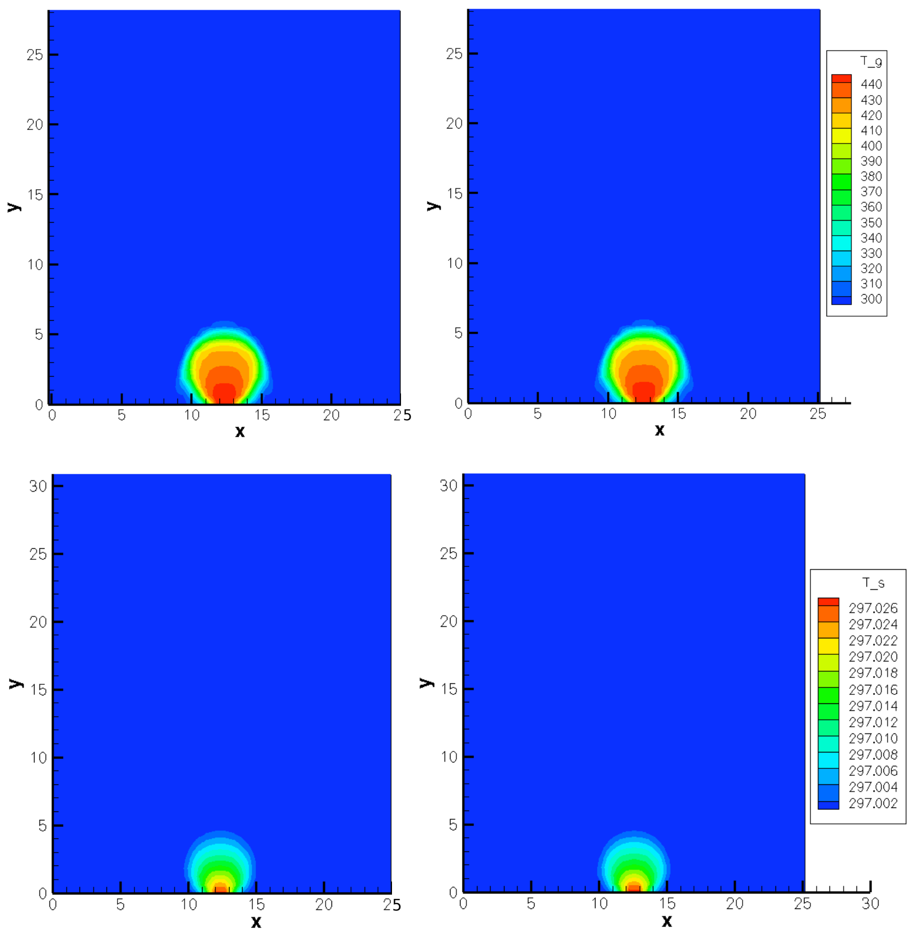


Figure 33: Case I zoomed contour plots at $t = 1.0$ s using MFIX (left), and ODEti (right): gas temperature, T_g, (top) and solids temperature, T_s (bottom).

2. Case II: Isothermal Flow with Granular Energy

Case II is a two-phase model for isothermal flow with granular energy. The model consists of one solids and one gas phase. The geometry and boundary conditions of case II are shown in Figure 34. The computational domain is a uniform rectangular grid as shown in Figure 12 (b). As with case I, air enters through the lower boundary of the bed. A central jet with a velocity $v_1 = 12.6$ cm/s is surrounded on both sides by a distributor with a velocity $v_2 = 1.0$ s. Both streams are injected at a temperature of 297 K. The central jet is approximately 1-cm wide. The particle diameter is constant throughout the bed, and each phase (gas and solids) is modeled as a non-reacting single species. The particles are given an initial granular energy of 10 cm^2/s^2 because it was determined that the full-order model had stability issues when the bed was given an initial granular energy of zero. The model parameters for this case are listed in Table XIII.

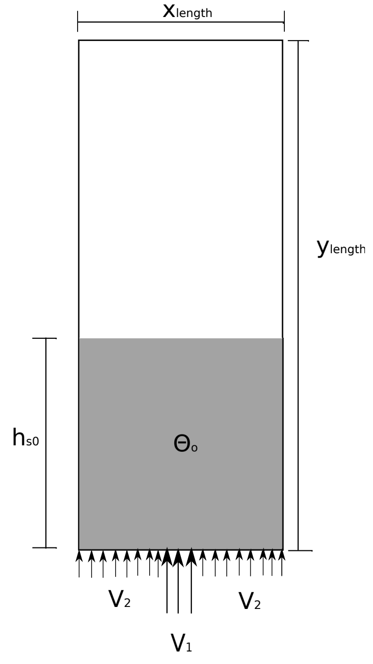


Figure 34: Case II: geometry and boundary conditions.

POD is applied to the database of snapshots computed by MFIX for the seven field variables modeled using the parameters of Table XIII. The first six POD basis functions of ε_g, p_g, u_g, u_s, v_g, v_s, and Θ are shown in Figures 35 - 41, respectively. The modes are normalized by maximum values. ϕ_0 is the basis function for the average mode.

Tables XIV and XV show the percentage of energy captured by each mode and the cumulative energy captured by the sum over the modes for each field variable.

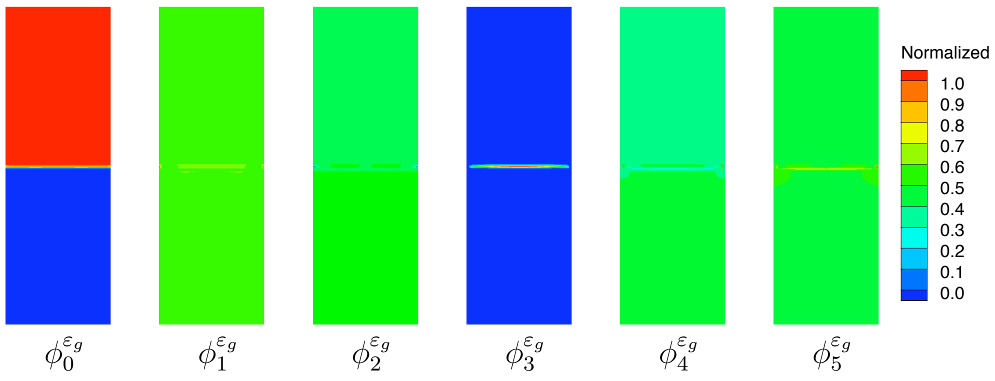


Figure 35: Case II: first six basis functions of void fraction, ε_g.

Table XIII: Parameters of case II.

Parameter	Description	Units	
x_{length}	Length of the domain in x-direction	cm	25.4
y_{length}	Length of the domain in y-direction	cm	76.5
i_{max}	Number of cells in x-direction	-	108
j_{max}	Number of cells in y-direction	-	124
v_1	Gas jet velocity	cm/s	12.6
v_2	Gas distributor velocity	cm/s	1.0
p_{g_s}	Static pressure at outlet	g/(cm/s^2)	1.01×10^6
T_{g_0}	Gas temperature	K	297
μ_{g_0}	Gas viscosity	g/(cm/s)	1.8×10^{-4}
Θ_0	Initial granular energy of bed	cm^2/s^2	10
t_{start}	Start time	s	0.0
t_{stop}	Stop time	s	1.0
Δt	Initial time step	s	1.0×10^{-4}
ρ_s	Particle density	g/cm^3	1.0
D_p	Particle diameter	cm	0.05
h_{s0}	Initial height of packed bed	cm	38.25
ϵ_g^*	Initial void fraction of packed bed	-	0.4

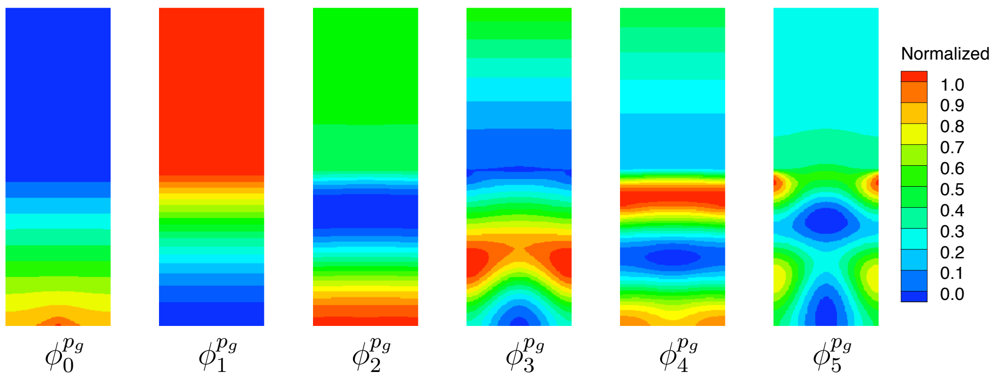


Figure 36: Case II: first six basis functions of gas pressure, p_g.

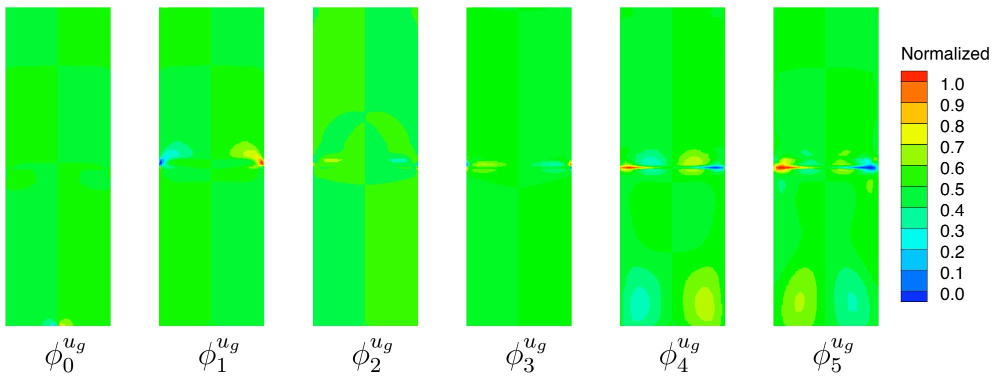


Figure 37: Case II: first six basis functions of gas velocity, u_g.

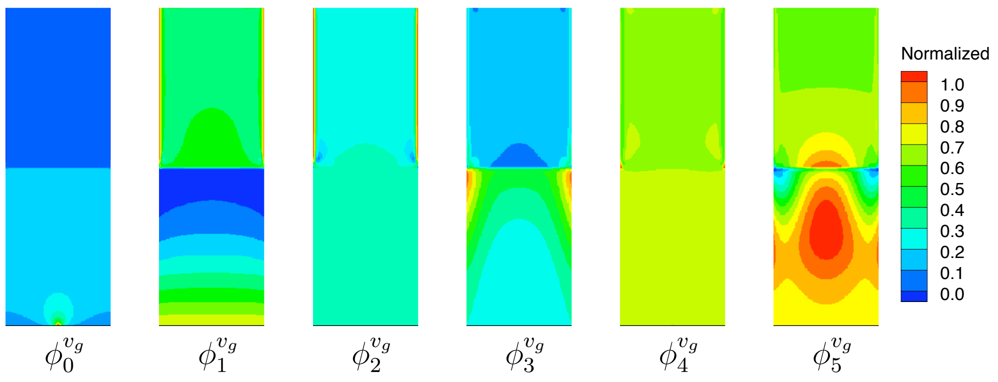


Figure 38: Case II: first six basis functions of gas velocity, v_g.

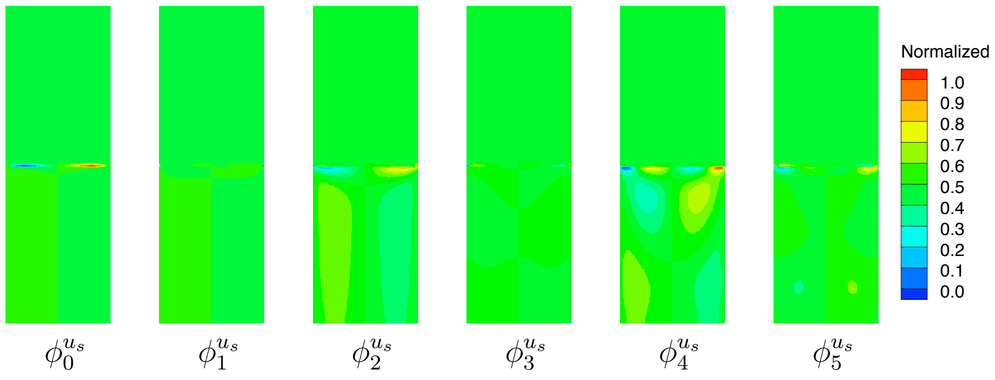


Figure 39: Case II: first six basis functions of solids velocity, u_s.

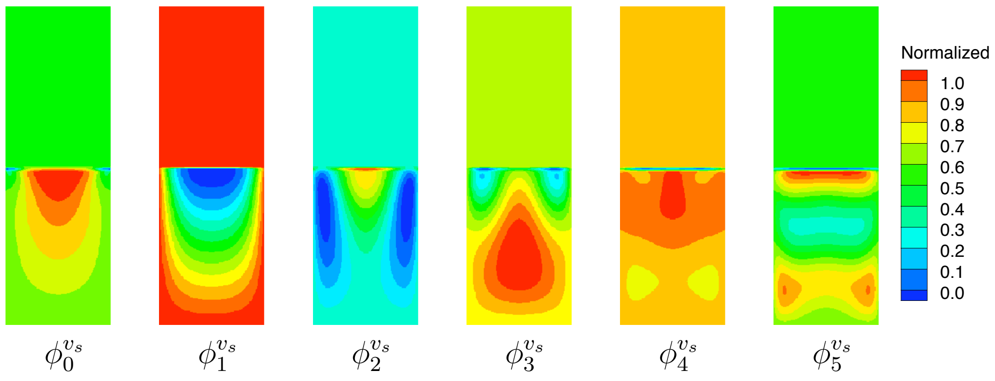


Figure 40: Case II: first six basis functions of solids velocity, v_s.

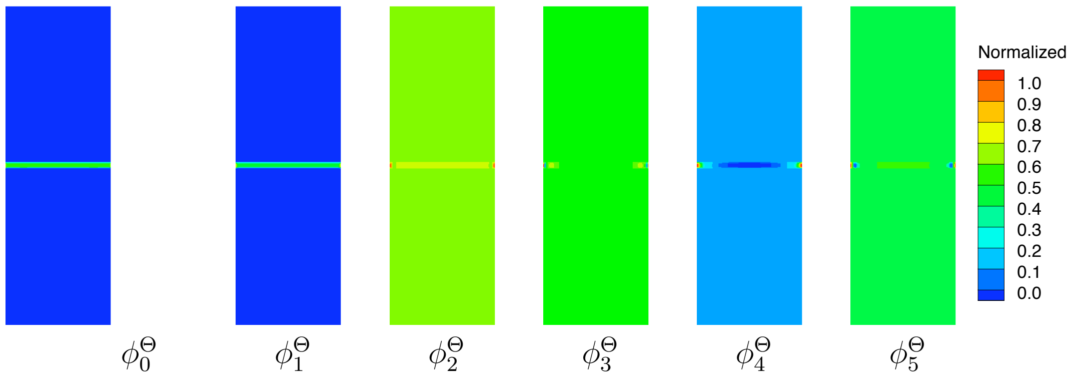


Figure 41: Case II: first six basis functions of granular energy, Θ.

Table XIV: Energy variation for the gas pressure and velocities of case II.

Number	p_g		u_g		v_g	
of Modes	Energy	Total	Energy	Total	Energy	Total
	[%]	[%]	[%]	[%]	[%]	[%]
1	99.936	99.936	89.091	89.091	84.990	84.990
2	0.061	99.996	18.085	95.944	13.609	98.600
3	0.003	99.999	4.518	99.202	1.025	99.625
4	0.000	99.999	2.119	99.811	0.267	99.891
5	0.000	99.999	1.029	99.931	0.078	99.970
6	0.000	99.999	0.589	99.961	0.013	99.984
7	0.000	99.999	0.424	99.975	0.008	99.992
8	0.000	99.999	0.223	99.986	0.005	99.997

Table XV: Energy variation for the void fraction, solids velocities, and granular energy of case II.

Number	ε_g		u_s		v_s		Θ	
of Modes	Energy	Total	Energy	Total	Energy	Total	Energy	Total
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1	81.264	81.264	82.567	82.567	96.534	96.534	99.989	99.989
2	16.662	97.885	12.592	82.518	2.088	98.623	0.009	99.997
3	1.975	99.860	2.915	93.189	1.027	99.650	0.002	99.999
4	0.091	99.951	1.052	97.026	0.197	99.847	0.000	99.999
5	0.038	99.989	0.439	98.706	0.123	99.970	0.000	99.999
6	0.009	99.997	0.192	99.407	0.020	99.990	0.000	99.999
7	0.002	99.999	0.146	99.701	0.007	99.997	0.000	99.999
8	0.000	99.999	0.061	99.823	0.001	99.998	0.000	99.999

The number of modes used and the energy spectrum captured by the chosen number of modes for each of the seven field variables is shown in Table XVI.

As with case I, the number of modes chosen for case II, shown in Table XVII was based on the best compromise between accuracy and computational speed-up of the reduced-order model. For this particular case, the mode sets of the field variables for the solids quantities contained at least 99.9%. This was to ensure that the solids particles were accurately modeled. The mode set used for Θ was varied, but for every combination other than 3 modes the solids velocities varied widely from the "exact" solution given by the full-order model. This was due to the highly coupled nature

Table XVI: Cumulative energy captured by the chosen number of modes for case II.

Variable	Number of Modes	Symbol	Cumulative Energy [%]
p_g	2	N_{p_g}	99.996
u_g	2	N_{u_g}	95.944
v_g	6	N_{v_g}	99.984
u_s	9	N_{u_s}	99.982
v_s	9	N_{v_s}	99.999
Θ	3	N_{Θ}	99.999
ε_g	7	N_{ε_g}	99.999

of the granular energy to the solids velocities (see (4.26)) and was only discovered through numerical experimentation.

Using the modes in Table XVII, it is possible to demonstrate that the POD-based ROM is more computationally efficient than the full-order model. The computational time of each is summarized in Table XVII. The computation time required for MFIX was 2917 seconds and 273 seconds for ODEg. The POD-based ROM ODEg, without Richardson's acceleration techinques [2], was 10.65 times faster than the full-order model, MFIX.

The accuracy of the ROM is analyzed by comparing the time coefficients computed by the ROM to those obtained by directly projecting the database of the snapshots onto the basis functions (from (2.19)). The time coefficients obtained via direct projection gives the "exact" solution of the time coefficients. Comparisons of the two time coefficients for p_g and u_g between ODEg and MFIX are shown in Fig-

Table XVII: Summary of case II CPU times for MFIX and ODEg.

Code	N_{p_g}	N_{u_g}	N_{v_g}	N_{u_s}	N_{v_s}	N_Θ	N_{ε_g}	t_{CPU} [s]
MFIX	-	-	-	-	-	-	-	2917
ODEg	2	2	6	9	9	3	7	273

ures 42 and 43. The first four time coefficients for ε_g, v_g, u_s, and v_s are compared in Figures 44 - 47 for ODEg and MFIX. These time coefficients tend to match well with the "exact" time coefficients. The solids velocities match the "exact" time coefficients best, which is critical for successfully modeling the granular energy as it is highly coupled to these quantities.

The three time coefficients of Θ are compared for ODEg and MFIX in Figure 48. The first time coefficient from ODEg follows closely its counterpart from MFIX. However, the other two time coefficients do not follow closely to the "exact" solution. These two time coefficients corresponding basis functions carry relatively little energy.

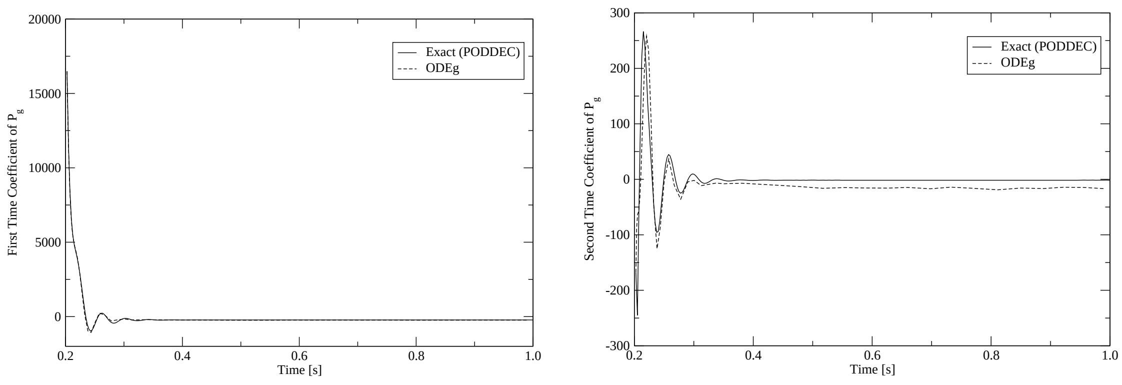


Figure 42: Case II: The two time coefficients of gas pressure, p_g, using ODEg and direct projection.

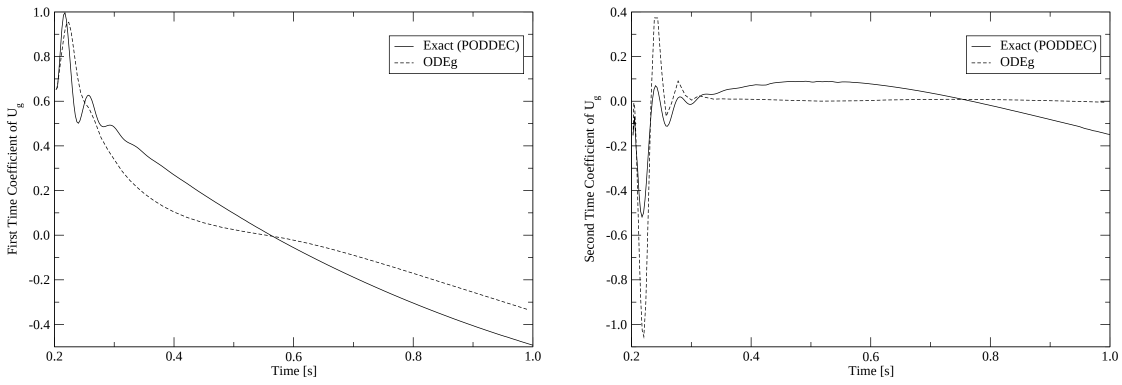


Figure 43: Case II: The two time coefficients of gas velocity, u_g, using ODEg and direct projection.

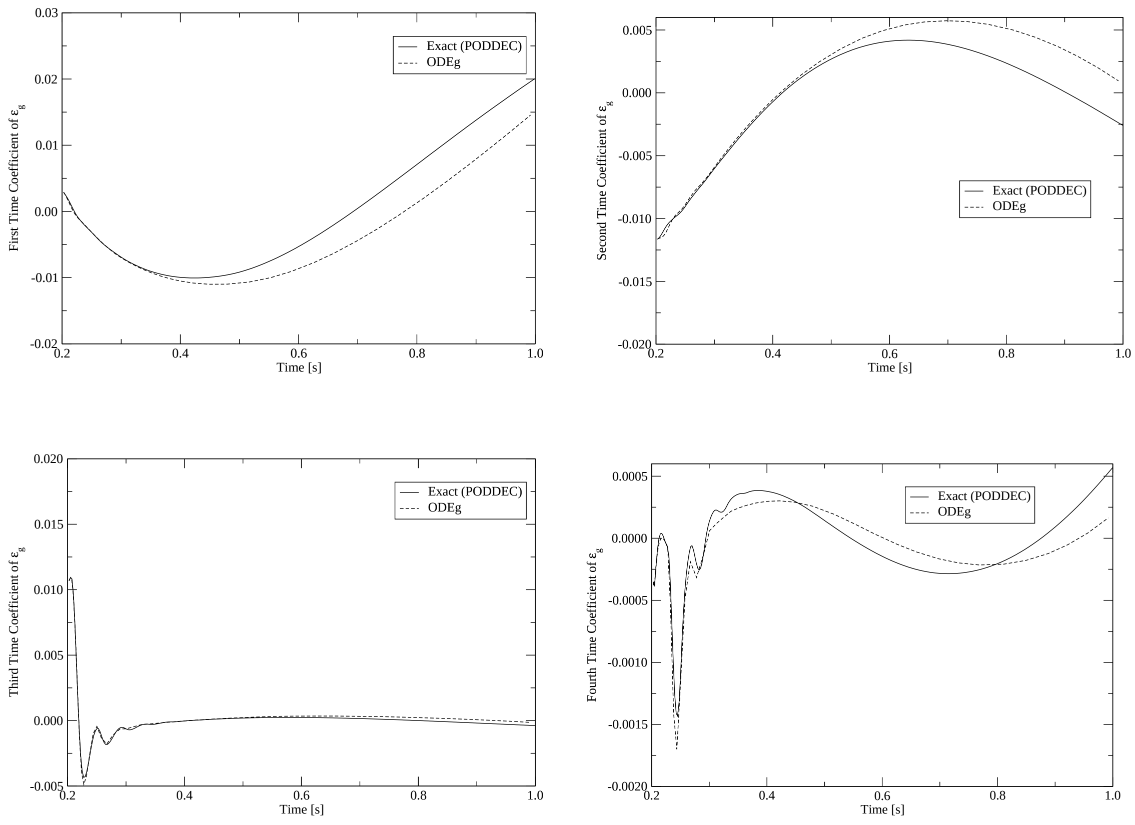


Figure 44: Case II: The first four time coefficients of void fraction, ε_g, using ODEg and direct projection.

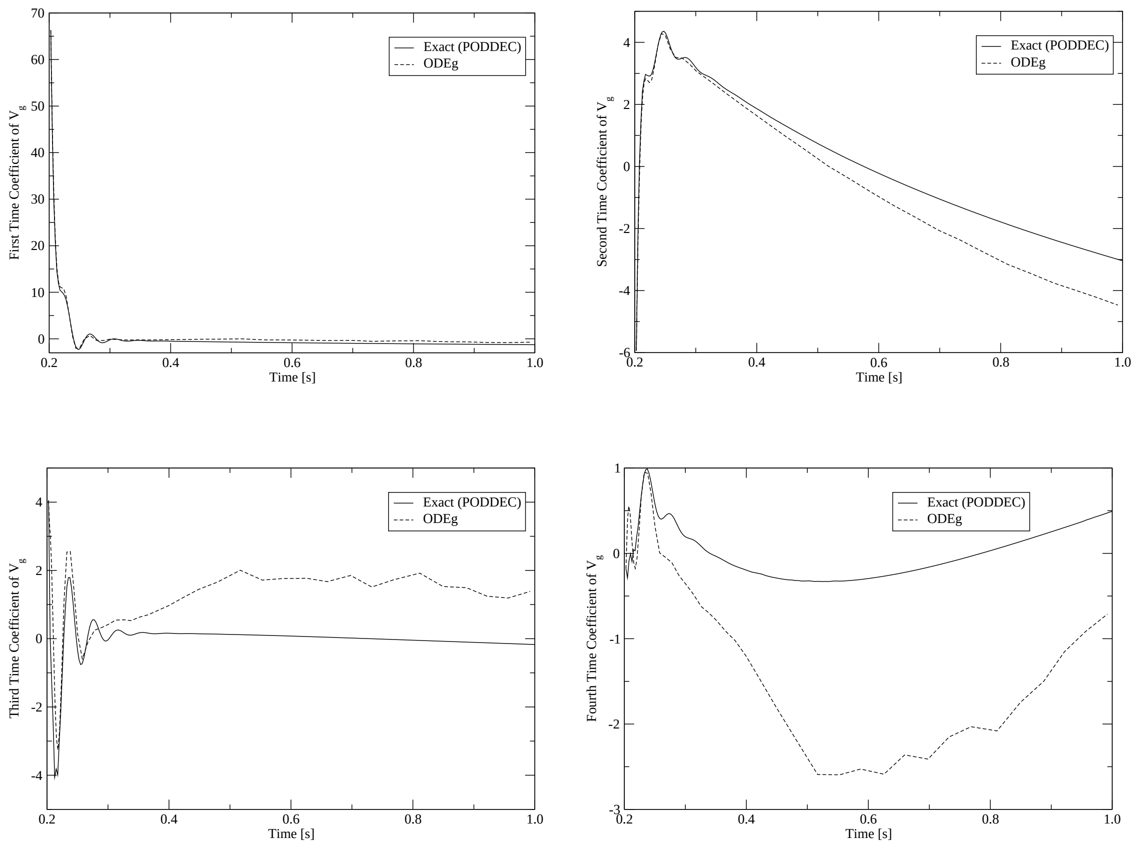


Figure 45: Case II: The first four time coefficients of gas velocity, v_g, using ODEg and direct projection.

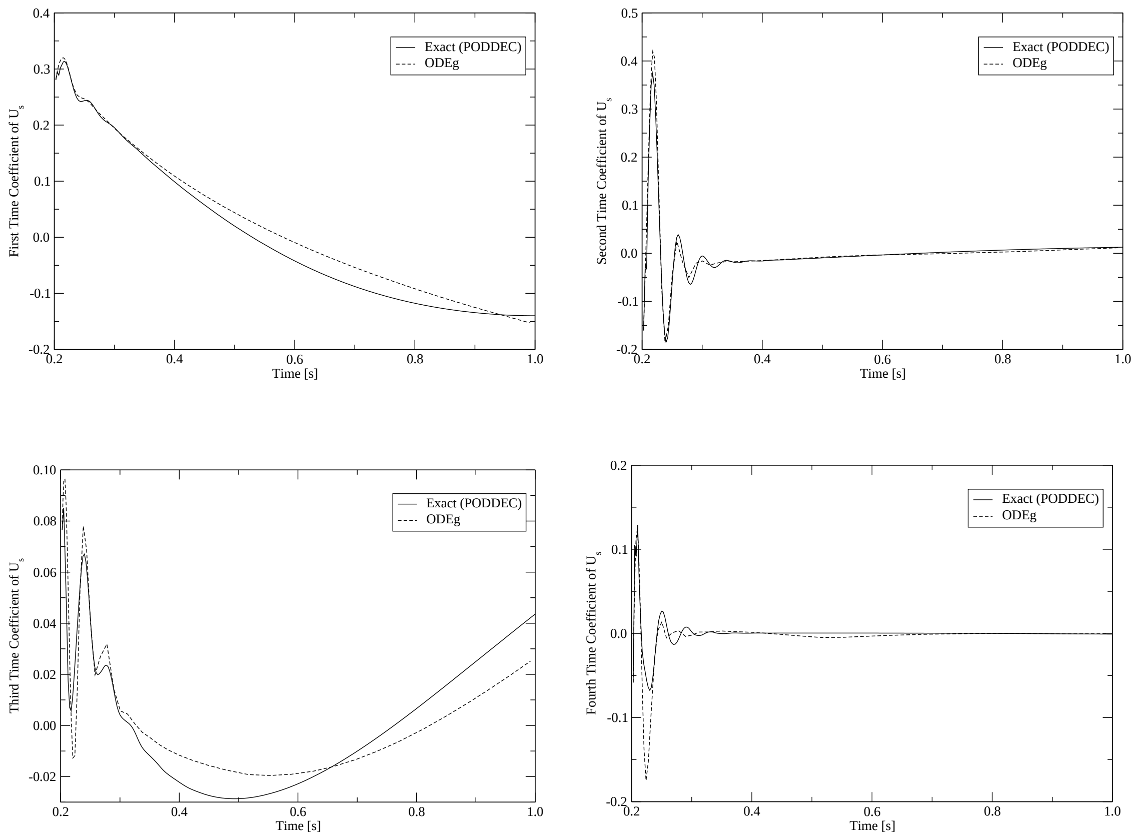


Figure 46: Case II: The first four time coefficients of solids velocity, u_s, using ODEg and direct projection.

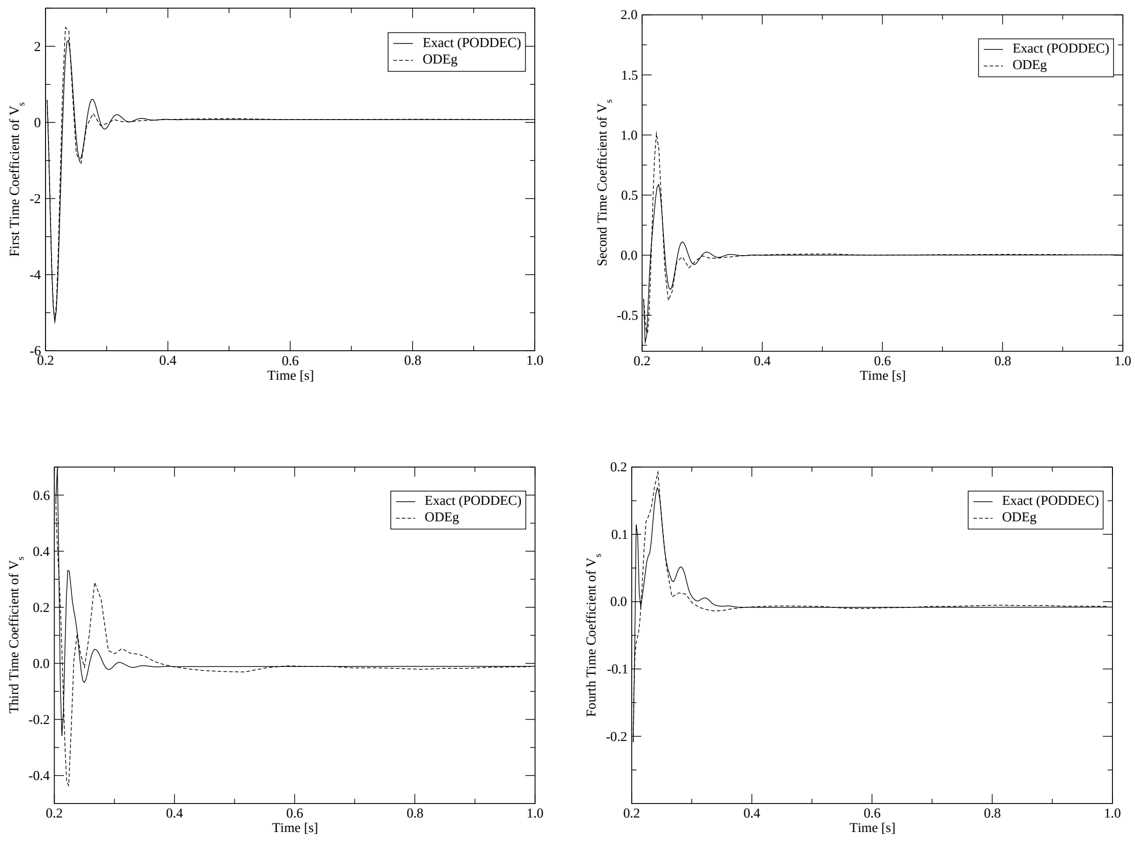


Figure 47: Case II: The first four time coefficients of solids velocity, v_s, using ODEg and direct projection.

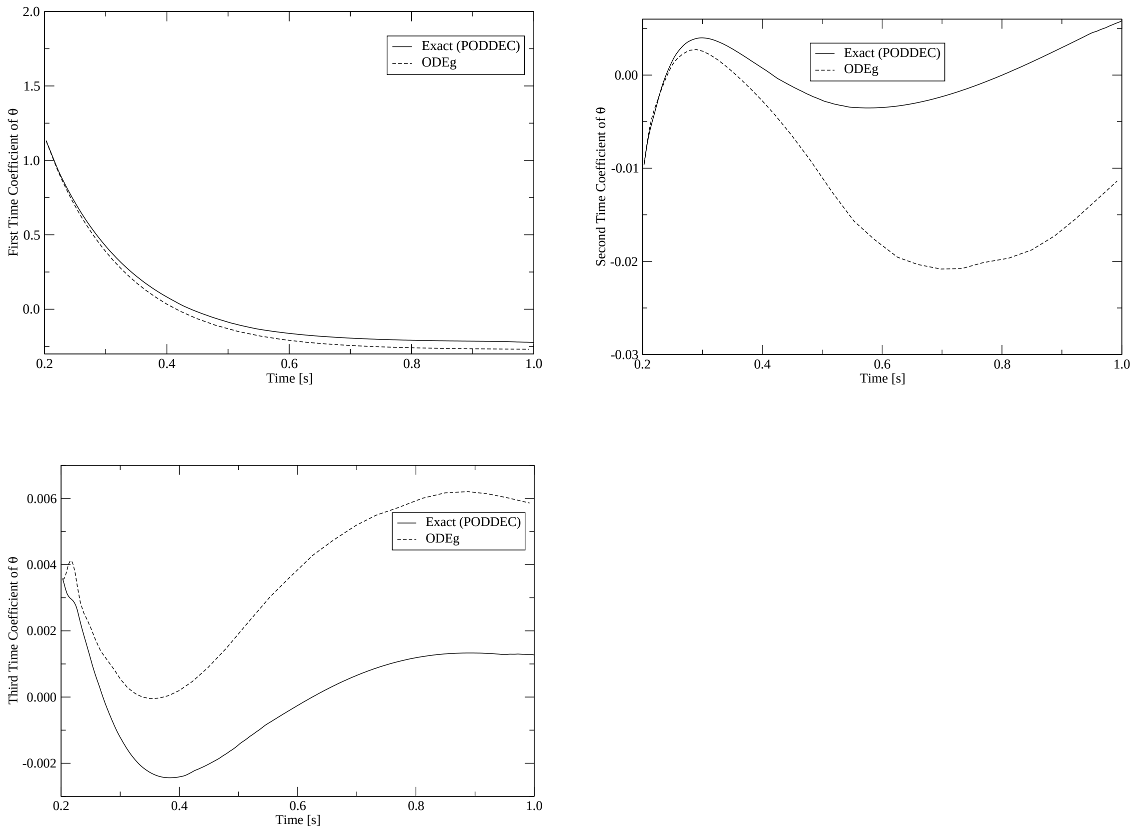


Figure 48: Case II: The three time coefficients of granular energy, Θ, using ODEg and direct projection.

Contour plots at $t = 1.0$ s were generated for each of the field variables at the end of the integration period. The numerical results at the end of the integration period have the largest error, which were accumulated during time integration. A comparison of the results between MFIX and ODEg for the pressure, void fraction, and gas and solids velocities is shown in Figures 49 - 51. The flow fields of MFIX and ODEg match well. Most importantly, the solids velocities computed at the surface of the bed, where the majority of the granular energy is found, from ODEg are similar to those computed by MFIX at surface of the bed. This should ensure that the granular energy is captured well.

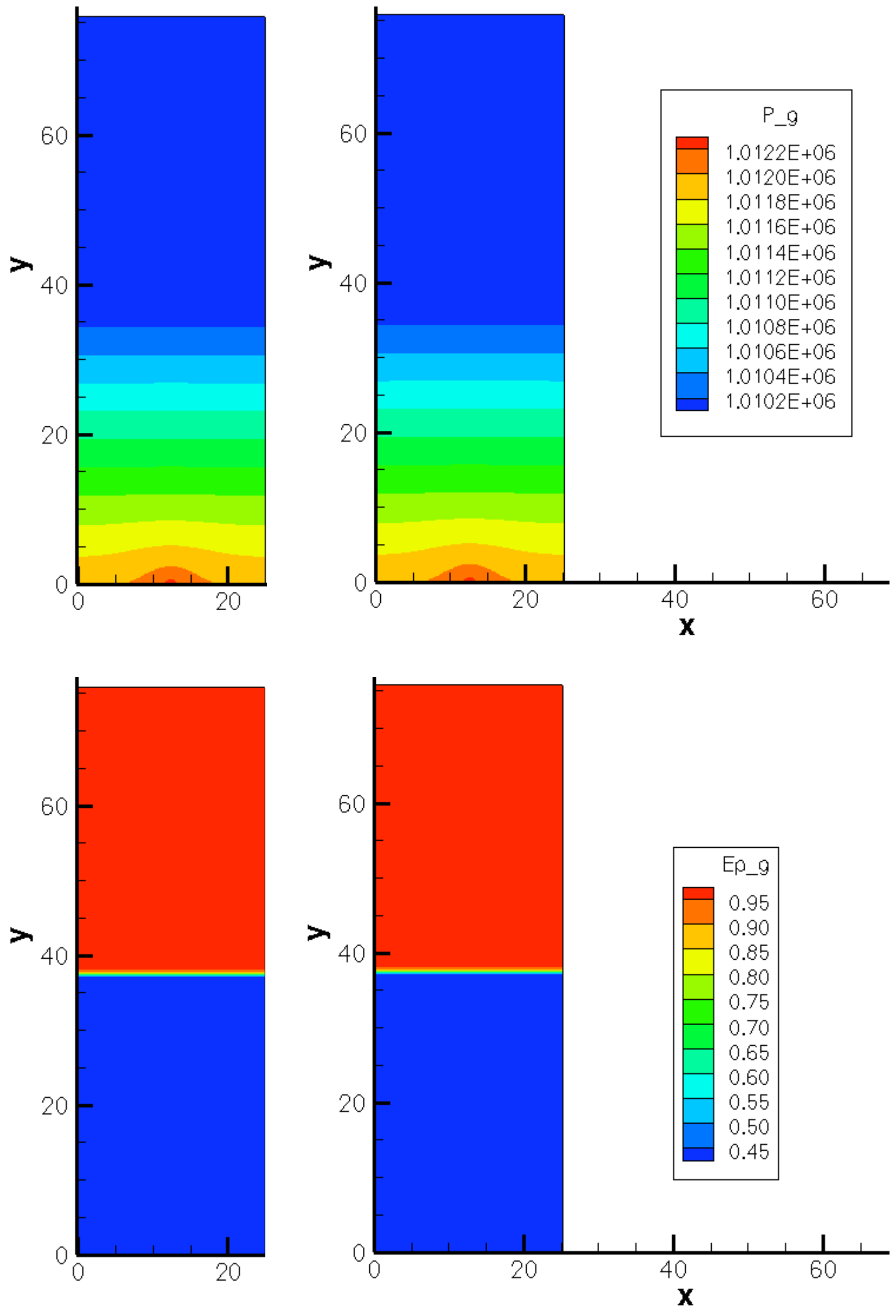


Figure 49: Case II contour plots at $t = 1.0$ s using MFIX (right) and ODEg (left): gas pressure, p_g, (top) and void fraction, ε_g (bottom).

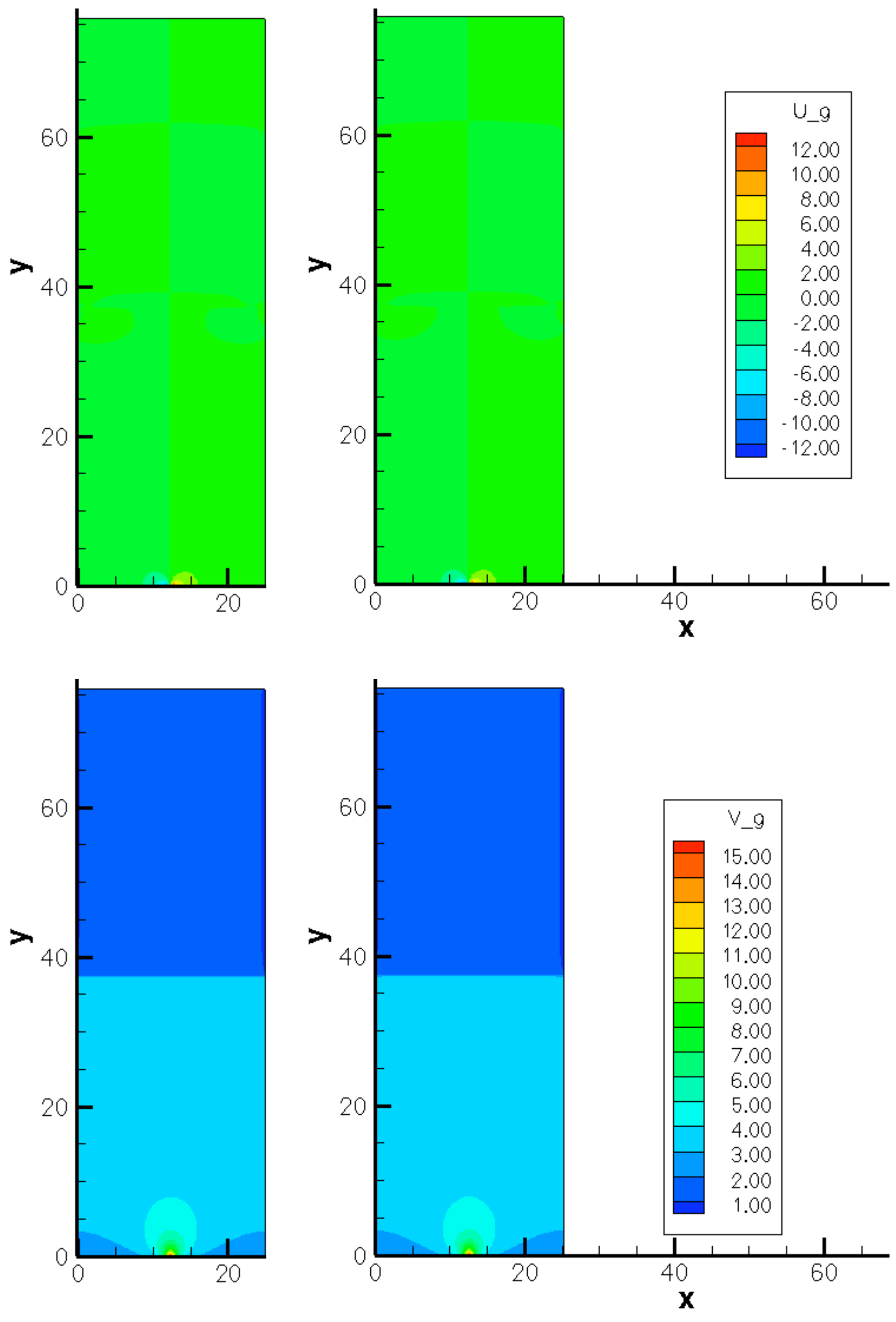
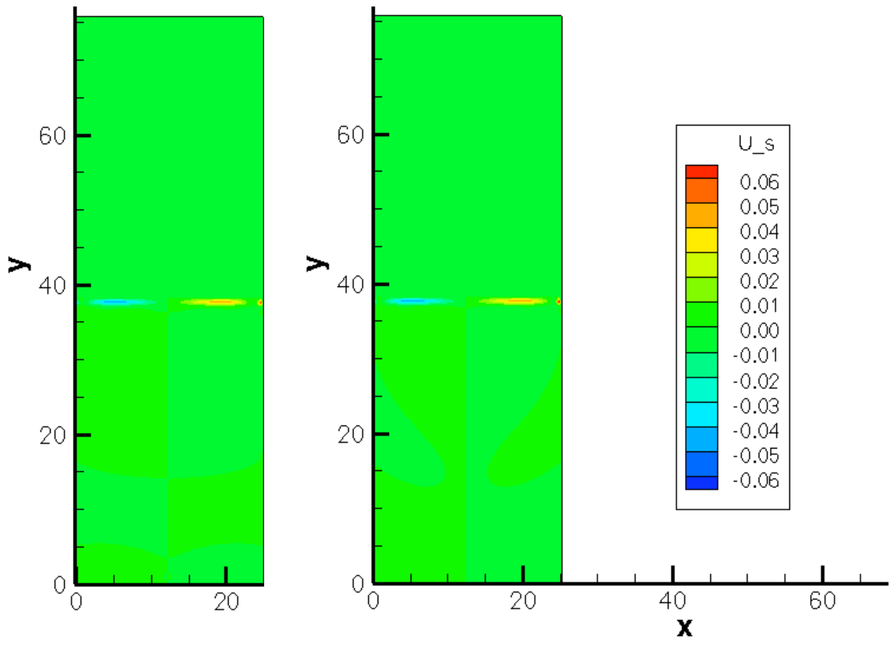


Figure 50: Case II contour plots at $t = 1.0$ s using MFIX (right) and ODEg (left): gas velocity, u_g, (top) and gas velocity, v_g (bottom).



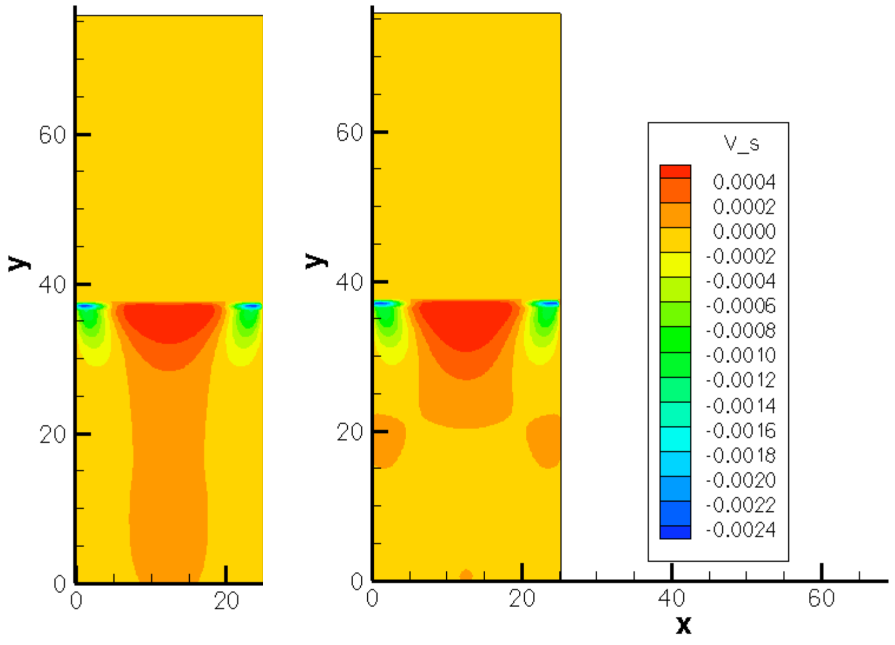


Figure 51: Case II contour plots at t = 1.0 s using MFIX (right) and ODEg (left): solids velocity, u_s, (top) and solids velocity, v_s (bottom).

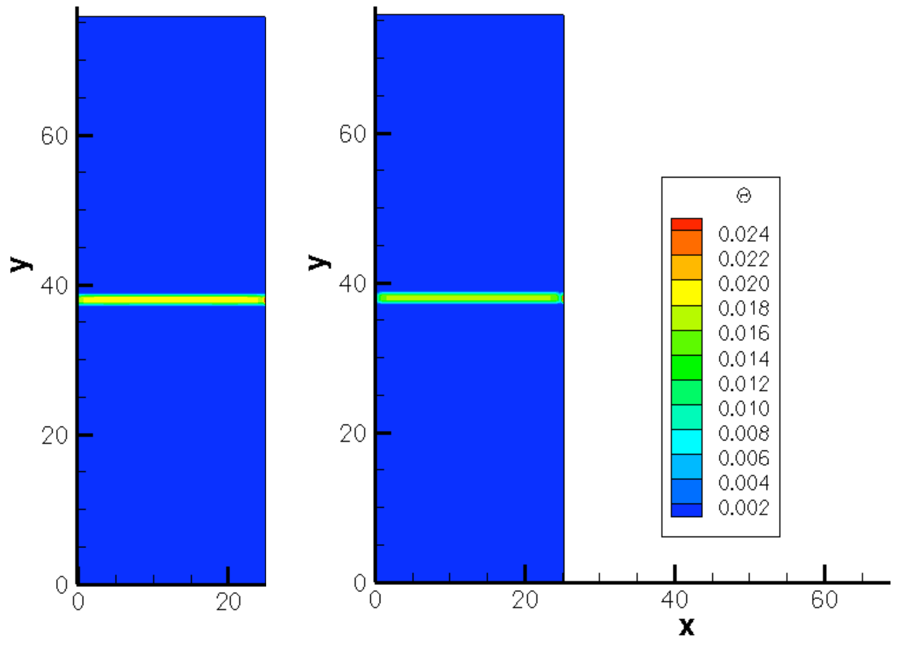


Figure 52: Case II contour plots at $t = 1.0$ s using MFIX (right) and ODEg (left): granular energy, Θ.

The granular energy contours of MFIX and ODEg are compared in Figure 52. Zoomed plots of the granular energy are given in Figure 53. As evidenced by both figures, ODEg captures the granular energy accurately when compared to the results from MFIX. Using the mode set in Table XVI, ODEg achieves two orders of magnitude computational speed-up factor over MFIX and is capable of calculating qualitatively accurate results.

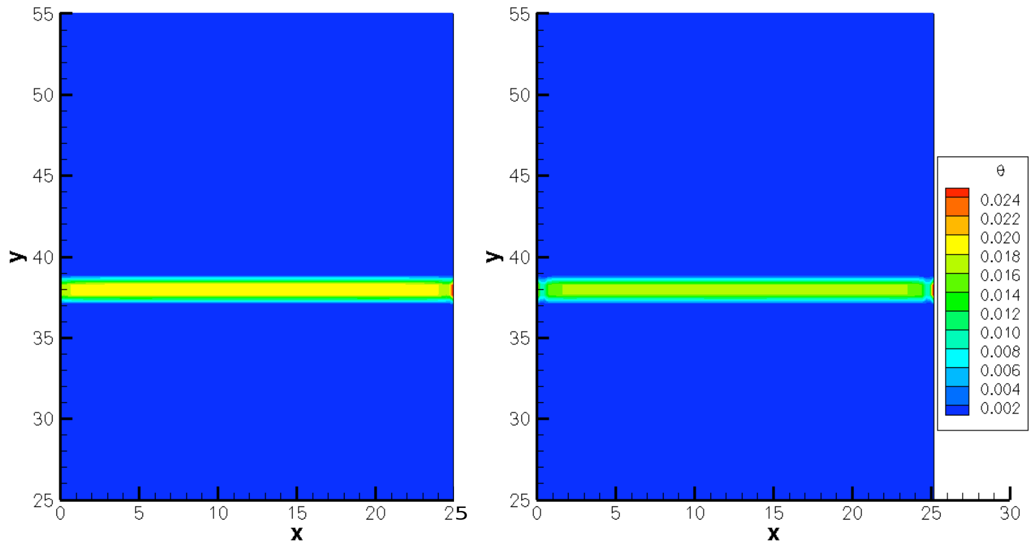


Figure 53: Case II zoomed contour plots at t = 1.0 s using MFIX (left) and ODEg (right) for the granular energy, Θ.

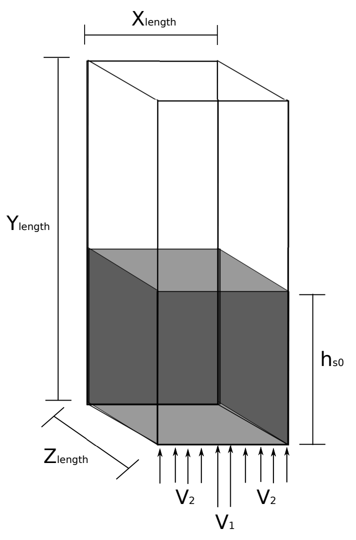
3. Case III: Three-Dimensional Isothermal Flow

Case III is a two-phase model for three dimensional isothermal flow. The model consists of one solids and one gas phase. The geometry and boundary conditions of case III are shown in Figure 54. The computational domain is a uniform rectangular grid. As with the previous two cases, air enters in through the lower boundary of the bed. The three dimensional case is a projection in the $z-$direction of the two dimensional cases, which changes the lower boundary inlet conditions. Instead of a central jet, a jet strip located in the center of the bed along the z axis injects air into the bed with a velocity $v_1 = 12.6$ cm/s. On either side of this jet, air is injected at a velocity $v_2 = 1.0$ cm/s through a distributor. The air is injected at a temperature of 297 K. The jet strip is approximately 1-cm wide. The grid spacing is equal in the x- and z-direction. The particle diameter is constant throughout the bed, and each phase (gas and solids) is modeled as a non-reacting single species. The model parameters for this case are listed in Table XVIII.

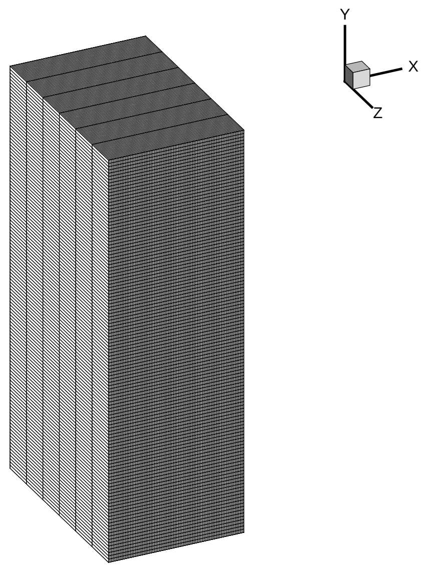
POD is applied to the database of snapshots computed by MFIX for the eight field variables modeled using the parameters of Table XVIII. The first six POD basis functions of ε_g, p_g, u_g, u_s, v_g, v_s, w_g, and w_s are shown in Figures 55 - 62, respectively. The modes are normalized by maximum values. ϕ_0 is the basis function for the average mode.

Table XVIII: Parameters of case III.

Parameter	Description	Units	
x_{length}	Length of the domain in x-direction	cm	25.4
y_{length}	Length of the domain in y-direction	cm	76.5
z_{length}	Length of the domain in z-direction	cm	1.646
i_{max}	Number of cells in x-direction	-	108
j_{max}	Number of cells in y-direction	-	124
k_{max}	Number of cells in z-direction	-	7
v_1	Gas jet velocity	cm/s	12.6
v_2	Gas distributor velocity	cm/s	1.0
p_{g_s}	Static pressure at outlet	g/(cm/s^2)	1.01×10^6
T_{g_0}	Gas temperature	K	297
μ_{g_0}	Gas viscosity	g/(cm/s)	1.8×10^{-4}
t_{start}	Start time	s	0.0
t_{stop}	Stop time	s	1.0
Δt	Initial time step	s	1.0×10^{-4}
ρ_s	Particle density	g/cm^3	1.0
D_p	Particle diameter	cm	0.05
h_{s0}	Initial height of packed bed	cm	38.25
ϵ_g^*	Initial void fraction of packed bed	-	0.4



(a)



(b)

Figure 54: Case III: (a) geometry and boundary conditions, and (b) computational grid.

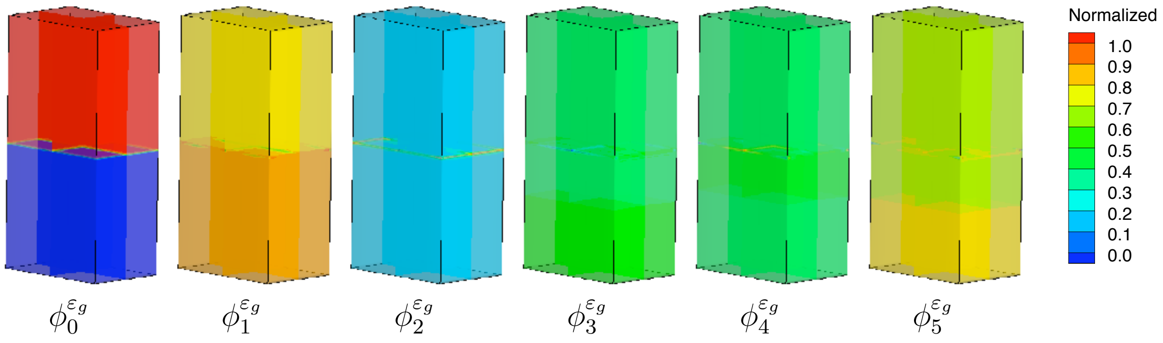


Figure 55: Case III: first six basis functions of void fraction, ε_g.

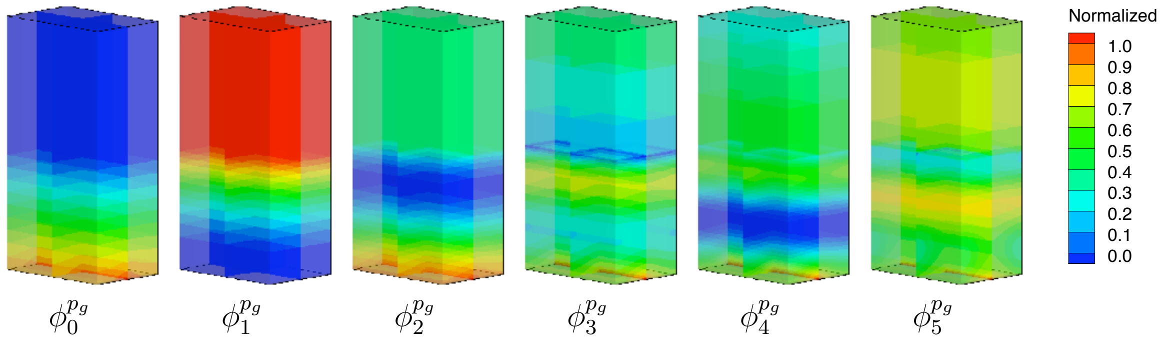


Figure 56: Case III: first six basis functions of gas pressure, p_g.

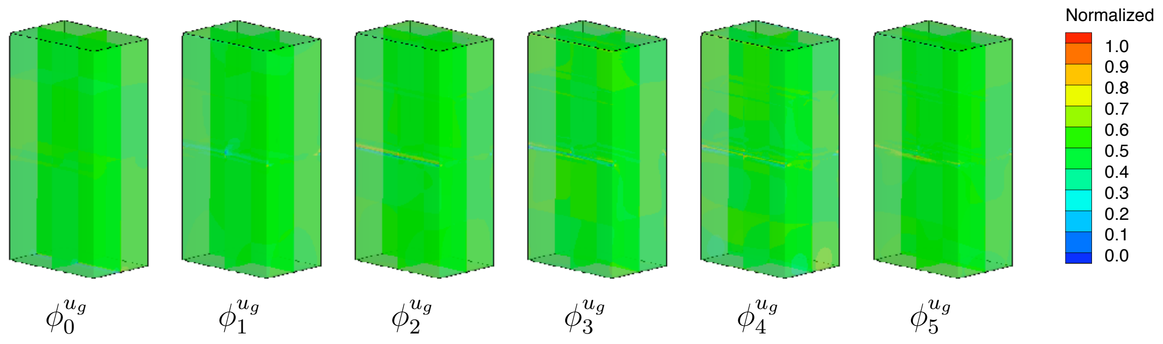


Figure 57: Case III: first six basis functions of gas velocity, u_g.

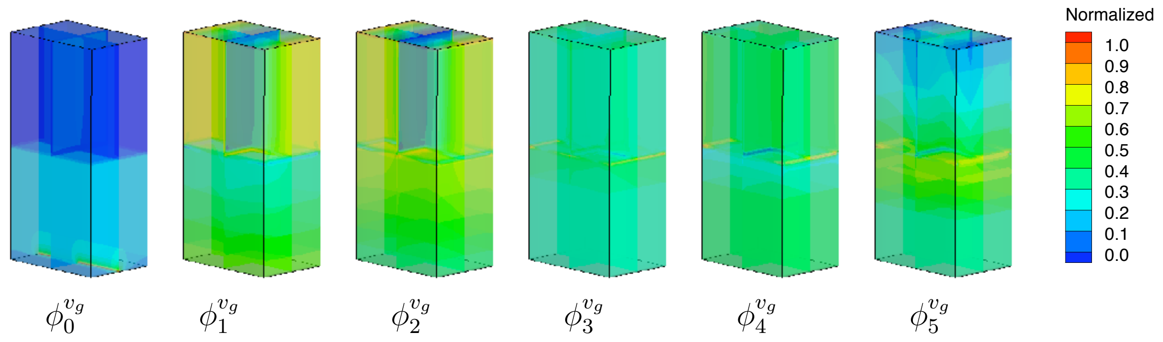


Figure 58: Case III: first six basis functions of gas velocity, v_g.

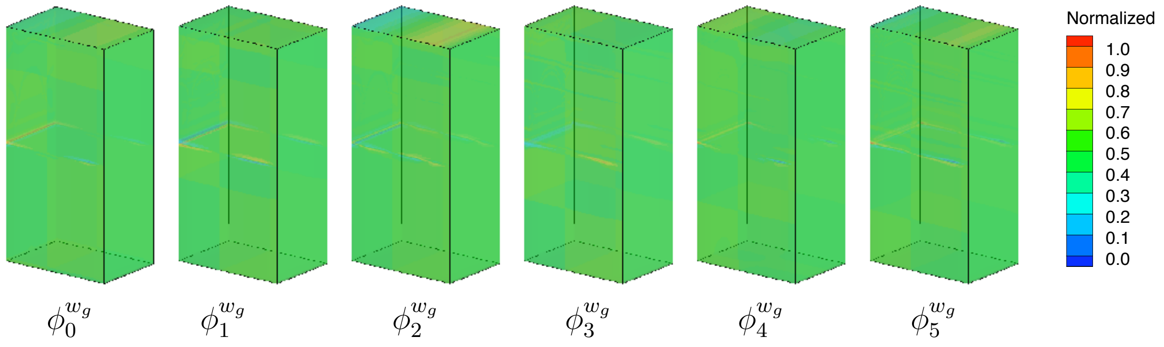


Figure 59: Case III: first six basis functions of gas velocity, w_g.

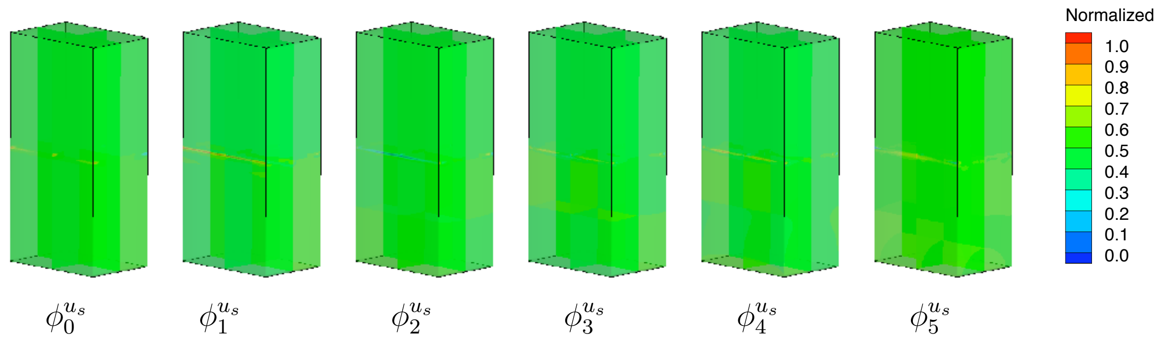


Figure 60: Case III: first six basis functions of solids velocity, u_s.

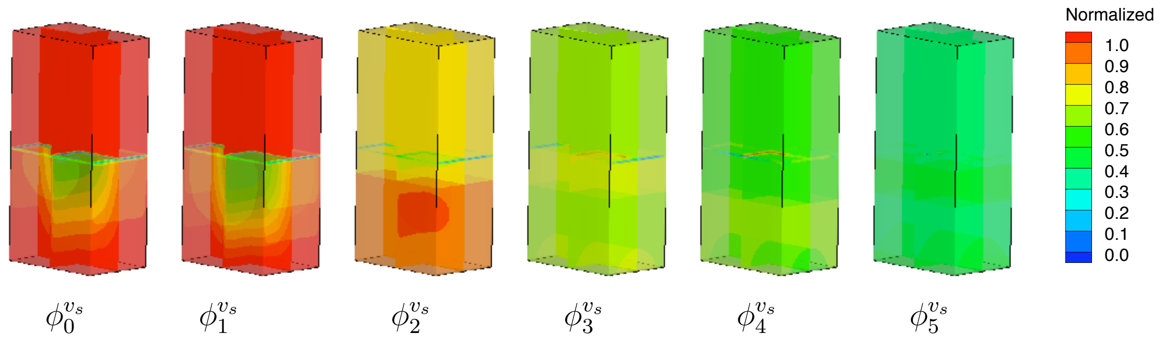


Figure 61: Case III: first six basis functions of solids velocity, v_s.

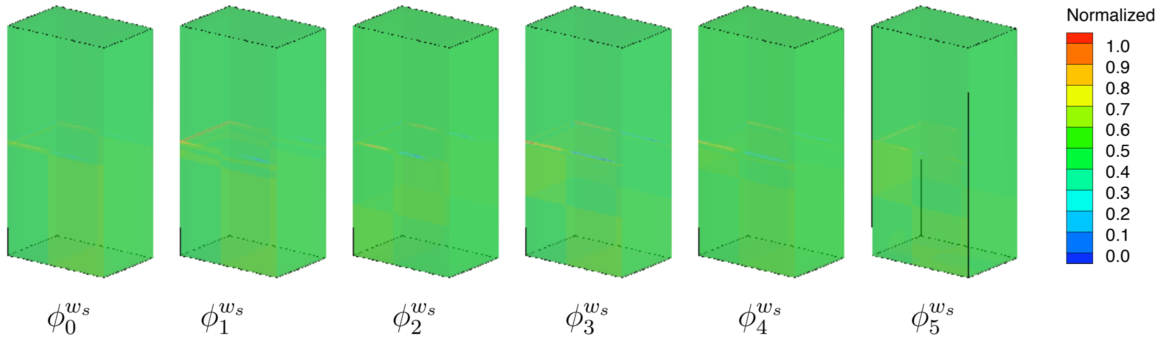


Figure 62: Case III: first six basis functions of solids velocity, w_s.

Tables XIX and XX show the percentage of energy captured by each mode and the cumulative energy captured by the sum over the modes for each field variable.

Table XIX: Energy variation for the gas pressure and velocities of case III.

Number	p_g		u_g		v_g		w_g	
of Modes	Energy	Total	Energy	Total	Energy	Total	Energy	Total
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1	99.948	99.948	91.084	91.084	55.526	55.526	61.935	61.935
2	0.048	99.995	8.510	99.594	42.462	97.988	37.765	99.700
3	0.004	99.998	0.303	99.896	1.857	99.845	0.253	99.953
4	0.001	99.999	0.071	99.968	0.135	99.980	0.026	99.980
5	0.000	99.999	0.016	99.985	0.016	99.996	0.014	99.994
6	0.000	99.999	0.003	99.988	0.002	99.999	0.002	99.996
7	0.000	99.999	0.002	99.990	0.000	99.999	0.001	99.997
8	0.000	99.999	0.001	99.991	0.000	99.999	0.000	99.997

Table XX: Energy variation for the void fraction and solids velocities of case III.

Number	ε_g		u_s		v_s		w_s	
of Modes	Energy	Total	Energy	Total	Energy	Total	Energy	Total
	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1	75.266	75.266	95.705	95.705	97.469	97.469	99.539	99.539
2	24.659	99.925	4.022	99.727	2.472	99.942	0.453	99.992
3	0.067	99.993	0.205	99.931	0.031	99.972	0.007	99.999
4	0.004	99.996	0.056	99.988	0.012	99.984	0.001	99.999
5	0.003	99.999	0.009	99.997	0.002	99.986	0.000	99.999
6	0.000	99.999	0.002	99.998	0.002	99.987	0.000	99.999
7	0.000	99.999	0.000	99.999	0.001	99.989	0.000	99.999
8	0.000	99.999	0.000	99.999	0.001	99.990	0.000	99.999

The number of modes used and the energy spectrum captured by the chosen number of modes for each of the eight field variables is shown in Table XXI .

The modes were chosen, shown in Table XXII based on the best compromise between accuracy and computational speed-up of the reduced-order model. For this case, the mode set chosen was the same as case II for p_g, ε_g, u_g, v_g, u_s, and v_s. This was due to the success of ODEg (see section 2) in capturing the solids velocities as well as the numerical experimentation with the two-dimensional isothermal case. For w_g and w_s, the number of modes for these variables was equal to the number of modes for u_g and u_s, respectively. It was assumed that because the grid spacing in the x- and z-directions are equal, the velocities in the x- and z-directions would behave similarly.

Table XXI: Cumulative energy captured by the chosen number of modes for case III.

Variable	Number of Modes	Symbol	Cumulative Energy [%]
p_g	2	N_{p_g}	99.995
u_g	2	N_{u_g}	99.594
v_g	6	N_{v_g}	99.999
w_g	2	N_{w_g}	99.700
u_s	9	N_{u_s}	99.999
v_s	9	N_{v_s}	99.999
w_s	9	N_{w_s}	99.999
ε_g	7	N_{ε_g}	99.999

Using the modes in Table XXII, it is possible to demonstrate that the POD-based ROM is more computationally efficient than MFIX. The computational time of each is summarized in Table XXII. The computation time required for MFIX was 125908 seconds and 10349 seconds for ODEV. The POD-based ROM ODEV, without Richardson's acceleration techniques [2], was 12.16 times faster than the full-order model, MFIX.

The accuracy of the ROM is analyzed by comparing the time coefficients computed by the ROM to those obtained by directly projecting the database of the snapshots onto the basis functions (from (2.19)). The time coefficients obtained via direct projection gives the "exact" solution of the time coefficients. Comparisons of the two time coefficients for p_g, u_g, and w_g between ODEV and MFIX are shown Figures 63 - 65. The first four time coefficients of ε_g, u_s, v_g, v_s, and w_s are compared

Table XXII: Summary of case III CPU times for MFIX and ODEV.

Code	N_{p_g}	N_{u_g}	N_{v_g}	N_{w_g}	N_{u_s}	N_{v_s}	N_{w_s}	N_{ε_g}	t_{CPU} [s]
MFIX	-	-	-	-	-	-	-	-	125908
ODEV	2	2	6	2	9	9	9	7	10349

in Figures 66 - 70. The time coefficients for p_g and the first two time coefficients of v_g and ε_g match the "exact" time coefficients. The errors for all of the other time coefficients becomes large almost immediately after the ROM simulation has started. The large errors in the time coefficients does not generally equate to accurate modeling of the flow field, but a final judgment cannot be made until the variables are reconstructed with (B).

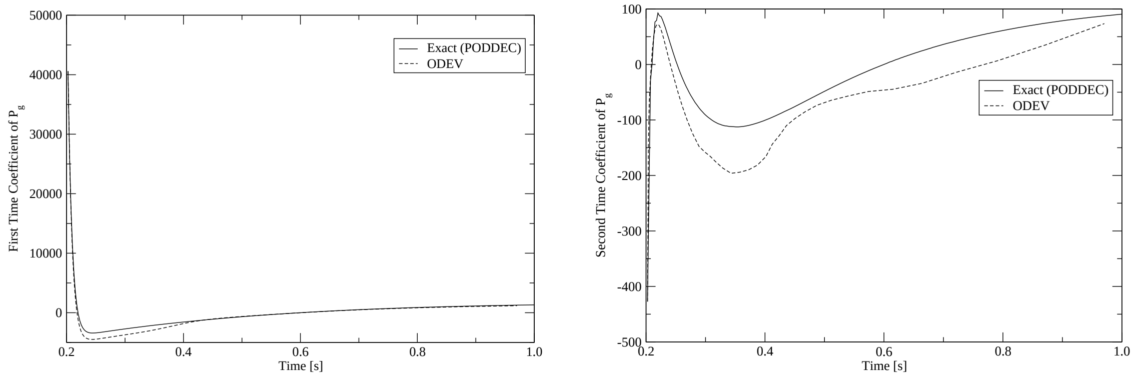


Figure 63: Case III: The two time coefficients of gas pressure, p_g, using ODEV and direct projection.

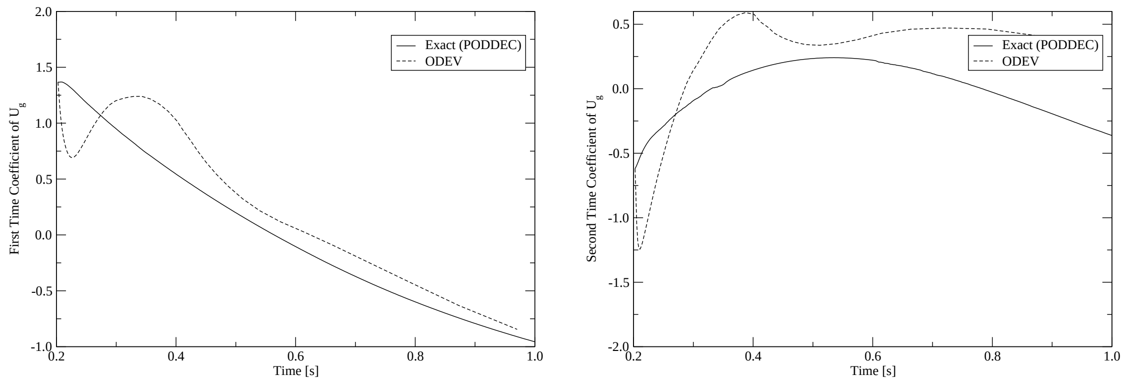


Figure 64: Case III: The two time coefficients of gas velocity, u_g, using ODEV and direct projection.

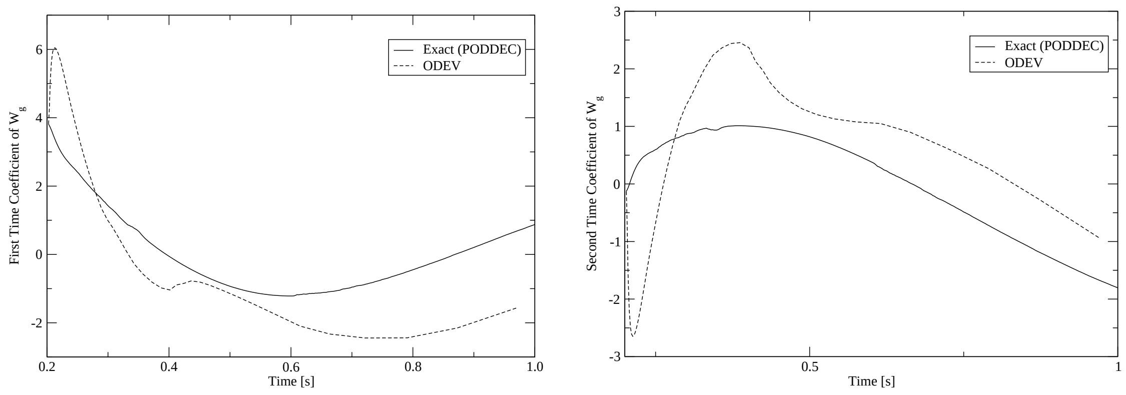


Figure 65: Case III: The two time coefficients of gas velocity, w_g, using ODEV and direct projection.

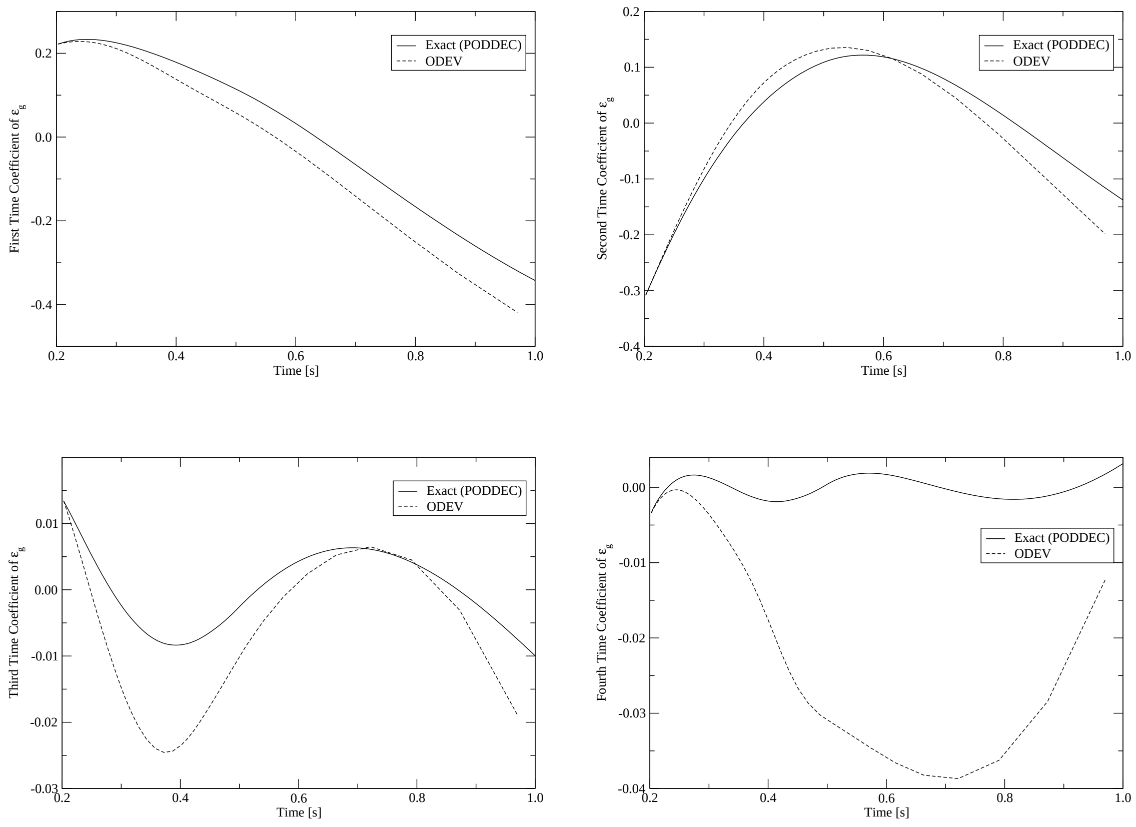


Figure 66: Case III: The first four time coefficients of gas velocity, ε_g, using ODEV and direct projection.

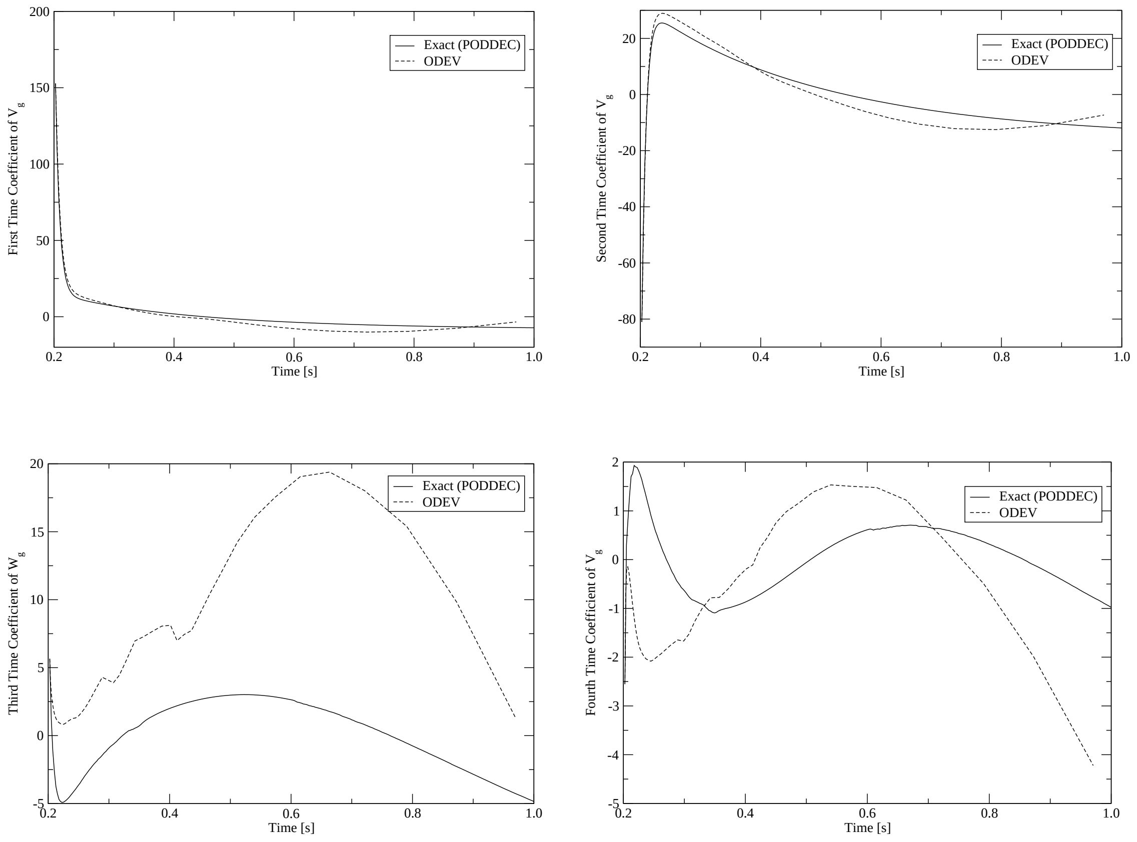


Figure 67: Case III: The first four time coefficients of gas velocity, v_g, using ODEV and direct projection.

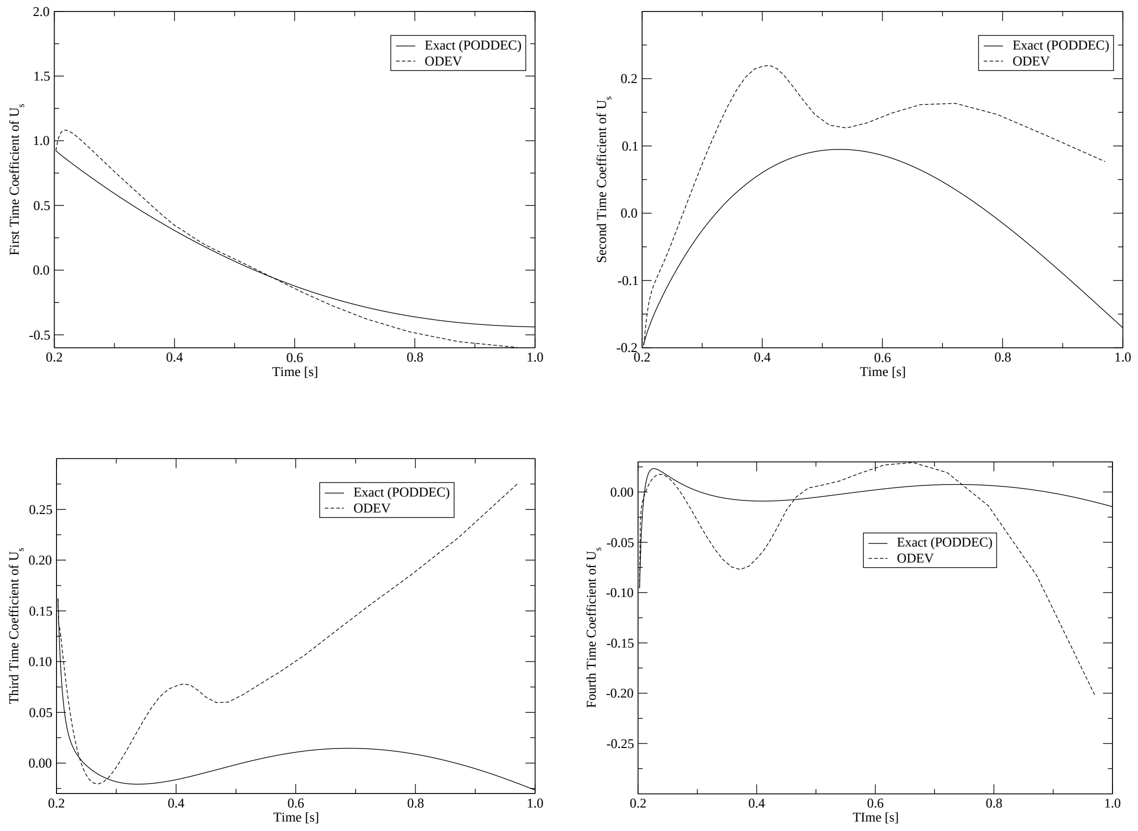


Figure 68: Case III: The first four time coefficients of solids velocity, u_s, using ODEV and direct projection.

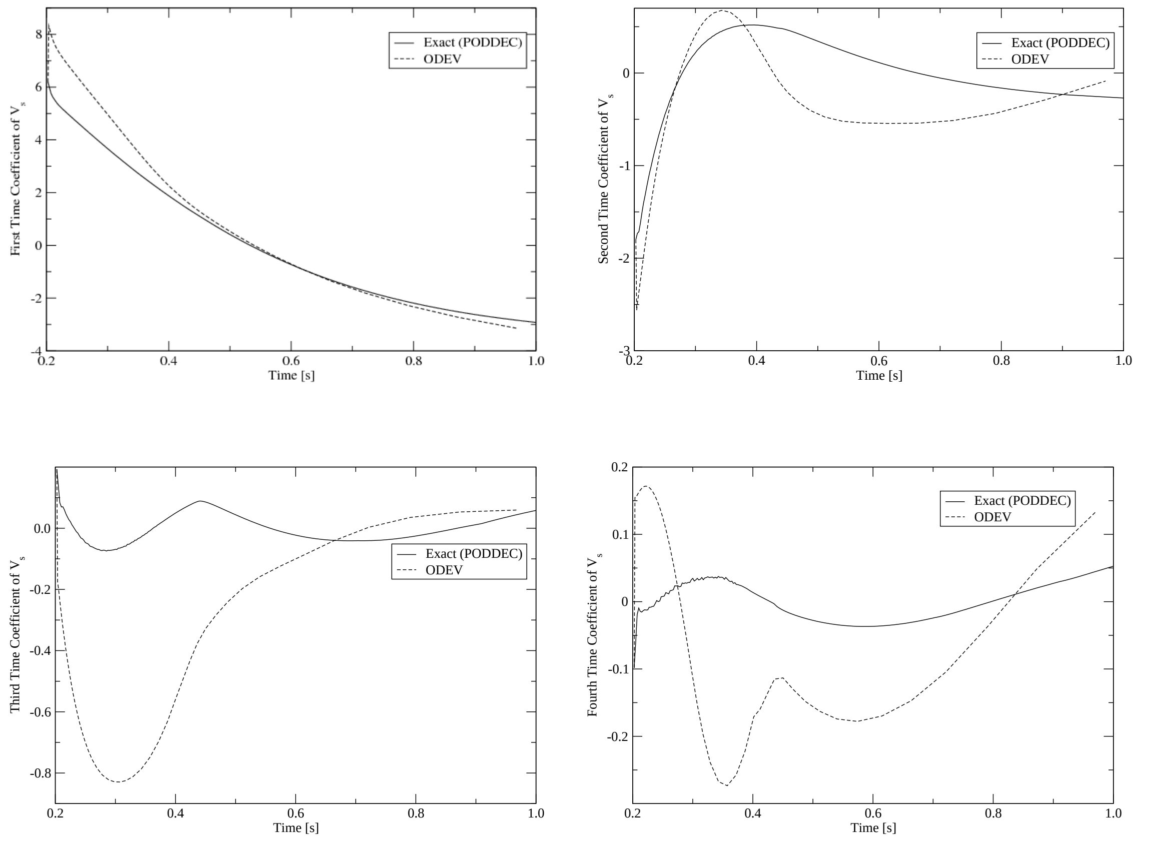


Figure 69: Case III: The first four time coefficients of solids velocity, v_s, using ODEV and direct projection.

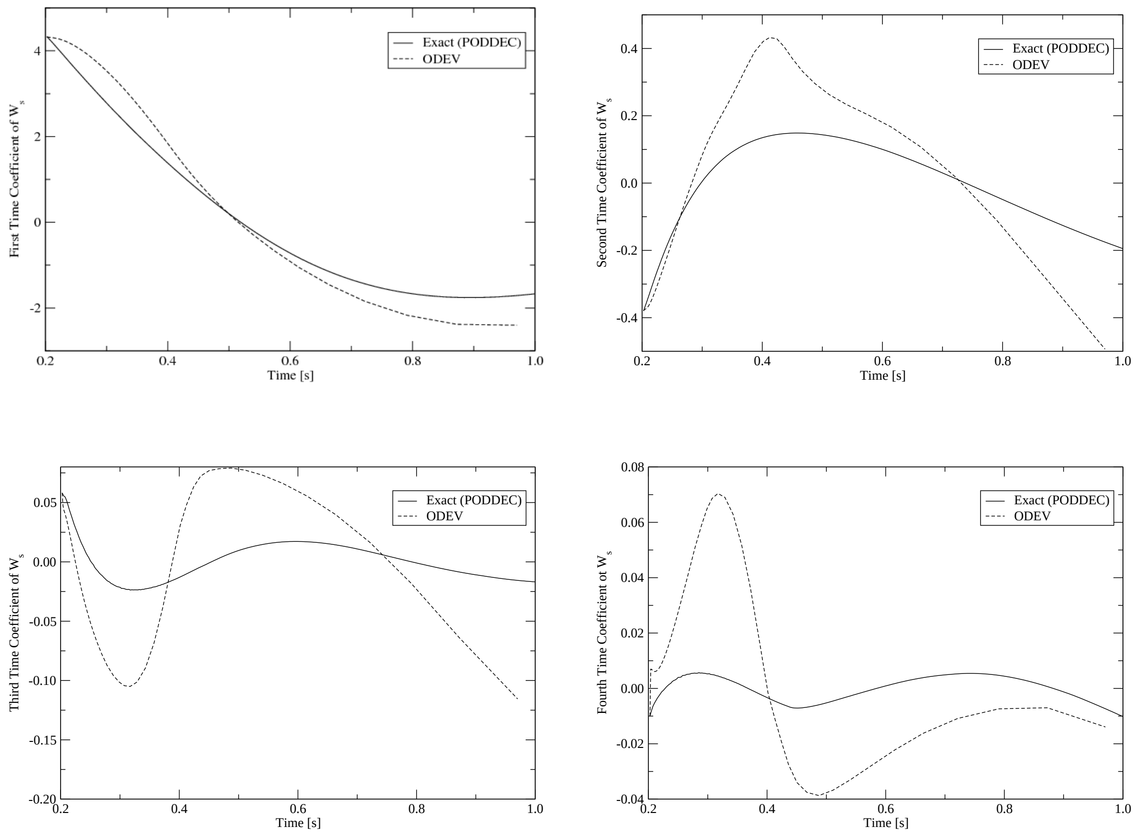


Figure 70: Case III: The first four time coefficients of solids velocity, w_s, using ODEV and direct projection.

Contour plots at $t = 1.0$ s were generated for each of the field variables at the end of the integration period. The numerical results at the end of the integration period have the largest error, which were accumulated during time integration. A comparison of the results between MFIX and ODEV for the pressure, void fraction, and gas and solids velocities is shown in Figures 71-74.

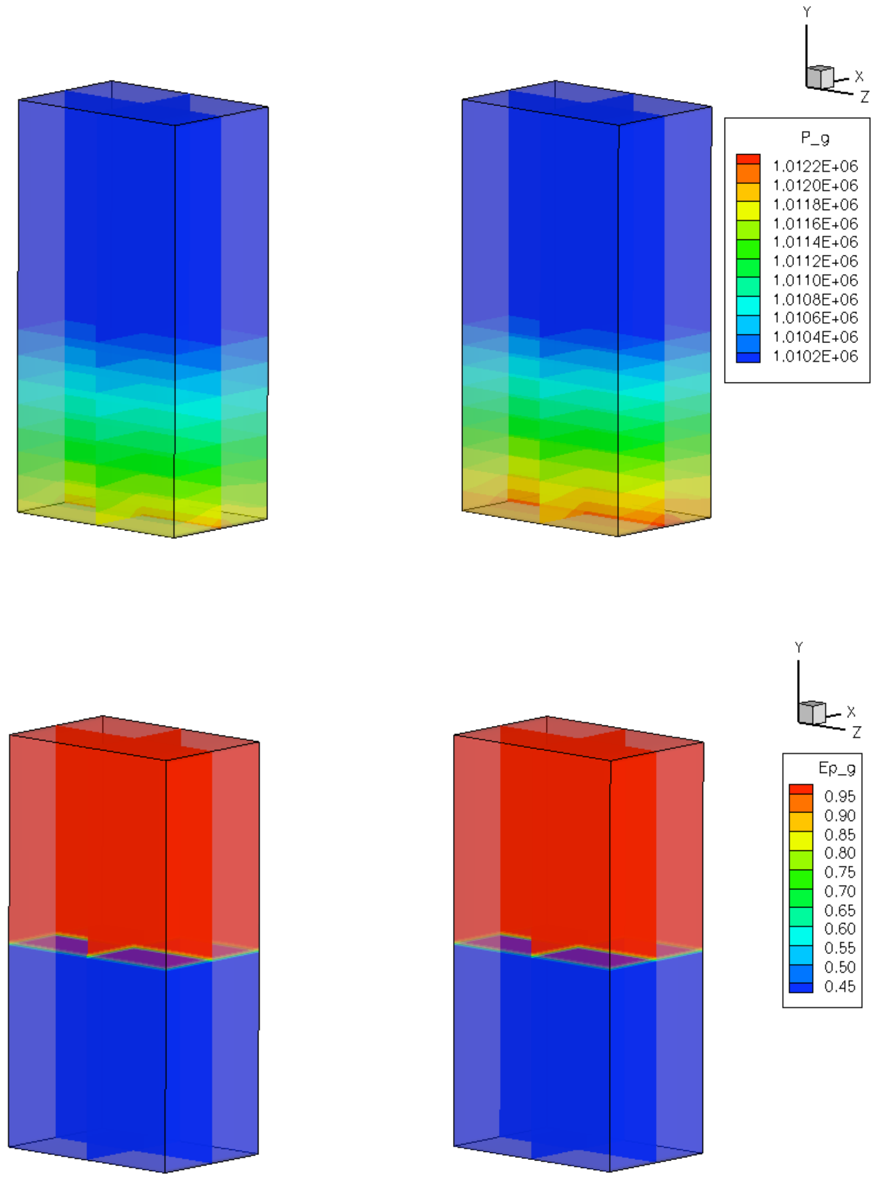


Figure 71: Case III contour plots at $t = 1.0$ s using MFIX (right) and ODEV (left): gas pressure, p_g, (top) and void fraction, ε_g (bottom).

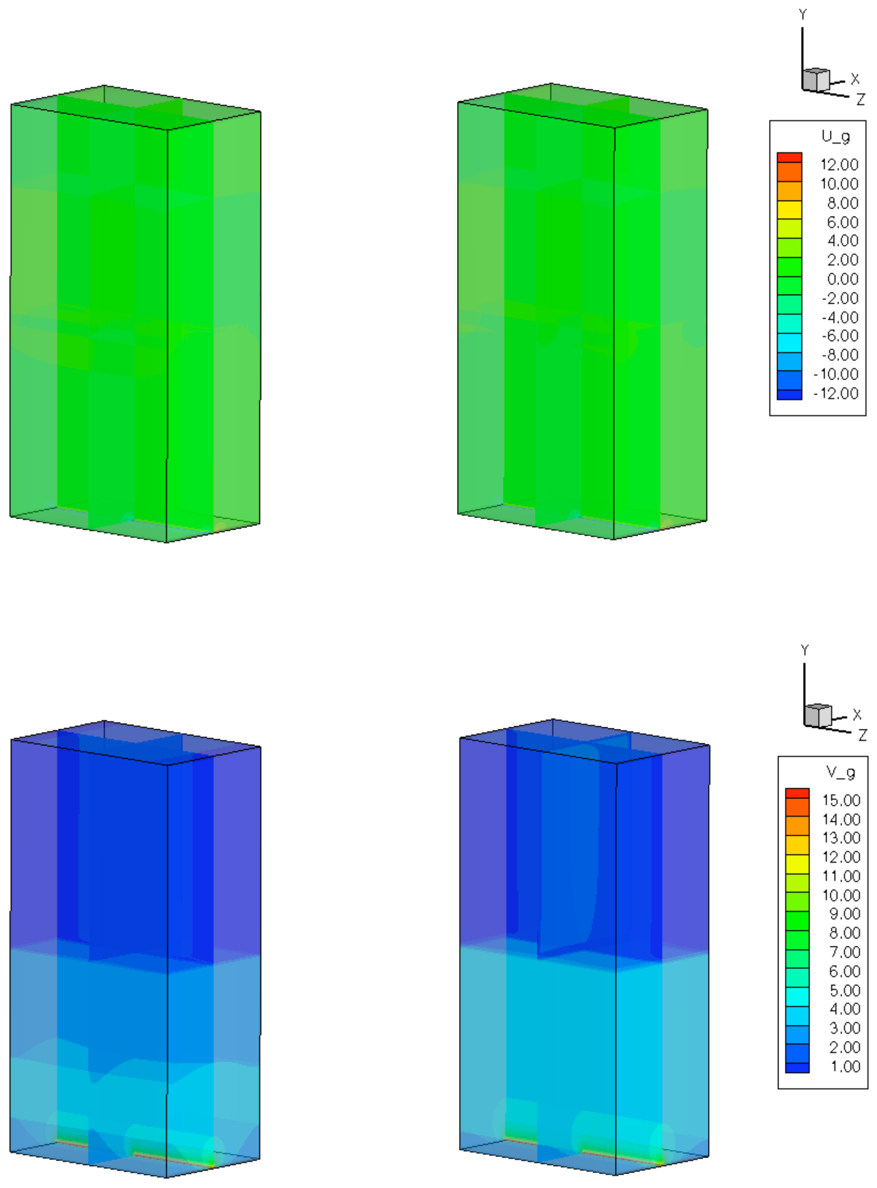


Figure 72: Case III contour plots at $t = 1.0$ s using MFIX (right) and ODEV (left): gas velocity, u_g (top) and gas velocity, v_g (bottom).

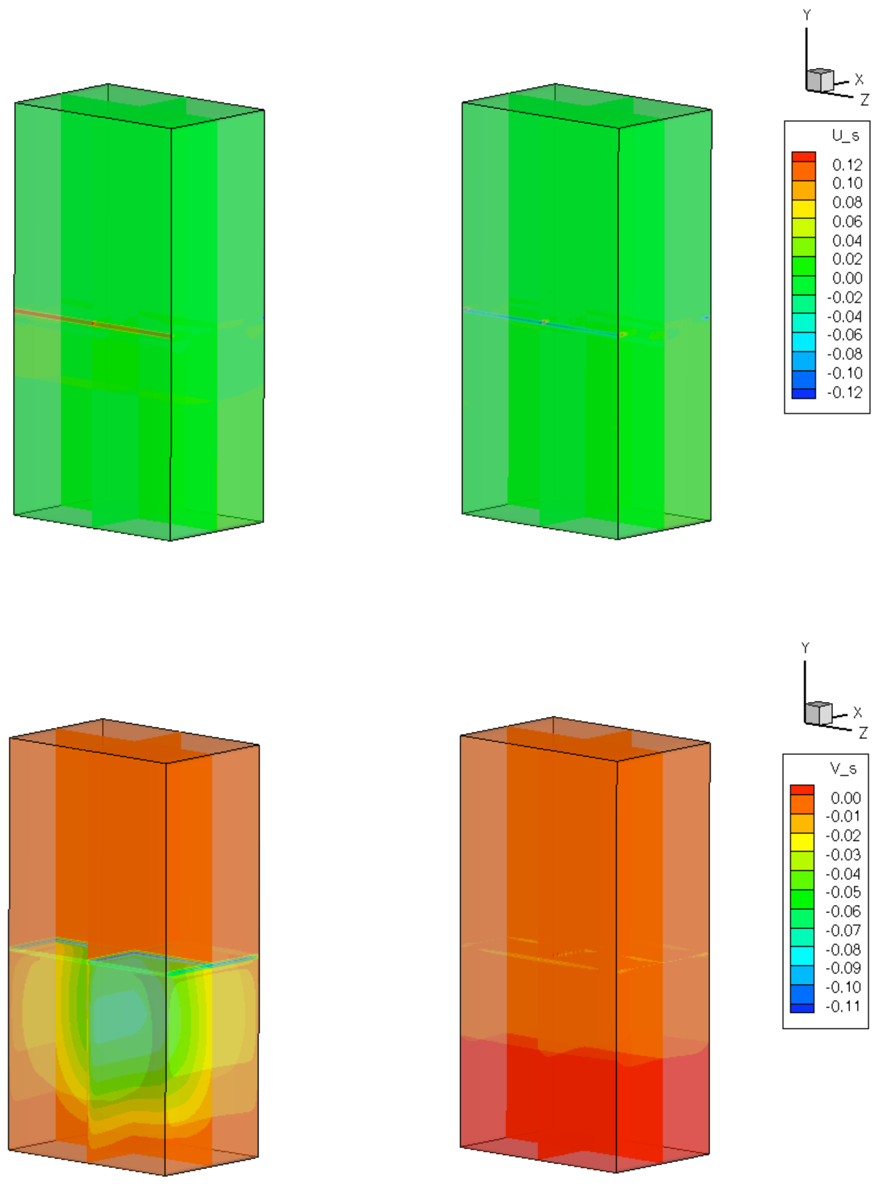


Figure 73: Case III contour plots at t = 1.0 s using MFIX (right) and ODEV (left): solids velocity, u_s, (top) and solids velocity, v_s (bottom).

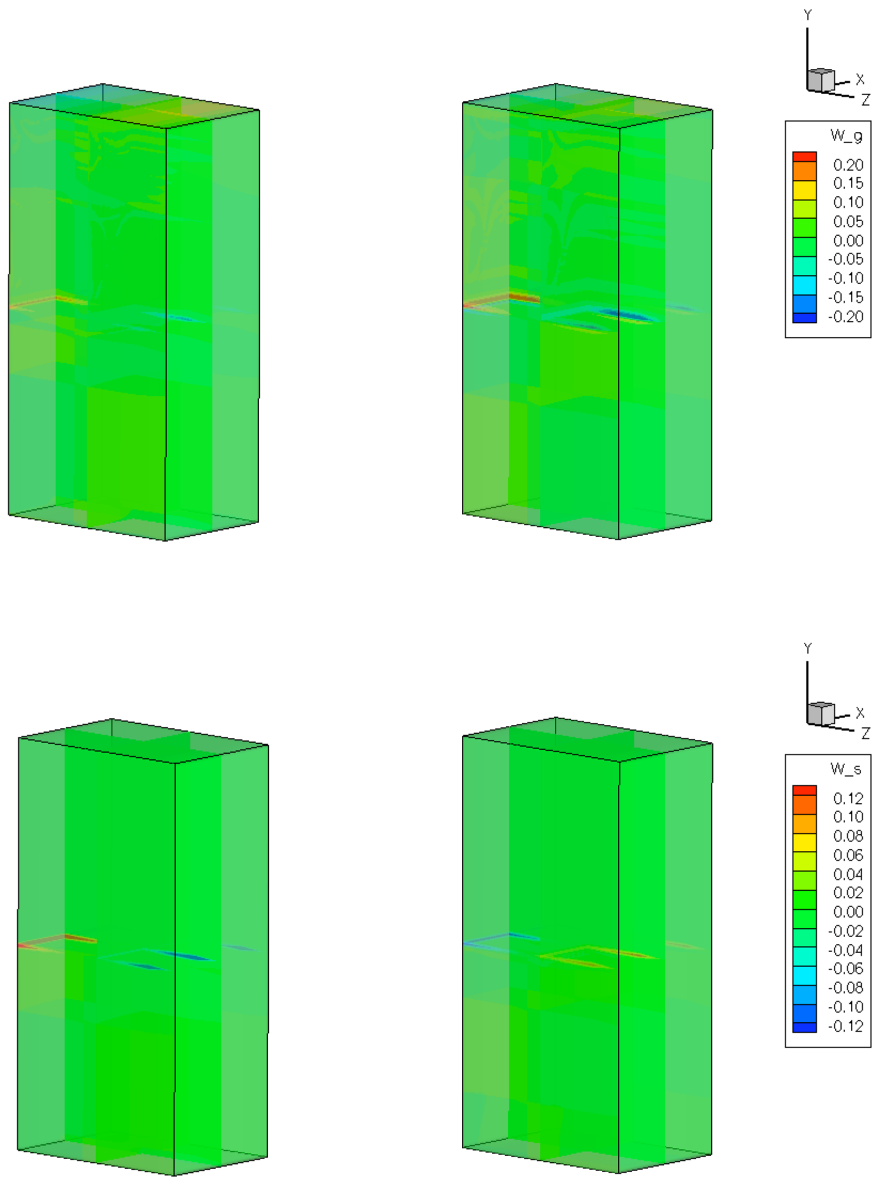


Figure 74: Case III contour plots at t = 1.0 s using MFIX (right) and ODEV (left): gas velocity, w_g, (top) and solids velocity, w_s (bottom).

From Figures 71- 74, the contour plots from ODEV that appear to match the best with those from MFIX are the pressure, void fraction, and x- and y-gas velocities. The remaining contour plots from ODEV for the solids velocities and the z-gas velocity do not appear to match those from MFIX. In the case of the z-gas velocity, ODEV over-predicts the velocity at the surface of the bed. For the x- and z-solids velocities, the velocity fields are the opposite of the corresponding MFIX results at the surface of the bed. The y-solids velocity computed by ODEV does not contain any of the structure computed by MFIX. To quantify the errors in the contour plots from MFIX and ODEV for each of the variables, a normalized error metric was used:

$$\epsilon = \frac{V_{MFIX} - V_{ODEV}}{\sqrt{V_{MFIX}^2 + V_{ODEV}^2}}, \tag{7.1}$$

where V is each of the field variables. Contour plots for errors associated with each of the field variables is shown in Figures 75- 78. The largest errors for each of the field variables, given in percentage error, is summarized in Table XXIII.

Table XXIII: Case III: Largest normalized error [%] for each field variable.

Variables	p_g	ε_g	u_g	u_s	v_g	v_s	w_g	w_s
Error [%]	0.018	5.78	141.4	141.4	53.2	138.7	141.4	141.1

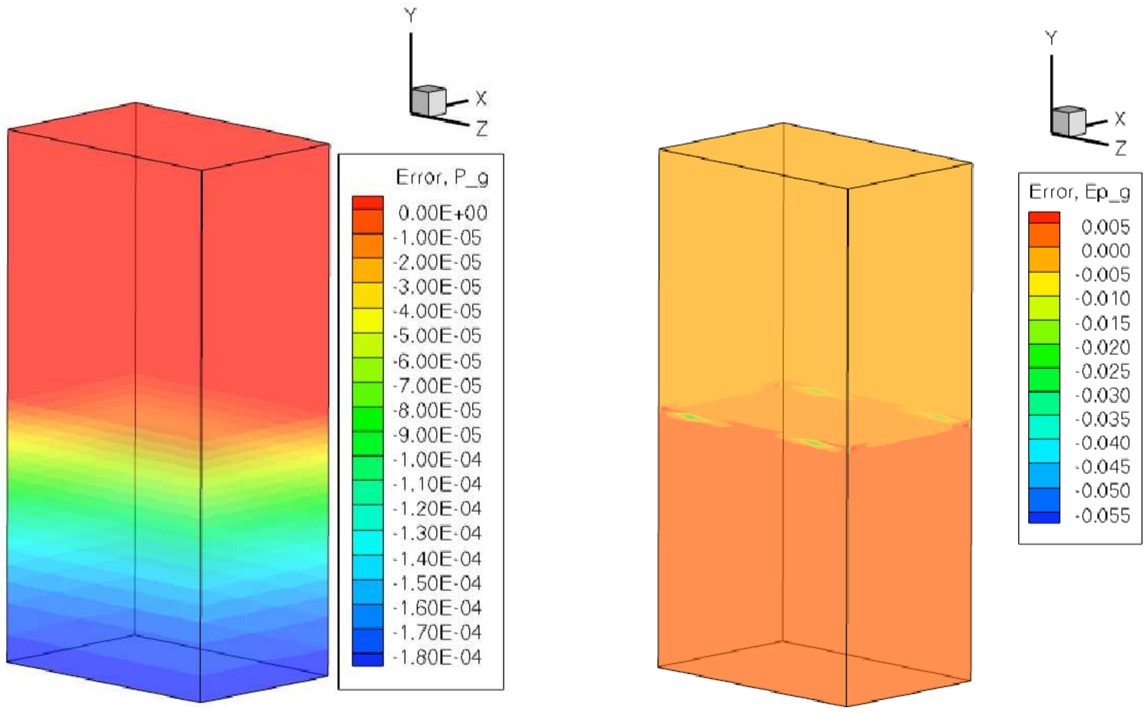
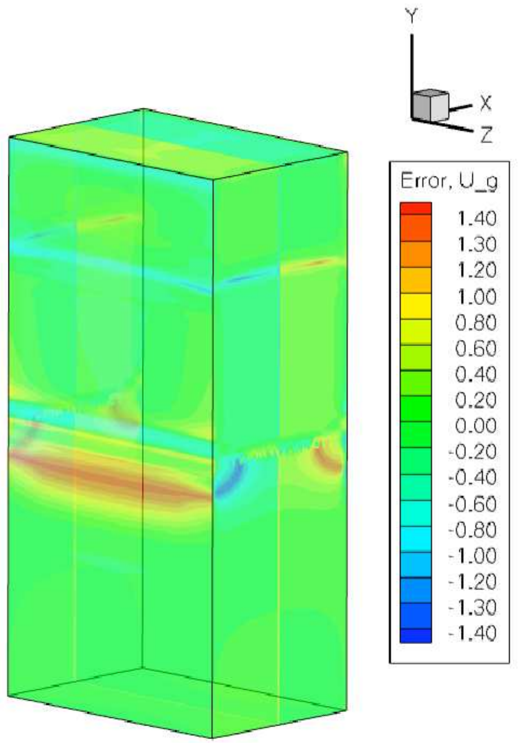


Figure 75: Case III error contour plots at $t = 1.0$ s between MFIX and ODEV for gas pressure, p_g, (left) and void fraction, ε_g (right).



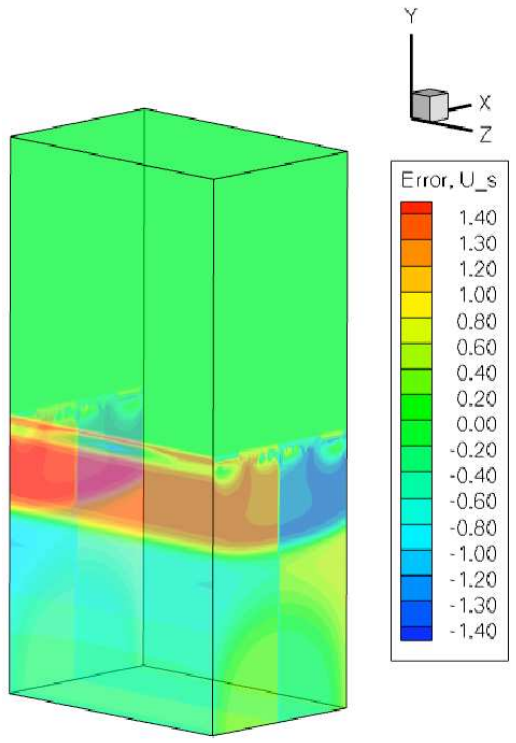
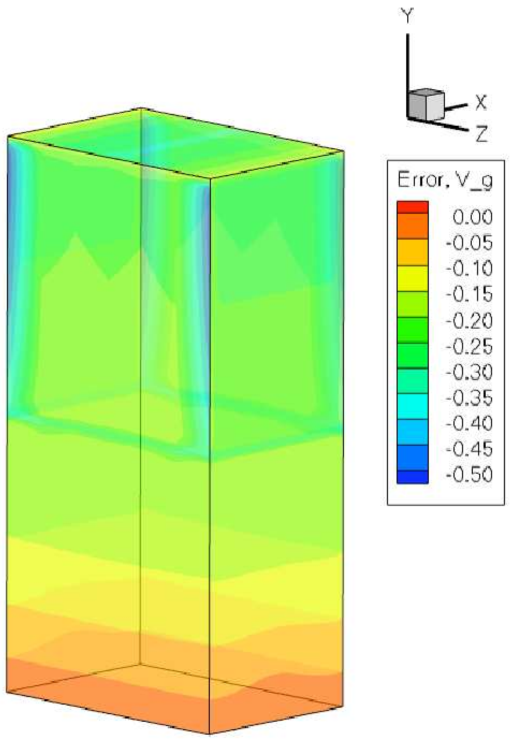


Figure 76: Case III error contour plots at $t = 1.0$ s between MFIX and ODEV for gas velocity, u_g, (left) and solids velocity, u_s (right).



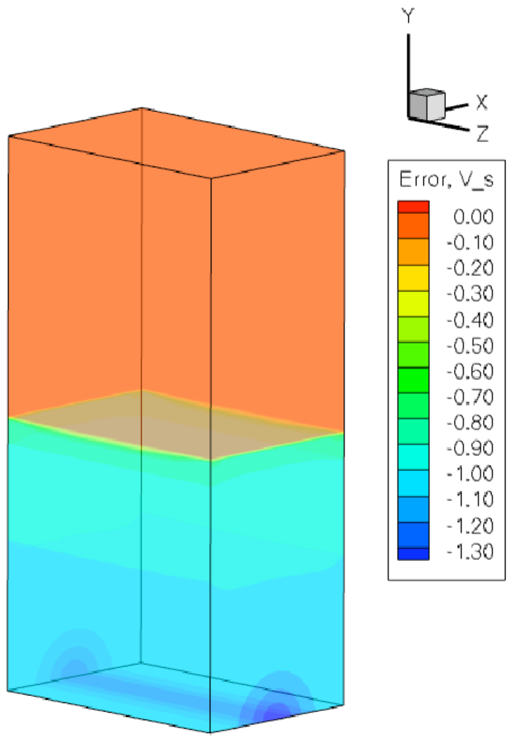


Figure 77: Case III error contour plots at $t = 1.0$ s between MFIX and ODEV for gas velocity, v_g, (left) and solids velocity, v_s (right).

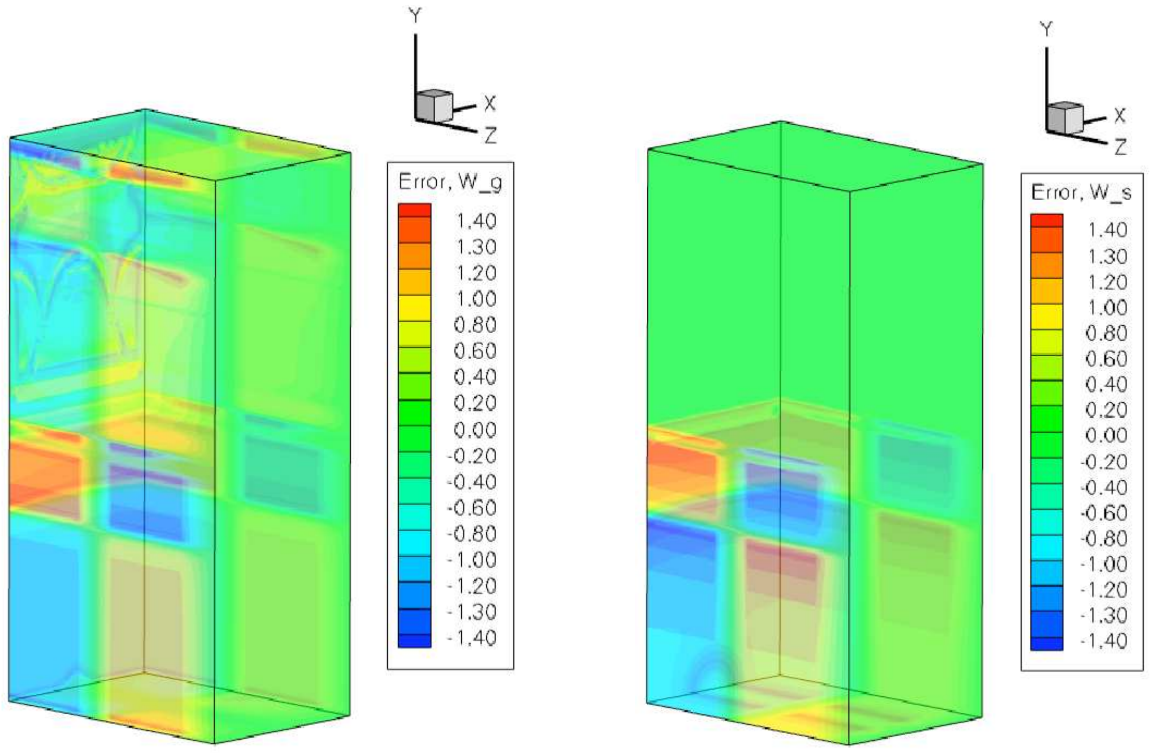


Figure 78: Case III error contour plots at $t = 1.0$ s between MFIX and ODEV for gas velocity, w_g, (left) and solids velocity, w_s (right).

The errors presented in Table XXIII show, with the exception of the void fraction and pressure, the field variables have errors much larger than 10%. The contour plots for the errors of u_g, u_s, w_g, and w_s, shown in Figures 76 and 78, prove that the results from ODEV are exactly opposite from the results of MFIX.

The poor performance of the ROM is most likely attributed to the inability to solve constrained ODEs using the current method. The limiting variable is the void fraction, which is bounded by the minimum packing fraction and a fully gaseous cell. Even though the maximum error associated with the void fraction from Table XXIII is only 5.78%, this error is large enough to cause the much larger errors associated with both the gas and solids velocities. Even slight changes in the void fraction computed by the solids correction equation (see the MFIX Numerics Guide for details [34]), as shown here, result in large changes to the computed velocities. It is concluded that while ODEV achieved a two order of magnitude speed-up factor over MFIX, the ROM did not achieve its goal of accurately modeling two-phase three-dimensional isothermal flows.

C. Single and Two-Phase Feature Extraction

This section presents results for three cases: (1) bubble extraction in two-phase fluidized beds, (2) shock and wake extraction on supercritical airfoils, and (3) shear layer and vortex extraction in turbomachinery seals.

1. Case I: Bubbling Fluidized Beds

a. Description of Phenomena

Bubbling in multiphase fluidized beds occurs when the velocity injected at the bottom of the bed (see Figure 6 in Chapter IV) reaches a critical velocity. A bubble is classified as a pocket of gas that surrounded by solids particles. The critical velocity, known as minimum bubbling velocity, varies for different particle sizes and compositions [32]. In the case of Geldart Group A particles, of which the solids particles in the study belong to, the minimum bubbling velocity is higher than the minimum fluidization velocity [32].

In the strict mathematical sense, bubbles are formed when shocks present in the bed intersect [32]. The formation of shocks in a fluidized bed has no closed form solution; therefore, only numerical solutions exist and no exact criteria exist for the formation of bubbles. Algebraic models exists, such as the Davidson model for bubble speed classification, but no model exists to determine the number, size, or location of bubbles in a fluidized bed [32, 35].

b. Results

The two-phase fluidized bed studied here is similar to the cases presented in the POD-based ROM results section (section B). The exact parameters are presented in Table XXIV [32]. This case was chosen because it was one of three cases presented by Gidaspow [32] where bubbles have been known to form within the fluidized bed.

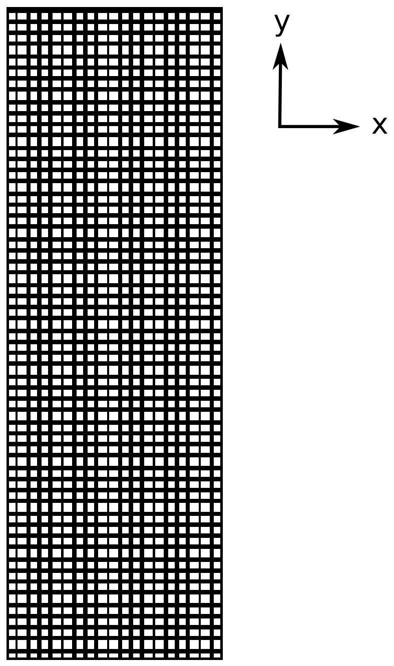


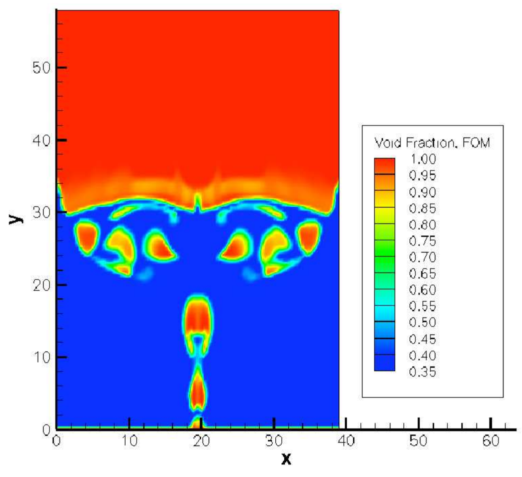
Figure 79: Case I: Computational domain for MEDts implementation of bubbling fluidized bed.

Table XXIV: Case I: Parameters of an isothermal bubbling fluidized bed.

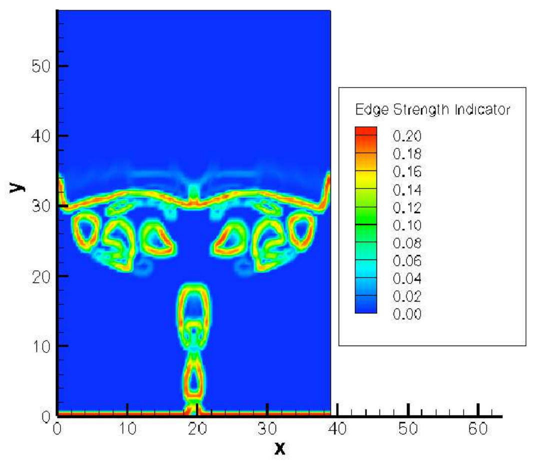
Parameter	Description	Units	
x_{length}	Length of the domain in x-direction	cm	39.37
y_{length}	Length of the domain in y-direction	cm	58.44
i_{max}	Number of cells in x-direction	-	108
j_{max}	Number of cells in y-direction	-	124
v_1	Gas jet velocity	cm/s	355.0
v_2	Gas distributor velocity	cm/s	28.4
p_{g_s}	Static pressure at outlet	g/(cm/s^2)	1.06×10^6
T_{g_0}	Gas temperature	K	297
μ_{g_0}	Gas viscosity	g/(cm/s)	1.8×10^{-4}
t_{start}	Start time	s	0.0
t_{stop}	Stop time	s	1.0
Δt	Initial time step	s	$1.0 \times 10{-}4$
ρ_s	Particle density	g/cm^3	2.61
D_p	Particle diameter	cm	0.08
h_{s0}	Initial height of packed bed	cm	29.2
ϵ_g^*	Initial void fraction of packed bed	-	0.4

The computational domain of case I is presented in Figure 79. The computational domain is disretized using a structured grid with uniform spacing in the x and y directions. Although the grid spacing in the x- and y-directions is not equal, no special treatment is required in the definition of the structuring elements due to the grid uniformity.

The void fraction, ε_g, is the best indicator of the presence of a bubble in bubbling fluidized beds. The morphology algorithm for structured domains, MEDts, is applied to the void fraction snapshot ensemble generated from the parameters of Table XXIV using MFIX. Figure 80 (a) shows the contour plot of the last snapshot of the void fraction at $t = 1$ s. The results from the algorithm without thresholding applied to the final snapshot is given in Figure 80 (b). The algorithm accurately depicts the bubbles located in the bed as well as the bed surface.



(a)



(b)

Figure 80: Case I: Void fraction, ε_g, in a bubbling fluidized bed at $t = 1$ s (a) with MEDts results without thresholding applied (b) to (a).

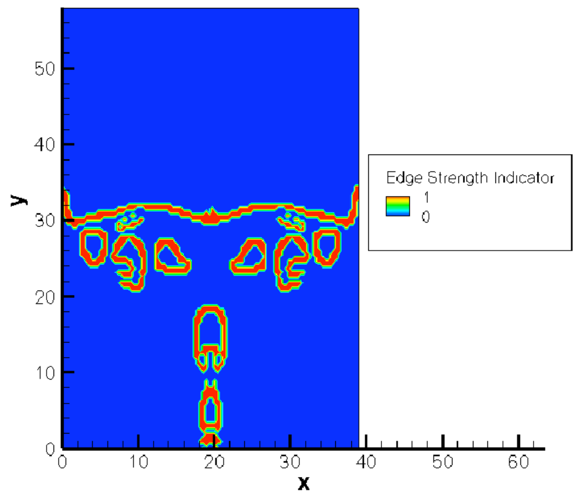


Figure 81: Case I: Results of MEDts with thresholding applied to a bubbling fluidized bed.

The next step is to apply thresholding using (3.12). The thresholding limits were ranged from 0.0 to 0.25, the upper and lower limits of the edge strength index. By numerical experimentaion, the optimal thresholding levels determined were $\alpha = 0.1$ and $\beta = 0.2$. Thresholding was applied at these levels to the void fraction snapshot of Figure 80. The results of applying thresholding to the image are shown in Figure 81.

The algorithm captures all bubbles with large gradients in the void fraction at their boundaries. However, bubbles with low gradients of the void fraction at their boundaries are not captured. This is not a problem, as the bubbles with low void fraction gradients should be captured by the normal POD basis functions. Actual testing of the augmented basis functions should verify this statement. The edges are generally two points in width [35], which shows the global thresholding technique (3.12) utilized in the algorithm is sufficient.

2. Case II: Transonic Supercritical Airfoils

a. Description of Phenomena

Transonic flow occurs at Mach numbers between 0.8 and 1.2. Some of the early work on understanding this complex regime was performed in 1948 by 1948 by Erikson and Robinson [48]. They measured local unsteady pressures on an oscillating wind tunnel model.

Transonic flows are known to exhibit strong unsteady characteristics [48]. These unsteady characteristics can be generated by either oscillating the airfoil or attached control surface [49] or on fixed supercritical airfoils, which are airfoils specially designed for this flow regime, at low angles of attack [50, 51]. Strong, moving shocks are the most common unsteady phenomena observed on airfoils at transonic speeds [52]. Tijdeman [49] observed three different unsteady shock motions on airfoils with moving control surfaces, shown in Figure 82. Type A shocks are present throughout the

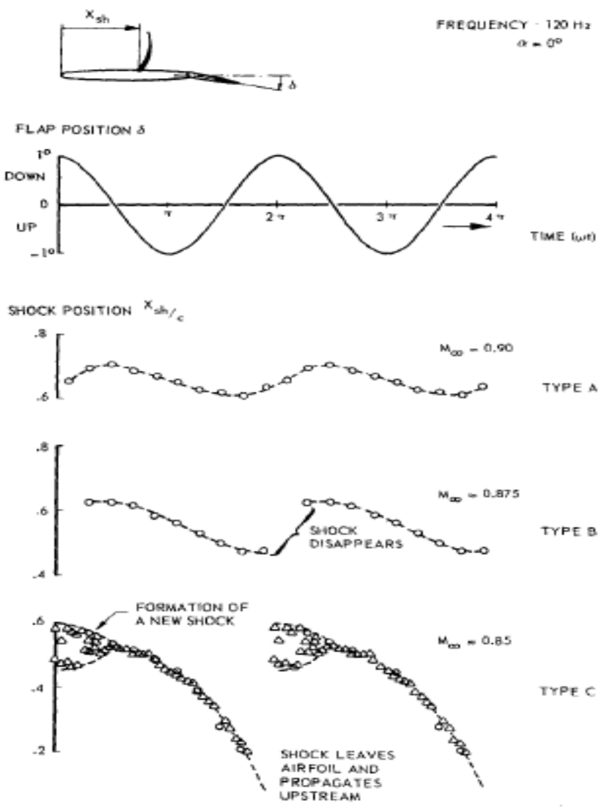


Figure 82: Typical shock oscillations for airfoil with oscillating flap.

oscillation and vary around a mean chord of 70%. Type B shocks are very similar to Type A shocks but appear to vanish once every period. Type C shocks are generated periodically at roughly 60% chord but propagate upstream into the flow [48].

Gibb [52] remarked that on supercritical airfoils Type A shocks are generated when: (i) the airfoil has a thickness/chord ratio greater than 10%, (ii) the shock must be strong with a Mach number of 1.3 just ahead of the shock, (iii) the airfoil mush have a large trailing edge angle, and (iv) the airfoil must have an average shock position aft of 50% chord. This can be seen in Figure 83 [52].

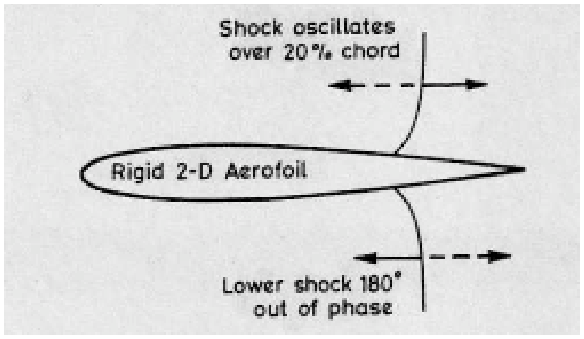


Figure 83: Typical shock oscillations for a stationary airfoil.

b. Results

The case presented here is similar to the one described by Gibb [52] for a 14% biconvex airfoil. This airfoil is common in wind tunnels and on aircraft for bomb fairings. The airfoil is subject to a freestream Mach number of 0.87. The Reynolds number for this flow, based the chord length, is 5.95×10^6. The specific parameters of the flow are given in Table XXV [52].

Table XXV: Case II: Parameters of an isolated transonic biconvex airfoil.

Parameter	Description	Units	
P_{tot}	Total Pressure	kPa	162.5
T_{tot}	Total Temperature	K	343.35
P_{ref}	Reference Pressure	kPa	101.325
T_{ref}	Reference Temperature	K	300
M_o	Freestream Mach Number	-	0.87
μ_o	Dynamic Viscosity	kg/(m s)	2.052×10^-5
Re_c	Reynolds Number	-	5.95×10^6
CFL	CFL number	-	1.75

The computational domain of case II is presented in Figure 84. This computational domain is discretized using a hybrid grid. The grid around the airfoil is an O-grid. The O-grid is grown outward from the airfoil until the majority of the viscous effects along the airfoil surface are contained within the O-grid. The remaining area of the computational domain is discretized using an unstructured triangular grid with a controlled growth rate.

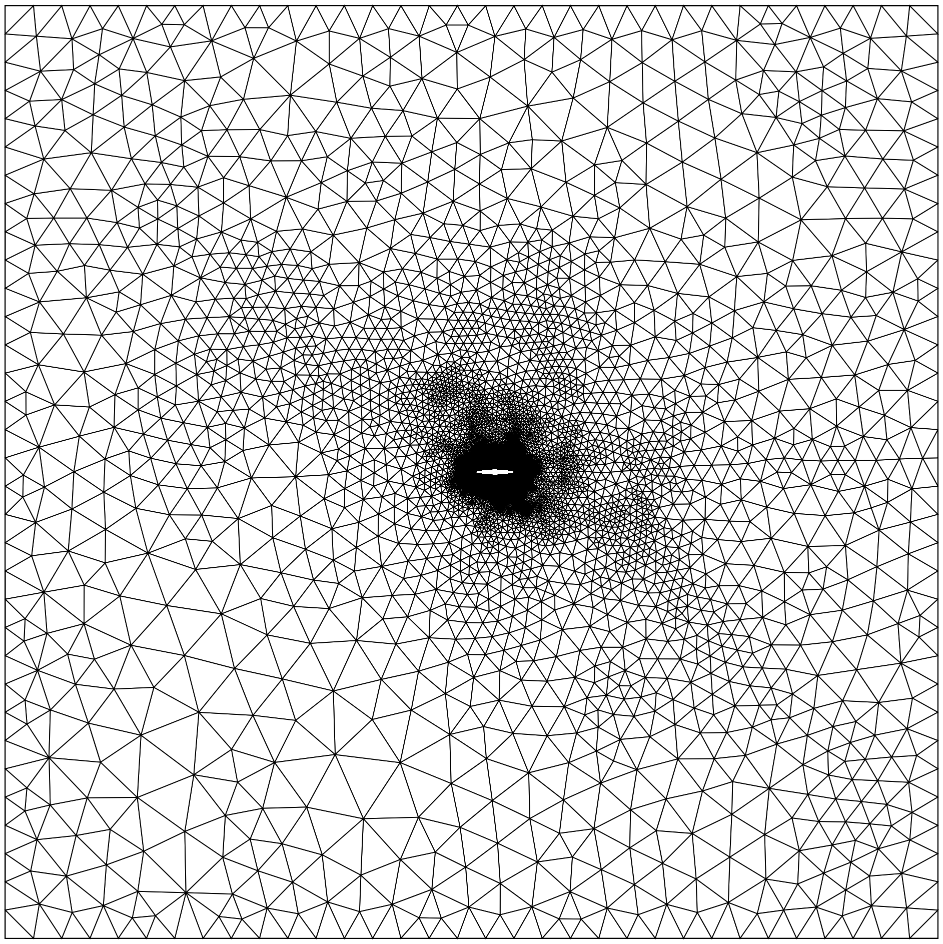
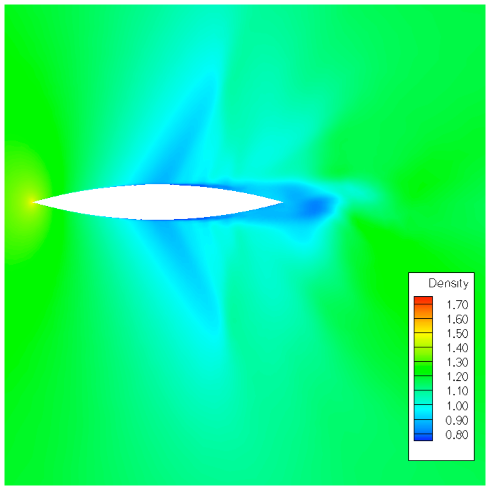
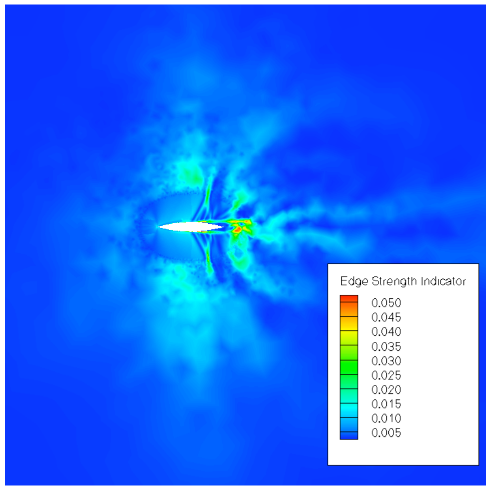


Figure 84: Case II: Computational domain for MEDuns implementation of a transonic biconvex airfoil.

For shocks, the density, ρ, is an indicator of the presence shocks on the surface. The morphology algorithm for general domains, MEDuns, is applied to the density snapshot ensemble generated from the parameters of Table XXV using UNS3D. Figure 85 (a) shows the last snapshot of the density from the unsteady simulation. The results from the algorithm applied to the final snapshot without thresholding or limiting applied is given in Figure 85 (b). The algorithm does a good job of depicting the wake and the vortices shed, but also picks up a significant amount of noise far away from the airfoil and at the interface of the structured and unstructured grids. This is attributed the varying sizes of the structuring elements throughout the grid.



(a)



(b)

Figure 85: Case II: Density contour from UNS3D (a) with MEDuns applied to transonic airfoil without thresholding or limiting applied (b) to (a).

Figure 86 shows the computational domain near the interface of the O-grid and the unstructured grid. Notice that some of the cell faces are much larger than their neighboring cell faces. The structuring elements for the nodes on these faces stretch into areas that contain shocks, leading to the “false” edges, that is image points with

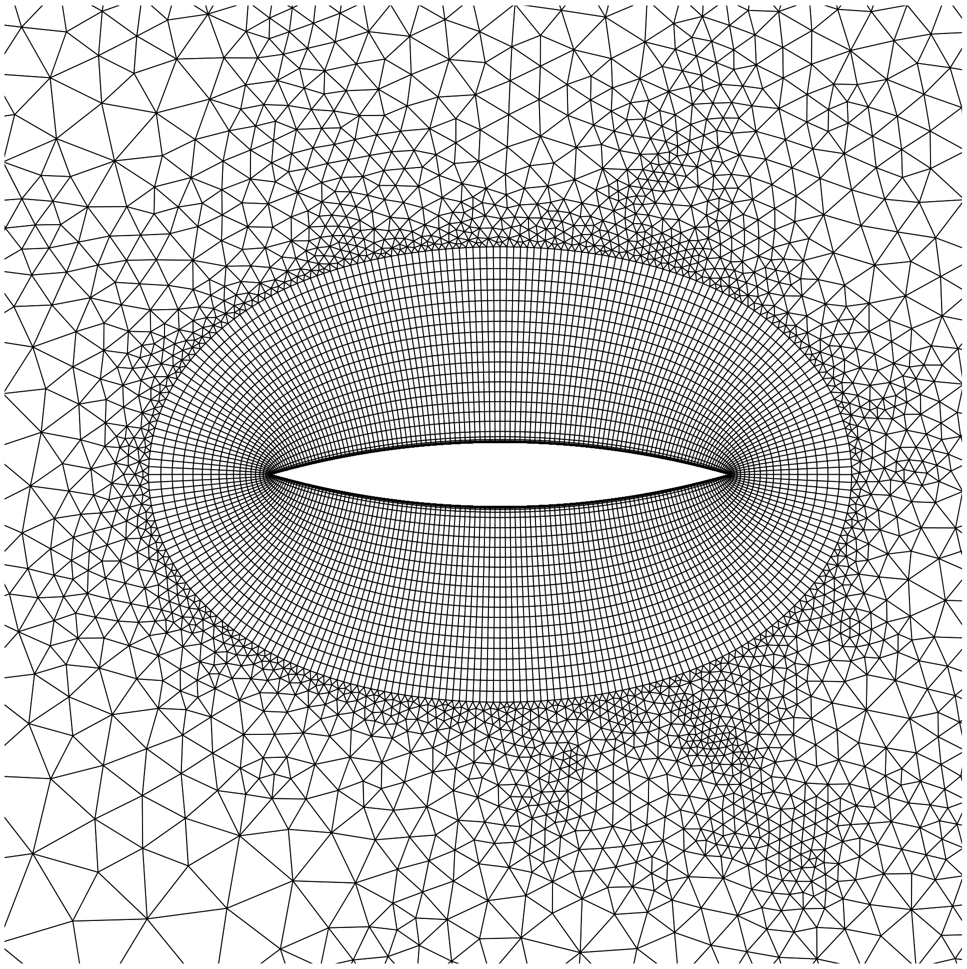


Figure 86: Case II: zoomed computational domain for MED implementation of a transonic biconvex airfoil.

large values for the edge strength index where the gradients of the indicated variable are low, far away from the airfoil. This problem is alleviated by limiting the size of the structuring elements to help enforce pseudo uniformity (see page 29). Through numerical experimentation, the upper limit for the size of the structuring element for case II was found to be 0.005, or roughly the average length of an edge in the O-grid. The results of the algorithm with structuring element size limiting applied are given in Figure 87. The noise is significantly reduced without any loss in accuracy locating the shocks and vortices.

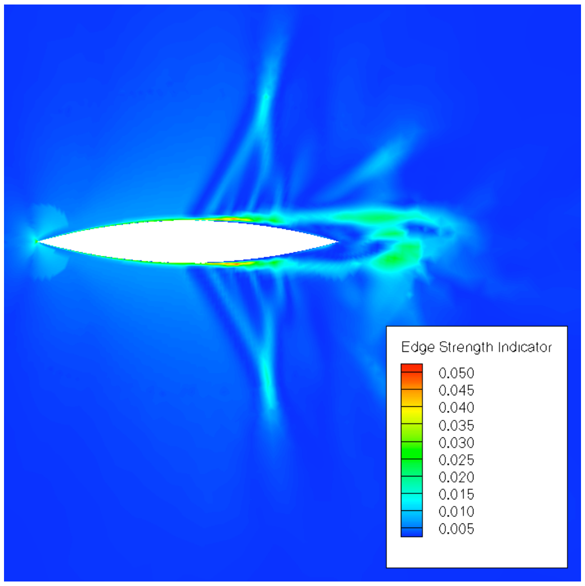


Figure 87: Case II: Results of MEDuns with structuring element limiter applied to transonic biconvex airfoil.

The final step is to apply thresholding using (3.12). The thresholding limits were ranged from 0.0 to 0.06, the upper and lower limits of the edge strength index. By numerical experimentation, it was determined that the optimal thresholding levels were $\alpha = 0.0049$ and $\beta = 0.03$. Thresholding was applied at these levels to the density snapshot of Figure 85a. The results of applying thresholding to the image are shown in Figure 88.

The algorithm captures the shocks and vortices, but it is difficult to define thresholding limits to accurately locate both the shocks and vortices. The difficulty in using a global thresholding technique for flows similar to this is that no one set of limits can properly handle edges of varying strength. The algorithm also locates the stagnation point and boundary layer before separation. The shocks also are roughly two

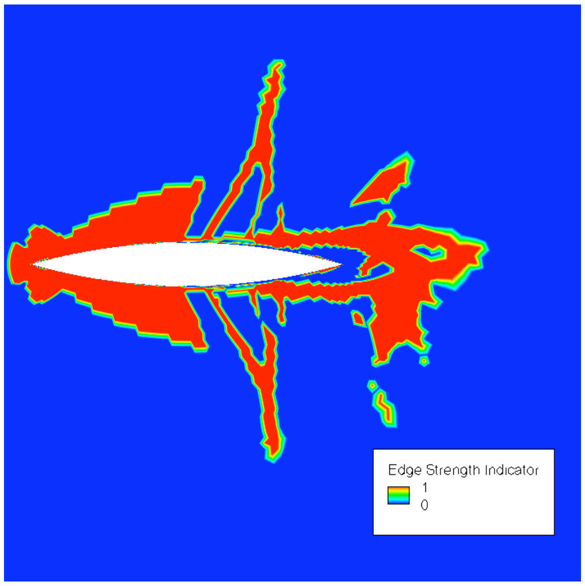


Figure 88: Case II: Results of MEDuns with thresholding applied to a transonic biconvex airfoil.

points in width, but the wake is significantly wider. To address these issues, a local thresholding technique should be applied. The use of a local thresholding technique would correctly extract edges of varying strength. The algorithm accurately indicates the shocks and vortices present in the flow, but thresholding of the shock-indicated image needs to be studied more to completely extract the shocks and vortices.

3. Case III: Cavity Flows in Turbomachinery Seals

a. Description of Phenomena

Cavity flow occurs in several applications, such as aircraft wheel and weapon bays [53], car doors [54], and turbomachine stator seals [55]. Cavity flows generate instabilities when the freestream or bulk flow interacts with the acoustics generated within the cavity. The interactions have been shown in experiments to generate a sharp increase in observed friction factors [56].

In open cavities, that is cavities with no opposing wall, flow instabilities are generated by the mean flow moving over the cavity producing a feed-back loop. The feed-back loop is formed when the shear layer oscillations in the cavity interact with vortices traveling along the shear layer and pressure wave acoustics radiating upstream from the cavity. Figure 89 [55] shows the interaction.

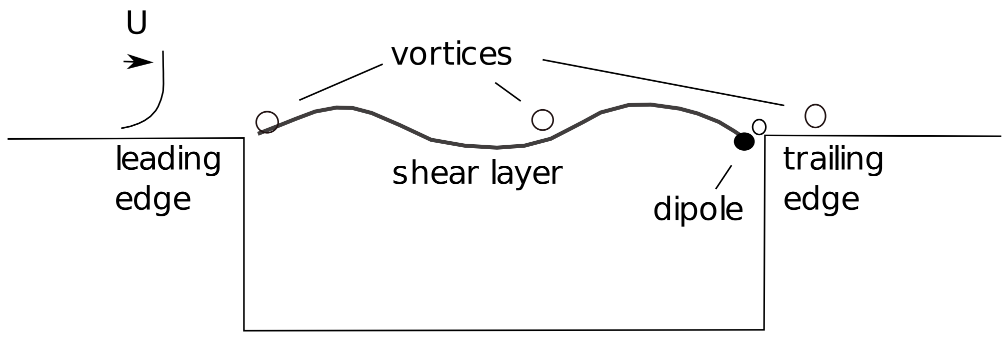


Figure 89: Cavity flow features.

When the boundary layer reaches the leading edge of the cavity it separates to form an oscillating shear layer across the cavity. The oscillation excites modes which generate vortices and causes them to propagate from the leading edge. These vortices travel downstream with the shear layer and impinge on the trailing edge of the cavity [55]. The vortices cause the shear layer to distort and warp as it reaches the trailing edge. This warping forces the attachment point to locate temporarily below the lip of the cavity generating a momentary increase in pressure. An acoustic dipole is exicted that in turn generates pressure waves. These pressure waves travel upstream and excite the shear layer causing more vortices to form and shed from the leading edge. This interaction between the vortices and pressure waves is what generates the feed-back loop. The same phenomenon is present in closed cavities, though the generated acoustics also interact with the boundary layer on the top channel wall, if the wall is sufficiently close, causing additional flow instabilities.

b. Results

The case presented here is similar to the one described by Liliedahl [55] for a two-dimensional turbomachinery seal. The seal inlet Mach number is 0.2. The Reynolds number based on the cavity length is 14,300. The seal height is 28 mils. The specific parameters of the flow are given in Table XXVI [55].

Table XXVI: Case III: Parameters of a two-dimensional turbomachinery seal.

Parameter	Description	Units	
P_{tot}	Total Pressure	kPa	104.191
T_{tot}	Total Temperature	K	305.4
P_{ref}	Reference Pressure	kPa	101.325
T_{ref}	Reference Temperature	K	300
M_o	Inlet Mach Number	-	0.2
ν_o	Kinematic Viscosity	m^2/s^2	1.6261×10^-5
Re_c	Reynolds Number	-	14300

The computational domain of case III is presented in Figure 90. This domain is discretized using a structured grid with nodes clustered on the leading and trailing edges of the cavity and along the upper and lower portions of the channel. The clustering is such that the computational domain will be able to capture the moving shear layer and vortices generated in the seal.

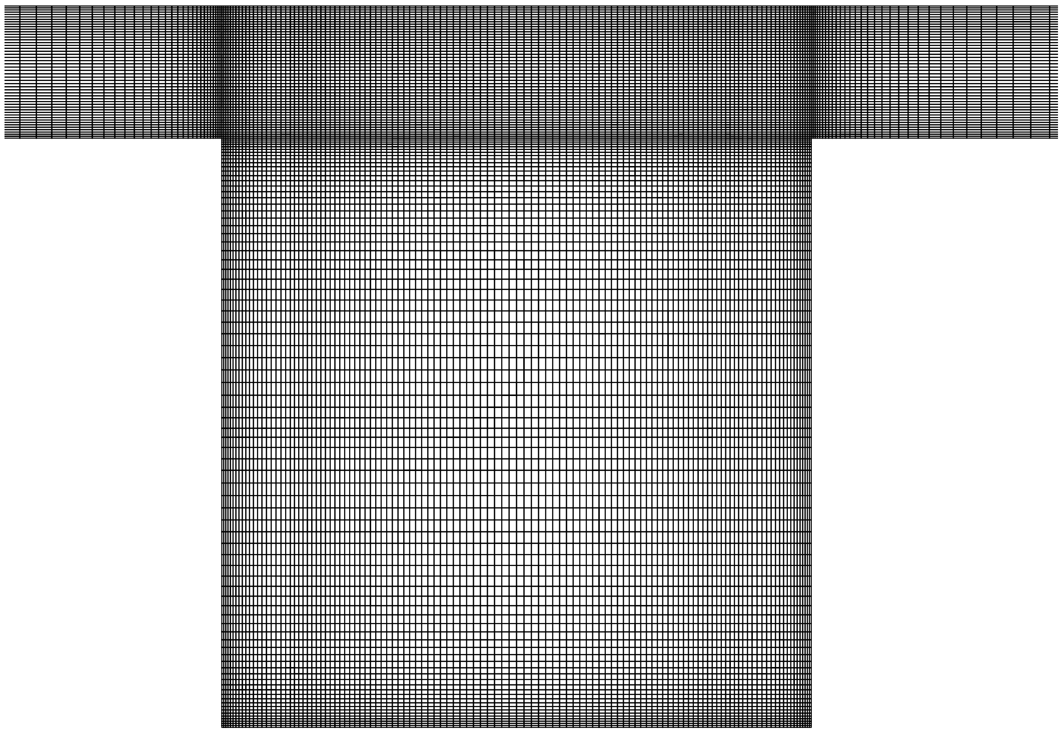


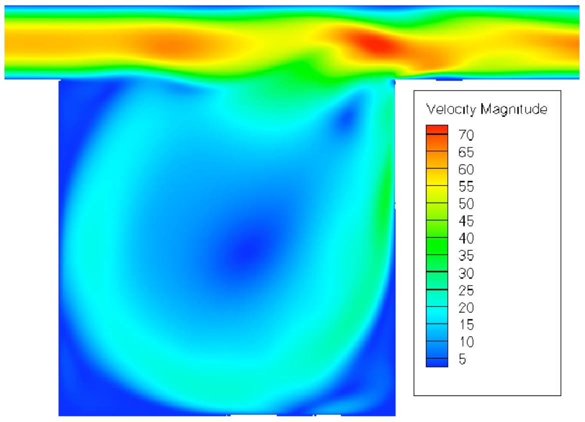
Figure 90: Case III: computational domain for MED implementation of 2-D turbomachinery seal.

The algorithm for general domains, MEDuns, is applied to the snapshot ensemble generated from the parameters of Table XXVI using UNS3D. For this case, the features of interest are the shear layer and any vorticies generated in the flow. Neither of these features can be considered shocks, so density is not a good candidate for indicating either the shear layer or vortices. Pressure is also a poor candidate for indicating either feature because of the small local pressure gradients present in the flow. Two choices considered herein are therefore the velocity magnitude, V, and viscosity, μ. Both terms appear in the definition of the shear stress in the fluid, given as:

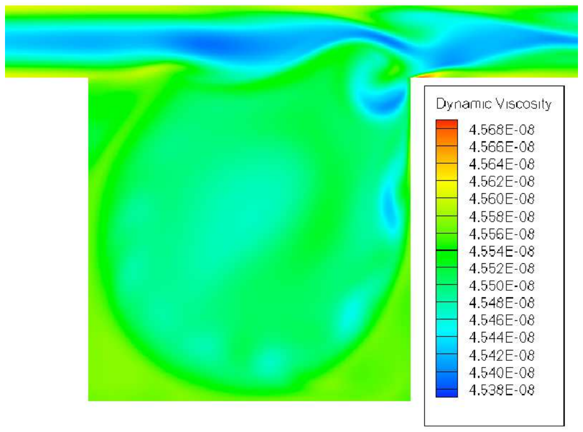
$$\tau = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \ . \tag{7.2}$$

Any changes in viscosity should take place within the shear layer as well.

The algorithm is applied to the snapshot ensemble for the velocity magnitude and viscosity generated from the parameters of Table XXVI using UNS3D. Figure 91 shows the last snapshot of the velocity magnitude and viscosity. The results from the algorithm applied to each of the final snapshots are given in Figure 92. The algorithm depicts both the shear layer and the vortices generated. The algorithm applied to the viscosity snapshot does an excellent job depicting the vortices as well. Both results obtained from the algorithm clearly show the effects of node clustering, which causes the "depressions" of lower edge strength to appear in the results.

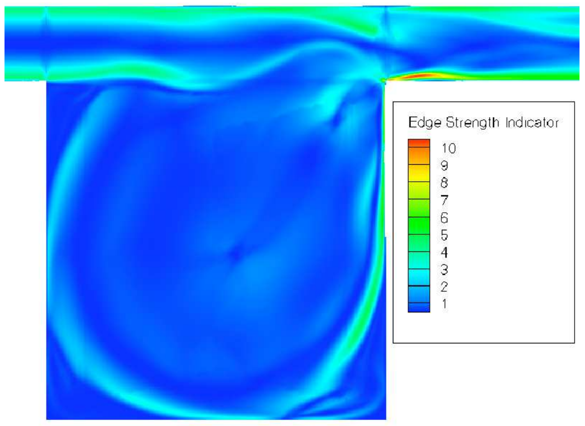


(a) Velocity Magnitude

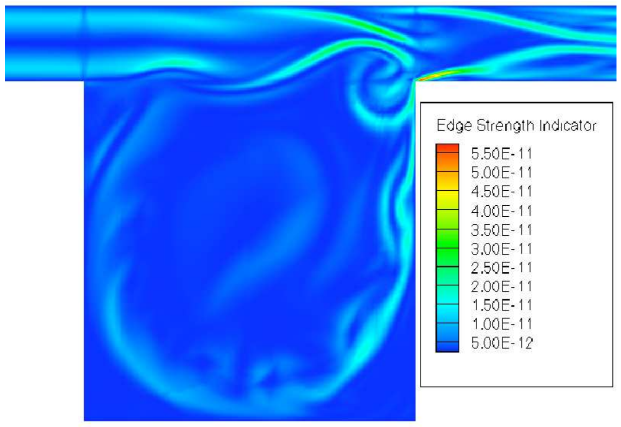


(b) Viscosity

Figure 91: Case III: Velocity magnitude (a) and viscosity (b) results from UNS3D for the turbomachinery seal.



(a) Velocity Magnitude



(b) Viscosity

Figure 92: Case III: MEDuns results for velocity magnitude (a) and viscosity (b) for the turbomachinery seal.

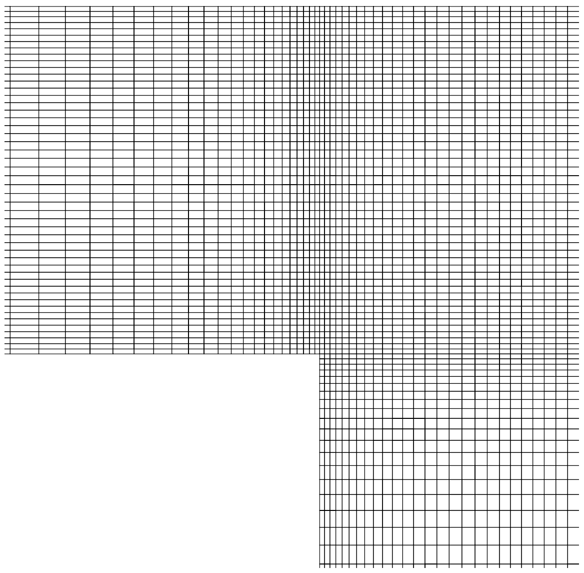
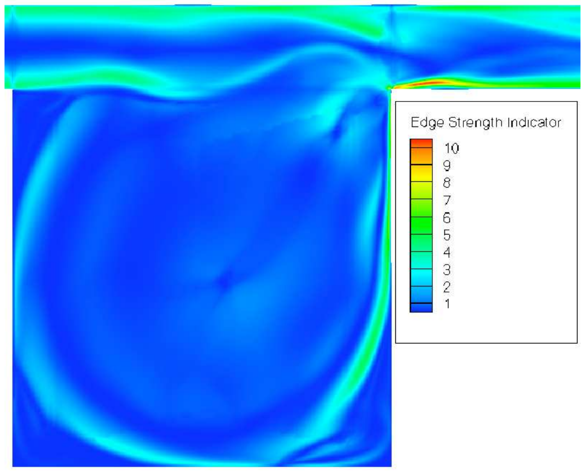
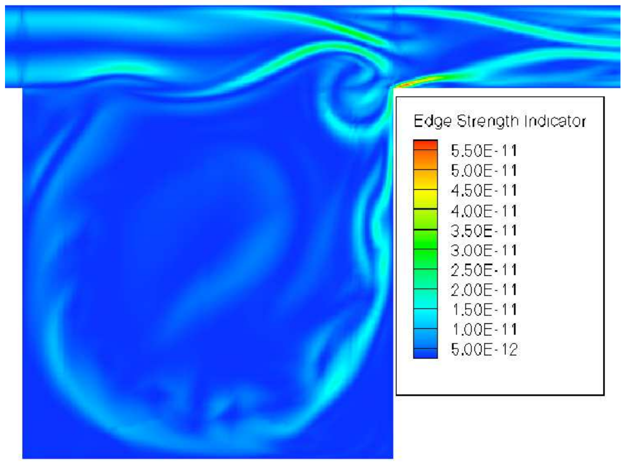


Figure 93: Case III: zoomed computational domain for MEDuns implementation of 2-D turbomachinery seal.

Figure 93 shows the computational domain near the leading edge of the seal. It is clear from Figure 93 that the faces near the channel wall are much smaller than those faces at 10% of the channel height. The structuring elements for feature detection are not reaching far enough into the domain, which causes the “depressions” seen in Figures 91a and 91b where node clustering is applied. For this case, the size of the structuring elements must have a lower size limit to enforce pseudo uniformity (see page 30). Through numerical experimentation, the lower limit for the size of the structuring elements was determined to be 1.77×10^{-5}, or the height of the faces at 10% of the channel height. The results of the algorithm with limiting applied are given in Figure 94. The effect of the cell growth rate is significantly reduced without any appreciable accuracy lost.



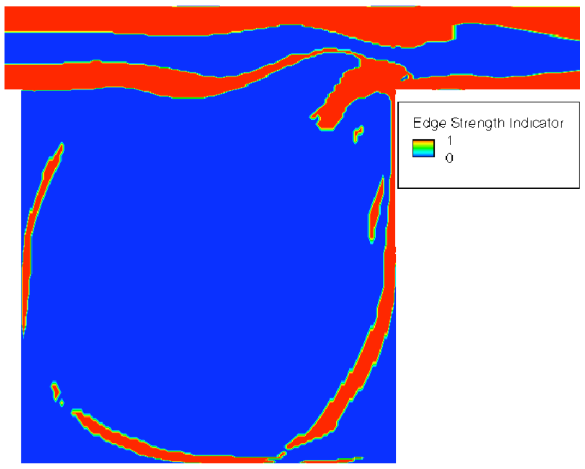
(a) Velocity Magnitude



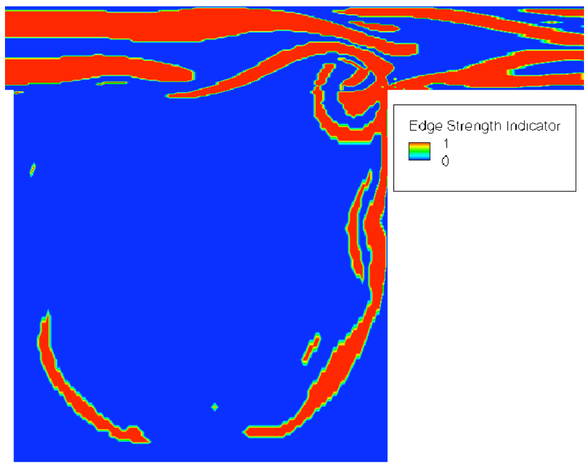
(b) Viscosity

Figure 94: Case III: MEDuns results for velocity magnitude (a) and viscosity (b) for the turbomachinery seal with limiter applied.

The final step is to apply thresholding using (3.12). The thresholding limits were ranged from 0.0 to 14.0 for the velocity magnitude and 0.0 to 7.25×10^{-11} for the viscosity. By numerical experimentation, it was determined that the optimal thresholding levels were 2.0 and 13.9 for the velocity magnitude and 7.5×10^{-12} and 7.25×10^{-11} for viscosity. Thresholding was applied at these levels to the velocity magnitude and viscosity snapshots of Figure 91. The results of applying thresholding to the image are shown in Figure 95. The algorithm captures the shear layer and vortices accurately, but as with case II the application of a local thresholding technique would improve the resolution and fully satisfy the edge thickness condition [35].



(a) Velocity Magnitude



(b) Viscosity

Figure 95: Case III: MEDuns results for velocity magnitude (a) and viscosity (b) for the turbomachinery seal with limiter and thresholding applied.

D. Summary

In this chapter the results of three cases for POD-based reduced-order models and three cases for the morphological-based feature detection algorithms were presented.

The first case for the POD-based ROM was the improved model for two dimensional non-isothermal flow first developed by Richardson [2]. Investigation showed that the POD-based ROM, ODEti, which did not use the acceleration techniques developed by Richardson [2], produced accurate results and was 8 times faster than the FOM, MFIX. The second case investigated was a two-dimensional isothermal flow with granular energy. This investigation showed that the POD-based ROM, ODEg, which did not use the acceleration techniques developed by Richardson [2], was 10.5 times faster than the FOM, MFIX, and produced accurate results. The last POD-based ROM case was an investigation of three-dimensional isothermal flow. While the POD-based ROM, ODEV, which did not use the acceleration techniques developed by Richardson [2], was 12 times faster than MFIX, ODEV produced errors for the gas and solids velocities larger than 10%. It was concluded that the POD-based ROM did not produce accurate results because of inability of the current method to model constrained ODEs, which exhibit the Gibbs phenomena [33] due to the approximation of the field variables. This yeilds inaccurate results as the variables approach the constraints on the void fraction imposed on the system.

The first case for the morphological-based feature detection algorithm was a two-dimensional bubbling two-phase flow. The algorithm successfully detected all of the bubbles and further showed the ability to extract bubbles with large void fraction gradients at the edges of the bubbles. The second case was a transonic airfoil. The algorithm detected the shocks and shed wake vortices. The algorithm, however, had a hard time extracting both the vortices and shocks because of the use of global thresholding by the morphological-based feature detection algorithm. The final case was a turbomachinery seal. The algorithm detected the shear layer within the cavity and the vortices shed from the shear layer interaction. The algorithm had a hard time extracting the shear layer and vortices because of the use of global thresholding by the morphological-based feature detection algorithm.

CHAPTER VIII

CONCLUSIONS & FUTURE WORK

A. Conclusions

This thesis presented the development of POD-based ROMs to simulate two-dimensional non-isothermal flow, two-dimensional isothermal flow with granular energy, and three-dimensional isothermal flow in two-phase fluidized beds. Results were presented for each of the POD-based ROMs and were compared to the full-order model. The presented cases used for comparison were minimum fluidization cases. The results showed that the two-dimensional POD-based ROMs produced accurate results while requiring relatively few basis functions for each field variable. The three-dimensional POD-based ROM did not produce accurate results because of the inability of the current method to solve constrained ODEs, which exhibit the Gibbs phenomena [33] due to the approximation of the field variables. This yeilds inaccurate results as the variables approach the constraints on the void fraction imposed on the system.

This thesis also presented the development of a feature detection algorithm based on mathematical morphology. This algorithm was utilized for structured and unstructured computational domains. The algorithm was applied to and results presented for a bubbling fluidized bed, a transonic airfoil, and a turbomachinery seal. The results show that the algorithm was able to extract the features of interest from each of the flows presented computed on structured and unstructured computational domains.

B. Future Work

The challenge for the future is to improve the ability of POD-based ROMs to model constrained ODEs. Additionally, the features extracted using the morphological feature detection algorithm need to be integrated and tested with the augmented POD algorithm currently in development [35] to model flows with moving discontinuities. The lessons learned from this thesis may eventually be applied to generate a POD-based ROM for turbomachinery.

REFERENCES

[1] T. Yuan, Reduced-order modeling for transport phenomena based on proper orthogonal decomposition, Master's Thesis, Texas A&M University, College Station, TX, 2003.

[2] B. R. Richardson, A reduced-order model based on proper orthogonal decomposition for two-phase flow non-isothermal transport phenomena, Master's Thesis, Texas A&M University, College Station, TX, 2007.

[3] D. J. Lucia, P. S. Beran, W. A. Silva, Reduced-order modeling: new approaches for computational physics, Journal of Progress in Aerospace Sciences 40 (2004) 51–117.

[4] R. Florea, K. C. Hall, P. G. A. Cizmas, Reduced-order modeling of unsteady viscous flow in a compressor cascade, AIAA Journal 36 (1998) 1039–1048.

[5] J. P. Thomas, E. H. Dowell, K. C. Hall, Static/dynamic correction approach for reduced-order modeling of unsteady aerodynamics, Journal of Aircraft 43 (2006) 865–878.

[6] M. C. Romanowski, E. H. Dowell, Reduced order Euler equations for unsteady aerodynamic flows - numerical techniques, in: 34th Aerospace Sciences Meeting and Exhibit, AIAA Paper 1996-528, Reno, Nevada, 1996.

[7] G. Berkooz, P. Holmes, J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annual Review of Fluid Mechanics 25 (1993) 539–575.

[8] L. Sirovich, Turbulence and the dynamics of coherent structures, Quarterly of Applied Mathematics 45 (1987) 561–590.

[9] P. G. A. Cizmas, A. Palacios, T. O'Brien, M. Syamlal, Proper-orthogonal decomposition of spatio-temporal patterns in fluidized beds, Chemical Engineering

Science 58 (2003) 4417–4427.

[10] P. G. A. Cizmas, A. Palacios, Proper orthogonal decomposition of turbine rotor-stator interaction, AIAA Journal of Propulsion and Power 19 (2003) 268–281.

[11] P. S. Beran, W. A. Silva, Reduced-order modeling: new approaches for computational physics, in: 39th AIAA Aerospace Sciences Meeting, Reno, NV, 2001.

[12] H. M. Park, M. W. Lee, An efficient method of solving the Navier-Stokes equations for flow control, International Journal for Numerical Methods in Engineering 41 (1998) 1133–1151.

[13] P. LeGresley, J. Alonso, Investigation of non-linear projection for POD based reduced-order models for aerodynamics, in: AIAA 39th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2001-0926, Reno, NV, 2001.

[14] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, S. A. Orszag, Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders, Physics of Fluids 3 (1991) 2337–2354.

[15] K. C. Hall, J. P. Thomas, E. H. Dowell, Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows, AIAA Journal 38 (2000) 1853–1862.

[16] O. K. Rediniotis, J. Ko, A. Kurdila, Reduced order nonlinear Navier-Stokes models for synthetic jets, Journal of Fluids Engineering 124 (2002) 433–443.

[17] C. L. Pettit, P. S. Beran, Application of proper orthogonal decomposition to the discrete euler equations, International Journal for Numerical Methods in Engineering 55 (2002) 479–497.

[18] D. J. Lucia, Reduced-order modeling for high speed flows with moving shocks, Ph.D. Thesis, Air Force Institute of Techonology, School of Engineering and Management, Wright-Patterson Air Force Base, OH, 2001.

[19] G. Matheron, J. Serra, The birth of mathematical morphology, in: H. Talbot,

R. Beare (Eds.), Proceedings of the 6th International Symposium ISMM, Sydney, Australia, 2002, pp. 1–16.

[20] P. Soille, Morphological Image Analysis: Principles and Applications, Springer-Verlag, Berlin, 2nd edition, 2003.

[21] H. J. A. M. Heijmans, Mathematical morphology: a modern approach to image processing based on algebra and geometry, SIAM Review 37 (1995) 2337–2354.

[22] P. Maragos, R. W. Schafer, Morphological systems for multidimensional signal processing, Proceedings of the IEEE 78 (1990) 690–710.

[23] F. Meyer, Mathematical morphology: from two dimensions to three dimensions, Journal of Microscopy 165 (1992) 5–28.

[24] J. Serra, Image Analysis and Mathematical Morphology, Academic, London, 1982.

[25] E. R. Dougherty, R. Lotufo, Hands on Morphological Image Processing, SPIE-The International Society for Optical Engineering, Bellingham, WA, 2003.

[26] J. S. J. Lee, R. Haralick, L. Shapiro, Morphological edge detection, IEEE J. of Robotics and Automation RA-3 (1987) 142–156.

[27] B. Chanda, M. K. Kundu, Y. V. Padmaja, A multi-scale morphologic edge detector, Pattern Recognition 31 (1998) 1469–1478.

[28] M. Kraft, K. Andrezej, Morphological edge detection algoirthm and its hardware implementation, Computer Recognition Systems 2 (2007) 132–139.

[29] A. N. Evans, X. U. Liu, A morphological gradient approach to color edge detection, IEEE Transactions on Image Processing 15 (2006) 1454–1463.

[30] P. Holmes, J. L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, Cambridge, UK, 1996.

[31] L. V. Kantorovich, V. I. Krylov, Approximate Methods of Higher Analysis, Interscience Publishers, Inc., New York, 1964.

[32] D. Gidaspow, Multiphase Flow and Fluidization, Academic Press, Boston, MA, 1994.

[33] D. Gottlieb, C. W. Shu, On the gibbs phenomenon and its resolution, SIAM Review 39 (1997) 644–668.

[34] M. Syamlal, MFIX Documentation Numerical Technique, Technical Report DE-AC21-95MC31346, EG&G Technical Services of West Virginia, Morgantown, WV, 1998.

[35] T. A. Brenner, R. L. Fontenot, P. G. A. Cizmas, T. J. O'Brien, R. W. Breault, Augmented proper orthogonal decomposition for problems with moving discontinuities, Powder Technology 203 (2010) 78–85.

[36] H. J. A. M. Heijmans, C. Ronse, The algebraic basis of mathematical morphology i. dilations and erosions, Computer Vision, Graphics, and Image Processing 50 (1990) 245–295.

[37] R. Lerallut, E. Deceniére, F. Meyer, Image filtering using morphological amoebas, Image And Vision Computing 25 (2007) 395–404.

[38] R. H. MacNeal, Finite Elements: Their Design and Performance, Marcel Dekker, New York, 1993.

[39] E. B. Becker, G. F. Carey, J. T. Oden, Finite Elements: An Introduction, Volume 1, University of Texas at Austin, Austin, TX, 1981.

[40] J. Blazek, Computational Fluid Dynamics: Principles and Applications, Elsevier, Amsterdam, 2001.

[41] F. R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA Journal 32 (1994) 1598–1605.

[42] J. I. Gargoloff, A Numerical Method for Fully Nonlinear Aeroelastic Analysis, Ph.D. Thesis, Texas A&M University, College Station, Texas, 2007.

[43] M. Syamlal, W. Rogers, T. J. O'Brien, MFIX Documentation Theory Guide,

Technical Report DOE/METC-94/1004, DOE/METC, 1994.

[44] S. Benyahia, M. Syamlal, T. J. O'Brien, Summary of MFIX Equations 2005-4, NETL, 2007. http://www.mfix.org/documentation/MfixEquations2005-4-3.pdf.

[45] Z. Han, P. G. A. Cizmas, Prediction of axial thrust load in centrifugal compressors, International Journal of Turbo & Jet-Engines 20 (2003) 1–16.

[46] K. S. Kim, Three-dimensional hybrid grid generator and unstructured flow solver for compressors and turbines, Ph.D. Thesis, Texas A&M University, College Station, TX, 2003.

[47] T. Yuan, P. G. Cizmas, T. O'Brien, A reduced-order model for a bubbling fluidized bed based on proper orthogonal decomposition, Computers and Chemical Engineering 30 (2005) 243–259.

[48] H. Tijdeman, R. Seebass, Transonic flow past oscillating airfoils, Annual Review Fluid Mechanics 12 (1980) 181–222.

[49] H. Tijdeman, Investigation of the Transonic Flow Around Oscillating Airfoils, Technical Report NLR TR-77-090U, National Aerospace Laboratory, Amsterdam, The Nederlands, 1977.

[50] D. G. Mabey, Oscillatory flow from shock induced separation on biconvex aerofoils of varying thickness in ventilated wind tunnels, Technical Report, Advisory Group for Aerospace Research and Development CP 296, paper 11, 1980.

[51] S. R. Mohan, Periodic flows on rigid aerofoils at transonic speeds, in: 29th Aerospace Sciences Meeting, AIAA Paper 91-0598, Reno, NV, 1991.

[52] J. Gibb, The cause and cure of periodic flows at transonic speeds, in: Proceedings 16th Congress of ICAS, Jerusalem, Israel, 1988, pp. 122–130.

[53] G. Srinivasan, Acoustics and unsteady flow of telescope cavity in an airplane, Journal of Aircraft 37 (2000) 274–281.

[54] B. Henderson, Automobile noise involving feedback-sound generation by low

speed cavity flows, in: Third Computational Aeroacoustics Workshop on Benchmarks, NASA/CP-2000-209790, 2000, pp. 95–100.

[55] D. Liliedahl, P. Cizmas, Prediction of fluid instabilities in hole-pattern stator seals, in: 47th AIAA Aerospace Sciences Meeting, AIAA Paper 2009-0786, Orlando, FL, 2009.

[56] L. A. Villasmil, D. W. Childs, H. C. Chen, Understanding friction factor behavior in liquid annular seals with deliberately roughened surfaces, AIAA Paper 2003-5070, 2003.

APPENDIX A

CONSTITUTIVE MODELS

Gas-solids drag

The gas-solid drag force F_{gs} implemented in MFIX is attributed to Syamlal and O'Brien, which is related to the terminal velocity v_{rs} of the solid particles:

$$F_{gs} = \frac{3\varepsilon_s \varepsilon_g \rho_g}{4v_{rs}^2 d_{ps}} C_{Ds}\left(\frac{Re_s}{v_{rs}}\right)|\vec{v}_s - \vec{v}_g|,$$

where the terminal velocity v_{rs} is defined as:

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

where

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

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

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

C_{Ds}, the solid particle drag coefficient, is given as:

$$C_{Ds}\left(\frac{Re_s}{v_{rs}}\right) = \left(0.63 + 4.8\sqrt{\frac{v_{rs}}{\left(\frac{Re_s}{v_{rs}}\right)}}\right)^2.$$

Gas phase stress tensor

The gas phase stress tensor is defined as:

$$\bar{\bar{S}}_g = -P_g \bar{\bar{I}} + \bar{\bar{\tau}}_g,$$

where P_g is the gas phase pressure and $\bar{\bar{\tau}}_g$ is the viscous stress tensor. $\bar{\bar{\tau}}_g$ takes 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 and $\lambda_g = -2/3\mu_g$ is the second coefficient of the gas phase viscosity. $\bar{\bar{I}}$ is an identity tensor. $\bar{\bar{D}}_g$ is the gas phase strain rate tensor, defined as:

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

Solid phase stress tensor

The solid phase experiences either plastic or viscous stresses. Which stresses the particles undergo is dependent on the packing fraction, as defined by:

$$\bar{\bar{\tau}}_s = \begin{cases} -P_s^{\mathcal{P}}\bar{\bar{I}} + \bar{\bar{\tau}}_s^{\mathcal{P}} & \text{if } \varepsilon_g \leq \varepsilon_g^*\text{: Plastic Regime} \\ -P_s^{\mathcal{V}}\bar{\bar{I}} + \bar{\bar{\tau}}_s^{\mathcal{V}} & \text{if } \varepsilon_g > \varepsilon_g^*\text{: Viscous Regime} \end{cases},$$

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

- *Plastic Regime*: The plastic solid phase pressure is defined as:

$$P_s^{\mathcal{P}} = \varepsilon_s P^*$$

where P^* is represented by the power law:

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

The solid phase plastic stress tensor, $\bar{\bar{\tau}}_s$, is defined as:

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

where the "plastic" viscosity $\mu_s^{\mathcal{P}}$ is given as:

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

$\bar{\bar{D}}_s$ denotes the solid phase strain rate tensor andϕ 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*: The solid phase viscous pressure is defined as:

$$p_s^{\mathcal{V}} = K_1 \varepsilon_s^2 \Theta$$

where Θ is the granular energy. K_1 is the first granular stress constant defined as:

$$K_1 = 2(1 + e_s)\rho_s g_0,$$

where e_s is the coefficient of restitution and g_0 is the radial distribution function given by:

$$g_0 = \frac{1}{\varepsilon_g} + 1.5\frac{\varepsilon_s}{\varepsilon_g^2} + 0.5\frac{\varepsilon_s^2}{\varepsilon_g^3}.$$

The granular stress is defined as:

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

where $\bar{\bar{D}}_s$ is the strain rate tensor, and $\lambda_s^{\mathcal{V}}$ is the second coefficient of viscosity for the solids phase:

$$\lambda_s^{\mathcal{V}} = K_{2s}\varepsilon_s\sqrt{\Theta}.$$

K_2 is the second granular stress constant given as:

$$K_2 = \frac{4d_{ps}\rho_s(1+e_s)\varepsilon_s g_0}{3\sqrt{\pi}} - \frac{2}{3}K_3$$

where d_{ps} is the solid particle diameter, and K_3 is the third granular stress constant

$$K_3 = \frac{d_{ps}\rho_s\sqrt{\pi}}{6(3-e_s)}\left[0.5(3e_s+1)+0.4(1+e_s)(3e_s-1)e_s g_0\right] + \frac{d_{ps}\rho_s 8\varepsilon_s g_0(1+e_s)}{10\sqrt{\pi}}.$$

The solid phase “viscous”, or shear, viscosity is given as:

$$\mu_s^{\mathcal{V}} = K_3\varepsilon_s\sqrt{\Theta}.$$

For simulations where the granular energy equation 4.26 is not solved, an algebraic granular energy equation is solved. It is given by:

$$\Theta = \left\{\frac{-K_1\varepsilon_s\mathrm{tr}(\bar{\bar{D}}_s) + \sqrt{K_1^2\mathrm{tr}^2(\bar{\bar{D}}_s) + 4K_4\varepsilon_s\left[K_2\mathrm{tr}^2(\bar{\bar{D}}_s) + 2K_3\mathrm{tr}(\bar{\bar{D}}_s^2)\right]}}{2\varepsilon_s K_4}\right\}^2,$$

where K_4 is the fourth granular stress constant:

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

Gas-solids heat transfer

The gas-solids heat transfer takes the typical difference form:

$$H_{gs} = -\gamma_{gs}(T_s - T_g),$$

where γ_{gs} is the gas-solids heat transfer coefficient. MFIX utilizes the following equation to compute the gas-solids heat transfer coefficient:

$$\gamma_{gs} = \frac{C_{pg} R_{gs}}{exp\left(\frac{C_{pg} R_{gs}}{\gamma_{gs}^0}\right) - 1}.$$

R_{gs} is the rate of transfer of mass from the solid to the gas phase. γ_{gs}^0 is the gas-solids heat transfer coefficient for no interphase mass transfer. It is related to the Nusselt number (Nu_s) by:

$$\gamma_{gs}^0 = \frac{6 k_g \varepsilon_s Nu_s}{d_{ps}^2}$$

where the particle Nusselt number is given by

$$Nu_s = (7 - 10\varepsilon_g + 5\varepsilon_g^2)(1 + 0.7 Re_s^{0.2} Pr^{1/3}) + (1.33 - 2.4\varepsilon_g + 1.2\varepsilon_g^2) Re_s^{0.7} Pr^{1/3}.$$

Pr is the Prandtl number defined as:

$$Pr = \frac{C_{pg} \mu_g}{k_g}.$$

Gas and solids conduction

MFIX defines the gas phase conduction as:

$$\vec{q_g} = -\varepsilon_g k_g \bigtriangledown T_g.$$

MFIX also defines the solid phase conduction as:

$$\vec{q_s} = -\varepsilon_s k_s \bigtriangledown T_s.$$

APPENDIX B

EXAMPLE OF EDGE DETECTION ALGORITHM

This appendix presents the Morphological Edge Detection algorithm using a one-dimensional signal. The example signal is presented below. The first step in the

SIGNAL: $0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0$

algorithm is to apply blurring to the signal using (3.6). L_{rs} is set to 3. This is shown below. Once blurring has been applied, the blurred signal is shifted to the left and

BLUR: $0 \quad 0 \quad \frac{1}{3} \quad \frac{2}{3} \quad 1 \quad 1 \quad 1 \quad 1 \quad \frac{2}{3} \quad \frac{1}{3} \quad 0 \quad 0$

to the right. The left shift is seen below.

LEFT SHIFT: $0 \quad \frac{1}{3} \quad \frac{2}{3} \quad 1 \quad 1 \quad 1 \quad 1 \quad \frac{2}{3} \quad \frac{1}{3} \quad 0 \quad 0 \quad 0$

The right shift is given below.

$$\text{RIGHT SHIFT:}\quad 0 \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{2}{3} \quad 1 \quad 1 \quad 1 \quad 1 \quad \tfrac{2}{3} \quad \tfrac{1}{3} \quad 0$$

Erosion and dilation are applied next to the signals using (3.5) and (3.4). The eroded signal is given below.

$$\text{EROSION:}\quad 0 \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{2}{3} \quad 1 \quad 1 \quad \tfrac{2}{3} \quad \tfrac{1}{3} \quad 0 \quad 0 \quad 0$$

The dilated signal is seen below.

$$\text{DILATION:}\quad 0 \quad \tfrac{1}{3} \quad \tfrac{2}{3} \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad \tfrac{2}{3} \quad \tfrac{1}{3} \quad 0$$

The combined difference operators are applied next with (3.7) and (3.8). The blur/eroded signal is given below.

$$\text{BLUR/ERODED:} \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad 0 \quad 0$$

The dilated/blur signal is seen below.

$$\text{DILATED/BLUR:} \quad 0 \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad 0 \quad 0 \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad 0$$

The signal's edges are indicated by taking the minimum of the blur/eroded and dilate/blurred signal using (3.11).

$$\text{INDICATED:} \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad 0 \quad 0 \quad 0 \quad 0 \quad \tfrac{1}{3} \quad \tfrac{1}{3} \quad 0 \quad 0$$

The edges present in the signal are exactly at the discontinuity present in the original signal. The edges also fit the edge width requirement.

APPENDIX C

INPUT FILE FOR ODETI

```
# ODEx input file for Case I
#
# 1.1 Run control section
#
  TSTART  = 0.20256
  TSTOP = 1.0
  DT    = 1.234567D-4
  MAX_NIT = 30
  DT_MAX  = 1.D0
  DT_MIN  = 1.D-6
  DT_FAC  = 0.9D0
  TOL_RESID = 1.D-4
  TOL_DIVERGE = 1.D+2
  SOLVE_ENERGY = .TRUE.
#
# 1.2 Geometry and discretization section
#
  DO_K = .FALSE.
#
  XLENGTH = 25.4D0     IMAX = 108
  YLENGTH = 76.5D0     JMAX = 124
  DISCRETIZE = 2
#
# 1.3 physical propertiessection
#
  MU_g0   = 1.8D-4
  K_g0    = 1.0E-10
  K_s0    = 1.505E-4
  MW_g0   = 29.D0
  T_g0    = 297.D0
  RO_s0   = 1.0
  C_PG0   = 0.25
  C_PS0   = 0.310713
  GAMA_RG0 = 0.0
  GAMA_RS0 = 0.0
  D_p     = 0.05
  C_e     = 0.8
  Phi     = 30.0
  EP_star = 0.4
#
# 1.4 POD
#
  nP_g    = 2
  nU_g    = 5
  nV_g    = 5
  nU_s    = 8
  nV_s    = 6
  nEP_g   = 7
  nT_g    = 12
  nT_s    = 3
```

APPENDIX D

SAMPLE INPUT FILE FOR ODEG

```
# ODEx input file for Case II
#
# 1.1 Run control section
#
  TSTART  =  0.202579076483097
  TSTOP = 1.0
  DT    = 1.234567D-4
  MAX_NIT = 30
  DT_MAX  = 1.D0
  DT_MIN  = 1.D-6
  DT_FAC  = 0.9D0
  TOL_RESID = 1.D-4
  TOL_DIVERGE = 1.D2
  SOLVE_ENERGY = .FALSE.
  GRANULAR_ENERGY = .TRUE.
#
# 1.2 Geometry and discretization section
#
  DO_K = .FALSE.
#
  XLENGTH = 25.4D0     IMAX = 108
  YLENGTH = 76.5D0     JMAX = 124
  DISCRETIZE = 2
#
# 1.3 physical propertiessection
#
  MU_g0   = 1.8D-4
  MW_g0   = 29.D0
  T_g0    = 297.D0
  RO_s0   = 1.0
  C_PG0   = 0.25
  GAMA_RG0 = 0.0
  GAMA_RS0 = 0.0
  D_p     = 0.05
  C_e     = 0.8
  Phi     = 30.0
  EP_star = 0.4
#
# 1.4 POD
#
  nP_g    = 2
  nU_g    = 2
  nV_g    = 6
  nU_s    = 9
  nV_s    = 9
  nEP_g   = 7
  nTheta_m = 3
```

APPENDIX E

INPUT FILE FOR ODEV

```
# ODEx input file for Case III
#
#
# 1.1 Run control section
#
  TSTART  = 0.20256
  TSTOP = 1.0
  DT    = 1.234567D-4
  MAX_NIT = 30
  DT_MAX  = 1.D0
  DT_MIN  = 1.D-6
  DT_FAC  = 0.9D0
  TOL_RESID = 1.D-4
  TOL_DIVERGE = 1.D+2
  SOLVE_ENERGY = .FALSE.
  GRANULAR_ENERGY = .FALSE.
#
# 1.2 Geometry and discretization section
#
  DO_K = .TRUE.
#
  XLENGTH = 25.4D0     IMAX = 108
  YLENGTH = 76.5D0     JMAX = 124
  ZLENGTH = 1.646D0    KMAX = 7
  DISCRETIZE = 2
#
# 1.3 physical propertiessection
#
  MU_g0   = 1.8D-4
  K_g0    = 1.0E-10
  K_s0    = 1.505E-4
  MW_g0   = 29.D0
  T_g0    = 297.D0
  RO_s0   = 1.0
  C_PG0   = 0.25
  C_PS0   = 0.310713
  GAMA_RG0 = 0.0
  GAMA_RS0 = 0.0
  D_p      = 0.05
  C_e      = 0.8
  Phi      = 30.0
  EP_star = 0.4
#
# 1.4 POD
#
  nP_g     = 2
  nU_g     = 2
  nV_g     = 6
  nU_s     = 9
  nV_s     = 9
  nW_g    = 2
  nW_s    = 9
  nEP_g    = 7
```

APPENDIX F

INPUT FILE FOR MEDTS

```
# MEDts Input File: Bubbling Bed
#
&card1
imax = 108    !Bed width
jmax = 126    !Bed height
nss = 320     !Number of snapshots
vss = 320     !Snapshot of interest
plv = 0.1     !Lower thresholding value
puv = 0.2     !Upper thresholding value
/
```

APPENDIX G

INPUT FILE FOR MEDUNS: TRANSONIC AIRFOIL

```
# MEDuns Input File: Transonic Airfoil Case
#
&card1
nss = 100                    ! Number of input files
plv = 0.0049                 ! Lower thresholding value
puv = 0.03                   ! Upper thresholding value
vmf1 = 'Bicon_fine.mesh'     ! Mesh file
c2nf1 = 'c2n.def'            ! Connectivity file
ef = 'bb1000.plt'            ! Last input file name
af = 'bb'                    ! Input file preamble
ff = 'edge_bb40t'            ! Output file preamble
lmtd = .true.                ! Structuring element limiter
dmm = 0.005D0                ! Structuring element maximum size
athd = .true.                ! Thresholding flag
ftol = 1D-11                 ! Tolerance for face location
p_check = .false.            ! Pointer check
klim = .true.                ! Single k-layer flag
lso = .true.                 ! Logical for structuring element type
/
```

APPENDIX H

INPUT FILE FOR MEDUNS: TURBOMACHINE SEAL

```
# MEDuns Input File: Turbomachine Seal
#
&card1
nss = 80                    ! Number of input files
plv = 1.95                  ! Lower thresholding value
puv = 10.0                  ! Upper thresholding value
vmf1 = 'vol.mesh'           ! Mesh file
c2nf1 = 'c2n.def'           ! Connectivity file
ef = 'bb0220.plt'           ! Last input file name
af = 'bb'                   ! Input file preamble
ff = 'edge_uvt'             ! Output file preamble
lmtd = .false.              ! Structuring element limiter
mtd = .true.                ! Stucuturing element maximizer
dmm = 1.77D-5               ! Structuring element maximum size
athd = .true.               ! Thresholding flag
ftol = 1D-14                ! Tolerance for face location
ntol = 1D-10                ! Tolerance for MNR
rv = 7                      ! Input variable morphology is applied over
p_check = .false.           ! Pointer check
klim = .true.               ! Single k-layer flag
lso = .true.                ! Logical for structuring element type
/
```

VITA

Raymond Lee Fontenot was born in Sulphur, Louisiana. He received his Bachelor of Science degree in mechanical engineering from McNeese State University, Lake Charles, Louisiana in May 2008.

He began his graduate study at Texas A&M University in July 2008 and received his Master of Science degree in aerospace engineering in December 2010. His research interests focused on reduced-order modeling of transport phenomena in two-phase flows and morphologocial feature detection algorithms used to extract features in single and two-phase flows.

Raymond Lee Fontenot's email address is rlf1582@aeromail.tamu.edu. He can be reached at:

Department of Aerospace Engineering

c/o Dr. Paul Cizmas

Texas A&M University

H.R. Bright Building, Ross Street - TAMU 3141

College Station, TX 77843-3141

The typist for this thesis was Raymond Fontenot.