# Reduced Order Geomechanics Models

by

Saeed Hatefi Ardakani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Civil Engineering

Waterloo, Ontario, Canada, 2024

© Saeed Hatefi Ardakani 2024

**Examining Committee Membership**

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

| | |
|---|---|
| External Examiner: | Dr. Masayuki Yano<br>Associate Professor, Institute for Aerospace Studies,<br>University of Toronto |
| Supervisor: | Dr. Robert Gracie<br>Associate Professor, Dept. of Civil and Environmental Engineering,<br>University of Waterloo |
| Internal Member: | Dr. Chul Min Yeum<br>Associate Professor, Dept. of Civil and Environmental Engineering,<br>University of Waterloo |
| Internal Member: | Dr. Shunde Yin<br>Professor, Dept. of Civil and Environmental Engineering,<br>University of Waterloo |
| Internal-External Member: | Dr. Maurice B. Dusseault<br>Professor, Dept. of Earth and Environmental Sciences,<br>University of Waterloo |

## Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Statement of Contributions

This thesis consists of three manuscripts written for publication. Exceptions to sole authorship of material are as follows:

Chapters 1, 2, and 5 are based on the following journal article:

S. Hatefi Ardakani, G. Zingaro, M. Komijani, and R. Gracie, "An efficient reduced order model for nonlinear transient porous media flow with time-varying injection rates," *Finite Elements in Analysis and Design*, vol. 241, p. 104237, 2024.

In this journal article, I was the primary author and was responsible for writing the article. I developed the methodology, conceptualized the verification case, performed the formal analysis of the developed model, implemented the computer code for the full order and reduced order models, and ran the simulations. Dr. Robert Gracie was responsible for editing the paper, conceptualization of the idea, and investigation of the results. Dr. Mohammad Komijani was responsible for editing the paper and conceptualization of the idea. Giovanni Zingaro was responsible for editing the paper and investigating the results.

Chapters 1, 3, and 5 are based on the following journal article:

S. Hatefi Ardakani, G. Zingaro, and R. Gracie, "An efficient parameterized local reduced order model for Drucker–Prager plasticity," *Submitted to Computational Mechanics*, September 2024.

In this journal article, I was the primary author and was responsible for writing the article. I developed the methodology, conceptualized the verification case, performed the formal analysis, implemented the plasticity code for the full order and reduced order models, and ran the simulations. Dr. Robert Gracie was responsible for editing the paper, conceptualization, and investigation of the results. Giovanni Zingaro was responsible for editing the paper and investigating the results.

Chapters 1, 4, and 5 are based on the following journal article:

S. Hatefi Ardakani and R. Gracie, "A parameterized local reduced order model of stimulated volume evolution in reservoirs," *Submitted to International Journal for Numerical and Analytical Methods in Geomechanics*, November 2024.

In this journal article, I was the primary author and was responsible for writing the article. I developed the methodology, conceptualized the verification case, performed the formal analysis, implemented the poro-elasto-plastic code for the full order and reduced order models, and ran the simulations. Dr. Robert Gracie was responsible for editing the paper, conceptualization, and investigation of the results.

## Abstract

Computational techniques are commonly used for real-time simulation of complex geomechanics problems, such as hydraulic dilation stimulation. A significant challenge in this realm is that high-fidelity mathematical models or full order models (FOMs) are computationally expensive as they must span multiple spatial and temporal length scales, often including nonlinearities and thermo-hydro-mechanical processing. The computationally intensive nature of these simulations continues to pose challenges in parameter estimation, uncertainty quantification, and optimization applications, where hundreds to thousands of simulations are required to achieve a solution. Intrusive reduced order models (ROMs) have emerged as a method to derive and train a computationally efficient surrogate/proxy model using the FOM. This thesis seeks to bridge the gap in existing intrusive ROMs in reservoir engineering by introducing efficient ROMs that are capable of capturing hydro-mechanical coupling behavior and path-dependent plastic deformation of rocks. A complex case involving hydraulic dilation stimulation is used to show the efficiency and accuracy of the ROM in addressing coupling, plasticity, and permeability enhancement features.

First, an efficient and accurate ROM is proposed for nonlinear porous media flow problems, with specific application to a two-dimensional layered reservoir with a two-well system. Standard projection-based intrusive ROMs without hyper-reduction, such as proper orthogonal decomposition-Galerkin (POD-Galerkin), have not demonstrated efficacy in reducing the computational cost of the ROM for nonlinear problems. In this context, we combine POD-Galerkin with discrete empirical interpolation method (DEIM) as a hyper-reduction technique to reduce the size of the system of equations and accelerate the computation of nonlinear terms (residual force vector and its Jacobian). Column-reduced Jacobian DEIM technique is employed to interpolate the Jacobian, leading to a significant reduction in the computational time of the online stage. The ROM is parameterized for the nonlinear transient injection rate (pumping schedule). Offline, training data is generated by the FOM runs with simple constant injection rates. Online, the ROM demonstrates high accuracy and efficiency for complex and time-varying pumping schedules, including sinusoidal, high-frequency, and time-discontinuous pumping schedules that are located outside of the training regime. It is shown that POD-DEIM ROM has about $10^3$ times fewer degrees of freedom (DoFs) and is approximately 190 times faster than the FOM for a reservoir model with $3 \times 10^4$ DoFs, while maintaining an accurate solution in the online stage. The accuracy and efficiency of the POD-DEIM motivate its potential use as a surrogate model in the real-time control and monitoring of fluid injection processes.

Intrusive ROMs have faced considerable difficulties in accurately capturing the history-dependent nonlinear evolution of plastic strain. In the second objective, an intrusive ROM is

developed and evaluated for a Drucker–Prager plasticity model, in which material properties and cyclic load path are parametric inputs. By constructing multiple local DEIM (LDEIM) approximations in combination with clustering and classifier techniques, a fast and accurate ROM is achieved. The FOM consists of a two-dimensional finite element analysis (FEA) of a deformable solid with Drucker–Prager plasticity. Offline, the temporal and parameterized training data generated from FOM runs are classified using the $k$-means clustering algorithm, whereby LDEIM basis vectors are computed. Online, a nearest neighbor classifier identifies the appropriate LDEIM. The ROM has three hyper-parameters (the size of the ROM, the number of clusters, and the number of DEIM measurement points per cluster), influencing both accuracy and speed-up. In a micromechanics porous media problem, parameterized by Young's modulus and hardening modulus, the ROM's performance is demonstrated for inputs within and outside of the training domain; error and speed-up vary with inputs - accuracy is highest for inputs within the training domain (Error: 1.0–3.5% vs 1.0–9.2%), while speed-up varies from 106 to 134 times. For a cyclic plasticity problem, parameterized by load path, the ROM exhibits stable and accurate online performance with a substantial speed-up for test load paths. Under FOMs with $\sim 10^3$ and $\sim 5 \times 10^4$ DoFs, speed-ups are 11 and 770 times, respectively. Larger speed-ups seem likely for larger FOMs.

Finally, the ROM for nonlinear transient porous media flow as a diffusion problem is coupled with the ROM for plasticity to develop a novel ROM formulation for poroplasticity problems. This ROM aims to significantly reduce the computational costs for nonlinear and fully-coupled hydro-mechanical simulations in large-scale reservoirs. The developed mathematical model integrates a coupled system of equations from a two-dimensional FEA of momentum and mass balance equations equipped with Drucker–Prager plasticity and stress-dependent permeability enhancement models. The proposed ROM combines various ROMs, including POD-Galerkin to reduce the number of DoFs, DEIM to accelerate the computation of nonlinear terms, and local POD and local DEIM (LPOD/LDEIM) for further reductions in poroplasticity problems. LPOD and LDEIM classify the parameterized training data, obtained from offline FOM runs, into multiple subspaces with similar dynamic features. A new strategy for clustering and classification techniques tailored for the coupled formulation framework is introduced. The advantages of this ROM are demonstrated in a large-scale application involving hydraulic dilation stimulation of a reservoir with a horizontal well pair. The ROM is parameterized not only by the material properties but also by the injection rate. Its effectiveness is evaluated for more realistic use cases, where ROM remains efficient for injection rates that extend beyond the training data.

In large-scale subsurface flow modeling of hydraulic dilation stimulation, a speed-up of $\sim 400$ times is achieved, with a ROM reducing the model dimension from $10^5$ DoFs to 100 DoFs. This substantial computational saving results from a real-time analysis of the ROM

and becomes even more highlighted in multi-query problems, where the model must be executed for multiple inputs and system configurations. This ROM has a high potential for accelerating various problems, such as uncertainty quantification, design, history matching, and well control optimization. It is also recommended that the proposed ROM be adopted for other real-world subsurface applications, including conventional and unconventional oil and gas production, hydraulic fracturing, and carbon storage.

**Acknowledgments**

I would like to thank my supervisor, Dr. Robert Gracie, for his guidance and continuous support throughout my time as his student. I would also like to thank the committee members for reviewing this thesis providing thoughts and feedback.

I must express my gratitude to my wife, Elaheh, for her eternal love and support. Without her, nothing is possible in my life.

## Dedication

To my better half, Elaheh, and my family for all their love and sacrifices.

# Table of Contents

**Examining Committee** ii

**Author's Declaration** iii

**Statement of Contributions** iv

**Abstract** v

**Acknowledgments** viii

**Dedication** ix

**List of Figures** xiii

**List of Tables** xvii

**1 Introduction** 1

1.1 Complexity of Large-scale Reservoir Simulation . . . . . 1

1.2 Reduced Order Models . . . . . 3

1.3 Intrusive Reduced Order Models . . . . . 4

1.4 Purpose and Objectives . . . . . 8

1.4.1 Objective 1: Develop an efficient intrusive ROM for nonlinear transient flow in porous media as a diffusion problem . . . . . 9

1.4.2 Objective 2: Develop a parameterized ROM for cyclic plasticity models (Drucker–Prager model) . . . 9
1.4.3 Objective 3: Develop a robust ROM for poroplasticity models with an application in hydraulic dilation stimulation . . . 10

**2 Intrusive ROM for Nonlinear Transient Porous Media Flow 11**
2.1 Introduction . . . 11
2.2 Mathematical Model . . . 13
2.3 Intrusive Model Order Reduction . . . 17
2.3.1 POD-Galerkin: Proper Orthogonal Decomposition-Galerkin Projection 17
2.3.2 DEIM: Discrete Empirical Interpolation Method . . . 21
2.4 Results and discussion . . . 24
2.4.1 Comparison between FOM, POD-Galerkin and POD-DEIM . . . 26
2.4.2 Sensitivity Analysis for Parametric ROM . . . 34
2.5 Conclusions . . . 40

**3 Parameterized ROM for Drucker–Prager Plasticity 41**
3.1 Introduction . . . 41
3.2 Full Order Model . . . 42
3.2.1 Drucker–Prager Elastoplasticity Model . . . 43
3.2.2 Newton–Raphson Solution to the Nonlinear System of Equations . . 44
3.2.3 Linear-Nonlinear Decomposition . . . 45
3.3 Intrusive Reduced Order Model . . . 46
3.3.1 Proper Orthogonal Decomposition-Galerkin Projection . . . 46
3.3.2 Discrete Empirical Interpolation Method . . . 47
3.3.3 Localized Discrete Empirical Interpolation Method . . . 49
3.3.4 Offline Stage . . . 53
3.4 Results and discussion . . . 58
3.4.1 FOM Verification . . . 58

3.4.2 Micromechanics of Porous Media . . . 60
3.4.3 Cyclic Plastic Loading . . . 69
3.5 Conclusions . . . 74

**4 Parameterized ROM of Stimulated Volume Evolution in Large-scale Reservoirs 76**
4.1 Mathematical model . . . 77
4.1.1 Poro-Elasto-Plastic Model . . . 78
4.1.2 Coupled Full Order Model . . . 80
4.2 Intrusive Local Reduced Order Model . . . 82
4.2.1 Proper Orthogonal Decomposition . . . 83
4.2.2 Discrete Empirical Interpolation Method . . . 84
4.2.3 Localized Discrete Empirical Interpolation Method . . . 85
4.2.4 Localized Proper Orthogonal Decomposition . . . 86
4.3 Results and discussion . . . 87
4.3.1 Example 1: Parameterized ROM with material inputs . . . 89
4.3.2 Example 2: Parameterized ROM with material inputs and injection rate . . . 98
4.4 Conclusions . . . 104

**5 Conclusions 105**
5.1 Concluding Remarks . . . 105
5.2 Research Contributions . . . 108
5.3 Directions for Future Study . . . 110

**References 113**

**APPENDICES 124**

**A Drucker–Prager Plasticity Constitutive Equations 125**

**B Finite Element Discretization for Poroplasticity Problem 127**

# List of Figures


1.1 a) Illustration of hydraulic dilation stimulation of SAGD well pair and b) typical mesh used in SAGD simulation, involving a large FEM mesh with a large number of DoFs. . . . . 2

2.1 Reservoir-well system consisting of two wells (well A with a time-varying injection rate Q($t$) and well B with a constant pressure P). . . . . 14

2.2 Coarse finite element mesh with 1423 DoFs for FE training process. . . . . 26

2.3 Fluid pressure overtime at two characteristic points using POD-DEIM and FOM. . . . . 27

2.4 Pressure field snapshots at t = 6.5 days, t = 34.3 days, t = 87 days, from top to bottom. Left column: FOM results and right column: POD-DEIM's relative error. . . . . 28

2.5 Mass flux snapshots at t = 6.5 days, t = 34.3 days, t = 87 days, from top to bottom. Left column: FOM results and right column: scaled difference between FOM and POD-DEIM results. . . . . 29

2.6 a) $L_2$ error versus CPU time and b) $H_1$ error versus CPU time for the various numerical techniques: POD-Galerkin, POD-DEIM and FOM. . . . . 32

2.7 Effect of the number of truncated modes ($k$) on the accuracy of the results. a) $L_2$ norm error and b) $H_1$ norm error. . . . . 33

2.8 Role of snapshot period in the training process on the accuracy of the results. a) $L_2$ norm error and b) $H_1$ norm error. . . . . 34

2.9 Online time-varying injection rates (test functions 1 to 4). . . . . 35

2.10 Pressure and mass flux over time for a) test function 1, b) test function 2, c) test function 3. . . . . 36

2.11 Comparison of the time history of the POD-DEIM and FOM for a highly variable injection rate. a) injection pressure and b) mass flux. . . . . 37

2.12 Rectangular waves with regular and irregular time sequences as online time-varying injection rates. a) load case 1, b) load case 2, c) load case 3, d) load case 4. . . . . 38

2.13 Pressure and mass flux overtime for different rectangular waves. a) results of rectangular wave 1, b) results of wave 2, c) results of wave 3, d) results of wave 4. . . . . 39

3.1 Workflow for the offline and online stages of the presented ROM. . . . . 54

3.2 $k$-folds cross-validation procedure. . . . . 56

3.3 (a) $k$-fold cross-validation of DEIM force vector for training dataset, (b) average testing error vs. measurement points for testing dataset ($k = 3$). . 57

3.4 Spatial distribution of DEIM measurement points when $q = 200$. . . . . 57

3.5 a) Two-dimensional slope stability problem, b) FEM mesh with 909 nodes and 282 Q8 elements. . . . . 59

3.6 a) Load factor vs. settlement diagram, b) increment of accumulated plastic strain at collapse. . . . . 60

3.7 a) Two-dimensional micromechanics of porous media, b) FEM mesh with 5209 nodes and 4992 Q4 elements, c) plastic strain bands. . . . . 61

3.8 a) Force-displacement diagram for the FOM, the ROM with DEIM and the ROM with LDEIM, b) energy of mode shapes in DEIM and LDEIM. . . . 62

3.9 Quality of nonlinear force vector and Jacobian approximations based on the number of measurement points, a) nonlinear force vector, b) Jacobian. . . . 63

3.10 Effect of the hyper-parameters $m$ and $q$ on the accuracy and the computational cost of the ROMs, a) hyper-parameter $q$ ($m = 60$), b) hyper-parameters $m$. 64

3.11 Global responses of the FOM for ten samples used for training purposes. . 65

3.12 Accuracy and speed-up analyses, a) relative $L_2$ error contour of LDEIM for inputs inside and outside of the training domain, b) ROM speed-up for inputs inside of the training domain. . . . . 67

3.13 Loss function of online phase at various load steps, b) load-displacement response when the ROM input is $E = 27.5$ MPa and $H = 475$ kPa. . . . . 67

3.14 Comparison of primary and secondary variables for the FOM and the ROM with LDEIM when the ROM input is $E = 27.5$ MPa and $H = 475$ kPa, a) displacements in $x$ and $y$ directions along the line S, b) von Mises stress, c) accumulated plastic strain. . . . . 68

3.15 (a) Two-dimensional plastic plate with a central hole, (b) quarter plate with a central hole due to symmetric, (c) FE mesh with 707 nodes and 649 Q4 elements. . . . . 70

3.16 Online loading-unloading paths, (a) load case 1, (b) load case 2, (c) load case 3, (d) load case 4. . . . . 71

3.17 Division of force vector into seven clusters. . . . . 72

3.18 Comparison of load-displacement responses for FOM and ROM with LDEIM using various load paths, (a) load case 1, (b) load case 2, (c) load case 3, (d) load case 4. . . . . 73

3.19 Plastic strain evolution for FOM and ROM with LDEIM at points A and B. 73

4.1 Offline/Online procedure of the presented LROM. . . . . 88

4.2 a) Two-dimensional hydraulic stimulation problem, b) comparison of the domain size to the size of the stimulated zone (accumulated plastic strain zone), c) FEM mesh with 32993 nodes and 32848 elements. . . . . 90

4.3 Ten samples non-uniformly distributed in the two-dimensional training domain. 91

4.4 Projection error for the displacement and pressure equations using POD and LPOD. . . . . 92

4.5 DEIM interpolation error for both the mechanical and hydraulic parts of the nonlinear force snapshots, a) mechanical part, b) hydraulic part. . . . . 93

4.6 Accuracy and efficiency analyses for inputs inside of the training domain, a) relative $L_2$ error contour of LROM for displacement, b) relative $L_2$ error contour of LROM for pressure, c) LROM speed-up. . . . . 95

4.7 Comparison of primary variables $u_x$, $u_y$, and $p$ for CFOM and LROM, a) displacement in $x$ direction $u_x$, b) displacement in $y$ direction $u_y$, c) pore pressure $p$. . . . . 96

4.8 Comparison of secondary variables $\bar{\varepsilon}^p$, $\kappa$, and $\phi$ for CFOM and LROM, a) $\bar{\varepsilon}^p$, b) $\kappa$, c) $\phi$. . . . . 97

4.9 Three-dimensional training domain, including two-dimensional material parameter space and one-dimensional injection rate space. . . . . 99

4.10 Top row: contours of the pressure field $p$ at $t = 2.5$ days, $t = 6$ days, and $t = 10$ days, from left to right. Bottom Row: contour of LROM relative error at $t = 2.5$ days, $t = 6$ days, and $t = 10$ days, from left to right. . . . . 100

4.11 Displacement field snapshots in the $x$ and $y$ directions ($u_x$ and $u_y$) at $t = 2.5$ days, $t = 6$ days, and $t = 10$ days, from left to right, a) $u_x$, b) $u_y$. In each subfigure, the top and bottom rows show CFOM results and LROM relative error, respectively. . . . . 102

4.12 The evolution of accumulated plastic strain for CFOM and LROM at point A located at the surface of the bottom well with coordinate $x = 100$ m and $y = 95.1$ m. . . . . 103

4.13 Porosity and permeability enhancement for CFOM and LROM at point A, a) Permeability, b) porosity. . . . . 103

5.1 Algorithm of history matching process using ROM. . . . . 112

# List of Tables


2.1 CPU time (second) needed from the FOM, POD-Galerkin, and POD-DEIM techniques for different training meshes . . . . . 30

2.2 Total number of Newton–Raphson iterations needed from the FOM, POD-Galerkin, and POD-DEIM techniques for different training meshes (note that the total number of time steps is 200) . . . . . 30

3.1 Material properties of slope stability problem . . . . . 59

3.2 Material properties for the generation of training data . . . . . 61

3.3 Samples used for training process . . . . . 65

3.4 CPU runtime in seconds for offline and online stages of the ROM with LDEIM. The CPU time for online phase is reported based on an average of 36 material inputs inside the training domain . . . . . 69

3.5 CPU time in seconds, ROM speed-up, and average number of Newton–Raphson iterations needed for the FOM and the ROM with LDEIM . . . . 74

4.1 Material properties for the two-dimensional injection wells problem . . . . 89

4.2 CPU runtime in hours for offline and online stages of LROM . . . . . 98

# Chapter 1

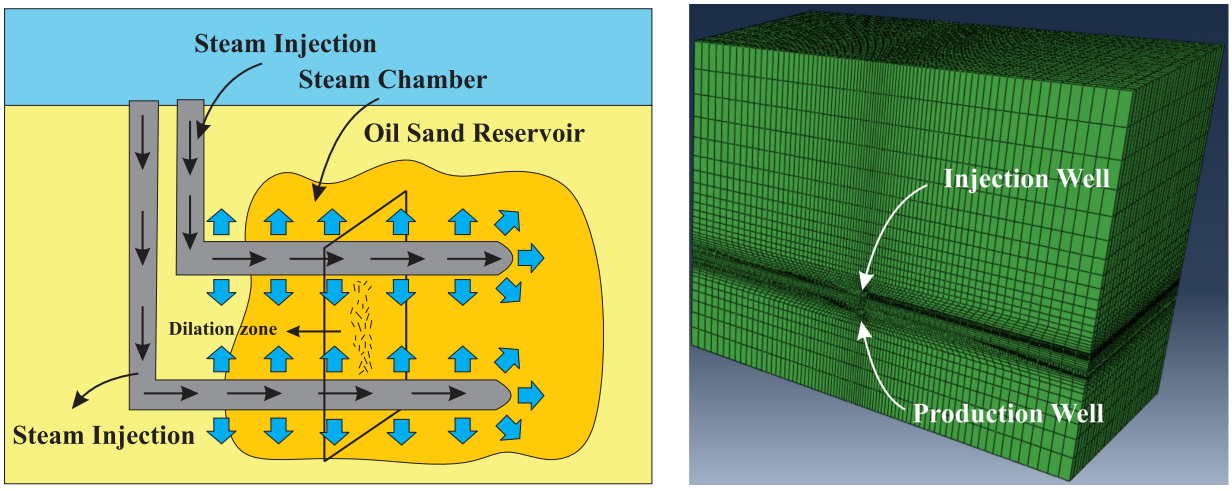
# Introduction

## 1.1 Complexity of Large-scale Reservoir Simulation

Reservoir engineers are interested in simulating the behavior of complex geomechanical problems under different conditions (temperature, pressure, and loading). Hydraulic dilation stimulation of heavy oil reservoirs is a complex process in reservoir engineering, which was introduced to enhance the productivity of steam-assisted gravity drainage (SAGD) [1]. Prior to commencing SAGD operations, by injecting high-pressure fluid into both injector and producer wells over a short period of time, the porosity and permeability of the vertical zone connecting the well pair increase. Consequently, less steam is needed during SAGD, the oil production rate increases, and steam circulation time decreases [2]. A schematic diagram of hydraulic dilation stimulation for SAGD well pair is illustrated in Figure 1.1a.

Computational techniques, such as finite element method (FEM), are frequently employed to numerically simulate the full physics SAGD dilation process [3, 4, 5]. In all studies, shear dilation and permeability enhancement were analyzed using thermo-hydro-mechanical (THM) modeling of plasticity couplings. Yin *et al.* presented a coupled thermo-poroplasticity model to simulate shear dilation and permeability enhancement in a multiphase oil sand reservoir [3]. They assessed the effect of dilation angle, initial lateral stress ratio, and stiffness of overburden rock on the magnitude of shear dilation and permeability enhancement. Xu and Wong proposed a fully coupled finite element model to simulate injection well testing in oil sand reservoirs by considering poroplastic constitutive relationships and strain-dependent permeability models [4]. Yang *et al.* simulated vertical-well-assisted SAGD dilation process (VW-SAGD-Dilation) where several vertical wells are utilized in the vicinity of SAGD well pair to enhance the interwell communication between SAGD well pair by making dilation zones [5].

One important challenge in the full physics models mentioned above is their very high computational cost. The runtime to complete a single simulation can easily take more than 100 hours. This high cost stems from the need to solve an extremely large nonlinear coupled system of equations at each time step. The governing equations are based on solid phase deformation, multiphase fluid flow, and heat transfer. Additionally, plasticity models are used to simulate the dilation process, and the large physical size of the domain ($10^2$–$10^3$ m) and the small size of the wells ($10^{-1}$ m) result in a large number of degrees of freedom (DoFs) in finite element (FE) simulations, as illustrated in Figure 1.1b. The computational cost issue is even more highlighted for (a) estimating parameters in inverse problems; or (b) history matching process, where the model must be executed hundreds to thousands of times to estimate the posterior distribution [4]. In this context, models that execute in minutes rather than requiring 10s to 100s of hours of computation are needed.




(a) (b)

Figure 1.1: a) Illustration of hydraulic dilation stimulation of SAGD well pair and b) typical mesh used in SAGD simulation, involving a large FEM mesh with a large number of DoFs.

In recent decades, much progress has been made to increase the computational efficiency of simulations through improved interpolation methodologies (e.g., the eXtended Finite Element method [6, 7], Meshfree method [8], and Isogeometric method [9, 10]), adaptive meshing techniques [11, 12], and high-performance computing [13, 14], to name a few. Confounding these efforts, the expectations and ambitions of modelers increase continuously. Current aspirations to apply multiphysics simulations in iterative design processes or real-time operational use motivate the development of more novel methodologies to reduce

computational costs. This thesis presents the derivation of such a novel method and evaluates its accuracy and efficiency against high-fidelity mathematical models. It is applied to large-scale industrial scenarios to demonstrate its potential use as a surrogate model for real-time control and monitoring of fluid injection processes.

## 1.2 Reduced Order Models

Model order reduction (MOR) techniques have been developed to reduce the computational cost of complex and large-scale simulations. The main idea of MOR is to compute/derive an approximation to the full order model (FOM), in which the ROM has significantly fewer DoFs than the FOM. The derived system with a small number of DoFs is called a reduced order model (ROM). In this thesis, we refer to the FOM as a discrete system of nonlinear equations derived using the FEM, which is common in geomechanics simulations. As an application, ROMs provide designers and operators with a rapid evaluation of the system instead of using the computationally expensive full order multiphysics models. MOR techniques have found applications in thermal analysis [15], aerodynamics [16, 17], structural dynamics [18], control [19], biomechanics [20], computational fluid dynamics (CFD) [21, 22], and reservoir simulations [23, 24].

ROMs are trained in a two-phase offline process. Phase 1 involves performing many full physics simulations to generate data for the training process. In phase 2, the ROM is trained. Although full physics simulations are considered computationally expensive, the data generation process is executed only once in the lifecycle of a ROM. The created ROM is utilized online to rapidly predict the response of the system. The accuracy of the ROM is less than that of the FOM; however, ideally it is in orders of magnitude more computationally efficient. In practice, the ROM designer must balance trade-offs between variables that increase ROM accuracy (e.g., more training data) and those that increase ROM efficiency (e.g., fewer ROM DoFs).

MOR techniques can be classified into two categories: intrusive [25, 26, 27, 28, 29, 30, 31, 32] and non-intrusive approaches [33, 34, 35, 36, 37, 38]. The algorithms of an intrusive ROM are directly integrated with that of the FOM. Knowledge of governing equations and modification of the FE simulator codes are required to develop an intrusive ROM. In contrast, non-intrusive (black box) MOR approaches train a surrogate (regression) model using FOM outputs without requiring modifications of the FE reservoir simulation code. Various types of surrogate models can be found in the literature, such as response surface methods (RSMs) [33, 34], Gaussian process regression [35, 36] and neural network architectures [37, 38].

Here, an important question is "Which MOR technique should be selected for large-scale reservoir simulation?". One consideration is how much access one has to the simulator's source code. Intrusive ROMs necessitate manipulating the FE source code or at least having a good interface to the source code, for instance, access to the stiffness matrices and residual forces. In cases where we are not able to modify the FE source code of the FOM, the best option for creating ROM is non-intrusive approach. For example, there are some limitations in accessing different parts of the FE subroutine in most commercial FE software. In this case, non-intrusive ROM is the only option. Another consideration is the offline and online computational costs. The offline computational cost includes the costs of data generation and training process, while the online cost involves the execution time of the trained ROM as a surrogate model. The offline computational cost of non-intrusive methods is very high compared to the intrusive methods due to the need for copious amounts of training data generated by computationally expensive high-fidelity models. However, the online computational cost of non-intrusive methods is often low, particularly when GPUs are used for computations. The huge computational cost of offline phase in non-intrusive methods might be considered acceptable due to very fast computations during the online phase. This trade-off is justifiable in cases where the ROM is used many times in the online phase. In our application, these high-fidelity models include plasticity and stress-dependent permeability models in a large-scale reservoir with many DoFs. Data generation and training non-intrusive ROMs can exceed the available computational budget. In contrast, intrusive ROMs require significantly smaller datasets as the underlying physics of the FOM is explicitly incorporated into the ROM. In this thesis, an intrusive ROM is selected as we desire a model that minimizes the computational cost associated with data generation in the offline stage.

## 1.3 Intrusive Reduced Order Models

The main focus of this thesis is on intrusive ROMs, whereby the governing equations are modified to derive a new system of equations with fewer DoFs. To illustrate the concept of an intrusive ROM within a simple reservoir context, a nonlinear and time-dependent hydro-mechanical coupled problem is considered, and the main MOR formulations are summarized. In the framework of FEM, the domain is discretized by a mesh with $ne$ finite elements, $n_u^{\text{nodes}}$ nodes for the displacement field, and $n_p^{\text{nodes}}$ nodes for the pressure field. The finite element interpolations are utilized to approximate the trial solution for the state variables (bulk displacement fields $\mathbf{u}^h(\mathbf{x}, t)$ and pore pressure $p^h(\mathbf{x}, t)$) in a two-dimensional system of coupled partial differential equations (PDEs) at point $\mathbf{x}$ and time $t$:

$$\mathbf{u}^h(\mathbf{x},t) = \mathbf{N}_u \mathbf{u}^t,\ \mathbf{x} \in \Omega,\ t \in [0,T] \tag{1.1}$$

$$p^h(\mathbf{x},t) = \mathbf{N}_p \mathbf{p}^t,\ \mathbf{x} \in \Omega,\ t \in [0,T] \tag{1.2}$$

in which $\mathbf{N}_u$ is a matrix of shape functions for displacement, $\mathbf{N}_p$ is a vector of shape functions for pressure, $\mathbf{u}^t$ is a vector of nodal displacements at time $t$, and $\mathbf{p}^t$ is a vector of nodal pressures at time $t$. FOM has $n_u = 2 \times n_u^{\text{nodes}}$ and $n_p = n_p^{\text{nodes}}$ DoFs for the displacement and pressure fields, respectively.

At time step $t$, a large nonlinear coupled system of equations must be solved for the nodal values of the unknown $\mathbf{d}^t = \begin{bmatrix} \mathbf{u}^t \\ \mathbf{p}^t \end{bmatrix}$. The solution of the nodal unknowns $\mathbf{d}^t$ at time step $t$, given the solution $\mathbf{d}^{t-1}$ at the previous time step $t-1$, leads to a large discrete coupled system of equations of the form

$$\mathbf{R}\left(\mathbf{d}^t\right) = \mathbf{0} \tag{1.3}$$

In FEM, the nonlinear system of equations (1.3) is solved iteratively by Newton–Raphson method, such that the increment, $\Delta\mathbf{d}^{t(j)}$, in the nodal values at time step $t$ and iteration $j$ is

$$\mathbf{K}^{t(j)} \Delta\mathbf{d}^{t(j)} = \mathbf{R}^{t(j)} \tag{1.4}$$

and the updated solution at time $t$ is obtained as

$$\mathbf{d}^{t(j+1)} = \mathbf{d}^{t(j)} + \Delta\mathbf{d}^{t(j)} \tag{1.5}$$

In (1.4), $\mathbf{R}^{t(j)} = \mathbf{R}\left(\mathbf{d}^{t(j)}\right)$ and $\mathbf{K}^{t(j)} = \mathbf{K}\left(\mathbf{d}^{t(j)}\right)$ are the residual force vector and the Jacobian matrix at time step $t$ and iteration $j$, respectively.

The goal of intrusive ROM is to replace the solution of (1.3) and (1.4) for a large number of DoFs $n_u$ and $n_p$ with the solution of a computationally efficient ROM for a smaller number of DoFs $m_u$ and $m_p$ ($m_u \ll n_u$ and $m_p \ll n_p$). Galerkin projection method is utilized to approximate the vectors $\mathbf{u}^t \in \mathbb{R}^{n_u}$ and $\mathbf{p}^t \in \mathbb{R}^{n_p}$ by means of a linear combination of the elements of the projection matrices $\mathbf{W}_u \in \mathbb{R}^{n_u \times m_u}$ and $\mathbf{W}_p \in \mathbb{R}^{n_p \times m_p}$, respectively:

$$\mathbf{u}^t = \mathbf{W}_u \widetilde{\mathbf{u}}^t \tag{1.6}$$

$$\mathbf{p}^t = \mathbf{W}_p \widetilde{\mathbf{p}}^t \tag{1.7}$$

where $\widetilde{\mathbf{u}}^t \in \mathbb{R}^{m_u}$ and $\widetilde{\mathbf{p}}^t \in \mathbb{R}^{m_p}$ denote the reduced vector of DoFs for ROM at time $t$. Equations (1.6) and (1.7) can be written in a coupled form

$$\mathbf{d}^t = \mathbf{W} \widetilde{\mathbf{d}}^t \tag{1.8}$$

where $\mathbf{W}$ is called the coupled projection matrix and is presented as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_p \end{bmatrix} \tag{1.9}$$

The displacement and pressure equations at time step $t$ and iteration $j$ are now solved in the reduced space by substituting (1.8) into (1.4) and multiplying both sides by $\mathbf{W}^\top$:

$$\widetilde{\mathbf{K}}^{t(j)} \Delta \widetilde{\mathbf{d}}^{t(j)} = \widetilde{\mathbf{R}}^{t(j)} \tag{1.10}$$

where the updated reduced solution is obtained as

$$\widetilde{\mathbf{d}}^{t(j+1)} = \widetilde{\mathbf{d}}^{t(j)} + \Delta \widetilde{\mathbf{d}}^{t(j)} \tag{1.11}$$

in which $\widetilde{\mathbf{R}}^{t(j)} \in \mathbb{R}^{m_u+m_p}$ and $\widetilde{\mathbf{K}}^{t(j)} \in \mathbb{R}^{(m_u+m_p)\times(m_u+m_p)}$ are the reduced residual force vector and the reduced Jacobian matrix:

$$\widetilde{\mathbf{R}}^{t(j)} = \mathbf{W}^\top \mathbf{R}^{t(j)} \tag{1.12}$$

$$\widetilde{\mathbf{K}}^{t(j)} = \mathbf{W}^\top \mathbf{K}^{t(j)} \mathbf{W} \tag{1.13}$$

So far, ROM construction has been discussed for linear systems such as (1.3); however, nonlinear transient coupled problems present additional challenges, including multiple sources of nonlinearity such as plasticity and permeability enhancement. In these cases, the computational cost of nonlinear terms scales with the dimension of the FOM. To address this issue, hyper-reduction techniques, ranging from global to local versions, are employed to accelerate the computation of the nonlinear terms [39, 40, 41]. More details are provided in Chapters 2, 3, and 4.

Projection-based intrusive ROMs are used in reservoir engineering and have been extended to some applications, including two-phase flow and transport systems. However, they come with a number of limitations. These ROMs neglect the effect of rock deformation in the reservoir and focus solely on a system of pressure-saturation equations. They fail to account for plastic deformation as a highly nonlinear behavior in rock mechanics. In fact, intrusive ROMs have faced significant challenges in learning the path-dependent nonlinear evolution of plastic strain [42]. Furthermore, these ROMs typically lack parameterization for material properties and loading paths, which are important variables in calibration and optimization processes.

There is therefore a need for a novel intrusive ROM that can significantly enhance the computational efficiency of numerical models in reservoirs where rock deformation plays a significant role. In this thesis, a ROM is developed to reduce computational costs in nonlinear and fully-coupled hydro-mechanical simulations. The advantages of ROM are demonstrated in a large-scale industrial application: hydraulic dilation stimulation of a reservoir with a horizontal well pair. The ROM is capable of capturing the path-dependent and nonlinear evolution of plastic strain. The ROM is parameterized by material properties and loading conditions, specifically injection rate. This advancement will provide reservoir engineers with a powerful tool to reduce the computational cost of complex and large-scale simulations. They may use this ROM to design pumping schedules in such a way that the hydraulic dilation process is optimized. In addition, this ROM allows them to accelerate history matching, a process for the calibration and validation of the reservoir models by minimizing the difference between production history data and simulator prediction.

Additional challenges also need to be considered during ROM development for geomechanical problems:

- **Modeling Error:** The modeling error can be significantly larger than the discretization error in geomechanics simulations. For instance, when comparing numerical results to experimental data from a simple triaxial test, large discrepancies often arise due to numerous sources of uncertainty that influence the numerical predictions. Plasticity models, in particular, introduce substantial modeling errors due to their assumptions and simplifications. For example, results obtained using the Drucker–Prager model can differ significantly from those generated by more advanced and complex soil plasticity models. This discrepancy emphasizes the significance of modeling errors compared to discretization errors in geomechanics simulations.

- **The Cost of Offline Training:** When designing the training regime of a ROM, it is essential to evaluate the computational cost of the data generation and training process. The aim is to minimize the effort of the offline stage while maintaining a

highly accurate solution in the online stage. The complexity of the problem, including the number of parametric inputs and the quality of interest, plays an important role in determining the training regime.

- **The Accuracy of the ROM:** The accuracy of the ROM is often a trade-off with its speed-up: as the ROM becomes faster, the accuracy decreases. A ROM must be parameterized by various hyper-parameters, such as the size of the ROM, which influence both its accuracy and computational efficiency. For each specific project, these parameters must be carefully configured to meet the desired level of accuracy while considering the acceptable trade-off with speed-up.

  Acceptable accuracy must reflect the practical considerations and uncertainties present in geomechanical modeling. In practice, the required accuracy depends on the specific application and its associated tolerances. Subsurface modeling is influenced by significant uncertainties arising from geomechanical data (e.g., material properties, in-situ stress conditions), heterogeneity (both quantified and unquantified), and limitations of constitutive models. Furthermore, the assumption of small deformations introduces additional simplifications that may influence the accuracy. For example, when comparing the FOM results to experimental data from a simple triaxial test, large discrepancies can be observed, highlighting the numerous sources of uncertainty that influence FOM accuracy. Thus, the acceptable accuracy of the ROM must be evaluated within these uncertainties.

- **The Domain of Applicability and Extrapolation:** It is essential to identify the valid ranges of inputs that yield reliable results and to define the domain of applicability in both space and time. The performance of the ROM must be evaluated for inputs within and outside the training domain. For example, in industrial applications, the injection rate can serve as an input to the ROM, and the rates in online use may differ significantly from those for training the ROM. An effective ROM should accurately reproduce the response of the reservoir to such dynamic changes in injection rates, including those that extend beyond the training regime."

## 1.4 Purpose and Objectives

Thus, given the need for the development of a robust ROM for the hydraulic dilation stimulation process, the objectives of this thesis are illustrated in this section.

### 1.4.1 Objective 1: Develop an efficient intrusive ROM for nonlinear transient flow in porous media as a diffusion problem

The first objective is to develop an intrusive ROM for nonlinear porous media flow problems with transient and time-discontinuous fluid injection rates. The performance of the ROM is studied in the context of a two-dimensional analysis of time-varying injection into a two-well system in a layered porous media reservoir. The developed ROM consists of the combination of proper orthogonal decomposition and discrete empirical interpolation method (POD-DEIM) as a hyper-reduction technique. Its computational efficiency is compared with the classical proper orthogonal decomposition-Galerkin (POD-Galerkin) without hyper-reduction and the FOM. An analysis of the trade-off between accuracy and run-time is conducted for ROMs of different sizes and hyper-parameters. Furthermore, the impact of the training regime on the performance of the ROMs is evaluated. Specifically, the training regime is generated using the FOM with constant injection rates during the offline stage, while the ROM is tested with more realistic complex online time-discontinuous pumping schedules. This objective is fulfilled in Chapter 2 which is based on the article:

S. Hatefi Ardakani, G. Zingaro, M. Komijani, and R. Gracie, "An efficient reduced order model for nonlinear transient porous media flow with time-varying injection rates," *Finite Elements in Analysis and Design*, vol. 241, p. 104237, 2024.

### 1.4.2 Objective 2: Develop a parameterized ROM for cyclic plasticity models (Drucker–Prager model)

One step in the simulation of hydraulic dilation stimulation is considering nonlinear elasto-plastic behavior for the porous media. Objective 2 will focus on the development of MOR technique for a two-dimensional system of nonlinear equations equipped with pressure-dependent plasticity constitutive model. Drucker–Prager plasticity model is considered for oil sand reservoir, and the challenge is to accurately capture plastic deformation around the wells using ROM. Intrusive ROMs have faced significant challenges in learning the path-dependent nonlinear evolution of plastic strain as a function of a loading path. In this objective, we develop and evaluate an intrusive ROM for a Drucker–Prager plasticity model, in which the material properties and the cyclic load path are parametric inputs. Efficiency is achieved via multiple reduction methods: POD-Galerkin to reduce the total number of DoFs, DEIM as a hyper-reduction technique to accelerate the computation of nonlinear terms, and clustering to enable the use of a local version of hyper-reduction technique (LDEIM). It is shown that the standard DEIM approximation is inefficient for constructing accurate plasticity ROMs, while LDEIM substantially improves the online computational

cost. The objective is completed by developing a ROM for two problems. First, in a micromechanics porous media problem, ROM is parameterized by material properties, and its performance is demonstrated for inputs within and outside of the training domain. Second, the ROM is parameterized by load path for a cyclic plasticity problem. This objective is completed in Chapter 3 which is based on the article:

S. Hatefi Ardakani, G. Zingaro, and R. Gracie, "An efficient parameterized local reduced order model for Drucker–Prager plasticity," *Submitted to Computational Mechanics*, September 2024.

### 1.4.3 Objective 3: Develop a robust ROM for poroplasticity models with an application in hydraulic dilation stimulation

Following the derivation of an intrusive ROM for nonlinear transient porous media flow and plasticity problems, the third objective is to develop a ROM for poroplasticity problems. A novel ROM formulation for fully-coupled hydro-mechanical simulations of deformable porous media with single-phase fluid flow is introduced. The coupled FOM (CFOM) employs a two-dimensional finite element analysis (FEA) equipped with Drucker–Prager plasticity and stress-induced permeability enhancement models to describe the dilation behavior of sandstone. A robust ROM is achieved by clustering the training data for the displacement and pressure equations. Clustering and classification techniques are tailored for the coupled formulation framework. The model is applied to a large-scale application: hydraulic dilation stimulation of a reservoir with a horizontal well pair. Finally, the performance of the ROM is evaluated for a complex case, where the ROM is parameterized with material properties and injection rate. This version of ROM is ready to be used as a powerful tool for the calibration and history matching processes. This objective is described in Chapter 4 which is based on the article:

S. Hatefi Ardakani and R. Gracie, "A parameterized local reduced order model of stimulated volume evolution in reservoirs," *Submitted to International Journal for Numerical and Analytical Methods in Geomechanics*, November 2024.

# Chapter 2

# Intrusive ROM for Nonlinear Transient Porous Media Flow

## 2.1 Introduction

In Chapter 2, an intrusive reduced order model (ROM) for nonlinear porous media flow problems with transient and time-discontinuous fluid injection rates is developed. Many intrusive model order reduction (MOR) approaches, such as Balanced Truncation [43] and proper orthogonal decomposition-Galerkin (POD-Galerkin) [44], efficiently reduce the computational cost of ROM for linear, steady-state, and time-dependent full order model (FOM). However, these standard projection-based techniques are not effective for nonlinear systems of interest here. The computational cost of the nonlinear terms in the ROM scales with the dimensionality of the FOM, which significantly reduces the computational efficiency of the ROM. Hyper-reduction techniques have been proposed to circumvent this drawback by reducing the computational cost without sacrificing the stability and accuracy of the underlying projection. They can be grouped into two families: project-then-approximate methods (energy-conserving sampling and weighting (ECSW) method [45, 46], empirical cubature method (ECM) [47], Linear program-based empirical quadrature method [48]) and approximate-then-project methods (missing point estimation (MPE) [49], empirical interpolation method (EIM) [50], discrete empirical interpolation method (DEIM) [30, 51], Gappy POD [52]).

The POD-Galerkin technique without hyper-reduction reduces the dimensionality of complex systems by determining the optimal basis vectors extracted from singular value decomposition (SVD) of high-fidelity snapshots; however, the complexity of evaluating the

nonlinear terms remains unchanged. To improve computational efficiency for nonlinear problems, the DEIM approach was developed [30, 51] and used to interpolate the nonlinear contributions to the residual and the Jacobian. DEIM has seen some applications in structural dynamics [53], plasticity problems [30], and reservoir simulations [54].

There are several variants of DEIM in the literature. The Q-DEIM (QR factorization DEIM) is similar to DEIM but uses a different method for selecting measurement points, which results in improved error bounds [55, 56]. Trajectory DEIM (TDEIM) approximates the nonlinear operators by combining the trajectory linearization technique with DEIM [54]. Tan *et al.* [54] used TDEIM to generate a ROM for a two-phase heterogeneous reservoir. Matrix DEIM (MDEIM) is the matrix version of DEIM, which allows us to directly calculate the reduced Jacobian approximation instead of approximating the full order Jacobian matrix [57, 58]. Adaptive DEIM (ADEIM) is a dynamic version of DEIM, proposed by Peherstorfer and Willcox [59]. Classical DEIM provides a linear interpolation of nonlinear operators, and it is based on offline data. However, ADEIM is a nonlinear interpolation as the DEIM basis vectors are dynamically regenerated through the online state. Peherstorfer and Willcox [59] proved that ADEIM shows improvements in the accuracy of ROMs. Localized DEIM (LDEIM) is an efficient ROM which approximates the nonlinear components using multiple local DEIM subspaces instead of a single global DEIM subspace [30, 60, 61, 62]. Each local DEIM subspace is tailored to a specific part of the properties of the system, which can be parameters and states [60]. LDEIM has found applications in fluid dynamics [60], solid mechanics [30], electromagnetic problems [61], and reservoir simulations [62].

In this chapter, we focus on the combination of the DEIM and the POD-Galerkin approaches, referred to as POD-DEIM. We study the accuracy and efficiency of the proposed model by solving nonlinear porous media flow problems, specifically those extending from fluid injection from a well. Furthermore, a fast and efficient version of the DEIM approach, that of [28], which is based on the approximated column-reduced Jacobian, has been adopted. In addition, previous studies of POD-Galerkin and similar ROMs have insufficiently studied the relationship between the accuracy of the ROM results and computational time, which will be addressed in this work.

Another challenge for applying ROM technologies to reservoir simulations is that the exact pumping schedule (injection rate vs. time) is not known in advance of the operation, as technical and logistical challenges impact schedules. Thus, the pumping schedule used for training the ROMs can differ greatly from online pumping schedules. Furthermore, pumping schedules are often characterized by active and inactive injection periods. Therefore, an effective ROM will need to accurately reproduce the response of the reservoir, subject to such sudden changes in injection rate. To date, the applications of ROMs to complex and time-discontinuous pumping schedules have not been adequately accessed.

This chapter evaluates (a) the effectiveness of the combination of the DEIM and the POD-Galerkin approaches (POD-DEIM) to develop ROMs for transient nonlinear porous media flow problems; (b) the trade-off between accuracy and run-time, in comparison to coarse FE mesh (less accurate, but fast); and (c) the impact of training regime on the performance of the ROMs, specifically investigating the use of simple training sets for ROMs with more realistic complex online time-discontinuous pumping schedules. Thus, the utility of the POD-DEIM approach for more realistic use cases can be beneficial in practice.

## 2.2 Mathematical Model

This section develops the mathematical formulation for a reservoir-well system consisting of two wells: The fluid is injected into one well, and the pressure is constant for another. First, the governing equations with boundary conditions for nonlinear porous media flow are described. Then, a brief description of the nonlinear finite element formulation is discussed.

As a model problem, consider a two-dimensional reservoir with two wells. In well A (left), fluid is injected at a given rate Q, and in well B (right), pore pressure is maintained at a given elevated level. All boundaries of the modeled reservoir are assumed to be impermeable. Figure 2.1 depicts the $20\,\mathrm{m} \times 20\,\mathrm{m}$ domain with the given boundary conditions (BCs). We also presume that the initial pore pressure is 2 MPa.

The transient flow equations in a porous medium $\Omega$ are given as [63, 64]

$$s\frac{\partial p}{\partial t} = \nabla \cdot \mathbf{q} + Q_p \quad \text{on } \Omega \times (0,T) \tag{2.1}$$

$$\mathbf{n} \cdot \mathbf{q} = h \quad \text{on } \Gamma_q \times (0,T) \tag{2.2}$$

$$p = g \quad \text{on } \Gamma_p \times (0,T) \tag{2.3}$$

$$p\,(\mathbf{x},0) = p_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \tag{2.4}$$

in which $s$ is the specific storage of the medium, and $Q_p$ is the body source. $\Gamma_p$ and $\Gamma_q$ are the boundaries in which pressure (Dirichlet) and mass flux (Neumann) boundary conditions are applied, respectively. $\mathbf{q}$ is the mass flux defined using Darcy's law:

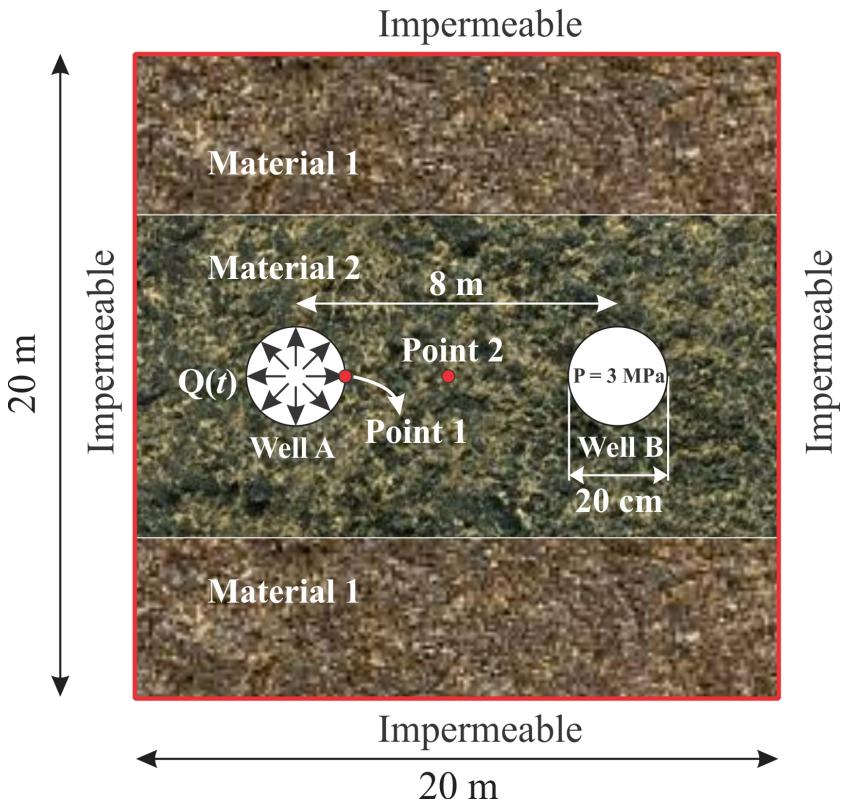



Figure 2.1: Reservoir-well system consisting of two wells (well A with a time-varying injection rate Q($t$) and well B with a constant pressure P).

$$\mathbf{q} = -\frac{\kappa(p)}{\mu}\nabla p \tag{2.5}$$

in which $\kappa$ is the permeability of the medium and $\mu$ is the dynamic viscosity of the fluid. To define a source of nonlinearity in (2.1), it is assumed that the permeability is pressure-dependent ($\kappa = \kappa(p)$).

By multiplying the strong form of (2.1) by a weight function $\delta p$, integrating over the domain, and applying integration by parts, the corresponding weak form can be written as

$$\int_{\Omega} \delta p\, s \frac{\partial p}{\partial t} d\Omega - \int_{\Omega} \delta p\, Q_p d\Omega + \int_{\Gamma} \delta p\, (\mathbf{q}\cdot\mathbf{n})\, d\Gamma - \int_{\Omega} \nabla \delta p \cdot \mathbf{q}(p) d\Omega = 0 \tag{2.6}$$

The aim here is to find the field $p(x, y, t)$ such that the weak form of the nonlinear transient

equation is satisfied.

The domain is discretized into 4-node quadrilateral (Q4) finite elements. Using Galerkin method, the finite element approximation for the pressure field $p^h(x, y, t)$ and the weight function $\delta p$ are

$$p^h(\mathbf{x}, t) = \sum_{I=1}^{4} N_I(\mathbf{x}) p_I(t) = \mathbf{N}\mathbf{p} \tag{2.7}$$

$$\delta p^h(\mathbf{x}) = \sum_{I=1}^{4} N_I(\mathbf{x}) \delta p_I = \mathbf{N}\delta\mathbf{p} \tag{2.8}$$

in which $p_I$ and $\delta p_I$ denote the nodal values of the trial solution and weight function, respectively. $N_I(\mathbf{x})$ are the shape functions of the conventional Q4 isoparametric element. The gradients of $p^h$ and $\delta p^h$ are written as

$$\nabla p^h(\mathbf{x}, t) = \sum_{I=1}^{4} \nabla N_I(\mathbf{x}) p_I(t) = \sum_{I=1}^{4} B_I(\mathbf{x}) p_I(t) = \mathbf{B}\mathbf{p} \tag{2.9}$$

$$\nabla \delta p^h(\mathbf{x}) = \sum_{I=1}^{4} \nabla N_I(\mathbf{x}) \delta p_I = \sum_{I=1}^{4} B_I(\mathbf{x}) \delta p_I = \mathbf{B}\delta\mathbf{p} \tag{2.10}$$

in which $B_I = \nabla N_I$.

Substituting the trial solution (2.7), weight function (2.8), and their gradients (2.9, 2.10) into the weak form (2.6) and evoking the arbitrariness of the $\delta p_I$ leads to the discretized form of the nonlinear transient flow equation

$$\mathbf{S}\dot{\mathbf{p}} + \mathbf{F}_p(\mathbf{p}) + \mathbf{F}_Q = 0 \tag{2.11}$$

in which

$$\mathbf{S} = \int_\Omega \mathbf{N}^\top s \mathbf{N} d\Omega \tag{2.12}$$

$$\mathbf{F}_p = -\int_\Omega \mathbf{B}^\top \mathbf{q}(p^h) d\Omega \tag{2.13}$$

$$\mathbf{F}_Q = \int_{\Gamma_q} h\mathbf{N}^\top d\Gamma - \int_\Omega \mathbf{N}^\top Q_p d\Omega \tag{2.14}$$

The implicit and unconditionally stable backwards Euler method is utilized to numerically integrate the semi-discrete equations (2.11) over time. The relation between the pressure degrees of freedom at time steps $n$ and $n+1$ is defined as

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \dot{\mathbf{p}}^{n+1}\Delta t \tag{2.15}$$

So, the residual of the nonlinear system of equations (2.11) for time step $n+1$ is

$$\mathbf{R}^{n+1} = \mathbf{S}\frac{\mathbf{p}^{n+1} - \mathbf{p}^n}{\Delta t} + \mathbf{F}_p^{n+1} + \mathbf{F}_Q^{n+1} = 0 \tag{2.16}$$

Equation (2.16) is a nonlinear system of equations due to the presumed dependence of the permeability on pressure ($\mathbf{F}_p(p) = \mathbf{F}_p(\kappa(p))$). Applying the Newton–Raphson iteration method to solve the nonlinear system of equations (2.16) results in

$$\mathbf{K}^{n+1(j)}\Delta\mathbf{p}^{n+1(j)} = -\mathbf{R}^{n+1(j)}, \qquad j = 1, ..., J \tag{2.17}$$

$$\mathbf{p}^{n+1(j+1)} = \mathbf{p}^{n+1(j)} + \Delta\mathbf{p}^{n+1(j)} \tag{2.18}$$

in which $\Delta\mathbf{p}^{n+1(j)}$ is the increment of the pressure vector at time step $n+1$ and iteration $j$. $\mathbf{R}^{n+1(j)}$ denotes the residual vector evaluated at $\mathbf{p}^{n+1(j)}$. In addition, $\mathbf{K}^{n+1(j)}$ is the Jacobian matrix and can be defined as

$$\mathbf{K}^{n+1(j)} = \frac{\partial\mathbf{R}^{n+1(j)}}{\partial\mathbf{p}} \mid_{\mathbf{p}^{n+1(j)}} = \frac{\mathbf{S}}{\Delta t} + (\mathbf{K}_p)^{n+1(j)} \tag{2.19}$$

where

$$\mathbf{K}_p = \int_\Omega \mathbf{B}^\top \frac{\kappa(p^h)}{\mu}\mathbf{B}d\Omega + \int_\Omega \mathbf{B}^\top \frac{\kappa'(p^h)}{\mu}\mathbf{B}\mathbf{p}\mathbf{N}d\Omega \tag{2.20}$$

The initial guess for the solution in time step $n+1$ is the converged solution in time step $n$:

$$\mathbf{p}^{n+1(0)} = \mathbf{p}^{n} \tag{2.21}$$

A summary of the Newton–Raphson iteration algorithm for the time integration of nonlinear fluid flow can be found in [65].

## 2.3 Intrusive Model Order Reduction

Intrusive MOR techniques are carried out in an offline-online decomposition. In the offline stage, computationally expensive FEM simulations are performed to generate training data (snapshots). Snapshot matrices are constructed by collating snapshots of pressure DoFs, residual vectors, and the Jacobian matrix at various time steps and for various inputted injection rates. This data set is utilized for training the ROM. Online, the trained ROM is employed to predict the response of the system to arbitrary time-varying injection rates.

We begin with the POD-Galerkin technique without hyper-reduction [44, 66] to reduce the dimensionality of complex systems by determining the optimal basis vectors useful for the projection of the FOM onto the reduced space. Since the complexity of evaluating the nonlinear terms remains unchanged in POD-Galerkin, DEIM [51] is then utilized to approximate the residual force vector and its Jacobian using empirical interpolation techniques. Finally, a ROM, referred to as POD-DEIM, is used to reduce the dimensionality and complexity of nonlinear transient flow from wells by combining POD-Galerkin and DEIM.

### 2.3.1 POD-Galerkin: Proper Orthogonal Decomposition-Galerkin Projection

#### Galerkin Projection

The most widely used methods for constructing an intrusive ROM are projection-based techniques. In this section, we use the Galerkin projection method to reduce the dimensionality of the nonlinear transient flow equations. By using a weighted linear combination of the optimal orthonormal basis vectors ($\mathbf{U} = [\phi_1 ..... \phi_k] \in \mathbb{R}^{N \times k}$ ($k \ll N$)), a projection process is employed to approximate the full order state from a reduced order state

$$\mathbf{p}(t) \approx \mathbf{U}\tilde{\mathbf{p}}(t) \tag{2.22}$$

where $\mathbf{p}(t) \in \mathbb{R}^N$ is the full order state vector, and $\tilde{\mathbf{p}}(t) \in \mathbb{R}^k$ is the reduced time-dependent coefficient vector of order $k \ll N$. By projecting the nonlinear system of equations (2.17, 2.18) using $\mathbf{U}$ and applying the orthogonality condition, the dimensionality reduced system of equations in time step $n+1$ and iteration $j$ for the Newton–Raphson iteration method are given as

$$\widetilde{\mathbf{K}}^{n+1(j)} \Delta \widetilde{\mathbf{p}}^{n+1(j)} = -\widetilde{\mathbf{R}}^{n+1(j)}, \qquad j = 1, ..., J \tag{2.23}$$

$$\widetilde{\mathbf{p}}^{n+1(j+1)} = \widetilde{\mathbf{p}}^{n+1(j)} + \Delta \widetilde{\mathbf{p}}^{n+1(j)} \tag{2.24}$$

where $\widetilde{\mathbf{R}} \in \mathbb{R}^k$ and $\widetilde{\mathbf{K}} \in \mathbb{R}^{k \times k}$ denote the reduced residual vector and the reduced Jacobian matrix, respectively. They are defined as

$$\widetilde{\mathbf{R}}^{n+1(j)} = \mathbf{U}^\top \mathbf{R}(\mathbf{U}\widetilde{\mathbf{p}})^{n+1(j)} = \widetilde{\mathbf{S}} \dot{\widetilde{\mathbf{p}}}^{n+1(j)} + (\widetilde{\mathbf{F}}_p)^{n+1(j)} + (\widetilde{\mathbf{F}}_Q)^{n+1(j)} = \mathbf{0} \tag{2.25}$$

$$\widetilde{\mathbf{K}}^{n+1(j)} = \frac{\partial \widetilde{\mathbf{R}}^{n+1(j)}}{\partial \widetilde{\mathbf{p}}} \mid_{\widetilde{\mathbf{p}}^{n+1(j)}} = \mathbf{U}^\top \mathbf{K}^{n+1(j)} \mathbf{U} = \frac{\widetilde{\mathbf{S}}}{\Delta t} + (\widetilde{\mathbf{K}}_p)^{n+1(j)} \tag{2.26}$$

In (2.25, 2.26), $\widetilde{\mathbf{S}} \in \mathbb{R}^{k \times k}$, $\widetilde{\mathbf{K}}_p \in \mathbb{R}^{k \times k}$, $\widetilde{\mathbf{F}}_Q \in \mathbb{R}^k$, and $\widetilde{\mathbf{F}}_p \in \mathbb{R}^k$ are of the form

$$\widetilde{\mathbf{S}} = \mathbf{U}^\top \mathbf{S} \mathbf{U} \tag{2.27}$$

$$\widetilde{\mathbf{K}}_p = \mathbf{U}^\top \mathbf{K}_Q \mathbf{U} \tag{2.28}$$

$$\widetilde{\mathbf{F}}_Q = \mathbf{U}^\top \mathbf{F}_Q \tag{2.29}$$

$$\widetilde{\mathbf{F}}_p = \mathbf{U}^\top \mathbf{F}_p \tag{2.30}$$

### POD: Proper Orthogonal Decomposition

The quality of approximations (2.27, 2.28, 2.29, 2.30) is clearly affected by the choice of basis vectors $\mathbf{U}$. One method for constructing the reduced basis vectors is the POD. POD is a technique to determine the optimal orthonormal basis vectors $\mathbf{U} = [\phi_1 \ldots\ldots \phi_k]$ which contain the overall physical behavior of a time-dependent system. In the literature, there are three different variants of POD: Karhunen–Loève decomposition (KLD), principal component analysis (PCA), and singular value decomposition (SVD) [67]. In this study, the variant of the POD known as singular value decomposition of snapshots is utilized. The aim is to find a set of basis vectors $\{\phi_1 \ldots\ldots \phi_k\}$ in which the snapshot matrices can be optimally represented. Snapshots can be produced by varying the parameters of the system, e.g. time steps or injection rates. Full FE analysis is carried out in the training process to compute the solution at each time step. Then, snapshot matrices are constructed by collecting the corresponding pressure, nonlinear force, and Jacobian vectors in matrices $\mathbf{P}^{\text{SNP}}$, $\mathbf{F}^{\text{SNP}}$, and $\mathbf{K}^{\text{SNP}}$:

$$\mathbf{P}^{\text{SNP}} = \begin{bmatrix} | & | & & | \\ \mathbf{p}_1 & \mathbf{p}_2 & \ldots & \mathbf{p}_{n_s} \\ | & | & & | \end{bmatrix} \tag{2.31}$$

$$\mathbf{F}^{\text{SNP}} = \begin{bmatrix} | & | & & | \\ \mathbf{F}_1 & \mathbf{F}_2 & \ldots & \mathbf{F}_{n_s} \\ | & | & & | \end{bmatrix} \tag{2.32}$$

$$\mathbf{K}^{\text{SNP}} = \begin{bmatrix} | & | & & | \\ \mathbf{K}_1 & \mathbf{K}_2 & \ldots & \mathbf{K}_{n_s} \\ | & | & & | \end{bmatrix} \tag{2.33}$$

According to [68], a set of orthonormal basis vectors $\{\phi_1 \ldots\ldots \phi_k\}$ solves the below minimization problem. In other words, the Euclidean norm error between snapshots and their projection on lower-dimensional space is minimized to calculate the basis vectors:

$$\underset{\{\phi_1 \ldots\ldots \phi_k\}}{\arg\min} \sum_{m=1}^{n_s} \left\| \mathbf{p}_m - \sum_{n=1}^{k} (\mathbf{p}_m^{\top} \phi_n) \phi_n \right\|_2^2 \tag{2.34}$$

where

$$\phi_m^\top \phi_n = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases} \tag{2.35}$$

Equation (2.35) shows one of the important properties for basis vectors, i.e. orthogonality. According to [68], the SVD of the snapshot matrix $\mathbf{P}^{\text{SNP}} \in \mathbb{R}^{N \times n_s}$ leads to a solution to the optimization problem (2.34). SVD decomposes the complex matrix $\mathbf{P}^{\text{SNP}}$ into the matrices $\mathbf{U}^{\text{F}}$, $\mathbf{\Sigma}^{\text{F}}$, and $\mathbf{V}^{\text{F}^\top}$:

$$\mathbf{P}^{\text{SNP}} = \mathbf{U}^{\text{F}} \mathbf{\Sigma}^{\text{F}} \mathbf{V}^{\text{F}^\top} \tag{2.36}$$

where $\mathbf{U}^{\text{F}} = [\phi_1 ..... \phi_r] \in \mathbb{R}^{N \times r}$ and $\mathbf{V}^{\text{F}} \in \mathbb{R}^{n_s \times r}$ are full orthonormal matrices, consisting of the eigenvectors of matrices $\mathbf{P}^{\text{SNP}} \left(\mathbf{P}^{\text{SNP}}\right)^\top$ and $\left(\mathbf{P}^{\text{SNP}}\right)^\top \mathbf{P}^{\text{SNP}}$, respectively. $\mathbf{\Sigma}^{\text{F}} \in \mathbb{R}^{r \times r}$ is the full diagonal matrix:

$$\mathbf{\Sigma}^{\text{F}} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_r) \tag{2.37}$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$ are the singular values.

### POD Truncation Error

By applying the SVD on (2.34), the minimum error between snapshots and their projection on lower-dimensional space is obtained by [51]

$$\sum_{m=1}^{n_s} \left\| \mathbf{p}_m - \sum_{n=1}^{k} (\mathbf{p}_m^\top \phi_n) \phi_n \right\|_2^2 = \sum_{n=k+1}^{r} \sigma_n^2 \tag{2.38}$$

The total number of POD basis vectors is determined by the rank of $\mathbf{P}^{\text{SNP}} (r \leq \min(N, n_s)$. Our goal is to truncate the POD basis vectors with a low-energy approximation to achieve the desired accuracy of the ROM. Equation (2.38) indicates that the projection error in POD only depends on the neglected singular values ($(k+1)$th singular value to $r$th singular value). Hence, a criterion for the relative energy is defined to determine the number of truncated POD basis vectors:

$$\frac{\sum_{n=k+1}^{r} \sigma_n^2}{\sum_{n=1}^{r} \sigma_n^2} \leq \text{TOL} \tag{2.39}$$

where TOL is a tolerance and can be a value between 0 and 1. As the value of TOL tends to zero, the number of truncated POD basis vectors increases, resulting in a more accurate ROM. A summary of POD basis computation is presented in Algorithm 1.

**Algorithm 1** POD Basis Computation

**Data:** Snapshot matrix $\mathbf{P}^{\text{SNP}} \in \mathbb{R}^{N \times n_s}$

**Result:** A set of truncated POD basis vectors $\mathbf{U} = [\phi_1 \ldots\ldots \phi_k] \in \mathbb{R}^{N \times k}$

Construct the SVD of $\mathbf{P}^{\text{SNP}}$: $\mathbf{P}^{\text{SNP}} = \mathbf{U}^{\text{F}} \mathbf{\Sigma}^{\text{F}} {\mathbf{V}^{\text{F}}}^{\top}$

Compute $\mathbf{U}$ by truncating the basis vectors $\mathbf{U}^{\text{F}}$ using (2.39)

### 2.3.2 DEIM: Discrete Empirical Interpolation Method

As illustrated in Section 2.3.1, the POD-Galerkin technique without hyper-reduction reduces the dimensionality of complex systems by determining the optimal basis vectors extracted from SVD of high-fidelity snapshots. However, the complexity of evaluating nonlinear terms $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{K}}$ in (2.25) and (2.26) scales with the dimensionality of the FOM, since they are dependent on the dimension of the full order terms $\mathbf{R}$ and $\mathbf{K}$. This makes the direct use of the POD-Galerkin too computationally expensive. To overcome this drawback, DEIM, as a hyper-reduction technique, is employed to accelerate the computation of the nonlinear terms. DEIM, known as a discrete version of the empirical interpolation method (EIM) [51, 69], utilizes empirical interpolation techniques for fast approximation of the reduced residual vector $\widetilde{\mathbf{R}}$ and the reduced Jacobian matrix $\widetilde{\mathbf{K}}$ in the reduced nonlinear system of equations (2.23).

#### Hyper-reduced Residual

In DEIM, by using a weighted linear combination of the truncated basis vectors ($\mathbf{Z} = [\mathbf{z}_1 \ldots\ldots \mathbf{z}_m] \in \mathbb{R}^{N \times m}$, $m \ll N$), a projection-based process is employed to approximate the full order nonlinear term $\mathbf{R}$ from a lower-dimensional subspace $\mathscr{Z} = \text{span}\,\{\mathbf{z}_1 \ldots\ldots \mathbf{z}_m\} \subset \mathbb{R}^N$:

$$\mathbf{R}(t) \approx \hat{\mathbf{R}} = \mathbf{Z}\mathbf{c}(t) \tag{2.40}$$

where $\mathbf{c}(t) \in \mathbb{R}^m$ is the reduced time-dependent coefficient vector of order $m \ll N$. The aim is to determine the set of basis vectors $\mathbf{Z}$ and the corresponding coefficient vector

$\mathbf{c}(t)$. To find the set of basis vectors $\mathbf{Z}$, a snapshot matrix for the nonlinear term $\mathbf{R}$ is first constructed ($\mathbf{R}^{\text{SNP}} = [\mathbf{R}_1 ... \mathbf{R}_{n_s}]$). Then, the left singular vectors obtained from SVD of $\mathbf{R}^{\text{SNP}}$ is considered as the set of basis vectors $\mathbf{Z}$:

$$\mathbf{R}^{\text{SNP}} = \mathbf{Z}\Sigma_R\mathbf{V}_R^\top \tag{2.41}$$

To determine the coefficient vector $\mathbf{c}(t)$, both sides of the overdetermined equation (2.40) are multiplied by the measurement matrix $\mathbf{P} = [\mathbf{e}_{\gamma 1}, ..., \mathbf{e}_{\gamma m}] \in \mathbb{R}^{N\times m}$. The result is the selection of $m$ rows of the system of equations

$$\mathbf{P}^\top\mathbf{R}(t) \approx \mathbf{P}^\top\mathbf{Z}\mathbf{c}(t) \tag{2.42}$$

$\mathbf{e}_{\gamma i} = [0\ 0\ ...\ 0\ 0\ 1\ 0\ 0\ ...\ 0\ 0]^\top \in \mathbb{R}^N$ denotes the $\gamma_i$th column of the $N$-dimensional identity matrix. Algorithm 2 shows the procedure used to compute $\mathbf{e}_{\gamma i}$. From (2.40) and (2.42), the final approximation to the nonlinear term $\mathbf{R}$ is of the form

$$\hat{\mathbf{R}}(t) = \mathbf{Z}(\mathbf{P}^\top\mathbf{Z})^{-1}\mathbf{P}^\top\mathbf{R}(t) \in \mathbb{R}^N \tag{2.43}$$

The above equation illustrates a reduced complexity in evaluating the nonlinear terms. In the offline stage, the matrices $\mathbf{Z}$ and $\mathbf{P}^\top$, and consequently $\mathbf{Z}(\mathbf{P}^\top\mathbf{Z})^{-1}$ are calculated. Then, in the online stage, $\mathbf{R}_{\text{DEIM}} = \mathbf{P}^\top\mathbf{R}(t) \in \mathbb{R}^m$, where $m \ll N$, is constructed for each iteration of the Newton–Raphson method.

### Hyper-reduced Jacobian

The main issue with DEIM is associated with the calculation of the Jacobian matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$, since its computational cost scales with the dimensionality of the FOM. There are several interpolation-based approaches to efficiently compute the large Jacobian matrix [28, 30, 51].

The first method method is to approximate the Jacobian with the same POD basis vectors calculated for the nonlinear residual vector [30, 51]:

$$\mathbf{K} \approx \hat{\mathbf{K}} = \frac{\partial\hat{\mathbf{R}}}{\partial\mathbf{p}} = \mathbf{Z}(\mathbf{P}^\top\mathbf{Z})^{-1}\frac{\partial\mathbf{R}_{\text{DEIM}}}{\partial\mathbf{p}} = \mathbf{Z}(\mathbf{P}^\top\mathbf{Z})^{-1}\mathbf{k}_{\text{DEIM}} \tag{2.44}$$

**Algorithm 2** DEIM Algorithm

**Data:** Snapshot matrix for nonlinear term $\mathbf{R}$ : $\mathbf{R}^{\text{SNP}} = [\mathbf{R}_1 ... \mathbf{R}_{n_s}]$

**Result:** Measurement matrix: $\mathbf{P} = [\mathbf{e}_{\gamma 1}, ..., \mathbf{e}_{\gamma m}]$ and location of the elements of the nonlinear term $\mathbf{R}$ selected by DEIM: $\mathbf{LOC} = [\gamma_1, ..., \gamma_m]$

Compute singular value decomposition of $\mathbf{R}^{\text{SNP}}$: $\mathbf{R}^{\text{SNP}} = \mathbf{Z}\Sigma_R\mathbf{V}_R^\top$

Construct $\mathbf{Z}_m = [\mathbf{z}_1 ..... \mathbf{z}_m]$

Initiate $\mathbf{LOC} = [\gamma_1]$ where $\gamma_1 =$ index of the maximum element of $|\mathbf{z}_1|$

Initiate measurement matrix $\mathbf{P}$: $\mathbf{P} = [\mathbf{e}_{\gamma_1}]$

**for** $j = 2, ..., m$ **do**

- Compute $\mathbf{c}_j$ by solving $\mathbf{P}^\top\mathbf{Z}_{j-1}\mathbf{c}_j = \mathbf{P}^\top\mathbf{z}_j$ where $\mathbf{Z}_{j-1} = [\mathbf{z}_1 ..... \mathbf{z}_{j-1}]$
- Compute residual $\mathbf{RES} = \mathbf{z}_j - \mathbf{Z}_{j-1}\mathbf{c}_j$
- Determine $\gamma_j =$ index of the maximum element of residual $|\mathbf{RES}|$
- Update matrices $\mathbf{P}$ and $\mathbf{LOC}$: $\mathbf{P} = \begin{bmatrix} \mathbf{P} & \mathbf{e}_{\gamma_j} \end{bmatrix}$ and $\mathbf{LOC} = \begin{bmatrix} \mathbf{LOC} & \gamma_j \end{bmatrix}$

The computational cost of $\mathbf{k}_{\text{DEIM}} \in \mathbb{R}^{m \times N}$ remains high as it requires looping over all elements. $\mathbf{k}_{\text{DEIM}}$ can be calculated by solely computing contributions from the elements that have shared measurement points, resulting in significant computational savings without a reduction in accuracy [30, 51].

The second method involves approximating Jacobian with a new set of POD basis vectors that are associated with the snapshots of Jacobian. As suggested in [28], rather than focusing on the full order Jacobian $\mathbf{K}^{(j)} \in \mathbb{R}^{N \times N}$, column-reduced Jacobian $\mathbf{K}^{(j)}\mathbf{U} \in \mathbb{R}^{N \times k}$ is approximated in (2.26), leading to a significant reduction in the computational cost. We consider the column-reduced Jacobian as a collection of columns; thus, its columns are approximated independently using DEIM. The procedure for approximating the full order column-reduced Jacobian $\mathbf{K}^{(j)}\mathbf{U}$ using DEIM is detailed in Algorithm 3.

**Algorithm 3** Procedure for approximating the full order column-reduced Jacobian using DEIM

**Data:** Snapshot matrices for full order column-reduced Jacobian ($\mathbf{K}_{cr_1}$ $\mathbf{K}_{cr_2}$ ... $\mathbf{K}_{cr_{n_s}}$)

**Result:** Aproximated column-reduced Jacobian matrix $\widehat{\mathbf{K}}_{cr}$ by using DEIM

**for** $j = 1, ..., k$ **do**

- Construct snapshot matrix for $j$th column of column-reduced Jacobian: $\mathbf{COL}_j^{\text{SNP}} = \left[\mathbf{K}_{cr_1}(:, j)\ \mathbf{K}_{cr_2}(:, j)\ .....\ \mathbf{K}_{cr_{n_s}}(:, j)\right]$
- Compute matrices $\mathbf{P}_j$ and $\mathbf{LOC}_j$ using Algorithm 2. The input data for the algorithm 2 is $\mathbf{COL}_j^{\text{SNP}}$
- Compute singular value decomposition of $\mathbf{COL}_j^{\text{SNP}}$: $\mathbf{COL}_j^{\text{SNP}} = \mathbf{Z}_j \Sigma_j \mathbf{V}_j^\top$
- Compute approximation to $j$th column of column-reduced Jacobian $\widehat{\mathbf{K}}_{cr_j}$: $\widehat{\mathbf{K}}_{cr_j} = \mathbf{Z}_j(\mathbf{P}_j^\top \mathbf{Z}_j)^{-1}\mathbf{P}_j^\top \mathbf{K}_{cr}(:, j)$

Construct approximated column-reduced Jacobian matrix $\widehat{\mathbf{K}}_{cr} = \left[\widehat{\mathbf{K}}_{cr_1}\ \widehat{\mathbf{K}}_{cr_2}\ .....\ \widehat{\mathbf{K}}_{cr_k}\right]$

In Algorithm 3, $\mathbf{K}_{cr}$ and $\widehat{\mathbf{K}}_{cr}$ represent the abbreviations defined for the column-reduced Jacobian matrix $\mathbf{K}\,\mathbf{U} \in \mathbb{R}^{N\times k}$ and its approximation, respectively.

The first method is computationally faster than the second one. However, the most important concern is the convergence of the Newton–Raphson method. The second method provides a more accurate approximation of the Jacobian, resulting in better performance of the Newton–Raphson algorithm and fewer iterations to converge. This study employs both approaches to evaluate the efficacy of the methods in different nonlinear problems. In this chapter, we adopt the first approach, while in Chapters 3 and 4, the second approach is used to approximate the Jacobian matrix.

The current study employs a combination of POD-Galerkin projection and DEIM to build a ROM, called POD-DEIM, for a nonlinear transient fluid flow.

## 2.4 Results and discussion

The wellbore injectivity responses during fluid production are known as complex phenomena in geomechanics due to their temporal and spatial behaviors. The movement of fluid is

essential in such porous media since it produces deformation in their solid skeletons. In addition, injecting fluid into a gas and oil reservoir through an injector well and pumping gas and oil from a producer well play an important role in conventional and unconventional oil and gas reservoirs. Hence, we consider a system of wellbores inside a porous medium as an oil and gas reservoir where the efficiency and accuracy of the proposed MOR technique are observed. As depicted in Figure 2.1, the system is a practical two-dimensional reservoir for nonlinear transient flow in the porous medium with two wells.

The material properties of the fluid are considered as follows,

$$\rho = 1000\,{}^{\mathrm{kg}}/_{\mathrm{m}^3};\ \mu = 0.0011\mathrm{Pa.s}$$

The medium is idealized as a layered rock mass with different permeability. The permeability of side layers is constant,

$$\kappa_1 = 1\,\mathrm{mD}$$

To make the problem more complex (nonlinear) and to evaluate the performance of our POD-DEIM, the below pressure-dependent relation is defined for the permeability of the middle layer:

$$\kappa_2 = 0.1 + \frac{100}{9}\,(p-2)^2\ \mathrm{mD}$$

With this $k$-$p$ relationship, permeability varies from 0.1 mD to nearly 50 mD over a pressure change of 2 MPa, implying a critically stressed media and that the pressure changes cause considerable plastic dilation, which is not directly modeled. The relationship is not based on measurements; however, using such an aggressive $k$-$p$ relationship will allow for initial conclusions about the utility of the ROM for well stimulation problems in which porosity and permeability changes are large.

The fluid is injected into the left well, flowing through the porous medium. For the right well, it is assumed that the pressure is constant. Figure 2.1 depicts the $20\,\mathrm{m} \times 20\,\mathrm{m}$ domain with the given BCs.

We assume that the initial pressure distribution is 2 MPa for all the DoFs since this two-dimensional problem is the cross-section of a three-dimensional model at a depth of about 200 m.

The rock mass field is discretized using 4-node quadrilateral elements (Q4). Figure 2.2 depicts the coarsest mesh (mesh 1) around the well with 1423 DoFs. Furthermore, five finer meshes are considered for FE training process (mesh 2: 5998 DoFs, mesh 3: 10810 DoFs, mesh 4: 17614 DoFs, mesh 5: 35114 DoFs and mesh 6: 101121 DoFs). It should

be mentioned that the finest mesh (mesh 6) acts as a reference solution since there is no analytical solution for this problem.

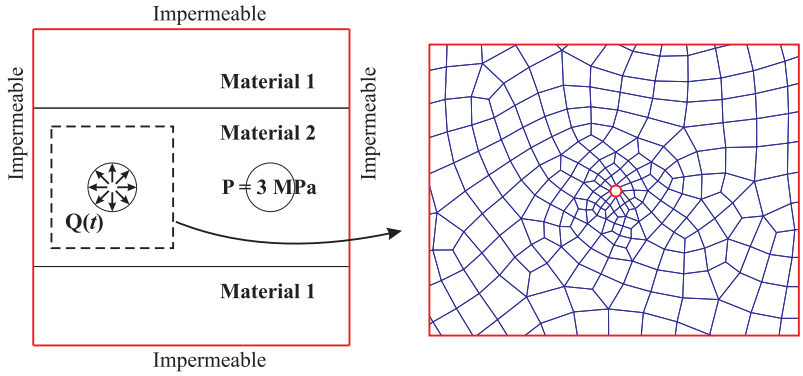



Figure 2.2: Coarse finite element mesh with 1423 DoFs for FE training process.

Numerical time integration is carried out using the backward Euler method for the total time = 87 days and the number of time steps is 200.

### 2.4.1 Comparison between FOM, POD-Galerkin and POD-DEIM

This section will investigate the accuracy and run-time of the POD-Galerkin, POD-DEIM and FOM for the nonlinear time-dependent system. In addition, the accuracy of the trained model and its dependency on mesh size, the number of truncated modes, and the snapshot period will be discussed.

In the training process, a full FE analysis is carried out to compute the solution at each time step. Then, to compute the POD basis, snapshot matrices are constructed by collecting the corresponding pressure, nonlinear force, and Jacobian vectors in matrices $\mathbf{P}^{\text{SNP}} = [\mathbf{p}_1 ... \mathbf{p}_{200}]$ and $\mathbf{F}^{\text{SNP}} = [\mathbf{F}_1 ... \mathbf{F}_{200}]$, and $\mathbf{K}^{\text{SNP}} = [\mathbf{K}_1 ... \mathbf{K}_{200}]$.

A ROM is generated using the mentioned snapshot matrices. As a result of our POD-DEIM technique, the pressure overtime at two characteristic points, as specified in Figure 2.1, is shown in Figure 2.3. It can be seen that the ROM captures the solution of the FOM with high accuracy. Here, the size of the system of equations is reduced to $40 \times 40$ from $10810 \times 10810$ while the online accuracy is guaranteed. Furthermore, using DEIM, the residual vector is interpolated using 400 measurement points versus the complete vector with 10810 components. Similarly, the Jacobian is interpolated using 10 measurement points rather than the complete Jacobian with 96158 number of non-zero entries. In similar

applications, it has been noted that fewer measurement points are required to accurately interpolate the Jacobian compared to the residual vector [57].

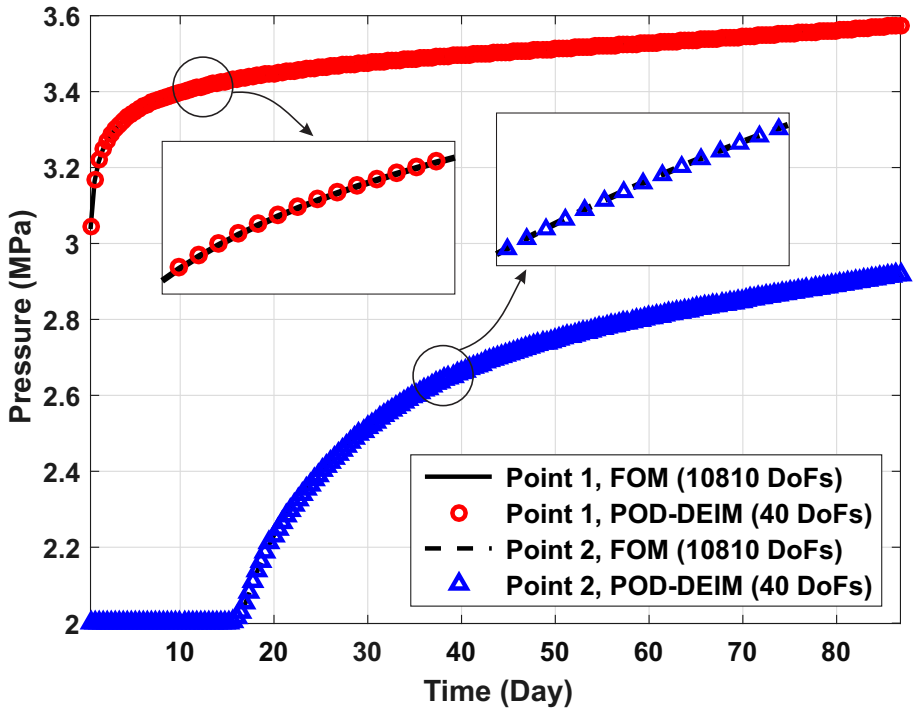



Figure 2.3: Fluid pressure overtime at two characteristic points using POD-DEIM and FOM.

Figure 2.4 illustrates pressure field snapshots at 6.5, 34.3 and 87 days. Initially, the fluid flows from well A to B in the porous media since the fluid is injected into well A. After some time, the fluid stream from well A begins to converge with the stream from well B near the midpoint between the wells. Whereby the stream from well A begins to dominate the rock mass. It should be mentioned that most of the fluid flow with the domain occurs in Material 2 due to its higher permeability than Material 1. In Figure 2.4, FOM results and the relative error of the POD-DEIM are presented. The relative error at each DoF is calculated by

$$e_I(t) = \frac{\left|p_I^{\mathrm{ROM}}(t) - p_I^{\mathrm{FOM}}(t)\right|}{\left|p_I^{\mathrm{FOM}}(t)\right|} \tag{2.45}$$

where subscript $I$ refers to the $I$th DoF. The error contour evidently shows that the

maximum relative error is $3.5 \times 10^{-3}$, meaning that the ROM is sufficiently accurate in the whole domain.

The mass flux and error between the POD-DEIM and the FOM are presented in Figure 2.5. It is worth mentioning that it is not possible to use (2.45) for the calculation of the error of mass flux as the amount of mass flux in some DoFs is zero, leading to division by zero in (2.45). Hence, a scaled difference between FOM and POD-DEIM mass fluxes is represented in Figure 2.5:

$$D_{s,I}(t) = \frac{\left|q_I^{\mathrm{ROM}}(t) - q_I^{\mathrm{FOM}}(t)\right|}{\max(|\mathbf{q}^{\mathrm{FOM}}(t)|)} \tag{2.46}$$

As shown in Figure 2.5, the maximal scaled difference does not exceed $2.5 \times 10^{-3}$, showing an acceptable accuracy mass flux for POD-DEIM.

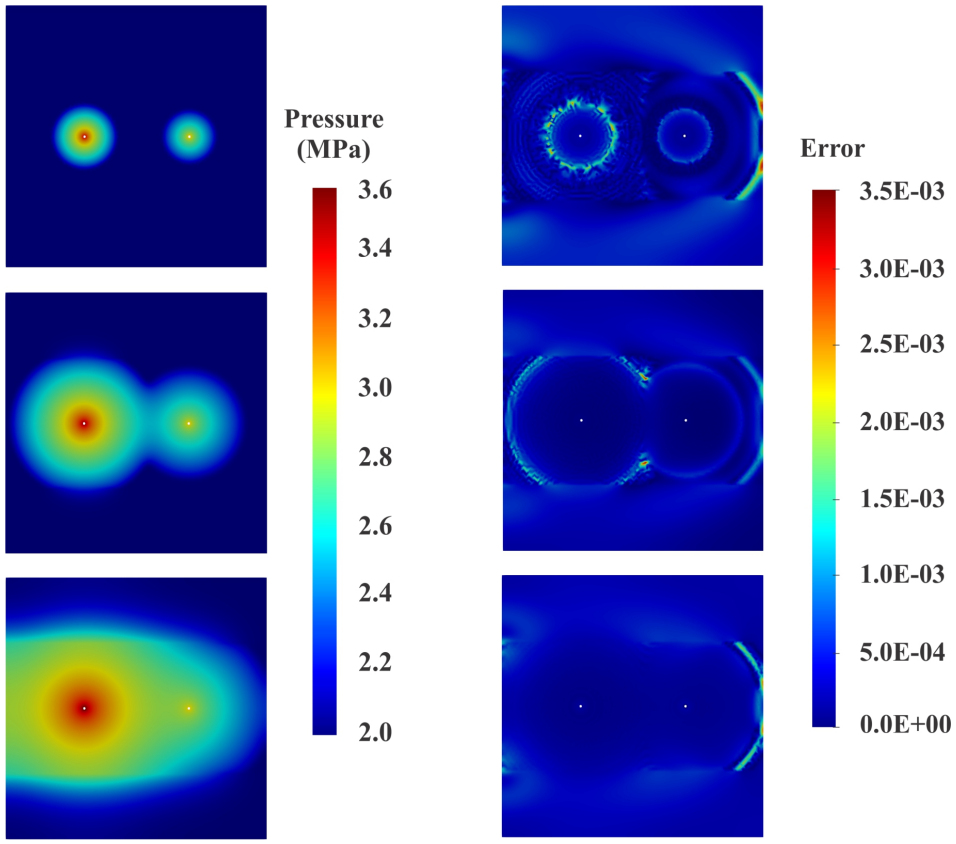



Figure 2.4: Pressure field snapshots at t = 6.5 days, t = 34.3 days, t = 87 days, from top to bottom. Left column: FOM results and right column: POD-DEIM's relative error.

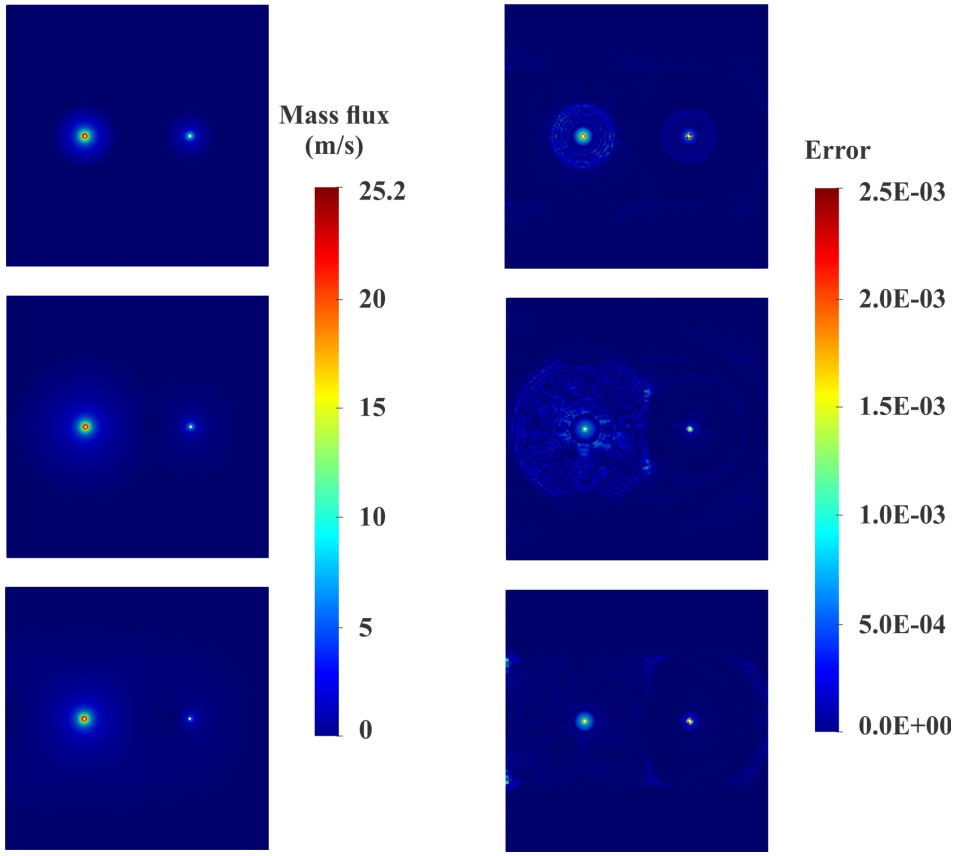



Figure 2.5: Mass flux snapshots at t = 6.5 days, t = 34.3 days, t = 87 days, from top to bottom. Left column: FOM results and right column: scaled difference between FOM and POD-DEIM results.

In Table 2.1 and Figure 2.6, the efficiency and accuracy of three different techniques (POD-Galerkin, POD-DEIM, and FOM) are compared for the designed nonlinear time-dependent system.

In order to investigate the computational cost of the three different techniques, the CPU time from different training meshes is presented in Table 2.1. It can be seen that POD-Galerkin is faster than FOM for different training meshes since POD-Galerkin reduces the dimensionality of the complex system using optimal basis vectors. However, the reduction in computational cost is not significant. This can be explained by noting that the POD-Galerkin does not reduce the complexity of evaluating the nonlinear terms, meaning that most CPU time is devoted to formulating the full order residual force vector and its full order Jacobian at each Newton–Raphson iteration. To improve the drawback of POD-Galerkin,

the POD-DEIM is utilized to approximate the time-dependent nonlinear contributions (residual force and its Jacobian). As a result, the POD-DEIM significantly decreases the CPU time compared to POD-Galerkin and FOM. For instance, POD-DEIM is about 190 times faster than two other techniques for a very large model, where the size of the system of equations is $35114 \times 35114$.

Table 2.2 also shows the total number of Newton–Raphson iterations required from the FOM, POD-Galerkin, and POD-DEIM techniques for different training meshes. Although the difference between the three numerical techniques in the total number of Newton iterations is small, the POD-DEIM requires more Newton–Raphson iterations than the FOM and POD-Galerkin methods. This is attributed to the fact that the POD-DEIM uses approximations to the full order residual force and its Jacobian to accelerate the computation of each iteration.

| Mesh | CPU Time (second) | | |
|---|---|---|---|
| | FOM | POD-Galerkin | POD-DEIM |
| 1 | 1285.4 | 657.1 | 768.0 |
| 2 | 7641.7 | 3587.5 | 1028.0 |
| 3 | 13825.0 | 7817.0 | 1122.4 |
| 4 | 42336.8 | 28885.4 | 1201.4 |
| 5 | 271042.1 | 210282.9 | 1396.1 |

Table 2.1: CPU time (second) needed from the FOM, POD-Galerkin, and POD-DEIM techniques for different training meshes

| Mesh | Total Number of Newton Iterations | | |
|---|---|---|---|
| | FOM | POD-Galerkin | POD-DEIM |
| 1 | 1286 | 1354 | 1439 |
| 2 | 1291 | 1451 | 1414 |
| 3 | 1300 | 1489 | 1608 |
| 4 | 1302 | 1513 | 1645 |
| 5 | 1546 | 1546 | 1668 |

Table 2.2: Total number of Newton–Raphson iterations needed from the FOM, POD-Galerkin, and POD-DEIM techniques for different training meshes (note that the total number of time steps is 200)

To show the performance of POD-DEIM, $L_2$ norm error (2.47) and $H_1$ norm error (2.48) in conjunction with CPU time for a single-core processor are depicted in Figure 2.6 for different

training meshes (from coarse mesh (mesh 1) to very fine mesh (mesh 5)). In other words, the results present the effect of the number of DoFs on the accuracy and the computational cost for three different techniques. It should be mentioned that the finest mesh (mesh 6) acts as a reference solution used to compute the error, since there is no analytical solution for this problem. As can be seen, for the cases of POD-Galerkin and FOM, increasing the number of DoFs decreases the $L_2$ error while the CPU time significantly increases, showing the inefficiency of these two techniques for large systems. However, in the light of POD-DEIM, increasing the number of DoFs does not significantly increase the CPU time. Rather, the accuracy is well guaranteed. For example, in the case of 35114 DoFs, POD-DEIM decreases the run time from 270000 seconds to 1400 seconds while also maintaining a highly accurate solution. Figure 2.6 shows that the online computational cost of POD-DEIM is not entirely independent of the FOM size. There are three main reasons for this behavior. The first reason is related to the total number of Newton–Raphson iterations required by the POD-DEIM technique for different FE meshes. The total number of iterations varies depending on the FOM size, directly impacting the online computational cost. Finer meshes generally require more Newton–Raphson iterations compared to coarser meshes. The second reason involves the computation of the approximation for the nonlinear force vector and the $j$th column of column-reduced Jacobian $\widehat{\mathbf{K}}_{cr_j} = \mathbf{Z}_j(\mathbf{P}_j^\top \mathbf{Z}_j)^{-1}\mathbf{P}_j^\top \mathbf{K}_{cr}(:, j)$. In this chapter, full matrices were used to perform these multiplications during the online phase. This process is time-consuming, particularly for large FOM sizes. As the FOM size increases, the computational cost of these multiplications grows. In Chapters 3 and 4, this issue is addressed by using sparse matrices instead of full matrices. The third reason relates to the DEIM algorithm used in this thesis. In the online phase, approximating $\mathbf{R}$ and $\mathbf{K}$ requires a loop over the measurement points, and within this loop, another loop iterates over the DEIM elements. If a DEIM element is shared among two or more measurement points, it is only considered once in the calculation of $\mathbf{R}_{\text{DEIM}}$ and $\mathbf{k}_{\text{DEIM}}$. For coarse meshes, measurement points share more DEIM elements, resulting in fewer iterations over the DEIM elements. In contrast, in finer meshes, the measurement points share fewer DEIM elements, necessitating additional iterations to calculate $\mathbf{R}_{\text{DEIM}}$ and $\mathbf{k}_{\text{DEIM}}$.

The computational budget and error tolerance are two important factors for design engineers. The computational budget depends on the time allocated to simulate the desired model based on the customer's budget. Error tolerance is also a desired accuracy considered in a specific project. If the computational budget and the error tolerance are defined at the levels depicted in Figure 2.6, it can be said that our POD-DEIM technique is a viable model for designing large models.

$$||\overline{e}||_{L_2} = \frac{||\mathbf{p}^{\text{ROM}} - \mathbf{p}^{\text{FOM}}||_{L_2}}{||\mathbf{p}^{\text{FOM}}||_{L_2}} \tag{2.47}$$

$$||\overline{e}||_{H_1} = \frac{||\mathbf{p}^{\mathrm{ROM}} - \mathbf{p}^{\mathrm{FOM}}||_{H_1}}{||\mathbf{p}^{\mathrm{FOM}}||_{H_1}} \tag{2.48}$$

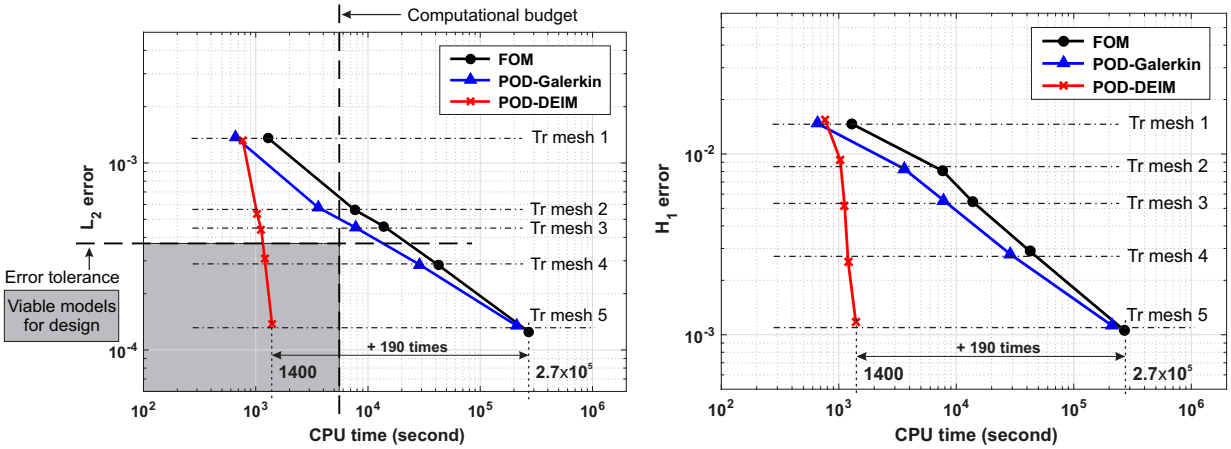
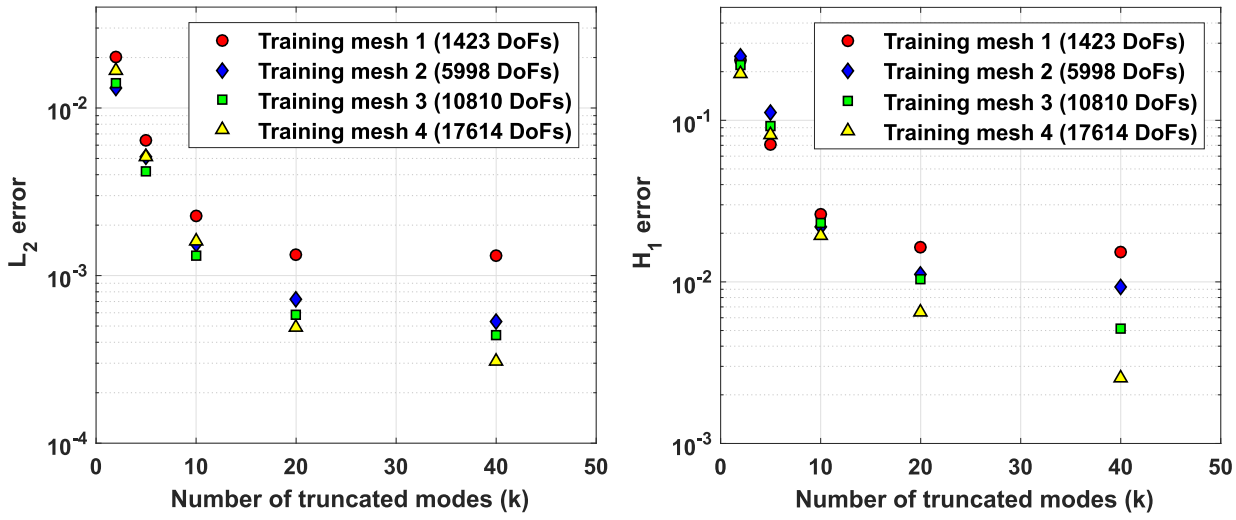



Figure 2.6: a) $L_2$ error versus CPU time and b) $H_1$ error versus CPU time for the various numerical techniques: POD-Galerkin, POD-DEIM and FOM.

In Figure 2.7, a sensitivity analysis is conducted to determine the effect of the number of truncated modes ($k$) on the accuracy of the results. Although increasing the number of truncated modes reduces the $L_2$ and $H_1$ norm errors, the error converges at a certain number of POD modes. For example, in the case of training mesh 1, the $L_2$ and $H_1$ errors are constant for values $k$ greater than 20.

Furthermore, the error tolerance of the design and the size of the training mesh specify the minimum dimensionality required for constructing a ROM. For instance, a ROM designer considers the tolerance error to be $5 \times 10^{-4}$. Hence, for two training meshes, 5998 DoFs and 17614 DoFs, the minimum dimensionality in our MOR technique is 40 and 20, respectively. In comparison to coarse training mesh (5998 DoFs), fine training mesh (17614 DoFs) needs smaller truncated modes since snapshots of fine training mesh for the extraction of the reduced basis functions have higher fidelity (more accurate) than the ones of coarse training mesh.

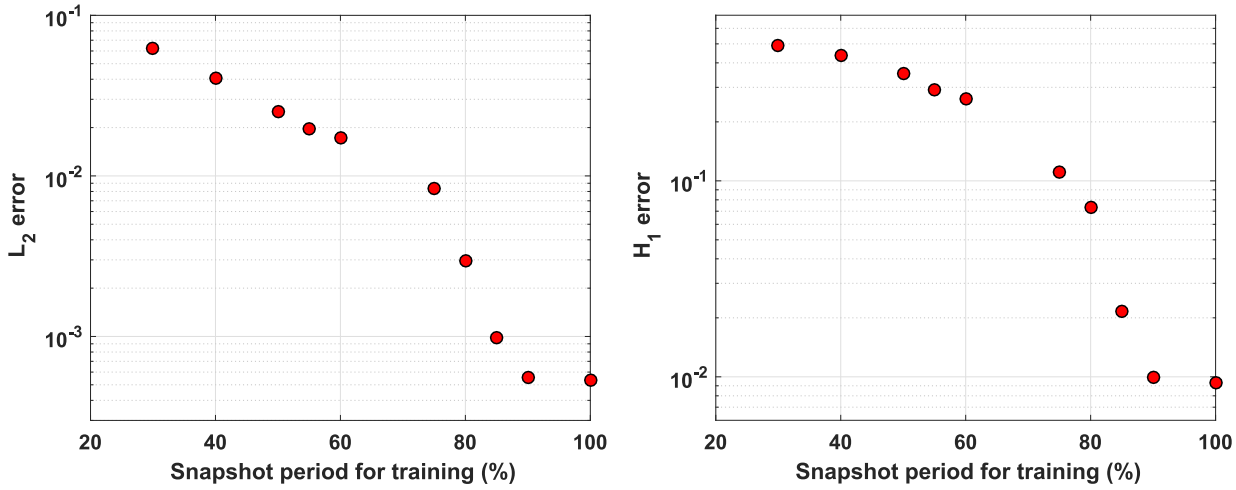



(a) (b)

Figure 2.7: Effect of the number of truncated modes ($k$) on the accuracy of the results. a) $L_2$ norm error and b) $H_1$ norm error.

One important point about the POD approach is the selection of high fidelity snapshots in the training process. In fact, snapshots should have enough physical information about the spatial and temporal evolution of the model. In order to demonstrate the effect of the number of snapshots on the accuracy of the results, $L_2$ and $H_1$ errors are plotted for different periods of snapshots in Figure 2.8. The snapshot period for the training data is the percentage of the data set used to construct the reduced basis functions useful for the projection of the FOM onto the reduced space. It is observable that decreasing the snapshot period reduces the accuracy of the model, driving us to be particular when selecting the snapshots in our ROMs. In addition, the error tolerance of the design plays a crucial role in determining the minimum snapshot period required for constructing a ROM. For instance, it is assumed that the error tolerance of the design is equal to 0.01. According to the Figure 2.8, at least 75% of the full data should be selected as the snapshots to achieve the desired accuracy.

This section compared the accuracy and run-time of the POD-Galerkin, POD-DEIM and FOM for the nonlinear time-dependent system without any parametric inputs. Section 2.4.2 will discuss the behavior of untrained parametric inputs on the performance of the POD-DEIM previously presented.




(a) (b)

Figure 2.8: Role of snapshot period in the training process on the accuracy of the results. a) $L_2$ norm error and b) $H_1$ norm error.

### 2.4.2 Sensitivity Analysis for Parametric ROM

We now evaluate the effectiveness of the presented POD-DEIM for the pumping schedules that fall outside of the training regime to assess the extrapolative ability of the ROM. The injection rate is typically an input parameter, which fluctuates in practice due to operational requirements or constraints, and different time-varying injection rates during the online stage. At the same time, the offline training process is based on constant injection rates. The idea here is to push the limits of the constructed POD-DEIM with a relatively small amount of training data and try to answer the question: how simple can the training data be while still yielding accurate results in the online stage? To this end, the ROM is first trained using 60 full FE simulations with mesh 1 (1423 DoFs) at constant injection rates $Q = 1, \ldots, 60$. Then, samples for pressure and nonlinear force are collected at each time step in matrices $\mathbf{P}_{Q_i}^{\text{SNP}}$ and $\mathbf{F}_{Q_i}^{\text{SNP}}$, respectively. These snapshot matrices are combined into two big matrices $\mathbf{P}^{\text{SNP}}$ and $\mathbf{F}^{\text{SNP}}$ using Algorithm 4. Next, a POD-DEIM is constructed using samples obtained from 60 different sets of injection rate. Finally, the surrogate model is tested through the online state with different time-varying injection rates, as depicted in Figure 2.9.

**Algorithm 4** Build Matrices $\mathbf{P}^{\text{SNP}}$ and $\mathbf{F}^{\text{SNP}}$

**Data:** Matrices $\mathbf{P}^{\text{SNP}}_{Q_i} \in \mathbb{R}^{N\times n_s}$ and $\mathbf{F}^{\text{SNP}}_{Q_i} \in \mathbb{R}^{N\times n_s}$ for $i = 1, ..., 60$
**Result:** Combined snapshot matrices $\mathbf{P}^{\text{SNP}} \in \mathbb{R}^{N\times 60n_s}$ and $\mathbf{F}^{\text{SNP}} \in \mathbb{R}^{N\times 60n_s}$
Initiate the combined snapshot matrices: $\mathbf{P}^{\text{SNP}} = [\,]$ and $\mathbf{F}^{\text{SNP}} = [\,]$
**for** $j = 1, ..., n_s$ **do**
  **for** $i = 1, ..., 60$ **do**
    $\mathbf{P}^{\text{SNP}} = \left[\ \mathbf{P}^{\text{SNP}} \quad \mathbf{P}^{\text{SNP}}_{Q_i}(:, j)\ \right]$
    $\mathbf{F}^{\text{SNP}} = \left[\ \mathbf{F}^{\text{SNP}} \quad \mathbf{F}^{\text{SNP}}_{Q_i}(:, j)\ \right]$

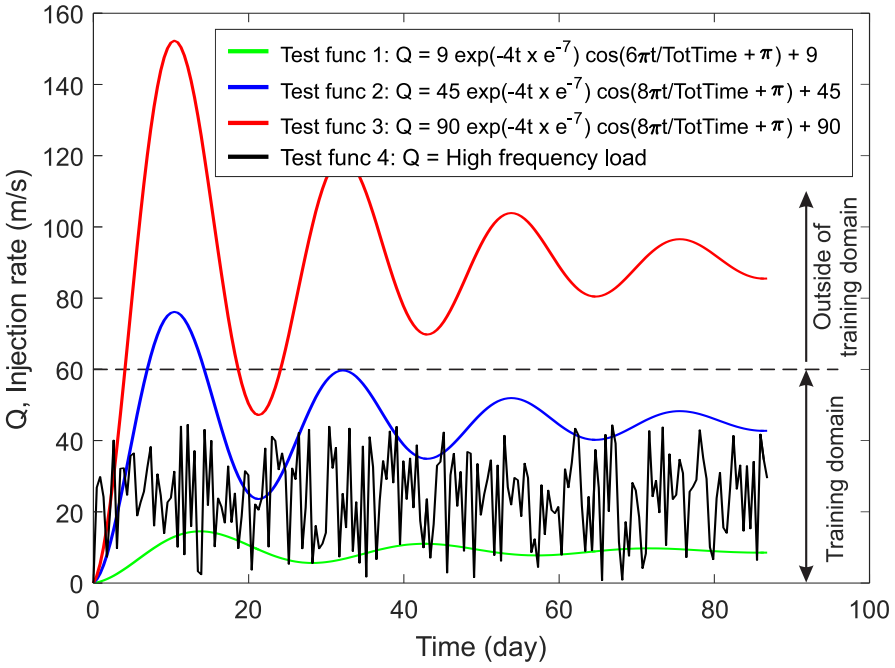



Figure 2.9: Online time-varying injection rates (test functions 1 to 4).

Figure 2.10 represents the pressure and mass flux over time for three test functions 1, 2, and 3. Pressure variations are reported at nodes 16 and 959. In addition, mass flux variations are plotted as the average of mass fluxes at the integration points of elements 262 and 433. In the ROM, the size of the system of equations using DEIM is reduced to $40 \times 40$, where the residual vector is approximated using 400 measurement points. As can be seen, the reduced order model captures the solution of the FOM with acceptable accuracy. It is worth mentioning that test function 3 falls outside of the training injection rates; however,

the POD-DEIM surrogate model provides accurate results for design and optimization purposes.

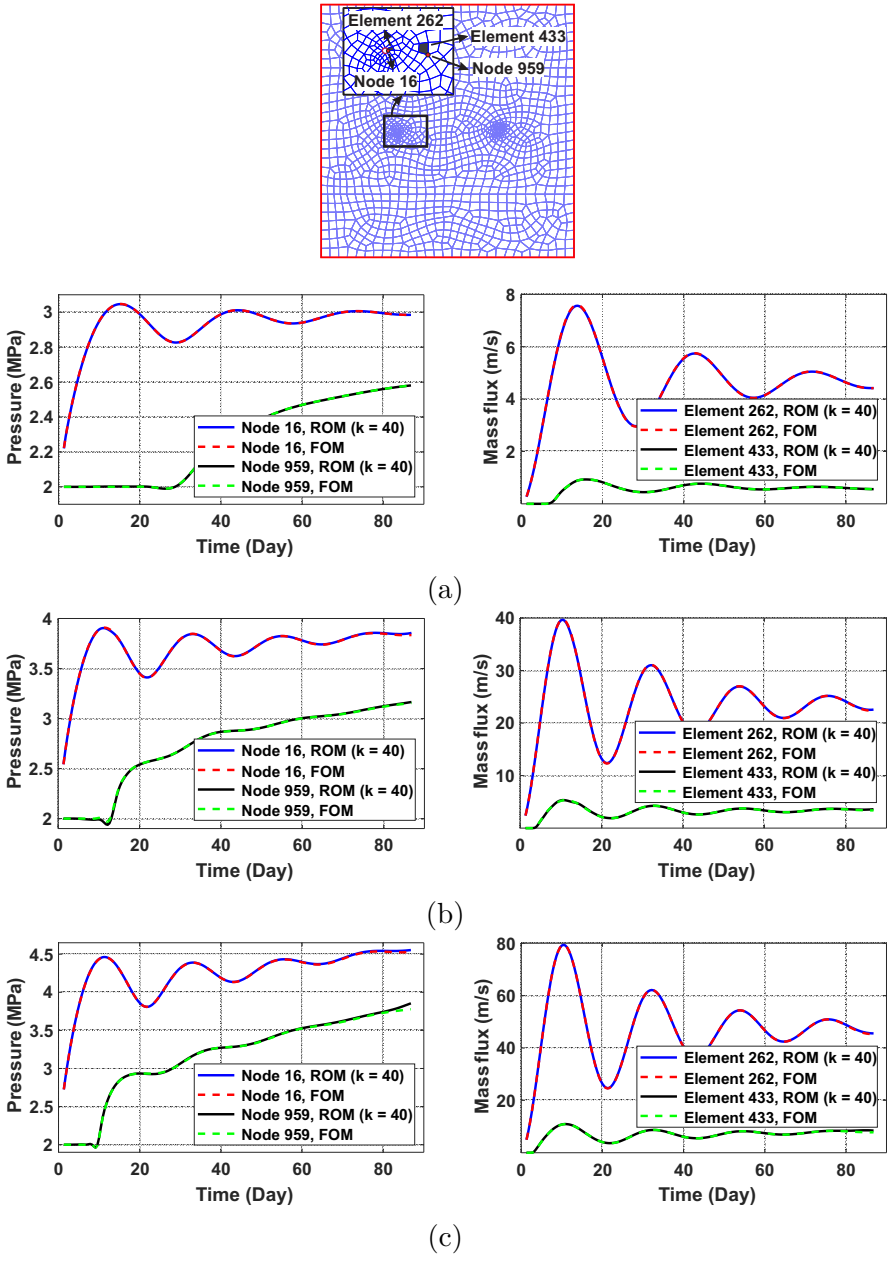
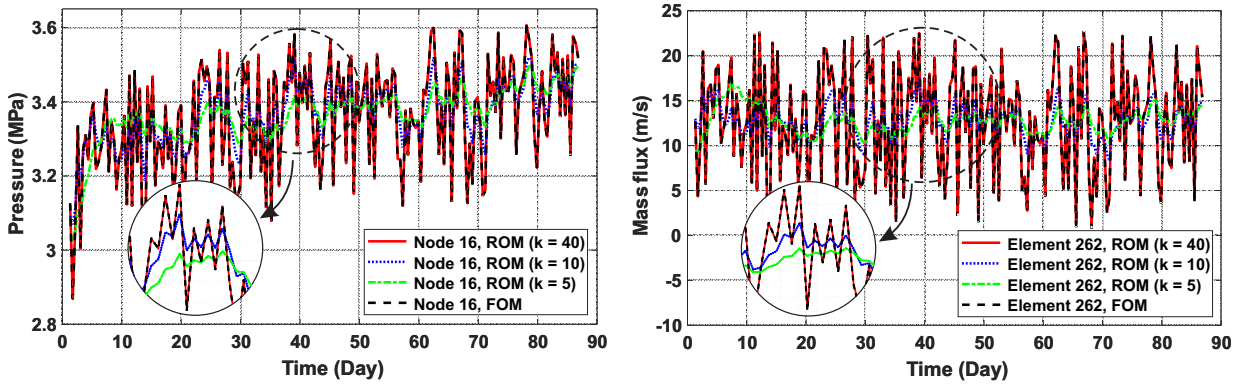



Figure 2.10: Pressure and mass flux over time for a) test function 1, b) test function 2, c) test function 3.

We now investigate the performance of the POD-DEIM for a highly variable injection rate (dark line in Figure 2.9). The aim is to design a complex high-frequency test function to find an answer to the question: how dissimilar can the parametric inputs (e.g., the pumping schedule) of the ROM be from those of the training regime while still ensuring an accurate ROM? Figure 2.11 contrasts the results of the FOM and POD-DEIM for the injection pressure (pressure at node 16) and mass flux (the average of mass fluxes at the integration points of the element 262). The response of ROM is based on three different values for the number of POD basis function ($k = 5, 10, 40$). It should be mentioned that the size of the DEIM basis function is considered 400 for all cases. Figure 2.11 demonstrates that the ROM captures the solution of the FOM with reasonable accuracy with a relatively small amount of ROM basis functions ($k = 40$). Furthermore, it is demonstrated that moderately sized ROMs ($k = 10$) can capture the majority of the dominant modes in the model, where its response closely resembles the FOM. In contrast, in the case of a small set of basis functions (e.g. $k = 5$), the dominant modes cannot adequately capture the response of the system. In the case of $k = 5$, we are missing most of the dominant modes of the model, resulting in larger errors. Hence, it can be concluded that the presented POD-DEIM yields an efficient and accurate ROM, which can capture the mode shapes of unseen complex injection rates.




(a) (b)

Figure 2.11: Comparison of the time history of the POD-DEIM and FOM for a highly variable injection rate. a) injection pressure and b) mass flux.

We finally try to solve a possible applied scenario, where the injection rate acts as a rectangular wave, using the presented ROM. Figure 2.12 illustrates the variations of rectangular waves with regular and irregular time sequences. In load case 1, the fluid is pumped into the wellbore at a constant rate for 22 days, whereby the pumping process is

stopped for the remaining run time. In load case 2, irregular time sequences of pumping are considered while the same maximum injection rate is applied for all time sequences. In load case 3, the pumping schedule alternates between short and long pumping sequences, with periods of no pumping in between. Finally, load case 4 presents the most complex rectangular wave, i.e., a wave with irregular time sequences and varying injection rates.

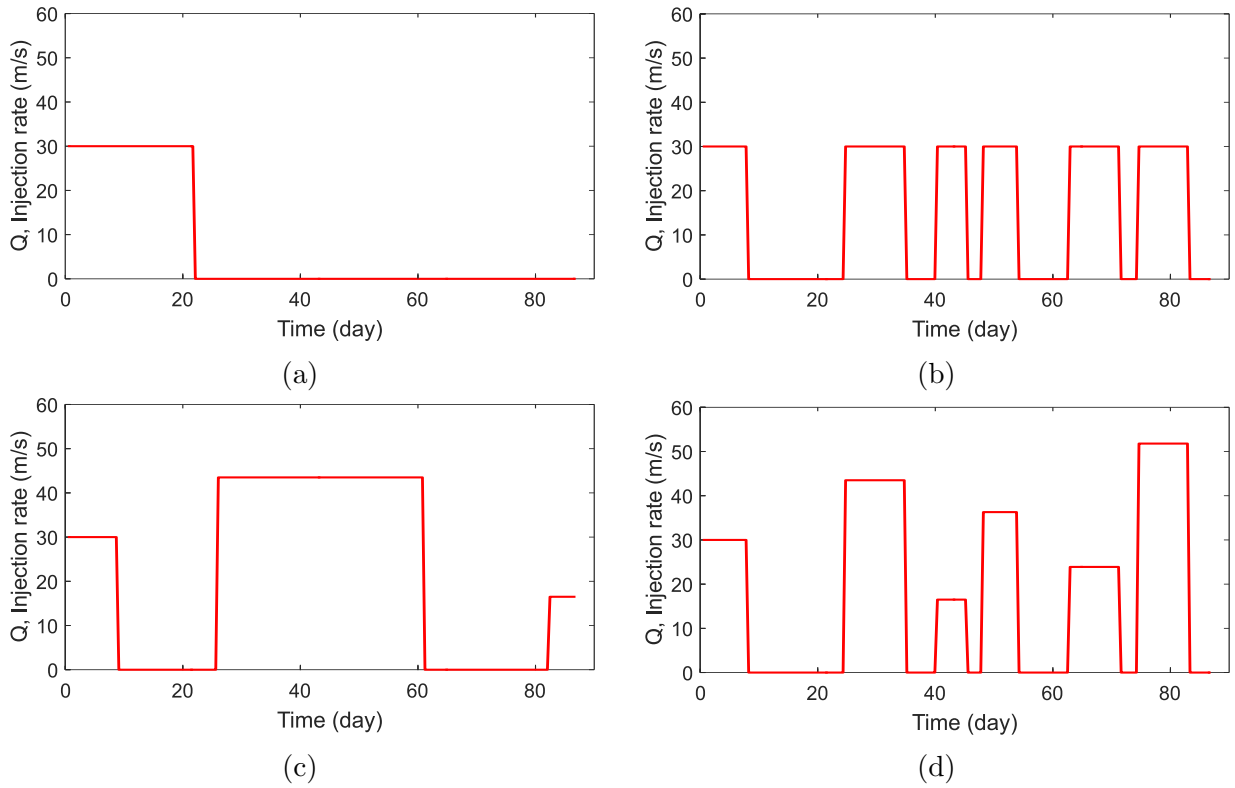
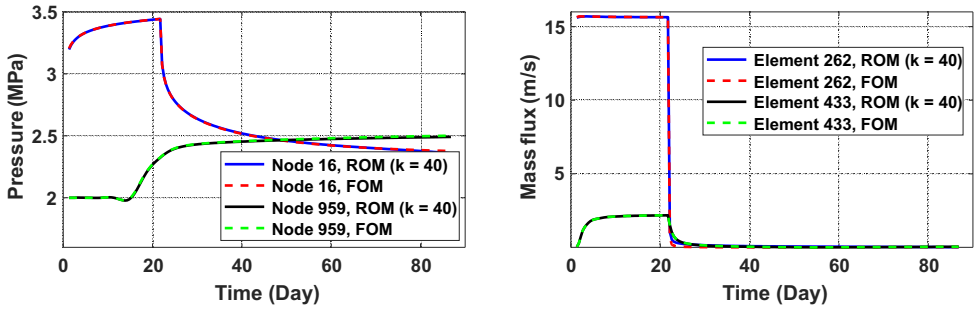
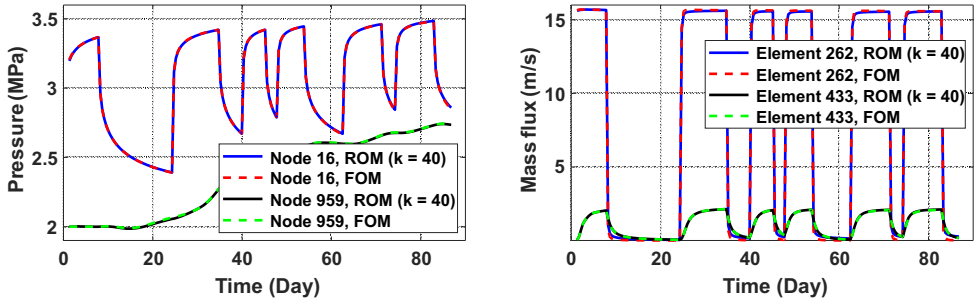



Figure 2.12: Rectangular waves with regular and irregular time sequences as online time-varying injection rates. a) load case 1, b) load case 2, c) load case 3, d) load case 4.

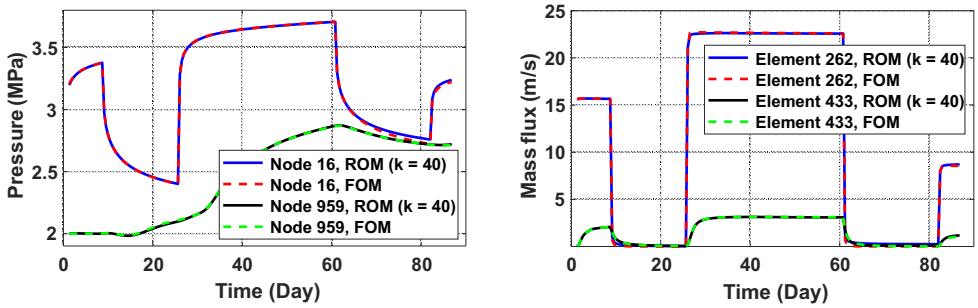
The pressure and mass flux over time for the defined rectangular waves are depicted in Figure 2.13. Our POD-DEIM can effectively and accurately approximate the FOM with highly complex pressure and mass flux behavior. Thus, the proposed model is suitable for use as a surrogate model with sufficient accuracy to calibrate and control the fluid injection pressure, which is a critical factor for preventing waterflooding through natural fractures.



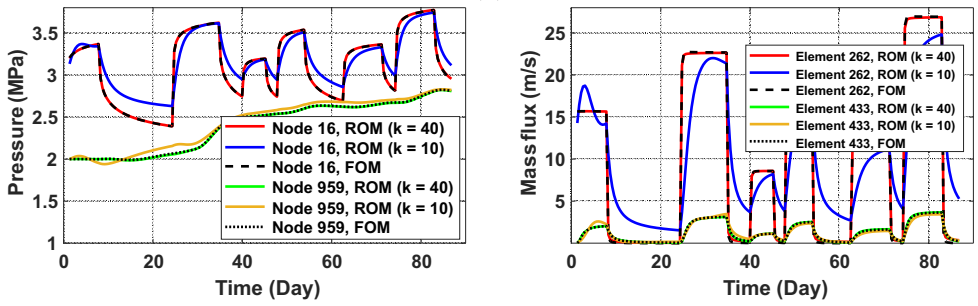

(a)




(b)




(c)




(d)

Figure 2.13: Pressure and mass flux overtime for different rectangular waves. a) results of rectangular wave 1, b) results of wave 2, c) results of wave 3, d) results of wave 4.

## 2.5 Conclusions

In this chapter, an efficient and accurate POD-DEIM ROM is proposed for the application of nonlinear porous media flow problems, particularly in reservoir simulations. POD-Galerkin and DEIM are utilized to reduce the size of the system of equations and interpolate the nonlinear terms, respectively. Column-reduced Jacobian DEIM technique is employed to interpolate the Jacobian, leading to a significant reduction in the computational time of the online stage. POD-DEIM ROM has about $10^3$ times fewer DoFs and is approximately 190 times faster than the FOM for a modest reservoir model with $3 \times 10^4$ DoFs while maintaining an accurate solution in the online stage.

The ROM proposed has been shown to be an effective surrogate model with sufficient accuracy to calibrate and control the fluid pressure during injection operations. The ROM is parameterized for nonlinear transient injection rate (pumping schedule). Offline, training data is generated by FOM runs with simple constant injection rates. Online, the ROMs are demonstrated to have both high accuracy and efficiency for complex and time-varying pumping schedules. These include sinusoidal and high-frequency pumping schedules. Furthermore, the ROM is online tested for a possible applied scenario, time-discontinuous pumping schedules with regular and irregular time sequences. The reservoir responses (pressure and mass flux) demonstrate that the ROM can capture the spatial-temporal evolution of the FOM with reasonable accuracy using a relatively small ROM basis ($k = 10$ and 40). ROMs with a very small basis ($k = 5$) cannot adequately capture the response of the FOM system. Furthermore, POD-Galerkin without DEIM does not yield a ROM with meaningful speed-up over the FOM.

POD-DEIM ROM is demonstrated to have strong potential for reservoir modeling; yet likely requires augmentation for more complex applications incorporating hydro-mechanical coupling behavior and path-dependent plastic deformation of rocks. The next step is to extend the current ROM framework to a plasticity problem, where capturing the history-dependent nonlinear evolution of plastic strain is a big challenge.

# Chapter 3

# Parameterized ROM for Drucker–Prager Plasticity

## 3.1 Introduction

Chapter 3 will focus on the development of model order reduction (MOR) technique for a two-dimensional system of nonlinear equations equipped with pressure-dependent plasticity constitutive model. Drucker–Prager plasticity model is considered for oil sand reservoir, and the challenge is to accurately capture plastic deformation around the wells using reduced order model (ROM).

Despite continued advancements in computing hardware, the simulation of plastic deformation of solids and structures remains computationally expensive as it involves solving a large system of nonlinear equations. The computationally intensive nature of these simulations continues to pose challenges in parameter estimation, uncertainty quantification, and optimization applications, where hundreds to thousands of simulations are required to achieve a solution [70, 71]. Additionally, in real-time applications, full physics simulation of plasticity by finite element analysis is often impractical. MOR techniques have emerged as a method to derive and train a computationally efficient surrogate/proxy model through the use of the full order model (FOM). In this chapter, the FOM is the solution to the system of discrete nonlinear equations derived through the finite element method (FEM) for the plastic deformation of a solid domain. The development of ROMs for cyclic plasticity requires new MOR techniques due to the path-dependent and nonlinear evolution of plastic strain, which is the focus of this chapter.

Recent research has made considerable efforts in the development of intrusive and non-

intrusive ROMs involving nonlinear and path-dependent constitutive laws, particularly plasticity and viscoelasticity [30, 72, 73, 74, 75, 76]. Gorji *et al.* generated artificial neural network-based (ANN-based) constitutive modeling for plasticity using recurrent neural network (RNN) [72]. Chen investigated the extrapolation ability of RNN in viscoelasticity constitutive laws [73]. Ghavamian *et al.* developed an intrusive ROM for the Perzyna viscoplastic model when the ROM is parameterized by only one material property [30]. In their case, the load path is not a parametric input for ROM. Intrusive ROMs have faced significant challenges in learning the path-dependent nonlinear evolution of plastic strain as a function of a loading path. This chapter develops and evaluates an intrusive ROM for a Drucker–Prager plasticity model, in which the material properties and the cyclic load path are parametric inputs. By constructing multiple local DEIM (LDEIM) approximations in combination with clustering and classifier techniques, a fast and accurate ROM can be achieved [60]. It is shown that the standard DEIM approximation of the FOM residual vector and Jacobian is inefficient for constructing accurate ROMs. The FOM response is path-dependent and strongly nonlinear as described in Section 3.2. Further, the online computational cost of LDEIM is substantially improved as each local DEIM requires fewer measurement points than the traditional DEIM. Section 3.3 discusses these concepts in more detail.

Each ROM is parameterized by several hyper-parameters (tuning parameters), which influence both the accuracy and computational efficiency of the ROM. The user must specify these parameters, as they are not determined directly in the training algorithm. The influence of hyper-parameters will be explored in several examples in Section 3.4. This chapter will demonstrate the effectiveness of the ROMs based on LDEIM when parameterized by material properties and loading paths for Drucker–Prager plasticity problems.

## 3.2 Full Order Model

In this chapter, the FOM consists of a finite element analysis of the elastic-plastic deformation of a two-dimensional solid domain, where loads are applied incrementally. At each step, a large nonlinear system of equations with $n$ degrees of freedom (DoFs) must be solved for unknown increments in the nodal displacements.

Consider the two-dimensional FEM approximation of the displacement of a solid domain $\Omega$:

$$\mathbf{u}^h(\mathbf{x},\tau) = \sum_{I=1}^{n_d} N_I(\mathbf{x})\,\mathbf{u}_I(\tau) = \mathbf{N}\mathbf{u}^\tau,\ \mathbf{x}\in\Omega \tag{3.1}$$

in which $\mathbf{x}$ denotes material points and $\tau$ denotes pseudo-time. $n_d$ is the number of nodes of the FE mesh, $N_I$ are the shape functions, $\mathbf{u}_I$ are two-dimensional nodal displacements, $\mathbf{N}$ is a $2 \times n$ matrix of shape functions, and $\mathbf{u}^\tau$ is a $n \times 1$ column vector of nodal displacements at load step $\tau$. The FOM has $n = 2 \times n_d$ DoFs for two-dimensional problems.

Assuming small elastoplastic deformations, a system of $n$ nonlinear discrete equilibrium equations can be derived for the solution of the nodal displacements $\mathbf{u}^\tau$ at load step $\tau$ given the solution $\mathbf{u}^{\tau-1}$ at load step $\tau - 1$:

$$\mathbf{R}\left(\mathbf{u}^\tau\right) = \mathbf{F}_{\text{ext}}\left(\tau\right) - \mathbf{F}_{\text{int}}\left(\mathbf{u}^\tau\right) = \mathbf{0} \tag{3.2}$$

in which $\mathbf{F}_{\text{ext}}$ is the external force vector

$$\mathbf{F}_{\text{ext}}\left(\tau\right) = \int_{\Gamma_t} \mathbf{N}^\top \bar{\mathbf{t}}\left(\tau\right) \mathrm{d}\Gamma + \int_\Omega \mathbf{N}^\top \mathbf{f}_b \mathrm{d}\Omega \tag{3.3}$$

and $\mathbf{F}_{\text{int}}$ is the internal force vector

$$\mathbf{F}_{\text{int}}\left(\mathbf{u}^\tau\right) = \int_\Omega \mathbf{B}^\top \boldsymbol{\sigma}\left(\mathbf{u}^\tau\right) \mathrm{d}\Omega \tag{3.4}$$

in which $\boldsymbol{\sigma}$ is the Cauchy stress in Voigt (matrix) notation and $\mathbf{f}_b$ is the body force. $\Gamma_t$ is the portion of the boundary where traction $\bar{\mathbf{t}}$ is prescribed. The outward pointing normal vector to $\Gamma_t$ is denoted as $\mathbf{n}_t$. $\mathbf{B}$ is the standard $3 \times n$ B-matrix of shape functions derivatives.

### 3.2.1 Drucker–Prager Elastoplasticity Model

The Drucker–Prager yield criterion is a widely used mathematical model that describes the plastic behavior of materials, particularly in the field of geotechnical engineering [77, 78]. It is a modified version of the von Mises criterion that accounts for pressure sensitivity and dilation. Dilation refers to the increase in volume when a material is subject to shear stress in the presence of compressive stresses.

According to the Drucker–Prager yield criterion, plastic yielding starts when a critical combination of the hydrostatic stress $p$ and the $J_2$ invariant of the deviatoric stress $\mathbf{s}$ is attained, specifically:

$$\Phi\left(\boldsymbol{\sigma}, c\right) = \sqrt{J_2\left(\mathbf{s}\left(\boldsymbol{\sigma}\right)\right)} + \eta p\left(\boldsymbol{\sigma}\right) - \xi c = 0 \tag{3.5}$$

where $c$ is the cohesion, the stress level at which a material transitions from the elastic to the plastic deformation regime. $\mathbf{s}$ is the deviatoric stress. The parameters $\eta$ and $\xi$ can be selected according to the approximation required to simplify the Mohr–Coulomb yield surface (compression cone, extension cone, and plane strain condition)[79]:

$$\eta = \frac{3\tan\phi}{\sqrt{9+12\tan^2\phi}} \qquad \xi = \frac{3}{\sqrt{9+12\tan^2\phi}} \tag{3.6}$$

in which $\phi$ denotes the friction angle. The linear hardening curve is defined as

$$c\left(\bar{\varepsilon}^p\right) = c_0 + H\bar{\varepsilon}^p \tag{3.7}$$

$\bar{\varepsilon}^p$ is the accumulated plastic strain. $c_0$ and $H$ represent the initial cohesion and the hardening modulus, respectively.

The details of the constitutive equation are given in Appendix A. The small deformation assumption is used so the total strain tensor can be additively decomposed into elastic and plastic parts. To solve the elastoplastic constitutive initial value problem (A.2, A.3, A.6), the elastic predictor/plastic corrector algorithm is used [79].

### 3.2.2 Newton–Raphson Solution to the Nonlinear System of Equations

The solution of (3.2) for the displacements at load step $\tau$ is sought iteratively using a Newton–Raphson solver. At each iteration $j$, a large $n \times n$ linear system of equations is solved for the increment of the solution (nodal displacements) $\Delta\mathbf{u}^{\tau(j)}$:

$$\mathbf{K}^{\tau(j)}\Delta\mathbf{u}^{\tau(j)} = \mathbf{R}^{\tau(j)} \tag{3.8}$$

and the solution is updated according to

$$\mathbf{u}^{\tau(j+1)} = \mathbf{u}^{\tau(j)} + \Delta\mathbf{u}^{\tau(j)} \tag{3.9}$$

where $\mathbf{R}^{\tau(j)} = \mathbf{R}\left(\mathbf{u}^{\tau(j)}\right)$ and $\mathbf{u}^{\tau(j)}$ are the residual vector and the nodal displacement vector at load step $\tau$ and iteration $j$, respectively. $\mathbf{K}^{\tau(j)}$ is the Jacobian matrix, defined as

$$\mathbf{K}^{\tau(j)} = \mathbf{K}\left(\mathbf{u}^{\tau(j)}\right) = -\left.\frac{\partial\mathbf{R}^{\tau(j)}}{\partial\mathbf{u}}\right|_{\mathbf{u}^{\tau(j)}} = \int_\Omega \mathbf{B}^\top\mathbf{D}^{ep}\mathbf{B}\mathrm{d}\Omega \tag{3.10}$$

$\mathbf{D}^{ep}$ is the consistent (algorithmic) tangent modulus in elastoplasticity. It is a function of $\mathbf{u}^{\tau(j)}$ and the plastic strain $\boldsymbol{\varepsilon}^{p,\tau(j)}$, which is defined in Appendix A [79, 80]. The FOM is based on a classical finite element implementation, further details of which are found in numerous textbooks [79, 81, 82]. The FOM is computationally expensive due to the cost of both assembly and solution (matrix inversion) of the large $n \times n$ system of equations (3.8) at each iteration. An efficient ROM must reduce these costs. The construction of such a ROM is described in the next section.

### 3.2.3 Linear-Nonlinear Decomposition

In this study, we adopt a linear-nonlinear decomposition as described by Ghavamian *et al.* [30]. They found that a more efficient ROM can be achieved by applying hyper-reduction solely to the nonlinear components of the residual vector and its Jacobian rather than to the full residual vector. To this end, the residual vector is decomposed into linear and nonlinear parts as follows:

$$\mathbf{R} = \mathbf{F}_{\text{ext}} - \mathbf{K}_{\text{L}}\mathbf{u} - \mathbf{F}(\mathbf{u}) \tag{3.11}$$

where $\mathbf{F}_{\text{ext}} - \mathbf{K}_{\text{L}}\mathbf{u}$ is the linear part. $\mathbf{F}(\mathbf{u})$, called the force vector, is the nonlinear part of the residual vector

$$\mathbf{F}(\mathbf{u}) = \int_{\Omega} \mathbf{B}^{\top}\boldsymbol{\sigma}(\mathbf{u})\,\mathrm{d}\Omega - \mathbf{K}_{\text{L}}\mathbf{u} \tag{3.12}$$

in which the linear elastic stiffness matrix is

$$\mathbf{K}_{\text{L}} = \int_{\Omega} \mathbf{B}^{\top}\mathbf{D}^{e}\mathbf{B}\mathrm{d}\Omega \tag{3.13}$$

Similarly, the Jacobian matrix (3.10) is decomposed into linear and nonlinear components:

$$\mathbf{K}(\mathbf{u}) = \mathbf{K}_{\text{L}} + \mathbf{K}_{\text{NL}}(\mathbf{u}) \tag{3.14}$$

in which $\mathbf{K}_{\text{NL}}(\mathbf{u})$ is the nonlinear part of Jacobian matrix and is defined as

$$\mathbf{K}_{\text{NL}}(\mathbf{u}) = \frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \int_{\Omega} \mathbf{B}^{\top}\left(\mathbf{D}^{ep}(\mathbf{u}) - \mathbf{D}^{e}\right)\mathbf{B}\mathrm{d}\Omega \tag{3.15}$$

## 3.3 Intrusive Reduced Order Model

This section describes the methods used to construct the intrusive ROM for plasticity. Multiple techniques are used to reduce the computational complexity of the ROM while minimizing the loss between the FOM and the ROM. Specifically, i) the POD-Galerkin technique is utilized to project the FOM with $n$ DoFs onto a reduced space of $m$ DoFs ($m \ll n$); ii) sparse sampling and interpolation (i.e., DEIM) are employed to reduce the complexity of evaluating nonlinear components of the force vector $\mathbf{F}$ and its Jacobian $\mathbf{K}_{\text{NL}}$ to be proportional to the number of measurement points $q$ ($q \approx m$ and $q \ll n$); iii) multiple local DEIMs are constructed, each of them characterizes a distinct region of the FOM response. Offline, regions are identified by clustering the training data. Online, a classification algorithm identifies the appropriate cluster/DEIM approximation based on the current state of the ROM.

### 3.3.1 Proper Orthogonal Decomposition-Galerkin Projection

We utilize the Galerkin projection method in conjunction with POD to reduce the dimensionality of the FOM nonlinear equilibrium equation (3.8) from $n$ to $m$ ($m \ll n$). Let $\mathbf{W}$ denote the matrix which projects the DOFs of a reduced order space $\widetilde{\mathbf{u}}$ at load step $\tau$ and iteration $j$ onto the DoFs of the FOM:

$$\mathbf{u}^{\tau(j)} = \mathbf{W}\widetilde{\mathbf{u}}^{\tau(j)} \tag{3.16}$$

in which $\widetilde{\mathbf{u}} \in \mathbb{R}^m$ and $\mathbf{u} \in \mathbb{R}^n$. $\mathbf{W}$ will be defined in detail later in this section.

The discrete equations of the ROM are constructed by multiplying the FOM equations (3.8) with $\mathbf{W}^\top$ and subsequently inserting (3.16) into (3.8) yielding

$$\widetilde{\mathbf{K}}^{\tau(j)}\Delta\widetilde{\mathbf{u}}^{\tau(j)} = \widetilde{\mathbf{R}}^{\tau(j)} \tag{3.17}$$

$$\widetilde{\mathbf{u}}^{\tau(j+1)} = \widetilde{\mathbf{u}}^{\tau(j)} + \Delta\widetilde{\mathbf{u}}^{\tau(j)} \tag{3.18}$$

in which $\widetilde{\mathbf{K}}^{\tau(j)} \in \mathbb{R}^{m\times m}$ represents the reduced Jacobian matrix and $\widetilde{\mathbf{R}}^{\tau(j)} \in \mathbb{R}^m$ represents the reduced residual vector :

$$\widetilde{\mathbf{K}}^{\tau(j)} = \mathbf{W}^\top\mathbf{K}^{\tau(j)}\mathbf{W} = \underbrace{\mathbf{W}^\top\mathbf{K}_{\text{L}}\mathbf{W}}_{\widetilde{\mathbf{K}}_{\text{L}}} + \underbrace{\mathbf{W}^\top\mathbf{K}_{\text{NL}}^{\tau(j)}\mathbf{W}}_{\widetilde{\mathbf{K}}_{\text{NL}}^{\tau(j)}} \tag{3.19}$$

$$\widetilde{\mathbf{R}}^{\tau(j)} = \mathbf{W}^\top \mathbf{R}^{\tau(j)} = \underbrace{\mathbf{W}^\top \mathbf{F}_{\text{ext}}^{\tau(j)}}_{\widetilde{\mathbf{F}}_{\text{ext}}^{\tau(j)}} - \widetilde{\mathbf{K}}_{\text{L}} \widetilde{\mathbf{u}}^{\tau(j)} - \underbrace{\mathbf{W}^\top \mathbf{F}^{\tau(j)}}_{\widetilde{\mathbf{F}}^{\tau(j)}} \tag{3.20}$$

in which the linear-nonlinear decompositions (3.11) and (3.14) have been used. The projection matrix $\mathbf{W}$ and reduced linear part of the Jacobian $\widetilde{\mathbf{K}}_{\text{L}} \in \mathbb{R}^{m \times m}$ are computed once offline. The nonlinear terms $\widetilde{\mathbf{F}}^{\tau(j)}$ and $\widetilde{\mathbf{K}}_{\text{NL}}^{\tau(j)}$ must be computed online at each iteration of the nonlinear solver. Since ROM quantities $\widetilde{\mathbf{F}}^{\tau(j)} \in \mathbb{R}^m$ and $\widetilde{\mathbf{K}}_{\text{NL}}^{\tau(j)} \in \mathbb{R}^{m \times m}$ are dependent on the corresponding the FOM quantities, the computational cost of their assembly scales with $n$. Methods to reduce the cost to $q$ ($q \approx m$ and $q \ll n$) are discussed in Section 3.3.2.

The projection matrix $\mathbf{W}$ is computed using snapshots of the high-fidelity FOM simulations conducted offline, with inputs sampled from the parameter space where the ROM is to operate. Samples of the FOM displacement vector $\mathbf{u} \in \mathbb{R}^n$ are collected at the end of each load step and assembled into the snapshot matrix $\mathbf{U}^{\text{SNP}}$, as

$$\mathbf{U}^{\text{SNP}} = \{\mathbf{u}_1, ..., \mathbf{u}_{n_s}\} \in \mathbb{R}^{n \times n_s} \tag{3.21}$$

where $n_s$ is the number of snapshots. Ideally, $n_s$ is chosen to minimize the offline training cost of the FOM executions, yet large enough to span the domain of the online use of the ROM.

Singular Value Decomposition (SVD) (2.36) is performed on $\mathbf{U}^{\text{SNP}}$ to extract the projection matrix $\mathbf{W}$. The columns of the projection matrix $\mathbf{W}$ are taken as the first $m$ left singular vectors $\mathbf{w}_i$, $\mathbf{W} = \{\mathbf{w}_1, ..., \mathbf{w}_m\} \in \mathbb{R}^{n \times m}$, where $m \ll n$.

### 3.3.2 Discrete Empirical Interpolation Method

POD-Galerkin without hyper-reduction is an effective approach for reducing the computational cost of ROMs for linear elastic problems. However, it is not efficient for plasticity problems due to the presence of the nonlinear terms $\mathbf{F}$ and $\mathbf{K}_{\text{NL}}$ in (3.19) and (3.20), which must be recomputed at every iteration of the Newton–Raphson algorithm. Using DEIM [39, 51], $\mathbf{F}$ and $\mathbf{K}_{\text{NL}}$ are replaced by the approximations based on interpolation and sparse sampling. Here, we summarize the DEIM approximation. The details were discussed in Section 2.3.2.

The DEIM approximation of $\mathbf{F}$, denoted as $\hat{\mathbf{F}}$, is determined by a linear map from $\mathbf{f}_{\text{DEIM}}$:

$$\hat{\mathbf{F}} = \mathbf{A}\mathbf{f}_{\text{DEIM}} \in \mathbb{R}^n \tag{3.22}$$

where

$$\mathbf{A} = \mathbf{Z}(\mathbf{P}^\top \mathbf{Z})^{-1} \tag{3.23}$$

$$\mathbf{f}_{\text{DEIM}} = \mathbf{P}^\top \mathbf{F} \tag{3.24}$$

In (3.23) and (3.24), the truncated POD basis $\mathbf{Z}$ ($\mathbf{Z} = [\mathbf{z}_1 \ldots . \mathbf{z}_q] \in \mathbb{R}^{n \times q}$, $q \ll n$) and the measurement matrix $\mathbf{P}$ were defined in Section 2.3.2. $q$ are measurement points determined using a greedy algorithm [69, 39, 51].

The matrix $\mathbf{A}$ is computed once offline. Online, the computation of $\mathbf{F}$ is replaced by the cheaper $\hat{\mathbf{F}}$, which only involves the evaluation of a small number of rows of $\mathbf{F}$, i.e., $\mathbf{f}_{\text{DEIM}}$. The cost of the evaluation of $\hat{\mathbf{F}}$ scales with $q$ which in practice is similar to the ROM size $m$.

In this chapter, we adopt the first method, as detailed in Section 2.3.2, to compute the nonlinear term $\mathbf{K}_{\text{NL}}$ using the same POD basis vectors that are determined for the term $\mathbf{F}$ [30, 51].

The computationally efficient form of the reduced order system of equations is

$$\widetilde{\mathbf{R}} = \underbrace{\widetilde{\mathbf{F}}_{\text{ext}}}_{m \times 1} - \underbrace{\widetilde{\mathbf{K}}_{\text{L}} \widetilde{\mathbf{u}}}_{m \times 1} - \underbrace{\mathbf{A}_w}_{m \times q} \underbrace{\mathbf{f}_{\text{DEIM}}}_{q \times 1} = \mathbf{0} \tag{3.25}$$

$$\widetilde{\mathbf{K}} = \underbrace{\widetilde{\mathbf{K}}_{\text{L}}}_{m \times m} + \underbrace{\mathbf{A}_w}_{m \times q} \underbrace{\mathbf{k}_{\text{DEIM}} \mathbf{W}}_{q \times m} \tag{3.26}$$

in which $\mathbf{A}_w = \mathbf{W}^\top \mathbf{A}$ and is computed offline. It can be shown that $\widetilde{\mathbf{K}}$ corresponds to the Jacobian matrix for the first-order Taylor series approximation of the reduced system $\widetilde{\mathbf{R}}$.

The DEIM technique offers notable computational cost savings, particularly in scenarios where $q$ is much smaller than $n$ ($q \ll n$). However, in many nonlinear plasticity problems, accurate estimation of $\mathbf{F}$ and $\mathbf{K}_{\text{NL}}$ necessitates a large number of measurement points ($q$), as the POD singular values exhibit slow decay. In such scenarios, utilizing small values for $q$ leads to ROMs that are very inaccurate compared to the FOM or that are unstable because the Newton–Raphson solver fails to converge. We can employ multiple local DEIM (LDEIM) subspaces instead of a single global DEIM subspace to address this limitation. The LDEIM implementation is explained in the next section.

### 3.3.3 Localized Discrete Empirical Interpolation Method

LDEIM follows an offline-online decomposition process. During the offline phase, the force sample matrix $\mathbf{F}^{\text{SNP}}$ is partitioned into multiple clusters, and DEIM is applied to each cluster individually to compute local DEIM interpolation matrix

$$\mathbf{A}_c = \mathbf{Z}_c(\mathbf{P}_c^\top \mathbf{Z}_c)^{-1} \tag{3.27}$$

in which $\mathbf{Z}_c$ and $\mathbf{P}_c$ denote the POD basis vectors and measurement point matrix associated with the $c^{th}$ cluster, respectively. LDEIM requires fewer measurement points for each cluster compared to global DEIM, as the energy associated with basis vectors for each cluster decays significantly faster than those for unclustered samples. Online, the most appropriate cluster is selected based on the current state of the system. The online phase in LDEIM is faster compared to DEIM because it involves a small number of measurement points for each cluster.

Two main questions must be addressed to develop a ROM with LDEIM: i) How can the force samples be divided into clusters during the offline phase?; ii) How can the ideal cluster be effectively determined during the online phase? There are different methods for clustering the samples [60, 83]. In this study, offline the $k$-means clustering algorithm is employed to divide the snapshot matrix $\mathbf{F}^{\text{SNP}}$ into clusters [60], while online the nearest neighbor classifier algorithm is used to identify the appropriate cluster [60].

#### $k$-Means Clustering Algorithm for LDEIM

The $k$-means clustering algorithm is a widely unsupervised machine learning technique that partitions a given dataset into a specified number of distinct clusters, denoted as $n_c$ [84]. In the standard $k$-means clustering algorithm, the cluster labels are initialized for each data point from the dataset, where cluster labels are updated through an iterative process by calculating the distance between each data point and each centroid (i.e., the center of cluster). The centroid is defined as the mean of the samples allocated to the cluster. Peherstorfer *et al.* [60] tailored the standard $k$-means clustering algorithm for DEIM by improving the definition of the distance. In the proposed $k$-means clustering algorithm, the distance is defined between a data point and its local DEIM approximation. By applying the modified $k$-means clustering algorithm to the snapshots resulting from elastoplasticity model, we can effectively group data points based on their similarity in loading paths and material properties, allowing us to uncover patterns and transitions within the material response, enabling better estimation of its complex behavior.

The objective is to partition the nonlinear force sample matrix $\mathbf{F}^{\text{SNP}}$ into mutually exclusive subsets $\mathbf{F}_1^{\text{SNP}}, ..., \mathbf{F}_{n_c}^{\text{SNP}}$ in the offline phase. Subsequently, for each cluster $\mathbf{F}_c^{\text{SNP}}$, a local DEIM interpolation matrix $\mathbf{A}_c$ is constructed. Furthermore, a classifier function is trained to designate the mapping between the sample $\mathbf{F}_i$ and cluster number $c$. In the algorithm, the input is the set $\mathbf{F}^{\text{SNP}}$ and the outputs are the classifier function and $n_c$ matrices $\mathbf{A}_c$. In this thesis, non-overlapping clusters are used. A more advanced method is to use overlapping clusters to make the approximation more robust at the cluster interfaces [85]. This investigation will be of interest in future studies.

The first step is to initialize the cluster labels for each sample force. The samples in the dataset are divided into $n_c$ clusters, and the classifier function **cls** is defined as a simple $n_s \times 1$ array that maps snapshots to clusters. In most applications, it is crucial to have an appropriate initial guess, as the $k$-means algorithm does not guarantee convergence to a global minimum. However, in this work, we found that a random initial guess for clusters works well. To ensure optimal clustering results, the $k$-means algorithm is performed multiple times with different initial conditions, and the clusters that make the minimum error for ROM are selected.

The second step involves the iterative assignment. The distance between each sample force $\mathbf{F}_i$ and its DEIM approximation $\hat{\mathbf{F}}_{i,c} = \mathbf{A}_c \mathbf{P}_c^\top \mathbf{F}_i$ is calculated using the Euclidean distance and cosine similarity

$$D = \sqrt{D_{\text{cosine}}^2 + D_{\text{Euclidean}}^2} \tag{3.28}$$

$$D_{\text{cosine}} = 1 - \frac{\left| \mathbf{F}_i^\top \hat{\mathbf{F}}_{i,c} \right|}{\left\| \mathbf{F}_i \right\| \left\| \hat{\mathbf{F}}_{i,c} \right\|} \tag{3.29}$$

$$D_{\text{Euclidean}} = \frac{\left\| \mathbf{F}_i - \hat{\mathbf{F}}_{i,c} \right\|}{\left\| \mathbf{F}_i \right\|} \tag{3.30}$$

The benefits of clustering using Euclidean distance and cosine similarity metrics are discussed in [30, 86]. The cosine similarity metric is more effective at capturing similarities in the orientations and features of high-dimensional data sets as it measures the cosine of the angle between vectors. In contrast, the Euclidean distance measures the absolute magnitude between points.

Next, for each sample force, we determine the cluster that provides the most accurate DEIM approximation by finding the minimum distance between $\mathbf{F}_i$ and $\hat{\mathbf{F}}_{i,c}$. Finally, the classifier function **cls** and the local DEIM interpolation matrix $\mathbf{A}_c$ are updated. This process is repeated until the convergence criterion is met. Convergence is achieved when the classifier function remains unchanged or the maximum number of iterations has been reached. A summary of the $k$-means clustering algorithm is presented in Algorithm 1.

**Algorithm 1:** $k$-means clustering algorithm

**Data:** Snapshot matrix for nonlinear force vector $\mathbf{F} : \mathbf{F}^{\text{SNP}}$
Maximum number of iterations: $M_I$
Number of clusters: $n_c$

**Result:** Local DEIM approximations for each cluster $c = 1, ..., n_c$: $\mathbf{P}_c$, $\mathbf{Z}_c$, $\mathbf{A}_c$
Classifier function **cls** : sample $\mathbf{F}_i \rightarrow$ cluster number $c$

Set convergence index: $conv = 1$
Initiate classifier function **cls** according to the initial guess

**for** $j = 1 : M_I$ **do**
  **for** $c = 1 : n_c$ **do**
    Calculate POD basis vectors: $\mathbf{Z}_c$
    Calculate measurement matrix $\mathbf{P}_c$ using the greedy algorithm [69, 39, 51]
    Calculate $\mathbf{A}_c$ using (3.27)
    **for** $i = 1 : \text{size}\left(\mathbf{F}^{\text{SNP}}, 2\right)$ **do**
      Calculate the DEIM approximation to the force sample $\mathbf{F}_i$: $\hat{\mathbf{F}}_{i,c}$
      Calculate the distance between $\mathbf{F}_i$ and $\hat{\mathbf{F}}_{i,c}$ using (3.28)
      Save the distance in $\mathbf{D}\,(i, c)$
  **for** $i = 1 : \text{size}\left(\mathbf{F}^{\text{SNP}}, 2\right)$ **do**
    Determine the old cluster number: $c_{\text{old}} = \mathbf{cls}\,(i)$
    Calculate the new cluster by finding the minimum distance between $\mathbf{F}_i$ and $\hat{\mathbf{F}}_{i,c}$: $c_{\text{new}} = \min(\mathbf{D}\,(i, :))$
    **if** $c_{\text{old}} \neq c_{\text{new}}$ **then**
      $conv = 0$
      Update the classifier function: $\mathbf{cls}\,(i) = c_{\text{new}}$
  **if** $conv = 1$ **then**
    Break (terminate the execution of for loop on $j$)

### Nearest Neighbor Classifier Algorithm

In the online phase, the nearest neighbor classifier is utilized to identify the most suitable cluster based on the current state of the system. The classifier operates on the low-dimensional representation of the data instead of using the full order data. At the end of each load step, the distance between the low-dimensional converged nonlinear force vector and each of the low-dimensional sample forces in the set $\mathbf{F}^{\text{SNP}}$ is calculated [60]. The low-dimensional representation of the converged nonlinear force vector $\hat{\mathbf{f}}_{\text{DEIM}} \in \mathbb{R}^q$ and the sample force data $\mathbf{f}_{\text{DEIM},i} \in \mathbb{R}^q$ are defined as

$$\hat{\mathbf{f}}_{\text{DEIM}} = \mathbf{P}_{c_{\text{old}}}^{\top} \hat{\mathbf{F}} \tag{3.31}$$

$$\mathbf{f}_{\text{DEIM},i} = \mathbf{P}_{c_{\text{old}}}^{\top} \mathbf{F}_i \tag{3.32}$$

in which the subscript $c_{\text{old}}$ denotes the cluster number in the current load step.

The cluster number associated with the sample force data that has the minimum distance to the converged force is determined. This cluster represents the training state that is closest to the current state of the system and is denoted as $c_{\text{new}}$. A summary of the nearest neighbor classifier algorithm is presented in Algorithm 2.

**Algorithm 2:** Nearest neighbor classifier algorithm

**Data:** Snapshot matrix for nonlinear force vector $\mathbf{F} : \mathbf{F}^{\text{SNP}}$
Classifier function: **cls**
Converged force vector $\hat{\mathbf{F}}$ at the end of load step $\tau$
Cluster number in the load step $\tau$: $c_{\text{old}}$
**Result:** Cluster number for the load step $\tau + 1$: $c_{\text{new}}$
Calculate the low-dimensional representation of $\hat{\mathbf{F}}$: $\hat{\mathbf{f}}_{\text{DEIM}}$

**for** $i = 1 : \text{size}\left(\mathbf{F}^{\text{SNP}}, 2\right)$ **do**
  Calculate the low-dimensional representation of sample force $\mathbf{F}_i$: $\mathbf{f}_{\text{DEIM},i}$
  Calculate the distance between $\hat{\mathbf{f}}_{\text{DEIM}}$ and $\mathbf{f}_{\text{DEIM},i}$ using (3.28)
  Save the distance in $\mathbf{d}\,(i)$
Determine the minimum of vector $\mathbf{d}$ and find its index $k$
Determine the cluster number for load step $\tau + 1$: $c_{\text{new}} = \mathbf{cls}\,(k)$

The online stage involves solving a nonlinear reduced order system of equations, which is significantly more computationally efficient than the FOM. Once the update cluster number

$c_{\text{new}}$ is determined, the appropriate local interpolation matrix is utilized to define and solve the reduced order system of equations for the next load step:

$$\widetilde{\mathbf{R}} = \widetilde{\mathbf{F}}_{\text{ext}} - \widetilde{\mathbf{K}}_{\text{L}}\widetilde{\mathbf{u}} - \mathbf{A}_{w,c_{\text{new}}}\mathbf{f}_{\text{DEIM},c_{\text{new}}} = \mathbf{0} \tag{3.33}$$

$$\widetilde{\mathbf{K}} = \widetilde{\mathbf{K}}_{\text{L}} + \mathbf{A}_{w,c_{\text{new}}}\mathbf{k}_{\text{DEIM},c_{\text{new}}}\mathbf{W} \tag{3.34}$$

### 3.3.4 Offline Stage

In this section, the offline stage of the presented ROM is summarized. We begin by presenting the algorithm used for data generation, whereby the training process for constructing training basis vectors is explained. Furthermore, the implementation of $k$-fold cross-validation for constructing approximated nonlinear terms is described. Figure 3.1 illustrates the procedure used during the offline stage.

#### Data Generation

The offline stage begins by selecting a sampling space. A sampling space is generated using the material properties space $\mathcal{M} = \{\mu_1, ..., \mu_{n_m}\}$, load steps space $\mathcal{T} = \{\tau_1, ..., \tau_{n_s}\}$, and the Newton–Raphson iterations space $\mathcal{J} = \{iter_1, ..., iter_J\}$, where $J$ denotes the total number of iterations across all load steps ($J > n_s$). The snapshot matrices $\mathbf{U}^{\text{SNP}}$, $\mathbf{F}^{\text{SNP}}$, and $\mathbf{F}^{\text{SNP}}_{\text{conv}}$ are constructed by combining the snapshots of the high-fidelity FOM simulations for each sampling point $s_i = (\mu_i, \tau_i, iter_i)$, described in more detail in Algorithm 3. Samples for the nonlinear force vectors are stored at each iteration in the matrix $\mathbf{F}^{\text{SNP}}_{s_i}$. Then, samples for displacement and converged nonlinear force vectors are collected at the end of each load step in matrices $\mathbf{U}^{\text{SNP}}_{s_i}$ and $\mathbf{F}^{\text{SNP}}_{\text{conv},s_i}$, respectively. These snapshots are combined into three large matrices $\mathbf{U}^{\text{SNP}}$, $\mathbf{F}^{\text{SNP}}_{\text{conv}}$, and $\mathbf{F}^{\text{SNP}}$, which are used in the construction of the ROM through the processes described in Section 3.3.4.

#### Training Process

In the training process, snapshots $\mathbf{U}^{\text{SNP}}$ and $\mathbf{F}^{\text{SNP}}$ are used to construct the global bases for POD and localized bases for DEIM, respectively. Additionally, $\mathbf{F}^{\text{SNP}}_{\text{conv}}$ is divided into $n_c$ clusters using the $k$-means algorithm. Finally, the local DEIM approximations are computed for each cluster. Figure 3.1 shows the training regime for the ROM with LDEIM

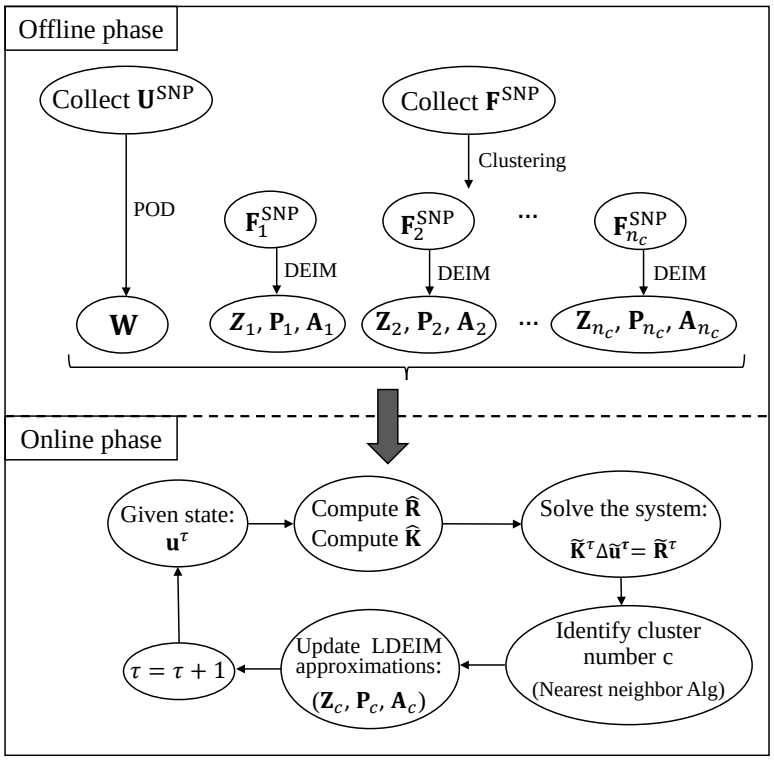



Figure 3.1: Workflow for the offline and online stages of the presented ROM.

approximations. It should be noted that the $k$-means algorithm is utilized to cluster the converged force vector $\mathbf{F}_{\mathrm{conv}}^{\mathrm{SNP}}$ rather than $\mathbf{F}^{\mathrm{SNP}}$, since the online nearest neighbor algorithm is based on the converged nonlinear force vector, resulting in a significant decrease in the computational cost of the clustering algorithm. A crucial step in the training process involves the precomputation of certain matrix multiplications, which reduces the computational cost during the online stage. In (3.33) and (3.34), the terms $\widetilde{\mathbf{K}}_{\mathrm{L}}$, $\mathbf{W}$, and $\mathbf{A}_{w,c}$, where $c = 1, \ldots, n_c$, are executed once offline.

### Cross-Validation

The primary concern associated with machine learning (ML) algorithms is overfitting. This situation occurs when the model learns the noise and outliers within the dataset. Cross-validation techniques are the most common strategies used to combat overfitting and improve the generalizability and predictability of models.

**Algorithm 3:** Data generation

**Data:** Material properties space $\mathcal{M} = \{\mu_1, ..., \mu_{n_m}\}$, load steps space $\mathcal{T} = \{\tau_1, ..., \tau_{n_s}\}$, and Newton–Raphson iterations space $\mathcal{J} = \{iter_1, ..., iter_J\}$

**Result:** Combined snapshot matrices: $\mathbf{U}^{\text{SNP}}$, $\mathbf{F}^{\text{SNP}}$ and $\mathbf{F}^{\text{SNP}}_{\text{conv}}$

Initiate the combined snapshot matrices: $\mathbf{U}^{\text{SNP}} = [\,]$, $\mathbf{F}^{\text{SNP}} = [\,]$, and $\mathbf{F}^{\text{SNP}}_{\text{conv}} = [\,]$

**for** $i = 1, ..., n_m$ **do**

Displacement and nonlinear converged force vectors at the end of each load step are collected in snapshot matrices $\mathbf{U}^{\text{SNP}}_{\mu_i}$ and $\mathbf{F}^{\text{SNP}}_{\text{conv},\mu_i}$, respectively:

$\mathbf{U}^{\text{SNP}}_{\mu_i} = \{\mathbf{u}_1, ..., \mathbf{u}_{n_s}\}$

$\mathbf{F}^{\text{SNP}}_{\text{conv},\mu_i} = \{\mathbf{F}_1, ..., \mathbf{F}_{n_s}\}$

The nonlinear unbalanced force vector at every iteration of each load step is stored in the snapshot matrix $\mathbf{F}^{\text{SNP}}_{\mu_i}$:

$\mathbf{F}^{\text{SNP}}_{\mu_i} = \{\mathbf{F}_1, ..., \mathbf{F}_J\}$

**for** $j = 1, ..., n_s$ **do**

**for** $i = 1, ..., n_m$ **do**

$\mathbf{U}^{\text{SNP}} = \left[\ \mathbf{U}^{\text{SNP}} \quad \mathbf{U}^{\text{SNP}}_{\mu_i}(:, j)\ \right]$

$\mathbf{F}^{\text{SNP}} = \left[\ \mathbf{F}^{\text{SNP}} \quad \mathbf{F}^{\text{SNP}}_{\mu_i}(:, j)\ \right]$

$\mathbf{F}^{\text{SNP}}_{\text{conv}} = \left[\ \mathbf{F}^{\text{SNP}}_{\text{conv}} \quad \mathbf{F}^{\text{SNP}}_{\text{conv},\mu_i}(:, j)\ \right]$

Overfitting is not of concern for intrusive ROMs due to the quasi-optimality of Galerkin projection. In this study, $k$-fold cross-validation is applied to assess the performance of ROM to ensure the model can generalize and extrapolate to unseen data. Furthermore, $k$-fold cross-validation provides computational savings during the offline stage, as only a portion of the generated data is used in the training process. This study introduces $k$-fold cross-validation to assess the predictive ability of the DEIM approximation of $\mathbf{F}$. However, future studies should implement cross-validation techniques to validate the complete ROM.

The approach involves randomly partitioning $\mathbf{F}^{\text{SNP}}$ into training, validation and testing datasets. 10% of the $\mathbf{F}^{\text{SNP}}$ is retained as the testing or unseen dataset. The remaining dataset is partitioned into $k$ folds where the $k^{th}$ fold is used as the validation set $\mathbf{F}^{\text{SNP}}_{\text{Validation}}$ and the remaining $k-1$ folds are used collectively as the training dataset $\mathbf{F}^{\text{SNP}}_{\text{Train}}$. The training dataset is used to train the DEIM approximation of $\mathbf{F}$. In contrast, the validation dataset is used to evaluate the candidate models. For each value of $k$, the model is trained $k$ times for all combinations of training and validation datasets, and then, the average validation error is reported. This procedure is iterated over different values of $k$ to determine the

optimal hyper-parameter $k$, which minimizes the testing error in the final constructed model. Finally, the testing dataset, withheld from the training process, is used to measure the extrapolation ability and generalizability of the model. Figure 3.2 illustrates the standard $k$-fold cross-validation procedure [69].

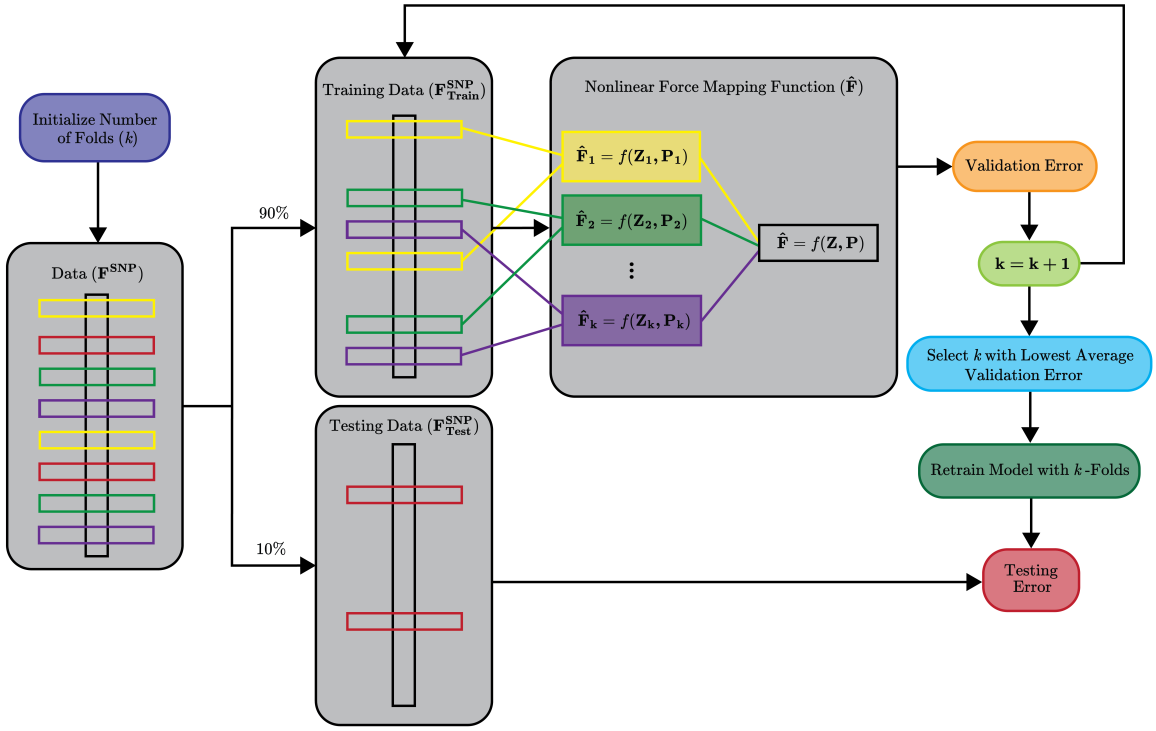
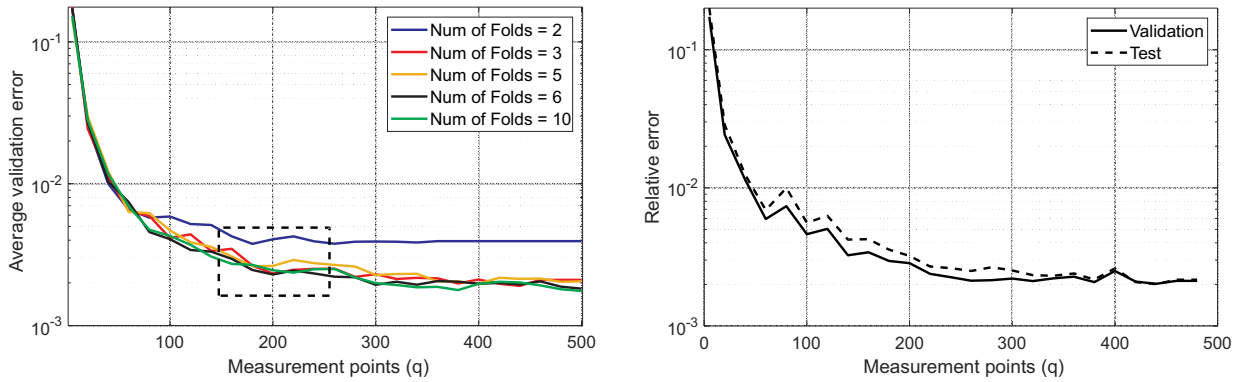



Figure 3.2: $k$-folds cross-validation procedure.

We show the process employed for selecting the hyper-parameter $q$, the number of DEIM measurement points for the problem discussed in Section 3.4.2. The process begins by performing $k$-fold cross-validation for values of $k$ ranging from 2 to 10, where the average validation error for each $k$ is reported, as shown in Figure 3.3a. Through $k$-fold cross-validation, the optimal hyper-parameter $k$ for the DEIM approximation should be set to 3 to achieve an accurate approximation of the force vector, where measurement points between 150 and 250 should be used to ensure adequate accuracy and minimize the computational cost. Figure 3.3b shows the average validation error for the constructed DEIM approximation of the nonlinear force vector against the testing dataset.




(a) (b)

Figure 3.3: (a) $k$-fold cross-validation of DEIM force vector for training dataset, (b) average testing error vs. measurement points for testing dataset ($k = 3$).

A 3-fold cross-validation is employed to demonstrate the consistency of the nonlinear force vector approximation where the common measurement points between folds are shown in Figure 3.4. The measurement points used in the DEIM approximation are consistent between folds, indicating that the trained DEIM approximation of the $\mathbf{F}$ is not overfitting to the training data. Furthermore, the common measurement points between folds tend to be located in areas where the plastic behavior occurs.

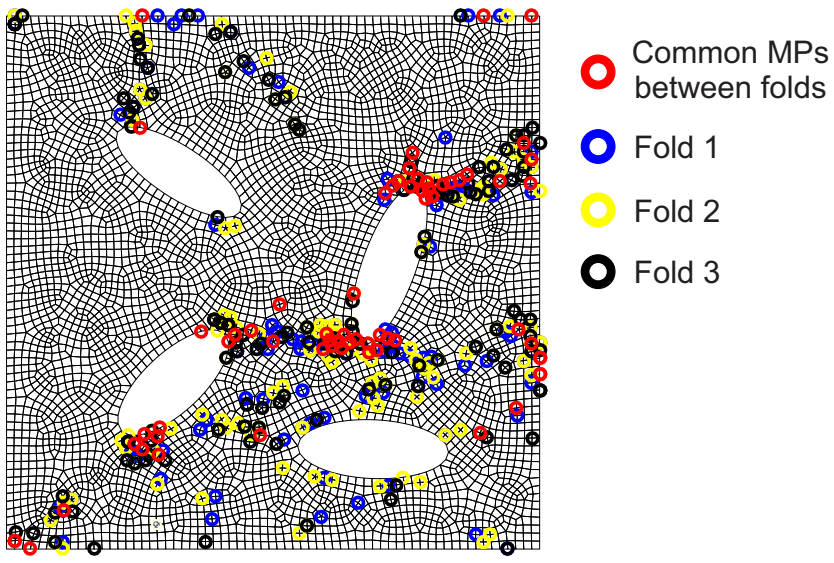



Figure 3.4: Spatial distribution of DEIM measurement points when $q = 200$.

## 3.4 Results and discussion

In this section, we first verify the FOM code using established verification examples from the literature. An inclined earth embankment under self-weight is analyzed in plane strain condition, and the results are compared with those presented by de Souza Neto *et al.* [79]. Next, we present two examples to demonstrate the capability of the presented ROM for plastic deformation of solids, in which the material properties and load path are parametric inputs. The first example considers a parameterized ROM with two material inputs for a micromechanical model of porous media. The effects of hyper-parameters ($m$ and $q$) and training regimes are investigated. In addition, the performance of the ROM with LDEIM over the ROM with classical DEIM and the FOM is demonstrated. In the second example, reduced two-dimensional plastic deformation of a plate with a central hole subject to cyclic plastic loading is considered. It will be shown that using LDEIM approximations, an efficient and accurate ROM can be constructed. In both examples, a sensitivity analysis is performed to evaluate the ability of ROM to extrapolate beyond the training data range.

### 3.4.1 FOM Verification

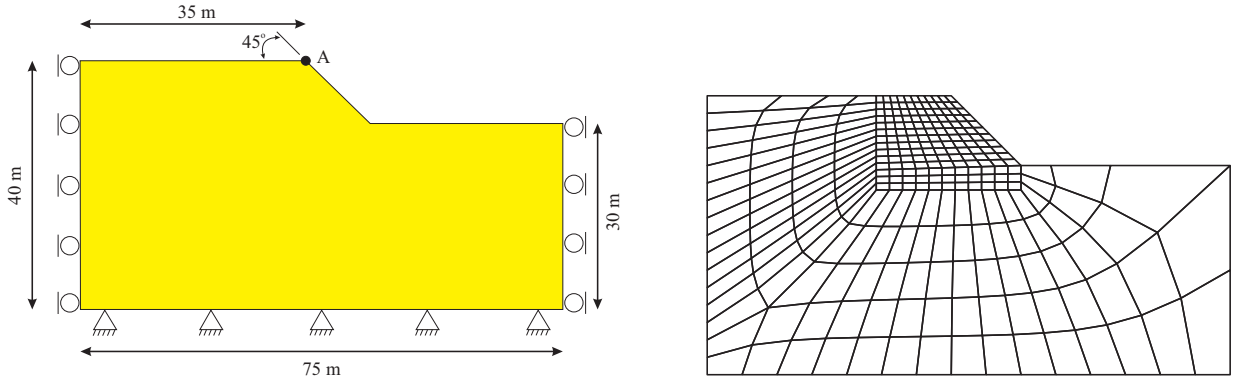
In this section, a slope stability analysis is conducted on an inclined earth embankment, as shown in Figure 3.5a. The problem is simulated under plane strain state, and the results are compared with the available numerical results [79].

The gravity acceleration load $\bar{g}$ is applied to the domain as follows:

$$\bar{g} = \alpha g \tag{3.35}$$

where $g$ is the standard acceleration of gravity, and $\alpha$ is the gravity load factor. In an incremental process, the value of $\alpha$, starting from 0, is gradually increased until collapse occurs. At the collapse level, the value of $\alpha$ represents the safety factor.

To verify the results, the same FEM mesh used in [79] is adopted. The FOM is discretized using eight-noded quadrilateral elements (Q8) with reduced integration ($2 \times 2$ integration points). Figure 3.5b shows a mesh with 909 nodes and 282 elements. The material properties of the soil, modeled as a perfectly plastic Drucker–Prager material, are listed in Table 3.1 [79].



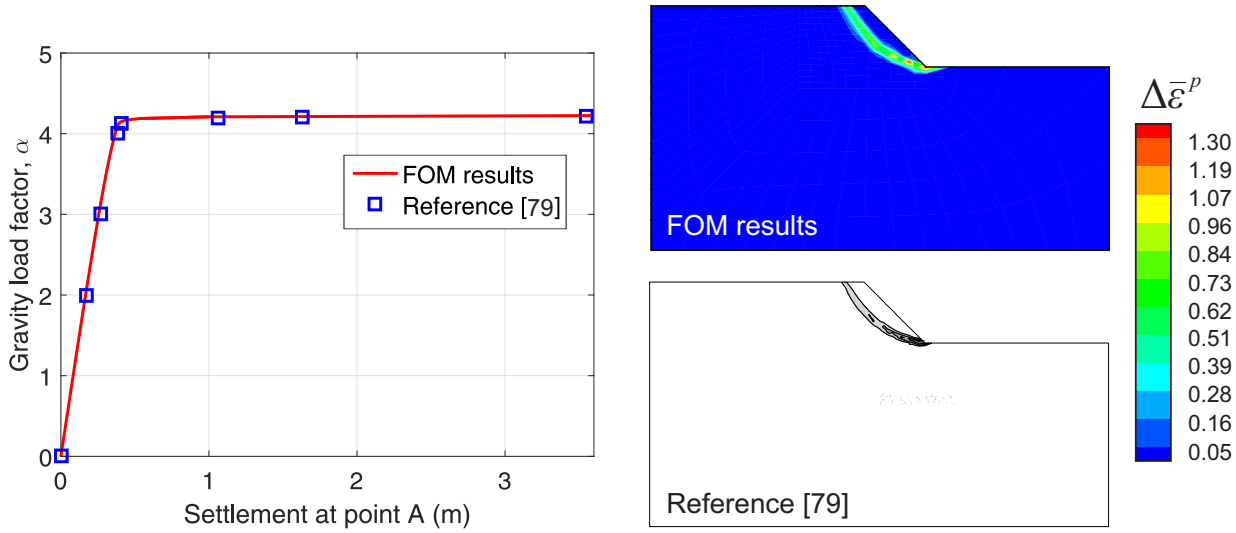

(a) (b)

Figure 3.5: a) Two-dimensional slope stability problem, b) FEM mesh with 909 nodes and 282 Q8 elements.

Table 3.1: Material properties of slope stability problem

| Variable | Symbol | Value |
|---|---|---|
| Young's Modulus | $E$ | 20 MPa |
| Poisson's Ratio | $\nu$ | 0.49 |
| Initial Cohesion | $c_0$ | 50 kPa |
| Friction Angle | $\phi$ | 20° |
| Hardening Modulus | $H$ | 0 kPa |
| Density | $\rho$ | 2000 $^{\text{kg}}/_{\text{m}^3}$ |
| Standard Acceleration of Gravity | $g$ | 10 $^{\text{m}}/_{\text{s}^2}$ |

Figure 3.6a compares the FOM results with the numerical model by de Souza Neto *et al.* [79] in predicting the settlement of the embankment at point A. The FOM results demonstrate good agreement with the reference FE results for various gravity load factors. The calculated safety factor is 4.21, which closely matches the value reported by [79], i.e., $\alpha = 4.2$. Figure 3.6b presents the contour map of the incremental accumulated plastic strain at collapse. As can be seen, the failure mechanism predicted by the FOM is in good agreement with the numerical results reported by [79].




(a) (b)

Figure 3.6: a) Load factor vs. settlement diagram, b) increment of accumulated plastic strain at collapse.

### 3.4.2 Micromechanics of Porous Media

The objective of this example is to compare the performance of parameterized ROM with that of the FOM for a readily reproduced, yet non-trivial, plasticity problem. The example is inspired by models of representative elementary volume (REV) used in $FE^2$ simulations of porous media [87, 88, 89]. The micromechanical behavior of a porous medium is studied through the analysis of a 2m $\times$ 2m domain consisting of four large elliptical cavities, as shown in Figure 3.7a. A uniform compressive load of 120 kPa is incrementally applied along the right edge of the domain; the displacements along the left edge are restrained in both $x$ and $y$ directions. The FOM is discretized by quadrilateral elements (Q4). Figure 3.7b depicts the coarser mesh (mesh 1) with 5209 nodes and 4992 elements. Furthermore, a finer mesh (mesh 2) with 45478 nodes and 44822 elements is considered for the FOM training process to ensure a more accurate capture of stress concentration around cavities. Figure 3.7c illustrates a complex plastic behavior due to the domain geometry and the applied load. Multiple plastic strain bands are formed in the domain and initiated from stress concentrations at the cavities. A successful MOR technique must capture such behavior. Table 3.2 outlines the material properties used in the training process.

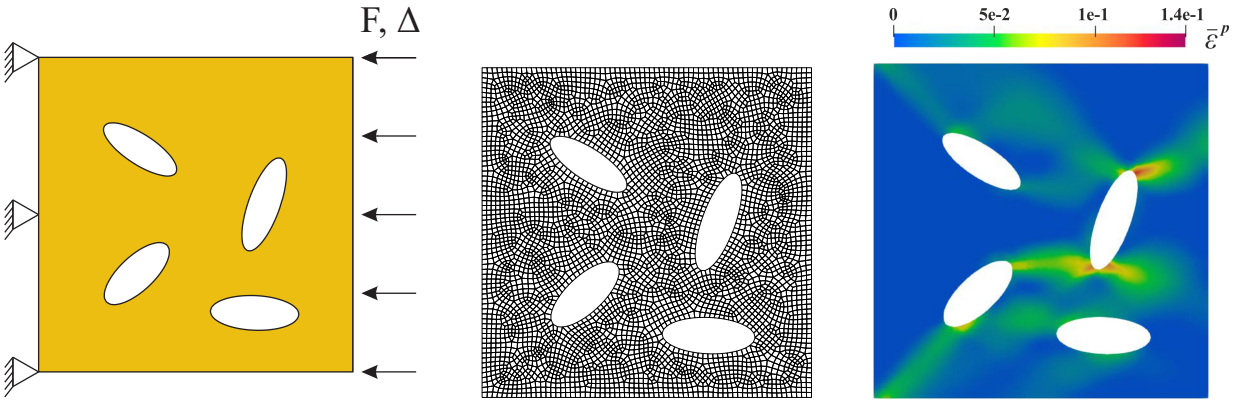



Figure 3.7: a) Two-dimensional micromechanics of porous media, b) FEM mesh with 5209 nodes and 4992 Q4 elements, c) plastic strain bands.

Table 3.2: Material properties for the generation of training data

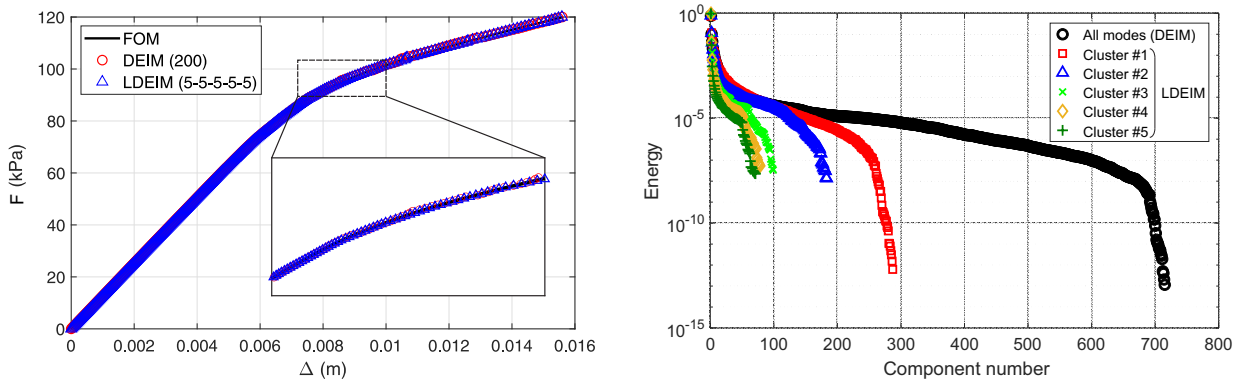
| Variable | Symbol | Value |
|---|---|---|
| Young's Modulus | $E$ | 20 MPa |
| Poisson's Ratio | $\nu$ | 0.49 |
| Initial Cohesion | $c_0$ | 50 kPa |
| Friction Angle | $\phi$ | 20° |
| Hardening Modulus | $H$ | 50 kPa |

**Non-parameterized ROM**

We begin by evaluating the accuracy and computational cost of the ROMs with DEIM and LDEIM for a nonlinear system without parametric inputs, comparing them to the FOM. In the presented example, the size of the ROM is reduced from 10418 DoFs and 90956 DoFs using POD to 60 DoFs for meshes 1 and 2, respectively. The low-energy POD modes are truncated to achieve an accuracy of 99% for the projection matrix $\mathbf{W}$. Based on our experience, reducing the number of truncated POD modes significantly affects the performance of the Newton–Raphson solver in plasticity problems. By excluding certain mode shapes, some crucial features of the problem are missed, leading to an ill-conditioned Jacobian matrix that is non-invertible.

Figure 3.8a illustrates that ROMs with DEIM and LDEIM can accurately capture the global response of the system. The ROM with DEIM approximates the nonlinear force vector using $q = 200$ measurement points compared to the 10418 DoFs of the FOM. It is

observed that the Newton–Raphson algorithm in DEIM will not converge when trained with fewer than 200 measurement points. However, the local DEIM approximations enable many fewer measurement points. In this example, LDEIM with 5 clusters and 5 measurement points per cluster is efficient and the Newton–Raphson algorithm is stable. In Figure 3.8a, the notation 5-5-5-5-5 refers to the use of an LDEIM approximation with 5 clusters and 5 measurement points per cluster. Figure 3.8b depicts the energy of modes in DEIM and LDEIM resulting from SVD analysis of nonlinear force snapshots. DEIM exhibits slower decay compared to LDEIM. In such a case, truncating a large number of POD singular values ($q$) with significant energy can exponentially increase the interpolation error. In contrast, LDEIM requires fewer measurement points to decrease the interpolation error through the approximation of the nonlinear terms.

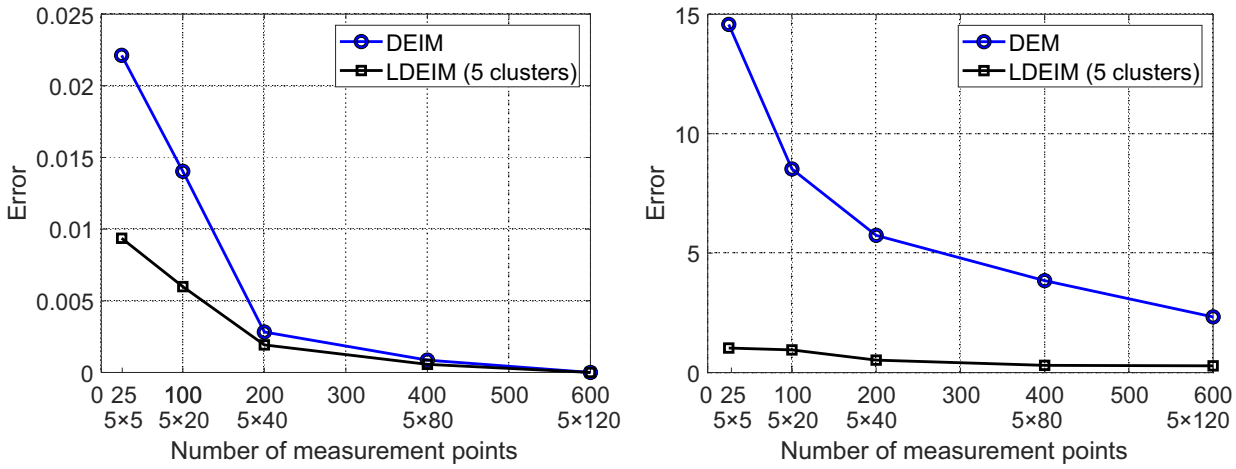



(a) (b)

Figure 3.8: a) Force-displacement diagram for the FOM, the ROM with DEIM and the ROM with LDEIM, b) energy of mode shapes in DEIM and LDEIM.

One cause of the instability of the Newton–Raphson algorithm observed with DEIM can be attributed to the quality of the DEIM approximations for the nonlinear force vector. Figure 3.9a represents the quality of the approximation of the nonlinear force vector. In this figure, a nonlinear force vector from the FOM solution is approximated using DEIM and LDEIM. As can be seen, DEIM shows poor performance with fewer measurement points compared to LDEIM. For instance, with 25 DEIM modes, the quality of approximation of the force vector is much lower than that of LDEIM with $5 \times 5$ modes. However, as the number of modes increases, the accuracy of DEIM and LDEIM becomes comparable. Therefore, the same number of measurement points is not expected for DEIM and LDEIM. The Newton–Raphson algorithm in DEIM will not converge when trained with fewer than 200 measurement points because of the poor approximation quality of the force vector. Another cause of the instability of the Newton–Raphson algorithm observed with DEIM

can be attributed to the accuracy of the DEIM approximation for the Jacobian. Figure 3.9b shows the quality of approximation of the Jacobian for DEIM and LDEIM. As shown, the approximation error of DEIM is consistently higher than that of LDEIM. For a small number of measurement points, the error of DEIM is significantly high, adversely affecting the convergence of the Newton–Raphson algorithm. However, as the number of modes increases, the DEIM error decreases, which improves the convergence rate of the nonlinear solver. It should be noted that the error for LDEIM remains relatively high at approximately 0.5. This high value is related to the method used to approximate the Jacobian. In this chapter, the Jacobian is approximated using the same DEIM bases determined for the nonlinear force vector. Thus, the approximated full order Jacobian matrix is different from the exact full order Jacobian matrix. Another potential cause of the instability of the Newton–Raphson algorithm may be the poor scaling of the Lebesgue constant with the number of DEIM interpolation points. Regression methods, such as Gappy POD, provide more accurate approximations for the nonlinear force vector and Jacobian [90]. Unlike DEIM, which uses interpolation to ensure exact alignment with the solution at measurement points, Gappy POD employs regression and least squares to approximate nonlinear terms. Furthermore, other DEIM variants, such as QDEIM, which has a tighter error bound compared to DEIM, could address these issues [55]. These investigations will be of interest in future studies.




(a) (b)

Figure 3.9: Quality of nonlinear force vector and Jacobian approximations based on the number of measurement points, a) nonlinear force vector, b) Jacobian.

Figure 3.10 shows the performance of the ROM with LDEIM compared to the ROM with DEIM. The results indicate the effect of hyper-parameters $m$ and $q$ on the accuracy and

the computational cost of the ROM. For mesh 1, DEIM shows a small speed-up of 7. In contrast, LDEIM produces significantly faster ROMs than the FOM. When $q$ is 5, the CPU time for LDEIM is 28 seconds, which is 170 times faster than the FOM. Furthermore, for a finer mesh (mesh 2), LDEIM provides a speed-up factor of 2933 when $q$ in each cluster is 5. We can conclude that LDEIM seems to be a more robust ROM than DEIM for Drucker–Prager plasticity problems.

From Figure 3.10a, it can be observed that the number of measurement points $q$ in each cluster does not affect the accuracy of LDEIM, as the POD error becomes dominant in this case. In Figure 3.10a, the ROM size $m$ is 60. The fluctuations in the response of the system, attributed to cluster changes, significantly affect the error. However, the CPU time increases with the number of measurement points. In addition, LDEIM error is generally smaller than DEIM. Figure 3.10b shows the effect of hyper-parameter $m$ (POD modes) on the accuracy and the computational cost of the ROM. Increasing the size of the hyper-parameter $m$ decreases the $L_2$ error for LDEIM, while the CPU time remains relatively unchanged. For $m > 50$, the error stalls because the DEIM error becomes dominant. It is noted that for mesh 2, LDEIM fails to converge for $m$ smaller than 50.

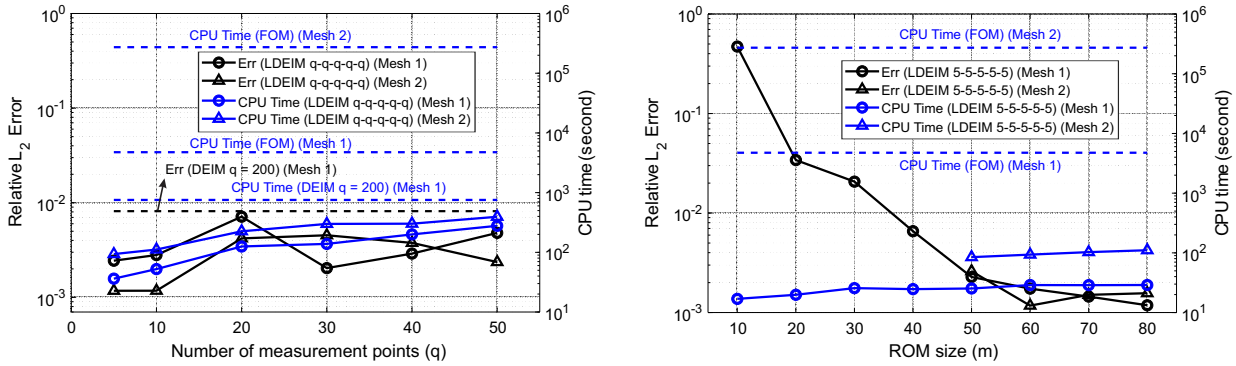



Figure 3.10: Effect of the hyper-parameters $m$ and $q$ on the accuracy and the computational cost of the ROMs, a) hyper-parameter $q$ ($m = 60$), b) hyper-parameters $m$.

### Parameterized ROM with Two Material Inputs

In the following section, we perform a parameter sensitivity analysis, focusing on Young's modulus and hardening modulus as parametric inputs of the presented ROM. This example discusses the ability of the ROM to solve plasticity problems with two parametric inputs. As outlined in Table 3.3, the ROM is trained using a small amount of data (only ten

samples). The ability of the ROM to interpolate and extrapolate the response of the system is evaluated using parametric inputs located inside and outside of the training domain. Figure 3.11 illustrates the global responses of the FOM, based on mesh 1, for ten samples used in the training set.

Table 3.3: Samples used for training process

| Sample | $E$ (MPa) | $H$ (kPa) | Sample | $E$ (MPa) | $H$ (kPa) |
|---|---|---|---|---|---|
| 1 | 10 | 300 | 6 | 10 | 500 |
| 2 | 12 | 300 | 7 | 12 | 500 |
| 3 | 15 | 300 | 8 | 15 | 500 |
| 4 | 20 | 300 | 9 | 20 | 500 |
| 5 | 30 | 300 | 10 | 30 | 500 |

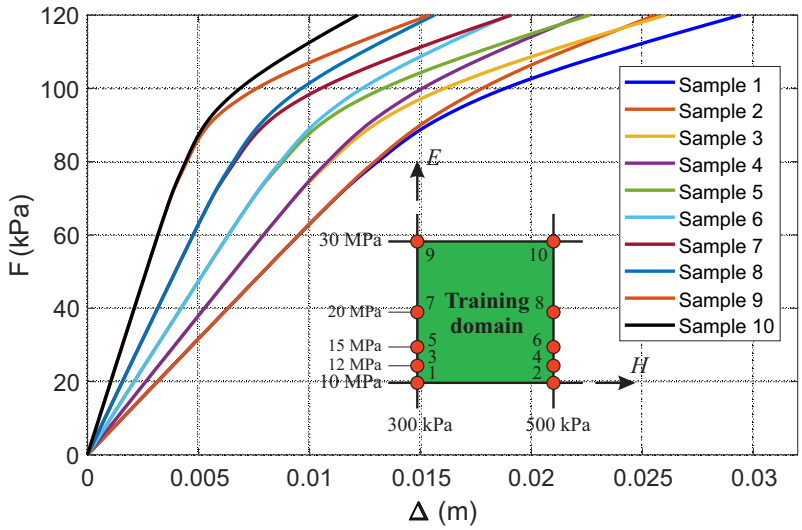



Figure 3.11: Global responses of the FOM for ten samples used for training purposes.

It is important to highlight that the Newton–Raphson algorithm in the ROM based on classical DEIM fails to converge unless a large number of measurement points $q$ are used. It is observed that the Newton integration does not converge when the ROM is trained with fewer than 2000 measurement points. An explanation could be that DEIM exhibits a slow decay in terms of the energy of the mode shapes for the nonlinear force snapshots over the two-dimensional material parameter space. Additionally, the approximated tangent stiffness matrix can be a cause of the instability of the Newton–Raphson solver. The tangent stiffness matrix is approximated using the same POD basis vectors that are determined for the residual force vector. This issue necessitates the use of LDEIM.

The converged nonlinear force vectors are grouped into 5 clusters. The size of the ROM is reduced from 10418 DoFs to 60 DoFs using the POD, and the nonlinear force vector is approximated with 5 measurement points in each cluster.

Figure 3.12a displays the relative $L_2$ error of the ROM with LDEIM for the inputs within and outside of the training domain. In the presented domain, Young's modulus and hardening modulus range from 2.5 to 35 MPa and 200 to 550 kPa, respectively. The accuracy is highest for the inputs within the training domain (error of 1.0–3.5%). However, the error increases when the input parameters are located outside of the training domain (error of 1.0–9.2%).

The computational efficiency of the ROM is shown in Figure 3.12b. The LDEIM substantially reduces CPU time compared to the FOM. Speed-up factors range from 106 to 134 across different ROM inputs. As can be seen, speed-up factor distribution is not uniform for the inputs in the training domain, as the Newton–Raphson performance in the ROM (number of Newton iterations) varies from one input to another one. It is expected that the upper bound of the speed-up factor is 170 based on the results of the unparameterized ROM shown in Section 3.4.2. Furthermore, the online computational cost savings scale with the size of the training FOM (DoFs) and the number of load steps ($n_s$). In this example, the training data is generated using FOM simulations with a coarse mesh. Potential cost savings are expected to grow exponentially if FOM simulations with a fine mesh are used during the offline stage, similar to the observations outlined in Section 3.4.2.

Figure 3.13a illustrates the online loss function over the load steps for inputs inside the training domain. The loss function is defined using (3.36), where $n_{test}$ refers to the number of online material inputs, which, in this case, is equal to 36. It can be observed that the loss function increases with the load, which indicates that the error between the FOM and the ROM with LDEIM solution increases due to the evolution of the plastic front in the domain. The load-displacement curve of LDEIM for the input parameters $E = 27.5$ MPa and $H = 475$ kPa is presented in Figure 3.13b. There is considerable agreement between the response of the FOM and the ROM with LDEIM. The observed fluctuations in the load-displacement curve for LDEIM, specifically in $\Delta = 0.0059$, 0.0077, 0.0102, and 0.0113 m, can be attributed to the cluster changes in DEIM basis vectors. One possible solution to remove these small fluctuations is to use overlapping clusters rather than non-overlapping ones which is beyond the scope of this work [85].

$$\text{Loss function} = \frac{\sum_{i=1}^{n_{test}} |\Delta_{\text{LDEIM}} - \Delta_{\text{FOM}}|}{n_{test}} \tag{3.36}$$

Figure 3.14a illustrates the displacements in $x$ and $y$ directions along the cross-section S for a set of input parameters inside the training domain: $E = 27.5$ MPa and $H = 475$ kPa.

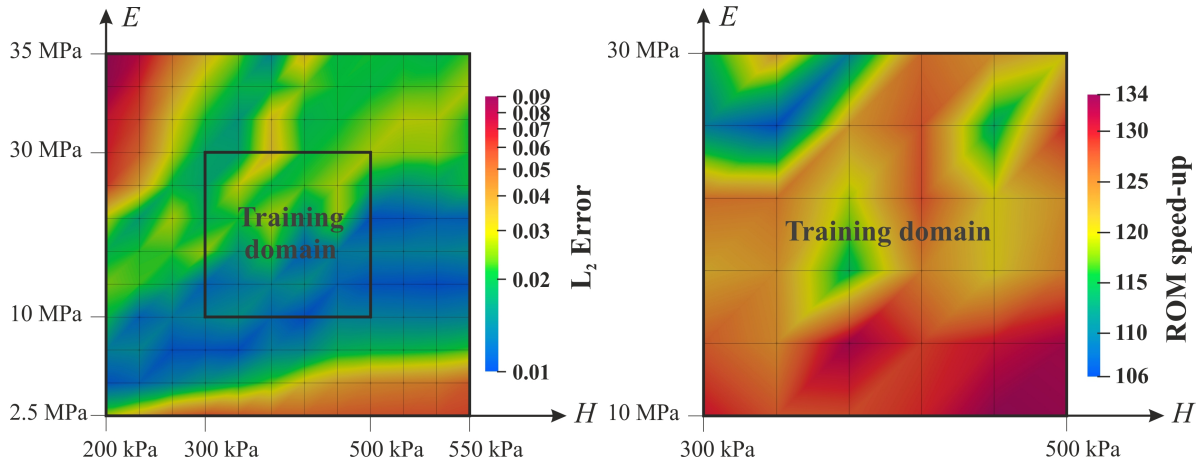



Figure 3.12: Accuracy and speed-up analyses, a) relative $L_2$ error contour of LDEIM for inputs inside and outside of the training domain, b) ROM speed-up for inputs inside of the training domain.

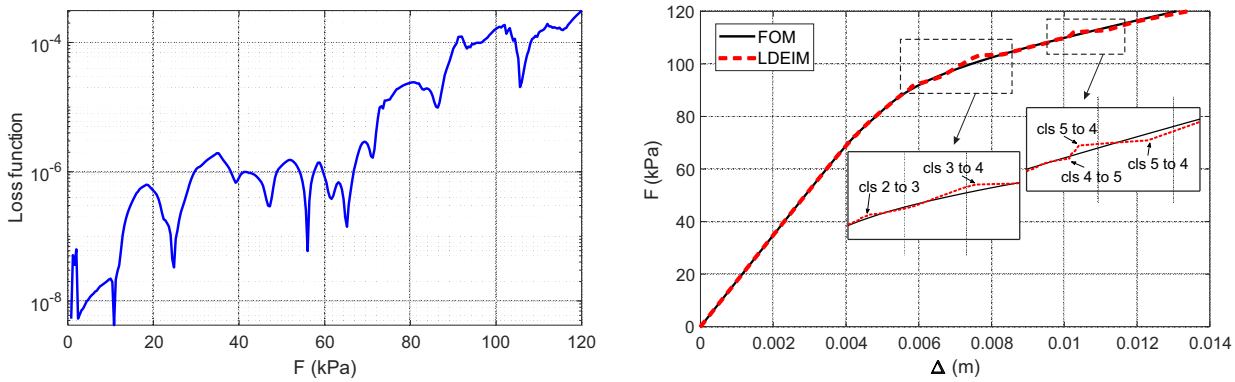



Figure 3.13: Loss function of online phase at various load steps, b) load-displacement response when the ROM input is $E = 27.5$ MPa and $H = 475$ kPa.

This is one of the configurations where the difference between the FOM and the ROM inside the training domain is the largest. There is a good agreement between the response of the FOM and the solution of the ROM with LDEIM. LDEIM calculates the displacement in the entire domain and satisfies the boundary conditions along the left edge of the domain. The von Mises stress $\sigma_{vm}$ and accumulated plastic strain $\bar{\varepsilon}^p$ are presented in Figures 3.14b

and 3.14c, respectively. The gray zone indicates areas where LDEIM does not compute $\sigma_{vm}$ and $\bar{\varepsilon}^p$. $\sigma_{vm}$ and $\bar{\varepsilon}^p$ are the secondary variables in the governing equations which are calculated only at LDEIM measurement points. In contrast, displacement is the primary variable that can be regenerated from the reduced order solution using (3.16).

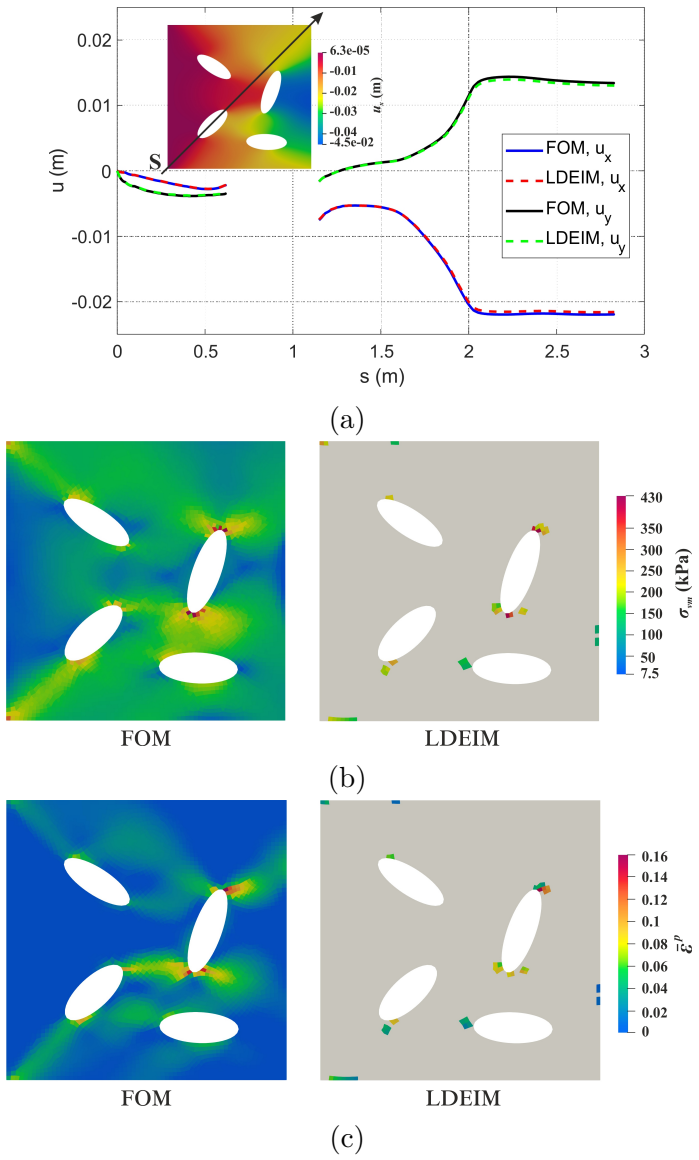



Figure 3.14: Comparison of primary and secondary variables for the FOM and the ROM with LDEIM when the ROM input is $E = 27.5$ MPa and $H = 475$ kPa, a) displacements in $x$ and $y$ directions along the line S, b) von Mises stress, c) accumulated plastic strain.

Figures 3.14b and 3.14c show that LDEIM measurement points are situated along the plastic bands, particularly close to the zones with stress concentration. As can be seen in the contours corresponding to LDEIM, $\sigma_{vm}$ and $\bar{\varepsilon}^p$ are evaluated at more than five measurement points. LDEIM contours provide information across all the measurement points of all the clusters, reflecting the history-dependent nature of the problem and the need to store the evolution of variables over time.

Table 3.4 summarizes the computational cost associated with the offline and online phases of the ROM with LDEIM. The offline phase consists of two main portions: data generation and data training. The online phase involves testing the trained ROM with different inputs. The computational cost of the online phase is significantly lower than that of the offline phase, highlighting the efficiency of the presented ROM for plasticity problems. In addition, intrusive ROMs, as physics-informed methods, are expected to require smaller training data than purely data-based methods. This can be explained by the concept that intrusive ROMs leverage the underlying physics of the system to reduce the data required to achieve an accurate solution. For this specific problem, due to the complexity of plastic distribution in the domain, we speculate that the computational cost of the offline phase for non-intrusive ROM techniques is significantly higher than the cost for intrusive ROM. This investigation will be of interest in future studies.

Table 3.4: CPU runtime in seconds for offline and online stages of the ROM with LDEIM. The CPU time for online phase is reported based on an average of 36 material inputs inside the training domain

| Offline phase (second) | | Online phase (second) | |
|---|---|---|---|
| Data generation | Data training | ROM | FOM |
| $4.5 \times 10^4$ | $4.1 \times 10^4$ | $3.6 \times 10^1$ | $4.5 \times 10^3$ |

### 3.4.3 Cyclic Plastic Loading

Intrusive ROMs have faced significant challenges in learning the path-dependent nonlinear evolution of plastic strain as a function of a cyclic loading path. This section demonstrates the capability of the presented ROM with LDEIM when parameterized by the loading path. A plate with a hole subject to compressive cyclic loading is analyzed. The plastic strain originates from the hole boundary exhibiting stress singularity, and then propagates through the plate as a plastic strain band.

Figure 3.15a depicts a two-dimensional plastic plate with a central hole. Due to the symmetry, a 1/4 model is used where Dirichlet boundary conditions are applied along the symmetric planes. A cyclic compressive load is considered to highlight the ability of ROM to capture pressure-dependent Drucker–Prager plasticity models. The domain is discretized by a coarse mesh consisting of 707 nodes and 649 Q4 elements, as shown in Figure 3.15c. The material properties utilized in the simulations are listed in Table 3.2.

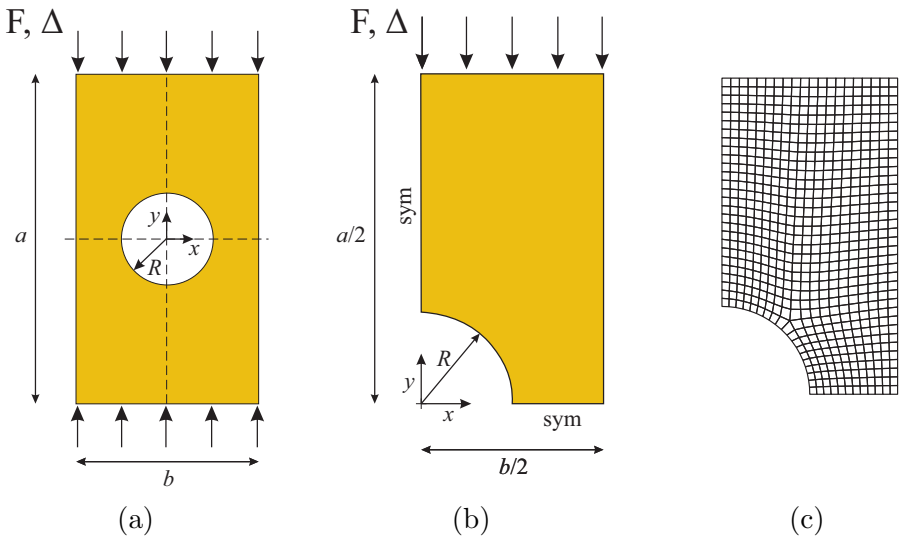



Figure 3.15: (a) Two-dimensional plastic plate with a central hole, (b) quarter plate with a central hole due to symmetric, (c) FE mesh with 707 nodes and 649 Q4 elements.

In this example, the parametric input of the ROM is the loading path. We define a simple training regime that can capture a wide range of responses while online load paths exhibit significant variations in both time and magnitude. Offline, the training dataset is generated by collecting the displacement, converged force vector and unbalanced force vector in matrices $\mathbf{U}^{\text{SNP}}$, $\mathbf{F}^{\text{SNP}}_{\text{conv}}$, and $\mathbf{F}^{\text{SNP}}$ for three loading and unloading cycles with compressive stresses ranging from 0 to 100 kPa, as shown in Figure 3.16a. Online, the ROM is evaluated with arbitrary loading paths, depicted in Figure 3.16.

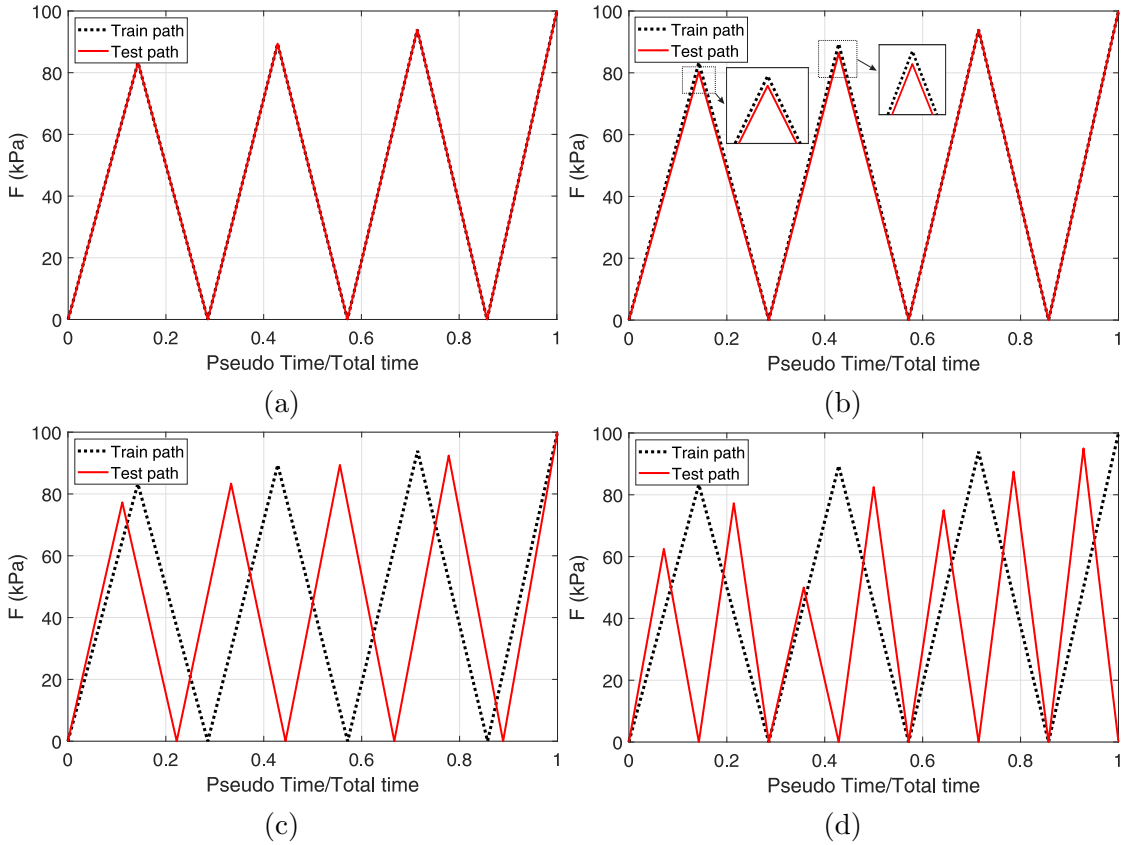



Figure 3.16: Online loading-unloading paths, (a) load case 1, (b) load case 2, (c) load case 3, (d) load case 4.

Here, the Newton–Raphson solver is inaccurate and unstable for the ROM with DEIM unless a large number of measurement points $q$ are used. A small number of measurement points does not appear to be an accurate approximation for the nonlinear terms, as DEIM shows a slow decay in the singular values of the mode shapes. However, LDEIM approximations enable many fewer measurement points. In this case, the snapshot matrix for $\mathbf{F}_{\text{conv}}^{\text{SNP}}$ is divided into 7 clusters. As depicted in Figure 3.17, the clusters correspond to specific sections of the training load-deflection curve and are defined using the $k$-means clustering method, yielding intuitive results.

In the presented ROM, the size of the system of equations is reduced to $m = 40$, where the nonlinear force vector is approximated using $q = 5$ measurement points at each cluster ($n_c = 7$). In Figure 3.18, it is shown that the ROM accurately captures the response of the FOM for all test cases. In addition, the ROM can reproduce the complex response of cases

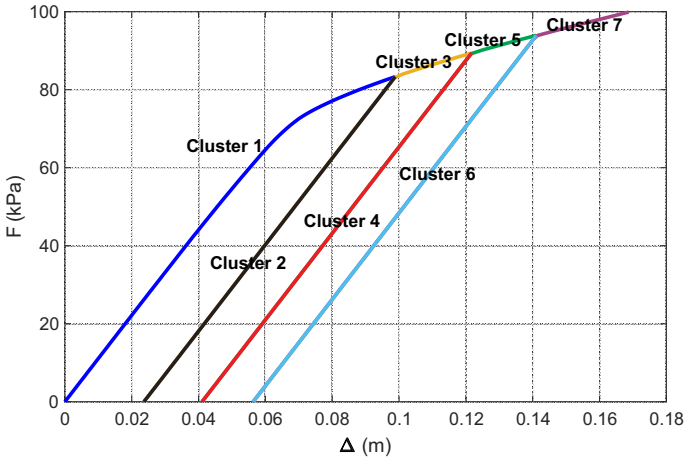



Figure 3.17: Division of force vector into seven clusters.

3 and 4, which are greatly different from the loading-unloading cycles used for the training process.

In Figure 3.19a, the history of accumulated plastic strain at points A and B is shown to evaluate the path-dependent nonlinear evolution of the plastic front. The figure represents the accuracy of the ROM with LDEIM in simulating the local plastic deformation in the domain throughout the loading cycles. There are five segments in the graph for plastic evolution, showing the loading parts of the cycles. The constant parts represent the unloading part of the cycles, as there should be no plastic evolution during unloading.

The computational cost of four load cases is summarized in Table 3.5. First, the FOM with a very coarse mesh size (1414 DoFs) is used to generate the training data. It is observed that the ROM with LDEIM is faster than the FOM for different load cases. For instance, the ROM speed-up factor for load case 4 is 11. Table 3.5 also shows the average number of Newton–Raphson iterations per load step required for the FOM and the ROM. The difference between the FOM and the ROM with LDEIM in the number of Newton iterations is small; however, ROMs with LDEIM generally need more Newton–Raphson iterations than the FOM. To further demonstrate the potential computational savings of the ROM, we utilize a very fine mesh with 46,352 DoFs for the most complex loading-unloading cycle of load case 4. The CPU speed-up for the fine mesh is approximately 770, while the average number of Newton–Raphson iterations per load-step is 5, close to that of the FOM. These results highlight the significant gain in computational efficiency provided by the ROMs with LDEIM. In addition, it is demonstrated that a more significant ROM speed-up is attained for problems with larger FOMs (i.e., DoFs).

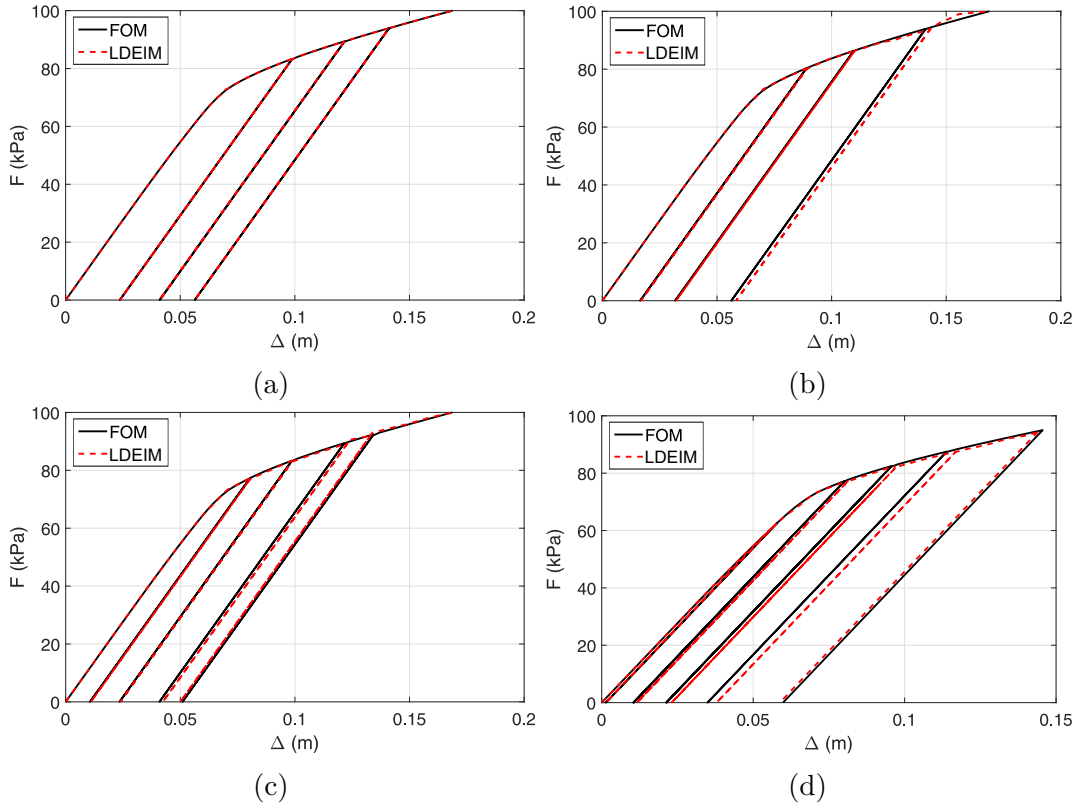



Figure 3.18: Comparison of load-displacement responses for FOM and ROM with LDEIM using various load paths, (a) load case 1, (b) load case 2, (c) load case 3, (d) load case 4.

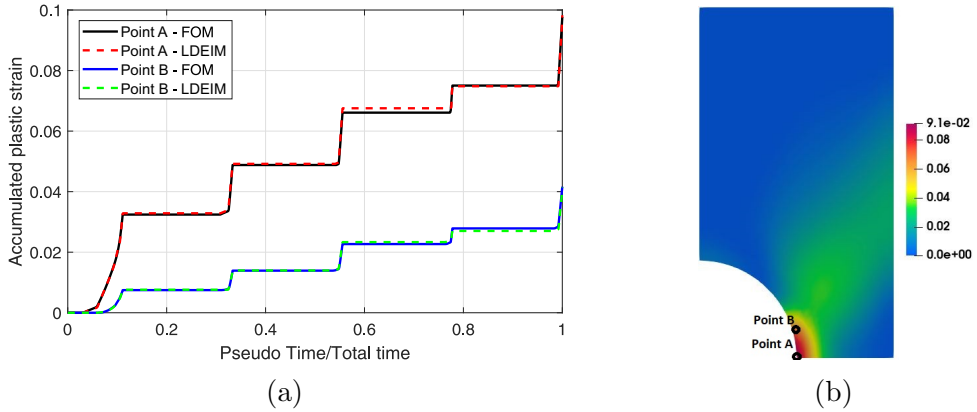



Figure 3.19: Plastic strain evolution for FOM and ROM with LDEIM at points A and B.

Table 3.5: CPU time in seconds, ROM speed-up, and average number of Newton–Raphson iterations needed for the FOM and the ROM with LDEIM

| Load Case | Mesh 1 - Coarse Mesh ($\sim 10^3$ DoFs) | | | | |
|---|---|---|---|---|---|
| | CPU Time (second) | | ROM Speed-up | Number of Newton Iterations | |
| | FOM | ROM | | FOM | ROM |
| 1 | $2.7 \times 10^2$ | $4.6 \times 10^1$ | 6 | 3 | 5 |
| 2 | $2.7 \times 10^2$ | $4.3 \times 10^1$ | 6 | 3 | 5 |
| 3 | $3.5 \times 10^2$ | $4.3 \times 10^1$ | 8 | 3 | 3 |
| 4 | $4.9 \times 10^2$ | $4.3 \times 10^1$ | 11 | 3 | 3 |
| Load Case | Mesh 2 - Fine Mesh ($\sim 5 \times 10^4$ DoFs) | | | | |
| | CPU Time (second) | | ROM Speed-up | Number of Newton Iterations | |
| | FOM | ROM | | FOM | ROM |
| 4 | $3.7 \times 10^5$ | $4.8 \times 10^2$ | 770 | 3 | 5 |

## 3.5 Conclusions

In this chapter, we have developed a robust ROM that can efficiently and accurately predict path-dependent nonlinearity in Drucker–Prager plasticity problems. DEIM provides computational cost savings by approximating the high-dimensional nonlinear terms (i.e., residual vector and its Jacobian). By clustering the temporal and parameterized training data generated from the FOM runs and localizing the DEIM approximations, further reductions are incorporated into the ROM. Using linear-nonlinear decomposition, an efficient and accurate ROM is achieved by applying sparse sampling and interpolation exclusively to the nonlinear components of the residual vector and its Jacobian. The accuracy and computational efficiency of the ROM are compared to the FOM when parameterized by three hyper-parameters (the size of the ROM, the number of clusters, and the number of DEIM measurement points per cluster).

The ROM is applied to a micromechanical porous media model, parameterized by elastic and hardening moduli; and plate with a hole problem, parameterized by a cyclic load. Observations confirm the benefits of LDEIM compared to global DEIM for problems in which there is a slow decay in the singular values of mode shapes. The lack of stability in the Newton–Raphson algorithm is one of the challenges in applying global DEIM to the Drucker–Prager model. The ROM built with classical DEIM is excessively inefficient as it requires a large number of measurement points. However, the ROM built with LDEIM exhibits stable, accurate, and efficient online performance because of its superior

ability to accurately interpolate the nonlinear terms over the parameter space with very few measurement points. A potential cause of the instability of the Newton–Raphson algorithm can be the poor scaling of the Lebesgue constant with the number of DEIM interpolation points. Regression methods, such as Gappy POD, can provide more accurate approximations for the nonlinear force vector and Jacobian. Unlike DEIM, which uses interpolation to ensure exact alignment with the solution at measurement points, Gappy POD employs regression and least squares to approximate nonlinear terms. We therefore recommend the adoption of Gappy POD in approximating nonlinear terms. The accuracy of LDEIM is highest for the input parameters within the training domain with minimum and maximum errors of 1.0% and 3.5%, respectively. For inputs outside of the training domain, the error increases to 9.2%. In addition, the ROM with LDEIM substantially reduces CPU time compared to the FOM. The speed-up ranges from 106 to 134 across different ROM inputs.

The performance of the ROM with LDEIM in cyclic plasticity models is also assessed when the load path is considered to be an input. Combining hyper-reduction techniques such as POD with LDEIM produces ROMs with reasonable accuracy that execute significantly faster than the FOM for cyclic loads, which differ from the training regime. The ROM clusters the training cyclic load path to better approximate the nonlinear force vector at each part of the online load path. Under FOMs with $\sim 10^3$ and $\sim 5 \times 10^4$ DoFs, the speed-ups range from 11 to 770 times, respectively.

This ROM successfully predicts the nonlinear mechanical behavior of porous media, but it does not capture the effect that pore pressure has on the deformations. The next step is to couple the ROM for plasticity with the diffusion-based ROM presented in Chapter 2 to develop a novel ROM formulation for hydro-mechanical poro-elasto-plastic problems.

# Chapter 4

# Parameterized ROM of Stimulated Volume Evolution in Large-scale Reservoirs

Following the derivation of an intrusive reduced order model (ROM) for a diffusion problem in Chapter 2 and for plasticity problems in Chapter 3, this chapter develops a ROM for poroplasticity problems. This ROM aims to significantly reduce the computational costs for nonlinear and fully-coupled hydro-mechanical simulations in large-scale reservoirs.

DEIM and its variants have been applied in various reservoir simulations [24, 39, 54, 91]. Ghasemi *et al.* significantly reduced computational costs by applying POD-DEIM to a two-phase flow and transport system [91]. Tan *et al.* combined POD-DEIM with trajectory piecewise linearization (TPWL) to interpolate the nonlinear terms in a two-dimensional heterogeneous reservoir simulation [54]. They demonstrated that this combination can lead to a more efficient and accurate ROM than the basic form of POD-DEIM. Ghadiri *et al.* used smooth orthogonal decomposition (SOD) instead of POD and combined it with DEIM to enhance the accuracy and speed-up of ROMs for well control optimization in two-dimensional and three-dimensional reservoirs [24]. Hatefi Ardakani *et al.* developed a parametric POD-DEIM for a layered reservoir with a two-well system [39]. They parameterized POD-DEIM for nonlinear injection rates and showed that ROM is efficient and accurate for injection rates inside and outside the training domain.

For systems characterized by a wide range of dynamic features and parameter variations, POD-DEIM becomes inefficient, as a large number of POD and DEIM basis vectors are required to sufficiently reduce the errors associated with the projection and the approximation of nonlinear terms. To address this limitation and achieve a further reduction,

one can partition the system of interest into multiple components with similar dynamic features and compute multiple local POD/DEIM (LPOD/LDEIM) basis functions for each part rather than relying on a single global basis function for the entire system [60]. Local reduced order models (LROMs) have found a few applications in reservoir simulation [92, 93]. Florez and Gildin proposed a ROM by manually time partitioning a linear/nonlinear thermal poroelastic system into subintervals and calculating local POD basis vectors for each local space [92]. Ghasemi and Gildin introduced a combination of LPOD and LDEIM to generate an efficient ROM for two-phase flow in reservoirs [93]. They neglected the effect of rock deformation in the reservoir and focused on a system of pressure-saturation equations, where the nonlinearity arose solely from the saturation equation.

In this chapter, an extension of LPOD-LDEIM is demonstrated to reduce the computational cost of numerical models in reservoirs with rock deformation. We introduce a novel LROM formulation for fully-coupled hydro-mechanical simulations of deformable porous media with single-phase fluid flow. A Drucker–Prager plasticity model, as a pressure-dependent yield criterion, is considered to describe the plastic behavior of sandstone. At the same time, the effect of stimulated dilation on porosity and permeability enhancement is investigated. A robust LROM is achieved by introducing distinct local DEIM subspaces for displacement and pressure equations. A new strategy is proposed for clustering and classification techniques that align with the coupled formulation framework. Furthermore, LPOD is applied to the pressure equation to reduce the size of the system of equations even further than global POD. In addition to material properties, the proposed LROM is parameterized with the injection rate. The effectiveness of LROM is evaluated for more realistic use cases, where LROM remains efficient for injection rates that extend beyond the training data.

The organization of this chapter is as follows: Section 4.1 describes CFOM for the hydraulic dilation stimulation problem, exploring the sources of nonlinearity, such as plastic deformation and stress-induced permeability enhancement. Section 4.2 discusses the proposed intrusive LROM for a fully-coupled hydro-mechanical problem. In Section 4.3, we evaluate the performance of LROM for a large-scale reservoir with a SAGD horizontal well pair. Finally, a summary of the work is presented in Section 4.4.

## 4.1 Mathematical model

This section describes the mathematical formulation for a system of reservoir-well experiencing hydraulic dilation stimulation. In this phenomenon, a short period of high-pressure fluid is injected into the injector and producer horizontal wells. Then, due to the shear dilation, the porosity and permeability of the vertical zone connecting the well pair drastically

increase. One step in the simulation of hydraulic dilation stimulation is to formulate the problem of coupled fluid flow and poroplasticity in a well-reservoir system. First, the governing equations of momentum and mass balances with boundary conditions for the poro-elasto-plastic model are discussed. Then, the coupled full order model consisting of a finite element analysis (FEA) of a two-dimensional poro-elasto-plastic domain is developed.

### 4.1.1 Poro-Elasto-Plastic Model

In this section, it is assumed that a deformable porous medium is saturated with a single-phase fluid. Pore pressure and bulk displacement fields are considered as the primary variables. According to Biot's formulation, two coupled governing equations are defined to describe the hydraulic-mechanical problem: linear momentum equation and mass balance [94, 95].

The first attempt is to define the linear momentum equation. The infinitesimal strain $\boldsymbol{\varepsilon}$ and the total stress $\boldsymbol{\sigma}$ are defined as

$$\boldsymbol{\varepsilon} = \frac{\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\top}\right)}{2} \tag{4.1}$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' - \alpha p \mathbf{I} \tag{4.2}$$

in which $\boldsymbol{\sigma}'$ and $p$ are the effective stress and the pore pressure, respectively. $\mathbf{I}$ is the unit tensor, and $\alpha$ represents the Biot's coefficient. The momentum balance equation with appropriate initial and boundary conditions is written as

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_b = \mathbf{0} \text{ in } \Omega \times [0, T] \tag{4.3}$$

where $\mathbf{f}_b$ represents the body force vector.

By substituting (4.2) into (4.3), the momentum balance equation is represented as

$$\nabla \cdot \boldsymbol{\sigma}' - \alpha \nabla \cdot (p\mathbf{I}) + \mathbf{f}_b = \mathbf{0} \text{ in } \Omega \times [0, T] \tag{4.4}$$

The Drucker–Prager plasticity model, as a pressure-dependent yield criterion, is considered to describe the plastic behavior of sandstone and to simulate the dilation:

$$\Phi\left(\boldsymbol{\sigma}', c\right) = \sqrt{J_2\left(\mathbf{s}\left(\boldsymbol{\sigma}'\right)\right)} + \eta p\left(\boldsymbol{\sigma}'\right) - \xi c = 0 \tag{4.5}$$

in which $\mathbf{s}$ is the deviatoric part of the effective stress. The plane strain condition is assumed to determine the parameters $\eta$ and $\xi$ as follows [79]:

$$\eta = \frac{3\tan\phi}{\sqrt{9+12\tan^2\phi}} \qquad \xi = \frac{3}{\sqrt{9+12\tan^2\phi}} \tag{4.6}$$

In (4.5), $c$ is the cohesion, defined by the hardening curve

$$c\left(\bar{\varepsilon}^p\right) = c_0 + H\bar{\varepsilon}^p \tag{4.7}$$

where $H$ is the hardening modulus, $c_0$ is the initial cohesion, and $\bar{\varepsilon}^p$ is the accumulated plastic strain. More details on the elasto-plastic constitutive equation can be found in [40].

The next governing equation is the mass balance equation defined as [95, 96]

$$\left(\frac{\phi}{K_f} + \frac{\alpha-\phi}{K_s}\right)\frac{\partial p}{\partial t} + \alpha\frac{\partial \varepsilon_v}{\partial t} + \nabla\cdot\mathbf{q} = \mathbf{0} \text{ in } \Omega\times[0,T] \tag{4.8}$$

$\varepsilon_v$ is the volumetric strain, $\phi$ is the porosity, and $K_s$ and $K_f$ are the solid and fluid bulk moduli, respectively. $\mathbf{q}$ is the mass flux defined using Darcy's law:

$$\mathbf{q} = -\frac{\kappa}{\mu_f}\nabla p \tag{4.9}$$

where $\kappa$ is the permeability of the reservoir, and $\mu_f$ represents the dynamic viscosity of the fluid.

To model the permeability enhancement due to shear dilation, a relation between permeability, porosity, and strain changes must be defined. There are several models in the literature [3, 97, 98, 99, 100]. According to [97, 100], for an anisotropic elaso-plastic model, a permeability change factor is defined based on volumetric plastic strain (dilation). However, for an isotropic elasto-plastic model, we can establish incremental relations between changes in permeability, porosity, and total strain (the sum of the elastic and plastic components of strain) [3, 98, 99]:

$$\phi^t = 1 - \left(1-\phi^{t-1}\right)e^{-\beta\Delta\varepsilon_v} \tag{4.10}$$

$$\kappa^t = \kappa^{t-1}\left(\frac{\phi^t}{\phi^{t-1}}\right)^{\gamma} \tag{4.11}$$

in which $\phi^t$ is the porosity at time step $t$, $\phi^{t-1}$ is the porosity at time step $t-1$, $\kappa^t$ is the permeability at time step $t$, $\kappa^{t-1}$ is the permeability at time step $t-1$, $\Delta\varepsilon_v$ is the volumetric strain change, and $\beta$ and $\gamma$ are constant coefficients.

### 4.1.2 Coupled Full Order Model

CFOM consists of an FEA of a two-dimensional poro-elasto-plastic domain. In the framework of FEM, the domain $\Omega$ is discretized by a mesh with $ne$ elements, $n_u^{\text{nodes}}$ nodes for the displacement field, and $n_p^{\text{nodes}}$ nodes for the pressure field. The finite element interpolations are utilized to approximate the trial solution for the state variables (bulk displacement fields $\mathbf{u}^h(\mathbf{x},t)$ and pore pressure $p^h(\mathbf{x},t)$) in a system of coupled partial differential equations (PDEs) at point $\mathbf{x}$ and time $t$:

$$\mathbf{u}^h(\mathbf{x},t) = \sum_{I=1}^{n_u^{\text{nodes}}} \left(N_u\right)_I (\mathbf{x})\mathbf{u}_I(t) = \mathbf{N}_u\mathbf{u}^t,\ \mathbf{x}\in\Omega,\ t\in[0,T] \tag{4.12}$$

$$p^h(\mathbf{x},t) = \sum_{I=1}^{n_p^{\text{nodes}}} \left(N_p\right)_I (\mathbf{x})p_I(t) = \mathbf{N}_p\mathbf{p}^t,\ \mathbf{x}\in\Omega,\ t\in[0,T] \tag{4.13}$$

in which $\mathbf{N}_u$ is a matrix of shape functions for displacement, $\mathbf{N}_p$ is a vector of shape functions for pressure, $\mathbf{u}^t$ is a vector of nodal displacements at time $t$, and $\mathbf{p}^t$ is a vector of nodal pressures at time $t$. CFOM has $n_u = 2\times n_u^{\text{nodes}}$ and $n_p = n_p^{\text{nodes}}$ DoFs for the displacement and pressure fields, respectively.

At time step $t$, a large nonlinear coupled system of equations must be solved for the nodal values of the unknowns $\mathbf{d}^t = \begin{bmatrix} \mathbf{u}^t \\ \mathbf{p}^t \end{bmatrix}$. The solution of the nodal unknowns $\mathbf{d}^t$ at time step $t$, given the solution $\mathbf{d}^{t-1}$ at the previous time step $t-1$, leads to a large discrete coupled system of equations of the form

$$\mathbf{R}\left(\mathbf{d}^t\right) = \mathbf{F}^{\text{ext}}(t) - \mathbf{F}^{\text{int}}\left(\mathbf{d}^t\right) = \mathbf{0} \tag{4.14}$$

$\mathbf{F}^{\text{ext}}$ is the external force vector

$$\mathbf{F}^{\text{ext}}(t) = \begin{bmatrix} \mathbf{F}_u^{\text{ext}}(t) \\ \Delta t\mathbf{F}_p^{\text{ext}}(t) \end{bmatrix} = \begin{bmatrix} \int_{\Gamma_t} {\mathbf{N}_u}^\top \bar{\mathbf{t}}\,\mathrm{d}\Gamma + \int_\Omega {\mathbf{N}_u}^\top \mathbf{f}_b\,\mathrm{d}\Omega \\ -\Delta t \int\limits_{\Gamma_q} \bar{\mathbf{q}}{\mathbf{N}_p}^\top d\Gamma \end{bmatrix} \tag{4.15}$$

in which $\mathbf{F}_u^{\text{ext}}$ and $\mathbf{F}_p^{\text{ext}}$ are the mechanical and hydraulic parts of the external force vector, respectively. $\Gamma_t$ and $\Gamma_q$ are the portions of the boundary where tractions $\bar{\mathbf{t}}$ and $\bar{\mathbf{q}}$ are prescribed, respectively.

In (4.14), $\mathbf{F}^{\text{int}}$ is the internal force vector

$$\mathbf{F}^{\text{int}}(t) = \begin{bmatrix} \overbrace{\mathbf{F}_u^{\text{int}}\left(\mathbf{u}^t\right)}^{\mathbf{F}_u} - \mathbf{Q}\mathbf{p}^t \\ \Delta t \underbrace{\mathbf{F}_p^{\text{int}}\left(\mathbf{p}^t\right)}_{\mathbf{F}_p} + \mathbf{Q}^\top\left(\mathbf{u}^t - \mathbf{u}^{t-1}\right) + \mathbf{S}^t\mathbf{p}^t - \mathbf{S}^{t-1}\mathbf{p}^{t-1} \end{bmatrix} \tag{4.16}$$

in which $\mathbf{F}_u$ and $\mathbf{F}_p$ are defined as

$$\mathbf{F}_u\left(\mathbf{u}^t\right) = \int_\Omega {\mathbf{B}_u}^\top \boldsymbol{\sigma}'\left(\mathbf{u}^h\right) \mathrm{d}\Omega \tag{4.17}$$

$$\mathbf{F}_p\left(\mathbf{p}^t\right) = \int_\Omega {\mathbf{B}_p}^\top \mathbf{q}\left(p^h\right) \mathrm{d}\Omega \tag{4.18}$$

where $\mathbf{B}_u$ and $\mathbf{B}_p$ are the matrices of derivatives of the standard shape functions for the displacement and pressure fields, respectively. The details of the finite element discretization for poroplasticity problems are given in Appendix B.

The matrices $\mathbf{Q}$ and $\mathbf{S}$ are the coupling and compressibility matrices, written as [94, 101]

$$\mathbf{Q} = \int_\Omega {\mathbf{B}_u}^\top \alpha \mathbf{m} \mathbf{N}_p \mathrm{d}\Omega \tag{4.19}$$

$$\mathbf{S}^t = \int_\Omega {\mathbf{N}_p}^\top \left(\frac{\phi^t}{K_f} + \frac{\alpha - \phi^t}{K_s}\right) \mathbf{N}_p \mathrm{d}\Omega \tag{4.20}$$

In FEM, the nonlinear system of equations (4.14) is solved iteratively by Newton–Raphson method, such that the increment, $\Delta\mathbf{d}^{t(j)}$, in the nodal values at time step $t$ and iteration $j$ is

$$\mathbf{K}^{t(j)} \Delta\mathbf{d}^{t(j)} = \mathbf{R}^{t(j)} \tag{4.21}$$

and the updated solution at time $t$ is obtained as

$$\mathbf{d}^{t(j+1)} = \mathbf{d}^{t(j)} + \Delta\mathbf{d}^{t(j)} \tag{4.22}$$

In (4.21), $\mathbf{R}^{t(j)} = \mathbf{R}\left(\mathbf{d}^{t(j)}\right)$ and $\mathbf{K}^{t(j)} = \mathbf{K}\left(\mathbf{d}^{t(j)}\right)$ are the residual force vector and the Jacobian matrix at time step $t$ and iteration $j$, respectively. $\mathbf{K}^{t(j)}$ is defined as

$$\mathbf{K}^{t(j)} = -\left.\frac{\partial \mathbf{R}^{t(j)}}{\partial \mathbf{d}}\right|_{\mathbf{d}^{t(j)}} = \begin{bmatrix} \mathbf{K}_u & -\mathbf{Q} \\ \mathbf{Q}^\top & \underbrace{\mathbf{S} + \Delta t \mathbf{H}}_{\mathbf{K}_p} \end{bmatrix} \tag{4.23}$$

where

$$\mathbf{K}_u = \int_\Omega {\mathbf{B}_u}^\top \mathbf{D}^{ep} \mathbf{B}_u \mathrm{d}\Omega \tag{4.24}$$

$$\mathbf{H} = \int_\Omega {\mathbf{B}_p}^\top \frac{\kappa\left(p^h\right)}{\mu} \mathbf{B}_p \mathrm{d}\Omega \tag{4.25}$$

$\mathbf{D}^{ep}$ is the algorithmic tangent modulus in elasto-plastic constitutive models, which is defined in [40, 79, 80]. CFOM is computationally expensive due to the nonlinear multiphysics phenomena involved. In fact, an extremely large nonlinear coupled system of equations must be solved at each iteration. An efficient hyper-reduced ROM drastically reduces the costs associated with both the assembly process of nonlinear terms and the solution of the system of equations.

## 4.2 Intrusive Local Reduced Order Model

In this section, we describe multiple reduction methods to reduce the computational complexity related to the simulation of the permeability enhancement due to shear dilation. It includes the need for POD-G to reduce the number of displacement and pressure DoFs, the need for DEIM to accelerate the computation of nonlinear terms, and the need for local ROM (LPOD/LDEIM) to apply a further reduction for nonlinear coupled transient problems.

### 4.2.1 Proper Orthogonal Decomposition

POD is performed to replace the solution of (4.14) and (4.21) for a large number of DoFs $n_u$ and $n_p$ with the solution of a computationally efficient ROM for a smaller number of DoFs $m_u$ and $m_p$ ($m_u \ll n_u$ and $m_p \ll n_p$). Galerkin projection method is utilized to approximate the vectors $\mathbf{u}^t \in \mathbb{R}^{n_u}$ and $\mathbf{p}^t \in \mathbb{R}^{n_p}$ by means of a linear combination of the elements of the projection matrices $\mathbf{W}_u \in \mathbb{R}^{n_u \times m_u}$ and $\mathbf{W}_p \in \mathbb{R}^{n_p \times m_p}$, respectively:

$$\mathbf{u}^t = \mathbf{W}_u \widetilde{\mathbf{u}}^t \tag{4.26}$$

$$\mathbf{p}^t = \mathbf{W}_p \widetilde{\mathbf{p}}^t \tag{4.27}$$

where $\widetilde{\mathbf{u}}^t \in \mathbb{R}^{m_u}$ and $\widetilde{\mathbf{p}}^t \in \mathbb{R}^{m_p}$ denote the displacement and pressure reduced vectors of DoFs for ROM at time $t$. Equations (4.26) and (4.27) can be written in a coupled form

$$\mathbf{d}^t = \mathbf{W} \widetilde{\mathbf{d}}^t \tag{4.28}$$

where $\widetilde{\mathbf{d}}^t = \begin{bmatrix} \widetilde{\mathbf{u}}^t \\ \widetilde{\mathbf{p}}^t \end{bmatrix}$ is the nodal values of the unknowns in the reduced space. $\mathbf{W}$ is called the coupled projection matrix and is presented as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_p \end{bmatrix} \tag{4.29}$$

Assume that the samples of displacement and pressure vectors are stored at the end of each time step as

$$\mathbb{S}_u = \{\mathbf{u}_1, ..., \mathbf{u}_{n_s}\} \in \mathbb{R}^{n_u \times n_s} \tag{4.30}$$

$$\mathbb{S}_p = \{\mathbf{p}_1, ..., \mathbf{p}_{n_s}\} \in \mathbb{R}^{n_p \times n_s} \tag{4.31}$$

in which $n_s$ is the number of time steps. The quality of approximations (4.26) and (4.27) is affected by the choice of basis vectors $\mathbf{W}_u$ and $\mathbf{W}_p$. Using singular value decomposition (SVD), the projection matrices are selected by truncating POD basis vectors with a low-energy approximation [40].

The displacement and pressure equations at time step $t$ and iteration $j$ are now solved in the reduced space by substituting (4.28) into (4.21) and multiplying both sides by $\mathbf{W}^\top$:

$$\widetilde{\mathbf{K}}^{t(j)}\Delta\widetilde{\mathbf{d}}^{t(j)} = \widetilde{\mathbf{R}}^{t(j)} \tag{4.32}$$

where the updated reduced solution is obtained as

$$\widetilde{\mathbf{d}}^{t(j+1)} = \widetilde{\mathbf{d}}^{t(j)} + \Delta\widetilde{\mathbf{d}}^{t(j)} \tag{4.33}$$

in which $\widetilde{\mathbf{R}}^{t(j)} \in \mathbb{R}^{m_u+m_p}$ and $\widetilde{\mathbf{K}}^{t(j)} \in \mathbb{R}^{(m_u+m_p)\times(m_u+m_p)}$ are the reduced residual force vector and the reduced Jacobian matrix

$$\widetilde{\mathbf{R}}^{t(j)} = \mathbf{W}^\top\mathbf{R}^{t(j)} \tag{4.34}$$

$$\widetilde{\mathbf{K}}^{t(j)} = \mathbf{W}^\top\mathbf{K}^{t(j)}\mathbf{W} \tag{4.35}$$

So far, ROM construction has been discussed for the linearized system (4.14); however, there are additional challenges that appear for nonlinear transient coupled problems where there are several sources of nonlinearity, such as plasticity and permeability enhancement. In this case, the nonlinear components $\mathbf{R}^{t(j)}$ and $\mathbf{K}^{t(j)}$ must be calculated online at each iteration $j$, and the computational cost of their assembly scales with the dimension of the CFOM. This issue can be addressed by DEIM, as discussed in the next section.

### 4.2.2 Discrete Empirical Interpolation Method

The same formulations provided in Section 3.3.2 are used for the fast approximation of the nonlinear terms $\mathbf{F}_u$, $\mathbf{F}_p$, $\mathbf{K}_u$, and $\mathbf{K}_p$ at time $t$ and iteration $j$ in (4.16) and (4.23). Here, it is necessary to define separate measurement matrices ($\mathbf{P}_u$ and $\mathbf{P}_p$), measurement points ($q_u$ and $q_p$), and DEIM interpolation matrices ($\mathbf{A}_u$ and $\mathbf{A}_p$) for the displacement and pressure contributions of the nonlinear terms.

In this chapter, we adopt the first method, as detailed in Section 2.3.2, to approximate the large nonlinear functions $\mathbf{K}_u$ and $\mathbf{K}_p$ using the same POD basis vectors that are determined for the nonlinear terms $\mathbf{F}_u$ and $\mathbf{F}_p$ [30, 40, 51].

The POD-DEIM provides a significant reduction in computational cost, particularly for problems with a small number of measurement points $q_u$ and $q_p$. However, for certain

problems, such as permeability enhancement due to shear dilation, a large number of measurement points are required to sufficiently reduce the error associated with the approximation of nonlinear terms, as depicted in Section 4.3.1. To address this limitation and achieve further reduction, one can partition the system of interest into multiple components with similar dynamic features and compute multiple local POD/DEIM (LPOD/LDEIM) basis functions for each part rather than relying on a single global basis function for the entire system.

### 4.2.3 Localized Discrete Empirical Interpolation Method

The main idea is to partition the snapshot matrices $\mathbb{S}_{\mathbf{F}_u}$ and $\mathbb{S}_{\mathbf{F}_p}$ into multiple groups and apply DEIM to each domain to obtain local DEIM interpolation matrices:

$$\mathbf{A}_u^c = \mathbf{Z}_u^c (\mathbf{P}_u^{c\top} \mathbf{Z}_u^c)^{-1} \tag{4.36}$$

$$\mathbf{A}_p^c = \mathbf{Z}_p^c (\mathbf{P}_p^{c\top} \mathbf{Z}_p^c)^{-1} \tag{4.37}$$

where $\mathbf{Z}_u^c$, $\mathbf{Z}_p^c$, $\mathbf{P}_u^c$, and $\mathbf{P}_p^c$ are the DEIM basis vectors and the measurement matrices corresponding to the displacement and the pressure contributions of the nonlinear terms for the $c^{th}$ cluster, respectively. The clustering process provides a smaller number of measurement points for each domain compared to global DEIM, leading to a fast online phase for LROM. The same offline/online procedure provided in Section 3.3.3 is used for the construction of LDEIM. $k$-means clustering algorithm and nearest neighbor classifier are used for offline and online stages, respectively [60].

$k$-means clustering algorithm is used to classify the snapshots $\mathbb{S}_{\mathbf{F}_u}$ and $\mathbb{S}_{\mathbf{F}_p}$ into different clusters:

$$\mathbb{S}_{\mathbf{F}_u} = \left\{ \mathbb{S}_{\mathbf{F}_u}^1, \ldots, \mathbb{S}_{\mathbf{F}_u}^{n_c} \right\} \tag{4.38}$$

$$\mathbb{S}_{\mathbf{F}_p} = \left\{ \mathbb{S}_{\mathbf{F}_p}^1, \ldots, \mathbb{S}_{\mathbf{F}_p}^{n_c} \right\} \tag{4.39}$$

in which the superscript $n_c$ denotes the number of clusters. It is important to note that the clustering process is done based on the snapshot $\mathbb{S}_{\mathbf{F}_u}$. After the convergence of the algorithm, the same clusters are applied to the snapshots $\mathbb{S}_{\mathbf{F}_p}$. In the algorithm, the inputs are snapshot $\mathbb{S}_{\mathbf{F}_u}$, $\mathbb{S}_{\mathbf{F}_p}$, and $n_c$, and the outputs are local DEIM interpolation matrices ($\mathbf{A}_u^c$

and $\mathbf{A}_p^c$) and a classifier function showing the mapping between each column of snapshot matrices ($\mathbf{F}_{u_i}$ and $\mathbf{F}_{p_i}$) and cluster number $c$: $\textbf{Class} : \mathbf{F}_{u_i}, \mathbf{F}_{p_i} \rightarrow c$. The procedure for $k$-means clustering algorithm was detailed in Section 3.3.3.

Our attention now turns to the online phase of LDEIM. The objective is to use the nearest neighbor classifier as a computationally efficient search algorithm to select the most appropriate cluster with respect to the current state of the system. Similar to the offline phase, the online classification process relies on the displacement contribution of the nonlinear force vectors. The procedure for the nearest neighbor classifier was detailed in Section 3.3.3.

### 4.2.4 Localized Proper Orthogonal Decomposition

The main idea is to partition the snapshot matrices of the primary variables into multiple clusters and apply POD to each cluster to obtain local POD basis vectors. This localization method allows to reduce the size of the system of the equations even further compared to global POD. Furthermore, it provides us with a better estimation of the solution in each cluster. Here, LPOD is applied only in the pressure equation. Based on our experience, in elasto-plastic hydromechanical problems with permeability enhancement, where the nonlinearity is due to mechanical and hydraulic parts of the coupled system of equations, localizing the displacement equation negatively impacts the stability of the Newton–Raphson algorithm, while localizing the pressure equation enhances the stability. A possible explanation for this behavior is that the displacement space does not approximate the inf-sup supremizer effectively. It is expected that a supremizer stabilization technique will address this issue by ensuring that the LBB condition is satisfied in the ROM.

The presented LPOD follows an offline-online decomposition. Offline, the clustering process for the pressure equation follows the same clusters determined in LDEIM. Thus, there is no need to rerun $k$-means clustering algorithm for the pressure snapshot. The same number of clusters $n_c$ and classifier functions ($\textbf{Class} : \mathbf{p}_i \rightarrow c$) are utilized to classify the pressure snapshot $\mathbb{S}_p$ into multiple clusters:

$$\mathbb{S}_p = \left\{ \mathbb{S}_p^1, \ldots, \mathbb{S}_p^{n_c} \right\} \tag{4.40}$$

POD is applied to each domain to obtain local projection matrices

$$\mathbf{W}^c = \begin{bmatrix} \mathbf{W}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_p^c \end{bmatrix} \tag{4.41}$$

where $\mathbf{W}_u$ is the global projection matrix for the displacement equation, and $\mathbf{W}_p^c$ is the local projection matrix for the pressure equation associated with the $c^{th}$ cluster.

In the online phase of LPOD, the projection matrix $\mathbf{W}^c$ is updated based on the current state of the system. At the end of the time step $t-1$, once the updated cluster number $c^t$ is determined using the nearest neighbor classifier, the appropriate local projection matrix is used to solve the reduced order coupled system of equations for the next time step:

$$\begin{aligned}\widetilde{\mathbf{R}}\left(\widetilde{\mathbf{d}}^t\right) &= \mathbf{W}^{c\top}\mathbf{F}^{\text{ext}}(t) \\ &- \mathbf{W}^{c\top}\begin{bmatrix} \hat{\mathbf{F}}_u(\widetilde{\mathbf{u}}^t) - \mathbf{Q}\mathbf{W}_p^c\widetilde{\mathbf{p}}^t \\ \Delta t\hat{\mathbf{F}}_p(\widetilde{\mathbf{p}}^t) + \mathbf{Q}^\top\mathbf{W}_u(\widetilde{\mathbf{u}}^t - \widetilde{\mathbf{u}}^{t-1}) + \mathbf{S}^t\mathbf{W}_p^c\widetilde{\mathbf{p}}^t - \mathbf{S}^{t-1}\mathbf{W}_p^c\widetilde{\mathbf{p}}^{t-1} \end{bmatrix} = \mathbf{0}\end{aligned} \tag{4.42}$$

$$\widetilde{\mathbf{K}}\left(\widetilde{\mathbf{d}}^t\right) = \mathbf{W}^{c\top}\begin{bmatrix} \hat{\mathbf{K}}_u(\widetilde{\mathbf{u}}^t) & -\mathbf{Q} \\ \mathbf{Q}^\top & \hat{\mathbf{K}}_p(\widetilde{\mathbf{p}}^t) \end{bmatrix}\mathbf{W}^c \tag{4.43}$$

In Figure 4.1, the offline/online procedure of the presented LROM is summarized.

## 4.3 Results and discussion

Hydraulic dilation stimulation of SAGD is a complex phenomenon in geomechanics due to its temporal and spatial behavior. Before the start of SAGD production, a short period of high-pressure fluid is injected into the injector and producer horizontal wells. Due to the shear dilation, the porosity and permeability of the vertical zone connecting the well pair drastically increase. We use this phenomenon to show the capability of the presented LROM, in which the material properties and injection rate are parametric inputs. Two prominent examples are considered. In the first example, a parameterized LROM with two material inputs (hardening modulus and initial permeability) is generated. It is demonstrated that using LPOD/LDEIM, a numerically stable LROM can be generated. The second example considers a possible applied scenario where the injection rate is not constant. LROM is trained and tested by adding another parametric input: injection rate.

The material distribution is assumed to be uniform along SAGD well length. Thus, numerical simulations are based on two-dimensional models cut perpendicular to SAGD well pair. As illustrated in Figure 4.2a, CFOMs are created for a 200m $\times$ 200m domain with the given BCs. SAGD well pair, with a diameter of 0.2 m, is located in the middle of the model, and

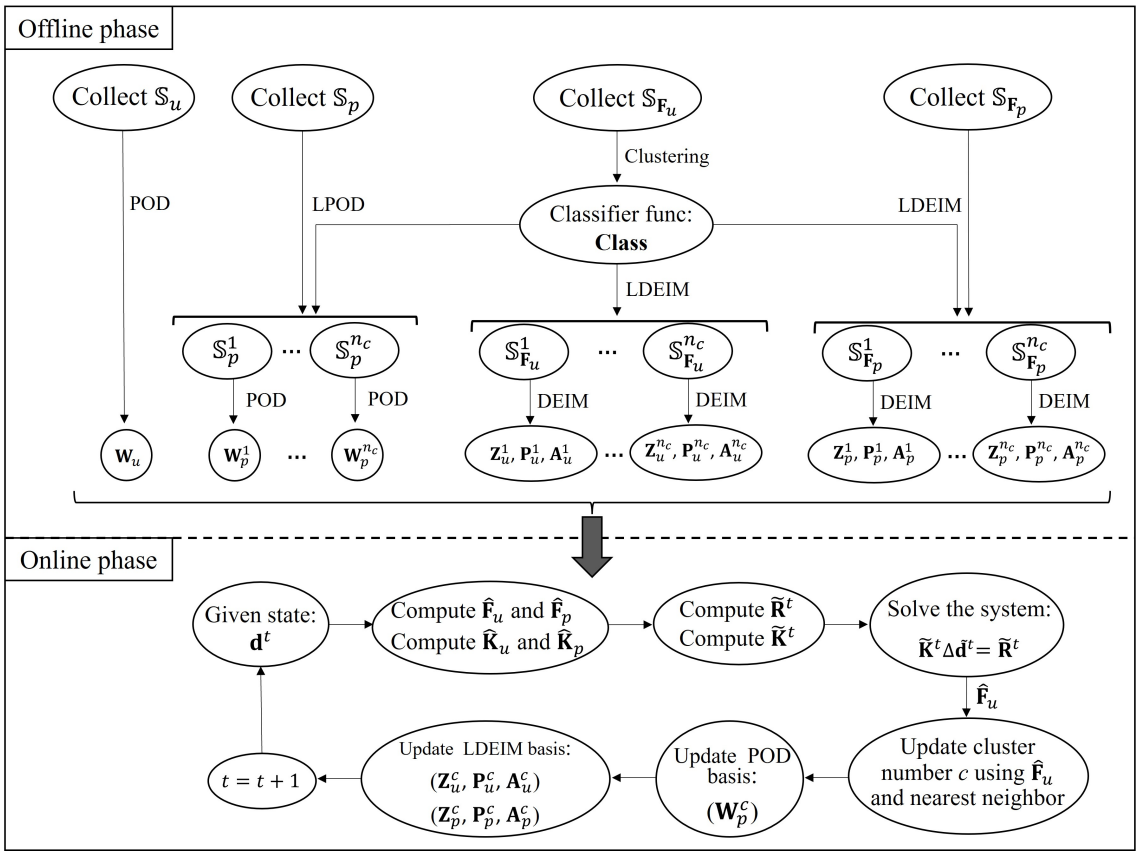



Figure 4.1: Offline/Online procedure of the presented LROM.

the distance between wells is 5 m. The size of the domain is chosen to be sufficiently large to ensure it is far from the plastic zone distributed around the well. As shown in Figure 4.2b, the size of the plastic zone is approximately 2.8 m. Therefore, the domain size is set to be about 70 times larger than the plastic zone, ensuring that the boundary conditions are adequately distanced from the stimulated zone. The fluid is injected at a rate of 18 L/s at the boundaries of both injector and producer wells, as shown in Figure 4.2a. The domain is subject to uniform in-situ stresses with a minimum horizontal stress, $\sigma_{Hmin} = 8$ MPa, a maximum horizontal stress, $\sigma_{Hmax} = 7$ MPa, and a vertical stress, $\sigma_V = 9$ MPa. The stresses are approximately equivalent to the stress fields at SAGD depth. The initial pore pressure also equals 4 MPa. Table 4.1 summarizes the material properties, including hydraulic and elasto-plastic mechanical properties. The loading process is performed in injection rate-controlled mode. The injection rate is the model input as the boundary condition, and the pressure is the output of the simulation.

Table 4.1: Material properties for the two-dimensional injection wells problem

| Symbol | Value | Unit | Symbol | Value | Unit |
|---|---|---|---|---|---|
| $E$ | 10 | GPa | $\mu_f$ | $10^{-12}$ | GPa·s |
| $\nu$ | 0.38 | - | $K_s$ | 36 | GPa |
| $c_0$ | 6 | MPa | $K_f$ | 3.3 | GPa |
| $\varphi$ | 20° | - | $\alpha$ | 1 | - |
| $H$ | 42 | MPa | $\beta$ | 3 | - |
| $k_0$ | 5 | mD | $\gamma$ | 20 | - |
| $\phi_0$ | 0.34 | - | | | |

The large physical size of the reservoir (200m $\times$ 200m) and the small size of the wells (0.2 m) lead to many DoFs in CFOM simulations. Figure 4.2c depicts the fine mesh considered for the FE training process. The reservoir is discretized using 4-node quadrilateral elements (Q4) instead of Taylor–Hood elements due to the failure to meet the Ladyzhenskaya–Babuška–Brezzi (LBB) condition in the reduced order space. When Taylor–Hood elements are used, pressure instabilities related to spurious pressure POD modes are observed in LROM. However, no LBB instabilities are observed in both CFOM and LROM after switching to the same-order elements. This issue has also been noted in computational fluid dynamics (CFD) applications [102, 103]. Ballarin *et al.* demonstrated that the POD-Galerkin approximation of incompressible Navier–Stokes equations becomes unstable when Taylor–Hood finite element spaces are employed in ROM [102]. They proposed a supremizer stabilization technique to ensure that the LBB condition is met in ROM. Although implementing the supremizer stabilization technique in porous media is beyond the scope of this work, it represents an area of interest for future studies.

### 4.3.1 Example 1: Parameterized ROM with material inputs

The objective is to compare the performance of the presented LROM parameterized with two material inputs, i.e., hardening modulus and initial permeability, with that of CFOM. The training domain for hardening modulus and initial permeability are considered in the ranges of $H \in [40, 60]$ MPa and $\kappa_0 \in [5, 10]$ mD, respectively. It is important to note that the injection rate is not a parametric input for LROM in this example. As depicted in Figure 4.3, LROM is trained using only 10 samples in the two-dimensional parameter space. The distribution of samples is not entirely uniform. From our observations, reservoirs with low permeability tend to experience more plastic deformation around the wells than those with higher permeability. Therefore, we have concentrated more samples near the lower

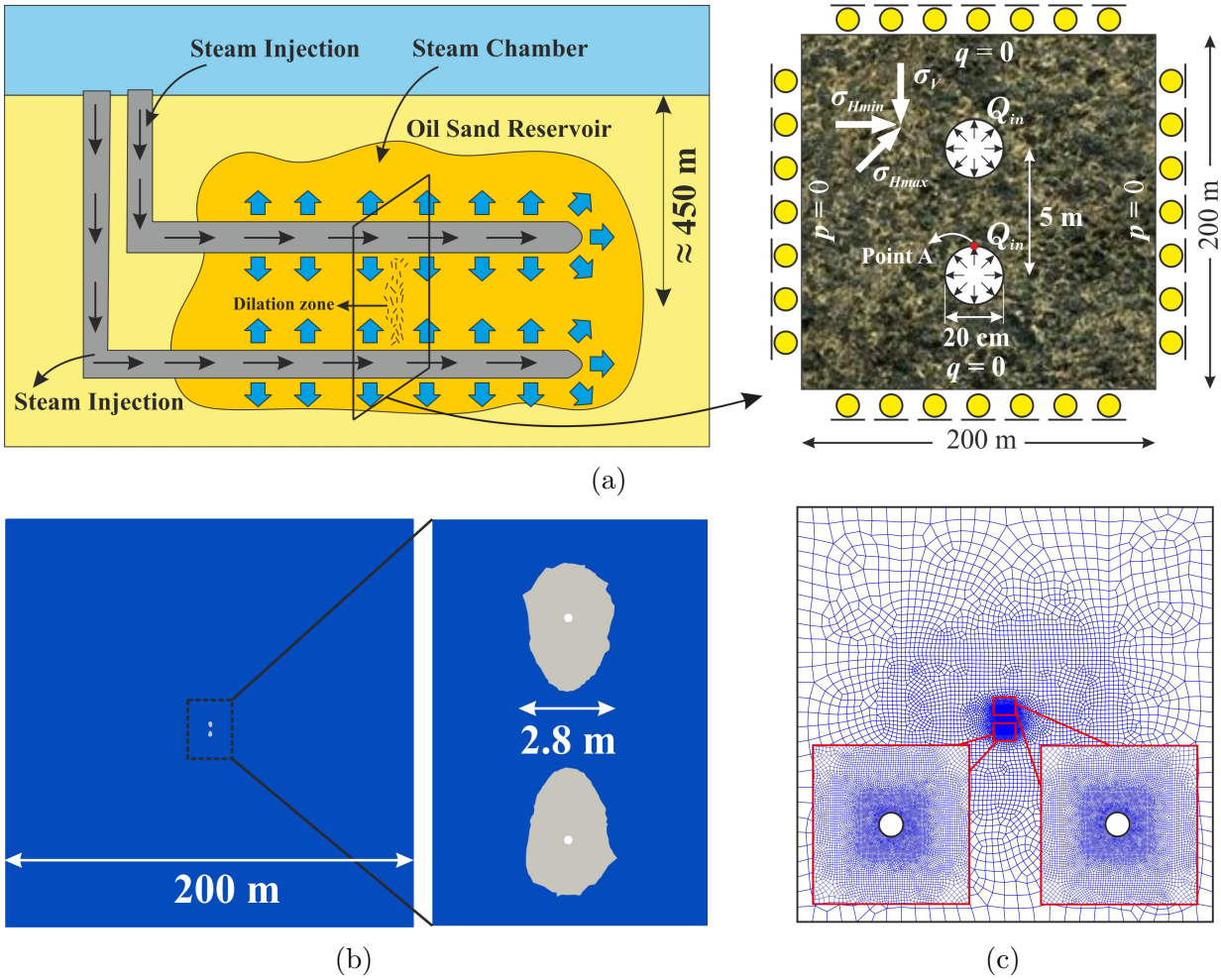



Figure 4.2: a) Two-dimensional hydraulic stimulation problem, b) comparison of the domain size to the size of the stimulated zone (accumulated plastic strain zone), c) FEM mesh with 32993 nodes and 32848 elements.

bound of the permeability range. It should be noted that the sampling method can be automated and less dependent on the user's knowledge of the problem. However, automated sampling methods, such as a uniform distribution of sampling points across the training domain, will require significantly more samples, resulting in a substantial increase in the offline computational cost for data generation. In contrast, leveraging knowledge of the problem helps reduce the number of required samples while maintaining accuracy.

To apply LROM, the reservoir is simulated with a training schedule for 10 days, and

snapshots of displacement, pressure, and nonlinear terms are stored every 36 minutes for each sample. Thus, we have 4000 snapshots for each variable. Next, an LROM is constructed using snapshots, as explained in Section 4.2. Finally, the surrogate model is tested online for the inputs inside the training domain.

the sampling method can be automated and made less reliant on the user's knowledge of the problem. However, automated sampling methods, such as a uniform distribution of sampling points across the training domain, will significantly increase the required number of samples. This will lead to a substantial increase in the offline computational cost for data generation. In contrast, leveraging knowledge of the problem helps reduce the number of required samples while maintaining accuracy. For example, in Section 4.3.1, we used a non-uniform distribution of samples within the training domain based on our knowledge of the problem. Specifically, we observed that reservoirs with lower permeability tend to exhibit greater plastic deformation around the wells compared to those with higher permeability. Based on this observation, we concentrated more samples near the lower bound of the permeability range. Adopting an automated sampling method instead will require at least twice as many samples to achieve comparable accuracy, resulting in a substantial increase in the offline computational cost.

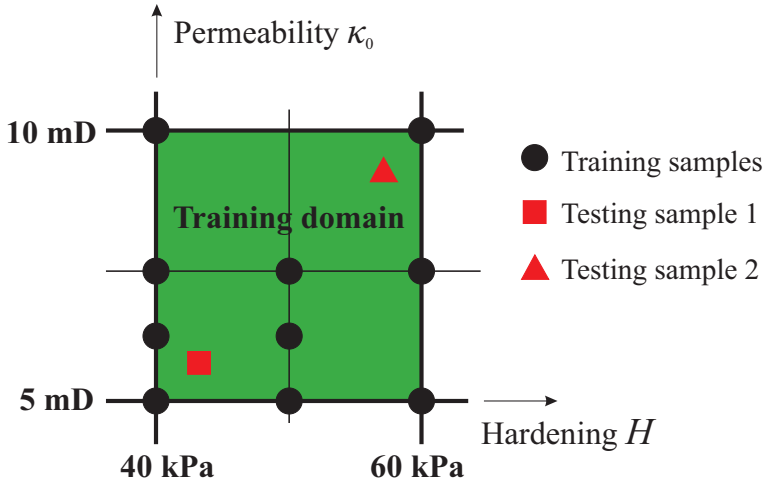



Figure 4.3: Ten samples non-uniformly distributed in the two-dimensional training domain.

By applying POD, the size of ROM for the displacement field is reduced from 65986 DoFs to 90 DoFs ($k_u = 90$). Figure 4.4 depicts the projection error for the displacement field as a function of POD mode. As can be seen, to achieve a POD projection error of $5 \times 10^{-3}\%$, the first 90 POD modes with the highest energy are kept, and the remaining modes are truncated. The projection error is defined as

$$Err_{\mathrm{Prj}}(k) = \left(1 - E_{\mathrm{Acc}}^{\mathrm{POD}}(k)\right) \times 100 \tag{4.44}$$

where $E_{\mathrm{Acc}}^{\mathrm{POD}}(k)$ refers to the cumulative normalized energy of the first $k$ POD modes. Based on our experience, in elasto-plastic hydromechanical problems with permeability enhancement, where the nonlinearity is due to mechanical and hydraulic parts of the coupled system of equations, localizing the displacement equation negatively impacts the stability of the Newton–Raphson algorithm, while localizing the pressure equation enhances the stability. The pressure field is localized by dividing its snapshot into 8 clusters. Figure 4.4 depicts the projection error for the pressure field using POD and LPOD derived from SVD analysis of the pressure snapshots. LPOD demonstrates a faster decay in the singular values of the mode shapes than global POD. Thus, LPOD allows for the truncation of a large number of singular values within each cluster. To achieve a POD projection error of $10^{-6}\%$, the first 10 POD modes with the highest energy are kept in each cluster, and the remaining modes are truncated. Thus, using LPOD, the size of ROM for the pressure field is reduced from 32993 DoFs to 10 DoFs ($k_p = 10$).

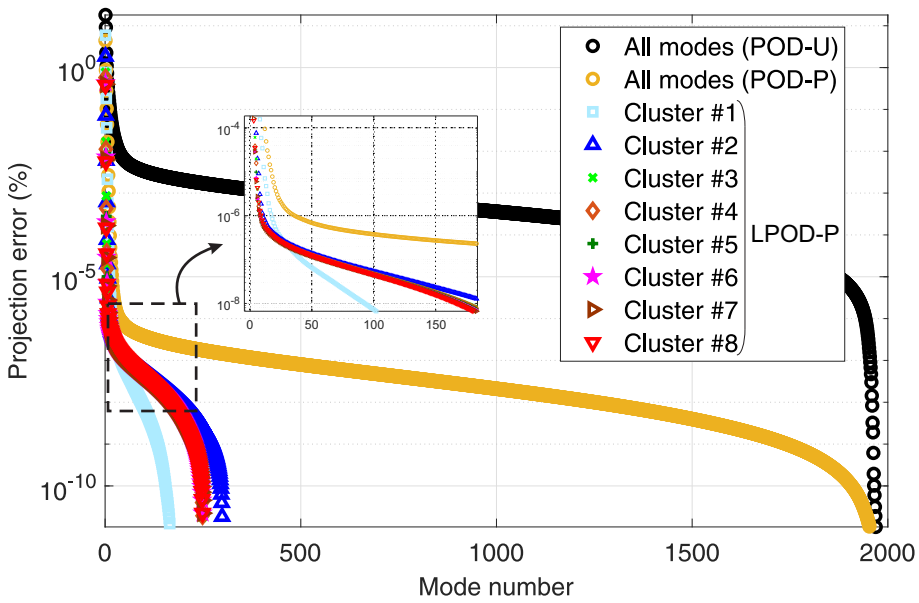



Figure 4.4: Projection error for the displacement and pressure equations using POD and LPOD.

It is important to note that a large number of measurement points $q$ (at least 2000) are required in global DEIM to ensure that the Newton–Raphson algorithm in ROM

converges to a correct solution. However, LDEIM drastically reduces the number of required measurement points while maintaining an accurate and stable ROM. The improved stability of the Newton–Raphson solver observed with LDEIM can be explained by the interpolation error in DEIM and LDEIM. Figure 4.5 illustrates DEIM interpolation error for both the mechanical and hydraulic parts of the nonlinear force snapshots resulting from SVD analysis. DEIM interpolation error is defined as

$$Err_{\text{Inp}}\left(q\right) = \left(1 - E_{\text{Acc}}^{\text{DEIM}}\left(q\right)\right) \times 100 \tag{4.45}$$

where $E_{\text{Acc}}^{\text{DEIM}}\left(q\right)$ refers to the cumulative normalized energy of the first $q$ DEIM modes. Figure 4.5 shows that the energy of modes in DEIM decays significantly slower than in LDEIM. Consequently, global DEIM requires a large number of measurement points to accurately approximate the nonlinear force vectors, whereas LDEIM approximations achieve this accuracy with significantly fewer measurement points.

In this example, the converged coupled nonlinear force vectors are grouped into 8 clusters. As explained in Section 4.2.3, clustering is based on the displacement part of the snapshots for the nonlinear force vector $\mathbb{S}_{\mathbf{F}_u}$ and the same clusters are subsequently applied to the pressure part of the snapshots $\mathbb{S}_{\mathbf{F}_p}$. Each cluster in LDEIM requires 30 mode shapes $((q_u = 30) \ll n_u$ and $(q_p = 30) \ll n_p)$ to accurately interpolate the nonlinear force vectors.

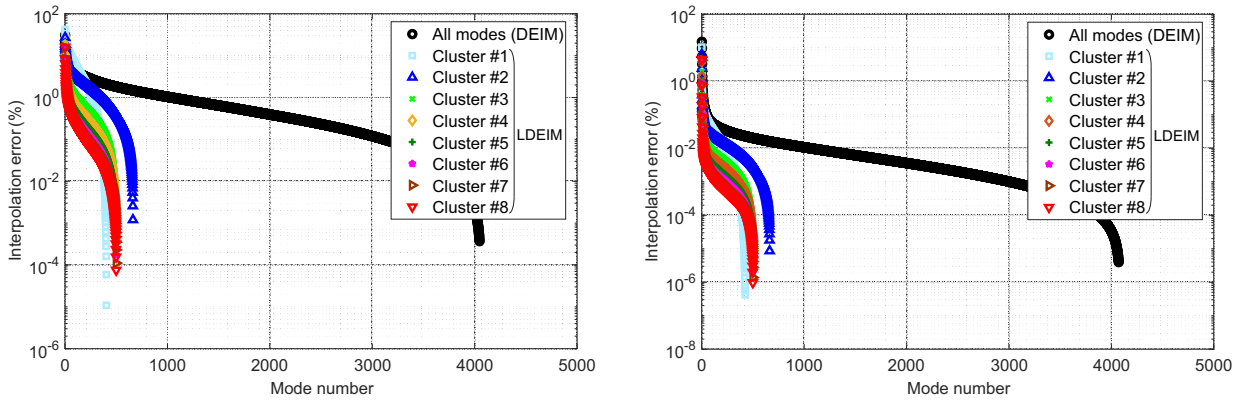



Figure 4.5: DEIM interpolation error for both the mechanical and hydraulic parts of the nonlinear force snapshots, a) mechanical part, b) hydraulic part.

The accuracy and efficiency of LROM are evaluated for inputs inside the training domain in Figure 4.6. The relative $L_2$ error of LROM for displacement and pressure equations

is demonstrated in Figures 4.6a and 4.6b, respectively. LROM accurately captures the reservoir responses (displacement and pressure) using a small ROM size ($k_u = 90$ and $k_p = 10$) and a small number of measurement points for each cluster ($q_u = 30$ and $q_p = 30$). The accuracy ranges from 0.2% to 4.5% for the displacement variable and from 0.2% to 4.1% for the pressure variable across different material inputs. The highest accuracy is observed for the upper bound of the initial permeability range. However, the error increases as the initial permeability decreases. Lower initial permeabilities lead to more significant plastic deformation in the formation, decreasing the accuracy of the Newton–Raphson algorithm in LROM.

A key metric in the evaluation of the performance of a ROM is the speed-up factor. Computational speed-up (with reference to CFOM) is illustrated in Figure 4.6c for different inputs within the training domain. LROM significantly reduces CPU time compared to CFOM, with speed-up factors ranging from 368 to 402 across the material parameter space. The maximum speed-up factor of 402 corresponds to the upper bounds of inputs in the training domain. Higher initial permeability leads to lower plastic deformation around the well. This lower level improves the Newton–Raphson performance by reducing the number of iterations per time step.

Figure 4.7 illustrates the displacement in $x$ direction $u_x$, the displacement in $y$ direction $u_y$, and pore pressure $p$ along the cross-section S. These are the primary variables in the governing equations. A vertical cross-section is used for $u_y$ and $p$, while a horizontal cross-section is used for $u_x$. There is a good agreement between CFOM results and the solution of LROM for two sets of the input parameters (Test 1: $H = 42$ MPa and $\kappa_0 = 5.2$ mD and Test 2: $H = 58$ MPa and $\kappa_0 = 8$ mD). As shown, LROM reproduces the complex responses of the reservoir, particularly those near the wellbore. Using (4.26) and (4.27), LROM computes the primary variables in the entire reservoir and satisfy the boundary conditions on the boundary of the domain. Discontinuities in the graphs indicate the location of wells. It should be mentioned that an initial pore pressure of 4 MPa must be added to the contour values to represent the total pore pressure.

Figure 4.8 demonstrates the ability of LROM to calculate the secondary variables in the governing equations. The figure depicts the stimulated dilation zone, represented as the accumulated plastic strain $\bar{\varepsilon}^p$, along with the permeability $\kappa$ and porosity $\phi$, after 10 days of fluid injection for online test case 1. As expected, the stimulated volume grows symmetrically from both wells in the direction of the maximum in-situ stress, defined by a region of increased porosity and permeability. There is considerable agreement between the contours produced by CFOM and LROM. However, as seen in all LROM's contours, the gray zone indicates areas where LROM cannot compute values for the secondary variables. LROM calculates these values only at LDEIM measurement points. Further efforts are

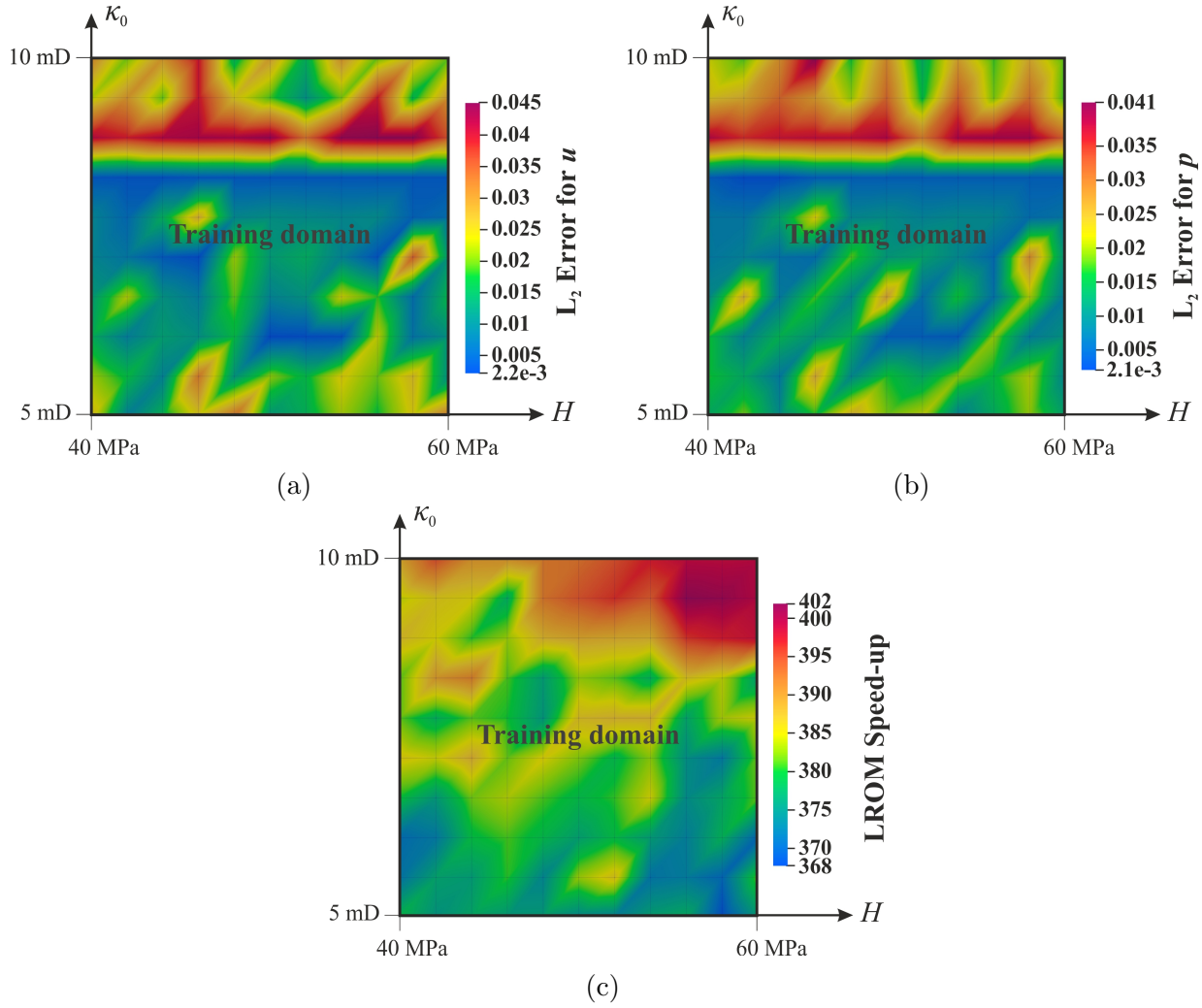
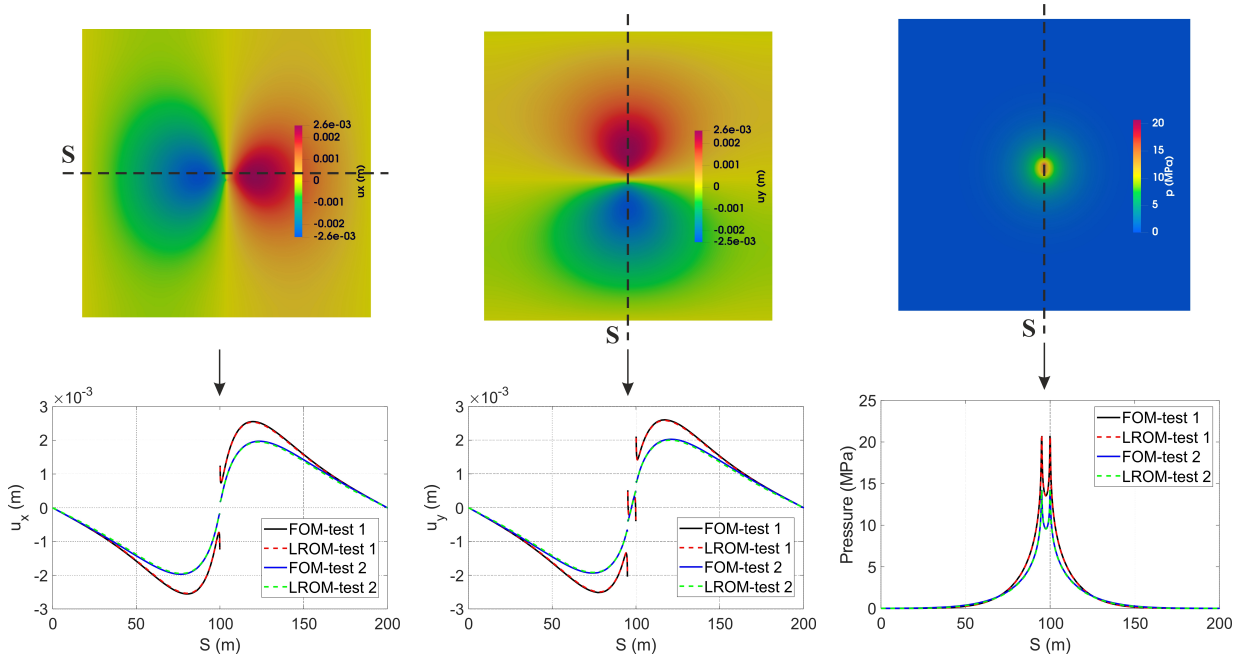



Figure 4.6: Accuracy and efficiency analyses for inputs inside of the training domain, a) relative $L_2$ error contour of LROM for displacement, b) relative $L_2$ error contour of LROM for pressure, c) LROM speed-up.

required to redistribute LROM results over the entire domain, which is beyond the scope of this work. For instance, using neural network architectures, a map function between the reduced order vector and the full order vector can be developed. By comparing Figure 4.8a with Figures 4.8b and 4.8c, it is observed that the measurement points for mechanical term $\bar{\varepsilon}^p$ differ from those for hydraulic terms $\kappa$ and $\phi$, as LROM utilizes different LDEIM basis vectors for mechanical and hydraulic nonlinear terms. Figure 4.8 shows that LDEIM

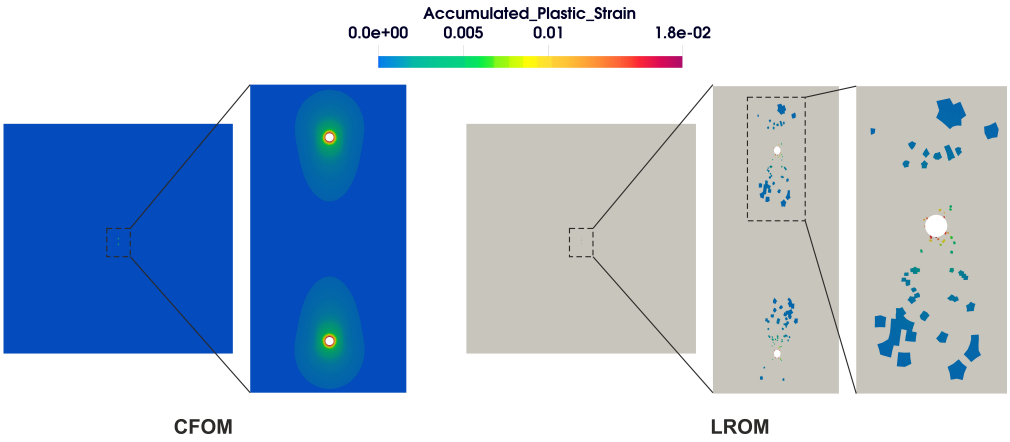



(a) (b) (c)

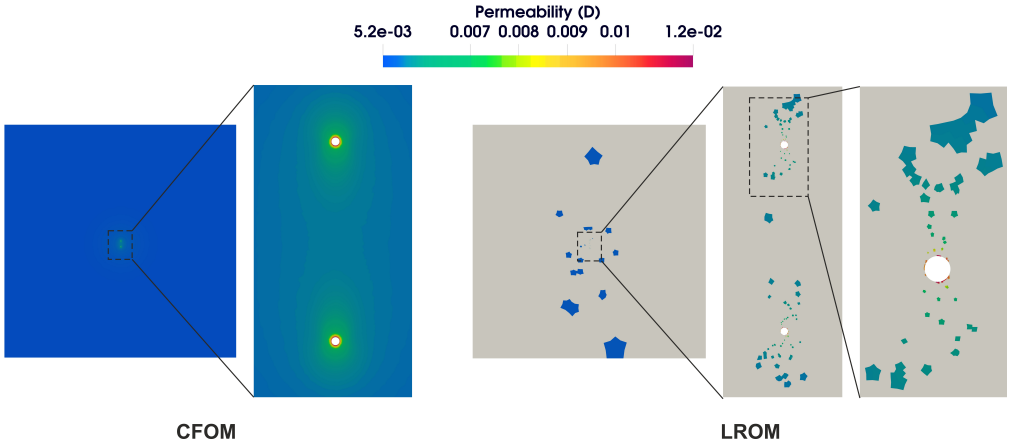
Figure 4.7: Comparison of primary variables $u_x$, $u_y$, and $p$ for CFOM and LROM, a) displacement in $x$ direction $u_x$, b) displacement in $y$ direction $u_y$, c) pore pressure $p$.

measurement points for $\bar{\varepsilon}^p$ are situated along the dilation zone, while those for $\kappa$ and $\phi$ are located in areas with enhanced permeability and porosity.

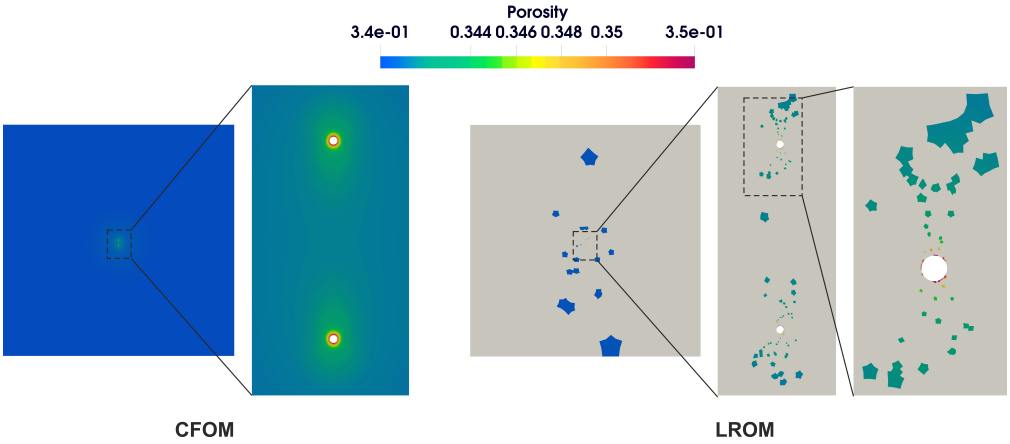
Table 4.2 summarizes the CPU performance in hours for both offline and online phases of LROM and CFOM. The offline phase involves multiple CFOM runs for the generation of snapshots and the training process using POD/LPOD/LDEIM. During the online phase, LROM is tested with two different sets of parametric inputs. The CPU runtime for the offline phase is significantly larger than the online phase. In addition, LROM provides a substantial computational speed-up (with reference to CFOM), with factors of 377 and 397 for test cases 1 and 2, respectively. These results highlight the efficiency of LROM for poro-elasto-plastic problems.




(a)




(b)




(c)

Figure 4.8: Comparison of secondary variables $\bar{\varepsilon}^p$, $\kappa$, and $\phi$ for CFOM and LROM, a) $\bar{\varepsilon}^p$, b) $\kappa$, c) $\phi$.

Table 4.2: CPU runtime in hours for offline and online stages of LROM

| Test case | Offline phase (hour) | | Online phase (hour) | |
|---|---|---|---|---|
| | Data generation | Data training | LROM | CFOM |
| 1 | 92.7 | 32 | 0.24 | 89.4 |
| 2 | | | 0.21 | 82.8 |

### 4.3.2 Example 2: Parameterized ROM with material inputs and injection rate

We now consider a possible applied scenario where the injection rate in LROM acts as an input. In practical applications, the injection rate often fluctuates due to operational requirements and various time-varying injection rates through the online phase. In this example, we aim to generate an LROM that remains efficient for pumping schedules that extend beyond the training domain. This model provides designers with a rapid evaluation of the reservoir instead of using computationally expensive coupled full order models. In addition to two material inputs, the injection rate is also treated as an input parameter for LROM. Thus, we have a parameterized LROM with three inputs: hardening modulus $H$, initial permeability $\kappa_0$, and injection rate $Q_{in}$. The training domain for hardening modulus and initial permeability is defined within the ranges of $H \in [40, 60]$ MPa and $\kappa_0 \in [5, 10]$ mD, respectively. At the same time, the offline training process is done using constant injection rates in the range of $Q_{in} \in [0, 22]$ L/s. Figure 4.9 illustrates the training regime and test schedules. In total, LROM is trained using 40 samples (10 samples in two-dimensional material space $\times$ 4 samples in injection rate space). First, the reservoir is analyzed with a training schedule for 10 days for each sample. Then, snapshots for displacement, pressure, and nonlinear force vectors are collected every 36 minutes. Next, LROM is trained using data obtained from 40 different sets of samples. Finally, online, LROM is evaluated with different time-varying injection rates and material properties, as depicted in Figure 4.9.

Global POD is applied to the displacement equation, and the size of the system of equations for the displacement part is reduced to 90. For the pressure part of the system, LPOD is applied to reduce its size to 10 for each cluster. The snapshot matrix of the pressure variable is partitioned into 8 clusters. Consequently, the total size of the system of equations in LROM is 100.

LDEIM is applied to both the mechanical and hydraulic parts of the converged nonlinear terms by dividing them into 8 clusters using $k$-means clustering method. Clustering is performed based on the displacement part of the snapshot, and the same clusters are

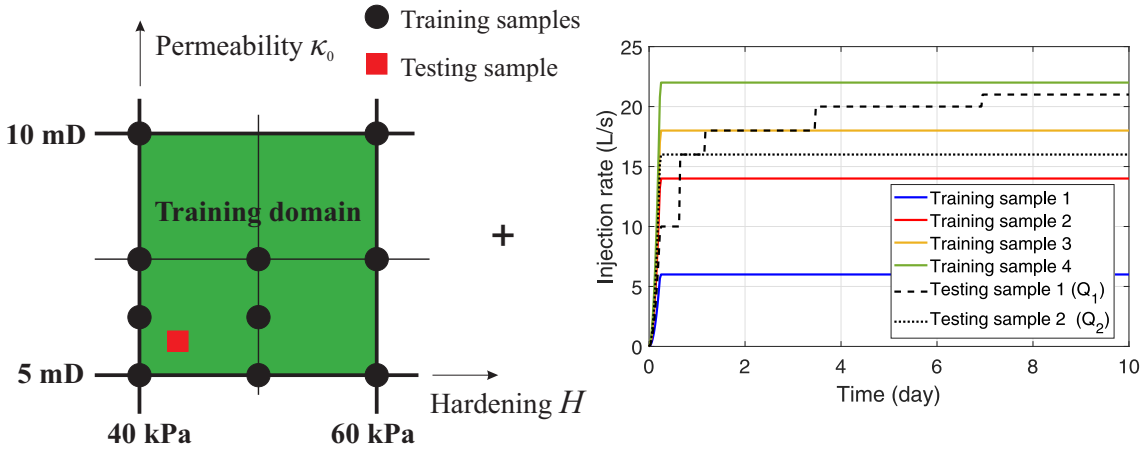



Figure 4.9: Three-dimensional training domain, including two-dimensional material parameter space and one-dimensional injection rate space.

subsequently used for the pressure part. In each cluster, the number of measurement points is set to 30.

To ensure the robustness of the presented LROM, it is necessary to test it with inputs that differ significantly from the training regime. Two online test samples are used, as depicted in Figure 4.9. Test sample 1 shows a pumping schedule acting as a rectangular wave with irregular time sequences: $Q_{in} = Q_1$. For test sample 2, a constant injection rate with a value different from the training schedules is used: $Q_{in} = Q_2$. In both test samples, the material parameter space employs inputs within the training domain: $H = 42$ MPa, $\kappa_0 = 5.2$ mD. This configuration is chosen because it represents one of the scenarios where the error between LROM and CFOM is the largest.

Figure 4.10 presents the pore pressure field $p$ and the relative error between LROM and CFOM. The top row of Figure 4.10 depicts the contour map of pore pressure snapshots for three given times during the pumping schedule: $t = 2.5$, $t = 6$, and $t = 10$ days. It should be noted that an initial pore pressure of 4 MPa must be added to the contour values to represent the total pore pressure. The material properties and injection rate are set to test sample 1. Initially, the fluid flows from both horizontal wells into the porous media. Over time, the fluid streams from both wells begin to converge. Subsequently, the fluid diffusion front propagates far beyond the stimulated zone. The maximum pore pressure, approximately 26 MPa, is located at the boundaries of the wellbore, while the minimum pore pressure remains equal to the initial pore pressure at the reservoir boundaries, far from the wells. The bottom row of Figure 4.10 illustrates the relative error of LROM for the inputs corresponding to test sample 1. LROM relative error at each DoF is calculated by

$$Err_I(t) = \frac{\left|p_I^{\mathrm{LROM}}(t) - p_I^{\mathrm{CFOM}}(t)\right|}{\left|p_I^{\mathrm{CFOM}}(t)\right|} \tag{4.46}$$

The error contour map indicates that LROM is sufficiently accurate in both temporal and spatial domains. The maximum relative error is $8.3 \times 10^{-3}$, occurring near the wellbore. In addition, the error is symmetrically distributed in the reservoir.

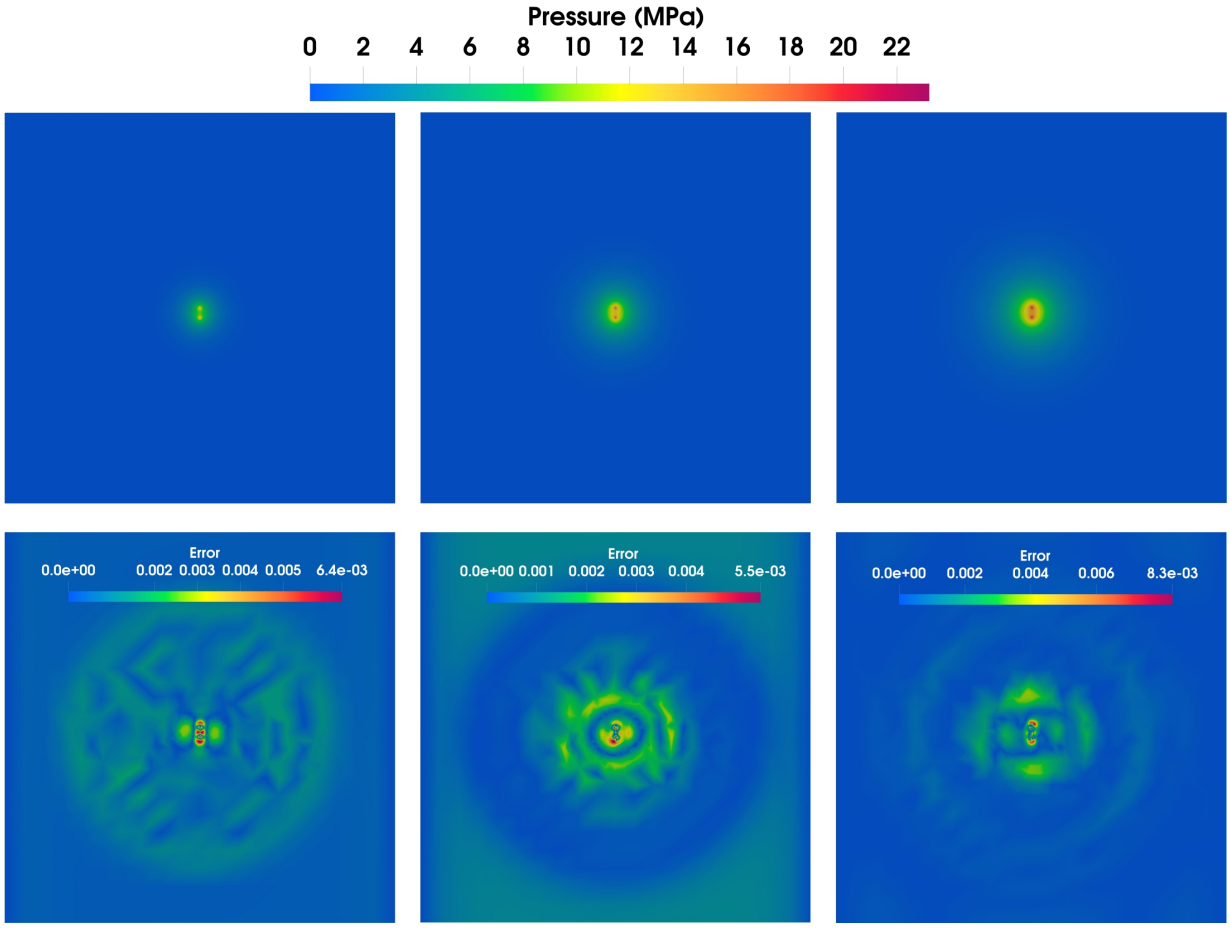



Figure 4.10: Top row: contours of the pressure field $p$ at $t = 2.5$ days, $t = 6$ days, and $t = 10$ days, from left to right. Bottom Row: contour of LROM relative error at $t = 2.5$ days, $t = 6$ days, and $t = 10$ days, from left to right.

The displacement in the $x$ and $y$ directions ($u_x$ and $u_y$), along with the relative error between LROM and CFOM, are presented in Figure 4.11 at 2.5, 6, and 10 days. Since the

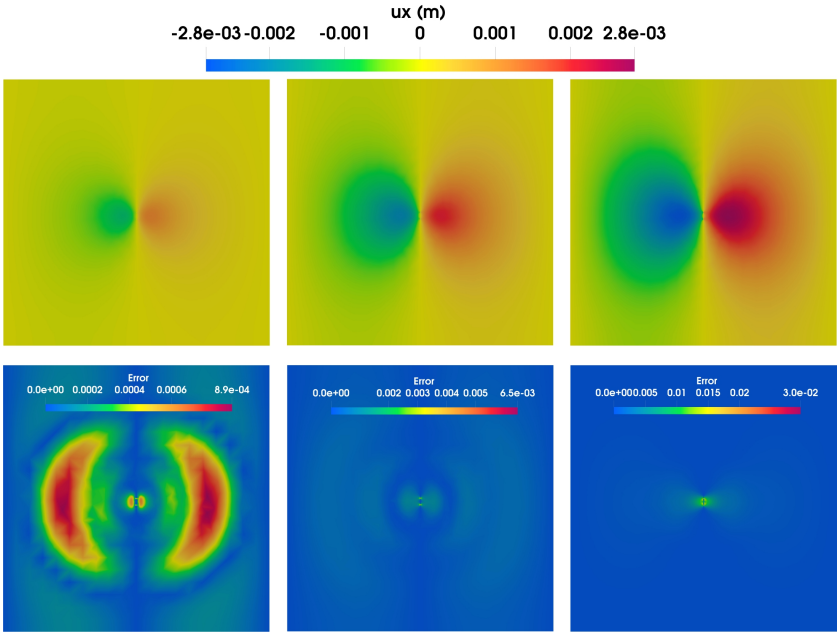
displacement near the reservoir boundaries is zero, the error is calculated using a scaled difference between CFOM and LROM results at each DOF as

$$D_{x,I}(t) = \frac{\left|u_{x,I}^{\mathrm{LROM}}(t) - u_{x,I}^{\mathrm{CFOM}}(t)\right|}{\max(|\mathbf{u}_x^{\mathrm{CFOM}}(t)|)} \tag{4.47}$$

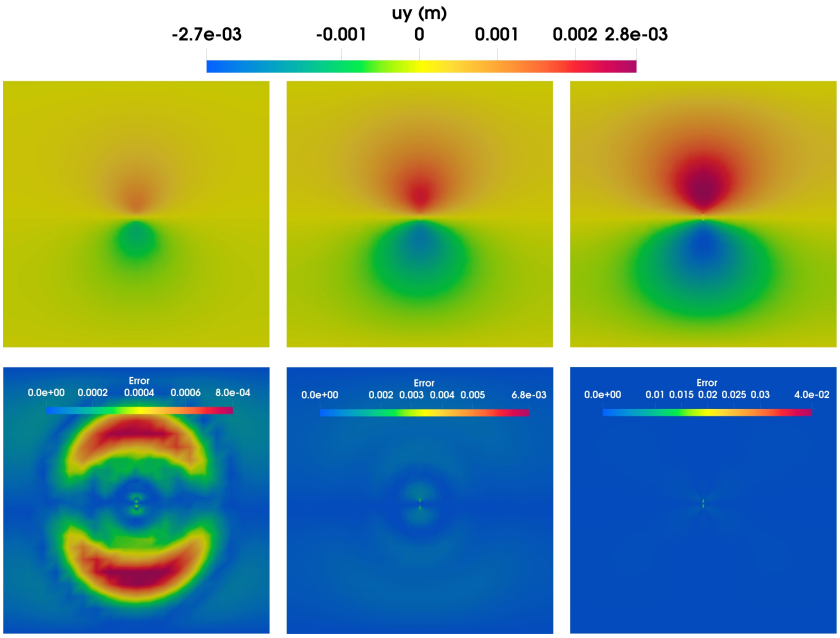
$$D_{y,I}(t) = \frac{\left|u_{y,I}^{\mathrm{LROM}}(t) - u_{y,I}^{\mathrm{CFOM}}(t)\right|}{\max(\left|\mathbf{u}_y^{\mathrm{CFOM}}(t)\right|)} \tag{4.48}$$

in which $D_{x,I}$ is the error for $u_x$, $D_{y,I}$ is the error for $u_y$, and the subscript $I$ denotes the $I$th DoF. As depicted in Figure 4.11, the maximal scaled difference increases over time but is constrained to $3 \times 10^{-2}$ for $u_x$ and $4 \times 10^{-2}$ for $u_y$. This indicates that the LROM maintains acceptable accuracy for the displacement field.

Figure 4.12 compares the time history of accumulated plastic strain $\bar{\varepsilon}^p$ for CFOM and LROM at point A located at the surface of the bottom well with coordinate $x = 100$ m and $y = 95.1$ m. As observed, there is a good agreement between CFOM and LROM results for both test samples. For test sample 2, LROM provides an almost perfect approximation of the high-fidelity model, in contrast to test sample 1. The offline training process is based on constant injection rates, and test sample 2 also uses a constant injection rate, which is inside the training regime. Thus, it is expected that LROM will converge to nearly exact results. However, test sample 1 pushes the limits of the constructed LROM by using online injection rates that are significantly different from the simple training data. The results confirm that LROM maintains acceptable accuracy even when the injection rate is dissimilar from those in the training regime.




(a)




(b)

Figure 4.11: Displacement field snapshots in the $x$ and $y$ directions ($u_x$ and $u_y$) at $t = 2.5$ days, $t = 6$ days, and $t = 10$ days, from left to right, a) $u_x$, b) $u_y$. In each subfigure, the top and bottom rows show CFOM results and LROM relative error, respectively.

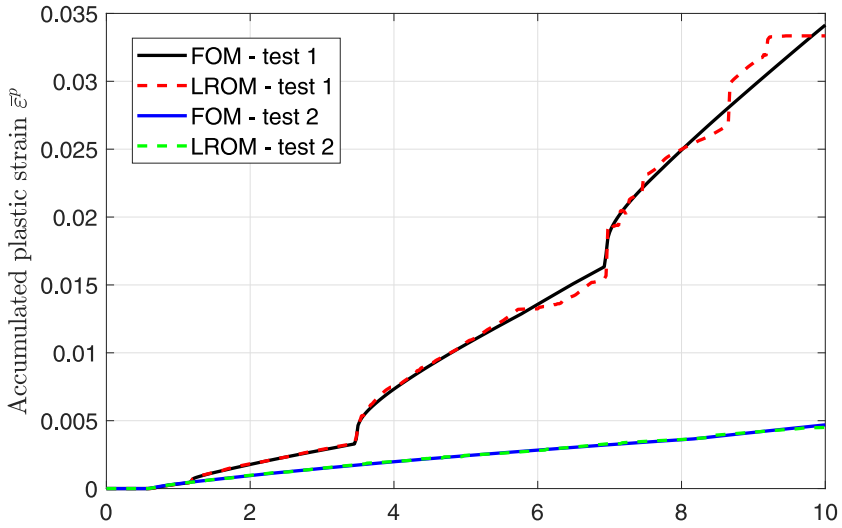



Figure 4.12: The evolution of accumulated plastic strain for CFOM and LROM at point A located at the surface of the bottom well with coordinate $x = 100$ m and $y = 95.1$ m.

Figure 4.13 shows the evolution of porosity and permeability due to shear dilation at point A. According to (4.10) and (4.11), there is an incremental relation between total strain and porosity, and subsequently between porosity and permeability. Our LROM accurately approximates the porosity and permeability enhancements for both test samples 1 and 2. In the more critical case of test sample 1, the porosity increases from 0.34 to 0.367, leading to a permeability enhancement from 5.2 mD to 24.5 mD.

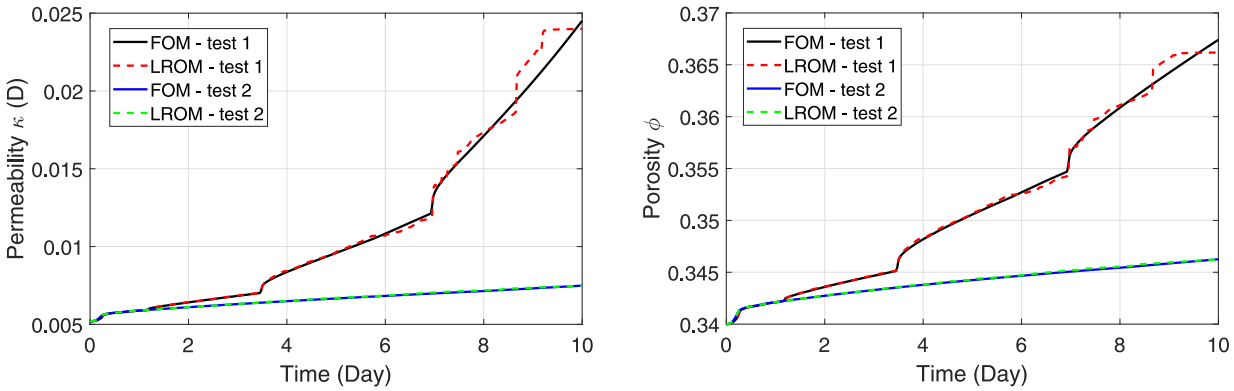



Figure 4.13: Porosity and permeability enhancement for CFOM and LROM at point A, a) Permeability, b) porosity.

## 4.4 Conclusions

In this chapter, a robust ROM formulation is introduced for poroplasticity problems, with specific application to hydraulic dilation stimulation of a reservoir with a horizontal well pair. A coupled projection matrix is used to reduce the size of the coupled system of equations. DEIM is adapted to efficiently approximate the coupled forms of the nonlinear terms (residual force vector and its Jacobian). Rather than applying DEIM to the fully-coupled nonlinear terms, it is separately applied to the displacement and pressure contributions of the nonlinear terms. In this way, different measurement points are defined for the displacement and pressure variables.

For further reductions in the approximation of nonlinear terms, a new local ROM (LROM) is proposed for clustering and classification techniques that align with the coupled formulation framework. A robust LROM is achieved by introducing distinct local DEIM subspaces for the displacement and pressure equations. Additionally, local POD is applied to the pressure equation to reduce the size of the ROM even further than global POD. It is observed that localizing the displacement equation negatively impacts the stability of the Newton-Raphson algorithm, while localizing the pressure equation enhances the stability. One possible solution to this limitation is to use overlapping clusters instead of non-overlapping clusters, which will be considered for future improvements.

Two examples demonstrate the advantages of LROM. First, LROM is parameterized for material properties (hardening modulus and initial permeability). The performance of LROM is evaluated for inputs inside the training domain. LROM accurately captures the reservoir responses (displacement and pressure) using a small ROM size ($k_u = 90$ and $k_p = 10$) and a small number of measurement points for each cluster ($q_u = 30$ and $q_p = 30$). The accuracy ranges from 0.2% to 4.5% for the displacement variable and from 0.2% to 4.1% for the pressure variable across different material inputs. Furthermore, a comparison between CFOM and LROM is accomplished for the spatial-temporal evolution of the stimulated dilation zone. The second example extends the study to a more realistic scenario by adding the injection rate as a third input for LROM. Results show that LROM maintains efficiency even for injection rates outside the training regime.

In large-scale subsurface flow modeling of hydraulic dilation stimulation, a substantial computational saving of 400 times is achieved for a single-query real-time simulation with a ROM reducing the model dimension from $10^5$ DoFs to $10^2$ DoFs. The saving is highlighted in multi-query tasks, where the model must be executed for multiple inputs and system configurations. This ROM has a high potential for accelerating various problems, such as uncertainty quantification, design, history matching, and well control optimization.

# Chapter 5

# Conclusions

## 5.1 Concluding Remarks

In this thesis, a gap in existing intrusive ROMs to capture hydro-mechanical coupling behavior and path-dependent plastic deformation of rocks was identified, and a new ROM was developed. Complex geomechanics problems were used to show the efficiency and accuracy of the ROM in addressing coupling, plasticity, and permeability enhancement features. The ROM was parameterized by material properties and loading conditions, providing reservoir engineers with a powerful tool to reduce the computational cost of complex and large-scale simulations. The cost of offline training, the accuracy, and the domain of applicability and extrapolation were discussed through the development of the ROMs.

The first objective was to develop an intrusive ROM for a nonlinear diffusion problem by combining POD-Galerkin and DEIM. POD-Galerkin was used to reduce the size of the system of equations. Classical DEIM was applied to accelerate the computation of the residual force vector. A robust version of the DEIM approach, which is based on the approximated column-reduced Jacobian, was adopted to approximate the Jacobian matrix. A significant reduction in the computational time of the online stage was achieved using the combination of these methods. The problem under analysis was modeled by a transient diffusion equation with a nonlinear permeability coefficient. The snapshots for the FOM were generated using a continuous Galerkin FE solver and a backward Euler scheme for time integration.

To demonstrate the efficiency and accuracy of the ROM, a two-dimensional layered reservoir with a two-well system was analyzed. An analysis of the trade-off between accuracy and

speed-up was conducted for ROMs of different sizes and hyper-parameters. The results showed that as the ROM becomes faster, the accuracy decreases. Under FOMs with $3 \times 10^4$ DoFs, it was observed that the ROM has about $10^3$ times fewer DoFs and is approximately 190 times faster than the FOM, while maintaining an accurate solution with reference to the FOM.

In terms of the domain of applicability and extrapolation, one important challenge in applying ROM technologies to reservoir simulations is the uncertainty in the exact pumping schedule (injection rate vs. time). The exact pumping schedule is not known in advance of the operation, as technical and logistical challenges impact schedules. Pumping schedules are often characterized by active and inactive injection periods, which can differ greatly from those used for training the ROMs. Previously, the applications of ROMs to complex and time-discontinuous pumping schedules have not been adequately accessed. The ROM presented addresses this issue by accurately reproducing the response of the reservoir, subject to such sudden changes in injection rate. Offline, training data was generated by FOM runs with simple constant injection rates. Online, the ROMs demonstrated high accuracy and efficiency for complex and time-varying pumping schedules, such as sinusoidal, high-frequency, and time-discontinuous pumping schedules.

Intrusive ROMs in reservoir engineering have been extended to some applications, including two-phase flow and transport systems. However, they come with a number of limitations. These ROMs neglect the effect of rock deformation in the reservoir and focus solely on a system of pressure-saturation equations. They fail to account for plastic deformation as a highly nonlinear behavior in rock mechanics. The second objective of this thesis was to address this gap by developing an intrusive ROM for a Drucker–Prager plasticity model, which is capable of capturing the history-dependent nonlinear evolution of plastic strain. POD-DEIM ROM presented in objective 1 was extended to POD-LDEIM by constructing multiple local DEIM (LDEIM) approximations in combination with $k$-means clustering algorithm and nearest neighbor technique. It was shown that the standard DEIM approximation of the FOM residual vector and Jacobian is inefficient for constructing accurate ROMs. Further, the online computational cost of LDEIM is substantially improved as each local DEIM requires fewer measurement points than the classical DEIM.

The ROM was parameterized by three hyper-parameters (the size of ROM, the number of clusters, and the number of DEIM measurement points per cluster), which influence both the accuracy and computational efficiency of the ROM. The user must specify these parameters, as they are not determined directly in the training algorithm. The performance of the ROM was compared for inputs within and outside of the training domain and it was observed that the accuracy is highest for the input parameters within the training domain with minimum and maximum errors of 1.0% and 3.5%, respectively. However, for inputs

outside of the training domain, the error increases to 9.2%.

To show the ability of the ROM to capture the path-dependent behavior, a cyclic plasticity model was considered, and the ROM was parameterized by the load path. Offline, training data was generated using a single load path with three loading and unloading cycles. Online, the ROM was tested with arbitrary load paths and successfully reproduced the response of the system for loading-unloading cycles that are greatly different from those used in the training process.

The presented intrusive ROM requires only a small amount of data as the underlying physics of the FOM is explicitly incorporated into the ROM. This substantially reduces the offline computational cost, which was approximately 24 hours. The offline computational cost includes the time for data generation, determining the POD and DEIM basis vectors, and clustering the data. For this specific problem, due to the complexity of plastic distribution in the domain, we expect that the computational cost of the offline phase for non-intrusive ROM techniques will be significantly higher than the cost for intrusive ROMs. This investigation will be of interest in future studies.

Real-time simulation of hydraulic dilation stimulation is computationally expensive as it must span multiple spatial and temporal length scales, including plasticity, permeability enhancement, and hydro-mechanical processing. Such a large-scale simulation typically requires many hours or even days to complete. The final objective was to couple the ROM for a diffusion problem with the ROM for plasticity to develop a novel local ROM formulation for nonlinear and fully-coupled hydro-mechanical simulations in large-scale reservoirs. The coupled FOM employed the FEA of a two-dimensional deformable porous media with single-phase fluid flow. Drucker–Prager plasticity model was used to describe the plastic behavior of sandstone. At the same time, the effect of stimulated dilation on porosity and permeability was investigated using stress-induced permeability enhancement models. The POD-Galerkin, DEIM, and their local versions (LPOD/LDEIM) were combined to reduce the computational costs. A new strategy for clustering and classification techniques tailored for the coupled formulation framework was introduced to calculate LPOD and LDEIM basis vectors. LDEIM is applied to nonlinear force vectors in momentum and mass balance equations, while LPOD is applied to the pressure state variable. It was observed that in elastoplastic hydromechanical problems with permeability enhancement, where the nonlinearity is due to mechanical and hydraulic parts of the coupled system of equations, localizing the displacement variable negatively improves the stability of the Newton–Raphson algorithm, while localizing the pressure enhances the stability. The ROM was parameterized not only by material properties but also by the injection rate for hydraulic dilation stimulation of a reservoir with a horizontal well pair. Under FOMs with $10^5$ DoFs, a speed-up of 400 times was achieved using a single CPU. This performance can

be significantly improved by the online implementation of multiple CPUs or GPUs.

Additional model parameters, such as friction angle and Young's modulus, can be considered to expand the parameterization. However, including more parameters will require additional training data and an increased number of samples. This will lead to a higher number of POD modes and DEIM basis vectors during the training process. Consequently, the size of the ROM will increase, reducing its speed-up. Furthermore, it is expected that the convergence of the Newton–Raphson method will be slower than the current scenario, which considers only a two-dimensional material parameter space.

This version of the ROM is expected to provide reservoir engineers with a powerful tool for multi-query tasks, such as history matching. In these tasks, an optimization process must be conducted, where hundreds to thousands of simulations are required to minimize the difference between field data and simulator prediction values to obtain an optimal target. The replacement of computationally expensive FE models with the presented ROM significantly accelerates the optimization process. For multi-query tasks, it is expected to achieve a speed-up significantly greater than single-query tasks (speed-up $\gg 400$), as the ROM is executed multiple times.

## 5.2 Research Contributions

In the process of accomplishing the objectives of this thesis, research contributions that expand the literature on intrusive geomechanics ROMs in reservoir applications were produced. The significant contributions produced by this thesis were disseminated in conferences and peer-reviewed journals and are summarized below.

The first objective was to develop an intrusive ROM for nonlinear porous media flow problems, with specific application to a two-dimensional layered reservoir. This work is disseminated in:

S. Hatefi Ardakani, G. Zingaro, M. Komijani, and R. Gracie, "An efficient reduced order model for nonlinear transient porous media flow with time-varying injection rates," *Finite Elements in Analysis and Design*, vol. 241, p. 104237, 2024.

The key contributions of this work are:

- An efficient and accurate parameterized ROM is derived for a diffusion problem by combining POD-Galerkin with classical and column-reduced Jacobian DEIM techniques.

- An analysis of the trade-off between accuracy and run-time is conducted for ROMs of different hyper-parameters (POD modes and DEIM sample points).
- Online, the ROM demonstrates significantly lower computation times and accurately models subsurface subject to complex time-varying injection rates.

The second objective of this thesis was to develop a parameterized ROM for a cyclic plasticity model to address challenges in capturing the history-dependent nonlinear evolution of plastic strain. This work is disseminated in:

S. Hatefi Ardakani, G. Zingaro, and R. Gracie, "An efficient parameterized local reduced order model for Drucker–Prager plasticity," *Submitted to Computational Mechanics*, September 2024.

The key contributions of this work are:

- The first ROM to solve a cyclic Drucker–Prager plasticity problem is developed. The ROM captures the history-dependent nonlinear evolution of plastic strain around the well.
- Numerical convergence issues observed in plasticity ROM with classical DEIM are addressed by introducing local DEIM approximations.
- Influence of hyper-parameters (ROM size, number of clusters, and DEIM measurement points) on accuracy and speed-up are investigated.

The third objective of this thesis was to develop a ROM formulation for the simulation of stimulated volume evolution in large-scale reservoirs. This work is disseminated in:

S. Hatefi Ardakani and R. Gracie, "A parameterized local reduced order model of stimulated volume evolution in reservoirs," *Submitted to International Journal for Numerical and Analytical Methods in Geomechanics*, November 2024.

S. Hatefi Ardakani and R. Gracie, "A hyper-reduced order model for nonlinear coupled porous media flow," *16th World Congress on Computational Mechanics and 4th Pan American Congress on Computational Mechanics*. Vancouver, Canada, July 2024.

The key contributions of this work are:

- The first intrusive hydro-mechanical ROM for poroplasticity problems is developed.

- ROM is applied on a coupled system of momentum and mass balance equations equipped with Drucker–Prager plasticity and stress-induced permeability enhancement models.
- A robust surrogate model is presented to simulate well communication in a large-scale reservoir.
- A new strategy for clustering and classification techniques tailored for the coupled formulation framework is introduced.

## 5.3 Directions for Future Study

There are many potential topics for future study on ROMs in reservoir engineering. Some of these ideas include:

**Level of Intrusion:** The ROM presented in this thesis is extremely intrusive, requiring the modification of the FE source code of the FOM. There are other ROMs with different levels of intrusion, such as hybrid ROMs and non-intrusive ROMs. One of the largest remaining challenges is to evaluate non-intrusive ROMs in terms of accuracy, offline and online computational costs, and generalizability beyond the training domain. Deep learning-based ROMs and probabilistic-based regression methods, such as Gaussian process regression (GPR), can be classified as non-intrusive ROMs. A primary challenge for deep learning methods is the data requirements. They require a large amount of training data to learn effectively. In geomechanical simulations, generating such data is computationally expensive. Another challenge is associated with interpretability. Deep learning methods can be complex to interpret since they are often seen as black boxes. The other challenge is generalization and overfitting. Deep learning methods require careful design to avoid issues such as overfitting or oscillations in output. GPRs can address many of these challenges. They perform well with small datasets because of their ability to interpolate between known data points. GPRs are often more interpretable because of their probabilistic framework. In addition, GPRs can handle continuous inputs and outputs, providing smooth predictions based on a probabilistic framework. Applying these ROMs to the same large-scale problems solved in this thesis is recommended to compare their performance with the intrusive ROM.

**Use of Other Hyper-reduction Techniques:** Hyper-reduction methods based on direct sampling, have demonstrated better stability than interpolation-based methods like DEIM. They preserve the symmetry and the positivity of the tangent stiffness matrices. In this thesis, a non-positive and non-symmetric approximated tangent stiffness matrix was observed using the interpolation based DEIM method, particularly in plasticity, causing

instabilities in the Newton–Raphson algorithm. While local DEIM helps to improve this drawback, the need for direct sampling methods, such as energy conserving mesh sampling and weighting (ECSW), to approximate the tangent stiffness matrix is remarked.

**Redistribution of the ROM Results:** The ROM developed for plasticity and poroplasticity in this thesis computes the primary variables, such as displacement and pressure fields, in the entire domain using POD projection matrices. However, the secondary variables, such as plastic strain and permeability, are calculated only at sampling points and not throughout the domain. Further efforts are required to redistribute the ROM results for the secondary variables across the entire domain. Regression-based models, such as neural network architectures and GPRs, can be developed to project the reduced order vector back into the full order vector. Since the problems in this thesis are history-dependent and the amount of training data is limited, choosing a proper method remains a big challenge.

**Model Extensions:** There are various physical behaviors that the presented ROM cannot currently capture but are of interest for industrial applications. These include multiphase fluid flow, thermal and chemical coupling effects. These features make the ROM suitable for a wide range of geomechanics applications in enhanced geothermal systems (EGS), thermal enhanced oil recovery (EOR), steam assisted gravity drainage (SAGD), and so forth.

**Extension to Three-dimensional ROMs:** A three-dimensional model provides a more realistic representation of porous media reservoirs, including complex geometries, heterogeneities, and multi-dimensional flow. The speed-up factor for three-dimensional ROMs is expected to be significantly larger than for two-dimensional ROMs because three-dimensional FOMs are much more computationally expensive compared to two-dimensional FOMs. In three-dimensional FOMs, elements have more nodes and DoFs compared to two-dimensional ones, and numerical integration requires more integration points. This increases the computational cost for evaluating large matrices, such as stiffness and permeability matrices. The reduction in the number of DoFs in the ROM, from a large value in the three-dimensional FOM to a much smaller value in the reduced model, results in a substantial speed-up, making three-dimensional intrusive ROMs much more efficient than their FOMs. Additionally, solving the linearized system of equations in a three-dimensional form involves larger matrices, making the inversion of the Jacobian matrix computationally expensive. In three-dimensional ROMs, the large Jacobian matrix in the FOM is replaced by a much smaller matrix, drastically reducing the computational cost of matrix inversion. An extension of the ROM to a three-dimensional form would be challenging but valuable contribution in terms of the efficiency.

**Use of other constitutive plasticity models:** Adopting a different constitutive plasticity model, such as Cap plasticity model, increases the complexity of both the FOM and the ROM. The yield surface in Drucker–Prager model is a simple cone, indicating a linear

relation between shear stress and pressure. In contrast, Cap plasticity model introduces a cap to the yield surface, requiring additional material parameters to define the cap geometry and its evolution. The added complexity is expected to result in challenging convergence behavior during Newton–Raphson algorithm. The algorithm requires more iterations per time step, resulting in higher computational costs for the FOM and the ROM. In the offline phase, the complexity impacts both data generation and training process. The computational cost of data generation increases due to the need for additional iterations per time step and the need for more sampling points to accurately capture the behavior of Cap plasticity model. Furthermore, the complexity of stress distributions in the domain makes the training process more demanding. In the online phase, the ROM is expected to require more POD modes and DEIM basis vectors to accurately represent the complexity of the system, resulting in a larger ROM. Implementing such a constitutive model is recommended to compare its offline and online performance with that of Drucker–Prager plasticity model.

**Multi-query Tasks in Reservoir Engineering:** In this thesis, a real-time simulation was conducted to evaluate the efficiency and the accuracy of the ROM. The ROM demonstrated substantial computational saving for single-query simulations. The saving is highlighted in multi-query tasks, such as history matching, where an optimization process is conducted to calibrate the reservoir models by minimizing the difference between field data and simulator predictions. Using the ROM in place of computationally expensive FE models would make the optimization algorithm more robust, as depicted in Figure 5.1.

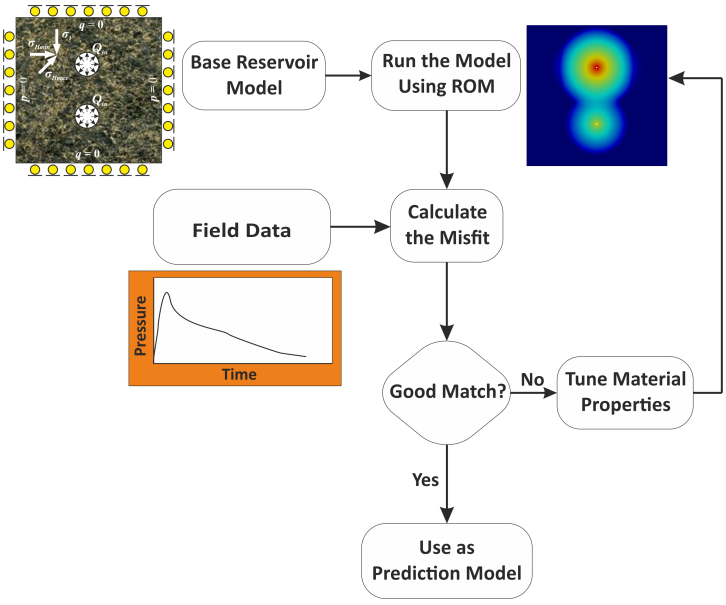



Figure 5.1: Algorithm of history matching process using ROM.

# References


[1] Y. Yuan, "Hydraulic dilation stimulation to improve steam injectivity and conformance in thermal heavy oil production," in *SPE International Heavy Oil Conference and Exhibition*, Society of Petroleum Engineers, 2018.

[2] X. Sun, B. Xu, G. Qian, and B. Li, "The application of geomechanical SAGD dilation startup in a Xinjiang oil field heavy-oil reservoir," *Journal of Petroleum Science and Engineering*, vol. 196, p. 107670, 2021.

[3] S. Yin, B. F. Towler, M. B. Dusseault, and L. Rothenburg, "Numerical experiments on oil sands shear dilation and permeability enhancement in a multiphase thermoporoelastoplasticity framework," *Journal of Petroleum Science and Engineering*, vol. 69, no. 3-4, pp. 219–226, 2009.

[4] B. Xu and R. C. Wong, "Coupled finite-element simulation of injection well testing in unconsolidated oil sands reservoir," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 37, no. 18, pp. 3131–3149, 2013.

[5] Z. Yang, X. Sun, C. Luo, B. Xu, B. Yang, and B. Li, "Vertical-well-assisted SAGD dilation process in heterogeneous super-heavy oil reservoirs: Numerical simulations," *Underground Space*, vol. 6, no. 6, pp. 603–618, 2021.

[6] T.-P. Fries and T. Belytschko, "The extended/generalized finite element method: an overview of the method and its applications," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 3, pp. 253–304, 2010.

[7] T. Belytschko, R. Gracie, and G. Ventura, "A review of extended/generalized finite element methods for material modeling," *Modelling and Simulation in Materials Science and Engineering*, vol. 17, no. 4, p. 043001, 2009.

[8] T. Rabczuk and T. Belytschko, "Cracking particles: a simplified meshfree method for arbitrary evolving cracks," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 13, pp. 2316–2343, 2004.

[9] V. P. Nguyen, C. Anitescu, S. P. Bordas, and T. Rabczuk, "Isogeometric analysis: an overview and computer implementation aspects," *Mathematics and Computers in Simulation*, vol. 117, pp. 89–116, 2015.

[10] Y. Li, T. Yu, C. Xing, and S. Natarajan, "Modeling quasi-static and dynamic thermo-elastic coupled brittle fracture using an adaptive isogeometric hybrid phase-field method," *Finite Elements in Analysis and Design*, vol. 224, p. 103993, 2023.

[11] R. Abedi, O. Omidi, and S. Enayatpour, "A mesh adaptive method for dynamic well stimulation," *Computers and Geotechnics*, vol. 102, pp. 12–27, 2018.

[12] P. Gupta and C. A. Duarte, "Simulation of non-planar three-dimensional hydraulic fracture propagation," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 38, no. 13, pp. 1397–1430, 2014.

[13] L. Li, C. Tang, G. Li, S. Wang, Z. Liang, and Y. Zhang, "Numerical simulation of 3D hydraulic fracturing based on an improved flow-stress-damage model and a parallel FEM technique," *Rock Mechanics and Rock Engineering*, vol. 45, no. 5, pp. 801–818, 2012.

[14] K. Wang, H. Liu, J. Luo, and Z. Chen, "Efficient CPR-type preconditioner and its adaptive strategies for large-scale parallel reservoir simulations," *Journal of Computational and Applied Mathematics*, vol. 328, pp. 443–468, 2018.

[15] F. Casenave, A. Gariah, C. Rey, and F. Feyel, "A nonintrusive reduced order model for nonlinear transient thermal problems with nonparametrized variability," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, pp. 1–19, 2020.

[16] P. V. Thomas, M. S. ElSayed, and D. Walch, "Review of model order reduction methods and their applications in aeroelasticity loads analysis for design optimization of complex airframes," *Journal of Aerospace Engineering*, vol. 32, no. 2, p. 04018156, 2019.

[17] K. Lindhorst, M. Haupt, and P. Horst, "Reduced-order modelling of non-linear, transient aerodynamics of the HIRENASD wing," *The Aeronautical Journal*, vol. 120, no. 1226, pp. 601–626, 2016.

[18] B. Besselink, U. Tabak, A. Lutowska, N. Van de Wouw, H. Nijmeijer, D. J. Rixen, M. Hochstenbach, and W. Schilders, "A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control," *Journal of Sound and Vibration*, vol. 332, no. 19, pp. 4403–4422, 2013.

[19] S. Lall, J. E. Marsden, and S. Glavaški, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 12, no. 6, pp. 519–535, 2002.

[20] M. Calka, P. Perrier, J. Ohayon, C. Grivot-Boichon, M. Rochette, and Y. Payan, "Machine-Learning based model order reduction of a biomechanical model of the human tongue," *Computer Methods and Programs in Biomedicine*, vol. 198, p. 105786, 2021.

[21] C. W. Rowley, "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005.

[22] S. Hijazi, G. Stabile, A. Mola, and G. Rozza, "Data-driven POD-Galerkin reduced order model for turbulent flows," *Journal of Computational Physics*, vol. 416, p. 109513, 2020.

[23] H. Florez and E. Gildin, "Global/local model order reduction in coupled flow and linear thermal-poroelasticity," *Computational Geosciences*, pp. 1–27, 2019.

[24] M. Ghadiri, A. Marjani, R. Daneshfar, and S. Shirazian, "Model order reduction of a reservoir simulation by SOD-DEIM," *Journal of Petroleum Science and Engineering*, vol. 200, p. 108137, 2021.

[25] P. Kerfriden, J.-C. Passieux, and S. P.-A. Bordas, "Local/global model order reduction strategy for the simulation of quasi-brittle fracture," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 2, pp. 154–179, 2012.

[26] D. Binion and X. Chen, "Coupled electrothermal-mechanical analysis for MEMS via model order reduction," *Finite Elements in Analysis and Design*, vol. 46, no. 12, pp. 1068–1076, 2010.

[27] A. Gaonkar and S. Kulkarni, "Model order reduction for dynamic simulation of beams with forcing and geometric nonlinearities," *Finite Elements in Analysis and Design*, vol. 76, pp. 50–62, 2013.

[28] K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 2, pp. 155–181, 2011.

[29] K. Carlberg, M. Barone, and H. Antil, "Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction," *Journal of Computational Physics*, vol. 330, pp. 693–734, 2017.

[30] F. Ghavamian, P. Tiso, and A. Simone, "POD-DEIM model order reduction for strain-softening viscoplasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 317, pp. 458–479, 2017.

[31] M. Ebrahimi and M. Yano, "A hyperreduced reduced basis element method for reduced-order modeling of component-based nonlinear systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 431, p. 117254, 2024.

[32] E. Du and M. Yano, "Efficient hyperreduction of high-order discontinuous Galerkin methods: Element-wise and point-wise reduced quadrature formulations," *Journal of Computational Physics*, vol. 466, p. 111399, 2022.

[33] C. Gogu and J.-C. Passieux, "Efficient surrogate construction by combining response surface methodology and reduced order modeling," *Structural and Multidisciplinary Optimization*, vol. 47, no. 6, pp. 821–837, 2013.

[34] A. Ghavidel, R. Gracie, and M. Dusseault, "Transient heat transfer in a horizontal well in hot dry rock–model, solution, and response surfaces for practical inputs," *Applied Thermal Engineering*, p. 117158, 2021.

[35] M. Guo and J. S. Hesthaven, "Reduced order modeling for nonlinear structural analysis using Gaussian process regression," *Computer Methods in Applied Mechanics and Engineering*, vol. 341, pp. 807–826, 2018.

[36] J. Yu, C. Yan, and M. Guo, "Non-intrusive reduced-order modeling for fluid problems: A brief review," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 16, pp. 5896–5912, 2019.

[37] M. Zhou, M. Shadabfar, H. Huang, Y. F. Leung, and S. Uchida, "Meta-modelling of coupled thermo-hydro-mechanical behaviour of hydrate reservoir," *Computers and Geotechnics*, vol. 128, p. 103848, 2020.

[38] J. S. Hesthaven and S. Ubbiali, "Non-intrusive reduced order modeling of nonlinear problems using neural networks," *Journal of Computational Physics*, vol. 363, pp. 55–78, 2018.

[39] S. Hatefi Ardakani, G. Zingaro, M. Komijani, and R. Gracie, "An efficient reduced order model for nonlinear transient porous media flow with time-varying injection rates," *Finite Elements in Analysis and Design*, vol. 241, p. 104237, 2024.

[40] S. Hatefi Ardakani, G. Zingaro, and R. Gracie, "An efficient parameterized local reduced order model for Drucker–Prager plasticity," *Submitted to Computational Mechanics*, 2024.

[41] S. Hatefi Ardakani and R. Gracie, "A parameterized local reduced order model of stimulated volume evolution in reservoirs," *Submitted to International Journal for Numerical and Analytical Methods in Geomechanics*, 2024.

[42] J. Kehls, S. Kastian, T. Brepols, and S. Reese, "Reduced order modeling of structural problems with damage and plasticity," *Proceedings in Applied Mathematics and Mechanics*, vol. 23, no. 4, p. e202300079, 2023.

[43] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[44] A. Fic, R. A. Białecki, and A. J. Kassab, "Solving transient nonlinear heat conduction problems by proper orthogonal decomposition and the finite-element method," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 48, no. 2, pp. 103–124, 2005.

[45] S. Grimberg, C. Farhat, R. Tezaur, and C. Bou-Mosleh, "Mesh sampling and weighting for the hyperreduction of nonlinear Petrov–Galerkin reduced-order models with local reduced-order bases," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 7, pp. 1846–1874, 2021.

[46] C. Farhat, T. Chapman, and P. Avery, "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1077–1110, 2015.

[47] J. A. Hernandez, M. A. Caicedo, and A. Ferrer, "Dimensional hyper-reduction of nonlinear finite element models via empirical cubature," *Computer Methods in Applied Mechanics and Engineering*, vol. 313, pp. 687–722, 2017.

[48] A. T. Patera and M. Yano, "An LP empirical quadrature procedure for parametrized functions," *Comptes Rendus Mathematique*, vol. 355, no. 11, pp. 1161–1167, 2017.

[49] P. Astrid, S. Weiland, K. Willcox, and T. Backx, "Missing point estimation in models described by proper orthogonal decomposition," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2237–2251, 2008.

[50] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation'method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, 2004.

[51] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.

[52] K. Willcox, "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Computers and Fluids*, vol. 35, no. 2, pp. 208–226, 2006.

[53] P. Tiso and D. J. Rixen, "Discrete empirical interpolation method for finite element structural dynamics," in *Topics in Nonlinear Dynamics, Volume 1*, pp. 203–212, Springer, 2013.

[54] X. Tan, E. Gildin, H. Florez, S. Trehan, Y. Yang, and N. Hoda, "Trajectory-based DEIM (TDEIM) model reduction applied to reservoir simulation," *Computational Geosciences*, vol. 23, no. 1, pp. 35–53, 2019.

[55] Z. Drmac and S. Gugercin, "A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions," *SIAM Journal on Scientific Computing*, vol. 38, no. 2, pp. A631–A648, 2016.

[56] V. Bussy, C. Vienne, and V. Kaftandjian, "Fast algorithms based on Empirical Interpolation Methods for selecting best projections in Sparse-View X-ray Computed Tomography using a priori information," *NDT and E International*, vol. 134, p. 102768, 2023.

[57] D. Bonomi, A. Manzoni, and A. Quarteroni, "A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 324, pp. 300–326, 2017.

[58] F. Negri, A. Manzoni, and D. Amsallem, "Efficient model reduction of parametrized systems by matrix discrete empirical interpolation," *Journal of Computational Physics*, vol. 303, pp. 431–454, 2015.

[59] B. Peherstorfer and K. Willcox, "Online adaptive model reduction for nonlinear systems via low-rank updates," *SIAM Journal on Scientific Computing*, vol. 37, no. 4, pp. A2123–A2150, 2015.

[60] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz, "Localized discrete empirical interpolation method," *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. A168–A192, 2014.

[61] F. Müller, A. Siokos, J. Kolb, M. Nell, and K. Hameyer, "Efficient estimation of electrical machine behavior by model order reduction," *IEEE Transactions on Magnetics*, vol. 57, no. 6, pp. 1–4, 2021.

[62] S. Yoon, Z. M. Alghareeb, and J. R. Williams, "Development of reduced-order oil reservoir models using localized DEIM," in *SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers*, p. 170741, 2014.

[63] S. U. Ansari, M. Hussain, A. Rashid, S. Mazhar, S. Ahmad, and K. J. Siddiqui, "Validating numerical solution of transient Darcy flow using the stabilized mixed finite element method," *Simulation*, vol. 94, no. 1, pp. 55–65, 2018.

[64] J. Istok, *Groundwater modeling by the finite element method*. Washington, DC (USA) American Geophysical Union, 1989.

[65] E. Wilson, K. Bathe, and F. Peterson, "Finite element analysis of linear and nonlinear heat transfer," *Nuclear Engineering and Design*, vol. 29, no. 1, pp. 110–124, 1974.

[66] A. Chatterjee, "An introduction to the proper orthogonal decomposition," *Current Science*, pp. 808–817, 2000.

[67] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu, "Proper orthogonal decomposition and its applications—part I: Theory," *Journal of Sound and vibration*, vol. 252, no. 3, pp. 527–544, 2002.

[68] S. Volkwein, "Proper orthogonal decomposition: Theory and reduced-order modelling," *Lecture Notes, University of Konstanz*, vol. 4, no. 4, pp. 1–29, 2013.

[69] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[70] X. Li, X. Zhang, and Y. Zhang, "Three-dimensional plasticity-based topology optimization with smoothed finite element analysis," *Computational Mechanics*, vol. 73, no. 3, pp. 533–548, 2024.

[71] J. Kuhn, J. Spitz, P. Sonnweber-Ribic, M. Schneider, and T. Böhlke, "Identifying material parameters in crystal plasticity by Bayesian optimization," *Optimization and Engineering*, pp. 1–35, 2021.

[72] M. B. Gorji, M. Mozaffar, J. N. Heidenreich, J. Cao, and D. Mohr, "On the potential of recurrent neural networks for modeling path dependent plasticity," *Journal of the Mechanics and Physics of Solids*, vol. 143, p. 103972, 2020.

[73] G. Chen, "Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity," *Computational Mechanics*, vol. 67, no. 3, pp. 1009–1019, 2021.

[74] M. Maia, I. B. Rocha, P. Kerfriden, and F. van der Meer, "Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate," *Computer Methods in Applied Mechanics and Engineering*, vol. 407, p. 115934, 2023.

[75] L. Borkowski, C. Sorini, and A. Chattopadhyay, "Recurrent neural network-based multiaxial plasticity model with regularization for physics-informed constraints," *Computers and Structures*, vol. 258, p. 106678, 2022.

[76] M. Eghbalian, M. Pouragha, and R. Wan, "A physics-informed deep neural network for surrogate modeling in classical elasto-plasticity," *Computers and Geotechnics*, vol. 159, p. 105472, 2023.

[77] D. C. Drucker and W. Prager, "Soil mechanics and plastic analysis or limit design," *Quarterly of Applied Mathematics*, vol. 10, no. 2, pp. 157–165, 1952.

[78] G.-F. Zhao, J. Lian, A. Russell, and N. Khalili, "Implementation of a modified Drucker–Prager model in the lattice spring model for plasticity and fracture," *Computers and Geotechnics*, vol. 107, pp. 97–109, 2019.

[79] E. A. de Souza Neto, D. Peric, and D. R. Owen, *Computational methods for plasticity: theory and applications*. Chichester: Wiley, 2008.

[80] J. C. Simo and R. L. Taylor, "Consistent tangent operators for rate-independent elastoplasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 48, no. 1, pp. 101–118, 1985.

[81] F. Dunne and N. Petrinic, *Introduction to computational plasticity*. New York: Oxford University Press, 2005.

[82] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary, *Nonlinear finite elements for continua and structures*. New York: wiley, 2014.

[83] M. Drohmann, B. Haasdonk, and M. Ohlberger, "Adaptive reduced basis methods for nonlinear convection-diffusion equations," in *Finite Volumes for Complex Applications VI Problems & Perspectives, Springer*, pp. 369–377, 2011.

[84] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, 2022.

[85] D. Amsallem, M. J. Zahr, and C. Farhat, "Nonlinear model order reduction based on local reduced-order bases," *International Journal for Numerical Methods in Engineering*, vol. 92, no. 10, pp. 891–916, 2012.

[86] D. Amsallem and B. Haasdonk, "Pebl-rom: Projection-error based local reduced-order models," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, pp. 1–25, 2016.

[87] C. Oskay and J. Fish, "Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 7, pp. 1216–1243, 2007.

[88] J. Fish, K. Shek, M. Pandheeradi, and M. S. Shephard, "Computational plasticity for composite structures based on mathematical homogenization: Theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 148, no. 1-2, pp. 53–73, 1997.

[89] P. F. Dehaghani, S. H. Ardakani, H. Bayesteh, and S. Mohammadi, "3D hierarchical multiscale analysis of heterogeneous SMA based materials," *International Journal of Solids and Structures*, vol. 118, pp. 24–40, 2017.

[90] B. Peherstorfer, Z. Drmac, and S. Gugercin, "Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points," *SIAM Journal on Scientific Computing*, vol. 42, no. 5, pp. A2837–A2864, 2020.

[91] M. Ghasemi, Y. Yang, E. Gildin, Y. Efendiev, and V. Calo, "Fast multiscale reservoir simulations using POD-DEIM model reduction," in *SPE Reservoir Simulation Symposium*, Society of Petroleum Engineers, 2015.

[92] H. Florez and E. Gildin, "Global/local model order reduction in coupled flow and linear thermal-poroelasticity," *Computational Geosciences*, vol. 24, no. 2, pp. 709–735, 2020.

[93] M. Ghasemi and E. Gildin, "Localized model order reduction in porous media flow simulation," *Journal of Petroleum Science and Engineering*, vol. 145, pp. 689–703, 2016.

[94] R. W. Lewis and B. A. Schrefler, *The finite element method in the static and dynamic deformation and consolidation of porous media*. John Wiley & Sons, 1999.

[95] M. A. Biot, "General theory of three-dimensional consolidation," *Journal of Applied Physics*, vol. 12, no. 2, pp. 155–164, 1941.

[96] J. Kim, H. A. Tchelepi, and R. Juanes, "Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 13-16, pp. 1591–1606, 2011.

[97] F. Cappa and J. Rutqvist, "Modeling of coupled deformation and permeability evolution during fault reactivation induced by deep underground injection of $CO_2$," *International Journal of Greenhouse Gas Control*, vol. 5, no. 2, pp. 336–346, 2011.

[98] L. Chin, R. Raghavan, and L. Thomas, "Fully coupled geomechanics and fluid-flow analysis of wells with stress-dependent permeability," *Spe Journal*, vol. 5, no. 01, pp. 32–45, 2000.

[99] G. Han and M. B. Dusseault, "Description of fluid flow around a wellbore with stress-dependent porosity and permeability," *Journal of Petroleum Science and Engineering*, vol. 40, no. 1-2, pp. 1–16, 2003.

[100] S. Hsiung, A. Chowdhury, and M. Nataraja, "Numerical simulation of thermal-mechanical processes observed at the Drift-Scale Heater Test at Yucca Mountain, Nevada, USA," *International Journal of Rock Mechanics and Mining Sciences*, vol. 42, no. 5-6, pp. 652–666, 2005.

[101] O. C. Zienkiewicz and R. L. Taylor, *The finite element method: Solid mechanics*, vol. 2. Butterworth-heinemann, 2000.

[102] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, "Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations," *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1136–1161, 2015.

[103] G. Rozza and K. Veroy, "On the stability of the reduced basis method for Stokes equations in parametrized domains," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 7, pp. 1244–1260, 2007.

[104] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

# APPENDICES

# Appendix A

# Drucker–Prager Plasticity Constitutive Equations

In Drucker–Prager plasticity, the equilibrium equation is equipped with the constitutive equation in the incremental form:

$$\mathrm{d}\boldsymbol{\sigma} = \mathbf{D}^e \left(\mathrm{d}\boldsymbol{\varepsilon} - \mathrm{d}\boldsymbol{\varepsilon}^p\right) \tag{A.1}$$

where $\mathrm{d}\,(\bullet)$ is the increment in variable $(\bullet)$, $\boldsymbol{\varepsilon}$ is the total strain, $\boldsymbol{\varepsilon}^p$ is the plastic strain, and $\mathbf{D}^e$ is the isotropic elasticity tensor. To determine the direction of the plastic strain increment and hardening variables, the plastic flow rules are written as

$$\mathrm{d}\boldsymbol{\varepsilon}^p = \mathrm{d}\lambda \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \tag{A.2}$$

and

$$\mathrm{d}\boldsymbol{\alpha} = -\mathrm{d}\lambda \frac{\partial \Phi}{\partial \mathbf{A}} \tag{A.3}$$

$\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}$ is the flow vector, the direction of the plastic strain increment. In addition, $\mathrm{d}\lambda$ is plastic multiplier which determines the magnitude of the flow vector. $\mathbf{A}$ is the hardening thermodynamic force, and $\boldsymbol{\alpha}$ is the hardening internal variable. For associative isotropic hardening, $\boldsymbol{\alpha}$ and $\mathbf{A}$ are defined as

$$\boldsymbol{\alpha} \equiv \bar{\varepsilon}^p \tag{A.4}$$

$$\mathbf{A} \equiv k\left(\bar{\varepsilon}^{p}\right) \tag{A.5}$$

The loading/unloading conditions can be described using the Kuhn–Tucker optimality conditions as

$$\Phi\left(\boldsymbol{\sigma}, c\right) \leq 0, \quad \mathrm{d}\lambda \geq 0, \quad \Phi\left(\boldsymbol{\sigma}, c\right) \mathrm{d}\lambda = 0 \tag{A.6}$$

$\mathbf{D}^{\mathrm{ep}}$ is the consistent (algorithmic) tangent modulus in elastoplasticity [79, 80] and is defined as

$$\begin{aligned}
\mathbf{D}^{ep} = & 2G\left(1 - \frac{\Delta\lambda}{\sqrt{2}\left\|\left(\boldsymbol{\varepsilon}_{dev}^{e,\mathrm{trial}}\right)^{n+1(j)}\right\|}\right)\mathbf{I}_{dev} \\
& + 2G\left(\frac{\Delta\lambda}{\sqrt{2}\left\|\left(\boldsymbol{\varepsilon}_{dev}^{e,\mathrm{trial}}\right)^{n+1(j)}\right\|} - GA\right)\mathbf{D}^{e} \otimes \mathbf{D}^{e} \\
& - \sqrt{2}GAK\left(\eta\mathbf{D}^{e} \otimes \mathbf{I} + \eta\mathbf{I} \otimes \mathbf{D}^{e}\right) + K\left(1 - K\eta\eta A\right)\mathbf{I} \otimes \mathbf{I}
\end{aligned} \tag{A.7}$$

where $\boldsymbol{\varepsilon}_{dev}^{e,\mathrm{trial}}$ is the deviatoric trial strain, which is defined in the elastic predictor/plastic corrector algorithm [79]. $\mathbf{I}$ and $\mathbf{I}_{dev}$ are the second order unit tensor and its deviatoric, respectively. $G$ is the shear modulus, $K$ is the bulk modulus, and $A$ is specified as

$$A = \frac{1}{G + K\eta\eta + \xi^{2}H} \tag{A.8}$$

# Appendix B

# Finite Element Discretization for Poroplasticity Problem

By multiplying the strong form of (4.4) by a weight function $\delta\mathbf{u}$, integrating over the domain, and applying integration by parts, the corresponding weak form can be written as

$$\int_{\Gamma} \delta\mathbf{u}\,(\boldsymbol{\sigma}' \cdot \mathbf{n})\,d\Gamma - \int_{\Omega} \nabla\delta\mathbf{u} \cdot \boldsymbol{\sigma}'(u) d\Omega - \int_{\Omega} \delta\mathbf{u}\,\alpha\,(\nabla \cdot (p\mathbf{I}))\,d\Omega + \int_{\Omega} \delta\mathbf{u}\,\mathbf{f}_b d\Omega = 0 \tag{B.1}$$

The weighted residual method is now applied to the mass balance equation (4.8) and to its boundary condition. The weak form is written as

$$\int_{\Omega} \delta p \left(\frac{\phi}{K_f} + \frac{\alpha - \phi}{K_s}\right) \frac{\partial p}{\partial t} d\Omega + \int_{\Omega} \delta p \left(\alpha \nabla \cdot \frac{\partial \mathbf{u}}{\partial t}\right) d\Omega + \int_{\Gamma} \delta p\,(\mathbf{q} \cdot \mathbf{n})\,d\Gamma - \int_{\Omega} \nabla\delta p \cdot \mathbf{q}(p) d\Omega = 0 \tag{B.2}$$

The aim here is to find the field $\mathbf{u}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ such that the weak forms (B.1) and (B.2) are satisfied.

As explained in Section 4.1.2, the trial solutions $\mathbf{u}^h$ and $p^h$ and the weight functions $\delta\mathbf{u}^h$ and $\delta p^h$ are written using the finite element interpolations

$$\mathbf{u}^h(\mathbf{x}, t) = \mathbf{N}_u \mathbf{u}^t \tag{B.3}$$

$$\delta \mathbf{u}^h(\mathbf{x}) = \mathbf{N}_u \delta \mathbf{u} \tag{B.4}$$

$$p^h(\mathbf{x}, t) = \mathbf{N}_p \mathbf{p}^t \tag{B.5}$$

$$\delta p^h(\mathbf{x}) = \mathbf{N}_p \delta \mathbf{p} \tag{B.6}$$

in which $\mathbf{u}^t$ is the vector of nodal displacement and $\mathbf{p}^t$ is the vector of nodal pressure. $\delta\mathbf{u}$ and $\delta\mathbf{p}$ are the nodal values of the weight functions. $\mathbf{N}_u$ is a matrix of shape functions for displacement and $\mathbf{N}_p$ is a vector of shape functions for pressure. The gradients of $\mathbf{u}^h$, $\delta\mathbf{u}^h$, $p^h$, and $\delta p^h$ are written as

$$\nabla \mathbf{u}^h(\mathbf{x}, t) = \nabla \mathbf{N}_u \mathbf{u}^t = \mathbf{B}_u \mathbf{u}^t \tag{B.7}$$

$$\nabla \delta \mathbf{u}^h(\mathbf{x}) = \nabla \mathbf{N}_u \delta \mathbf{u} = \mathbf{B}_u \delta \mathbf{u} \tag{B.8}$$

$$\nabla p^h(\mathbf{x}, t) = \nabla \mathbf{N}_p \mathbf{p}^t = \mathbf{B}_p \mathbf{p}^t \tag{B.9}$$

$$\nabla \delta p^h(\mathbf{x}) = \nabla \mathbf{N}_p \delta \mathbf{p} = \mathbf{B}_p \delta \mathbf{p} \tag{B.10}$$

where $\mathbf{B}_u$ and $\mathbf{B}_p$ are the matrices of derivatives of the standard shape functions for the displacement and pressure fields, respectively.

Substituting the trial solutions (B.3, B.5), weight functions (B.4,B.6), and their gradients (B.7, B.8, B.9, B.10) into the weak forms (B.1, B.2) and evoking the arbitrariness of the $\delta\mathbf{u}$ and $\delta\mathbf{p}$ leads to the discretized form of the momentum and mass balance equations

$$\int_\Omega {\mathbf{B}_u}^\top \boldsymbol{\sigma}' \left(\mathbf{u}^h\right) \mathrm{d}\Omega - \mathbf{Q}\mathbf{p}^t = \mathbf{F}_u^{\mathrm{ext}}(t) \tag{B.11}$$

$$\int_\Omega {\mathbf{B}_p}^\top \mathbf{q}\left(p^h\right) \mathrm{d}\Omega + \mathbf{Q}^\top \frac{\partial \mathbf{u}^t}{\partial t} + \mathbf{S}^t \frac{\partial \mathbf{p}^t}{\partial t} = \mathbf{F}_p^{\mathrm{ext}}(t) \tag{B.12}$$

in which $\mathbf{F}_u^{\mathrm{ext}}$ and $\mathbf{F}_p^{\mathrm{ext}}$ are defined in (4.15). The matrices $\mathbf{Q}$ and $\mathbf{S}$ are the coupling and compressibility matrices, defined in (4.19) and (4.20).

Equations (B.11) and (B.12) can be written in the coupling form as

$$\begin{bmatrix} \mathbf{F}_u^{\text{int}}(\mathbf{u}^t) \\ \mathbf{F}_p^{\text{int}}(\mathbf{p}^t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{d}^t + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{Q}^\top & \mathbf{S}^t \end{bmatrix} \frac{\partial \mathbf{d}^t}{\partial t} = \begin{bmatrix} \mathbf{F}_u^{\text{ext}}(t) \\ \mathbf{F}_p^{\text{ext}}(t) \end{bmatrix} \tag{B.13}$$

where

$$\mathbf{F}_u^{\text{int}}(\mathbf{u}^t) = \int_\Omega \mathbf{B}_u{}^\top \boldsymbol{\sigma}'(\mathbf{u}^h)\, \mathrm{d}\Omega \tag{B.14}$$

$$\mathbf{F}_p^{\text{int}}(\mathbf{p}^t) = \int_\Omega \mathbf{B}_p{}^\top \mathbf{q}(p^h)\, \mathrm{d}\Omega \tag{B.15}$$

The implicit and unconditionally stable backwards Euler method is utilized to numerically integrate the semi-discrete equations (B.13) over time. The relation between the nodal unknowns at time steps $t-1$ and $t$ is defined as

$$\mathbf{d}^t = \mathbf{d}^{t-1} + \dot{\mathbf{d}}^t \Delta t \tag{B.16}$$

So, the coupled system of equations (B.13) for time step $t$ is written as

$$\mathbf{R}(\mathbf{d}^t) = \begin{bmatrix} \mathbf{F}_u^{\text{ext}}(t) \\ \Delta t \mathbf{F}_p^{\text{ext}}(t) \end{bmatrix} - \begin{bmatrix} \mathbf{F}_u^{\text{int}}(\mathbf{u}^t) - \mathbf{Q}\mathbf{p}^t \\ \Delta t \mathbf{F}_p^{\text{int}}(\mathbf{p}^t) + \mathbf{Q}^\top(\mathbf{u}^t - \mathbf{u}^{t-1}) + \mathbf{S}^t\mathbf{p}^t - \mathbf{S}^{t-1}\mathbf{p}^{t-1} \end{bmatrix} = \mathbf{0} \tag{B.17}$$