

© Copyright by Mohammad Sarraf Joshaghani 2019
All Rights Reserved

**MULTI-SCALE AND INTERFACE MECHANICS FOR POROUS MEDIA:
MATHEMATICAL MODELS AND COMPUTATIONAL FRAMEWORKS**

A Dissertation

Presented to

the Faculty of the Department of Civil and Environmental Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Civil Engineering

By

Mohammad Sarraf Joshaghani

August 2019

**MULTI-SCALE AND INTERFACE MECHANICS FOR POROUS MEDIA:
MATHEMATICAL MODELS AND COMPUTATIONAL FRAMEWORKS**

Mohammad Sarraf Joshaghani

Approved:

Chair of the Committee
Kalyana B. Nakshatrala, Associate Professor,
Dept. of Civil & Environmental Engineering

Committee Members:

Cumaraswamy Vipulanandan, Professor,
Dept. of Civil & Environmental Engineering

Yi. L. Mo, Professor,
Dept. of Civil & Environmental Engineering

Rodolfo Ostialla-Mónico, Assistant Professor,
Dept. of Mechanical Engineering

Matthew G. Knepley, Associate Professor,
Dept. of Computer Science & Engineering,
University at Buffalo

Béatrice Rivière, Professor and Chair,
Dept. of Computational & Applied Mathematics,
Rice University

Suresh K. Khator, Associate Dean,
Cullen College of Engineering

Roberto Ballarini, Professor and Chair,
Dept. of Civil & Environmental Engineering

Acknowledgments

My decision of going back to school to pursue a Ph.D. degree was one of the best decisions I made so far. It has truly been a life-changing experience for me, and it would not have been possible to do without the support and guidance that I received from many people. My greatest gratitude is given to my advisor Dr. Kalyana Babu Nakshatrala, for his guidance and encouragement throughout my graduate studies. Your commitment to delivering high-quality work and educating students is unmatched and has made a deep impression on me. Thank you for teaching me how to approach and deal with a problem scientifically.

Research in our CAML group would not have been fruitful without stimulating discussions on a variety of topics with Dr. Mohammad Shabouei, Dr. Saeed Karimi, and Tejasree Phatak. Special thanks to Dr. Hanie Joodat and Dr. Justin Chang for their mentorship and expertise in the field of computational mechanics and high performance computing. You have helped me greatly with my research.

My time in Houston was made enjoyable in large part due to the many friends and groups that became a part of my life. I am grateful for time spent with roommates and friends. Thank you Bavand, Amir, Ebrahim, Aram, Bijan, Taher, and Ahmed for creating great memories. I am also grateful to Dr. Tahereh Daneshi, who encouraged me during her visit to my hometown in 2011 that I should pursue graduate studies in the United States.

Lastly, I would like to say a heartfelt thank you to my parents Shirin and Ali; my siblings Marjan, Hossein, and Hasan; my nieces Niloofer, Yasaman, and Romina for always believing in me and encouraging me to follow my dreams. And Reza and Debbie for helping in whatever way they could during this challenging period. I love you all, and without your unconditional love, sacrifice, and support, I would not be where I am today. This dissertation is dedicated to you.

**MULTI-SCALE AND INTERFACE MECHANICS FOR POROUS MEDIA:
MATHEMATICAL MODELS AND COMPUTATIONAL FRAMEWORKS**

An Abstract

of a

Dissertation

Presented to

the Faculty of the Department of Civil and Environmental Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Civil Engineering

By

Mohammad Sarraf Joshaghani

August 2019

Abstract

There are many challenges in subsurface modeling. First, many important subsurface processes occur at the interfaces, either the interface of two different porous media (e.g., layered media) or the interface of free-porous media (e.g., hyporheic zones, arterial mass transport). Second, these processes (flow, transport, and mechanical deformation) are complex, coupled, and multi-physics by nature. Third, natural geomaterials such as fissured rocks often exhibit a pore-size distribution with two dominant pore scales. Fourth, the practical problems are invariably large-scale by nature. Thus, successful modeling of such processes in complex porous media requires: (i) an accurate prescription of flow dynamics within each region and at the interface, (ii) development of robust and accurate computational methods, and (iii) implementation and understanding of these models in a parallel and scalable high performance computing (HPC) environment.

This dissertation develops modeling strategies to advance the current state-of-the-art in subsurface modeling to address the challenges mentioned above. The specific aims are three-fold: First, we develop a comprehensive mathematical framework that provides a self-consistent set of conditions for flow dynamics at an interface. It will be shown that many of the popular interface conditions form special cases of the proposed framework. The approach hinges on extending the principle of virtual power to account for the power expended at the interface and then appealing to the calculus of variations.

Second, we present a discontinuous Galerkin formulation for the double porosity/permeability (DPP) model. We present a numerical procedure to discretize the interface conditions accurately. We develop numerical strategies to simulate and study the flow of fluids in porous media with complex pore-networks by using the DPP model. We also devise solver and parallel computing strategies to solve large-scale practical problems.

Third, we address the coupling of mechanical deformation of the porous solid with transport processes. We assume the porous solid to be an elastoplastic material, and transport of chemical species to be Fickian and develop a mathematical model and a

robust computational framework. These modeling tools can be applied to a variety of problems such as moisture diffusion in cementitious materials and consolidation of soils under severe loading-unloading regimes.

Table of Contents

Acknowledgments	v
Abstract	vii
Table of Contents	ix
List of Figures	xiii
List of Tables	xx
List of Abbreviations	xxi
Nomenclature	xxii
1 Introduction	1
1.1 An outline of the proposal	8
2 Background Material and Preliminaries	10
2.1 Double porosity/permeability (DPP) model	10
2.2 Grouping of field variables in continuous setting	12
2.3 Geometrical definitions	14
2.4 Average and jump operators	16
2.5 Mesh-related quantities	16
2.6 Functional analysis aspects	19
2.6.1 Inverse and trace inequalities	21
3 Classical and Stabilized Mixed Weak Formulations	23
3.1 Classical mixed formulation using $H(\text{div})$ elements	26
3.1.1 $H(\text{div})$ elements	26

3.2	Stabilized mixed continuous Galerkin formulation (CG-VMS)	28
3.3	Stabilized mixed DG formulation (DG-VMS)	29
3.3.1	Weak form in terms of numerical fluxes	31
3.3.2	The classical mixed DG formulation	33
3.3.3	Proposed stabilized mixed DG formulation	35
4	Theoretical Analysis	38
4.1	Stability norm	38
4.2	Convergence theorem and error analysis	39
5	Numerical Analyses and Results	52
5.1	Patch tests	52
5.1.1	Velocity-driven patch test	53
5.1.2	Non-conforming discretization	54
5.1.3	Non-constant Jacobian elements	67
5.2	Numerical convergence analysis	71
5.2.1	2D numerical convergence analysis:	71
5.2.2	3D numerical convergence analysis	72
5.3	Canonical problem and structure preserving properties	79
5.3.1	Quarter five-spot checkerboard problem	80
5.3.2	Element-wise mass balance	83
5.4	Coupled problem with heterogeneous medium properties	85
6	Proposed Solvers	91
6.1	Proposed four-field solvers	92
6.1.1	Method 1: splitting by scales	96
6.1.2	Method 2: splitting by fields	98
6.1.3	Computer implementation	100
7	Performance Model	101
7.1	Performance spectrum modeling	101

7.1.1	Mesh convergence	102
7.1.2	Strong-scaling	103
7.1.3	Static-scaling	104
7.1.4	Digits-of-Efficacy (DoE)	105
7.2	Representative numerical results	105
7.2.1	Two-dimensional study	106
7.2.2	Three-dimensional study	112
8	Interface Conditions for Flows in Coupled Free-Porous Media	133
8.1	Experimental observation and prior works	134
8.2	The proposed framework	136
8.2.1	The interface	136
8.2.2	Free flow region	138
8.2.3	Porous region	138
8.2.4	Fluid properties	139
8.2.5	Kinematically admissible and virtual fields	140
8.2.6	Other notation for convenience	141
8.2.7	Proposed principle of virtual power	141
8.3	Derivation of interface conditions and field equations	144
8.3.1	Handling internal constraints	145
8.3.2	Consequences of (P1) statement	145
8.4	Special cases	147
8.4.1	Beavers-Joseph condition	148
8.4.2	Beavers-Joseph-Saffman condition	150
8.4.3	No-slip condition	151
8.5	Minimum power theorem for a class of coupled flows	152
8.5.1	On construction of the potentials	153
8.5.2	Minimum power theorem	154

8.6	Uniqueness of solutions	155
9	A Modeling Framework for Coupling Plasticity With Species Diffusion	157
9.1	Mathematical model	157
9.1.1	Deformation subproblem	158
9.1.2	Transport subproblem	165
9.2	Proposed computational framework	169
9.2.1	A solver for deformation subproblem	169
9.2.2	A solver for diffusion subproblem	175
9.2.3	A coupling algorithm	179
9.2.4	Computer implementation	179
9.3	Performance of the computational framework	181
9.3.1	Degrdration of plate with a circular hole	181
9.3.2	Non-negative (NN) vs continuous Galerkin (CG)	184
9.3.3	Performance of the staggered scheme and the trust-region method	188
9.3.4	Physics of coupled deformation-diffusion problem	188
10	Conclusions and Future Work	199
10.1	Future Work	203
References		205
Appendices		227
A	Computer Implementation	228
B	PETSc Command-line Options	236
C	Spectral Performance Codes	238
D	Mathematical Proofs for Interface Condition	249

List of Figures

1.1	The left figure (adapted from the US Geological Survey [170]) shows a typical hyporheic zone. The right figure (adapted from the British Environment Agency [41]) depicts important processes that take place in a hyporheic zone.	2
1.2	The left figure shows a microscopic blood vessel in capillary network [127]. The right figure shows that by changing scale of observation, lumen wall membrane itself is composed of stack of porous media [176].	3
1.3	Different features of a typical coupled free-porous media model with complex pore-networks and highly heterogeneous permeability.	5
1.4	This figure shows the scope of current dissertation, which lies at the intersection of four topics.	7
2.1	This figure shows the decomposition of the domain into subdomains. External ($\partial\Omega$) and internal (Γ^{int}) boundaries of the domain, the interface (Υ) between two adjacent elements, and normal vectors to the boundaries are shown.	17
2.2	This figure illustrates the element diameter parameter h_ω and the diameter of the inscribed circle h_ω^{inc} for a typical element $\omega \in \mathcal{T}$	18
3.1	This figure shows the two-dimensional and three-dimensional elements that are employed in this chapter. The degrees-of-freedom (DoF) for each element are also indicated.	28
5.1	Velocity-driven patch test: This figure shows the computational domain, boundary conditions, and macro- and micro-permeabilities in each layer.	54
5.2	Velocity-driven patch test: Velocities are constant within each layer and pressures are linearly varying in the horizontal direction which are in agreement with the exact solution of this problem.	55
5.3	Velocity-driven patch test: This figure compares the velocities profiles obtained under the stabilized mixed CG formulation and the proposed DG formulation.	55
5.4	Non-conforming polynomial orders: The computational domain in the 2D setting is a unit square. Pressures are prescribed on the entire boundary of both pore-networks.	57

5.5	Non-conforming polynomial orders: In the left part of the domain, third order interpolation polynomials are used for velocities and pressures, while in the right part, first order interpolation polynomials are used.	58
5.6	Non-conforming polynomial orders: This parametric study demonstrates that an increase in η_p in the absence of η_u slightly improves the accuracy in capturing the jumps of the (macro- and micro-) velocities across a non-conforming edge.	59
5.7	Non-conforming polynomial orders: This figure shows a parametric study performed on the effect of η_u on velocity profiles. For the case of $\eta_p = 0$ and non-zero η_u , a drastic enhancement is captured with η_u of order 1. . .	60
5.8	Non-conforming polynomial orders: This figure shows a parametric study performed on the combined effect of η_u and η_p on minimizing the drifts of macro and micro-velocity fields.	62
5.9	Non-conforming polynomial orders: In the left half of the domain, third order interpolation polynomials are used for velocities and pressures, while in the right half, first order interpolation polynomials are used.	63
5.10	Non-conforming polynomial orders: This figure shows the exact and numerical solutions for the velocity profiles within the domain.	64
5.11	Non-conforming element refinement: The top figure shows the representative computational domain with non-conforming element refinement. The bottom figure shows the DG discretization of this domain.	66
5.12	Non-conforming element refinement: These results show that the proposed stabilized DG formulation is capable of handling non-conforming element refinement (with hanging nodes in the mesh).	67
5.13	Non-constant Jacobian elements: This figure shows two different computational domains and their corresponding meshes for the constant flow patch test. For this problem, non-constant Jacobian brick elements are used. . .	68
5.14	Non-constant Jacobian elements: Pressure and velocity profiles are shown for Mesh #1 (Fig. 5.13(a)) with non-constant Jacobian elements. . .	69
5.15	Non-constant Jacobian elements: Pressure and velocity contours are shown for Mesh #2 (Fig. 5.13(b)) with non-constant Jacobian elements. . .	70
5.16	Schematic of mesh types for convergence analysis: This figure provides a pictorial description of different types of mesh used in convergence study. The size of h is taken as 0.1 in the figure.	72
5.17	2D numerical convergence analysis on TRI mesh: This figure provides the convergence rates under h -refinement for various equal-order polynomial on triangular mesh.	73

5.18 2D numerical convergence analysis on TRI mesh: This figure shows the results of numerical convergence under p -refinement for a fixed mesh size ($h = 0.2$). The number of degrees-of-freedom corresponds to $p = 1$ to 8.	74
5.19 2D numerical convergence analysis for QUAD mesh: This figure provides the convergence rates under h -refinement for two sets of polynomials order of: 1 and 2, respectively, for velocity and pressure fields; and 2 for both velocity and pressure fields in the two pore-networks.	75
5.20 2D numerical convergence analysis for trapezoidal mesh: This figure provides the convergence rates under h -refinement for two sets of polynomials order of: 1 and 2, respectively, for velocity and pressure; and 2 for both velocity and pressure fields in the two pore-networks.	76
5.21 3D numerical convergence analysis on TRI mesh: This figure provides the convergence rates under h -refinement for various equal-order polynomials. The rate of convergence is polynomial, which is in accordance with the theory.	78
5.22 3D numerical convergence analysis on TRI mesh: This figure shows the results of numerical convergence under p -refinement for a fixed mesh size ($h = 0.2$). The number of degrees-of-freedom corresponds to $p = 1$ to 4. . . .	79
5.23 Quarter five-spot checkerboard problem: This figure shows the computational domain and boundary conditions. The heterogeneous domain is divided into four sub-regions with permeabilities shown in equation (5.10). . . .	81
5.24 Quarter five-spot checkerboard problem: This figure shows that steep pressure gradients near the injection and production wells are correctly captured under the proposed DG formulation. Results are obtained for $\eta_u = \eta_p = 0$	83
5.25 Quarter five-spot checkerboard problem: This figure compares the x-component of the macro-velocity (top) and micro-velocity (bottom) under the CG and the proposed DG formulations with different η_u and η_p	84
5.26 Element-wise mass balance: This figure shows the variation of the maximum element-wise inflow/outflow flux with interpolation polynomial orders. . .	86
5.27 Element-wise mass balance: This figure shows the local mass balance error under both CG and DG formulations for cubic equal-order interpolation for all the variables. As can be seen, the DG formulation returns smaller errors.	86
5.28 Coupled flow and transport problem: This figure shows the pictorial description of coupled flow-transport problem with heterogeneous medium properties along with initial and boundary conditions.	89

5.29	Coupled flow and transport problem: This figure shows that the proposed formulation can capture well-known instabilities in fluid mechanics, similar to viscous-fingering instability, in a heterogeneous, layered porous domain.	90
7.1	Two-dimensional problem: This figure provides a pictorial description of the boundary value problem.	107
7.2	Two-dimensional problem: This figure shows the typical meshes employed in our numerical simulations.	107
7.3	Two-dimensional problem using TRI mesh: This figure compares the mesh convergence results for the chosen finite element formulations under various solver tolerances. The results are shown for field-splitting block solver methodology.	110
7.4	Two-dimensional problem using QUAD mesh: This figure compares the mesh convergence results for the chosen finite element formulations under various solver tolerances. The results are shown for field-splitting methodology.	111
7.5	Two-dimensional problem: This figure compares the static-scaling results for the chosen finite element formulations using TRI and QUAD meshes.	113
7.6	Two-dimensional problem for TRI mesh: This figure compares the Digits of Efficacy (DoE) among the chosen finite element formulations. Results for both solver methodologies with a tolerance of 10^{-7} are reported.	114
7.7	Two-dimensional problem for QUAD element: This figure compares the Digits of Efficacy (DoE) among the chosen finite element formulations. Results for both solver methodologies with a tolerance of 10^{-7} are reported.	115
7.8	Three-dimensional domain: This figure provides a pictorial description of the boundary value problem.	116
7.9	Three-dimensional domain: This figure shows the typical meshes employed in our numerical simulations.	116
7.10	Three-dimensional problem: This figure compares the mesh convergence results for the chosen finite element formulations using TET and HEX meshes. The solver tolerance is taken to be 10^{-7}	124
7.11	Three-dimensional problem: This figure compares the static-scaling results for the chosen finite element formulations using TET and HEX meshes.	125
7.12	Three-dimensional problem using TET mesh: This figure compares the Digits of Efficacy (DoE) for the chosen finite element formulations.	127
7.13	Three-dimensional problem using HEX mesh: This figure compares the Digits of Efficacy (DoE) for the chosen formulations.	128

8.1	A pictorial description of the rectilinear flow in a horizontal channel between an impervious upper wall and a pervious lower wall (at $y = 0$).	135
8.2	A pictorial description of coupled free-porous media. The free flow region $\mathcal{K}_{\text{free}}$ and the porous region \mathcal{K}_{por} share a common interface Γ_{int}	137
9.1	Model I: left figure shows the effect of coupling parameter c_{ref} on the onset of plastic yielding. Right figure shows one-dimensional uniaxial stress-strain relationship ($\sigma-\epsilon$) undertaking “degradation via elasticity parameters” model.	164
9.2	Model II: left figure depicts consistent decline in elastic limit function as the coupling power $ \zeta c $ increases. Right figure shows one-dimensional stress-strain path under uniaxial tension ($\sigma-\epsilon$), employing the degradation via model II.	165
9.3	Uncoupled models: This schematic shows one-dimensional stress-strain relationship for uncoupled model I and model II; and compares them with standard models under one-dimensional uniaxial tension.	166
9.4	One-way and two-way coupling strategies for deformation-diffusion system. In this figure \mathbf{R} denotes the rotation tensor.	168
9.5	Plate with a circular hole: This figure provides a pictorial description of the geometry and boundary value problem.	182
9.6	Plate with a circular hole: This figure shows the tree-node triangular mesh used in the numerical results. Position of points A and B, and lines C and D are also marked.	182
9.7	Degradation model I: This figure compares concentration profiles from the continuous Galerkin formulation and the non-negative formulation at two stages. Gray-scale region in the left figures indicates violated concentrations.	185
9.8	Degradation model II: This figure compares concentration profiles from the continuous Galerkin formulation and the non-negative formulation. Gray-scale region in the left figure indicates violated concentrations.	186
9.9	Continuous Galerkin vs non-negative formulation: This figure illustrates the concentration profile on path D during three loading steps. Coordinates of path D is marked in Fig 9.6	187
9.10	Residual loading step: This figure compares stress profile and effective plastic strain contours from the continuous Galerkin formulation and the non-negative formulation at residual loading step.	187

9.11 Plate with a circular hole: This figure shows displacement history at point A and B.	192
9.12 Stress, degradation model I: This figure illustrates concentration profile and contour of effective plastic strain at maximum and residual loading steps.	194
9.13 Stress, degradation model II: This figure illustrates concentration profile and contour of effective plastic strain at maximum and residual loading steps.	195
9.14 degradation model I: This figure shows the plastic zone at maximum and residual loading steps.	196
9.15 degradation model II: This figure shows the plastic zone at maximum and residual loading steps.	197
9.16 Plate with a circular hole: This figure shows percentage of yielded area for degradation model I and II.	198

List of Tables

3.1	The element-level discretization for different mesh types and the chosen three formulations. \oplus denotes the direct sum operator between two finite element spaces.	29
5.1	Model parameters for velocity-driven patch test.	53
5.2	Model parameters for non-conforming polynomial orders, element-wise mass balance study, and 2D numerical convergence analysis.	57
5.3	Model parameters for non-conforming element refinement problem. . . .	66
5.4	Model parameters for 3D computational domains with non-constant Jacobian elements.	68
5.5	Model parameters for 3D numerical convergence analysis.	77
5.6	Model parameters for the quarter five-spot checkerboard problem. . . .	82
5.7	Model parameters for coupled flow and transport problem in the heterogeneous domain.	89
7.1	Parameters for two-dimensional problem.	106
7.2	This table illustrates degrees-of-freedom for two-dimensional h-size refinement study.	109
7.3	Parameters for three-dimensional problem.	114
7.4	This table shows h-size and corresponding degrees-of-freedom for three-dimensional strong-scaling studies. In test1, we have fixed the h -size to $1/16$ for all finite element formulations. In test2, we selected h -size for each formulation such that all end up to have roughly the same DoF of approximately 200k.	119
7.5	3D problem: Strong-scaling results for H(div) formulation with same h-size.	120
7.6	3D problem: Strong-scaling results for CG-VMS formulation with same h-size.	121
7.7	3D problem: Strong-scaling results for DG-VMS formulation with same h-size.	122
7.8	3D problem: Strong-scaling results for H(div) formulation with same DoF count.	129

7.9	3D problem: Strong-scaling results for CG-VMS formulation with same DoF count.	130
7.10	3D problem: Strong-scaling results for DG-VMS formulation with same DoF count.	131
7.11	This table illustrates three-dimensional h-size refinement for each discretization such that at each step DoF approximately doubles.	132
9.1	Parameters for plane with a circular hole problem.	183
9.2	Degradation model I: This table shows time-to-solution and iteration counts under continuous Galerkin and non-negative strategies for both deformation and diffusion subproblems.	189
9.3	Degradation model II: This table shows time-to-solution and iteration counts under continuous Galerkin and non-negative strategies for both deformation and diffusion subproblems.	190

List of Abbreviations

AMG	Algebraic Multi-Grid;
BDFM	Brezzi-Douglas-Fortin-Marini;
BDM	Brezzi-Douglas-Marini;
BJ	Beaver-Joseph;
BLMVM	Bound Limited Memory Variable Metric;
CG	Continuous Galerkin;
C&N	Carbon and Nitrogen;
DG	Discontinuous Galerkin;
DoA	Digits of Accuracy;
DoE	Digits of Efficacy;
DoF	Degrees of Freedom;
DoS	Digits of Size;
DPP	Double Porosity Permeability;
GMRES	Generalized Minimal Residual Method;
HEX	Hexahedron;
HYPRE	Parallel High Performance Preconditioners;
ILU	Incomplete LU;
KSP	Abstract PETSc object that manages all Krylov methods;
LBB	Ladyzhenskaya–Babuška–Brezzi;
LDL	Low Density Lipoproteins;
MMS	Method of Manufactured Solutions;
MPI	Message Passing Interface;

MREP	Maximum Rate of Entropy Production;
PETSc	Portable Extensible Toolkit for Scientific Computation;
QUAD	Quadrilateral;
RT	Raviart-Thomas;
RTN	Raviart-Thomas-Nédélec;
SUPG	Streamline Upwind Petrov-Galerkin;
TAS	Time-Accuracy-Size;
TET	Tetrahedron;
TIC	Theory of Interacting Continua;
TRI	Triangle;
VF	Viscous Fingering;
VMS	Variational Multi-Scale;

Nomenclature

The following key symbols are used:

- 1 = Macro-pore network;
- 2 = Micro-pore network;
- $\mathbf{u}_1, \mathbf{u}_2, p_1, p_2$ = Velocity and pressure solution fields in the two pore-networks;
- k = Isotropic permeability;
- k_1, k_2 = Permeabilities in the two pore-networks;
- γ, μ = True density and coefficient of viscosity of the fluid;
- $u_{n1}, u_{n2}, p_{01}, p_{02}$ = Prescribed velocities and pressures;
- χ = Mass exchange across the pore-networks;

- α = BJ material constant;
 β = Parameter in the inter-pore mass transfer;
 η = Flow characterization parameter in the DPP model;
 σ = One-dimensional stress;
 σ_y^* = Elastic limit function;
 ε = One-dimensional strain;
 κ = Effective plastic strain;
 ϕ = Effective stress measure;
 η_u, η_p = Stabilization parameters;
 ϕ = Effective stress measure;
 h = Mesh-size;
 h_ω = Element diameter of ω ;
 h_ω^{inc} = Diameter of the inscribed circle in ω ;
 h_Y = Characteristic length of an edge;
 nd = Number of spatial dimensions;
 c, D = Concentration and diffusivity;
 $\omega^i, \partial\omega^i$ = The i -th subdomain and its boundary;
 $m(\omega)$ = Net rate of volumetric flux from element ω ;
 m_{\max}^{out} = Maximum element-wise mass outflow flux;
 m_{\max}^{in} = Maximum element-wise mass inflow flux;
 I_{kin} = Kinetic hardening function;
 f = Yield criterion function;
 u_B = Discharge velocity at the interface;
 E_t = Tangent modulus;

H	=	Isotropic hardening modulus;
Q	=	Discharge velocity in porous domain;
$\Omega, \bar{\Omega}, \partial\Omega$	=	Computational porous domain, its set closure, and its boundary;
$\Gamma_1^u, \Gamma_2^u, \Gamma_1^p, \Gamma_2^p$	=	Velocity and pressure boundaries;
$\mathcal{T}, \mathcal{T}_h$	=	A mesh, and a mesh with mesh-size h ;
Ψ	=	Power expended density along free-porous interface;
Ψ_{nrg}	=	Free energy function;
$\mathcal{C}_{\text{drag},1}, \mathcal{C}_{\text{drag},2}$	=	Bounds on drag coefficients;
$\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}}, \mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}}, \mathcal{C}_{\text{int}}$	=	Constant in standard estimate for interpolation error;
\mathcal{C}_{inv}	=	Constant in discrete inverse inequality;
\mathcal{C}_{lqu}	=	Locally quasi-uniform coefficient;
\mathcal{C}_{sp}	=	Shape parameter;
$\mathcal{C}_{\text{trace}}$	=	Constant in continuous trace inequality;
N_{ele}	=	Number of subdomains (elements);
$\tilde{\Omega}$	=	Union of all open subdomains;
$\mathcal{E}, \mathcal{E}^{\text{int}}$	=	Sets of all and interior edges, respectively;
Υ	=	A typical edge (i.e., $\Upsilon \in \mathcal{E}$ or $\Upsilon \in \mathcal{E}^{\text{int}}$);
Γ^{int}	=	Union of internal boundaries;
$\mathcal{K}_{\text{free}}, \mathcal{K}_{\text{por}}$	=	Free and porous regions;
$\partial\mathcal{K}_{\text{free}}, \partial\mathcal{K}_{\text{pore}}$	=	Whole boundaries of free and porous regions;
$\partial\mathcal{K}_{\text{free}}^{\text{ext}}, \partial\mathcal{K}_{\text{por}}^{\text{ext}}$	=	External boundaries of free and porous regions;
\mathcal{W}	=	Strain energy density;
\mathcal{H}	=	Hardening potential;
$(\cdot; \cdot)_{\mathcal{K}}, (\cdot; \cdot)$	=	L_2 inner-products over \mathcal{K} and $\tilde{\Omega}$, respectively;
$\ \cdot\ _{\mathcal{K}}, \ \cdot\ $	=	L_2 norms over \mathcal{K} and $\tilde{\Omega}$, respectively;

$\mathbf{w}_1, \mathbf{w}_2, q_1, q_2$	=	Weighting functions for velocities and pressures;
$\mathbf{u}^*, \mathbf{u}^*_2, p^*_1, p^*_2$	=	Numerical fluxes;
$\mathbf{b}_1, \mathbf{b}_2$	=	Specific body forces in the pore-networks;
$\mathbf{t}_{\text{free}}, \mathbf{t}_{\text{por}}$	=	Traction on the free and porous regions;
$\hat{\mathbf{n}}(\mathbf{x})$	=	Unit outward normal vector at $\mathbf{x} \in \partial\Omega$;
$\hat{\mathbf{n}}_{\text{free}}, \hat{\mathbf{n}}_{\text{free}}$	=	Unit outward normals of free and porous domains;
$\hat{\mathbf{n}}_{\text{free}}^{\text{ext}}, \hat{\mathbf{n}}_{\text{free}}^{\text{ext}}$	=	Unit outward normals tot the external boundaries;
\mathbf{i}^{por}	=	Interaction term for fluid;
$\hat{\mathbf{s}}$	=	Unit tangential vector along the interface;
$\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}$	=	Velocity in the free and porous region;
$\boldsymbol{\alpha}$	=	back stress tensor;
$\boldsymbol{\xi}$	=	Shifted stress;
Υ_1	=	Scale-splitting of DPP field variables;
Υ_2	=	Field-splitting of DPP field variables;
$\text{grad}[\cdot], \text{div}[\cdot]$	=	The gradient and divergence operators w.r.t a spatial point $\mathbf{x} \in \Omega$;
$\{\!\!\{\cdot\}\!\!\}, [\![\cdot]\!]$	=	Average and jump operators;
$\mathcal{P}^m(\omega)$	=	Set of all polynomials over ω up to and including m -th order;
\mathcal{E}	=	Set of internal variables;
$\mathcal{P}^{\text{internal}}$	=	Internal virtual power;
\mathcal{Q}	=	Set of stress like hardening variables;
\mathcal{L}	=	Linear differential operator;
\mathcal{F}	=	Forcing function;
\oplus	=	Direct sum operator between two finite element spaces;
\mathbf{I}, \mathbf{O}	=	Identity tensor, and zero tensor;

- \mathbf{R} = Rotation tensor;
 \mathbf{T} = Cauchy stress tensor;
 $\mathbf{T}_{\text{free}}^{\text{extra}}, \mathbf{T}_{\text{por}}^{\text{extra}}$ = Extra Cauchy stress in the free and porous regions;
 $\mathbf{E}^e, \mathbf{E}^p$ = Elastic and plastic strain tensor;
 \mathbf{K} = Stiffness matrix;
 \mathbf{S} = Schur complement matrix;
 \mathbf{S}_p = preconditioner needed for \mathbf{S}^{-1} ;
 $\hat{\mathbf{N}}$ = Unit deviatoric tensor;
 $\mathbf{D}_T, \mathbf{D}_S$ = Reference diffusivity tensor under tensile and shear strains;
 $\text{tr}[\cdot]$ = Standard trace of second-order tensor;
 \mathbb{C} = Fourth-order elastic tensor;
 \mathbb{E} = Fourth-order tangent constitutive tensor.

Chapter 1

Introduction

Emergence of computational technologies and demand of industries have pushed the approach of combining mechanics, computational methods and HPC to the forefront. Many important science and engineering problems involve flow and transport in a domain which comprises free flow and porous regions. In these problems, a plethora of vital processes takes place within each regions and also near the interface of free flow and porous regions. One has to capture these processes accurately to discern the overall dynamics and all the interactions in the entire domain. We now discuss two such problems, which have motivated us to undertake the research presented hereby.

The *first problem* pertains to the hyporheic zone, located beneath a stream bed, where mixing of surface and shallow groundwater (subsurface water) takes place. Myriads of geological and hydrochemical (coupled) processes happen at this zone, which are vital to water distribution, environmental safety and can affect the whole food chain [95, 166]. As shown in **Fig. 1.1**, the zone may contain multiple layers of porous medium (with different material properties) deposited vertically where top layer interacts (i.e., exchange water and species) at the interface with free-flow domain.

The *second problem* pertains to the arterial mass transfer—the transport of atherogenic macromolecules, such as low density lipoproteins (LDL), from bulk blood flow into artery walls and vice versa [168, 174] (see **Fig. 1.2**). Accumulation of LDL at the interface of bulk blood flow and the endothelial layer—the part of lumen next to the blood

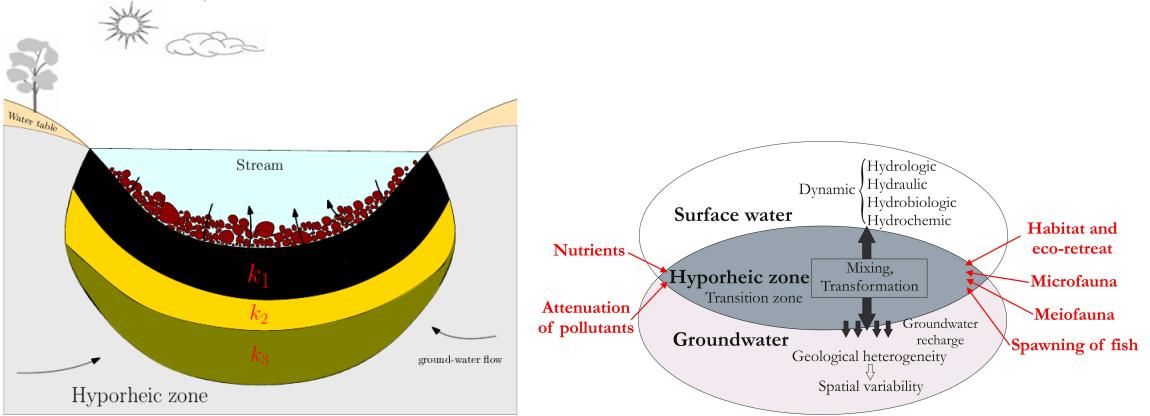


Figure 1.1: The left figure (adapted from the US Geological Survey [170]) shows a typical hyporheic zone. The right figure (adapted from the British Environment Agency [41]) depicts important processes that take place in a hyporheic zone.

flow—is a primary cause of various cardiovascular diseases; for example, atherosclerotic lesions within the intima of arteries [43, 84]. A firm understanding of transluminal flow and transport will enable physicians to administer better therapeutic procedures. By changing the scale of observation, the artery wall itself is composed of stack of porous membranes wrapped around the (free-flow) lumen.

Predictive modeling of coupled free-porous, multi-physics problems (with strong material heterogeneity) such as arterial mass transfer and hyporheic zone needs: (i) a firm understanding of *underlying mechanics* (both at the interface of free-porous and within each regions), (ii) development of robust and accurate *computational methods*, and (iii) implementation and understanding of these models in a parallel and scalable *HPC* environment.

On the mechanics front, three outstanding problems will involve:

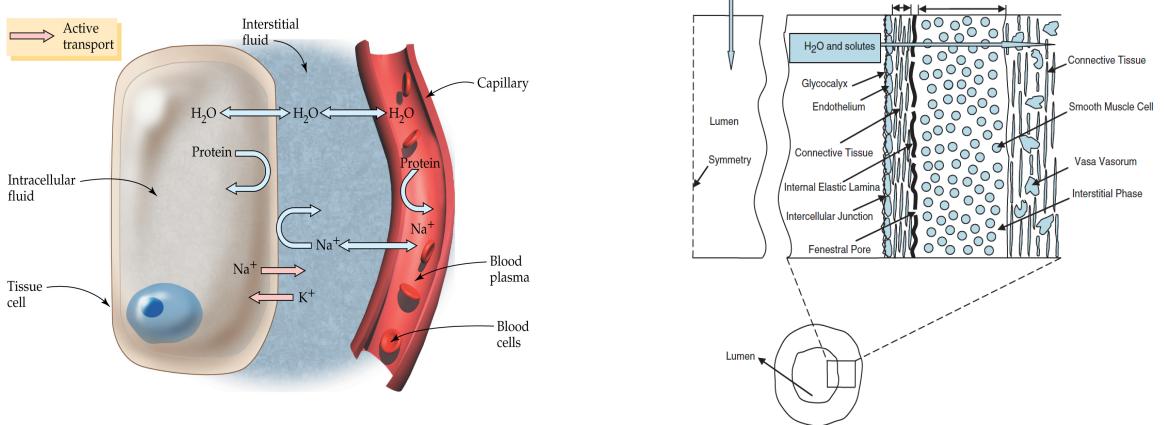


Figure 1.2: The left figure shows a microscopic blood vessel in capillary network [127]. The right figure shows that by changing scale of observation, lumen wall membrane itself is composed of stack of porous media [176].

1. *Theoretical interface conditions for coupled flow dynamics at the interfaces.* Employing purely empirical interface conditions at the physics disparity (e.g., free-porous media interface) are prevalent in the literature¹. However, there is a knowledge gap on a unified theoretical framework (with underpinning mechanics) to obtain a complete set of interface conditions which capture the prior experimental relations.
2. *Porous media with complex microstructure.* In the recent years, there has been an increasing demand on the exploration of hydrocarbons from unconventional sources (i.e., oil and gas from tight shale), [167]. Laboratory experiments confirm that many of these geomaterials such as (tight) shale rocks exhibit two or more dominant pore-scales connected by two dominant pore-networks [130]. On the other hand, additive manufacturing now enables us to easily create porous media with complex porous structure. We can now direct hardware to deposit material layer upon layer, in

¹Physics disparity interfaces should not be confused with interfaces at the material disparity. Material disparity, also known as heterogeneity, are observed within a region at the vicinity of jump in material properties such as permeability.

precise geometric shapes. One example of this complex porous media is biological implants that are designed to replace, support, or enhance a damaged biological structure. These demands well justify to account for multi-porosity in our coupled free flow-porous system. As illustrated in **Fig. 1.3**, at each layer of porous domain two or more dominant pore networks could coexist with the possibility of mass exchange between pore-scales. Hence, a mathematical model with strong continuum thermomechanics underpinning is required to address double porosity/permeability (DPP) or multi-porosity problem.

3. *Systematic development of multi-physics of coupled problems* Cementitious materials are susceptible to degradation in their lifespan due to external stimuli, which can be in the form of mechanical loading, temperature, transport of chemical species within the material, chemical reactions, or radiation. Degradation, which involves deterioration of host material, is usually modeled via a coupled mechanism such as coupled deformation-diffusion system. A firm understanding of this system and its effects on the mechanical response is central to a wide variety of problems. Prior experiments have shown that the presence and diffusion of a chemical species affect the plastic material properties; for example, the elastic yield functions depends on the concentration of the species. Such dependence on material properties affects the plastic deformation of the material.

In addition to deriving a robust mathematical model which incorporates various aspects of mechanics, one should also be cognizant that these mathematical models are not amenable to analytical solutions and existing numerical formulations are not adequate. Thus an accurate and convergent computational framework for modeling these systems needs to be developed. Excessive care should be taken with respect to modeling diffusion of a chemical species which is an omnipresent phenomenon in geophysical problems. One

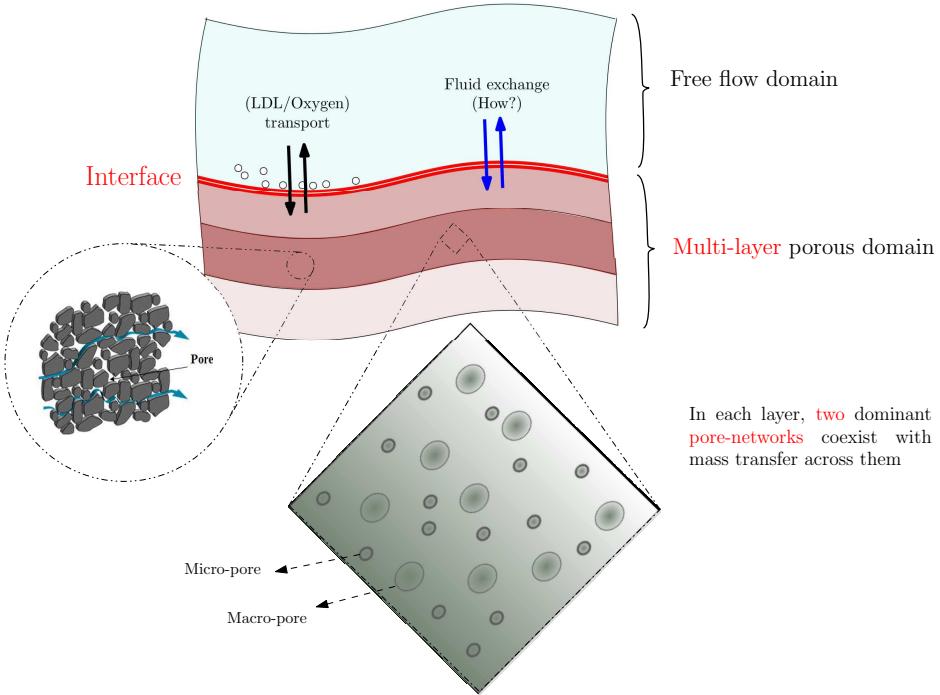


Figure 1.3: Different features of a typical coupled free-porous media model with complex pore-networks and highly heterogeneous permeability.

encounters several challenges in solving coupled problems that involve a diffusion process (e.g., degradation problems). The central one is about producing non-negative solutions under the diffusion equations. It is well-known that popular numerical formulations for transport equations do not satisfy maximum principle and the non-negative constraint. These violations are prominent when the diffusion process is anisotropic. Using such formulations for diffusion problems in conjunction with other problems such as elastoplastic deformation will produce unreliable solutions.

Apart from the above challenges, the application problems in these areas are typically large-scale in nature. These large-scale problems cannot be solved on a standard desktop or by employing direct solvers, as such a computation will be prohibitively expensive. and hence one needs to employ parallel computing tools and HPC techniques to tackle these problems.

It is beyond the scope of a PhD thesis to address all the aspects of the overarching goal. Herein, critical knowledge gaps are identified, and solutions will be provided. Below is the list of these objectives, which all stems from the overarching goal.

- Develop a framework for obtaining appropriate conditions for coupled flow dynamics at the interface of free-porous media. Also, recover some popular conditions available in the literature for coupled flows as special cases of the proposed framework.
- Propose a stabilized formulation for the multi-scale porous media (with high material heterogeneity), which suppresses the non-physical numerical instabilities in coupled flow-transport problems, but captures the physical instabilities.
- Propose two composable block solver methodologies to solve $\mathbf{Kx} = \mathbf{b}$ that arise from finite element modeling of multi-scale porous media. Also, use performance spectrum model to gauge solvers scalability and comment on the choice of finite element formulations in large scale problems.
- Propose a mathematical model which describes the response of an elastoplastic material due to the diffusion of a chemical species within the material. Also, developing a computational framework for the resulting system of coupled equations that satisfies maximum principle and the non-negative constraint. This problem could be seen as a precursor to incorporating plastic deformations of the porous matrix into the interface problem.

The theme of the current work is described in **Fig. 1.4**. We resort to different theoretical and computational approaches to pursue the objectives outlined earlier. For deriving a self-consistent boundary conditions at the interface of the free-porous region, our approach will utilize the principle of virtual power and the theory of interacting

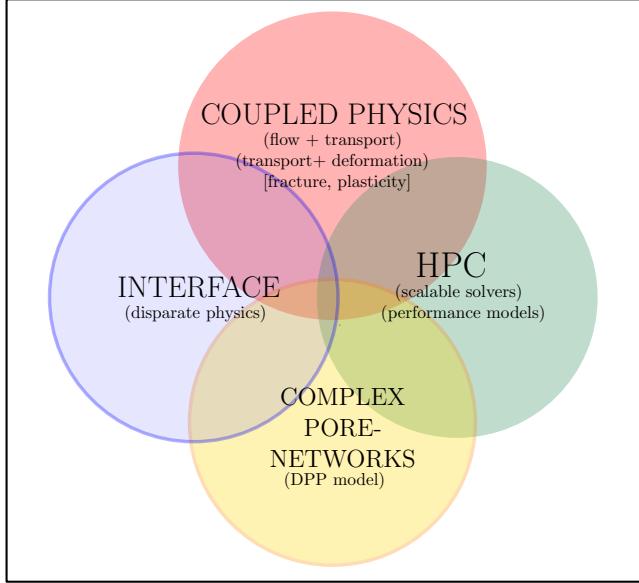


Figure 1.4: This figure shows the scope of current dissertation, which lies at the intersection of four topics.

continua. We invoke a geometric argument to enforce the internal constraints, impose the principle of material frame-indifference on all the constitutive relations and use the standard results from the calculus of variations. To numerically solve the DPP model in a highly heterogeneous domain, we propose a stabilized mixed discontinuous Galerkin formulation. This formulation will enjoy several attractive features. Using the composable solvers feature available in PETSc and the finite element libraries available under the Firedrake Project, we illustrate two different ways by which one can effectively precondition systems of equations resulting from large-scale DPP problem. Also, we employ the performance model called the Time-Accuracy- Size (TAS) spectrum model to demonstrate that the proposed composable block solvers are scalable in both the parallel and algorithmic sense. Finally, we develop a computational framework for modeling coupled elastoplastic-diffusion system. Our approach will employ an optimization-based solver that respects maximum principle and satisfies physical constraints like the non-negative constraint.

1.1 An outline of the proposal

Background material and preliminaries (including the governing equations of the DPP model, and convenient grouping of the variables) are provided in chapter 2. The weak forms of the classical, and continuous Galerkin mixed finite element formulations are presented along with the proposed stabilized mixed DG formulation in chapter 3. A systematic convergence analysis and the error estimation of the proposed DG formulation are carried out in chapter 4. Chapter 5 begins with constant flow patch tests and sensitivity studies on the stabilization parameters. Numerical convergence analysis and structure preserving properties are provided. In addition, the proposed DG formulation is implemented to study viscous-fingering-type physical instabilities in heterogeneous porous media with double pore-networks. Two block solver methodologies for preconditioning linear algebraic system arises from DPP are discussed in detail in chapter 6. The framework of the performance spectrum model along with the guidelines on how to interpret the resulting diagrams are presented in chapter 7. Using numerical simulations, the performance of the proposed block solvers are gauged. In the same chapter, we also compare the performance of the chosen finite element discretizations using the TAS performance spectrum model. Using the principle of virtual power at the interface of free-porous regions, we obtain complete interface conditions in chapter 8. We then proceed to show that the popular conditions – Beavers-Joseph (BJ) and Beavers-Joseph-Saffman (BJS) conditions – are special cases of the proposed framework. In chapter 9, we present a mathematical model which describes the response of a elastoplastic material due to the diffusion of a chemical species within the material. We also present a computational framework for the resulting system of coupled equations and illustrate the predictive capabilities of the proposed computational framework. This chapter ends with a discussion on the physics of the deformation of an elasto-plastic material under material degradation

due to the transport of a chemical species. Finally, conclusions and future trajectories are drawn in chapter 10.

Chapter 2

Background Material and Preliminaries

2.1 Double porosity/permeability (DPP) model

A class of models, which is commonly referred to as double porosity/permeability (DPP) models, have been found to be particularly attractive in modeling flows in porous media with two pore-networks (e.g., see [21, 175, 66, 27, 135, 51]). Recently, a DPP mathematical model with strong continuum thermomechanics underpinning has been derived in [135]. This model, which will be central to this text and will be referred to as *the DPP model* from here on, describes the flow of a single-phase incompressible fluid in a rigid porous medium with two distinct pore-networks, with a possible mass transfer across the pore-networks. The governing equations form a boundary value problem in terms of four-fields and the nature of the PDE is elliptic under steady-state responses. We refer to the two pore-networks as macro-pore and micro-pore networks, which are denoted by subscripts 1 and 2, respectively. We denote the porous domain by $\Omega \subset \mathbb{R}^{nd}$, where “ nd ” represents the number of spatial dimensions. For a precise mathematical treatment, we assume that Ω is an open bounded domain. The boundary $\partial\Omega = \bar{\Omega} - \Omega$ is assumed to be smooth, where the superposed bar denotes the set closure. A spatial point is denoted by $\mathbf{x} \in \bar{\Omega}$. The gradient operator with respect to \mathbf{x} is denoted by $\text{grad}[\cdot]$ and

the corresponding divergence operator is denoted by $\text{div}[\cdot]$. The unit outward normal to the boundary is denoted by $\hat{\mathbf{n}}(\mathbf{x})$. The pressure and the discharge (or Darcy) velocity fields in the macro-pore network are, respectively, denoted by $p_1(\mathbf{x})$ and $\mathbf{u}_1(\mathbf{x})$, and the corresponding fields in the micro-pore network are denoted by $p_2(\mathbf{x})$ and $\mathbf{u}_2(\mathbf{x})$. We denote the viscosity and true density of the fluid by μ and γ , respectively.

The abstract boundary value problem under the DPP model takes the following form:

Find $\mathbf{u}_1(\mathbf{x})$, $\mathbf{u}_2(\mathbf{x})$, $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ such that

$$\mu k_1^{-1} \mathbf{u}_1(\mathbf{x}) + \text{grad}[p_1(\mathbf{x})] = \gamma \mathbf{b}(\mathbf{x}) \quad \text{in } \Omega, \quad (2.1a)$$

$$\mu k_2^{-1} \mathbf{u}_2(\mathbf{x}) + \text{grad}[p_2(\mathbf{x})] = \gamma \mathbf{b}(\mathbf{x}) \quad \text{in } \Omega, \quad (2.1b)$$

$$\text{div}[\mathbf{u}_1(\mathbf{x})] = +\chi(\mathbf{x}) \quad \text{in } \Omega, \quad (2.1c)$$

$$\text{div}[\mathbf{u}_2(\mathbf{x})] = -\chi(\mathbf{x}) \quad \text{in } \Omega, \quad (2.1d)$$

$$\chi(\mathbf{x}) = -\frac{\beta}{\mu}(p_1(\mathbf{x}) - p_2(\mathbf{x})) \quad \text{in } \Omega, \quad (2.1e)$$

$$\mathbf{u}_1(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}) = u_{n1}(\mathbf{x}) \quad \text{on } \Gamma_1^u, \quad (2.1f)$$

$$\mathbf{u}_2(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}) = u_{n2}(\mathbf{x}) \quad \text{on } \Gamma_2^u, \quad (2.1g)$$

$$p_1(\mathbf{x}) = p_{01}(\mathbf{x}) \quad \text{on } \Gamma_1^p, \text{ and} \quad (2.1h)$$

$$p_2(\mathbf{x}) = p_{02}(\mathbf{x}) \quad \text{on } \Gamma_2^p, \quad (2.1i)$$

where $k_1(\mathbf{x})$ and $k_2(\mathbf{x})$, respectively, denote the (isotropic) permeabilities of the macro-pore and micro-pore networks, $\mathbf{b}(\mathbf{x})$ denotes the specific body force, and β is a dimensionless characteristic of the porous medium. $\chi(\mathbf{x})$ accounts for the mass exchange across the pore-networks and is the rate of volume transfer of the fluid between the two pore-networks per unit volume of the porous medium. The dimension of $\chi(\mathbf{x})$ is one over the time [$M^0 L^0 T^{-1}$]. Γ_i^u denotes that part of the boundary on which the normal component of velocity is prescribed in the macro-pore ($i = 1$) and micro-pore ($i = 2$) networks, and $u_{n1}(\mathbf{x})$ and $u_{n2}(\mathbf{x})$ denote the prescribed normal components of the velocities on Γ_1^u and

Γ_2^u , respectively. Γ_i^p is that part of the boundary on which the pressure is prescribed in the macro-pore ($i = 1$) and micro-pore ($i = 2$) networks, and $p_{01}(\mathbf{x})$ and $p_{02}(\mathbf{x})$ denote the prescribed pressures on Γ_1^p and Γ_2^p , respectively.

For mathematical well-posedness, we assume that

$$\Gamma_1^u \cup \Gamma_1^p = \partial\Omega, \quad \Gamma_1^u \cap \Gamma_1^p = \emptyset, \quad \Gamma_2^u \cup \Gamma_2^p = \partial\Omega, \quad \text{and} \quad \Gamma_2^u \cap \Gamma_2^p = \emptyset. \quad (2.2)$$

However, if $\Gamma_1^p = \emptyset$ and $\Gamma_2^p = \emptyset$ hold simultaneously then one will be able to find the pressure in each pore-network only up to an arbitrary constant. We assume that the drag coefficients in the two pore-networks, μ/k_1 and μ/k_2 , are bounded below and above. That is,

$$0 < \inf_{\mathbf{x} \in \Omega} \frac{\mu}{k_i(\mathbf{x})} \leq \sup_{\mathbf{x} \in \Omega} \frac{\mu}{k_i(\mathbf{x})} < +\infty \quad i = 1, 2. \quad (2.3)$$

This also means that there exist two non-dimensional constants $1 \leq \mathcal{C}_{\text{drag},1}$, $\mathcal{C}_{\text{drag},2} < +\infty$ where

$$\mathcal{C}_{\text{drag},1} := \left(\sup_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} \right) \left(\inf_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} \right)^{-1} \quad \text{and} \quad \mathcal{C}_{\text{drag},2} := \left(\sup_{\mathbf{x} \in \Omega} \frac{\mu}{k_2(\mathbf{x})} \right) \left(\inf_{\mathbf{x} \in \Omega} \frac{\mu}{k_2(\mathbf{x})} \right)^{-1}. \quad (2.4)$$

2.2 Grouping of field variables in continuous setting

We now discuss two ways of grouping the field variables, which form the basis for the proposed composable block solvers. Under the first approach, the field variables are grouped based on the scale of the pore-network. That is, all the field variables (i.e., velocity and pressure) pertaining to the macro-pore network are placed in one group, and the field variables of the micro-pore network are placed into another. We refer to this splitting of field variables as the *scale-split* and the associated grouping takes the

following form

$$\boldsymbol{\Upsilon}_1 = \begin{Bmatrix} \mathbf{u}_1(\mathbf{x}) \\ \frac{p_1(\mathbf{x})}{\mathbf{u}_2(\mathbf{x})} \\ p_2(\mathbf{x}) \end{Bmatrix}. \quad (2.5)$$

The governing equations of the DPP model under the scale-split can be compactly written as

$$\mathcal{L}_1[\boldsymbol{\Upsilon}_1] = \mathcal{F}_1. \quad (2.6)$$

In the above equation, the differential operator takes the following form

$$\mathcal{L}_1 := \left[\begin{array}{cc|cc} \mu k_1^{-1} \mathbf{I} & \text{grad}[\cdot] & \mathbf{O} & 0 \\ \text{div}[\cdot] & \frac{\beta}{\mu} & \mathbf{O} & -\frac{\beta}{\mu} \\ \hline \mathbf{O} & 0 & \mu k_2^{-1} \mathbf{I} & \text{grad}[\cdot] \\ \mathbf{O} & -\frac{\beta}{\mu} & \text{div}[\cdot] & \frac{\beta}{\mu} \end{array} \right], \quad (2.7)$$

where \mathbf{I} denotes the identity tensor, \mathbf{O} denotes the zero tensor, and the forcing function takes the following form

$$\mathcal{F}_1 = \begin{Bmatrix} \gamma \mathbf{b}(\mathbf{x}) \\ 0 \\ \frac{\gamma \mathbf{b}(\mathbf{x})}{\gamma \mathbf{b}(\mathbf{x})} \\ 0 \end{Bmatrix}. \quad (2.8)$$

Under the second approach, the field variables are grouped based on the nature of the fields. That is, field variables of a similar kind are placed in the same group. We refer to this splitting of field variables as the *field-split* and the associated grouping takes the

following form

$$\boldsymbol{\Upsilon}_2 = \begin{Bmatrix} \mathbf{u}_1(\mathbf{x}) \\ \mathbf{u}_2(\mathbf{x}) \\ p_1(\mathbf{x}) \\ p_2(\mathbf{x}) \end{Bmatrix}. \quad (2.9)$$

The governing equations of the DPP model under the field-split can be compactly written as

$$\mathcal{L}_2[\boldsymbol{\Upsilon}_2] = \mathcal{F}_2, \quad (2.10)$$

where the differential operator takes the following form

$$\mathcal{L}_2 := \left[\begin{array}{cc|cc} \mu k_1^{-1} \mathbf{I} & \mathbf{O} & \text{grad}[\cdot] & 0 \\ \mathbf{O} & \mu k_2^{-1} \mathbf{I} & 0 & \text{grad}[\cdot] \\ \hline \text{div}[\cdot] & \mathbf{O} & \frac{\beta}{\mu} & -\frac{\beta}{\mu} \\ \mathbf{O} & \text{div}[\cdot] & -\frac{\beta}{\mu} & \frac{\beta}{\mu} \end{array} \right] \quad (2.11)$$

and the forcing function can be written as

$$\mathcal{F}_2 = \begin{Bmatrix} \gamma \mathbf{b}(\mathbf{x}) \\ \gamma \mathbf{b}(\mathbf{x}) \\ 0 \\ 0 \end{Bmatrix}. \quad (2.12)$$

2.3 Geometrical definitions

The domain is partitioned into “*Nel*e” subdomains, which will be elements in the context of the finite element method. These elements form a mesh on the domain. Mathematically, a mesh \mathcal{T} on Ω is a finite collection of disjoint polyhedra $\mathcal{T} = \{\omega^1, \dots, \omega^{Nel}\}$

such that

$$\overline{\Omega} = \bigcup_{i=1}^{N_{ele}} \overline{\omega^i}. \quad (2.13)$$

(Recall that an overline denotes the set closure.) We refer to ω^i as the i -th subdomain (element). The union of all open subdomains is denoted by

$$\tilde{\Omega} = \bigcup_{i=1}^{N_{ele}} \omega^i \quad (2.14)$$

with the understanding that an integration over $\tilde{\Omega}$ is interpreted as

$$\int_{\tilde{\Omega}} (\cdot) d\Omega = \sum_{i=1}^{N_{ele}} \int_{\omega^i} (\cdot) d\Omega. \quad (2.15)$$

The boundary of element ω^i is denoted by $\partial\omega^i := \overline{\omega^i} - \omega^i$. The set of all edges¹ in the mesh is denoted by \mathcal{E} and the set of all interior edges is denoted by \mathcal{E}^{int} . The entire boundary of the skeleton of the mesh (i.e., the union of all the interior and exterior edges) is denoted by

$$\Gamma = \bigcup_{\Upsilon \in \mathcal{E}} \Upsilon \equiv \bigcup_{i=1}^{N_{ele}} \partial\omega^i. \quad (2.16)$$

The entire interior boundary (i.e., the union of all the interior edges) is denoted by

$$\Gamma^{\text{int}} = \bigcup_{\Upsilon \in \mathcal{E}^{\text{int}}} \Upsilon \equiv \Gamma \setminus \partial\Omega. \quad (2.17)$$

Similar to the broken integral over $\tilde{\Omega}$ (i.e., equation (2.15)), the integral over Γ^{int} should be interpreted as

$$\int_{\Gamma^{\text{int}}} (\cdot) d\Gamma = \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \int_{\Upsilon} (\cdot) d\Gamma. \quad (2.18)$$

¹For simplicity, we use “edge” to refer to a node in 1D, an edge in 2D and a face in 3D. The context will be clear from the particular discussion.

2.4 Average and jump operators

Consider an interior edge $\Upsilon \in \mathcal{E}^{\text{int}}$. We denote the elements that juxtapose Υ by ω_{Υ}^{+} and ω_{Υ}^{-} . The unit normal vectors on this interior edge pointing outwards to ω_{Υ}^{+} and ω_{Υ}^{-} are, respectively, denoted by $\hat{\mathbf{n}}_{\Upsilon}^{+}$ and $\hat{\mathbf{n}}_{\Upsilon}^{-}$ (see **Fig. 2.1**). The average $\{\cdot\}$ and jump $[\![\cdot]\!]$ operators on Υ for a scalar field $\varphi(\mathbf{x})$ are, respectively, defined as

$$\{\varphi\} := \frac{1}{2} (\varphi_{\Upsilon}^{+}(\mathbf{x}) + \varphi_{\Upsilon}^{-}(\mathbf{x})) \quad \text{and} \quad [\![\varphi]\!] := \varphi_{\Upsilon}^{+}(\mathbf{x}) \hat{\mathbf{n}}_{\Upsilon}^{+}(\mathbf{x}) + \varphi_{\Upsilon}^{-}(\mathbf{x}) \hat{\mathbf{n}}_{\Upsilon}^{-}(\mathbf{x}) \quad \forall \mathbf{x} \in \Upsilon, \quad (2.19)$$

where $\varphi_{\Upsilon}^{+}(\mathbf{x})$ and $\varphi_{\Upsilon}^{-}(\mathbf{x})$ are the restrictions of $\varphi(\mathbf{x})$ onto the elements ω_{Υ}^{+} and ω_{Υ}^{-} , respectively. Mathematically,

$$\varphi_{\Upsilon}^{+}(\mathbf{x}) := \varphi(\mathbf{x})|_{\partial\omega_{\Upsilon}^{+}} \quad \text{and} \quad \varphi_{\Upsilon}^{-}(\mathbf{x}) := \varphi(\mathbf{x})|_{\partial\omega_{\Upsilon}^{-}} \quad \forall \mathbf{x} \in \Upsilon. \quad (2.20)$$

For a vector field $\boldsymbol{\tau}(\mathbf{x})$, these operators on Υ are defined as

$$\{\boldsymbol{\tau}\} := \frac{1}{2} (\boldsymbol{\tau}_{\Upsilon}^{+}(\mathbf{x}) + \boldsymbol{\tau}_{\Upsilon}^{-}(\mathbf{x})) \quad \text{and} \quad [\![\boldsymbol{\tau}]\!] := \boldsymbol{\tau}_{\Upsilon}^{+}(\mathbf{x}) \cdot \hat{\mathbf{n}}_{\Upsilon}^{+}(\mathbf{x}) + \boldsymbol{\tau}_{\Upsilon}^{-}(\mathbf{x}) \cdot \hat{\mathbf{n}}_{\Upsilon}^{-}(\mathbf{x}) \quad \forall \mathbf{x} \in \Upsilon, \quad (2.21)$$

where $\boldsymbol{\tau}_{\Upsilon}^{+}(\mathbf{x})$ and $\boldsymbol{\tau}_{\Upsilon}^{-}(\mathbf{x})$ are defined similar to equation (2.20). It is important to note that the jump operator acts on a scalar field to produce a vector field and vice-versa. It is also important to note that the above definitions are independent of the ordering of the elements. The following identity will be used in the rest of this chapter:

$$[\![\varphi\boldsymbol{\tau}]\!] = [\![\boldsymbol{\tau}]\!] \{\varphi\} + \{\boldsymbol{\tau}\} \cdot [\![\varphi]\!]. \quad (2.22)$$

2.5 Mesh-related quantities

We denote the *element diameter* (i.e., the length of the largest edge) of $\omega \in \mathcal{T}$ by h_{ω} . The maximum element diameter in a given mesh is referred to as *the mesh-size* and is

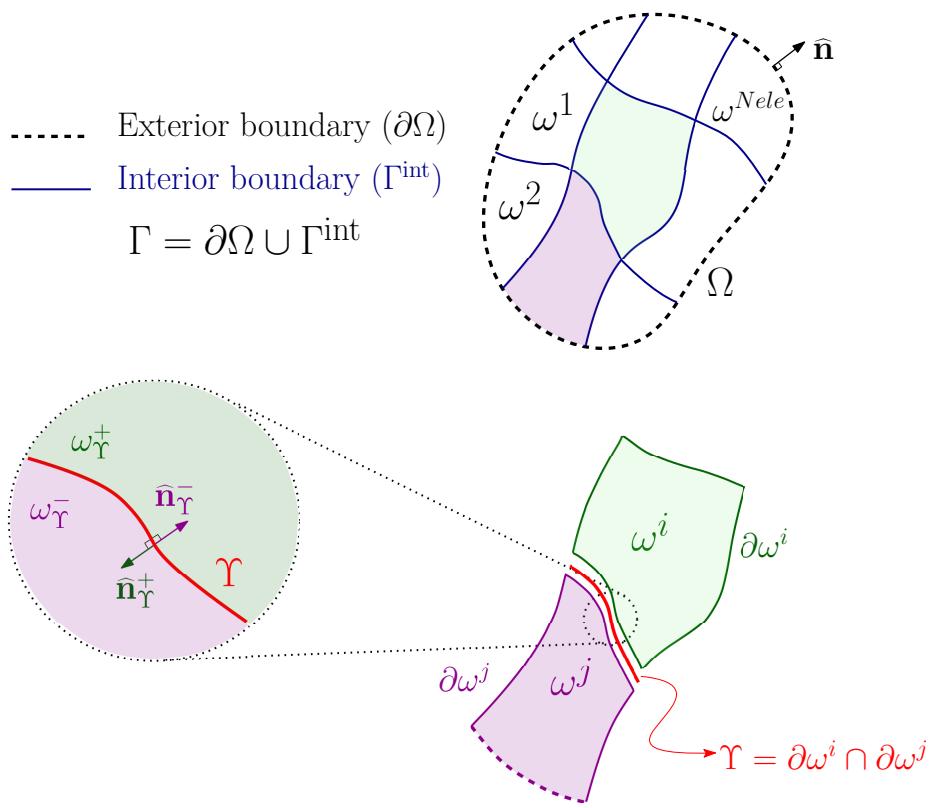


Figure 2.1: This figure shows the decomposition of the domain into subdomains. External ($\partial\Omega$) and internal (Γ^{int}) boundaries of the domain, the interface (Υ) between two adjacent elements, and normal vectors to the boundaries are shown.

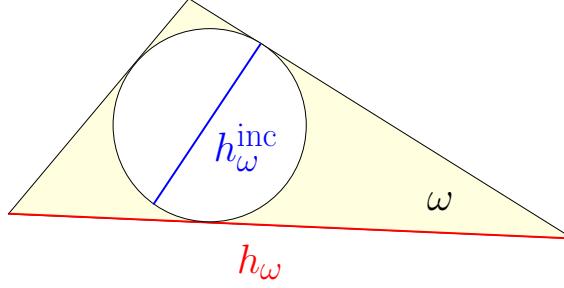


Figure 2.2: This figure illustrates the element diameter parameter h_ω and the diameter of the inscribed circle h_ω^{inc} for a typical element $\omega \in \mathcal{T}$.

denoted by

$$h := \max_{\omega \in \mathcal{T}} h_\omega. \quad (2.23)$$

We denote the *diameter of the inscribed circle* in $\omega \in \mathcal{T}$ by h_ω^{inc} (see Fig. 2.2). For an internal edge $\Upsilon \in \mathcal{E}^{\text{int}}$, shared by elements ω_Υ^+ and ω_Υ^- , we define the characteristic length h_Υ as

$$h_\Upsilon = \frac{1}{2} (h_{\omega_\Upsilon^+} + h_{\omega_\Upsilon^-}). \quad (2.24)$$

For an external edge $\Upsilon \in \mathcal{E} \setminus \mathcal{E}^{\text{int}}$, h_Υ is set to be equal to the element diameter of the element containing the edge Υ .

We place two restrictions on a mesh, and we refer to a mesh satisfying these two restrictions as an *admissible mesh*.

- (i) The mesh is *shape regular* [28], which means that there exists a constant number C_{sp} such that

$$C_{\text{sp}} h_\omega \leq h_\omega^{\text{inc}} \quad \forall \omega \in \mathcal{T}. \quad (2.25)$$

The constant C_{sp} is commonly referred to as the *shape parameter*.

- (ii) The mesh is *locally quasi-uniform*, which also goes by the name *contact regularity* [62]. This condition requires that the element diameters of any two neighboring

elements obey an equivalence relation. That is, there exists a constant number $\mathcal{C}_{\text{lqu}} > 0$ such that

$$\frac{1}{\mathcal{C}_{\text{lqu}}} h_{\omega_{\Upsilon}^+} \leq h_{\omega_{\Upsilon}^-} \leq \mathcal{C}_{\text{lqu}} h_{\omega_{\Upsilon}^+} \quad \forall \Upsilon \in \mathcal{E}^{\text{int}}. \quad (2.26)$$

The ordering of the neighboring elements (i.e., which element is “+” and which one is “−”) in the above inequality is arbitrary. This means that the above inequality holds even if ω_{Υ}^+ and ω_{Υ}^- are interchanged. The locally quasi-uniform condition implies the following useful bound:

$$\frac{1}{2} \left(1 + \frac{1}{\mathcal{C}_{\text{lqu}}} \right) h_{\omega_{\Upsilon}^+} \leq h_{\Upsilon} \leq \frac{1}{2} (1 + \mathcal{C}_{\text{lqu}}) h_{\omega_{\Upsilon}^+} \quad \forall \Upsilon \in \mathcal{E}^{\text{int}} \text{ and} \quad (2.27a)$$

$$\frac{1}{2} \left(1 + \frac{1}{\mathcal{C}_{\text{lqu}}} \right) h_{\omega_{\Upsilon}^-} \leq h_{\Upsilon} \leq \frac{1}{2} (1 + \mathcal{C}_{\text{lqu}}) h_{\omega_{\Upsilon}^-} \quad \forall \Upsilon \in \mathcal{E}^{\text{int}}. \quad (2.27b)$$

A mesh \mathcal{T} with mesh-size h will be denoted by \mathcal{T}_h . A sequence of meshes will be denoted by $\mathcal{T}_{\mathcal{H}}$, where $\mathcal{H} = (0, \bar{h})$. $\mathcal{T}_{\mathcal{H}}$ is said to be an admissible sequence of meshes if \mathcal{T}_h is admissible for every $h \in \mathcal{H}$.

Remark 2.5.1. *There are other notions of characteristic mesh sizes which are employed for DG methods. For example, an element length scale has been employed in [92], which takes the following form under our notation*

$$\hat{h} = \frac{\text{meas}(\omega_{\Upsilon}^+) + \text{meas}(\omega_{\Upsilon}^-)}{2 \text{meas}(\Upsilon)}, \quad (2.28)$$

where $\text{meas}(\cdot)$ denotes the measure of a set. A good discussion on various mesh-based characteristic lengths can be found in [62].

2.6 Functional analysis aspects

We denote the standard L_2 inner-product over a set \mathcal{K} by $(\cdot; \cdot)_{\mathcal{K}}$. That is,

$$(a; b)_{\mathcal{K}} := \int_{\mathcal{K}} a \cdot b \, d\mathcal{K} \quad (2.29)$$

and the associated standard L_2 norm is denoted by $\|\cdot\|_{\mathcal{K}}$ as

$$\|a\|_{\mathcal{K}} = \sqrt{(a; a)_{\mathcal{K}}}. \quad (2.30)$$

The subscript in the L_2 inner-product and the associated norm will be dropped if $\mathcal{K} = \tilde{\Omega}$.

In a subsequent section on the interpolation error, we employ a general order Sobolev semi-norm. To this end, let $\alpha = (\alpha_1, \dots, \alpha_{nd}) \in \mathbb{N}^{nd}$ be a nd -tuple (i.e., multi-index), the order of which is denoted by $|\alpha| := \sum_{i=1}^{nd} \alpha_i$. We denote the multi-index (classical or distributional) partial derivative by $D^\alpha(\cdot)$. For a scalar function $\varphi(\mathbf{x}) \in C_c^\infty(\mathcal{K})$ (which is a set of infinitely differentiable functions with compact support in \mathcal{K}) [67], the multi-index (classical) partial derivative with respect to a given coordinate system $\mathbf{x} = (x_1, \dots, x_{nd})$ is defined as

$$D^\alpha \varphi(\mathbf{x}) := \frac{\partial^{|\alpha|} \varphi(\mathbf{x})}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_{nd}^{\alpha_{nd}}}. \quad (2.31)$$

Then, the multi-index distributional partial derivative of a scalar field $a : \mathcal{K} \rightarrow \mathbb{R}$ is defined as

$$(D^\alpha a(\mathbf{x}); \varphi(\mathbf{x}))_{\mathcal{K}} := (-1)^{|\alpha|} (a(\mathbf{x}); D^\alpha \varphi(\mathbf{x}))_{\mathcal{K}} \quad \forall \varphi(\mathbf{x}) \in C_c^\infty(\mathcal{K}). \quad (2.32)$$

For a scalar field $a : \mathcal{K} \rightarrow \mathbb{R}$, the s -th order Sobolev semi-norm over \mathcal{K} is defined as

$$|a|_{H^s(\mathcal{K})} := \left(\sum_{|\alpha|=s} \|D^\alpha a(\mathbf{x})\|_{\mathcal{K}}^2 \right)^{1/2} \quad (2.33)$$

and for a vector field $\mathbf{a} : \mathcal{K} \rightarrow \mathbb{R}^{nd}$ with scalar components a_i ($i = 1, \dots, nd$), the corresponding semi-norm is defined as

$$|\mathbf{a}|_{H^s(\mathcal{K})} := \left(\sum_{i=1}^{nd} |a_i|_{H^s(\mathcal{K})}^2 \right)^{1/2}. \quad (2.34)$$

2.6.1 Inverse and trace inequalities²

The inequalities given below play a crucial role in obtaining bounds on the error due to terms defined on the element interface. Mathematical proofs to these estimates can be found in [173, 9, 62, 145].

Lemma 2.6.1. (*Continuous trace inequality.*) *For an admissible mesh \mathcal{T}_h , the following estimates hold $\forall \omega \in \mathcal{T}_h$:*

$$\|v\|_{\partial\omega} \leq C_{\text{trace}} \left(\frac{1}{\sqrt{h_\omega}} \|v\|_\omega + \sqrt{h_\omega} \|\text{grad}[v]\|_\omega \right) \quad \forall v(\mathbf{x}) \in H^1(\omega) \text{ and} \quad (2.35)$$

$$\|\mathbf{v}\|_{\partial\omega} \leq C_{\text{trace}} \left(\frac{1}{\sqrt{h_\omega}} \|\mathbf{v}\|_\omega + \sqrt{h_\omega} \|\text{grad}[\mathbf{v}]\|_\omega \right) \quad \forall \mathbf{v}(\mathbf{x}) \in (H^1(\omega))^{nd}, \quad (2.36)$$

where the C_{trace} depends on the shape parameter (i.e., C_{sp}) and the number of spatial dimensions (nd) but it is not dependent on h_ω .

Let $\mathcal{P}^m(\omega)$ denote the set of all polynomials up to and including m -th order over $\omega \in \mathcal{T}_h$. We then have the following discrete inequalities.

Lemma 2.6.2. (*Discrete inverse inequality.*) *Let \mathcal{T}_h be an admissible mesh. Then the following estimates hold $\forall \omega \in \mathcal{T}_h$:*

$$\|\text{grad}[v^h]\|_\omega \leq C_{\text{inv}} h_\omega^{-1} \|v^h\|_\omega \quad \forall v^h(\mathbf{x}) \in H^1(\omega) \cap \mathcal{P}^m(\omega) \text{ and} \quad (2.37)$$

$$\|\text{grad}[\mathbf{v}^h]\|_\omega \leq C_{\text{inv}} h_\omega^{-1} \|\mathbf{v}^h\|_\omega \quad \forall \mathbf{v}^h(\mathbf{x}) \in (H^1(\omega))^{nd} \cap (\mathcal{P}^m(\omega))^{nd}, \quad (2.38)$$

²For these results we assume that the velocity fields belong to $(H^1(\omega))^{nd}$ instead of $H(\text{div}, \omega)$, which was the case in the function space (3.8a). The reason is that one has to deal with half-Sobolev spaces and corresponding dual spaces (i.e., negative half-spaces) for trace inequalities under $H(\text{div})$; which makes the convergence and error analyses more involved. Moreover, the authors are not aware of any discrete trace inequalities available in the mathematical analysis literature that can be easily used under half-Sobolev spaces.

where \mathcal{C}_{inv} is a constant dependent on the shape parameter (\mathcal{C}_{sp}), the number of spatial dimensions (nd) and the polynomial order (m), but it does not depend on h_ω or on the fields $v^h(\mathbf{x})$ and $\mathbf{v}^h(\mathbf{x})$.

Lemma 2.6.3. *(Discrete trace inequality.) For an admissible mesh \mathcal{T}_h , the following estimates hold $\forall \omega \in \mathcal{T}_h$:*

$$\|v^h\|_{\partial\omega} \leq \mathcal{C}_{\text{trace}} (1 + \mathcal{C}_{\text{inv}}) \frac{1}{\sqrt{h_\omega}} \|v^h\|_\omega \quad \forall v^h(\mathbf{x}) \in H^1(\omega) \cap \mathscr{P}^m(\omega) \text{ and} \quad (2.39)$$

$$\|\mathbf{v}^h\|_{\partial\omega} \leq \mathcal{C}_{\text{trace}} (1 + \mathcal{C}_{\text{inv}}) \frac{1}{\sqrt{h_\omega}} \|\mathbf{v}^h\|_\omega \quad \forall \mathbf{v}^h(\mathbf{x}) \in (H^1(\omega))^{nd} \cap (\mathscr{P}^m(\omega))^{nd}. \quad (2.40)$$

Chapter 3

Classical and Stabilized Mixed Weak Formulations

Except for some academic problems, it is not possible to obtain analytical solutions for the governing equations under the DPP model. Hence, there is a need to resort to numerical solutions. A continuous stabilized mixed formulation proposed recently by [96] for the DPP mathematical model. However, continuous Galerkin (CG) based formulations suffer from the so-called Gibbs phenomenon when applied to problems with highly heterogeneous medium properties such as layered media; which manifests in the form of spurious oscillations (overshoots and undershoots) at the interface of a sharp change in medium properties (e.g., permeability). [92] have clearly demonstrated that conventional continuous finite element methods for *Darcy equations* fall short in accurately capturing jumps in the solution fields at the location of material discontinuities. Since disparate medium properties are frequently encountered in subsurface modeling, the stabilized mixed four-field CG formulation recently proposed for DPP model [96] will not be able to accurately capture the velocity profiles in highly heterogeneous porous media and will not suffice for realistic subsurface modeling. This will be clearly demonstrated using numerical simulations later in this document. We, therefore, develop a stabilized mixed discontinuous Galerkin (DG) formulation for the DPP model, which is robust, stable and capable of capturing possible jumps in the solution fields due to the existing disparate

medium properties.

It is important to mention that one can also capture disparate medium properties and satisfy the LBB *inf-sup* stability condition [34] by employing an element from the $H(\text{div})$ family; which include Raviart-Thomas spaces [149], Nédélec spaces [138] Brezzi-Douglas-Marini (BDM) spaces [32], Brezzi-Douglas-Fortin-Marini (BDFM) spaces [30] and Crouzeix-Raviart spaces [58]. Although there is an on-going debate on using $H(\text{div})$ elements vs. DG methods, the latter do enjoy some unique desirable features. DG methods combine the attractive features of both finite element and finite volume methods. Application of completely discontinuous basis functions in the form of piecewise polynomials in DG methods provides them with the flexibility to support common non-conforming spaces (e.g., non-matching grids and hanging nodes, $h\text{-}p$ adaptivity, variable degrees of local interpolations) and handle jumps in the profiles of variables [154, 52, 111, 112]. DG methods also enjoy high parallel efficiency. Unlike the conventional continuous formulations, they are known to exhibit better local (or element-wise) mass balance [92, 153].

The origins of DG methods can be traced back to [113] and [139]. One of the first successful applications of DG formulation to solve a practical problem was by [151], which addressed neutron transport. Over the years, DG methods have been successfully employed to solve hyperbolic PDEs [37, 142], elliptic PDEs [63, 155, 156, 10, 22, 54], parabolic PDEs [63, 107], coupling algorithms [137] and space-time finite elements [143, 2]. Several variants of DG formulations have been developed over the years with varying merits for each variant. Some popular variants are Runge-Kutta DG [56], local DG [44], embedded DG [78], compact DG [144], hybridizable DG [53] and adjoint-type variational multiscale DG [92, 17]. Although these variants may look very different, a unified framework has been laid out by [10], to derive DG methods systematically, and these methods differ in their choices of numerical fluxes. However, to the best of authors' knowledge, there is no

clear cut winner among these variants.

Under a mixed formulation, velocities and pressures are taken to be the primary variables. However, for numerical stability, a mixed formulation should either satisfy or circumvent the Ladyzhenskaya-Babuška-Brezzi (LBB) *inf-sup* stability condition [35]. This naturally places all the mixed formulations into either of two categories. A mixed formulation in the first category is built on the classical mixed formulation (which is based on the Galerkin formalism) but places restrictions on the interpolation functions for the independent field variables to satisfy the LBB condition. To put it differently, not all combinations of interpolation functions for the field variables satisfy the LBB condition under the classical mixed formulation. A mixed formulation in the second category augments the classical mixed formulation with stabilization terms so as to circumvent the LBB condition and to render a stable formulation. In this chapter, we consider one mixed formulation from the first category and two from the second category.

Let us define the following function spaces for the velocities and pressures fields as follows:

$$\mathcal{U}_1 := \left\{ \mathbf{u}_1(\mathbf{x}) \in (L_2(\Omega))^{nd} \mid \operatorname{div}[\mathbf{u}_1] \in L_2(\Omega), \mathbf{u}_1(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}) = u_{n1}(\mathbf{x}) \in H^{-1/2}(\Gamma_1^u) \right\}, \quad (3.1a)$$

$$\mathcal{U}_2 := \left\{ \mathbf{u}_2(\mathbf{x}) \in (L_2(\Omega))^{nd} \mid \operatorname{div}[\mathbf{u}_2] \in L_2(\Omega), \mathbf{u}_2(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}) = u_{n2}(\mathbf{x}) \in H^{-1/2}(\Gamma_2^u) \right\}, \quad (3.1b)$$

$$\mathcal{W}_1 := \left\{ \mathbf{w}_1(\mathbf{x}) \in (L_2(\Omega))^{nd} \mid \operatorname{div}[\mathbf{w}_1] \in L_2(\Omega), \mathbf{w}_1(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}) = 0 \text{ on } \Gamma_1^u \right\}, \quad (3.1c)$$

$$\mathcal{W}_2 := \left\{ \mathbf{w}_2(\mathbf{x}) \in (L_2(\Omega))^{nd} \mid \operatorname{div}[\mathbf{w}_2] \in L_2(\Omega), \mathbf{w}_2(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}) = 0 \text{ on } \Gamma_2^u \right\}, \quad (3.1d)$$

$$\mathcal{P} := \left\{ (p_1(\mathbf{x}), p_2(\mathbf{x})) \in L_2(\Omega) \times L_2(\Omega) \mid \left(\int_{\Omega} p_1(\mathbf{x}) d\Omega \right) \left(\int_{\Omega} p_2(\mathbf{x}) d\Omega \right) = 0 \right\}, \text{ and} \quad (3.1e)$$

$$\mathcal{Q} := \left\{ (p_1(\mathbf{x}), p_2(\mathbf{x})) \in H^1(\Omega) \times H^1(\Omega) \mid \left(\int_{\Omega} p_1(\mathbf{x}) d\Omega \right) \left(\int_{\Omega} p_2(\mathbf{x}) d\Omega \right) = 0 \right\}, \quad (3.1f)$$

where “ nd ” denotes the number of spatial dimensions, $H^1(\Omega)$ is a standard Sobolev space, and $H^{-1/2}(\cdot)$ is the dual space corresponding to $H^{1/2}(\cdot)$. Rigorous discussion of Sobolev spaces are accessible in [106]; and further discussion of function spaces are provided by [35].

3.1 Classical mixed formulation using $\mathbf{H}(\text{div})$ elements

The classical mixed formulation can be written as: Find $(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x})) \in \mathcal{U}_1 \times \mathcal{U}_2$ and $(p_1(\mathbf{x}), p_2(\mathbf{x})) \in \mathcal{P}$ such that we have

$$\begin{aligned} \mathcal{B}_{\text{Gal}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2; \mathbf{u}_1, \mathbf{u}_2, p_1, p_2) &= \mathcal{L}_{\text{Gal}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2) \\ \forall (\mathbf{w}_1(\mathbf{x}), \mathbf{w}_2(\mathbf{x})) \in \mathcal{W}_1 \times \mathcal{W}_2, \quad (q_1(\mathbf{x}), q_2(\mathbf{x})) \in \mathcal{P}, \end{aligned} \quad (3.2)$$

where the bilinear form and the linear functional are, respectively, defined as

$$\begin{aligned} \mathcal{B}_{\text{Gal}} := & (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1) - (\text{div}[\mathbf{w}_1]; p_1) + (q_1; \text{div}[\mathbf{u}_1]) + (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2) \\ & - (\text{div}[\mathbf{w}_2]; p_2) + (q_2; \text{div}[\mathbf{u}_2]) + (q_1 - q_2; \beta/\mu(p_1 - p_2)) \text{ and} \end{aligned} \quad (3.3a)$$

$$\mathcal{L}_{\text{Gal}} := (\mathbf{w}_1; \gamma \mathbf{b}) - (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_{01})_{\Gamma_1^p} + (\mathbf{w}_2; \gamma \mathbf{b}) - (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_{02})_{\Gamma_2^p}. \quad (3.3b)$$

3.1.1 $\mathbf{H}(\text{div})$ elements

Classes of $\mathbf{H}(\text{div})$ finite element discretizations such as Raviart-Thomas (RT) [150], generalized RTN [138], BDM [33], and BDFM [31] have been shown to satisfy the LBB condition. Moreover, these finite element discretizations satisfy element-wise mass balance property [35].

The classical mixed formulation based on discretizations from the lowest-order Raviart-Thomas spaces is commonly referred to as the RT0 formulation; which is frequently used

in subsurface modeling [49]. The unknowns under the RT0 formulation on a triangle are fluxes at the midpoints of edges of the element and element-wise constant pressures. The finite dimensional subspaces for each velocity and pressure fields under the lowest-order Raviart-Thomas discretization on a triangle, which are collectively denoted by $\text{RTF}_1 \oplus \text{DP}_0$, take the following form:

$$\mathcal{U}^h := \{\mathbf{u} = (u, v) \mid u_K = a_K + b_K x, v_K = c_K + b_K y; a_K, b_K, c_K \in \mathbb{R}; K \in \mathcal{T}_h\} \text{ and} \quad (3.4a)$$

$$\mathcal{P}^h := \{p \mid p = \text{constant on each triangle } K \in \mathcal{T}_h\}, \quad (3.4b)$$

where \mathcal{T}_h is a triangulation on Ω . These subspaces on a tetrahedron, which are denoted by $\text{N1F}_1 \oplus \text{DP}_0$, take the following form:

$$\begin{aligned} \mathcal{U}^h := & \{\mathbf{u} = (u, v, w) \mid u_K = a_K + b_K x, v_K = c_K + b_K y, w_K = d_K + b_K z; \\ & a_K, b_K, c_K, d_K \in \mathbb{R}; K \in \mathcal{T}_h\} \text{ and} \end{aligned} \quad (3.5a)$$

$$\mathcal{P}^h := \{p \mid p = \text{constant on each tetrahedron } K \in \mathcal{T}_h\}, \quad (3.5b)$$

where \mathcal{T}_h , in this case, is a tetrahedralization on Ω .

In addition to $\text{H}(\text{div})$ discretizations on simplicial meshes, we also consider the corresponding discretizations on non-simplicial element – QUAD and HEX. The velocity spaces for QUAD and HEX elements are, respectively, RCTF_1 and NCF_1 [128, 13]. The (macro- and micro-) pressures are element-wise constants, and DG_0 is commonly used to denote element-wise constant discretization on non-simplicial elements. See Figure 3.1 and Table 3.1 for a description of these discretizations. The finite dimensional subspaces for the RCTF_1 and NCF_1 discretizations can be written precisely using the language of finite element exterior calculus. But such a description needs introduction of additional jargon and notation, which is beyond the scope of this document. We, therefore, refer the reader to [12, 11, 13]. However, to guide the reader, the degrees-of-freedom for these

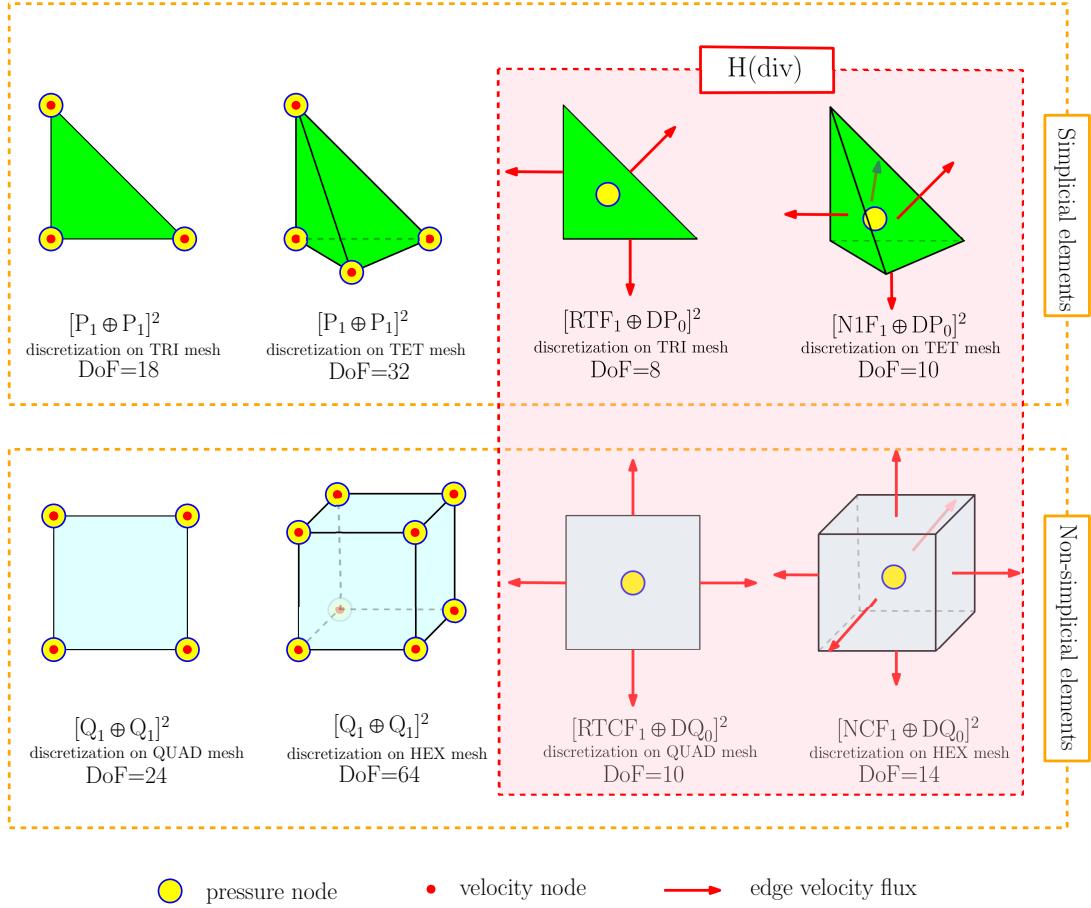


Figure 3.1: This figure shows the two-dimensional and three-dimensional elements that are employed in this chapter. The degrees-of-freedom (DoF) for each element are also indicated.

discretizations are shown in **Fig. 3.1** and Table 3.1.

3.2 Stabilized mixed continuous Galerkin formulation (CG-VMS)

The weak form of the CG-VMS formulation can be written as: Find $(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x})) \in \mathcal{U}_1 \times \mathcal{U}_2$ and $(p_1(\mathbf{x}), p_2(\mathbf{x})) \in \mathcal{Q}$ such that we have

$$\mathcal{B}_{\text{stab}}^{\text{CG}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2; \mathbf{u}_1, \mathbf{u}_2, p_1, p_2) = \mathcal{L}_{\text{stab}}^{\text{CG}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2)$$

Table 3.1: The element-level discretization for different mesh types and the chosen three formulations. \oplus denotes the direct sum operator between two finite element spaces.

Mesh type	Finite element formulation	
	H(div)	CG-VMS/DG-VMS
TRI	$[\text{RTF}_1 \oplus \text{DP}_0]^2$	$[\text{P}_1 \oplus \text{P}_1]^2$
QUAD	$[\text{RTCF}_1 \oplus \text{DQ}_0]^2$	$[\text{Q}_1 \oplus \text{Q}_1]^2$
TET	$[\text{N1F}_1 \oplus \text{DP}_0]^2$	$[\text{P}_1 \oplus \text{P}_1]^2$
HEX	$[\text{NCF}_1 \oplus \text{DQ}_0]^2$	$[\text{Q}_1 \oplus \text{Q}_1]^2$

$$\forall (\mathbf{w}_1(\mathbf{x}), \mathbf{w}_2(\mathbf{x})) \in \mathcal{W}_1 \times \mathcal{W}_2, (q_1(\mathbf{x}), q_2(\mathbf{x})) \in \mathcal{Q}, \quad (3.6)$$

where the bilinear form and the linear functional are defined, respectively, as

$$\begin{aligned} \mathcal{B}_{\text{stab}}^{\text{CG}} &:= \mathcal{B}_{\text{Gal}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2; \mathbf{u}_1, \mathbf{u}_2, p_1, p_2) \\ &\quad - \frac{1}{2} \left(\mu k_1^{-1} \mathbf{w}_1 - \text{grad}[q_1]; \frac{1}{\mu} k_1 (\mu k_1^{-1} \mathbf{u}_1 + \text{grad}[p_1]) \right) \\ &\quad - \frac{1}{2} \left(\mu k_2^{-1} \mathbf{w}_2 - \text{grad}[q_2]; \frac{1}{\mu} k_2 (\mu k_2^{-1} \mathbf{u}_2 + \text{grad}[p_2]) \right) \text{ and} \end{aligned} \quad (3.7a)$$

$$\begin{aligned} \mathcal{L}_{\text{stab}}^{\text{CG}} &:= \mathcal{L}_{\text{Gal}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2) - \frac{1}{2} \left(\mu k_1^{-1} \mathbf{w}_1 - \text{grad}[q_1]; \frac{1}{\mu} k_1 \gamma \mathbf{b} \right) \\ &\quad - \frac{1}{2} \left(\mu k_2^{-1} \mathbf{w}_2 - \text{grad}[q_2]; \frac{1}{\mu} k_2 \gamma \mathbf{b} \right). \end{aligned} \quad (3.7b)$$

An attractive feature of the CG-VMS formulation is that nodal-based equal-order interpolation for all the field variables (micro- and macro- velocities and pressures) is stable, which is not the case with the classical mixed formulation. The stability is achieved by the addition of stabilization terms, which circumvent the LBB condition.

3.3 Stabilized mixed DG formulation (DG-VMS)

We employ the adjoint-type variational multiscale approach to develop a stabilized mixed four-field DG formulation for the DPP model. In order to circumvent the LBB

inf-sup stability condition we add residual-based, adjoint-type stabilization terms defined over the elements. In order to avoid Gibbs phenomenon and at the same time maintain stability, we choose appropriate and consistent numerical fluxes, which are in the form of jumps and averages of the medium properties and solution fields. The resulting stabilized mixed DG formulation enjoys several attractive features, which include: (i) The formulation is capable of eliminating the spurious numerical instabilities in the profiles of solutions and capturing the existing jumps in the material properties. (ii) Equal-order interpolation, which is convenient for computer implementation as simple underlying data structures are needed, is stable for all the field variables. (iii) The formulation is mathematically shown to be consistent, stable, and hence convergent. (iv) *A priori* error estimation is systematically obtained. (v) The DG formulation exhibits improved element-wise mass balance compared to its continuous counterpart. (vi) The formulation can be utilized to capture physical instabilities in heterogeneous porous media and to eliminate numerical instabilities at the same time.

Formulations under the discontinuous Galerkin (DG) method inherit attractive features of both finite element and finite volume methods by allowing discontinuous basis functions (e.g., in the form of piecewise polynomials) [82]. The DG method supports non-matching grids and hanging nodes, and hence ideal for *hp* adaptivity [55]. Moreover, the method can naturally handle jumps in the profiles of the solution variables [93, 98].

We propose a stabilized *four-field* formulation for the DPP model. The proposed formulation draws its inspiration from the stabilized *two-field* formulations proposed by [92, 17] for Darcy equations, which describe the flow of an incompressible fluid through a porous medium with a single pore-network.

We introduce the following broken Sobolev spaces (which are piece-wise discontinuous

spaces):

$$\mathcal{U}^{\text{dg}} := \left\{ \mathbf{u}(\mathbf{x}) \mid \mathbf{u}(\mathbf{x})|_{\omega^i} \in (L_2(\omega^i))^{nd}; \operatorname{div}[\mathbf{u}] \in L_2(\omega^i); i = 1, \dots, Nele \right\}, \quad (3.8a)$$

$$\tilde{\mathcal{P}}^{\text{dg}} := \left\{ p(\mathbf{x}) \mid p(\mathbf{x})|_{\omega^i} \in L_2(\omega^i); i = 1, \dots, Nele \right\}, \quad (3.8b)$$

$$\tilde{\mathcal{Q}}^{\text{dg}} := \left\{ p(\mathbf{x}) \mid p(\mathbf{x})|_{\omega^i} \in H^1(\omega^i); i = 1, \dots, Nele \right\}, \quad (3.8c)$$

$$\mathcal{P}^{\text{dg}} := \left\{ (p_1(\mathbf{x}), p_2(\mathbf{x})) \in \tilde{\mathcal{P}} \times \tilde{\mathcal{P}} \mid \left(\int_{\tilde{\Omega}} p_1(\mathbf{x}) d\Omega \right) \left(\int_{\tilde{\Omega}} p_2(\mathbf{x}) d\Omega \right) = 0 \right\}, \text{ and} \quad (3.8d)$$

$$\mathcal{Q}^{\text{dg}} := \left\{ (p_1(\mathbf{x}), p_2(\mathbf{x})) \in \tilde{\mathcal{Q}} \times \tilde{\mathcal{Q}} \mid \left(\int_{\tilde{\Omega}} p_1(\mathbf{x}) d\Omega \right) \left(\int_{\tilde{\Omega}} p_2(\mathbf{x}) d\Omega \right) = 0 \right\}, \quad (3.8e)$$

where $L_2(\omega^i)$ denotes the set of all square-integrable functions defined on ω^i , and $H^1(\omega^i)$ is a standard Sobolev space [67].

Remark 3.3.1. *The following condition in \mathcal{P}^{dg} and \mathcal{Q}^{dg} spaces (which is expressed in terms of the mean pressures in the two pore-networks):*

$$\left(\int_{\tilde{\Omega}} p_1(\mathbf{x}) d\Omega \right) \left(\int_{\tilde{\Omega}} p_2(\mathbf{x}) d\Omega \right) = 0$$

is one of the ways to fix the datum for the pressure. However, this condition is seldom employed in a numerical implementation. Alternatively, one can prescribe the pressure on a portion of the boundary in one of the pore-networks. For further details refer to [96].

3.3.1 Weak form in terms of numerical fluxes

Multiplying the governing equations (2.1a)–(2.1d) by weighting functions, integrating over an element ω , and using equation (2.1e) and the divergence theorem, we obtain the following:

$$\begin{aligned} & (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1)_\omega - (\operatorname{div}[\mathbf{w}_1]; p_1)_\omega + \left(\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_1^* \right)_{\partial\omega} + (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2)_\omega - (\operatorname{div}[\mathbf{w}_2]; p_2)_\omega \\ & + \left(\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_2^* \right)_{\partial\omega} + (q_1; \operatorname{div}[\mathbf{u}_1])_\omega + \left(q_1; \left(\mathbf{u}_1^* - \mathbf{u}_1 \right) \cdot \hat{\mathbf{n}} \right)_{\partial\omega} + (q_2; \operatorname{div}[\mathbf{u}_2])_\omega \end{aligned}$$

$$+ \left(q_2; \left(\mathbf{\hat{u}}_2 - \mathbf{u}_2 \right) \cdot \widehat{\mathbf{n}} \right)_{\partial\omega} + \left(q_1 - q_2; \frac{\beta}{\mu} (p_1 - p_2) \right)_{\omega} = (\mathbf{w}_1; \gamma \mathbf{b}_1)_{\omega} + (\mathbf{w}_2; \gamma \mathbf{b}_2)_{\omega}, \quad (3.9)$$

where \hat{p}_1 and \hat{p}_2 are the numerical fluxes for the pressures and $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ are the numerical fluxes for the velocities. Summing the above equation over all the elements and using the identity (2.22), we obtain the following weak form in terms of numerical fluxes:

$$\begin{aligned} & (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1) - (\operatorname{div}[\mathbf{w}_1]; p_1) + \left(\{\!\{ \mathbf{w}_1 \}\!\}; [\![\hat{p}_1]\!] \right)_{\Gamma^{\text{int}}} + \left([\![\mathbf{w}_1]\!]; \{\!\{ \hat{p}_1 \}\!\} \right)_{\Gamma^{\text{int}}} + \left(\mathbf{w}_1 \cdot \widehat{\mathbf{n}}; \hat{p}_1 \right)_{\partial\Omega} \\ & + (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2) - (\operatorname{div}[\mathbf{w}_2]; p_2) + \left(\{\!\{ \mathbf{w}_2 \}\!\}; [\![\hat{p}_2]\!] \right)_{\Gamma^{\text{int}}} + \left([\![\mathbf{w}_2]\!]; \{\!\{ \hat{p}_2 \}\!\} \right)_{\Gamma^{\text{int}}} + \left(\mathbf{w}_2 \cdot \widehat{\mathbf{n}}; \hat{p}_2 \right)_{\partial\Omega} \\ & + (q_1; \operatorname{div}[\mathbf{u}_1]) + \left(\{\!\{ q_1 \}\!\}; [\![\hat{\mathbf{u}}_1]\!] - [\![\mathbf{u}_1]\!] \right)_{\Gamma^{\text{int}}} + \left([\![q_1]\!]; \{\!\{ \hat{\mathbf{u}}_1 \}\!} - \{\!\{ \mathbf{u}_1 \}\!} \right)_{\Gamma^{\text{int}}} + \left(q_1; (\hat{\mathbf{u}}_1 - \mathbf{u}_1) \cdot \widehat{\mathbf{n}} \right)_{\partial\Omega} \\ & + (q_2; \operatorname{div}[\mathbf{u}_2]) + \left(\{\!\{ q_2 \}\!\}; [\![\hat{\mathbf{u}}_2]\!] - [\![\mathbf{u}_2]\!] \right)_{\Gamma^{\text{int}}} + \left([\![q_2]\!]; \{\!\{ \hat{\mathbf{u}}_2 \}\!} - \{\!\{ \mathbf{u}_2 \}\!} \right)_{\Gamma^{\text{int}}} + \left(q_2; (\hat{\mathbf{u}}_2 - \mathbf{u}_2) \cdot \widehat{\mathbf{n}} \right)_{\partial\Omega} \\ & + \left(q_1 - q_2; \frac{\beta}{\mu} (p_1 - p_2) \right) = (\mathbf{w}_1; \gamma \mathbf{b}_1) + (\mathbf{w}_2; \gamma \mathbf{b}_2). \end{aligned} \quad (3.10)$$

Physically, the jumps in pressures and the normal component of velocities should vanish on any curve which is entirely inside the domain, and in particular, on any interior edge. That is,

$$[\![p_1]\!] = \mathbf{0}, \quad [\![p_2]\!] = \mathbf{0}, \quad [\![\mathbf{u}_1]\!] = 0 \text{ and } [\![\mathbf{u}_2]\!] = 0 \quad \text{on } \Gamma^{\text{int}}. \quad (3.11)$$

Numerical fluxes are important components of DG methods, which have to be selected carefully. The choice of these numerical fluxes can greatly affect the stability of a DG formulation. Herein, we consider the following general expressions for the numerical fluxes:

$$\hat{p}_1 = \begin{cases} \lambda_1^{(1)} \{\!\{ p_1 \}\!} + \frac{\lambda_1^{(2)}}{2} [\![p_1]\!] \cdot \widehat{\mathbf{n}} + \lambda_1^{(3)} [\![\mathbf{u}_1]\!] & \text{on } \Gamma^{\text{int}} \\ p_1 & \text{on } \Gamma_1^u \\ p_{01} & \text{on } \Gamma_1^p, \end{cases} \quad (3.12)$$

$$\hat{p}_2 = \begin{cases} \lambda_2^{(1)} \{\!\{ p_2 \}\!} + \frac{\lambda_2^{(2)}}{2} [\![p_2]\!] \cdot \widehat{\mathbf{n}} + \lambda_2^{(3)} [\![\mathbf{u}_2]\!] & \text{on } \Gamma^{\text{int}} \\ p_2 & \text{on } \Gamma_2^u \\ p_{02} & \text{on } \Gamma_2^p, \end{cases} \quad (3.13)$$

$$\dot{\mathbf{u}}_1 = \Lambda_1^{(1)} \{\!\{ \mathbf{u}_1 \}\!\} + \frac{\Lambda_1^{(2)}}{2} [\![\mathbf{u}_1]\!] \hat{\mathbf{n}} + \Lambda_1^{(3)} [\![p_1]\!] \quad \text{on } \Gamma^{\text{int}}, \quad (3.14a)$$

$$\dot{\mathbf{u}}_1 \cdot \hat{\mathbf{n}} = \begin{cases} u_{n1} & \text{on } \Gamma_1^u \\ \mathbf{u}_1 \cdot \hat{\mathbf{n}} & \text{on } \Gamma_1^p, \end{cases} \quad (3.14b)$$

$$\dot{\mathbf{u}}_2 = \Lambda_2^{(1)} \{\!\{ \mathbf{u}_2 \}\!\} + \frac{\Lambda_2^{(2)}}{2} [\![\mathbf{u}_2]\!] \hat{\mathbf{n}} + \Lambda_2^{(3)} [\![p_2]\!] \quad \text{on } \Gamma^{\text{int}}, \text{ and} \quad (3.15a)$$

$$\dot{\mathbf{u}}_2 \cdot \hat{\mathbf{n}} = \begin{cases} u_{n2} & \text{on } \Gamma_2^u \\ \mathbf{u}_2 \cdot \hat{\mathbf{n}} & \text{on } \Gamma_2^p, \end{cases} \quad (3.15b)$$

where $\lambda_i^{(j)}$ and $\Lambda_i^{(j)}$ ($i, j = 1$ or 2) are constants. It is easy to check that these numerical fluxes satisfy the following relations on Γ^{int} :

$$\{\!\{ p_1^* \}\!\} = \lambda_1^{(1)} \{\!\{ p_1 \}\!\} + \lambda_1^{(3)} [\![p_1]\!], \quad \text{and} \quad [\![p_1^*]\!] = \lambda_1^{(2)} [\![p_1]\!], \quad (3.16a)$$

$$\{\!\{ p_2^* \}\!\} = \lambda_2^{(1)} \{\!\{ p_2 \}\!\} + \lambda_2^{(3)} [\![p_2]\!], \quad \text{and} \quad [\![p_2^*]\!] = \lambda_2^{(2)} [\![p_2]\!], \quad (3.16b)$$

$$\{\!\{ \mathbf{u}_1^* \}\!\} = \Lambda_1^{(1)} \{\!\{ \mathbf{u}_1 \}\!\} + \Lambda_1^{(3)} [\![p_1]\!], \quad \text{and} \quad [\![\mathbf{u}_1^*]\!] = \Lambda_1^{(2)} [\![\mathbf{u}_1]\!], \text{ and} \quad (3.16c)$$

$$\{\!\{ \mathbf{u}_2^* \}\!\} = \Lambda_2^{(1)} \{\!\{ \mathbf{u}_2 \}\!\} + \Lambda_2^{(3)} [\![p_2]\!], \quad \text{and} \quad [\![\mathbf{u}_2^*]\!] = \Lambda_2^{(2)} [\![\mathbf{u}_2]\!]. \quad (3.16d)$$

3.3.2 The classical mixed DG formulation

This formulation is based on the Galerkin formalism and can be obtained by making the following choices:

$$\lambda_1^{(1)} = \lambda_2^{(1)} = \Lambda_1^{(1)} = \Lambda_2^{(1)} = 1 \quad (3.17)$$

and the other constants in equations (3.12)–(3.15) are taken to be zeros. The numerical fluxes on Γ^{int} under the classical mixed DG formulation take the following form:

$$p_1^* = \{\!\{ p_1 \}\!\}, \quad p_2^* = \{\!\{ p_2 \}\!\}, \quad \mathbf{u}_1^* = \{\!\{ \mathbf{u}_1 \}\!\} \text{ and } \mathbf{u}_2^* = \{\!\{ \mathbf{u}_2 \}\!\}. \quad (3.18)$$

The above numerical fluxes are similar to the ones employed by [23], which are known to be consistent but do not result in a stable DG method [10]. The corresponding weak formulation reads: Find $(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x})) \in \mathcal{U}^{\text{dg}} \times \mathcal{U}^{\text{dg}}$, $(p_1(\mathbf{x}), p_2(\mathbf{x})) \in \mathcal{P}^{\text{dg}}$ such that we have

$$\begin{aligned} \mathcal{B}_{\text{Gal}}^{\text{DG}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2; \mathbf{u}_1, \mathbf{u}_2, p_1, p_2) &= \mathcal{L}_{\text{Gal}}^{\text{DG}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2) \\ \forall (\mathbf{w}_1(\mathbf{x}), \mathbf{w}_2(\mathbf{x})) \in \mathcal{U}^{\text{dg}} \times \mathcal{U}^{\text{dg}}, \quad (q_1(\mathbf{x}), q_2(\mathbf{x})) \in \mathcal{P}^{\text{dg}}, \end{aligned} \quad (3.19)$$

where the bilinear form and the linear functional are, respectively, defined as:

$$\begin{aligned} \mathcal{B}_{\text{Gal}}^{\text{DG}} := & (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1) - (\text{div}[\mathbf{w}_1]; p_1) + (q_1; \text{div}[\mathbf{u}_1]) + ([\![\mathbf{w}_1]\!]; [\!\{p_1\}\!])_{\Gamma^{\text{int}}} - (\{q_1\}; [\![\mathbf{u}_1]\!])_{\Gamma^{\text{int}}} \\ & + (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2) - (\text{div}[\mathbf{w}_2]; p_2) + (q_2; \text{div}[\mathbf{u}_2]) + ([\![\mathbf{w}_2]\!]; [\!\{p_2\}\!])_{\Gamma^{\text{int}}} - (\{q_2\}; [\![\mathbf{u}_2]\!])_{\Gamma^{\text{int}}} \\ & + \left(q_1 - q_2; \frac{\beta}{\mu} (p_1 - p_2) \right) + (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_1)_{\Gamma_1^u} + (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_2)_{\Gamma_2^u} \\ & - (q_1; \mathbf{u}_1 \cdot \hat{\mathbf{n}})_{\Gamma_1^u} - (q_2; \mathbf{u}_2 \cdot \hat{\mathbf{n}})_{\Gamma_2^u} \quad \text{and} \end{aligned} \quad (3.20a)$$

$$\begin{aligned} \mathcal{L}_{\text{Gal}}^{\text{DG}} := & (\mathbf{w}_1; \gamma \mathbf{b}_1) + (\mathbf{w}_2; \gamma \mathbf{b}_2) - (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_{01})_{\Gamma_1^p} - (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_{02})_{\Gamma_2^p} - (q_1; u_{n1})_{\Gamma_1^u} - (q_2; u_{n2})_{\Gamma_2^u}. \end{aligned} \quad (3.20b)$$

The classical mixed DG formulation is not stable under all combinations of interpolation functions for the field variables, which is due to the violation of the LBB *inf-sup* stability condition [34]. Specifically, equal-order interpolation for all the field variables is not stable under the classical mixed DG formulation. This numerical instability (due to the interpolation functions) is different from the aforementioned instability due to the numerical fluxes (i.e., Bassi-Rebay DG method). We develop a stabilized mixed DG formulation which does not suffer from any of the aforementioned instabilities. This is achieved by adding adjoint-type, residual-based stabilization terms (which are defined over the subdomains and circumvent the LBB *inf-sup* stability condition) and by incorporating appropriate numerical fluxes (which are consistent and stable and are defined along the subdomain interfaces).

3.3.3 Proposed stabilized mixed DG formulation

This formulation makes the following choices:

$$\lambda_1^{(1)} = \lambda_2^{(1)} = 1, \quad \lambda_1^{(3)} = \eta_u h_\gamma \{\mu k_1^{-1}\} \text{ and } \lambda_2^{(3)} = \eta_u h_\gamma \{\mu k_2^{-1}\} \text{ and} \quad (3.21a)$$

$$\Lambda_1^{(1)} = \Lambda_2^{(1)} = 1, \quad \Lambda_1^{(3)} = \frac{\eta_p}{h_\gamma} \{\mu^{-1} k_1\} \text{ and } \Lambda_2^{(3)} = \frac{\eta_p}{h_\gamma} \{\mu^{-1} k_2\} \quad (3.21b)$$

and the other constants in equations (3.12)–(3.15) are taken to be zero. η_u and η_p are non-negative, non-dimensional bounded constants. The corresponding numerical fluxes on Γ^{int} take the following form:

$$\begin{aligned} \overset{*}{p}_1 &= \{p_1\} + \eta_u h_\gamma \{\mu k_1^{-1}\} [\![\mathbf{u}_1]\!], \quad \overset{*}{p}_2 = \{p_2\} + \eta_u h_\gamma \{\mu k_2^{-1}\} [\![\mathbf{u}_2]\!], \\ \overset{*}{\mathbf{u}}_1 &= \{\mathbf{u}_1\} + \frac{\eta_p}{h_\gamma} \{\mu^{-1} k_1\} [\![p_1]\!] \text{ and } \overset{*}{\mathbf{u}}_2 = \{\mathbf{u}_2\} + \frac{\eta_p}{h_\gamma} \{\mu^{-1} k_2\} [\![p_2]\!]. \end{aligned} \quad (3.22)$$

The mathematical statement of the proposed stabilized mixed DG formulation reads as: Find $(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x})) \in \mathcal{U}^{\text{dg}} \times \mathcal{U}^{\text{dg}}$, $(p_1(\mathbf{x}), p_2(\mathbf{x})) \in \mathcal{Q}^{\text{dg}}$ such that we have

$$\begin{aligned} \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2; \mathbf{u}_1, \mathbf{u}_2, p_1, p_2) &= \mathcal{L}_{\text{stab}}^{\text{DG}}(\mathbf{w}_1, \mathbf{w}_2, q_1, q_2) \\ \forall (\mathbf{w}_1(\mathbf{x}), \mathbf{w}_2(\mathbf{x})) \in \mathcal{U}^{\text{dg}} \times \mathcal{U}^{\text{dg}}, \quad (q_1(\mathbf{x}), q_2(\mathbf{x})) \in \mathcal{Q}^{\text{dg}} \end{aligned} \quad (3.23)$$

where the bilinear form and the linear functional are, respectively, defined as:

$$\begin{aligned} \mathcal{B}_{\text{stab}}^{\text{DG}} &:= \mathcal{B}_{\text{Gal}}^{\text{DG}} - \frac{1}{2} (\mu k_1^{-1} \mathbf{w}_1 - \text{grad}[q_1]; \mu^{-1} k_1 (\mu k_1^{-1} \mathbf{u}_1 + \text{grad}[p_1])) \\ &\quad - \frac{1}{2} (\mu k_2^{-1} \mathbf{w}_2 - \text{grad}[q_2]; \mu^{-1} k_2 (\mu k_2^{-1} \mathbf{u}_2 + \text{grad}[p_2])) \\ &\quad + (\eta_u h_\gamma \{\mu k_1^{-1}\} [\![\mathbf{w}_1]\!]; [\![\mathbf{u}_1]\!])_{\Gamma^{\text{int}}} + (\eta_u h_\gamma \{\mu k_2^{-1}\} [\![\mathbf{w}_2]\!]; [\![\mathbf{u}_2]\!])_{\Gamma^{\text{int}}} \\ &\quad + \left(\frac{\eta_p}{h_\gamma} \{\mu^{-1} k_1\} [\![q_1]\!]; [\![p_1]\!] \right)_{\Gamma^{\text{int}}} + \left(\frac{\eta_p}{h_\gamma} \{\mu^{-1} k_2\} [\![q_2]\!]; [\![p_2]\!] \right)_{\Gamma^{\text{int}}} \end{aligned} \quad (3.24a)$$

$$\begin{aligned} \mathcal{L}_{\text{stab}}^{\text{DG}} &:= \mathcal{L}_{\text{Gal}}^{\text{DG}} - \frac{1}{2} (\mu k_1^{-1} \mathbf{w}_1 - \text{grad}[q_1]; \mu^{-1} k_1 \gamma \mathbf{b}_1) - \frac{1}{2} (\mu k_2^{-1} \mathbf{w}_2 - \text{grad}[q_2]; \mu^{-1} k_2 \gamma \mathbf{b}_2) \end{aligned} \quad (3.24b)$$

To completely define the formulation, the parameters η_u and η_p have to be prescribed. We make the following recommendation, which is based on the theoretical convergence analysis (see Chapter 4) and extensive numerical simulations (see section 5.1–section 5.4):

- (i) For conforming approximations, the parameters can be taken to be $\eta_u = \eta_p = 0$.
- (ii) For non-conforming approximations, the parameters can be taken to be $\eta_u = \eta_p = 10$ or 100 . (See section 5.1.2.1).

A few remarks about the stabilized formulation are in order.

- (a) The above stabilized formulation is an adjoint-type formulation. We have posed even the classical mixed formulation as an adjoint-type (see the bilinear form (3.20a)). In addition, the stabilization terms within the elements (i.e., in $\tilde{\Omega}$) are of adjoint-type, which look similar to the one proposed by [92] for the two-field Darcy equations.
- (b) Since the formulation is of adjoint-type, the formulation will not give rise to symmetric coefficient (“stiffness”) matrix. But the coefficient matrix will be positive definite, which can be inferred from Lemma 4.1.1. Alternatively, the above stabilized formulation can be posed as an equivalent symmetric formulation by replacing q_1 and q_2 with $-q_1$ and $-q_2$, respectively; which is justified as q_1 and q_2 are arbitrary weighting functions. In this case, the resulting symmetric formulation will not result in positive-definite coefficient matrix.
- (c) In order to minimize the drift in the solution fields, especially in the case of non-conforming discretization, additional stabilization terms on the interior boundaries

(i.e., terms containing η_u and η_p) are required in both networks. The necessity of employing such stabilization parameters has been addressed by [17] for the case of Darcy equations. It is noteworthy that η_u parameter was not included in the formulation proposed by [92], as they did not consider non-conforming approximations.

- (d) Due to the presence of the terms containing η_u and η_p , the above numerical fluxes are no longer similar to the ones proposed by [23]. The numerical fluxes employed in the proposed formulation are not the same as any of the DG methods discussed in the review paper [10].
- (e) In the case of Darcy equations, a stabilized formulation without edge stabilization terms has been developed and its convergence has been established by utilizing a lifting operator [36]. The question about whether such an approach can be extended to the DPP model is worthy of an investigation, but is beyond the scope of this text.

Chapter 4

Theoretical Analysis

4.1 Stability norm

We start by grouping the field variables and their corresponding weighting functions as

$$\mathbf{U} = (\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), p_1(\mathbf{x}), p_2(\mathbf{x})) \in \mathbb{U} \text{ and} \quad (4.1a)$$

$$\mathbf{W} = (\mathbf{w}_1(\mathbf{x}), \mathbf{w}_2(\mathbf{x}), q_1(\mathbf{x}), q_2(\mathbf{x})) \in \mathbb{U}, \quad (4.1b)$$

where the product space \mathbb{U} is defined as

$$\mathbb{U} = \mathcal{U}^{\text{dg}} \times \mathcal{U}^{\text{dg}} \times \mathcal{Q}^{\text{dg}}. \quad (4.2)$$

The proposed stabilized mixed DG formulation (3.23) can then be compactly written as: Find $\mathbf{U} \in \mathbb{U}$ such that we have

$$\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{W}; \mathbf{U}) = \mathcal{L}_{\text{stab}}^{\text{DG}}(\mathbf{W}) \quad \forall \mathbf{W} \in \mathbb{U}. \quad (4.3)$$

The stability of the proposed weak formulation will be established under the following norm:

$$\begin{aligned} (\|\mathbf{W}\|_{\text{stab}}^{\text{DG}})^2 := \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{W}; \mathbf{W}) &= \frac{1}{2} \left\| \sqrt{\frac{\mu}{k_1}} \mathbf{w}_1 \right\|^2 + \frac{1}{2} \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[q_1] \right\|^2 \\ &\quad + \frac{1}{2} \left\| \sqrt{\frac{\mu}{k_2}} \mathbf{w}_2 \right\|^2 + \frac{1}{2} \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[q_2] \right\|^2 + \left\| \sqrt{\frac{\beta}{\mu}} (q_1 - q_2) \right\|^2 \end{aligned}$$

$$\begin{aligned}
& + \left\| \sqrt{\eta_u h_\Upsilon \{\!\{ \mu k_1^{-1} \}\!}} [\![\mathbf{w}_1]\!] \right\|_{\Gamma^{\text{int}}}^2 + \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\!\{ \mu^{-1} k_1 \}\!}} [\![q_1]\!] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \left\| \sqrt{\eta_u h_\Upsilon \{\!\{ \mu k_2^{-1} \}\!}} [\![\mathbf{w}_2]\!] \right\|_{\Gamma^{\text{int}}}^2 + \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\!\{ \mu^{-1} k_2 \}\!}} [\![q_2]\!] \right\|_{\Gamma^{\text{int}}}^2 \quad \forall \mathbf{W} \in \mathbb{U}
\end{aligned} \tag{4.4}$$

Lemma 4.1.1. (*Stability norm*) $\|\cdot\|_{\text{stab}}^{\text{DG}}$ is a norm on \mathbb{U} .

Proof. The mathematical proof is similar to that of the continuous formulation, which is provided in [96].

4.2 Convergence theorem and error analysis

In order to perform the error analysis of the proposed stabilized mixed DG formulation, we need to define the finite element solution \mathbf{U}^h and the corresponding weighting function as

$$\mathbf{U}^h = (\mathbf{u}_1^h(\mathbf{x}), \mathbf{u}_2^h(\mathbf{x}), p_1^h(\mathbf{x}), p_2^h(\mathbf{x})) \in \mathbb{U}^h \text{ and} \tag{4.5a}$$

$$\mathbf{W}^h = (\mathbf{w}_1^h(\mathbf{x}), \mathbf{w}_2^h(\mathbf{x}), q_1^h(\mathbf{x}), q_2^h(\mathbf{x})) \in \mathbb{U}^h. \tag{4.5b}$$

\mathbb{U}^h is the closed linear subspace of \mathbb{U} and is defined as

$$\mathbb{U}^h = \mathcal{U}^h \times \mathcal{U}^h \times \mathcal{Q}^h, \tag{4.6}$$

where

$$\begin{aligned}
\mathcal{U}^h := & \left\{ \mathbf{u}^h(\mathbf{x}) \in \mathcal{U}^{\text{dg}} \mid \mathbf{u}^h(\mathbf{x}) \in (C^0(\bar{\omega}^i))^{nd}; \mathbf{u}^h(\mathbf{x})|_{\omega^i} \in (\mathcal{P}^k(\omega^i))^{nd}; i = 1, \dots, N_{\text{ele}} \right\} \\
& \tag{4.7a}
\end{aligned}$$

$$\begin{aligned}
\mathcal{Q}^h := & \left\{ (p_1^h, p_2^h) \in \mathcal{Q}^{\text{dg}} \mid p_1^h(\mathbf{x}), p_2^h(\mathbf{x}) \in C^0(\bar{\omega}^i); p_1^h(\mathbf{x}), p_2^h(\mathbf{x})|_{\omega^i} \in \mathcal{P}^l(\omega^i); i = 1, \dots, N_{\text{ele}} \right\} \\
& \tag{4.7b}
\end{aligned}$$

and $C^0(\bar{\omega}^i)$ is the set of all continuous functions defined on $\bar{\omega}^i$ (which is the set closure of ω^i).

The finite element formulation corresponding to the proposed stabilized mixed DG formulation is defined as: Find $\mathbf{U}^h \in \mathbb{U}^h$ such that we have

$$\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h; \mathbf{U}^h) = \mathcal{L}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h) \quad \forall \mathbf{W}^h \in \mathbb{U}^h. \quad (4.8)$$

The error in the finite element solution \mathbf{E} is defined as the difference between the finite element solution and the exact solution. If we define $\tilde{\mathbf{U}}^h$ as an “interpolate” of \mathbf{U} onto \mathbb{U}^h [29], decomposition of the error can be performed as

$$\mathbf{E} := \mathbf{U}^h - \mathbf{U} = \mathbf{E}^h + \mathbf{H}, \quad (4.9)$$

where $\mathbf{E}^h = \mathbf{U}^h - \tilde{\mathbf{U}}^h$ is the approximation error and $\mathbf{H} = \tilde{\mathbf{U}}^h - \mathbf{U}$ is the interpolation error. The components of \mathbf{E} and \mathbf{H} are as

$$\mathbf{E} = \{\mathbf{e}_{\mathbf{u}_1}, \mathbf{e}_{\mathbf{u}_2}, e_{p_1}, e_{p_2}\} \quad \text{and} \quad \mathbf{H} = \{\boldsymbol{\eta}_{\mathbf{u}_1}, \boldsymbol{\eta}_{\mathbf{u}_2}, \eta_{p_1}, \eta_{p_2}\}. \quad (4.10)$$

Lemma 4.2.1. (*Estimates for approximation errors on Γ^{int} .*) *On a sequence of admissible meshes, the following estimates hold:*

$$\left\| \sqrt{h_\gamma \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \mathbf{e}_{\mathbf{u}_1} \}\!} \right\|_{\Gamma^{\text{int}}}^2 \leq C_{\mathbf{e}_{\mathbf{u}_1}} \left\| \sqrt{\mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|^2 \quad \text{and} \quad (4.11)$$

$$\left\| \sqrt{h_\gamma \{\!\{ \mu k_2^{-1} \}\!}} \{\!\{ \mathbf{e}_{\mathbf{u}_2} \}\!} \right\|_{\Gamma^{\text{int}}}^2 \leq C_{\mathbf{e}_{\mathbf{u}_2}} \left\| \sqrt{\mu k_2^{-1}} \mathbf{e}_{\mathbf{u}_2} \right\|^2. \quad (4.12)$$

Proof. We first note that

$$\left\| \sqrt{h_\gamma \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \mathbf{e}_{\mathbf{u}_1} \}\!} \right\|_{\Gamma^{\text{int}}}^2 = \sum_{\gamma \in \mathcal{E}^{\text{int}}} \left\| \sqrt{h_\gamma \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \mathbf{e}_{\mathbf{u}_1} \}\!} \right\|_\gamma^2. \quad (4.13)$$

We now bound the approximation error of \mathbf{u}_1 on an interior edge $\Upsilon \in \mathcal{E}^{\text{int}}$. The Cauchy-Schwarz inequality implies the following

$$\left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \{\mathbf{e}_{\mathbf{u}_1}\} \right\|_\Upsilon^2 \leq \frac{1}{2} \left(\left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^+ \cap \Upsilon}^2 + \left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^- \cap \Upsilon}^2 \right). \quad (4.14)$$

Noting the boundedness of the drag coefficients (i.e., equation (2.4)), we obtain the following

$$\left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \{\mathbf{e}_{\mathbf{u}_1}\} \right\|_\Upsilon^2 \leq \frac{1}{2} \mathcal{C}_{\text{drag},1} \left(\left\| \sqrt{h_\Upsilon \mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^+ \cap \Upsilon}^2 + \left\| \sqrt{h_\Upsilon \mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^- \cap \Upsilon}^2 \right). \quad (4.15)$$

Using the bound based on the locally quasi-uniform condition (i.e., inequality (2.27)) we obtain the following

$$\left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \{\mathbf{e}_{\mathbf{u}_1}\} \right\|_\Upsilon^2 \leq \frac{1}{4} \mathcal{C}_{\text{drag},1} (1 + \mathcal{C}_{\text{lqu}}) \left(\left\| \sqrt{h_{\omega_\Upsilon^+} \mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^+ \cap \Upsilon}^2 + \left\| \sqrt{h_{\omega_\Upsilon^-} \mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^- \cap \Upsilon}^2 \right). \quad (4.16)$$

By summing over all the interior edges we obtain the following

$$\begin{aligned} \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \{\mathbf{e}_{\mathbf{u}_1}\} \right\|_\Upsilon^2 &\leq \frac{1}{4} \mathcal{C}_{\text{drag},1} (1 + \mathcal{C}_{\text{lqu}}) \sum_{\omega \in \mathcal{T}} \left\| \sqrt{h_\omega \mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega \setminus \partial\Omega}^2 \\ &\leq \frac{1}{4} \mathcal{C}_{\text{drag},1} (1 + \mathcal{C}_{\text{lqu}}) \sum_{\omega \in \mathcal{T}} \left\| \sqrt{h_\omega \mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_{\partial\omega}^2. \end{aligned} \quad (4.17)$$

By invoking the discrete trace inequality (2.40) we obtain the following

$$\begin{aligned} \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \left\| \sqrt{h_\Upsilon \{\mu k_1^{-1}\}} \{\mathbf{e}_{\mathbf{u}_1}\} \right\|_\Upsilon^2 &\leq \frac{1}{4} \mathcal{C}_{\text{drag},1} (1 + \mathcal{C}_{\text{lqu}}) \mathcal{C}_{\text{trace}}^2 (1 + \mathcal{C}_{\text{inv}})^2 \sum_{\omega \in \mathcal{T}} \left\| \sqrt{\mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_\omega^2 \\ &\leq \frac{1}{4} \mathcal{C}_{\text{drag},1} \mathcal{C}_{\text{trace}}^2 (1 + \mathcal{C}_{\text{inv}})^2 (1 + \mathcal{C}_{\text{lqu}}) \left\| \sqrt{\mu k_1^{-1}} \mathbf{e}_{\mathbf{u}_1} \right\|_\omega^2. \end{aligned} \quad (4.18)$$

(Recall that the subscript will be dropped if the L_2 norm is over $\tilde{\Omega} := \cup_{\omega \in \mathcal{T}} \omega$.) Thus,

$$\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}} := \frac{1}{4} \mathcal{C}_{\text{drag},1} \mathcal{C}_{\text{trace}}^2 (1 + \mathcal{C}_{\text{inv}})^2 (1 + \mathcal{C}_{\text{lqu}}). \quad (4.19)$$

On similar lines, one can establish the estimate (4.12) with

$$\mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}} := \frac{1}{4} \mathcal{C}_{\text{drag},2} \mathcal{C}_{\text{trace}}^2 (1 + \mathcal{C}_{\text{inv}})^2 (1 + \mathcal{C}_{\text{lqu}}). \quad (4.20)$$

If a p -th order polynomial is employed for a field variable $f(\mathbf{x})$ on an element $\omega \in \mathcal{T}$ and the corresponding interpolate denoted by \tilde{f}^h , the following estimate holds for the interpolation error [34]

$$\|f - \tilde{f}^h\|_\omega \leq \mathcal{C}_{\text{int}} h_w^{p+1} |f|_{H^{p+1}(\omega)}, \quad (4.21)$$

where h_ω is the element diameter of ω , \mathcal{C}_{int} is a non-dimensional constant independent of h_ω and f , and $|\cdot|_{H^{p+1}(\omega)}$ is a Sobolev semi-norm, which is defined in equation (2.33).

To avoid further introduction of constants, we employ the notation $A \lesssim B$ to denote that there exists a constant C , independent of the mesh size, such that $A \leq CB$. A similar definition holds for $A \gtrsim B$. The notation $A \sim B$ denotes the case when $A \lesssim B$ and $A \gtrsim B$ hold simultaneously.

Lemma 4.2.2. (*Estimates for interpolation errors on Γ^{int} .*) *If polynomial orders used for interpolation of \mathbf{u}_1 , \mathbf{u}_2 , p_1 and p_2 are, respectively, p , q , r and s then the following estimates hold for the interpolation errors on Γ^{int} :*

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}\} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(p+1)} |\mathbf{u}_1|_{H^{p+1}(\omega)}^2, \quad (4.22)$$

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_2^{-1} \}\!}\} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_2} \}\!} \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(q+1)} |\mathbf{u}_2|_{H^{q+1}(\omega)}^2, \quad (4.23)$$

$$\left\| \sqrt{h_\Upsilon^{-1} \{\!\{ \mu^{-1} k_1 \}\!}\} [\![\eta_{p_1}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2r} |p_1|_{H^{r+1}(\omega)}^2, \text{ and} \quad (4.24)$$

$$\left\| \sqrt{h_\Upsilon^{-1} \{\!\{ \mu^{-1} k_2 \}\!}\} [\![\eta_{p_2}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2s} |p_2|_{H^{s+1}(\omega)}^2. \quad (4.25)$$

Proof. We first establish the estimate (4.22). The boundedness of the drag coefficient $\mu/k_1(\mathbf{x})$ and the linearity of a norm imply the

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Upsilon}^2 \leq \frac{1}{\eta_p} \left(\sup_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} \right) \left\| \sqrt{h_\Upsilon} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Upsilon}^2 \quad \forall \Upsilon \in \mathcal{E}^{\text{int}}. \quad (4.26)$$

Using the triangle inequality and the bound from the locally quasi-uniform condition (2.27), we obtain the:

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Upsilon}^2 \lesssim \frac{1}{4} (1 + C_{\text{lqu}}) \left(\left\| \sqrt{h_{\omega_\Upsilon^+}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^+ \cap \Upsilon}^2 + \left\| \sqrt{h_{\omega_\Upsilon^-}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|_{\partial\omega_\Upsilon^- \cap \Upsilon}^2 \right) \quad \forall \Upsilon \in \mathcal{E}^{\text{int}}. \quad (4.27)$$

By summing over all the interior edges and noting the linearity of a norm, we obtain

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Gamma^{\text{int}}}^2 = \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Upsilon}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} (h_\omega \|\boldsymbol{\eta}_{\mathbf{u}_1}\|_{\partial\omega}^2). \quad (4.28)$$

By invoking the discrete trace inequality (2.40), we obtain the following inequality

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} \|\boldsymbol{\eta}_{\mathbf{u}_1}\|_{\omega}^2. \quad (4.29)$$

If a polynomial of order p is employed for approximating \mathbf{u}_1 , then the standard estimate for the interpolation error (4.21) provides

$$\left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!} \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(p+1)} |\mathbf{u}_1|_{H^{p+1}(\omega)}^2, \quad (4.30)$$

which is the estimate (4.22). By reasoning out on similar lines, one can establish the estimate (4.23).

We now establish the estimate (4.24). The boundedness of the drag coefficient $\mu/k_1(\mathbf{x})$ and the linearity of a norm imply

$$\left\| \sqrt{h_\Upsilon^{-1} \{\!\{ \mu^{-1} k_1 \}\!}} [\![\eta_{p_1}]\!] \right\|_{\Upsilon}^2 \leq \left(\inf_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} \right) \left\| \sqrt{h_\Upsilon^{-1}} [\![\eta_{p_1}]\!] \right\|_{\Upsilon}^2 \quad \forall \Upsilon \in \mathcal{E}^{\text{int}}. \quad (4.31)$$

Using the triangle inequality and the bound from the locally quasi-uniform condition (2.27), we obtain

$$\left\| \sqrt{h_\Upsilon^{-1} \{\!\{ \mu^{-1} k_1 \}\!}} [\![\eta_{p_1}]\!] \right\|_{\Upsilon}^2 \lesssim 4 \left(1 + \frac{1}{C_{\text{lqu}}} \right)^{-1} \left(\left\| \sqrt{h_{\omega_\Upsilon^+}^{-1}} \eta_{p_1} \right\|_{\partial\omega_\Upsilon^+ \cap \Upsilon}^2 + \left\| \sqrt{h_{\omega_\Upsilon^-}^{-1}} \eta_{p_1} \right\|_{\partial\omega_\Upsilon^- \cap \Upsilon}^2 \right) \quad \forall \Upsilon \in \mathcal{E}^{\text{int}}. \quad (4.32)$$

By summing over all the interior edges and noting the linearity of a norm, we obtain

$$\left\| \sqrt{h_{\Upsilon}^{-1}\{\mu^{-1}k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 = \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \left\| \sqrt{h_{\Upsilon}^{-1}\{\mu^{-1}k_1\}} [\eta_{p_1}] \right\|_{\Upsilon}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} \left(h_{\omega}^{-1} \|\eta_{p_1}\|_{\partial\omega}^2 \right). \quad (4.33)$$

By invoking the discrete trace inequality (2.39), we obtain the following inequality

$$\left\| \sqrt{h_{\Upsilon}^{-1}\{\mu^{-1}k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} \left(h_{\omega}^{-2} \|\eta_{p_1}\|_{\omega}^2 \right). \quad (4.34)$$

If a polynomial of order r is employed for approximating p_1 , then the standard estimate for the interpolation error (4.21) provides

$$\left\| \sqrt{h_{\Upsilon}^{-1}\{\mu^{-1}k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_{\omega}^{2r} |p_1|_{H^{r+1}(\omega)}^2, \quad (4.35)$$

which is the estimate (4.24). By reasoning out on similar lines, one can establish the estimate (4.25).

Lemma 4.2.3. (*Estimate for \mathbf{H} under the stability norm.*) *If polynomial orders used for interpolation of \mathbf{u}_1 , \mathbf{u}_2 , p_1 and p_2 are, respectively, p , q , r and s then the following estimate holds*

$$\begin{aligned} (\|\mathbf{H}\|_{\text{stab}}^{\text{DG}})^2 &\lesssim \sum_{\omega \in \mathcal{T}_h} \left(h_{\omega}^{2(p+1)} |\mathbf{u}_1|_{H^{p+1}(\omega)}^2 + h_{\omega}^{2(q+1)} |\mathbf{u}_2|_{H^{q+1}(\omega)}^2 + (1 + h_{\omega}^2) h_{\omega}^{2r} |p_1|_{H^{r+1}(\omega)}^2 \right. \\ &\quad \left. + (1 + h_{\omega}^2) h_{\omega}^{2s} |p_2|_{H^{s+1}(\omega)}^2 \right), \end{aligned} \quad (4.36)$$

where the constant in the estimate is independent of the characteristic mesh length (h or h_{ω}) and the solution fields (\mathbf{u}_1 , \mathbf{u}_2 , p_1 and p_2).

Proof. The definition of the stability norm (4.4) and the components of \mathbf{H} (4.10) imply

$$\begin{aligned} (\|\mathbf{H}\|_{\text{stab}}^{\text{DG}})^2 &= \frac{1}{2} \left\| \sqrt{\frac{\mu}{k_1}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|^2 + \frac{1}{2} \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[\eta_{p_1}] \right\|^2 + \frac{1}{2} \left\| \sqrt{\frac{\mu}{k_2}} \boldsymbol{\eta}_{\mathbf{u}_2} \right\|^2 + \frac{1}{2} \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[\eta_{p_2}] \right\|^2 \\ &\quad + \left\| \sqrt{\frac{\beta}{\mu}} (\eta_{p_1} - \eta_{p_2}) \right\|^2 + \left\| \sqrt{\eta_u h_{\Upsilon} \{\mu k_1^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_1}] \right\|_{\Gamma^{\text{int}}}^2 + \left\| \sqrt{\frac{\eta_p}{h_{\Upsilon}} \{\mu^{-1}k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 \\ &\quad + \left\| \sqrt{\eta_u h_{\Upsilon} \{\mu k_2^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_2}] \right\|_{\Gamma^{\text{int}}}^2 + \left\| \sqrt{\frac{\eta_p}{h_{\Upsilon}} \{\mu^{-1}k_2\}} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2. \end{aligned} \quad (4.37)$$

Using the boundedness of the drag coefficient of the first pore-network, linearity of a norm and the standard estimate for the interpolation error (4.21), and noting that the polynomial order of approximation for \mathbf{u}_1 is p , we obtain

$$\frac{1}{2} \left\| \sqrt{\frac{\mu}{k_1}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|^2 \leq \frac{1}{2} \sup_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} \sum_{\omega \in \mathcal{T}_h} \|\boldsymbol{\eta}_{\mathbf{u}_1}\|_\omega^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(p+1)} |\mathbf{u}_1|_{H^{p+1}(\omega)}^2. \quad (4.38)$$

Similarly,

$$\frac{1}{2} \left\| \sqrt{\frac{\mu}{k_2}} \boldsymbol{\eta}_{\mathbf{u}_2} \right\|^2 \leq \frac{1}{2} \sup_{\mathbf{x} \in \Omega} \frac{\mu}{k_2(\mathbf{x})} \sum_{\omega \in \mathcal{T}_h} |\boldsymbol{\eta}_{\mathbf{u}_2}|_\omega^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(q+1)} |\mathbf{u}_2|_{H^{q+1}(\omega)}^2. \quad (4.39)$$

For the *second* term, we proceed as follows by first noting the boundedness of the drag coefficient in the first pore-network

$$\begin{aligned} \frac{1}{2} \left\| \sqrt{\frac{k_1}{\mu}} \operatorname{grad}[\eta_{p_1}] \right\|^2 &\leq \frac{1}{2} \inf_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} \sum_{\omega \in \mathcal{T}_h} \|\operatorname{grad}[\eta_{p_1}]\|_\omega^2 \\ &\leq \frac{1}{2} \inf_{\mathbf{x} \in \Omega} \frac{\mu}{k_1(\mathbf{x})} C_{\text{inv}}^2 \sum_{\omega \in \mathcal{T}_h} h_\omega^{-2} \|\eta_{p_1}\|_\omega^2 \quad [\text{inverse estimate (2.37)}] \\ &\lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2r} |p_1|_{H^{r+1}(\omega)}^2 \quad [\text{interpolation estimate (4.21)}]. \end{aligned} \quad (4.40)$$

Similarly, one can derive the following estimate for the *fourth* term

$$\frac{1}{2} \left\| \sqrt{\frac{k_2}{\mu}} \operatorname{grad}[\eta_{p_2}] \right\|^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2s} |p_2|_{H^{s+1}(\omega)}^2. \quad (4.41)$$

The estimate for the *fifth* term utilizes the triangle inequality and the interpolation estimate (4.21) and it can be obtained as

$$\left\| \sqrt{\frac{\beta}{\mu}} (\eta_{p_1} - \eta_{p_2}) \right\|^2 \leq \frac{\beta}{\mu} \sum_{\omega \in \mathcal{T}_h} \left(\|\eta_{p_1}\|_\omega^2 + \|\eta_{p_2}\|_\omega^2 \right) \lesssim \sum_{\omega \in \mathcal{T}_h} \left(h_\omega^{2(r+1)} |p_1|_{H^{r+1}(\omega)}^2 + h_\omega^{2(s+1)} |p_2|_{H^{s+1}(\omega)}^2 \right). \quad (4.42)$$

Using the boundedness of η_u and the drag coefficient of the first pore-network and noting the linearity of a norm, we obtain the following estimate for the *sixth* term

$$\left\| \sqrt{\eta_u h_\Gamma \{ \mu k_1^{-1} \}} [\![\boldsymbol{\eta}_{\mathbf{u}_1}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \left\| \sqrt{h_\Upsilon} [\![\boldsymbol{\eta}_{\mathbf{u}_1}]\!] \right\|_\Upsilon^2. \quad (4.43)$$

Using the bound based on the locally quasi-uniform condition (2.27) and the triangle inequality, we obtain

$$\begin{aligned} \left\| \sqrt{\eta_u h_\Gamma \{ \mu k_1^{-1} \}} [\![\boldsymbol{\eta}_{\mathbf{u}_1}]\!] \right\|_{\Gamma^{\text{int}}}^2 &\lesssim \sum_{\Upsilon \in \mathcal{E}^{\text{int}}} \left(\left\| \sqrt{h_{\omega_\Upsilon^+}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|_{\partial \omega_\Upsilon^+ \cap \Upsilon}^2 + \left\| \sqrt{h_{\omega_\Upsilon^-}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|_{\partial \omega_\Upsilon^- \cap \Upsilon}^2 \right) \\ &\lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega \left\| \boldsymbol{\eta}_{\mathbf{u}_1} \right\|_{\partial \omega}^2. \end{aligned} \quad (4.44)$$

Using the discrete trace inequality (2.40) and the standard interpolation estimate (4.21), we obtain

$$\left\| \sqrt{\eta_u h_\Gamma \{ \mu k_1^{-1} \}} [\![\boldsymbol{\eta}_{\mathbf{u}_1}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} \left\| \boldsymbol{\eta}_{\mathbf{u}_1} \right\|_\omega^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(p+1)} |\mathbf{u}_1|_{H^{p+1}(\omega)}^2. \quad (4.45)$$

A similar argument gives rise to the following estimate for the *eighth* term

$$\left\| \sqrt{\eta_u h_\Gamma \{ \mu k_2^{-1} \}} [\![\boldsymbol{\eta}_{\mathbf{u}_2}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2(q+1)} |\mathbf{u}_2|_{H^{q+1}(\omega)}^2. \quad (4.46)$$

Noting that η_p is a bounded constant, estimates (4.24) and (4.25) immediately imply the following estimates for the *seventh* and *ninth* terms:

$$\left\| \sqrt{\frac{\eta_p}{h_\Gamma} \{ \mu^{-1} k_1 \}} [\![\boldsymbol{\eta}_{p_1}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2r} |p_1|_{H^{r+1}(\omega)}^2 \text{ and} \quad (4.47)$$

$$\left\| \sqrt{\frac{\eta_p}{h_\Gamma} \{ \mu^{-1} k_2 \}} [\![\boldsymbol{\eta}_{p_2}]\!] \right\|_{\Gamma^{\text{int}}}^2 \lesssim \sum_{\omega \in \mathcal{T}_h} h_\omega^{2s} |p_2|_{H^{s+1}(\omega)}^2. \quad (4.48)$$

By adding up the individual estimates for all the terms, we obtain the desired result.

Theorem 4.2.1. (*Consistency*) *The error in the finite element solution satisfies*

$$\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h; \mathbf{E}) = 0 \quad \forall \mathbf{W}^h \in \mathbb{U}^h \subset \mathbb{U}. \quad (4.49)$$

Proof. The proof follows a standard procedure employed in the literature. Equation (4.3) implies that for all $\mathbf{W}^h \in \mathbb{U}^h \subset \mathbb{U}$ we have:

$$\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h; \mathbf{U}^h) = \mathcal{L}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h) \text{ and} \quad (4.50\text{a})$$

$$\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h; \mathbf{U}) = \mathcal{L}_{\text{stab}}^{\text{DG}}(\mathbf{W}^h) \quad (4.50\text{b})$$

By subtracting the above two equations, invoking the linearity in the second slot of $\mathcal{B}_{\text{stab}}^{\text{DG}}(\cdot; \cdot)$ and noting the definition of \mathbf{E} given by (4.9), we obtain the desired result.

Theorem 4.2.2. (*Convergence*) Under a sequence of admissible meshes, the finite element solution $\mathbf{U}^h \in \mathbb{U}^h$ tends to the exact solution $\mathbf{U} \in \mathbb{U}$ almost everywhere¹ as the mesh-size $h \rightarrow 0$.

Proof. The error with respect to the stability norm can be rewritten as

$$(\|\mathbf{E}\|_{\text{stab}}^{\text{DG}})^2 = \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{E}; \mathbf{E}) = \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{E}^h + \mathbf{H}; \mathbf{E}) = \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{E}^h; \mathbf{E}) + \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{H}; \mathbf{E}) = \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{H}; \mathbf{E}). \quad (4.51)$$

We invoked the definition of $\|\cdot\|_{\text{stab}}^{\text{DG}}$ norm (i.e., Eq. (4.4)) for establishing the first equality, the decomposition of the error (i.e., Eq. (4.9)) for the second equality, linearity in the first slot of $\mathcal{B}_{\text{stab}}^{\text{DG}}(\cdot; \cdot)$ for the third equality, and consistency (i.e., Theorem 4.2.1) for the fourth equality. We now expand $\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{H}; \mathbf{E})$ as

$$\begin{aligned} \mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{H}; \mathbf{E}) &= \mathcal{B}_{\text{stab}}^{\text{DG}}(\boldsymbol{\eta}_{\mathbf{u}_1}, \boldsymbol{\eta}_{\mathbf{u}_2}, \eta_{p_1}, \eta_{p_2}; \mathbf{e}_{\mathbf{u}_1}, \mathbf{e}_{\mathbf{u}_2}, e_{p_1}, e_{p_2}) \\ &= \frac{1}{2}(\boldsymbol{\eta}_{\mathbf{u}_1}; \mu k_1^{-1} \mathbf{e}_{\mathbf{u}_1}) + \frac{1}{2}(\boldsymbol{\eta}_{\mathbf{u}_1}; \text{grad}[e_{p_1}]) - \frac{1}{2}(\text{grad}[\eta_{p_1}]; \mathbf{e}_{\mathbf{u}_1}) \\ &\quad + \frac{1}{2}(\boldsymbol{\eta}_{\mathbf{u}_2}; \mu k_2^{-1} \mathbf{e}_{\mathbf{u}_2}) + \frac{1}{2}(\boldsymbol{\eta}_{\mathbf{u}_2}; \text{grad}[e_{p_2}]) - \frac{1}{2}(\text{grad}[\eta_{p_2}]; \mathbf{e}_{\mathbf{u}_2}) \\ &\quad + \frac{1}{2} \left(\text{grad}[\eta_{p_1}]; \frac{k_1}{\mu} \text{grad}[e_{p_1}] \right) + \frac{1}{2} \left(\text{grad}[\eta_{p_2}]; \frac{k_2}{\mu} \text{grad}[e_{p_2}] \right) \end{aligned}$$

¹Two quantities that are the same except on a set of measure zero are said to be equal almost everywhere [67].

$$\begin{aligned}
& + \left((\eta_{p_1} - \eta_{p_2}); \frac{\beta}{\mu} (e_{p_1} - e_{p_2}) \right) \\
& - (\{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!\}; \llbracket e_{p_1} \rrbracket)_{\Gamma^{\text{int}}} + (\llbracket \eta_{p_1} \rrbracket; \{\!\{ \mathbf{e}_{\mathbf{u}_1} \}\!\})_{\Gamma^{\text{int}}} - (\{\!\{ \boldsymbol{\eta}_{\mathbf{u}_2} \}\!\}; \llbracket e_{p_2} \rrbracket)_{\Gamma^{\text{int}}} + (\llbracket \eta_{p_2} \rrbracket; \{\!\{ \mathbf{e}_{\mathbf{u}_2} \}\!\})_{\Gamma^{\text{int}}} \\
& + (\eta_u h_{\Upsilon} \{\!\{ \mu k_1^{-1} \}\!\} \llbracket \boldsymbol{\eta}_{\mathbf{u}_1} \rrbracket; \llbracket \mathbf{e}_{\mathbf{u}_1} \rrbracket)_{\Gamma^{\text{int}}} + \left(\frac{\eta_p}{h_{\Upsilon}} \{\!\{ \mu^{-1} k_1 \}\!\} \llbracket \eta_{p_1} \rrbracket; \llbracket e_{p_1} \rrbracket \right)_{\Gamma^{\text{int}}} \\
& + (\eta_u h_{\Upsilon} \{\!\{ \mu k_2^{-1} \}\!\} \llbracket \boldsymbol{\eta}_{\mathbf{u}_2} \rrbracket; \llbracket \mathbf{e}_{\mathbf{u}_2} \rrbracket)_{\Gamma^{\text{int}}} + \left(\frac{\eta_p}{h_{\Upsilon}} \{\!\{ \mu^{-1} k_2 \}\!\} \llbracket \eta_{p_2} \rrbracket; \llbracket e_{p_2} \rrbracket \right)_{\Gamma^{\text{int}}}. \quad (4.52)
\end{aligned}$$

By employing Cauchy-Schwarz and Peter-Paul inequalities, we obtain the following bound²

$$\begin{aligned}
2\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{H}; \mathbf{E}) &\leq \frac{\varepsilon_1}{2} \left\| \sqrt{\frac{\mu}{k_1}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|^2 + \frac{1}{2\varepsilon_1} \left\| \sqrt{\frac{\mu}{k_1}} \mathbf{e}_{\mathbf{u}_1} \right\|^2 + \frac{\varepsilon_2}{2} \left\| \sqrt{\frac{\mu}{k_1}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|^2 + \frac{1}{2\varepsilon_2} \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[e_{p_1}] \right\|^2 \\
& + \frac{\varepsilon_3}{2} \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[\eta_{p_1}] \right\|^2 + \frac{1}{2\varepsilon_3} \left\| \sqrt{\frac{\mu}{k_1}} \mathbf{e}_{\mathbf{u}_1} \right\|^2 + \frac{\varepsilon_4}{2} \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[\eta_{p_1}] \right\|^2 + \frac{1}{2\varepsilon_4} \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[e_{p_1}] \right\|^2 \\
& + \frac{\varepsilon_5}{2} \left\| \sqrt{\frac{\mu}{k_2}} \boldsymbol{\eta}_{\mathbf{u}_2} \right\|^2 + \frac{1}{2\varepsilon_5} \left\| \sqrt{\frac{\mu}{k_2}} \mathbf{e}_{\mathbf{u}_2} \right\|^2 + \frac{\varepsilon_6}{2} \left\| \sqrt{\frac{\mu}{k_2}} \boldsymbol{\eta}_{\mathbf{u}_2} \right\|^2 + \frac{1}{2\varepsilon_6} \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[e_{p_2}] \right\|^2 \\
& + \frac{\varepsilon_7}{2} \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[\eta_{p_2}] \right\|^2 + \frac{1}{2\varepsilon_7} \left\| \sqrt{\frac{\mu}{k_2}} \mathbf{e}_{\mathbf{u}_2} \right\|^2 + \frac{\varepsilon_8}{2} \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[\eta_{p_2}] \right\|^2 + \frac{1}{2\varepsilon_8} \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[e_{p_2}] \right\|^2 \\
& + \varepsilon_9 \left\| \sqrt{\frac{\beta}{\mu}} (\eta_{p_1} - \eta_{p_2}) \right\|^2 + \frac{1}{\varepsilon_9} \left\| \sqrt{\frac{\beta}{\mu}} (e_{p_1} - e_{p_2}) \right\|^2 \\
& + \varepsilon_{10} \left\| \sqrt{\frac{h_{\Upsilon}}{\eta_p} \{\!\{ \mu k_1^{-1} \}\!\}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!\} \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{10}} \left\| \sqrt{\frac{\eta_p}{h_{\Upsilon}} \{\!\{ \mu^{-1} k_1 \}\!\}} \llbracket e_{p_1} \rrbracket \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{11} \left\| \sqrt{h_{\Upsilon}^{-1} \{\!\{ \mu^{-1} k_1 \}\!\}} \llbracket \eta_{p_1} \rrbracket \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{11}} \left\| \sqrt{h_{\Upsilon} \{\!\{ \mu k_1^{-1} \}\!\}} \{\!\{ \mathbf{e}_{\mathbf{u}_1} \}\!\} \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{12} \left\| \sqrt{\frac{h_{\Upsilon}}{\eta_p} \{\!\{ \mu k_2^{-1} \}\!\}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_2} \}\!\} \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{12}} \left\| \sqrt{\frac{\eta_p}{h_{\Upsilon}} \{\!\{ \mu^{-1} k_2 \}\!\}} \llbracket e_{p_2} \rrbracket \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{13} \left\| \sqrt{h_{\Upsilon}^{-1} \{\!\{ \mu^{-1} k_2 \}\!\}} \llbracket \eta_{p_2} \rrbracket \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{13}} \left\| \sqrt{h_{\Upsilon} \{\!\{ \mu k_2^{-1} \}\!\}} \{\!\{ \mathbf{e}_{\mathbf{u}_2} \}\!\} \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{14} \left\| \sqrt{\eta_u h_{\Upsilon} \{\!\{ \mu k_1^{-1} \}\!\}} \{\!\{ \boldsymbol{\eta}_{\mathbf{u}_1} \}\!\} \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{14}} \left\| \sqrt{\eta_u h_{\Upsilon} \{\!\{ \mu k_1^{-1} \}\!\}} \llbracket \mathbf{e}_{\mathbf{u}_1} \rrbracket \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{15} \left\| \sqrt{\frac{\eta_p}{h_{\Upsilon}} \{\!\{ \mu^{-1} k_1 \}\!\}} \llbracket \eta_{p_1} \rrbracket \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{15}} \left\| \sqrt{\frac{\eta_p}{h_{\Upsilon}} \{\!\{ \mu^{-1} k_1 \}\!\}} \llbracket e_{p_1} \rrbracket \right\|_{\Gamma^{\text{int}}}^2
\end{aligned}$$

²For convenience of the reader, we color-coded the terms. (See the online version for the colored text.)

The red-colored terms contain interpolation errors and contribute to $\|\mathbf{H}\|_{\text{stab}}^{\text{DG}}$. The blue-colored terms contain approximation errors and contribute to $\|\mathbf{E}\|_{\text{stab}}^{\text{DG}}$. We employ Lemma 4.2.1 on the magenta-colored terms and employ Lemma 4.2.2 on the green-colored terms.

$$\begin{aligned}
& + \varepsilon_{16} \left\| \sqrt{\eta_u h_\Upsilon \{\mu k_2^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_2}] \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{16}} \left\| \sqrt{\eta_u h_\Upsilon \{\mu k_2^{-1}\}} [\mathbf{e}_{\mathbf{u}_2}] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{17} \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\mu^{-1} k_2\}} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{17}} \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\mu^{-1} k_2\}} [e_{p_2}] \right\|_{\Gamma^{\text{int}}}^2,
\end{aligned} \tag{4.53}$$

with ε_i ($i = 1, \dots, 17$) are arbitrary positive constants. After employing Lemma 4.2.1, the above inequality can be grouped as

$$\begin{aligned}
2\mathcal{B}_{\text{stab}}^{\text{DG}}(\mathbf{H}; \mathbf{E}) & \leq \left(\frac{1}{2\varepsilon_1} + \frac{1}{2\varepsilon_3} + \frac{\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}}}{\varepsilon_{11}} \right) \left\| \sqrt{\frac{\mu}{k_1}} \mathbf{e}_{\mathbf{u}_1} \right\|^2 + \left(\frac{1}{2\varepsilon_2} + \frac{1}{2\varepsilon_4} \right) \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[e_{p_1}] \right\|^2 \\
& + \left(\frac{1}{2\varepsilon_5} + \frac{1}{2\varepsilon_7} + \frac{\mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}}}{\varepsilon_{13}} \right) \left\| \sqrt{\frac{\mu}{k_2}} \mathbf{e}_{\mathbf{u}_2} \right\|^2 + \left(\frac{1}{2\varepsilon_6} + \frac{1}{2\varepsilon_8} \right) \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[e_{p_2}] \right\|^2 \\
& + \frac{1}{\varepsilon_9} \left\| \sqrt{\frac{\beta}{\mu}} (e_{p_1} - e_{p_2}) \right\|^2 \\
& + \left(\frac{1}{\varepsilon_{10}} + \frac{1}{\varepsilon_{15}} \right) \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\mu^{-1} k_1\}} [e_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{14}} \left\| \sqrt{\eta_u h_\Upsilon \{\mu k_1^{-1}\}} [\mathbf{e}_{\mathbf{u}_1}] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \left(\frac{1}{\varepsilon_{12}} + \frac{1}{\varepsilon_{17}} \right) \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\mu^{-1} k_2\}} [e_{p_2}] \right\|_{\Gamma^{\text{int}}}^2 + \frac{1}{\varepsilon_{16}} \left\| \sqrt{\eta_u h_\Upsilon \{\mu k_2^{-1}\}} [\mathbf{e}_{\mathbf{u}_2}] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \left(\frac{\varepsilon_1 + \varepsilon_2}{2} \right) \left\| \sqrt{\frac{\mu}{k_1}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|^2 + \left(\frac{\varepsilon_3 + \varepsilon_4}{2} \right) \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[\eta_{p_1}] \right\|^2 \\
& + \left(\frac{\varepsilon_5 + \varepsilon_6}{2} \right) \left\| \sqrt{\frac{\mu}{k_2}} \boldsymbol{\eta}_{\mathbf{u}_2} \right\|^2 + \left(\frac{\varepsilon_7 + \varepsilon_8}{2} \right) \left\| \sqrt{\frac{k_2}{\mu}} \text{grad}[\eta_{p_2}] \right\|^2 \\
& + \varepsilon_9 \left\| \sqrt{\frac{\beta}{\mu}} (\eta_{p_1} - \eta_{p_2}) \right\|^2 \\
& + \varepsilon_{14} \left\| \sqrt{\eta_u h_\Upsilon \{\mu k_1^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_1}] \right\|_{\Gamma^{\text{int}}}^2 + \varepsilon_{15} \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\mu^{-1} k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{16} \left\| \sqrt{\eta_u h_\Upsilon \{\mu k_2^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_2}] \right\|_{\Gamma^{\text{int}}}^2 + \varepsilon_{17} \left\| \sqrt{\frac{\eta_p}{h_\Upsilon} \{\mu^{-1} k_2\}} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{10} \left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\mu k_1^{-1}\}} \{\boldsymbol{\eta}_{\mathbf{u}_1}\} \right\|_{\Gamma^{\text{int}}}^2 + \varepsilon_{11} \left\| \sqrt{h_\Upsilon^{-1} \{\mu^{-1} k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 \\
& + \varepsilon_{12} \left\| \sqrt{\frac{h_\Upsilon}{\eta_p} \{\mu k_2^{-1}\}} \{\boldsymbol{\eta}_{\mathbf{u}_2}\} \right\|_{\Gamma^{\text{int}}}^2 + \varepsilon_{13} \left\| \sqrt{h_\Upsilon^{-1} \{\mu^{-1} k_2\}} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2.
\end{aligned} \tag{4.54}$$

We choose the coefficients of the first nine terms (i.e., blue-colored terms) in such a way that these nine terms add up to the square of $\|\mathbf{E}\|_{\text{stab}}^{\text{DG}}$. This can be achieved by choosing these coefficients as:

$$\frac{1}{2\varepsilon_1} + \frac{1}{2\varepsilon_3} + \frac{\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}}}{\varepsilon_{11}} = \frac{1}{2\varepsilon_5} + \frac{1}{2\varepsilon_7} + \frac{\mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}}}{\varepsilon_{13}} = \frac{1}{2}, \quad \left(\frac{1}{2\varepsilon_2} + \frac{1}{2\varepsilon_4} \right) = \left(\frac{1}{2\varepsilon_6} + \frac{1}{2\varepsilon_8} \right) = \frac{1}{2},$$

$$\frac{1}{\varepsilon_9} = \frac{1}{\varepsilon_{14}} = \frac{1}{\varepsilon_{16}} = 1, \quad \text{and} \quad \left(\frac{1}{\varepsilon_{10}} + \frac{1}{\varepsilon_{15}} \right) = \left(\frac{1}{\varepsilon_{12}} + \frac{1}{\varepsilon_{17}} \right) = 1. \quad (4.55)$$

One way to satisfy the above constraints is to make the following choices for the individual constants:

$$\varepsilon_1 = \varepsilon_3 = \varepsilon_5 = \varepsilon_7 = 4, \quad \varepsilon_2 = \varepsilon_4 = \varepsilon_6 = \varepsilon_8 = \varepsilon_{10} = \varepsilon_{12} = \varepsilon_{15} = \varepsilon_{17} = 2, \quad \varepsilon_9 = \varepsilon_{14} = \varepsilon_{16} = 1,$$

$$\varepsilon_{11} = 4\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}} \quad \text{and} \quad \varepsilon_{13} = 4\mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}}. \quad (4.56)$$

By incorporating the above choices into inequality (4.54), we obtain

$$\begin{aligned} 2(\|\mathbf{E}\|_{\text{stab}}^{\text{DG}})^2 &\leq \left(\|\mathbf{E}\|_{\text{stab}}^{\text{DG}} \right)^2 + 3 \left\| \sqrt{\frac{\mu}{k_1}} \boldsymbol{\eta}_{\mathbf{u}_1} \right\|^2 + 3 \left\| \sqrt{\frac{k_1}{\mu}} \text{grad}[\eta_{p_1}] \right\|^2 \\ &\quad + 3 \left\| \sqrt{\frac{\mu}{k_2}} \boldsymbol{\eta}_{\mathbf{u}_2} \right\|^2 + 3 \left\| \frac{k_2}{\sqrt{\mu}} \text{grad}[\eta_{p_2}] \right\|^2 + \left\| \sqrt{\frac{\beta}{\mu}} (\eta_{p_1} - \eta_{p_2}) \right\|^2 \\ &\quad + 2 \left\| \sqrt{\frac{\eta_p}{h_\gamma}} \{\mu^{-1} k_1\} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 + 2 \left\| \sqrt{\frac{\eta_p}{h_\gamma}} \{\mu^{-1} k_2\}^{1/2} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2 \\ &\quad + \left\| \sqrt{\eta_u h_\gamma \{\mu k_1^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_1}] \right\|_{\Gamma^{\text{int}}}^2 + \left\| \sqrt{\eta_u h_\gamma \{\mu k_2^{-1}\}} [\boldsymbol{\eta}_{\mathbf{u}_2}] \right\|_{\Gamma^{\text{int}}}^2 \\ &\quad + 4\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}} \left\| \sqrt{h_\gamma^{-1} \{\mu^{-1} k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 + 4\mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}} \left\| \sqrt{h_\gamma^{-1} \{\mu^{-1} k_2\}} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2 \\ &\quad + 2 \left\| \sqrt{\frac{h_\gamma}{\eta_p} \{\mu k_1^{-1}\}} \{\boldsymbol{\eta}_{\mathbf{u}_1}\} \right\|_{\Gamma^{\text{int}}}^2 + 2 \left\| \sqrt{\frac{h_\gamma}{\eta_p} \{\mu k_2^{-1}\}} \{\boldsymbol{\eta}_{\mathbf{u}_2}\} \right\|_{\Gamma^{\text{int}}}^2 \\ &\leq \left(\|\mathbf{E}\|_{\text{stab}}^{\text{DG}} \right)^2 + 6(\|\mathbf{H}\|_{\text{stab}}^{\text{DG}})^2 \\ &\quad + 4\mathcal{C}_{\mathbf{e}_{\mathbf{u}_1}} \left\| \sqrt{h_\gamma^{-1} \{\mu^{-1} k_1\}} [\eta_{p_1}] \right\|_{\Gamma^{\text{int}}}^2 + 4\mathcal{C}_{\mathbf{e}_{\mathbf{u}_2}} \left\| \sqrt{h_\gamma^{-1} \{\mu^{-1} k_2\}} [\eta_{p_2}] \right\|_{\Gamma^{\text{int}}}^2 \\ &\quad + 2 \left\| \sqrt{\frac{h_\gamma}{\eta_p} \{\mu k_1^{-1}\}} \{\boldsymbol{\eta}_{\mathbf{u}_1}\} \right\|_{\Gamma^{\text{int}}}^2 + 2 \left\| \sqrt{\frac{h_\gamma}{\eta_p} \{\mu k_2^{-1}\}} \{\boldsymbol{\eta}_{\mathbf{u}_2}\} \right\|_{\Gamma^{\text{int}}}^2. \quad (4.57) \end{aligned}$$

Lemma 4.2.2 implies

$$\begin{aligned} (\|\mathbf{E}\|_{\text{stab}}^{\text{DG}})^2 &\lesssim 6(\|\mathbf{H}\|_{\text{stab}}^{\text{DG}})^2 + \sum_{\omega \in \mathcal{T}_h} \left(h_\omega^{2(p+1)} |\mathbf{u}_1|_{H^{p+1}(\omega)}^2 + h_\omega^{2(q+1)} |\mathbf{u}_2|_{H^{q+1}(\omega)}^2 \right. \\ &\quad \left. + h_\omega^{2r} |p_1|_{H^{r+1}(\omega)}^2 + h_\omega^{2s} |p_2|_{H^{s+1}(\omega)}^2 \right). \quad (4.58) \end{aligned}$$

As $h \rightarrow 0$, $h_\omega \rightarrow 0 \ \forall \omega \in \mathcal{T}_h$, which in turn implies that $\|\mathbf{H}\|_{\text{stab}}^{\text{DG}} \rightarrow 0$ (using Lemma 4.2.3) and all other terms on the right hand side tend to zero (using Lemma 4.2.2). Thus, $\|\mathbf{E}_{\text{stab}}^{\text{DG}}\| \rightarrow 0$ as $h \rightarrow 0$. Since $\|\cdot\|_{\text{stab}}^{\text{DG}}$ is a norm (i.e., Lemma 4.1.1), one can conclude that $\mathbf{U}^h \rightarrow \mathbf{U}$ almost everywhere as $h \rightarrow 0$.

Remark 4.2.1. *The selection of constants ε_i ($i = 1, \dots, 17$) in equation (4.56) is arbitrary. We do not claim that this selection provides an optimal bound, which is not the aim of this work. However, the selection is sufficient to establish the convergence of the proposed formulation.*

Lemmas 4.2.2 and 4.2.3 immediately give the following results:

Corollary 4.2.1. *(Rates of convergence.) Let p, q, r and s be the polynomial orders for approximating the fields $\mathbf{u}_1, \mathbf{u}_2, p_1$ and p_2 . Let the orders of regularity in terms of the Sobolev semi-norm for these solution fields be $\hat{p}, \hat{q}, \hat{r}$ and \hat{s} . Then the rates of convergence for these fields will be, respectively, $\min[p+1, \hat{p}]$, $\min[q+1, \hat{q}]$, $\min[r, \hat{r}]$ and $\min[s, \hat{s}]$.*

Remark 4.2.2. *In order for Lemma 4.2.2 to hold, $\eta_p \neq 0$, as η_p is in the denominator of the estimates (4.22) and (4.23). Since the convergence theorem utilizes Lemma 4.2.2, the convergence of the proposed DG formulation is thus established for the case $\eta_p \neq 0$. However, numerical simulations suggest that the parameters η_u and η_p do not seem to have a noticeable effect on the results for problems involving conforming meshes and conforming interpolations.*

Chapter 5

Numerical Analyses and Results

5.1 Patch tests

Patch tests are generally used to indicate the quality of a finite element. Despite some debated mathematical controversies regarding the patch test, “the patch test is the most practically useful technique for assessing element behavior” as nicely pinpointed by [90]. In this section, different constant flow patch tests are used to showcase various features of the proposed stabilized mixed DG formulation. First, the capability of the proposed formulation for modeling flow in a highly heterogeneous, layered porous domain with abrupt changes in macro- and micro-permeabilities, is shown. Then, the ability of the proposed stabilized mixed DG formulation for supporting non-conforming discretization, in the form of non-conforming order refinement and non-conforming element refinement, is assessed. Finally, the proposed stabilized mixed DG formulation is employed on meshes with non-constant Jacobian elements. For the case of non-conforming order refinement, a parametric study is performed to assess the sensitivity of the solutions with respect to the stabilization parameters η_u and η_p .

5.1.1 Velocity-driven patch test

In reality, heterogeneity of the material properties is indispensable when it comes to porous domains. In many geological systems, medium properties can vary by many orders of magnitude and rapid changes may occur over small spatial scales. The aim of this boundary value problem is to show that the proposed stabilized mixed DG formulation can perform satisfactorily when the medium properties are heterogeneous.

The heterogeneous domain consists of five horizontal layers with different macro- and micro-permeabilities in each layer. As shown in **Fig. 5.1**, on the left side of each layer, a constant normal velocity ($\mathbf{u}_i \cdot \hat{\mathbf{n}} = -\frac{k_i^{\# \text{ layer}}}{\mu}$) is applied and on the right side, $\mathbf{u}_i \cdot \hat{\mathbf{n}} = \frac{k_i^{\# \text{ layer}}}{\mu}$ is prescribed. On the top and bottom of the domain, normal components of macro- and micro-velocities are prescribed to be zero. For uniqueness of the solution, pressure is prescribed on one corner of the domain. Table 5.1 provides the model parameters for this problem.

Table 5.1: Model parameters for velocity-driven patch test.

Parameter	Value
$\gamma \mathbf{b}$	{0.0, 0.0}
L_x	5.0
L_y	4.0
μ	1.0
β	1.0
k	0.2
η_u	100.0
η_p	100.0
h	structured TRI mesh of size 0.04 used

As can be seen in **Fig. 5.2**, velocities are constant and pressures are linearly varying in the horizontal direction in each layer, which are in agreement with the exact solution of this problem as remarked by [92]. This problem is also solved using the stabilized

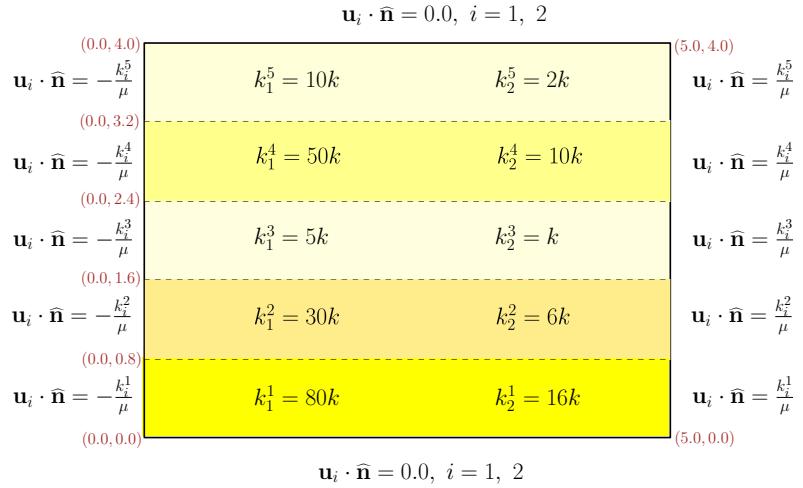


Figure 5.1: Velocity-driven patch test: This figure shows the computational domain, boundary conditions, and macro- and micro-permeabilities in each layer.

continuous Galerkin (CG) formulation of the DPP model developed by [96] and the x-components of velocity profiles are compared under both DG and CG formulations at $x=2.5$ throughout the domain as shown in **Fig. 5.3**. As can be seen, spurious oscillations are observed along the interfaces of the layers under the CG formulation. Under the DG formulation, however, such oscillations are completely eliminated and the physical jumps in the velocity profiles are accurately captured across the interfaces.

5.1.2 Non-conforming discretization

One of the features of DG formulations is that the global error of the computation can be controlled by adjusting the numerical resolution in a selected set of the elements. Such a *non-conforming discretization* can be obtained in two ways [83]: One can either modify the local order of the interpolation, or locally change the element size in parts of the computational domain. [16, 15] have discussed that the former method, also known as *non-conforming order refinement* or *non-conforming polynomial orders*, is more preferred for smooth problems. However, for the non-smooth case, which is due to the geometric

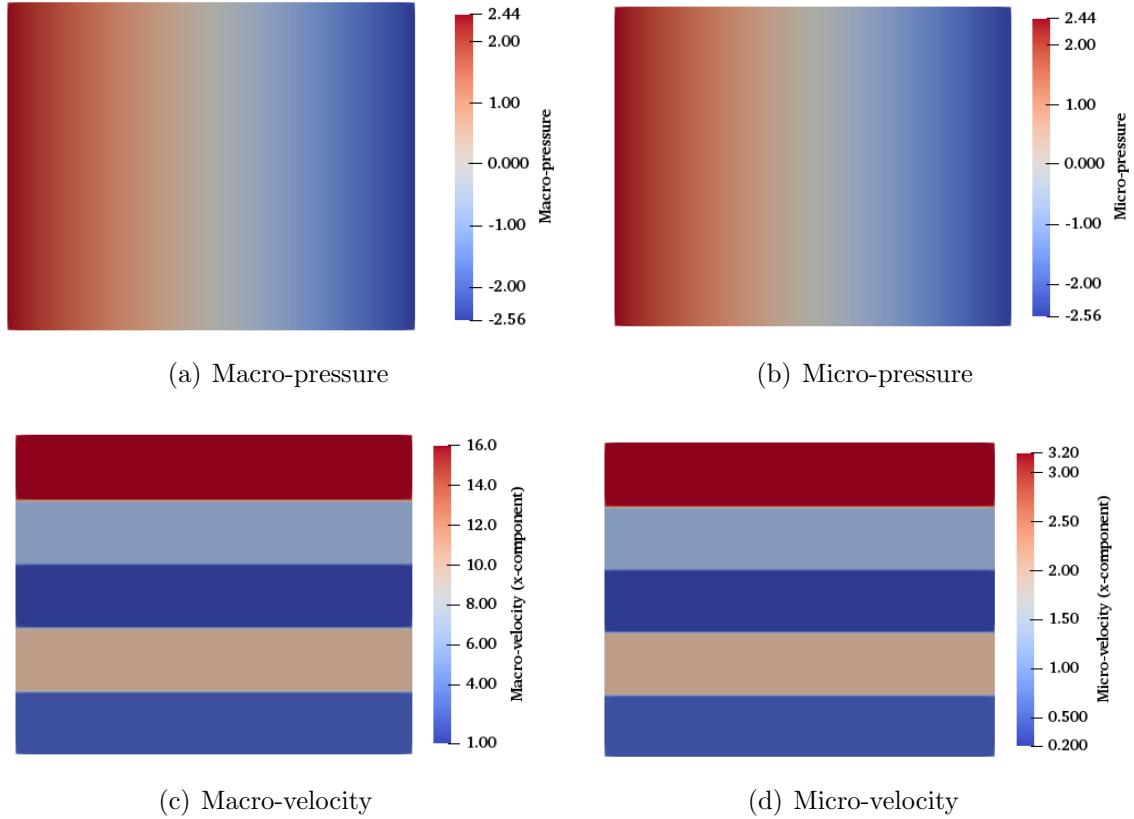


Figure 5.2: Velocity-driven patch test: Velocities are constant within each layer and pressures are linearly varying in the horizontal direction which are in agreement with the exact solution of this problem.

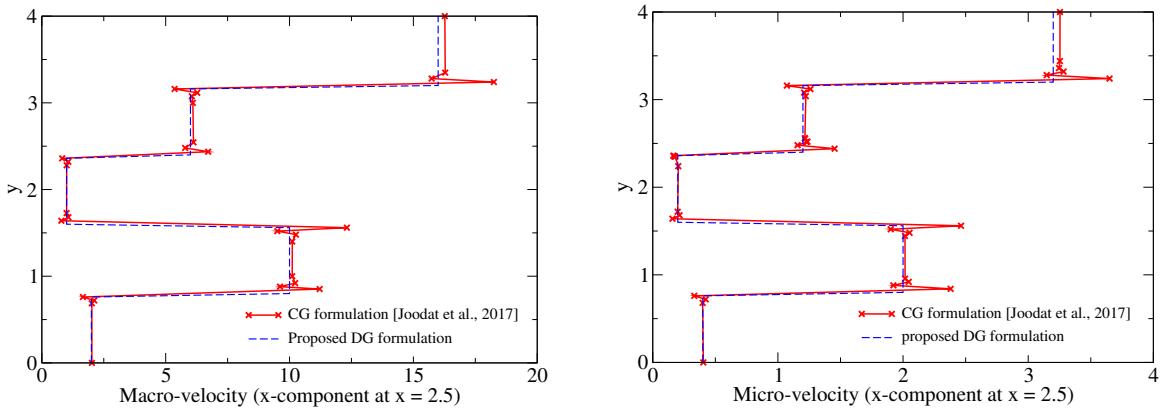


Figure 5.3: Velocity-driven patch test: This figure compares the velocities profiles obtained under the stabilized mixed CG formulation and the proposed DG formulation.

features, sources, or boundary conditions, *non-conforming element refinement* is the best choice. In the following, we show the application of non-conforming discretization under the proposed stabilized DG formulation using simply designed boundary value problems.

5.1.2.1 Non-conforming polynomial orders

Since the element communication under the DG formulations takes place through fluxes, each element can independently possess a desired order of interpolation. Hence, the DG methods can easily support the non-conforming polynomial orders (see [152, 42, 81]).

In order to investigate the performance of our proposed stabilized mixed DG formulation under non-conforming polynomial orders, a problem taken from [96] is used. The domain is considered to be a unit square, with pressures being prescribed on the entire boundary of both pore-networks as shown in **Fig. 5.4**. Prescribed pressure values on the respective boundary edges are obtained using the analytical solutions of this problem. The analytical solution for the pressure and velocity fields can be written as:

$$p_1(x, y) = \frac{\mu}{\pi} \exp(\pi x) \sin(\pi y) - \frac{\mu}{\beta k_1} \exp(\eta y), \quad (5.1)$$

$$p_2(x, y) = \frac{\mu}{\pi} \exp(\pi x) \sin(\pi y) + \frac{\mu}{\beta k_2} \exp(\eta y), \quad (5.2)$$

$$\mathbf{u}_1(x, y) = -k_1 \begin{pmatrix} \exp(\pi x) \sin(\pi y) \\ \exp(\pi x) \cos(\pi y) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\eta}{\beta} \exp(\eta y) \end{pmatrix}, \text{ and} \quad (5.3)$$

$$\mathbf{u}_2(x, y) = -k_2 \begin{pmatrix} \exp(\pi x) \sin(\pi y) \\ \exp(\pi x) \cos(\pi y) \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{\eta}{\beta} \exp(\eta y) \end{pmatrix}. \quad (5.4)$$

where

$$\eta := \sqrt{\beta \frac{k_1 + k_2}{k_1 k_2}}. \quad (5.5)$$

η is a useful parameter to characterize the flow of fluids through porous media with double porosity/permeability [135].

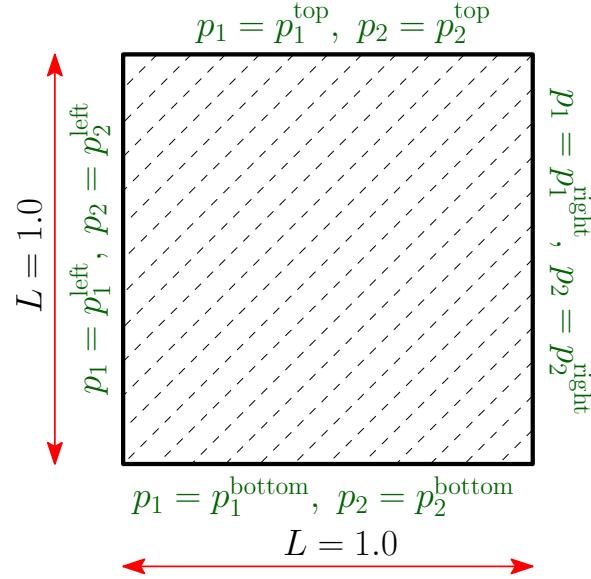


Figure 5.4: Non-conforming polynomial orders: The computational domain in the 2D setting is a unit square. Pressures are prescribed on the entire boundary of both pore-networks.

Table 5.2: Model parameters for non-conforming polynomial orders, element-wise mass balance study, and 2D numerical convergence analysis.

Parameter	Value
$\gamma \mathbf{b}$	{0.0, 0.0}
L	1.0
μ	1.0
β	1.0
k_1	1.0
k_2	0.1
η	$\sqrt{11} \simeq 3.3166$
η_u	10.0
η_p	1.0
h	structured TRI mesh of size 0.1 used
$p_i^{\text{left}}, i = 1, 2$	Obtained by evaluating
$p_i^{\text{right}}, i = 1, 2$	the analytical solution
$p_i^{\text{top}}, i = 1, 2$	(equations (5.1) and (5.2))
$p_i^{\text{bottom}}, i = 1, 2$	on the respective boundaries.

Table 5.2 provides the parameter values for this problem. In the left and right parts of the domain, two different sets of equal-order interpolation are employed for velocities

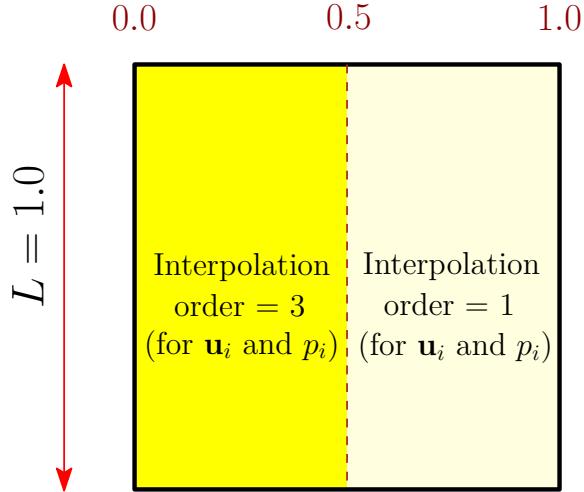


Figure 5.5: **Non-conforming polynomial orders:** In the left part of the domain, third order interpolation polynomials are used for velocities and pressures, while in the right part, first order interpolation polynomials are used.

and pressures as shown in **Fig. 5.5**. In the left half, third order interpolation polynomials are employed for velocities and pressures in each pore-network while in the right half, first-order interpolation polynomials are used.

Smooth velocity profiles along the non-conforming edge ($x = 0.5$) are not achievable for a coarse mesh (e.g., of size 10×10 elements mesh) without using extra stabilization terms (i.e., $\eta_u = \eta_p = 0$). One can either apply exhaustive mesh refinement, which in turn leads to a much higher computational cost, or can circumvent the unnecessary refinements by alternatively taking advantage of non-zero η_u and η_p . **Figs. 5.6–5.8** illustrate the sensitivity of x-component of velocities along the non-conforming edge with respect to η_u , η_p and their combined effect. According to **Figs. 5.6(a)** and **5.6(b)**, the increase in η_p per se in the absence of η_u slightly improves the results. However, for the case of $\eta_p = 0$ and non-zero η_u , a drastic enhancement is captured with η_u of order one as shown in **Figs. 5.7(a)** and **5.7(b)**. **Figs. 5.8(a)** and **5.8(b)** show the combined effect of η_u and η_p along the non-conforming edge in minimizing the drifts of macro and micro-velocity fields.

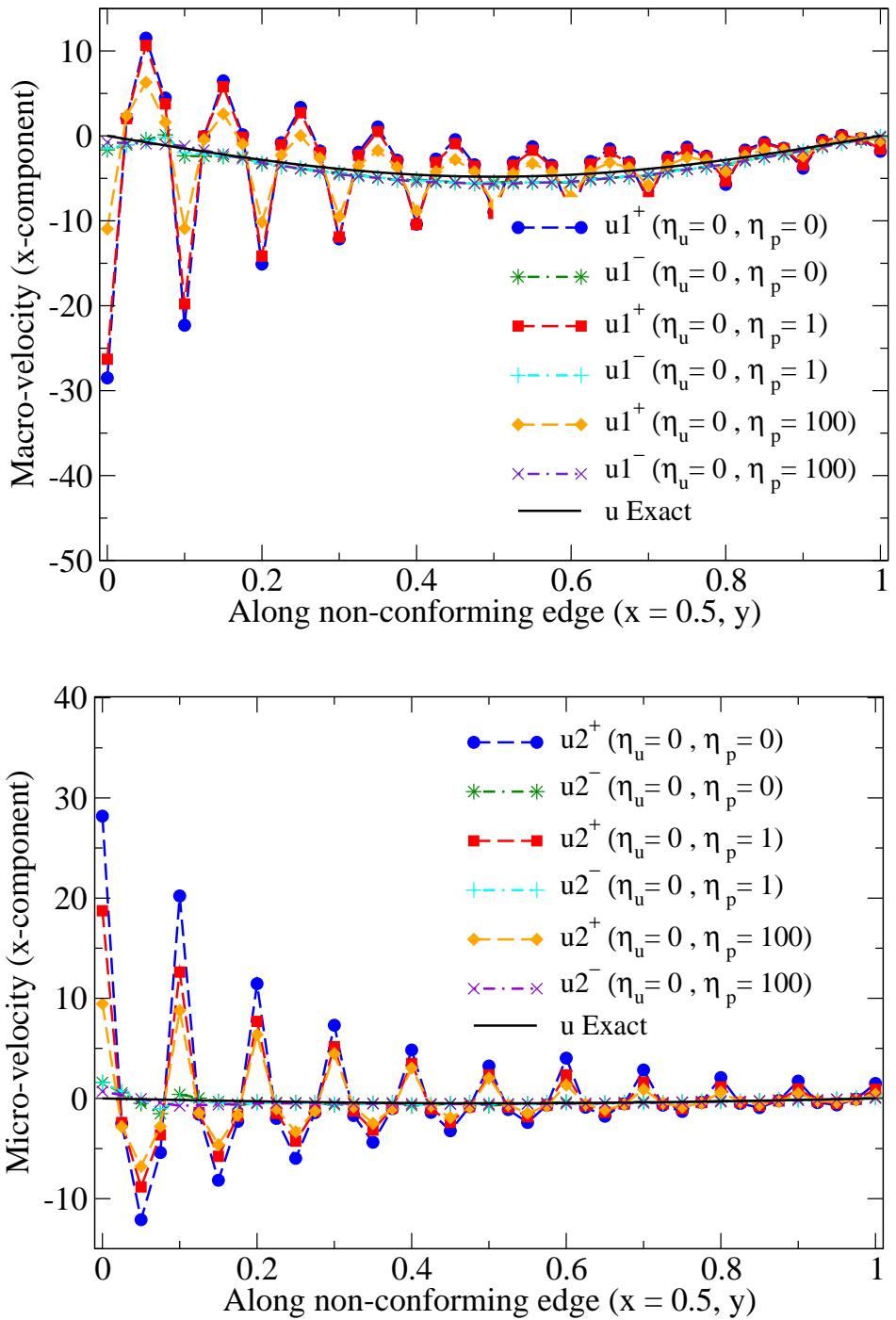


Figure 5.6: Non-conforming polynomial orders: This parametric study demonstrates that an increase in η_p in the absence of η_u slightly improves the accuracy in capturing the jumps of the (macro- and micro-) velocities across a non-conforming edge.

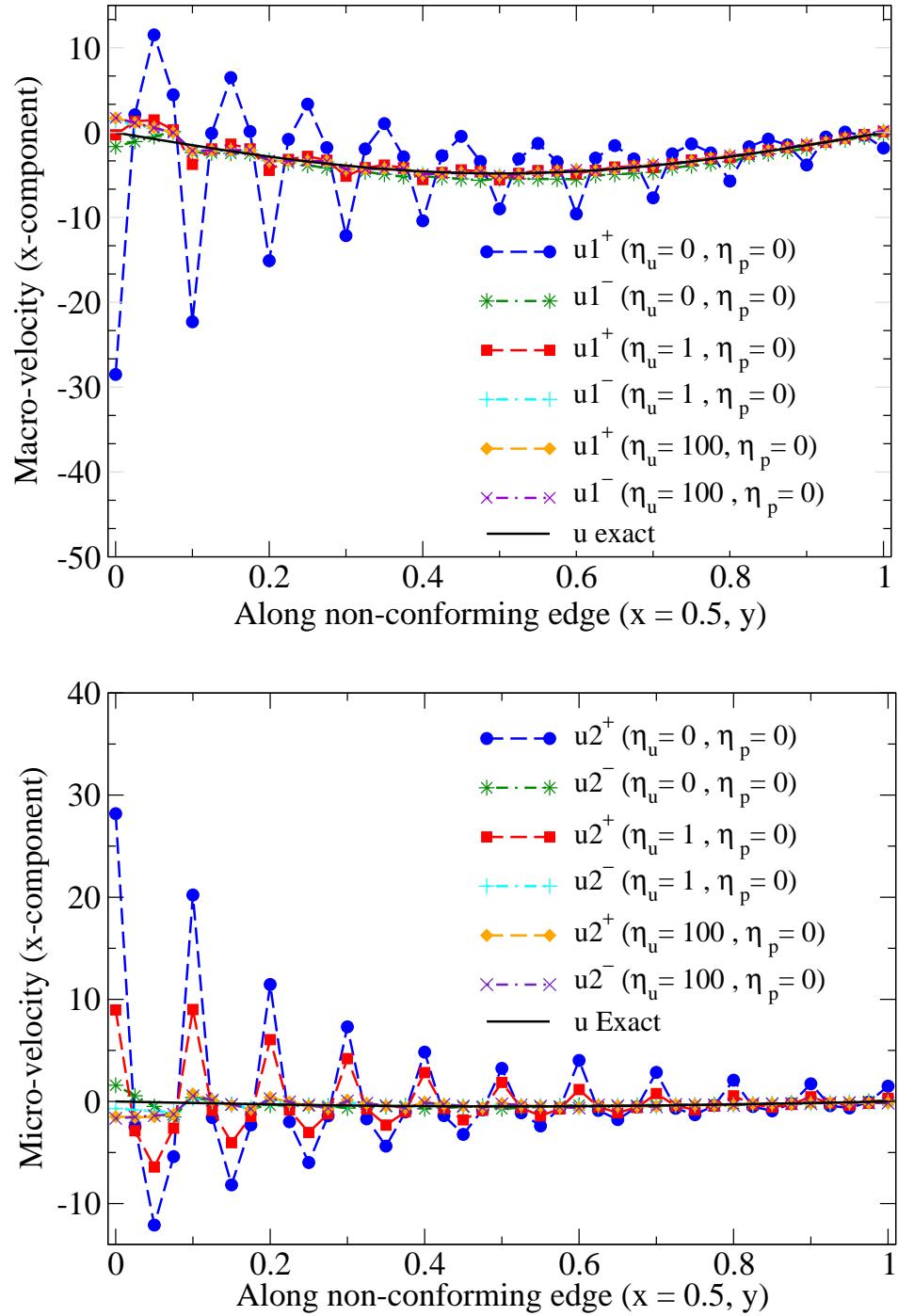


Figure 5.7: Non-conforming polynomial orders: This figure shows a parametric study performed on the effect of η_u on velocity profiles. For the case of $\eta_p = 0$ and non-zero η_u , a drastic enhancement is captured with η_u of order 1.

Figs. 5.9 and **5.10** compare the exact and numerical solutions for the pressure and velocity fields by taking $\eta_u = 10$ and $\eta_p = 1$. As can be seen, the numerical and the exact solutions match, which implies that the proposed mixed DG formulation can nicely handle non-conforming polynomial orders. [17] suggests the need for such additional stabilization terms for modeling flow under Darcy equations. However, to the best of the authors' knowledge, no numerical simulation has been reported to quantify the effect of these stabilization parameters on the accuracy of results under the DPP model for the problems exhibiting mismatching interpolation order.

5.1.2.2 Non-conforming element refinement

In mesh refinement procedures, one can either uphold the conformity of the mesh or produce irregular (non-conforming) meshes. The ability of DG formulations to support non-conforming elements obviates the user from propagating refinements beyond the desired elements [83]. The non-conforming meshes introduce *hanging nodes* on the edge of neighboring elements. In general, there are two strategies for handling non-matching interface discretization. In the first approach, extra degrees of freedom are assigned to the hanging nodes; hence the shape functions are generated on both regular and hanging nodes in such a way that both Kronecker delta and partition of unity properties are satisfied. Constructing these special shape functions for two- and three-dimensional problems is discussed in [77, 132]. In the second approach, which is known as constrained approximation, the shape functions are generated only on the corner nodes of each element and the stiffness matrix is assembled via conventional algorithms. The constraints at hanging nodes are then designed to be the average of their neighboring corner nodes. This can be enforced either through Lagrange multipliers or multiplication by the connectivity matrix [3, 20]. This method is a classical standard procedure in treatment of mismatching girds

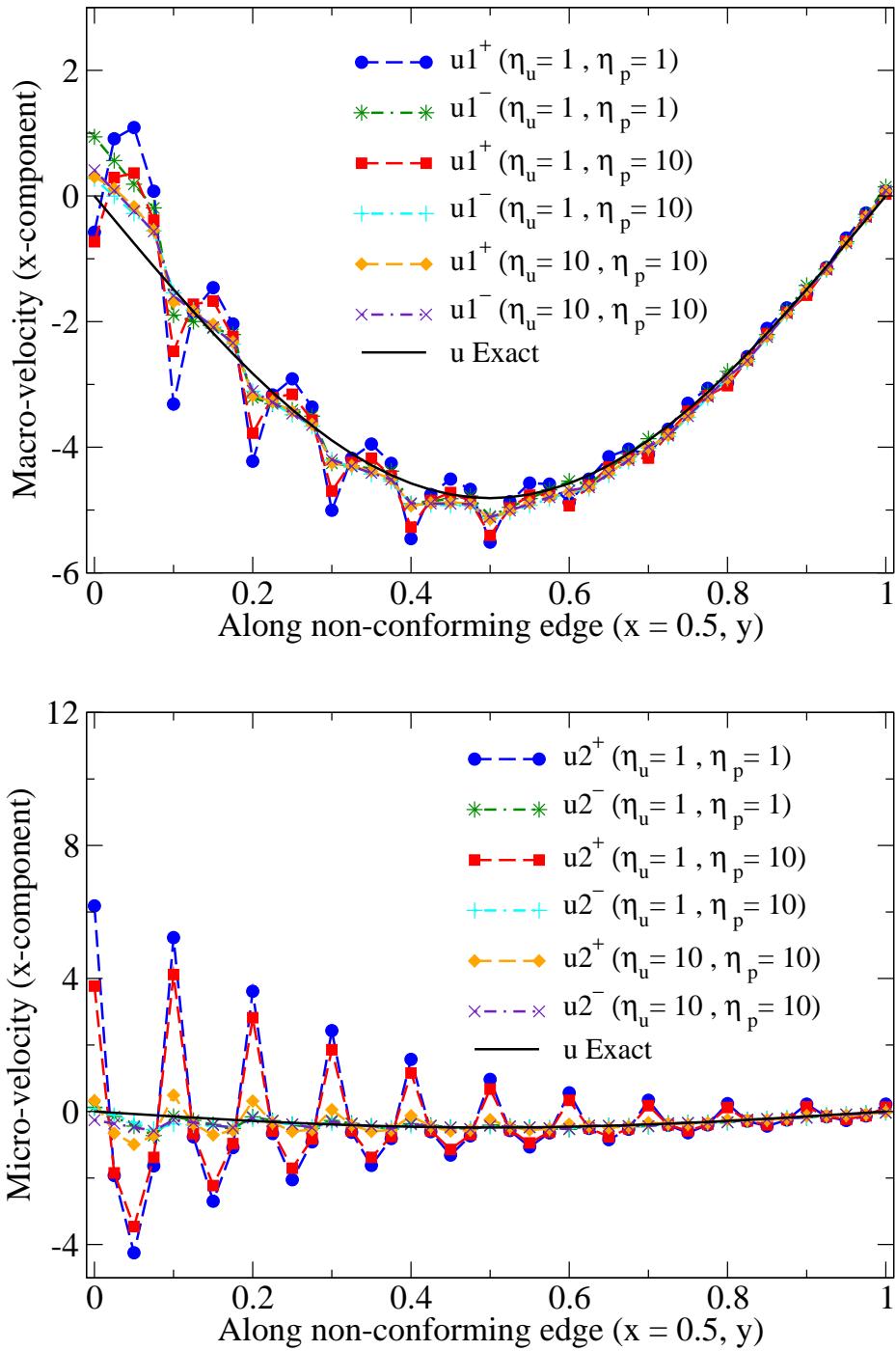


Figure 5.8: Non-conforming polynomial orders: This figure shows a parametric study performed on the combined effect of η_u and η_p on minimizing the drifts of macro and micro-velocity fields.

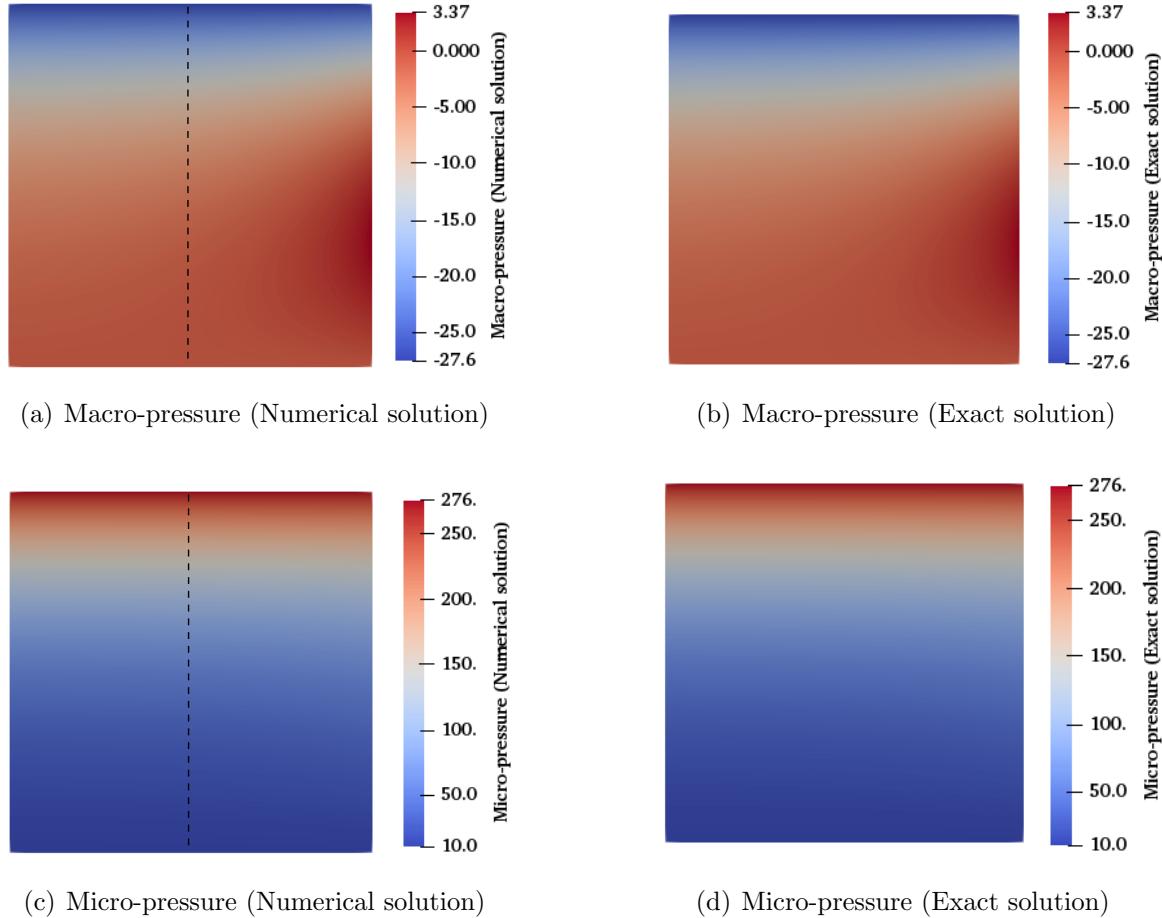


Figure 5.9: Non-conforming polynomial orders: In the left half of the domain, third order interpolation polynomials are used for velocities and pressures, while in the right half, first order interpolation polynomials are used.

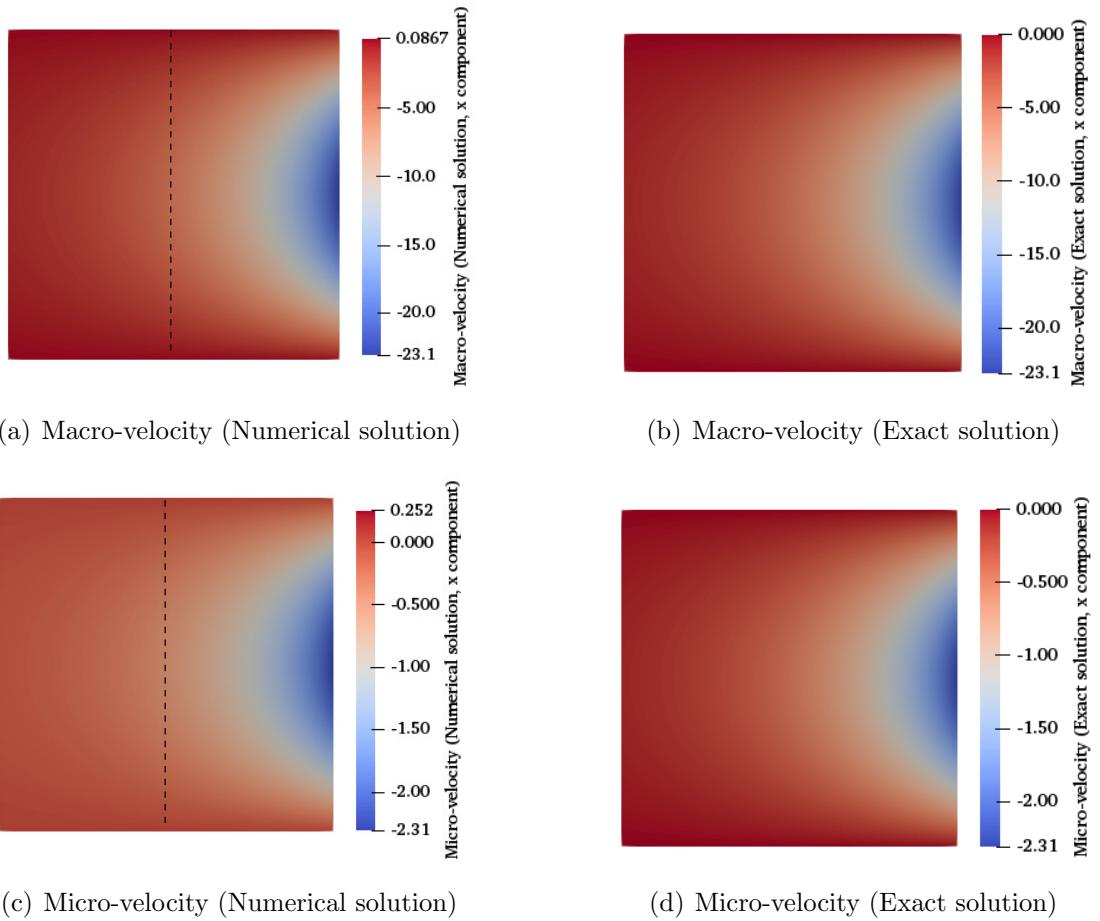


Figure 5.10: **Non-conforming polynomial orders:** This figure shows the exact and numerical solutions for the velocity profiles within the domain.

and hanging nodes. For further details refer to [141]. Herein, we resort to the second approach by introducing *virtual nodes*, as the refinement algorithm is more straightforward compared to the first approach [70].

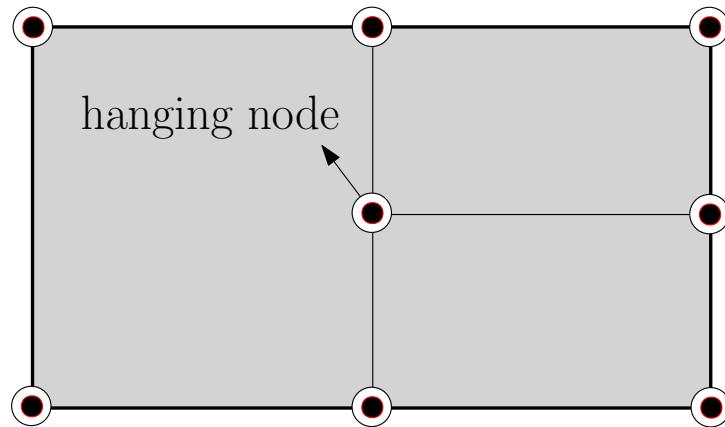
Applications of mesh refinement in the light of DG formulations are provided by [40, 104, 79], where the numerical fluxes on the non-conforming meshes are incorporated in the DG solver. In the following problem, the capability of our proposed stabilized mixed DG formulation for supporting the non-conforming element refinement is investigated. The domain is homogeneous with pressures being prescribed on the left and right boundaries of both pore-networks. The normal components of velocities are zero on top and bottom of the domain. The model parameters for this problem can be found in Table 5.3. The refinement provided is based on physical considerations and takes place on the right half of the domain, where the mismatching edge is shared by more than two elements, as can be seen in **Fig. 5.11(a)**.

The virtual nodes laid down on the non-conforming boundary face (nodes 13 and 14 in **Fig. 5.11(b)**), each store a linear interpolation of nodes 2 and 3. These nodes (similar to hanging nodes 8 and 9) do not initially impose any additional degrees of freedom and are merely auxiliary nodes on the edge of element 1 for programming convenience. The usual DG algorithm for the assembly of the global stiffness matrix is followed. Then, we enforce constraints for degrees of freedom corresponding to hanging nodes (and virtual nodes) by Lagrange multiplier's approach as described in details in [99, 70]. At this stage, the interactions of node 8 with nodes 2 and 3 was facilitated via virtual node 13, and similarly, the interaction of node 9 with nodes 2 and 3 was assisted via virtual node 14.

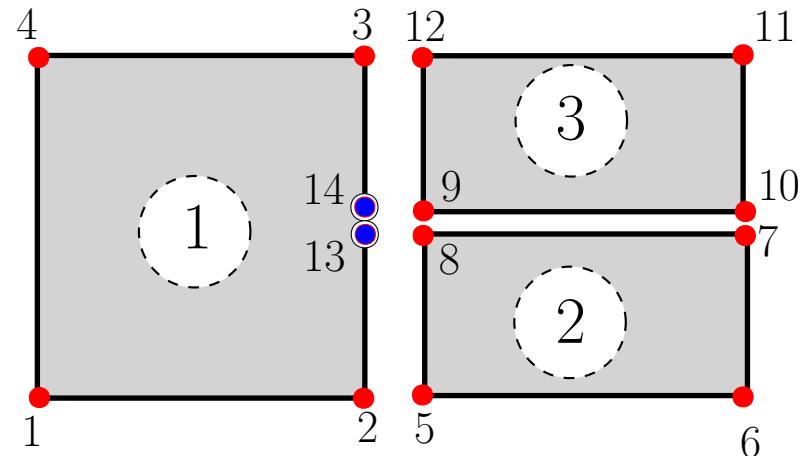
Fig. 5.12 shows the velocity and pressure profiles within the domain. Pressures in both pore-networks are varying linearly and velocities are constant throughout the domain. These results show that the proposed stabilized DG formulation is capable of handling

Table 5.3: Model parameters for non-conforming element refinement problem.

Parameter	Value
$\gamma \mathbf{b}$	{0.0, 0.0}
L_x	2.0
L_y	1.0
μ	1.0
β	1.0
k_1	1.0
k_2	0.1
η_u	0.0
η_p	0.0



(a) Computational domain



(b) Mesh discretization

Figure 5.11: Non-conforming element refinement: The top figure shows the representative computational domain with non-conforming element refinement. The bottom figure shows the DG discretization of this domain.

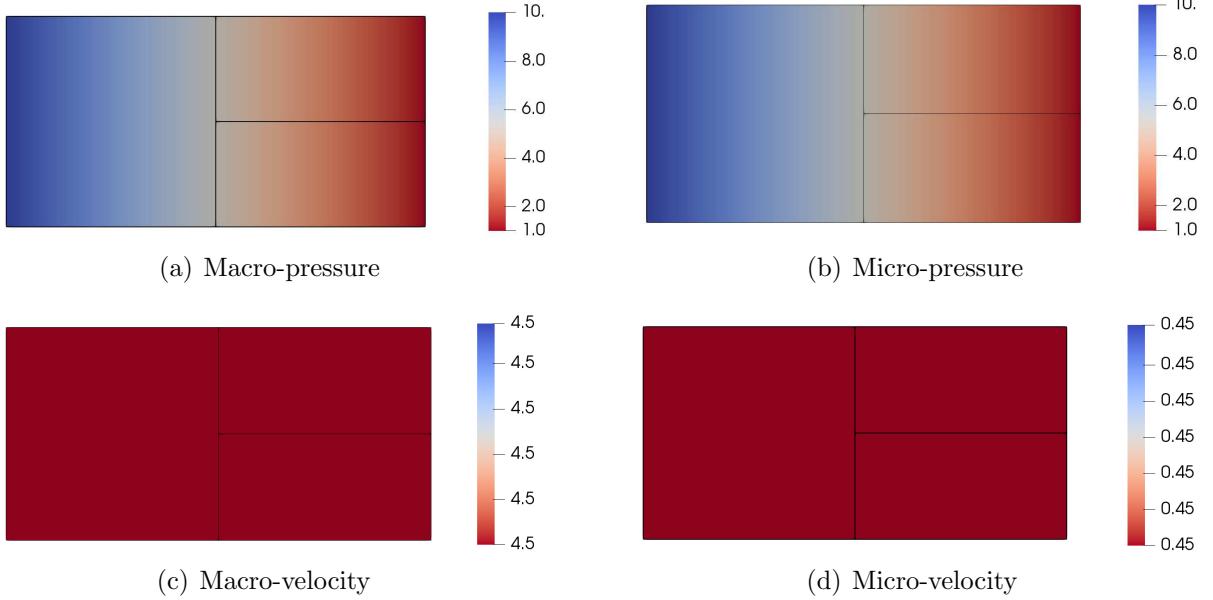
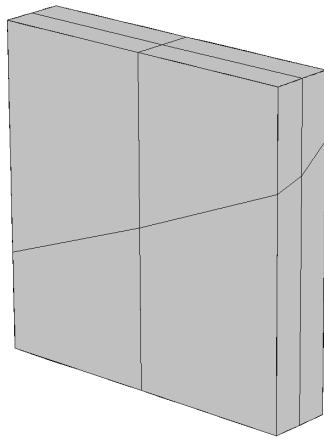


Figure 5.12: **Non-conforming element refinement:** These results show that the proposed stabilized DG formulation is capable of handling non-conforming element refinement (with hanging nodes in the mesh).

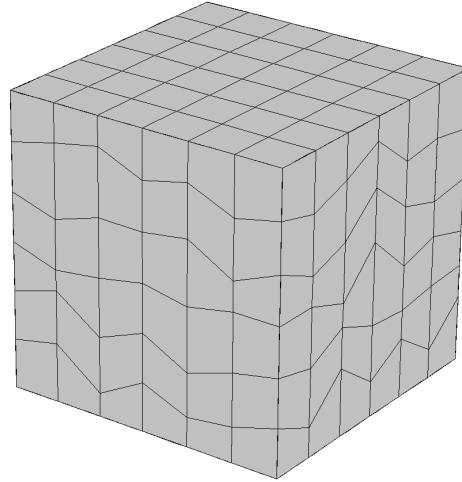
non-conforming element refinement (with hanging nodes in the mesh).

5.1.3 Non-constant Jacobian elements

In practice, many hydrogeological systems have complex shapes and modeling of such domains, especially in the 3D settings, requires using of elements with irregular shapes. Divergent boundaries in such elements result in non-constant Jacobian determinants. Herein, the aim is to show that the proposed stabilized mixed DG formulation can perform satisfactorily to model flow through computational domains composed of non-constant Jacobian elements. It will be shown that under the equal-order interpolation for the field variables, our proposed formulation is still able to pass the constant flow patch test with irregular elements. Two different computational domains with sample meshes having non-constant Jacobian brick elements are depicted in **Fig. 5.13** and model parameters are provided in Table 5.4. Pressures are prescribed at both left and right faces of the



(a) Mesh #1



(b) Mesh #2

Figure 5.13: Non-constant Jacobian elements: This figure shows two different computational domains and their corresponding meshes for the constant flow patch test. For this problem, non-constant Jacobian brick elements are used.

Table 5.4: Model parameters for 3D computational domains with non-constant Jacobian elements.

Parameter	Mesh #1	Mesh #2
$\gamma \mathbf{b}$	{0.0, 0.0, 0.0}	{0.0, 0.0, 0.0}
L_x	1.0	1.0
L_y	1.0	0.2
L_z	1.0	1.0
μ	1.0	1.0
β	1.0	1.0
k_1	1.0	1.0
k_2	0.1	0.1
η_u	0.0	0.0
η_p	0.0	0.0

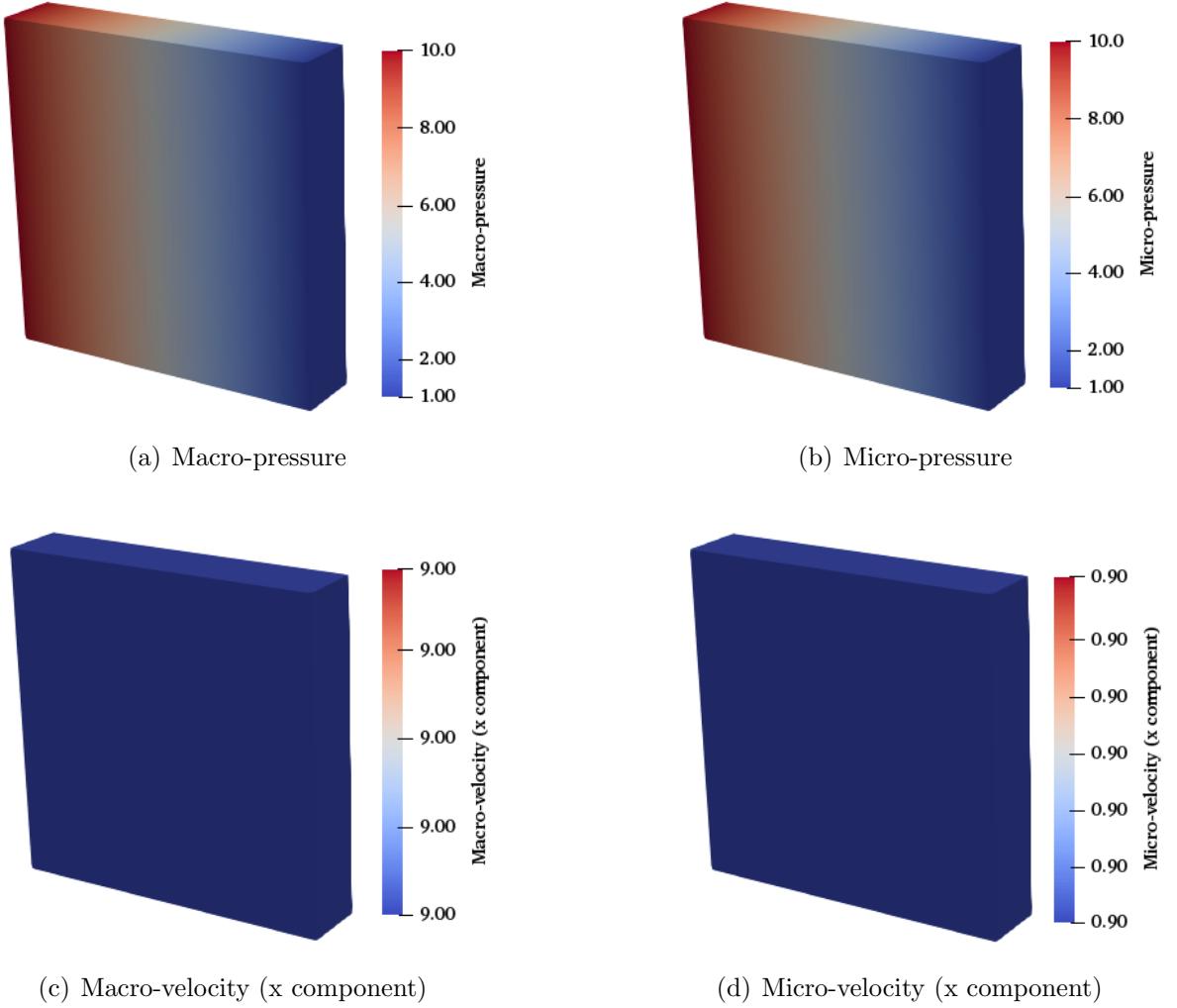


Figure 5.14: **Non-constant Jacobian elements:** Pressure and velocity profiles are shown for Mesh #1 (**Fig. 5.13(a)**) with non-constant Jacobian elements.

two pore-networks ($p_1(x = 0, y, z) = p_2(x = 0, y, z) = p^L$ and $p_1(x = 1, y, z) = p_2(x = 1, y, z) = p^R$). On the other faces, the normal component of velocity in both pore-networks is assumed to be zero (i.e., $\mathbf{u}_1 \cdot \hat{\mathbf{n}} = \mathbf{u}_2 \cdot \hat{\mathbf{n}} = 0$). The pressure and velocity profiles for both domains are shown in **Figs. 5.14** and **5.15**. In both domains, pressures are varying linearly from the left face to the right one and velocities are constant throughout the domain as expected. These results show that the proposed mixed DG formulation is capable of providing accurate results using non-constant Jacobian elements.

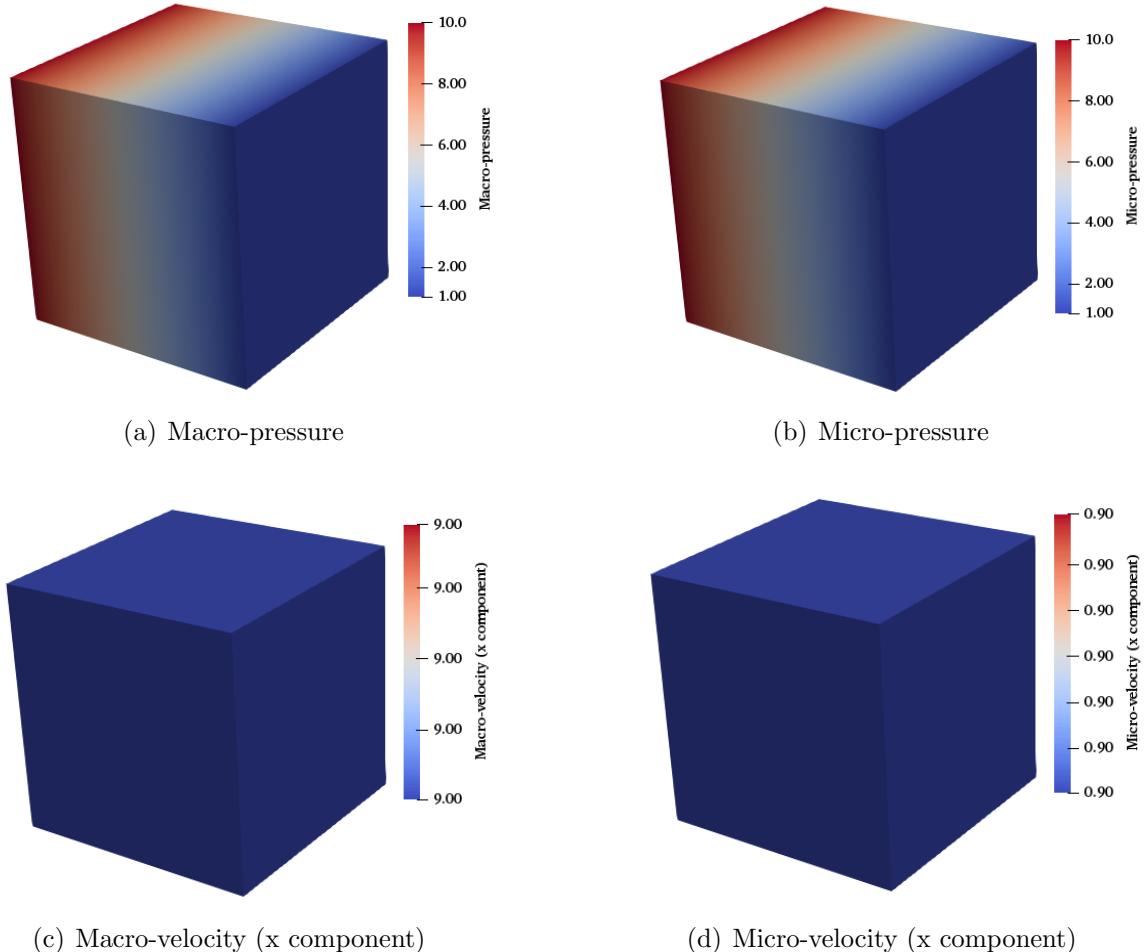


Figure 5.15: Non-constant Jacobian elements: Pressure and velocity contours are shown for Mesh #2 (Fig. 5.13(b)) with non-constant Jacobian elements.

5.2 Numerical convergence analysis

In this section, we perform numerical convergence analysis of the proposed stabilized DG formulation with respect to both h - and p -refinements.

5.2.1 2D numerical convergence analysis:

Convergence analysis in the 2D setting is performed on the boundary value problem described in Section 5.1.2.1. This problem earlier employed by [96] for the convergence analysis of the stabilized mixed *continuous* Galerkin (CG) formulation of the DPP model on a triangular (TRI) mesh. The exact solutions for the pressures and velocities are provided by equations (5.1) – (5.4). The domain for this problem is homogeneous (macro- and micro-permeabilities are constant within the domain). The computational domain is shown in **Fig. 5.4** and the parameter values are provided in Table 5.2.

The types of meshes used in this study are shown in **Fig. 5.16**. The three-node triangular element (TRI) (which is a simplicial mesh), four-node square element (QUAD) and four-node trapezoidal mesh [8] (which are non-simplicial meshes) are employed in the numerical simulation and the convergence is obtained under *h-refinement*. We performed h-convergence study on equal-order interpolations of order 1 to 4, and *p-refinement* of up to order 8, on TRI mesh. In **Figs. 5.17** and **5.18**, the convergence rates under *h-refinement* and *p-refinement* are provided for the L_2 -norm and the H^1 -norm of the pressure fields in the two pore-networks. The rates of convergence under *h*- and *p*-refinements are observed to be polynomial and exponential, respectively.

On non-simplicial meshes we performed two h-convergence analyses. In case I, equal-order interpolations of $p = q = r = s = 2$; and in case II, interpolation order of $p = q = 1$

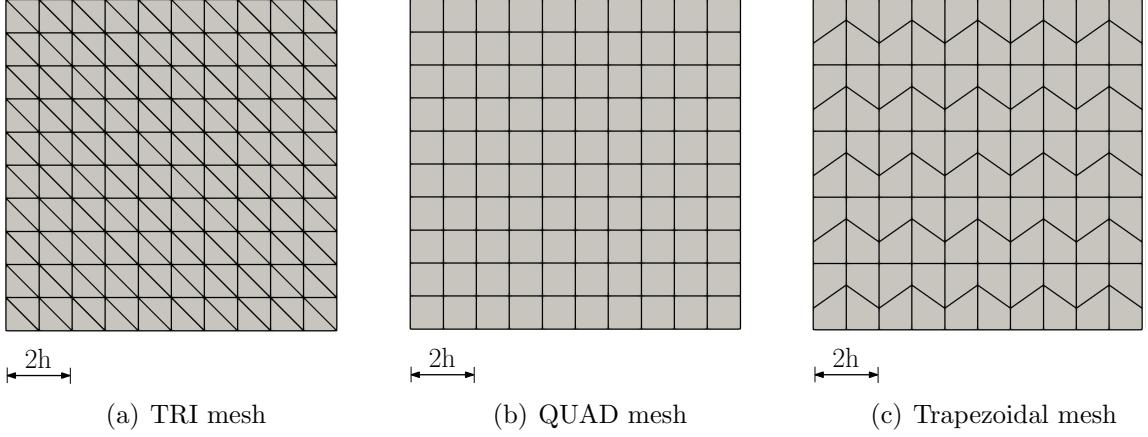


Figure 5.16: Schematic of mesh types for convergence analysis: This figure provides a pictorial description of different types of mesh used in convergence study. The size of h is taken as 0.1 in the figure.

and $r = s = 2$ were considered. In **Figs. 5.19** and **5.20**, the convergence rates under h -refinement for QUAD and trapezoidal meshes are provided for the L_2 -norm and the H^1 -norm of the pressure fields, and the L_2 -norm and the $Hdiv$ -norm of the velocity fields. The rates of convergence are observed to be polynomial. And for L_2 - and H^1 -norms, it is shown that both interpolation cases lead to the same rate of convergence on pressure and velocity fields for both pore networks. These results are in accordance with corollary 4.2.1 and equation (4.58). However, for $Hdiv$ -norm, the rate of convergence is sensitive to the choice of interpolation order for the velocity fields.

5.2.2 3D numerical convergence analysis

The computational domain of this problem is a unit cube with pressure being prescribed on the entire boundary of the two pore-networks. The analytical solution takes the following form:

$$p_1(x, y, z) = \frac{\mu}{\pi} \exp(\pi x) (\sin(\pi y) + \sin(\pi z)) - \frac{\mu}{\beta k_1} (\exp(\eta y) + \exp(\eta z)), \quad (5.6)$$

$$p_2(x, y, z) = \frac{\mu}{\pi} \exp(\pi x) (\sin(\pi y) + \sin(\pi z)) + \frac{\mu}{\beta k_2} (\exp(\eta y) + \exp(\eta z)), \quad (5.7)$$

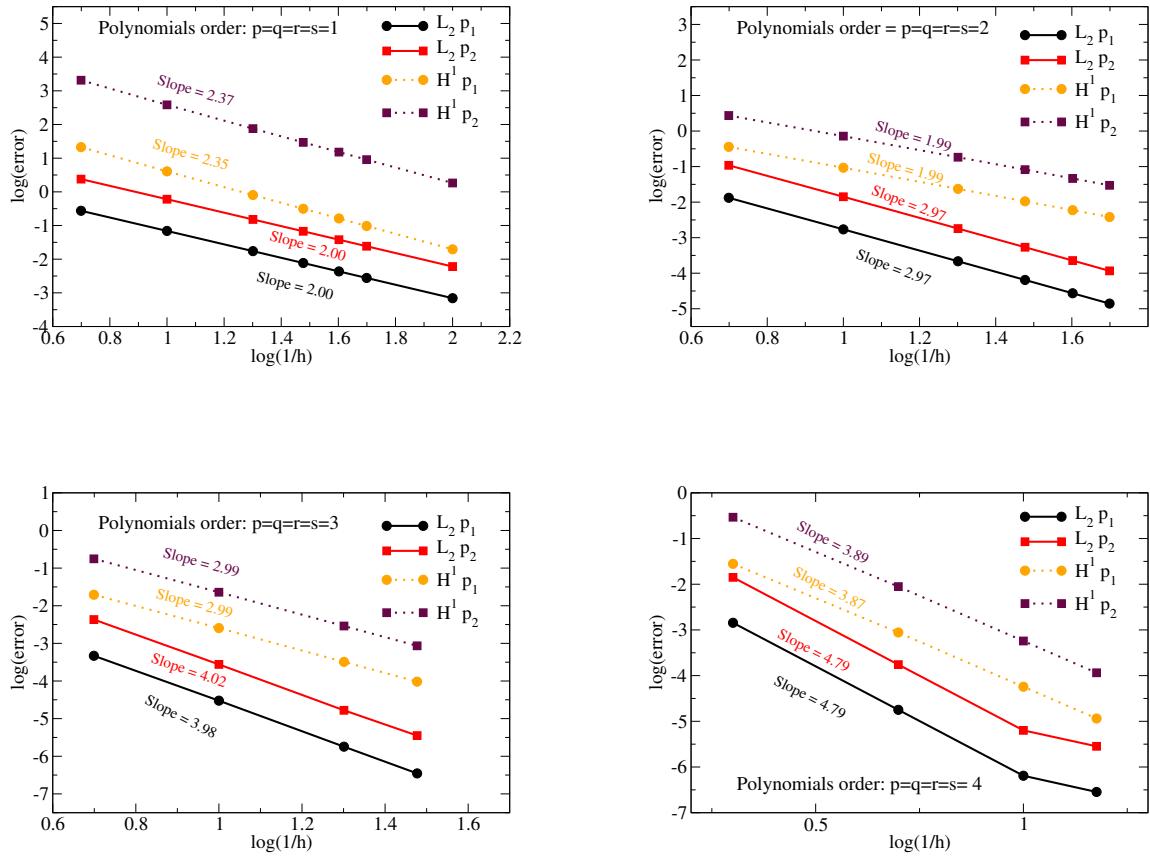


Figure 5.17: 2D numerical convergence analysis on TRI mesh: This figure provides the convergence rates under h -refinement for various equal-order polynomial on triangular mesh.

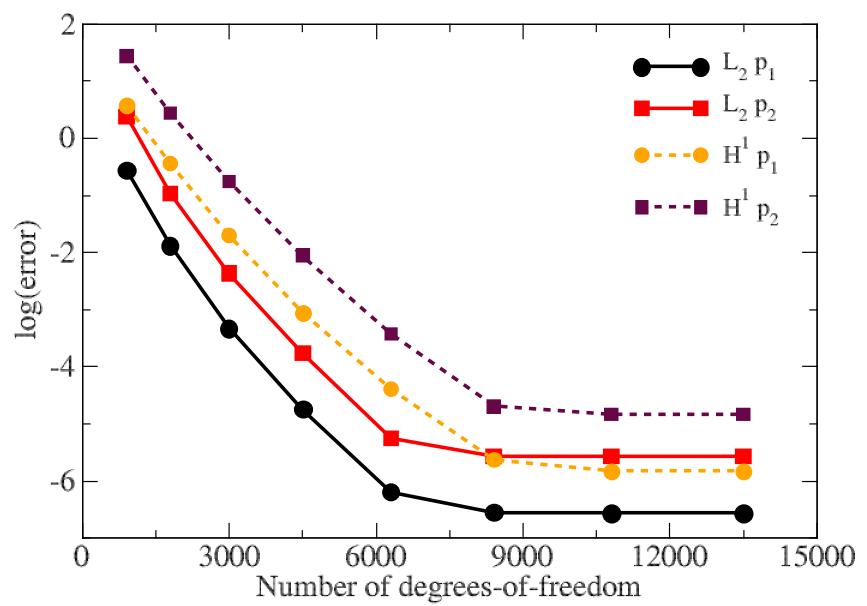


Figure 5.18: 2D numerical convergence analysis on TRI mesh: This figure shows the results of numerical convergence under p -refinement for a fixed mesh size ($h = 0.2$). The number of degrees-of-freedom corresponds to $p = 1$ to 8.

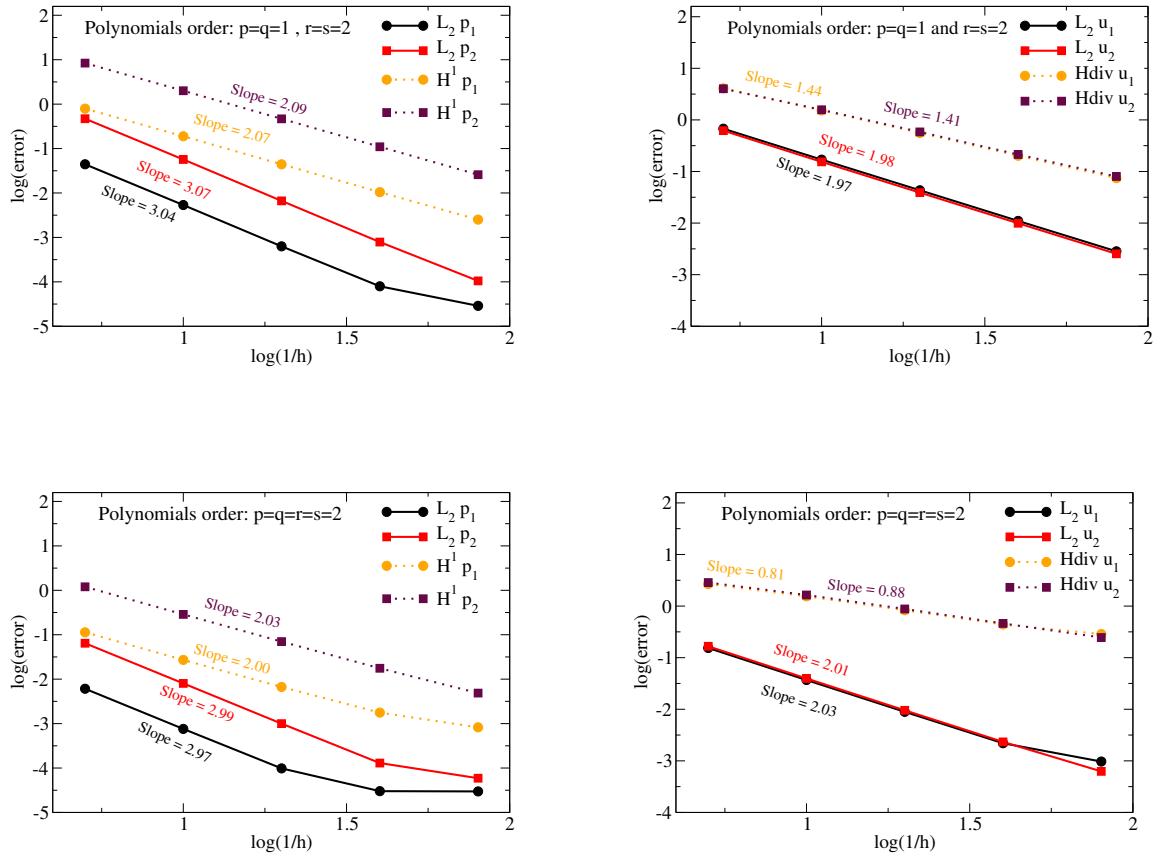


Figure 5.19: 2D numerical convergence analysis for QUAD mesh: This figure provides the convergence rates under h -refinement for two sets of polynomials order of: 1 and 2, respectively, for velocity and pressure fields; and 2 for both velocity and pressure fields in the two pore-networks.

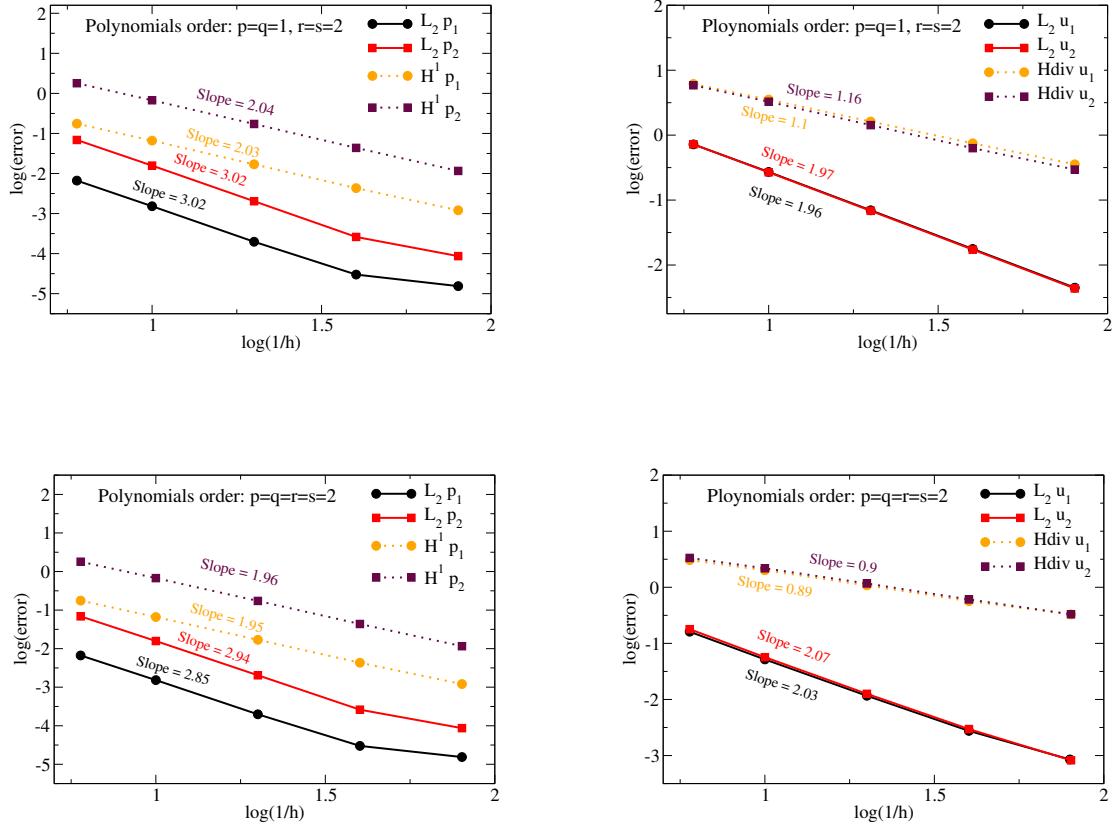


Figure 5.20: 2D numerical convergence analysis for trapezoidal mesh: This figure provides the convergence rates under h -refinement for two sets of polynomials order of: 1 and 2, respectively, for velocity and pressure; and 2 for both velocity and pressure fields in the two pore-networks.

Table 5.5: Model parameters for 3D numerical convergence analysis.

Parameter	Value
$\gamma \mathbf{b}$	{0.0, 0.0, 0.0}
L_x	1.0
L_y	1.0
μ	1.0
β	1.0
k_1	1.0
k_2	0.1
η	$\sqrt{11} \simeq 3.3166$
η_u	100.0
η_p	0.0
$p_i^{\text{left}}, i = 1, 2$	Obtained by evaluating
$p_i^{\text{right}}, i = 1, 2$	the analytical solution
$p_i^{\text{top}}, i = 1, 2$	(equations (5.6) and (5.7))
$p_i^{\text{bottom}}, i = 1, 2$	on the respective boundaries.

$$\mathbf{u}_1(x, y, z) = -k_1 \exp(\pi x) \begin{pmatrix} \sin(\pi y) + \sin(\pi z) \\ \cos(\pi y) \\ \cos(\pi z) \end{pmatrix} + \frac{\eta}{\beta} \begin{pmatrix} 0 \\ \exp(\eta y) \\ \exp(\eta z) \end{pmatrix}, \text{ and} \quad (5.8)$$

$$\mathbf{u}_2(x, y, z) = -k_2 \exp(\pi x) \begin{pmatrix} \sin(\pi y) + \sin(\pi z) \\ \cos(\pi y) \\ \cos(\pi z) \end{pmatrix} - \frac{\eta}{\beta} \begin{pmatrix} 0 \\ \exp(\eta y) \\ \exp(\eta z) \end{pmatrix}. \quad (5.9)$$

Pressure boundary conditions on each face are obtained by evaluating the analytical solution on the corresponding boundary of each pore-network. Table 5.5 provides the parameter values employed in the numerical simulation.

The eight-node brick element HEX, which is a non-simplicial element, is employed in this numerical simulation. **Figs. 5.21** and **5.22** respectively provide the convergence rates under h -refinement and p -refinement for the L_2 -norm and the H^1 -norm of the pressure fields in the two pore-networks. As can be seen, the rates of convergence under the h - and p -refinements are polynomial and exponential, respectively; which are in accordance with the theory (viz. Corollary 4.2.1).

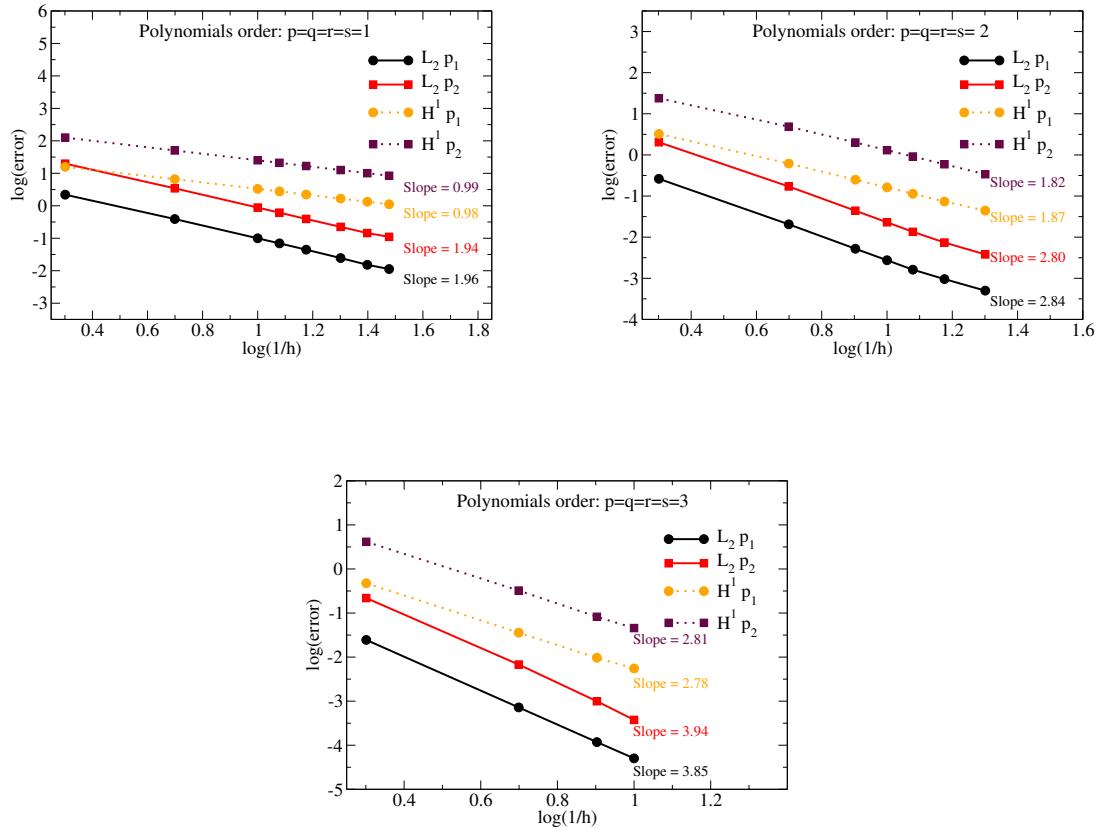


Figure 5.21: 3D numerical convergence analysis on TRI mesh: This figure provides the convergence rates under h -refinement for various equal-order polynomials. The rate of convergence is polynomial, which is in accordance with the theory.

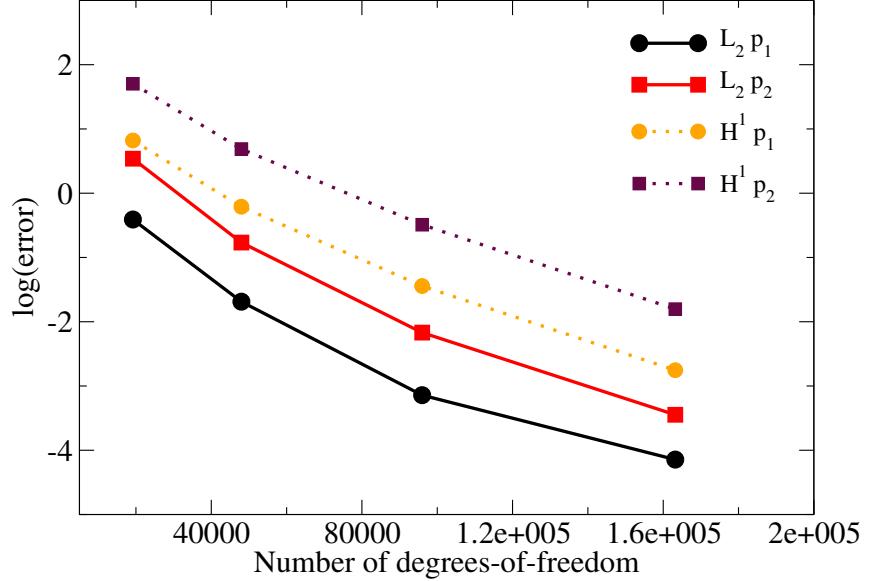


Figure 5.22: 3D numerical convergence analysis on TRI mesh: This figure shows the results of numerical convergence under p -refinement for a fixed mesh size ($h = 0.2$). The number of degrees-of-freedom corresponds to $p = 1$ to 4.

5.3 Canonical problem and structure preserving properties

In this section, first, robustness of the proposed stabilized mixed DG formulation is assessed using a standard test problem, with abrupt changes in material properties and elliptic singularities. In the literature, this problem is typically referred to as the quarter five-spot checkerboard problem. Second, the element-wise mass balance property associated with the CG and DG formulations is compared.

5.3.1 Quarter five-spot checkerboard problem

The original form of this problem, known as “five-spot problem” with homogeneous properties, has been firstly designed for the Darcy equations. Herein, we extend this problem to the DPP model with modified boundary conditions and heterogeneous medium properties. **Fig. 5.23** shows the computational domain and the boundary conditions for the five-spot problem. An injection well surrounded by four production wells placed at four corners of a square domain form a typical setting in the enhanced oil recovery applications. The underlying symmetry allows for solving the problem only in the top right quadrant, which is referred to as a “quarter” five-spot problem. In the well-known “checkerboard problem”, such a computational domain is divided into four sub-regions I, II, III, and IV with abrupt changes in the permeability.

In this problem, elliptic singularities are observed near the injection and production wells which are located at the opposite corners of the diagonals (denoted by C_{inj} and C_{prod} , respectively). The normal component of velocity is prescribed to be zero on the entire boundary of the micro-pore network. In the macro-network, however, velocity at the injection and production wells is prescribed by applying a source/sink term while zero normal velocity is assumed on the rest of the boundary. It is worth mentioning that the prescribed source and sink strengths at injection and production wells are, respectively, equal to +1 and -1. However, instead of applying a pointwise sink/source at the location of wells, the normal component of velocity is applied along the external edges of the corner element in x - and y -directions with an equivalent distribution as shown in **Fig. 5.23**.

Table 5.6 provides the parameter values for this problem. The permeability parameters in sub-regions I and IV and the ones in sub-regions II and III are mutually equal. Herein, we assume that sub-regions I and IV are more permeable compared to sub-regions

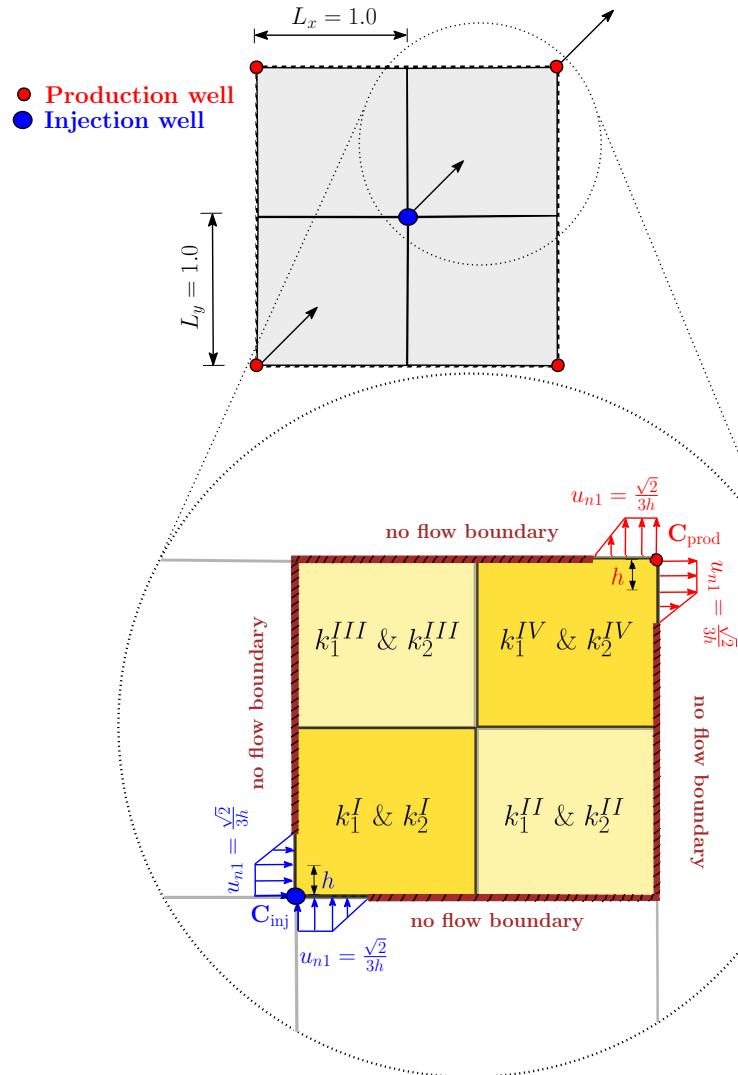


Figure 5.23: **Quarter five-spot checkerboard problem:** This figure shows the computational domain and boundary conditions. The heterogeneous domain is divided into four sub-regions with permeabilities shown in equation (5.10).

II and III with the following drag coefficients:

$$\begin{aligned} \left(\frac{\mu}{k_1}\right)_I &= \left(\frac{\mu}{k_1}\right)_{IV} = 1, & \left(\frac{\mu}{k_1}\right)_{II} &= \left(\frac{\mu}{k_1}\right)_{III} = 100, \\ \left(\frac{\mu}{k_2}\right)_I &= \left(\frac{\mu}{k_2}\right)_{IV} = 10, & \text{and} & \quad \left(\frac{\mu}{k_2}\right)_{II} = \left(\frac{\mu}{k_2}\right)_{III} = 1000. \end{aligned} \quad (5.10)$$

Table 5.6: Model parameters for the quarter five-spot checkerboard problem.

Parameter	Value
$\gamma\mathbf{b}$	{0.0, 0.0}
L_x	1.0
L_y	1.0
μ	1.0
β	1.0
k_1, k_2	refer to Eqn. (5.10)
u_{n1}	0.0 On $\partial\Omega - \{C_{\text{prod}} \& C_{\text{inj}}\}$
u_{n2}	0.0 On $\partial\Omega$
source and sink strength	-1 at C_{prod} +1 at C_{inj}
η_u	0, 10, 100
η_p	0, 10, 100
h	structured TRI mesh of size 0.01 used

Fig. 5.24 shows the macro- and micro pressure profiles for this problem. Steep gradients near the injection and production wells with no spurious oscillation in the pressure fields are observed under the proposed DG formulation which confirm the robustness of the numerical formulation. In order to further explore the effect of stabilization parameters on the solution profiles, this problem has been solved for different combinations of η_u and η_p as shown in **Fig. 5.25**. As can be seen, η_u and η_p have no noticeable effect on x-component of velocities under the DG formulation. However, spurious oscillations are observed under the CG formulation at the interface of sub-regions with different permeability values which implies that CG formulations fall short in capturing material discontinuities.

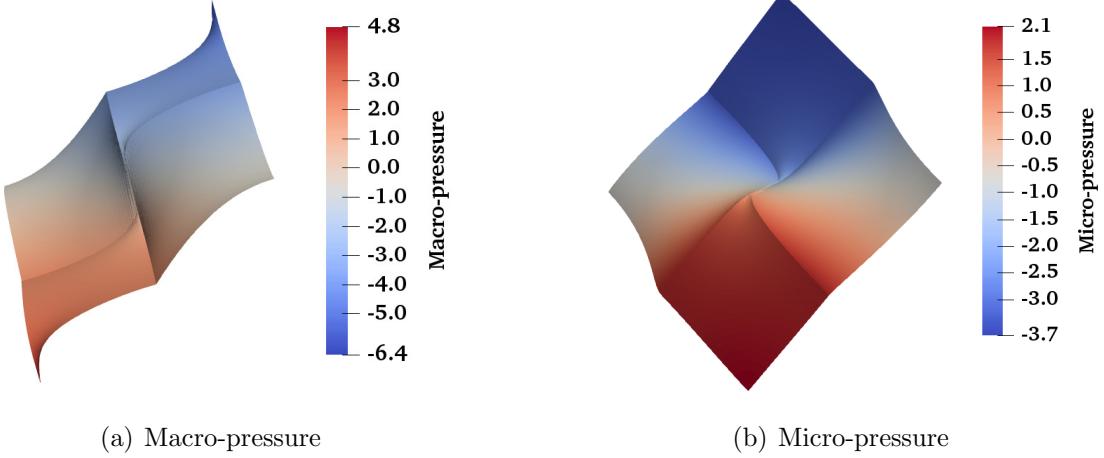


Figure 5.24: **Quarter five-spot checkerboard problem:** This figure shows that steep pressure gradients near the injection and production wells are correctly captured under the proposed DG formulation. Results are obtained for $\eta_u = \eta_p = 0$.

5.3.2 Element-wise mass balance

A DG method, when designed properly, can exhibit superior element-wise properties compared to its continuous counterpart. CG formulations may suffer from poor element-wise conservation; however, they satisfy a global mass balance [91]. The importance of element-wise mass balance in subsurface modeling is discussed in [171], which is particularly true when the flow is coupled with transport and/or chemical reactions.

In this section, element-wise mass balance error is investigated under the proposed stabilized mixed DG formulation for the DPP model, and the results are compared with its continuous counterpart. In the context of DPP, the net rate of volumetric flux from both pore-networks can be obtained as follows for an element $\omega \in \mathcal{T}_h$:

$$m(\omega) := \int_{\partial\omega} (\mathbf{u}_1 + \mathbf{u}_2) \cdot \hat{\mathbf{n}} \, d\Gamma. \quad (5.11)$$

After calculation, this equation should result in a zero value. The maximum element-wise

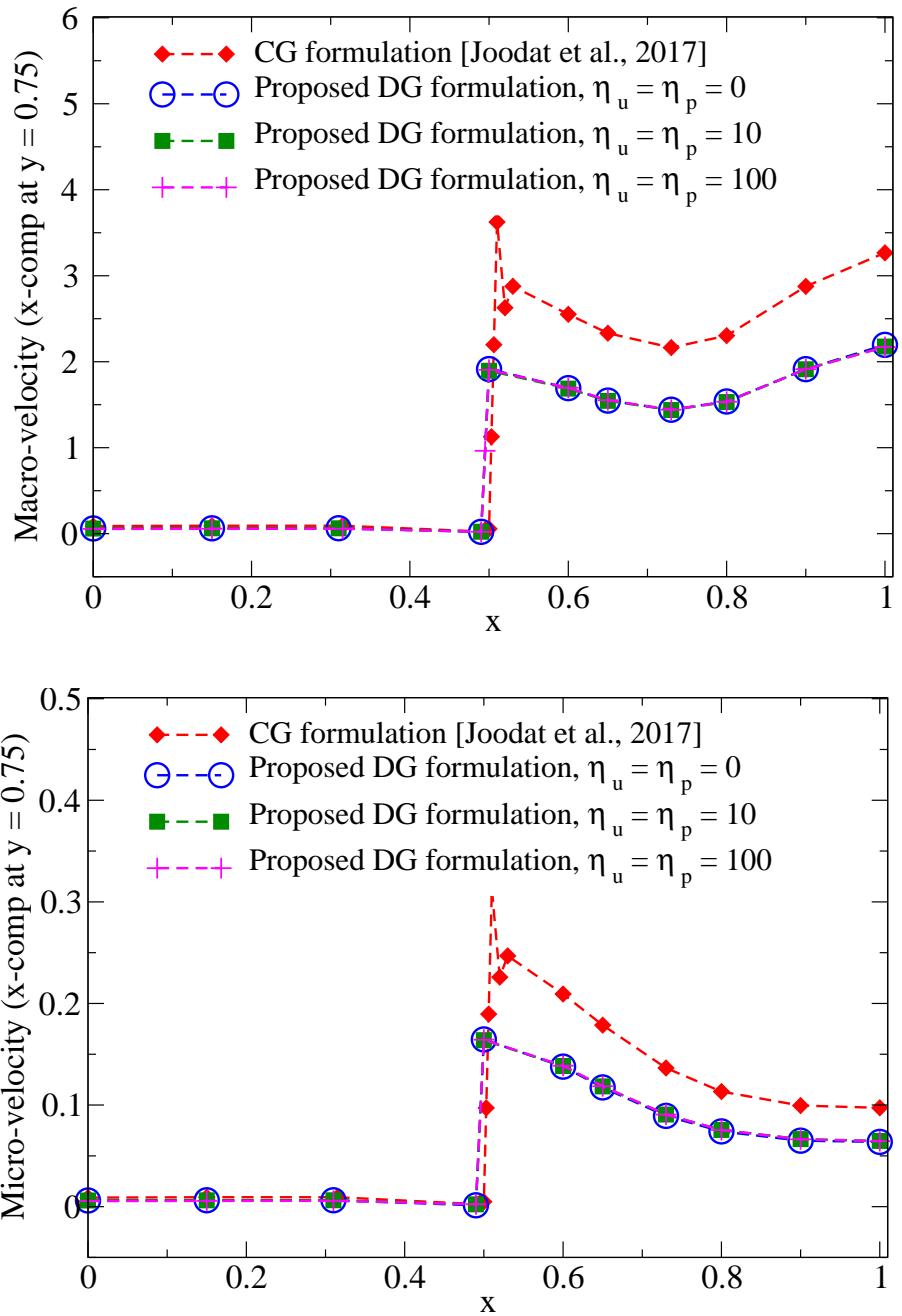


Figure 5.25: Quarter five-spot checkerboard problem: This figure compares the x-component of the macro-velocity (top) and micro-velocity (bottom) under the CG and the proposed DG formulations with different η_u and η_p .

mass inflow/outflow flux can be obtained as

$$m_{\max}^{\text{out}} := \max_{\omega \in \mathcal{T}_h} [\max[m(\omega), 0]] \quad \text{and} \quad (5.12a)$$

$$m_{\max}^{\text{in}} := \max_{\omega \in \mathcal{T}_h} [\max[-m(\omega), 0]]. \quad (5.12b)$$

It should be noted that the definition of the local mass flux presented in equation (5.11) is different from the corresponding one under the Darcy equations. For the case of single porosity and under Darcy equations, the net flux is zero for the velocity. However, under the DPP model the net flux need not be zero for the individual velocities and it is shown to be zero for the summation of \mathbf{u}_1 and \mathbf{u}_2 . The domain is discretized with structured TRI mesh of size 0.2. We employ the same boundary value problem as stated in subsection 5.1.2.1 with parameter values provided in Table 5.2. Pressures are prescribed on the whole boundary in both pore-networks.

Comparisons of maximum local mass inflow/outflow with respect to different combinations of equal-order interpolation are illustrated in **Fig. 5.26** for both DG and CG formulations. **Fig. 5.27** shows the local mass balance error in each element for cubic equal-order polynomials. The error values obtained under CG and DG formulations suggest that the DG formulation returns smaller errors.

5.4 Coupled problem with heterogeneous medium properties

In the previous sections, we used patch tests and canonical problems to demonstrate that the proposed stabilized mixed DG formulation can accurately capture the jumps in the solution fields across material interfaces. We will further illustrate the performance of this formulation using a representative problem pertaining to viscous fingering

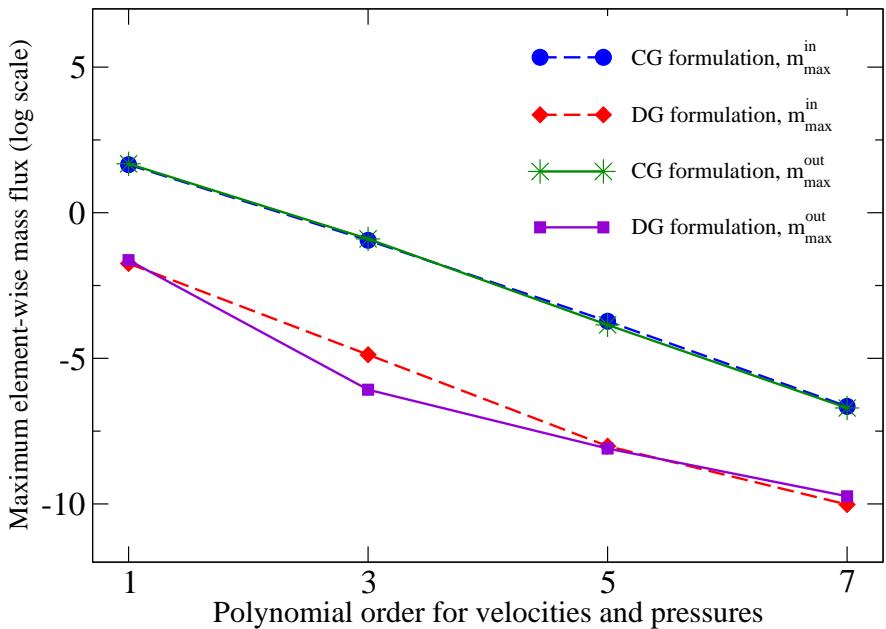


Figure 5.26: **Element-wise mass balance:** This figure shows the variation of the maximum element-wise inflow/outflow flux with interpolation polynomial orders.

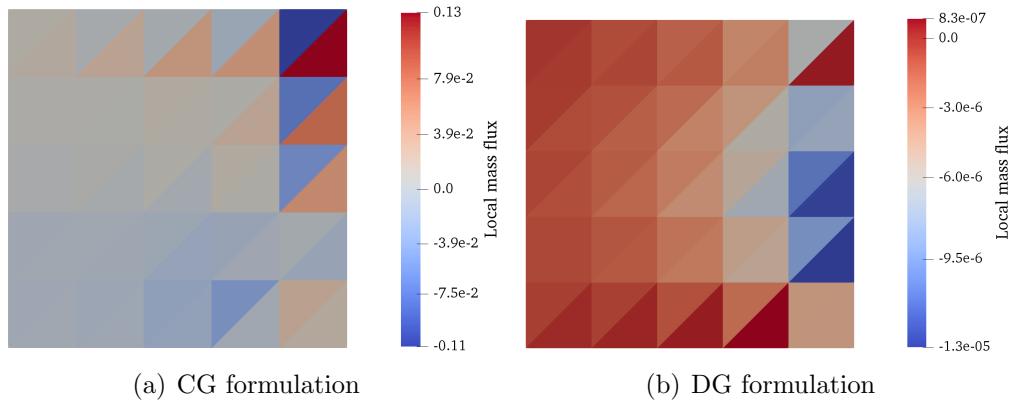


Figure 5.27: **Element-wise mass balance:** This figure shows the local mass balance error under both CG and DG formulations for cubic equal-order interpolation for all the variables. As can be seen, the DG formulation returns smaller errors.

in heterogeneous porous media.

Viscous fingering is a coupled phenomenon which involves both flow and transport [64]. In the flow of two immiscible fluids in a thin cell, typically called the Hele-Shaw cell, a more viscous fluid (with viscosity μ_H) is invaded by a less viscous one (with viscosity $\mu_L < \mu_H$), resulting in the creation of physical (displacement) instabilities [88]. The classical viscous fingering in porous media with a single pore-network (i.e., under Darcy equations) has been studied by [160], and therefore, this instability is sometimes referred to as the Saffman-Taylor instability in the literature [64]. Recently, [96] have numerically shown that viscous-fingering-type instabilities can also occur in homogeneous porous media with double pore-networks. They employed the continuous Galerkin (CG) formulation of the DPP model, as their studies were restricted to homogeneous porous media.

Herein, we will employ the proposed DG formulation to study the effect of *heterogeneity* on the appearance and growth of viscous-fingering-type physical instabilities in porous media with two pore-networks. The governing equations for this two-way coupled flow and transport problem consist of two parts. Flow under the DPP model is governed by equations (2.1a)–(2.1i) and the transient advection-diffusion problem is governed by the following set of equations:

$$\frac{\partial c(\mathbf{x}, t)}{\partial t} + \operatorname{div} [\mathbf{u}(\mathbf{x}, t)c(\mathbf{x}, t) - D(\mathbf{x}, t)\operatorname{grad}[c(\mathbf{x}, t)]] = f(\mathbf{x}, t) \quad \text{in } \Omega \times (0, T), \quad (5.13a)$$

$$c(\mathbf{x}, t) = c^p(\mathbf{x}, t) \quad \text{on } \Gamma^D \times (0, T), \quad (5.13b)$$

$$\hat{\mathbf{n}}(\mathbf{x}) \cdot (\mathbf{u}(\mathbf{x}, t)c(\mathbf{x}, t) - D(\mathbf{x}, t)\operatorname{grad}[c(\mathbf{x}, t)]) = q^p(\mathbf{x}, t) \quad \text{on } \Gamma^N \times (0, T), \text{ and} \quad (5.13c)$$

$$c(\mathbf{x}, t = 0) = c_0(\mathbf{x}) \quad \text{in } \Omega, \quad (5.13d)$$

where $c(\mathbf{x}, t)$ denotes the concentration, $D(\mathbf{x}, t)$ is the diffusivity, and the advection velocity $\mathbf{u}(\mathbf{x}, t)$ is sum of the macro- and micro-velocity fields (which are obtained from the flow problem). That is,

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_1(\mathbf{x}, t) + \mathbf{u}_2(\mathbf{x}, t). \quad (5.14)$$

The concentration for the more viscous fluid is assumed to be zero and for the less viscous fluid is considered to be equal to 1. In order to complete the coupling of the flow and transport equations and upon introducing μ_0 as the base viscosity of the less viscous fluid and $R_c = \log(\mu_H/\mu_L)$ as the log-mobility ratio, the viscosity of the fluid is assumed to exponentially depend on the concentration of the diffusant as

$$\mu(c(\mathbf{x}, t)) = \mu_0 \exp[R_c(1 - c(\mathbf{x}, t))]. \quad (5.15)$$

We consider a domain consisting of two horizontal layers with different permeabilities. The pictorial description of the problem is provided in **Fig. 5.28**. The values of macro- and micro-permeabilities in the bottom layer are assumed to be higher than those of the upper layer. Such heterogeneity in the permeability imposes a perturbation on the interface of the two fluids which causes the appearance of unstable finger-like patterns throughout the domain at the fluid-fluid interface. Moreover, a random function is used for defining the initial condition for the transport problem within the domain. Parameter values for this coupled flow and transport problem are provided in Table 5.7. For the advection-diffusion model given by equations (5.13a)–(5.13d), we have utilized Streamline Upwind Petrov-Galerkin (SUPG) formulation, as described in [38]. Also, see the computer code provided in Appendix C.

Fig. 5.29 shows the concentration profile at different time steps throughout the heterogeneous domain. The more viscous fluid is shown in dark blue and the less viscous fluid is shown in dark red. As can be seen, physical instabilities in form of separate

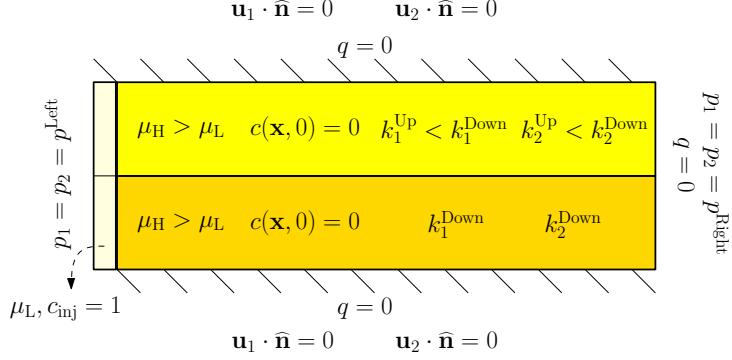


Figure 5.28: **Coupled flow and transport problem:** This figure shows the pictorial description of coupled flow-transport problem with heterogeneous medium properties along with initial and boundary conditions.

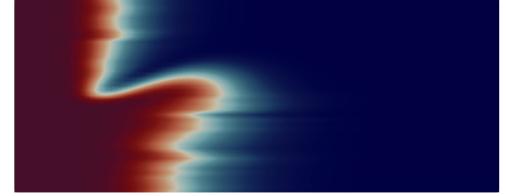
Table 5.7: Model parameters for coupled flow and transport problem in the heterogeneous domain.

Parameter	Value	Parameter	Value
$\gamma \mathbf{b}$	{0.0, 0.0}	k_2^{Down}	0.011
f	0.0	k_2^{Up}	0.009
L_x, L_y	1.0, 0.4	c_0	0.0
μ_0	1×10^{-3}	c_{inj}	1.0
R_c	3.0	p^{Left}	10.0
D	2×10^{-6}	p^{Right}	1.0
β	1.0	q	0.0
k_1^{Down}	1.1	Δt	5×10^{-5}
k_1^{Up}	0.9	T	1.5×10^{-3}
h	structured TRI mesh of size 0.01 used	η_u	0
		η_P	0

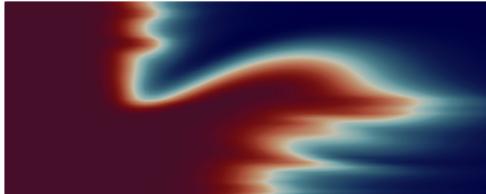
finger-like intrusions are created at the fluid-fluid interface. These intrusions are similar to the viscous-fingering-type instabilities. At the early time steps, we have a larger number of fingers compared to the later time steps. These smaller fingers merge and form fewer but much larger fingers as time goes by. It should be noted that finger-like physical instabilities grow at a higher rate in the bottom layer due to its higher permeability, as can be seen in **Fig. 5.29**. Moreover, at the later time steps, the fingers formed in the bottom layer tend to move towards the interface and enter the top layer. The proposed



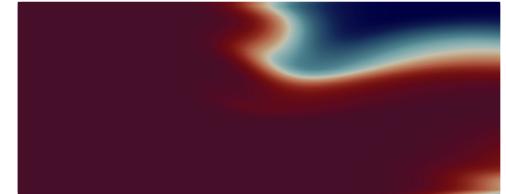
(a) $t=5 \Delta t$



(b) $t=10 \Delta t$



(c) $t=15 \Delta t$



(d) $t=20 \Delta t$

Figure 5.29: **Coupled flow and transport problem:** This figure shows that the proposed formulation can capture well-known instabilities in fluid mechanics, similar to viscous-fingering instability, in a heterogeneous, layered porous domain.

DG formulation eliminated the numerical instabilities (like Gibbs phenomenon and spurious node-to-node oscillations) but yet accurately captured the physical instabilities. It is worth mentioning that in our numerical simulations, the parameters η_u and η_p had no noticeable effect on the generation of fingers.

Chapter 6

Proposed Solvers

Recent numerical endeavors for solving DPP address small-scale problems [50, 97, 98]. However, The current iterative solver methodologies have been developed and successfully employed for either single-field problems (e.g., Poisson’s equation, linear elasticity) or for two-field problems (e.g., Darcy equations, Stokes equations) using two-field composable solvers [157, 147, 39]. However, there is a gap in knowledge when one wants to solve large-scale problems under a complex pore-structure model such as DPP model. Unlike Darcy equations, the governing equations under the DPP model cannot be written as a single-field Poisson equation solely in terms of pressures [97], or even as a two-field problem.

To facilitate solving large-scale problems under the DPP model, we present two four-field composable block solver methodologies. Appealing to PETSc’s composable solver features [19, 39] and Firedrake Project’s finite element libraries [147], we will show that the proposed composable block solvers can be effectively implemented in a parallel setting. The two salient features of the proposed block solvers are: they are scalable in both the algorithmic and parallel senses. They can be employed under a wide variety of finite element discretizations. Both of these features will be illustrated later in this document using representative two- and three-dimensional problems.

We illustrate that the proposed composable solvers can be used under a wide variety

of finite element discretizations, and will employ three popular finite element discretizations – the classical mixed formulation (which is based on the Galerkin formalism) using $H(\text{div})$ elements, the CG-VMS stabilized formulation [97] and our proposed stabilized mixed discontinuous Galerkin formulation [98], which will be referred to as DG-VMS formulation. We will consider $H(\text{div})$ discretizations for simplicial elements (triangle [TRI] and tetrahedron [TET]) and non-simplicial elements (quadrilateral [QUAD] and hexahedron [HEX]). In particular, we employ the lowest-order Raviart-Thomas spaces for simplicial elements [150, 26]. For non-simplicial elements, the velocity spaces for QUAD and HEX elements are, respectively, RCTF₁ and NCF₁ [128, 13].

This chapter will be valuable to subsurface modelers on three fronts. First and the obvious one is that the proposed composable block solver methodologies facilitate solving large-scale problems involving flow through porous media with multiple pore-networks. Second, our work can guide an application scientist to choose a finite element discretization among several choices. Third, our work illustrates on how to utilize performance metrics other than the commonly used metric – the total time to solution – in subsurface modeling. A couple of these other metrics include Digits of Efficacy (DoE) and the total Degrees-of-Freedom (DoF) processed per second (DoF/s).

6.1 Proposed four-field solvers

The fully discrete formulations for the DPP model can be assembled into the following linear problem

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \quad (6.1)$$

where \mathbf{K} is the stiffness matrix, \mathbf{u} is the vector of unknown velocities and pressure, and \mathbf{f} is the corresponding forcing or RHS vector. Solving the system of equations (6.1) in a

fast and scalable way requires careful composition and manipulation of the four different physical fields. In this section, we demonstrate how this can be done through PETSc [19, 18, 59] and its composable solver capabilities [39]. The individual block components of the stiffness matrix \mathbf{K} for the mixed Galerkin formulation using $H(\text{div})$ elements can be categorized into the following:

$$\mathbf{K}_{uu}^1 \leftarrow (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1), \quad (6.2a)$$

$$\mathbf{K}_{up}^1 \leftarrow -(\text{div}[\mathbf{w}_1]; p_1), \quad (6.2b)$$

$$\mathbf{K}_{pu}^1 \leftarrow (q_1; \text{div}[\mathbf{v}_1]), \quad (6.2c)$$

$$\mathbf{K}_{pp}^1 \leftarrow \left(q_1; \frac{\beta}{\mu} p_1 \right), \quad (6.2d)$$

$$\mathbf{K}_{uu}^2 \leftarrow (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2), \quad (6.2e)$$

$$\mathbf{K}_{up}^2 \leftarrow -(\text{div}[\mathbf{w}_2]; p_2), \quad (6.2f)$$

$$\mathbf{K}_{pu}^2 \leftarrow (q_2; \text{div}[\mathbf{v}_2]), \quad (6.2g)$$

$$\mathbf{K}_{pp}^2 \leftarrow \left(q_2; \frac{\beta}{\mu} p_2 \right), \quad (6.2h)$$

$$\mathbf{K}_{pp}^{12} \leftarrow -\left(q_1; \frac{\beta}{\mu} p_2 \right), \text{ and} \quad (6.2i)$$

$$\mathbf{K}_{pp}^{21} \leftarrow -\left(q_2; \frac{\beta}{\mu} p_1 \right). \quad (6.2j)$$

For the CG-VMS formulation, the individual block components of the stiffness matrix can be categorized into the following:

$$\mathbf{K}_{uu}^1 \leftarrow \frac{1}{2} (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1), \quad (6.3a)$$

$$\mathbf{K}_{up}^1 \leftarrow -(\text{div}[\mathbf{w}_1]; p_1) - \frac{1}{2} (\mathbf{w}_1; \text{grad}[p_1]), \quad (6.3b)$$

$$\mathbf{K}_{pu}^1 \leftarrow (q_1; \text{div}[\mathbf{v}_1]) + \frac{1}{2} (\text{grad}[q_1]; \mathbf{u}_1), \quad (6.3c)$$

$$\mathbf{K}_{pp}^1 \leftarrow \frac{1}{2} \left(\text{grad}[q_1]; \frac{1}{\mu} k_1 \text{grad}[p_1] \right) + \left(q_1; \frac{\beta}{\mu} p_1 \right), \quad (6.3d)$$

$$\mathbf{K}_{uu}^2 \leftarrow \frac{1}{2} (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2), \quad (6.3e)$$

$$\mathbf{K}_{up}^2 \leftarrow -(\operatorname{div}[\mathbf{w}_2]; p_2) - \frac{1}{2} (\mathbf{w}_2; \operatorname{grad}[p_2]), \quad (6.3f)$$

$$\mathbf{K}_{pu}^2 \leftarrow (q_2; \operatorname{div}[\mathbf{v}_2]) + \frac{1}{2} (\operatorname{grad}[q_2]; \mathbf{u}_2), \quad (6.3g)$$

$$\mathbf{K}_{pp}^2 \leftarrow \frac{1}{2} \left(\operatorname{grad}[q_2]; \frac{1}{\mu} k_2 \operatorname{grad}[p_2] \right) + \left(q_2; \frac{\beta}{\mu} p_2 \right), \quad (6.3h)$$

$$\mathbf{K}_{pp}^{12} \leftarrow -\left(q_1; \frac{\beta}{\mu} p_2 \right), \text{ and} \quad (6.3i)$$

$$\mathbf{K}_{pp}^{21} \leftarrow -\left(q_2; \frac{\beta}{\mu} p_1 \right). \quad (6.3j)$$

Likewise, the block components of the stiffness matrix for the DG-VMS formulation read:

$$\mathbf{K}_{uu}^1 \leftarrow \frac{1}{2} (\mathbf{w}_1; \mu k_1^{-1} \mathbf{u}_1) + \eta_u h (\{\!\{ \mu k_1^{-1} \}\!\} [\![\mathbf{w}_1]\!]; [\![\mathbf{u}_1]\!])_{\Gamma^{\text{int}}}, \quad (6.4a)$$

$$\mathbf{K}_{up}^1 \leftarrow -(\operatorname{div}[\mathbf{w}_1]; p_1) - \frac{1}{2} (\mathbf{w}_1; \operatorname{grad}[p_1]) + ([\![\mathbf{w}_1]\!]; \{\!\{ p_1 \}\!\})_{\Gamma^{\text{int}}} + (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_1)_{\Gamma_1^u}, \quad (6.4b)$$

$$\mathbf{K}_{pu}^1 \leftarrow (q_1; \operatorname{div}[\mathbf{v}_1]) + \frac{1}{2} (\operatorname{grad}[q_1]; \mathbf{u}_1) - (\{\!\{ q_1 \}\!\}; [\![\mathbf{v}_1]\!])_{\Gamma^{\text{int}}} - (q_1; \mathbf{v}_1 \cdot \hat{\mathbf{n}})_{\Gamma_1^u}, \quad (6.4c)$$

$$\mathbf{K}_{pp}^1 \leftarrow \frac{1}{2} \left(\operatorname{grad}[q_1]; \frac{1}{\mu} k_1 \operatorname{grad}[p_1] \right) + \left(q_1; \frac{\beta}{\mu} p_1 \right) + \frac{\eta_p}{h} (\{\!\{ \mu^{-1} k_1 \}\!\} [\![q_1]\!]; [\![p_1]\!])_{\Gamma^{\text{int}}}, \quad (6.4d)$$

$$\mathbf{K}_{uu}^2 \leftarrow \frac{1}{2} (\mathbf{w}_2; \mu k_2^{-1} \mathbf{u}_2) + \eta_u h (\{\!\{ \mu k_2^{-1} \}\!\} [\![\mathbf{w}_2]\!]; [\![\mathbf{u}_2]\!])_{\Gamma^{\text{int}}}, \quad (6.4e)$$

$$\mathbf{K}_{up}^2 \leftarrow -(\operatorname{div}[\mathbf{w}_2]; p_2) - \frac{1}{2} (\mathbf{w}_2; \operatorname{grad}[p_2]) + ([\![\mathbf{w}_2]\!]; \{\!\{ p_2 \}\!\})_{\Gamma^{\text{int}}} + (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_2)_{\Gamma_2^u}, \quad (6.4f)$$

$$\mathbf{K}_{pu}^2 \leftarrow (q_2; \operatorname{div}[\mathbf{v}_2]) + \frac{1}{2} (\operatorname{grad}[q_2]; \mathbf{u}_2) - (\{\!\{ q_2 \}\!\}; [\![\mathbf{v}_2]\!])_{\Gamma^{\text{int}}} - (q_2; \mathbf{v}_2 \cdot \hat{\mathbf{n}})_{\Gamma_2^u}, \quad (6.4g)$$

$$\mathbf{K}_{pp}^2 \leftarrow \frac{1}{2} \left(\operatorname{grad}[q_2]; \frac{1}{\mu} k_2 \operatorname{grad}[p_2] \right) + \left(q_2; \frac{\beta}{\mu} p_2 \right) + \frac{\eta_p}{h} (\{\!\{ \mu^{-1} k_2 \}\!\} [\![q_2]\!]; [\![p_2]\!])_{\Gamma^{\text{int}}}, \quad (6.4h)$$

$$\mathbf{K}_{pp}^{12} \leftarrow -\left(q_1; \frac{\beta}{\mu} p_2 \right), \text{ and} \quad (6.4i)$$

$$\mathbf{K}_{pp}^{21} \leftarrow -\left(q_2; \frac{\beta}{\mu} p_1 \right). \quad (6.4j)$$

The components of the corresponding RHS vector \mathbf{f} for equations (6.2), (6.3) and (6.4)

are:

$$\mathbf{f}_u^1 \leftarrow (\mathbf{w}_1; \gamma \mathbf{b}) - (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_{01})_{\Gamma_1^p}, \quad (6.5a)$$

$$\mathbf{f}_p^1 \leftarrow \mathbf{0}, \quad (6.5b)$$

$$\mathbf{f}_u^2 \leftarrow (\mathbf{w}_2; \gamma \mathbf{b}) - (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_{02})_{\Gamma_2^p}, \quad (6.5c)$$

$$\mathbf{f}_p^2 \leftarrow \mathbf{0}, \quad (6.5d)$$

and

$$\mathbf{f}_u^1 \leftarrow \frac{1}{2} (\mathbf{w}_1; \gamma \mathbf{b}) - (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_{01})_{\Gamma_1^p}, \quad (6.6a)$$

$$\mathbf{f}_p^1 \leftarrow \frac{1}{2} \left(\text{grad}[q_1]; \frac{1}{\mu} k_1 \gamma \mathbf{b} \right), \quad (6.6b)$$

$$\mathbf{f}_u^2 \leftarrow \frac{1}{2} (\mathbf{w}_2; \gamma \mathbf{b}) - (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_{02})_{\Gamma_2^p}, \quad (6.6c)$$

$$\mathbf{f}_p^2 \leftarrow \frac{1}{2} \left(\text{grad}[q_2]; \frac{1}{\mu} k_2 \gamma \mathbf{b} \right), \quad (6.6d)$$

and

$$\mathbf{f}_u^1 \leftarrow \frac{1}{2} (\mathbf{w}_1; \gamma \mathbf{b}) - (\mathbf{w}_1 \cdot \hat{\mathbf{n}}; p_{01})_{\Gamma_1^p}, \quad (6.7a)$$

$$\mathbf{f}_p^1 \leftarrow \frac{1}{2} \left(\text{grad}[q_1]; \frac{1}{\mu} k_1 \gamma \mathbf{b} \right) - (q_1; u_{n1})_{\Gamma_1^u}, \quad (6.7b)$$

$$\mathbf{f}_u^2 \leftarrow \frac{1}{2} (\mathbf{w}_2; \gamma \mathbf{b}) - (\mathbf{w}_2 \cdot \hat{\mathbf{n}}; p_{02})_{\Gamma_2^p}, \quad (6.7c)$$

$$\mathbf{f}_p^2 \leftarrow \frac{1}{2} \left(\text{grad}[q_2]; \frac{1}{\mu} k_2 \gamma \mathbf{b} \right) - (q_2; u_{n2})_{\Gamma_2^u}, \quad (6.7d)$$

respectively. Specifically, we employ PETSc's block solver capabilities, in the PCFIEDL-SPLIT class, taking two fields at a time. However, the global DPP model is a four field problem so we subdivide our problem recursively such that we end up with 2×2 blocks. Conceptually, PETSc can employ a wide variety of block solver methodologies on a 2×2 matrix:

$$\mathbf{K} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}, \quad (6.8)$$

where \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are individual block matrices which also consist of 2×2 blocks. Although equation (6.8) is conceptually a 4×4 block matrix, PETSc's field-splitting capabilities enables us to break the system down dynamically at runtime into two levels of 2×2 blocks.

We now propose two different ways one can compose scalable and efficient solvers and preconditioners for blocks \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} with the individual components shown in equations (6.2), (6.3), and (6.4).

6.1.1 Method 1: splitting by scales

One option is to split the global problem by scales. That is, each macro- or micro-scale 2×2 block will contain its corresponding velocity and pressure fields. Under this solver strategy, equation (6.1) is then rewritten as

$$\begin{bmatrix} \mathbf{K}_{uu}^1 & \mathbf{K}_{up}^1 & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{pu}^1 & \mathbf{K}_{pp}^1 & \mathbf{0} & \mathbf{K}_{pp}^{12} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{uu}^2 & \mathbf{K}_{up}^2 \\ \mathbf{0} & \mathbf{K}_{pp}^{21} & \mathbf{K}_{pu}^2 & \mathbf{K}_{pp}^2 \end{bmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{p}_1 \\ \mathbf{u}_2 \\ \mathbf{p}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_u^1 \\ \mathbf{f}_p^1 \\ \mathbf{f}_u^2 \\ \mathbf{f}_p^2 \end{pmatrix}, \quad (6.9)$$

where $\mathbf{0}$ is a zero matrix, \mathbf{u}_1 and \mathbf{p}_1 are the respective macro-scale velocity and pressure vectors, \mathbf{v}_2 and \mathbf{p}_2 are the respective micro-scale velocity and pressure vectors. The individual 2×2 blocks from equation (6.8) would be:

$$\begin{aligned} \mathbf{A} &:= \begin{bmatrix} \mathbf{K}_{uu}^1 & \mathbf{K}_{up}^1 \\ \mathbf{K}_{pu}^1 & \mathbf{K}_{pp}^1 \end{bmatrix}, & \mathbf{B} &:= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{pp}^{12} \end{bmatrix}, \\ \mathbf{C} &:= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{pp}^{21} \end{bmatrix}, & \mathbf{D} &:= \begin{bmatrix} \mathbf{K}_{uu}^2 & \mathbf{K}_{up}^2 \\ \mathbf{K}_{pu}^2 & \mathbf{K}_{pp}^2 \end{bmatrix}. \end{aligned} \quad (6.10)$$

Although the off diagonal blocks \mathbf{B} and \mathbf{C} contain the inter-scale pressure coupling terms, they are very sparse so we will ignore these blocks for now. The composition of the \mathbf{A} and \mathbf{D} blocks are similar to the classical mixed Poisson problem so the Schur complement approach outlined in [47, 124] and the references within can be applied.

The task is to individually precondition the decoupled \mathbf{A} and \mathbf{D} blocks. We note that

they admit factorizations of

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K}_{pu}^1 (\mathbf{K}_{uu}^1)^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{uu}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & (\mathbf{K}_{uu}^1)^{-1} \mathbf{K}_{up}^1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \text{ and} \quad (6.11)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K}_{pu}^2 (\mathbf{K}_{uu}^2)^{-1} & \mathbf{I} \end{bmatrix} \text{ and} \begin{bmatrix} \mathbf{K}_{uu}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & (\mathbf{K}_{uu}^2)^{-1} \mathbf{K}_{up}^2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (6.12)$$

where \mathbf{I} is the identity matrix and

$$\mathbf{S}^1 = \mathbf{K}_{pp}^1 - \mathbf{K}_{pu}^1 (\mathbf{K}_{uu}^1)^{-1} \mathbf{K}_{up}^1 \text{ and} \quad (6.13)$$

$$\mathbf{S}^2 = \mathbf{K}_{pp}^2 - \mathbf{K}_{pu}^2 (\mathbf{K}_{uu}^2)^{-1} \mathbf{K}_{up}^2 \quad (6.14)$$

are the Schur complements for the \mathbf{A} and \mathbf{D} blocks, respectively. The inverses can therefore be written as

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{I} & -(\mathbf{K}_{uu}^1)^{-1} \mathbf{K}_{up}^1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{K}_{uu}^1)^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{S}^1)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{K}_{pu}^1 (\mathbf{K}_{uu}^1)^{-1} & \mathbf{I} \end{bmatrix} \text{ and} \quad (6.15)$$

$$\mathbf{D}^{-1} = \begin{bmatrix} \mathbf{I} & -(\mathbf{K}_{uu}^2)^{-1} \mathbf{K}_{up}^2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{K}_{uu}^2)^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{S}^2)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{K}_{pu}^2 (\mathbf{K}_{uu}^2)^{-1} & \mathbf{I} \end{bmatrix}. \quad (6.16)$$

The task at hand is to approximate the inverses of the \mathbf{K}_{vv}^1 , \mathbf{K}_{uu}^2 , \mathbf{S}^1 , and \mathbf{S}^2 blocks. The first two blocks are simply mass matrices so we can invert them using the ILU(0) (incomplete lower upper) solver. For the Schur complement blocks, we employ a diagonal mass-lumping of \mathbf{K}_{uu}^1 and \mathbf{K}_{uu}^2 to estimate $(\mathbf{K}_{uu}^1)^{-1}$ and $(\mathbf{K}_{uu}^2)^{-1}$ because they are spectrally equivalent to the identity. That is,

$$\mathbf{S}_p^1 = \mathbf{K}_{pp}^1 - \mathbf{K}_{pu}^1 \text{diag}(\mathbf{K}_{uu}^1)^{-1} \mathbf{K}_{up}^1 \text{ and} \quad (6.17)$$

$$\mathbf{S}_p^2 = \mathbf{K}_{pp}^2 - \mathbf{K}_{pu}^2 \text{diag}(\mathbf{K}_{uu}^2)^{-1} \mathbf{K}_{up}^2 \quad (6.18)$$

to precondition the inner solvers responsible for inverting \mathbf{S}^1 and \mathbf{S}^2 . For these blocks we employ the multigrid V-cycle on \mathbf{S}_p^1 and \mathbf{S}_p^2 from the HYPRE BoomerAMG package [68].

We expect these to work because the \mathbf{S} blocks are spectrally equivalent to the Laplacian, modulo the penalty terms. In [124] it turns out the presence of the VMS stabilization terms in the \mathbf{K}_{pp}^1 and \mathbf{K}_{pp}^2 blocks do not drastically affect the performance or scalability of this solver strategy.

Instead of completely solving for the \mathbf{K}_{uu}^{-1} and \mathbf{S}_p of both scales, we apply only a single sweep of ILU(0)/block Jacobi and V-cycle, respectively, and rely on GMRES [158] to solve the entire 4×4 block system. Thus this outer GMRES is able to pick up the inter-scale pressure coupling blocks \mathbf{B} and \mathbf{C} . The PETSc command-line options for this solver methodology is given in listing 2 (see Appendix B).

6.1.2 Method 2: splitting by fields

Another option is to group the velocities and pressures of both scales into two different blocks. If this approach is taken, equation (6.1) is then rewritten as:

$$\begin{bmatrix} \mathbf{K}_{uu}^1 & \mathbf{0} & \mathbf{K}_{up}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{uu}^2 & \mathbf{0} & \mathbf{K}_{up}^2 \\ \mathbf{K}_{pu}^1 & \mathbf{0} & \mathbf{K}_{pp}^1 & \mathbf{K}_{pp}^{12} \\ \mathbf{0} & \mathbf{K}_{pu}^2 & \mathbf{K}_{pp}^{21} & \mathbf{K}_{pp}^2 \end{bmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_u^1 \\ \mathbf{f}_u^2 \\ \mathbf{f}_p^1 \\ \mathbf{f}_p^2 \end{pmatrix}, \quad (6.19)$$

and the individual blocks in equation (6.8) would now look like

$$\begin{aligned} \mathbf{A} &:= \begin{bmatrix} \mathbf{K}_{uu}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{uu}^2 \end{bmatrix}, & \mathbf{B} &:= \begin{bmatrix} \mathbf{K}_{up}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{up}^2 \end{bmatrix}, \\ \mathbf{C} &:= \begin{bmatrix} \mathbf{K}_{pu}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{pu}^2 \end{bmatrix}, & \mathbf{D} &:= \begin{bmatrix} \mathbf{K}_{pp}^1 & \mathbf{K}_{pp}^{12} \\ \mathbf{K}_{pp}^{21} & \mathbf{K}_{pp}^2 \end{bmatrix}. \end{aligned} \quad (6.20)$$

Unlike the previous methodology, we can work directly with the above stiffness matrix, which admits a factorization of

$$\mathbf{K} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ C\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (6.21)$$

where the Schur complement \mathbf{S} is

$$\mathbf{S} = \mathbf{D} - C\mathbf{A}^{-1}\mathbf{B}. \quad (6.22)$$

The inverse can therefore be written as

$$\mathbf{K}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -C\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix}. \quad (6.23)$$

Although \mathbf{A} is a 2×2 block containing velocities spanning across two different scales, we can still approximate \mathbf{A}^{-1} by inverting the entire \mathbf{A} block using ILU(0) because the off-diagonal blocks are zero and the diagonal blocks consist of only mass matrices. Approximating \mathbf{S}^{-1} is a little trickier because equation (6.22) is a dense 2×2 block with off-diagonal terms. However, we can still employ a diagonal mass-lumping of \mathbf{A} to estimate \mathbf{A}^{-1} because it is again spectrally equivalent to the identity. The preconditioner needed for \mathbf{S}^{-1} is

$$\begin{aligned} \mathbf{S}_p &= \mathbf{D} - C\text{diag}(\mathbf{A})^{-1}\mathbf{B} \\ &= \begin{bmatrix} \mathbf{K}_{pp}^1 & \mathbf{K}_{pp}^{12} \\ \mathbf{K}_{pp}^{21} & \mathbf{K}_{pp}^2 \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{pu}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{pu}^2 \end{bmatrix} \text{diag} \left(\begin{bmatrix} \mathbf{K}_{uu}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{uu}^2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{K}_{up}^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{up}^2 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_{pp}^1 - \mathbf{K}_{pu}^1 \text{diag}(\mathbf{K}_{uu}^1) \mathbf{K}_{up}^1 & \mathbf{K}_{pp}^{12} \\ \mathbf{K}_{pp}^{21} & \mathbf{K}_{pp}^2 - \mathbf{K}_{pu}^2 \text{diag}(\mathbf{K}_{uu}^2) \mathbf{K}_{up}^2 \end{bmatrix}. \end{aligned} \quad (6.24)$$

The off-diagonal blocks only consist of mass-matrix terms but the decoupled diagonal blocks are identical to equations (6.17) and (6.18). Thus, we individually employ multi-grid V-cycle on each of the diagonal blocks. As in the previous solver methodology, only

a single sweep of ILU(0) and the two multigrid V-cycles are needed for the \mathbf{A}^{-1} matrix and the two diagonal terms within the \mathbf{S}_p matrix, respectively, and the GMRES method is employed to solve the entire block system. The PETSc implementation is shown in listing 3 (see Appendix B).

6.1.3 Computer implementation

The finite element capabilities are provided by the Firedrake Project package [147, 122, 85, 128, 25, 121, 87, 86] with GNU compilers. This sophisticate finite element simulation package and its software dependencies can be found at [180, 183, 184, 177, 185, 179, 178, 181, 182]. The computational meshes are built on top of the DMplex unstructured grid format [103, 108, 109] and partitioned through the Chaco package [80]. Pictorial descriptions of the specific elements represented by this mesh format and utilized in this chapter are illustrated in Figure 3.1. The DMplex data structure interfaces very nicely with PETSc’s suite of parallel solvers and provide excellent scalability across thousands of MPI processes [46, 48]. Sample Firedrake codes for some of these benchmark problems can be found in Appendix C.

In our PETSc implementation, the same global matrix will be assembled for both solvers. The preconditioners differ by the subblocks which are extracted. The different sparsity pattern of the subblocks contributes to the performance differences seen in the solvers, but the overall assembly time remains unchanged for either solvers.

Chapter 7

Performance Model

Recently, [45] have proposed the Time-Accuracy-Size (TAS) performance spectrum model, which is an enhanced version of the original spectrum model proposed in [48] obtained by incorporating accuracy into the spectrum model. The TAS spectrum model can be used to study the performance of numerical formulations in a parallel setting. Herein, we will utilize the TAS model specifically to achieve the following: (i) We show that the proposed composable solvers are algorithmically scalable. (ii) We compare the performance of the two proposed composable solvers on a particular hardware. (iii) We discuss how the choice of finite element mesh type could affect the solver performance. (iv) We compare the performance of the chosen three finite element discretizations (the classical mixed formulation with $H(\text{div})$ elements, the CG-VMS stabilized formulation and the DG-VMS stabilized formulation) for solving the governing equations under the DPP model.

7.1 Performance spectrum modeling

To understand the parallel performance and algorithmic scalability of the proposed DPP composable block solver methodologies for the three finite element formulations, a performance model is needed. The performance model based on the Time-Accuracy-Size (TAS) spectrum analysis outlined in [45] shall be used as the basis for understanding

the quality of these finite element formulations with the proposed block solvers. We now briefly highlight the performance metrics used in this section and why they are each important in each of their own ways.

7.1.1 Mesh convergence

This criterion uses the convergence notion to account for numerical accuracy of a solution in the performance spectrum. In this text, we are adopting L_2 norm of the error defined as

$$L_2^{\text{norm}} = \|u_h - u\|_{L_2}, \quad (7.1)$$

where u is the exact solution, u_h is the the finite element solution, and h is measure of element size. Based on theory, most finite element discretizations will have an upper-bound for L_2 error norm as

$$L_2^{\text{norm}} \leq Ch^\alpha, \quad (7.2)$$

where α is known as *convergence rate* and C is some constant. When reporting and comparing how much accuracy is attained for each discretization, we use the notation of *Digits of Accuracy* (DoA) defined as

$$\text{DoA} := -\log_{10}(L_2^{\text{norm}}) \quad (7.3)$$

and plot DoA against *Digits of Size* (DoS), which is defined as

$$\text{DoS} := -\log_{10}(\text{DoF}). \quad (7.4)$$

Noting that for most formulations $\text{DoF} = Dh^{-nd}$, where D is some constant, the slope of DoA vs DoS plot is in the order of $\frac{\alpha}{nd}$. Any tailing off from the line plot is an indicator of incorrect implementation or solver convergence tolerances being too relaxed. Furthermore, the ratio DoA/DoS can be a good indicator of how much accuracy is achieved per DoF.

7.1.2 Strong-scaling

In this basic parallel scaling while the size of problem remains unchanged, the number of processes increases. In general, this metric comments on the marginal efficiency of each additional processes assigned to a problem. It is conventional to plot number of processes against the parallel efficiency defined as

$$\text{Parallel eff. (\%)} = \frac{T_1}{T_p \times proc} \times 100\%, \quad (7.5)$$

where *proc* is the number of MPI processes, T_1 is the total wall-clock time needed on a single MPI processes, and T_p is the total wall-clock time needed with *proc* MPI processes. However, this metric must be interpreted carefully for the following reasons:

1. **Solver iteration counts:** The number of solver iterations may fluctuate as the number of MPI processes changes. This can happen for a number of reasons, whether it is algorithmic implementation or relaxed convergence criterion. It is necessary to also report the number of KSP iterations required as the number of processes changes.
2. **Problems too small:** If the DoF count is too small for a particular MPI concurrency, communication time will swamp the computation time, thus reducing the parallel efficiency. This issue may arise when making comparative studies between different finite element discretizations, as different formulations have different DoF counts for a given *h*-size. Furthermore, for Python-based simulation packages like Firedrake, overheads from just-in-time compilation and instantiation of objects can also affect the strong-scaling .
3. **Problems too large:** If the DoF count is too large for a particular MPI concurrency, the problems not only drop out of the various levels of cache in the memory hierarchy but also invoke several expensive cache misses which can slow down the overall

performance. This may result in superlinear speedups, like the BLMVM bound-constrained optimization solver in [46].

Lastly, the global problem size for this scaling analysis is fixed which does not indicate how a particular solver or algorithm scales as the size increases. Traditionally, *weak-scaling* which increases both the global problem size and MPI concurrency with a fixed size per MPI process, can help one understand the algorithmic scalability of a particular solver, but it may be difficult to see from either the strong- or weak-scaling diagrams how a given machine or algorithm will handle a variety of workloads. Thus, an additional scaling metric needs to be introduced to help explain whether the performance of solvers might degrade due to increased KSP iteration counts or memory contention on a machine as the problem size increases.

7.1.3 Static-scaling

As described in [48], is a scaling analysis where the MPI concurrency is fixed but the problem size is increased. The essential metric for this analysis is the computation rate (DoF over Time). In this chapter, we run a series of problem sizes at a fixed parallelism and plot the computation rate against the wall-clock time. Note that the time need not be the total time to solution, instead one could look at various phases like the finite element assembly or solver computation rates.

Static-scaling returns information on performance and scalability of software and solvers across different hardware architectures. This scaling analysis also captures both strong-scaling and weak-scaling effects. Assuming that the block solvers are of $\mathcal{O}(N)$ scalability, where $N = \text{DoF}$, optimal scaling is indicated by a horizontal curve. Any tail offs at small problem sizes suggests strong-scaling effects whereas tail offs at large

problem sizes indicate suboptimal algorithmic or memory effects. The exact reasoning for the tail offs towards the right can be verified through arithmetic intensity, which is the measure of the total work over the total bytes transferred (see [48] and the references within).

7.1.4 Digits-of-Efficacy (DoE)

The final metric needed for our performance spectrum study is the *Digits of Efficacy* (DoE). This metric measures the accuracy production by a particular scheme in a given amount of time. The DoE could be defined as

$$\text{DoE} := -\log_{10}(L_2^{\text{norm}} \times \text{Time}). \quad (7.6)$$

Assuming that straight lines are captured in both the mesh convergence and static-scaling diagrams, the DoE has a linear dependence on problem size and returns a slope of $nd - \alpha$ (see [45] for details on the exact derivation). This efficacy measure is analogous to the *action* of a mechanical system, that is the product of energy and time. In the TAS spectrum analysis, the DoE represents an analogous action for computation, and we speculate that an optimal algorithm minimizes this product over its runtime. Since the DoE takes the negative logarithm of *action*, a higher DoE is desirable.

7.2 Representative numerical results

In this section, after clarifying the terminology and framework adopted for the performance spectrum model, we solve the four-field DPP model in two- and three-dimensional settings in order to demonstrate the implementation of the proposed composable block solvers and gauge their performances. The two-dimensional problem will be conducted

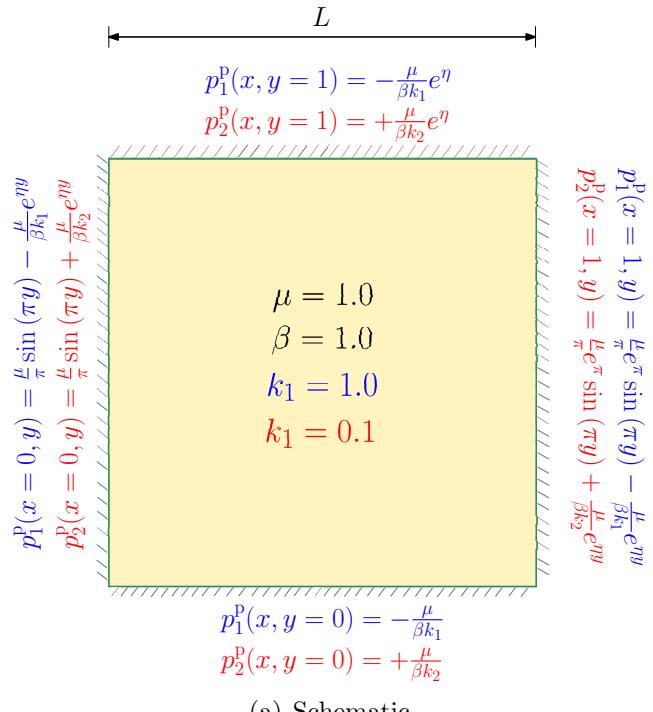
Table 7.1: Parameters for two-dimensional problem.

Parameter	Value
L	1.0
$\gamma \mathbf{b}$	{0.0, 0.0}
μ	1.0
β	1.0
k_1	1.0
k_2	0.1
η	$\sqrt{11}$
η_p	10
η_u	10

in serial (one MPI process) on a dual socket Intel Xeon E5-2609v3 server node. The three-dimensional problems will be conducted on a dual socket Intel Xeon E5-2698v3 server node and will utilize up to 16 MPI processes (8 MPI processes per socket). On different performance metrics, H(div), CG-VMS, and DG VMS formulations are compared for both simplicial (TRI, TET) and non-simplicial (QUAD, HEX) meshes. Both two-dimensional and three-dimensional problems were adopted by [97] for the convergence analysis of continuous stabilized mixed formulation (CG-VMS) and by [98] for the convergence analysis of discontinuous stabilized mixed formulation (DG-VMS) for the DPP model. We are generating three series of outputs for first-order CG-VMS, DG-VMS with $\eta_p = \eta_u = 10$, and H(div) formulations.

7.2.1 Two-dimensional study

For this first problem, let us consider a two-dimensional DPP boundary value problem with governing equations stated in equation (2.1). The homogeneous (i.e., constant macro and micro-permeabilities) bi-unit square computational domain and boundary conditions for this study are shown in **Fig. 7.1(a)**, and the corresponding parameters are described in Table 7.1.



(a) Schematic

Figure 7.1: Two-dimensional problem: This figure provides a pictorial description of the boundary value problem.

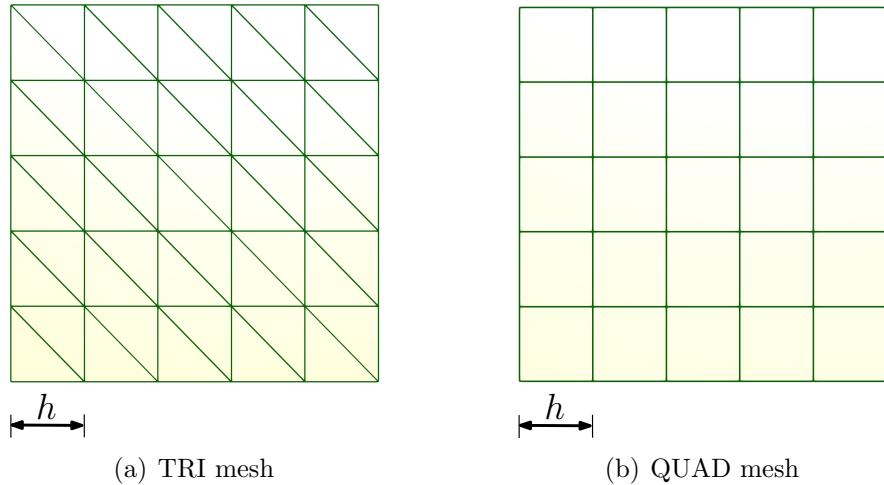


Figure 7.2: Two-dimensional problem: This figure shows the typical meshes employed in our numerical simulations.

The analytical solution for the pressure and velocity fields takes the following form:

$$\mathbf{u}_1(x, y) = -k_1 \begin{pmatrix} e^{\pi x} \sin(\pi y) \\ e^{\pi x} \cos(\pi y) - \frac{\eta}{\beta k_1} e^{\eta y} \end{pmatrix}, \quad (7.7a)$$

$$p_1(x, y) = \frac{\mu}{\pi} e^{\pi x} \sin(\pi y) - \frac{\mu}{\beta k_1} e^{\eta y}, \quad (7.7b)$$

$$\mathbf{u}_2(x, y) = -k_2 \begin{pmatrix} e^{\pi x} \sin(\pi y) \\ e^{\pi x} \cos(\pi y) + \frac{\eta}{\beta k_2} e^{\eta y} \end{pmatrix}, \text{ and} \quad (7.7c)$$

$$p_2(x, y) = \frac{\mu}{\pi} e^{\pi x} \sin(\pi y) + \frac{\mu}{\beta k_2} e^{\eta y}, \quad (7.7d)$$

where η is defined as

$$\eta := \sqrt{\beta \frac{k_1 + k_2}{k_1 k_2}}. \quad (7.8)$$

For two-dimensional performance spectrum analysis, all three finite element formulations will start off with the same h -sizes and will be refined up to 6 times. The initial TRI and QUAD coarse meshes are shown in **Fig. 7.2(a)** and **Fig. 7.2(b)** and the corresponding DoF counts for each formulation is shown in Table 7.2. The mesh convergence results with respect to DoA and DoS are performed under field-splitting solver and are shown in **Fig. 7.3** and **Fig. 7.4** respectively for TRI and QUAD meshes.

It should be noted that by applying scale-splitting solver, very same results could be obtained and for brevity, we decided not to plot them in figures. It can be seen in these diagrams that the CG-VMS and DG-VMS lines exhibit a slope $\frac{\alpha}{nd} = 1$, which verifies that our Firedrake implementation of these discretizations is correct. The H(div) lines exhibit a slope of 0.5 for TRI meshes but appear to have superlinear convergence for the QUAD meshes, which has also been observed in other Firedrake endeavors [72]. It can also be seen that if the solver tolerances are not strict enough, the mesh convergence lines will tail off. Nonetheless, the CG-VMS and DG-VMS have the highest ratios of DoA over DoS in most of these diagrams which suggests that each DoF in VMS formulations

Table 7.2: This table illustrates degrees-of-freedom for two-dimensional h-size refinement study.

h-size	DoF	CG-VMS		DG-VMS		H(div)		QUAD	
		TRI	QUAD	TRI	QUAD	TRI	DoF	h-size	DoF
5	216	5	216	5	900	5	600	5	270
10	726	10	726	10	3600	10	2400	10	1040
20	2646	20	2646	20	14400	20	9600	20	4080
40	10086	40	10086	40	57600	40	38400	40	16160
80	39366	80	39366	80	230400	80	153600	80	64320
160	155526	160	155526	160	921600	160	614400	160	256640

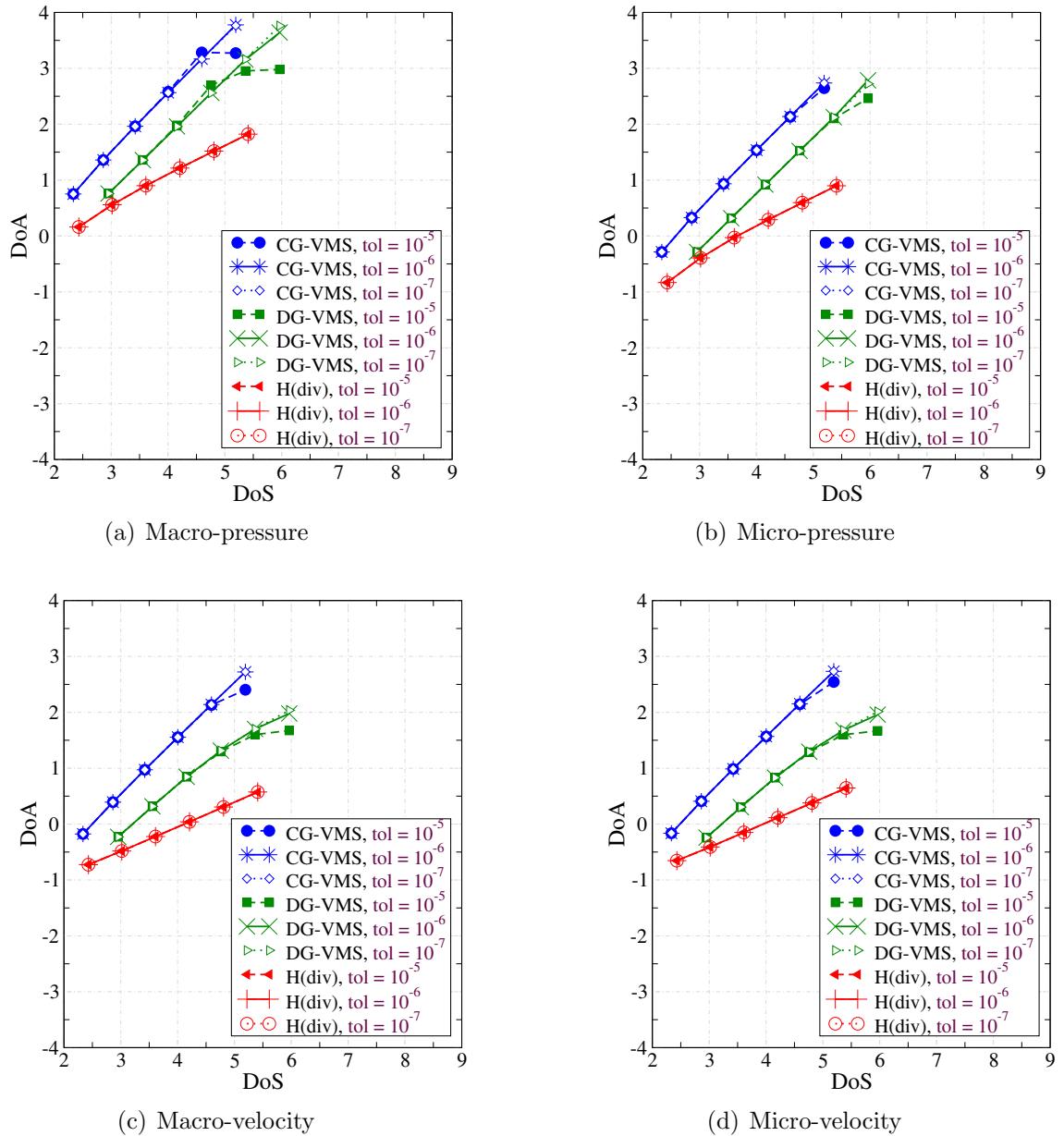


Figure 7.3: Two-dimensional problem using TRI mesh: This figure compares the mesh convergence results for the chosen finite element formulations under various solver tolerances. The results are shown for field-splitting block solver methodology.

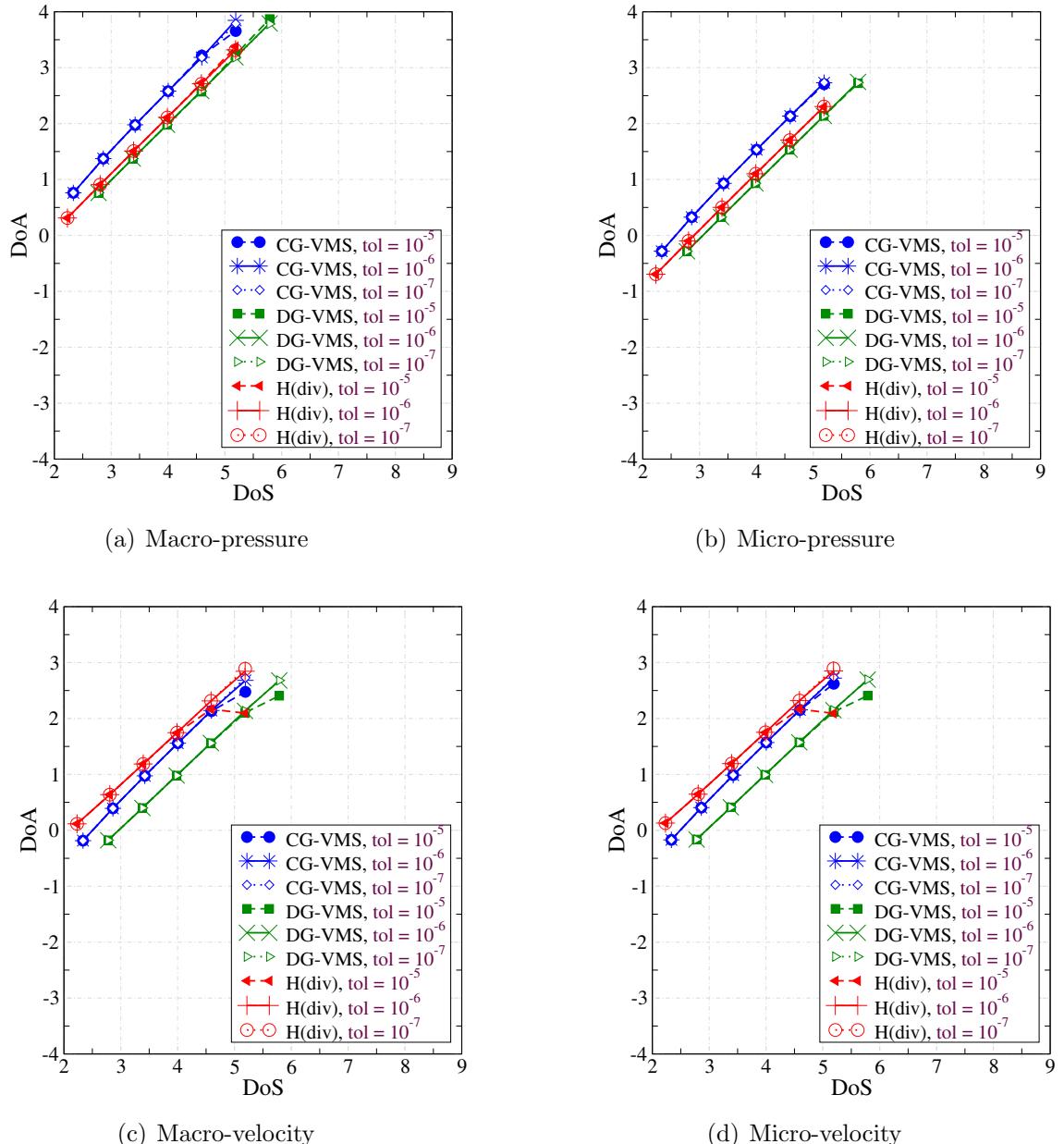


Figure 7.4: Two-dimensional problem using QUAD mesh: This figure compares the mesh convergence results for the chosen finite element formulations under various solver tolerances. The results are shown for field-splitting methodology.

has a greater level of contribution to the overall numerical accuracy than their $H(\text{div})$ counterparts.

Static-scaling results for both block solver strategies are shown in Figure 7.5, and we see that the total wall clock time is almost equally distributed among the assemble and solve phases. The field-splitting methodologies are slightly worse than their scale-splitting counterparts for the VMS formulations. However, the difference in performance is almost negligible when we look at the total time. The DoF counts are too small as the line curves for both the assembly and solve phases flatten out when all three formulations have roughly 10K DoF or more. No matter which mesh is utilized, the $H(\text{div})$ formulation processes its DoF count faster than either VMS formulations.

Figs. 7.6 and **7.7** contain DoE diagrams for TRI and QUAD meshes, respectively. Although $H(\text{div})$ appears to have the highest computation rates for both TRI and QUAD meshes, it has a lower DoA than its VMS counterparts for TRI meshes, which results in a much smaller DoE. For QUAD mesh on the other hand, $H(\text{div})$ formulation has a very high DoA and it beats out its VMS counterparts for all the fields, as shown in **Fig. 7.7**.

7.2.2 Three-dimensional study

In this section, we are solving a three-dimensional problem which is constructed by the Method of Manufactured Solutions (MMS) [140]. The homogeneous computational domain and boundary conditions for this problem are illustrated in **Fig. 7.8**, and related parameters are listed in Table 7.3. Also, a representative TET and HEX coarse meshes are shown in **Figs. 7.9(a)** and **7.9(b)**, respectively. The analytical solution for the

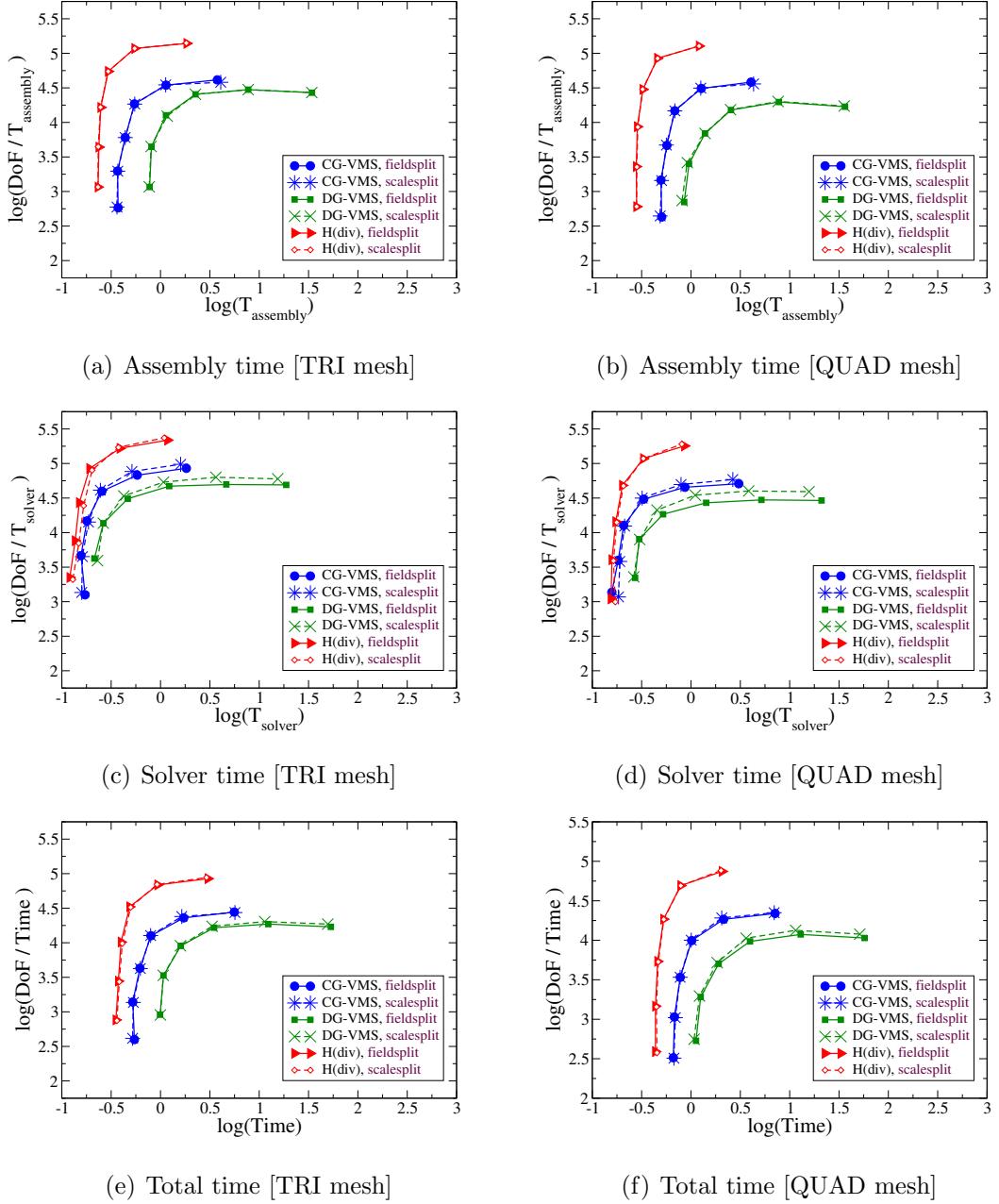


Figure 7.5: Two-dimensional problem: This figure compares the static-scaling results for the chosen finite element formulations using TRI and QUAD meshes.

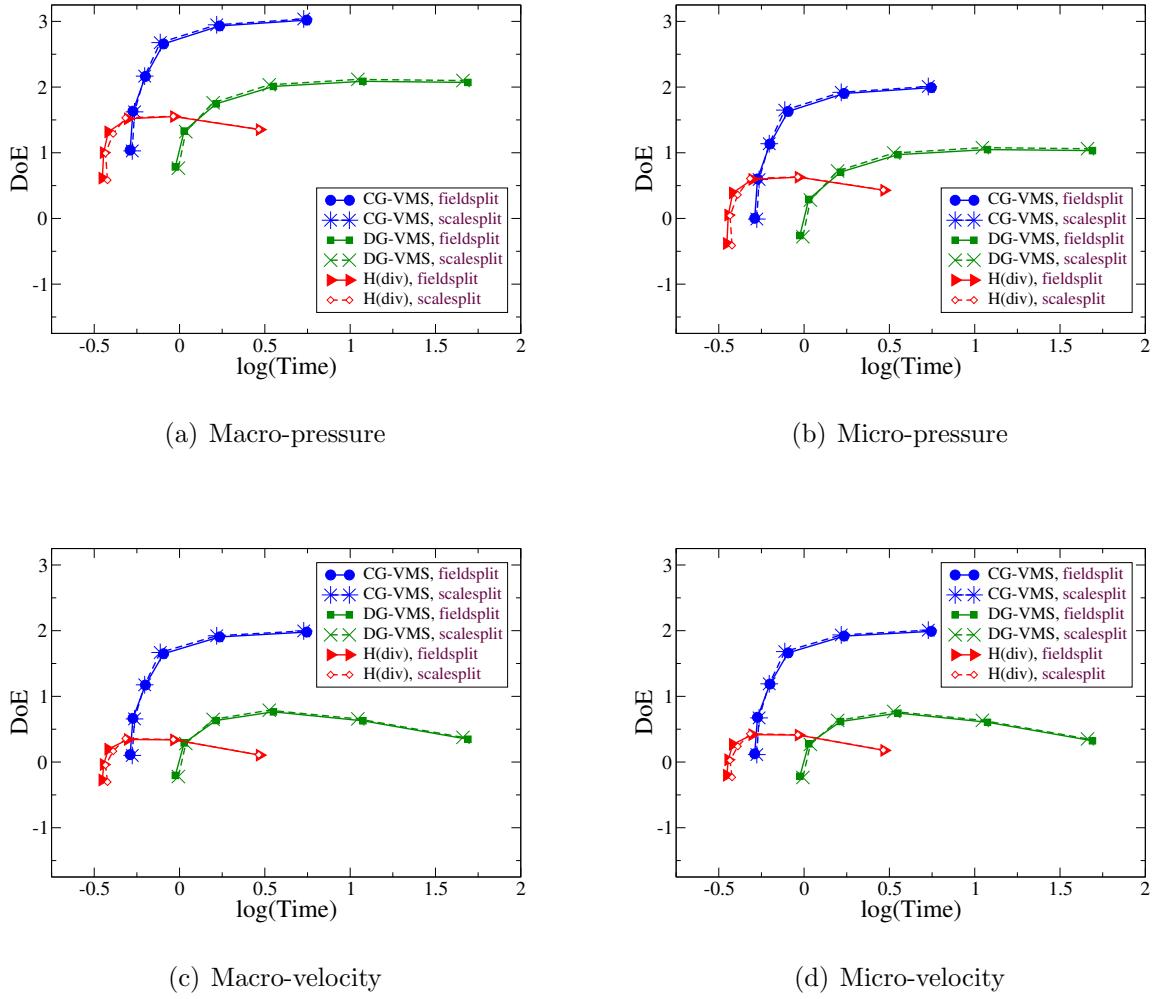


Figure 7.6: Two-dimensional problem for TRI mesh: This figure compares the Digits of Efficacy (DoE) among the chosen finite element formulations. Results for both solver methodologies with a tolerance of 10^{-7} are reported.

Table 7.3: Parameters for three-dimensional problem.

Parameter	Value
L	1.0
$\gamma \mathbf{b}$	$\{0.0, 0.0, 0.0\}$
μ	1.0
β	1.0
k_1	1.0
k_2	0.1
η	$\sqrt{11}$

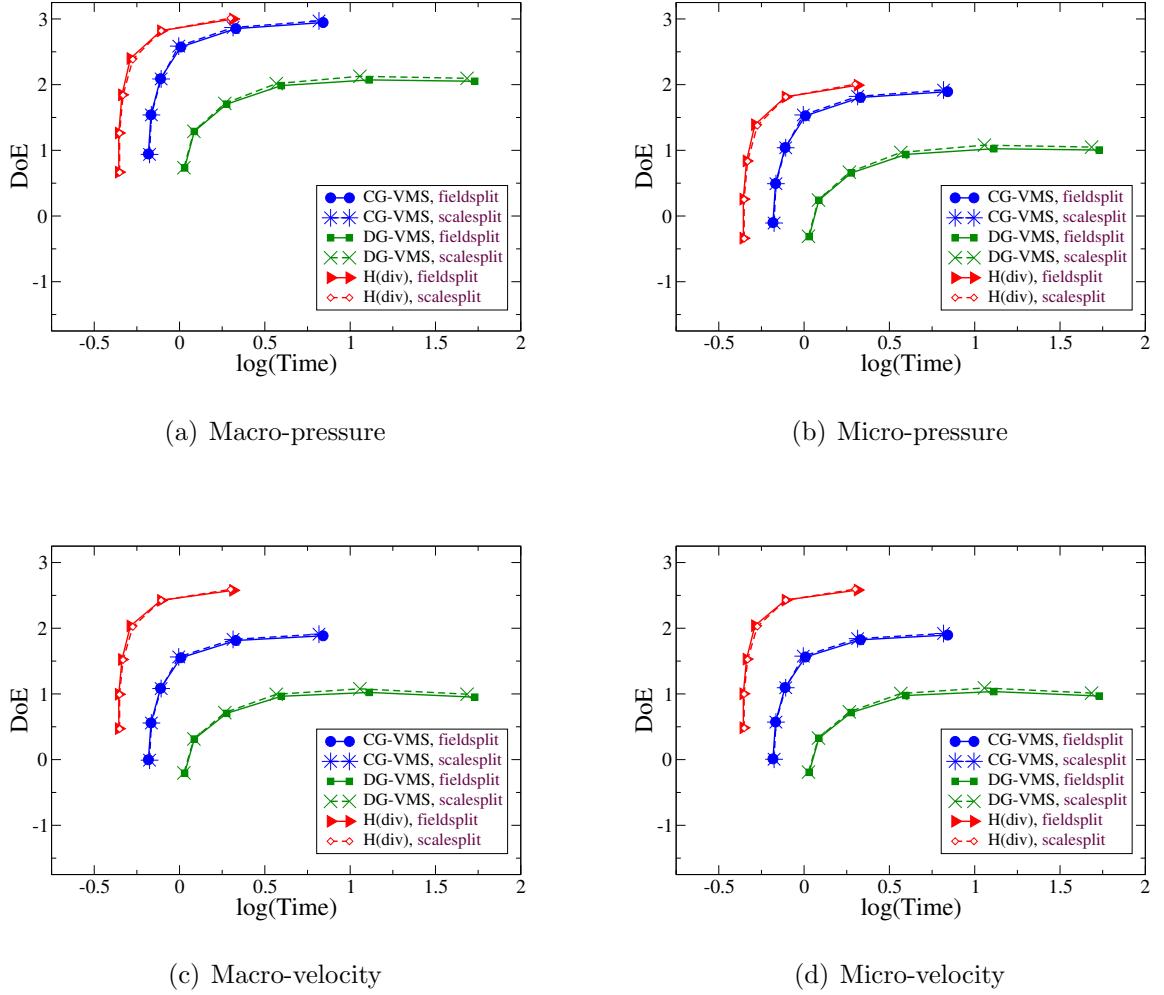


Figure 7.7: Two-dimensional problem for QUAD element: This figure compares the Digits of Efficacy (DoE) among the chosen finite element formulations. Results for both solver methodologies with a tolerance of 10^{-7} are reported.

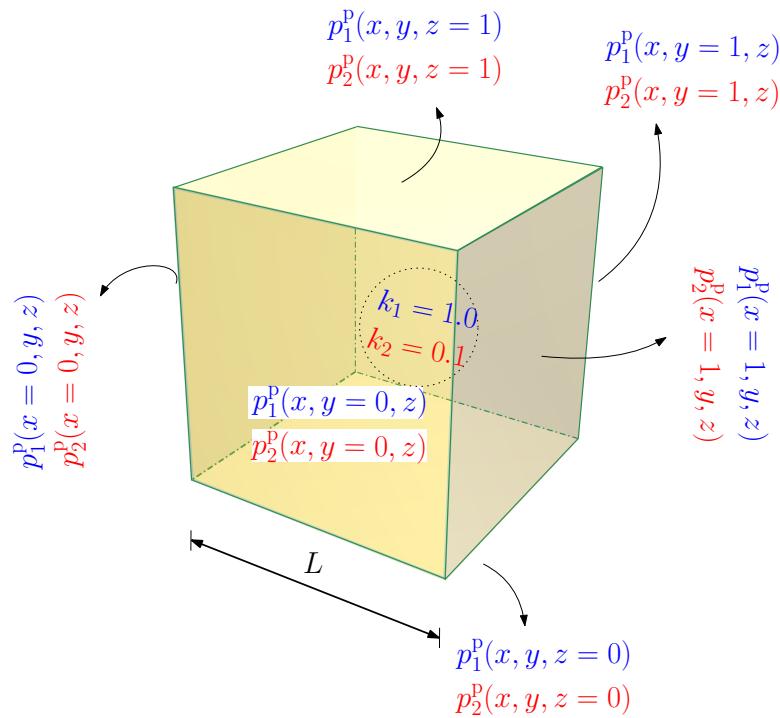


Figure 7.8: Three-dimensional domain: This figure provides a pictorial description of the boundary value problem.

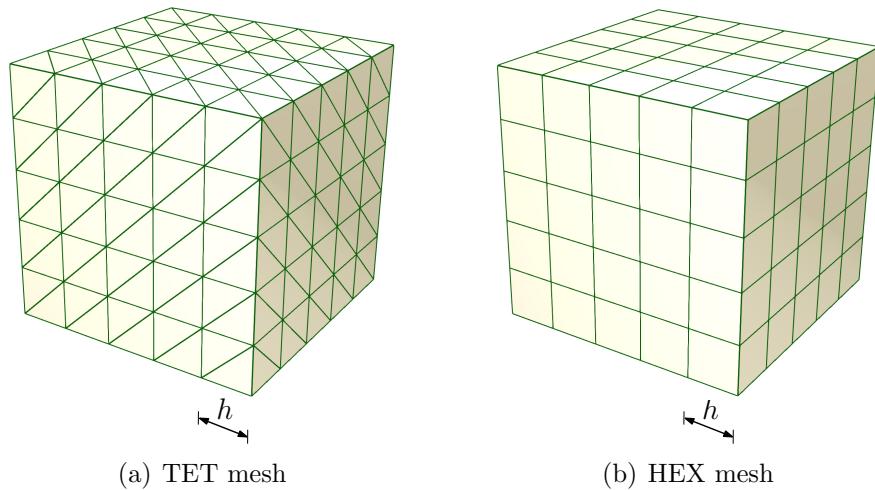


Figure 7.9: Three-dimensional domain: This figure shows the typical meshes employed in our numerical simulations.

pressure and velocity fields in the two pore-networks takes the following form:

$$\mathbf{u}_1(x, y, z) = -k_1 \begin{pmatrix} e^{\pi x}(\sin(\pi y) + \sin(\pi z)) \\ e^{\pi x} \cos(\pi y) - \frac{\eta}{\beta k_1} e^{\eta y} \\ e^{\pi x} \cos(\pi z) - \frac{\eta}{\beta k_1} e^{\eta z} \end{pmatrix}, \quad (7.9a)$$

$$p_1(x, y, z) = \frac{\mu}{\pi} e^{\pi x} (\sin(\pi y) + \sin(\pi z)) - \frac{\mu}{\beta k_1} (e^{\eta y} + e^{\eta z}), \quad (7.9b)$$

$$\mathbf{u}_2(x, y, z) = -k_2 \begin{pmatrix} e^{\pi x}(\sin(\pi y) + \sin(\pi z)) \\ e^{\pi x} \cos(\pi y) + \frac{\eta}{\beta k_2} e^{\eta y} \\ e^{\pi x} \cos(\pi z) + \frac{\eta}{\beta k_2} e^{\eta z} \end{pmatrix}, \text{ and} \quad (7.9c)$$

$$p_2(x, y, z) = \frac{\mu}{\pi} e^{\pi x} (\sin(\pi y) + \sin(\pi z)) + \frac{\mu}{\beta k_1} (e^{\eta y} + e^{\eta z}). \quad (7.9d)$$

7.2.2.1 Test 1: Strong-scaling results

First we investigate the strong-scaling performance of the proposed block solvers when applied to different finite element formulations. Two case studies are shown: first we fix the h -size for all finite element formulations, and second we modify each formulation's h -size such that they all have roughly matching DoF counts. Table 7.4 contains the corresponding h -sizes and DoF counts needed for both case studies.

First, we consider when all discretizations have an h -size = 1/16. Strong-scaling results for both field-splitting and scale-splitting block solver methodologies for H(div), CG-VMS, and DG-VMS can be found in Tables 7.5, 7.6, and 7.7, respectively. All three Tables indicate that the KSP iteration counts between field-splitting and scale-splitting are identical whereas the wall-clock time for scale-splitting is slightly smaller. The KSP counts for H(div) and CG-VMS do not change much when the number of MPI processes increases, whereas DG-VMS's KSP counts increase drastically. We suspect that this proliferation in iteration count is attributed to the fact that the DG-VMS introduces

several penalty terms and average/jump operators, all of which could affect the quality of the Schur complement approach presented in this chapter. But even with our proposed approach, the iterations appears to stabilize as more MPI processes are added. The increase in KSP iteration counts will affect the parallel efficiency so one has to be careful when interpreting these results. Nonetheless, we see that the DG-VMS parallel efficiency is the highest, even with its proliferated KSP counts. This is attributed to the fact that the DoF count for DG-VMS is larger than CG-VMS and H(div). All three tables indicate that higher DoF counts bring in more efficiency in the parallel sense.

Second, we consider the case when all discretizations contain approximately 200K degrees-of-freedom. Strong-scaling results for both field-splitting and scale-splitting block solver methodologies for H(div), CG-VMS, and DG-VMS can be found in Tables 7.8, 7.9, and 7.10, respectively. Like with the same h -size case, the scale-splitting method appears to be more efficient in terms of wall-clock time needed despite having the same KSP counts as the field-splitting method. It can also be seen that the H(div) and CG-VMS KSP counts do not fluctuate much with MPI processes and that DG-VMS KSP counts still increase dramatically. However, tuning the mesh sizes such that all finite element discretizations have the same DoF count enables us to have better understanding of the parallel performance, especially for three-dimensional problems. It can be seen that H(div) requires the least amount of wall-clock time resulting in the lowest parallel efficiency, but that does not mean this is a bad formulation. In order to understand the quality of the H(div) discretizations, we need to take into consideration the numerical accuracy and perform a TAS spectrum analysis.

Table 7.4: This table shows h-size and corresponding degrees-of-freedom for three-dimensional strong-scaling studies. In test1, we have fixed the h -size to 1/16 for all finite element formulations. In test2, we selected h -size for each formulation such that all end up to have roughly the same DoF of approximately 200k.

Strong scaling tests	CG-VMS			DG-VMS			H(div)		
	TET	HEX	TET	HEX	TET	HEX	TET	HEX	TET
h-size	DoF	h-size	Dof	h-size	DoF	h-size	DoF	h-size	DoF
Test 1	1/16	39305	1/16	39305	1/16	786432	1/16	262144	1/16
Test 2	1/28	195112	1/29	216000	1/10	192000	1/15	216000	1/17
								180336	1/29
									200158

Table 7.5: 3D problem: Strong-scaling results for $H(\text{div})$ formulation with same h-size.

Field-splitting							Scale-splitting						
No. of MPI proc.	TET mesh			HEX mesh			No. of MPI proc.	TET mesh			HEX mesh		
	Time Assembly	Solver	Total	KSP	Parallel eff. (%)	Time Assembly	Solver	Total	KSP	Parallel eff. (%)	Time Assembly	Solver	Total
1	6.82E-01	5.80E-01	1.26E+00	15	100	5.45E-01	2.97E-01	8.42E-01	17	100	6.81E-01	3.97E-01	10.78E-01
2	4.77E-01	4.16E-01	8.93E-01	15	70.6363	5.47E-01	2.66E-01	8.13E-01	17	51.8154	4.77E-01	2.66E-01	7.43E-01
4	3.67E-01	2.91E-01	6.58E-01	16	47.8812	5.15E-01	2.44E-01	7.59E-01	17	27.7229	3.67E-01	2.44E-01	6.11E-01
8	3.27E-01	2.43E-01	5.70E-01	16	27.6535	5.58E-01	2.89E-01	8.47E-01	17	12.4233	3.27E-01	2.89E-01	5.16E-01
16	3.38E-01	2.83E-01	6.21E-01	16	12.6851	6.60E-01	3.98E-01	1.06E+00	17	4.97401	3.38E-01	3.98E-01	4.97401

Table 7.6: 3D problem: Strong-scaling results for CG-VMS formulation with same h-size.

Field-splitting							HEX mesh						
No. of MPI proc.	TET mesh			Parallel			Time			KSP			Parallel eff. (%)
	Assembly	Solver	Total		KSP	Parallel eff. (%)		Assembly	Solver	Total		KSP	Parallel eff. (%)
1	2.37E+00	1.02E+00	3.39E+00	12	100	2.67E+00	1.67E+00	4.34E+00	16	100			
2	2.06E+00	7.28E-01	2.78E+00	13	60.8118	2.16E+00	1.20E+00	3.36E+00	18	64.5344			
4	1.55E+00	4.77E-01	2.02E+00	14	41.8438	1.65E+00	7.54E-01	2.40E+00	19	45.1499			
8	1.06E+00	3.75E-01	1.44E+00	14	29.4128	1.41E+00	5.94E-01	2.00E+00	20	27.0719			
16	8.90E-01	3.64E-01	1.25E+00	15	16.876	1.19E+00	5.13E-01	1.70E+00	20	15.9673			

Scale-splitting							HEX mesh						
No. of MPI proc.	TET mesh			Parallel			Time			KSP			Parallel eff. (%)
	Assembly	Solver	Total		KSP	Parallel eff. (%)		Assembly	Solver	Total		KSP	Parallel eff. (%)
1	2.39E+00	8.62E-01	3.25E+00	12	100	2.67E+00	1.28E+00	3.96E+00	16	100			
2	2.03E+00	6.35E-01	2.67E+00	13	61.0049	2.17E+00	9.79E-01	3.15E+00	18	62.8295			
4	1.57E+00	4.34E-01	2.00E+00	14	40.6141	1.66E+00	6.57E-01	2.32E+00	19	42.6401			
8	1.04E+00	3.49E-01	1.39E+00	14	29.2416	1.39E+00	5.27E-01	1.92E+00	20	25.8155			
16	9.62E-01	3.56E-01	1.32E+00	15	15.4423	1.21E+00	5.10E-01	1.72E+00	20	14.3786			

Table 7.7: 3D problem: Strong-scaling results for DG-VMS formulation with same h-size.

Field-splitting		TET mesh				HEX mesh			
No. of MPI proc.	Time Assembly	Solver	Total	KSP	Parallel eff. (%)	Time Assembly	Solver	Total	KSP Parallel eff. (%)
1	5.51E+01	2.42E+01	7.93E+01	19	100	5.32E+01	2.53E+01	7.85E+01	22 100
2	2.64E+01	2.37E+01	5.01E+01	44	79.1343	2.62E+01	2.08E+01	4.70E+01	39 83.4645
4	9.65E+00	1.88E+01	2.85E+01	81	69.6209	1.14E+01	1.10E+01	2.24E+01	42 87.6787
8	5.45E+00	1.26E+01	1.80E+01	100	55.0972	7.70E+00	6.81E+00	1.45E+01	45 67.6172
16	3.66E+00	8.67E+00	1.23E+01	110	40.2496	6.48E+00	4.35E+00	1.08E+01	46 45.3385

Scale-splitting		TET mesh				HEX mesh			
No. of MPI proc.	Time Assembly	Solver	Total	KSP	Parallel eff. (%)	Time Assembly	Solver	Total	KSP Parallel eff. (%)
1	5.52E+01	1.86E+01	7.37E+01	19	100	5.35E+01	1.82E+01	7.17E+01	22 100
2	2.65E+01	1.88E+01	4.52E+01	44	81.4987	2.62E+01	1.51E+01	4.13E+01	39 86.6804
4	1.00E+01	1.57E+01	2.56E+01	81	71.8994	1.10E+01	8.13E+00	1.91E+01	42 93.881
8	5.68E+00	1.04E+01	1.61E+01	100	57.1805	7.92E+00	5.11E+00	1.30E+01	45 68.7356
16	3.78E+00	7.45E+00	1.12E+01	110	41.0762	6.63E+00	3.40E+00	1.00E+01	46 44.6028

7.2.2.2 Test 2: TAS Spectrum Analysis

For the TAS spectrum analysis, we consider a range of problems, shown in Table 7.11, such that all finite element formulations in each refinement step have roughly the same DoF count. The mesh convergence results with respect to DoA and DoS, for both TET and HEX meshes, are shown in [Fig. 7.10](#). CG-VMS and DG-VMS lines indicate a slope of $\frac{2}{3}$, which again corroborates that our Firedrake implementation of these formulations are correct. The H(div) lines exhibit a slope of $\frac{1}{3}$ for TET mesh. However, similar to the two-dimensional problem for non-simplicial element QUAD, H(div) exhibits super linear convergence for the HEX meshes. We are not observing any tail-offs in these results as the solver relative convergence tolerance of $1e - 7$ was strict enough. The observation that both CG- and DG-VMS have the highest DoA over DoS ratio for almost all velocity and pressure fields implies that they have greater levels of contribution to the overall numerical accuracy than the H(div) schemes.

Static-scaling results for both block solver strategies are presented in [Fig. 7.11](#). Flat lines appear in all six subfigures, indicating that the proposed block-solver methodologies are scalable under the chosen h -sizes and hardware environment. It is a common belief among application scientists that a solver exhibits worse scaling than an assembly procedure, since assembly is almost entirely local. However, the results show that for all the chosen discretizations—no matter what solver methodology is employed—time to assemble stiffness matrix is higher than the solver time. This infers that we have successfully optimized solvers to such an extent that the assembly procedure as implemented in the Firedrake Project is more dominant.

Analogous to the two-dimensional problem, the scale-splitting methodologies are slightly better than their field-splitting counterparts for the all formulations. Evidently, this disparity is more clear for VMS formulations at the solve time level. However, the difference

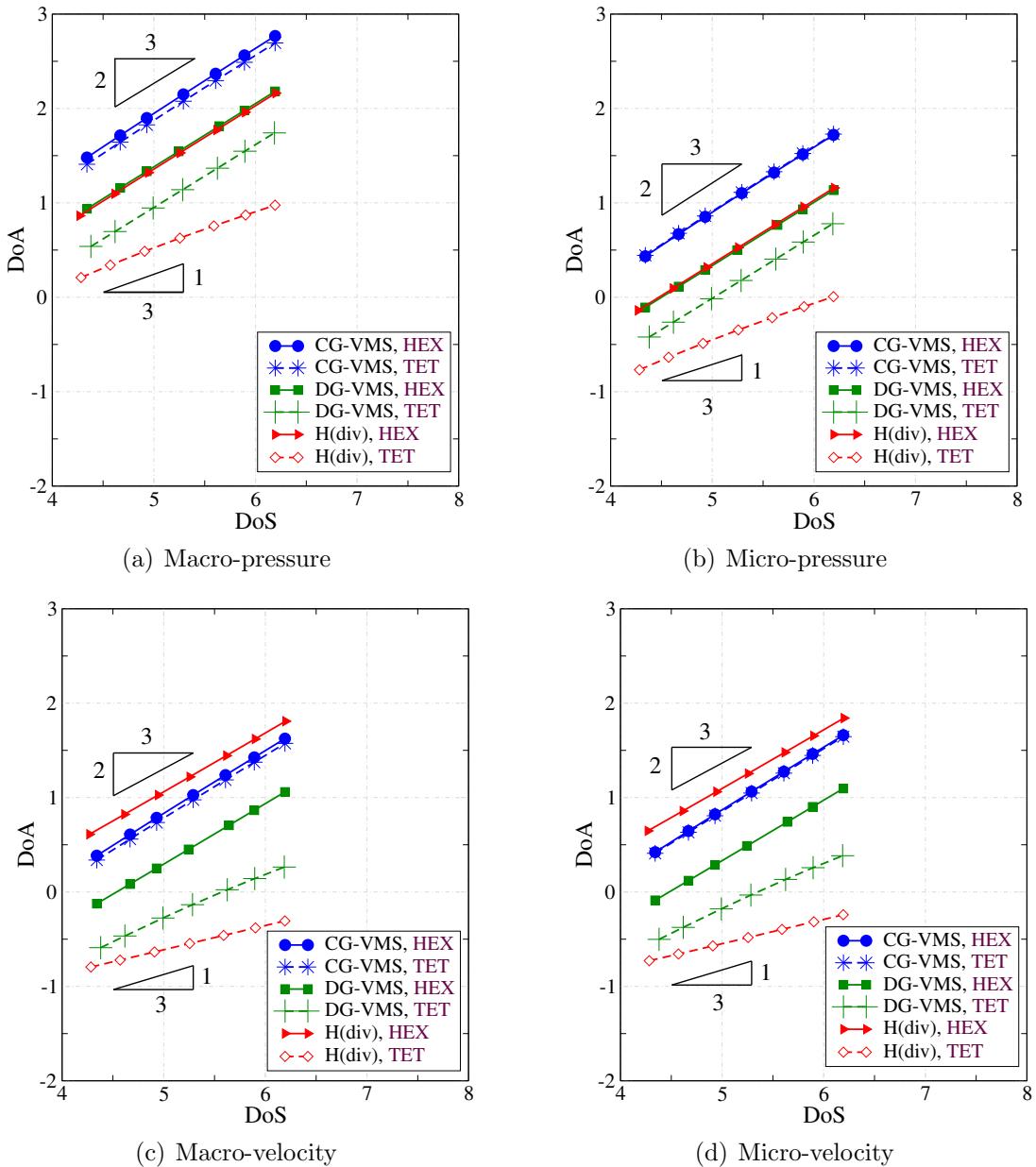


Figure 7.10: Three-dimensional problem: This figure compares the mesh convergence results for the chosen finite element formulations using TET and HEX meshes. The solver tolerance is taken to be 10^{-7} .

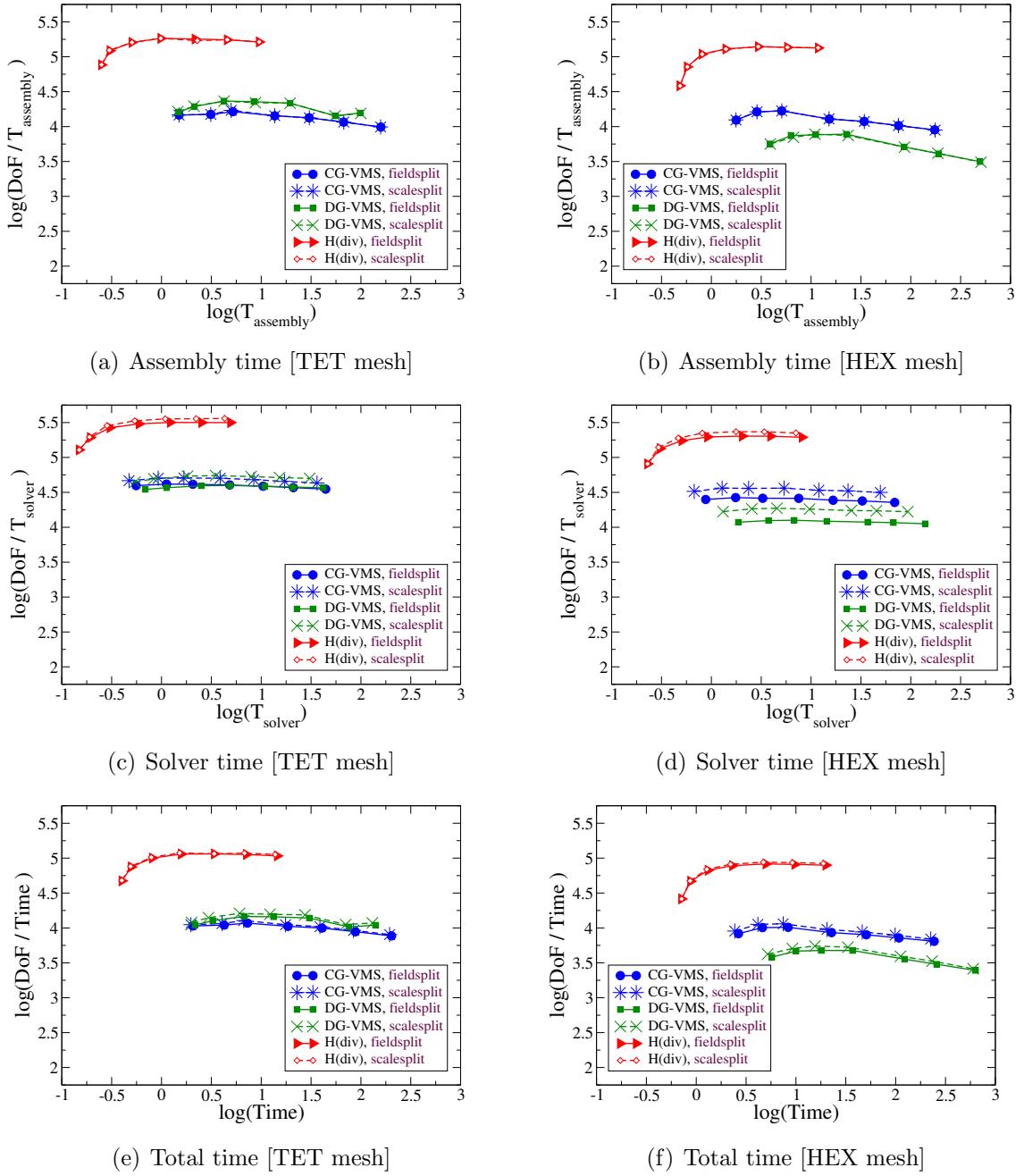


Figure 7.11: Three-dimensional problem: This figure compares the static-scaling results for the chosen finite element formulations using TET and HEX meshes.

in performance is almost inconsequential when we look at the total time. It can be seen that the DoF counts are sufficient to level out the curves when all three formulations have roughly 20K DoFs or more. Regardless of the mesh type, the H(div) formulation processes its DoF count faster than either VMS formulations.

Figs. 7.12 and **7.13** contain DoE diagrams for TET and HEX meshes, respectively. For the case of TET mesh type, in spite of H(div) having the fastest computation rates, it has a lower DoA than its VMS counterparts which in turns lead to a much smaller DoE with steep declining curve. On the contrary, for the case of the HEX mesh, H(div) surpasses its VMS counterparts due to its high DoA values. These diagrams demonstrate how numerical accuracy can have a drastic effect on the overall computational performance of these various finite element formulations.

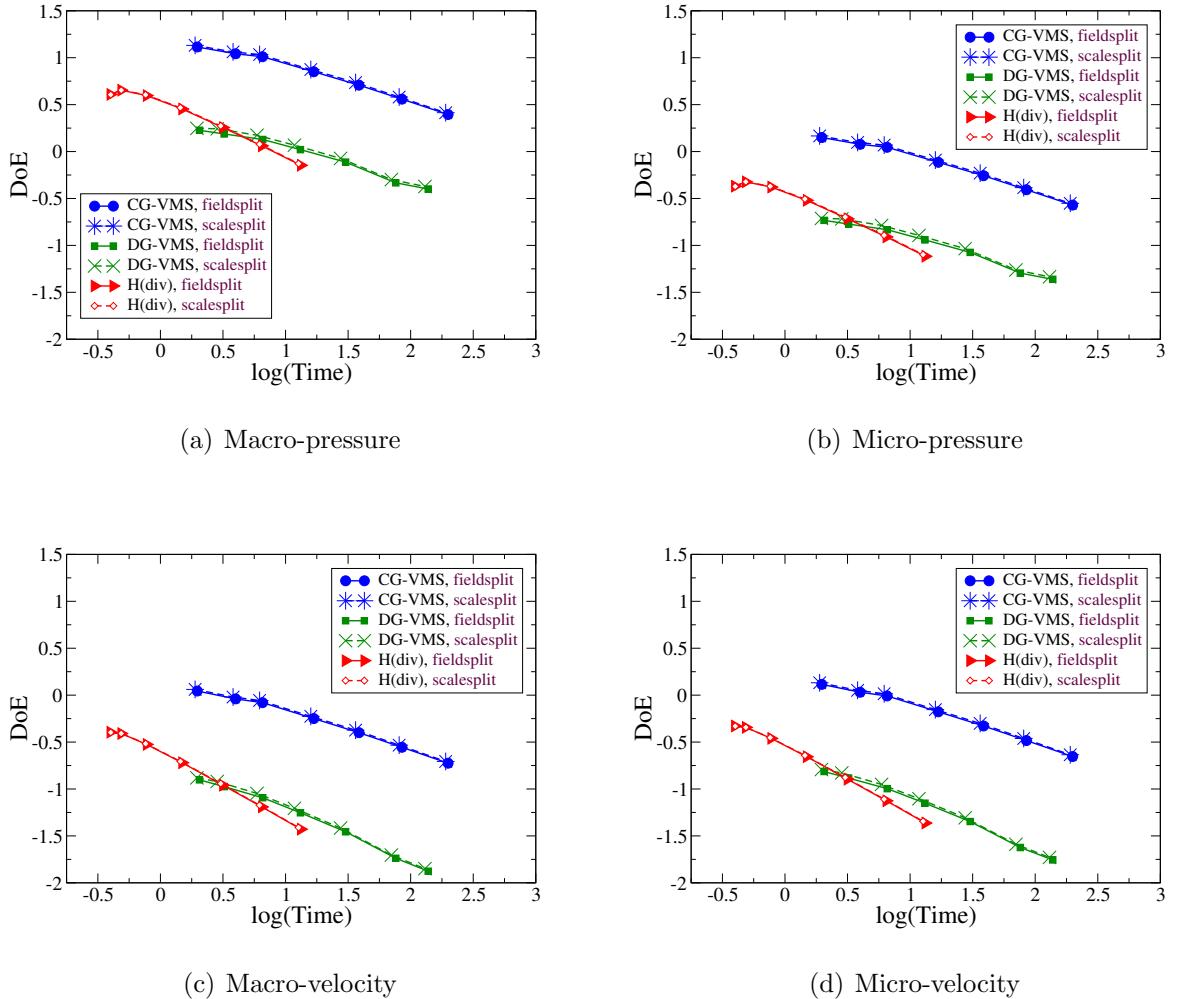


Figure 7.12: Three-dimensional problem using TET mesh: This figure compares the Digits of Efficacy (DoE) for the chosen finite element formulations.

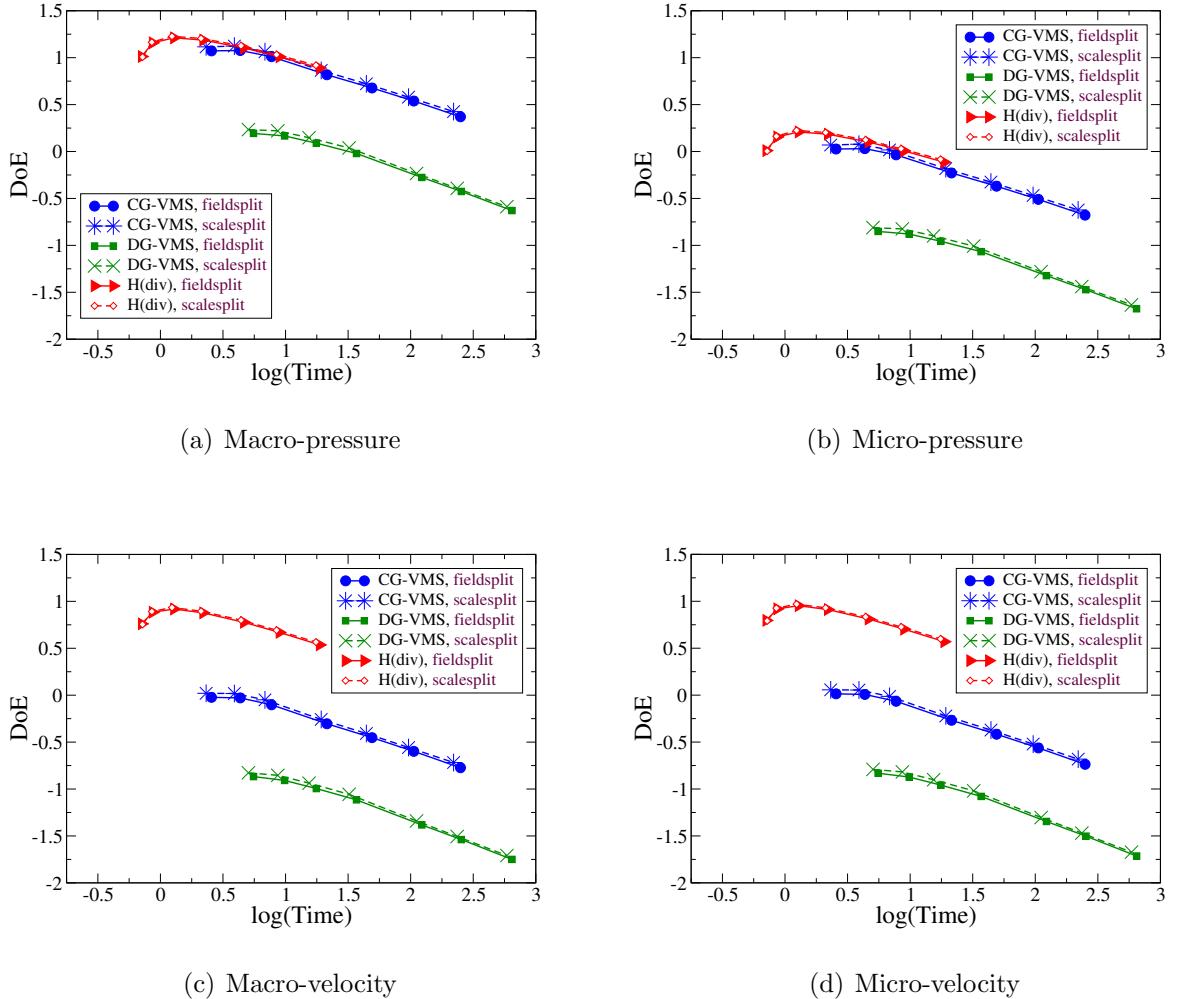


Figure 7.13: Three-dimensional problem using HEX mesh: This figure compares the Digits of Efficacy (DoE) for the chosen formulations.

Table 7.8: 3D problem: Strong-scaling results for $H(\text{div})$ formulation with same DoF count.

Field-splitting							Scale-splitting						
No. of MPI proc.	TET mesh						HEX mesh						
	Assembly	Solver	Total	KSP	Parallel eff. (%)	Time	Assembly	Solver	Total	KSP	Parallel eff. (%)		
1	9.11E-01	6.62E-01	1.57E+00	15	100	1.39E+00	1.13E+00	2.52E+00	20	100			
2	6.25E-01	5.20E-01	1.15E+00	16	68.7336	9.78E-01	7.72E-01	1.75E+00	20	71.9143			
4	4.10E-01	3.51E-01	7.60E-01	16	51.7627	7.13E-01	5.21E-01	1.23E+00	20	50.9927			
8	3.63E-01	2.70E-01	6.33E-01	16	31.092	6.72E-01	4.58E-01	1.13E+00	21	27.8429			
16	3.63E-01	3.00E-01	6.62E-01	16	14.8513	7.45E-01	4.19E-01	1.16E+00	21	13.5148			

Field-splitting							Scale-splitting						
No. of MPI proc.	TET mesh						HEX mesh						
	Assembly	Solver	Total	KSP	Parallel eff. (%)	Time	Assembly	Solver	Total	KSP	Parallel eff. (%)		
1	9.14E-01	6.19E-01	1.53E+00	15	100	1.39E+00	1.02E+00	2.41E+00	20	100			
2	6.07E-01	4.92E-01	1.10E+00	16	69.8543	9.72E-01	7.26E-01	1.70E+00	20	71.0542			
4	4.25E-01	3.36E-01	7.61E-01	16	50.4008	7.07E-01	5.32E-01	1.24E+00	20	48.6885			
8	3.81E-01	2.83E-01	6.63E-01	16	28.9085	7.48E-01	4.57E-01	1.21E+00	21	25.0311			
16	3.82E-01	2.73E-01	6.55E-01	16	14.6352	7.95E-01	4.50E-01	1.25E+00	21	12.1135			

Table 7.9: 3D problem: Strong-scaling results for CG-VMS formulation with same DoF count.

Field-splitting									
No. of MPI proc.	TET mesh					HEX mesh			
	Assembly	Solver	Total	KSP	Parallel eff. (%)	Time	Solver	Total	KSP eff. (%)
1	1.23E+01	5.01E+00	1.73E+01	13	100	1.53E+01	9.33E+00	2.46E+01	17
2	7.95E+00	3.39E+00	1.13E+01	15	76.455	9.20E+00	6.22E+00	1.54E+01	19
4	4.81E+00	1.99E+00	6.80E+00	15	63.7406	5.08E+00	3.37E+00	8.45E+00	20
8	3.16E+00	1.26E+00	4.42E+00	16	49.0052	3.53E+00	2.34E+00	5.87E+00	21
16	2.42E+00	9.54E-01	3.38E+00	16	32.1111	2.78E+00	1.66E+00	4.44E+00	21
Scale-splitting									
No. of MPI proc.	TET mesh					HEX mesh			
	Assembly	Solver	Total	KSP	Parallel eff. (%)	Time	Solver	Total	KSP eff. (%)
1	1.25E+01	4.14E+00	1.66E+01	13	100	1.53E+01	6.93E+00	2.22E+01	17
2	7.84E+00	2.80E+00	1.06E+01	15	77.9605	9.16E+00	4.64E+00	1.38E+01	19
4	4.84E+00	1.64E+00	6.48E+00	15	64.0244	5.23E+00	2.68E+00	7.91E+00	20
8	3.32E+00	1.11E+00	4.43E+00	16	46.8009	3.62E+00	1.93E+00	5.55E+00	21
16	2.44E+00	8.38E-01	3.28E+00	16	31.6313	2.75E+00	1.37E+00	4.12E+00	21

Table 7.10: 3D problem: Strong-scaling results for DG-VMS formulation with same DoF count.

Field-splitting								Scale-splitting									
No. of MPI proc.	TET mesh				HEX mesh				No. of MPI proc.	TET mesh				HEX mesh			
	Assembly	Solver	Total	KSP	Parallel eff. (%)	Time	Assembly	Solver	Total	Assembly	Solver	Total	KSP	Parallel eff. (%)			
1	8.08E+00	5.40E+00	1.35E+01	19	100	3.40E+01	2.03E+01	5.42E+01	22	100	7.81E+00	4.11E+00	1.19E+01	4.70E+01	22	100	
2	4.76E+00	8.27E+00	1.30E+01	79	51.7268	1.82E+01	1.71E+01	3.54E+01	40	76.6544	4.57E+00	6.62E+00	1.12E+01	1.24E+01	3.06E+01	40	76.8715
4	2.55E+00	5.45E+00	7.99E+00	102	42.1566	1.02E+01	9.67E+00	1.99E+01	45	68.2402	2.54E+00	4.43E+00	6.97E+00	1.02E+01	6.95E+00	45	68.6369
8	1.91E+00	3.37E+00	5.27E+00	109	31.9492	7.32E+00	5.92E+00	1.32E+01	46	51.1801	1.85E+00	2.88E+00	4.73E+00	7.41E+00	4.19E+00	46	50.6789
16	1.61E+00	2.52E+00	4.13E+00	119	20.4045	6.04E+00	3.98E+00	1.00E+01	52	33.8136	1.61E+00	2.26E+00	3.87E+00	6.00E+00	3.40E+00	52	31.2566

Table 7.11: This table illustrates three-dimensional h-size refinement for each discretization such that at each step DoF approximately doubles.

CG-VMS			DG-VMS			H(div)		
TET		HEX		TET		HEX		TET
h-size	Dof	h-size	Dof	h-size	Dof	h-size	Dof	h-size
1/13	21952	1/13	21952	1/5	24000	1/7	21952	1/8
1/17	46656	1/17	46656	1/6	41472	1/9	46686	1/10
1/21	85184	1/21	85184	1/8	98304	1/11	85184	1/13
1/28	195112	1/28	195112	1/10	192000	1/14	175616	1/17
1/36	405224	1/36	405224	1/13	421824	1/19	438976	1/22
1/45	778688	1/45	778688	1/16	786432	1/23	778688	1/28
1/57	1560896	1/57	1560896	1/20	1536000	1/29	1560896	1/35

Chapter 8

Interface Conditions for Flows in Coupled Free-Porous Media

Before we elaborate on some prior works and present our approach, we outline below the following desired properties of the interface conditions.

- (i) Interface conditions may directly stem from the balance laws and the associated jump conditions. For example, the no-penetration boundary condition at a stationary impervious wall, commonly employed in fluid mechanics, stems from the jump condition associated with the balance of mass.
- (ii) Alternatively, they may be constitutive specifications. If this is the case, they should be compatible with the balance laws and satisfy the essential invariance properties (e.g., the principle of material frame-indifference or the Galilean invariance).
- (iii) It is needless to say that they should agree with the experiments.
- (iv) They should apply to a wide variety of problems.
- (v) They should give rise to mathematical models (i.e., boundary value problems and initial boundary value problems) that are mathematically well-posed.

This chapter fills the gap in our understanding of interface conditions for flows in coupled free-porous media. Our treatment of the problem will be at the continuum

(or the so-called Darcy) scale. The specific aims of this text are twofold. *First*, to develop a framework for obtaining appropriate conditions for coupled flow dynamics at the interface of free-porous media. *Second*, to recover some popular conditions available in the literature for coupled flows as special cases of the proposed framework. Our approach will utilize the principle of virtual power and the theory of interacting continua.

8.1 Experimental observation and prior works

The two most popular approaches are the Beavers-Joseph (BJ) condition [24] and the Beavers-Joseph-Saffman (BJS) condition [159]. The experiments conducted by [24] provided the following two pieces of information regarding flows near the interface of coupled free-porous media:

- (i) The no-slip condition, commonly used for free flows at a boundary, is no longer satisfied at the interface.
- (ii) There is a jump in the tangential components of velocity on either side of the interface.

[24] also proposed an empirical relation, which advocates that the shear stress on the free flow side of the interface is linearly proportional to the jump in the tangential velocities across the interface. Based on the velocity profile and the notation introduced in **Fig. 8.1**, the BJ condition takes the following form

$$u_B - Q = \left(\frac{k^{1/2}}{\alpha} \right) \frac{\partial u}{\partial y} \Big|_{y=0^+}, \quad (8.1)$$

where $y = 0^+$ is the boundary limit point from the free flow region, k denotes the isotropic permeability of the porous medium, and α is a constant that depends only on the properties of the fluid and the porous material.

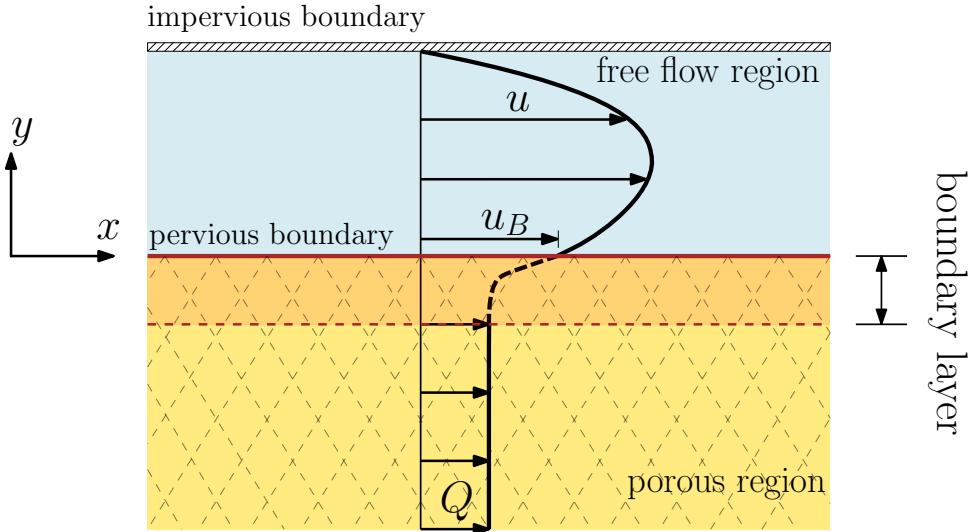


Figure 8.1: A pictorial description of the rectilinear flow in a horizontal channel between an impervious upper wall and a porous lower wall (at $y = 0$).

Later, [159] performed a statistical analysis and suggested a modification to the BJ condition, and this new condition is popularly referred to as the BJS condition. Specifically, using a one-dimensional geometrical setting and assuming uniform pressure gradient in the porous medium, [159] argued that the velocity on the porous medium side is a higher-order term compared to the velocity on the free flow side of the interface, and hence one can neglect the higher-order term. The BJS condition takes the following form

$$u_B = \left(\frac{k^{1/2}}{\alpha} \right) \frac{\partial u}{\partial y} \Big|_{y=0^+} + O(k), \quad (8.2)$$

where $O(\cdot)$ is the standard “big O notation,” which describes the limiting behavior of a function when the argument tends towards a particular value.

Although these two approaches have laid the foundation for much of the works in this field, they suffer from some drawbacks, which became clear because of new experimental and numerical studies. *First*, the slip coefficients under the BJ and BJS conditions are independent of the velocities in the free flow and porous regions. However, [115] have shown the linear dependence of the slip coefficient on the Reynolds number, so the slip coefficient can depend on the velocities. *Second*, their primary interest is free flows in a

region with a part of its boundary to be pervious due to a juxtaposed porous medium. Their approaches were aimed at replacing the slip condition with an alternate boundary condition which is appropriate for free flows due to a pervious boundary. Thus their treatments do not provide sufficient information to study flows in coupled free-porous media, as there was no discussion on appropriate boundary conditions for the flows in the porous region. *Third*, their treatment of the boundary conditions is rather ad hoc and are not amenable to generalization to other porous media models.

One can find in the literature great efforts towards extending these two empirical conditions; for example, see [110, 161]. However, the literature does not meet the desired properties of the interface conditions. We will present a framework for getting a complete set of interface conditions (not just boundary conditions for free flows due to the presence of a pervious boundary) suitable for modeling flows in coupled free-porous media.

8.2 The proposed framework

Consider a domain $\Omega \subset \mathbb{R}^{nd}$ in which an incompressible fluid flows, where “ nd ” denotes the number of spatial dimensions and \mathbb{R} denotes the set of real numbers. A spatial point in the domain is denoted by \mathbf{x} . The gradient and divergence operators with respect to \mathbf{x} are, respectively, denoted by $\text{grad}[\cdot]$ and $\text{div}[\cdot]$. The domain consists of two non-overlapping but adjoining regions: a porous region and a free flow region. See **Fig. 8.2** for a pictorial description.

8.2.1 The interface

The interface—the surface that demarcates the two regions—is denoted by Γ_{int} . The face of Γ_{int} that is adjacent to the free flow region is denoted by Γ_{free} , and the face of Γ_{int}

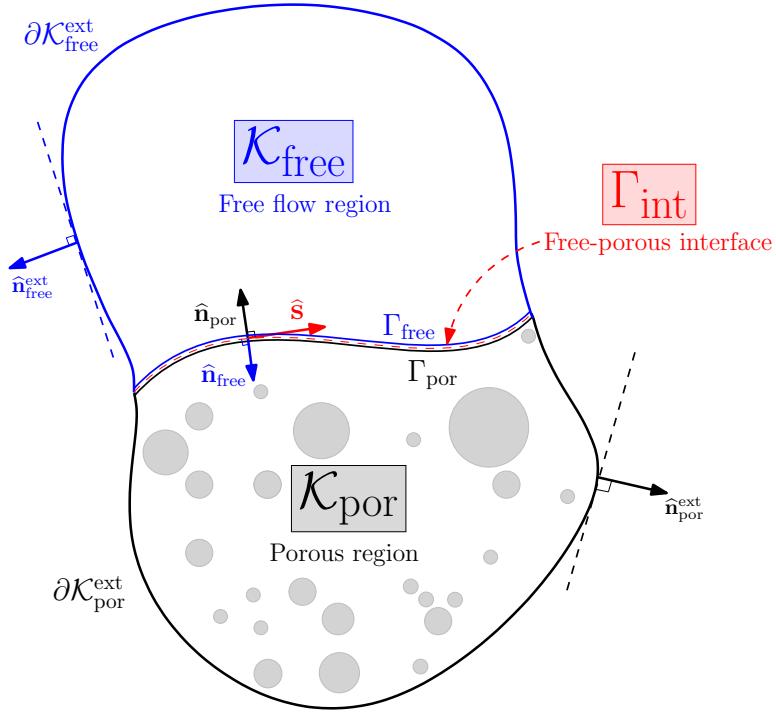


Figure 8.2: A pictorial description of coupled free-porous media. The free flow region $\mathcal{K}_{\text{free}}$ and the porous region \mathcal{K}_{por} share a common interface Γ_{int} .

that is adjacent to the porous region is denoted by Γ_{por} . Note that Γ_{int} , for our purposes, has a zero thickness, and the faces Γ_{free} and Γ_{por} have been introduced for mathematical convenience. The unit outward normal on Γ_{free} emanating away from the free flow region is denoted by $\hat{\mathbf{n}}_{\text{free}}$. Similarly, the unit outward normal on Γ_{por} emanating away from the porous region is denoted by $\hat{\mathbf{n}}_{\text{por}}$. Clearly, these normals on the interface satisfy

$$\hat{\mathbf{n}}_{\text{free}}(\mathbf{x}) + \hat{\mathbf{n}}_{\text{por}}(\mathbf{x}) = \mathbf{0} \quad \forall \mathbf{x} \in \Gamma_{\text{int}}. \quad (8.3)$$

A unit tangent vector on Γ_{int} is denoted by $\hat{\mathbf{s}}$.

8.2.2 Free flow region

We denote the region in which free flow occurs by $\mathcal{K}_{\text{free}}$, and its whole boundary and external boundary are, respectively, denoted by $\partial\mathcal{K}_{\text{free}}$ and $\partial\mathcal{K}_{\text{free}}^{\text{ext}}$. We thus have

$$\partial\mathcal{K}_{\text{free}} = \partial\mathcal{K}_{\text{free}}^{\text{ext}} \cup \Gamma_{\text{free}} \quad \text{and} \quad \partial\mathcal{K}_{\text{free}}^{\text{ext}} \cap \Gamma_{\text{free}} = \emptyset. \quad (8.4)$$

The unit outward normal to the external boundary $\mathcal{K}_{\text{free}}^{\text{ext}}$ is denoted by $\hat{\mathbf{n}}_{\text{free}}^{\text{ext}}$. We denote the velocity vector field in the free flow region by $\mathbf{v}_{\text{free}}(\mathbf{x})$, and the corresponding pressure field by $p_{\text{free}}(\mathbf{x})$. Mathematically, $\mathbf{v}_{\text{free}} : \mathcal{K}_{\text{free}} \cup \partial\mathcal{K}_{\text{free}} \rightarrow \mathbb{R}^{nd}$ and $p_{\text{free}} : \mathcal{K}_{\text{free}} \cup \partial\mathcal{K}_{\text{free}} \rightarrow \mathbb{R}$. We denote the specific body force and the stress tensor in the free flow region by $\mathbf{b}_{\text{free}}(\mathbf{x})$ and \mathbf{T}_{free} , respectively. The external boundary $\partial\mathcal{K}_{\text{free}}^{\text{ext}}$ is divided into two parts: Γ_{free}^v and Γ_{free}^t , such that

$$\Gamma_{\text{free}}^v \cup \Gamma_{\text{free}}^t = \partial\mathcal{K}_{\text{free}}^{\text{ext}} \quad \text{and} \quad \Gamma_{\text{free}}^v \cap \Gamma_{\text{free}}^t = \emptyset. \quad (8.5)$$

Γ_{free}^v is the part of the external boundary of the free flow region on which velocity boundary condition is prescribed, and Γ_{free}^t is that part of the external boundary of the free flow region on which traction boundary condition is prescribed. We thus have

$$\partial\mathcal{K}_{\text{free}} = \partial\mathcal{K}_{\text{free}}^{\text{ext}} \cup \Gamma_{\text{free}} = \Gamma_{\text{free}}^v \cup \Gamma_{\text{free}}^t \cup \Gamma_{\text{free}}. \quad (8.6)$$

We denote the prescribed velocity vector on Γ_{free}^v by $\mathbf{v}_{\text{free}}^p(\mathbf{x})$, and the prescribed traction on Γ_{free}^t by $\mathbf{t}_{\text{free}}^p(\mathbf{x})$.

8.2.3 Porous region

We denote the porous region by \mathcal{K}_{por} , and its whole boundary and external boundary are, respectively, denoted by $\partial\mathcal{K}_{\text{por}}$ and $\partial\mathcal{K}_{\text{por}}^{\text{ext}}$. Similar to the free flow region, we have

$$\partial\mathcal{K}_{\text{por}} = \partial\mathcal{K}_{\text{por}}^{\text{ext}} \cup \Gamma_{\text{por}} \quad \text{and} \quad \partial\mathcal{K}_{\text{por}}^{\text{ext}} \cap \Gamma_{\text{por}} = \emptyset. \quad (8.7)$$

The unit outward normal to the external boundary $\mathcal{K}_{\text{por}}^{\text{ext}}$ is denoted by $\hat{\mathbf{n}}_{\text{por}}^{\text{ext}}$. *The porous solid is assumed to be rigid, and its motion can be ignored.* We denote the porosity by $\phi_{\text{por}}(\mathbf{x})$. We denote the discharge velocity and the pressure of the fluid in the porous region by $\mathbf{v}_{\text{por}}(\mathbf{x})$ and $p_{\text{por}}(\mathbf{x})$, respectively. It is important to note that the discharge velocity is equal to the true (or seepage) velocity times the porosity. We denote the specific body force and the stress of the fluid in the porous region by $\mathbf{b}_{\text{por}}(\mathbf{x})$ and \mathbf{T}_{por} , respectively. We denote the interaction term for the fluid in the porous region by \mathbf{i}_{por} , which accounts for the momentum supply due to the coexistence of the other constituent – the porous solid. As mentioned earlier, the interaction term should be interpreted in the context of TIC. The external boundary $\partial\mathcal{K}_{\text{por}}^{\text{ext}}$ is divided into two parts: Γ_{por}^v and Γ_{por}^t , such that

$$\Gamma_{\text{por}}^v \cup \Gamma_{\text{por}}^t = \partial\mathcal{K}_{\text{por}}^{\text{ext}} \quad \text{and} \quad \Gamma_{\text{por}}^v \cap \Gamma_{\text{por}}^t = \emptyset. \quad (8.8)$$

Γ_{por}^v is the part of the external boundary of the porous region on which velocity boundary condition is prescribed, and Γ_{por}^t is that part of the external boundary of the porous region on which traction boundary condition is prescribed. We thus have

$$\partial\mathcal{K}_{\text{por}} = \partial\mathcal{K}_{\text{por}}^{\text{ext}} \cup \Gamma_{\text{por}} = \Gamma_{\text{por}}^v \cup \Gamma_{\text{por}}^t \cup \Gamma_{\text{por}}. \quad (8.9)$$

We denote the prescribed velocity on Γ_{por}^v by $\mathbf{v}_{\text{por}}^p(\mathbf{x})$ and the prescribed traction on Γ_{por}^t by $\mathbf{t}_{\text{por}}^p(\mathbf{x})$.

8.2.4 Fluid properties

The dynamic coefficient of viscosity of the fluid is denoted by μ . The true density of the fluid in the free flow region is denoted by γ_{free} , and the corresponding quantity of the fluid in the porous region is denoted by γ_{por} . Note that the bulk density of the fluid in

porous media is equal to the true density of the fluid times the porosity of the porous medium. The interface conditions are derived under the realistic case of $\gamma_{\text{free}} = \gamma_{\text{por}} = \gamma$.

8.2.5 Kinematically admissible and virtual fields

We introduce the following space for the pairs of vector fields defined on free flow and porous regions

$$\mathcal{W} := \{(\mathbf{w}_{\text{free}}(\mathbf{x}), \mathbf{w}_{\text{por}}(\mathbf{x})) \mid \mathbf{w}_{\text{free}} : \mathcal{K}_{\text{free}} \cup \partial\mathcal{K}_{\text{free}} \rightarrow \mathbb{R}^{nd}, \mathbf{w}_{\text{por}} : \mathcal{K}_{\text{por}} \cup \partial\mathcal{K}_{\text{por}} \rightarrow \mathbb{R}^{nd}\}. \quad (8.10)$$

For a given pair of vector fields $(\mathbf{w}_{\text{free}}, \mathbf{w}_{\text{por}}) \in \mathcal{W}$, we introduce the following normal components:

$$w_{\text{free}}^{(n)}(\mathbf{x}) := \mathbf{w}_{\text{free}}(\mathbf{x}) \cdot \hat{\mathbf{n}}_{\text{free}}(\mathbf{x}), \quad (8.11a)$$

$$w_{\text{por}}^{(n)}(\mathbf{x}) := \mathbf{w}_{\text{por}}(\mathbf{x}) \cdot \hat{\mathbf{n}}_{\text{por}}(\mathbf{x}), \quad (8.11b)$$

and the following decomposition:

$$\mathbf{w}_{\text{free}}(\mathbf{x}) = w_{\text{free}}^{(n)}(\mathbf{x})\hat{\mathbf{n}}_{\text{free}}(\mathbf{x}) + \mathbf{w}_{\text{free}}^*(\mathbf{x}), \quad (8.12a)$$

$$\mathbf{w}_{\text{por}}(\mathbf{x}) = w_{\text{por}}^{(n)}(\mathbf{x})\hat{\mathbf{n}}_{\text{por}}(\mathbf{x}) + \mathbf{w}_{\text{por}}^*(\mathbf{x}), \quad (8.12b)$$

where $\mathbf{w}_{\text{free}}^*(\mathbf{x})$ and $\mathbf{w}_{\text{por}}^*(\mathbf{x})$ denote the corresponding tangential components of the vector fields.

We refer to a pair of vector fields $(\mathbf{w}_{\text{free}}(\mathbf{x}), \mathbf{w}_{\text{por}}(\mathbf{x})) \in \mathcal{W}$ to be *kinematically admissible* if the following properties are satisfied:

(i) $\text{div}[\mathbf{w}_{\text{free}}] = 0$ in $\mathcal{K}_{\text{free}}$ and $\text{div}[\mathbf{w}_{\text{por}}] = 0$ in \mathcal{K}_{por} ,

(ii) $w_{\text{free}}^{(n)}(\mathbf{x}) + w_{\text{por}}^{(n)}(\mathbf{x}) = 0$ on the interface Γ_{int} , and

- (iii) $\mathbf{w}_{\text{free}}(\mathbf{x})$ and $\mathbf{w}_{\text{por}}(\mathbf{x})$ satisfy the velocity boundary conditions on the external boundary (i.e., on Γ_{free}^v and Γ_{por}^v , respectively).

We denote the set of all kinematically admissible pairs of vector fields by \mathcal{V} . Certainly, the exact velocity fields are kinematically admissible; that is $(\mathbf{v}_{\text{free}}(\mathbf{x}), \mathbf{v}_{\text{por}}(\mathbf{x})) \in \mathcal{V}$.

We refer to a pair of vector fields $(\mathbf{w}_{\text{free}}(\mathbf{x}), \mathbf{w}_{\text{por}}(\mathbf{x})) \in \mathcal{W}$ to be a pair of *virtual vector fields* if the first two properties under kinematical admissibility are met, and $\mathbf{w}_{\text{free}}(\mathbf{x})$ and $\mathbf{w}_{\text{por}}(\mathbf{x})$ vanish on Γ_{free}^v and Γ_{por}^v , respectively. We denote the set of all pairs of virtual vector fields by $\tilde{\mathcal{V}}$.

8.2.6 Other notation for convenience

We occasionally use the following notation:

$$\mathbf{L}_{\text{free}} = \text{grad}[\mathbf{v}_{\text{free}}], \mathbf{L}_{\text{por}} = \text{grad}[\mathbf{v}_{\text{por}}], \mathbf{D}_{\text{free}} = \frac{1}{2} (\mathbf{L}_{\text{free}} + \mathbf{L}_{\text{free}}^T), \text{ and } \mathbf{D}_{\text{por}} = \frac{1}{2} (\mathbf{L}_{\text{por}} + \mathbf{L}_{\text{por}}^T). \quad (8.13)$$

8.2.7 Proposed principle of virtual power

The mathematical statement of the proposed principle of virtual power for flows in coupled free-porous media, which will be in the form of balance of virtual power, can be written as follows:

Find $(\mathbf{v}_{\text{free}}(\mathbf{x}), \mathbf{v}_{\text{por}}(\mathbf{x})) \in \mathcal{V}$ such that the following two properties are met:

$$(P1) \quad \mathcal{P}^{(\text{internal})} = \mathcal{P}^{(\text{external})} \quad \forall (\mathbf{w}_{\text{free}}(\mathbf{x}), \mathbf{w}_{\text{por}}(\mathbf{x})) \in \tilde{\mathcal{V}} \text{ and} \quad (8.14)$$

$$(P2) \quad \mathcal{P}^{(\text{internal})} = 0 \quad \forall (\mathbf{w}_{\text{free}}(\mathbf{x}), \mathbf{w}_{\text{por}}(\mathbf{x})) \in \mathcal{W}_{\text{rigid}}, \quad (8.15)$$

where the internal virtual power expended (i.e., virtual stress power) in the free flow region is given by

$$\mathcal{P}_{\text{free}}^{(\text{internal})} := \int_{\mathcal{K}_{\text{free}}} \mathbf{T}_{\text{free}} \cdot \text{grad}[\mathbf{w}_{\text{free}}] \, d\Omega. \quad (8.16)$$

The internal virtual power expended in the porous region is written as

$$\mathcal{P}_{\text{por}}^{(\text{internal})} := \mathcal{P}_{\text{por, stress}}^{(\text{internal})} + \mathcal{P}_{\text{por, interactions}}^{(\text{internal})}, \quad (8.17)$$

where virtual stress power in the porous region is defined as

$$\mathcal{P}_{\text{por, stress}}^{(\text{internal})} := \int_{\mathcal{K}_{\text{por}}} \mathbf{T}_{\text{por}} \cdot \text{grad}[\mathbf{w}_{\text{por}}] \, d\Omega, \quad (8.18)$$

and the internal virtual power expended due to interactions between the constituents in the porous region is written as

$$\mathcal{P}_{\text{por, interactions}}^{(\text{internal})} := \int_{\mathcal{K}_{\text{por}}} \mathbf{i}_{\text{por}} \cdot \left(\mathbf{w}_{\text{por}} - \overrightarrow{\mathbf{w}_{\text{por}}^{(\text{solid})}} \right)^0 \, d\Omega = \int_{\mathcal{K}_{\text{por}}} \mathbf{i}_{\text{por}} \cdot \mathbf{w}_{\text{por}} \, d\Omega. \quad (8.19)$$

In the above equation, $\mathbf{w}_{\text{por}}^{(\text{solid})}$ denotes the vector field associated with the porous solid. Since we assumed the porous solid to be rigid and neglected its motion, this term becomes zero. The internal virtual power expended at the interface is written as

$$\mathcal{P}_{\text{int}}^{(\text{internal})} := \int_{\Gamma_{\text{int}}} \delta\Psi \, d\Gamma, \quad (8.20)$$

where $\delta\Psi$ denotes the virtual power expended density at the interface and depends on both the true velocity fields and their virtual counterparts. The total internal virtual power expended takes the following form

$$\mathcal{P}^{(\text{internal})} := \mathcal{P}_{\text{free}}^{(\text{internal})} + \mathcal{P}_{\text{por}}^{(\text{internal})} + \mathcal{P}_{\text{int}}^{(\text{internal})}. \quad (8.21)$$

The total external virtual power expended takes the following form

$$\begin{aligned} \mathcal{P}^{(\text{external})} := & \underbrace{\int_{\mathcal{K}_{\text{free}}} \gamma \mathbf{b}_{\text{free}} \cdot \mathbf{w}_{\text{free}} \, d\Omega + \int_{\Gamma_{\text{free}}^t} \mathbf{t}_{\text{free}}^p \cdot \mathbf{w}_{\text{free}} \, d\Gamma}_{\text{external virtual power expended on the free flow region}} \\ & + \underbrace{\int_{\mathcal{K}_{\text{por}}} \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} \cdot \mathbf{w}_{\text{por}} \, d\Omega + \int_{\Gamma_{\text{por}}^t} \mathbf{t}_{\text{por}}^p \cdot \mathbf{w}_{\text{por}} \, d\Gamma}_{\text{external virtual power expended on the porous region}}. \end{aligned} \quad (8.22)$$

We will show that an appropriate set of interface conditions can be derived by prescribing a functional form for $\delta\Psi$, and this prescription will be a constitutive specification. We place the following restrictions on the functional, and these restrictions are based on either invariance requirements, physical properties or convenience.

- (i) *Positive semi-definiteness.* The total power expended at the interface should be *physically* non-negative. This can be ensured by assuming Ψ to be a positive semi-definite functional. Mathematically,

$$\Psi[\boldsymbol{\chi}] \geq 0 \quad \forall \boldsymbol{\chi}. \quad (8.23)$$

- (ii) *Dependence of Ψ on velocities.* We take the set of variables for the functional dependence of Ψ as

$$\boldsymbol{\chi} = \{\mathbf{v}_{\text{free}}^*(\mathbf{x}), \mathbf{v}_{\text{por}}^*(\mathbf{x}), v_n(\mathbf{x})\}, \quad (8.24)$$

where

$$v_n(\mathbf{x}) := v_{\text{free}}^{(n)}(\mathbf{x}). \quad (8.25)$$

Recall that the tangential velocities have been defined in equation (8.12). Since the true fluid densities in the porous and free flow regions are assumed to be the same,

the balance of mass across the interface implies that

$$v_n(\mathbf{x}) = -v_{\text{por}}^{(n)}(\mathbf{x}). \quad (8.26)$$

The chosen functional dependence will imply that

$$\delta\Psi = \frac{\partial\Psi}{\partial \mathbf{v}_{\text{free}}^*} \cdot \delta \mathbf{v}_{\text{free}}^* + \frac{\partial\Psi}{\partial \mathbf{v}_{\text{por}}^*} \cdot \delta \mathbf{v}_{\text{por}}^* + \frac{\partial\Psi}{\partial v_n} \cdot \delta v_n. \quad (8.27)$$

Noting that $\delta \mathbf{v}_{\text{free}}$ and $\delta \mathbf{v}_{\text{por}}$ are relative velocities with respect to the rigid porous solid, they vanish under a rigid body motion of the entire domain. Hence, $\delta\Psi$ vanishes under a rigid body motion of the virtual velocities. This point is important to satisfy the statement (P2) under the proposed principle of virtual power.

(iii) *Invariance.* We require the constitutive relations emanating from the functional Ψ to satisfy the principle of material frame-indifference. This implies that Ψ can depend only on the following individual and joint invariants [165]:

$$\mathbf{v}_{\text{free}}^* \cdot \mathbf{v}_{\text{free}}^*, \mathbf{v}_{\text{por}}^* \cdot \mathbf{v}_{\text{por}}^*, \mathbf{v}_{\text{free}}^* \cdot \mathbf{v}_{\text{por}}^* \text{ and } v_n.$$

8.3 Derivation of interface conditions and field equations

The main consequence of the statement (P2) is the symmetry of the Cauchy stresses in both the regions, which is equivalent to the balance of angular momentum. That is,

$$\mathbf{T}_{\text{free}}(\mathbf{x}) = \mathbf{T}_{\text{free}}^T(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{K}_{\text{free}} \quad \text{and} \quad \mathbf{T}_{\text{por}}(\mathbf{x}) = \mathbf{T}_{\text{por}}^T(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{K}_{\text{por}}. \quad (8.28)$$

8.3.1 Handling internal constraints

The Cauchy stresses under the constrained motion due to internal constraints can be written as:

$$\mathbf{T}_{\text{free}}(\mathbf{x}) = -p_{\text{free}}(\mathbf{x})\mathbf{I} + \mathbf{T}_{\text{free}}^{\text{extra}}(\mathbf{x}) \text{ and} \quad (8.29\text{a})$$

$$\mathbf{T}_{\text{por}}(\mathbf{x}) = -p_{\text{por}}(\mathbf{x})\mathbf{I} + \mathbf{T}_{\text{por}}^{\text{extra}}(\mathbf{x}), \quad (8.29\text{b})$$

where the extra stresses, $\mathbf{T}_{\text{free}}^{\text{extra}}$ and $\mathbf{T}_{\text{por}}^{\text{extra}}$, which should be prescribed through constitutive specifications.

8.3.2 Consequences of (P1) statement

Using Green's identity and noting that the virtual velocity fields vanish on Γ_{free}^v and Γ_{por}^v , the (P1) statement (8.14) can be rewritten as

$$\begin{aligned} & \int_{\Gamma_{\text{free}}^t} \mathbf{w}_{\text{free}} \cdot \{\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}}^{\text{ext}} - \mathbf{t}_{\text{free}}^p\} d\Gamma - \int_{\mathcal{K}_{\text{free}}} \mathbf{w}_{\text{free}} \cdot \{\text{div}[\mathbf{T}_{\text{free}}] + \gamma \mathbf{b}_{\text{free}}\} d\Omega \\ & + \int_{\Gamma_{\text{por}}^t} \mathbf{w}_{\text{por}} \cdot \{\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}}^{\text{ext}} - \mathbf{t}_{\text{por}}^p\} d\Gamma - \int_{\mathcal{K}_{\text{por}}} \mathbf{w}_{\text{por}} \cdot \{\text{div}[\mathbf{T}_{\text{por}}] + \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} - \mathbf{i}_{\text{por}}\} d\Omega \\ & + \int_{\Gamma_{\text{int}}} \left\{ \mathbf{w}_{\text{free}} \cdot \mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}} + \mathbf{w}_{\text{por}} \cdot \mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}} + \mathbf{w}_{\text{free}}^* \cdot \frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*} + \mathbf{w}_{\text{por}}^* \cdot \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*} + w_n \cdot \frac{\partial \Psi}{\partial v_n} \right\} d\Gamma = 0 \\ & \forall (\mathbf{w}_{\text{free}}, \mathbf{w}_{\text{por}}) \in \tilde{\mathcal{V}}. \end{aligned} \quad (8.30)$$

We now invoke the arbitrariness of the fields $\mathbf{w}_{\text{free}}(\mathbf{x})$ and $\mathbf{w}_{\text{por}}(\mathbf{x})$ but respecting the requirements of kinematic admissibility. The first two terms give rise to the following governing equations for the free flow region except along the part of the boundary that shares with the interface:

$$\text{div}[\mathbf{T}_{\text{free}}] + \gamma \mathbf{b}_{\text{free}} = \mathbf{0} \quad \text{in } \mathcal{K}_{\text{free}}, \quad (8.31\text{a})$$

$$\operatorname{div}[\mathbf{v}_{\text{free}}] = 0 \quad \text{in } \mathcal{K}_{\text{free}}, \quad (8.31\text{b})$$

$$\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}}^{\text{ext}}(\mathbf{x}) = \mathbf{t}_{\text{free}}^{\text{p}}(\mathbf{x}) \quad \text{on } \Gamma_{\text{free}}^t, \text{ and} \quad (8.31\text{c})$$

$$\mathbf{v}_{\text{free}}(\mathbf{x}) = \mathbf{v}_{\text{free}}^{\text{p}}(\mathbf{x}) \quad \text{on } \Gamma_{\text{free}}^v. \quad (8.31\text{d})$$

The third and fourth terms give rise to the following governing equations for the porous region except along the part of the boundary that shares with the interface:

$$\operatorname{div}[\mathbf{T}_{\text{por}}] + \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} - \mathbf{i}_{\text{por}} = \mathbf{0} \quad \text{in } \mathcal{K}_{\text{por}}, \quad (8.32\text{a})$$

$$\operatorname{div}[\mathbf{v}_{\text{por}}] = 0 \quad \text{in } \mathcal{K}_{\text{por}}, \quad (8.32\text{b})$$

$$\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}}^{\text{ext}}(\mathbf{x}) = \mathbf{t}_{\text{por}}^{\text{p}}(\mathbf{x}) \quad \text{on } \Gamma_{\text{por}}^t, \text{ and} \quad (8.32\text{c})$$

$$\mathbf{v}_{\text{por}}(\mathbf{x}) = \mathbf{v}_{\text{por}}^{\text{p}}(\mathbf{x}) \quad \text{on } \Gamma_{\text{por}}^v. \quad (8.32\text{d})$$

Noting the decomposition given in equation (8.12), the fifth term gives rise to the following interface conditions on Γ_{int} :

$$v_{\text{free}}^{(n)}(\mathbf{x}) + v_{\text{por}}^{(n)}(\mathbf{x}) = 0, \quad (8.33\text{a})$$

$$\hat{\mathbf{n}}_{\text{free}}(\mathbf{x}) \cdot \mathbf{T}_{\text{free}}(\mathbf{x}) \hat{\mathbf{n}}_{\text{free}}(\mathbf{x}) + \frac{\partial \Psi}{\partial v_n} = \hat{\mathbf{n}}_{\text{por}}(\mathbf{x}) \cdot \mathbf{T}_{\text{por}}(\mathbf{x}) \hat{\mathbf{n}}_{\text{por}}(\mathbf{x}), \quad (8.33\text{b})$$

$$\hat{\mathbf{s}}(\mathbf{x}) \cdot \mathbf{T}_{\text{free}}^{\text{extra}} \hat{\mathbf{n}}_{\text{free}}(\mathbf{x}) = -\hat{\mathbf{s}}(\mathbf{x}) \cdot \frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*}, \text{ and} \quad (8.33\text{c})$$

$$\hat{\mathbf{s}}(\mathbf{x}) \cdot \mathbf{T}_{\text{por}}^{\text{extra}} \hat{\mathbf{n}}_{\text{por}}(\mathbf{x}) = -\hat{\mathbf{s}}(\mathbf{x}) \cdot \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*}. \quad (8.33\text{d})$$

Equation (8.33a) is in fact the jump condition corresponding to the balance of mass. The other three conditions are in general not jump conditions and they stem from a constitutive specification in the form of a prescription for the functional Ψ . If Ψ is independent of v_n (which is assumed in section 8.4 to obtain some popular conditions like the BJ and BJS conditions) then the second condition (8.33b) will reduce to the normal component of the jump condition for the balance of linear momentum. To summarize,

the complete set of governing equations for flows in coupled free-porous media is:

- the equations in the free flow region along with the boundary conditions on the external boundary (not including Γ_{int}) of the region (8.31a)–(8.31d),
- the equations in the porous region along with the boundary conditions on the external boundary (not including Γ_{int}) of the region (8.32a)–(8.32d),
- the symmetry of Cauchy stresses (8.28),
- the decomposition of Cauchy stresses (8.29a)–(8.29b),
- the interface conditions (8.33a)–(8.33d) and
- the (prescribed) constitutive specifications for $\mathbf{T}_{\text{free}}^{\text{extra}}$, $\mathbf{T}_{\text{por}}^{\text{extra}}$, \mathbf{i}_{por} and Ψ .

The solution fields will be $\mathbf{v}_{\text{free}}(\mathbf{x})$, $\mathbf{v}_{\text{por}}(\mathbf{x})$, $p_{\text{free}}(\mathbf{x})$ and $p_{\text{por}}(\mathbf{x})$.

8.4 Special cases

We now show the BJ and BJS conditions, and the no-slip condition (which is commonly employed in the fluid mechanics for free flows) are, respectively, special cases and a limiting case of the proposed framework. The following assumptions will be common to all the mentioned conditions:

- (A1) The normal component of the velocity at the interface does not contribute towards the power expended density at the interface. That is, Ψ is independent of v_n .
- (A2) Ψ is a quadratic functional of the tangential (relative) velocities, and the invariance requirements demand that this functional has to be in terms of individual and joint

invariants of the tangential (relative) velocities. Thus, mathematically, we write the functional as

$$\Psi[\mathbf{v}_{\text{free}}^*, \mathbf{v}_{\text{por}}^*, v_n] = \alpha_{11} \mathbf{v}_{\text{free}}^* \cdot \mathbf{v}_{\text{free}}^* + 2\alpha_{12} \mathbf{v}_{\text{free}}^* \cdot \mathbf{v}_{\text{por}}^* + \alpha_{22} \mathbf{v}_{\text{por}}^* \cdot \mathbf{v}_{\text{por}}^*, \quad (8.34)$$

where α_{11} , α_{12} and α_{22} are constants, and $\mathbf{v}_{\text{free}}^*$ and $\mathbf{v}_{\text{por}}^*$ are the tangential velocities.

(A3) The non-negativity of Ψ is enforced by assuming that

$$\alpha_{11}\alpha_{22} \geq \alpha_{12}^2. \quad (8.35)$$

(A4) The Stokes model is assumed to describe the flow in the free flow region. That is, the flow in the free flow region is assumed to be a creeping flow, which implies the following

$$\mathbf{T}_{\text{free}}^{\text{extra}} = 2\mu \mathbf{D}_{\text{free}}. \quad (8.36)$$

The above assumptions give rise to the following interface conditions for the tangential component of the tractions:

$$\hat{\mathbf{s}} \cdot \mathbf{T}_{\text{free}}^{\text{extra}} \hat{\mathbf{n}}_{\text{free}} = -\frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*} \cdot \hat{\mathbf{s}} = -2(\alpha_{11} \mathbf{v}_{\text{free}}^* + \alpha_{12} \mathbf{v}_{\text{por}}^*) \cdot \hat{\mathbf{s}} \quad \text{on } \Gamma_{\text{free}} \text{ and} \quad (8.37a)$$

$$\hat{\mathbf{s}} \cdot \mathbf{T}_{\text{por}}^{\text{extra}} \hat{\mathbf{n}}_{\text{por}} = -\frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*} \cdot \hat{\mathbf{s}} = -2(\alpha_{12} \mathbf{v}_{\text{free}}^* + \alpha_{22} \mathbf{v}_{\text{por}}^*) \cdot \hat{\mathbf{s}} \quad \text{on } \Gamma_{\text{por}}, \quad (8.37b)$$

where $\hat{\mathbf{s}}(\mathbf{x})$ denotes an arbitrary unit tangent vector field along the interface.

8.4.1 Beavers-Joseph condition

The BJ condition can be obtained by further making the following choices:

$$\alpha_{11} = \alpha_{22} = \frac{\alpha\mu\sqrt{3}}{2\sqrt{\text{tr}[\mathbf{K}]}} \quad \text{and} \quad \alpha_{12} = \frac{-\alpha\mu\sqrt{3}}{2\sqrt{\text{tr}[\mathbf{K}]}} , \quad (8.38)$$

where $\text{tr}[\cdot]$ denotes the trace of a second-order tensor. Then equation (8.37a) will reduce to

$$\hat{\mathbf{s}} \cdot (-2\mu \mathbf{D}_{\text{free}})\hat{\mathbf{n}}_{\text{free}} = \frac{\alpha\mu\sqrt{3}}{\sqrt{\text{tr}[\mathbf{K}]}} \hat{\mathbf{s}} \cdot (\mathbf{v}_{\text{free}} - \mathbf{v}_{\text{por}}). \quad (8.39)$$

which is the “boundary” condition proposed in [24] for the free flow region due to the presence of a previous boundary. By aligning the coordinate axes similar to the one shown in **Fig. 8.1** and by taking the x-component of \mathbf{v}_{por} to be Q , one will get an expression similar to the one provided in [24] (cf. equation (8.1)). It should be however noted that [24] do not provide a corresponding condition for the flow in the porous media, which lies on the other side of the interface.

On the other hand, using the proposed framework, one can obtain a corresponding condition for the flow on the other side of the interface (i.e., the porous medium); which is needed if one wants to simulate a coupled flow in both free and porous regions. Using equation (8.37b), the interface condition on Γ_{por} can be written as

$$\hat{\mathbf{s}} \cdot \mathbf{T}_{\text{por}}^{\text{extra}} \hat{\mathbf{n}}_{\text{por}} = \frac{\alpha\mu'\sqrt{3}}{\sqrt{\text{tr}[\mathbf{K}]}} \hat{\mathbf{s}} \cdot (\mathbf{v}_{\text{free}} - \mathbf{v}_{\text{por}}). \quad (8.40)$$

8.4.1.1 A discussion on the BJ condition

The velocity field in the porous region is assumed to be known *a priori*. Moreover, the flow in the porous region is tacitly assumed to be uniform beyond a boundary layer (see **Fig. 8.1**). But the velocity field in the porous region is seldom known *a priori* and this is particularly true in the case of flows in coupled free-porous media. Even if the velocity field in the porous region is known, this field will not be uniform due to spatial heterogeneity of medium properties (e.g., permeability). (Heterogeneity is inherent to the two application problems that we discussed in the introduction.) This will create an ambiguity in assigning a value to Q (cf. equation (8.1)). Specifically, at what depth one

has to sample the (horizontal or tangential) velocity to specify Q (cf. **Fig. 8.1**).

Last but not least, the BJ condition may not be compatible with all porous media model. For example, if the flow in the porous region is modeled using the Darcy model, for which, $\mathbf{T}_{\text{por}}^{(\text{extra})} = \mathbf{0}$. Equation (8.40) will then imply that

$$\hat{\mathbf{s}} \cdot (\mathbf{v}_{\text{free}} - \mathbf{v}_{\text{por}}) = 0,$$

which, based on the BJ condition (8.39), will further imply that

$$\hat{\mathbf{s}} \cdot \mathbf{D}_{\text{free}} \hat{\mathbf{n}} = 0.$$

But this condition will not be met in general, as, for example, the horizontal velocity can depend on the y -coordinate or the vertical velocity can depend on the x -coordinate.

8.4.2 Beavers-Joseph-Saffman condition

In addition to the aforementioned four assumptions (A1)–(A4), we make the following choices to obtain the BJS condition:

$$\alpha_{11} = \alpha_{22} = \frac{\alpha\mu\sqrt{3}}{2\sqrt{\text{tr}[\mathbf{K}]}} \quad \text{and} \quad \alpha_{12} = 0. \quad (8.41)$$

Then, using equation (8.37a), the *boundary* condition at Γ_{free} for the flow in the free region due to a juxtaposed porous region takes the following form

$$\hat{\mathbf{s}} \cdot (-2\mu \mathbf{D}_{\text{free}}) \hat{\mathbf{n}}_{\text{free}} = \frac{\alpha\mu\sqrt{3}}{\sqrt{\text{tr}[\mathbf{K}]}} \hat{\mathbf{s}} \cdot \mathbf{v}_{\text{free}}. \quad (8.42)$$

Using equation (8.37b), the interface condition on Γ_{por} takes the following form

$$\hat{\mathbf{s}} \cdot \mathbf{T}_{\text{free}}^{\text{extra}} \hat{\mathbf{n}}_{\text{free}} = \frac{\alpha\mu\sqrt{3}}{\sqrt{\text{tr}[\mathbf{K}]}} \hat{\mathbf{s}} \cdot \mathbf{v}_{\text{por}}. \quad (8.43)$$

8.4.2.1 A discussion on the BJS condition

Since the BJS condition (8.42) does not contain Q (the mean velocity in the porous region beyond the boundary layer), it does not assume the velocity field in the porous region is neither known *a priori* nor uniform. However, the BJS condition need not be compatible with all porous media models. If one again considers the Darcy model to describe the flow in the porous region, equation (8.43) implies that $\mathbf{v}_{\text{por}} = \mathbf{0}$ – the no-slip boundary condition for the porous region along the interface – which is not what has been observed in the experiments [24].

On the other hand, if one uses the Darcy-Brinkman model, for which $\mathbf{T}_{\text{por}}^{\text{extra}} = 2\mu\mathbf{D}_{\text{por}}$, the BJS condition will be compatible with the chosen model. Saffman did recognize that his condition is actually compatible with the Darcy-Brinkman model and not the Darcy model. However, by using asymptotic analysis, he argued that solutions from the Darcy-Brinkman model and the Darcy model do not differ significantly outside the boundary layer, and the size of the boundary layer is in the order of the square-root of the (trace of) permeability.

8.4.3 No-slip condition

The classical no-slip condition can be obtained by making the following choices for the constants:

$$\alpha_{11} = \frac{\alpha}{2\sqrt{\text{tr}[\mathbf{K}]}} , \alpha_{22} = 0 \quad \text{and} \quad \alpha_{12} = 0, \quad (8.44)$$

and then by letting $\text{tr}[\mathbf{K}] \rightarrow 0$. To wit, based on the choices made in equation (8.44), the interface condition (8.37a) reduces to the following:

$$\hat{\mathbf{s}} \cdot \mathbf{v}_{\text{free}}^* = - \left(\frac{\sqrt{\text{tr}[\mathbf{K}]}}{\alpha} \right) \hat{\mathbf{s}} \cdot \mathbf{T}_{\text{free}}^{\text{extra}} \hat{\mathbf{n}}_{\text{free}}. \quad (8.45)$$

By letting $\text{tr}[\mathbf{K}] \rightarrow 0$ and noting that $\hat{\mathbf{s}}$ is an arbitrary tangent vector along the interface, one can conclude that $\mathbf{v}_{\text{free}}^* = \mathbf{0}$ on Γ_{free} , which is the no-slip condition. Note that $\text{tr}[\mathbf{K}] \rightarrow 0$ basically implies that the boundary is impervious, and the no-slip boundary condition is typically enforced at an impervious boundary in an uncoupled free flow.

8.5 Minimum power theorem for a class of coupled flows

It is well-known that an uncoupled creeping flow, which is governed by the incompressible Stokes equations, enjoys a minimum power theorem [76]. It has also been established that an uncoupled flow through porous media based on either Darcy equations or Darcy-Brinkman equations enjoys a minimum power theorem [163]. *It is thus natural to ask whether a flow in coupled free-porous media enjoys a minimum power theorem.*

We now show that the answer to this question is affirmative for a class of coupled flows. This class of flows is characterized by these two requirements:

(R1) There exists two potentials, Φ_{free} and Φ_{por} , with the following properties:

- (i) They satisfy the form-invariance and the invariance under a Euclidean transformation (i.e., they satisfy the principle of material frame indifference). Specifically these potentials can be expressed as $\Phi_{\text{free}}[\mathbf{D}_{\text{free}}]$ and $\Phi_{\text{por}}[\mathbf{D}_{\text{por}}, \mathbf{v}_{\text{por}}]$.
- (ii) They provide the constitutive relations of the following form for the extra Cauchy stresses and the interaction term:

$$\mathbf{T}_{\text{free}}^{\text{extra}} = \frac{\partial \Phi_{\text{free}}}{\partial \mathbf{D}_{\text{free}}}, \quad (8.46a)$$

$$\mathbf{T}_{\text{por}}^{\text{extra}} = \frac{\partial \Phi_{\text{por}}}{\partial \mathbf{D}_{\text{por}}}, \text{ and} \quad (8.46b)$$

$$\mathbf{i}_{\text{por}} = \frac{\partial \Phi_{\text{por}}}{\partial \mathbf{v}_{\text{por}}} \quad (8.46c)$$

(iii) Each of the potentials has a positive definite Hessian.

(R2) The functional Ψ has a positive definite Hessian.

The requirement (R2) is in addition to the properties that outlined in §8.2 for Ψ to satisfy. It is easy to construct Ψ to have a positive definite Hessian; the functional (8.34) satisfying the condition (8.35) is one such example.

8.5.1 On construction of the potentials

For many popular uncoupled free flow models (e.g., Stokes equations) and porous media models (e.g., Darcy equations, Darcy-Brinkman equations), the rate of internal dissipation density satisfies the conditions (8.46a)–(8.46c). One can take the same approach to construct the potentials Φ_{free} and Φ_{por} even for the case of coupled flows. This approach can be best illustrated by the following examples.

Under the Stokes model, the Cauchy stress and the extra Cauchy stress are given by

$$\mathbf{T}_{\text{free}} = -p_{\text{free}} \mathbf{I} + 2\mu \mathbf{D}_{\text{free}} = -p_{\text{free}} \mathbf{I} + \mathbf{T}_{\text{free}}^{\text{extra}}, \quad (8.47)$$

and the rate of internal dissipation density is given by

$$2\mu \mathbf{D}_{\text{free}} \cdot \mathbf{D}_{\text{free}}.$$

Clearly, by choosing the potential Φ_{free} to be

$$2\Phi_{\text{free}}[\mathbf{v}_{\text{free}}] = 2\mu \mathbf{D}_{\text{free}} \cdot \mathbf{D}_{\text{free}}, \quad (8.48)$$

one can satisfy the requirement (8.46a). Similarly, under the Darcy model, the extra Cauchy stress and interaction term are, respectively, given by

$$\mathbf{T}_{\text{por}}^{\text{extra}} = \mathbf{0} \quad \text{and} \quad \mathbf{i}_{\text{por}} = \mu \mathbf{K}^{-1} \mathbf{v}_{\text{por}}(\mathbf{x}). \quad (8.49)$$

By choosing the potential Φ_{por} to be

$$2\Phi_{\text{por}}[\mathbf{v}_{\text{por}}] = \underbrace{\mu \mathbf{K}^{-1} \mathbf{v}_{\text{por}}(\mathbf{x}) \cdot \mathbf{v}_{\text{por}}(\mathbf{x})}_{\text{rate of internal dissipation density}}, \quad (8.50)$$

one can satisfy the requirements (8.46b) and (8.46c). Under the Darcy-Brinkman model, the extra Cauchy stress and interaction term are, respectively, given by

$$\mathbf{T}_{\text{por}}^{\text{extra}} = 2\mu \mathbf{D}_{\text{por}} \cdot \mathbf{D}_{\text{por}} \quad \text{and} \quad \mathbf{i}_{\text{por}} = \mu \mathbf{K}^{-1} \mathbf{v}_{\text{por}}(\mathbf{x}). \quad (8.51)$$

By choosing the potential Φ_{por} to be

$$2\Phi_{\text{por}}[\mathbf{v}_{\text{por}}] = \underbrace{2\mu \mathbf{D}_{\text{por}} \cdot \mathbf{D}_{\text{por}} + \mu \mathbf{K}^{-1} \mathbf{v}_{\text{por}}(\mathbf{x}) \cdot \mathbf{v}_{\text{por}}(\mathbf{x})}_{\text{rate of internal dissipation density}}, \quad (8.52)$$

one can satisfy the requirements (8.46b) and (8.46c).

If the coupled flow is modeled based on Stokes-Darcy equations (i.e., Stokes model is used for the free flow region, and Darcy model is used for the porous region), then the two potentials for the coupled flow can be chosen based on equations (8.48) and (8.50), which are for uncoupled flows. Similarly, if the coupled flow is based on Stokes-Darcy-Brinkman equations (i.e., Stokes model is used for the free flow region and Darcy-Brinkman model is used for the porous region), then the two potentials for the coupled flow can be chosen based on equations (8.48) and (8.52).

8.5.2 Minimum power theorem

We define the total mechanical power functional as

$$\begin{aligned} \mathcal{P}_{\text{coupled}}[\mathbf{z}_{\text{free}}(\mathbf{x}), \mathbf{z}_{\text{por}}(\mathbf{x})] := & \int_{\mathcal{K}_{\text{free}}} \Phi_{\text{free}}[\mathbf{z}_{\text{free}}(\mathbf{x})] \, d\Omega + \int_{\mathcal{K}_{\text{por}}} \Phi_{\text{por}}[\mathbf{z}_{\text{por}}(\mathbf{x})] \, d\Omega \\ & + \int_{\Gamma_{\text{int}}} \Psi[\mathbf{z}_{\text{free}}^*(\mathbf{x}), \mathbf{z}_{\text{por}}^*(\mathbf{x}), z_n(\mathbf{x})] \, d\Gamma \end{aligned}$$

$$\begin{aligned}
& - \int_{\mathcal{K}_{\text{free}}} \gamma \mathbf{b}_{\text{free}}(\mathbf{x}) \cdot \mathbf{z}_{\text{free}}(\mathbf{x}) \, d\Omega - \int_{\Gamma_{\text{free}}^t} \mathbf{t}_{\text{free}}^p(\mathbf{x}) \cdot \mathbf{z}_{\text{free}}(\mathbf{x}) \, d\Gamma \\
& - \int_{\mathcal{K}_{\text{por}}} \gamma \phi_{\text{por}}(\mathbf{x}) \mathbf{b}_{\text{por}}(\mathbf{x}) \cdot \mathbf{z}_{\text{por}}(\mathbf{x}) \, d\Omega - \int_{\Gamma_{\text{por}}^t} \mathbf{t}_{\text{por}}^p(\mathbf{x}) \cdot \mathbf{z}_{\text{por}}(\mathbf{x}) \, d\Gamma,
\end{aligned} \tag{8.53}$$

where $\mathbf{z}_{\text{free}} : \mathcal{K}_{\text{free}} \rightarrow \mathbb{R}^{nd}$ and $\mathbf{z}_{\text{por}} : \mathcal{K}_{\text{por}} \rightarrow \mathbb{R}^{nd}$ are vector fields; $\mathbf{z}_{\text{free}}^*$ and $\mathbf{z}_{\text{por}}^*$ denote, respectively the tangential components of \mathbf{z}_{free} and \mathbf{z}_{por} ; and

$$z_n(\mathbf{x}) := \mathbf{z}_{\text{free}}(\mathbf{x}) \cdot \hat{\mathbf{n}}_{\text{free}}(\mathbf{x}).$$

We then establish the following result with a proof provided in Appendix D.

Theorem 8.5.1 (Minimum power theorem for coupled flows). *For the class of coupled flows satisfying the requirements (R1)–(R2), any pair of kinematically admissible vector fields $(\tilde{\mathbf{v}}_{\text{free}}(\mathbf{x}), \tilde{\mathbf{v}}_{\text{por}}(\mathbf{x}))$ satisfies*

$$\mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}(\mathbf{x}), \mathbf{v}_{\text{por}}(\mathbf{x})] \leq \mathcal{P}_{\text{coupled}}[\tilde{\mathbf{v}}_{\text{free}}(\mathbf{x}), \tilde{\mathbf{v}}_{\text{por}}(\mathbf{x})], \tag{8.54}$$

in which $\mathbf{v}_{\text{free}}(\mathbf{x})$ is the velocity field in the free flow region and $\mathbf{v}_{\text{por}}(\mathbf{x})$ is the velocity field in the porous region.

8.6 Uniqueness of solutions

We will use the minimum power theorem to establish the uniqueness of solutions under the proposed interface conditions. For brevity, we will show for the case of coupled Stokes-Darcy-Brinkman equations; however, with straightforward alterations, one can show for the case of Darcy equations coupled with the Stokes equations. We establish the uniqueness under the following functional form for Ψ , which is (slightly) more general than the one considered in §8.4:

$$\Psi[\mathbf{v}_{\text{free}}^*(\mathbf{x}), \mathbf{v}_{\text{por}}^*(\mathbf{x}), v_n(\mathbf{x})] = \alpha_{11} \mathbf{v}_{\text{free}}^*(\mathbf{x}) \cdot \mathbf{v}_{\text{free}}^*(\mathbf{x}) + 2\alpha_{12} \mathbf{v}_{\text{free}}^*(\mathbf{x}) \cdot \mathbf{v}_{\text{por}}^*(\mathbf{x})$$

$$+ \alpha_{22} \mathbf{v}_{\text{por}}^*(\mathbf{x}) \cdot \mathbf{v}_{\text{por}}^*(\mathbf{x}) + \beta v_n(\mathbf{x}) \cdot v_n(\mathbf{x}),$$
(8.55)

with

$$\alpha_{11}\alpha_{22} \geq \alpha_{12}^2 \quad \text{and} \quad \beta \geq 0.$$
(8.56)

To establish uniqueness under more general conditions (e.g., a more general functional form for Ψ), one needs to resort to techniques from functional analysis, which is beyond the scope of this paper. We establish the following theorem with a proof provided in Appendix D.

Theorem 8.6.1 (Uniqueness). *Under the prescribed data given by $\mathbf{b}_{\text{free}}(\mathbf{x})$, $\mathbf{b}_{\text{por}}(\mathbf{x})$, $\mathbf{v}_{\text{free}}^p(\mathbf{x})$, $\mathbf{v}_{\text{por}}^p(\mathbf{x})$, $\mathbf{t}_{\text{free}}^p(\mathbf{x})$ and $\mathbf{t}_{\text{por}}^p(\mathbf{x})$; and under Ψ given by equation (8.55); the solution to the coupled Stokes-Darcy-Brinkman equations is unique up to an arbitrary constant for the pressures.*

Chapter 9

A Modeling Framework for Coupling Plasticity With Species Diffusion

9.1 Mathematical model

We shall consider a chemical species that diffuses in a deformable solid. The deformation of the solid is modeled using small-strain elastoplasticity and the transport of the chemical species is treated as a Fickian-type diffusion process. We now present a mathematical model by coupling the deformation process of the solid with the transport process of the chemical species. We consider two types of coupling: one-way and two-way. Under the one-way coupling, the presence and transport of the chemical species affect the material parameters of the deformation process, but the deformation of the solid does not affect the transport process. To put it another way, neither the diffusion parameters, such as diffusivity, depend on the deformation or stress in the solid, nor the kinematics of the deformation enter the governing equations for the transport process. Under the two-way coupling, each of the two processes—deformation and transport—affect each other. We first introduce the necessary notation. We then outline the governing equations for each of the processes and describe the nature of the coupling between them.

Let $\Omega \subset \mathbb{R}^{nd}$ be an open bounded domain, where “ nd ” is the number of spatial dimensions. Let $\partial\Omega$ denote its smooth boundary. A spatial point is denoted by $\mathbf{x} \in \overline{\Omega}$,

where a superposed bar denote the set closure. The gradient and divergence operators with respect to \mathbf{x} are denoted by $\text{grad}[\cdot]$ and $\text{div}[\cdot]$, respectively. The unit outward normal to the boundary is denoted by $\hat{\mathbf{n}}(\mathbf{x})$. We denote the displacement of the solid by $\mathbf{u}(\mathbf{x})$ and concentration field by $c(\mathbf{x})$. For the deformation subproblem, the boundary is divided into two complementary parts: Γ_u^D and Γ_u^N . Γ_u^D denotes that part of the boundary on which Dirichlet (displacement) boundary condition is prescribed, and Γ_u^N it the part of the boundary on which Neumann (traction) boundary condition is prescribed. Likewise, for the diffusion subproblem, the boundary is divided into Γ_c^D —part of the boundary on which Dirichlet (concentration) boundary condition is prescribed—and Γ_c^N , part of the boundary on which Neumann (flux) boundary condition is prescribed. For mathematical well-posedness, we assume that $\Gamma_u^D \cap \Gamma_u^N = \emptyset$, $\Gamma_u^D \cup \Gamma_u^N = \partial\Omega$, $\Gamma_c^D \cap \Gamma_c^N = \emptyset$, and $\Gamma_c^D \cup \Gamma_c^N = \partial\Omega$. Moreover, for uniqueness, we assume that Γ_u^D and Γ_c^D have zero (set) measure.

9.1.1 Deformation subproblem

We account for the solid undergoing elasto-plastic deformations and the material is degrading due to the presence of the chemical species. We make the following assumptions for the elasto-plastic deformations: (i) the strains are small, (ii) kinematic hardening is neglected, (iii) the plasticity is associative, and (iv) J_2 flow theory is applicable. We will consider two different degradation models, referred to as Model I and Model II. We make the following assumptions for Model I: (a) the elastic material parameters at a spatial point depend on the concentration of the chemical species at that point, and (b) the material can undergo linear isotropic hardening. The assumptions behind Model II are: (a) the elastic limit function depends on the concentration of the chemical species, and (b) the material can undergo nonlinear isotropic hardening.

Since we consider plasticity based on small strains, linearized strain and additive

decomposition of strain will suffice. We denote the linearized strain by¹

$$\mathbf{E} := \frac{1}{2}(\text{grad}[\mathbf{u}] + \text{grad}[\mathbf{u}]^T). \quad (9.1)$$

The additive decomposition of the strain tensor takes the following form

$$\mathbf{E} = \mathbf{E}^e + \mathbf{E}^p, \quad (9.2)$$

where \mathbf{E}^e and \mathbf{E}^p denote the elastic and plastic components, respectively.

The governing equations of the initial boundary-value deformation problem under quasi-static condition could be written as:

$$\text{div}[\mathbf{T}] + \rho(\mathbf{x})\mathbf{b}(\mathbf{x}) = \mathbf{0} \quad \text{in } \Omega, \quad (9.3a)$$

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}^p(\mathbf{x}, t) \quad \text{on } \Gamma_u^D \times (0, \mathcal{T}], \text{ and} \quad (9.3b)$$

$$\mathbf{T}\hat{\mathbf{n}} = \mathbf{t}^p(\mathbf{x}, t) \quad \text{on } \Gamma_u^N \times (0, \mathcal{T}], \quad (9.3c)$$

where ρ denotes the density, \mathbf{b} denotes the specific body force, \mathbf{u}^p is the prescribed time-varying displacement, and \mathbf{t}^p is the prescribed time-varying surface traction. In addition, we consider the following initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega, \quad (9.4)$$

where \mathbf{u}_0 is an initial displacement response that satisfies equations (9.3a)–(9.3c) at $t = 0$. To close the formulation of this initial boundary-value problem, we require that Cauchy stress satisfy the rate constitution equation as

$$\dot{\mathbf{T}} = \mathbb{E}\dot{\mathbf{E}}, \quad (9.5)$$

¹In continuum mechanics, \mathbf{E} is typically reserved to denote the Lagrangian strain. Since we do not consider large-deformations in this text, there should be no confusion in our usage of \mathbf{E} to denote the linearized strain.

where \mathbb{E} is a fourth order tangent constitutive tensor and It should be noted that the rate notations in the constitutive equation are pseudo and denote an incremental constitutive response, and should not be interpreted to mean that the response may contain rate-dependent effects.

A common choice for the free energy function Ψ_{nrg} is given by

$$\Psi_{\text{nrg}}(\mathbf{E}^e, \mathcal{E}) = \mathcal{W}(\mathbf{E}^e) + \mathcal{H}(\mathcal{E}), \quad (9.6)$$

where \mathcal{W} is the strain energy density, \mathcal{H} is the hardening potential, and \mathcal{E} represents a general set of internal variables modeling the hardening of the material. In the context of plasticity the set of \mathcal{E} is defined as

$$\mathcal{E} := \{\kappa, \boldsymbol{\alpha}\}, \quad (9.7)$$

where κ is an internal variable measures the accumulated equivalent plastic strain, $\boldsymbol{\alpha}$ is the back stress that is determined by a kinematic hardening model. The energy form explained in equation (9.6) enables us to decouple the elastic and hardening response of the material. A classical quadric form of \mathcal{W} could be considered as

$$\mathcal{W}(\mathbf{E}^e) = \frac{1}{2} \mathbf{E}^e \cdot \mathbb{C} \mathbf{E}^e, \quad (9.8)$$

where \mathbb{C} is a fourth-order elasticity tensor. The Cauchy stress can be obtained as

$$\mathbf{T} = \frac{\partial \mathcal{W}}{\partial \mathbf{E}^e} = \mathbb{C} \mathbf{E}^e = \mathbb{C} (\mathbf{E} - \mathbf{E}^p). \quad (9.9)$$

The deviatoric part of the stress tensor, \mathbf{S} , is defined as

$$\mathbf{S} := \mathbf{T} - \frac{1}{nd} \text{tr}[\mathbf{T}] \mathbf{I},$$

where $\text{tr}[\cdot]$ denotes the trace of a second-order tensor, and \mathbf{I} denotes the second-order identity tensor. By differentiating \mathcal{H} with respect to components of \mathcal{E} , we also introduce the set of stress-like hardening variables

$$\mathcal{Q} := \left\{ -\frac{\partial \mathcal{H}}{\partial \kappa}, -\frac{\partial \mathcal{H}}{\partial \boldsymbol{\alpha}} \right\}. \quad (9.10)$$

The stress tensor in classical plasticity model must satisfy the yield criterion, which determines whether the material is still elastic or it has undergone an irreversible plastic deformation. This criterion, which holds at any material point and at any loading instance, is defined as

$$f(\boldsymbol{\xi}, \kappa, c) = \phi(\boldsymbol{\xi}, I_{\text{kin}}(\kappa, c)) - \sigma_y^*(\kappa, c, \sigma_0) \leq 0, \quad (9.11)$$

where ϕ is a scalar effective stress measure, σ_y^* is elastic limit function, $\boldsymbol{\xi} = \mathbf{S} - \boldsymbol{\alpha}$ is the shifted stress, I_{kin} is the function used to model kinematic hardening, and σ_0 is the initial scalar yield stress in the absent of diffusant. Our material is represented by von Mises yield condition (also known as J_2 -flow) and equation (9.11) could be reduced to

$$f(\boldsymbol{\xi}, \kappa, c) = \|\boldsymbol{\xi}(I_{\text{kin}}(\kappa, c))\| - \sigma_y^*(\kappa, c, \sigma_0) \leq 0. \quad (9.12)$$

The evolution of plastic strain could be determined as

$$\dot{\mathbf{E}}^p = \dot{\gamma} \frac{\partial g(\mathbf{T}, \mathcal{E})}{\partial \mathbf{T}} = \dot{\gamma} \hat{\mathbf{N}}, \quad (9.13)$$

where $\dot{\gamma}$ is the rate of the plastic multiplier that is nonnegative, the scalar function g is the plastic potential, and $\hat{\mathbf{N}}$ is a unit deviatoric tensor that is normal to the yield surface. In this study, we assume associative plastic flow and hence $f = g$. The term $\dot{\gamma}$ determines the magnitude of the plastic strain rate, and the direction is given by $\hat{\mathbf{N}}$. As the material undergoes plastic deformation, the plastic variables also change according to the hardening model. A general form of hardening rule can be stated as

$$\dot{\mathcal{E}} = \dot{\gamma} \mathbf{h}(\boldsymbol{\xi}, \mathcal{Q}, \kappa, c) = \dot{\gamma} \frac{\partial f(\boldsymbol{\xi}, \kappa, c)}{\partial \mathcal{Q}}. \quad (9.14)$$

In particular, the rate of back stress, and the rate of effective plastic strain could be obtained from equation (9.14) as:

$$\dot{\boldsymbol{\alpha}} = I_{\text{kin}}(\kappa, c) \dot{\gamma} \frac{\partial f(\boldsymbol{\xi}, \kappa, c)}{\partial \boldsymbol{\xi}} = I_{\text{kin}}(\kappa, c) \dot{\gamma} \hat{\mathbf{N}} \text{ and} \quad (9.15a)$$

$$\dot{\kappa} = \sqrt{\frac{2}{3}} \|\dot{\mathbf{E}}^p\| = \sqrt{\frac{2}{3}} \dot{\gamma}. \quad (9.15b)$$

Finally, the loading/unloading conditions can be expressed in the Kuhn-Tucker form as:

$$\dot{\gamma} \geq 0, \quad f \leq 0, \text{ and } \dot{\gamma}f = 0. \quad (9.16)$$

Now we introduce two models for taking into account the coupling effect of diffusion of species on deformation problem.

Remark 9.1.1. *In the degradation models introduced in the rest of this section, degraded material incorporates only isotropic hardening, and no kinematic hardening behavior will be assumed (i.e., $I_{\text{kin}} = 0$). This hypothesis could be justified as the material is assumed to undergo a monolithic loading regime, and hence, the Bauschinger effect could be neglected. However, in case of the emergence of supportive experimental results that observe kinematic hardening phenomenon for the coupled deformation-diffusion system, the proposed framework in here could be extended with no difficulties.*

9.1.1.1 Model I: degradation via elasticity parameters

In this model, linear isotropic hardening model is employed and the Cauchy stress tensor dependence on c , is directly from Lamé parameters. Hence, the yield condition could be simplified as

$$f(\mathbf{T}, \kappa, c) = \sqrt{\frac{3}{2}} \|\mathbf{S}\| - \sigma_y^* = \sqrt{\frac{3}{2}} \|\mathbf{S}\| - H\kappa - \sigma_0 \leq 0 \quad (9.17)$$

where the constant scalar $H > 0$ is the isotropic hardening modulus, and the stress-strain relationship, for any given concentration, could be written as

$$\mathbf{T}(\mathbf{u}, \mathbf{x}, c) = \lambda(\mathbf{x}, c) \text{tr}[\mathbf{E}^e] \mathbf{I} + 2\mu(\mathbf{x}, c) \mathbf{E}^e, \quad (9.18)$$

where λ , and μ are the Lamé parameters. To consider degradation/healing of the material due to the presence of diffusive species, the Lamé parameters depend on concentration and can be written as:

$$\lambda(\mathbf{x}, c) = \lambda_0(\mathbf{x}) + \lambda_1(\mathbf{x}) \frac{c(\mathbf{x})}{c_{\text{ref}}} \quad \text{and} \quad (9.19\text{a})$$

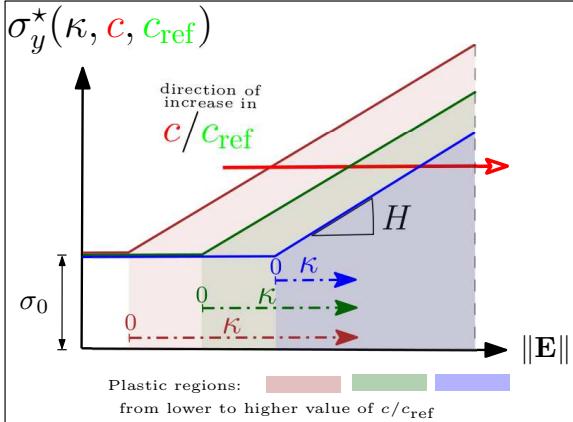
$$\mu(\mathbf{x}, c) = \mu_0(\mathbf{x}) + \mu_1(\mathbf{x}) \frac{c(\mathbf{x})}{c_{\text{ref}}}, \quad (9.19\text{b})$$

where c_{ref} is the reference concentration, λ_0 and μ_0 are the Lamé parameters for the virgin material; and λ_1 and μ_1 incorporate the effect of concentration on the Lamé parameters. This type of coupling for the problem of pure elasticity-diffusion was earlier implemented by [60, 133]. **Fig 9.1(a)** shows the effect of coupling power $\frac{c}{c_{\text{ref}}}$ on the elastic limit function. It could be seen that the elastic limit function has shifted to the right as coupling power increases, which implies a delay in plastic yielding. However, the initial yield stress remains independent of coupling power. **Fig 9.1(b)** shows the stress path for a representative one-dimensional problem under uniaxial tension loading when degradation model I is employed. In one-dimensional analysis, stress, and strain are respectively denoted by σ , and ε . Tangent modulus tensor \mathbb{E} reduces E_t , which could be related to isotropic hardening modulus as

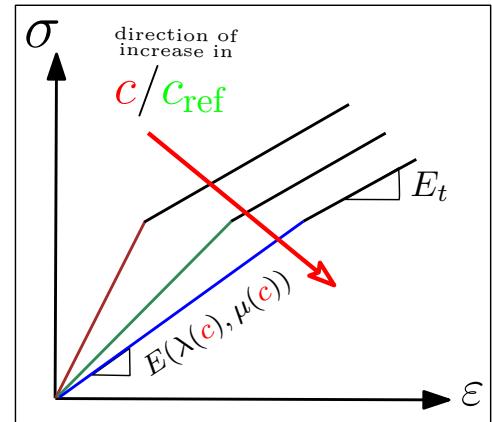
$$E_t = \frac{H}{1 + \frac{H}{E}}. \quad (9.20)$$

9.1.1.2 Model II: degradation via elastic limit function

Nonlinear isotropic hardening model, which was initially proposed by [169], is adopted for this part of study. In recent years, variants of nonlinear isotropic models have been employed for modeling healing/degradation problems. Using this hardening model, [94, 105, 129, 101, 162] have captured hydrogen embrittlement of metals, where the materials



(a) Elastic limit function



(b) One-dimensional stress-strain

Figure 9.1: Model I: left figure shows the effect of coupling parameter c_{ref} on the onset of plastic yielding. Right figure shows one-dimensional uniaxial stress-strain relationship ($\sigma-\epsilon$) undertaking “degradation via elasticity parameters” model.

become softened at microscale due to enhancing the dislocation mobility. In this coupling category, the yield function takes the following form

$$f(\mathbf{T}, \kappa, c) = \sqrt{\frac{3}{2}} \|\mathbf{S}\| - \sigma_y^* = \sqrt{\frac{3}{2}} \|\mathbf{S}\| - \sigma_c \left(1 + \frac{\kappa}{\kappa_0}\right)^{n_w}, \quad (9.21)$$

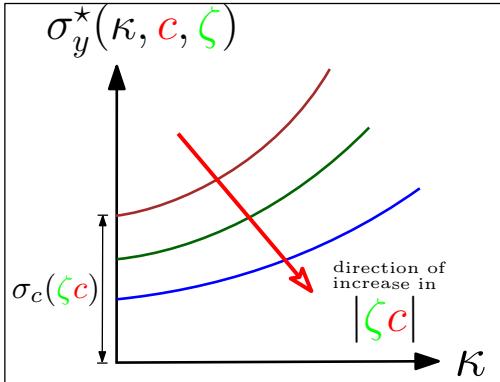
where $\kappa_0 = \frac{\sigma_0}{E}$, with E being the Young's modulus, n_w is the work hardening exponent, and σ_c is the initial yield stress in the presence of diffusant. A linear form for σ_c is chosen as

$$\sigma_c = (\zeta c + 1)\sigma_0. \quad (9.22)$$

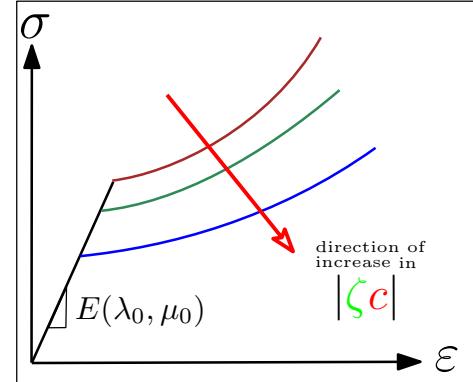
ζ is the coupling parameter used to adjust the behavior of elastic limit function based on diffusant absorption in solid structure. The stress-strain relationship in this model takes the following form

$$\mathbf{T}(\mathbf{u}, \mathbf{x}) = \lambda_0(\mathbf{x}) \text{tr}[\mathbf{E}^e] \mathbf{I} + 2\mu_0(\mathbf{x}) \mathbf{E}^e. \quad (9.23)$$

Fig 9.2(a) signifies the effect of coupling power $|\zeta c|$ on the behavior of elastic limit function. Unlike model I, initial yield stress depends on the coupling power. Schematic



(a) Elastic limit function



(b) One-dimensional stress-strain

Figure 9.2: Model II: left figure depicts consistent decline in elastic limit function as the coupling power $|\zeta c|$ increases. Right figure shows one-dimensional stress-strain path under uniaxial tension ($\sigma-\epsilon$), employing the degradation via model II.

of stress-strain relationship for one-dimensional uniaxial loading is illustrated in Fig 9.2(b).

Remark 9.1.2. Throughout the chapter, we refer to the case of the “uncoupled problems”, which implies pure plastic deformation problem or deformation problem in the absence of diffusant in the domain. This could be equally achieved by assuming $c_{\text{ref}} \rightarrow \infty$ and $\zeta \rightarrow 0$ for model I and model II, respectively. Uncoupled model I is the conventional linear isotropic hardening and uncoupled model II, which is nonlinear isotropic hardening, is known as Swift model (see figure 9.3).

9.1.2 Transport subproblem

Mass transport of diffusant species (e.g., hydrogen or material void) in the material domain is a major factor in the degradation of the solid structure. Steady-state diffusion is taken for the transport phenomenon, which assumes no chemical reactions causing the formation of reaction products or phase transformations to occur at the diffusion problem.

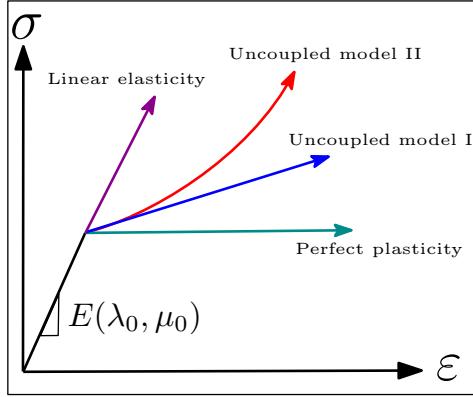


Figure 9.3: **Uncoupled models:** This schematic shows one-dimensional stress-strain relationship for uncoupled model I and model II; and compares them with standard models under one-dimensional uniaxial tension.

The governing equations for the steady-state response of diffusion the system takes the following form:

$$-\operatorname{div}[\mathbf{D}(\mathbf{x})\operatorname{grad}[c(\mathbf{x})]] = m(\mathbf{x}) \quad \text{in } \Omega, \quad (9.24a)$$

$$c(\mathbf{x}) = c^p(\mathbf{x}) \quad \text{on } \Gamma_c^D, \text{ and} \quad (9.24b)$$

$$-\hat{\mathbf{n}}(\mathbf{x}) \cdot \mathbf{D}(\mathbf{x})\operatorname{grad}[c(\mathbf{x})] = h^p(\mathbf{x}) \quad \text{on } \Gamma_c^N, \quad (9.24c)$$

where \mathbf{D} is the diffusivity tensor, $m(\mathbf{x})$ is the prescribed volumetric source, and $h^p(\mathbf{x})$ is the prescribed diffusive flux. Assumption of steady-state for the diffusion equation is realistic as the solid deformation is quasi-static (which implies that mechanical loading is to be applied very slowly) and hence at any loading instance, the diffusion process reaches equilibrium. We now discuss two types of coupling for the system of coupled deformation-diffusion.

9.1.2.1 One-way vs two-way coupling

In both one-way and two-way systems, the concentration of the diffusant mass (which is calculated by using the pure diffusion equation) is coupled into deformation system to determine the elastoplastic behavior. The difference arises in the choice of the diffusivity

tensor. As in one-way coupling strategy, the diffusivity tensor is irrespective of the strain accumulated in the domain from elastoplastic deformation. In other words:

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}_0 = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}, \quad (9.25)$$

where \mathbf{D}_0 is the reference diffusivity tensor and d_1 and d_2 are parameters specifying material anisotropy. On the other hand, in the two-way system, for a given total strain field with its first invariant defined as

$$I_E := \text{tr}[\mathbf{E}], \quad (9.26)$$

we take into account the effect of deformation on the diffusivity tensor as

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}_0(\mathbf{x}) + (\mathbf{D}_T(\mathbf{x}) - \mathbf{D}_0(\mathbf{x})) \left(\frac{\exp[\eta_T I_E] - 1}{\exp[\eta_T I_E^{\text{ref}}] - 1} \right) + (\mathbf{D}_S(\mathbf{x}) - \mathbf{D}_0(\mathbf{x})) \left(\frac{\exp[\eta_S I_E] - 1}{\exp[\eta_S I_E^{\text{ref}}] - 1} \right), \quad (9.27)$$

where the non-negative η_T and η_S are material parameters. \mathbf{D}_T and \mathbf{D}_S are respectively the reference diffusivity tensor under tensile and shear strains; and E_{ref} is a reference measure of the strain. For convenience \mathbf{D}_T and \mathbf{D}_S can be chosen as

$$\mathbf{D}_T = \phi_T \mathbf{D}_0 \text{ and} \quad (9.28a)$$

$$\mathbf{D}_S = \phi_S \mathbf{D}_0, \quad (9.28b)$$

where ϕ_T and ϕ_S are some positive real number material parameters specifying the corresponding anisotropy induced from the deformation problem. **Fig 9.4** summarizes one-way and two-way coupling strategies.

Remark 9.1.3. *Two-way coupling strategy could be reduced to one-way coupling if \mathbf{D}_T , and \mathbf{D}_S are chosen to be equal to \mathbf{D} . This will establish the independence of diffusivity tensor from deformation problem (i.e., $\mathbf{D}_E(\mathbf{x}) = \mathbf{D}_0$) at any solve step.*

Remark 9.1.4. The effect of anisotropy of diffusivity tensor on deformation responses will be studied in Section 3, and 4. It should be clarified that in this chapter for brevity, the term anisotropy refers to diffusivity anisotropy and should not be confused with anisotropy in deformation problem as the elasticity tensor remains isotropic in all the problems.

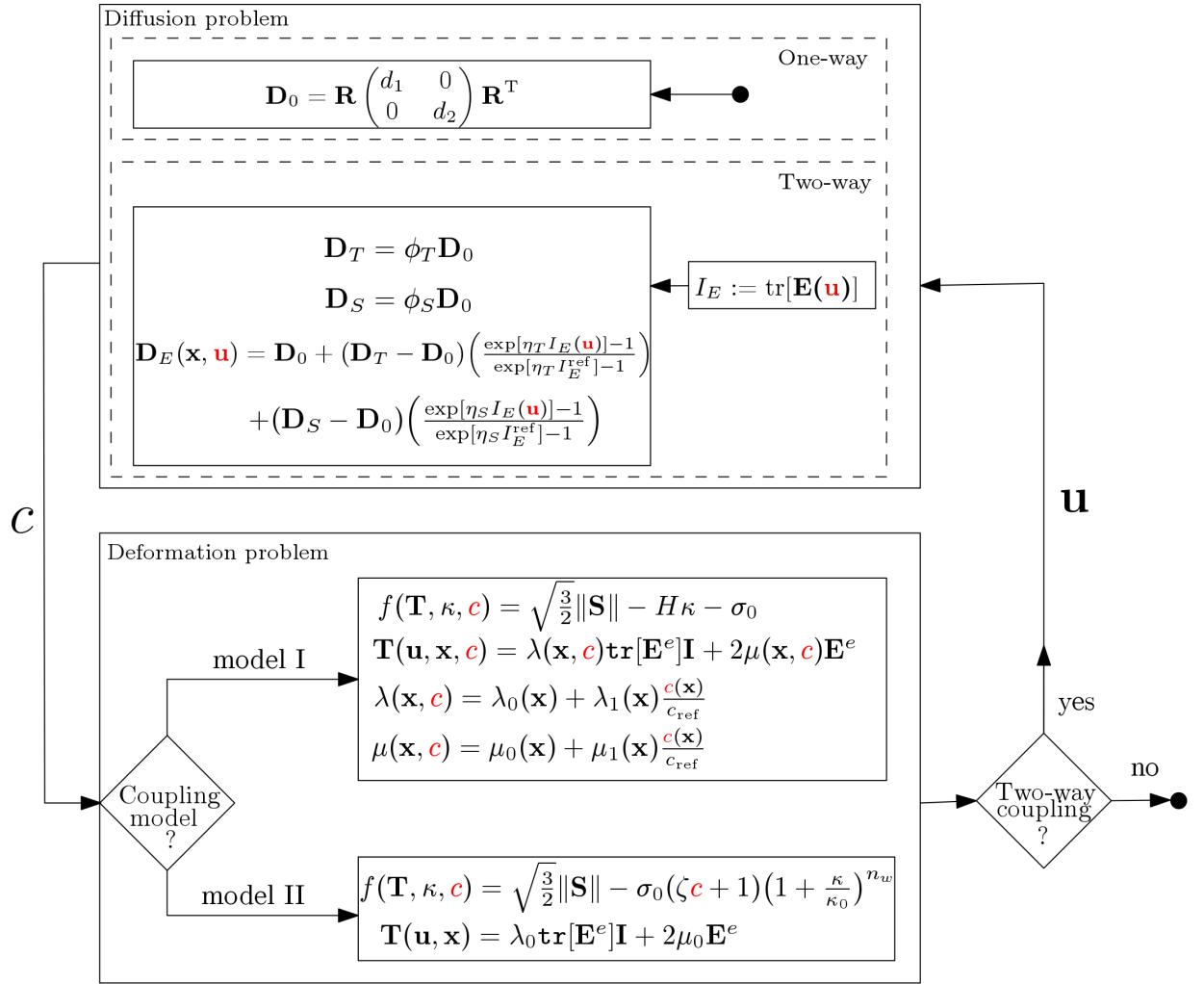


Figure 9.4: One-way and two-way coupling strategies for deformation-diffusion system.
In this figure \mathbf{R} denotes the rotation tensor.

9.2 Proposed computational framework

We shall develop a computational framework based on a staggered coupling approach, which allows decomposing the problem into two subproblems—deformation and diffusion. By solving these two subproblems iteratively until convergence, keeping the field variables from the other subproblem constant in each iteration of a subproblem, one can get the coupled response. Besides a coupling algorithm, the proposed computational framework comprises individual solvers for the two subproblems. We use low-order finite elements and the same computational mesh for solving both the subproblems. We describe below the mentioned ingredients of the proposed computational framework.

9.2.1 A solver for deformation subproblem

We build the solver for the deformation subproblem by combining the single-field displacement-based continuous Galerkin formulation, backward Euler, predictor-corrector return mapping algorithm, and Newton-Raphson method. We define the following function spaces:

$$\mathcal{U} := \left\{ \mathbf{u}(\mathbf{x}, \cdot) \in (H^1(\Omega))^{nd} \mid \mathbf{u}(\mathbf{x}, t) = \mathbf{u}^p(\mathbf{x}, t) \text{ on } \Gamma_u^D \right\} \text{ and} \quad (9.29a)$$

$$\mathcal{W} := \left\{ \mathbf{w}(\mathbf{x}, \cdot) \in (H^1(\Omega))^{nd} \mid \mathbf{u}(\mathbf{x}, t) = \mathbf{0} \text{ on } \Gamma_u^D \right\}, \quad (9.29b)$$

where $H^1(\Omega)$ is a standard Sobolov space on Ω [35]. The load step is divided into $\mathcal{T} + 1$ sub-intervals and for any quantity ψ , we have

$$\psi_n(\mathbf{x}) \approx \psi(\mathbf{x}, t_n), \quad n = 0, \dots, \mathcal{T}. \quad (9.30)$$

We assume the analysis procedure has been completed up to load increment t_n . The single-field Galerkin formulation for the pure deformation problem at load increment

t_{n+1} reads: Find $\mathbf{u}_{n+1} \in \mathcal{U}$ such that we have

$$\mathcal{F}(\mathbf{u}_{n+1}, \mathbf{w}) = 0 \quad \forall \mathbf{w} \in \mathcal{W}, \quad (9.31)$$

where \mathcal{F} is the residual.

$$\mathcal{F}(\mathbf{u}_{n+1}, \mathbf{w}) := \int_{\Omega} \underbrace{\mathbf{T}[\mathbf{E}(\mathbf{u}_{n+1})]}_{\mathbf{T}_{n+1}} \cdot \text{grad}[\mathbf{w}] \, d\Omega - \int_{\Omega} \rho \mathbf{b} \cdot \mathbf{w} \, d\Omega + \int_{\Gamma_{\mathbf{u}}^N} \mathbf{w} \cdot \mathbf{t}_n^p \, d\Gamma. \quad (9.32)$$

A solution to problems with nonlinear constitutive models, such as plasticity requires linearization. Assuming that the applied load is independent of displacement, only the first term of equation (9.32) requires linearization through Newton's method. Let the superscript (i) denote the current Newton or nonlinear iteration. The Jacobian $\mathcal{J}[\mathbf{u}_{n+1}^{(i)}; \delta \mathbf{u}, \mathbf{w}]$ is computed by taking the Gâteaux variation of the residual $\mathcal{F}(\mathbf{u}_{n+1}, \mathbf{w})$ at $\mathbf{u}_{n+1} = \mathbf{u}_{n+1}^{(i)}$ in the directions of $\delta \mathbf{u}$. Formally, this is derived by

$$\mathcal{J}[\mathbf{u}_{n+1}^{(i)}; \delta \mathbf{u}, \mathbf{w}] := \lim_{\epsilon \rightarrow 0} \frac{\mathcal{F}(\mathbf{u}_{n+1}^{(i)} + \epsilon \delta \mathbf{u}; \mathbf{w}) - \mathcal{F}(\mathbf{u}_{n+1}^{(i)}; \mathbf{w})}{\epsilon} \equiv \left[\frac{d}{d\epsilon} \mathcal{F}(\mathbf{u}_{n+1}^{(i)} + \epsilon \delta \mathbf{u}; \mathbf{w}) \right]_{\epsilon=0}, \quad (9.33)$$

provided the limit exists. Further discussions on Gâteaux variations are provide by [75].

Following through with the calculation above, the Jacobian for our formulation reads

$$\mathcal{J}[\mathbf{u}_{n+1}^{(i)}; \delta \mathbf{u}, \mathbf{w}] := \int_{\Omega} \mathbb{C}_{\text{alg}}^{(n+1,i)} \frac{\partial \mathbf{E}(\mathbf{u}_{n+1}^{(i)})}{\partial \mathbf{u}_{n+1}^{(i)}} \delta \mathbf{u} \cdot \text{grad}[\mathbf{w}] \, d\Omega, \quad (9.34)$$

where $\mathbb{C}_{\text{alg}}^{(n+1,i)} = \frac{\partial \mathbf{T}_{n+1}^{(i)}}{\partial \mathbf{E}(\mathbf{u}_{n+1}^{(i)})}$ denotes algorithmic tangent modulus.

In each Newton iteration, we thus solve the following linear variational problem: Find $\delta \mathbf{u} \in \mathcal{U}$ such that we have

$$\mathcal{J}[\mathbf{u}_{n+1}^{(i)}; \delta \mathbf{u}, \mathbf{w}] = -\mathcal{F}(\mathbf{u}_{n+1}^{(i)}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathcal{W}. \quad (9.35)$$

The fully discrete formulations for our deformation model at each Newton's iteration can be assembled into the following linear problem

$$\mathbf{K}_{\mathbf{u}}^{(n+1,i)} \delta \mathbf{u} = \mathbf{r}_{\mathbf{u}}^{(n+1,i)}. \quad (9.36)$$

where \mathbf{K}_u is called the tangent stiffness matrix and \mathbf{r}_u is the residual vector. Two different definitions of displacement increments could be considered for the incremental deformation problem as:

$$\delta\mathbf{u} = \mathbf{u}_{n+1}^{(i+1)} - \mathbf{u}_{n+1}^{(i)} \text{ and} \quad (9.37a)$$

$$\Delta\mathbf{u}^{(i)} = \mathbf{u}_{n+1}^{(i)} - \mathbf{u}_n, \quad (9.37b)$$

where $\delta\mathbf{u}$ is the displacement increment calculated at each Newton's iteration, while $\Delta\mathbf{u}$, which is the increment from the last converged load increment to the previous iteration, will be used to calculate stress increment. In other words, $\delta\mathbf{u}$ is accumulated into $\Delta\mathbf{u}$ during the iterations. $\Delta\mathbf{u}$ is set to 0 before starting a new load increment. After obtaining the nodal displacement increments by solving equation (9.36), the displacement increment $\Delta\mathbf{u}$ is achieved by the following update equation

$$\Delta\mathbf{u}^{(i+1)} = \Delta\mathbf{u}^{(i)} + \delta\mathbf{u}^{(i)}. \quad (9.38)$$

Once the residual meets the prescribed tolerance the process will be terminated. While calculating residual, the stress $\mathbf{T}_{n+1}^{(i)}$ needs to be obtained. Stress calculation is complicated due to history dependency of stress and non-linearity with respect to strain as the plastic deformation occurs. Before elaborating on the stress determination strategy, we need to formulate numerical algorithms to integrate the rate-form constitutive relations in the deformation problem. We resort to the backward Euler marching scheme to ensure numerical stability. It is well-known that the backward Euler method as discussed by [7] leads to the closest point projection in the elastoplasticity problem. Substituting equation (9.13) into equation (9.9), incremental stress could be written as

$$\mathbf{T}_{n+1} = \underbrace{\mathbf{T}_n + \mathbb{C}\Delta\mathbf{E}_{n+1}}_{\mathbf{T}_{n+1}^{\text{trial}}} - \mathbb{C}\Delta\mathbf{E}_{n+1}^p = \mathbf{T}_{n+1}^{\text{trial}} - 2\mu\Delta\gamma\hat{\mathbf{N}}_{n+1}. \quad (9.39)$$

Internal variables at $t = t_{n+1}$ are also updated as

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + I_{\text{kin}}\Delta\gamma\hat{\mathbf{N}}_{n+1} \text{ and} \quad (9.40a)$$

$$\kappa_{n+1} = \kappa_n + \sqrt{\frac{2}{3}} \Delta\gamma. \quad (9.40b)$$

Coaxiality of \mathbf{S}_{n+1} and $\mathbf{S}_{n+1}^{\text{trial}}$ tensors could be easily established, which implies $\widehat{\mathbf{N}}_{n+1} = \widehat{\mathbf{N}}_{n+1}^{\text{trial}}$. As a result, shifted stress takes the following form

$$\boldsymbol{\xi}_{n+1} = \mathbf{S}_{n+1} - \boldsymbol{\alpha}_{n+1} = \mathbf{T}_{n+1}^{\text{trial}} - \alpha_n - (2\mu + I_{\text{kin}})\Delta\gamma \widehat{\mathbf{N}}_{n+1}^{\text{trial}}. \quad (9.41)$$

Incremental form of equation (9.16) implies that under plastic yielding ($\Delta\gamma \neq 0$), stress must stay on the yielding surface (i.e., $f = 0$). This condition is known as *plastic consistency condition* and using equations (9.40a) and (9.41), it takes the following general form

$$f(\boldsymbol{\xi}_{n+1}, \kappa_{n+1}, c) = \|\boldsymbol{\xi}_{n+1}\| - \sigma_y^*(\kappa_{n+1}) = \|\boldsymbol{\xi}_{n+1}^{\text{trial}}\| - \{2\mu + I_{\text{kin}}(\kappa_{n+1})\Delta\gamma\} - \sqrt{\frac{2}{3}}\sigma_y^*(\kappa_{n+1}) = 0. \quad (9.42)$$

Updated stress and updated internal variables for an applied incremental strain at a given material point will be obtained via a separate algorithm outside of the main form. The response is computed using an iterative predictor-corrector return mapping algorithm embedded in the global Newton iteration discussed earlier. This procedure for both degradation models is summarized in Algorithm 1.

Remark 9.2.1. *In this chapter, function spaces for deformation problem will be a standard linear CG space for the displacement while the stress and internal variables will be represented by using a linear quadrature element. If all functions are assumed to be a finite element space, or are interpolated in a finite element space, suboptimal convergence of a Newton method will be observed. This is a well-known point in computational plasticity and has been extensively discussed in [65, 164]. The choice of quadrature element will make it possible to express the complex non-linear material constitutive equation at the Gauss (quadrature) point only, without involving any interpolation of non-linear expressions throughout the element. It will ensure an optimal convergence rate for the Newton's method. For a thorough discussion of the quadrature element refer to [118].*

Algorithm 1 Stress update algorithm for degradation model I and model II

Input: \mathbf{T}_n , κ_n , and $\Delta\mathbf{E}^{(i+1)}$.

▷ $\Delta\mathbf{E}^{(i+1)} = \mathbf{E}_{n+1}^{(i+1)} - \mathbf{E}_n$ (from the last load increment to the current iteration)

Output: \mathbf{T}_{n+1} , κ_{n+1} .

1. Compute the elastic trial state

$$\mathbf{S}_{n+1}^{\text{trial}} = \mathbf{S}_n + \mathbb{C}\Delta\mathbf{E}_{\text{dev}}^{(i)} = \mathbf{S}_n + 2\mu\Delta\mathbf{E}_{\text{dev}}^{(i)}.$$

▷ Note that in **model I**: $\mu = \hat{\mu}(c)$.

2. Compute $f_{n+1}^{\text{trial}} = f(\mathbf{T}_{n+1}^{\text{trial}}, \kappa_n)$ and check consistency of trial state

$$\boldsymbol{\xi}_{n+1}^{\text{trial}} = \mathbf{S}_{n+1}^{\text{trial}} - \boldsymbol{\alpha}_{n+1}^{\text{trial}},$$

$$f_{n+1}^{\text{trial}} = \|\boldsymbol{\xi}_{n+1}^{\text{trial}}\| - \sqrt{\frac{2}{3}}\sigma_y^*(\kappa_n),$$

model I: $f_{n+1}^{\text{trial}} = \|\mathbf{S}_{n+1}^{\text{trial}}\| - \sqrt{\frac{2}{3}}(\sigma_0 - H\kappa_n)$ or

model II: $f_{n+1}^{\text{trial}} = \|\mathbf{S}_{n+1}^{\text{trial}}\| - \sqrt{\frac{2}{3}}(\zeta c + 1)\sigma_0(1 + \frac{\kappa}{\kappa_0})^{n_w},$

```

if  $f_{n+1}^{\text{trial}} \leq 0$  then
     $(\cdot)_{n+1} = (\cdot)_{n+1}^{\text{trial}}$  and EXIT (elastic step).
else
    solve for  $\Delta\gamma > 0$  in step 3 (plastic step).
end if

```

Algorithm 1 continued

3. Plastic step or return mapping algorithm: solve for $\Delta\gamma$ (refer to equation (9.42))

model I: $f(\mathbf{S}_{n+1}, \kappa_{n+1})$ is linear w.r.t $\Delta\gamma$

$$\Delta\gamma = \frac{f_{n+1}^{\text{trial}}}{2\mu + \frac{2}{3}H} \text{ or}$$

model II: $f(\mathbf{S}_{n+1}, \kappa_{n+1})$ is non-linear w.r.t $\Delta\gamma \rightarrow$ Local Newton's method Initialize: $k = 0$, κ^k , $\Delta\gamma^k = 0$, f_{Tol} , and k_{max} .

while $\tilde{\mathcal{F}} > f_{Tol}$ AND $k < k_{max}$ **do**

$$\tilde{\mathcal{J}}[\Delta\gamma^k; \delta\Delta\gamma] = -\tilde{\mathcal{F}}(\Delta\gamma^k),$$

where

$$\begin{aligned} \tilde{\mathcal{J}} &= -2\mu\delta\Delta\gamma - \sqrt{\frac{2}{3}} \frac{\partial\sigma_y^*}{\partial\kappa_{n+1}} \frac{\partial\kappa_{n+1}}{\partial\Delta\gamma} \delta\Delta\gamma = \left\{ -2\mu - \frac{2}{3} \frac{n_w\sigma_0}{\kappa_0} (\zeta c + \right. \\ &\left. 1) \left(1 + \frac{\Delta\gamma^k}{\kappa_0} \right)^{n-1} \right\} \delta\Delta\gamma \end{aligned}$$

and

$$\Delta\gamma^{k+1} = \Delta\gamma^k + \delta\Delta\gamma$$

end while

4. Update stress and plastic variables

$$\mathbf{T}_{n+1} = \mathbf{T}_{n+1}^{\text{trial}} - 2\mu\Delta\gamma\hat{\mathbf{N}}_{n+1}^{\text{trial}}; \quad \boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n - H\Delta\gamma\hat{\mathbf{N}}_{n+1}^{\text{trial}}; \quad \text{and} \quad \kappa_n^{k+1} = \kappa_n + \sqrt{\frac{2}{3}}\Delta\gamma^{k+1}.$$

Remark 9.2.2. *The algorithmic tangent modulus is needed for the calculation of global Jacobian introduced in equation (9.34). This modulus should be consistent with time integration, and stress update algorithm discussed earlier. By differentiation of incremental stress (refer to equation (9.39)) with respect to the incremental strain, this modulus in incremental form could be obtained as*

$$\mathbb{C}_{\text{alg}} = \frac{\partial \Delta \mathbf{T}}{\partial \Delta \mathbf{E}} = \mathbb{C} - 2\mu \widehat{\mathbf{N}}^{\text{trial}} \otimes \frac{\partial \Delta \gamma}{\partial \Delta \mathbf{E}} - 2\mu \Delta \gamma \frac{\partial \widehat{\mathbf{N}}^{\text{trial}}}{\partial \Delta \mathbf{E}}. \quad (9.43)$$

For von Mises yield criterion, we obtain

$$\mathbb{C}_{\text{alg}} = \mathbb{C} - 4 \frac{\mu^2}{\mathcal{M}} \widehat{\mathbf{N}}^{\text{trial}} \otimes \widehat{\mathbf{N}}^{\text{trial}} - \frac{4\mu^2 \Delta \gamma}{\|\boldsymbol{\xi}^{\text{trial}}\|} \{ \mathbb{I} - \mathbf{I} \otimes \mathbf{I} - \widehat{\mathbf{N}}^{\text{trial}} \otimes \widehat{\mathbf{N}}^{\text{trial}} \}, \quad (9.44)$$

where \otimes denote tensor product, \mathbb{I} is fourth order symmetric identity tensor, and scalar coefficient \mathcal{M} is defined as:

$$\mathcal{M} = 2\mu + I_{\text{kin}} + \sqrt{\frac{2}{3} \frac{\partial I_{\text{kin}}}{\partial \kappa} \Delta \gamma} + \frac{2}{3} \frac{\partial \sigma_y^*}{\partial \kappa}$$

We refer to [102] for complete derivation of equation (9.44). The coefficient \mathcal{M} for model I and model II could be obtained as:

$$\textbf{Model I} \quad \mathcal{M} = 2\mu + \frac{2}{3}H \text{ and}$$

$$\textbf{Model II} \quad \mathcal{M} = 2\mu + \frac{n_w \sigma_0 (\zeta c + 1)}{\kappa_0} \left(1 + \frac{\kappa}{\kappa_0} \right)^{n_w}.$$

9.2.2 A solver for diffusion subproblem

The maximum-principles-preserving solver for the diffusion subproblem is devised by posing the subproblem as a convex quadratic program and employing associated optimization solvers.

Before elaborating on the numerical scheme for solving the diffusion problem, we provide a mathematical argument that establishes bounds for $c(\mathbf{x})$ in Ω for the coupled problem. From theory, we know that elliptic boundary value problems such as the diffusion equation enjoy a maximum principle under appropriate regularity assumptions on the domain and input parameters [73]. The non-negativity constraint is the physical implication of maximum principles under certain conditions on the forcing function and boundary conditions. In a continuous system, this property could *a priori* estimate the non-negativity of concentration field via assessing c values on boundaries. A maximum principle for diffusion with decay was first proposed by [89] and could be stated as

Let $c(\mathbf{x}) \in C^2(\Omega) \cup C^0(\bar{\Omega})$ satisfy the following differential inequality

$$-\operatorname{div}[\mathbf{D}(\mathbf{x})\operatorname{grad}[c]] = m(\mathbf{x}) \leq 0 \quad \text{in } \Omega, \quad (9.46)$$

where diffusivity tensor, which is a function of displacement field in a two-way coupling, is symmetric, continuously differentiable, and uniformly elliptic (i.e., there exists $0 < c_1 \leq c_2 < +\infty$, such that $c_1 \mathbf{y}^T \mathbf{y} \leq \mathbf{y}^T \mathbf{D}(\mathbf{x}) \mathbf{y} \leq c_2 \mathbf{y}^T \mathbf{y}$ for every $\mathbf{x} \in \Omega$ and $\mathbf{y} \in \mathbb{R}^{nd}$). Then $c(\mathbf{x})$ satisfies a continuous maximum principle of the following form

$$\max_{\mathbf{x} \in \bar{\Omega}} c(\mathbf{x}) \leq \max \left[0, \max_{\mathbf{x} \in \Gamma_c^D} [c^p(\mathbf{x})] \right]. \quad (9.47)$$

Note that if $f(\mathbf{x}) \geq 0$ and $c^p \geq 0$, based on the above equation we can conclude that $c(\mathbf{x}) \geq 0$ in the whole domain. When employing well-known discretization methods, the consequent discrete system should also preserve such fundamental properties. However, many numerical formulations such as finite element, finite difference, finite volume, lattice-Boltzmann, discontinuous Galerkin method, and spectral element method are not expected to satisfy maximum principles and the non-negative constraints for diffusion equation, even with exhaustive mesh refinements and polynomial refinements [134]. This is due to the lack of an included mechanism in these numerical formulations to address

these constraints. We now start with the variational form of single-field (concentration) formulation and then modify the ensuing discrete problem to meet the non-negative constraint. We shall define the following function spaces:

$$\mathcal{P} := \{c(\mathbf{x}) \in H^1(\Omega) \mid c(\mathbf{x}) = c^p(\mathbf{x}) \text{ on } \Gamma_c^D\} \text{ and} \quad (9.48a)$$

$$\mathcal{Q} := \{q(\mathbf{x}) \in H^1(\Omega) \mid q(\mathbf{x}) = 0 \text{ on } \Gamma_c^D\}. \quad (9.48b)$$

The single-field Galerkin formulation for the pure tensorial diffusion problem reads: Find $c \in \mathcal{P}$ such that we have

$$\mathcal{B}_c(q; c) = \mathcal{L}_c(q) \quad \forall q(\mathbf{x}) \in \mathcal{Q}. \quad (9.49)$$

where bilinear form and linear functional are, respectively, defined as:

$$\begin{aligned} \mathcal{B}_c(q; c) &:= \int_{\Omega} \text{grad}[q] \cdot \mathbf{D}(\mathbf{x}) \text{grad}[c] \, d\Omega \text{ and} \\ \mathcal{L}_c(q) &:= \int_{\Omega} q(\mathbf{x}) m(\mathbf{x}) \, d\Omega + \int_{\Gamma_c^N} q(\mathbf{x}) h^p(\mathbf{x}) \, d\Gamma. \end{aligned}$$

Since bilinear form is symmetric, by using Vainberg's theorem our weak form has a corresponding variational statement, which can be written as

$$\underset{c(\mathbf{x}) \in \mathcal{P}}{\text{minimize}} = \frac{1}{2} \mathcal{B}_c(c; c) - \mathcal{L}_c(c). \quad (9.51)$$

9.2.2.1 Optimized-based solver for diffusion problem

It is of paramount importance to note that the concentration is a non-negative quantity, and a robust numerical solver must not violate the non-negative constraint at any instances. [114, 136] proposed an optimization-based finite element method that is the result of imposing the bounds on nodal solutions of minimization statement, constructed by invoking Vainberg's theorem on Galerkin formulation. In the rest of the chapter, we use the symbols \preceq and \succeq to denote component-wise inequalities for vectors (i.e., for any

two finite dimensional vectors \mathbf{a} and \mathbf{b} , $\mathbf{a} \preceq \mathbf{b}$, means $a_i \leq b_i$). After spatial discretization using finite elements, for a given nodal displacement \mathbf{u} , the discrete equation for the diffusion problem takes the following form

$$\mathbf{K}_c(\mathbf{u})\mathbf{c} = \mathbf{f}_c, \quad (9.52)$$

where \mathbf{K}_c is symmetric positive definite matrix, \mathbf{c} is the vector containing nodal concentrations, and \mathbf{f}_c is the nodal source vector. To enforce maximum principle, we shall write our weak form as the following minimization problem

$$\underset{\mathbf{c} \in \mathbb{R}^{\text{ndof}}}{\text{minimize}} = \frac{1}{2} \langle \mathbf{c}; \mathbf{K}_c(\mathbf{u})\mathbf{c} \rangle - \langle \mathbf{c}; \mathbf{f}_c \rangle \quad (9.53a)$$

$$\text{subject to } c_{\min} \mathbf{1} \preceq \mathbf{c} \preceq c_{\max} \mathbf{1}, \quad (9.53b)$$

where $\langle \cdot; \cdot \rangle$ represents the standard inner product on Euclidean space, $\mathbf{1}$ denotes a vector of ones of size $\text{ndofs} \times 1$ and ndofs denotes number of degrees-of-freedom in the nodal concentration vector. $c_{\min} := \min_{\mathbf{x} \in \partial\Omega} [0, \min c^p(\mathbf{x})]$ and $c_{\max} := \max_{\mathbf{x} \in \partial\Omega} [0, \max c^p(\mathbf{x})]$ are, respectively the lower and upper bounds for c . Note that by setting $c_{\min} = 0$ and $c_{\max} = +\infty$, we can obtain non-negative constraint. Equation (9.53) is a constrained optimization problem that belongs to convex quadratic programming and unique global minimizer could be found. The first order optimality condition for this problem could be stated as:

$$\mathbf{K}_c(\mathbf{u})\mathbf{c} = \mathbf{f}_c + \boldsymbol{\lambda}_{\min} - \boldsymbol{\lambda}_{\max}, \quad (9.54a)$$

$$c_{\min} \mathbf{1} \preceq \mathbf{c} \preceq c_{\max} \mathbf{1}, \quad (9.54b)$$

$$\boldsymbol{\lambda}_{\min} \succeq \mathbf{0}, \quad (9.54c)$$

$$\boldsymbol{\lambda}_{\max} \succeq \mathbf{0}, \quad (9.54d)$$

$$(\mathbf{c} - c_{\min} \mathbf{1}) \cdot \boldsymbol{\lambda}_{\min} = 0, \text{ and} \quad (9.54e)$$

$$(c_{\max} \mathbf{1} - \mathbf{c}) \cdot \boldsymbol{\lambda}_{\max} = 0, \quad (9.54f)$$

where $\boldsymbol{\lambda}_{\min}$ and $\boldsymbol{\lambda}_{\max}$ are vectors of Lagrange multipliers corresponding to $\mathbf{c} \succeq c_{\min} \mathbf{1}$ and $\mathbf{c} \preceq c_{\max} \mathbf{1}$, respectively. There are many robust numerical algorithms available to solve this constrained optimization problem (e.g., active set strategy, interior point methods, trust region, \dots). In this chapter, we have applied the trust-region-reflective method in all our simulation. Detailed discussions of trust-region is provided by [131, 74].

9.2.3 A coupling algorithm

Solution strategies for multiphysics problems are mainly divided into monolithic and staggered methods. The monolithic approach treats both problems (deformation and diffusion) in a single system of equations. Despite its unconditional stability, it leads to a large and non-symmetric system of equations that requires a high memory bandwidth and thus high computational cost. The staggered approach (which hinges on operator-split techniques) is designed to reduce the computational costs via partitioning the problem into two sub-problems, and each sub-problem is treated by a different numerical scheme. Detailed discussion on staggered and monolithic methods can be found in [69, 100, 125]. In this chapter, we will focus only on the staggered method as there is no straightforward way to incorporate out optimization-based formulation within a monolithic scheme. The various step of our coupling algorithm is summarized in Algorithm 2.

9.2.4 Computer implementation

The numerical simulations performed through scripting via COMSOL [116], which is a bidirectional computational interface. This allowed us to utilize finite element capabilities in [57] and integrate it with optimization packages in the MATLAB [126].

Algorithm 2 Staggered coupling algorithm for elastoplasticity-diffusion system

```

1: Initialize  $\mathbf{u}_0 = \mathbf{0}$ ,  $\mathbf{q}_0^p = \mathbf{0}$ .
2: Set  $\mathbf{c}_0 \succeq \mathbf{0}$ .
3: for  $n = 0, \dots, \mathcal{T}$  do ▷ Begin load step
4:   CALL DEFORMATION SOLVER: obtain  $\mathbf{u}_{n+1}$ .
5:   Traction increment:  $\Delta\mathbf{q}_n^p$ .
6:   Initialize:  $\Delta\mathbf{u}_n^{(i)} = \mathbf{0}$ ,  $\Delta\gamma_n^{(i)} = 0$ .
7:   for  $i = 0, \dots$  do ▷ Begin Newton's iteration
8:     Residual vector:  $\mathbf{r}_u^{(n,i)}$ ,
9:     if  $\|\mathbf{r}_u^{(n,i)}\| < \epsilon_{\text{Tol}}$  then ▷ Check convergence
10:      BREAK ▷ Go to next load step
11:    else ▷ Continue iterations
12:      Tangent stiffness:  $\mathbf{K}_u^{(n,i)}(\mathbf{c}_n)$ .
13:      Solve:  $\mathbf{K}_u^{(n,i)}(\mathbf{c}_n)\delta\mathbf{u} = -\mathbf{r}^{(n,i)}$ .
14:    end if
15:    Update:  $\Delta\mathbf{u}_n^{(i+1)} = \Delta\mathbf{u}_n^{(i)} + \delta\mathbf{u}$ .
16:    Calculate:  $\Delta\mathbf{E}_n^{(i+1)}$ .
17:    Stress update:  $\mathbf{T}_{n+1}^{(i+1)}$ ,  $\Delta\gamma_n^{(i+1)}$ . ▷ (check Algorithm1)
18:
19:  end for ▷ End Newton's iteration
20:  Update:  $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta\mathbf{u}_n^{(i+1)}$  and
21:           $\mathbf{q}_{n+1}^p = \mathbf{q}_n^p + \Delta\mathbf{q}_n^p$ .
22:  CALL DIFFUSION SOLVER: obtain  $\mathbf{c}_{n+1}$  by solving the following minimization
   problem:

```

$$\underset{c \in \mathbb{R}^{\text{nDof}}}{\text{minimize}} = \frac{1}{2}(\mathbf{c}_{n+1}; \mathbf{K}_c(\mathbf{u}_{n+1})\mathbf{c}_{n+1}) - (\mathbf{c}_{n+1}; \mathbf{f}_c)$$

$$\text{subject to } c_{\min}\mathbf{1} \preceq \mathbf{c}_{n+1} \preceq c_{\max}\mathbf{1}$$

```

23: end for ▷ End load step


---



```

In our numerical simulations, GMRES iterative solver [158] preconditioned with algebraic multigrid V-cycle based on smoothed aggregation [172] was used for deformation problem, and the relative convergence tolerance of $1e^{-6}$ was employed. For diffusion problem, trust-region-reflective algorithm from [146] in MATLAB used and the relative tolerance set to $1e^{-14}$.

9.3 Performance of the computational framework

In this section, we solve the coupled elastoplasticity-diffusion model in a plane stress problem to demonstrate the implementation of the framework proposed in Section 9.2. We first establish the need for a non-negative algorithm in both degradation model I and II by illustrating the failure of conventional CG formulation in capturing correct c profiles. These failures appear as unphysical c values that cascade to next loading step and results in numerical errors also in deformation problem. We show that the proposed computational framework suppresses the source of numerical artifacts and produces physical and reliable solutions. We also systematically study the effect of the concentration of the diffusant on the deformation of solid and vice versa and compare the results with uncoupled cases. We will also comment on the performance of the proposed solver in terms of iteration count and time-to-solution.

9.3.1 Degradation of plate with a circular hole

The computational domain and boundary conditions for deformation and diffusion problems are depicted in figure 9.5. The pure deformation problem has been frequently solved for linear anisotropic hardening problem and can be found in [186]. Due to the double symmetry of the geometry, we only analyze the one-quarter of the plate. A total

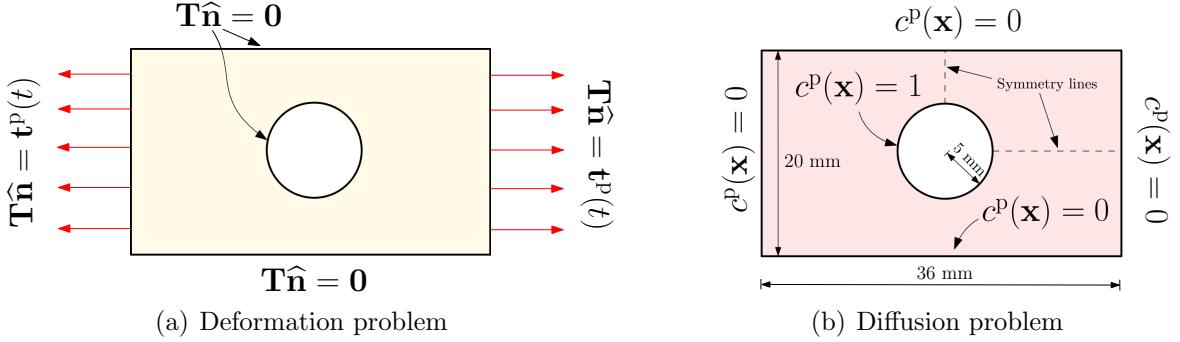


Figure 9.5: Plate with a circular hole: This figure provides a pictorial description of the geometry and boundary value problem.

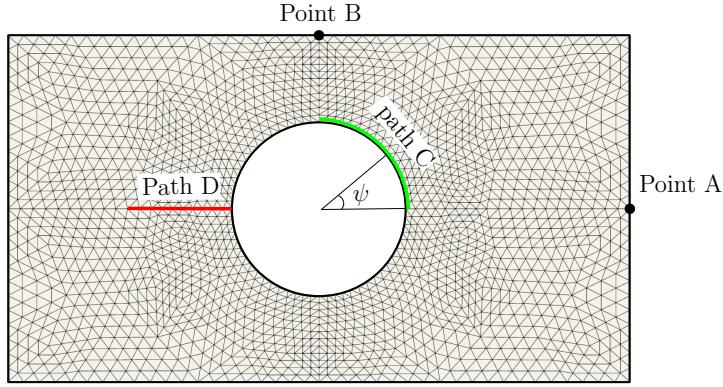


Figure 9.6: Plate with a circular hole: This figure shows the tree-node triangular mesh used in the numerical results. Position of points A and B, and lines C and D are also marked.

of 56 three-node isoparametric triangles with linear interpolation of concentration and displacement fields are employed in the calculation (see figure 9.6).

In this problem, a unit thickness is assumed, and the calculation is performed by imposing the right edge to a uniform tensile load, which linearly increases from zero to a maximum value of 133 MPa in 1.2 s and is linearly unloaded in the next 1.2 s. The maximum traction is chosen such that the mean stress over the section passing through the hole is 10% above the yield stress for the uncoupled model I (linear isotropic hardening). The traction is prescribed in a total of 2.2 s in 26 steps. We took one large increment (0.44 s) up to elastic limit followed by equal increments of 0.05 s up

Table 9.1: Parameters for plane with a circular hole problem.

General parameters	Value
m	0
ψ	$\pi/3$
(λ_0, μ_0)	$(1.94e^{10}, 2.92e^{10})$
σ_0	$243e^6$
E_{ref}	0.001
(d_1, d_2)	(50,1)
Model I parameters	value
(λ_1, μ_1)	$(-8.5e^8, -8.5e^8)$
E_t	$2.171e^9$
c_{ref}	0.05
(ϕ_T, ϕ_S)	(1.2,1.2)
Model II parameters	value
n_w	5
ζ	-0.3
(ϕ_T, ϕ_S)	(1.25,1.25)

to maximum load. Due to path-dependency of elastoplastic solution, relatively small increments chosen when anticipating a plastic flow in the loading stage. The loading increment relaxed at the onset of the unloading stage to 0.2 s. As reversed plastic flow can occur during the unloading, relatively small increments (0.05 s) assigned at the end of the unloading stage. Although this loading pattern designed based on the uncoupled model I, in order to be consistent, we use the same loading pattern for all problems solved in this section. The material parameters and data-set for this problem are described in Table 9.1. One should note that since the plate is thin and the loads are in-plane, we can assume a plane stress condition and hence no special treatment of the incompressibility constraint is needed.

9.3.2 Non-negative (NN) vs continuous Galerkin (CG)

In this subsection, we highlight the importance of non-negative solutions and its impact on coupled elastoplastic-diffusion analyses of a plate with a circular hole undergoing one cycle of uniaxial loading-unloading. We applied two-way coupling strategy and studied both degradation model I and II. From **Figs 9.7** and **9.8**, it is evident that the proposed non-negative formulation satisfies the earlier mentioned condition and produces physically meaningful concentration, whereas the continuous Galerkin formulation produces negative, unphysical concentrations for both models I and II. In the degradation model I, all violations occur as negative values. However, as shown in **Fig 9.8** for degradation model II, continuous Galerkin formulation violates both upper-bound and lower-bound constraints.

Fig 9.9 shows the evolution of concentration profile measured on path D for three loading steps. The discrepancy between continuous Galerkin and non-negative formulation is not limited to maximum loading step but it is observed throughout the whole loading process and varies in degradation model I and model II. The success of coupled elastoplastic-diffusion analysis relies on the performance of each subproblem, and the violations occurred in diffusion solution affects the deformation solution. **Fig 9.10** compares the stress profiles and effective plastic strain contours at the residual loading step. It can be seen that continuous Galerkin formulation generates slightly different stress profiles compared to the non-negative formulation.

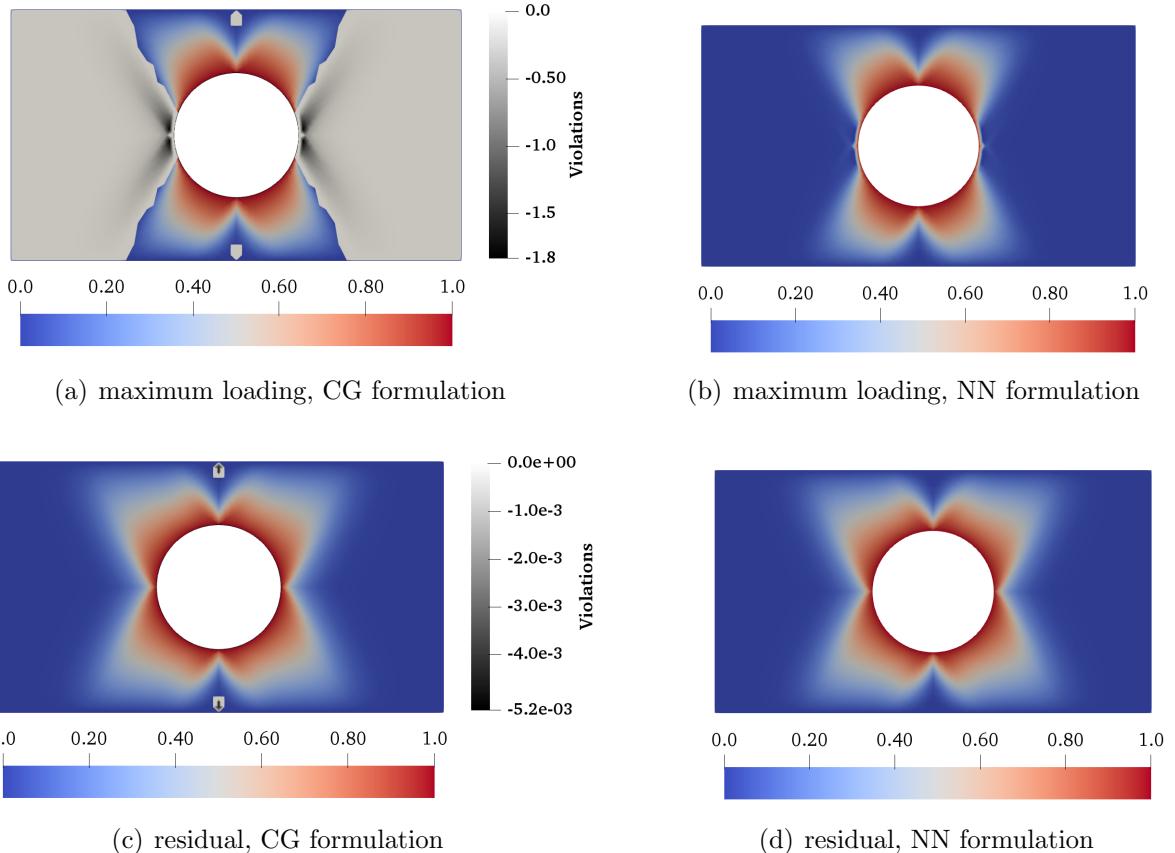


Figure 9.7: **Degradation model I:** This figure compares concentration profiles from the continuous Galerkin formulation and the non-negative formulation at two stages. Gray-scale region in the left figures indicates violated concentrations.

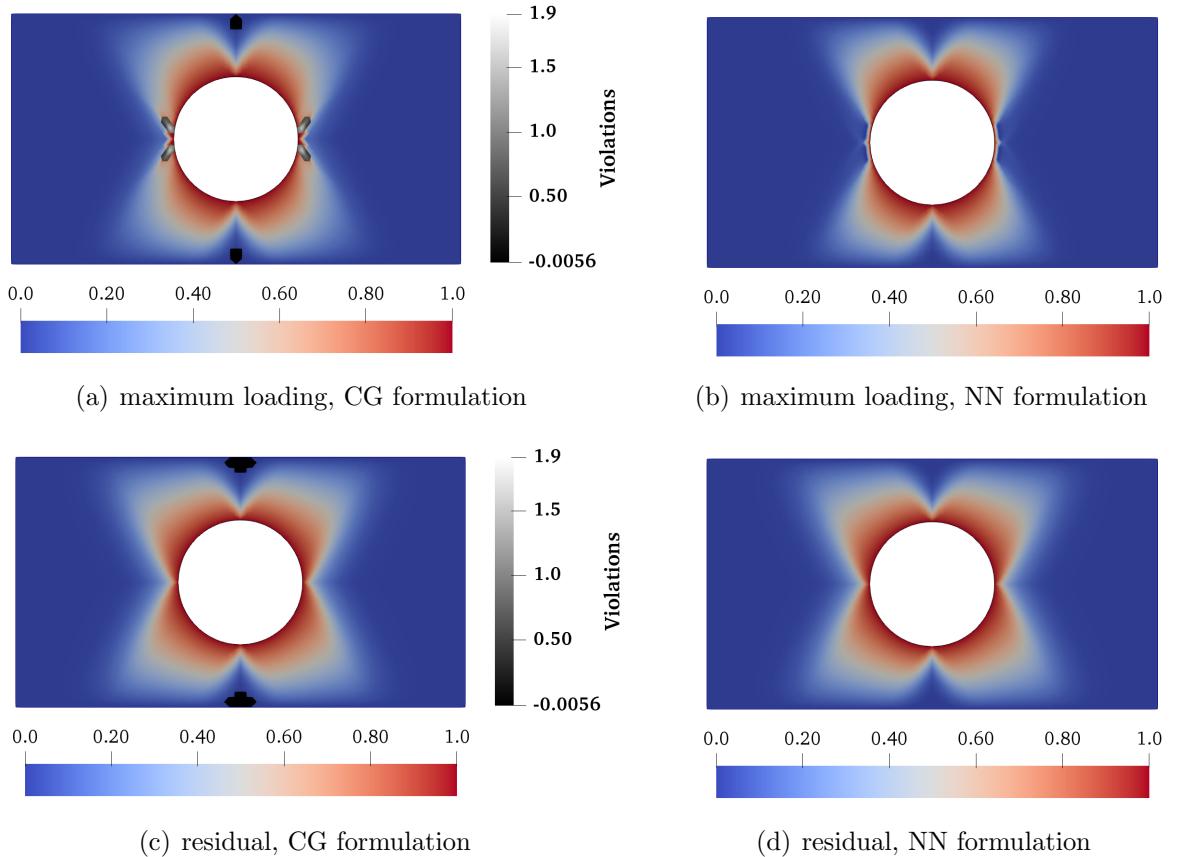


Figure 9.8: **Degradation model II:** This figure compares concentration profiles from the continuous Galerkin formulation and the non-negative formulation. Gray-scale region in the left figure indicates violated concentrations.

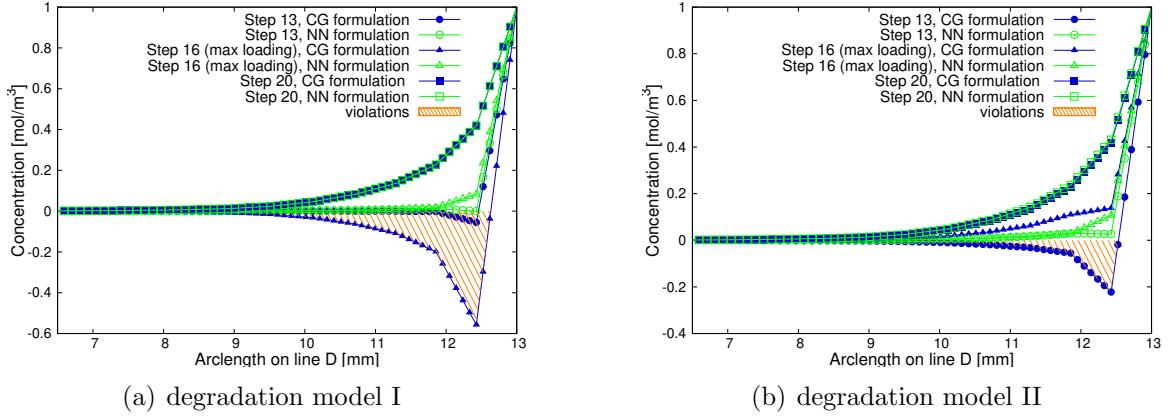


Figure 9.9: **Continuous Galerkin vs non-negative formulation:** This figure illustrates the concentration profile on path D during three loading steps. Coordinates of path D is marked in **Fig 9.6**.

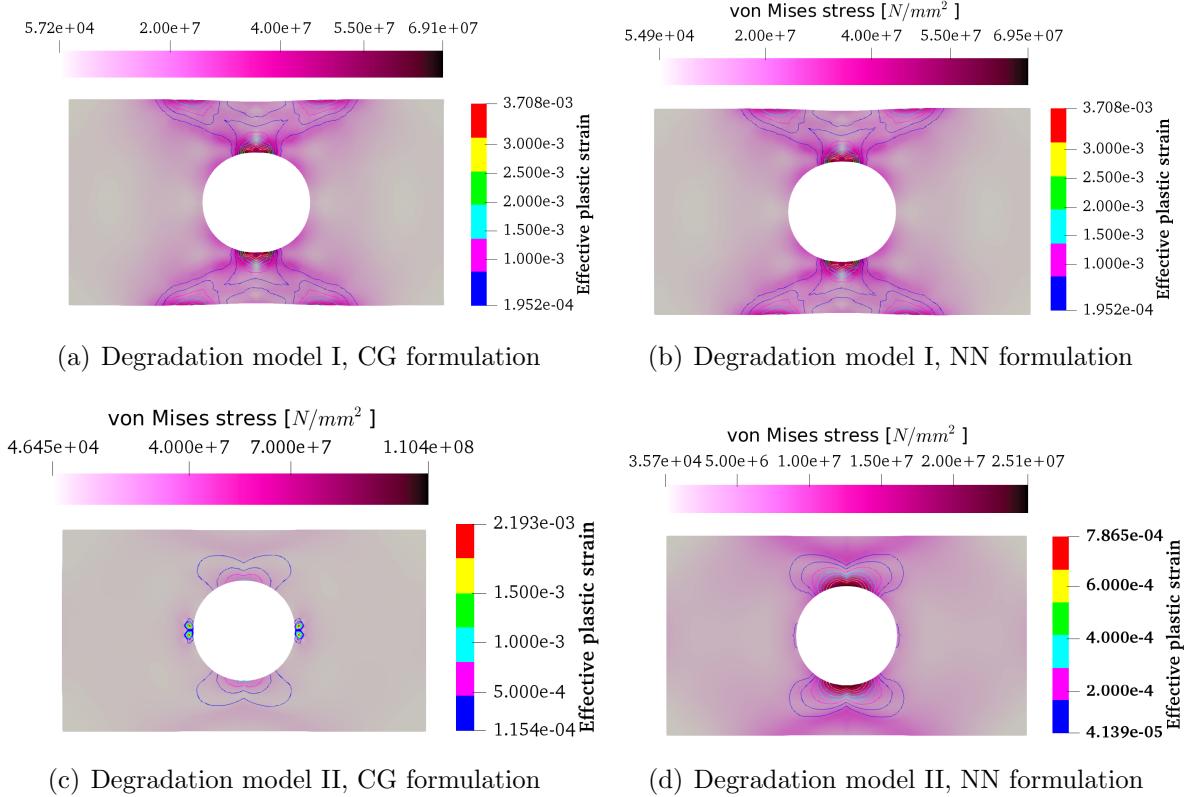


Figure 9.10: **Residual loading step:** This figure compares stress profile and effective plastic strain contours from the continuous Galerkin formulation and the non-negative formulation at residual loading step.

9.3.3 Performance of the staggered scheme and the trust-region method

It should be noted that for each loading step in the staggered coupling algorithm, the deformation solver, and either continuous Galerkin or trust-region algorithm for diffusion problem should converge. The convergence and time-to-solution histories of the plate with a circular hole under two-way coupling are shown in Table 9.2 for degradation model I. The data are collected for six loading steps, and we see that for deformation problem, time-to-solution and number of iterations remained almost unchanged regardless of the formulation employed in diffusion solver. In deformation solver in both formulations, the majority of clock-time is spent on assembly phase. However, as expected, non-negative formulation appears to take higher clock-time in solver phase than continuous Galerkin formulation. Despite requiring lower solver clock-time and a fixed iteration count, continuous Galerkin formulation leads to violations of maximum principal. The percent of these violations increases near the maximum loading step. Table 9.3 contains performance results captured for degradation model II. We observe similar trends to model I with respect to iteration count and time-to-solution for both solvers.

9.3.4 Physics of coupled deformation-diffusion problem

We illustrate how accumulated stress and propagation of plastic zone in solid affect the diffusion process, and how the concentration of the diffusant affect the deformation of the solid. We perform the calculation using different coupling strategies (uncoupled, one-way, two-way) and compare the results with standard linear elasticity and perfect plasticity problems. The problem set and material properties for this problem described earlier in subsection 9.3.1. In this chapter, in plots pertaining to the deformation problem,

Table 9.2: Degradation model I: This table shows time-to-solution and iteration counts under continuous Galerkin and non-negative strategies for both deformation and diffusion subproblems.

Loading Step	Continuous Galerkin formulation						Proposed NN formulation					
	# of iter.	Deformation Time	# of iter.	Time	% of Solver violations	Deformation iter. time	# of iter.	Total time	# of PCG iter.	Assembly	Diffusion iter.	Solver Time
1	1	0.589	2	1.385	0.003	0.61	1	0.653	23	948	1.556	0.085
6	28	0.701	2	1.891	0.003	0.41	28	0.886	19	693	1.807	0.064
11	28	0.906	2	1.875	0.002	0.41	28	0.664	22	953	1.720	0.085
16	33	0.707	2	1.674	0.003	61.76	33	0.705	23	641	1.899	0.069
21	28	0.654	2	1.670	0.002	0.41	28	0.851	21	814	1.661	0.064
26	28	1.034	2	1.819	0.002	0.41	28	0.710	21	675	1.7178	0.064

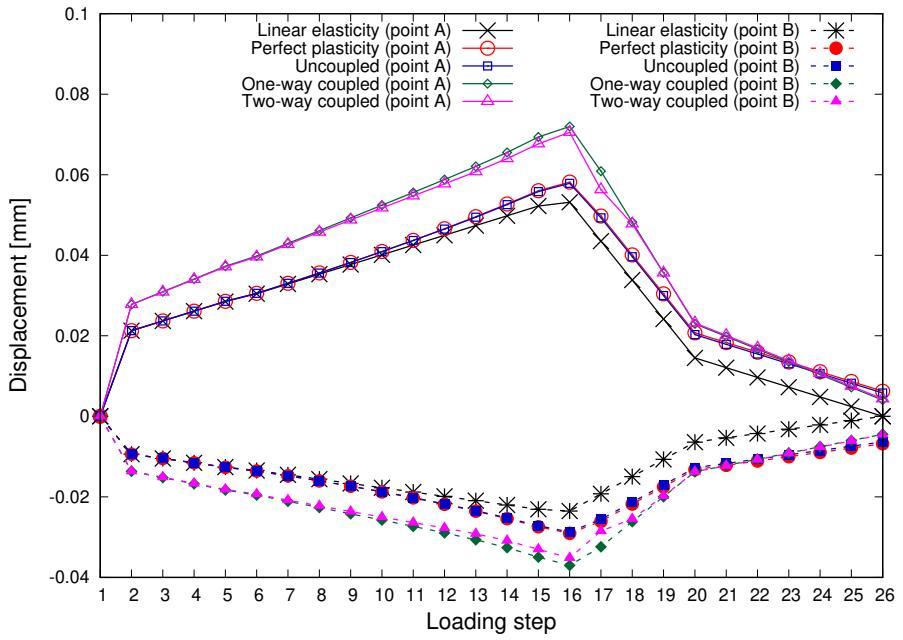
Table 9.3: Degradation model II: This table shows time-to-solution and iteration counts under continuous Galerkin and non-negative strategies for both deformation and diffusion subproblems.

Loading Step	Continuous Galerkin formulation						Proposed NN formulation					
	Deformation		Diffusion				Deformation		Diffusion			
# of iter.	Total Time	# of iter.	Time	Assembly	Solver	% of violations	# of iter.	Total time	# of iter.	# of PCG iter.	Assembly Time	Solver
1	1	0.556	2	1.294	0.003	0.61	1	0.540	23	805	1.259	0.064
6	28	0.690	2	1.892	0.002	0.41	28	0.679	21	828	1.938	0.065
11	28	0.909	2	1.748	0.002	0.41	28	1.025	25	1049	1.621	0.079
16	28	0.717	2	1.517	0.002	0.82	28	0.711	22	563	1.504	0.055
21	28	0.620	2	1.964	0.003	0.41	28	0.684	22	862	2.082	0.082
26	28	0.957	2	1.669	0.002	0.61	28	0.855	23	939	1.784	0.083

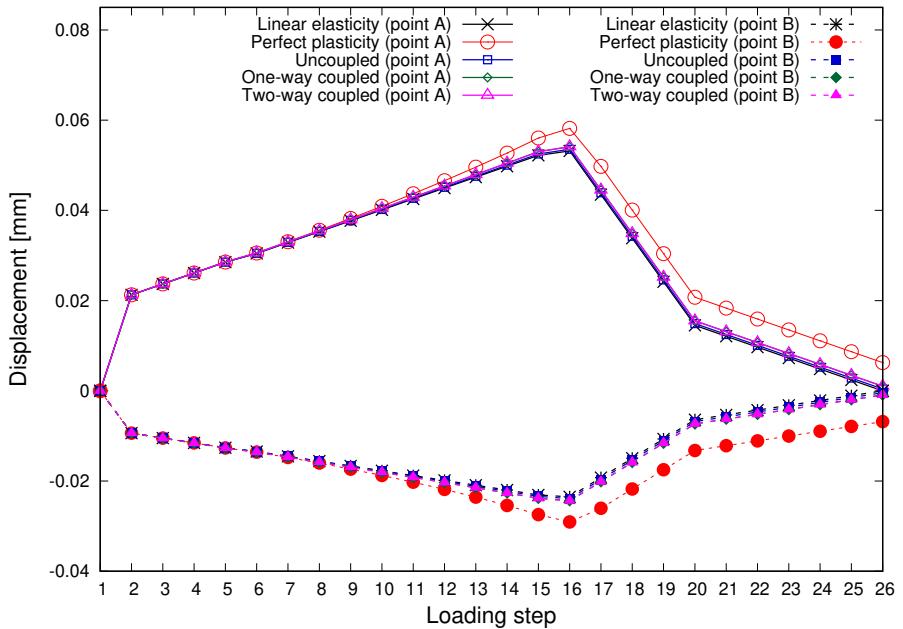
the displacement fields are scaled up to 50 times for better visualization.

Fig 9.11(a) shows displacement of point A and B for degradation model I. As expected, the diffusant propagation in the domain has degenerated the stiffness of solid matrix for coupled cases and compared to uncoupled and perfectly plastic cases (which are unaffected by diffusant), increased displacement is observed at every loading step. At maximum loading step, one-way and two-way coupled cases showed, respectively, 24% and 21% increase in displacement at point A, compared to perfect plasticity case. From **Fig 9.11(b)**, one can observe that under degradation model II, unlike model I, the displacement values are not exceeding the perfect plasticity case. This difference stems from the mechanism model I and model II are taking to be coupled with diffusion problem. In model I, as shown in **Fig 9.1(a)**, the initial yield stress is fixed during degradation, and as the concentration of diffusant grows, displacement at which yielding occurs is postponed. This results in a global relaxation of structure. However, model II shrinks the upper bound of elastic limit function at affected areas (see **Fig 9.2(a)**). This implies that yielding occurs at lower displacement. In other words, The structure experience a pseudo-softening behavior.

Stress profile and contour of effective plastic strain for the degradation model I is shown in **Fig 9.12**. At maximum loading step, it is evident that thicker shear bands are forming for one-way and two-way coupled cases. However, the intensity of stress profiles remained unchanged on the shear bands. Residual stresses for coupled cases are spatially more distributed but are significantly lower than stresses observed of for the uncoupled case. In degradation model II, similar to model I, stresses during maximum loading are approximately equal for all three cases. However, unlike model I, one-way and two-way coupling strategies contain higher residual stresses than uncoupled case (see **Fig 9.13**). We know that during unloading steps, the material follows the elastic stress path. As



(a) Degradation model I



(b) Degradation model II

Figure 9.11: Plate with a circular hole: This figure shows displacement history at point A and B.

discussed earlier, model I decreases the slope of the elastic path as degradation grows. This will result in lower stress at the end of the unloading step.

When the effective plastic strain is positive, the material yields. For uncoupled problems, during the loading stage, the plastic zone monotonically grows and will not change during the unloading stage. This could be seen in **Figs 9.14(a)–9.14(b)** and **Fig 9.15(a)–9.15(b)** for model I and model II, respectively. However, for either one-way or two-way coupled cases, the yielding function introduced in equations 9.17 and 9.21 updates. Hence the plastic zone could vary at each load step as the concentration profile evolves. **Fig 9.14** presents the location of the plastic zone during maximum and residual loading steps for degradation model I. For the case of coupled problems, the zone has grown at maximum loading and spreads to upper and lower edges at the vertical centerline. But, significantly shrinks as the unloading stage ends. Results for degradation model II are shown in **Fig 9.15**. X-patterns appear at maximum loading for both coupled cases, and the pattern does not change at the residual loading step. To qualitatively compare model I and II with respect to the evolution of the plastic zone, we use a global metric of the percentage of yielded area in a domain. **Fig 9.16** contains the result of this metric for three cases of uncoupled, one-way coupled, and two-way coupled. It is evident that coupled strategies increase the area of the plastic zone in the degradation model I. This increase is also observed in model II but is less pronounced. It could be seen that one-way coupling overestimates the area of the plastic zone for both models I and II.

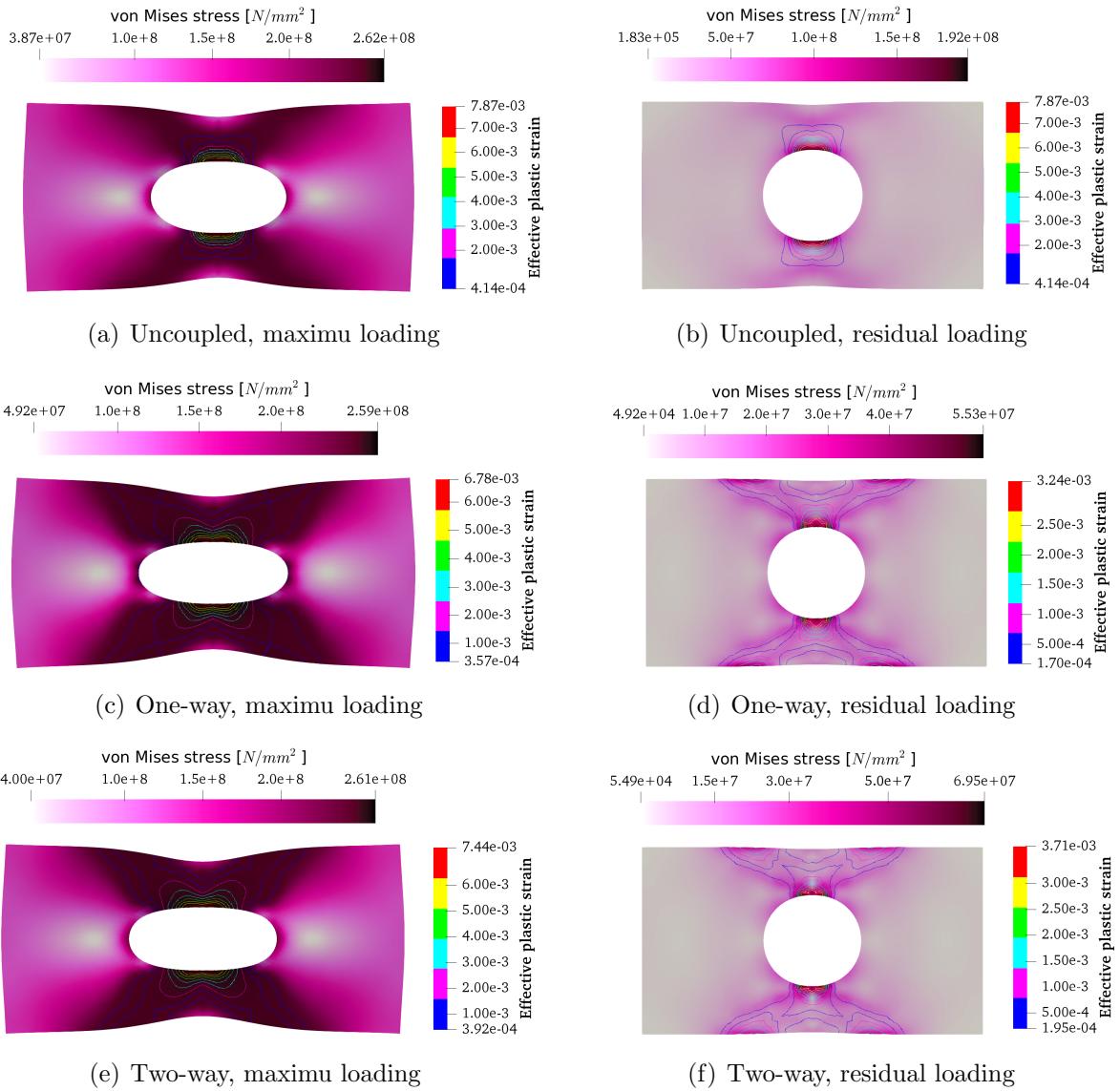
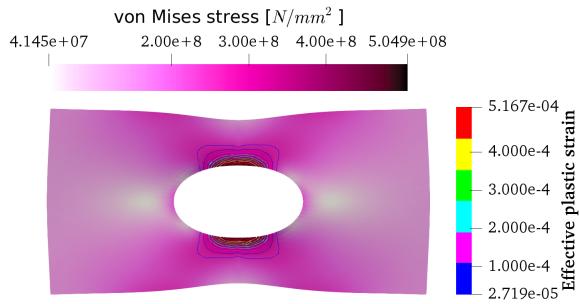


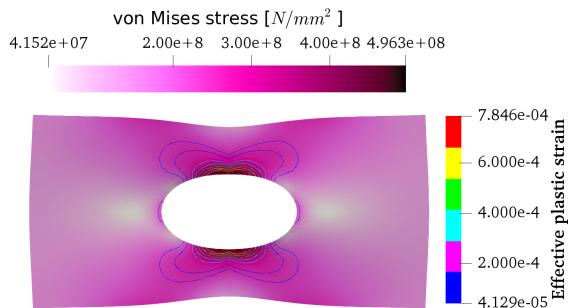
Figure 9.12: **Stress, degradation model I:** This figure illustrates concentration profile and contour of effective plastic strain at maximum and residual loading steps.



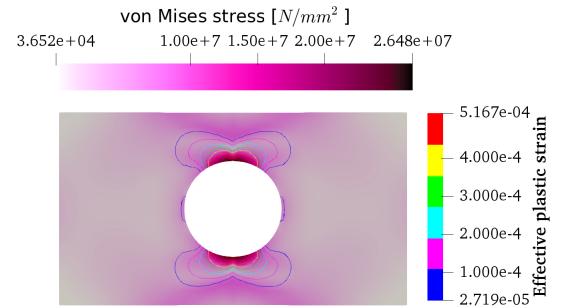
(a) Uncoupled, maximum loading



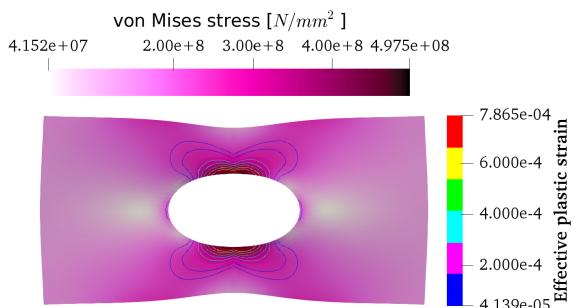
(b) Uncoupled, residual loading



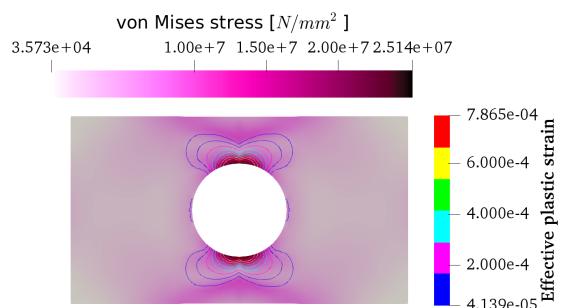
(c) One-way, maximum loading



(d) One-way, residual loading

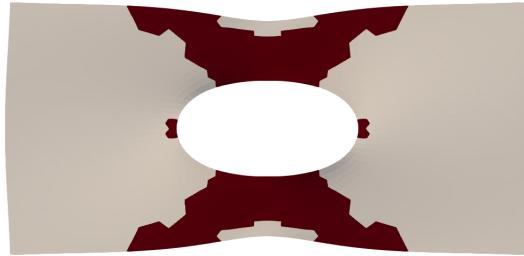


(e) Two-way, maximum loading

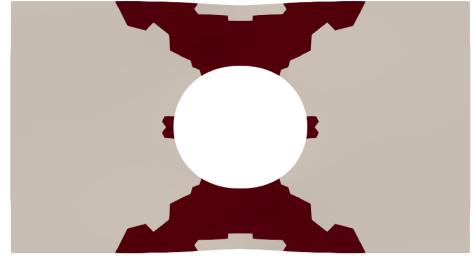


(f) Two-way, residual loading

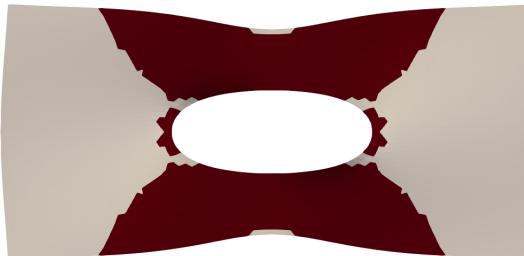
Figure 9.13: Stress, degradation model II: This figure illustrates concentration profile and contour of effective plastic strain at maximum and residual loading steps.



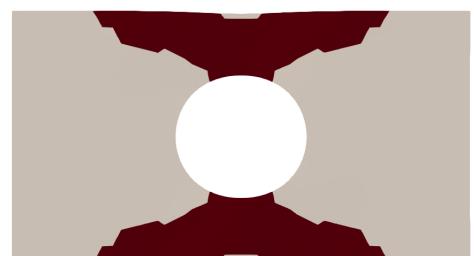
(a) Uncoupled, maximum loading



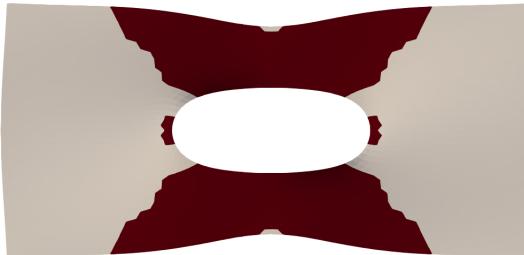
(b) Uncoupled, residual loading



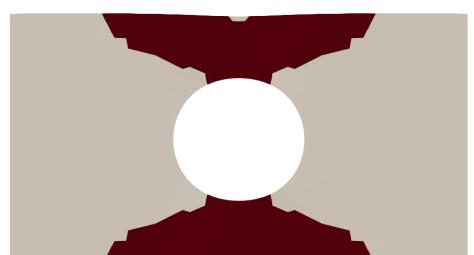
(c) One-way, maximum loading



(d) One-way, residual loading

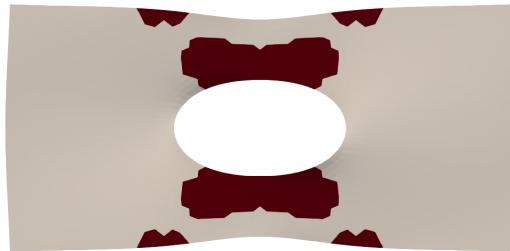


(e) Two-way, maximum loading

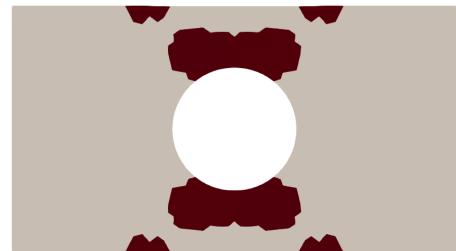


(f) Two-way, residual loading

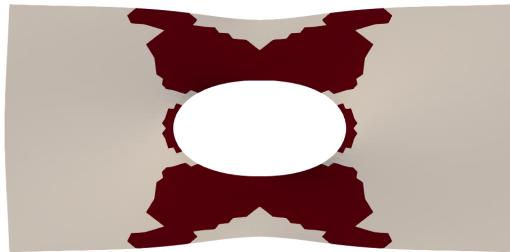
Figure 9.14: degradation model I: This figure shows the plastic zone at maximum and residual loading steps.



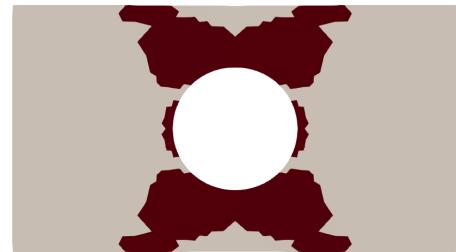
(a) Uncoupled, maximum loading



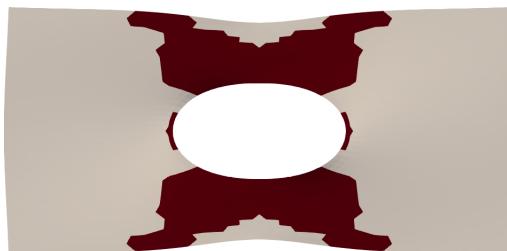
(b) Uncoupled, residual loading



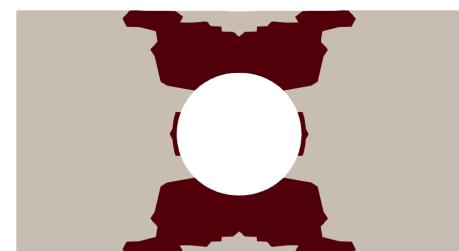
(c) One-way, maximum loading



(d) One-way, residual loading



(e) Two-way, maximum loading



(f) Two-way, residual loading

Figure 9.15: degradation model II: This figure shows the plastic zone at maximum and residual loading steps.

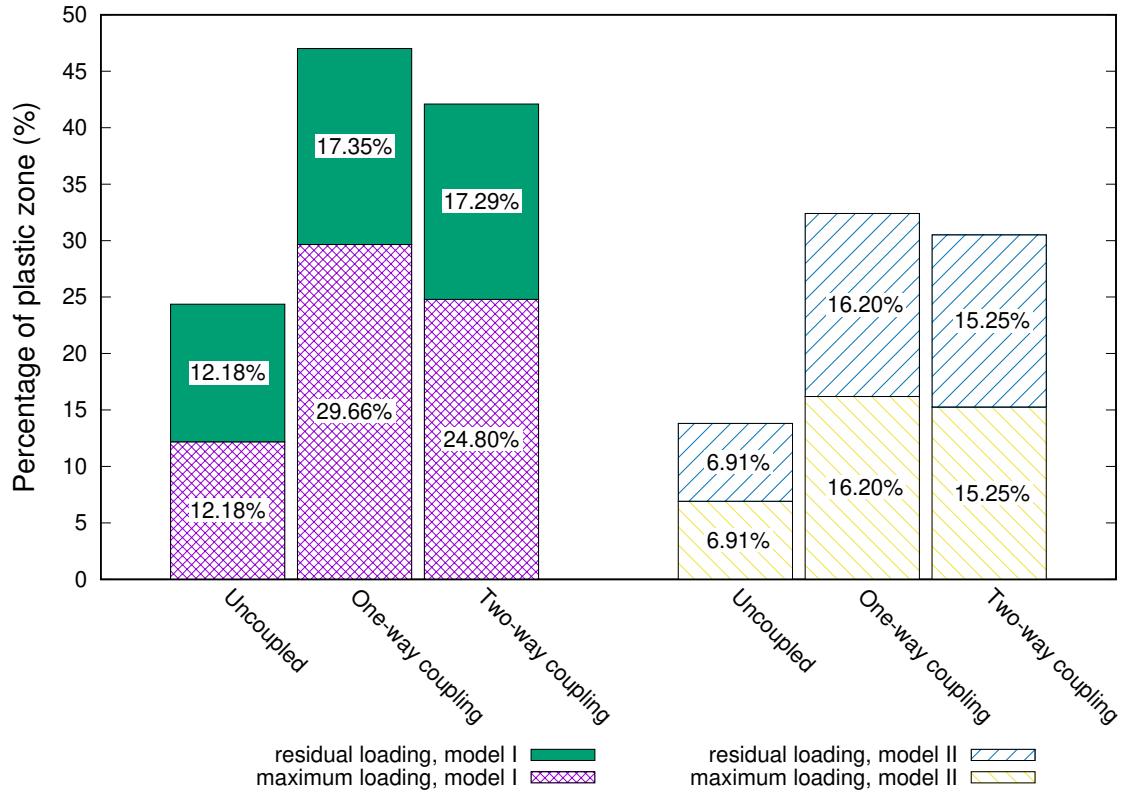


Figure 9.16: Plate with a circular hole: This figure shows percentage of yielded area for degradation model I and II.

Chapter 10

Conclusions and Future Work

A new stabilized mixed DG formulation has been introduced for the DPP mathematical model in Chapter 3, which describes the flow of a single-phase incompressible fluid through a porous medium with two dominant pore-networks. Some of the main findings on the *computational front* and the *nature of flow* through porous media with double pore-networks can be summarized as:

- (i) Arbitrary combinations of interpolation functions for the field variables are stable under the proposed DG formulation. Unlike the classical mixed DG formulation, which violates the LBB *inf-sup* stability condition under the equal-order interpolation for all the field variables, the proposed DG formulation circumvents the LBB condition. This implies that the proposed DG formulation does not suffer from node-to-node spurious oscillations when the computationally convenient equal-order interpolation for all the field variables is employed.
- (ii) Due to a careful selection of numerical fluxes, the proposed DG formulation does not suffer from the inherent instabilities that DG methods typically suffer from; for example, the Bassi-Rebay DG method.
- (iii) The stabilization terms inside the domain are of adjoint-type and residual-based, and the corresponding stabilization parameters do not contain any mesh-dependent parameters.

- (iv) The proposed stabilized formulation performs remarkably well, in comparison with its continuous counterpart, in the presence of heterogeneity in material properties. In other words, under the proposed DG formulation no unphysical numerical instabilities are generated at the vicinity of discontinuities in material properties due to Gibbs phenomenon.
- (v) The formulation passes patch tests, even on meshes with non-constant Jacobian elements, in 2D and 3D settings.
- (vi) The proposed DG formulation can support non-conforming discretization in form of non-conforming polynomial orders or non-conforming element refinement, thus allowing efficient h -, p -, and hp -adaptivities.
- (vii) A sensitivity study revealed the importance of η_u and η_p (i.e., jump terms with respect to the normal components of the velocities and pressures, respectively) to reduce the drift along the interior edges for the case of non-conforming polynomial orders.
- (viii) It is shown, theoretically, that the proposed formulation is convergent. The convergence rates obtained under both h - and p -refinement methods in several numerical experiments are in accordance with the theory.
- (ix) It is shown that the proposed DG formulation can be employed to solve coupled flow-transport problems in porous media with double pore-networks. In particular, the effect of heterogeneity of medium properties is studied on the appearance and growth of fingers under viscous-fingering-type instability. The proposed formulation is capable of suppressing the non-physical numerical instabilities (like Gibbs phenomenon and spurious node-to-node oscillations), yet capturing the underlying physical ones.

In chapter 6, we have developed two block solver methodologies which are capable of solving *large-scale* problems under the four-field DPP mathematical model. We subsequently presented a systematic performance analysis of various finite element discretizations for the DPP model using the recently proposed TAS spectrum model, which takes into consideration important metrics such as mesh convergence, static-scaling, and DoE. We have also identified strong-scaling issues one needs to be cognizant of when the block solvers are applied to various finite elements. In our numerical studies, two- and three-dimensional problems had analogous performance trends, despite their marked discrepancy in time to solution.

Some salient features of the proposed composable block solver methodologies are as:

1. Both composable solvers are compatible with different kinds of mixed finite element formulations: H(div) and non-H(div) elements, simplicial and non-simplicial elements, node- and edge-based discretizations, and continuous and discontinuous approximations.
2. Both composable solvers are scalable in both parallel and algorithmic senses.
3. The solvers can be implemented seamlessly using the existing PETSc's composable solver options. Hence, one can leverage on the existing parallel computing tools to implement these composable solvers into existing simulators.

Some main conclusions from the performance analysis using the TAS spectrum model are as:

1. *Scale-split vs. field-split.* For a fixed problem size, the scale-splitting methodology tends to be slightly more efficient in terms of wall-clock time needed despite having the same KSP counts as the field-splitting method. However, selecting either

solver methodologies will be left to the programmer's convenience and limitations as switching from one strategy to another exerts negligible overall effects on performance metrics.

2. *H(div) vs. VMS formulations.* (a) No matter what mesh type is chosen, DoFs are processed the fastest under the H(div) formulation compared to the CG-VMS or DG-VMS formulations. (b) The VMS formulations yield much higher overall numerical accuracy for all velocity and pressure fields than their H(div) counterparts. The exception is for non-simplicial meshes, where the H(div) formulation exhibits superlinear convergence.

In chapter 8, we have considered the flows of incompressible fluids in coupled free-porous media. We have presented a theoretical framework to obtain a complete set of self-consistent conditions, which describes the flow dynamics at the interface of free flow and porous regions. The interface conditions are essential for the closure of the mathematical model. The framework is primarily built upon the principle of virtual power, theory of interacting continua, and a geometric argument for enforcing internal constraints, which in our case is the incompressibility of the fluid. The central idea in the proposed principle of virtual power is to account for the power expended at the interface and thereby making it possible to circumvent the need to estimate the partial stress in the porous solid.

Under the proposed framework, the set of interface conditions is a combination of jump conditions and a constitutive specification, which is provided by prescribing the physically meaningful power expended density at the interface. We have also shown that the jump conditions by themselves do not provide a workable set of conditions, which is because of the inability to quantify the traction taken by the rigid porous solid under the theory of interacting continua. The salient features of the proposed framework of obtaining

interface conditions are: (i) The framework enjoys a strong theoretical underpinning. (ii) The resulting interface conditions make the resulting mathematical model well-posed. Specifically, we have shown that the resulting mathematical model has a unique solution. (iii) The framework is amenable to generalizations, and the resulting interface conditions are valid for a wide variety of porous media models. (iv) Several popular conditions in the literature are special cases of the proposed framework. (v) Similar to uncoupled free flows and uncoupled flows in porous media, the flows in coupled free-porous media under the proposed interface conditions also enjoy a minimum power theorem.

In chapter 9, We have developed a comprehensive modeling framework, consisting of a mathematical model and a computational framework, for studying the elastoplastic mechanical response due to the transport of a chemical species within the host material. We have demonstrated that the proposed computational framework respects mathematical principles, such as maximum principles, and physical constraints, such as the non-negative constraint for the concentration field. We also compared the physics of a degrading elastoplastic material with that of the response of a non-degrading material.

10.1 Future Work

One other possible numerical approach for addressing large-scale geophysical flow models is to employ a method known as hybridization, where the discrete equations are transformed such that classic static condensation and local post-processing methods can be applied. Future endeavors could be focused on developing a hybridized discontinuous Galerkin computational framework for DPP model and comparing its performance with the proposed stabilized mixed discontinuous Galerkin.

On the interface of free flow and porous region, it is possible to construct a weak

formulation based on the minimum power theorem, which can be utilized in a numerical implementation of the mathematical model under the finite element method. Assessing the efficacy of such a weak formulation is worthy of a future investigation. This framework is applicable in numerical modeling of real-life interface problems in geological systems (e.g., hyporheic zone modeling) and biological systems (e.g., arterial mass transfer). Also, the current theoretical framework for deriving interface condition could be extended to derive the fluid-porous-structure interface, using the principle of virtual power and maximization of rate entropy production.

One can extend the work presented on coupled elastoplastic-diffusion systems on three possible fronts. The first one is to incorporate chemical reactions into the model; for example, considering oxidation of the material. The second study can be directed towards understanding and modeling the initiation and propagation of fractures in degrading materials, and phase-field modeling can be a leading candidate for such a study. The third aspect can be on extending the current study to large deformation plasticity models, where interesting phenomena such as metal necking could be studied. These improvements enable application scientists to better understand the coupled behavior of structures under degradation.

References

- [1] VisIt. Developed by Departement of Energy, <https://wci.llnl.gov/codes/visit>.
- [2] R. Abedi, B. Petracovici, and R. B. Haber. A space–time discontinuous Galerkin method for linearized elastodynamics with element-wise momentum balance. *Computer Methods in Applied Mechanics and Engineering*, 195(25–28):3247–3273, 2006.
- [3] M. Ainsworth and B. Senior. Aspects of an adaptive hp -finite element method: Adaptive strategy, conforming approximation and efficient solvers. *Computer Methods in Applied Mechanics and Engineering*, 150:65–87, 1997.
- [4] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCC project version 1.5. *Archive of Numerical Software*, 3:9–23, 2015.
- [5] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40:9, 2014.
- [6] P. R. Amestoy, I. S. Duff, J. Koster, and J. Y. L’Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23:15–41, 2001.
- [7] F. Armero. Elastoplastic and viscoplastic deformations in solids and structures. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–41, 2018.
- [8] D. Arnold, D. D. Boffi, and R. Falk. Approximation by quadrilateral finite elements. *Mathematics of Computation*, 71(239):909–922, 2002.

- [9] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19:742–760, 1982.
- [10] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39:1749–1779, 2002.
- [11] D. N. Arnold, R. Falk, and R. Winther. Finite element exterior calculus: from Hodge theory to numerical stability. *Bulletin of the American mathematical society*, 47(2):281–354, 2010.
- [12] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta numerica*, 15:1–155, 2006.
- [13] D. N. Arnold and A. Logg. Periodic table of the finite elements. *SIAM News*, 47(9):212, 2014.
- [14] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2015.
- [15] I. Babuška and M. R. Dorr. Error estimates for the combined h and p versions of the finite element method. *Numerische Mathematik*, 37:257–277, 1981.
- [16] I. Babuška and T. Strouboulis. *The Finite Element Method and Its Reliability*. Oxford University Press, 2001.
- [17] S. Badia and R. Codina. Stabilized continuous and discontinuous Galerkin techniques for Darcy flow. *Computer Methods in Applied Mechanics and Engineering*, 199:1654–1667, 2010.
- [18] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dal-cin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C.

McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Web page, 2018.

- [19] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.
- [20] R. E. Bank, A. H. Sherman, and A. Weiser. Some refinement algorithms and data structures for regular local mesh refinement. *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences*, 1:3–17, 1983.
- [21] G. I. Barenblatt, I. P. Zheltov, and I. N. Kochina. Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]. *Journal of Applied Mathematics and Mechanics*, 24:1286–1303, 1960.
- [22] T. P. Barrios and R. Bustinzal. An augmented discontinuous Galerkin method for elliptic problems. *Comptes Rendus Mathematique*, 344:53–58, 2007.
- [23] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.
- [24] G. S. Beavers and D. D. Joseph. Boundary conditions at a naturally permeable wall. *Journal of Fluid Mechanics*, 30(1):197–207, 1967.
- [25] G. T. Bercea, A. T. T. McRae, D. A. Ham, L. Mitchell, F. Rathgeber, L. Nardi, F. Luporini, and P. H. J. Kelly. A structure-exploiting numbering algorithm for finite elements on extruded meshes, and its performance evaluation in firedrake. *Geoscientific Model Development*, 9(10):3803–3815, 2016.

- [26] D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Element Methods and Applications*, volume 44. Springer-Verlag, Berlin, 2013.
- [27] C. Boutin and P. Royer. On models of double porosity poroelastic media. *Geophysical Journal International*, 203:1694–1725, 2015.
- [28] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, 2007.
- [29] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, New York, 1994.
- [30] F. Brezzi, J. Douglas, M. Fortin, and L. D. Marini. Efficient rectangular mixed finite elements in two and three space variables. *Mathematical Modelling and Numerical Analysis*, 21:581–604, 1987.
- [31] F. Brezzi, J. Douglas, M. Fortin, and L. D. Marini. Efficient rectangular mixed finite elements in two and three space variables. *ESAIM: Mathematical Modelling and Numerical Analysis*, 21(4):581–604, 1987.
- [32] F. Brezzi, J. Douglas, and L. D. Marini. Two families of mixed elements for second order elliptic problems. *Numerische Mathematik*, 47:217–235, 1985.
- [33] F. Brezzi, J. Douglas, and L. D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2):217–235, 1985.
- [34] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer series in computational mathematics*. Springer-Verlag, New York, 1991.
- [35] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, volume 15. Springer Science & Business Media, 2012.

- [36] F. Brezzi, T. J. R. Hughes, L. D. Marini, and A. Masud. Mixed discontinuous Galerkin method for Darcy flow. *SIAM Journal of Scientific Computing*, 22:119–145, 2005.
- [37] F. Brezzi, L. D. Marini, and E. Süli. Discontinuous Galerkin methods for first-order hyperbolic problems. *Mathematical Models and Methods in Applied Sciences*, 14:1893–1903, 2004.
- [38] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32:199–259, 1982.
- [39] J. Brown, M. G. Knepley, D. A. May, L. C. McInnes, and B. Smith. Composable linear solvers for multiphysics. In *Parallel and Distributed Computing (ISPDC), 2012 11th International Symposium on*, pages 55–62. IEEE, 2012.
- [40] C. Burstedde, O. Ghattas, M. Gurnis, G. Stadler, E. Tan, T. Tu, L. C. Wilcox, and H. Zhong. Scalable adaptive mantle convection simulation on petascale supercomputers. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, page 62. IEEE Press, 2008.
- [41] S. Buss. *The hyporheic handbook: a handbook on the groundwater-surfacewater interface and hyporheic zone for environmental managers*. Hyporheic Network, British Environment Agency, Bristol, U.K., 2009.
- [42] N. Canouet, L. Fezoui, and S. Piperno. Discontinuous Galerkin time-domain solution of Maxwell’s equations on locally-refined nonconforming Cartesian grids. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 24:1381–1401, 2005.

- [43] C. G. Caro, J. M. Fitzgerald, and R. C. Schroter. Atheroma and arterial wall shear; observation, correlation and proposal of a shear dependent mass transfer mechanism for atherogenesis. *Proceedings of the Royal Society of London B*, 177:109–133, 1971.
- [44] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau. An a priori error analysis of the local discontinuous galerkin method for elliptic problems. *SIAM Journal on Numerical Analysis*, 38(5):1676–1706, 2000.
- [45] J. Chang, M. S. Fabien, M. G. Knepley, and R. T. Mills. Comparative study of finite element methods using the Time-Accuracy-Size (TAS) spectrum analysis. *SIAM Journal of Scientific Computing (under revision)*, 2018. Available on arXiv: 1802.07832.
- [46] J. Chang, S. Karra, and K. B. Nakshatrala. Large-scale optimization-based non-negative computational framework for diffusion equations: parallel implementation and performance studies. *Journal of Scientific Computing*, 70:243–271, 2017.
- [47] J. Chang and K. B. Nakshatrala. Variational inequality approach to enforce the non-negative constraint for advection-diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 320:287–334, 2017.
- [48] J. Chang, K. B. Nakshatrala, M. G. Knepley, and L. Johnsson. A performance spectrum for parallel computational frameworks that solve PDEs. *Concurrency and Computation: Practice and Experience*, 30:e4401, 2018.
- [49] Z. Chen, G. Huan, and Y. Ma. *Computational Methods for Multiphase Flows in Porous Media*. Society for Industrial and Applied Mathematics, Philadelphia, 2006.
- [50] J. Choo and R. I. Borja. Stabilized mixed finite elements for deformable porous media with double porosity. *Computer Methods in Applied Mechanics and Engineering*, 293:131–154, 2015.

- [51] J. Choo, J. White, and R. I. Borja. Hydromechanical modeling of unsaturated flow in double porosity media. *International Journal of Geomechanic*, 16:D4016002, 2016.
- [52] B. Cockburn. Discontinuous Galerkin methods. *Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 83:731–754, 2003.
- [53] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365, 2009.
- [54] B. Cockburn, J. Guzmán, and H. Wang. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Mathematics of Computation*, 78:1–24, 2009.
- [55] B. Cockburn, G. Karniadakis, and C. Shu, editors. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer-Verlag, New York, USA, 2000.
- [56] B. Cockburn and C. Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.
- [57] COMSOL Java API . *Reference Guide, Version 4.3*. COMSOL, Inc., Burlington, Massachusetts, www.comsol.com, 2012.
- [58] M. Crouzeix and P.-A. Raviart. Conforming and nonconforming finite element methods for solving the stationary stokes equations i. *Revue française d'automatique informatique recherche opérationnelle. Mathématique*, 7(R3):33–75, 1973.

- [59] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo. Parallel distributed computing using Python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- [60] S. Darbha and K. R. Rajagopal. Unsteady motions of degrading or aging linearized elastic solids. *International Journal of Non-Linear Mechanics*, 44(5):478–485, 2009.
- [61] T. A. Davis. Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, 30:196–199, 2004.
- [62] V. Dolejší and M. Feistauer. *Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow*. Springer, 2015.
- [63] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic Galerkin methods. *Computing Methods in Applied Sciences*, pages 207–216, 1976.
- [64] P. G. Drazin. *Introduction to Hydrodynamic Stability*. Cambridge University Press, Cambridge, U.K., 2002.
- [65] F. Dunne and N. Petrinic. *Introduction to Computational Plasticity*. Oxford University Press, 2005.
- [66] R. C. Dykhuizen. A new coupling term for dual-porosity models. *Water Resources Research*, 26(02):351–356, 1990.
- [67] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, Rhode Island, 1998.
- [68] R. D. Falgout and U. M. Yang. HYPRE: A library of high performance preconditioners. In *International Conference on Computational Science*, pages 632–641. Springer, 2002.

- [69] C. A. Felippa, K. C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24–25):3247–3270, 2001.
- [70] T. P. Fries, A. Byfut, A. Alizada, K. W. Cheng, and A. Schröder. Hanging nodes and XFEM. *International Journal for Numerical Methods in Engineering*, 86:404–430, 2011.
- [71] C. Geuzaine and J. F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.
- [72] T. H. Gibson, L. Mitchell, D. A. Ham, and C. J. Cotter. A domain-specific language for the hybridization and static condensation of finite element methods. *Available on arXiv: 1802.00303*, 2018.
- [73] D. Gilbarg and N. S. Trudinger. *Elliptic partial differential equations of second order*. Springer, 2015.
- [74] P. E. Gill, W. Murray, and M. H. Wright. *Numerical Linear Algebra and Optimization*, volume 1. Addison-Wesley Redwood City, CA, 1991.
- [75] R. Glowinski. *Lectures on Numerical Methods for Non-linear Variational Problems*. Springer Science & Business Media, 2008.
- [76] E. Guazzelli and J. F. Morris. *A Physical Introduction to Suspension Dynamics*, volume 45. Cambridge University Press, Cambridge, U.K., 2011.
- [77] A. K. Gupta. A finite element for transition from a fine to a coarse grid. *International Journal for Numerical Methods in Engineering*, 12:35–45, 1978.

- [78] S. Güzey, B. Cockburn, and H. K. Stolarski. The embedded discontinuous galerkin method: application to linear shell problems. *International Journal for Numerical Methods in Engineering*, 70(7):757–790, 2007.
- [79] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183:508–532, 2002.
- [80] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing '95: Proceedings of the 1995 ACM/IEEE Conference on Supercomputing (CDROM)*, page 28, New York, 1995. ACM Press.
- [81] J. S. Hesthaven and T. Warburton. High-order nodal discontinuous Galerkin methods for the Maxwell eigenvalue problem. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 362:493–524, 2004.
- [82] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Science & Business Media, New York, 2007.
- [83] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Science & Business Media, 2007.
- [84] H. F. Hoff, C. K. Heideman, R. L. Jackson, R. J. Bayardo, S. K. H, and A. M. Gotto. Localization patterns of plasma apolipoproteins in human atherosclerotic lesions. *Circulation Research*, 37:72–79, 1975.
- [85] M. Homolya and D. A. Ham. A parallel edge orientation algorithm for quadrilateral meshes. *SIAM Journal on Scientific Computing*, 38(5):48–61, 2016.

- [86] M. Homolya, R. C. Kirby, and D. A. Ham. Exposing and exploiting structure: optimal code generation for high-order finite element methods. *Available on arXiv: 1711.02473*, 2017.
- [87] M. Homolya, L. Mitchell, F. Luporini, and D. A. Ham. Tsfc: a structure-preserving form compiler. *SIAM Journal on Scientific Computing*, 40(3):C401–C428, 2018.
- [88] G. M. Homsy. Viscous fingering in porous media. *Annual Review of Fluid Mechanics*, 19:271–311, 1987.
- [89] E. Hopf. *Elementare Bemerkungen über die Lösungen Partieller Differentialgleichungen zweiter Ordnung vom elliptischen Typus*. 1927.
- [90] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Inc., New York, 2012.
- [91] T. J. R. Hughes, G. Engel, L. Mazzei, and M. G. Larson. The continuous Galerkin method is locally conservative. *Journal of Computational Physics*, 163:467–488, 2000.
- [92] T. J. R. Hughes, A. Masud, and J. Wan. A stabilized mixed discontinuous Galerkin method for Darcy flow. *Computer Methods in Applied Mechanics and Engineering*, 195:3347–3381, 2006.
- [93] T. J. R. Hughes, A. Masud, and J. Wan. A stabilized mixed discontinuous Galerkin method for Darcy flow. *Computer Methods in Applied Mechanics and Engineering*, 195(25-28):3347–3381, 2006.
- [94] K. Ikeda, Y. Yamakawa, and S. Tsutsumi. Simulation and interpretation of diffuse mode bifurcation of elastoplastic solids. *Journal of the Mechanics and Physics of Solids*, 51(9):1649–1673, 2003.

- [95] J. B. Jones and R. M. Holmes. Surface-subsurface interactions in stream ecosystems. *Trends in Ecology & Evolution*, 11(6):239–242, 1996.
- [96] S. H. S. Joodat, K. B. Nakshatrala, and R. Ballarini. Modeling flow in porous media with double porosity/permeability: A stabilized mixed formulation, error analysis, and numerical solutions. *Computer Methods in Applied Mechanics and Engineering*, 337:632–676, 2018.
- [97] S. H. S. Joodat, K. B. Nakshatrala, and R. Ballarini. Modeling flow in porous media with double porosity/permeability: A stabilized mixed formulation, error analysis, and numerical solutions. *Computer Methods in Applied Mechanics and Engineering*, 337:632–676, 2018.
- [98] M. S. Joshaghani, S. H. S. Joodat, and K. B. Nakshatrala. A stabilized mixed discontinuous Galerkin formulation for double porosity/permeability model. Available on arXiv: 1805.01389, 2018.
- [99] G. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, New York, 2013.
- [100] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, and others. Multiphysics simulations: Challenges and opportunities. *The International Journal of High Performance Computing Applications*, 27(1):4–83, 2013.
- [101] J. H. Kim, A. Serpantié, F. Barlat, F. Pierron, and M. G. Lee. Characterization of the post-necking strain hardening behavior using the virtual fields method. *International Journal of Solids and Structures*, 50(24):3829–3842, 2013.
- [102] N. H. Kim. *Introduction to Nonlinear Finite Element Analysis*. Springer Science & Business Media, 2014.

- [103] M. G. Knepley and D. A. Karpeev. Mesh algorithms for PDE with Sieve I: Mesh distribution. *Scientific Programming*, 17(3):215–230, 2009.
- [104] M. A. Kopera and G. X. Giraldo. Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *Journal of Computational Physics*, 275:92–117, 2014.
- [105] H. Kotake, R. Matsumoto, S. Taketomi, and N. Miyazaki. Transient hydrogen diffusion analyses coupled with crack-tip plasticity under cyclic loading. *International Journal of Pressure Vessels and Piping*, 85(8):540–549, 2008.
- [106] N. V. Krylov. *Lectures on Elliptic and Parabolic Equations in Sobolev Spaces*, volume 96. American Mathematical Society., 2008.
- [107] D. V. Kulkarni, D. V. Rovas, and D. A. Tortorelli. Discontinuous Galerkin framework for adaptive solution of parabolic problems. *International Journal for Numerical Methods in Engineering*, 70:1–24, 2007.
- [108] M. Lange, M. G. Knepley, and G. J. Gorman. Flexible, scalable mesh and data management using PETSc DMplex. In *Proceedings of the Exascale Applications and Software Conference*, April 2015.
- [109] M. Lange, L. Mitchell, M. G. Knepley, and G. J. Gorman. Efficient mesh management in Firedrake using PETSc-DMplex. *SIAM Journal on Scientific Computing*, 38(5):S143–S155, 2016.
- [110] R. E. Larson and J. J. L. Higdon. Microscopic flow near the surface of two-dimensional porous media. Part 2. Transverse flow. *Journal of Fluid Mechanics*, 178:119–136, 1987.

- [111] J. Li and B. Rivière. Numerical solutions of the incompressible miscible displacement equations in heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 292:107–121, 2015.
- [112] J. Li and B. Rivière. Numerical modeling of miscible viscous fingering instabilities by high-order methods. *Transport in Porous Media*, 113:607–628, 2016.
- [113] J. L. Lions. Problèmes aux limites non homogènes et donées irrégulières: Une méthode d'approximation. *Numerical Analysis of Partial Differential Equations (CIME 2 Ciclo, Ispra, 1967)*, Edizioni Cremonese, Rome, pages 283–292, 1968.
- [114] R. Liska and M. Shashkov. Enforcing the discrete maximum principle for linear finite element solutions of second-order elliptic problems. *Communications in Computational Physics*, 3:852–877, 2008.
- [115] Q. Liu and A. Prosperetti. Pressure-driven flow in a channel with porous walls. *Journal of Fluid Mechanics*, 679:77–100, 2011.
- [116] LiveLink for MATLAB . *User's Guide, Version 5.1*. COMSOL, Inc., Burlington, Massachusetts, www.comsol.com, 2012.
- [117] A. Logg, K. A. Mardal, and G. N. Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.
- [118] A. Logg, K. A. Mardal, G. N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.
- [119] A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)*, 37:20, 2010.

- [120] A. Logg, G. N. Wells, and J. Hake. DOLFIN: A C++/Python finite element library. In *Automated Solution of Differential Equations by the Finite Element Method*, pages 173–225. Springer, 2012.
- [121] F. Luporini, D. A. Ham, and P. H. J. Kelly. An algorithm for the optimization of finite element integration loops. *ACM Transactions on Mathematical Software*, 44:3:1–3:26, 2017.
- [122] F. Luporini, A. L. Varbanescu, F. Rathgeber, G. T. Bercea, J. Ramanujam, D. A. Ham, and P. H. J. Kelly. Cross-loop optimization of arithmetic intensity for finite element local assembly. *ACM Transactions on Architecture and Code Optimization*, 11(4):57:1–57:25, 2015.
- [123] F. Luporini, A. L. Varbanescu, F. Rathgeber, G. T. Bercea, J. Ramanujam, D. A. Ham, and P. H. J. Kelly. Cross-loop optimization of arithmetic intensity for finite element local assembly. *ACM Transactions on Architecture and Code Optimization (TACO)*, 11:57, 2015.
- [124] N. K. Mapakshi, J. Chang, and K. B. Nakshatrala. A scalable variational inequality approach for flow through porous media models with pressure-dependent viscosity. *Journal of Computational Physics*, 359:137–163, 2018.
- [125] B. Markert. *Weak or Strong: on Coupled Problems in Continuum Mechanics*. 2010.
- [126] MATLAB. *version 7.10.0 (R2016a)*. The MathWorks Inc., Natick, Massachusetts, 2016.
- [127] J. McMurry, D. S. Ballantine, C. A. Hoeger, V. E. Peterson, and M. Castellion. *Fundamentals of General, Organic, and Biological Chemistry*. Pearson Education, 2010.

- [128] A. T. T. McRae, G. T. Bercea, L. Mitchell, D. A. Ham, and C. J. Cotter. Automated generation and symbolic manipulation of tensor product finite elements. *SIAM Journal on Scientific Computing*, 38(5):25–47, 2016.
- [129] R. Miresmaeili, M. Ogino, T. Nakagawa, and H. Kanayama. A coupled elastoplastic-transient hydrogen diffusion analysis to simulate the onset of necking in tension by using the finite element method. *International Journal of Hydrogen Energy*, 35(3):1506–1514, 2010.
- [130] S. Mitchell, N. L. Michels, K. Kunze, and J. Pérez-Ramírez. Visualization of hierarchically structured zeolite bodies from macro to nano length scales. *Nature Chemistry*, 4(10):825, 2012.
- [131] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.
- [132] D. J. Morton, J. M. Tyler, and J. R. Dorroh. A new 3D finite element for adaptive h -refinement in 1-irregular meshes. *International Journal for Numerical Methods in Engineering*, 38:3989–4008, 1995.
- [133] M. K. Mudunuru and K. B. Nakshatrala. A framework for coupled deformation–diffusion analysis with application to degradation/healing. *International Journal for Numerical Methods in Engineering*, 89(9):1144–1170, 2012.
- [134] H. Nagarajan and K. B. Nakshatrala. Enforcing the non-negativity constraint and maximum principles for diffusion with decay on general computational grids. *International Journal for Numerical Methods in Fluids*, 67:820–847, 2011.
- [135] K. B. Nakshatrala, S. H. S. Joodat, and R. Ballarini. Modeling flow in porous media with double porosity/permeability: Mathematical model, properties, and analytical solutions. *Journal of Applied Mechanics*, 2018. Available on arXiv:1605.07658.

- [136] K. B. Nakshatrala and A. J. Valocchi. Non-negative mixed finite element formulations for a tensorial diffusion equation. *Journal of Computational Physics*, 228:6726–6752, 2009.
- [137] P. B. Nakshatrala, K. B. Nakshatrala, and D. A. Tortorelli. A time-staggered partitioned coupling algorithm for transient heat conduction. *International Journal for Numerical Methods in Engineering*, 78(12):1387–1406, 2009.
- [138] J.-C. Nédélec. Mixed finite elements in \mathbb{R}^3 . *Numerische Mathematik*, 35(3):315–341, 1980.
- [139] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.
- [140] W. L. Oberkampf and C. J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.
- [141] J. T. Oden, L. Demkowicz, W. Rachowicz, and T. A. Westermann. Toward a universal hp adaptive finite element strategy, Part 2. A posteriori error estimation. *Computer Methods in Applied Mechanics and Engineering*, 77(1-2):113–180, 1989.
- [142] R. K. Pal, R. Abedi, A. Madhukar, and R. B. Haber. Adaptive spacetime discontinuous Galerkin method for hyperbolic advection–diffusion with a non-negativity constraint. *International Journal for Numerical Methods in Engineering*, 105:963–989, 2016.
- [143] J. Palaniappan, R. B. Haber, and R. L. Jerrard. A spacetime discontinuous Galerkin method for scalar conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 193(33-35):3607–3631, 2004.

- [144] J. Peraire and P. O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.
- [145] D. A. D. Pietro and A. Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*, volume 69. Springer Science & Business Media, 2011.
- [146] quadratic programming package. <https://www.mathworks.com/help/optim/ug/quadprog.html>, 2018.
- [147] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G. T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):24, 2016.
- [148] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G. T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43:24, 2017.
- [149] P. A. Raviart and J. M. Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical Aspects of the Finite Element Method*, pages 292–315, Springer-Verlag, New York, 1977.
- [150] P. A. Raviart and J. M. Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical aspects of finite element methods*, pages 292–315. Springer, 1977.
- [151] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479*, 1973.

- [152] J. F. Remacle, J. E. Flaherty, and M. S. Shephard. An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review*, 45:53–72, 2003.
- [153] B. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*. SIAM, 2008.
- [154] B. Rivière and M. F. Wheeler. Discontinuous Galerkin methods for flow and transport problems in porous media. *International Journal for Numerical Methods in Biomedical Engineering*, 18:63–68, 2002.
- [155] B. Rivière, M. F. Wheeler, and V. Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. Part I. *Computational Geosciences*, 3:337–360, 1999.
- [156] T. Rusten, P. Vassilevski, and R. Winther. Interior penalty preconditioners for mixed finite element approximations of elliptic problems. *Mathematics of Computation of the American Mathematical Society*, 65:447–466, 1996.
- [157] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, USA, 2003.
- [158] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [159] P. G. Saffman. On the boundary condition at the surface of a porous medium. *Studies in Applied Mathematics*, 50(2):93–101, 1971.
- [160] P. G. Saffman and G. Taylor. The penetration of a fluid into a porous medium or

Hele-Shaw cell containing a more viscous liquid. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 245:312–329, 1958.

- [161] M. Sahraoui and M. Kaviany. Slip and no-slip velocity boundary conditions at interface of porous, plain media. *International Journal of Heat and Mass Transfer*, 35(4):927–943, 1992.
- [162] D. Sasaki, M. Koyama, K. Higashida, K. Tsuzaki, and H. Noguchi. Effects of hydrogen-altered yielding and work hardening on plastic-zone evolution: A finite-element analysis. *International Journal of Hydrogen Energy*, 40(31):9825–9837, 2015.
- [163] M. Shabouei and K. B. Nakshatrala. Mechanics-based solution verification for porous media models. *Communications in Computational Physics*, 20(5):1127–1162, 2016.
- [164] J. C. Simo and H. J. Hughes. *Computational Inelasticity*, volume 7. Springer Science & Business Media, 2006.
- [165] G. F. Smith. On isotropic functions of symmetric tensors, skew-symmetric tensors and vectors. *International Journal of Engineering Science*, 9(10):899–916, 1971.
- [166] M. Sophocleous. Interactions between groundwater and surface water: the state of the science. *Hydrogeology Journal*, 10(1):52–67, 2002.
- [167] B. Straughan. *Mathematical Aspects of Multi-porosity Continua*. Springer, Cham, Switzerland, 2017.
- [168] N. Sun, N. B. Wood, A. D. Hughes, A. S. Thom, and X. Y. Xu. Fluid-wall modelling of mass transfer in an axisymmetric stenosis: effects of shear-dependent transport properties. *Annals of Biomedical Engineering*, 34(7):1119–1128, 2006.

- [169] H. W. Swift. Plastic instability under plane stress. *Journal of the Mechanics and Physics of Solids*, 1(1):1–18, 1952.
- [170] C. W. Thomas, J. W. Harvey, O. L. Franke, and W. A. Alley. *Ground water and surface water: a single resource*. United States Geological Survey, Circular 1139, Denver, Colorado, 1998.
- [171] D. Z. Turner, K. B. Nakshatrala, M. J. Martinez, and P. K. Notz. Modeling subsurface water resource systems involving heterogeneous porous media using the variational multiscale formulation. *Journal of Hydrology*, 428:1–14, 2012.
- [172] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996.
- [173] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Oxford Science Publications, New Jersey, 2013.
- [174] S. Wada and T. Karino. Theoretical study on flow-dependent concentration polarization of low density lipoproteins at the luminal surface of a straight artery. *Biorheology*, 36:207–223, 1999.
- [175] J. E. Warren and P. J. Root. The behavior of naturally fractured reservoirs. *Society of Petroleum Engineering Journal*, 3:245–255, 1963.
- [176] N. Yang and K. Vafai. Modeling of low-density lipoprotein (ldl) transport in the artery—effects of hypertension. *International Journal of Heat and Mass Transfer*, 49(5-6):850–867, 2006.
- [177] Zenodo/COFFEE. COFFEE: a compiler for fast expression evaluation, nov 2017.
- [178] Zenodo/FIAT. FIAT: the finite element automated tabulator, apr 2018.

- [179] Zenodo/FInAT. FInAT: a smarter library of finite elements, jan 2018.
- [180] ZENODO/firedrake. Firedrake: an automated finite element system, may 2018.
- [181] Zenodo/PETSc. PETSc: Portable, extensible toolkit for scientific computation, apr 2018.
- [182] Zenodo/petsc4py. petsc4py: The python interface to PETSc, apr 2018.
- [183] Zenodo/PyOP2. PyOP2: framework for performance-portable parallel computations on unstructured meshes, may 2018.
- [184] Zenodo/TSFC. TSFC: the two stage form compiler, may 2018.
- [185] Zenodo/UFL. UFL: the unified form language, apr 2018.
- [186] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method: Solid Mechanics*, volume 2. Butterworth-Heinemann, 2000.

Appendices

A Computer Implementation

The numerical results pertaining to the non-conforming discretization (Section 5.1.2) and non-constant Jacobian elements (Section 5.1.3), have been obtained using COMSOL Java API [57]. The numerical simulations for the 2D numerical convergence analysis on QUAD and trapezoidal meshes, and 3D numerical convergence analysis (Section 5.2.2); and the coupled problem (Section 5.4) were carried out using the Firedrake Project [148, 123]. All the remaining numerical results were generated using the FEniCS Project [117, 4].

The FEniCS and Firedrake Projects are built upon several scientific packages and provide automated frameworks to solve partial differential equations in serial and parallel environments. Both provide an easy-to-use Python-based interface to develop computer codes, to access the scientific packages on which they are built upon, and to generate the output in various formats which are compatible with popular visualization software packages such as **ParaView** [14] and **VisIt** [1]. Under both these projects, mesh generation can be performed either within the code or using the third party mesh generators such as **GMSH** [71].

Among the various components available in FEniCS, we have used the Unified Form Language (UFL) [5] and the DOLFIN library [119, 120] in our implementations. The former enables the user to declare the finite element discretization of variational forms and the latter is used for the automated assembly of the finite element discrete formulations. The Firedrake Project employs the UFL from the FEniCS Project. However, the main difference between the FEniCS and Firedrake Projects is that all data structures, linear solvers and non-linear solvers for the former are provided by DOLFIN library and for the latter are provided entirely by the PETSc library [19]. Another notable difference is that the FEniCS Project offers only simplicial element (e.g., triangular and tetrahedron

elements), whereas the Firedrake Project offers non-simplicial elements in addition to the simplicial ones.

In our numerical simulations, MUMPS [6] direct solver and the sparse LU decomposition direct solver from the UMFPACK [61] were, respectively, employed with default settings under the COMSOL Java API and the FEniCS Project. The GMRES iterative solver with “bjacobi” preconditioner and the relative convergence tolerance of 10^{-7} was employed under the Firedrake Project.

Below, we have provided a Firedrake-based computer code, which can be used to generate the results for the coupled problem, which is discussed in Section 5.4.

Listing 1: Firedrake code for solving the coupled problem in the heterogeneous porous medium

```

1  from firedrake import *
2  import numpy
3  import random
4  try:
5      import matplotlib.pyplot as plt
6  except:
7      warning("Matplotlib not imported")
8
9  #== Create mesh ==
10 nx, ny = 100, 40
11 Lx, Ly = 1.0, 0.4
12 mesh = RectangleMesh(nx,ny,Lx,Ly)
13
14 #== Function spaces ==
15 #—Double porosity/permeability flow problem—
16 velSpace = VectorFunctionSpace(mesh, "DG", 2)
17 pSpace = FunctionSpace(mesh, "DG", 2)
18 wSpace = MixedFunctionSpace([velSpace,pSpace,velSpace,pSpace])
19
20 #—Advection-diffusion problem—

```

```

21 uSpace = FunctionSpace(mesh, "CG",1)
22
23 #——Permeability——
24 kSpace = FunctionSpace(mesh, "DG", 0)
25
26 == Material properties and parameters ==
27 mu0, Rc, D = Constant(1e-3), Constant(3.0), Constant(2e-6)
28 k1_0 = 1.1
29 k1_1 = 0.9
30 tol = 1E-14
31
32 class myk1(Expression): #Macro-permeability
33     def eval(self, values, x):
34         if x[1] < Ly/2 + tol:
35             values[0] = k1_0
36         else:
37             values[0] = k1_1
38
39 k1 = interpolate(myk1(),kSpace)
40
41 k2_0 = 0.01 * 1.1
42 k2_1 = 0.01 * 0.9
43
44 class myk2(Expression): #Micro-permeability
45     def eval(self, values, x):
46         if x[1] < Ly/2 + tol:
47             values[0] = k2_0
48         else:
49             values[0] = k2_1
50
51 k2 = interpolate(myk2(),kSpace)
52
53 #——Drag coefficients——
54
55 def alphal(c):
56     return mu0 * exp(Rc * (1.0 - c))/k1
57
58 def invalphal(c):
59     return 1/alphal(c)

```

```

60
61 def alpha2(c):
62     return mu0 * exp(Rc * (1.0 - c))/k2
63
64 def invalpha2(c):
65     return 1/alpha2(c)
66
67 == Boundary and initial conditions ==
68 v_topbottom = Constant(0.0)
69 p_L = Constant(10.0)
70 p_R = Constant(1.0)
71 c_inj = Constant(1.0)
72
73 == Perturbation function for initial concentration ==
74 #—Needed to trigger the instability—
75 class c_0(Expression):
76     def eval(self, values, x):
77         if x[0] < 0.010*Lx:
78             values[0] = abs(.10*exp(-x[0]*x[0]) * random.random())
79         else:
80             values[0] = 0.0
81
82 == Define trial and test functions ==
83 #—DPP flow problem—
84 (v1,p1,v2,p2) = TrialFunctions(wSpace)
85 (w1,q1,w2,q2) = TestFunctions(wSpace)
86 DPP_solution = Function(wSpace)
87
88 #—AD problem—
89 c1 = TrialFunction(uSpace)
90 u = TestFunction(uSpace)
91 conc = Function(uSpace)
92 conc_k = interpolate(c_0(),uSpace)
93
94 == Time parameters ==
95 T = 0.0015 # Total simulation time
96 dt = 0.00005 # Time step
97
98 == Boundary conditions ==

```

```

99  #——DPP velocity BCs——
100 bcDPP = []
101
102 #——AD concentration BCs——
103 bcleft_c = DirichletBC(uSpace,c_inj,1,method = "geometric")
104
105 bcAD = [bcleft_c]
106
107 == Define source terms ==
108 #——DPP model——
109 rhob1, rhob2 = Constant((0.0,0.0)), Constant((0.0,0.0))
110
111 #——AD problem——
112 f = Constant(0.0)
113
114 == Normal vectors and mesh size ==
115 n = FacetNormal(mesh)
116 h = CellSize(mesh)
117 h_avg = (h('+') + h('-'))/2
118
119 == Penalty parameters ==
120 eta_p, eta_u = Constant(0.0), Constant(0.0)
121
122 == Define variational forms ==
123
124 #——DPP stabilized mixed DG formulation——
125 aDPP = dot(w1, alphal(conc_k) * v1) * dx + \
126     dot(w2, alpha2(conc_k) * v2) * dx - \
127     div(w1) * p1 * dx - \
128     div(w2) * p2 * dx + \
129     q1 * div(v1) * dx + \
130     q2 * div(v2) * dx + \
131     q1 * (p1 - p2) * dx - \
132     q2 * (p1 - p2) * dx + \
133     jump(w1,n) * avg(p1) * dS + \
134     jump(w2,n) * avg(p2) * dS - \
135     avg(q1) * jump(v1,n) * dS - \
136     avg(q2) * jump(v2,n) * dS + \
137     dot(w1,n) * p1 * ds(3) +

```

```

138      dot(w2,n) * p2 * ds(3) - \
139      q1 * dot(v1,n) * ds(3) - \
140      q2 * dot(v2,n) * ds(3) + \
141      dot(w1,n) * p1 * ds(4) + \
142      dot(w2,n) * p2 * ds(4) - \
143      q1 * dot(v1,n) * ds(4) - \
144      q2 * dot(v2,n) * ds(4) - \
145      0.5 * dot( alpha1(conc_k) * w1 - grad(q1), \
146                  invalpha1(conc_k) * (alpha1(conc_k) * v1 + grad(p1)) ) * dx - \
147      0.5 * dot( alpha2(conc_k) * w2 - grad(q2), \
148                  invalpha2(conc_k) * (alpha2(conc_k) * v2 + grad(p2)) ) * dx + \
149      (eta_u * h_avg) * avg(alpha1(conc_k)) * (jump(v1,n) * jump(w1,n)) * dS + \
150      (eta_u * h_avg) * avg(alpha2(conc_k)) * (jump(v2,n) * jump(w2,n)) * dS + \
151      (eta_p / h_avg) * avg(1 / alpha1(conc_k)) * dot(jump(q1,n),jump(p1,n)) * dS + \
152      (eta_p / h_avg) * avg(1 / alpha2(conc_k)) * dot(jump(q2,n),jump(p2,n)) * dS
153
154 LDPP = dot(w1,rhob1) * dx + \
155      dot(w2,rhob2) * dx - \
156      dot(w1,n) * p_L * ds(1) - \
157      dot(w2,n) * p_L * ds(1) - \
158      dot(w1,n) * p_R * ds(2) - \
159      dot(w2,n) * p_R * ds(2) - \
160      0.5 * dot( alpha1(conc_k) * w1 - grad(q1), \
161                  invalpha1(conc_k) * rhob1 ) * dx - \
162      0.5 * dot( alpha2(conc_k) * w2 - grad(q2), \
163                  invalpha2(conc_k) * rhob2 ) * dx
164
165
166 #—AD formulation with SUPG Stabilization—
167 vnorm = sqrt(dot((DPP_solution.sub(0)+DPP_solution.sub(2)), \
168                 (DPP_solution.sub(0)+DPP_solution.sub(2))))
169
170 taw = h/(2*vnorm)*dot((DPP_solution.sub(0)+DPP_solution.sub(2)), \
171                         grad(u))
172
173 a_r = taw*(c1 + dt*(dot((DPP_solution.sub(0)+DPP_solution.sub(2)), \
174                         grad(c1)) - div(D*grad(c1))))*dx
175
176 L_r = taw*(conc_k + dt*f)*dx

```

```

177
178 #——Weak form (GL + SUPG)——
179 aAD = a_r + u*c1*dx + dt*(u*dot((DPP_solution.sub(0)+DPP_solution.sub(2)), \
180                                     grad(c1))*dx + dot(grad(u),D*grad(c1))*dx)
181
182 LAD = L_r + u*conc_k*dx + dt*u*f*dx
183
184 #——Create files for storing solution——
185 cfile = File("Concentration.pvd")
186 v1file = File("Macro_Velocity.pvd")
187 p1file = File("Macro_Pressure.pvd")
188 v2file = File("Micro_Velocity.pvd")
189 p2file = File("Micro_Pressure.pvd")
190
191 #== Solver for flow problem ==
192 solver_parameters = { # Default solver — medium sized problems
193     'ksp_type': 'gmres',
194     'pc_type': 'bjacobi',
195     'mat_type': 'aij',
196     'ksp_rtol': 1e-7,
197     'ksp_monitor': True
198 }
199
200 problem_flow = LinearVariationalProblem(aDPP, LDPP, DPP_solution, bcs=bcDPP,
201                                         constant_jacobian=False)
202 solver_flow = LinearVariationalSolver(problem_flow, options_prefix="flow_",
203                                         solver_parameters=solver_parameters)
204
205 #== March the solution over time ==
206 t = dt
207 while t <= T:
208     print '====='
209     print '          time = ', t
210     print '====='
211     c_0.t = t
212
213 #——Compute DPP model——
214 solver_flow.solve()
215

```

```

216     #——Compute AD problem——
217     solve(aAD == LAD,conc,bcs=bcAD)
218     conc_k.assign(conc)    # update for next iteration
219
220     #——Dump solutions for each time step——
221     cfile.write(conc, time = t)
222     v1file.write(DPP_solution.sub(0), time = t)
223     p1file.write(DPP_solution.sub(1), time = t)
224     v2file.write(DPP_solution.sub(2), time = t)
225     p2file.write(DPP_solution.sub(3), time = t)
226     t += dt
227
228 print "total time = ", t
229
230 vlsol, plsol, v2sol, p2sol = DPP_solution.split()
231
232 #== Dump solution fields to file in VTK format ==
233 file = File("Concentration.pvd")
234 file.write(conc)
235
236 file = File('Macro.Velocity.pvd')
237 file.write(vlsol)
238
239 file = File('Macro.Pressure.pvd')
240 file.write(plsol)
241
242 file = File('Micro.Velocity.pvd')
243 file.write(v2sol)
244
245 file = File('Micro.Pressure.pvd')
246 file.write(p2sol)

```

B PETSc Command-line Options

In both the listing provided in this appendix, we have assumed that the global ordering of the mixed function space is macro-scale velocity (0), macro-scale pressure (1), micro-scale velocity (2), and micro-scale pressure (3).

Listing 2: PETSc command-line options for splitting by fields

```
1   -ksp_type gmres
2   -pc_type fieldsplit
3   -pc_fieldsplit_0_fields 0,1
4   -pc_fieldsplit_1_fields 2,3
5   -pc_fieldsplit_type additive
6   -fieldsplit_0_ksp_type preonly
7   -fieldsplit_0_pc_type fieldsplit
8   -fieldsplit_0_pc_fieldsplit_type schur
9   -fieldsplit_0_pc_fieldsplit_schur_fact_type full
10  -fieldsplit_0_pc_fieldsplit_schur_precondition selfp
11  -fieldsplit_0_fieldsplit_0_ksp_type preonly
12  -fieldsplit_0_fieldsplit_0_pc_type bjacobi
13  -fieldsplit_0_fieldsplit_1_ksp_type preonly
14  -fieldsplit_0_fieldsplit_1_pc_type hypre
15  -fieldsplit_1_ksp_type preonly
16  -fieldsplit_1_pc_type fieldsplit
17  -fieldsplit_1_pc_fieldsplit_type schur
18  -fieldsplit_1_pc_fieldsplit_schur_fact_type full
19  -fieldsplit_1_pc_fieldsplit_schur_precondition selfp
20  -fieldsplit_1_fieldsplit_0_ksp_type preonly
21  -fieldsplit_1_fieldsplit_0_pc_type bjacobi
22  -fieldsplit_1_fieldsplit_1_ksp_type preonly
23  -fieldsplit_1_fieldsplit_1_pc_type hypre
```

Listing 3: PETSc command-line options for splitting by scale

```
1      -ksp_type gmres
2      -pc_type fieldsplit
3      -pc_fieldsplit_0_fields 0,2
4      -pc_fieldsplit_1_fields 1,3
5      -pc_fieldsplit_type schur
6      -pc_fieldsplit_schur_fact_type full
7      -pc_fieldsplit_schur_precondition selfp
8      -fieldsplit_0_ksp_type preonly
9      -fieldsplit_0_pc_type bjacobi
10     -fieldsplit_1_ksp_type preonly
11     -fieldsplit_1_pc_type fieldsplit
12     -fieldsplit_1_pc_fieldsplit_type additive
13     -fieldsplit_1_fieldsplit_0_ksp_type preonly
14     -fieldsplit_1_fieldsplit_0_pc_type hypre
15     -fieldsplit_1_fieldsplit_1_ksp_type preonly
16     -fieldsplit_1_fieldsplit_1_pc_type hypre
```

C Spectral Performance Codes

In the following, we have provided Firedrake-based computer codes for $H(\text{div})$ formulation (listing 4), CG-VMS formulation (listing 5) and DG-VMS formulation (listing 6). These three formulations were discussed earlier in Chapter 3.

Listing 4: Firedrake code for 3D problem with TET mesh using $H(\text{div})$ formulation

```

1 from firedrake import *
2 import numpy as np
3
4 #== Create mesh ==
5 mesh = BoxMesh(5,5,5,1,1,1)
6
7 #== Function spaces ==
8 vSpace = FunctionSpace(mesh, "RT",1)
9 pSpace = FunctionSpace(mesh, "DG",0)
10 wSpace = MixedFunctionSpace([vSpace,pSpace,vSpace,pSpace])
11
12 #== Define trial and test functions ==
13 (v1,p1,v2,p2) = TrialFunctions(wSpace)
14 (w1,q1,w2,q2) = TestFunctions(wSpace)
15
16 #== Parameters and material properties ==
17 rhob1, rhob2 = Constant((0.0,0.0,0.0)), Constant((0.0,0.0,0.0))
18 mu = Constant(1.0)
19 beta = Constant(1.0)
20 fact = 1.0
21 k1, k2 = Constant(1.0), Constant(0.1)
22 alphal, alpha2 = Constant(mu/k1), Constant(mu/k2)
23 eta = np.sqrt(1.0 *(1.0+0.1)/(1.0 * 0.1))
24
25 #== Boundary conditions ==
26 p1.left = interpolate(Expression("(1/pi)*(sin(pi*x[1])+sin(pi*x[2])) -\\
27 (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
28 p1.right = interpolate(Expression("(1/pi)*exp(pi)*(sin(pi*x[1])+sin(pi*x[2])) -\\

```

```

29             (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
30 p1_bottom = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) -\
31                         (1.0 + exp(eta*x[2]))",eta=eta) , pSpace)
32 p1_top = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) -\
33                         (exp(eta) + exp(eta*x[2]))",eta=eta) , pSpace)
34 p1_back = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) -\
35                         (exp(eta*x[1]) + 1.0)",eta=eta) , pSpace)
36 p1_front = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) -\
37                         (exp(eta) + exp(eta*x[1]))",eta=eta) , pSpace)
38 p2_left = interpolate(Expression("(1/pi)*(sin(pi*x[1])+sin(pi*x[2])) +\
39                         10.0 * (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
40 p2_right = interpolate(Expression("(1/pi)*exp(pi)*(sin(pi*x[1])+sin(pi*x[2])) +\
41                         10.0 * (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
42 p2_bottom = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) +\
43                         10.0 * (1.0 + exp(eta*x[2]))",eta=eta) , pSpace)
44 p2_top = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) +\
45                         10.0 * (exp(eta) + exp(eta*x[2]))",eta=eta) , pSpace)
46 p2_back = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) +\
47                         10.0 * (exp(eta*x[1]) + 1.0)",eta=eta) , pSpace)
48 p2_front = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) +\
49                         10.0 * (exp(eta) + exp(eta*x[1]))",eta=eta) , pSpace)
50 bcs = []
51
52 ## Normal vectors ==#
53 n = FacetNormal(mesh)
54
55 ## Define variational forms ==#
56 a = dot(w1, alpha1*v1)*dx + dot(w2, alpha2*v2)*dx \
57     - div(w1) * p1 * dx - div(w2) * p2 * dx + q1 * div(v1) * dx + q2 * div(v2) * dx + \
58     q1 * fact * (p1 - p2) * dx - q2 * fact * (p1 - p2) * dx
59
60 L = dot(w1,rhob1)*dx + dot(w2,rhob2)*dx - \
61     dot(w1,n) * p1.left * ds(1) - dot(w2,n) * p2.left * ds(1) - \
62     dot(w1,n) * p1.right * ds(2) - dot(w2,n) * p2.right * ds(2) - \
63     dot(w1,n) * p1.bottom * ds(3) - dot(w2,n) * p2.bottom * ds(3) - \
64     dot(w1,n) * p1.top * ds(4) - dot(w2,n) * p2.top * ds(4) - \
65     dot(w1,n) * p1.back * ds(5) - dot(w2,n) * p2.back * ds(5) - \
66     dot(w1,n) * p1.front * ds(6) - dot(w2,n) * p2.front * ds(6)
67

```

```

68  #== Solver options ==#
69  parameters_twofields = {
70      "ksp_type": "gmres",
71      "pc_type": "fieldsplit",
72      "pc_fieldsplit_0_fields": "0,2",
73      "pc_fieldsplit_1_fields": "1,3",
74      "pc_fieldsplit_type": "schur",
75      "pc_fieldsplit_schur_fact_type": "full",
76      "pc_fieldsplit_schur_precondition": "selfp",
77      "fieldsplit_0_ksp_type": "preonly",
78      "fieldsplit_0_pc_type": "bjacobi",
79      "fieldsplit_1_ksp_type": "preonly",
80      "fieldsplit_1_pc_type": "fieldsplit",
81      "fieldsplit_1_pc_fieldsplit_type": "additive",
82      "fieldsplit_1_fieldsplit_0_ksp_type": "preonly",
83      "fieldsplit_1_fieldsplit_0_pc_type": "hypre",
84      "fieldsplit_1_fieldsplit_0_pc_hypre_boomeramg_strong_threshold": 0.75,
85      "fieldsplit_1_fieldsplit_0_pc_hypre_boomeramg_agg_nl": 2,
86      "fieldsplit_1_fieldsplit_1_ksp_type": "preonly",
87      "fieldsplit_1_fieldsplit_1_pc_type": "hypre",
88      "fieldsplit_1_fieldsplit_1_pc_hypre_boomeramg_strong_threshold": 0.75,
89      "fieldsplit_1_fieldsplit_1_pc_hypre_boomeramg_agg_nl": 2,
90      "ksp_rtol": 1e-5
91  }
92
93  #== Solve problem ==#
94  solution = Function(wSpace)
95  A = assemble(a, bcs=bcs, mat_type='aij')
96  b = assemble(L)
97  solver = LinearSolver(A, P=None, options_prefix="twofields_",
98                         solver_parameters=parameters_twofields)
99  solver.solve(solution, b)
100 v1sol, p1sol, v2sol, p2sol = solution.split()
101
102 #== Define exact solutions ==#
103 p1_ex = Function(pSpace)
104 p2_ex = Function(pSpace)
105 v1_ex = Function(vSpace)
106 v2_ex = Function(vSpace)

```

```

107 p1_exact = Expression("(1/pi)*exp(pi*x[0])*(sin(pi*x[1]) +\\
108                         sin(pi*x[2])) - (1/(1.0*1.0))*(exp(3.316625*x[1]) +\\
109                         exp(3.316625*x[2]))", degree = 5)
110 p2_exact = Expression("(1/pi)*exp(pi*x[0])*(sin(pi*x[1]) +\\
111                         sin(pi*x[2])) + (1/(1*0.1))*(exp(3.316625*x[1]) +\\
112                         exp(3.316625*x[2]))", degree = 5)
113 v1_exact = Expression((" -1*exp(pi*x[0])* (sin(pi*x[1]) +\\
114                         sin(pi*x[2])) ", "-1*exp(pi*x[0])*cos(pi*x[1]) +\\
115                         (3.316625/1.0)*exp(3.316625*x[1]) ", "-1*exp(pi*x[0])*cos(pi*x[2]) +\\
116                         (3.316625/1.0)*exp(3.316625*x[2]) "), degree = 5)
117 v2_exact = Expression((" -0.1*exp(pi*x[0])* (sin(pi*x[1]) +\\
118                         sin(pi*x[2])) ", "-0.1*exp(pi*x[0])*cos(pi*x[1]) -\\
119                         (3.316625/1.0)*exp(3.316625*x[1]) ", " -0.1*exp(pi*x[0])*cos(pi*x[2]) -\\
120                         (3.316625/1.0)*exp(3.316625*x[2]) "), degree = 5)
121 p1_ex = project(p1_exact, pSpace)
122 v1_ex = project(v1_exact, vSpace)
123 p2_ex = project(p2_exact, pSpace)
124 v2_ex = project(v2_exact, vSpace)
125 #== L2 error norms ==#
126 L2_p1 = errornorm(p1_ex,plsol,norm_type='L2',degree_rise= 3)
127 L2_v1 = errornorm(v1_ex,v1sol,norm_type='L2',degree_rise= 3)
128 L2_p2 = errornorm(p2_ex,p2sol,norm_type='L2',degree_rise= 3)
129 L2_v2 = errornorm(v2_ex,v2sol,norm_type='L2',degree_rise= 3)

```

Listing 5: Firedrake code for 3D problem with TET mesh using CG-VMS formulation

```

1 from firedrake import *
2 import numpy as np
3
4 #== Create mesh ==
5 mesh = BoxMesh(5,5,5,1,1,1)
6
7 #== Function spaces ==
8 vSpace = VectorFunctionSpace(mesh, "CG",1)
9 pSpace = FunctionSpace(mesh, "CG",1)
10 wSpace = MixedFunctionSpace([vSpace,pSpace,vSpace,pSpace])
11
12 #== Define trial and test functions ==

```

```

13 (v1,p1,v2,p2) = TrialFunctions (wSpace)
14 (w1,q1,w2,q2) = TestFunctions (wSpace)
15
16 ## Parameters and material properties ==#
17 rhob1, rhob2 = Constant((0.0,0.0,0.0)), Constant((0.0,0.0,0.0))
18 mu = Constant(1.0)
19 beta = Constant(1.0)
20 fact = 1.0
21 k1, k2 = Constant(1.0), Constant(0.1)
22 alpha1, alpha2 = Constant(mu/k1), Constant(mu/k2)
23 eta = np.sqrt(1.0 *(1.0+0.1)/(1.0 * 0.1))
24 invalphal = 1.0 / alpha1
25 invalpha2 = 1.0 / alpha2
26
27 ## Boundary conditions ==#
28 p1_left = interpolate(Expression("(1/pi)*(sin(pi*x[1])+sin(pi*x[2])) -\
29                         (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
30 p1_right = interpolate(Expression("(1/pi)*exp(pi)*(sin(pi*x[1])+sin(pi*x[2])) -\
31                         (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
32 p1_bottom = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) -\
33                         (1.0 + exp(eta*x[2]))",eta=eta) , pSpace)
34 p1_top = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) -\
35                         (exp(eta) + exp(eta*x[2]))",eta=eta) , pSpace)
36 p1_back = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) -\
37                         (exp(eta*x[1]) + 1.0)",eta=eta) , pSpace)
38 p1_front = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) -\
39                         (exp(eta) + exp(eta*x[1]))",eta=eta) , pSpace)
40 p2_left = interpolate(Expression("(1/pi)*(sin(pi*x[1])+sin(pi*x[2])) +\
41                         10.0 * (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
42 p2_right = interpolate(Expression("(1/pi)*exp(pi)*(sin(pi*x[1])+sin(pi*x[2])) +\
43                         10.0 * (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
44 p2_bottom = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) +\
45                         10.0 * (1.0 + exp(eta*x[2]))",eta=eta) , pSpace)
46 p2_top = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) +\
47                         10.0 * (exp(eta) + exp(eta*x[2]))",eta=eta) , pSpace)
48 p2_back = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) +\
49                         10.0 * (exp(eta*x[1]) + 1.0)",eta=eta) , pSpace)
50 p2_front = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) +\
51                         10.0 * (exp(eta) + exp(eta*x[1]))",eta=eta) , pSpace)

```

```

52 bcs = []
53
54 ## Normal vectors ==#
55 n = FacetNormal(mesh)
56
57 ## Define variational forms ==#
58 a = dot(w1, alphal*v1)*dx + dot(w2, alpha2*v2)*dx \
59     - div(w1) * p1 * dx - div(w2) * p2 * dx \
60     + q1 * div(v1) * dx + q2 * div(v2) * dx +\
61     q1 * fact * (p1 - p2) * dx - q2 * fact * (p1 - p2) * dx -\
62     0.5 * dot( alphal * w1 - grad(q1), invalphal * (alphal * v1 + grad(p1)) ) * dx -\
63     0.5 * dot( alpha2 * w2 - grad(q2), invalpha2 * (alpha2 * v2 + grad(p2)) ) * dx
64
65 L = dot(w1,rhob1)*dx + dot(w2,rhob2)*dx -\
66     0.5 * dot( alphal * w1 - grad(q1), invalphal * rhob1 ) * dx -\
67     0.5 * dot( alpha2 * w2 - grad(q2), invalpha2 * rhob2 ) * dx -\
68     dot(w1,n) * p1_left * ds(1) - dot(w2,n) * p2_left * ds(1) -\
69     dot(w1,n) * p1_right * ds(2) - dot(w2,n) * p2_right * ds(2) -\
70     dot(w1,n) * p1_bottom * ds(3) - dot(w2,n) * p2_bottom * ds(3) -\
71     dot(w1,n) * p1_top * ds(4) - dot(w2,n) * p2_top * ds(4) -\
72     dot(w1,n) * p1_back * ds(5) - dot(w2,n) * p2_back * ds(5) -\
73     dot(w1,n) * p1_front * ds(6) - dot(w2,n) * p2_front * ds(6)
74
75 ## Solver options ==#
76 parameters.twofields = {
77     "ksp_type": "gmres",
78     "pc_type": "fieldsplit",
79     "pc_fieldsplit_0_fields": "0,2",
80     "pc_fieldsplit_1_fields": "1,3",
81     "pc_fieldsplit_type": "schur",
82     "pc_fieldsplit_schur_fact_type": "full",
83     "pc_fieldsplit_schur_precondition": "selfp",
84     "fieldsplit_0_ksp_type": "preonly",
85     "fieldsplit_0_pc_type": "bjacobi",
86     "fieldsplit_1_ksp_type": "preonly",
87     "fieldsplit_1_pc_type": "fieldsplit",
88     "fieldsplit_1_pc_fieldsplit_type": "additive",
89     "fieldsplit_1_fieldsplit_0_ksp_type": "preonly",
90     "fieldsplit_1_fieldsplit_0_pc_type": "hypre",

```

```

91   "fieldsplit_1.fieldsplit_0.pc_hypre_boomeramg_strong_threshold": 0.75,
92   "fieldsplit_1.fieldsplit_0.pc_hypre_boomeramg_agg_nl": 2,
93   "fieldsplit_1.fieldsplit_1.ksp_type": "preonly",
94   "fieldsplit_1.fieldsplit_1.pc_type": "hypre",
95   "fieldsplit_1.fieldsplit_1.pc_hypre_boomeramg_strong_threshold": 0.75,
96   "fieldsplit_1.fieldsplit_1.pc_hypre_boomeramg_agg_nl": 2,
97   "ksp_rtol": 1e-5
98 }
99
100 #== Solve problem ==#
101 solution = Function(wSpace)
102 A = assemble(a, bcs=bcs, mat_type='aij')
103 b = assemble(L)
104 solver = LinearSolver(A, P=None, options_prefix="twofields_",
105                         solver_parameters=parameters_twofields)
106 solver.solve(solution, b)
107 v1sol, p1sol, v2sol, p2sol = solution.split()
108
109 #== Define exact solutions ==#
110 p1_ex = Function(pSpace)
111 p2_ex = Function(pSpace)
112 v1_ex = Function(vSpace)
113 v2_ex = Function(vSpace)
114 p1_exact = Expression("(1/pi)*exp(pi*x[0])*(sin(pi*x[1]) + \
115                         sin(pi*x[2])) - (1/(1.0*1.0))*(exp(3.316625*x[1]) + \
116                         exp(3.316625*x[2]))", degree = 5)
117 p2_exact = Expression("(1/pi)*exp(pi*x[0])*(sin(pi*x[1]) + \
118                         sin(pi*x[2])) + (1/(1*0.1))*(exp(3.316625*x[1]) + \
119                         exp(3.316625*x[2]))", degree = 5)
120 v1_exact = Expression((-1*exp(pi*x[0]))*(sin(pi*x[1]) + \
121                         sin(pi*x[2])), "-1*exp(pi*x[0])*cos(pi*x[1]) + \
122                         (3.316625/1.0)*exp(3.316625*x[1]), "-1*exp(pi*x[0])*cos(pi*x[2]) + \
123                         (3.316625/1.0)*exp(3.316625*x[2])", degree = 5)
124 v2_exact = Expression((" -0.1*exp(pi*x[0])*(sin(pi*x[1]) + \
125                         sin(pi*x[2])) ", "-0.1*exp(pi*x[0])*cos(pi*x[1]) - \
126                         (3.316625/1.0)*exp(3.316625*x[1]) ", "-0.1*exp(pi*x[0])*cos(pi*x[2]) - \
127                         (3.316625/1.0)*exp(3.316625*x[2]) ", degree = 5)
128 p1_ex = interpolate(p1_exact, pSpace)
129 v1_ex = interpolate(v1_exact, vSpace)

```

```

130 p2_ex = interpolate(p2_exact, pSpace)
131 v2_ex = interpolate(v2_exact, vSpace)
132
133 #= L2 error norms ==#
134 L2_p1 = errornorm(p1_ex,p1sol,norm_type='L2',degree_rise= 3)
135 L2_v1 = errornorm(v1_ex,v1sol,norm_type='L2',degree_rise= 3)
136 L2_p2 = errornorm(p2_ex,p2sol,norm_type='L2',degree_rise= 3)
137 L2_v2 = errornorm(v2_ex,v2sol,norm_type='L2',degree_rise= 3)

```

Listing 6: Firedrake code for 3D problem with TET mesh using DG-VMS formulation

```

1 from firedrake import *
2 import numpy as np
3
4 #= Create mesh ==#
5 mesh = BoxMesh(5,5,5,1,1,1)
6
7 #= Function spaces ==#
8 vSpace = VectorFunctionSpace(mesh, "DG",1)
9 pSpace = FunctionSpace(mesh, "DG",1)
10 wSpace = MixedFunctionSpace([vSpace,pSpace,vSpace,pSpace])
11
12 #= Define trial and test functions ==#
13 (v1,p1,v2,p2) = TrialFunctions(wSpace)
14 (w1,q1,w2,q2) = TestFunctions(wSpace)
15
16 #= Parameters and material properties ==#
17 rhob1, rhob2 = Constant((0.0,0.0,0.0)), Constant((0.0,0.0,0.0))
18 mu = Constant(1.0)
19 beta = Constant(1.0)
20 fact = 1.0
21 k1, k2 = Constant(1.0), Constant(0.1)
22 alpha1, alpha2 = Constant(mu/k1), Constant(mu/k2)
23 eta = np.sqrt(1.0 * (1.0+0.1)/(1.0 * 0.1))
24 invalpha1 = 1.0 / alpha1
25 invalpha2 = 1.0 / alpha2
26
27 #= Boundary conditions ==#

```

```

28 p1_left = interpolate(Expression("(1/pi)*(sin(pi*x[1])+sin(pi*x[2])) -\  

29                         (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
30 p1_right = interpolate(Expression("(1/pi)*exp(pi)*(sin(pi*x[1])+sin(pi*x[2])) -\  

31                         (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
32 p1_bottom = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) -\  

33                         (1.0 + exp(eta*x[2]))",eta=eta) , pSpace)
34 p1_top = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) -\  

35                         (exp(eta) + exp(eta*x[2]))",eta=eta) , pSpace)
36 p1_back = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) -\  

37                         (exp(eta*x[1]) + 1.0)",eta=eta) , pSpace)
38 p1_front = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) -\  

39                         (exp(eta) + exp(eta*x[1]))",eta=eta) , pSpace)
40 p2_left = interpolate(Expression("(1/pi)*(sin(pi*x[1])+sin(pi*x[2])) +\  

41                         10.0 * (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
42 p2_right = interpolate(Expression("(1/pi)*exp(pi)*(sin(pi*x[1])+sin(pi*x[2])) +\  

43                         10.0 * (exp(eta*x[1])+exp(eta*x[2]))",eta=eta) , pSpace)
44 p2_bottom = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) +\  

45                         10.0 * (1.0 + exp(eta*x[2]))",eta=eta) , pSpace)
46 p2_top = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[2]) +\  

47                         10.0 * (exp(eta) + exp(eta*x[2]))",eta=eta) , pSpace)
48 p2_back = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) +\  

49                         10.0 * (exp(eta*x[1]) + 1.0)",eta=eta) , pSpace)
50 p2_front = interpolate(Expression("(1/pi)*exp(pi*x[0])*sin(pi*x[1]) +\  

51                         10.0 * (exp(eta) + exp(eta*x[1]))",eta=eta) , pSpace)
52 bcs = []
53
54 ## Define normal vector, h_avg, and penalty parameters ==#
55 n = FacetNormal(mesh)
56 h = CellSize(mesh)
57 h_avg = (h('+') + h('-'))/2
58 eta_u, eta_p = Constant(10.), Constant(10.)
59
60 ## Define variational forms ==#
61 a = dot(w1, alphal*v1)*dx + dot(w2, alpha2*v2)*dx \
62     - div(w1) * p1 *dx - div(w2) * p2 * dx \
63     + q1 * div(v1) * dx + q2 * div(v2) * dx +\
64     q1 * fact * (p1 - p2) * dx - q2 * fact * (p1 - p2) * dx -\
65     0.5 * dot( alphal * w1 - grad(q1),invalphal * (alphal * v1 + grad(p1)) ) * dx -\
66     0.5 * dot( alpha2 * w2 - grad(q2),invalpha2 * (alpha2 * v2 + grad(p2)) ) * dx +\

```

```

67     jump(w1,n) * avg(p1) * dS + jump(w2,n) * avg(p2) * dS - \
68     avg(q1) * jump(v1,n) * dS - avg(q2) * jump(v2,n) * dS + \
69     eta_u * h_avg * avg(alpha1) * (jump(w1,n) * jump(v1,n)) * dS + \
70     eta_u * h_avg * avg(alpha2) * (jump(w2,n) * jump(v2,n)) * dS + \
71     (eta_p/h_avg) * avg(1/alpha1) * dot(jump(q1,n),jump(p1,n)) * dS + \
72     (eta_p / h_avg) * avg(1 / alpha2) * dot(jump(q2,n),jump(p2,n)) * dS
73
74 L = dot(w1,rhob1)*dx + dot(w2,rhob2)*dx - \
75     0.5 * dot( alpha1 * w1 - grad(q1),invalpha1 * rhob1 ) * dx - \
76     0.5 * dot( alpha2 * w2 - grad(q2),invalpha2 * rhob2 ) * dx - \
77     dot(w1,n) * p1_left * ds(1) - dot(w2,n) * p2_left * ds(1) - \
78     dot(w1,n) * p1_right * ds(2) - dot(w2,n) * p2_right * ds(2) - \
79     dot(w1,n) * p1_bottom * ds(3) - dot(w2,n) * p2_bottom * ds(3) - \
80     dot(w1,n) * p1_top * ds(4) - dot(w2,n) * p2_top * ds(4) - \
81     dot(w1,n) * p1_back * ds(5) - dot(w2,n) * p2_back * ds(5) - \
82     dot(w1,n) * p1_front * ds(6) - dot(w2,n) * p2_front * ds(6)
83
84 == Solver options ==
85 parameters_twofields = {
86     "ksp_type": "gmres",
87     "pc_type": "fieldsplit",
88     "pc_fieldsplit_0_fields": "0,2",
89     "pc_fieldsplit_1_fields": "1,3",
90     "pc_fieldsplit_type": "schur",
91     "pc_fieldsplit_schur_fact_type": "full",
92     "pc_fieldsplit_schur_precondition": "selfp",
93     "fieldsplit_0_ksp_type": "preonly",
94     "fieldsplit_0_pc_type": "bjacobi",
95     "fieldsplit_1_ksp_type": "preonly",
96     "fieldsplit_1_pc_type": "fieldsplit",
97     "fieldsplit_1_pc_fieldsplit_type": "additive",
98     "fieldsplit_1_fieldsplit_0_ksp_type": "preonly",
99     "fieldsplit_1_fieldsplit_0_pc_type": "hypre",
100    "fieldsplit_1_fieldsplit_0_pc_hypre_boomeramg_strong_threshold": 0.75,
101    "fieldsplit_1_fieldsplit_0_pc_hypre_boomeramg_agg_nl": 2,
102    "fieldsplit_1_fieldsplit_1_ksp_type": "preonly",
103    "fieldsplit_1_fieldsplit_1_pc_type": "hypre",
104    "fieldsplit_1_fieldsplit_1_pc_hypre_boomeramg_strong_threshold": 0.75,
105    "fieldsplit_1_fieldsplit_1_pc_hypre_boomeramg_agg_nl": 2,

```

```

106     "ksp_rtol": 1e-5
107 }
108
109 ## Solve problem ==#
110 solution = Function(wSpace)
111 A = assemble(a, bcs=bcs, mat_type='aij')
112 b = assemble(L)
113 solver = LinearSolver(A,P=None,options_prefix="twofields_",
114                         solver_parameters=parameters_twofields)
115 solver.solve(solution,b)
116 v1sol,plsol,v2sol,p2sol = solution.split()
117
118 ## Define exact solutions ==#
119 p1_ex = Function(pSpace)
120 p2_ex = Function(pSpace)
121 v1_ex = Function(vSpace)
122 v2_ex = Function(vSpace)
123 p1_exact = Expression("(1/pi)*exp(pi*x[0])*(sin(pi*x[1]) +\
124                         sin(pi*x[2])) - (1/(1.0*1.0))*(exp(3.316625*x[1]) +\
125                         exp(3.316625*x[2]))", degree = 5)
126 p2_exact = Expression("(1/pi)*exp(pi*x[0])*(sin(pi*x[1]) +\
127                         sin(pi*x[2])) + (1/(1*0.1))*(exp(3.316625*x[1]) +\
128                         exp(3.316625*x[2]))", degree = 5)
129 v1_exact = Expression((" -1*exp(pi*x[0])*(sin(pi*x[1]) +\
130                         sin(pi*x[2])) ", "-1*exp(pi*x[0])*cos(pi*x[1]) +\
131                         (3.316625/1.0)*exp(3.316625*x[1]) ", "-1*exp(pi*x[0])*cos(pi*x[2]) +\
132                         (3.316625/1.0)*exp(3.316625*x[2])"), degree = 5)
133 v2_exact = Expression((" -0.1*exp(pi*x[0])*(sin(pi*x[1]) +\
134                         sin(pi*x[2])) ", "-0.1*exp(pi*x[0])*cos(pi*x[1]) -\
135                         (3.316625/1.0)*exp(3.316625*x[1]) ", " -0.1*exp(pi*x[0])*cos(pi*x[2]) -\
136                         (3.316625/1.0)*exp(3.316625*x[2])"), degree = 5)
137 p1_ex = interpolate(p1_exact, pSpace)
138 v1_ex = interpolate(v1_exact, vSpace)
139 p2_ex = interpolate(p2_exact, pSpace)
140 v2_ex = interpolate(v2_exact, vSpace)
141
142 ## L2 error norms ==#
143 L2_p1 = errornorm(p1_ex,plsol,norm_type='L2',degree_rise= 3)
144 L2_v1 = errornorm(v1_ex,v1sol,norm_type='L2',degree_rise= 3)

```

```

145 L2_p2 = errornorm(p2_ex,p2sol,norm_type='L2',degree_rise= 3)
146 L2_v2 = errornorm(v2_ex,v2sol,norm_type='L2',degree_rise= 3)

```

D Mathematical Proofs for Interface Condition

D.1 A proof of the minimum power theorem

Based on the first-order optimality condition it will suffice to show that

$$\begin{aligned} \delta\mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}; \delta\mathbf{v}_{\text{free}}, \delta\mathbf{v}_{\text{por}}] &:= \left[\frac{d}{d\epsilon} \mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}} + \epsilon\delta\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}} + \epsilon\delta\mathbf{v}_{\text{por}}] \right]_{\epsilon=0} = 0 \\ &\quad \forall (\delta\mathbf{v}_{\text{free}}, \delta\mathbf{v}_{\text{por}}) \in \mathcal{W}. \end{aligned} \tag{D.1}$$

The positive definite Hessians will ensure that the extremum is in fact a minimum. The Gâteaux variation can be written as follows¹

$$\begin{aligned} \delta\mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}; \delta\mathbf{v}_{\text{free}}, \delta\mathbf{v}_{\text{por}}] &= \int_{\mathcal{K}_{\text{free}}} \frac{\partial\Phi_{\text{free}}}{\partial\mathbf{D}_{\text{free}}} \cdot \delta\mathbf{D}_{\text{free}} \, d\Omega \\ &\quad + \int_{\mathcal{K}_{\text{por}}} \left(\frac{\partial\Phi_{\text{por}}}{\partial\mathbf{v}_{\text{por}}} \cdot \delta\mathbf{v}_{\text{por}} + \frac{\partial\Phi_{\text{por}}}{\partial\mathbf{D}_{\text{por}}} \cdot \delta\mathbf{D}_{\text{por}} \right) \, d\Omega \\ &\quad + \int_{\Gamma_{\text{int}}} \left(\frac{\partial\Psi}{\partial\mathbf{v}_{\text{free}}^*} \cdot \delta\mathbf{v}_{\text{free}}^* + \frac{\partial\Psi}{\partial\mathbf{v}_{\text{por}}^*} \cdot \delta\mathbf{v}_{\text{por}}^* + \frac{\partial\Psi}{\partial v_n} \cdot \delta v_n \right) \, d\Gamma \\ &\quad - \int_{\mathcal{K}_{\text{free}}} \gamma \mathbf{b}_{\text{free}} \cdot \delta\mathbf{v}_{\text{free}} \, d\Omega - \int_{\Gamma_{\text{free}}^t} \mathbf{t}_{\text{free}}^p(\mathbf{x}) \cdot \delta\mathbf{v}_{\text{free}}(\mathbf{x}) \, d\Gamma \\ &\quad - \int_{\mathcal{K}_{\text{por}}} \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} \cdot \delta\mathbf{v}_{\text{por}} \, d\Omega - \int_{\Gamma_{\text{por}}^t} \mathbf{t}_{\text{por}}^p(\mathbf{x}) \cdot \delta\mathbf{v}_{\text{por}}(\mathbf{x}) \, d\Gamma. \end{aligned} \tag{D.3}$$

Using the conditions (8.46a)–(8.46c) under the requirement (R1), we obtain the following

$$\delta\mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}; \delta\mathbf{v}_{\text{free}}, \delta\mathbf{v}_{\text{por}}] = \int_{\mathcal{K}_{\text{free}}} \mathbf{T}_{\text{free}}^{\text{extra}} \cdot \delta\mathbf{D}_{\text{free}} \, d\Omega + \int_{\mathcal{K}_{\text{por}}} (\mathbf{i}_{\text{por}} \cdot \delta\mathbf{v}_{\text{por}} + \mathbf{T}_{\text{por}}^{\text{extra}} \cdot \delta\mathbf{D}_{\text{por}}) \, d\Omega$$

¹ $\delta\mathbf{D}_{\text{free}} := \frac{1}{2}(\text{grad}[\delta\mathbf{v}_{\text{free}}] + \text{grad}[\delta\mathbf{v}_{\text{free}}]^T)$ and $\delta\mathbf{D}_{\text{por}} := \frac{1}{2}(\text{grad}[\delta\mathbf{v}_{\text{por}}] + \text{grad}[\delta\mathbf{v}_{\text{por}}]^T)$.

$$\begin{aligned}
& + \int_{\Gamma_{\text{int}}} \left(\frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*} \cdot \delta \mathbf{v}_{\text{free}}^* + \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*} \cdot \delta \mathbf{v}_{\text{por}}^* + \frac{\partial \Psi}{\partial v_n} \cdot \delta v_n \right) d\Gamma \\
& - \int_{\mathcal{K}_{\text{free}}} \gamma \mathbf{b}_{\text{free}} \cdot \delta \mathbf{v}_{\text{free}} d\Omega - \int_{\Gamma_{\text{free}}^t} \mathbf{t}_{\text{free}}^p(\mathbf{x}) \cdot \delta \mathbf{v}_{\text{free}}(\mathbf{x}) d\Gamma \\
& - \int_{\mathcal{K}_{\text{por}}} \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} \cdot \delta \mathbf{v}_{\text{por}} d\Omega - \int_{\Gamma_{\text{por}}^t} \mathbf{t}_{\text{por}}^p(\mathbf{x}) \cdot \delta \mathbf{v}_{\text{por}}(\mathbf{x}) d\Gamma.
\end{aligned} \tag{D.4}$$

Noting the internal constraints (8.31b) and (8.32b), utilizing the decomposition of the Cauchy stresses (8.29), and invoking the Green's identity, we obtain the following

$$\begin{aligned}
\delta \mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}; \delta \mathbf{v}_{\text{free}}, \delta \mathbf{v}_{\text{por}}] = & - \int_{\mathcal{K}_{\text{free}}} \underbrace{(\text{div}[\mathbf{T}_{\text{free}}] + \gamma \mathbf{b}_{\text{free}})}_{= \mathbf{0} \text{ due to (8.31a)}} \cdot \delta \mathbf{v}_{\text{free}} d\Omega \\
& - \int_{\mathcal{K}_{\text{por}}} \underbrace{(\text{div}[\mathbf{T}_{\text{por}}] + \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} - \mathbf{i}_{\text{por}})}_{= \mathbf{0} \text{ due to (8.32a)}} \cdot \delta \mathbf{v}_{\text{por}} d\Omega \\
& + \int_{\partial \mathcal{K}_{\text{free}}} (\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}}) \cdot \delta \mathbf{v}_{\text{free}}(\mathbf{x}) d\Gamma - \int_{\Gamma_{\text{free}}^t} \mathbf{t}_{\text{free}}^p(\mathbf{x}) \cdot \delta \mathbf{v}_{\text{free}}(\mathbf{x}) d\Gamma \\
& + \int_{\partial \mathcal{K}_{\text{por}}} (\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}}) \cdot \delta \mathbf{v}_{\text{por}}(\mathbf{x}) d\Gamma - \int_{\Gamma_{\text{por}}^t} \mathbf{t}_{\text{por}}^p(\mathbf{x}) \cdot \delta \mathbf{v}_{\text{por}}(\mathbf{x}) d\Gamma \\
& + \int_{\Gamma_{\text{int}}} \left(\frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*} \cdot \delta \mathbf{v}_{\text{free}}^* + \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*} \cdot \delta \mathbf{v}_{\text{por}}^* + \frac{\partial \Psi}{\partial v_n} \cdot \delta v_n \right) d\Gamma.
\end{aligned} \tag{D.5}$$

Noting the decomposition of the boundaries $\partial \mathcal{K}_{\text{free}}$ and $\partial \mathcal{K}_{\text{por}}$, given by equations (8.6) and (8.7), we obtain the following

$$\delta \mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}; \delta \mathbf{v}_{\text{free}}, \delta \mathbf{v}_{\text{por}}] = \int_{\Gamma_{\text{free}}^t} \underbrace{(\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}}^{\text{ext}} - \mathbf{t}_{\text{free}}^p(\mathbf{x})) \cdot \delta \mathbf{v}_{\text{free}}(\mathbf{x})}_{= \mathbf{0} \text{ due to (8.31c)}} d\Gamma \tag{D.6}$$

$$\begin{aligned}
& + \int_{\Gamma_{\text{free}}^v} (\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}}^{\text{ext}}) \cdot \delta \mathbf{v}_{\text{free}}(\mathbf{x}) d\Gamma \\
& + \int_{\Gamma_{\text{por}}^t} \underbrace{(\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}}^{\text{ext}} - \mathbf{t}_{\text{por}}^p(\mathbf{x})) \cdot \delta \mathbf{v}_{\text{por}}(\mathbf{x})}_{= \mathbf{0} \text{ due to (8.32c)}} d\Gamma \tag{D.7}
\end{aligned}$$

$$\begin{aligned}
& + \int_{\Gamma_{\text{por}}^v} (\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}}^{\text{ext}}) \cdot \delta \mathbf{v}_{\text{por}}(\mathbf{x}) d\Gamma \\
& + \int_{\Gamma_{\text{int}}} (\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}}) \cdot \delta \mathbf{v}_{\text{free}}(\mathbf{x}) d\Gamma + \int_{\Gamma_{\text{int}}} (\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}}) \cdot \delta \mathbf{v}_{\text{por}}(\mathbf{x}) d\Gamma
\end{aligned}$$

$$+ \int_{\Gamma_{\text{int}}} \left(\frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*} \cdot \delta \mathbf{v}_{\text{free}}^* + \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*} \cdot \delta \mathbf{v}_{\text{por}}^* + \frac{\partial \Psi}{\partial v_n} \cdot \delta v_n \right) d\Gamma. \quad (\text{D.8})$$

Invoking that $\delta \mathbf{v}_{\text{free}}(\mathbf{x})$ and $\delta \mathbf{v}_{\text{por}}(\mathbf{x})$, respectively, vanish on Γ_{free}^v and Γ_{por}^v (see §8.2.5), and using the first interface condition (8.33a) and the notation introduced in (8.25), we obtain the following

$$\begin{aligned} \delta \mathcal{P}_{\text{coupled}}[\mathbf{v}_{\text{free}}, \mathbf{v}_{\text{por}}; \delta \mathbf{v}_{\text{free}}, \delta \mathbf{v}_{\text{por}}] &= \int_{\Gamma_{\text{int}}} \left(\hat{\mathbf{n}}_{\text{free}} \cdot \mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}} - \hat{\mathbf{n}}_{\text{por}} \cdot \mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}} + \frac{\partial \Psi}{\partial v_n} \right) \cdot \delta v_n d\Gamma \\ &\quad + \int_{\Gamma_{\text{int}}} \left(\mathbf{T}_{\text{free}} \hat{\mathbf{n}}_{\text{free}} + \frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^*} \right) \cdot \delta \mathbf{v}_{\text{free}}^* d\Gamma \\ &\quad + \int_{\Gamma_{\text{int}}} \left(\mathbf{T}_{\text{por}} \hat{\mathbf{n}}_{\text{por}} + \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^*} \right) \cdot \delta \mathbf{v}_{\text{por}}^* d\Gamma. \end{aligned} \quad (\text{D.9})$$

Finally, by utilizing the interface conditions (8.33b)–(8.33d) we have established that the first variation of $\mathcal{P}_{\text{coupled}}$ vanishes.

D.2 A proof of the uniqueness theorem

On the contrary, assume that

$$\{\mathbf{v}_{\text{free}}^{(1)}(\mathbf{x}), p_{\text{free}}^{(1)}(\mathbf{x}), \mathbf{v}_{\text{por}}^{(1)}(\mathbf{x}), p_{\text{por}}^{(1)}(\mathbf{x})\} \quad \text{and} \quad \{\mathbf{v}_{\text{free}}^{(2)}(\mathbf{x}), p_{\text{free}}^{(2)}(\mathbf{x}), \mathbf{v}_{\text{por}}^{(2)}(\mathbf{x}), p_{\text{por}}^{(2)}(\mathbf{x})\},$$

are two solutions to the coupled Stokes-Darcy-Brinkman equations for the prescribed data. That is, $\{\mathbf{v}_{\text{free}}^{(1)}(\mathbf{x}), p_{\text{free}}^{(1)}(\mathbf{x})\}$ and $\{\mathbf{v}_{\text{free}}^{(2)}(\mathbf{x}), p_{\text{free}}^{(2)}(\mathbf{x})\}$ satisfy the Stokes equations in $\mathcal{K}_{\text{free}}$, and $\{\mathbf{v}_{\text{por}}^{(1)}(\mathbf{x}), p_{\text{por}}^{(1)}(\mathbf{x})\}$ and $\{\mathbf{v}_{\text{por}}^{(2)}(\mathbf{x}), p_{\text{por}}^{(2)}(\mathbf{x})\}$ satisfy the Darcy-Brinkman equations in \mathcal{K}_{por} . Moreover, $\mathbf{v}_{\text{free}}^{(1)}$, $\mathbf{v}_{\text{free}}^{(2)}$, $\mathbf{v}_{\text{por}}^{(1)}$ and $\mathbf{v}_{\text{por}}^{(2)}$ satisfy

$$\operatorname{div} [\mathbf{v}_{\text{free}}^{(1)}] = 0 \quad \text{and} \quad \operatorname{div} [\mathbf{v}_{\text{free}}^{(2)}] = 0 \quad \text{in } \mathcal{K}_{\text{free}}; \quad (\text{D.10})$$

$$\operatorname{div} [\mathbf{v}_{\text{por}}^{(1)}] = 0 \quad \text{and} \quad \operatorname{div} [\mathbf{v}_{\text{por}}^{(2)}] = 0 \quad \text{in } \mathcal{K}_{\text{por}}. \quad (\text{D.11})$$

Since the pairs $\{\mathbf{v}_{\text{free}}^{(1)}(\mathbf{x}), \mathbf{v}_{\text{por}}^{(1)}(\mathbf{x})\}$ and $\{\mathbf{v}_{\text{free}}^{(2)}(\mathbf{x}), \mathbf{v}_{\text{por}}^{(2)}(\mathbf{x})\}$ are both kinematically admissible, the minimum power theorem implies that

$$\mathcal{P}_{\text{coupled}} \left[\mathbf{v}_{\text{free}}^{(1)}(\mathbf{x}), \mathbf{v}_{\text{por}}^{(1)}(\mathbf{x}) \right] = \mathcal{P}_{\text{coupled}} \left[\mathbf{v}_{\text{free}}^{(2)}(\mathbf{x}), \mathbf{v}_{\text{por}}^{(2)}(\mathbf{x}) \right]. \quad (\text{D.12})$$

Using the definition of $\mathcal{P}_{\text{coupled}}$ given by equation (8.53), the above equation can be expanded as

$$\begin{aligned} & \frac{1}{2} \left(\Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(1)} \right] - \Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(2)} \right] \right) + \frac{1}{2} \left(\Phi_{\text{por}} \left[\mathbf{v}_{\text{por}}^{(1)} \right] - \Phi_{\text{por}} \left[\mathbf{v}_{\text{por}}^{(2)} \right] \right) \\ & + \int_{\Gamma_{\text{int}}} \left(\Psi \left[\overset{*}{\mathbf{v}}_{\text{free}}^{(1)}, \overset{*}{\mathbf{v}}_{\text{por}}^{(1)}, v_n^{(1)} \right] - \Psi \left[\overset{*}{\mathbf{v}}_{\text{free}}^{(2)}, \overset{*}{\mathbf{v}}_{\text{por}}^{(2)}, v_n^{(2)} \right] \right) d\Gamma \\ & = \int_{\mathcal{K}_{\text{free}}} \gamma \mathbf{b}_{\text{free}} \cdot \left(\mathbf{v}_{\text{free}}^{(1)} - \mathbf{v}_{\text{free}}^{(2)} \right) d\Omega + \int_{\Gamma_{\text{free}}^t} \mathbf{t}_{\text{free}}^p \cdot \left(\mathbf{v}_{\text{free}}^{(1)} - \mathbf{v}_{\text{free}}^{(2)} \right) d\Omega \\ & + \int_{\mathcal{K}_{\text{por}}} \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} \cdot \left(\mathbf{v}_{\text{por}}^{(1)} - \mathbf{v}_{\text{por}}^{(2)} \right) d\Omega + \int_{\Gamma_{\text{por}}^t} \mathbf{t}_{\text{por}}^p \cdot \left(\mathbf{v}_{\text{por}}^{(1)} - \mathbf{v}_{\text{por}}^{(2)} \right) d\Omega. \end{aligned} \quad (\text{D.13})$$

Noting the rate of internal dissipation in the Stokes model, it is easy to establish the following

$$\frac{1}{2} \left(\Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(1)} \right] - \Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(2)} \right] \right) = \frac{1}{2} \Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(1)} - \mathbf{v}_{\text{free}}^{(2)} \right] + \int_{\mathcal{K}_{\text{free}}} 2\mu \mathbf{D}_{\text{free}}^{(2)} \cdot \left(\mathbf{D}_{\text{free}}^{(1)} - \mathbf{D}_{\text{free}}^{(2)} \right) d\Omega. \quad (\text{D.14})$$

Using equation (D.11)₂ the above equation can be written as

$$\frac{1}{2} \left(\Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(1)} \right] - \Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(2)} \right] \right) = \frac{1}{2} \Phi_{\text{free}} \left[\mathbf{v}_{\text{free}}^{(1)} - \mathbf{v}_{\text{free}}^{(2)} \right] + \int_{\mathcal{K}_{\text{free}}} \mathbf{T}_{\text{free}}^{(2)} \cdot \left(\mathbf{D}_{\text{free}}^{(1)} - \mathbf{D}_{\text{free}}^{(2)} \right) d\Omega, \quad (\text{D.15})$$

where

$$\mathbf{T}_{\text{free}}^{(2)} = -p_{\text{free}}^{(2)} \mathbf{I} + 2\mu \mathbf{D}_{\text{free}}^{(2)}. \quad (\text{D.16})$$

On similar lines, one can establish the following relation

$$\frac{1}{2} \left(\Phi_{\text{por}} \left[\mathbf{v}_{\text{por}}^{(1)} \right] - \Phi_{\text{por}} \left[\mathbf{v}_{\text{por}}^{(2)} \right] \right) = \frac{1}{2} \Phi_{\text{por}} \left[\mathbf{v}_{\text{por}}^{(1)} - \mathbf{v}_{\text{por}}^{(2)} \right] + \int_{\mathcal{K}_{\text{por}}} \mathbf{T}_{\text{por}}^{(2)} \cdot \left(\mathbf{D}_{\text{por}}^{(1)} - \mathbf{D}_{\text{por}}^{(2)} \right) d\Omega$$

$$+ \int_{\mathcal{K}_{\text{por}}} \mu \mathbf{K}^{-1} \mathbf{v}_{\text{por}}^{(2)} \cdot (\mathbf{v}_{\text{por}}^{(1)} - \mathbf{v}_{\text{por}}^{(2)}) d\Omega, \quad (\text{D.17})$$

where

$$\mathbf{T}_{\text{por}}^{(2)} = -p_{\text{por}}^{(2)} \mathbf{I} + 2\mu \mathbf{D}_{\text{por}}^{(2)}. \quad (\text{D.18})$$

We note the fields under the second solution satisfy the balance of linear momentum; that is:

$$\operatorname{div} [\mathbf{T}_{\text{free}}^{(2)}] + \gamma \mathbf{b}_{\text{free}} = \mathbf{0} \quad \text{in } \mathcal{K}_{\text{free}} \text{ and} \quad (\text{D.19})$$

$$\operatorname{div} [\mathbf{T}_{\text{por}}^{(2)}] + \gamma \phi_{\text{por}} \mathbf{b}_{\text{por}} = \mu \mathbf{K}^{-1} \mathbf{v}_{\text{por}}^{(2)} \quad \text{in } \mathcal{K}_{\text{por}}, \quad (\text{D.20})$$

and the prescribed tractions on the external boundary; that is:

$$\mathbf{t}_{\text{free}}^{(2)} := \mathbf{T}_{\text{free}}^{(2)} \hat{\mathbf{n}}_{\text{free}} = \mathbf{t}_{\text{free}}^p \quad \text{on } \Gamma_{\text{free}}^t \text{ and} \quad (\text{D.21})$$

$$\mathbf{t}_{\text{por}}^{(2)} := \mathbf{T}_{\text{por}}^{(2)} \hat{\mathbf{n}}_{\text{por}} = \mathbf{t}_{\text{por}}^p \quad \text{on } \Gamma_{\text{por}}^t. \quad (\text{D.22})$$

Using the interface conditions (8.33b)–(8.33d), we have the following

$$\begin{aligned} & \frac{1}{2} \Phi_{\text{free}} [\mathbf{v}_{\text{free}}^{(1)} - \mathbf{v}_{\text{free}}^{(2)}] + \frac{1}{2} \Phi_{\text{por}} [\mathbf{v}_{\text{por}}^{(1)} - \mathbf{v}_{\text{por}}^{(2)}] \\ & + \int_{\Gamma_{\text{int}}} \left(\Psi \left[\mathbf{v}_{\text{free}}^{*(1)}, \mathbf{v}_{\text{por}}^{*(1)}, v_n^{(1)} \right] - \Psi \left[\mathbf{v}_{\text{free}}^{*(2)}, \mathbf{v}_{\text{por}}^{*(2)}, v_n^{(2)} \right] \right) d\Gamma \\ & = \int_{\Gamma_{\text{int}}} \left(\frac{\partial \Psi}{\partial \mathbf{v}_{\text{free}}^{*(2)}} \cdot \left(\mathbf{v}_{\text{free}}^{*(1)} - \mathbf{v}_{\text{free}}^{*(2)} \right) + \frac{\partial \Psi}{\partial \mathbf{v}_{\text{por}}^{*(2)}} \cdot \left(\mathbf{v}_{\text{por}}^{*(1)} - \mathbf{v}_{\text{por}}^{*(2)} \right) + \frac{\partial \Psi}{\partial v_n^{(2)}} \cdot (v_n^{(1)} - v_n^{(2)}) \right) d\Gamma. \end{aligned} \quad (\text{D.23})$$

Noting the functional form of Ψ , the above equation reduces to the following

$$\frac{1}{2} \Phi_{\text{free}} [\mathbf{v}_{\text{free}}^{(1)} - \mathbf{v}_{\text{free}}^{(2)}] + \frac{1}{2} \Phi_{\text{por}} [\mathbf{v}_{\text{por}}^{(1)} - \mathbf{v}_{\text{por}}^{(2)}] + \int_{\Gamma_{\text{int}}} \Psi \left[\mathbf{v}_{\text{free}}^{*(1)} - \mathbf{v}_{\text{free}}^{*(2)}, \mathbf{v}_{\text{por}}^{*(1)} - \mathbf{v}_{\text{por}}^{*(2)}, v_n^{(1)} - v_n^{(2)} \right] d\Gamma = 0. \quad (\text{D.24})$$

Using the fact that $\Phi_{\text{free}}[\cdot]$, $\Phi_{\text{por}}[\cdot]$ and $\Psi[\cdot]$ are individually norms (and hence individually non-negative), each term in the above equation is individually zero. This further implies that

$$\mathbf{v}_{\text{free}}^{(1)}(\mathbf{x}) = \mathbf{v}_{\text{free}}^{(2)}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{K}_{\text{free}}, \quad (\text{D.25a})$$

$$\mathbf{v}_{\text{por}}^{(1)}(\mathbf{x}) = \mathbf{v}_{\text{por}}^{(2)}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{K}_{\text{por}}, \quad (\text{D.25b})$$

$$\mathbf{v}_{\text{free}}^{*(1)}(\mathbf{x}) = \mathbf{v}_{\text{free}}^{*(2)}(\mathbf{x}) \quad \forall \mathbf{x} \in \Gamma_{\text{free}}, \quad (\text{D.25c})$$

$$\mathbf{v}_{\text{por}}^{*(1)}(\mathbf{x}) = \mathbf{v}_{\text{por}}^{*(2)}(\mathbf{x}) \quad \forall \mathbf{x} \in \Gamma_{\text{por}}, \text{ and} \quad (\text{D.25d})$$

$$v_n^{(1)}(\mathbf{x}) = v_n^{(2)}(\mathbf{x}) \quad \forall \mathbf{x} \in \Gamma_{\text{int}}. \quad (\text{D.25e})$$

The balance of linear momentum in $\mathcal{K}_{\text{free}}$ and \mathcal{K}_{por} , respectively, implies that:

$$\text{grad} \left[p_{\text{free}}^{(1)}(\mathbf{x}) - p_{\text{free}}^{(2)}(\mathbf{x}) \right] = \mathbf{0} \quad \forall \mathbf{x} \in \mathcal{K}_{\text{free}} \text{ and} \quad (\text{D.26a})$$

$$\text{grad} \left[p_{\text{por}}^{(1)}(\mathbf{x}) - p_{\text{por}}^{(2)}(\mathbf{x}) \right] = \mathbf{0} \quad \forall \mathbf{x} \in \mathcal{K}_{\text{por}}, \quad (\text{D.26b})$$

which further implies that:

$$p_{\text{free}}^{(1)}(\mathbf{x}) = p_{\text{free}}^{(2)}(\mathbf{x}) + C_1 \quad \forall \mathbf{x} \in \mathcal{K}_{\text{free}} \quad \text{and} \quad p_{\text{por}}^{(1)}(\mathbf{x}) = p_{\text{por}}^{(2)}(\mathbf{x}) + C_2 \quad \forall \mathbf{x} \in \mathcal{K}_{\text{por}}, \quad (\text{D.27})$$

where C_1 and C_2 are arbitrary constants. Using the interface condition given by equation (8.33b) and noting that the velocity fields are continuous fields, we conclude that $C_1 = C_2 = C$ and

$$p_{\text{free}}^{(1)}(\mathbf{x}) = p_{\text{free}}^{(2)}(\mathbf{x}) + C \quad \text{and} \quad p_{\text{por}}^{(1)}(\mathbf{x}) = p_{\text{por}}^{(2)}(\mathbf{x}) + C \quad \forall \mathbf{x} \in \Gamma_{\text{int}}. \quad (\text{D.28})$$

Physically, the constant C fixes the datum for the pressure field. This completes the proof.

