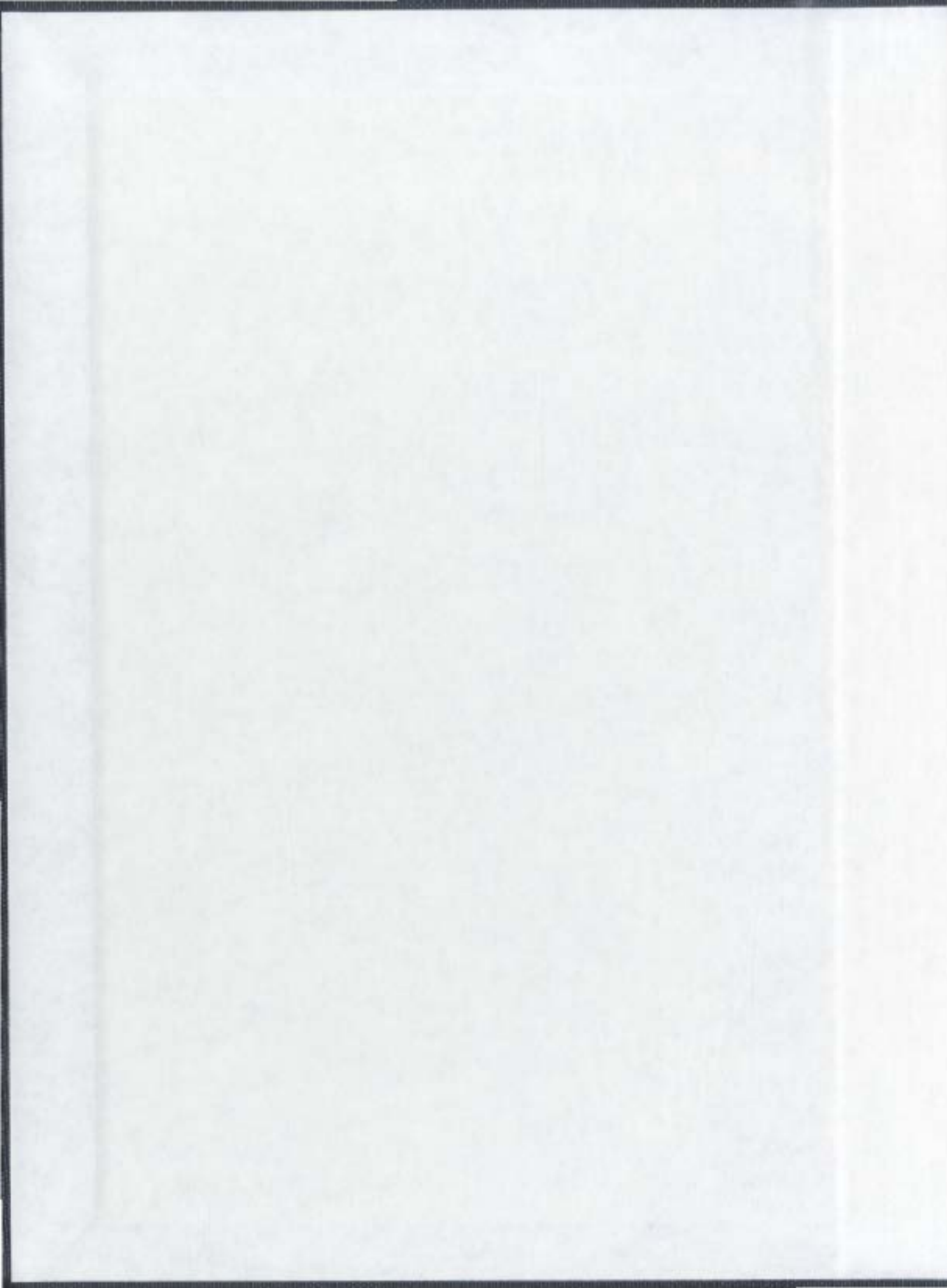


WELL MODELING INCORPORATING COMPOSITIONAL
AND NON-ISOTHERMAL EFFECTS

WORAKANOK THANYAMANTA



Well Modeling Incorporating Compositional and Non-Isothermal Effects

by
© Worakanok Thanyamanta

A thesis submitted to the School of Graduate Studies in partial fulfilment
of the requirements for the degree of Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

March 2007

St. John's

Newfoundland



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-31331-2

Our file Notre référence

ISBN: 978-0-494-31331-2

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

ABSTRACT

This study proposed a methodology to predicting asphaltene precipitation in wells with advanced completions. A fundamental comprehensive two-phase flow model was proposed to predict asphaltene precipitation in horizontal production wells. The main objective of the research was to incorporate compositional and non-isothermal effects into an isothermal physical flow model and to investigate conditions that promote asphaltene precipitation. The precipitation induced by flow restriction found in equipment installed as parts of advanced well completion was the main focus. The proposed model consisted of a black-oil network model and a compositional asphaltene model allowing investigation of both physical and phase behaviour of flowing fluids in non-isothermal environment. The original network model was an isothermal model. In order to take into account heat transport in wellbores, an approach to predict wellbore temperature profile using a network-type model was proposed. This enabled the proposed network model to predict pressure, temperature, flow rate, and phase fractions of the produced fluid in different parts of the well. Local asphaltene precipitation predictions were able to be conducted at locations where asphaltene formation was likely to occur. In this research an asphaltene model was proposed. The model was developed based on a pseudo-three-phase solid-type asphaltene model. An isenthalpic flash was used to reflect the characteristics of flows through restrictions. By using the proposed asphaltene model, asphaltene onset conditions can be predicted and asphaltene phase behaviour at the conditions of interest can be determined.

In this study the proposed methodology was used and successfully predicted flow behaviour in example well networks. It was found that different completion schemes have different effects on fluid conditions in the well and, in turn, asphaltene

precipitation behaviour. The example simulations suggested that the drastic pressure drop induced by the valve restriction can cause asphaltene to precipitate. In the example cases the precipitation occurred inside the restriction thus downstream and upstream conditions were not sufficient for evaluation of asphaltene precipitation. In addition temperature also has effects on asphaltene precipitation prediction. The isothermal assumption for production systems, where the temperature in the well is always constant and equal to the reservoir temperature, may not be sufficient to accurately describe the asphaltene phase behaviour in the well. An increase in fluid temperature inside the restricted flow path also has effects on prediction of the asphaltene onset pressure. The extent of these temperature effects depend on the shift in the predicted onset pressure compared with the prevailing pressure drops in the entire well network.

ACKNOWLEDGEMENTS

Throughout these years of my research the most important lesson in my life is that opportunity is the best thing to give to someone. I therefore feel deeply grateful to my two co-supervisors, Dr.Thormod Johansen and Dr.Kelly Hawboldt, who gave me such a great opportunity to try and do what I love. Thank you so much for your kindness and understanding, for believing in me, and for all guidance and supports. You two are the reason I can call Newfoundland my second home.

Other than my co-supervisors this research could not be completed without a few more people. Firstly I would like to thank Vitaly Khorikov and Christian Johansen for supports on developing the model. Your help was really appreciated. Many thanks are also extended to Dr.Majid Abdi (my supervisory committee), Dr.Neil Bose, and Dr.Faisal Khan who are always willing to provide supports whenever I need. Without these people doing this research would not only be harder but also less pleasant. I would like to also thank Kristian Krejbjerg from Calsep Inc. for thermodynamics support and WellDynamics for the useful information on ICV's.

I had a chance to know the following great people on the way doing this research. First of all I am so very grateful to Zhongxin Huo from Shell for having been so kind and helpful since we first met. I also thank George and Michael from Schlumberger for taking time to explain and show me around the asphaltene lab.

Being away from home I have come to realize how blessed I am to be surrounded by loving family and friends. I would like to first give a big hug to my mom, my dad and my brother for being so supportive and pushing me toward the best that I can be. I thank my life-long friends June, Prae, Duang, and Dan for always being with me through all the rough times (and good times of course!). The support from friend colleagues, Jamal Siavoshi, Lisa O'Brien, Haibo Niu, and Liping Wang is very appreciated.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
NOMENCLATURE	xiv
1 INTRODUCTION	1
1.1 Background	1
1.2 Objectives of the Research	2
1.3 Scope of the Study	2
1.4 Organization of the Thesis	3
2 NETWORK MODEL	5
2.1 Structure of Network Model	6
2.2 Multi-Phase Flow Models	8
2.2.1 Compositional Models	10
2.2.2 Black-Oil Models	11
2.3 Conservation Equations	16
2.3.1 Mass Balance Equation	16
2.3.2 Momentum Balance Equation	17
2.3.3 Energy Balance Equation	18
2.3.4 Heat Transfer at Solid-Liquid Interface	22
2.4 Isothermal Flow Parameter Calculations	23
2.4.1 Material Balance at Nodes	24
2.4.2 Inflow Equations	25
2.4.3 Momentum Balance Equations for Other Bridges	26
2.4.4 Split Equations	30
2.4.5 Boundary Conditions	31
2.5 Temperature Calculations	32
2.5.1 Heat Transfer in Wellbore	32
2.5.2 Overall Heat Transfer Coefficients	35
2.5.3 Energy Balance at Nodes	38
2.5.4 Boundary Conditions	41
2.6 Network Model Integration	41

2.7	Network Model Verification	46
2.7.1	Inaccuracy Caused by Discretization	46
2.7.2	Comparison with Published Literature	50
3	THERMODYNAMIC PHASE EQUILIBRIUM	56
3.1	Phase Equilibrium	56
3.2	Cubic Equations of State	65
3.2.1	Two-parameter Cubic Equations of State	68
3.2.2	Volume Shift	71
3.2.3	Mixing Rules	72
3.3	Vapour-Liquid Phase Equilibrium	74
4	ASPHALTENE PRECIPITATION PREDICTION	78
4.1	Asphaltene Models	78
4.2	Pressure Calculations	90
4.3	Temperature and Phase Equilibrium Calculations	98
4.3.1	Isenthalpic Flash	98
4.3.2	Pseudo-Three-Phase Equilibrium Calculation	101
4.3.3	Precipitation Prediction	106
5	COUPLED MODEL AND APPLICATIONS	110
5.1	Coupled Model	110
5.2	Model Applications	112
5.2.1	Fluid Characterization	112
5.3	Case 1: Well with an Inflow Control Valve	114
5.3.1	Network Model	115
5.3.2	Asphaltene Precipitation Prediction	130
5.4	Case 2: Production from Reservoir with Varied Reservoir Temperature	142
5.4.1	Network Model	143
5.4.2	Asphaltene Precipitation Prediction	151
5.5	Case 3: Well with Multiple Inflow Control Valves	152
5.6	Case 4: Well with Restricted Flow in the Annulus	156
5.6.1	Network Model	159
5.6.2	Asphaltene Precipitation Prediction	160
5.7	Discussions	164
6	CONCLUSIONS, RESEARCH NOVELTY, AND RECOMMENDATIONS	167
6.1	Summary and Conclusions	167
6.2	Novelty of the Research	169
6.3	Recommendations for Future Research	170
	References	172

Appendix A: Example Fluid Properties	177
Appendix B: Matlab Files	182

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 Well basic characteristics for discretization error analysis.	48
2-2 Results from two different discretization schemes used in model verification.	48
2-3 Well and fluid properties reported in Table 5.2 and Table 5.3 in Dawkrajai et al. (2005)	51
2-4 Well basic characteristics used for comparisons with published literature.	51
2-5 Results from the proposed model compared with Dawkrajai et al. (2005)	52
5-1 Well basic characteristics for wells in Case 1.	116
5-2 Well basic characteristics for the well in Case 2-C.	146
5-3 Well basic characteristics for the well in Case 3.	152
5-4 Well basic characteristics for the well in Case 4.	159
A-1 Static precipitation test results for Oil 1 at 100 °C at different pressures (adapted from Burke et al. [11]).	177
A-2 Fluid compositions and basic properties.	177
A-3 Characterized fluid properties.	178

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 Modeling of well flow paths.	6
2-2 Network model and flow parameters.	7
2-3 Typical types of reservoir.	9
2-4 Black-oil-model parameters as a function of pressure.	12
2-5 Flow control volume.	16
2-6 Energy transported by convection across a surface element.	18
2-7 Temperature profile by heat conduction of a solid slab between two parallel plates.	19
2-8 Stress vector acting on a surface element creating work done on the plus-side fluid.	20
2-9 Acceleration pressure loss through completion due to radial inflow. . .	27
2-10 Radial flow convergence into a slot opening.	30
2-11 Heat transfer in one-dimensional flow.	33
2-12 Overall heat transfer between fluid and surroundings.	35
2-13 Energy balance at a node.	38
2-14 Network model schematic.	42
2-15 Variation of fluid properties with pressure and temperature.	43
2-16 Curve fitting of pre-generated property values.	44
2-17 Dead oil viscosity as a function of temperature.	45
2-18 Well network used in evaluating errors caused by well discretization. .	47
2-19 Pressure profiles for different discretization schemes.	49
2-20 Total well flow profiles for different discretization schemes.	49
2-21 Temperature profiles in tubing for different discretization schemes. . .	50

2-22 Pressure drop in well tubing compared with results from Dawkrajai et al. (2005)	53
2-23 Production rate compared with results from Dawkrajai et al. (2005) .	53
2-24 Matching the predicted temperature profile of fluid in tubing with the results from Dawkrajai et al. (2005) by varying the Joule-Thompson coefficient	54
3-1 Vapour-liquid flash algorithm.	77
4-1 Possible locations for asphaltene deposition in oil production.	79
4-2 Schematic representation of a crude/micelle/precipitate system. . . .	80
4-3 Asphaltene phase envelope.	81
4-4 Asphaltene solubility based on a thermodynamic model.	82
4-5 Asphaltene content and pressure relation from gravimetric method .	82
4-6 Effect of temperature on asphaltene solubility.	83
4-7 Asphaltene solubility as a function of crude in-situ density.	85
4-8 Asphaltene instability correlations.	87
4-9 Asphaltene precipitation behaviour determined using refractive index.	89
4-10 Simplified restriction geometry.	90
4-11 Restriction geometry divided into sections.	92
4-12 Pressure drop prediction using Equation 4.9b for single-phase liquid. .	95
4-13 Pressure drop prediction using Equation 4.10 for two-phase fluid. . . .	96
4-14 Effects of frictional and accelerational pressure drop for different compressibility factors.	97
4-15 Variation of compressibility factor of example fluid at 100 °C.	98
4-16 Isenthalpic flash schematic.	99
4-17 Asphaltene model schematic.	107
4-18 Asphaltene precipitation onset pressure predicted by the model. . . .	108
4-19 Effect of volume shift parameter on asphaltene precipitation.	108
5-1 Proposed model schematic.	111
5-2 Example well network: Case 1.	114

5-3	Pressure profile for the well with ICV.	118
5-4	Flow rate profile for the well with ICV.	118
5-5	Pressure profile for the conventionally-completed well.	119
5-6	Flow rate profile for the conventionally-completed well.	119
5-7	Temperature profile at wellbore-reservoir interface.	121
5-8	Temperature profile for the conventionally-completed well.	122
5-9	Temperature profile for the well with ICV.	122
5-10	Energy transport in a well network.	123
5-11	Energy flow profile.	124
5-12	Effects of heat transport mechanisms on well temperature profile . . .	125
5-13	Effects of overall heat transfer coefficients on well temperature profile	125
5-14	Convergence of viscosity values in the proposed model.	126
5-15	Viscosity profile for the well with ICV.	127
5-16	Effects of temperature on viscosity profile.	128
5-17	Network model results in the case of saturated fluid.	129
5-18	Modeling of flow through the ICV using a simple geometry.	131
5-19	Pressure drop through ICV for various upstream radii.	132
5-20	Pressure drop through ICV for various downstream radii.	133
5-21	Pressure drop through ICV for various restriction lengths.	134
5-22	Pressure drop through ICV for various friction factors.	135
5-23	Simplified restriction geometry used to model the ICV in Case 1. . . .	136
5-24	Pressure profile across ICV.	137
5-25	Temperature profile across ICV.	138
5-26	Effects of temperature on calculations of reference asphaltene fugacities.	139
5-27	Asphaltene fugacity profile across ICV.	140
5-28	Asphaltene onset conditions predicted by the model.	141

5-29 Temperature profile in the case with linearly decreasing temperature in reservoir.	144
5-30 Temperature profile in the case with linearly increasing temperature in reservoir.	144
5-31 Example well network: Case 2-C.	146
5-32 Pressure profile for the well in two reservoir zones with distinct temperatures.	148
5-33 Flow rate profile for the well in two reservoir zones with distinct temperatures.	148
5-34 Temperature profile for the well in two reservoir zones with distinct temperatures.	149
5-35 Pressure profile for the well in two reservoir zones with distinct temperatures and pressures.	149
5-36 Flow rate profile for the well in two reservoir zones with distinct temperatures and pressures.	150
5-37 Temperature profile for the well in two reservoir zones with distinct temperatures and pressures.	150
5-38 Example well network: Case 3.	153
5-39 Pressure profile in the case with two ICVs.	154
5-40 Flow rate profile in the case with two ICVs.	154
5-41 Temperature profile in the case with two ICVs.	155
5-42 Fugacity profile through ICVs in Case 3 indicating asphaltene pre- cipitation.	155
5-43 Pressure profile in the case where the ICVs caused reversed flow in the annulus.	157
5-44 Flow rate profile in the case where the ICVs caused reversed flow in the annulus.	157
5-45 Temperature profile in the case where the ICVs caused reversed flow in the annulus.	158
5-46 Example well network: Case 4.	158
5-47 Pressure profile in well with restricted flow in annulus.	160
5-48 Flow rate profile in well with restricted flow in annulus.	161

5-49	Temperature profile in well with restricted flow in annulus.	161
5-50	Simplified restriction geometry used to model the restricted annulus in Case 4.	162
5-51	Pressure profile through the restricted annulus.	163
5-52	Temperature profile through the restricted annulus.	163
5-53	Fugacity profile through the restricted annulus.	164
A-1	Oil formation volume factor of example fluid at 100 °C.	178
A-2	Gas formation volume factor of example fluid at 100 °C.	179
A-3	Gas solubility of example fluid at 100 °C.	179
A-4	Dead-oil viscosity of example fluid.	180
A-5	Density of example fluid at 100 °C.	180
A-6	Compressibility factor of example fluid at 100 °C.	181
A-7	Molar volume of example fluid at 100 °C.	181

NOMENCLATURE

Abbreviations

EOS	Equation of state
GOR	Gas oil ratio
ICV	Inflow control valve
RI	Refractive index
STC	Stock tank condition
VLE	Vapour-liquid equilibrium

Greek Symbols

α	Liquid volume fraction or liquid holdup
γ	Specific gravity or Permeable fraction of surface area
μ	Chemical potential or viscosity
μ_{od}	Dead-oil viscosity
μ_{os}	Saturated-oil viscosity
Ω	Constant for parameter “ a ”, “ b ”, or “ a_c ” in two-parameter cubic EOS
ω	Acentric factor
$\bar{\rho}$	Average two-phase density
ϕ	Fugacity coefficient
π	Stress tensor
ρ	Density

τ_w	Wall shear stress
θ	Index for phases
ε_D	Pipe roughness

Symbols

A	Cross-sectional area
a, A	Attractive-term parameter in two-parameter cubic EOS
b, B	Repulsive-term parameter in two-parameter cubic EOS
B_g	Gas formation volume factor
B_o	Oil formation volume factor
\hat{C}_p	Specific heat capacity
c	Correction term in Peneloux’s volume shift calculation
C_p	Specific heat capacity
D	Diameter
D_h	Hydraulic diameter
\dot{E}	Energy flow rate
\dot{e}	Energy flux
E	Energy
E_k	Kinetic energy
E_p	Potential energy
f	Friction factor or fugacity

f_a	Asphaltene fugacity	N_v	Number of points evaluated inside the valve restriction
\bar{G}_i	Partial molar Gibbs energy		
G	Gibbs or Gibbs free energy	Nu	Nusselt number
g	Gravitational acceleration or Molar Gibbs energy	P	Wetting perimeter or pressure
		p	Pressure
\hat{H}	Enthalpy per unit mass	P^s	Vapour pressure
H	Enthalpy	p_b	Bubble point pressure
h	Heat transfer coefficient	PI	Productivity index
H^{id}	Ideal-gas molar enthalpy	Pr	Prandtl number
H^{res}	Residual enthalpy	Q	Heat
H_{ij}	Partial molar enthalpy of component i in phase j	q	Flow rate
		R	Universal gas constant
H_{spec}	Specific molar enthalpy	r	Radius
K	Absolute permeability or equilibrium ratio	r_e	Drainage radius
		r_i	Well inner radius
k	Thermal conductivity	r_o	Well outer radius
k_{rg}	Gas relative permeability	R_s	Gas solubility
k_{ro}	Oil relative permeability	Re	Reynolds number
L	Length	S	Entropy
\dot{m}	Mass flux	s	Skin factor or molar entropy
N	Number of network segments	T	Temperature
N	Total number of components	t	Time
n	Number of mole	\hat{U}	Specific internal energy
n	Number of moles	U	Internal energy
n^L	Liquid mole fraction	U	Overall heat transfer coefficient
n^V	Vapour mole fraction	u, U	Parameters in general form of two-parameter cubic EOS

\hat{V}	Specific volume	r	Reduced property or radial direction
V	Volume		
v	Velocity or molar volume	ref	Reference condition
v_a	Molar volume of pure asphaltene	res	Reservoir or reservoir conditions
\dot{W}	Work rate	S	Slot
MW	Molecular weight	STC	Standard conditions
W	Work	SYS	System
w, W	Parameters in general form of two-parameter cubic EOS	TP	Two-phase property
x	Liquid mole fraction	$UNIV$	Universe
y	Vapour mole fraction	w	Wall or work done by molecular motions
\bar{Z}_i	Partial molar compressibility factor	x	x direction
Z	Compressibility factor	y	y direction
z	Distance in z direction or feed mole fraction	z	z direction
		Superscripts	
Z_{RA}	Rackett compressibility factor	cor	Corrected value
		g	Gas phase
		id	Ideal property
		ig	Ideal-gas property
		k	iteration index
		L	Lower bound
		o	Oil phase
		R	Residual property
		RC	Reservoir conditions
		STC	Standard conditions
		U	Upper bound
		Subscripts	
b	Bulk fluid		
c	Critical property		
$cond$	Conductive heat transfer		
$conv$	Convective heat transfer		
dg	dissolved gas		
I	Inflow		
i, j, h	Index for phases or components		
o	Oil phase		

CHAPTER 1

INTRODUCTION

1.1 Background

In the last decades advances in oil production technologies have been developed toward production optimization and solutions to challenging production problems. Such problems include undesired solid formation during oil production. One of the most common solid precipitation is caused by asphaltene.

To ensure smooth operation it is crucial to be able to accurately predict fluid flow behavior in the wellbore. Physical properties of the fluid including density, viscosity, flow rates, and phase fractions are important parameters. Determination of fluid pressure and temperature is also mandatory especially when complex well completions are considered and solid formation is a concern. In the case where asphaltene phase is present in the system compositional properties of the fluid are also required. These flow parameters are interrelated and have influence on the overall behaviour of the fluid. Consequently comprehensive understanding of the flow system requires combinations of various disciplines including fluid dynamics, heat-transfer, and thermodynamics.

In this research a comprehensive model is proposed in order to simulate flow behaviour of oil mixtures produced through completed horizontal wells where asphaltene problem is expected. The model is an attempt to incorporate compositional and non-isothermal effects into a physical, constant-composition network model. The proposed model is constructed of a network solver which considers both

momentum-transport and heat-transport in the flow system to provide simultaneous solutions to the fluid's physical transport in non-isothermal environment. The ability of the model to predict fluid phase behaviour is achieved and optimized by associating the network model with a compositional asphaltene model which is only used locally to provide detailed thermodynamic phase behaviour calculations. Below summarizes the main objectives of this research.

1.2 Objectives of the Research

1. To develop a methodology for evaluating asphaltene precipitation in completed horizontal production wells.
2. To develop a fundamental comprehensive two-phase flow model incorporating compositional and non-isothermal effects.
3. To use the proposed model to predict flow conditions and local asphaltene precipitation in different well completion scenarios.
4. To investigate effects of well completions on pressure and temperature distribution inside the wells.
5. To investigate conditions that promote asphaltene precipitation focusing on flow restrictions introduced by advanced well completions.
6. To provide recommendations on further development of the model.

1.3 Scope of the Study

This research is an attempt to provide a methodology for evaluating asphaltene precipitation in completed advanced production wells. A fundamental comprehensive model is developed based on existing models to predict flow conditions and phase behaviour of reservoir fluids during primary depletion. The model is a two-phase model considering only oil and gas phases of fluids in wellbores. For the solid phase, even though there are other kinds of solid problems (e.g. wax and hydrates),

only asphaltene precipitation model will be considered and integrated into the proposed model. The asphaltene model used in this research is developed based on an available solid asphaltene model that is compatible with integrating into a network model. The model is meant to provide a methodology for predicting asphaltene precipitation. Moreover, the model is designed to be able to predict asphaltene phase behaviour at a specified condition. Asphaltene amount and deposition mechanisms will not be investigated.

1.4 Organization of the Thesis

This thesis is divided into six chapters. The first chapter, Introduction, provides an introduction to the thesis: background problems, objectives of the research, scope of the study, and organization of the thesis. In Chapter 2, Network Model, formulations of the network model are presented. Firstly the chapter shows how a horizontal well is modeled using the basic structure of the network model. In order to provide a brief understanding of what happens inside a wellbore, fundamental equations including multi-phase flow models and basic conservation equations that describe flow transport mechanisms are first presented. These equations are used as basic equations throughout the thesis to derive working equations for the proposed model. Later in the chapter, formulation of the network model's governing equations and how they are solved for unknown flow parameters under both isothermal and non-isothermal conditions are outlined.

The following chapters present derivations of detailed compositional analysis. Chapter 3, Thermodynamic Phase Equilibrium, provides the basis to thermodynamics used in the compositional asphaltene model presented in Chapter 4. This chapter is therefore dedicated to fluid phase equilibrium calculations focusing on vapour-liquid phase equilibrium and equations of state. In Chapter 4, Asphaltene Precipitation Prediction, the proposed asphaltene model is formulated. The first

section of the chapter discusses asphaltene precipitation mechanisms and available asphaltene models. The next section presents the use of simplified geometries to model and predict pressure variations of flows through restrictions. Models for predicting such pressure variations are developed and proposed for single-phase liquids and two-phase fluids. The last section outlines the development of the phase equilibrium calculations used in the proposed asphaltene model and how local asphaltene precipitation can be predicted.

Chapter 5, Coupled Model and Applications, presents example simulations using the proposed methodology and models presented in the previous chapters. The network model and the compositional asphaltene model are coupled in this chapter and used to predict asphaltene precipitation in example wells with different completion scenarios. The results and findings from the example simulations are also discussed. The thesis is then summarized and concluded in Chapter 6 where novelty of the research and recommendations for future development of the proposed model are also presented.

CHAPTER 2

NETWORK MODEL

During oil production, hydrocarbon fluid flows from the reservoir through perforations into the wellbore and enters the vertical well section at the heel and passes through various completion components of the well. Flow simulation using mathematical models is one of the effective and commonly used methods to predict flow conditions and flow behavior under this complex situation. One of these models is the “network solver” [9]. Also called network model, the network solver is widely used and provides accurate prediction for flow parameters in horizontal production wells.

In this chapter formulation of a network model to solve for unknown flow parameters will be presented. The first section presents in general basic structure of the network model and how a horizontal well can be modeled using such structure. Section 2.2 and 2.3 present fundamental models and equations used in constructing the network model. These include multi-phase-flow models and basic conservation equations. Section 2.4 and Section 2.5 describe how momentum-transport and energy-transport problems, respectively, can be solved using the network model. Due to complex nature of flows in completed wellbores the aforementioned transport problems are in many cases required to be solved together. Section 2.6 presents how the momentum and energy calculations are coupled in the proposed network model. The last section of the chapter discusses errors associated with this approach. Results from the proposed model are also compared with published literature.

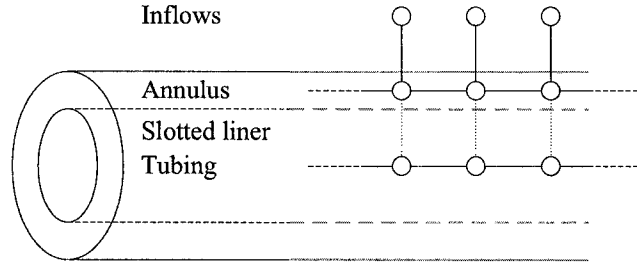


Figure 2-1: Modeling of well flow paths.

2.1 Structure of Network Model

In a network model a horizontal well is modeled using a network consisting of nodes to discretize the well geometry. Each pair of the nodes is connected with bridges that represent flows of the fluid from one point to another. Figure 2-1 shows different parts of a horizontal well modeled using nodes and connecting bridges. These nodes and bridges form a network of flow paths that represents the entire well (Figure 2-2). The series of nodes and bridges are designed so that the entire length of the well is divided into a finite number of segments (N) of specified length as shown in Figure 2-2(a). The number of segments depends on the desired accuracy of the model.

As shown in Figure 2-2(b) a fundamental segment of the well network consists of three nodes and four connecting bridges (except for Segment N). Each segment may have different reservoir properties (e.g. pressure (p_{res}) and permeability (K)). These nodes locate points in the reservoir, annulus, and tubing. There are four types of bridges in the network namely inlet, annular, annulus-to-tubing, and tubing bridges. In the well network inlet bridges represent inflows from the reservoir into the well. In the production process these reservoir inflows enter the well, e.g. through perforations into the annular space. Flows in the annular space of the well are represented by annular bridges connecting two consecutive annular nodes. The flow direction of fluid in the annular space is parallel to the main well flow in the tubing. However it can either flow toward the heel of well or the opposite depending on the

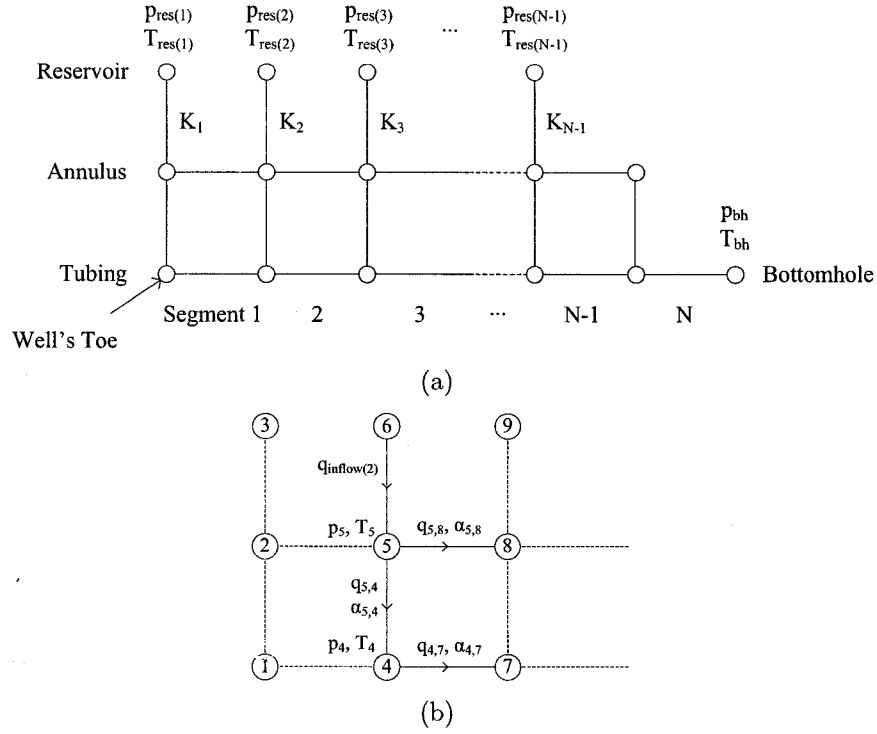


Figure 2-2: Network model and flow parameters. (a) Well network divided into N segments. (b) Unknowns in Segment 2 of the well network.

structure of the network. The annular fluid then may enter the tubing continuously over the length of the well tubing (e.g. through slotted liners) or at certain locations (e.g. through inflow control devices). This type of flow is modeled using bridges perpendicular to the main tubing flow. These annular-to-tubing bridges connect the annular and tubing nodes of the same segment. Finally tubing bridge connecting two consecutive tubing nodes represent the main fluid flow in the tubing space. The fluid in the tubing always flows from the toe toward the heel of well. As there are inflows continuously coming in from the reservoir they accumulate and cause the flow rate in the tubing to increase toward the heel of well and thus pressure drop.

In order to solve for unknown flow parameters the unknowns are assigned to the nodes or bridges. In this research unknown pressures and temperatures are assigned at the nodes and other flow parameters including flow rates and phase fractions are assigned for the bridges (Figure 2-2(b)). Based on this, in momentum calculations,

each segment is allowed to have the maximum of 9 unknowns (2 pressures, 4 flow rates, and 3 liquid holdups) except for the last segment where there can be the maximum of 6 unknowns (2 pressures, 2 flow rates, and 2 liquid holdups). This leads to the maximum of $9 \times N - 3$ unknowns for the entire network. In energy calculations there is one unknown temperature at each node. As it is of interest to predict the temperature at the bottomhole of the well, the bottomhole temperature (T_{bh}) is also considered an unknown leading to the total of $2 \times N + 1$ unknowns in the temperature calculations. Equations describing flow through each segment can be constructed using material balance, momentum balance (in the case of pressure), and energy balance (in the case of temperature) equations. The corresponding sets of equations are solved simultaneously using an iterative method (e.g. Newton-Raphson method) for the unknown flow parameters.

When there is more than one phase a multi-phase flow model is required in constructing the governing equations. The model takes into account mass transfer between phases. According to Brill and Mukherjee two types of multi-phase flow models have been used: black-oil model and compositional model [10]. The black-oil model assumes constant composition and the compositional model considers compositional variations in each phase. These models are briefly outlined below.

2.2 Multi-Phase Flow Models

In order to accurately simulate fluid properties for multi-phase flows it is crucial that an appropriate model describing mass transfer between phases is used. Each model has different advantages and drawbacks and each is appropriate for different fluids. As the model determines overall phase behaviour of the hydrocarbon mixtures during depletion, the type of the reservoir from which the fluid is produced is one of the first factors considered in selecting a multi-phase flow model. The classifications of reservoirs are generally based on reservoir temperature and fluid phase envelope.

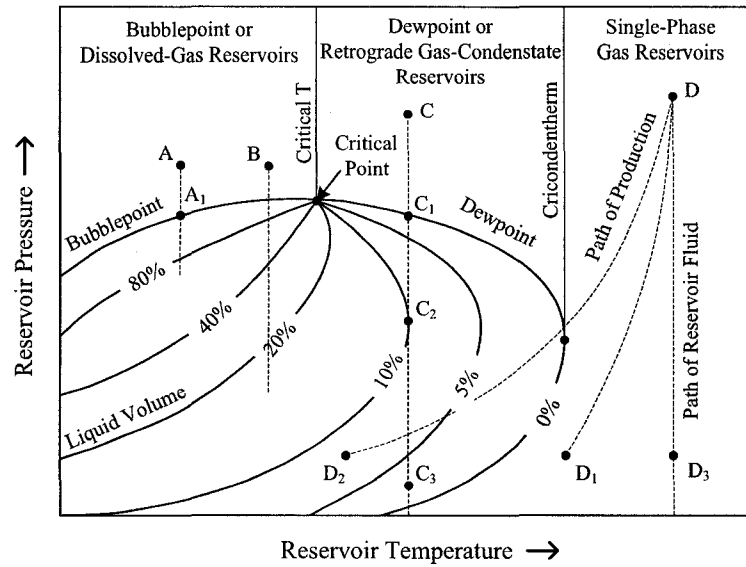


Figure 2-3: Typical types of reservoir [10].

Figure 2-3 shows typical types of reservoirs in oil and gas systems. Black oil reservoirs are the most common types of reservoirs. Temperatures of typical black-oil reservoirs are well below the critical temperature of the hydrocarbon mixture. Point A in Figure 2-3 shows that under reservoir conditions the black-oil-reservoir fluid is initially in the liquid form. When the fluid is produced from the reservoir the pressure is continuously decreased and finally falls below the bubble point pressure (Point A_1) where the gas phase starts to form. Inside the phase envelope there are two phases: oil and gas. Because black-oil fluids are composed of heavier compounds compared with other types of reservoir fluids they have wide phase envelope and broadly-spaced quality (iso-volume) lines. This leads to low shrinkage of the produced oil and less amount of gas produced at the separator [16].

When the reservoir temperature lies closer to the critical point (Point B in Figure 2-3) the reservoir fluid is considered “volatile” and often referred to as “near-critical” oil [16]. As the quality lines of volatile oils are closer near the critical point pressure has significant effects on oil volume fractions below the bubble point. There

is more gas produced at the separator conditions. Moreover, the produced gas is rich and shows retrograde characteristics in the reservoir [16].

Condensate reservoirs have temperatures falling between the critical temperature and the cricondenthem (Point C in Figure 2-3). In the reservoir, when the pressure drops below the dewpoint, liquid drops out as a result of retrograde condensation (Point C₂ has more liquid volume than the zero% liquid at Point C₁). The liquid volume decreases again after Point C₂.

In a gas reservoir the reservoir temperature is above the cricondenthem (Point D in Figure 2-3). There is only gas phase under initial reservoir conditions. After production if the gas remains single phase at the separator conditions (Point D₁) it is considered “dry” gas reservoir. On the other hand if it forms two phases at the separator conditions (Point D₂) the gas is considered a “wet” gas. In the reservoir wet gas does not form condensates because the phase envelope lies below the reservoir temperature.

The above characteristics of reservoir fluids during production determine which multi-phase flow model is more appropriate in describing the fluid behaviour below its bubble or dew point. Once equilibrium is assumed between the two phases (oil and gas) mass transfer between these phases under specified conditions can be described by one of the models described below.

2.2.1 Compositional Models

Compositional model treats fluids as multi-component systems consisting of a specified number of hydrocarbon components. Thorough thermodynamic calculations are used based on compositional material balance to determine how the feed fluid is distributed in the two phases at specified conditions. By knowing phase fractions and compositions, other physical and thermodynamic properties can be calculated [10]. As volatile and condensate fluids exhibit high variations in phase change

and phase compositions during depletion, as well as retrograde behaviour, compositional models are required. Such models use vapour-liquid equilibrium (VLE) or “flash” calculations to describe the mass transfer between the phases more accurately. Details on VLE calculations will be presented in detail later in Chapter 3.

2.2.2 Black-Oil Models

For non-volatile oils the reservoir temperatures are well away from fluids’ critical temperatures and the bubble point pressures are relatively low. The quality lines are also broadly spaced. Phase compositions of the fluids are nearly constant within the two-phase envelope [10]. The evolved gas also does not contain much heavy compounds [16]. In such cases a black-oil model which is based on volumetric material balance provides sufficiently accurate results for less computational effort.

As opposed to the compositional models black-oil models consider a fluid as a system consisting of only two components: oil and gas at a fixed state, e.g. at stock-tank conditions [16]. The liquid phase is described as a black oil or a liquid with associated dissolved gas. The mass transfer between liquid and gas phases is therefore indicated by the ability of the gas to dissolve into the oil phase at specified conditions. This ability is represented by gas solubility parameter (R_s). Because the gas solubility does not reflect retrograde condensation effects, the black-oil models are restricted to non-volatile oils and are not applicable for volatile oils and condensates [10]. Physical properties of the oil system can be calculated based on this parameter along with other black-oil-parameters outlined below.

Black-Oil-Model Parameters

(i) *Gas solubility*

Gas solubility (R_s) or solution gas oil ratio (GOR) is defined as the volume of gas dissolved in one stock tank barrel of oil at Stock tank condition (STC).

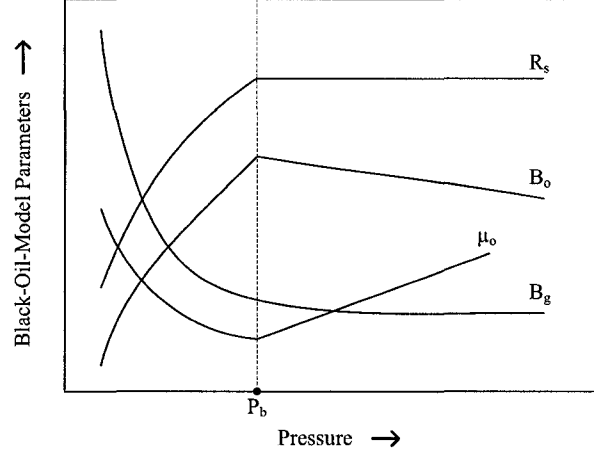


Figure 2-4: Black-oil-model parameters as a function of pressure [30].

It can be expressed as [10]:

$$R_s = \frac{V_{dg}^{STC}}{V_o^{STC}} \quad (2.1)$$

where R_s is the gas solubility [Sm^3/Sm^3]; V_{dg}^{STC} and V_o^{STC} are the volumes at standard conditions of deliberated dissolved gas and oil respectively.

As mentioned before this parameter determines mass transfer between oil and gas phases when there is change in fluid conditions. As shown in Figure 2-4 gas solubility increases with the increased pressure until the oil is saturated at its bubble point pressure (p_b). Above that pressure the gas solubility stays constant and the oil is considered “undersaturated”. The gas solubility parameter can generally be obtained from experiments or correlations.

(ii) ***Oil formation volume factor***

Oil formation volume factor (B_o) is a factor to represent the volume change of the oil phase in response to the change in pressure and temperature conditions. It is defined as the volume that one unit volume stock tank oil occupies at specified temperature and pressure [10]. The value of B_o is always equal to or

greater than 1.0 and can be expressed mathematically as:

$$B_o = \frac{(V_o)_{p,T}}{(V_o)_{STC}} \quad (2.2)$$

where B_o is the oil formation volume factor [m^3/Sm^3]; $(V_o)_{p,T}$ and $(V_o)_{STC}$ are the volumes of oil at reservoir conditions (p, T) and standard conditions respectively.

The change in oil volume with its conditions is caused mainly by the dissolved gas. Other factors affecting oil volume but to a less extent include oil compressibility and oil thermal expansion. These combined effects are illustrated in Figure 2-4. Below the bubble point pressure B_o increases with the pressure as more gas is dissolved in the oil. Above the bubble point pressure there is no more gas dissolving and B_o starts to gradually decrease due to compression. This parameter can be determined using experiments or correlations where different correlations are required for pressures below and above oil's bubble point pressure [10].

(iii) ***Gas formation volume factor***

Gas formation volume factor is defined in a similar manner as the oil formation volume factor but for the gas phase. It is expressed as [10]:

$$B_g = \frac{(V_g)_{p,T}}{(V_g)_{STC}} \quad (2.3)$$

where B_g is the gas formation volume factor [m^3/Sm^3]. $(V_g)_{p,T}$ and $(V_g)_{STC}$ are the volumes of gas at conditions (p, T) and standard conditions respectively.

Figure 2-4 shows gas formation volume factor as a function of pressure where the value monotonically decreases with pressure. This gas formation volume factor can easily be evaluated using the real gas law:

$$pV = ZnRT \quad (2.4)$$

where Z is the fluid compressibility factor, n is the number of mole, and R is the Universal gas constant.

By assuming the compressibility factor of a unity for hydrocarbon gases at standard conditions gas formation volume factor can be calculated from the expression [30]:

$$B_g = Z \frac{p^{STC}}{T^{STC}} \frac{T}{p} \quad (2.5)$$

where p^{STC} and T^{STC} are the standard conditions for pressure and temperature respectively.

Oil Viscosity Calculations

Fluid viscosity manifests itself as a resistance to flow of fluids [7]. The viscosity of black oil is therefore an important parameter in pressure drop calculations. As shown in Figure 2-4 oil viscosity varies with pressure. Below the bubble point oil viscosity decreases with increased pressure as the solution-gas content increases. The value then increases at pressures above the bubble point pressure due to oil compressibility.

Again, oil viscosity can be measured in laboratory or calculated using empirical correlations. In most cases in oil production viscosity are required at various conditions of pressure and temperature. Various correlations were proposed to facilitate determination of viscosity under conditions that are unavailable from PVT analysis [7].

Using correlations, viscosity at specified pressure and temperature can be calculated based on “dead-oil” viscosity [7]. The dead-oil viscosity is the viscosity of no-dissolved-gas oil evaluated at the atmospheric pressure and system temperature (T). Even though determination of the dead-oil viscosity using experiments is preferred there are empirical correlations for calculating this value. One of those is

proposed by Beggs and Robinson [5]:

$$\mu_{od} = 10^{(x-3)} - 0.001 \quad (2.6)$$

where μ_{od} is dead-oil viscosity [Pa·s] and

$$x = (1.8T - 460)^{-1.163} \exp \left(13.108 - \frac{6.591}{\gamma_o} \right) \quad (2.7)$$

where T is the temperature of interest [K] and γ_o is the stock-tank oil gravity.

The oil viscosity at the bubble point pressure (p_b) and temperature (T) can then be calculated using a correlation for saturated oil viscosity. Beggs and Robinson suggested the correlation providing corrections for oil viscosity due to dissolved gas at pressures above the atmospheric pressure [5]:

$$\mu_{os} = A\mu_{od}^B \quad (2.8)$$

where μ_{os} is the saturated-oil viscosity [cp]; μ_{od} is the dead-oil viscosity [cp]; and

$$A = 10.715 (5.615R_s + 100)^{-0.515} \quad (2.9a)$$

$$B = 5.44 (5.615R_s + 150)^{-0.338} \quad (2.9b)$$

where R_s is the gas solubility [Sm³/Sm³] at the conditions of interest.

Finally for viscosity at a pressure above the bubble point pressure another correlation is available that accounts for increased viscosity due to oil compressibility. Vazquez and Beggs proposed [59]:

$$\mu_o = \mu_{os} \left(\frac{p}{p_b} \right)^m \quad (2.10)$$

where μ_o is the oil viscosity at pressures above the bubble point pressure [Pa·s]; p and p_b are the system pressure and the bubble point pressure in kPa respectively.

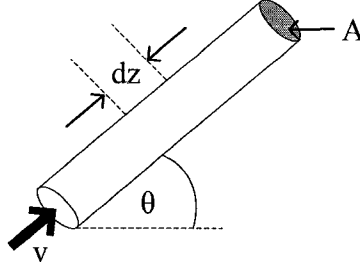


Figure 2-5: Flow control volume.

The parameter m is obtained from:

$$m = 0.263p^{1.187} \exp(-11.513 - 1.302 \times 10^{-5}p) \quad (2.11)$$

2.3 Conservation Equations

As mentioned before the flow behaviour of the fluid in a completed wellbore is determined by both momentum transport and energy transport of the wellbore fluid. For this reason material balance, momentum balance, and energy balance of the flow system are also required in addition to multi-phase flow model presented in the previous section. In this section basic concepts of these conservation equations will be presented. These equations will be referred to frequently throughout this thesis in deriving governing equations for the proposed model.

2.3.1 Mass Balance Equation

Consider a one-dimensional flow system as shown in Figure 2-5. The total mass flux of a fluid flowing through any cross section (A) along the flow direction (z) is always constant. This can be expressed as:

$$\begin{aligned} \dot{m} &= \rho A v \\ &= \text{constant} \end{aligned} \quad (2.12)$$

where \dot{m} is the mass flux, ρ is fluid density, A is the flow cross-sectional area, and v is fluid velocity.

2.3.2 Momentum Balance Equation

Consider equations below for one-dimensional steady-state momentum balance equation [10]. The total pressure drop is composed of pressure drops due to acceleration, friction, and gravity.

$$\dot{m}v = -\Delta p A - \tau_w P \Delta z - \rho g A \Delta z \sin \theta \quad (2.13)$$

or in differential form:

$$\frac{dp}{dz} = -\frac{\dot{m}}{A} \frac{dv}{dz} - \frac{\tau_w P}{A} - \rho g \sin \theta \quad (2.14)$$

where τ_w is the wall shear stress and P is the wetting perimeter.

Consider the frictional term in Equation 2.14. The wall shear stress (τ_w) can be expressed in terms of friction factor. By definition, a friction factor represents the ratio of the wall shear stress to dynamic pressure ($\rho v^2/2$) [30]. This parameter takes into account the effects on the shear stress due to pipe's perimeter (e.g. circular or non-circular), roughness, and Reynolds number. In this research a Moody-type friction factor will be used.

$$\begin{aligned} f &= 4 \frac{\text{Wall shear stress}}{\text{Dynamic pressure}} \\ &= 4 \frac{\tau_w}{\rho v^2/2} \\ &= 8 \frac{\tau_w}{\rho v^2} \end{aligned} \quad (2.15)$$

The frictional term in Equation 2.14 can now be written as:

$$\begin{aligned} \frac{\tau_w P}{A} &= \frac{f \rho v^2}{8} \frac{P}{A} \\ &= \frac{f \rho v^2}{2 D_h} \end{aligned} \quad (2.16)$$

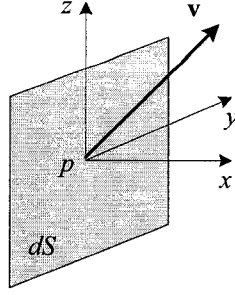


Figure 2-6: Energy transported by convection across a surface element [6].

where D_h is the hydraulic diameter defined as $D_h = 4A/P$.

The momentum balance equation (Equation 2.14) is now expressed in terms of the friction factor as:

$$\frac{dp}{dz} = -\frac{\dot{m}}{A} \frac{dv}{dz} - \frac{f \rho v^2}{2D_h} - \rho g \sin \theta \quad (2.17)$$

2.3.3 Energy Balance Equation

Considering a flow system the net energy flux is the sum of three energy components: energy flowing into or out of the system by conduction, by convection, and work done by molecular motions [6]. The conduction component accounts for the energy transferred by molecular motions and the convection component accounts for the energy transported by bulk fluid motion. The combined energy flux vector is therefore expressed as:

$$\dot{\mathbf{e}} = \dot{\mathbf{e}}_{conv} + \dot{\mathbf{e}}_{cond} + \dot{\mathbf{e}}_w \quad (2.18)$$

where $\dot{\mathbf{e}}$ is combined heat flux; $\dot{\mathbf{e}}_{conv}$, $\dot{\mathbf{e}}_{cond}$, and $\dot{\mathbf{e}}_w$ are energy flux posed by convection, conduction, and molecular work respectively.

Heat Transfer by Convection

To calculate energy transported by the bulk motion of fluid, consider a flow across a surface element dS perpendicular to the x -axis (Figure 2-6). The volumetric

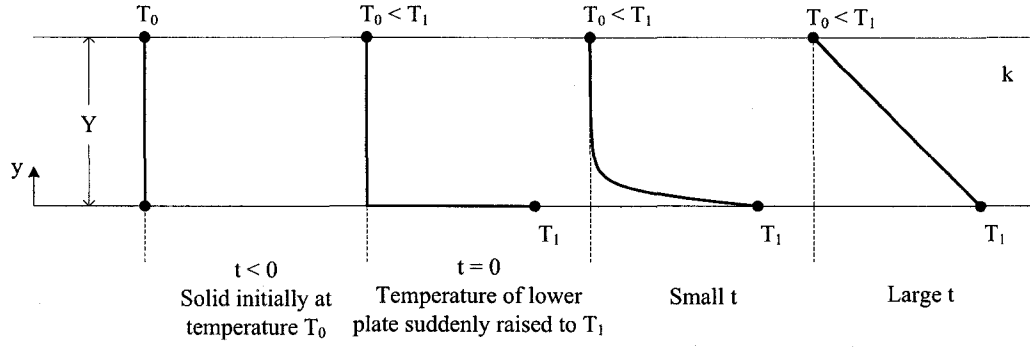


Figure 2-7: Temperature profile by heat conduction of a solid slab between two parallel plates [6].

flow rate across the element is $v_x dS$. The energy flow rate (\dot{E}_x) can be expressed as [6]:

$$\dot{E}_x = \left(\frac{1}{2} \rho v^2 + \rho \hat{U} \right) v_x dS \quad (2.19)$$

where \dot{E}_x is the convective heat flow rate across the surface element perpendicular to the x -axis. The kinetic energy per unit volume ($\frac{1}{2} \rho v^2$) is the expression for $\frac{1}{2} \rho (v_x^2 + v_y^2 + v_z^2)$; and \hat{U} is the specific internal energy.

Three-dimensional convective heat flux which is the sum of heat transported across the surfaces normal to all the axes (x, y, z) is expressed as [6]:

$$\dot{\mathbf{e}}_{conv} = \left(\frac{1}{2} \rho v^2 + \rho \hat{U} \right) \mathbf{v} \quad (2.20)$$

Heat Transfer by Conduction

The energy transported through a medium by molecular motions can be expressed using Fourier's law of heat conduction. The law states that the rate of heat flux is proportional to the temperature difference (ΔT) over a distance (Y) (Figure 2-7) [6].

$$\dot{e} = k \frac{\Delta T}{Y} \quad (2.21)$$

where k is the thermal conductivity of the heat transfer medium.

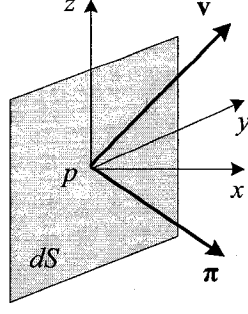


Figure 2-8: Stress vector (π) acting on a surface element creating work done on the plus-side fluid [6].

In differential form the heat flux by conduction is proportional to temperature gradient. Written for the heat flux in the positive y direction (Figure 2-7) [6]:

$$\dot{e}_y = -k \frac{dT}{dy} \quad (2.22)$$

In the case of “isotropic” media where thermal conductivity (k) is of the same value in all directions three-dimensional Fourier’s law is [6]:

$$\dot{e}_{cond} = -k \nabla T \quad (2.23)$$

Work Done by Molecular Motions

In addition to convective and conductive heat transports, work done to the system by molecular motions also contribute to the energy change in the system.

To calculate the amount of molecular work done to the system, consider first the surface element dS perpendicular to the x -axis (Figure 2-8). Force exerted by the fluid on the left side of the surface ($-x$) on the fluid on the right side ($+x$) is $\pi_x dS$. The rate of work done by the left-side fluid on the right-side fluid is therefore [6]:

$$\dot{W}_x = (\pi_x \cdot \mathbf{v}) dS \quad (2.24)$$

where π_x is x -component of the molecular stress tensor and \mathbf{v} is the fluid velocity.

The combined work flux in three dimensions is expressed as:

$$\dot{\mathbf{e}}_w = [\pi \cdot \mathbf{v}] \quad (2.25)$$

Substitute the above three energy components into Equation 2.18 to obtain the combined energy flux [6]:

$$\dot{\mathbf{e}} = \left(\frac{1}{2} \rho v^2 + \rho \hat{U} \right) \mathbf{v} + [\pi \cdot \mathbf{v}] + \dot{\mathbf{e}}_{cond} \quad (2.26)$$

Equation 2.26 is the basic combined energy equation. For practicability purposes the equation can also be written in terms of measurable parameter, enthalpy (H).

The molecular stress tensor (π) can be written as the sum of two stress components: normal stress ($p\delta$) and shear stress (τ) or $\pi = p\delta + \tau$. The term $[\pi \cdot \mathbf{v}]$ becomes $p\mathbf{v} + [\tau \cdot \mathbf{v}]$ [6]. By combining the first term ($p\mathbf{v}$) with the internal energy term Equation 2.26 can now be written in terms of enthalpy as [6]:

$$\begin{aligned} \dot{\mathbf{e}} &= \frac{1}{2} \rho v^2 \mathbf{v} + (\rho \hat{U} \mathbf{v} + p\mathbf{v}) + [\tau \cdot \mathbf{v}] + \dot{\mathbf{e}}_{cond} \\ &= \frac{1}{2} \rho v^2 \mathbf{v} + \rho \left(\hat{U} + \left(\frac{p}{\rho} \right) \right) \mathbf{v} + [\tau \cdot \mathbf{v}] + \dot{\mathbf{e}}_{cond} \\ &= \frac{1}{2} \rho v^2 \mathbf{v} + \rho (\hat{U} + p\hat{V}) \mathbf{v} + [\tau \cdot \mathbf{v}] + \dot{\mathbf{e}}_{cond} \\ &= \left(\frac{1}{2} \rho v^2 + \rho \hat{H} \right) \mathbf{v} + [\tau \cdot \mathbf{v}] + \dot{\mathbf{e}}_{cond} \end{aligned} \quad (2.27)$$

where \hat{H} is the enthalpy per unit mass. The enthalpy can be expressed in terms of measurable thermodynamic parameters [6]:

$$\begin{aligned} d\hat{H} &= \left(\frac{d\hat{H}}{dT} \right)_p dT + \left(\frac{d\hat{H}}{dp} \right)_T dP \\ &= \hat{C}_p dT + \left[\hat{V} - T \left(\frac{\partial \hat{V}}{\partial T} \right)_p \right] dp \end{aligned} \quad (2.28)$$

where \hat{C}_p and \hat{V} are the specific heat capacity and specific volume of the fluid respectively.

Integrating the above equation from a reference state (p°, T°) to the state of interest (p, T) to get [6]:

$$\hat{H} - \hat{H}^\circ = \int_{T^\circ}^T \hat{C}_p dT + \int_{p^\circ}^p \left[\hat{V} - T \left(\frac{\partial \hat{V}}{\partial T} \right)_p \right] dp \quad (2.29)$$

where \hat{H}° is the enthalpy per unit mass at the reference state.

For an ideal gas the last term in Equation 2.29 becomes zero. For a constant-density fluid with a constant heat capacity, the enthalpy can be calculated from [6]:

$$\hat{H} - \hat{H}^\circ = \hat{C}_p (T - T^\circ) + \left(\frac{1}{\rho} \right) (p - p^\circ) \quad (2.30)$$

and for a compressible fluid:

$$\hat{H} - \hat{H}^\circ = \hat{C}_p (T - T^\circ) + \left(\frac{1}{\rho} \right) (1 - \beta T) (p - p^\circ) \quad (2.31)$$

where β is the isobaric thermal expansion coefficient defined as [17]:

$$\beta = -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p = \frac{1}{\hat{V}} \left(\frac{\partial \hat{V}}{\partial T} \right)_p \quad (2.32)$$

2.3.4 Heat Transfer at Solid-Liquid Interface

In order to take into account heat transport between a fluid and its surroundings Newton's law of cooling is used. When there is temperature difference between the flowing fluid and the adjacent solid surface, the heat flux at the solid-liquid interface can be expressed as [6]:

$$q = h (T_0 - T_b) \quad (2.33)$$

where q is the heat flux normal to the interface; T_0 and T_b are the temperatures of the solid at the surface and that of the bulk fluid respectively; and h is the heat transfer coefficient.

The basic mass, momentum, and energy conservation equations presented in this section will be used in constructing governing equations to solve fluid's momentum- and energy-transport problems using the network model. The following sections present in detail how the model is formulated.

2.4 Isothermal Flow Parameter Calculations

As mentioned before the proposed network model is meant for solving two flow problems: momentum and energy transport problems. In the isothermal calculations, only the momentum transport is considered. It is assumed there is no heat transfer in the flow system thus the temperature is constant and equal to the reservoir temperature of the corresponding segment. Based on one-phase network solver developed by Johansen, the governing equations are formulated using the mass and momentum balances [29]. For the model to be applicable for two-phase fluids the black-oil model is incorporated. Flow parameters including pressures, flow rates, and phase fractions can be predicted for the entire well network. The pressures are solved at the nodes representing fluid pressures at different points along the well starting from the reservoir up to the bottomhole of the well. The flow rates and the phase fractions are calculated for all the flows between two consecutive nodes (represented by the network bridges).

In order to formulate the isothermal network model consider first Segment 2 as shown in Figure 2-2(b). Provided reservoir pressure (p_{res}), liquid holdups (α_{res}), and properties of the fluid and the reservoir, there are nine unknowns in Segment 2. These unknowns are pressures (p_4 p_5), flow rates ($q_{inflow(2)}$, $q_{5,8}$, $q_{5,4}$, $q_{4,7}$), and liquid phase fractions ($\alpha_{5,8}$, $\alpha_{5,4}$, $\alpha_{4,7}$). The subscripts indicate nodes or bridges. For example p_4 indicates the pressure at Node 4 and $q_{5,8}$ represents the flow rate in Bridge 5-8 entering the bridge at Node 5 and leaving at Node 8. In order to solve for these unknowns nine equations are required. For each segment there are four

possible types of equations to be written and used to relate all the unknowns. These are:

- Mass balance at nodes for oil or gas phases
- Inflow equation
- Momentum balance equations for annular, annulus-to-tubing, and tubing bridges, and
- Split equation.

Below outlines how each of the above equations are derived.

2.4.1 Material Balance at Nodes

Consider the established well network as in Figure 2–2 there is no accumulation of mass allowed anywhere in the system. Consider for example Segment 2 (Figure 2–2(b)) Two material balance equations can be written for each of the nodes as:

$$\begin{aligned}\sum \dot{m} &= \sum \rho q \\ &= 0\end{aligned}\tag{2.34}$$

where q is the total volumetric flow rate (oil and gas) and is assigned a positive sign when the direction of flow is toward the considered node and a negative sign when it is away from the node.

Oil-phase mass flux entering (or leaving) a node through a particular bridge is $\rho_o^{RC} q \alpha$. Using this expression along with Equation 2.34 the material balance at Node 4 for example is:

$$\rho_{1,4}^{RC} q_{1,4} \alpha_{1,4} + \rho_{5,4}^{RC} q_{5,4} \alpha_{5,4} + \rho_{4,7}^{RC} q_{4,7} \alpha_{4,7} = 0\tag{2.35}$$

where ρ^{RC} 's are liquid density at reservoir conditions.

Oil reservoir density (ρ^{RC}) is related to density at standard conditions (ρ^{STC}) through oil formation volume factor ($\rho^{RC} B_o = \rho^{STC}$). Equation 2.35 can then be

written as:

$$\frac{q_{1,4}\alpha_{1,4}}{B_{o1,4}} + \frac{q_{5,4}\alpha_{5,4}}{B_{o5,4}} + \frac{q_{4,7}\alpha_{4,7}}{B_{o4,7}} = 0 \quad (2.36)$$

Similarly, the material balance equation for the gas phase can be written for each node. The mass flux of the gas phase through a bridge consists of two components: free gas ($\rho_g^{RC} q(1 - \alpha) = \rho_g^{STC} q(1 - \alpha)/B_g$) and the dissolved gas in the liquid phase ($\rho_g^{RC} q_{dg}$). With help from gas solubility parameter (R_s) the latter can be expressed as shown below where q_{dg} is the dissolved gas volumetric flow rate.

$$\begin{aligned} \rho_g^{RC} q_{dg}^{RC} &= \rho_g^{RC} B_g q_{dg}^{STC} \\ &= \rho_g^{RC} B_g q_o^{STC} R_s \\ &= \rho_g^{RC} B_g \frac{q_o^{RC}}{B_o} R_s \\ &= \frac{\rho_g^{STC} q_o^{RC} R_s}{B_o} \end{aligned} \quad (2.37)$$

Gas material balance can now be written as (for Node 4):

$$\begin{aligned} \left(\frac{q_{1,4}(1 - \alpha_{1,4})}{B_{g1,4}} + \frac{q_{1,4}\alpha_{1,4}R_{s1,4}}{B_{o1,4}} \right) &+ \left(\frac{q_{5,4}(1 - \alpha_{5,4})}{B_{g5,4}} + \frac{q_{5,4}\alpha_{5,4}R_{s5,4}}{B_{o5,4}} \right) \\ &+ \left(\frac{q_{4,7}(1 - \alpha_{4,7})}{B_{g4,7}} + \frac{q_{4,7}\alpha_{4,7}R_{s4,7}}{B_{o4,7}} \right) = 0 \end{aligned} \quad (2.38)$$

2.4.2 Inflow Equations

When produced, a reservoir fluid enters the wellbore through perforations. In real situations inflows come in every direction toward the well. However to simplify the problem it can be assumed that the inflows are all in radial directions and perpendicular to the direction of the main well flow. This assumption is acceptable for near wellbore zone as the large pressure gradient into the well in this region leads to dominant perpendicular radial flow [9]. The flow situation is further simplified in the network model. The total radial inflow into a particular segment of the well is represented by using one perpendicular bridge connecting the reservoir node and the

annular node (inlet bridge). The inflows through these inlet bridges are described using a Darcy-based correlation [19].

$$q_{inflow(2)} = PI \left(\frac{k_{ro}}{\mu_o} + \frac{k_{rg}}{\mu_g} \right) (p_{res(2)} - p_5) \quad (2.39)$$

where $p_{res(2)}$ and p_5 are the reservoir pressure the pressure in the annulus (Node 5) of Segment 2 respectively; k_{ro} and k_{rg} are the relative permeabilities and μ_o and μ_g are the viscosities of the oil and gas phases respectively.

Various models for calculating the productivity index (PI) are available in the literature (e.g. Economides et al. [19]) as to account for anisotropy and inflow behaviour. For a homogeneous and isotropic reservoir they reduce to:

$$PI = \frac{2\pi KL}{\ln\left(\frac{r_e}{r_o}\right) + s} \quad (2.40)$$

where r_e is the drainage radius, r_o is the outer radius of the well, and s is the skin factor. The reservoir absolute permeability (K) is the “apparent” permeability which represents the overall permeability of the entire segment. The value may differ for different segment along the well length (Figures 2–2) but within each segment one value is used where reservoir heterogeneity is accounted.

2.4.3 Momentum Balance Equations for Other Bridges

Besides the inlet bridge the other bridges (annular, annulus-to-tubing, tubing) represent what happen inside the wellbore. This section presents how flow equations in internal parts of the well are formulated. In horizontal wells friction is a significant factor contributing to pressure drop in the flow direction. The friction especially important in the case of long wells and high permeability reservoirs [9]. In this research pressure drop through internal bridges of the network model are assumed to result mainly from wall friction. Only frictional term in the momentum balance (Equation 2.17) will be considered. For flows in the annular space and tubing the

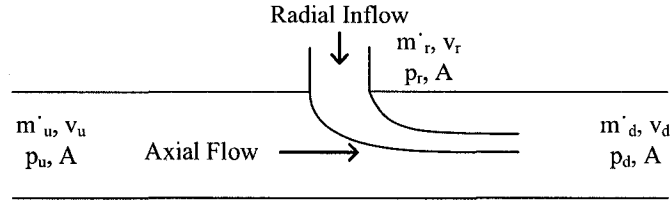


Figure 2-9: Acceleration pressure loss through completion due to radial inflow [9].

relations between flow rates and pressure drops are expressed in terms of Moody-type friction factor. For annulus-to-tubing flows simple pressure drop correlations for flows through nozzles are used.

Due to complexity of well completions it is more difficult to predict pressure drop in completed wellbores. Frictional pressure drop in a completed wellbore is different from that in a solid-wall pipe because of the additional roughness posed by perforations [57]. In addition there are also pressure loss caused by reservoir inflows [3]. Various studies have been conducted to determine appropriate model for pressure drop in completed horizontal wells and are available in the literature.

The inflow fluid radially enters the wellbore through perforations and combines with the main flow in the well as shown in Figure 2-9 [3]. This not only changes the well's boundary layer but also creates pressure loss due to acceleration of the fluid from upstream to downstream. To take into account accelerational pressure drop a momentum balance can be used as discussed by Brekke [8]. In spite of the complication in evaluating wellbore pressure drop a well-defined friction factor is possible to account for all the above pressure drop components.

Asheim et al. presented a model for smooth pipes to calculate an "equivalent" friction factor [3]. This friction factor addresses the additional pressure loss due to active inflows. The total friction factor which is the sum of the wall friction factor and the inflow equivalent friction factor is then used to calculate pressure drop in completed horizontal wells.

The existence of perforations on the sidewall of the well also alters the well's roughness. Su researched effects of perforations on pipe roughness [57]. Different types of friction factors were used to match experimental data in order to obtain the most appropriate expressions of friction factors for perforated wells.

A simple friction factor expression for turbulent flows in smooth pipes was also proposed by Blasius [10]. This friction factor can be used for fluid flows with a Reynolds number ranging from 3,000 to 10^5 .

$$f = \frac{0.3164}{\sqrt[4]{Re_{D_h}}} \quad (2.41)$$

where Re_{D_h} is Reynolds number calculated based on hydraulic diameter, D_h .

Su modified the above equation and develop a “Blasius-type” expression for completed-well friction factor [57]:

$$f = \frac{a}{Re^m} \quad (2.42)$$

The parameters a and m were obtained by matching experimental data and used to account for the pipe roughness caused by perforations. The Reynolds number (Re) was calculated based on the average fluid velocity and properties.

A friction factor expression for rough pipes was proposed by Haaland [23]:

$$f = \frac{1}{\left(-1.8 \log \left(\frac{6.9}{Re} + \frac{\varepsilon_D}{3.7} \right)^{1.11} \right)^2} \quad (2.43)$$

where ε_D is pipe roughness.

Su used Haaland equation to determine “equivalent sand grain roughness” of perforated pipes [57]. It was found that perforated smooth pipes were closer to blank smooth pipes than completely rough pipes. The friction factor data all fell on the smoother side in the transition region of friction factor plots.

For simplicity purposes the original Blasius's friction factor (Equation 2.41) is used in this research to formulate pressure drop equations for flows in the annulus and tubing as presented below. A more-refined value of friction factor can be used as necessary in the future development of the model.

Annular Flow Equations

The momentum balance equation for flows in the annular space is:

$$\frac{dp}{dz} = \frac{f\rho v^2}{2D_h} \quad (2.44)$$

Based on the above equation the flow equation for Segment 2's annular bridge can be written as an example:

$$p_5 - p_8 = \frac{q_{5,8}^2 L f \bar{\rho}}{2D_h A^2} \quad (2.45)$$

where L is the length of Segment 2, A is the annular cross-sectional area, and $\bar{\rho}$ is the average density of the two-phase fluid.

Tubing Flow Equations

Based on Equation 2.44 an equation relating flow and pressure for the fluid in well tubing can be expressed as:

$$p_4 - p_7 = \frac{q_{4,7}^2 L f \bar{\rho}}{2DA^2} \quad (2.46)$$

where D is the tubing diameter. Other variables are defined similarly to those in Equation 2.45.

Annulus-to-Tubing Flow Equations

For flows from the annulus into the tubing pressure loss is not only posed by wall friction but also by convergence of the flow into small cross-sectional area of slotted liners or valve openings. To enter a slot opening uniform radial inflows converge and accelerate as shown in Figure 2-10. This causes pressure drop due to acceleration of fluid [32]. For flows through slots this effect may be accounted for by using an equivalent skin or "slot" factor that can be used in the inflow equations (Equation

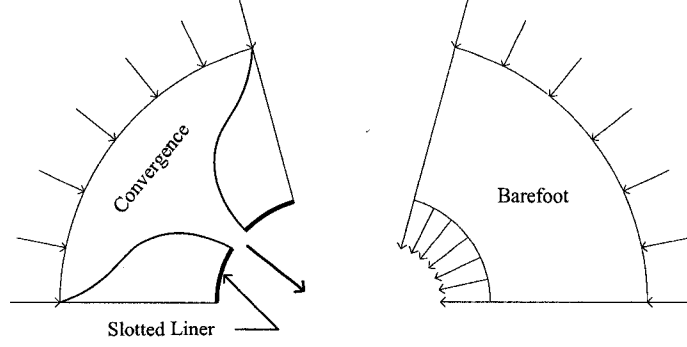


Figure 2-10: Radial flow convergence into a slot opening [32].

2.39) as suggested by Kaiser [32]. In this study however a discharge coefficient (c) is used to form a simple pressure drop equation similar to the case of flows through nozzles [29]:

$$p_{ann} - p_{tub} = c\rho v^2 \quad (2.47)$$

The annular-to tubing flow equations can now be written as:

$$p_5 - p_4 = \frac{q_{5,4}^2 c \bar{\rho}}{A^2} \quad (2.48)$$

where A is the total slot cross-sectional area calculated from $A = \omega LHW$ with ω as the number of slots per length, H as the slot height, and W as the slot width.

2.4.4 Split Equations

In order to match the number of unknowns and equations another flow condition must be established. Where there is a split of flow in the flow direction, the liquid-phase volume fractions of the two streams are assumed equal. For example at Node 5 in Figure 2-2(b) the annular flow leaving Segment 2 ($q_{2,5}$) splits into two streams. A portion of the fluid flows into the annular space of Segment 3 ($q_{5,8}$) and the rest through slots into the tubing ($q_{5,4}$). The liquid volume fractions of these two streams are assumed equal or mathematically expressed as:

$$\alpha_{5,4} = \alpha_{5,8} \quad (2.49)$$

If the flow directions are known before the simulations it is possible to specify where these splits of flow are located in the network and where the split equations are required.

Similar equations to those presented above can be established for all of the N segments in the network. The maximum number of equations is $9 \times N - 3$ equations (only six equations can be written for the last segment). These equations containing the same number of unknowns the governing equations can be solved simultaneously using a Newton-Raphson iterative method provided that required boundary conditions are specified.

2.4.5 Boundary Conditions

This problem requires boundary conditions at the inlet and outlet nodes. Reservoir pressures (p_{res}) are used as the boundary conditions at the reservoir inlets. Provided the inlet boundary conditions, pressure or flow rate can be specified at the outlet (heel of well) as the outlet boundary condition. In this research the bottomhole pressure (p_{bh}) at the heel is used as the outlet boundary condition.

It is worth it to mention here that for simplicity the flow directions of the fluid are assumed to be known before the simulation. Moreover in order for the model to simulate wells in different completion scenarios, some flexibility to modify the network structure is added by using “bridge indices”. These indices are used to indicate flow directions of the fluid. The fluid flowing axially toward the heel of well or radially toward the well tubing are assigned a positive bridge index of +1. The flows going in the opposite directions are given a negative bridge index of -1. However only annular bridges are allowed to have a negative bridge index. As mentioned earlier the tubing flow is always toward the heel of well. Similarly the flows in inlet and annulus-to-tubing bridges are always toward the well tubing.

Only the fluid in the annulus can flow toward the toe of well and thus allowed to have a bridge index of -1 .

Bridge indices are also used to indicate whether there is flow between the particular nodes. For example when a part of the well is packed off there is a discontinuity in the flow in the annulus. In this case the annular bridge occupied by the packing-off material is assigned a bridge index of zero to represent the discontinuity of the flow. This zero bridge index is also applied when the well is partially perforated. Only the inlet bridges of the segments that have connections with the reservoir are assigned the bridge indices of $+1$, the rest are assigned zero bridge indices.

2.5 Temperature Calculations

In oil production processes there are significant heat transport mechanisms which lead to changes in fluid temperature as it flows through the wellbore and in turn changes in fluid properties. For the above reason assuming an isothermal production process might not provide sufficiently accurate results to the fluid flow problems especially when phase behaviour of the fluid is an important issue. In this section how non-isothermal effects can be incorporated into the isothermal network model will be presented. The energy transport in wellbores will be first outlined. The governing equations used to solve for unknown temperatures of the entire well network will then be presented.

2.5.1 Heat Transfer in Wellbore

When produced the oil initially has approximately the same temperature as that of the reservoir. As it flows through wellbore its temperature changes as a result of heat transport. The difference in temperatures between the fluid and the surrounded formation or well completions leads to heat conduction through solid surroundings. Friction in the wellbore is another factor that may cause alteration

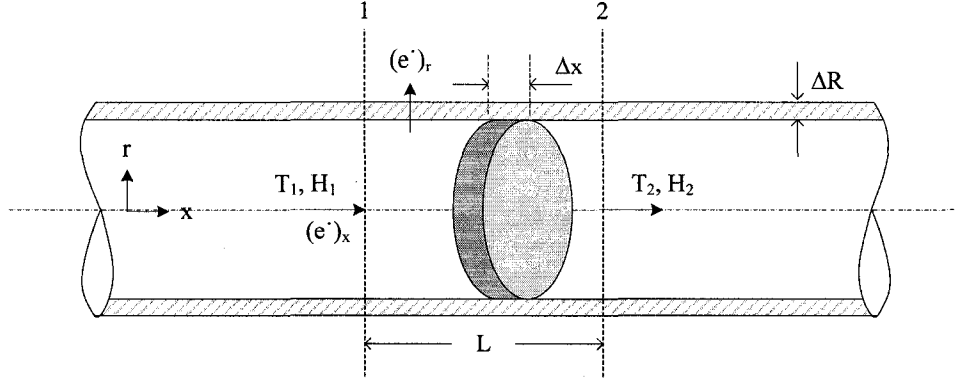


Figure 2-11: Heat transfer in one-dimensional flow.

in the fluid temperature [33]. Particularly in the case of very large drawdown where fluid experiences a wide range of pressure drop, its temperature may increase or decrease as a result of expansion. This effect is also known as Joule-Thompson (J-T) effect [24]. This expanding effect causes the temperature to increase in the case of liquids but on the other hand may cause a temperature drop in the case of gas flows depending on the flow conditions.

To account for these energy transports, consider the combined energy flux equation (Equation 2.18). For fully developed flow changes in enthalpy flux dominate [6]. The terms involving kinetic energy ($\frac{1}{2}\rho v^2 \mathbf{v}$) and shear stress ($\boldsymbol{\tau} \cdot \mathbf{v}$) can therefore be neglected. A study by Dawkrajai et al. confirmed the negligible effects of the kinetic term in modeling temperature of horizontal wells [17]. Considering Figure 2-11 for one-dimensional fluid flow in circular pipe axial heat conduction term ($(\dot{e}_{cond})_x$) can also be neglected relative to the enthalpy term in the same direction ($\rho \hat{H} v_x$). In this case the net energy rate at a point in the system written for the axial and radial axes, assuming that the fluid density is constant.

For the axial direction:

$$\dot{E}_x = \rho q \hat{H} \quad (2.50a)$$

For the radial direction:

$$\dot{E}_r = \rho q \hat{H} + Q_r \quad (2.50b)$$

The first equation is the energy flux in the flow direction (x) and the second addresses heat exchange between fluid and the reservoir. The first term in Equation 2.50b accounts for the heat transfer associated with radial inflows. The radial heat transfer at the fluid-solid interface is represented by Q_r . Assuming a constant heat capacity (\hat{C}_p), the above equations can be derived further.

For the axial direction,

$$\begin{aligned}\dot{E}_x &= \rho q (\hat{H} - \hat{H}^\circ) \\ &= \rho q \hat{C}_p (T - T^\circ) + q (1 - \beta T) (p - p^\circ)\end{aligned}\quad (2.51)$$

where \hat{H}° is the specific enthalpy at the reference state.

Introducing Joule-Thompson coefficient defined as [17]:

$$K_{JT} = \frac{\beta T - 1}{\rho \hat{C}_p} \quad (2.52)$$

Equation 2.51 becomes:

$$\dot{E}_x = \rho q \hat{C}_p (T - T^\circ) - \rho q \hat{C}_p K_{JT} (p - p^\circ) \quad (2.53)$$

Similarly, the equation for the radial heat transfer can be derived further as:

$$\begin{aligned}\dot{E}_r &= (\rho q \hat{H})_r + Q_r \\ &= \rho q \hat{C}_p (T - T^\circ) + Q_r\end{aligned}\quad (2.54)$$

By assuming that the pressure at the outer surface of the well is equal to the pressure of the fluid on the inside of the well, the Joule-Thompson effect in the radial direction is neglected. The radial heat transfer (Q_r) can be expressed using an overall heat transfer coefficient (U) which represents the heat transfer at the fluid-solid interface by convected fluid and conduction through the surrounding materials (well casing and cement for example).

$$(Q_r) = A_r U (T_o - T_b) \quad (2.55)$$

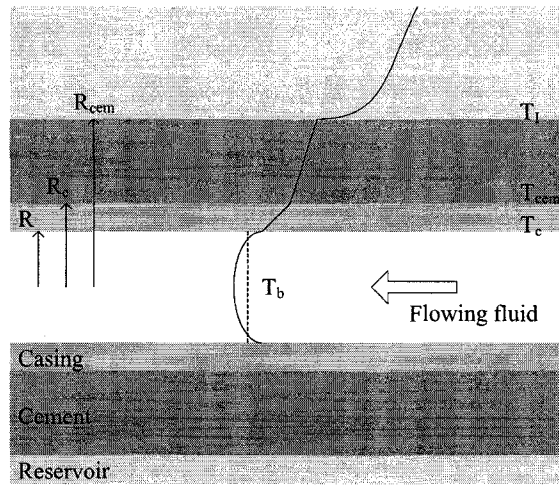


Figure 2-12: Overall heat transfer between fluid and surroundings (adapted from [17].)

where A_r is the area for radial heat transfer; T_o and T_b are the temperature at the outer surface of the solid surroundings and the temperature of the bulk fluid on the inside respectively. Determination of the overall heat transfer coefficient (U) is as described below.

2.5.2 Overall Heat Transfer Coefficients

Heat transfer between wellbore fluid and its surroundings involves complex energy transport mechanisms including radial conductive heat transfer through a series of heat transfer media and heat transport between the solid wall and the flowing fluid. The overall heat transfer coefficient is therefore used for simplicity purposes to represent the combined effect of the aforementioned heat transports. Typically the coefficient is derived by considering a steady-state system with equal heat flows through each layer of the interfaces. For example Dawkrajai et al. derived the overall heat transfer coefficient in cased and cemented wells [17]. Consider Figure 2-12. Assuming a constant thermal conductivity for each heat transfer medium the heat flows through each layer of the material were obtained [17]:

for the casing

$$\begin{aligned}
Q_c &= -2\pi (1 - \gamma) k_c \left. \frac{dT}{dr} \right|_{r=R} \\
&= 2\pi (1 - \gamma) k_c \frac{T_c - T_{cem}}{\ln \left(\frac{R_c}{R} \right)}
\end{aligned} \tag{2.56a}$$

for the cement

$$\begin{aligned}
Q_{cem} &= -2\pi (1 - \gamma) k_{cem} \left. \frac{dT}{dr} \right|_{r=R_c} \\
&= 2\pi (1 - \gamma) k_{cem} \frac{T_{cem} - T_I}{\ln \left(\frac{R_{cem}}{R_c} \right)}
\end{aligned} \tag{2.56b}$$

and for the flowing fluid

$$Q_{fl} = -2\pi (1 - \gamma) h (T_c - T_b) \tag{2.56c}$$

where the subscripts c , cem , and fl indicate casing, cement, and fluid respectively. The temperature of the inflow fluid and the average temperature of the bulk fluid inside the well are T_I and T_b respectively. The thermal conductivity of each material is represented in the equation by the parameter k and h is the heat transfer coefficient for fluid-solid interface. In the above equation the heat transfer was only considered through the area without radial fluid inflow [17]. Thus the radial solid-fluid heat transfer was only evaluated for the areas impermeable to fluid flow. The fraction of the total opening areas (e.g. slot openings or perforated parts of the well) was γ and the heat transfer between the fluid and the surroundings was therefore evaluated for the $(1-\gamma)$ portion of the total circumferential surface area.

At steady state all the heat flows through each layer could be equated [17].

$$Q_c = Q_{cem} = Q_{fl} \equiv Q \tag{2.57}$$

This led to

$$T_b - T_I = \frac{Q}{2\pi(1-\gamma)} \left[\frac{\ln\left(\frac{R_c}{R}\right)}{k_c} + \frac{\ln\left(\frac{R_{cem}}{R_c}\right)}{k_{cem}} + \frac{1}{R\alpha} \right] \quad (2.58)$$

The overall heat transfer coefficient was therefore

$$U = \frac{Q}{(T_b - T_I) 2\pi R (1 - \gamma)} = \left[\frac{R \ln\left(\frac{R_c}{R}\right)}{k_c} + \frac{R \ln\left(\frac{R_{cem}}{R_c}\right)}{k_{cem}} + \frac{1}{\alpha} \right] \quad (2.59)$$

In order to calculate the heat transfer coefficient (h) in laminar flow, Dawkrajai et al. used [17]:

$$h = 3.656 \frac{k_{fl}}{2R} \quad (2.60)$$

Hasan and Kabir suggested calculating the heat transfer coefficient (h) for forced-convective, turbulent-flow fluids in circular pipes or wells using the Nusselt number expression [24]:

$$Nu \equiv \frac{hd}{k} = 0.023 (Re)^{0.8} (Pr)^{0.33} \left(\frac{\mu}{\mu_w} \right)^{0.14} \quad (2.61)$$

where Pr is the Prandtl number ($Pr = C_p \mu / k$), Re is the Reynolds number, d is the pipe or well diameter, k is the fluid thermal conductivity, and μ_w is the fluid viscosity at wall temperature. According to Hasan and Kabir the term (μ/μ_w) is only significant in the case of very high viscosity fluid but becomes negligible in most of other cases [24].

In this research Equation 2.59 is used to calculate the overall heat transfer coefficients for fluids in annular and tubing spaces which will be described in more detail in Chapter 5. Fluid heat transfer coefficients are calculated using Equation 2.60 or Equation 2.61 depending on Reynolds number of the fluid.

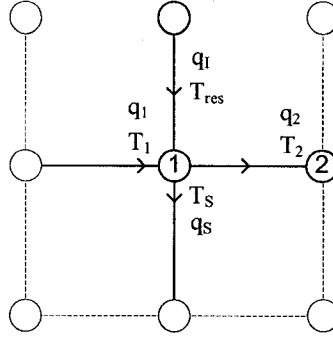


Figure 2-13: Energy balance at a node.

2.5.3 Energy Balance at Nodes

Similar to the isothermal calculations energy balance equation is established at each node in the network. The black-oil parameters are used to account for the two-phase behaviour of the fluid. These equations can be solved for temperatures at each node in the well network. For a particular segment, two energy balance equations can be written at the annular and tubing nodes. These equations relate heat transfer components entering and leaving each point (node) in the well.

In the network model bridges perpendicular to the main flow direction (inlet and annulus-to-tubing bridges) are used to represent the radial heat flows in each particular segment. The energy gain (or loss) through these perpendicular bridges therefore contribute to the temperature change within the particular segment. Consider the annular node (Node 1) of the segment in Figure 2-13. It is acceptable to assume that the fluid enters the segment (q_I) with the temperature of T_1 . Before the fluid leaves the segment, there are energy transports due to the fluid entering the segment from the reservoir (q_I) and the fluid leaving the segment through slots (q_S). The temperature of these two streams are assumed to be T_{res} for the inflow and T_S for the slot flow. The heat transfer occurring within the segment (e.g. mixing with the inflow stream and heat loss through the solid surroundings) causes the fluid temperature to change from T_1 to T_2 when it leaves the segment at Node 2.

The energy balance at Node 1 can now be written as:

$$\begin{aligned}
\sum \dot{E} &= \dot{E}_1 - \dot{E}_2 + \dot{E}_I - \dot{E}_S \\
&= (\rho q \hat{H})_1 - (\rho q \hat{H})_2 + [(\rho q \hat{H})_I + (Q_r)_I] \\
&\quad - [(\rho q \hat{H})_S + (Q_r)_S] \\
&= 0
\end{aligned} \tag{2.62}$$

where the subscripts 1, 2, I , and S denote properties of the flow upstream of Node 1, the flow downstream of Node 1, reservoir inflow, and slot flow, respectively. From material balance we have $(\rho q)_2 = (\rho q)_1 + (\rho q)_I - (\rho q)_S$, thus the above equation can be written as:

$$\begin{aligned}
\sum \dot{E} &= [\rho_2 q_2 \hat{H}_1 - \rho_I q_I \hat{H}_1 + \rho_S q_S \hat{H}_1] - (\rho q \hat{H})_2 \\
&\quad + [(\rho q \hat{H})_I + (Q_r)_I] - [(\rho q \hat{H})_S + (Q_r)_S] \\
&= -\rho_2 q_2 (\hat{H}_2 - \hat{H}_1) - \rho_I q_I (\hat{H}_1 - \hat{H}_I) \\
&\quad + \rho_S q_S (\hat{H}_1 - \hat{H}_S) + [(Q_r)_I + (Q_r)_S]
\end{aligned} \tag{2.63}$$

Assume that the fluid leaves the segment through slots with the conditions of the fluid at Node 1 ($\hat{H}_1 = \hat{H}_S$). In addition, for simplicity purposes, we assume that the temperature at the outer surface of the tubing wall is equal to that of the liquid in the annulus ($(Q_r)_S = 0$). Thus heat the radial conductive heat transfer for the annular fluid is only considered at the well-reservoir interface. As mentioned earlier the pressure at the outside surface of the well casing (or of the tubing wall) is assumed equal to the pressure of the fluid on the inside. Thus the Joule-Thompson terms drop out when the energy rates in the radial direction are considered. The reservoir temperature (T_{res}) is also assumed at the outer surface of the casing and equal to the temperature of the inflow fluid. Based on the above assumptions the

Equation 2.63 reduces to:

$$\begin{aligned}
\sum \dot{E} &= -\rho_2 q_2 (\hat{H}_2 - \hat{H}_1) - \rho_I q_I (\hat{H}_1 - \hat{H}_I) + (Q_r)_I \\
&= -\rho_2 q_2 [\hat{C}_p (T_2 - T_1) - \hat{C}_p K_{JT} (p_2 - p_1)] \\
&\quad - \rho_I q_I \hat{C}_p (T_1 - T_{res}) + (Q_r)_I
\end{aligned} \tag{2.64}$$

The heat capacity (\hat{C}_p) and the Joule-Thompson coefficient (K_{JT}) can either be evaluated separately for each of the bridge or determined at the averaged wellbore pressure and temperature.

The sign convention of the the enthalpy terms is according to the fluid flow direction. It is assigned a positive sign when the fluid flows toward the considered node and a negative sign when the flow is away from the node. According to Equation 2.55 the heat conduction term is negative when the the fluid loses heat to its surroundings ($T_1 < T_{res}$).

To derive the equation further for two-phase flows, similar to the momentum balance, the energy balance equations can be expressed in terms of black-oil-model parameters. For Bridge 5-8 for example, the enthalpy term can be written as the sum of the enthalpy change in oil and gas phases:

$$\begin{aligned}
\rho q \hat{C}_p (T_8 - T_5) &= \frac{\rho_{5,8}^o q_{5,8} \alpha_{5,8}}{B_{o5,8}} \hat{C}_p^o (T_8 - T_5) \\
&\quad + \frac{\rho_{5,8}^g q_{5,8} (1 - \alpha_{5,8})}{B_{g5,8}} \hat{C}_p^g (T_8 - T_5) \\
&\quad + \frac{\rho_{5,8}^g q_{5,8} \alpha_{5,8} R_{s5,8}}{B_{o5,8}} \hat{C}_p^g (T_8 - T_5)
\end{aligned} \tag{2.65}$$

where \hat{C}_p^o and \hat{C}_p^g are the heat capacity of oil and gas phases respectively; and ρ^o and ρ^g are the densities of the oil and gas phases at standard conditions. These values are evaluated at the averaged pressure and reservoir temperature.

Energy balance equations similar to Equation 2.64 can be written for all of the N segments to form $2 \times N$ equations. These are to be solved using a Newton-Raphson

iterative method for $2 \times N$ unknown temperatures resulting in a temperature profile of the entire network. In order to solve the equations boundary conditions are required.

2.5.4 Boundary Conditions

As mentioned before that there are $2 \times N + 1$ unknown temperatures including the temperature at the bottomhole (T_{bh}), one temperature at a node needs to be specified and used as a boundary condition in order to solve the equations. It is appropriate here to assume that the temperature at the toe of well where the reservoir fluid first enters the well is equal to the reservoir temperature. Thus the temperature at the annular node of the first segment is used as a boundary condition in this model and equal to the reservoir temperature of Segment 1 ($T_{res(1)}$). The $2 \times N$ unknown temperatures are then assigned starting at Node 1 (tubing node of Segment 1) up to the last node at the bottomhole of the well. The reservoir temperatures (T_{res} 's) are used as the other required boundary conditions.

2.6 Network Model Integration

In order to solve both momentum and heat transfer problem together the isothermal and the temperature calculations as presented in the previous sections are integrated. The momentum and energy transports in the well system are considered in a stepwise manner as shown in Figure 2-14. Firstly temperature of the fluid is assumed constant and set equal to the reservoir temperature of the corresponding segment. Fluid properties required in the simulation are generated at the local fluid pressure (at the nodes) and the averaged reservoir temperature. As will be presented in Chapter 5 there are two types of fluid properties required in the proposed model: black-oil-model properties and characterized-oil properties. In the network model only the black-oil properties including B_o , B_g , R_s , and density are

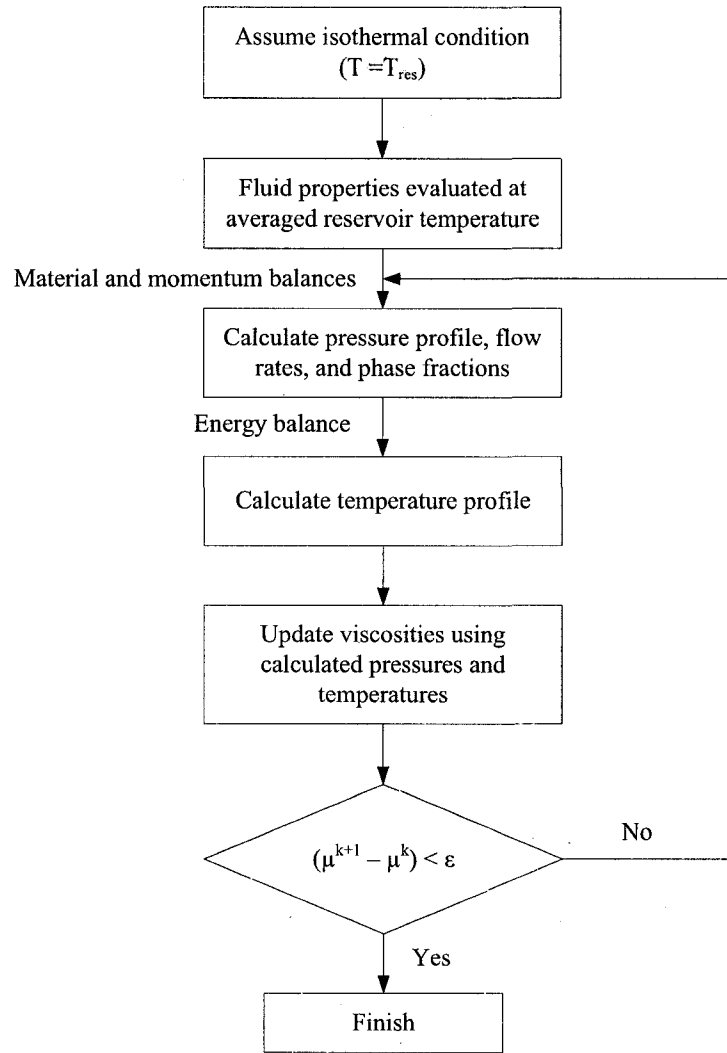


Figure 2-14: Network model schematic.

required. The network pressure profile is then calculated using the material and momentum balance equations provided that the reservoir pressure and the bottom-hole pressure are assigned as the inlet and outlet boundary conditions respectively. Other flow parameters including flow rates and phase fractions are also obtained from this part of the calculations. Provided the calculated flow parameters, the network temperature can then be calculated using the energy balance. In this case the reservoir temperature and the temperature at the toe of well are used as the boundary conditions.

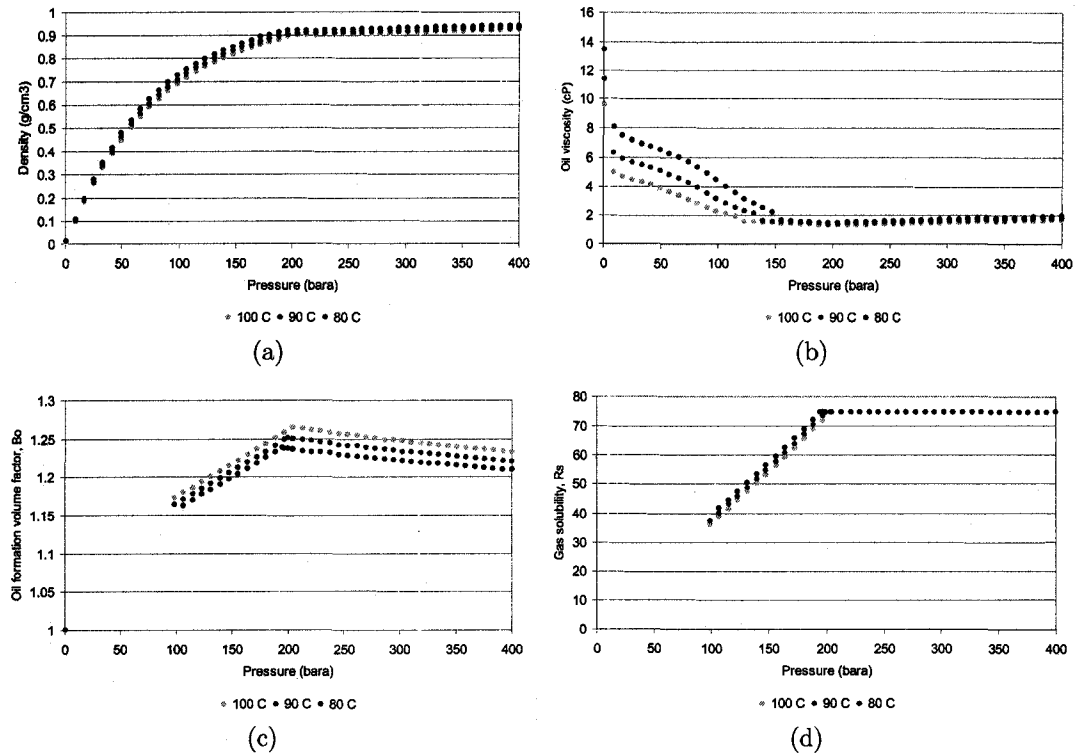


Figure 2-15: Variation of fluid properties with pressure and temperature. (a) density. (b) oil viscosity. (c) oil formation volume factor. (d) gas solubility.

At this point fluid properties can be recalculated using the calculated temperature. Based on the new property values the network pressures are then recalculated. This process is repeated until the property values converge providing the final results of fluid conditions for the entire network.

Figure 2-15 shows black-oil fluid properties at different temperature of the example fluid in Appendix A. It can be seen that there are relatively high variations in oil viscosity with temperature for pressures below the saturation point. However for undersaturated fluids there are little change in all fluid properties with temperature. In addition, provided a small change in temperature relative to pressure in production systems, the temperature-independent assumption for other black-oil properties are acceptable. In this research only oil viscosity recalculation was therefore selected to be used in the network model. Oil viscosity is a function of temperature and is

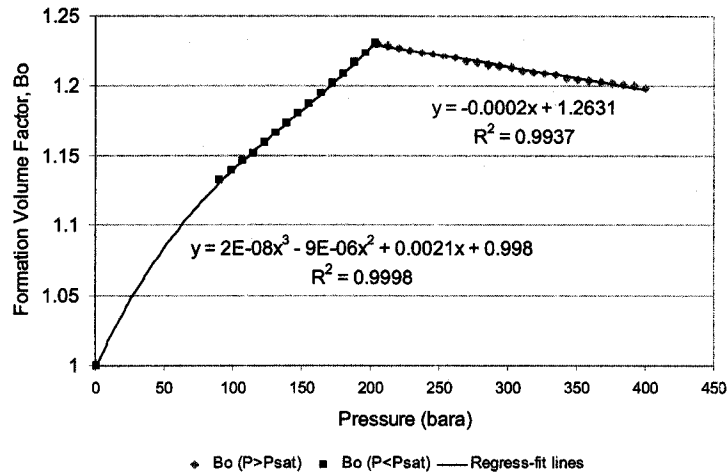


Figure 2-16: Curve fitting of pre-generated property values.

an important factor in describing resistance to liquid flow. Therefore, this property affects calculations in the network model in many ways including prediction of inflow rate. Updates of oil viscosity can be achieved by using empirical correlations. In this research the dead-oil viscosities at different temperature are generated from a fluid characterization software based on available PVT data. Provided the dead-oil viscosities the saturated- and undersaturated-oil viscosities can then be calculated using Equation 2.8 and Equation 2.10 respectively. Using the correlations allows fluid viscosities to be reevaluated according to the change in network temperature.

As opposed to oil viscosity other fluid properties are assumed to be pressure-dependent only. As small change in temperature (relative to the change in pressure) is expected in the well without thermal treatment the values are at the averaged reservoir temperature. These black-oil properties are generated prior to the simulations using an equation of state. Properties generated for various pressures can be translated to mathematical expressions by curve-fitting. As an example Figure 2-16 shows two expressions for oil formation volume factor (B_o) as a function of pressure: one for pressures below and the other for those above the saturation pressure. In the figure, the data points are the values of B_o generated using a thermodynamic

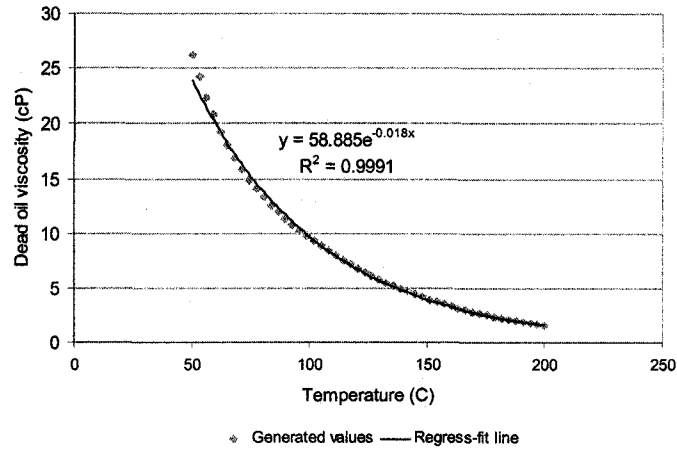


Figure 2-17: Dead oil viscosity as a function of temperature.

package and the lines were regressed-fit to the data using MS Excel. The R-squared values (R^2 's) indicating the reliability of estimated trendline values are also presented in the figure. The line is best fit to the data when the value of R^2 is close to one. By using these fluid-specific expressions, the fluid properties are allowed to be updated during the simulation in response to the change in pressure. In addition, in the case where fluid's PVT data are available the calculations of fluid properties based on a tuned equation of state helps increase the model accuracy compared with the use of fluid property correlations. Relations between other black-oil properties (B_o , B_g , R_s) and pressure can all be translated into mathematical expressions using the same approach.

Even though correlations are used to calculate oil viscosity in the proposed model, the dead-oil viscosity used in the correlations can be calculated at various temperatures based on a corresponding states principle [12]. An expression of the values in terms of temperature can then be developed in a similar manner as that for the pressure-dependent properties described above. In this research the values of the dead-oil viscosity are generated using a thermodynamic package [12] at various temperatures and used to develop a mathematical expression as shown in Figure 2-17. This way the dead-oil viscosity can be re-calculated during the simulations

according to the change in fluid temperature. Based on the calculated dead-oil viscosity, oil viscosity can be calculated.

Similar to the oil viscosity updates of other properties (e.g. density) in response to the change in temperature are also possible. The viscosity update was merely included to demonstrate the capability of the model to be used when higher temperature variation is involved. In order to recalculate a property in the network model, an accurate property correlation is required. In this research fluid properties (besides oil viscosity) are considered a function of pressure only and calculated using an equation of state which is believed to provide more accurate, fluid-specific results in property calculations. Whether or not to include temperature effect depends on many factors including the degree of temperature variation in the well and the temperature-sensitivity of each particular fluid property.

2.7 Network Model Verification

In this section the proposed model is validated. Errors associated with the network model caused by discretization of the well into segments was evaluated. In addition the results from the proposed model were compared with published literature.

2.7.1 Inaccuracy Caused by Discretization

The network model evaluates fluid behaviour based on discretizing the well into a finite number of segments. The approach is therefore associated with discretization errors. The smaller the segment length the closer the results are to those obtained from analytical solutions. In this section the magnitude of the error was investigated by simulating a 2000-meter well using two different discretization schemes. The first is that the well was divided equally into 100 segments of 20 meters long and the other into 200 segments of 10 meters long. Both schemes were used to predict pressure,

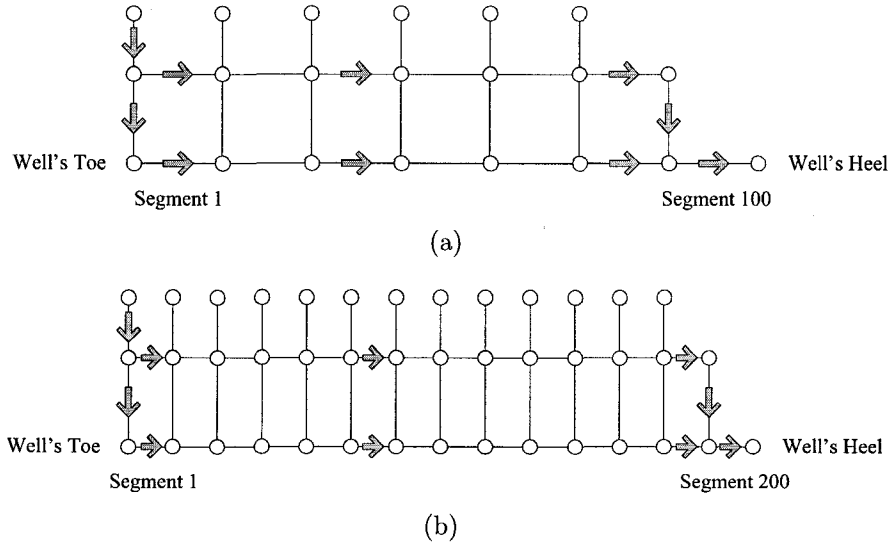


Figure 2-18: Well network used in investigating errors caused by well discretization. (a) well network divided into 100 segments. (b) well network divided into 200 segments.

flow rate, and temperature in the well using the proposed network model presented earlier. In both cases complete well network structures were used (i.e. no bridges were removed) as illustrated in Figure 2-18. Other conditions of the well systems were identical as listed in Table 2-1.

The results from the simulations (Figure 2-19 to Figure 2-21) showed that all the flow parameters from both schemes were comparable. The errors accumulated toward the heel of well but were small relative to the variations of the flow parameters for the entire length of the well. The total pressure drops in tubing, production rates, and the total changes in temperature of the fluid in tubing from the two schemes are compared in Table 2-2. Consider the production rates predicted by the two discretization schemes. When the well was divided into 100 segments the predicted production rate was 11,452 m³/d compared with 11,966 m³/d in the 200-segment scheme (-4.30% error). The difference was partly caused by the fact that, based on the network model structure, the well was perforated until the second last segment (Segment $N - 1$). This led to 10 meters longer of perforated section in the

Table 2-1: Well basic characteristics for discretization error analysis.

Property	Value
Well length (m)	2000
Reservoir pressure (bara)	368
Pressure at heel (bara)	357.5
Reservoir temperature (°C)	100
Permeability (Darcy)	1
Near-wellbore skin factor	2
Drainage radius (m)	20
Oil saturation	1
Tubing diameter (m)	0.127
Well outside diameter (m)	0.167
Discharge coefficient for flow through slots ($\text{Pa} \cdot (\text{kg}/\text{m}^3)^{-1} \cdot (\text{m}/\text{s})^{-2}$)	10
Steel tubing thermal conductivity ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	50
Outer casing thermal conductivity ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	11.99
Cement thermal conductivity ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	6.95
Joule-Thompson coefficient (°C/bar)	-0.0321
Fluid heat capacity ($\text{J}/\text{mol } ^\circ\text{C}$)	368.93

Table 2-2: Results from two different discretization schemes used in model verification.

Parameters	100 segments	200 segments
Total pressure drop (bara)	9.878	9.870
Production rate (m^3/d)	11,452	11,966
Total temperature change (°C)	0.1607	0.1654

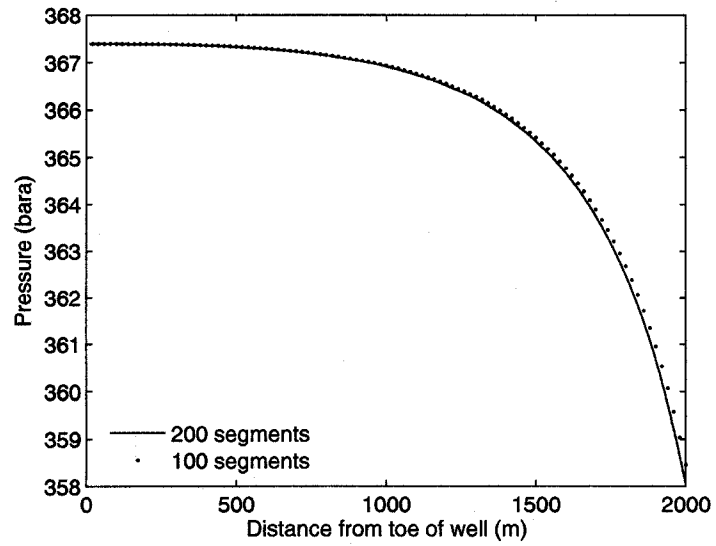


Figure 2-19: Pressure profiles for different discretization schemes.

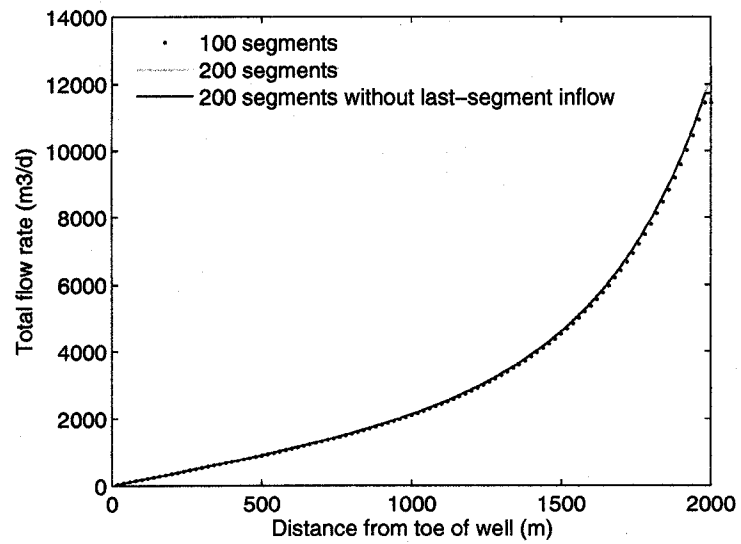


Figure 2-20: Total well flow profiles for different discretization schemes.

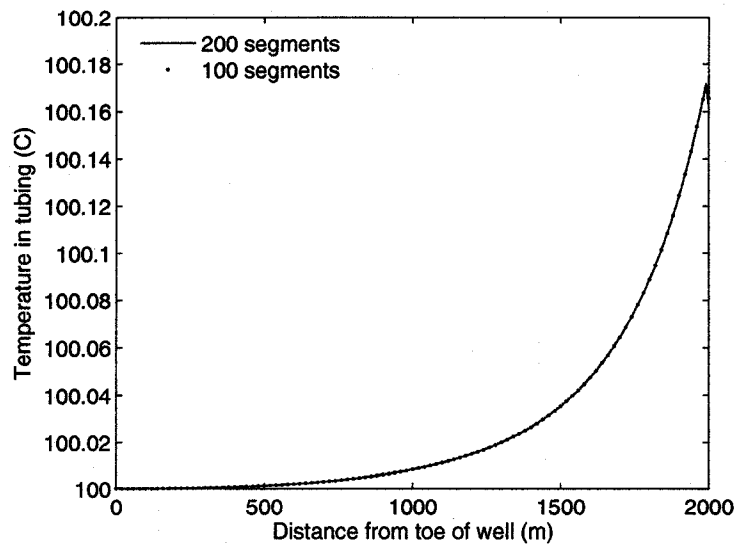


Figure 2-21: Temperature profiles in tubing for different discretization schemes.

200-segment scheme as shown in Figure 2-18. When the inlet bridge of the 199th segment in the 200-segment scheme was removed (the bridge was assigned a bridge index of zero) the lengths of the perforated section became equal in both schemes. The production rate in the 200-segment scheme decreased and became 11,715 m³/d closer to that obtained from the other scheme (-2.24% error).

2.7.2 Comparison with Published Literature

In order to verify that the proposed network model provides reasonable ranges of predicted results the results were compared with those from published literature. The proposed model was used to simulate a well with similar conditions as the small-diameter, high-flow-rate, single-phase oil well used in Dawkrajai et al. [17]. The reference well was cased, perforated well and had the properties as listed in Table 2-3.

A well with conditions as close as possible to the conditions reported in the literature was used in this analysis. The well conditions were set to match the

Table 2-3: Well and fluid properties reported in Table 5.2 and Table 5.3 in Dawkrajai et al. [17].*

Property	Value
Well inner diameter, ID (in)	2.602
Well outer diameter, OD (in)	3.5
Diameter with cement (in)	5
Casing thermal conductivity, K_c ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	11.99
Cement thermal conductivity, K_{cem} ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	6.95
Relative roughness	0.01
Total length (m)	601
Pipe opened ratio (%)	2
Oil heat transfer coefficient, K_o ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	0.1378
Oil API	45.176

* Units of values converted to be consistent with this research.

Table 2-4: Well basic characteristics used for comparisons with published literature.

Property	Value
Well length (m)	601
Permeability (Darcy)	0.05
Near-wellbore skin factor	0
Drainage radius (m)	300
Oil saturation	1
Oil density at reservoir conditions (kg/m^3)	800.9
Tubing diameter (in)	1.815
Well outside diameter (annulus) (in)	2.602
Discharge coefficient for flow through slots ($\text{Pa}\cdot(\text{kg}/\text{m}^3)^{-1}\cdot(\text{m}/\text{s})^{-2}$)	10
Slot ratio (%)	60
Perforated ratio (%)	2
Casing thickness (in)	0.898
Cement thickness (in)	1.5
Oil heat transfer coefficient, k_{fl} ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	0.1378
Steel tubing thermal conductivity, k_w ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	50
Outer casing thermal conductivity, k_c ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	11.99
Cement thermal conductivity, k_{cem} ($\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$)	6.95
Fluid heat capacity ($\text{J}/\text{kg } ^\circ\text{C}$)	2209.152

production rate of 3,138.7 m³/d in the reference well. Similar properties of well, fluid, and reservoir were used as available. The properties of the well system used in the analysis are listed in Table 2-4.

In the reference flow parameters were solved using a wellbore model based on the mass, momentum, and energy balances. The model differential equations were solved numerically using a finite difference method [17]. Figure 2-22 to Figure 2-24 compare the results predicted by the proposed model and those predicted by the reference. Note that the reference values were read directly from Figure 5.9 and Figure 5.10 in Dawkrajai et al. [17]. The results showed that the profiles of the flow parameters were comparable. The total pressure drops in tubing, production rates, and total temperature change in tubing were also comparable (as listed in Table 2-5). The results, regardless of the differences, were of the same order of magnitude. The differences in the results owed to many factors including differences in fluid, well, and reservoir properties. The main difference to be noted was that in Dawkrajai et al. the well was simulated without an annular space [17]. In this analysis the diameter of the reference well's tubing was equal to the outer diameter of the annulus of the well (Table 2-4). In addition different correlations were used to calculate friction factors and fluid heat transfer coefficient. In this research the well friction was modeled using a correlation for smooth pipes. Neglecting wall roughness would result in an

Table 2-5: Results from the proposed model and those from Dawkrajai et al. [17].*

Parameters	Proposed model	Literature
Total pressure drop (bara)	45.315	42.595
Production rate (m ³ /d)	3140.1	3138.73
Total temperature change (°C)	3.19	3.5

* Units of values converted to be consistent with this research.

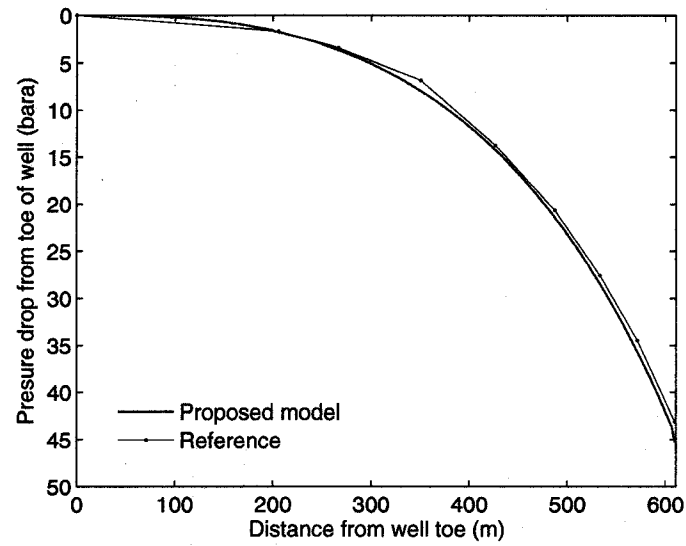


Figure 2-22: Pressure drop in well tubing compared with results from Dawkrajai et al. [17].

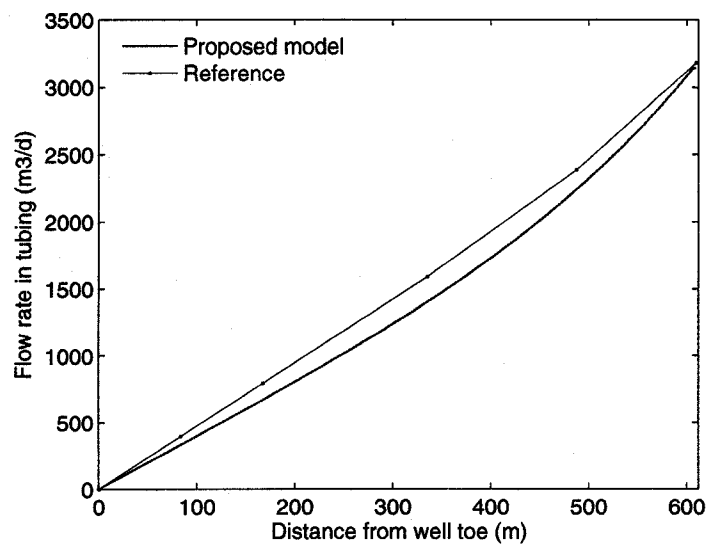


Figure 2-23: Production rate compared with results from Dawkrajai et al. [17].

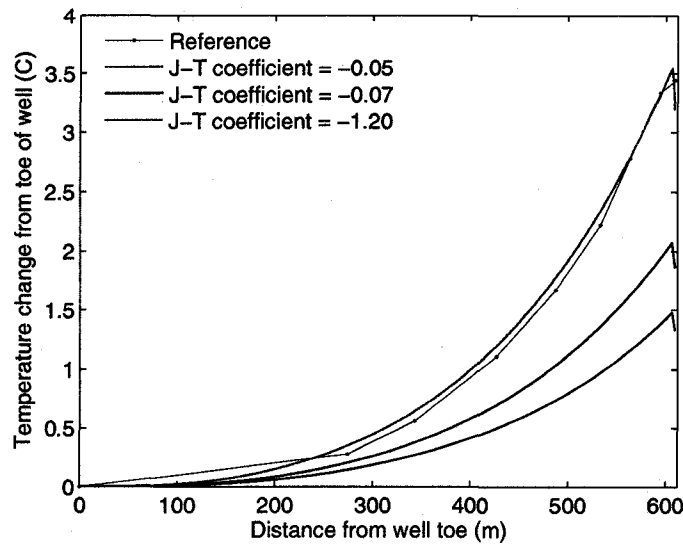


Figure 2-24: Matching the predicted temperature profile of fluid in tubing with the results from Dawkrajai et al. [17] by varying the Joule-Thompson coefficient.

underestimated frictional pressure drop. However because of the annulus, a higher pressure drop in the well than that without the annulus was expected.

As shown in Figure 2-24, the temperature change in the well matched the reference value when a Joule-Thompson coefficient of $-1.20\text{ }^{\circ}\text{C}/\text{bara}$ was used. Basically the temperature change in the well is a result of the balance between heating and cooling effects. In an oil well the heating effect is caused by Joule-Thompson effect and is a function of the well pressure drop. The cooling effects include mixing with cooler inflow streams and heat loss to the surroundings. In this analysis the fluid was a single-phase liquid thus the only effect that caused the fluid temperature to rise was the Joule-Thompson effect. The fluid with a more typical Joule-Thompson coefficient of $-0.05\text{ }^{\circ}\text{C}/\text{bara}$ for example would cause the maximum change in temperature of $2.27\text{ }^{\circ}\text{C}$ for a 45.315 bara of pressure drop. In this specific case, without considering any heat loss in the system, the minimum value of the coefficient to provide such a change in temperature (the reference $3.5\text{ }^{\circ}\text{C}$) for the provided pressure drop of 45.315 bara is $-0.077\text{ bara}/^{\circ}\text{C}$. For this reason, even though the value of

-0.12 °C/bar was considered quite large for a reservoir fluid, it provided reasonable temperature prediction in this case.

Furthermore using the Joule-Thompson coefficient of -1.2 °C/bar there was a slight difference in the predicted temperature change from the toe to the heel of well (3.19 °C from the proposed model and 3.5 °C from the literature). In spite of the difference it should be noted that the maximum temperature predicted by the proposed model did not occur at the heel of the well but in the tubing of the last segment before the tubing fluid was mixed with the cooler stream from the annulus. The temperature at this location was approximately 3.55 °C which was closer to the reference value. The difference in well conditions (i.e. the presence of the annular space) again could be accounted for the difference in the predicted temperature.

CHAPTER 3

THERMODYNAMIC PHASE EQUILIBRIUM

In addition to the constant-composition network model presented in the previous section, a compositional model is used in the proposed model to predict asphaltene precipitation. The asphaltene model is a three-phase model where solid-phase formation is determined by phase equilibrium calculations. As thermodynamic phase equilibrium is a nontrivial issue the basic concepts required to perform phase equilibrium calculations will be briefly described in this chapter.

3.1 Phase Equilibrium

A system at equilibrium is a system under a static condition where all forces in the system (i.e. resistance and driving forces) are in balance with no change or tendency toward change microscopically in the state of the system [53]. A system can be assumed at its equilibrium when it is given sufficient time for phases to interact under unchanged pressure and temperature conditions. This also applies to oil production systems. In oil production there are slow changes in fluid properties and the number of phases. However all of the co-existing phases can still be assumed to attain an equilibrium at any point in the system after a period of time. Using this assumption fluid properties, flow behaviour, and phase behaviour of production fluids can be evaluated at its equilibrium state.

In this section the criteria for equilibrium will be derived. The resulting criteria are always referred to in phase equilibrium calculations as a basic constraint that needs to be satisfied.

To derive the equilibrium criteria consider the first law of thermodynamics. For a closed system (a system with constant composition or no mass exchanging with its surroundings), the change in the total energy (E) results from heat transfer (Q) and work done (W) across the system's boundary.

$$\begin{aligned}\Delta E &= \Delta U + \Delta E_p + \Delta E_k \\ &= Q - W\end{aligned}\tag{3.1}$$

where U is the internal energy; E_p and E_k are the potential and kinetic energy respectively. The heat transported into the system and work done by the system are assigned positive signs. At equilibrium there is no change in energy thus the change in internal energy (dU) can be written for a process with no changes in kinetic and potential energy as:

$$dU = dQ - dW\tag{3.2}$$

The term dQ can be written in terms of entropy change in the universe (dS_{UNIV}) and the change in system entropy (dS_{SYS}). The entropy change in the universe is larger than zero in an irreversible process and equal to zero in an ideal reversible process ($dS_{UNIV} = dS_{SYS} - \frac{dQ}{T} \geq 0$). Also, the term dW can be expressed in terms of pressure (P) and volume (V) as $dW = PdV$. Equation 3.2 can now be written as:

$$dU \leq TdS - PdV\tag{3.3}$$

where S and V are the system entropy and volume, respectively. The equal sign is for an ideal reversible process and the inequality sign is for an irreversible process.

Introducing the Gibbs energy (G) defined as:

$$G = H - TS\tag{3.4}$$

where H is the system enthalpy defined as:

$$H = U + PV \quad (3.5)$$

After some rearrangement Equation 3.3 is written in terms of the Gibbs energy as:

$$dG \leq -SdT + VdP \quad (3.6)$$

Consider Equation 3.6. At constant T and P (dT and dP equal zero) while the system is approaching the equilibrium state, the change in the Gibbs energy decreases in a real process and remains constant in a reversible process, i.e. $((dG)_{T,P} \leq 0)$. Based on this, the criterion for equilibrium can now be derived such that the system's Gibbs energy reaches its minimum at equilibrium state. The mathematical expression for the equilibrium criterion is:

$$(dG)_{T,P} = 0 \quad (3.7a)$$

and

$$(d^2G)_{T,P} > 0 \quad (3.7b)$$

For a system to be considered at its equilibrium state its “global” minimum Gibbs energy must be achieved.

For a heterogeneous closed system where a number of phases co-exist, it is acceptable to treat the system as a group of single-phase open systems with mass transfer between the phases. Similar to the closed system the change in the Gibbs energy in an open system can be expressed using Equation 3.8. Note that the compositional term is added to Equation 3.6 to account for the variation of energy due to mass transfer between phases:

$$dG = -SdT + VdP + \sum_i^N \left(\frac{\partial G}{\partial n_i} \right)_{T,P,n_{j \neq i}} dn_i \quad (3.8)$$

The term $(\partial G/\partial n_i)_{T,P,n_{j \neq i}}$ is the partial molar Gibbs energy which is also called chemical potential (μ_i):

$$\mu_i = \left(\frac{\partial G}{\partial n_i} \right)_{T,P,n_{j \neq i}} \quad (3.9)$$

The compositional term in Equation 3.8 is over the total number of components (N) present in the system; n_i is the number of moles of component i ; and $n_{j \neq i}$ is the number of moles of each and all components except component i .

Substituting Equation 3.9 into Equation 3.8 to get:

$$dG = -SdT + VdP + \sum_i^N \mu_i dn_i \quad (3.10)$$

Considering a closed system containing θ phases, the change in the Gibbs energy for the entire system is the sum of the energy change in all of the existing phases:

$$dG = \sum_{h=1}^{\theta} (-S_h) dT + \sum_{h=1}^{\theta} (V_h) dP + \sum_{h=1}^{\theta} \left(\sum_i^N \mu_i dn_i \right)_h \quad (3.11)$$

where h is the index for each phase.

Again, at uniform and constant temperature and pressure (dT and dP equal zero) the equilibrium criterion for a multi-phase system can now be deduced from Equation 3.7a and Equation 3.11 as:

$$\begin{aligned} (dG)_{T,P} &= \sum_{h=1}^{\theta} \left(\sum_i^N \mu_i dn_i \right)_h \\ &= 0 \end{aligned} \quad (3.12)$$

The above criterion can be derived further. Consider a pair of phases α and β in a system consisting of θ phases. Equation 3.10 can be written for each phase at constant temperature and pressure as:

$$(dG)_{T,P}^{\alpha} = \sum_i^N \mu_i^{\alpha} dn_i^{\alpha} \quad (3.13a)$$

$$(dG)_{T,P}^{\beta} = \sum_i^N \mu_i^{\beta} dn_i^{\beta} \quad (3.13b)$$

The total change in the Gibbs energy for these two phases is:

$$\begin{aligned}(dG)_{T,P} &= \sum_i^N \mu_i^\alpha dn_i^\alpha + \sum_i^N \mu_i^\beta dn_i^\beta \\ &= 0\end{aligned}\tag{3.14}$$

From the conservation of mass the mass of component i leaving one phase must equal to the mass gained by the other. Thus:

$$dn_i^\alpha + dn_i^\beta = 0\tag{3.15}$$

Equation 3.15 and Equation 3.14 gives:

$$\sum_i^N (\mu_i^\alpha - \mu_i^\beta) dn_i^\alpha = 0\tag{3.16}$$

Since changes in the mass of a component (dn_i^α) is independent and arbitrary, it can be concluded from Equation 3.16 that $\mu_i^\alpha = \mu_i^\beta$. In other words the chemical potentials of component i in both phases are equal.

By comparing each pair of the existing phases in the system equilibrium criteria can also be established for a multi-phase mixture such that the chemical potentials of each component in all phases are equal at equilibrium:

$$\mu_i^{(1)} = \mu_i^{(2)} = \mu_i^{(3)} = \dots = \mu_i^{(\theta)}\tag{3.17}$$

where $i = 1, 2, \dots, N$.

Equation 3.17 will be of practical use if the chemical potential is expressed in terms of measurable parameters. The concept of fugacity is therefore introduced for this purpose. Consider the change in the Gibbs energy in an ideal process (Equation 3.6), written in terms of molar values:

$$dg = -sdT + vdP\tag{3.18}$$

where g , s , and v are the molar Gibbs energy, molar entropy, and molar volume respectively. For an ideal gas the pressure volume relation is $Pv = RT$. Thus Equation 3.18 can be written for an ideal gas at a constant temperature as:

$$\begin{aligned} dg &= vdP \\ &= \frac{RT}{P}dP \\ &= RTd\ln P \end{aligned} \tag{3.19}$$

where R is the Universal gas constant.

For a real fluid a “corrected pressure” or fugacity is introduced and replace P in Equation 3.19 to get:

$$dg = RTd\ln f \tag{3.20}$$

For a multi-component mixture the change in partial molar Gibbs energy of component i in the mixture ($\bar{G}_i = \partial G / \partial n_i$) can be expressed by analogy with Equation 3.20 as [53]:

$$d\bar{G}_i = RTd\ln f_i \tag{3.21a}$$

or in terms of chemical potential (Equation 3.9) as:

$$d\mu_i = RTd\ln f_i \tag{3.21b}$$

where f_i is the fugacity of component i in the mixture.

Integrate Equation 3.21b at a constant temperature to get [53]:

$$\mu_i = RT \ln f_i + C_i(T) \tag{3.22}$$

Consider the above equation, at equilibrium all phases are at equal temperature and thus the temperature-dependent integration constants (C_i 's) are all of the same values for all the phases. To satisfy the equal-chemical-potential criteria as in Equation 3.17, the fugacity of each component (i) must also be equal in all phases at

equilibrium:

$$f_i^{(1)} = f_i^{(2)} = f_i^{(3)} = \dots = f_i^{(\theta)} \quad (3.23)$$

where $i = 1, 2, \dots, N$.

An advantage of using fugacity to indicate equilibrium is that fugacity can be related to measurable properties through another parameter called fugacity coefficient (ϕ). To introduce the fugacity coefficient, consider the residual Gibbs energy (g^R) which represents the deviation of the Gibbs energy of a real system from that of the ideal gas at the same pressure and temperature conditions:

$$g^R = g - g^{ig} \quad (3.24)$$

The superscript R and ig indicate residual and ideal gas properties respectively.

Using Equations 3.19 and 3.20 the change in residual Gibbs energy in a pure substance (consisting of one component) can be expressed as:

$$\begin{aligned} dg^R &= d(g - g^{ig}) \\ &= RTd \ln f - RTd \ln P \\ &= RTd \ln \frac{f}{P} \\ &= RTd \ln \phi \end{aligned} \quad (3.25)$$

where ϕ is the fugacity coefficient defined as the ratio of fugacity to pressure. Thus for a pure substance,

$$\phi = \frac{f}{P} \quad (3.26)$$

The deviation of fugacity coefficient from unity reflects non-ideality of the system. As pressure approaches zero all systems behave as ideal gases and the fugacity coefficients approach the value of one.

Integration of Equation 3.25 at a constant temperature gives:

$$g^R = RT \ln \phi + C(T) \quad (3.27)$$

where C is the integration constant and is a function of temperature only. Consider an ideal gas. Its fugacity is equal to pressure ($\phi = 1$ and $g^R = 0$). The constant $C(T)$ is found to be zero and we get:

$$\frac{g^R}{RT} = \ln \phi \quad (3.28)$$

The fugacity coefficient can now be calculated through $\frac{g^R}{RT}$. First consider the differential form of $\frac{g}{RT}$,

$$d\frac{g}{RT} = \frac{1}{RT}dg - \frac{g}{RT^2}dT \quad (3.29)$$

Substitute Equations 3.4 and 3.18 into the above equation to get:

$$d\frac{g}{RT} = \frac{v}{RT}dP - \frac{h}{RT^2}dT \quad (3.30)$$

Deriving $\frac{g^{ig}}{RT}$ in a similar manner gives:

$$\begin{aligned} d\frac{g^R}{RT} &= d\left(\frac{g}{RT} - \frac{g^{ig}}{RT}\right) \\ &= \frac{v^R}{RT}dP - \frac{h^R}{RT^2}dT \end{aligned} \quad (3.31)$$

At a constant temperature and $v = ZRT/P$ for a real system, we get:

$$\begin{aligned} d\frac{g^R}{RT} &= \frac{v^R}{RT}dP \\ &= \frac{v - v^{ig}}{RT}dP \\ &= \frac{Z - Z^{ig}}{P}dP \\ &= (Z - 1)\frac{dP}{P} \end{aligned} \quad (3.32)$$

where the compressibility factor equals to one for an ideal gas ($Z^{ig} = 1$).

At constant temperature and composition the fugacity coefficient can be calculated by integrating Equation 3.32 from an ideal-gas state ($P = 0$ and $g^R/RT = 0$) to P .

$$\begin{aligned}\ln \phi &= \frac{g^R}{RT} \\ &= \int_0^P (Z - 1) \frac{dP}{P}\end{aligned}\quad (3.33)$$

For a multi-component mixture Equation 3.25 can be written as [53]:

$$d\bar{G}_i^R = RT d \ln \phi_i \quad (3.34)$$

where \bar{G}_i^R is the partial molar residual Gibbs energy. The fugacity coefficient of component i in the mixture ($\ln \phi_i$) can be perceived as a partial molar property with respect to $\ln \phi$ [53]:

$$\begin{aligned}\ln \phi_i &= \left[\frac{\partial n \ln \phi}{\partial n_i} \right]_{P, T, n_{j \neq i}} \\ &= \int_0^P (\bar{Z}_i - 1) \frac{dP}{P}\end{aligned}\quad (3.35)$$

and the fugacity coefficient of component i in the mixture is defined as:

$$\phi_i = \frac{f_i}{P z_i} \quad (3.36)$$

where z_i is the mole fraction of component i in the mixture.

Equation 3.35 facilitates calculations of component fugacities in multi-component systems through the use of measurable variables, i.e. Z_i and P . The equilibrium state of a system is determined where the equal-fugacity criteria are satisfied (as per Equation 3.23). Fluid properties and phase behaviour of the system can then be calculated at the equilibrium state. In order to calculate fugacities using Equation 3.35, however, an equation of state is required. Equations of state are equations relating the compressibility factor (Z) to pressure (P) which allow fugacities to be calculated through Equation 3.33 in the case of pure components or Equation 3.35

in the case of a multi-component mixtures. The next section outlines different types of cubic equations of state commonly used in oil and gas industry.

3.2 Cubic Equations of State

Equations of state (EOS) are equations relating pressure, temperature, volume, and compositions. Different types of equations of state have been successfully used to predict phase behaviour of hydrocarbon reservoir fluids. The simplest are semi-empirical van der Waals equations with two or three parameters which will be presented here in detail. Other more complex equations include Benedict-Webb-Rubin (BWR) type equations which are also applicable for reservoir fluids. More parameters in these equations provide higher flexibility but more complexity compared with van der Waals type equations of state [16].

In petroleum industry van der Waals type equations (e.g. Soave-Redlich-Kwong and Peng-Robinson) are commonly used and suitable for hydrocarbon mixtures. The original van der Waals equation was first introduced in 1873 [53]. The van der Waals equation which takes a cubic form was developed based on the ideal gas equation ($Pv = RT$) by adding parameters representing the attractive and repulsive intermolecular forces [53]:

$$\left(P + \frac{a}{v^2}\right)(v - b) = RT \quad (3.37)$$

where a/v^2 and b represent the attractive and repulsive terms respectively and v is the molar volume. The parameter b is also called a “co-volume” and always has a value less than the molar volume (v). Parameters a and b in the original van der Waals equation of state are expressed as:

$$a = \frac{9}{8}RT_c v_c = \frac{27}{64} \frac{R^2 T_c^2}{P_c} \quad (3.38a)$$

$$b = \frac{1}{3}v_c = \frac{1}{8} \frac{RT_c}{P_c} \quad (3.38b)$$

where the subscript c refers to the values at the critical point.

The above equation can be expressed in terms of compressibility factor as:

$$Z^3 - (1 + B)Z^2 + AZ - AB = 0 \quad (3.39)$$

where A and B are dimensionless parameters defined as:

$$A \equiv \frac{aP}{RT^2} \quad (3.40a)$$

$$B \equiv \frac{bP}{RT} \quad (3.40b)$$

By solving Equation 3.37 or Equation 3.39 for a pure compound at specified pressure and temperature below the critical temperature it can result in three real roots for volume or compressibility factor. The highest value corresponds to that of the vapour, the lowest corresponds to that of the liquid, and the middle value has no physical significance. At temperatures above the critical point the equation provides only one physically possible root.

The van der Waals equation is the basic equation upon which other cubic equations of state were developed. Several modifications have been introduced to the attractive and repulsive terms of the van der Waals equation. In addition experimental data for pure fluids have been used in determining the a and b parameters in attempt to improve the equation's accuracy. Despite the variety of modifications all van der Waals type equations take the same general form of:

$$P = \frac{RT}{v - b} - \frac{a}{v^2 + uv - w^2} \quad (3.41)$$

or in terms of the compressibility factor:

$$Z^3 - (1 + B - U)Z^2 + (A - BU - U - W^2)Z - (AB - BW^2 - W^2) = 0 \quad (3.42)$$

The dimensionless parameters A and B are defined as in Equations 3.40; U and W are defined as:

$$U \equiv \frac{uP}{RT} \quad (3.43a)$$

$$W \equiv \frac{wP}{RT} \quad (3.43b)$$

The basic parameters a and b in the cubic EOS can be determined by satisfying the condition at the critical temperature and pressure that:

$$\left(\frac{\partial P}{\partial v}\right)_{T=T_c} = \left(\frac{\partial^2 P}{\partial v^2}\right)_{T=T_c} = 0 \quad (3.44)$$

For two-parameter cubic equations of state the general expressions of the parameters a and b are:

$$a = \Omega_a \frac{R^2 T_c^2}{P_c} \quad (3.45a)$$

$$b = \Omega_b \frac{RT_c}{P_c} \quad (3.45b)$$

where Ω_a and Ω_b are different for each modified equation.

An important modification to the van der Waals equation is to replace the “ a ” parameter with a temperature-dependent parameter. This is achieved by adding another parameter $\alpha(T)$ to account for the temperature dependency.

$$a = a_c \alpha \quad (3.46)$$

where a_c is expressed as:

$$a_c = \Omega_{ac} \frac{R^2 T_c^2}{P_c} \quad (3.47)$$

As mentioned earlier, the fugacity coefficient of a pure substance can be calculated from Equation 3.33 provided the relation between compressibility factor and pressure. When Equation 3.42 is used to provide such relation the expression of the fugacity coefficient becomes:

$$\ln \phi = (Z - 1) - \ln(Z - B) + \frac{A}{\sqrt{U^2 + 4W^2}} \ln \frac{2Z + U - \sqrt{U^2 + 4W^2}}{2Z + U + \sqrt{U^2 + 4W^2}} \quad (3.48)$$

Depending on the modification made to the repulsive and/or attractive terms van der Waals type equations of state differ in their features. The equations are therefore suitable for use in different situations. In oil and gas applications two- or three-parameter equations of state are found sufficient [16].

3.2.1 Two-parameter Cubic Equations of State

Among various equations of state two-parameter cubic equations are the most widely applied. For hydrocarbon mixtures in petroleum industry the most commonly used are Soave-Redlich-Kwong and Peng-Robinson equations of state.

Redlich and Kwong Equation of State (RK)

Redlich and Kwong equation is a modification of the original van der Waals equation of state where the attractive term was modified by adding the temperature-dependent parameter $\alpha = T_r^{0.5}$ [49]:

$$P = \frac{RT}{v-b} - \frac{aT_r^{0.5}}{v(v+b)} \quad (3.49)$$

where T_r is the reduced temperature defined as $T_r = T/T_c$; a and b are defined as:

$$a = 0.42747 \frac{R^2 T_c^2}{P_c} \quad (3.50a)$$

$$b = 0.08664 \frac{RT_c}{P_c} \quad (3.50b)$$

Soave-Redlich-Kwong Equation of State (SRK)

Soave later modified the RK equation of state by replacing the temperature dependency of the attractive term ($T_r^{0.5}$) with a temperature-dependent function (α) [54]:

$$P = \frac{RT}{v-b} - \frac{a_c \alpha}{v(v+b)} \quad (3.51)$$

The parameters a_c and b are defined similarly to the a and b in RK equation of state correspondingly and α is defined as [16]:

$$\alpha = \left[1 + m \left(1 - T_r^{0.5}\right)\right]^2 \quad (3.52)$$

where m is a function of the acentric factor (ω) defined as:

$$m = 0.480 + 1.574\omega - 0.176\omega^2 \quad (3.53)$$

The acentric factor (ω) represents the deviation of molecular structure from spherical molecules (it has the value of zero for a simple spherical molecule). The expression for the acentric factor is [16]:

$$\omega = -\log(P^s/P_c)_{\text{at } T_r=0.7} - 1.0 \quad (3.54)$$

where P^s is the vapour pressure.

The improved expression of m was later suggested by Soave et al. that the value calculated from Equation 3.53 is divided by 1.18 [55]. Graboski and Daubert also suggested the m value to be calculated as [21]:

$$m = 0.48508 + 1.555171\omega - 0.15613\omega^2 \quad (3.55)$$

In terms of compressibility factor, the SRK equation of state is expressed as:

$$Z^3 - Z^2 + (A - B - B^2)Z - AB = 0 \quad (3.56)$$

where A and B can be obtained from Equations 3.40.

Peng-Robinson Equation of State (PR)

Due to SRK's lack in reliability to predict liquid density, Peng and Robinson attempted to improve this lacking feature by modifying the attractive term [44].

The Peng-Robinson equation of state is:

$$P = \frac{RT}{v - b} - \frac{a_c \alpha}{v(v + b) + b(v - b)} \quad (3.57)$$

where a_c and b are defined as:

$$a = 0.457235 \frac{R^2 T_c^2}{P_c} \quad (3.58a)$$

$$b = 0.077796 \frac{RT_c}{P_c} \quad (3.58b)$$

The temperature-dependent parameter (α) is defined as in Equation 3.52 and the value of m is:

$$m = 0.37464 + 1.5422\omega - 0.26992\omega^2 \quad (3.59)$$

The correlation for m was also provided for heavier components as [50]:

$$m = 0.3796 + 1.485\omega - 0.1644\omega^2 + 0.01667\omega^3 \quad (3.60)$$

Finally, written in terms of compressibility factor the Peng-Robinson equation of state becomes:

$$Z^3 - (1 - B)Z^2 + (A - 2B - 3B^2)Z - (AB - B^2 - B^3) = 0 \quad (3.61)$$

When the Soave-Redlich-Kwong or the Peng-Robinson equations of state is used along with random mixing rules (to be presented in Section 3.2.3), the fugacity coefficients in a multi-component mixture can be calculated based on Equation 3.35 as:

$$\ln \phi_i = \frac{b_i}{b} (Z - 1) - \ln(Z - B) - \frac{A}{B(\delta_2 - \delta_1)} \left(\frac{(2 \sum_{j=1}^N x_j a_{ij})}{a} - \frac{b_i}{b} \right) \ln \left(\frac{Z + \delta_2 B}{Z + \delta_1 B} \right) \quad (3.62)$$

where the equations of state take the following form:

$$P = \frac{RT}{v - b} - \frac{a}{(v + \delta_1 b)(v + \delta_2 b)} \quad (3.63)$$

and the parameters δ_1 and δ_2 are 1 and 0 for the Soave-Redlich-Kwong, and $1 + \sqrt{2}$ and $1 - \sqrt{2}$ for the Peng-Robinson equations of state respectively.

Two-parameter equations of state's lack in accuracy of volume predictions near the critical point and the flexibility of the equations to match both vapour pressure and liquid volume lead to the introduction of three-parameter equations of state. In three-parameter equations of state the flexibility of the equations is increased by the use of an additional parameter. The parameter is for example an additional parameter in the attractive term or an acentric factor. Further details on three-parameter equations of state can be found in the literature (e.g. Danesh [16]).

3.2.2 Volume Shift

Another solution to increase the accuracy in liquid volume prediction in cubic equations of state is to use the volume shift concept. The use of two-parameter equations of state are generally associated with a systematic deviation of liquid molar volume from experimental data when the pressure is within a range away from the critical pressure. This deviation can therefore be offset by the use of a volume shift introduced by Peneloux et al. [43]. The expression below is substituted directly into the equation of state.

$$v^{cor} = v - c \quad (3.64)$$

In Equation 3.64, v^{cor} is the corrected molar volume and c is the correction term obtained by comparing experimental and the predicted saturated liquid volumes at $T_r = 0.7$. The volume translation parameter (c) can be calculated according to Peneloux et al. as below [43]:

$$c = 0.40768 (0.29441 - Z_{RA}) \frac{RT_c}{P_c} \quad (3.65)$$

where Z_{RA} is the Rackett compressibility factor in the modified Rackett equation [56]:

$$v^s = \left(\frac{RT_c}{P_c} \right) Z_{RA}^{1+(1-T_r)^{\frac{2}{7}}} \quad (3.66)$$

where v^s is the saturated molar volume.

For multi-component fluids the correction term (c) can be obtained using a mixing rule.

$$c = \sum_i^N c_i x_i \quad (3.67)$$

where x_i is the mole fraction of component i in the mixture.

By introducing the parameter c the equilibrium condition is not affected when the same equation of state is used for both vapour and liquid phases. When volume translation parameters are used the fugacity coefficient (ϕ) can then be obtained from [16]:

$$\phi_i^{cor} = \phi_i \exp \left(\frac{-c_i P}{RT} \right) \quad (3.68)$$

where ϕ_i^{cor} and ϕ_i are the modified and original fugacity coefficients of component i respectively.

3.2.3 Mixing Rules

As equations of state were originally developed for pure components they can be applied to multi-component systems provided that the mixture parameters are determined. This can be achieved by the use of mixing rules.

Random mixing rules

Also known as van der Waals mixing rules, random mixing rules assume that compounds are randomly distributed within the mixture. This assumption is sufficient for hydrocarbon mixtures. The attractive-force and repulsive-force parameters

between a pair of molecules i and j (a_{ij} and b_{ij} respectively) can be expressed as:

$$a_{ij} = (a_i a_j)^{\frac{1}{2}} \quad (3.69a)$$

$$b_{ij} = \frac{(b_i + b_j)}{2} \quad (3.69b)$$

The parameters a and b to be used in an equation of state are:

$$a = \sum_i \sum_j x_i x_j (a_i a_j)^{0.5} (1 - k_{ij}) \quad (3.70a)$$

$$\begin{aligned} b &= \sum_i \sum_j x_i x_j b_{ij} \\ &= \sum_i \sum_j x_i x_j \frac{(b_i + b_j)}{2} \\ &= \sum_i x_i b_i \end{aligned} \quad (3.70b)$$

The binary interaction parameter (k_{ij}) in Equation 3.70a accounts for the differences between two molecules. The parameter k_{ij} is zero for two hydrocarbons with little difference in size. The value, on the other hand, is non-zero for different molecules such as non-hydrocarbon/hydrocarbon or light/heavy hydrocarbons. The most commonly used correlation for the binary interaction parameter is [13]:

$$k_{ij} = \vartheta \left[1 - \left(\frac{2 \left(v_{c_i}^{1/3} v_{c_j}^{1/3} \right)^{1/2}}{v_{c_i}^{1/3} + v_{c_j}^{1/3}} \right)^{\theta} \right] \quad (3.71)$$

where ϑ and θ are constants which vary depending on each equation of state. Inclusion of binary interaction parameters in mixing rules provides more flexibility to the equation of state. These parameters are frequently used as tuning parameters in regressions of equations of state to experimental data.

Non-random mixing rules

In many cases asymmetric compounds such as water or methanol (added as hydrate inhibitor) are present in the fluid system. The random mixing rules are not sufficient in these situations and more complex mixing rules are required. In non-random mixing rules the attractive term is separated into two parts [16].

$$a = a^C + c^A \quad (3.72)$$

where a^C represents the conventional random mixing term given by Equation 3.70a and a^A represents the asymmetric term due to polarity. Various expressions for the asymmetric term have been proposed and successfully applied to binary systems (see for example Danesh for more detail) [16].

3.3 Vapour-Liquid Phase Equilibrium

As mentioned earlier in, oil and gas production, fluid phase behaviour is determined at its equilibrium. Hydrocarbon fluid systems generally involve a number of compounds as well as coexisting phases. This adds complexity to phase behaviour calculations. In most cases, however, vapour-liquid phase equilibrium can be calculated independently of the equilibrium calculation for the solid phase or second liquid phase (e.g. water) [16]. This allows the complex multi-phase behaviour calculations to be reduced to vapour-liquid phase equilibrium. This also applies to the case of asphaltene precipitation. For the above reason this section is dedicated to vapour-liquid equilibrium. Phase equilibrium calculations involving asphaltene will be described in detail later in Chapter 4 where the asphaltene model is presented.

A basic vapour-liquid phase calculation can be conducted at fixed pressure and temperature using an isothermal flash algorithm. The calculation results in phase fractions (n^L and n^V) and the compositions (x_i and y_i) of the fluid provided feed

composition (z_i). Consider the material balance for the liquid and vapour phases:

$$n^V + n^L = 1 \quad (3.73)$$

where n^L is the number of moles of liquid and n^V is the number of moles of vapour after one mole of mixture is flashed into two phases. A material balance can also be written for each component i :

$$z_i = x_i n^L + y_i n^V \quad (3.74)$$

where z_i , x_i , and y_i are the mole fractions of component i in the mixture, liquid, and vapour phases, respectively. In an N -component system the mole fractions of all the components in each phase sum up to one:

$$\sum_{i=1}^N x_i = \sum_{i=1}^N y_i = 1 \quad (3.75)$$

Introducing equilibrium ratio (K_i) defined as the ratio of mole fraction of component i in the vapour phase (y_i) to that in the liquid phase (x_i):

$$K_i = \frac{y_i}{x_i} \quad (3.76)$$

where $i = 1, 2, \dots, N$ and substituting it into Equation 3.74, the mole fraction of component i in the liquid and vapour phases can be expressed as:

$$x_i = \frac{z_i}{1 + (K_i - 1) n^V} \quad (3.77a)$$

$$y_i = \frac{z_i K_i}{1 + (K_i - 1) n^V} \quad (3.77b)$$

Rachford-Rice equation is an equation expressing n^V in terms of K_i [48].

$$\begin{aligned} f(n^V) &= \sum_{i=1}^N (y_i - x_i) \\ &= \sum_{i=1}^N \frac{z_i (K_i - 1)}{1 + (K_i - 1) n^V} \\ &= 0 \end{aligned} \quad (3.78)$$

Provided that the values of K_i 's are known, n^V which indicates the state of the mixture at the given pressure and temperature can be calculated using the above equation. In most cases however K_i 's are unknown. In this case, an iterative method is used to solve for n^V . A simple successive substitution flash iteration as shown in Figure 3-1 can be used for this purpose given appropriate initial guess values for K_i 's. The widely used Wilson equation provides the initial guess K -values [16]:

$$K_i = \frac{P_{c_i}}{P} \exp \left[5.37 (1 + \omega_i) \left(\frac{1 - T_{c_i}}{T} \right) \right] \quad (3.79)$$

Given the initial K -values solving Equation 3.77 for n^V allows the phase compositions to be calculated using Equation 3.77. The fugacities of all components in the mixture can then be calculated based on Equation 3.62 using two equations of state: one for the liquid phase and the other for the vapour phase. The steps are repeated by updating the K -values until the equal-fugacity equilibrium criteria are satisfied. This can be verified by comparing the fugacities of each component in the liquid and vapour phases (f_i^L and f_i^V respectively). To update the K -values the expression below is used:

$$K_i^{new} = K_i^{old} \frac{f_i^L}{f_i^V} \quad (3.80)$$

where $i = 1, 2, \dots, N$.

For a two-phase system where two phases co-exist at specified pressure and temperature, n^V has a value between 0 and 1. In the case that n^V is more than 1 or n^V is less than 0, the mixture is considered to be all vapour or all liquid respectively. Different methods may also be used to help improve the convergence rate of the flash iteration. These methods can be found in the literature (see for example Danesh [16]).

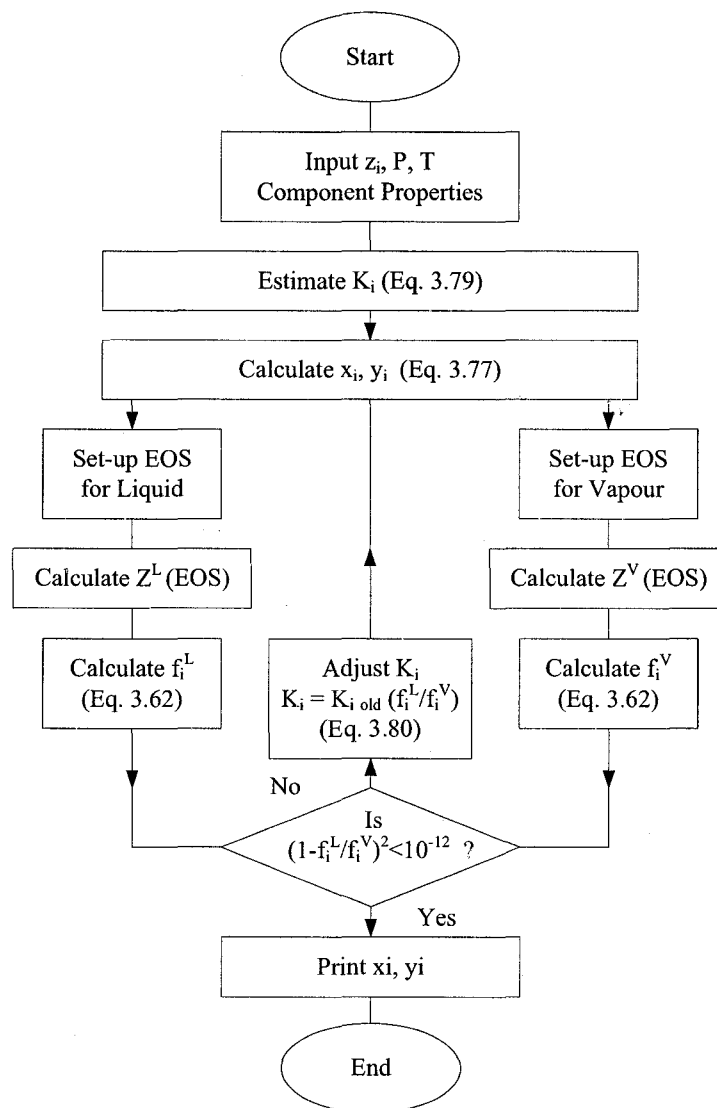


Figure 3-1: Vapour-liquid flash algorithm [16].

CHAPTER 4

ASPHALTENE PRECIPITATION PREDICTION

As mentioned, the proposed model consists of two parts: a network model and a compositional asphaltene model. In this chapter the latter which deals with detailed asphaltene precipitation analysis at local points in the network will be presented. In this research, flows through restrictions (e.g. valves installed as parts of advanced well completion) are the case of interest because the drastic pressure drop inside the restriction is suspected of inducing asphaltene to precipitate out of the fluid stream. Based on simplified restriction geometry and the local pressure-temperature condition of the fluid, asphaltene precipitation can be determined at any point inside and along the restricted flow path. Below asphaltene precipitation mechanisms, available asphaltene model, and asphaltene prediction methodology as a part of the proposed comprehensive model will be discussed in detail.

4.1 Asphaltene Models

Asphaltene is an important component in crude oils. During oil production process, asphaltene may be flocculated or precipitated out from crude oils causing undesirable problems. Asphaltene problems can occur in various locations as shown in Figure 4-1 for a vertical production well system. It is therefore crucial to be able to accurately determine the conditions where solid asphaltene possibly exists in the flow system.

In order to determine asphaltene precipitation conditions the nature of asphaltene and its precipitation mechanisms must be well understood. Joshi et al.

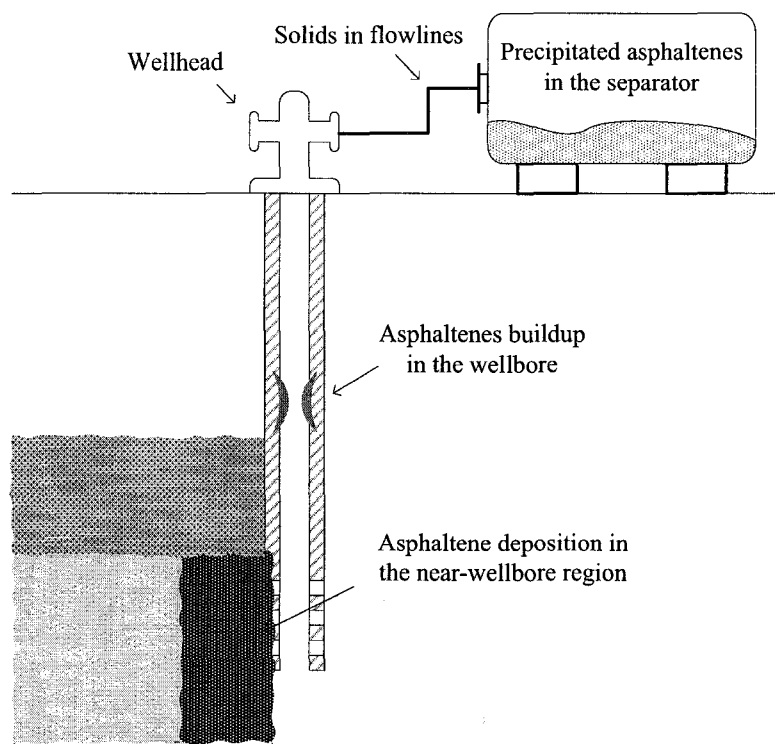


Figure 4-1: Possible locations for asphaltene deposition in oil production [26].

categorized asphaltene into “lab” and “field” asphaltenes [31]. The lab asphaltenes are defined mainly by their solubility. They are the substance in crude oils which are insoluble in excess normal alkanes (straight chain alkanes) but soluble in excess benzene and toluene at room temperature [4]. By this definition asphaltenes consist of a range of high molecular weight hydrocarbons with complex aromatic ring structures containing O, N, S, and heavy metals and are considered the most aromatic part of the crude [4].

The field asphaltenes on the other hand are defined as the substance precipitating from depressurization of live crude oils [31]. As opposed to the lab asphaltenes where precipitation is induced by addition of solvent the field-asphaltene precipitation is influenced by interaction between asphaltene and other substances in the oil mixture. Under reservoir conditions asphaltene is present in crude oils by being

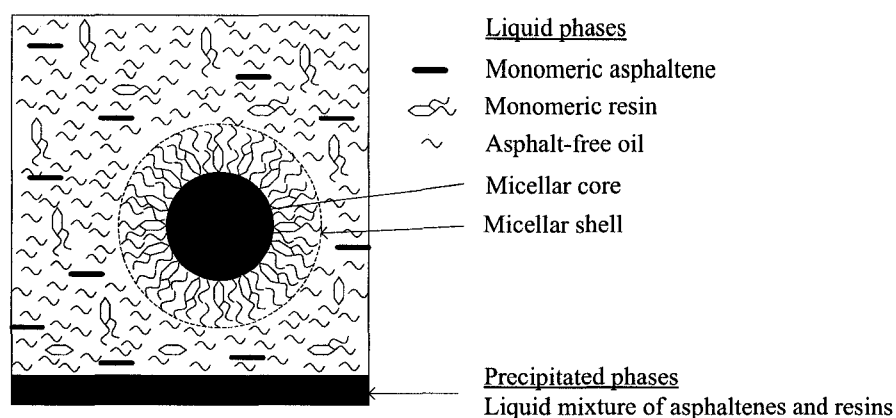


Figure 4-2: Schematic representation of a crude/micelle/precipitate system [20].

dissolved in oil solution or suspended as colloidal particles [31]. Prior to the precipitation, asphaltene micelles are stabilized by stabilizer (resins) keeping asphaltene dispersing in alkane solvents (Figure 4-2). The system can therefore be viewed as a mixture governed by non-polar hydrocarbons (e.g. paraffins, naphthenes, and aromatics of moderate molecular weight) and polar polyaromatic materials (resins and asphaltenes) [20]. The polar nature and the dynamic stability of the mixture defines asphaltene precipitation [20].

Various studies have been reported in the literature describing conditions leading to destabilization of these stabilized micelles and in turn precipitation. For example Burke et al. provided experimental data on asphaltene precipitation for various oil-solvent mixtures to determine the effects of pressure, temperature, and composition on asphaltene precipitation [11]. Hirschberg et al. used a thermodynamic model to predict precipitation and compare the results with experimental data [25]. The data included onset precipitation obtained by titration with various liquid and gas solvents as well as precipitate amounts measured at different conditions and solvent concentrations. Rydahl et al. conducted experiments to determine the effects of pressure and composition on asphaltene precipitation [52].

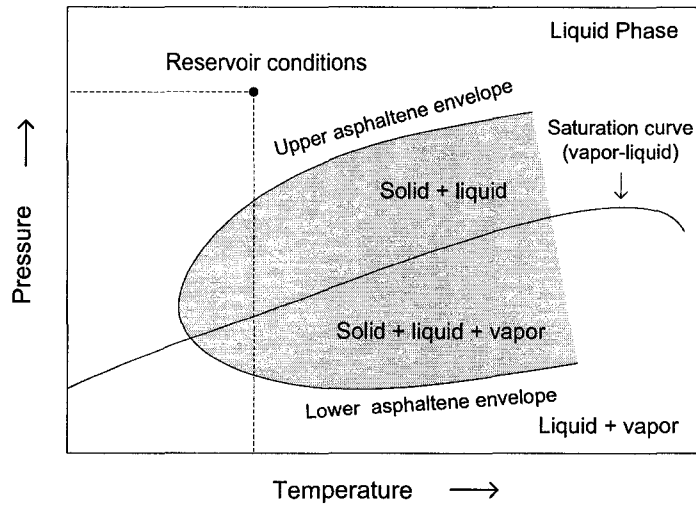


Figure 4-3: Asphaltene phase envelope (modified from Nghiem and Coombe [39]).

Overall, all the studies are consistent. The destabilization of asphaltene system is caused by “expanding” crude oil or when “interparticle distances” of asphaltenes are increased [60]. The interparticle expansion is generally caused by the change in pressure, temperature, and/or composition. To consider the effect of pressure and temperature on asphaltene precipitation consider asphaltene envelope for a constant-composition system (Figure 4-3). The asphaltene envelope is constructed of the pressures where precipitation is first detected (onset pressures) at each specified temperature [26]. The decrease in pressure may either increase or decrease the amount of the precipitates. When an undersaturated oil (consider the point representing reservoir conditions in Figure 4-3) is depressurized asphaltene starts to precipitate when the pressure reaches the upper asphaltene envelope. At this point the pressure of the fluid is still higher than its saturation pressure represented by the liquid-vapour saturation curve. The amount of precipitate increases with decreased pressure and reaches a maximum value somewhere near the saturation pressure. The amount of precipitates decreases again after the pressure is reduced below the saturation pressure. The reduction in precipitate amount is due to the change in fluid composition. Below the saturation pressure, gas (containing light

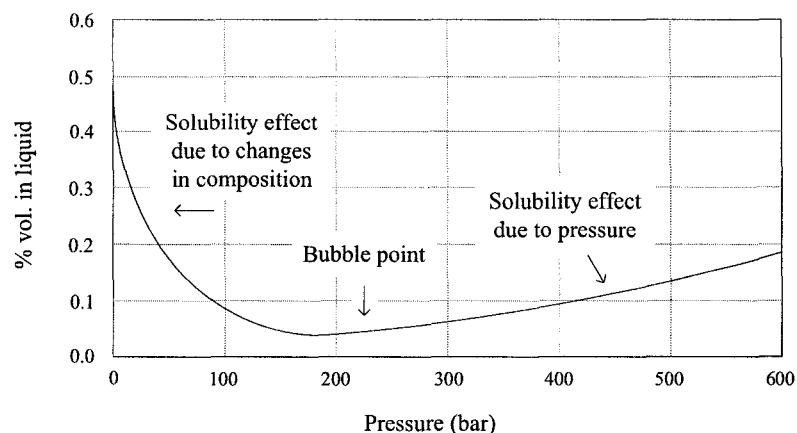


Figure 4-4: Asphaltene solubility based on a thermodynamic model [18].

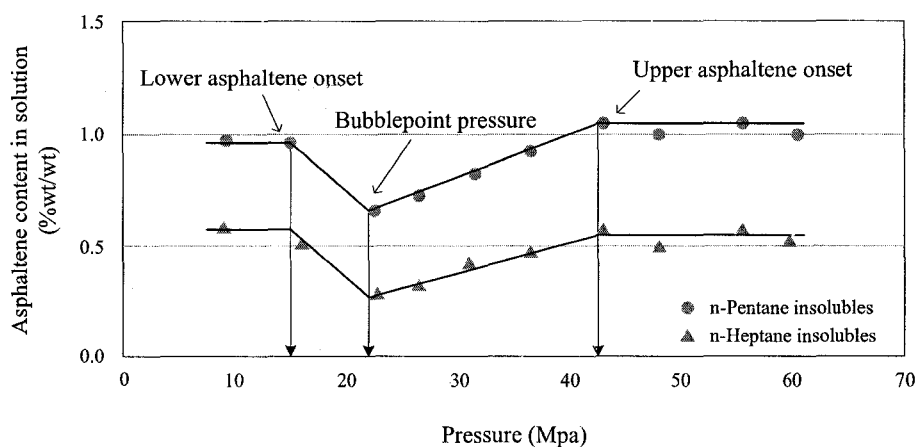


Figure 4-5: Asphaltene content and pressure relation from gravimetric method [26].

alkanes) evolves from the oil thus reducing the light components which act as solvents in the liquid phase. This leads to an increase in asphaltene solubility. Figure 4-4 showing asphaltene solubility as a function of pressure and Figure 4-5 showing the experimental data of precipitate amount at different pressures confirm the aforementioned precipitation phenomenon.

Temperature has a weaker effect on asphaltene precipitation [20]. Burke et al. conducted experiments to study the effects of temperature on asphaltene precipitation and found that asphaltene solubility increases with decreased temperature [11]. Figure 4-6 shows experimental results where less precipitates are detected at lower

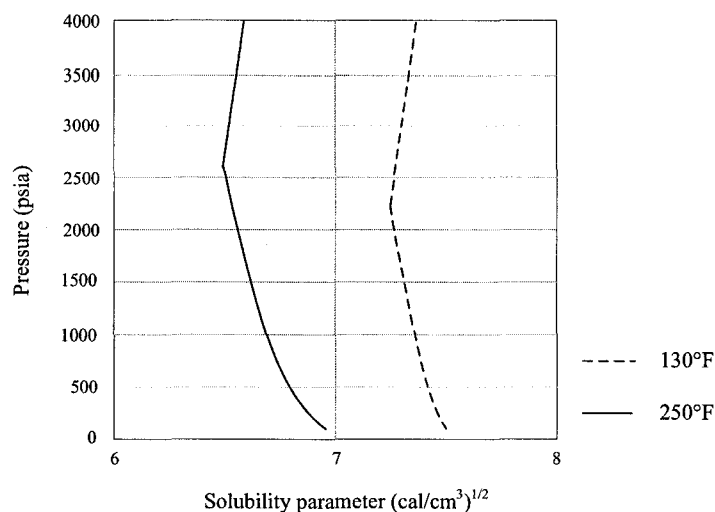


Figure 4-6: Effect of temperature on asphaltene solubility (adapted from Burke [11]).

temperature [11]. This is consistent with the observation that precipitation occurs when asphaltene interparticle distances increase.

Composition is also known to play a major role in asphaltene precipitation. Because of the asphaltene insolubility, precipitation tends to occur in crudes with high content of light hydrocarbons [20]. In oil production, changes in fluid composition can be caused by commingling of incompatible fluids, acid stimulation, and/or gas injection as in enhanced oil recovery process [26]. These alter solvent fractions in the crude oil and may lead to precipitation of asphaltene.

Other important characteristics of asphaltene precipitation are reversibility and supersaturation. These characteristics are very important in understanding precipitation mechanisms and in validating predictive asphaltene models. It was found that in field operation asphaltene shows some supersaturation before it actually precipitates [18]. The degree of supersaturation depends on the nature of the crude and the conditions in the well including temperature and turbulence [18]. With regard to asphaltene reversibility different assumptions are used in asphaltene models. Thermodynamic theories assume fully reversible processes while the colloidal theory

considers precipitation as an irreversible process [45]. Joshi et al. mentioned that asphaltene is “largely” reversible according to laboratory analysis [31]. From the studies when the pressure of the sample was reduced and then increased back to the original pressure within minutes the asphaltene were deflocculated [31]. However some irreversibility was shown in the crude that experienced depressurization for the first time [31]. Peramanu et al. investigated the effects of solvent concentration and temperature on the reversibility of asphaltene precipitation [45]. It was found from the study that solvent addition was able to reverse asphaltene precipitation provided sufficient agitation that broke up asphaltene particles. On the other hand temperature treatments only showed partial reversibility of asphaltene precipitates [45].

In primary production, there is no change in fluid composition. As a result, pressure and temperature conditions are the main factors determining precipitation in live crude oils. When a fluid is produced from a reservoir it undergoes pressure reduction and possibly destabilization of the asphaltene colloidal system. Asphaltene precipitation in primary production generally occurs above the bubble point pressure of the fluid [2]. Typically asphaltene precipitation is found in highly compressible reservoir fluids. Such precipitation also commonly occurs in undersaturated fluids where the reservoir pressure is much higher than the bubble point pressure. Higher reservoir pressure relative to the bubble point pressure implies that the fluid will experience a large pressure drop before the gas starts to evolve. Fortunately being undersaturated and experiencing high degree of depressurization do not ensure that all fluids exhibit asphaltene problem during productions. The crudes which have potential of forming solid asphaltene (unstable crudes) share some other characteristics. They are normally light crudes with high C_1 – C_3 contents and with relatively low C_{7+} content [36]. de Boer et al. showed strong correlations between asphaltene

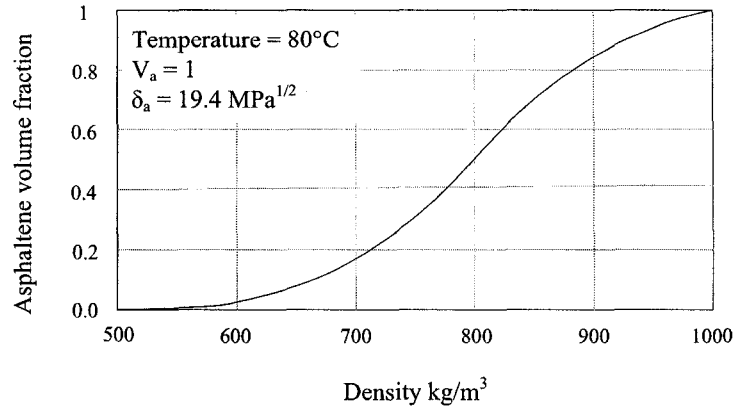


Figure 4–7: Asphaltene solubility as a function in-situ density of a crude with asphaltene solubility parameter (δ_a) of $19.4 \text{ MPa}^{1/2}$ (adapted from de Boer et al. [18]).

precipitation and in-situ oil density where asphaltene solubility increases with increased in-situ density (Figure 4–7) [18]. Rogel et al. found that unstable crudes also show high hydrogen deficiency, high aromaticity, and high degree of aromatic ring condensation [51].

Once asphaltene precipitates it is still not certain that it will cause a problem to the oil production. For example if the produced particles do not deposit or the particles are small enough to pass through all the openings without plugging the flow path, they are not of any concern. The conditions where the precipitates start to cause problems and the extent of the problems depend on the nature of the fluid and can be determined by various methods. In practice to avoid precipitation in field operation “preliminary instability screening” of the crude is first conducted to determine the tendency of a crude to precipitate asphaltene and the asphaltene formation behaviour [28]. Typically basic reservoir and stock-tank oil properties are first determined from PVT analysis. The stability of the oil may then be determined using various types of empirical instability tests [28]. These instability correlations use different indices and plots to help indicate severity of asphaltene problem in the field. de Boer et al. proposed a screening method based

on Flory and Huggins's thermodynamic model of asphaltene solubility and Hildebrand's solubility parameters [18]. The asphaltene solubility was correlated to oil properties including the aforementioned in-situ density and oil molar volume. An empirical plot was also developed where the degrees of severity of asphaltene problems were distinguished by the maximum supersaturation (occurring at the bubble point of various oil samples) lines as shown in Figure 4-8(a). From the plot the crudes that cause severe problems were light crudes and those with no problems did not have high undersaturation experience. Other available tests to verify the stability of crude oils include asphaltene to resin ratio, Baker-Petrolite's colloidal instability index, and Schlumberger's asphaltene instability index criterion (Figure 4-8(b)-4-8(d)) [28]. In addition experiments may also be used to determine the asphaltene onset and precipitation amount at different conditions. These experiments include fixed-wavelength isothermal depressurization where asphaltene instability is determined based on light transmittance as the fluid is continuously depressurized and high-pressure-high-temperature (HPHT) filtration techniques that quantify the amounts of precipitate [28]. The prescreening and asphaltene characterization methods are found to be sufficient to determine asphaltene problem potential and the necessity for further analysis [28]. The precipitation mechanisms in field operations however also depend on other factors such as turbulence in the well [18]. Thus other measures are available and may also be required to ensure problem-free operations.

These measures include addition of asphaltene inhibitors (or stabilizers) to stabilize asphaltene. These chemicals consist of acid compounds that exhibit amphiphilic characteristics. They act in a similar manner as resins and help keep asphaltene colloids suspending in the solution [51]. Different types of inhibitors are used depending on the characteristics of each specific crude.

In addition to the above measures, attempts were also given to the development of predictive thermodynamic and kinetic models to describe asphaltene formation

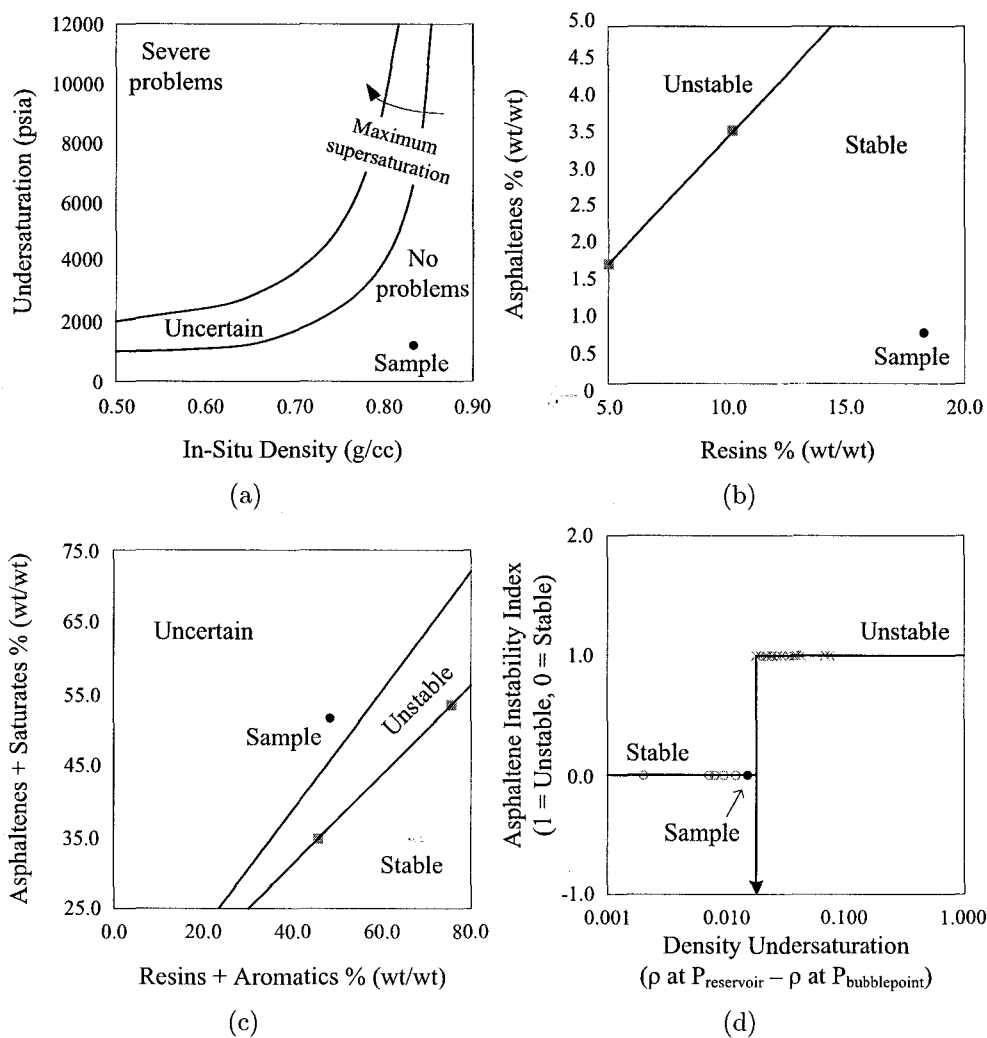


Figure 4-8: Asphaltene instability correlations: (a) de Boer plot (b) Asphaltene resin ratio (c) Colloidal instability index (d) Asphaltene instability index (adapted from de Boer et al. [18] and Jamaluddin et al. [27]).

behaviour at different conditions. As the prediction of the precipitation is a complex task and based on complex thermodynamics, it is difficult to develop an accurate model. In order to predict asphaltene precipitation the available asphaltene models use different assumptions and thermodynamic concepts in attempt to match the calculated results to real precipitation behaviour. Qin et al. broadly divided these models into four categories namely solubility, micellization, colloidal, and solid models [47]. Firstly solubility models are based on the “simplified Flory-Huggins theory” of polymer solution. The liquid phase is considered to be a mixture of three components: asphaltene, resin, and the remaining oil and solvent and the reversible solution equilibrium is used to describe asphaltene stability [47]. Models under this category include those proposed by Hirschberg et al. [25], Kawanaka et al. [34], Cimino et al. [14], Park and Mansoori [42], and Nor-Azlan and Adewumi [41]. Secondly thermodynamic colloidal model is based on statistical thermodynamics and colloidal science. In this model, proposed by Leontaritis and Mansoori [37] and later completed by Park and Mansoori [42], asphaltenes are assumed to be present in the oil as solid colloidal particles which are stabilized by resins adsorbed on their surface. The model is more applicable to the cases where dissociation of asphaltene micelles is involved [40]. The third category includes thermodynamic micellization models which consider asphaltenes as molecules that form a micelle core with stabilizers (resin molecules) absorbing onto its surface. The structure and concentration of the micelle are obtained based on the concept of the Gibbs free energy minimization [47]. Finally solid models are based on the assumption that the precipitating asphaltene is in pure solid phase (containing only asphaltene component). The solid phase is therefore handled independently of the other phases in the system. Examples of earlier studies on solid models include reports by Gupta [22] and Thomas et al. [58]. The difficulty in applying this type of model is that empirical parameters may be required and tuning the model with experimental data can be excessive [40]. More

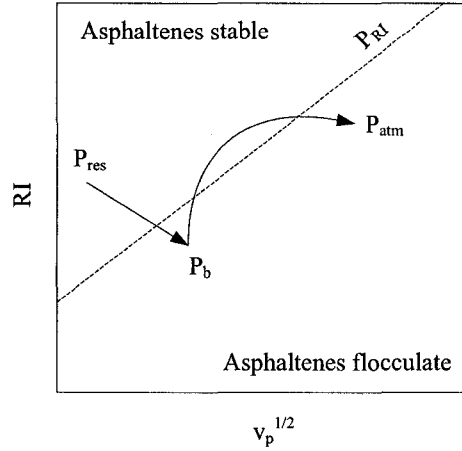


Figure 4-9: Asphaltene precipitation behaviour determined using refractive index [62].

recent solid models (such as Nghiem and Coombe [39] and Li et al. [38]) provide simple yet accurate models for asphaltene precipitation and is more applicable to modeling in oil production processes.

Even though various models are available for predicting asphaltene precipitation, rigorous use of experimental data to calibrate the models is inevitable due to fluid-specific nature of asphaltene precipitation. Wang and Buckley recently proposed an experimental approach to anticipating asphaltene problem and prediction of asphaltene precipitation at reservoir conditions [61]. In their studies asphaltene instability was determined based on the refractive index (RI) at asphaltene onset conditions. Refractive indices of stock tank oil samples were measured where asphaltene precipitate started to form. The onset refractive index (P_{RI}) for each particular oil had a linear relationship with square root of molar volume of precipitant (i.e. light fractions of the oil samples). In order to predict asphaltene precipitation at reservoir conditions, the onset refractive index (P_{RI}) were first translated to the value at the reservoir temperature assuming that P_{RI} was a linear function of temperature (e.g. that P_{RI} shifted by $-0.0008 RI$ units per $^{\circ}C$ [62]). The refractive index of the oil was then calculated from PVT data. To calculate the value, parameters

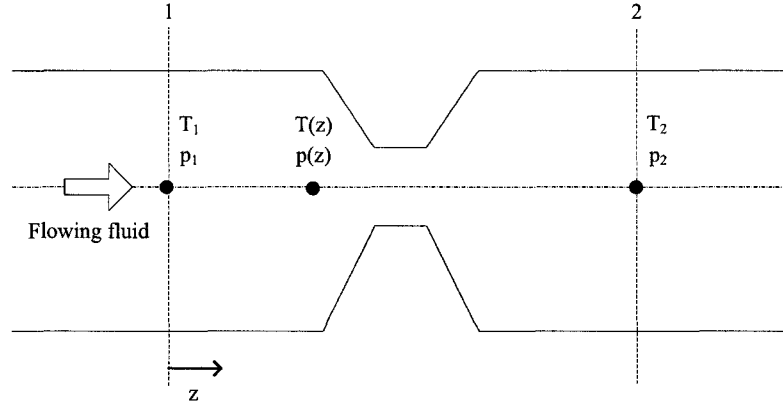


Figure 4-10: Simplified restriction geometry.

including R_s , B_o , and RI of dead oil were required [61]. The value was compared with P_{RI} at the same pressure and temperature. Asphaltene precipitate appeared if the RI was lower than P_{RI} as illustrated in Figure 4-9.

In this research an asphaltene model developed based on a solid-asphaltene model is used. The following sections describe how the proposed asphaltene model was formulated in detail. The model was designed to predict asphaltene precipitation specifically in the case of restricted flow paths. In order to do so, the model was divided into two main parts. The first is the calculation of pressure condition upon which temperature and phase behaviour calculations are based. Using the calculated pressures the temperature and phase behaviour are determined in the second part of the model using isenthalpic-flash phase equilibrium calculations.

4.2 Pressure Calculations

In order to predict pressure drop across a restriction the actual restriction is first modeled using a simplified geometry (Figure 4-10). Based on that, the conservations of mass and momentum presented in Chapter 2 are used to derive a one-dimensional equation to calculate the pressure profile inside the restriction ($p(z)$ in the Figure) starting from point 1 up to point 2. Recall Equation 2.17 for the momentum balance:

$$\frac{dp}{dz} = -\frac{\dot{m}}{A} \frac{dv}{dz} - \frac{f\rho v^2}{2D_h} - \rho g \sin \theta \quad (2.17)$$

Consider also the conservation of mass ($\dot{m} = \rho A v = \text{constant}$). The acceleration term in the above equation reduces to:

$$\begin{aligned} \frac{\dot{m}}{A} \frac{dv}{dz} &= \frac{\dot{m}}{A} \frac{d}{dz} \left(\frac{\dot{m}}{\rho A} \right) \\ &= \frac{\dot{m}^2}{A} \frac{d}{dz} \left(\frac{1}{\rho A} \right) \\ &= \frac{\dot{m}^2}{A} \left[\frac{1}{A} \frac{d}{dz} \left(\frac{1}{\rho} \right) + \frac{1}{\rho} \frac{d}{dz} \left(\frac{1}{A} \right) \right] \\ &= \frac{\dot{m}^2}{A} \left[-\frac{1}{A} \frac{1}{\rho^2} \frac{d\rho}{dz} - \frac{1}{\rho} \frac{1}{A^2} \frac{dA}{dz} \right] \end{aligned} \quad (4.1)$$

Neglecting changes in elevation the momentum balance equation becomes:

$$\frac{dp}{dz} = \frac{\dot{m}^2}{A} \left(\frac{1}{A} \frac{1}{\rho^2} \frac{d\rho}{dz} + \frac{1}{\rho} \frac{1}{A^2} \frac{dA}{dz} \right) - \frac{f\rho v^2}{2D_h} \quad (4.2)$$

After some rearrangement we get the equation which can be used for calculating pressure drop, provided that $\frac{d\rho}{dz} = \frac{d\rho}{dp} \frac{dp}{dz}$:

$$\frac{dp}{dz} = \frac{\frac{\dot{m}^2}{A^3} \frac{1}{\rho} \frac{dA}{dz} - \frac{f\dot{m}^2}{2\rho A^2 D_h}}{1 - \frac{\dot{m}^2}{A^2} \frac{1}{\rho^2} \frac{d\rho}{dp}} \quad (4.3)$$

The above equation is applicable for predicting pressure drop in sub-critical flows where the pressure is a function of position (z) in the flow direction. In order to include the effect of fluid compressibility Equation 4.3 is modified with the use of the real gas equation:

$$\begin{aligned} pV &= ZnRT \\ &= \frac{m}{MW} ZRT \end{aligned} \quad (4.4)$$

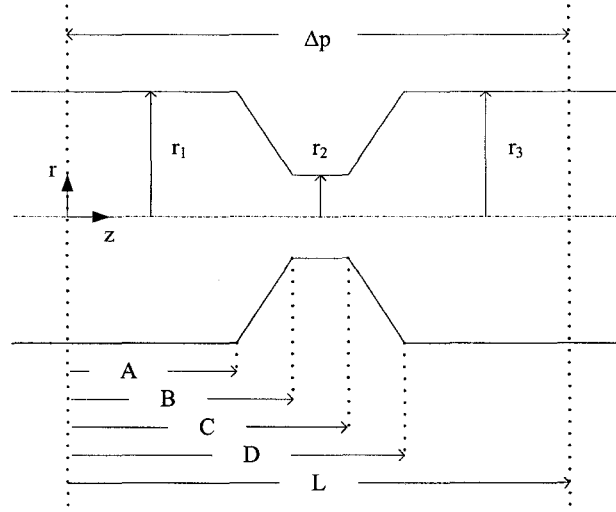


Figure 4-11: Restriction geometry divided into sections.

where MW is the average molecular weight and m is the mass of fluid. From the above equation density can now be written as a function of pressure:

$$\rho = \frac{m}{V} = \frac{pMW}{ZRT} \quad (4.5)$$

and the derivative of density with respect to pressure is:

$$\frac{d\rho}{dp} = \frac{MW}{ZRT} \quad (4.6)$$

Substituting the above two expressions into Equation 4.3 gives:

$$\frac{dp}{dz} = \frac{\frac{\dot{m}^2}{A^3} \frac{1}{p} \frac{dA}{dz} - \frac{1}{p} \frac{f \dot{m}^2}{2A^2 D_h}}{\frac{MW}{ZRT} - \frac{\dot{m}^2}{A^2} \frac{1}{p^2}} \quad (4.7)$$

assuming that the compressibility factor (which is also a function of pressure) is constant provided a sufficiently small pressure drop (dp).

Consider the simplified flow geometry in Figure 4-11 the general expression for the circular cross-sectional area in terms of position (z) in the flow direction can be

written:

$$A(z) = \begin{cases} \pi r_1^2 & 0 \leq z \leq A \\ \pi \left(\frac{r_2 - r_1}{B - A} (z - A) + r_1 \right)^2 & A \leq z \leq B \\ \pi r_2^2 & B \leq z \leq C \\ \pi \left(\frac{r_3 - r_2}{D - C} (z - C) + r_2 \right)^2 & C \leq z \leq D \\ \pi r_3^2 & D \leq z \leq L \end{cases} \quad (4.8)$$

where the symbols in the above equations are defined as in Figure 4-11.

Given the expression for cross-sectional area as a function of z Equation 4.7 can now be used to calculate the pressure drop for compressible single-phase fluids. For gases (compressible fluids), Equation 4.9a (rearranged from Equation 4.7) is used. For the liquid phase where its compressibility is considered low compared with that of the gas Equation 4.9b is used. Note that the equation for the liquid phase is a special case of Equation 4.3 where $\frac{d\rho}{dp} = 0$.

Single-phase gas:

$$p \left(\frac{MW}{ZRT} - \frac{\dot{m}^2}{A^2 p^2} \right) dp = \left(\frac{\dot{m}^2}{A^3} \frac{dA}{dz} - \frac{f \dot{m}^2}{2A^2 D_h} \right) dz \quad (4.9a)$$

Single-phase liquid:

$$\frac{dp}{dz} = \frac{\dot{m}^2}{A^3} \frac{1}{\rho} \frac{dA}{dz} - \frac{f \dot{m}^2}{2\rho A^2 D_h} \quad (4.9b)$$

In order to calculate pressure drop for two-phase (liquid-gas) fluid Equation 4.9a can be used by substituting two-phase properties for all the single-phase parameters as in Equations 4.10. The subscript TP 's represent two-phase properties.

$$p \left(\frac{MW}{Z_{TP}RT} - \frac{\dot{m}^2}{A^2 p^2} \right) dp = \left(\frac{\dot{m}^2}{A^3} \frac{dA}{dz} - \frac{f_{TP} \dot{m}^2}{2A^2 D_h} \right) dz \quad (4.10)$$

In the above equation the two-phase compressibility factor (Z_{TP}) can be evaluated before the simulation. The two-phase friction factor (f_{TP}) is considered the tuning parameter to match the proposed model with experimental data or correlations. Where experimental data or values of the friction factors for single-phase liquid and

single-phase gas (f_L and f_G respectively) are available an f_{TP} can be calculated as a weighted average of the liquid and gas friction factors:

$$f_{TP} = (\alpha_L f_L^e + \alpha_G f_G^e)^{1/e} \quad (4.11)$$

where α is the phase volume fraction for each phase. Once the constant e is specified calculations of two-phase friction factors at conditions other than those available from the experiments can be calculated.

For commercial valves, downstream pressure is generally the parameter of interest. Pressure drops across these valves are determined by manufacturers using extensive experiments. The pressure drop and flow rate through the valve are then related through “resistance coefficient”, “equivalent length”, or ‘flow coefficient’ concepts [15]. These coefficients are valve characteristics and independent of flow conditions. The coefficients account for combined pressure drop consisting of pressure loss through the valve itself and the additional pressure drops at the upstream and downstream locations of the valves. In practice a flow control valve is selected or designed to provide the required pressure drop in oil production process for production control purposes. The valve coefficient (C_v) is used to determine the pressure drop across the valve and may be obtained by combination of various means (e.g. analytical modeling, experiments, etc.) [35]. The value of C_v accounts for complex geometry and flow behaviour such as non-circular shape of the valve opening, multi-phase flow, and non-Newtonian fluid behaviour. The general correlation for pressure drop used in the case of commercial valves is [35]:

$$q_L = C_v \sqrt{\frac{\Delta p}{\gamma_L}} \quad (4.12)$$

where C_v is the rate of water flow at one unit pressure drop, q_L is the fluid flow rate through the valve and γ_L is the specific density of the fluid (water = 1). The two-phase friction factor (f_{TP}) in Equation 4.10 can be related to the coefficient

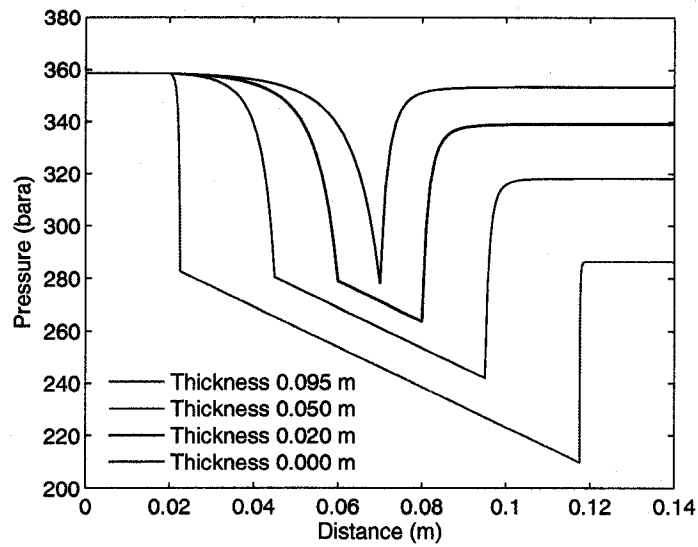


Figure 4-12: Pressure drop prediction using Equation 4.9b for single-phase liquid.

(C_v) and used to calculate pressure inside the restriction required in the proposed model.

Figure 4-12 and Figure 4-13 present the pressure drop calculations, using Equation 4.9b and Equation 4.10 respectively, for various thicknesses of valve throats (the length from B to C in Figure 4-11). Other geometry parameters were kept the same for all analyses. In both cases the pressure drops were calculated for a flow rate of 3,442 m³/d and a friction factor of 0.2. For the two-phase calculation, a Z_{TP} of 1 was used. The pressure was evaluated by taking a small increment along the length of the restriction (Δz). The pressure was calculated at each local point inside the restriction (shown as dots on the pressure plots in Figure 4-13).

In order to investigate the effects of the Z_{TP} and f_{TP} on pressure predictions, some analyses were conducted. Consider a symmetric valve with $D_1 = D_3$. Physically, regardless of friction, the pressures upstream and downstream of such a valve should be equal. However, Figure 4-14(a) shows that the increase in the compressibility factor (Z_{TP}) tends to decrease the accelerational pressure drop (evaluated at $f_{TP} = 0$) and in turn increase the downstream pressure when Equation 4.10 was

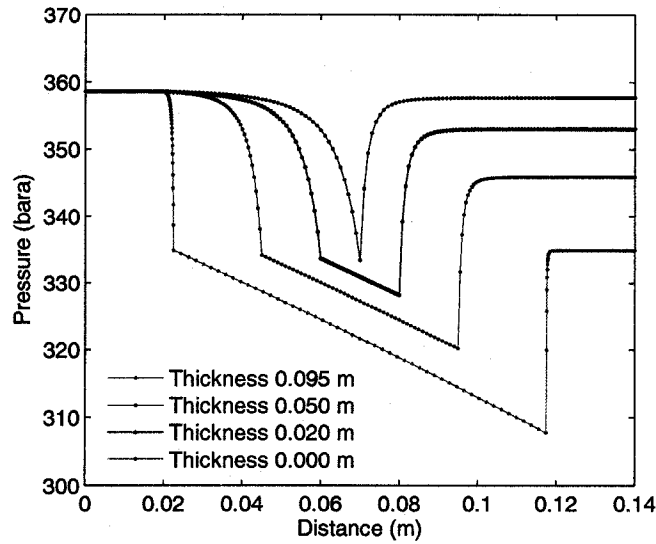


Figure 4-13: Pressure drop prediction using Equation 4.10 for two-phase fluid.

used and solved numerically along the length of the restriction. On the other hand the increase in the friction factor (f_{TP}) tends to decrease the downstream pressure as shown in Figure 4-14(b). For this reason the f_{TP} is used in this model as the tuning parameter to match the actual valve pressure drop.

It was found from the analysis that the f_{TP} is not independent of fluid properties. For fluids with a low Z_{TP} (0.6 and 1.0 in Figure 4-14) the downstream pressure decreased proportionally with the Z_{TP} when the same value of f_{TP} was used. For example for an f_{TP} of 0.1 the downstream pressure decreased more in the fluid with a Z_{TP} of 1.0 than that with a Z_{TP} of 0.6. This however did not apply to fluids with higher values of Z_{TP} . For these fluids a higher f_{TP} was required in order to reduce the downstream pressure. As seen in Figure 4-14(b) the friction factor of 0.1 did not cause the downstream pressure of the fluid with a Z_{TP} of 1.4 to decrease as much as it did to the other two fluids. As seen in Figure 4-15 for compressibility factors of the example fluid (as in Appendix A) at 100 °C, the two-phase (total) Z factor increases with pressure and ranges from approximately 0.7 to 1.4. It can therefore be concluded from the analysis that even though the tuning of the model

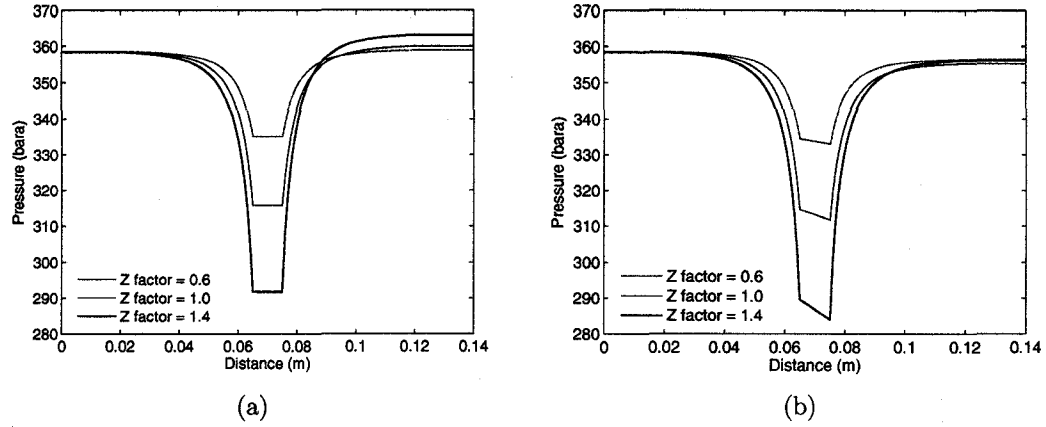


Figure 4-14: Effects of friction and acceleration on the pressure drop for different compressibility factors. (a) accelerational pressure drop (friction factor = 0). (b) frictional pressure drop (friction factor = 0.1).

may become more difficult as the pressure becomes closer to the saturation pressure (high Z_{TP}), Equation 4.10 can still be used to model two-phase fluids of various values of Z_{TP} by using the f_{TP} as the tuning parameter.

In the proposed model Equation 4.9b is used for pressures higher than the fluid's saturation pressure or Equation 4.10 is used. Using the appropriate equation, pressures at any locations inside the valve restriction ($p(z)$) can be calculated by dividing the restriction length (L) into a finite number of length steps (N_v) along the flow direction. The pressure of the fluid can be determined at each length (z) starting from the upstream of the restriction ($z = 0$) up to the N_v^{th} point at the downstream location of the restriction ($z = L$). Based on the calculated pressures, phase behaviour analysis can be performed as will be described in the next section. The phase behaviour calculations are conducted at each of the N_v points at the corresponding pressure and result in profiles of local temperature and fluid phase behaviour along the restriction.

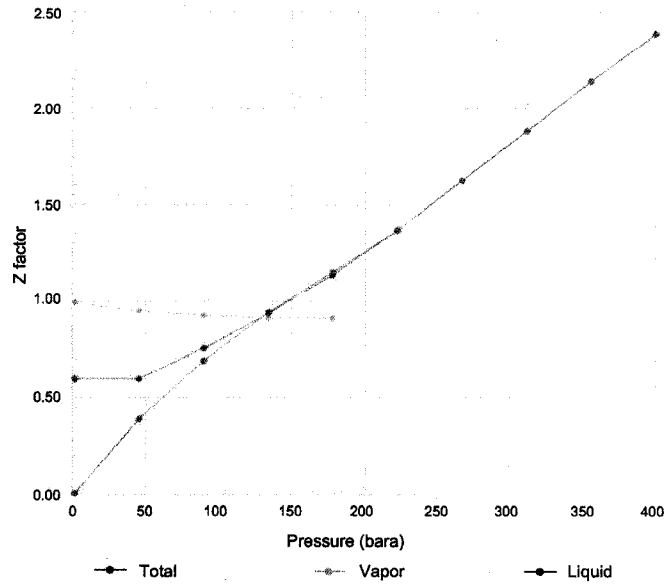


Figure 4-15: Variation of compressibility factor of example fluid at 100 °C [12].

4.3 Temperature and Phase Equilibrium Calculations

For multi-phase mixtures the change in temperature may occur when fluids flow through a restriction. Even though the gas fraction in the fluid mixture expands when the pressure decreases there is little time for heat transfer and thus the process is considered adiabatic [46]. For this reason the enthalpy of such system can be considered constant [46]. In this proposed asphaltene model an isenthalpic flash is therefore used instead of an isothermal flash to reflect the adiabatic nature of the process. This allows temperature to be calculated along with the fluid phase behaviour.

4.3.1 Isenthalpic Flash

Using an isenthalpic flash an additional energy conservation equation is required in addition to the equations used in the isothermal flash described in the previous chapter. As opposed to specifying pressure and temperature, pressure (calculated from the previous section) and enthalpy are specified. The phase equilibrium calculations using an isenthalpic flash algorithm can be performed using a few different

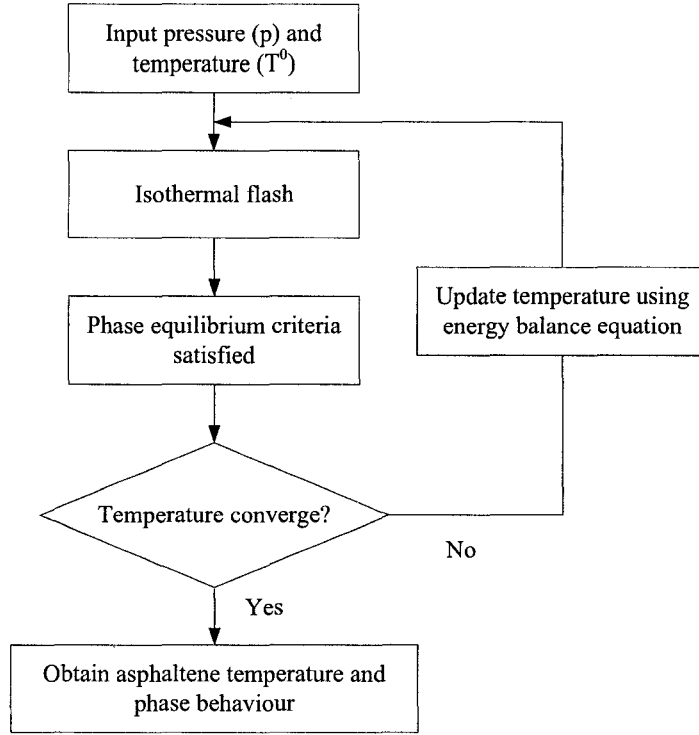


Figure 4–16: Isenthalpic flash schematic.

schemes [1]. One way is to use a series of conventional isothermal flash calculations at a specified pressure and various temperatures. The solution is obtained at the temperature where the conservation of enthalpy condition is satisfied. An isothermal flash calculation can also be performed at an intermediate temperature and allows the temperature to be updated in an outer loop using the energy balance equation (Figure 4–16). The latter scheme is used in this research and will be outlined here in detail. According to Figure 4–16 an isothermal flash is first performed on a mixture of known feed composition (z_i) at the pressure of interest ($p(z)$ inside the valve in this case) and an intermediate temperature (T^0). After the isothermal convergence is achieved, the energy balance is evaluated. The condition for energy balance is that the enthalpy of the fluid system is conserved [1]:

$$\begin{aligned}
 g_{n_p} &\equiv H - H_{sys} \\
 &= 0
 \end{aligned}
 \tag{4.13}$$

where H_{sys} is the specific molar enthalpy of the system and H is the fluid enthalpy evaluated in each isenthalpic-flash iteration.

The enthalpy of fluid (H) at different conditions can be calculated as the sum of the ideal-gas enthalpy (H^{id}) and the residual enthalpy (H^{res}) [12]:

$$H = \sum_i z_i H_i^{id} + H^{res} \quad (4.14)$$

where z_i is the mole fraction of component i in the considered phase (liquid or gas) and H_i^{id} is the ideal-gas enthalpy of component i defined as [12]:

$$H_i^{id} = \int_{T_{ref}}^T C_{p_i}^{id} dT \quad (4.15)$$

The integral is from the reference temperature (T_{ref}) to the system temperature (T). The molar ideal-gas heat capacity of component i ($C_{p_i}^{id}$) is a function of temperature and can be obtained from correlations [12]. The residual enthalpy can then be calculated using an equation of state from:

$$H^{res} = \sum_j F_j \sum_i y_{ij} H_{ij}^{res} \quad (4.16)$$

where F_j is the mole fraction of phase j ; y_{ij} and H_{ij}^{res} are the mole fraction and residual enthalpy of component i in phase j , respectively. The residual enthalpy H_{ij}^{res} can be obtained using an equation of state as [12]:

$$H_{ij}^{res} = -RT^2 \frac{\partial \ln \phi_{ij}}{\partial T} \quad (4.17)$$

In this research the system enthalpy (H_{sys}) is the initial fluid enthalpy at the upstream location of the restriction. The value is calculated at the upstream pressure and temperature using Equation 4.14 prior to the compositional simulations. For every value of temperature in the isenthalpic-flash iteration, the fluid enthalpy (H) is evaluated using the same equation. The resulting value is compared with H_{sys} to determine whether the energy balance condition is satisfied. If not so, the

temperature is updated using the equation below for the first iteration [1]:

$$T^1 = T^0 - \frac{g_{n_p}^0}{\sum_{j=1}^{n_p} F_j C_{pj}} \quad (4.18)$$

where C_p is the specific heat capacity of phase j . For the following iterations the equation below is used [1]:

$$T^{k+1} = T^k - \frac{g_{n_p}^k (T^k - T^{k-1})}{g_{n_p}^k - g_{n_p}^{k-1}} \quad (4.19)$$

where k is the iteration index.

The process of updating temperature is repeated until the energy balance is satisfied ($g_{np}=0$). As the enthalpy (H) in Equation 4.13 is a monotonically increasing function of temperature, a valid temperature range may be specified and used to check the temperature T^k in each iteration [1]. If T^k lies outside the temperature range Agarwal et al. suggested a Regula Falsi update (Equation 4.20) instead of Equation 4.19 to update the temperature [1].

$$T^{k+1} = T^k - \frac{g_{n_p}^k (T^L - T^U)}{g_{n_p}^L - g_{n_p}^U} \quad (4.20)$$

where L and U indicates the values for the lower and the upper bounds respectively.

The use of the isothermal flash in the inner loop permits stability analysis to be conducted and determine the correct number of phases in each of the iterations. More details of possible isenthalpic flash schemes and their limitations can be found in Agarwal et al. [1].

4.3.2 Pseudo-Three-Phase Equilibrium Calculation

This section presents the isothermal phase equilibrium calculation for liquid-vapour-solid systems to be used in the inner loop of the isenthalpic flash presented in the previous section. As mentioned before the asphaltene model proposed in this

research is a solid-type asphaltene model. The model uses a pseudo-three-phase equilibrium calculation where vapour-liquid phase equilibrium calculations are conducted independently from the solid phase. Because of the possible solid formation the fluid is characterized a little differently in this model according to asphaltene characterization approach proposed by Nghiem et al. [40]. As not all the compounds in the heaviest fraction precipitate (e.g. resins and paraffins may not precipitate) Nghiem et al. considered asphaltene as a part of the heaviest fraction [40]. This is consistent with Rydahl et al. where the experiments suggested that asphaltene can be considered the aromatic part of C_{50+} compounds [52]. Based on that, the heaviest component in the oil can therefore be split into two components: precipitating and non-precipitating. The precipitating component is considered a dense asphaltene phase composed of only one pure component (asphaltene) [47]. The properties and acentric factors of the two components are equal but the binary interaction coefficients are different. The precipitating component has larger interaction coefficients with light components which means that the component is more likely to form an asphaltene phase.

Using the above characterization approach, the heaviest (n^{th}) component in an n -component mixture can be divided into two components: non-precipitating component (C_{nA}) and precipitating component (C_{nB}) [38]. This leads to the total of $n+1$ components in the mixture. By using the average weight fraction of asphaltene from precipitation experiments the mole fraction of the precipitating component (C_{nB}) in the mixture can be estimated [39]:

$$z_{C_{nB}} = \frac{W_{C_{nB}} MW_{Oil}}{MW_{C_{nB}}} \quad (4.21)$$

where $W_{C_{nB}}$ is the average weight fraction of asphaltene precipitates and MW_{oil} and $MW_{C_{nB}}$ are the molecular weight of the entire mixture and the precipitating component respectively.

Similar to the liquid-vapour phase equilibrium calculations described in Chapter 3 the mass balance equations for three-phase systems are derived with an additional solid phase provided the assumption that there are no heavy hydrocarbons (asphaltene) in the vapour phase. The mole fractions of all three phases sum up to one [38]:

$$n^V + n^L + n^S = 1 \quad (4.22)$$

where n^V , n^L , and n^S are the mole fractions of vapour, liquid, and solid phases respectively. For each component i [38]:

$$n^V x_i^V + n^L x_i^L = z_i \quad i = 1, 2, \dots, nA \quad (4.23a)$$

$$n^L x_{nB}^L + n^S x_{nB}^S = z_i \quad i = nB \quad (4.23b)$$

where x_i is the mole fraction of component i in the indicated phase (vapour, liquid, or solid). Using the equilibrium ratios defined as [38]:

$$K_i^{VL} = \frac{x_i^V}{x_i^L} = \frac{x_i^V f_i^L}{x_i^L f_i^V} \quad (4.24a)$$

$$K_{nB}^{SL} = \frac{x_{nB}^S}{x_{nB}^L} = \frac{x_{nB}^S f_{nB}^L}{x_{nB}^L f_a} \quad (4.24b)$$

the phase compositions of component i in the vapour and the liquid phases can be obtained as follows [38]:

$$x_i^V = \frac{z_i K_i^{VL}}{n^V (K_i^{VL} - 1) - n^S + 1} \quad i = 1, 2, \dots, nA \quad (4.25a)$$

$$x_i^L = \frac{x_i^V}{K_i^{VL}} \quad i = 1, 2, \dots, nA \quad (4.25b)$$

$$x_{nB}^L = \frac{z_{nB}}{1 - n^V + n^S (K_{nB}^{SL} - 1)} \quad (4.25c)$$

The material balances for vapour, liquid, and asphaltene phases can now be expressed as:

$$\sum_{i=1}^{nA} x_i^V = \sum_{i=1}^{nA} \frac{z_i K_i^{VL}}{n^V (K_i^{VL} - 1) - n^S + 1} = 1 \quad (4.26a)$$

$$\sum_{i=1}^{nB} x_i^L = \sum_{i=1}^{nB} \frac{z_i}{n^V(K_i^{VL} - 1) - n^S(K_{nB}^{SL} - 1) + 1} = 1 \quad (4.26b)$$

$$i \neq nB, K_i^{SL} = 0$$

$$i = nB, K_i^{VL} = 0$$

$$x_{nB}^S = x_{nB}^L K_{nB}^{SL} = \frac{z_{nB} K_{nB}^{SL}}{1 - n^V + n^S(K_{nB}^{SL} - 1)} = 1 \quad (4.26c)$$

In order to predict asphaltene precipitation at a pressure-temperature condition n^S is first specified. The liquid-vapour phase behaviour problem is simply solved based on the equal-fugacity criteria as presented in Chapter 3. To consider the solid phase the equal-fugacity criteria is also applied. When there is asphaltene phase in equilibrium with the other phases the fugacities of asphaltene component in the liquid and solid phases must be equal and equal to asphaltene fugacity (f_a).

$$\ln f_{nB}^L = \ln f_{nB}^S = \ln f_a \quad (4.27)$$

The asphaltene fugacity (f_a) can be calculated from the definition of the Gibbs energy. Because the asphaltene phase can be considered either liquid or solid, its fugacity can be calculated in a similar manner to that for an undersaturated pure substance. Consider Equation 3.19 and 3.20, at a constant temperature:

$$dg = vdP \quad (3.19)$$

$$dg = RTd \ln f \quad (3.20)$$

By equating the above two equations we get:

$$d \ln f = \frac{v}{RT} dP \quad (4.28)$$

Integration of the above equation from a reference state (p^*) to the condition of interest (p) gives:

$$\int_{f_a^*}^{f_a} d \ln f = \int_{p^*}^p \frac{v}{RT} dP \quad (4.29)$$

As the molar volume is constant for any liquid or solid phase the fugacity of the asphaltene phase can be expressed as:

$$\ln f_a = \ln f_a^* + \frac{v_a (p - p^*)}{RT} \quad (4.30)$$

where f_a and f_a^* are the fugacities of pure asphaltene at the pressure p and the reference pressure (p^*) respectively; v_a is the molar volume of pure asphaltene.

The reference pressure (p^*) is the condition selected from experimental data where asphaltene precipitate is present and at the thermodynamic equilibrium with other phases in the mixture. The amount of the precipitate at the selected reference condition is read directly from the experimental data. To calculate reference asphaltene fugacity (f_a^*) the amount of the precipitate is deducted from the known composition of the mixture. An equation of state is then used on the remaining mixture composition to calculate component fugacities at the reference conditions. Because the mixture is under equilibrium condition it can be concluded that the fugacity of asphaltene in the solid phase at the reference condition (f_a^*) is equal to the calculated fugacity of the precipitating component in the liquid phase (f_{nB}^L). Provided the calculated reference asphaltene fugacity, asphaltene fugacity at other conditions (f_a) can now be calculated from Equation 4.30.

Note that Equation 4.30 was derived under a constant-temperature condition. The equation therefore does not reflect variations in fugacities due to changes in temperature. Asphaltene precipitation experiments conducted at the temperature of or close to the temperature of interest are therefore required. In the case where such experimental data are not available, a relationship between reference asphaltene fugacity (f_a^*) and temperature is required in order to translate f_a^* from the temperature where the experiments are conducted to other temperature conditions. Such relationship can be developed by, for example, relating the onset asphaltene fugacities obtained from asphaltene experimental data measured at two or more

temperatures. The fugacity-temperature relation developed from the data points should be able to be extended to other temperatures at least within a specified range of temperature.

As for temperature in restriction, small change in temperature is expected and asphaltene precipitation is a weak function of temperature. As a result this equation is expected to be sufficient. The effect of temperature on calculated asphaltene fugacities will be investigated in an example model application presented Chapter 5.

4.3.3 Precipitation Prediction

At any conditions asphaltene precipitation and the amount precipitated can be determined by comparing the fugacity of asphaltene in the liquid phase (f_{nB}^L) with the calculated asphaltene fugacity (f_a). At specified pressure and temperature a vapour-liquid equilibrium calculation is conducted by first assuming no precipitation ($n^S = 0$). If the calculated fugacity of asphaltene in the liquid phase is less than that in the solid phase ($f_{nB}^L < f_a$) there is no precipitation as shown in the asphaltene model schematic (Figure 4-17).

By assigning $n^S = 0$ the conditions where f_{nB}^L is equal to f_a is the onset point for asphaltene precipitation (the points where two fugacity plots intersect as in Figure 4-18). In the case that the above analysis reports asphaltene precipitation ($f_{nB}^L > f_a$) the amount of the precipitate can be estimated by adjusting the solid amount (n^S) until the values of f_{nB}^L and f_a are matched [38]. However approximating the amount of precipitate is not in the scope of this research. Refer to Li et al. for more detail on estimating precipitation amount [38].

In primary production pressure is the main influence in asphaltene precipitation. Nghiem and Coombe suggested that volume shift parameter of asphaltene component was an important parameter for modeling pressure effect as to account

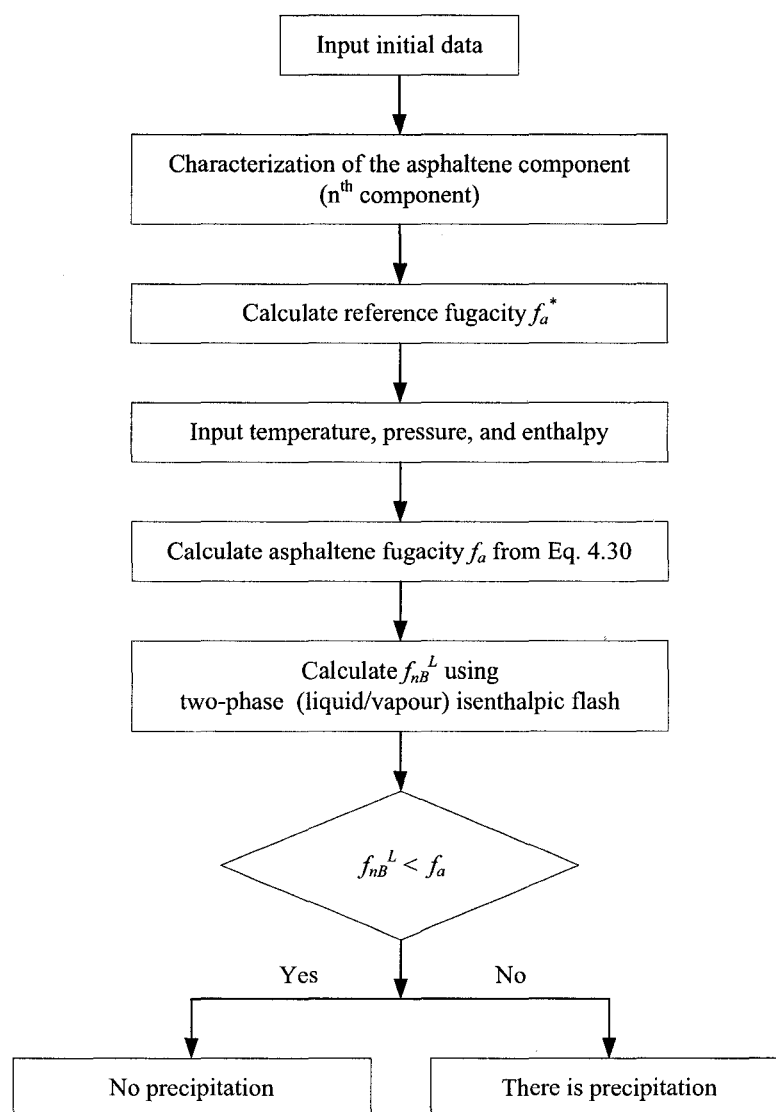


Figure 4–17: Asphaltene model schematic.

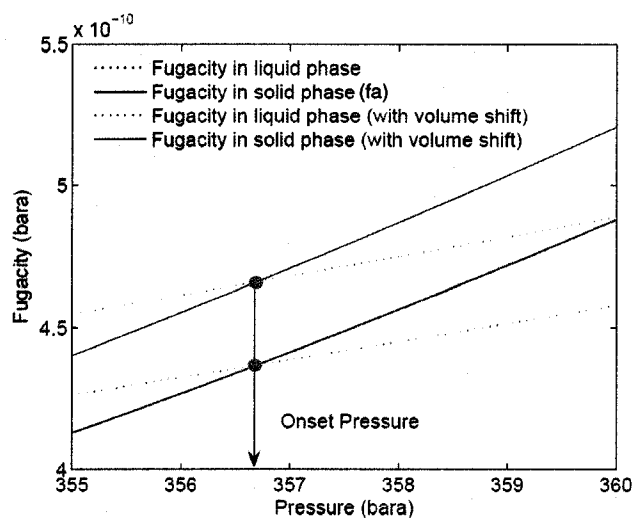


Figure 4-18: Asphaltene precipitation onset pressure predicted by the model.

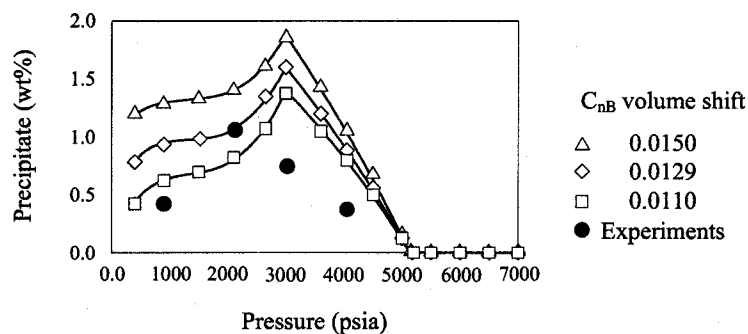


Figure 4-19: Effect of volume shift parameter on asphaltene precipitation [39].

for the inaccuracy in solid molar volume estimation [39]. From the study, the volume shift had effect on the prediction of precipitate amount. While larger volume shift showed more estimated precipitation the shapes of the precipitation curves predicted by the model were similar and the predicted onset pressure was not affected (Figure 4-18 and Figure 4-19). This conclusion is only true for the case where the onset is used as the reference conditions in calculating asphaltene fugacity (f_a).

To include volume shift into the model an equation of state with volume shift parameters is used. The fugacities of the fluid components at pressure p and temperature T are calculated as [39]:

$$\ln f_{ij} = \ln f_{ij}^{EOS} + \frac{c_i b_i p}{RT} \quad (4.31)$$

where f_{ij} and f_{ij}^{EOS} are the fugacities of component i in phase j calculated with and without volume shift, respectively. The parameters c_i and b_i are the volume shift and the EOS parameter b for component i , respectively.

In this research, as the main purpose of the equation of state does not involve volume estimation of asphaltene precipitates, volume shift parameters are not used in the model. Nevertheless, the effect of volume shift on asphaltene onset predictions at temperatures other than the reference temperature will be investigated where model applications are presented in Chapter 5.

The accuracy of the proposed asphaltene model depends on the regression of the model to experimental data. The interaction coefficients between hydrocarbons are used as the regression parameters. Li et al. suggested calculating the interaction parameters between hydrocarbons up to and including C_{nA} using the following expression [38]:

$$\delta_{ij} = 1 - \left[\frac{2v_{c_i}^{1/6} v_{c_j}^{1/6}}{v_{c_i}^{1/3} + v_{c_j}^{1/3}} \right]^e \quad (4.32)$$

where v_c is the molar critical volume. The exponent e was used to match the experimental saturation pressures while an equal value for the interaction parameters between C_{nB} and light hydrocarbon components was used to match the amount of asphaltene precipitates obtained from experiments.

CHAPTER 5

COUPLED MODEL AND APPLICATIONS

In the previous chapters the network model and the compositional asphaltene model were presented. Using the proposed methodology these models can be incorporated and used to simulate a production well and predict local asphaltene precipitation. In this chapter the methodology is first outlined. Example cases of production wells with different completion and production scenarios will then be provided to demonstrate how the proposed methodology can be used to predict asphaltene precipitation.

5.1 Coupled Model

Figure 5-1 shows the schematic of the proposed methodology. Provided the fluid's PVT data the fluid is first characterized and properties of the fluid are generated. There are two types of properties required in the proposed model: black-oil-model properties and characterized-oil properties. The black-oil properties are generated by considering the fluid as a two-component system consisting of stock-tank oil and gas. The characterized-oil properties are generated for all of the characterized thirteen components. The black-oil properties required in the non-compositional network model include B_o , B_g , R_s , density, and viscosity. Characterized-oil properties are required in the asphaltene model for compositional calculations. These include components' critical properties, acentric factors, and coefficients for enthalpy correlations. All of these properties can be generated prior to the simulations using a thermodynamic package.

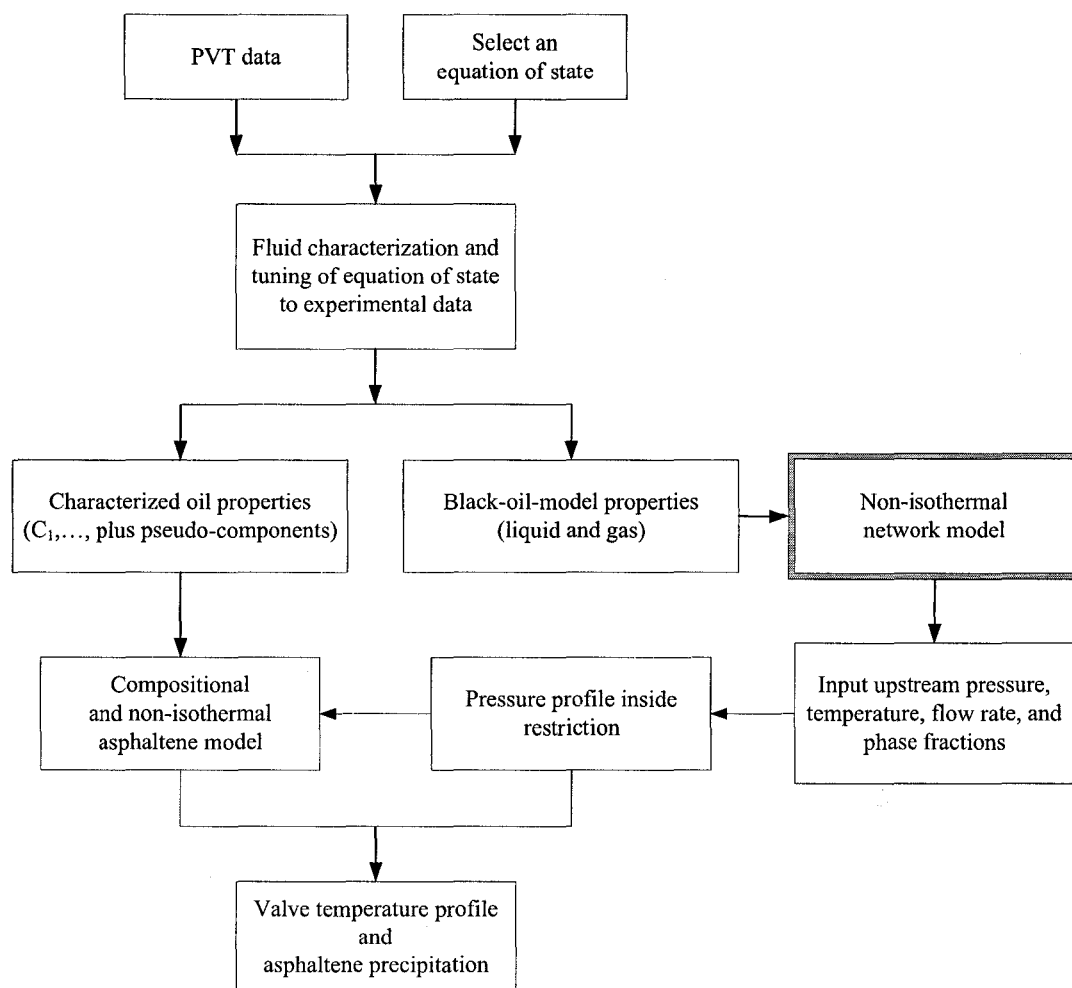


Figure 5-1: Proposed model schematic.

The well is then modeled using the non-isothermal network model to generate profiles of physical flow parameters. These profiles provide a rough picture of what happens in the well and allow the locations suspected for asphaltene precipitation to be determined. These locations are commonly where the fluid experiences a drastic change in conditions such as where the flow path is restricted. At the suspected locations the compositional asphaltene model can be used. Microscopically, as the local pressure is significantly influenced by the geometry of the restricted flow path, the local geometry is first modeled using a simplified geometry and detailed local pressure calculations are performed. Based on the calculated pressure the proposed asphaltene model can then be used to predict asphaltene precipitation. If the system is considered adiabatic (such as the case of flows through valves) an isenthalpic-flash phase equilibrium calculations are used where the local temperature and asphaltene precipitation can be determined.

5.2 Model Applications

To demonstrate the above methodology example simulations are provided. The simulations were performed on wells with different completion schemes. The results from the simulations are presented below.

5.2.1 Fluid Characterization

For all example cases the same fluid were used. In this case a fluid where asphaltene precipitation was observed in experiments was used and its available precipitation data are shown in Table A-1 in Appendix A. Nghiem and Coombe determined the asphaltene onset pressure by extrapolating using the last two data points and it was found to be approximately 356.69 bara at 100 °C [39]. As mentioned in order to perform simulations using the proposed model the fluid must be

first characterized. As required by the asphaltene model the example fluid was characterized into thirteen pseudo-components after Nghiem and Coombe [39]. The fluid was first characterized into 12 components using a thermodynamic package [12]. According to the characterization method outlined in Section 4.3.2 the 12th component (C_{13+}) of the fluid was divided into two components: non-precipitating component (C_{13A+}) and precipitating component (C_{13B+}). The C_{13B+} component was given the same properties of the C_{13+} characterized by the thermodynamic package. Only the binary interaction parameters between components were assigned differently. The mole fraction of the precipitating component in the fluid mixture was calculated from Equation 4.20 as below [39]:

$$\begin{aligned}
 z_{C_{nB}} &= \frac{W_{C_{nB}} MW_{Oil}}{MW_{C_{nB}}} \\
 &= \frac{0.158035 \times 171.343}{665.627} \\
 &= 0.040681
 \end{aligned} \tag{5.1}$$

where $W_{C_{nB}}$ is the average weight fraction of asphaltene precipitates obtained from the last column of Table A-1. Note that the values used in the calculation above were directly adopted from Nghiem and Coombe [39]. The characterized thirteen components of the fluid are listed in Table A-2. Some other basic properties required in calculations of other fluid properties can also be found in the table.

As the experimental data were available for 100 °C a reservoir temperature of 100 °C was used for most of the example cases. All black-oil properties were generated at the reservoir temperature of 100 °C except for Case 2 where reservoir temperature varied from 80 to 100 °C. In that case the fluid properties were calculated at the averaged reservoir temperature of 90 °C. Correlations were however used to calculate oil viscosity as described in Chapter 2 in order to take into account temperature effects. The dead-oil viscosity was calculated at the fluid temperature during the simulations. Oil viscosity was then calculated based on the calculated

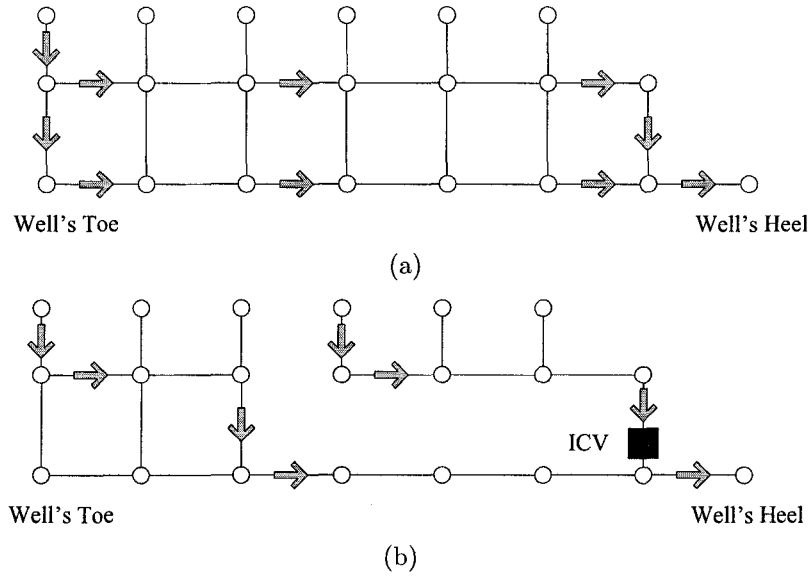


Figure 5-2: Example well network: Case 1. (a) conventional completion (b) advanced completion with an ICV

dead-oil viscosity.

5.3 Case 1: Well with an Inflow Control Valve

In this example case, the proposed model was used to simulate a 2000-meter horizontal well as illustrated in Figure 5-2. Two completions schemes were used. In the first scheme (Figure 5-2(a)) the well was conventionally completed. In the second scheme (Figure 5-2(b)) the well was equipped with an inflow control valve (ICV) to provide an extra pressure drop required as to control the inflow rates. The drastic pressure drop introduced by the valve however was suspected for causing asphaltene to precipitate.

To analyze the physical flow behaviour in this well the proposed network model was used. The well of 2000-meter length was first divided into 200 segments. Each had an equal length of 10 meters. The reservoir conditions were assumed to be 365 bara and 100 °C with the oil saturation (S^o) of one. The fluid was undersaturated which was a typical case for asphaltene precipitation in primary depletion. These

conditions were used for the entire length of the well (e.g. $p_{res(1)} = p_{res(2)} = \dots = p_{res(N-1)} = 365$ bara). In addition the reservoir was assumed to be homogeneous and equal reservoir properties (e.g. permeability (K)) were applied for all the segments.

In the advanced-completion case, at the end of the well a 500-meter interval was packed off and the well was completed with an inflow control valve (ICV) at the end the interval (N^{th} segment), right before the fluid reached the heel of well. All the inflows in this interval were collected and entered the tubing through the ICV adjusted to introduce a total pressure drop of approximately 0.989 bara as to achieve the targeted production rate of approximately 8,000 m³/d at the heel of well provided the bottomhole pressure of 357.3 bara. The bottomhole pressure in the conventional-completion case was set at 359 bara to provide approximately the same amount of production rate. Table 5–1 summarizes basic characteristics of the well systems and reservoir properties required in the simulations.

5.3.1 Network Model

In this part the momentum-transport and energy-transport problems were analyzed here in two separated but interconnected steps as described in Section 2.6. Firstly the momentum-transport problem was solved assuming an isothermal process. In the isothermal calculations, well pressure profile, flow rate, and phase fraction were calculated at the reservoir temperature. The flow directions were assumed to be known where the flow everywhere in the network was toward the heel of well as shown in Figure 5–2. As also shown in the figure, in the case with ICV, some of the bridges were removed (by giving bridge indices of zero's) in the advanced completed well to simulate the disconnection in fluid flow due to and inside the packed-off interval. An annular bridge was removed to represent the part of the well where the packed-off material resided. Annular-to-tubing bridges inside the

Table 5-1: Well basic characteristics for wells in Case 1.

Property	Value
Well length (m)	2000
Reservoir pressure (bara)	365
Pressure at heel without ICV (bara)	359
Pressure at heel with ICV (bara)	357.3
Reservoir temperature ($^{\circ}\text{C}$)	100
Production rate (m^3/d)	8000
Permeability (Darcy)	1
Near-wellbore skin factor	2
Drainage radius (m)	20
Oil saturation	1
Tubing diameter, r_i (m)	0.127
Well outside diameter, r_o (m)	0.167
Discharge coefficient for flow through slots ($\text{Pa} \cdot (\text{kg}/\text{m}^3)^{-1} \cdot (\text{m}/\text{s})^{-2}$)	10
Slot ratio (%)	60
Perforated ratio (%)	2
Casing thickness (m)	0.03
Cement thickness (m)	0.05
Valve opening (m)	0.02
Fluid heat transfer coefficient, k_{fl} ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	0.306
Steel tubing thermal conductivity, k_w ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	50
Outer casing thermal conductivity, k_c ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	11.99
Cement thermal conductivity, k_{cem} ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	6.95
Joule-Thompson coefficient ($^{\circ}\text{C}/\text{bara}$)	-0.0487
Fluid heat capacity ($\text{J}/\text{mol } ^{\circ}\text{C}$)	358.25

packed-off interval were also removed as all of the flow inside the interval were to be gathered and entered the well tubing through the ICV at the end of the interval.

Fluid properties of the flow in each bridge were evaluated based on the pressure at the entering node of each particular bridge. For example, for Bridge 2-1 where the fluid flowed from Node 2 to Node 1, the density of the fluid in this bridge was evaluated at the pressure at Node 2.

The simulations of the well resulted in Figure 5-3 for the pressure profile and Figure 5-4 for the flow rate profile in the well with ICV. Figure 5-5 and Figure 5-6 show the simulation results in the well without ICV or packed-off interval. As shown in the figures the pressure in the well decrease faster when the fluid approach the heel of well due to the increase in flow rate. With the ICV the sandface pressure drop (pressure in annulus) in the area close to the heel of well was reduced. This is required in some cases where a control of reservoir inflow is needed (e.g to prevent water or gas breakthrough). Consider the flow rate profile plots. The plots show the material balance of the fluid in the well. As seen in the figure the inflow rates were the sum of the flows in tubing and the annulus. The increase in tubing flow rate as the fluid flowed toward the heel of well equals to the annulus-to-tubing flow rate entering the tubing at each segment.

Using the calculated flow rate the well temperature profile was obtained based on the energy balance as described in Section 2.5. The reservoir temperature (100 °C) was used as the boundary conditions at all the inlet nodes. The other required boundary condition was the temperature at the toe (the annular node of Segment 1) where it was assumed equal to the reservoir temperature of 100 °C.

In the temperature simulations there were a couple of parameters required to represent different heat transfer mechanisms. These parameters included heat capacity (C_p) and Joule-Thompson coefficient (K_{JT}). The values of these parameters used in the model were evaluated at the averaged well pressure of 361.15 bara

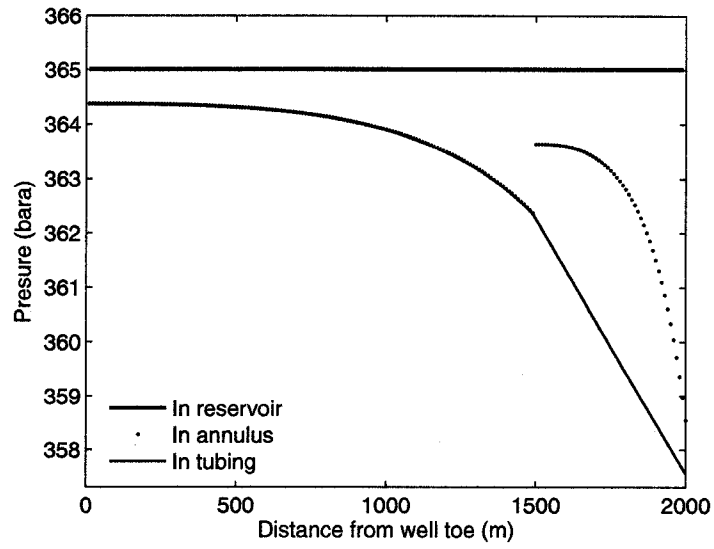


Figure 5-3: Pressure profile for the well with ICV.

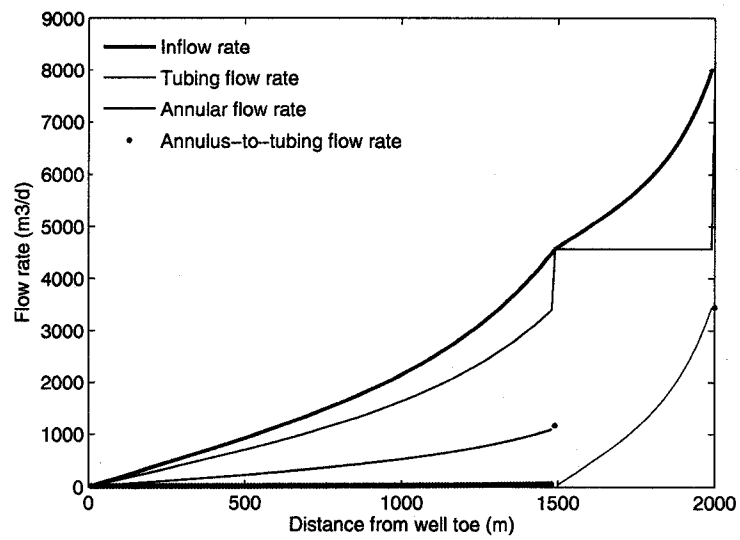


Figure 5-4: Flow rate profile for the well with ICV.

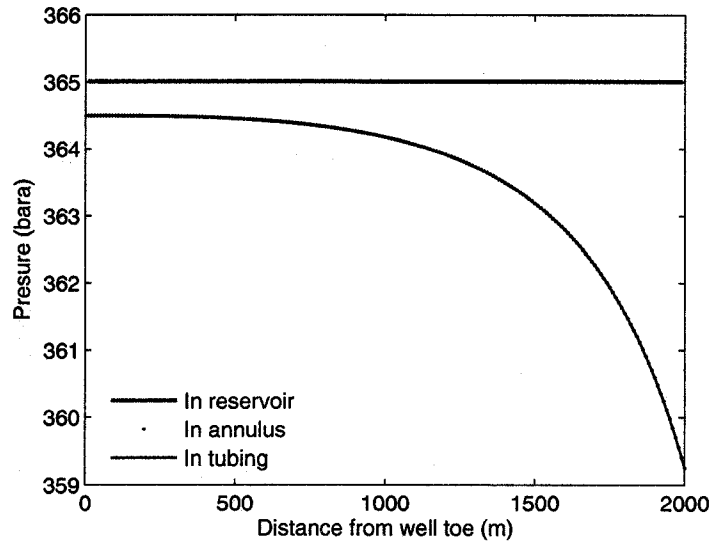


Figure 5-5: Pressure profile for the conventionally-completed well.

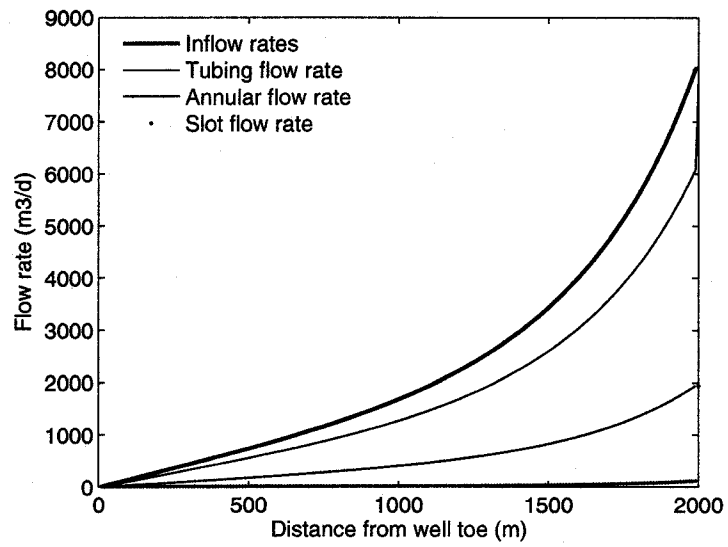


Figure 5-6: Flow rate profile for the conventionally-completed well.

$((p_{res}+p_{bh})/2)$ and reservoir temperature (100 °C). At the specified conditions the parameter K_{JT} and C_p were found to be -0.0487 °C/bar and 358.25 J/mol °C, respectively. The overall heat transfer coefficients between the fluid and its surroundings (U 's) were calculated as described in Section 2.5.2. To calculate the overall heat transfer coefficient for the tubing-annulus interface (U_t) for example, there were two heat transfer components: heat conduction through the tubing wall and heat transfer between the wall and the tubing fluid.

for the heat flow through tubing wall:

$$Q_w = -2\pi R (1 - \gamma) k_w \frac{T_o - T_i}{\Delta R_w} \quad (5.2a)$$

for the heat flow by the tubing fluid:

$$Q_{fl-t} = -2\pi (1 - \gamma) h_t (T_i - T_{fl-t}) \quad (5.2b)$$

where the subscripts w , fl , and t indicate pipe wall, fluid, and tubing, respectively. In this case the tubing wall had a thickness of ΔR_w which was assumed to be 0.01 meter and had a steel thermal conductivity (k_w) of 50 W·m⁻¹·K⁻¹. The temperatures at the inside and the outside surfaces of the tubing were T_i and T_o , respectively and the temperature of the fluid in the tubing was T_{fl-t} . The heat transfer coefficient between the tubing fluid and the tubing wall was represented by h_t . In addition it was assumed that the slot openings took the area of 60% of the total surface area of the tubing wall, thus the permeable fraction (γ) of 0.6 was used for flows through slots. For the flow inside the packed-off section where there was no flow through the tubing wall, γ was zero.

Consider Figure 5-7. By assuming the annular fluid temperature at the outer surface of the tubing wall the overall heat transfer coefficient for tubing fluid was derived as:

$$U_t = \frac{Q}{(T_{fl-t} - T_{fl-a}) 2\pi r_i (1 - \gamma)} = \left[\frac{\Delta R_w}{k_w} + \frac{1}{h_t} \right] \quad (5.3)$$

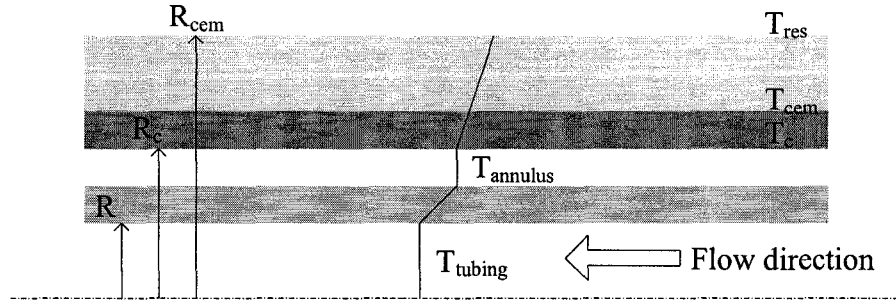


Figure 5-7: Temperature profile at wellbore-reservoir interface.

where T_{fla} is the fluid temperature in the annulus and r_i is the inner radius of the tubing wall. The fluid heat transfer coefficients were calculated using the Nusselt number as presented in Section 2.5.2:

$$h_t = \frac{Nu_t \cdot k_{fl}}{2R} \quad (5.4)$$

In this case the heat conductivity of the fluid (k_{fl}) was $0.306 \text{ W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$ at 361 bara and 100°C .

The overall heat transfer coefficient for the annulus-reservoir interface (U_a) was derived similarly by considering the heat conduction through the outer well casing, through cement, and the heat transfer at the fluid-casing interface.

$$U_a = \frac{Q}{(T_{fla} - T_{res}) 2\pi r_o (1 - \gamma)} = \left[\frac{\Delta R_c}{k_c} + \frac{\Delta R_{cem}}{k_{cem}} + \frac{1}{h_{ann}} \right] \quad (5.5)$$

the heat transfer coefficients for fluid in the annulus were calculated from:

$$h_a = \frac{Nu_a \cdot k_{fl}}{r_o - r_i} \quad (5.6)$$

where r_o and r_i are the outer and inner radii of the annulus, respectively. The thermal conductivities of 11.99 and $6.95 \text{ W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$ were used for the 0.03-meter casing and the 0.05-meter cement, respectively.

The resulting temperature profiles are shown in Figure 5-8 for the conventionally completed well and Figure 5-9 for the well with the ICV. The figures show that

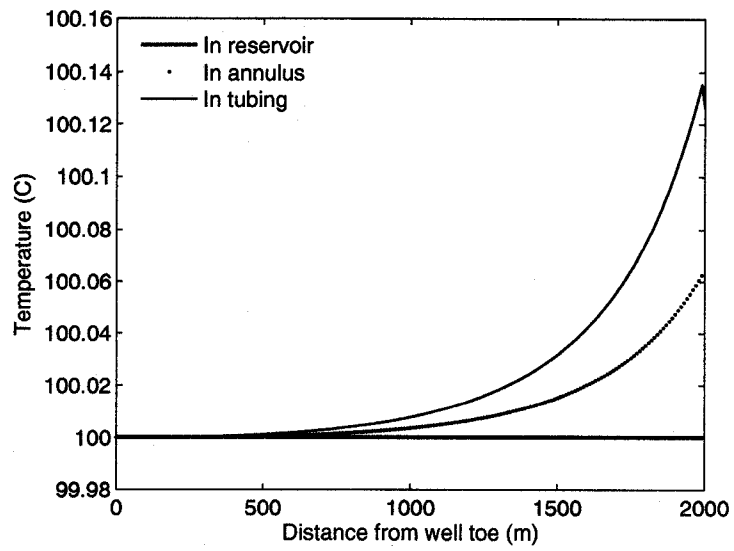


Figure 5-8: Temperature profile for the conventionally-completed well.

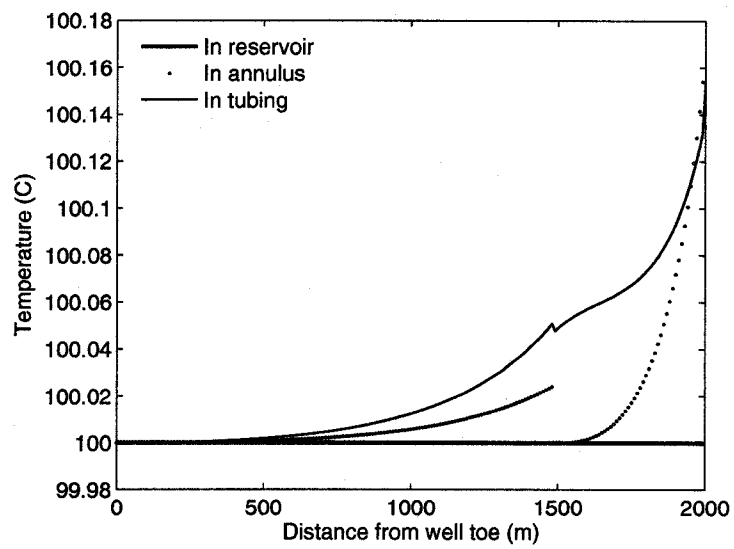


Figure 5-9: Temperature profile for the well with ICV.

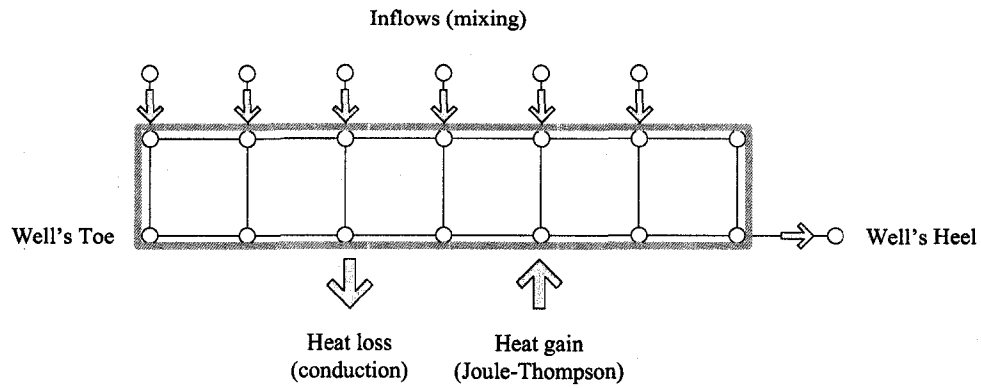


Figure 5-10: Energy transport in a well network.

the differences between the reservoir and the bottomhole temperatures are approximately 0.126 °C and 0.155 °C in the conventional and advanced-completion wells, respectively. The differences are mainly due to the differences in fluid pressure drop as Joule-Thompson effect was dominant.

As in Figure 5-10 the entire well network can be viewed as one control volume. The energy transport into and out of this control volume included the energy associated with the radial inflows from the reservoir, energy loss (or gain) between the fluid and the surroundings, and energy loss (or gain) from Joule-Thompson effect. The net energy from the aforementioned energy transport mechanisms led to the change in convective energy held by the fluid (defined here as $\dot{m}\hat{C}_pT$ for fluid in both tubing and annulus) from the initial value at the toe of well as the fluid flowed toward the heel of well.

To illustrate the balance of the above heat transport components, Figure 5-11 compares the values of each term in the energy balance equation. The values were relative values based on the energy of fluid at the toe of well. Also note that these values were the total values of both fluid in the annulus and that in the tubing. The figure shows that the Joule-Thompson effect contributed the most to the change in fluid temperature in this case. The Joule-Thompson term had a positive value

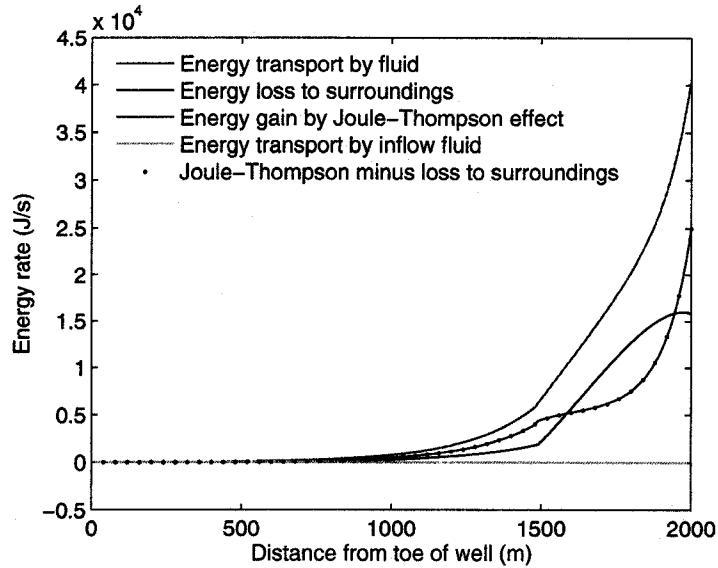


Figure 5-11: Energy flow profile.

meaning that the energy was added to the fluid in the well. As the inflow fluid had the same temperature as that at the toe of well (100 °C), it had no effect on the relative change in fluid energy. The energy loss to surroundings (represented by the overall heat transfer in the equation) played the second most important role in the change in fluid energy. This value was a negative value suggesting that the energy was transported away from the well. The fluid energy at any point along the length of the well (the pink line in the figure) was therefore the net result of the Joule-Thompson heating effect and the energy loss to surroundings described above (dotted line in the figure). This therefore verified the energy balance calculated by the proposed network model.

Figure 5-12 showing the effects of the conductive heat transfer (through surroundings) and Joule-Thompson on the temperature profile also confirms that the Joule-Thompson effect was dominant. Without the Joule-Thompson effect there was not at all a change in fluid temperature. As also seen in the figure the Joule-Thompson effect had a heating effect while the conduction term represented by the

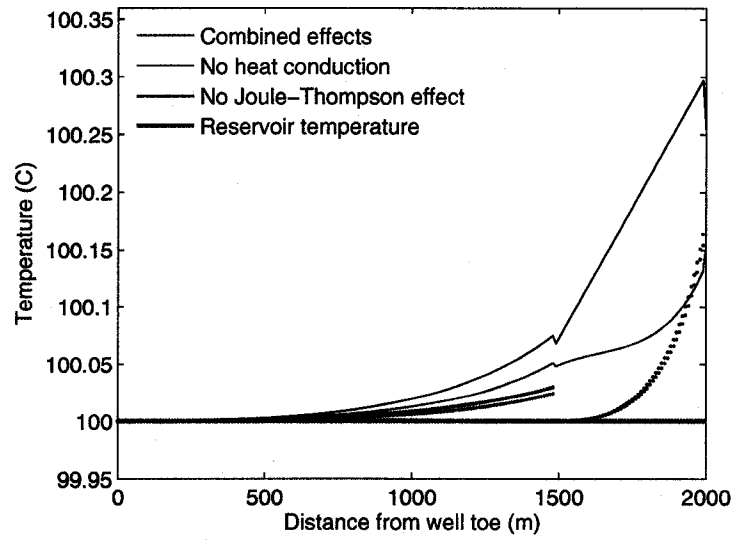


Figure 5-12: Effects of heat transport mechanisms on well temperature profile

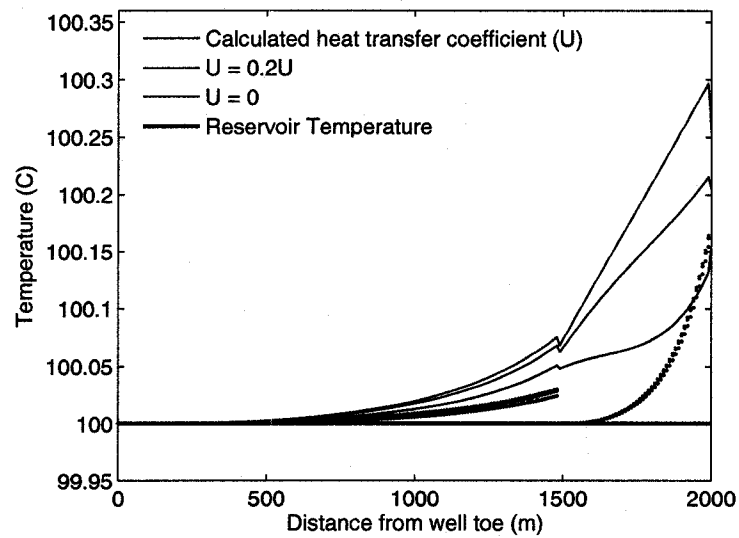


Figure 5-13: Effects of overall heat transfer coefficients on well temperature profile

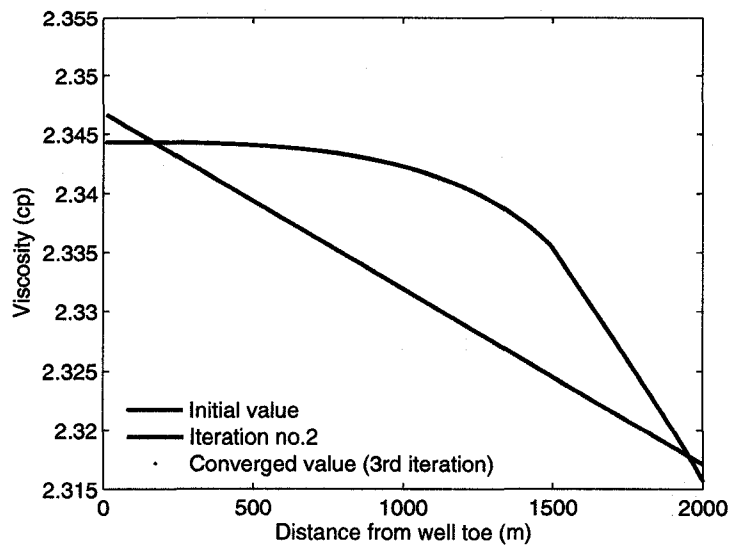


Figure 5-14: Convergence of viscosity values in the proposed model.

overall heat transfer (U) had a cooling effect on the fluid temperature. Figure 5-13 shows the effects of overall heat transfer coefficient on the well temperature profile. In this particular case, the larger value of U suggested higher heat loss to the surroundings and in turn resulted in smaller in fluid temperature. The curve with $U = 0$ represents the temperature profile when the well was completely insulated.

Because oil viscosity is a function of temperature and the value may change considerably in the case where thermal treatment is applied, the proposed model was designed in such a way that the oil viscosity is allowed to be recalculated in response to the change in temperature. Using correlations and the pre-generated dead-oil viscosities the updated viscosities were used to recalculate pressures and temperatures using the procedures presented in Section 2.6. These steps were repeated until the fluid viscosities converged. Figure 5-14 shows the convergence of the oil viscosities in the well's tubing as the well's pressure-temperature condition was updated. The initial values presented in the plot were those evaluated at the initial guessed pressures where the pressure was assumed to be linearly decreasing

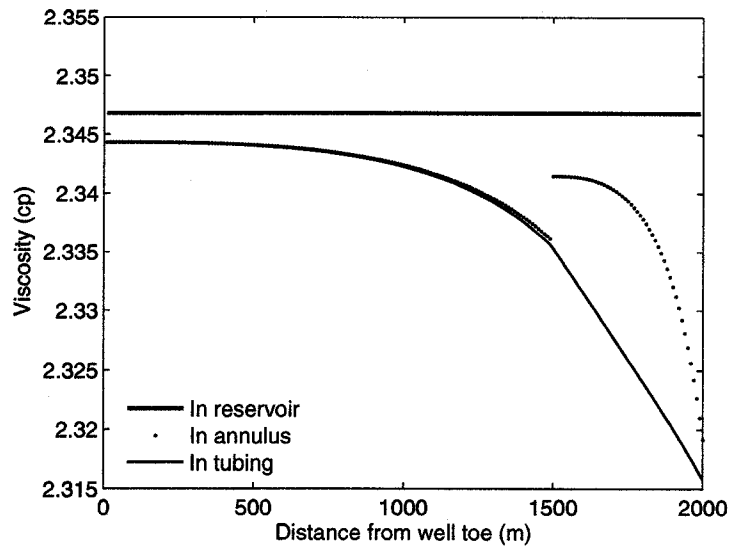


Figure 5-15: Viscosity profile for the well with ICV.

from the reservoir to the heel of well. As only a small change in temperature was observed the viscosity converged after the second update of the viscosity.

Figure 5-15 shows the converged values of the oil viscosity. It is shown in the figure that the viscosity profile resembles the shape of the pressure profile. This is because the pressure variation had an effect on both the fluid temperature and the oil viscosity. Simulations were conducted for the cases where more degree of temperature variations were observed, i.e. larger difference between the reservoir and the bottomhole temperatures (dT in Figure 5-16). As dT became larger the temperature started to have more effect on the viscosity profile.

As the pressure in the well in both cases of completions never went below the fluid's saturation pressure the liquid volume fraction was always 1. To demonstrate the ability of the proposed model to predict liquid holdups, an example of two-phase system was simulated. Consider a conventionally completed well producing fluid from a saturated reservoir with the reservoir conditions of 180 bara and 100°C. The pressure of the fluid was originally below the saturation pressure. Assuming that the oil saturation in the reservoir was 0.9 and the bottomhole pressure was set

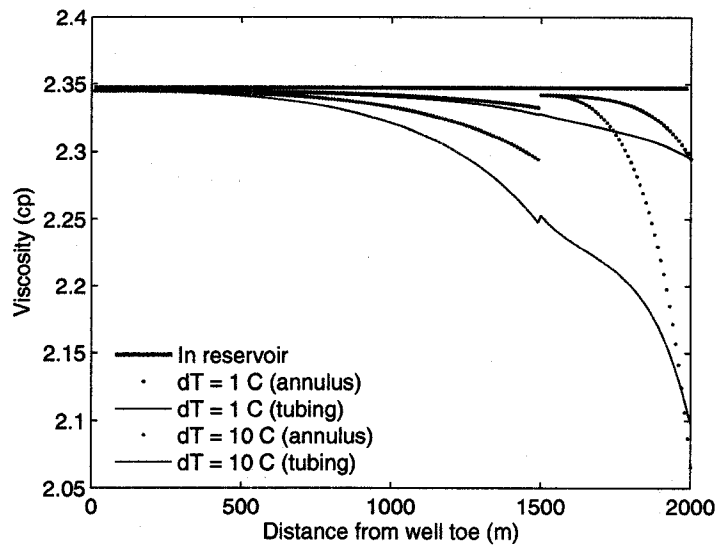


Figure 5-16: Effects of temperature on viscosity profile.

to 175 bara, the production rate was found to be 10,180 m³/d. Figure 5-17(a) shows the variation of liquid holdup in the well predicted by the model. The liquid holdup in the tubing was found to range from 0.5308 at the toe to 0.5171 at the heel of well reduced from 0.5318 in the reservoir as the pressure was decreased. The pressure and flow rate profiles are shown in Figure 5-17(b) and Figure 5-17(c), respectively.

For the temperature profile (Figure 5-17(d)) it can be seen that, in contrary to the case of undersaturated fluid, the the results showed a decrease in temperature as the fluid flowed toward the heel of well. The temperature at the heel was 0.12 °C lower than that at the toe. This can be explained by the fact that this case involved gas phase. The Joule-Thompson coefficient of a gas phase has a positive value meaning that pressure drop has a cooling effect on the gas temperature. In the simulation a two-phase Joule-Thompson coefficient was calculated as a volume average of the coefficients for the liquid and the gas phases. At the averaged pressure and 100 °C the Joule-Thompson coefficients were -0.0471 and 0.1691 bara/°C for the liquid and the gas phases, respectively. When averaged based on the liquid holdup in the reservoir (0.5318) the values resulted in a positive two-phase Joule-Thompson

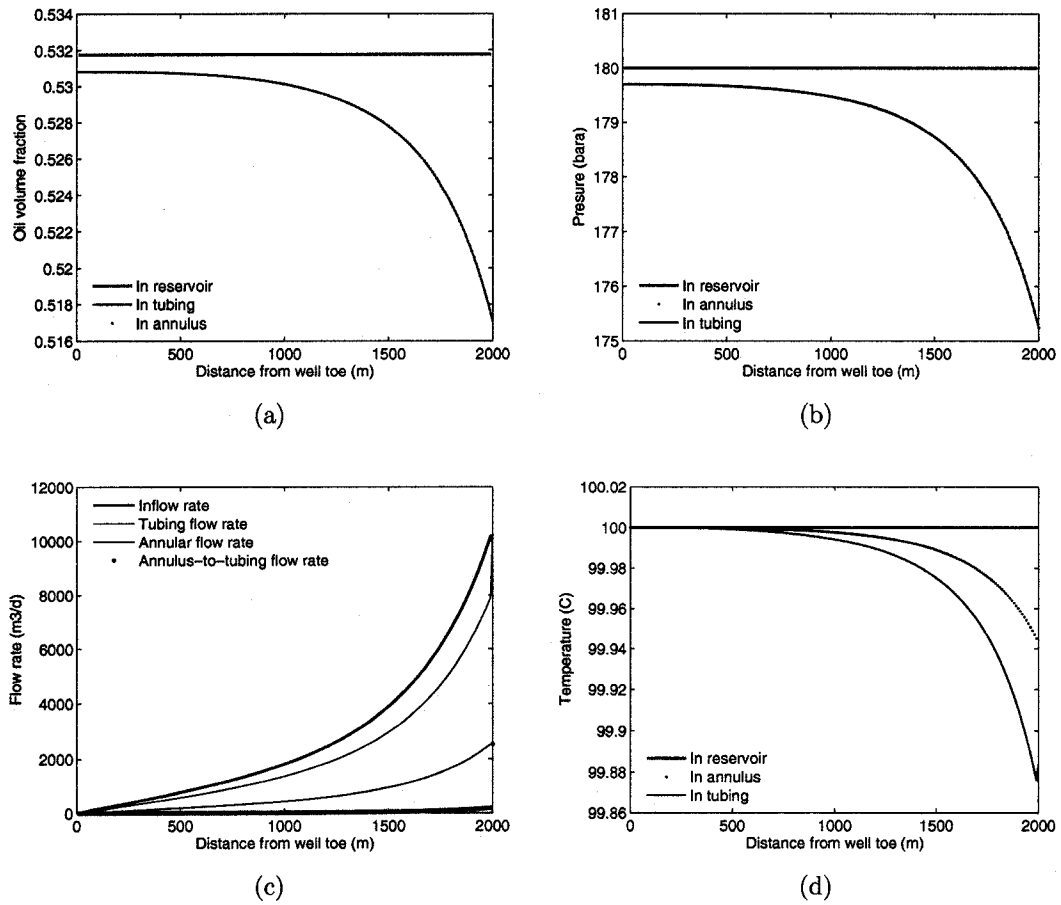


Figure 5-17: Network model results in the case of saturated fluid. (a) Liquid holdup profile. (b) Pressure profile. (c) Flow rate profile. (d) Temperature profile.

coefficient of 0.0509 bara/°C causing the fluid temperature to drop as the pressure decreased toward the heel of well. Determination of a two-phase Joule-Thompson coefficient that accurately represents the behaviour of the fluid however needs to be further investigated.

5.3.2 Asphaltene Precipitation Prediction

Once the final flow conditions in the well were obtained the analysis for asphaltene precipitation was conducted. Considering the well pressure profile it was obvious that the pressure in the well was above the asphaltene onset of 356.69 bara at 100 °C. However there was a large pressure drop through the ICV. The high flow rate (inflows were accumulated and forced through the valve opening) led to the most drastic pressure drop as seen in Figure 5-3. In addition as a result of the reduced cross-sectional area it was possible that the fluid experienced a pressure below the asphaltene onset inside the valve throat. Therefore a detailed analysis was performed on the annulus-to-tubing bridge at this location (Segment 200 or 2000 meters from the well's toe).

As described in Chapter 4 the asphaltene precipitation analysis was divided into two main parts. The first was to determine pressures at a finite number of points inside the restriction. The local pressures were then used to calculate local temperature and phase behaviour at the corresponding points using an isenthalpic flash algorithm.

To calculate the valve pressure profile the complex flow geometry through the valve was first modeled using a simpler geometry as shown in Figure 5-18. As the fluid pressure was well above the saturation pressure the pressure inside the restriction was analytically calculated using Equation 4.9b for single-phase liquids. For simplicity purposes the friction factor was used as the tuning parameter that took into account all the non-ideal effects occurring in the complex flow situation

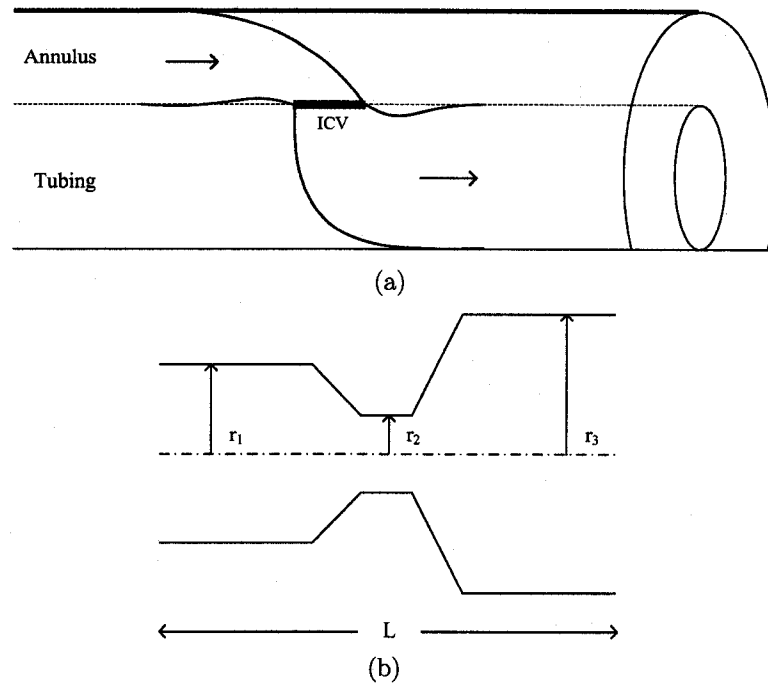


Figure 5-18: Modeling of flow through the ICV using a simple geometry. (a) flow of wellbore fluid through a restriction. (b) simplified geometry.

(e.g. change in the flow direction, turbulence, etc.). In this part of the analysis, as small changes in temperature and pressure were expected, fluid properties (such as density) was assumed constant and evaluated at the upstream conditions.

As seen in Figure 5-18(a) the flow geometry such as upstream and downstream diameters were not certain. In order to determine an appropriate geometry that represented the actual flow through the ICV a few analyses on the effects of the model geometry were first conducted.

(i) *Valve entrance*

Firstly the analysis on the effects of the valve entrance geometry (upstream diameter, D_1) was investigated. The investigation was carried out by varying the upstream diameter starting from 0.022 m (the width of the annulus measured from the tubing to the outer casing is 0.02 m but the value 0.022 m was used to avoid the same diameter as the valve opening (D_2) of 0.02 m). The

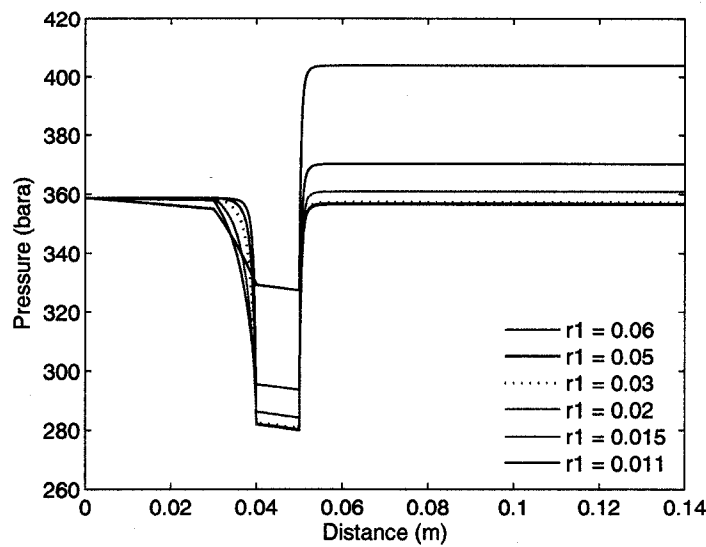


Figure 5-19: Pressure drop through ICV for various upstream radii.

value was increased up to 0.12 m (the area-based “equivalent” diameter of the annular cross section was 0.108 m).

Figure 5-19 shows the resulting valve pressure profiles for the friction factor of 0.05 and other geometry parameters were kept constant. When the entrance radius (r_1) was increased from 0.011 the total pressure drop across the valve (Δp) increased. The pressure profile started to converge when r_1 reached the value of 0.03 m. Increasing r_1 further had very little effect on the pressure profile. Therefore, it was concluded from this investigation that even though it was difficult to determine the actual upstream diameter, using any value larger than 0.03 m would not significantly affect the results of the valve pressure prediction. In this example the value 0.06 m was acceptable and was therefore used as the upstream diameter

(ii) *Valve exit*

Similar to the case of the valve entrance, the flow geometry on the downstream side of the valve was also uncertain and asymmetric. However for simplicity

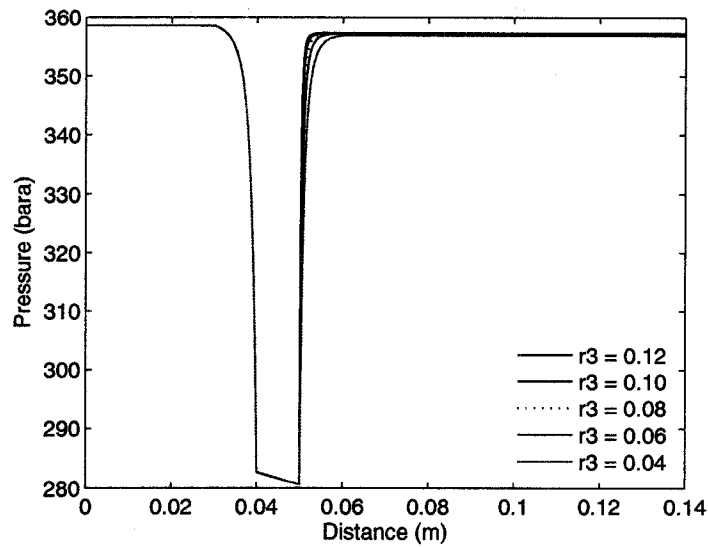


Figure 5-20: Pressure drop through ICV for various downstream radii.

purposes a circular geometry was assumed. In order to determine the appropriate downstream diameter to represent the actual flow geometry the effects of the downstream geometry was therefore investigated. The pressure profiles for different values of downstream diameters (D_3) ranging from 0.08 m to 0.24 m were generated. The tubing diameter was 0.127 m and was therefore included within the range. The same values of friction factor and the other geometry parameters were used for all the cases. The results from this investigation are shown in Figure 5-20. The exit diameter had very slight effects on the valve pressure drop (Δp). In addition similar to the case of valve entrance the valve pressure profiles became indistinguishable as the downstream diameter was increased beyond a value which was 0.16 m in this case. Therefore the downstream diameter of 0.08 was used.

(iii) *Valve length*

Another factor that might affect the valve pressure drop prediction is the length of the valve (L). When a fluid flows through a restriction there is a

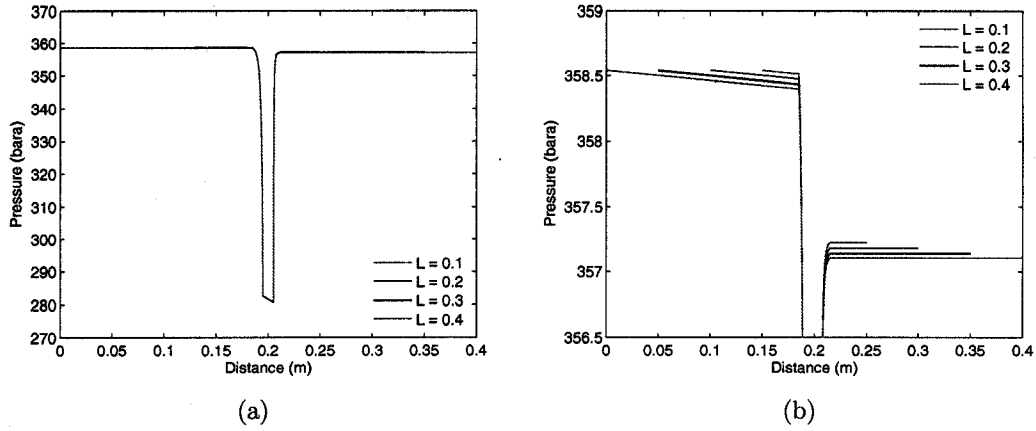


Figure 5-21: Pressure drop through ICV for various restriction lengths.

certain affected length starting from where the flow pattern is altered until it recovers and becomes fully developed at a point downstream of the restriction. The effect on pressure drop of the valve length was investigated and the results are shown in Figure 5-21. In this analysis the length (L) was varied from 0.1 to 0.4 m (the thickness of the valve opening remained at 0.01 m). The friction factor of 0.05 and the diameters D_1 and D_3 of 0.06 m and 0.16 m respectively were used. As seen in the figure, L did not affect the shape of the pressure profile. The differences in the predicted downstream pressure merely resulted from friction. In addition the effect was very small compared with the overall change in pressure across the valve.

In this case it was desired to evaluate the downstream pressure at the immediate point from the valve throat before the fluid leaving the valve combined with the fluid from the main tubing stream. Therefore, the shortest possible value of L was preferred. Half a diameter before and after the valve (i.e. $0.5D_1$ and $0.5D_3$, respectively) were assumed. This led to the total valve length (L) of 0.14 m.

(iv) *Friction factor*

As mentioned earlier the friction factor is used in this model as the tuning

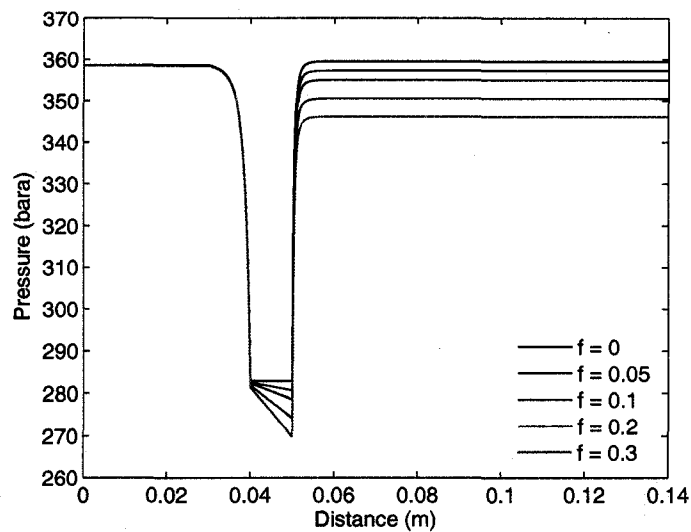


Figure 5-22: Pressure drop through ICV for various friction factors.

parameter that also takes into account additional pressure drop (other than acceleration and friction) caused by the ICV. Figure 5-22 shows pressure drop across the ICV for the friction factor ranging from 0 to 0.3. The curve with zero friction factor represents the pressure drop due to acceleration only. It is shown in the figure that the friction factor had insignificant effect on the shape of the pressure profile both downstream and upstream of the valve. Its contribution to the pressure variation only occurred inside the valve throat, yet in turn affected the downstream pressure.

In this example case the ICV was to introduce 0.989 bara pressure drop to the system keeping the designed production rate. The flow rate through the valve obtained from the network model was 3,442 m³/d. By trial and error the friction factor of 0.0427 was found to provide the pressure drop of approximately 0.990 bara and therefore used in the model to predict the valve pressure drop. Note that such a relation between the flow rate and the pressure drop required the valve coefficient (C_v) of 151.98 USgpm/(psi)^½ which was considered extremely low resistant for a commercial valve.

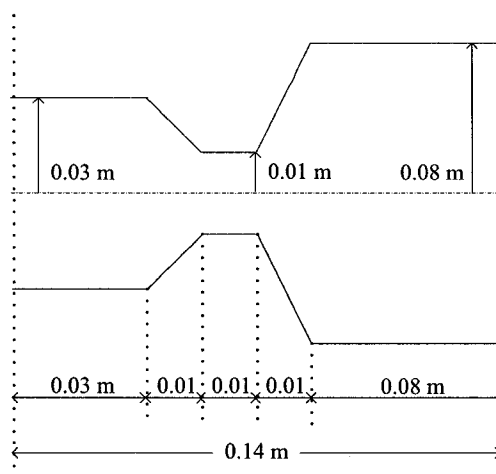


Figure 5-23: Simplified restriction geometry used to model the ICV in Case 1.

Figure 5-23 shows the simplified valve geometry used to represent the flow through the ICV in this example case. Using D_1 , D_2 , and D_3 of 0.06, 0.02, and 0.16 respectively and the friction factor of 0.0427 the pressure drop model was used to predict the pressure profile across the ICV. In order to do so the local conditions inside the restriction were analyzed discretely. The total length of the valve was divided into 201 points including the upstream and downstream points. The resulting pressures were plotted as shown in Figure 5-24. Because of the large flow rate and the small valve opening, a large pressure drop was observed. These calculated pressures were to be used in calculating local temperatures and to determine solid phase formation using the isenthalpic-flash asphaltene model as described below.

Based on the calculated pressures, the compositional asphaltene model was then used to determine asphaltene precipitation behaviour of the fluid flowing through the ICV. In the compositional calculations the fluid was considered a multi-component system as opposed to liquid-gas components considered in the network model. The thirteen pseudo-components characterized in Section 5.2.1 were used to perform pseudo-three-phase equilibrium calculations on each of the local points inside the restriction based on the corresponding pressure previously calculated.

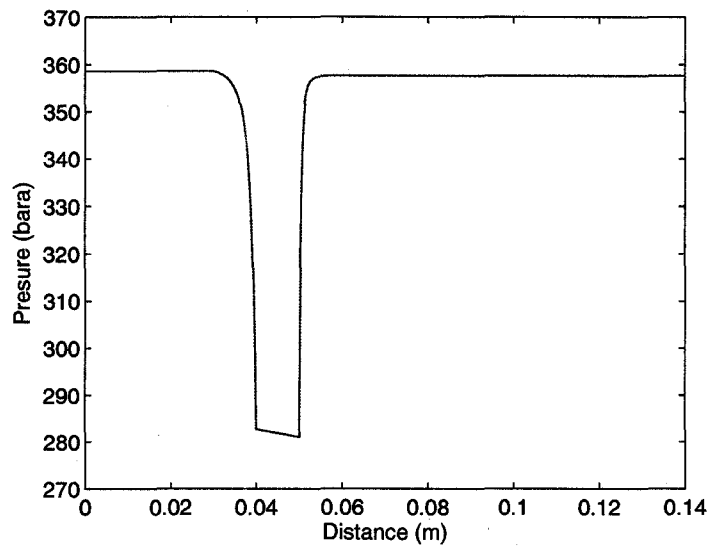


Figure 5-24: Pressure profile across ICV.

In order to predict asphaltene precipitation the reference condition was first selected and the reference asphaltene fugacity (f_a^*) was calculated. At the onset pressure precipitation amount was zero. Thus, using the onset as the reference conditions, f_a^* was calculated as the fugacity of the asphaltene component in the liquid phase (f_{nB}^L) of the entire fluid system in Table A-2. An isothermal flash was conducted on the fluid at the reference conditions (356.69 bara and 100 °C) and the reference asphaltene fugacity (f_a^*) was found to be 5.5E-10 bara. The asphaltene fugacities (f_a) at different pressures were then calculated based on this value using Equation 4.30 by using the onset experiment temperature (100 °C) as the reference temperature. In phase equilibrium calculations, the Peng-Robinson equation of state was used. The equation was tuned to the saturation pressure prior to the simulations. The exponent for the binary interaction parameter of 1.5807 was found to match the experimental saturation pressure. The binary interaction parameters between C_{13B+} and the light components of 0.22 was used [39]. In addition, the enthalpy at the upstream location (H_{sys}) was also calculated and used

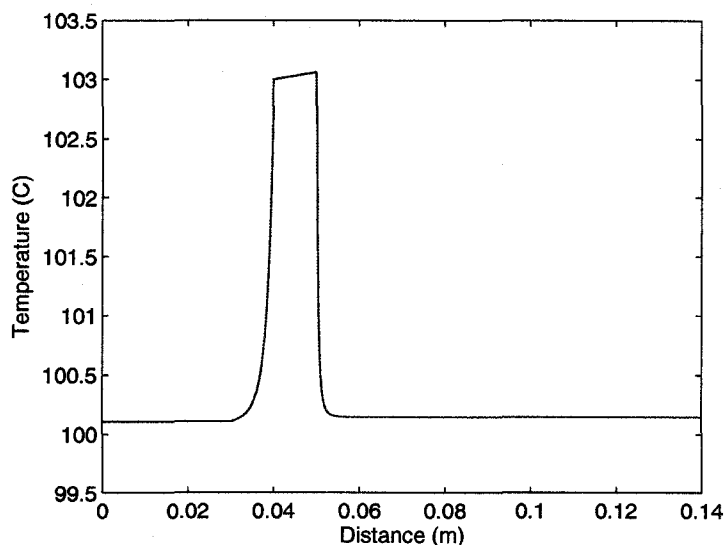


Figure 5-25: Temperature profile across ICV.

as the initial system enthalpy required in the isenthalpic flash calculations. The upstream fluid conditions were obtained from the network model. In this case the conditions of the fluid at the annular node of Segment 200's were used: 358.54 bara and 100.154 °C. The temperatures of the consecutive points inside the restriction were determined where the enthalpy-conservation constraint was satisfied. Once the local temperature was determined the fugacities of fluid components were calculated at the local pressure-temperature conditions.

The process was repeated for all the 201 points along the flow directions providing the temperature profile as shown in Figure 5-25 and the fugacity profile of the precipitating component in the liquid phase (f_{nB}^L) was also calculated. The fugacities were compared with asphaltene fugacities (f_a) calculated at the corresponding pressure and the reference temperature (100°C). As shown in Figure 5-26, in spite of the change in fluid temperature from the reference temperature, there was no significant difference in the f_a calculation when the reference temperature or the local temperature ($T(z)$) were used in Equation 4.30. Consider the fugacity plots

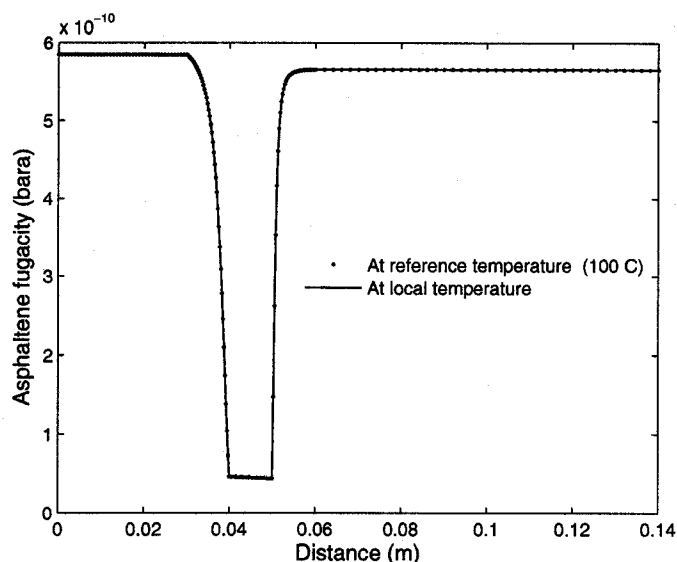
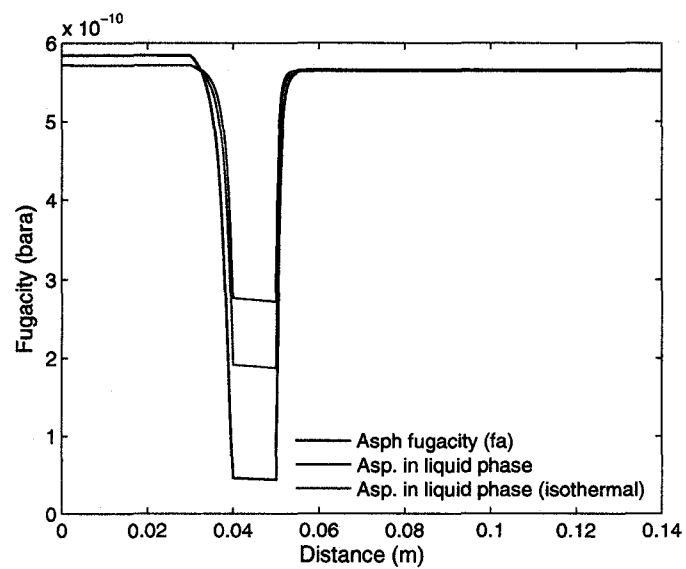
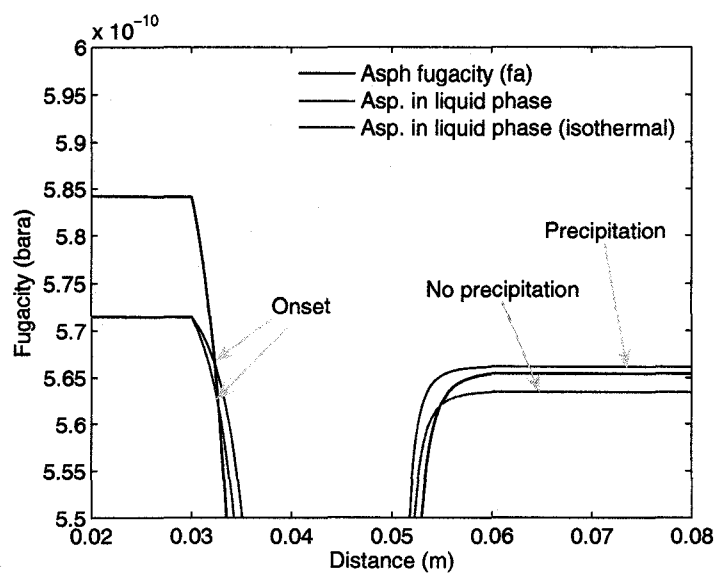


Figure 5-26: Effects of temperature on calculations of reference asphaltene fugacities.

in Figure 5-27. The point where the two curves first intersect represents the asphaltene precipitation onset. The part where the f_a -curve falls below the f_{nB}^L -curve ($f_a < f_{nB}^L$) specifies the conditions where asphaltene precipitation occurred. It is also shown in the figure that there is a slight difference in asphaltene onsets in the isenthalpic and the isothermal cases. When temperature was taken into account asphaltene started to precipitate upstream of the valve corresponding to the pressure of 357.788 bara and the temperature of 100.183 °C. When the upstream temperature ($T_1 = 100.154$ °C) was used and assumed constant throughout the length of the restriction the precipitation onset occurred a little further from the valve entrance and the onset pressure was found to be 357.652 bara. The isothermal case however showed no precipitation downstream of the valve (Figure 5-27(b)). This onset pressure predictions were consistent with the observation in experiments that asphaltene solubility decreases with increased temperature. Therefore the onset pressure was expected to be higher in the isenthalpic case where the increase in temperature was observed.



(a)



(b)

Figure 5-27: Asphaltene fugacity profile across ICD.

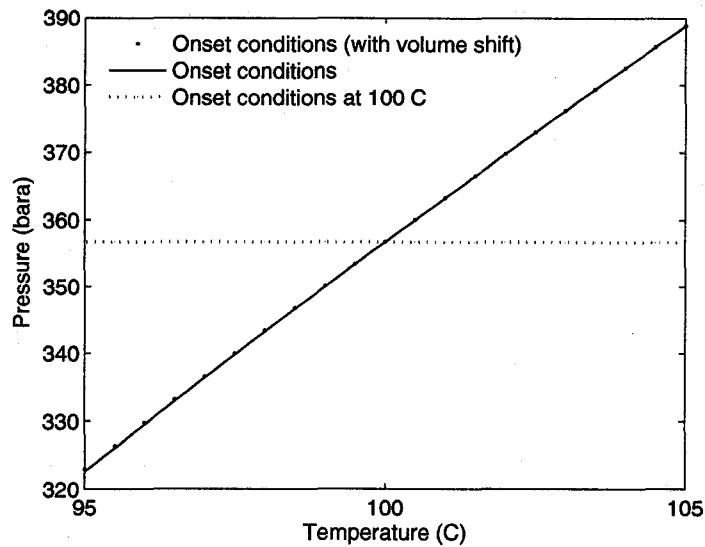


Figure 5-28: Asphaltene onset conditions predicted by the model.

Figure 5-28 shows asphaltene onset conditions predicted by the model at various temperatures using the asphaltene onset at 100 °C as the reference conditions. It is shown in the figure that the model provided an approximately linearly increasing function of asphaltene onset pressure and temperature. When a volume shift parameter of 0.0129 (as used in Nghiem and Coombe [39]) was used for the precipitating (nB^{th}) component, it had negligible effect on the predicted onset conditions. Note that the predicted onset pressures were not yet verified due to lack of experimental data. Figure 5-28 is therefore presented here only to provide the trend of the onset pressures predicted by the model and the effect of the volume shift on the onset prediction.

It was found from the analysis that the pressure drop in the restriction had more significant effect on asphaltene precipitation. Temperature also had influence on the precipitation prediction but to a less extent. There was a slight difference of 0.136 bara in the onset pressures calculated in the isothermal and the isenthalpic processes. The significance of the difference depended on many factors including

the prevailing pressure drop in the well. The difference however was small compared with the pressure drop induced by the restriction. In both isothermal and isenthalpic cases asphaltene started to precipitate somewhere inside the restriction. This means that the pressure drop induced by the restriction was the cause of solid asphaltene formation. As both the upstream and downstream pressures were higher than the experimental onset pressure of 356.69 bara at 100 °C, if only the pressures upstream and downstream of the valve were considered, the precipitation may not be detected. In addition the asphaltene onset pressure predicted by the model at 100.183 °C were higher than the 356.69 bara (experiment) at 100 °C by 1.098 bara. The difference is considered very large compared with the pressure drawdown of the well (the difference between the reservoir and the bottomhole pressure was 7.70 bara).

5.4 Case 2: Production from Reservoir with Varied Reservoir Temperature

In many cases oil production takes place in a reservoir with geothermal gradient or fluids are produced from a reservoir with two different temperature zones. In this example three production scenarios with varied temperature in the reservoir were investigated.

- Case A: Well with linearly decreasing reservoir temperature along the flow direction,
- Case B: Well with linearly increasing reservoir temperature along the flow direction, and
- Case C: Production of two distinct temperature zones.

5.4.1 Network Model

In Case A and Case B the well completed in a similar manner as the advanced completed well with ICV in case 1 (Figure 5-2(b)) was simulated. All the conditions for production were identical to that used in Case 1. The only difference was that the reservoir temperature in Case A was linearly decreasing from 100 °C at the toe to 80 °C at the heel of well. In Case B the temperature was linearly increasing from 80 °C to 100 °C at the heel. In both cases the fluid properties used in the network model were calculated at the averaged reservoir temperature of 90 °C except for oil viscosity where the values were calculated at the temperature of the fluid. The resulting temperature profiles in Case A and Case B are shown in Figure 5-29 and Figure 5-30, respectively. The temperature profiles illustrate the effect of energy transport by inflow fluid. In Case A for example, as the fluid flowed toward the heel of well the mixing with cooler inflow fluid from the reservoir led to a decrease in temperature. The green lines show the temperature profiles of the fluid when the well was insulated (ignoring the cooling effect of heat conduction through the surroundings). As we know from Case 1 that a single-phase oil tends to get hotter as it flows through the well, the temperature decrease shown by the green lines resulted only from the mixing with the cooler reservoir fluid. In addition when conduction (through solid surroundings) was considered, more significant temperature drop was observed compared with that in Case 1. This was due to the larger difference between the fluid temperature and the reservoir temperature leading to a higher rate of heat loss. The heat transport mechanisms in Case B can be explained in a similar manner. The fluid temperature in Case B increased more rapidly due to the higher-temperature reservoir fluid. The heat conduction effect observed in the simulation results (represented by the difference between the green and pink lines) also suggested the heat transport from the higher-temperature reservoir to the lower-temperature fluid in the well.

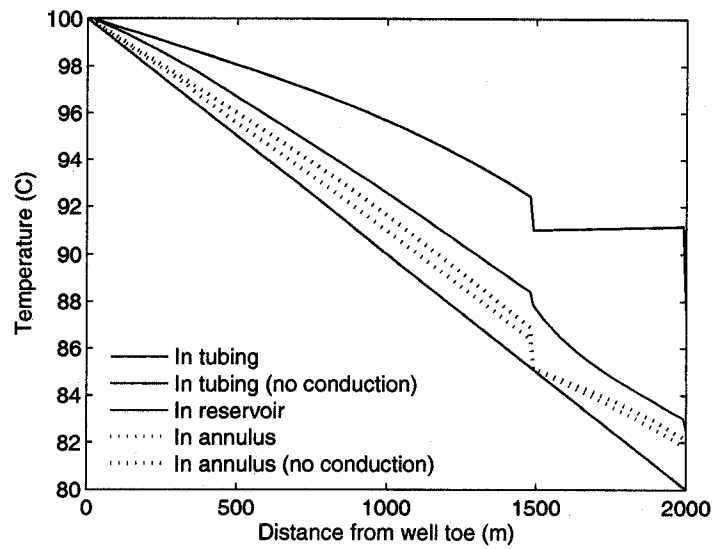


Figure 5-29: Temperature profile in the case with linearly decreasing temperature in reservoir.

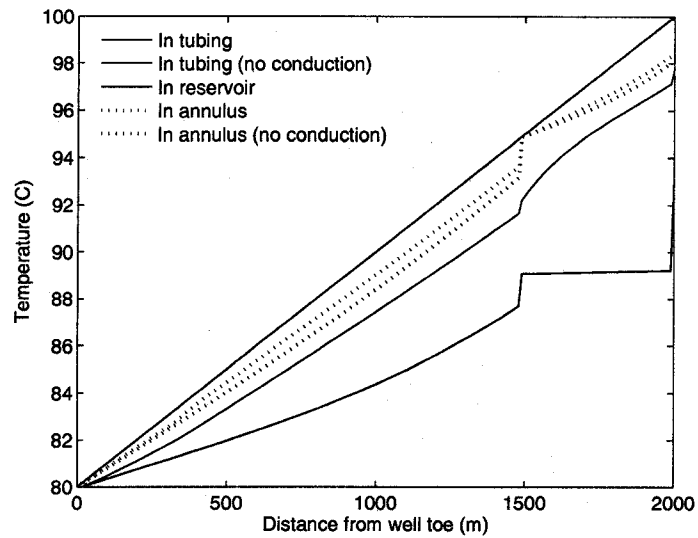


Figure 5-30: Temperature profile in the case with linearly increasing temperature in reservoir.

The pressure and flow rate profiles for these two cases were of the same shapes as those in Case 1. The effect of the different temperature scenarios on the production was also reflected in the predicted production rate. Due to lower oil viscosity at higher temperature, a slightly lower production rate of 7,895.4 m³/d was predicted in Case B compared with that of 7,996.9 m³/d in Case 1. For the same reason an even lower production rate of 7,615.9 m³/d was predicted in Case A as the majority of the fluid in the well was at a lower temperature.

Case C was an attempt to demonstrate the potential of the proposed model to be further modified and used with cases where fluids are produced from two different reservoirs or the case of multilateral wells. In this example the main well (Zone-1 well) was a 2000-meter well with a 1000-meter perforated section producing fluid from Zone-1 reservoir. The well was then connected with another well (Zone-2 well) producing fluid from Zone-2 reservoir. The Zone-2 well was 1000 meters long and was perforated throughout. This well system was modeled using the network model as shown in Figure 5–31. As seen in the figure the first half of Zone-1 well had both tubing and annulus while the second half only had tubing. The annular bridges of Segment 101 to Segment 200 of the well network were modified and used as the tubing of the well in Zone 2. The well basic characteristics were assumed equal to those used in Case 1 unless listed in Table 5–2.

In addition the two reservoirs had two distinct temperatures: 80 °C in Zone 1 and 100 °C in Zone 2. The reservoir pressures were assumed to be 370 bara in both Zone 1 and Zone 2. Similar to Case A and Case B, fluid properties were calculated at the averaged temperature of 90 °C. Furthermore, in order to simulate the wells that were situated in two separated reservoirs, it was assumed that there was no heat transfer between the two wells. Thus for simplicity purposes, the wells were assumed to be insulated ($U = 0$). The boundary conditions used in the temperature calculations included the reservoir temperatures of the corresponding

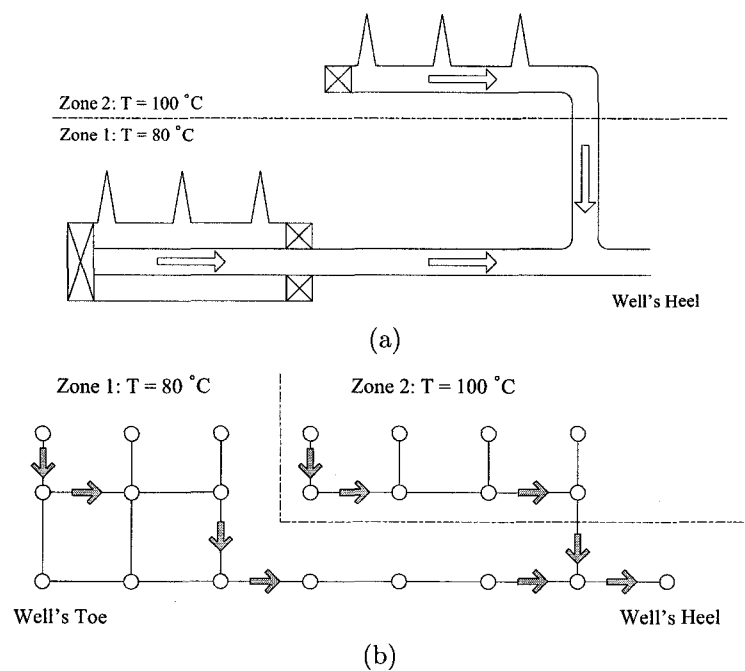


Figure 5-31: Example well network: Case 2-C. (a) actual system. (b) network model of the system.

Table 5-2: Well basic characteristics for the well in Case 2-C.

Property	Value
Well length (m)	2000
Reservoir pressure (bara)	370
Pressure at heel (bara)	358
Reservoir temperature Zone 1 ($^\circ\text{C}$)	80
Reservoir temperature Zone 2 ($^\circ\text{C}$)	100
Production rate (m^3/d)	12,300
Tubing diameter Zone 1 (m)	0.102
Well outside diameter Zone 1 (m)	0.122
Tubing diameter Zone 2 (m)	0.127
Joule-Thompson coefficient ($^\circ\text{C}/\text{bara}$)	-0.0499
Fluid heat capacity ($\text{J}/\text{mol } ^\circ\text{C}$)	351.94

segments; i.e. 100 °C at the inlet nodes of segments in Zone 1 and 80 °C at the inlet nodes of segments in Zone 2. The temperature at the toe of well was also used as a boundary condition and equal to 100 °C. In addition the set of equations used in the temperature calculations was modified such that the temperature in the annulus of Segment 101 where the Zone-2 inflow first enters the well was fixed at 80 °C.

Figure 5-32 shows the pressure profile of the well. The pressure drop in the tubing of Zone-1 well increased more rapidly after Segment 100 as the size of the well was reduced (no annulus). Figure 5-33 shows that more fluid was produced from Zone 2 than from Zone 1. This was because the diameter of the tubing of Zone-2 well was larger (0.127 m in Zone 2 compared with 0.102 m in Zone 1) and the pressure drawdown was larger near the bottomhole of the well. The large volume of fluid from Zone-2 well was combined with the tubing fluid of Zone-1 well at Segment 200 resulting in an abrupt increase in tubing flow rate and temperature. The temperature in Zone-1 well's tubing increased to 96.55 °C after mixing with the fluid from Zone-2 well (Figure 5-34).

To analyze the case further, in addition to temperature difference it was assumed that the two reservoirs also had different pressures. The pressures of 370 bara and 360 bara were assumed for Zone 1 and Zone 2 respectively. The results of the simulations were shown in Figure 5-35 to Figure 5-37. Due to lower reservoir pressure in Zone 2 less fluid was produced compared with the previous case of equal-pressure reservoirs resulting in a lower production rate of approximately 5,000 m³/d. The temperature of the tubing fluid after the mixing with the less amount of the fluid from Zone 2 was therefore lower than that in the equal-pressure case where the temperature of the mixed stream was found to be 90.48 °C.

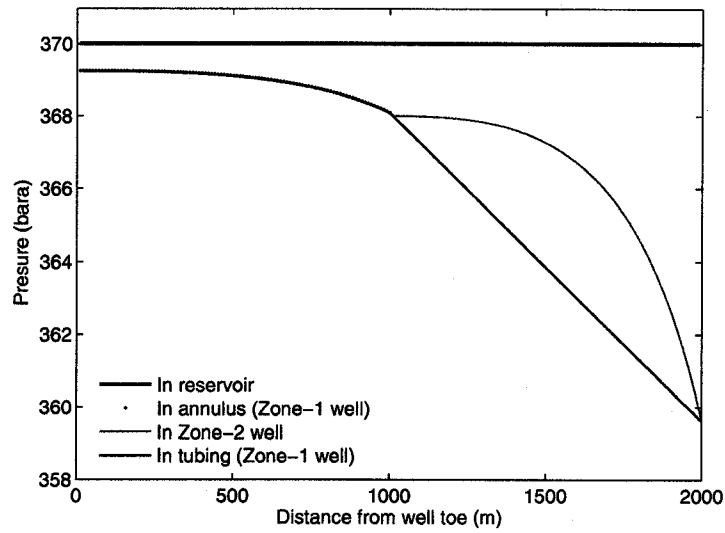


Figure 5-32: Pressure profile for the well in two reservoir zones with distinct temperatures.

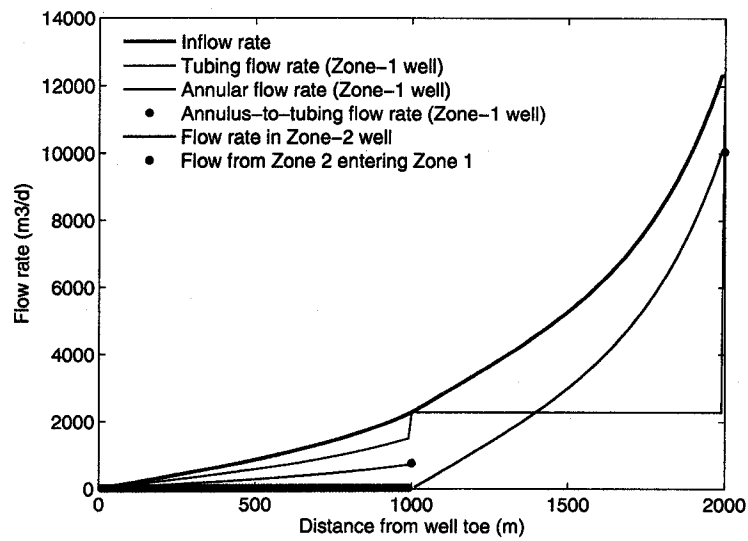


Figure 5-33: Flow rate profile for the well in two reservoir zones with distinct temperatures.

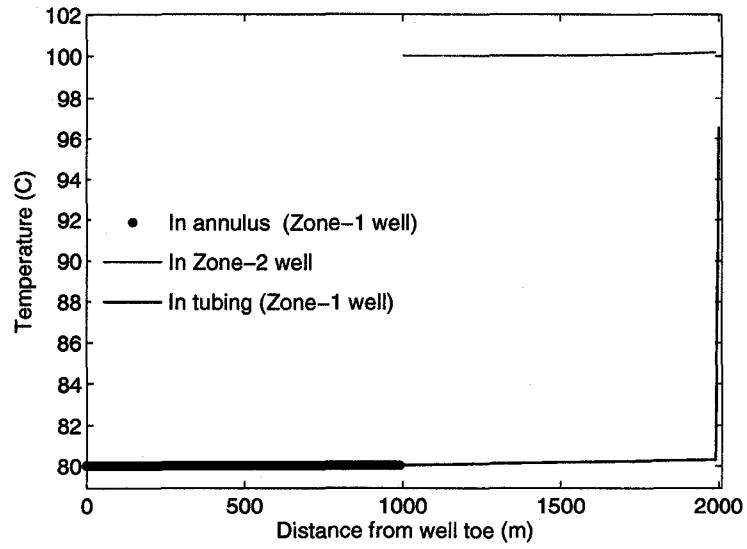


Figure 5-34: Temperature profile for the well in two reservoir zones with distinct temperatures.

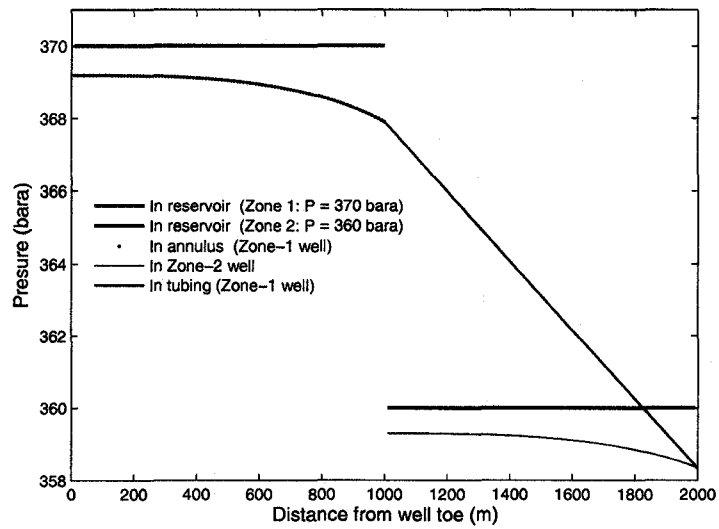


Figure 5-35: Pressure profile for the well in two reservoir zones with distinct temperatures and pressures.

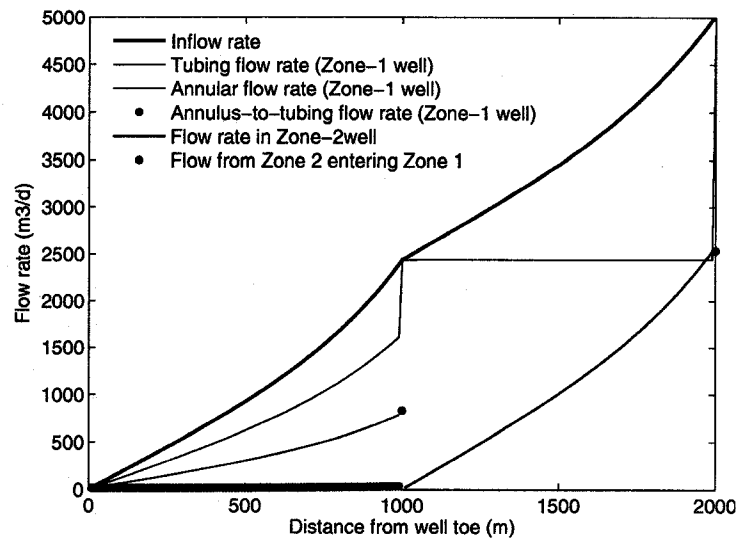


Figure 5-36: Flow rate profile for the well in two reservoir zones with distinct temperatures and pressures.

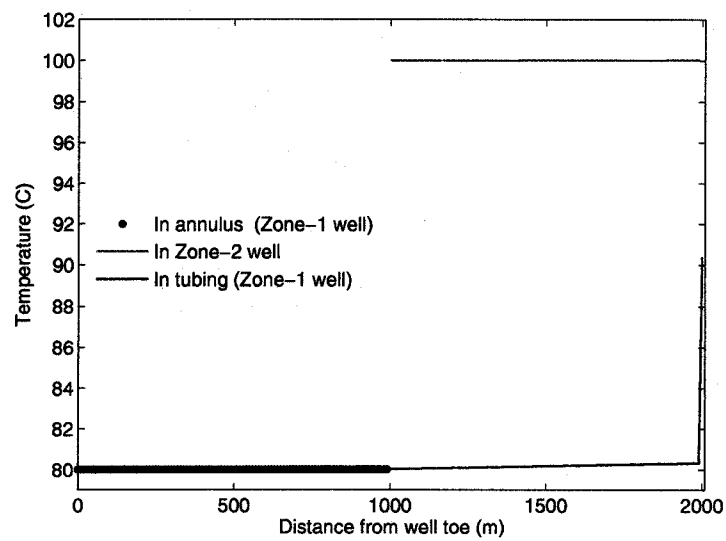


Figure 5-37: Temperature profile for the well in two reservoir zones with distinct temperatures and pressures.

5.4.2 Asphaltene Precipitation Prediction

Considering the well pressure profile there was a large pressure drop in the last segment. In addition due to mixing of two fluid streams from the two temperature zones there was also a drastic increase in temperature known for increasing the onset pressure for asphaltene precipitation. As there is no ICV the pressure profile due to the valve restriction were not evaluated. Instead, a one-point asphaltene precipitation analysis was performed at the location where the two streams were combined. The fluid conditions were obtained from the network model. In this case the pressure and the temperature at the heel of well representing the mixed fluid conditions at the lowest pressure were used. From the simulation, the conditions were 358 bara and 96.69 °C for the equal-pressure case and 358 bara and 90.48 °C for the different-pressure case.

As mentioned earlier, in order to provide an accurate prediction, asphaltene onset experiments performed at the temperature of consideration is required. Otherwise a relation between the reference asphaltene fugacity and temperature is needed to translate the reference asphaltene fugacity from the experimental to field conditions. As there were no other asphaltene onset data available for the example fluid to develop such relation, for demonstration purposes, it was assumed that the reference asphaltene fugacity (f_a^*) at 100 °C could be used as the reference fugacity at the interesting well temperatures. The pure asphaltene fugacity (f_a) was then calculated based on the reference asphaltene fugacity and found to be 5.74E-10 bara at 358 bara and 100 °C. Using the isothermal flash part of the proposed asphaltene model, the liquid-fugacity in the different-pressure case was found to be 1.63E-10 bara at 358 bara and 90.48 °C lower than the calculated asphaltene fugacity. Thus there was no asphaltene precipitation. For the equal-pressure case the liquid-fugacity was found to be 3.68E-10 bara compared with the asphaltene fugacity of 5.74E-10 bara

Table 5-3: Well basic characteristics for the well in Case 3.

Property	Value
Well length (m)	1000
Reservoir pressure (bara)	362
Pressure at heel (bara)	356.9
Reservoir temperature (°C)	100
Production rate (m ³ /d)	6,192.2
Tubing diameter (m)	0.127
Well outside diameter (m)	0.167
Valve opening (m)	0.02
Fluid heat transfer coefficient, k_{fl} (W·m ⁻¹ ·K ⁻¹)	0.306
Joule-Thompson coefficient (°C/bar)	-0.0487
Fluid heat capacity (J/mol °C)	358.38

indicating no precipitation as well.

5.5 Case 3: Well with Multiple Inflow Control Valves

To extend the application of the model to cases with more than one ICV, a 1000-meter well equipped with two ICVs was simulated. The ICVs, ICV 1 and ICV 2, were installed at 1000 meters and 950 meters from the toe of well respectively. The well was packed off so that the inflow from the last 300 meters of the well entered the well tubing through the two ICVs. Other well and fluid properties used in Case 1 were assumed in and used in this case unless shown in Table 5-3.

Using the network model the well was divided into 100 segments and modeled as shown in Figure 5-38(a). Assume that at the beginning of the production the two ICVs had the same valve discharge coefficients. Fluid in the annulus anywhere in the network were assumed to flow toward the heel of well. The resulting well pressure and flow rate profiles can be found in Figure 5-39 and Figure 5-40 respectively. Due to the difference in the flow rates through the two ICVs (there was a higher flow rate through ICV 1) the pressure drops across the valves were 1.124 bara and 0.858

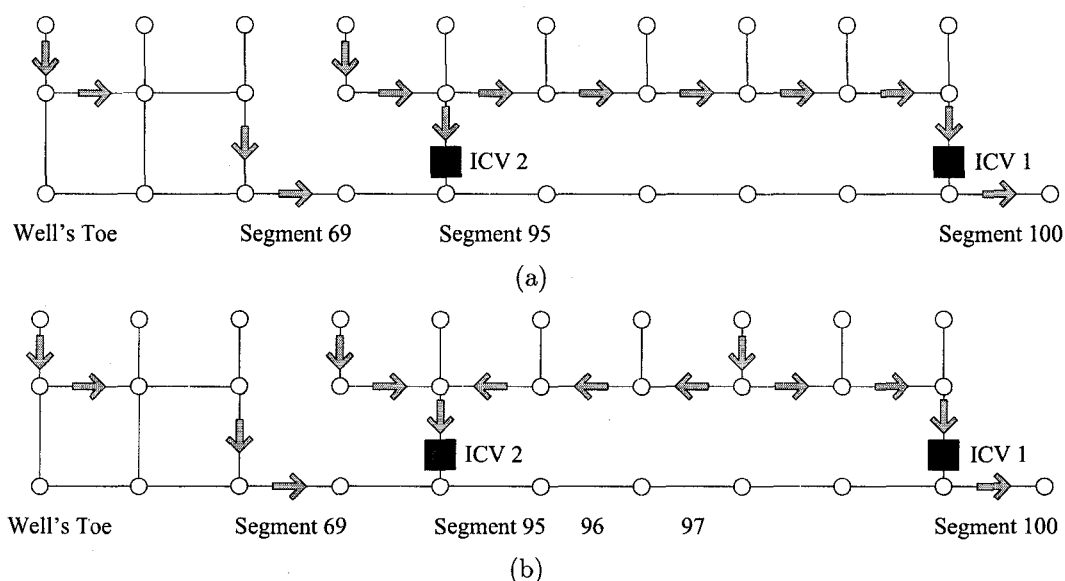


Figure 5-38: Example well network: Case 3. (a) without reversed flow in annulus. (b) with reversed flow in annulus.

bara for ICV 1 and ICV 2 respectively. The temperature profile was also generated and shown in Figure 5-41.

Using the ICV geometry from Case 1, detailed asphaltene analyses were conducted for ICV 1 and ICV 2 as described in Case 1. A friction factor of 0.2042 was used for both ICVs to match the pressure drops across the valves corresponding to the valve discharge coefficient of $52.65 \text{ USgpm}/(\text{psi})^{\frac{1}{2}}$. The results of the fugacity analyses showed asphaltene precipitations in both ICVs as shown in Figure 5-42.

Assume that ICV 1 was plugged by the asphaltene precipitate causing the area of the valve opening to reduce by approximately 21.8% and the valve pressure drop to increase. However, ICV 2 was assumed intact. Due to the increased resistance to the flow, the fluid sought a less resistant flow path which was through ICV 2. Therefore a lower flow rate was expected through ICV 1. Regardless of the change all the flow directions in the well network were first assumed toward the heel of well similar to the previous case before the valve was plugged. The results of the simulations at the convergence were all imaginary numbers.

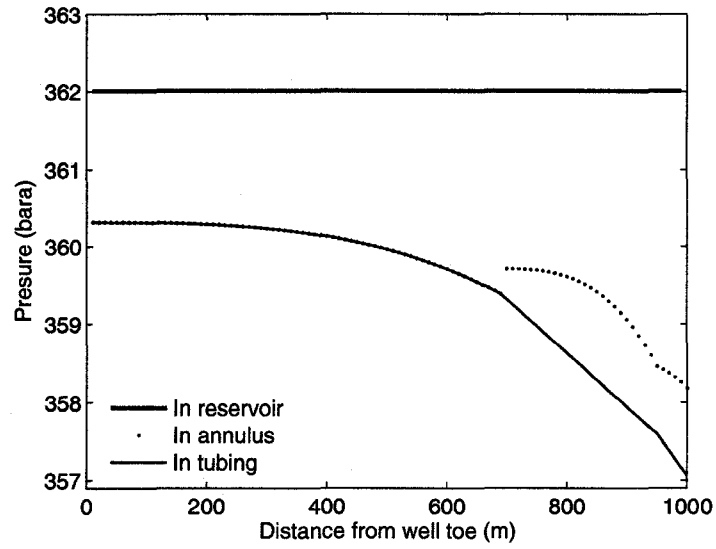


Figure 5-39: Pressure profile in the case with two ICVs.

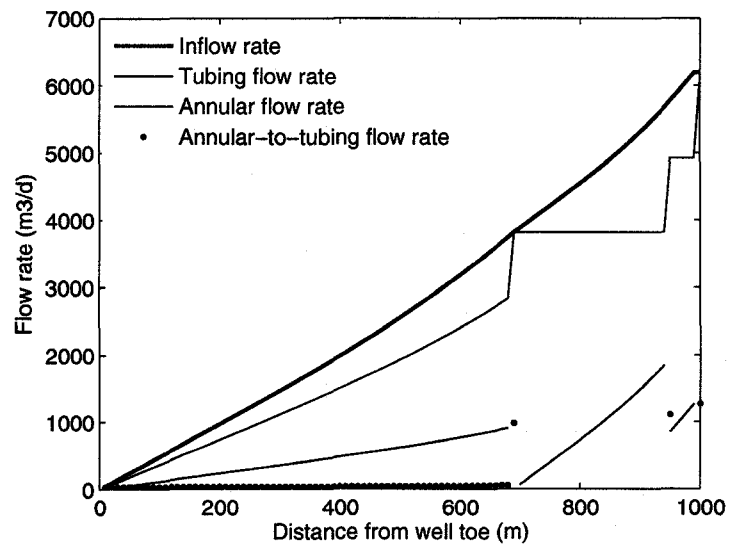


Figure 5-40: Flow rate profile in the case with two ICVs.

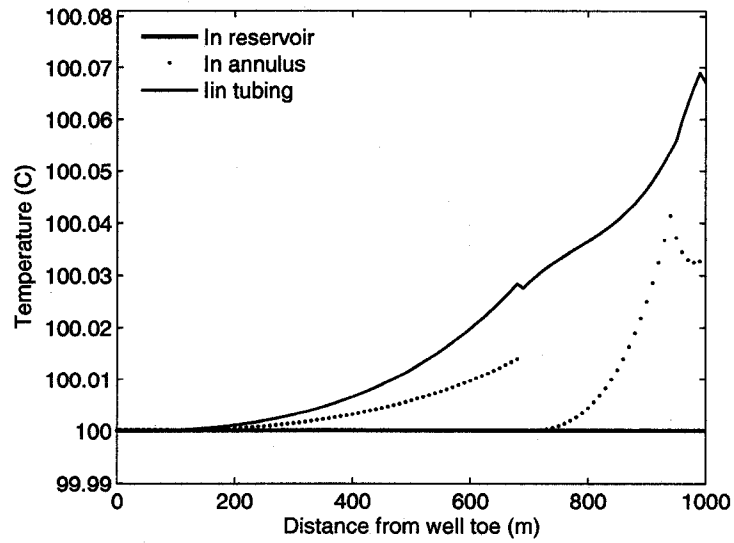


Figure 5-41: Temperature profile in the case with two ICVs.

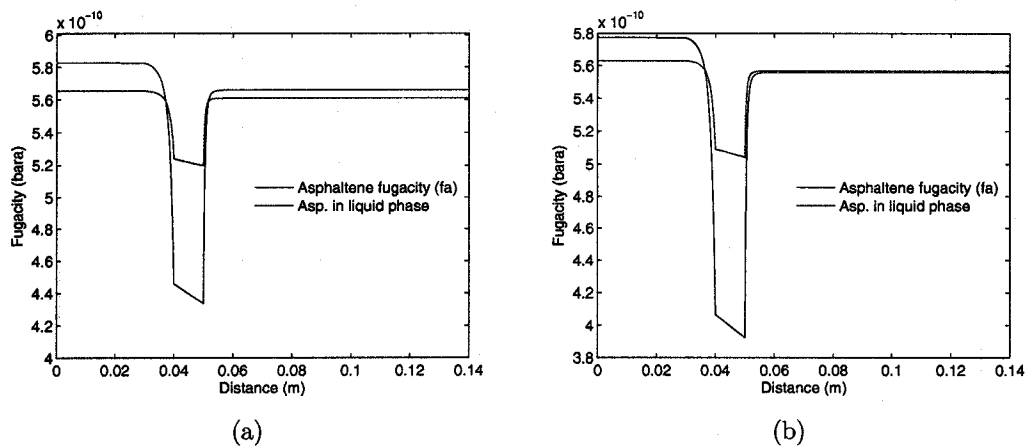


Figure 5-42: Fugacity profile through ICVs in Case 3 indicating asphaltene precipitation. (a) for flow through the ICV at 950 meters from the toe of well. (a) for flow through the ICV at 1000 meters from the toe of well.

As expected, the fluid in the annulus might have flowed toward the preferred lower-resistant ICV 2, the incorrect assignment of the flow directions for annular bridges of segments between the ICV 1 and ICV 2 could be the cause of the imaginary solutions. The annular flow direction in Segment 95 was therefore switched toward ICV 2 and the simulation was repeated. It was found that the magnitude of the imaginary parts in the results decreased as the annular flows in more segments (starting from Segment 95) were switched in direction. By trial and error it was found that real-number results were obtained when the flow in the annulus of 3 segments after the location of ICV 2 (i.e. Segment 95, 96, and 97) were reversed toward the valve as shown in Figure 5–38(b). The results from the network model after the flow directions were altered are shown in Figure 5–43 to 5–45. In Figure 5–43, instead of the continuous decrease in pressure toward the heel of well, an increase in pressure was observed in a small section in the annulus between 950 and 1000 meters from the toe of the well. Figure 5–44 shows negative flow rate illustrating the reversed direction of the flow in the annulus toward the ICV 2. The tubing flow rate at 950 meters from the toe of the well was also found higher than the sum of the inflows up to Section 95 (represented by the Inflow rate curve) because the inflows from the next two segments also entered the tubing at this location. In the temperature calculations the energy equations were modified such that the first node connected to the reversed flow (annular node of Segment 97) had a temperature equal to the reservoir temperature. As a result, as seen in the temperature profile (Figure 5–45), a reservoir temperature was observed at the annular node of Segment 97 where the only flow entering the node was from the reservoir.

5.6 Case 4: Well with Restricted Flow in the Annulus

In oil and gas productions equipment (e.g. flow measuring equipment or flow control valves) is commonly installed on the tubing. In this example an equipment

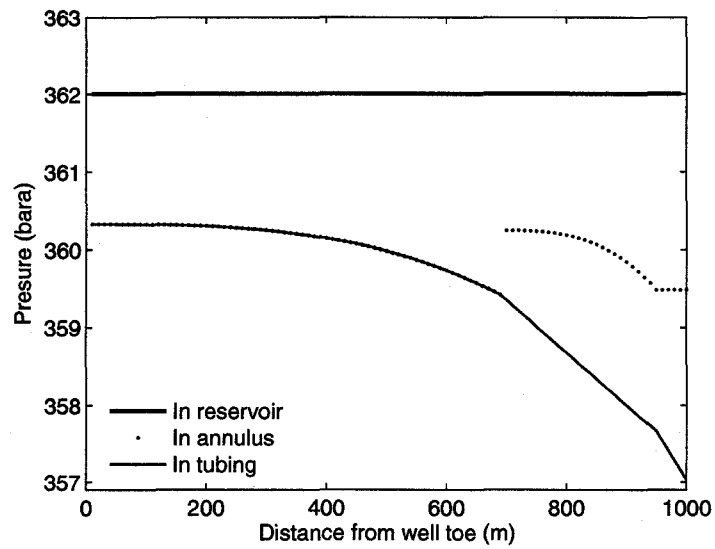


Figure 5-43: Pressure profile in the case where the ICVs caused reversed flow in the annulus.

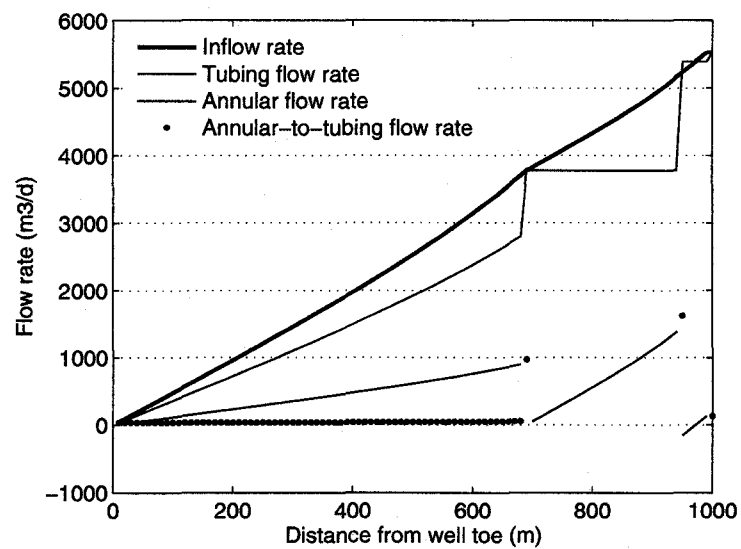


Figure 5-44: Flow rate profile in the case where the ICVs caused reversed flow in the annulus.

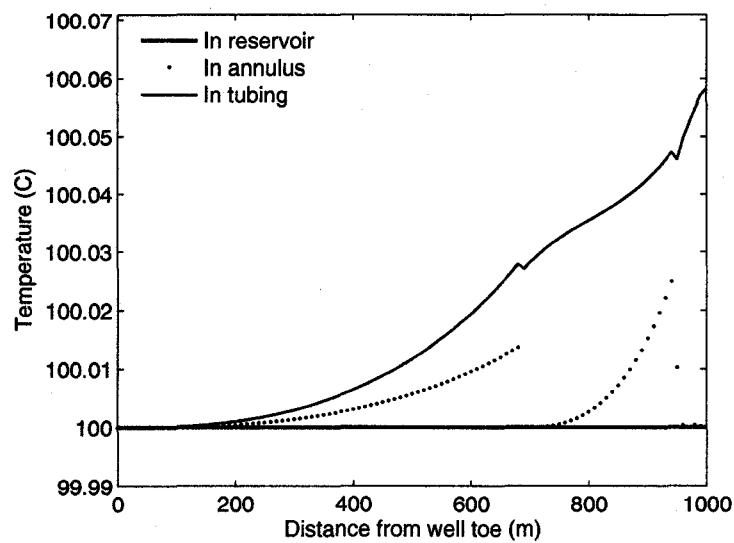


Figure 5-45: Temperature profile in the case where the ICVs caused reversed flow in the annulus.

was assumed to be installed in the tubing and caused the cross-sectional area in the annulus to decrease restricting the flow in the annulus. To analyze this problem a 2000-meter well with an equipment installed at 1,700 meters from the toe of well was investigated. The well was modeled using a network model as shown in Figure 5-46. Again, well and fluid characteristics used in Case 1 were assumed unless reported in Table 5-4.

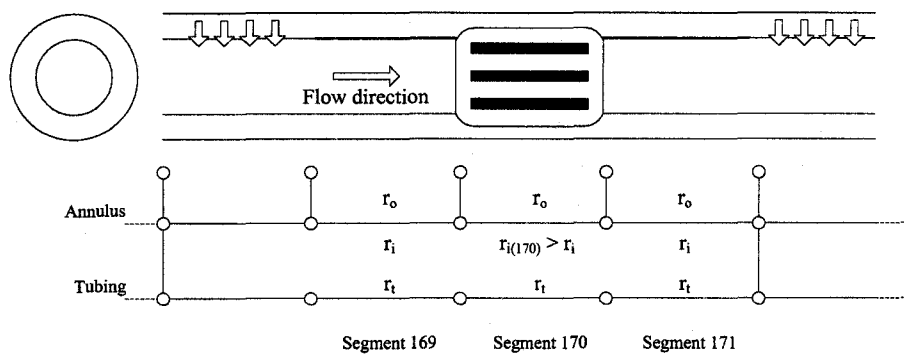


Figure 5-46: Example well network: Case 4.

Table 5–4: Well basic characteristics for the well in Case 4.

Property	Value
Well length (m)	2000
Reservoir pressure (bara)	365
Pressure at heel (bara)	357
Reservoir temperature (°C)	100
Production rate (m ³ /d)	9,856.6
Tubing diameter (m)	0.127
Tubing diameter of Segment 170 (m)	0.137
Well outside diameter (m)	0.167
Fluid heat transfer coefficient, k_{fl} (W·m ⁻¹ ·K ⁻¹)	0.306
Joule-Thompson coefficient (°C/bar)	-0.0487
Fluid heat capacity (J/mol °C)	358.36

5.6.1 Network Model

As shown in Figure 5–46 the well was first divided into 200 segments with the equipment installed in Segment 170. The restriction was placed in the annulus and was assumed to take up 10 meters of the well length which was equal to the entire length of the 170th segment. The change in the annular cross-section caused by the equipment was modeled by assigning an inner radius (r_i) for Segment 170's annulus differently from those in the adjacent segments. The inner radius of the particular segment was assumed to increase by 0.01 meters from the equal inner radii of other segments. This caused the annular cross-sectional area of Segment 170 to decrease by 22.45%. The outer radius of the annulus (r_o) and the radius of the tubing (r_t) in Segment 170 remained equal to those in other parts of the well. Due to the existence of the equipment and in order to obtain a well-developed flow through the equipment it was assumed that there were no annulus-to-tubing flow in Segment 169, 170, and 171 as shown in the figure. It was found from the simulations that removal of the annulus-to-tubing helped in the convergence of the solutions by reducing abrupt changes in flow conditions of the fluid. Using the network model, the

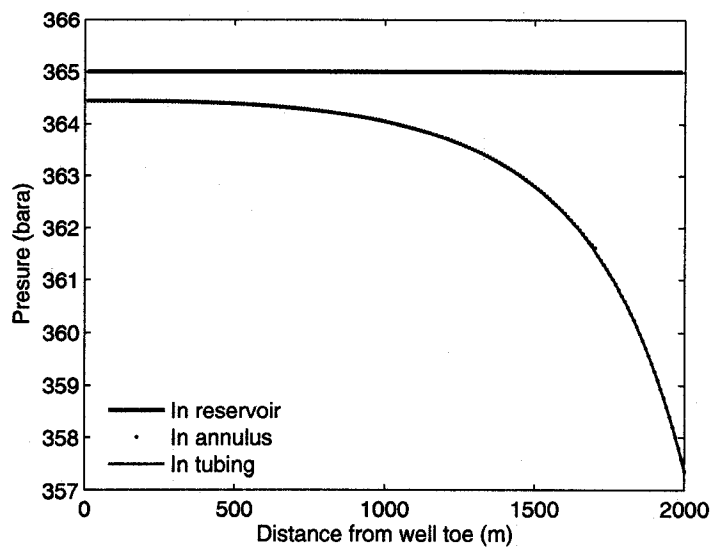


Figure 5-47: Pressure profile in well with restricted flow in annulus.

results of the simulations are shown in Figure 5-47 to Figure 5-49. Even though there was a restriction in the annular space of the well it can be seen from the results that this type of restriction did not significantly affect the pressure profile in the annulus. This was partly because of the reduced flow rate through the restricted annulus (Figure 5-48). Similar to the case with two ICVs, the fluid tended to seek a lower-resistant flow path and thus entered the tubing before the restriction (seen as the abrupt increase in tubing flow rate of Segment 168).

5.6.2 Asphaltene Precipitation Prediction

A detailed asphaltene analysis was then conducted for the fluid flowing through the restricted annulus. For this example the restriction geometry was defined a little differently from the previous cases. An annular cross section (as opposed to the circular cross section) was used to model the restriction in this case. For the simplified geometry as shown in Figure 5-50 there was an additional parameter r_4 representing the outer radius of the annulus. The cross-sectional areas of the restriction were then expressed in terms of the distance from the valve entrance (z)

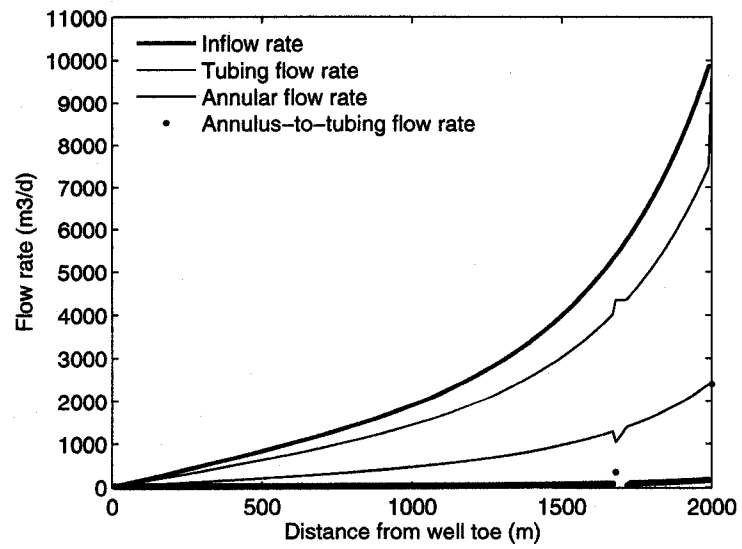


Figure 5-48: Flow rate profile in well with restricted flow in annulus.

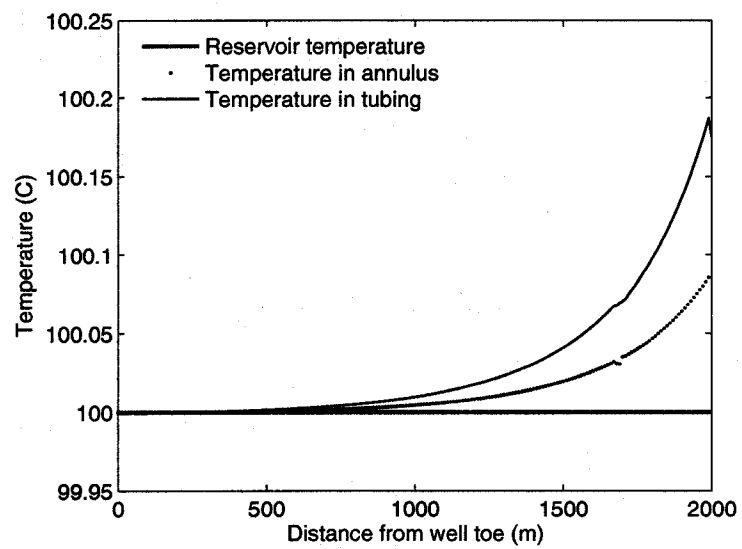


Figure 5-49: Temperature profile in well with restricted flow in annulus.

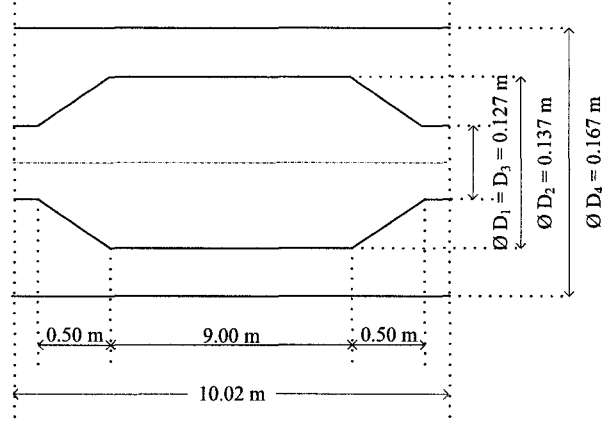


Figure 5-50: Simplified restriction geometry used to model the restricted annulus in Case 4.

as:

$$A(z) = \begin{cases} \pi r_4^2 - \pi r_1^2 & 0 \leq z \leq A \\ \pi r_4^2 - \pi \left(\frac{r_2 - r_1}{B - A} (z - A) + r_1 \right)^2 & A \leq z \leq B \\ \pi r_4^2 - \pi r_2^2 & B \leq z \leq C \\ \pi r_4^2 - \pi \left(\frac{r_3 - r_2}{D - C} (z - C) + r_2 \right)^2 & C \leq z \leq D \\ \pi r_4^2 - \pi r_3^2 & D \leq z \leq L \end{cases} \quad (5.7)$$

where other symbols in the above equations are defined as in Figure 4-11.

The pressure drop in the annulus from Segment 170 to Segment 171 was found to be 0.155 bara from the network model. A friction factor of 0.0255 was used to match this pressure drop for the flow rate of 1,224 m³/d through the particular annular space. The asphaltene analysis was then conducted in a similar manner as that described in Case 1. The calculated pressure and temperature profiles through the restriction are shown in Figure 5-51 and Figure 5-52 respectively. The fugacity of the precipitating component at the conditions inside the restriction was also compared with the corresponding asphaltene fugacity as shown in Figure 5-53. As expected, because the fluid condition only changed slightly in this case, the results showed no asphaltene precipitation.

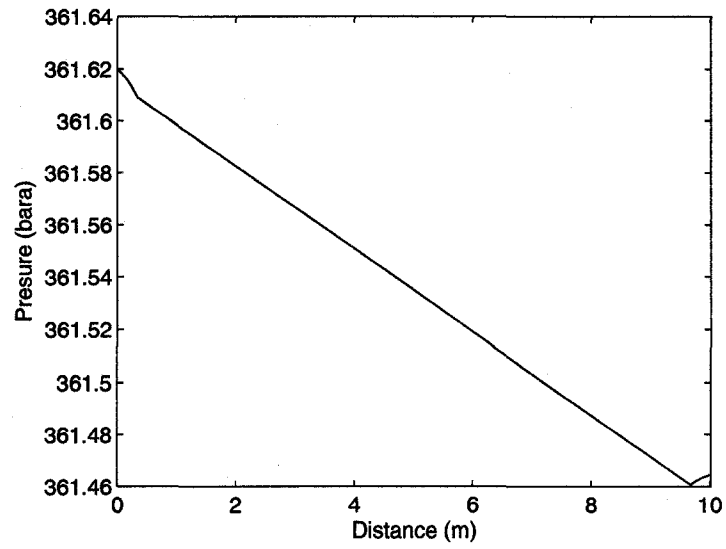


Figure 5-51: Pressure profile through the restricted annulus.

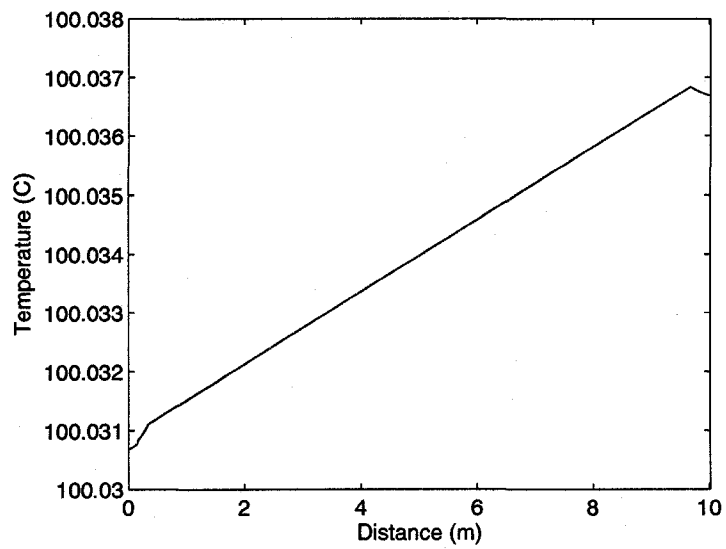


Figure 5-52: Temperature profile through the restricted annulus.

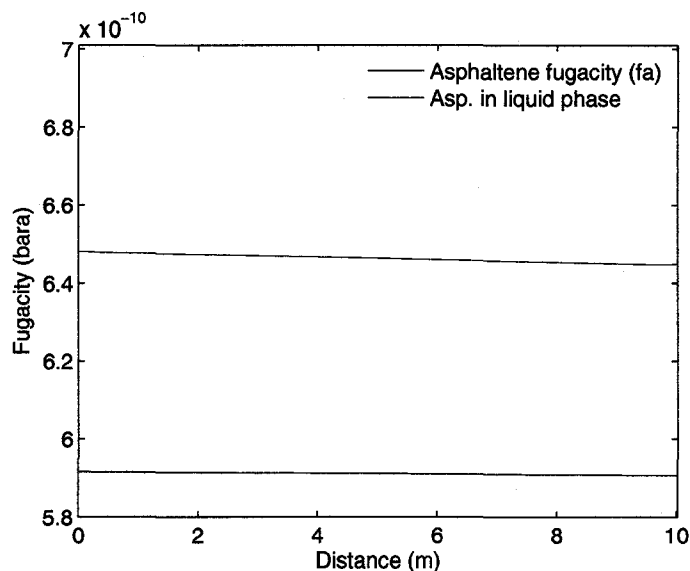


Figure 5-53: Fugacity profile through the restricted annulus.

5.7 Discussions

In this chapter the proposed methodology was used to analyze the wells with different completion scenarios. It was found from the example applications that the proposed network model was able to accurately portray the fluid behaviour in the wellbore. The model was successfully used to predict pressure, flow rate, liquid holdup, and temperature profiles influenced by the characteristics of different well completions. At the locations where asphaltene precipitation was expected to be a risk, the proposed asphaltene model was used to predict local asphaltene precipitation potential in cases with or without flow restrictions. It was found from the asphaltene analysis that pressure drop in restriction has significant effect on asphaltene precipitation. Because asphaltene started to precipitate somewhere inside the restriction it means that the pressure drop introduced by the restriction was the cause of solid asphaltene formation. Based on the finding of this work, if the asphaltene onset pressure of 356.69 bara at 100 °C (experiment) was the only factor used to determine asphaltene precipitation and only the pressures upstream and

downstream of the valve were considered, the precipitation would not be detected. Different flow geometries and well completions also have different effects on the asphaltene precipitation. This depends highly on the extent to which the geometries (or completions) cause fluid conditions to change. There is usually a significant variation in fluid conditions when fluid is forced to flow through a small cross-sectional area or enter a space at a single point in the well. When the fluid has an alternative flow path with lower flow resistance, it tends to avoid the high-resistant path causing less abrupt change in fluid conditions.

The model was also used to investigate the effects of temperature. Even though there was a lack of experimental data for model verifications the model provided a reasonable trend of asphaltene onset predictions. Based on the asphaltene model used, it was found that temperature also has influence on the precipitation onset prediction. In the example Case 1 the difference in the onset pressures calculated at the reservoir temperature (isothermal process) and at the predicted wellbore temperature was 1.098 bara. The difference could be significant depending on the prevailing pressure drop in the well.

Nonetheless, even though the study showed that a restriction can cause asphaltene precipitation, the extent of asphaltene problem may need to be further investigated. Many studies have shown reversibility of solid asphaltene when the fluid conditions are quickly retrieved back to the no-asphaltene stage. This phenomenon is similar to what happens in the valve. The pressure decreases and then quickly increases downstream of the valve. In addition asphaltene formation also showed some supersaturation before it actually precipitated as discussed in Chapter 4. Consequently, even though asphaltene precipitation is found inside the restriction it cannot be concluded just yet that the precipitated solid will cause any problem downstream of the valve. Further investigations on the effect of restriction on such

precipitation and the problems it may cause to the production process are therefore required.

CHAPTER 6

CONCLUSIONS, RESEARCH NOVELTY, AND RECOMMENDATIONS

6.1 Summary and Conclusions

A methodology to predict asphaltene precipitation in advanced production wells is proposed. The methodology is executed using the proposed comprehensive flow model incorporating non-isothermal and compositional effects. The proposed model is a simplified model which is able to portray flow behaviour in the wellbore by predicting flow behaviour of fluids in wells with advanced completions. The model can also be used to predict phase behaviour of the fluids focusing on asphaltene precipitation induced by flow restrictions typically found as parts of advanced well completions.

The proposed model consists of a black-oil, non-isothermal network model and a compositional asphaltene model. To simulate a well the network model is first used. The network model is a two-phase, non-isothermal flow model which is able to predict pressure, temperature, flow rate, and liquid holdup of fluids for the entire well. Given the overall behaviour of the fluid in the well, it is possible to determine the location in the well network where asphaltene precipitation is most likely to occur. A local detailed analysis can then be conducted at the suspected location to predict asphaltene precipitation. Local pressure profile influenced by local restricted flow paths is first determined. The proposed semi-empirical solid asphaltene model

is then used to predict solid phase behaviour under an isenthalpic condition. Local temperatures inside the valve are determined based on energy balance. Using pseudo-three-phase equilibrium calculations, the precipitation is determined based on pre-calculated solid asphaltene fugacity.

Example applications of the methodology and the model were provided. It can be concluded from the example simulations that the first objectives of the research have been achieved. Various horizontal production wells with different completion schemes were successfully modeled using the proposed semi-compositional model. The compositional effect was also effectively incorporated through the use of local asphaltene precipitation analysis. In the example applications, wells with various fluid and reservoir conditions (e.g. different rates of production, pressure draw-downs, and reservoir properties) were investigated. The wells were also completed differently. The example cases include wells equipped with one or more inflow control valves, a well with a section of restricted annulus due to increase in tubing diameter, and wells producing from two zones of reservoir with different conditions. Due to the drastic change in flow conditions where the abrupt change in flow geometry existed, asphaltene precipitations were suspected. For this reason asphaltene precipitation analyses were conducted to determine whether solid asphaltenes were likely to form at such locations.

From this study changes in pressure and temperature conditions induced by restricted flow paths actually have effects on asphaltene precipitation behaviour. It was found that valve restrictions can cause asphaltene to precipitate due to drastic decrease in pressure. Even though it is not observed under downstream and upstream conditions, the precipitation may occur at conditions inside the restriction. Changes in temperature during a production process also have effects on asphaltene onset pressure. The extent of the effect depends on the characteristics of the system. The shift in the onset pressure may be large compared with the prevailing

pressure drops in the network. Based on the simulation results as large as 1.098 bara difference in predicted asphaltene onset pressures could be observed when an isothermal production process was assumed instead of considering the actually fluid temperature. However even though it was found that a valve restriction can promote asphaltene precipitation, the precipitated solid might not exist or cause any problem downstream of the valve. This depends on many factors including size of precipitates, asphaltene reversibility, and supersaturation.

6.2 Novelty of the Research

To the best of our knowledge, attempts have never been given to integrating an asphaltene model into a well network model. This research proposed a comprehensive semi-compositional model that predicts asphaltene in horizontal production wells. An approach to predict temperature profile in wellbores using a network-type model was also proposed. The proposed network model is able to predict temperature in the well including the fluid in the annular space of the well. By incorporating the compositional and non-isothermal effects the proposed model provides better understanding of flow behaviour of fluids in different parts of production systems including flow through restrictions.

The advanced well completion is very common nowadays as to achieve more control in oil and gas production. Therefore, determination of flow behaviour through well equipment becomes more necessary. This research proposed a model to predict conditions and phase behaviour of the fluid flowing through the well completion equipment using simplified restriction geometries.

A modification has been introduced to an existing solid asphaltene model in order to make it more applicable to the case with flow restrictions. This was achieved by incorporating an isenthalpic flash into an isothermal-flash solid asphaltene model.

6.3 Recommendations for Future Research

As mentioned, this research was meant to provide a simplified comprehensive approach to modeling of wellbore flows. The proposed model is a fundamental model and is at its early stage of development. Further research is therefore recommended to improve the model:

1. In the proposed model there are a few parameters used to represent fluid behaviour in complex environments. These values should be further investigated in order to best represent the fluid. Such parameters include two-phase friction factor and heat transfer coefficient of the fluid in the annulus. The overall heat transfer coefficient is generally evaluated between fluid in the tubing and the reservoir. In the network model temperature of the annular fluid is also predicted. Appropriate overall heat transfer coefficients for each interface i.e. tubing-annulus and annulus-reservoir interfaces should be determined in order to provide accurate temperature predictions.
2. In order to improve the accuracy of the model, further verification of the proposed model using available field data will prove beneficial. In addition, the proposed model is constructed with various models including inflow model, pressure drop through valve model, and asphaltene model. The accuracy of the proposed model highly depends on the accuracy of these models. The models can be modified to better suit different flow characteristics (e.g. laminar/turbulent flows, or two-phase flow patterns). Regarding the asphaltene model, asphaltene precipitation is a very fluid-specific problem, other available asphaltene models (such as experiment-based models) are worth being considered. The models not only can be highly compatible with the network model, but also more predictive in some situations providing case-specific and more accurate results.

3. Further modifications to the network model allowing customization to a wider range of well completion cases are recommended. At the current stage the model is still unstable in many cases especially when abrupt changes in flow parameters occur in the well. This problem should be further resolved. In addition even though the objective of the research is to provide a simplified solution, the model has potential of being used with more complex well designs or restriction geometries. Appropriate modifications to serve more complex situations are also recommended.
4. In this research, as the problem of interest is asphaltene, the study was focusing on undersaturated fluid. Further investigation on the cases of two-phase fluids or to include a third phase (e.g. to include water) should not be neglected. Incorporating models to predict other solid problems such as hydrates, scale, and wax is also possible.
5. The model may also be extended for use in transient flow problems. For a very short period after the startup of the well abrupt changes in fluid conditions occur close to the heel of the well. Immediate expansion of the fluid may cause solids to precipitate particularly hydrates and probably asphaltene as well.
6. In order to provide a complete analysis of the production system integration of the proposed model with a reservoir model is also possible and recommended.

References

- [1] R. K. Agarwal, Y. Li, L. X. Nghiem, and D. A. Coombe. Multiphase multicomponent isenthalpic flash calculations. *Journal of Canadian Petroleum Technology*, 30(3):69–75, May–Jun 1991.
- [2] M. A. Aquino-Olivos, E. Buenrostro-Gonzalez, S. I. Andersen, and C. Lira-Galeana. Investigations of inhibition of asphaltene precipitation at high pressure using bottomhole samples. *Energy & Fuels*, 15(1):236–240, 2001.
- [3] H. Asheim, J. Kolnes, and P. Oudeman. A flow resistance correlation for completed wellbore. *Journal of Petroleum Science and Engineering*, 8:97–104, 1992.
- [4] K. Baker. Understanding paraffin and asphaltene problems in oil and gas wells. South Midcontinent Region Workshop, Smackover, Arkansas, July 2003.
- [5] H. D. Beggs and J. R. Robinson. Estimating the viscosity of crude oil systems. *Journal of Petroleum Technology*, pages 1140–1141, Sep 1975.
- [6] R. B. Bird, W. E. Stewart, and E. N. Lighthfoot. *Transport Phenomena*. John Wiley & Sons, Inc., New York, 2nd edition, 2002.
- [7] H. B. Bradley, editor. *Petroleum Engineering Handbook*. Society of Petroleum Engineers, Richardson, TX, U.S.A, 1992.
- [8] K. Brekke. *Horizontal Well Productivity and Risk Assessment*. Ph.d. thesis, The Graduate School. The University of Tulsa, 1996.
- [9] K. Brekke, T. E. Johansen, and R. Olufsen. A new modular approach to comprehensive simulation of horizontal wells. 1993. SPE paper 26518.
- [10] J. P. Brill and H. Mukherjee. *Multiphase Flow in Wells*, volume Monograph volume 17 of *SPE Henry L. Doherty Series*. Society of Petroleum Engineers Inc., Richardson, Texas, 1999.
- [11] N. E. Burke, R. E. Hobbs, and S. F. Kashou. Measurement and modeling of asphaltene precipitation. *Journal of Petroleum Technology*, pages 1440–1456, Nov 1990.
- [12] Calsep Inc. Method documentation pvtsim 15. Software documentation, 2005.
- [13] P. L. Chueh and J. M. Prausnitz. Vapour-liquid equilibria at high pressures calculation of partial molar volume in non-polar liquid mixtures. *AIChE*, 13(6):1099–1113, 1967.

- [14] R. Cimino, S. Corraera, and P. A. Sacomani. Thermodynamic modeling for prediction of asphaltene deposition in live oil. The 1995 SPE International Symposium on Oilfield Chemistry, San Antonio, Texas, Feb 1995. Paper SPE 28993.
- [15] Crane Co. Flow of fluids through valves, fittings, and pipe. Technical Paper No. 410M, Woodlands, TX, 1999.
- [16] A. Danesh. *PVT and Phase Behaviour of Petroleum Reservoir Fluids*. Elsevier, Amsterdam, The Netherlands, 1998.
- [17] P. Dawkrajai, K. Yoshioka, A. A. Romero, D. Zhu, A. D. Hill, and L. W. Lake. A comprehensive statistically-based method to interpret real-time flowing measurements. Annual Report DE-FC26-03NT15402, The University of Texas at Austin, Texas A&M University, Oct 2005.
- [18] R. B. de Boer, K. Leerlooyer, M. R. P. Eigner, and A. R. D. van Bergen. Screening of crude oils for asphalt precipitation: Theory, practice, and the selection of inhibitors. *SPE Production & Facilities*, pages 55–61, Feb 1995.
- [19] M. J. Economides, A. D. Hill, and C. Ehlig-Economides. *Petroleum Production Systems*. Prentice Hall Petroleum Engineering Series. Prentice Hall, Inc., Upper Saddle River, NJ, 1994.
- [20] A. Firoozabadi. *Thermodynamics of Hydrocarbon Reservoirs*. McGraw Hill, Boston, 1999.
- [21] M. S. Graboski and T. E. Daubert. A modified Soave equation of state for phase equilibrium calculations. 1. hydrocarbon systems. *Ind. Eng. Chem. Process Des. Dev.*, 17(4):443–448, 1978.
- [22] A. K. Gupta. A model for asphaltene flocculation using an equation of state. M.Sc. thesis, Department of Chemical and Petroleum Engineering, University of Calgary, Jun 1986.
- [23] S. E. Haaland. Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering*, 105, Mar 1983.
- [24] A. R. Hasan and C. S. Kabir. *Fluid Flow and Heat Transfer in Wellbores*. Society of Petroleum Engineers, Richardson, Texas, 2002.
- [25] A. Hirschberg, L. N. J. deJong, B. A. Schipper, and J. G. Meijer. Influence of temperature and pressure on asphaltene flocculation. *Society of Petroleum Engineers Journal*, pages 283–293, Jun 1984.
- [26] A. K. M. Jamaluddin, J. Creek, C. S. Kabir, J. D. McFadden, D. D'Cruz, M. T. Joseph, N. Joshi, and B. Ross. A comparison of various laboratory techniques to measure thermodynamic asphaltene instability. 2001. Paper SPE 72154.

- [27] A. K. M. Jamaluddin, J. Nighswander, and N. Joshi. Systematic approach for characterizing hydrocarbon solid. *SPE Journal*, pages 304–12, Sep 2003. SPE paper 86573.
- [28] A. K. M. Jamaluddin, J. Nighswander, N. Joshi, D. Calder, and B. Ross. Asphaltenes characterization: A key to deepwater developments. 2002. Paper SPE 77936.
- [29] A. C. Johansen. Work term report, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL, Canada, 2005.
- [30] T. E. Johansen. Principles of reservoir engineering. Advanced Reservoir Engineering course note (Draft), 2005. Faculty of Engineering and Applied Science, Memorial University of Newfoundland.
- [31] N. B. Joshi, O. C. Mullins, A. Jamaluddin, J. Creek, and J. McFadden. Asphaltene precipitation from live crude oil. *Energy & Fuels*, 15(4):979–986, 2001.
- [32] T. M. V. Kaiser, S. Wilson, and L. A. Venning. Inflow analysis and optimization of slotted liners. *SPE Drilling & Completion*, pages 200–208, Dec 2002.
- [33] D. L. Katz. Thermodynamics analysis of frictional heat effects in pipeline flow. *Oil and Gas Journal*, Mar 1972.
- [34] S. Kawanaka, S. J. Park, and G. A. Mansoori. The role of asphaltene deposition in eor gas flooding: A predictive technique. The 1988 SPE/DOE Symposium on Enhanced Oil Recovery, Richardson, Texas, Feb 1988. Paper SPE 17376.
- [35] M. Konopczynski and A. Ajayi. Design of intelligent well downhole valves for adjustable flow control. 2004. SPE paper 90664.
- [36] H. M. S. Lababidi, A. A. Garrouch, and M. A. Fahim. A fuzzy heuristic approach for predicting asphaltene precipitation potential. *Energy & Fuels*, 18(1):242–250, 2004.
- [37] K.J. Leontaritis and G.A. Mansoori. Asphaltene flocculation during oil production and processing: A thermodynamic colloidal model. The 1987 SPE International Symposium on Oilfield Chemistry, San Antonio, Texas, Feb 1987. Paper SPE 16258.
- [38] M. Li, P. Guo, and S. Li. Modeling technique of asphaltene precipitation. SPE Permian Basin Oil and Gas Recovery Conference, Midland, Texas, May 2001.
- [39] L. X. Nghiem and D. A. Coombe. Modeling asphaltene precipitation during primary depletion. 1997. SPE paper 36106.
- [40] L. X. Nghiem, M. S. Hassam, and R. Nutakki. Efficient modelling of asphaltene precipitation. The 68th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, Houston, Texas, Oct 1993.

- [41] N. Nor-Azlan and M. A. Adewumi. Development of asphaltene phase equilibrium predictive model. The 1993 SPE Eastern Regional Conference and Exhibition, Pittsburgh, PA, Nov 1993. Paper SPE 26905.
- [42] S.J. Park and G. A. Mansoori. Organic deposition from heavy petroleum crude. Proceedings of the UNITAR/UNDP 4th International Conference on Heavy Crudes and Tar Sands, Aug 1988.
- [43] A. Peneloux, E. Rauzy, and R. Freze. A consistent correction for Redlich-Kwong-Soave volumes. *Journal of Fluid Phase Equilibria*, 8:7–23, 1982.
- [44] D. Y. Peng and D. B. Robinson. A new two-constant equation of state. *Ind. Eng. Chem. Fundam.*, 15(1):59–64, 1976.
- [45] S. Peramanu, C. Singh, M. Agrawala, and H. W. Yarranton. Investigation on the reversibility of asphaltene precipitation. *Energy & Fuels*, 15(4):910–917, 2001.
- [46] T. K. Perkins. Critical and subcritical flow of multiphase mixtures through chokes. *SPE Drilling*, 8:271–276, Dec 1993.
- [47] X. Qin, P. Wang, K. Sepehrnoori, and G. A. Pope. Modeling asphaltene precipitation in reservoir simulation. *Ind. Eng. Chem. Res.*, 39(8):2644–2654, 2000.
- [48] H. H. Rachford and J. D. Rice. Procedure for use of electronic digital computers in calculating flash vaporisation hydrocarbon equilibrium. *Petrol. Technol.*, 4(10), sect. 1, page 19; sect. 2, page 3, 1952.
- [49] O. Redlich and J. N. S. Kwong. On the thermodynamics of solutions. *Chemical Review*, 44:233–244, 1948.
- [50] D. B. Robinson and D. Y. Peng. The characterisation of the heptanes and heavier fractions for the gpa peng-robinson programs. GPA Research Report 28, Tulsa, 1978.
- [51] E. Rogel, O. León, J. Espidel, and J. González. Asphaltene stability in crude oils. 1999. SPE paper 53998.
- [52] A. K. Rydahl, K. S. Pedersen, and H. P. Hjermstad. Modeling of live oil asphaltene precipitation. AIChE Spring National Meeting, Houston, Texas, Mar 1997.
- [53] J. M. Smith and H. C. Van Ness. *Introduction to Chemical Engineering Thermodynamics*. McGraw-Hill, New York, 4th edition, 1987.
- [54] G. Soave. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chem. Eng. Sci.*, 27:1197–1203, 1972.

- [55] G. Soave, M. Barolo, and A. Bertucco. Estimation of high pressure fugacity coefficients of pure gaseous fluids by a modified SRK equation of state. *Journal of Fluid Phase Equilibria*, 91:87–100, 1993.
- [56] C. F. Spencer and R.P. Danner. Prediction of bubble point pressure of mixtures. *Journal of Chem. Eng. Data*, 18(2):230–234, 1973.
- [57] Z. Su and J. S. Gudmundsson. Friction factor of perforation roughness in pipes. 1993. SPE paper 26521.
- [58] F. B Thomas, D. B. Bennion D. W. Bennion, and B. E. Hunter. Experimental and theoretical studies of solids precipitation from reservoir fluid. *J. Can. Petrol. Technol*, 31(1):22–31, Jan 1992.
- [59] M. Vazquez and H. D. Beggs. Correlations for fluid physical property prediction. *Journal of Petroleum Technology*, pages 968–70, Jun 1980.
- [60] A. I. Victorov and N. A. Smirnova. Thermodynamic model of petroleum fluids containing polydisperse asphaltene aggregates. *Ind. Eng. Chem. Res.*, 37(8):3242–3251, 1998.
- [61] J. X. Wang, J. S. Buckley, N. E. Burke, and J. L. Creek. A practical method for anticipating asphaltene problems. *SPE Production & Facilities*, pages 152–160, Aug 2004.
- [62] J.X. Wang and J. S. Buckley. An experimental approach to prediction of asphaltene flocculation. The 2001 SPE International Symposium of Oilfield Chemistry, Houston, Texas, Feb 2001. Paper SPE 64994.

Appendix A: Example Fluid Properties

Table A-1: Static precipitation test results for Oil 1 at 100 °C at different pressures (adapted from Burke et al. [11]).

Pressure (bara)	Precipitates live oil (wt%)	Precipitates remaining (wt%)	Total precipitates (wt%)
69.9611	0.403	15.73	16.133
138.9087	1.037	14.98	16.107
209.2352	0.742	15.06	15.802
276.8038	0.402	14.86	15.262
356.6934	Onset P		

Table A-2: Fluid compositions and basic properties [39].

Component	Mole%	MW
CO_2	2.46	44.01
$N_2 + C_1$	36.94	16.228
C_2	3.47	30.07
C_3	4.05	44.097
C_4	1.93	58.124
C_5	1.57	72.151
$C_6 - C_7$	4.35	92.667
$C_8 - C_{13}$	13.4093	143.321
$C_{14} - C_{19}$	9.4338	227.478
$C_{20} - C_{24}$	5.6864	305.371
$C_{25} - C_{30}$	4.9512	381.763
C_{31A+}	7.6812	665.624
C_{31B+}	4.0681	665.624
C_{7+} MW	329	
C_{7+} SG	0.9594	
Oil MW	171.4	
Sat pressure (bara)	203.39534	

Table A-3: Characterized fluid properties.

Comp	Mol%	MW	T_c	P_c	v_c	ω	1	Cp coefficients			4
								2	3		
CO_2	2.46	44.01	31.05	73.76	94.00	0.23	1.98E+01	7.34E-02	-5.60E-05	1.72E-08	
$N_2 + C_1$	36.94	16.23	96.65	42.46	148.00	0.15	1.96E+01	5.04E-02	1.24E-05	-1.13E-08	
C_2	3.47	30.07	-84.27	45.68	203.00	0.01	5.41E+00	1.78E-01	-6.94E-05	8.71E-09	
C_3	4.05	44.10	32.25	48.84	98.75	0.10	-4.22E+00	3.06E-01	-1.59E-04	3.21E-08	
C_4	1.93	58.12	146.82	37.53	257.45	0.19	6.16E+00	3.48E-01	-1.33E-04	6.89E-09	
C_5	1.57	72.15	192.11	33.79	304.94	0.24	-6.41E+00	4.96E-01	-2.65E-04	5.50E-08	
$C_6 - C_7$	4.35	92.67	262.43	36.94	353.81	0.41	-1.51E+01	5.44E-01	-2.49E-04	2.37E-08	
$C_8 - C_{13}$	13.41	143.32	343.46	28.94	523.32	0.60	-1.86E+01	7.66E-01	-3.21E-04	0.00E+00	
$C_{14} - C_{19}$	9.43	227.48	433.82	20.27	907.81	0.82	-1.65E+01	1.23E+00	-5.18E-04	0.00E+00	
$C_{20} - C_{24}$	5.69	305.37	497.46	17.67	1270.78	0.99	-1.55E+01	1.65E+00	-6.89E-04	0.00E+00	
$C_{25} - C_{30}$	4.95	381.76	554.27	16.40	1656.99	1.14	-1.43E+01	2.07E+00	-8.63E-04	0.00E+00	
C_{31A+}	7.68	665.62	741.58	15.05	3285.15	1.26	-8.49E+00	3.73E+00	-1.54E-03	0.00E+00	
C_{31B+}	4.07	665.62	741.58	15.05	3285.15	1.26	-8.49E+00	3.73E+00	-1.54E-03	0.00E+00	

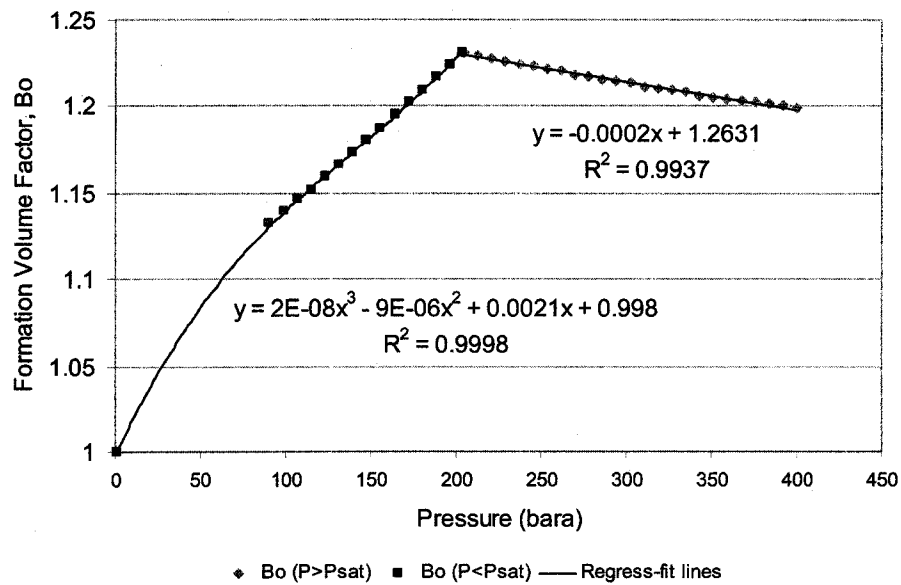


Figure A-1: Oil formation volume factor of example fluid at 100 °C.

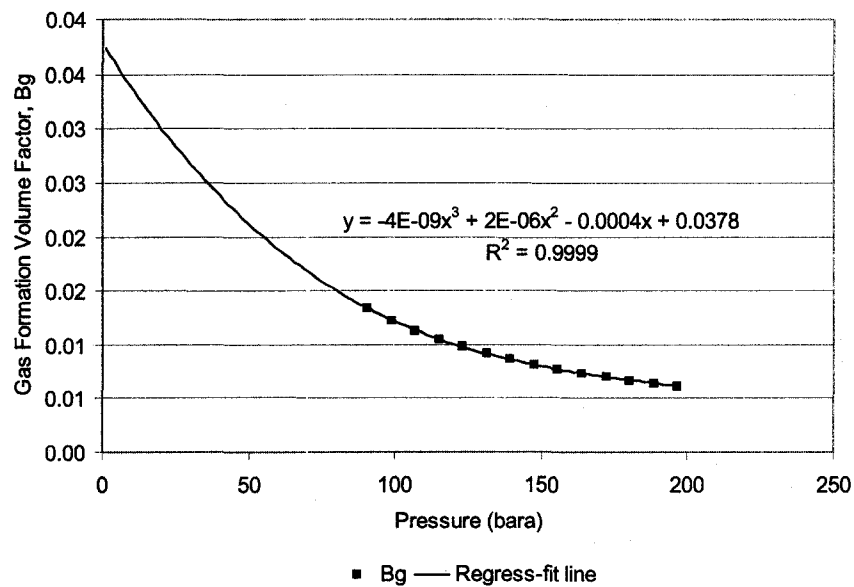


Figure A-2: Gas formation volume factor of example fluid at 100 °C.

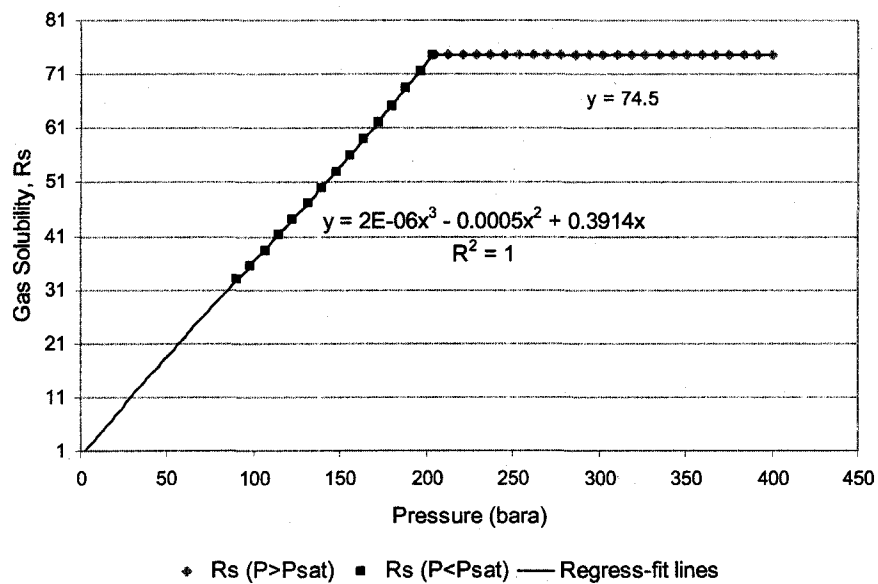


Figure A-3: Gas solubility of example fluid at 100 °C.

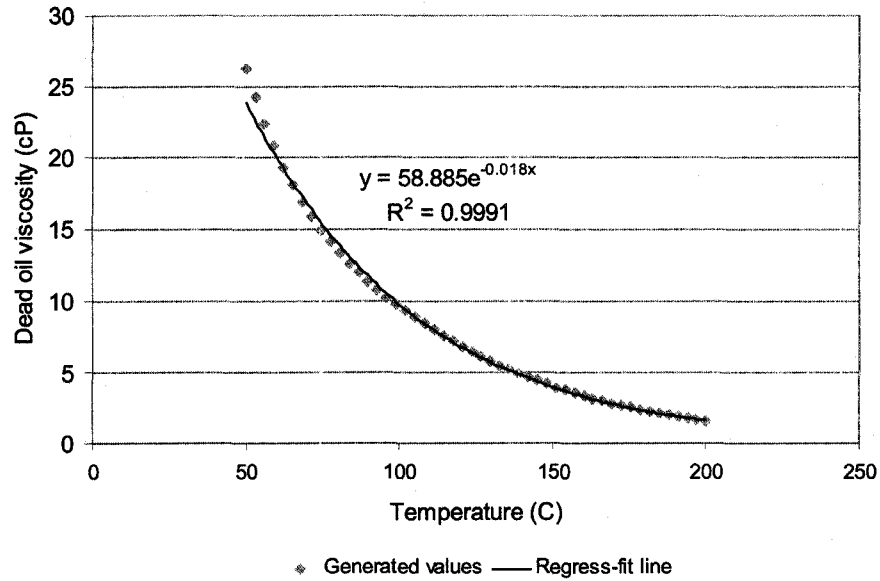


Figure A-4: Dead-oil viscosity of example fluid.

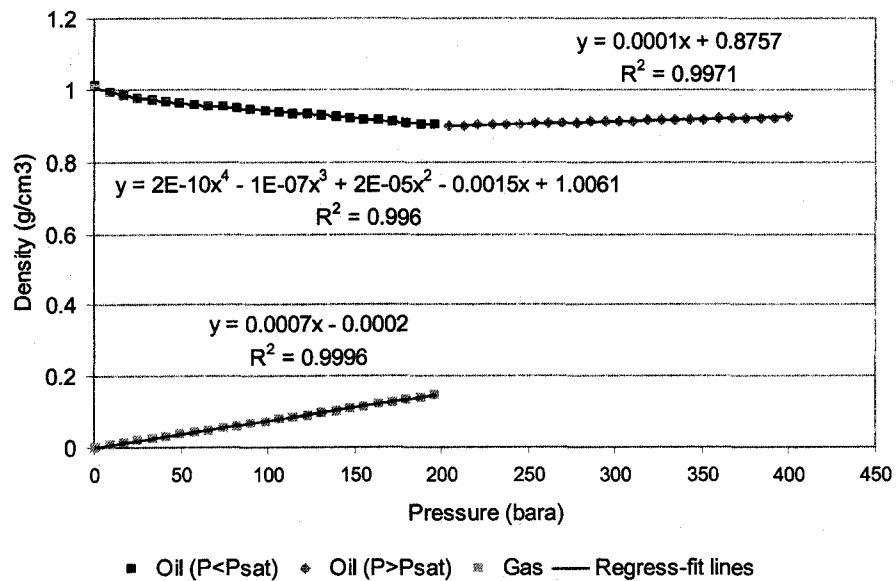


Figure A-5: Density of example fluid at 100 °C.

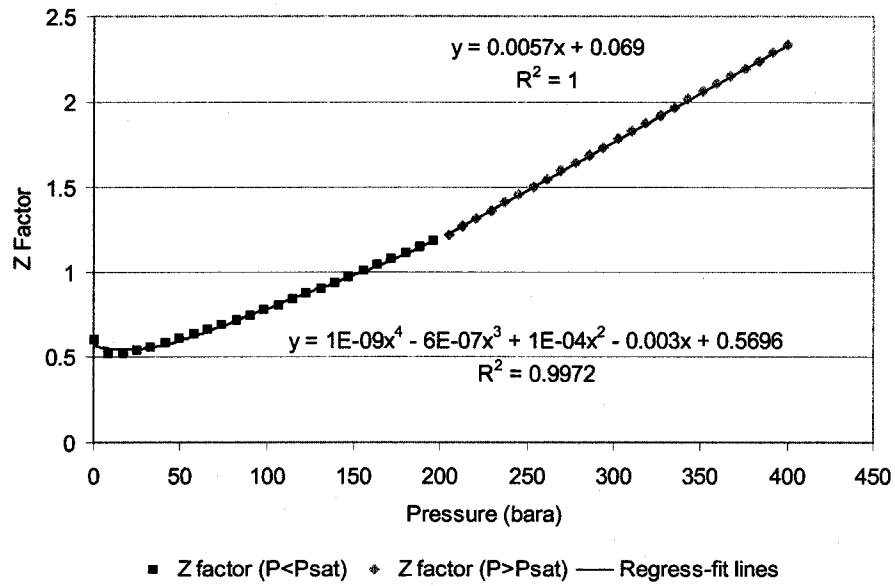


Figure A-6: Compressibility factor of example fluid at 100 °C.

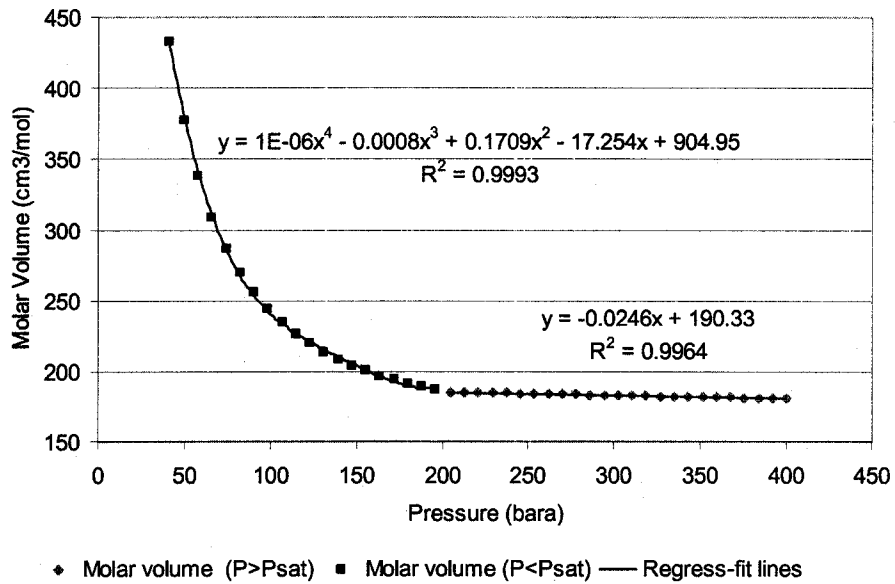


Figure A-7: Molar volume of example fluid at 100 °C.

Appendix B: Matlab Files

Main programme of the network model

```
networkSolver;
TempSolver;
flagmu = true;
muindex = 0;
convergemu = 0;
while (flagmu)
    mu2P_temp = updatemu(XT,X1,Tres,pres,pref,pb,N,num_var,num_varT,...
        Nodes,bridges,Rs,mu2P_res,mu_res);
    % Recalculate pressure/temperature dependent viscosities
    [flagmu epsilonmu] = checkconvmu(mu2P,mu2P_temp,bridges,threshold);
    % Check for convergence
    convergemu(muindex+1) = epsilonmu
    % Assign calculated temperatures for use in recalculating flow parameters
    % in isothermal calculations
    mu2P=mu2P_temp; % Update mu2P
    T_temp(1)=Tres(1); % Generate new T_temp for pressure calculations
    for i=2:num_varT
        T_temp(i)=XT(i-1);
    end
    muindex = muindex+1
    flag = true;
    sentinelCount = 0;
    iteration
    TempSolver
end
% Categorize viscosity values
%displaymu;
% Get flow parameters at the location where asphaltene precipitation is
% suspected to be used in the detailed analysis
location;
```

Isothermal network solver

```
%-----DATA LOADING BEGIN-----
input_data; % Input data file
input_p; % Input reservoir pressures
input_L; % Input segment lengths
input_c; % Input slot/valve discharge coefficients
input_K; % Input absolute permeabilities
input_kro; % Calculate oil relative permeabilities
input_krg; % Calculate gas relative permeabilities
input_s; % Input skin factors
%-----GENERATING RESERVOIR FLUID PROPERTIES BEGIN-----
% Calculate fluid properties at inlet (reservoir) nodes.
% Calculate pressure-dependent black-oil properties at reservoir conditions
[Bo_res Bg_res Rs_res] = generateResprop(N,pres,pb);
% Calculate pressure/temperature dependent oil viscosities
mu_res = generatemures(Tres,pres,pb,N,Rs_res);
% Calculate liquid holdups in the reservoir
input_alpha;
% Calculate pressure-dependent densities for each phase and two-phase (TP)
```

```

% fluid
[rho2P_res rho_res] = generateRhoes(pres,pb,N,alpha_res);
% Calculate two-phase (TP) fluid viscosities
mu2P_res = generatemu2P(mu_res,N,alpha_res,pres,pb);
%-----GENERATING RESERVOIR FLUID PROPERTIES END-----
%-----DATA LOADING END-----
clc;
%-----PRE-CALCULATIONS BEGIN-----
% Precalculate some coefficient values to help increase the calculation
% rate
precalculations;
% Initial guessed values of unknown parameters
guess;
%-----GENERATING WELLBORE FLUID PROPERTIES BEGIN-----
% Calculate fluid properties at all nodes in the well network based on the
% guessed unknown parameters.
% Calculate pressure-dependent gas solubilities
Rs = generateRs(X1,pres,Rs_res,pref,pb,N,num_var,Nodes);
% Calculate pressure-dependent oil formation volume factors
Bo = generateBo(X1,pres,Bo_res,pref,pb,N,num_var,Nodes);
% Calculate pressure-dependent gas formation volume factors
Bg = generateBg(X1,pres,Bg_res,pref,pb,N,num_var,Nodes);
% Calculate pressure/temperature dependent viscosities for each phase and
% two-phase (TP) fluid
[mu2P mu] = generatemu(T_temp,X1,Tres,pref,pb,N,num_var,num_varT,...
    Nodes,bridges,Rs,mu2P_res,mu_res);
% Calculate pressure-dependent densities for each phase and two-phase (TP)
% fluid
[rho2P rho] = generaterho(X1,pres,rho2P_res,rho_res,pref,pb,N,...
    num_var,Nodes,bridges);
%-----GENERATING WELLBORE FLUID PROPERTIES END-----
%-----PRE-CALCULATIONS END-----
%-----ITERATIVE PROCESS BEGIN-----
% SentinelCount counts how many iterations that has been done, so that the
% program stops when the desired number of iterations are reached.
sentinelCount = 0;
% Flag determines whether the iteration should stop (solutions converged)
flag = true;
% Generate indexes for directions of flow through each bridge
generateBindex;
% Iterate function calculations to solve for unknowns using Newton-Raphson
% method
iteration
%-----ITERATIVE PROCESS END-----
% Isothermal network model end

```

Data input file for isothermal network solver

```

% Input data
N          = 200;          % Number of segments
L          = 10;          % Segment length (when all segments are of
                          % the same length or modify input_L.m)
threshold  = 1e-12;        % Tolerance value to check for convergence
stop       = 100;         % Number of iterations before stop
                          % iterating
re         = 20;          % Drainage radius (m)
ri         = 5*0.0254/2;  % Inner radius (m)
ro         = ri+0.02;     % Outer radius (m)

```

```

ri          = ri*ones(1,N);      % Generate ri for all N segments
ro          = ro*ones(1,N-1);    % Generate ro for all N-1 segments
r_temp      = ri;                % Inner radius of annulus (m)
pres        = 365;                % Reservoir pressure (bara) (when all
                                % segments have the same reservoir
                                % pressures or modify input_p.m)
pbh         = 357.3;              % Bottomhole pressure (bara)
pb          = 203.4;              % Bubblepoint pressure (bara)
slot_den    = 30000;              % Slot density (slots/m)
slot_L      = 0.01;              % Slot length (m)
slot_W      = 0.001;              % Slot width (m)
So          = 1*ones(1,N-1);     % Oil saturation (when So's are equal for
                                % all segments)
n_k         = 2*ones(1,N-1);     % Exponent for calculating relative
                                % permeabilities (when n_k's are equal for
                                % all segments)

generateTres;                     % Generate reservoir temperatures
num_var = 9*N-3;                  % Calculate number of unknowns in
                                % isothermal calculations
Nodes = 3*N-1;                   % Calculate number of nodes
bridges = 4*N-2;                  % Calculate number of bridges
num_varT = 2*N;                   % Calculate number of temperature unknowns
                                % in temperature calculations

pbh = pbh*10^5;                   % Change unit of pressure from bara to Pa
generateT;                         % Generate wellbore temperature to be used
                                % in isothermal calculations

% For Case 2-C (well with two temperature zones), different-pressure case,
% ri          = 4*0.0254/2;      % Inner radius (m)
% ro          = ri+0.02;         % Outer radius (m)
% ri          = ri*ones(1,N);    % Generate ri for all N segments
% ro          = ro*ones(1,N-1);  % Generate ro for all N-1 segments
% r_temp      = ri;              % Inner radius of annulus (m)
% for i = 101:N-1
%     r_temp(i)=0;                % Annulus used as tubing in the well
%                               % in Zone 2
% end
% for i = 101:N-1
%     ro(i)    = 5*0.0254/2;      % Outer radius of well in Zone 2
% end
% pres1       = 370;              % Pressure in Zone 1
% pres2       = 360;              % Pressure in Zone 2

% For case 3 (well with multiple inflow control valves),
% N = 100;                % Well divided into 100 segments

% For Case 4 (well with restricted flow in annulus), assign
% r_temp(170) = ri(169)+0.005;

% Generate reservoir temperatures
Tres = 100*ones(1,N);

% For Case 2-A (well with linearly decreasing reservoir temperature),
% Tres(1) = 100;          % Temperature at the toe
% sumL = 0;
% for i = 2:N              % Temperature decreases at the rate 0.01 C/m
%     sumL = sumL+L(i);
%     Tres(i) = 100-sumL*0.01;
% end

% For Case 2-B (well with linearly increasing reservoir temperature),
% Tres(1) = 80;           % Temperature at the toe
% sumL = 0;
% for i = 2:N              % Temperature increases at the rate 0.01 C/m

```



```

%      sumL = sumL+L(i);
%      Tres(i) = 80+sumL*0.01;
% end

% For Case 2-C (well with two temperature zones),
% Tres = 80*ones(1,N);      % Temperatures in Zone 1
%
% for i=101:200              % Temperatures in Zone 2
%     Tres(i) = 100;
% end

% Generate wellbore temperature to be used in isothermal calculations
% Temperatures in annulus and tubing are equal to the reservoir temperature
% of the corresponding segment
T_temp = zeros(1,num_varT);
for i = 0:N-1
    T_temp(i*2+1) = Tres(i+1);      %Annulus
    T_temp(i*2+2) = Tres(i+1);      %Tubing
end

```

Iterative method to solve for unknown parameters in isothermal network solver

```

% Iterate function calculations to solve for unknowns using Newton-Raphson
% method

f1 = zeros(1,9);                % Function matrix for Segment 1
f2 = zeros(1,num_var-(9+6));    % Function matrix for Segment 2 to N-1
f3 = zeros(1,6);                % Function matrix for Segment N

j1 = zeros(9,num_var);          % Jacobian matrix for Segment 1
j2 = zeros(num_var-(9+6),num_var); % Jacobian matrix for Segment 2 to N-1
j3 = zeros(6,num_var);          % Jacobian matrix for Segment N

converge = 0; % Convergence value to be compared with the tolerance value
while(flag)

    % Generating function matrices
    f1 = f1Generator(X1,I,pres,beta,alpha,B,Bo,Bg,Rs,mu2P,rho2P,...
        alpha_res,f1,pref,qref);
    f2 = f2Generator(X1,beta,alpha,B,I,pres,Bo,Bg,Rs,mu2P,rho2P,...
        alpha_res,f2,pref,qref,N,bindex);
    f3 = f3Generator(X1,beta,B,Bo,Bg,Rs,mu2P,rho2P,f3,pref,pbh,...
        N,num_var,Nodes,bridges,bindex);

    f = [f1 f2 f3]; % Combine the matrices

    % Generating jacobian matrices
    j1 = j1Generator(I,X1,beta,alpha,B,j1,Bo,Bg,Rs,rho2P,mu2P,...
        alpha_res,pref,qref);
    j2 = j2Generator(X1,beta,alpha,I,B,j2,Bo,Bg,Rs,rho2P,mu2P,...
        alpha_res,pref,qref,N,bindex);
    j3 = j3Generator(X1,beta,B,j3,Bo,Bg,Rs,rho2P,mu2P,pref,N,...
        num_var,Nodes,bridges,bindex);

    jac = [j1;j2;j3]; % Combine the matrices

    % LU factorization -- Inversion of the jacobian matrix
    [L1 U1] = lu(jac);
    L1_INV = inv(L1);
    U1_INV = inv(U1);

    temp1 = L1_INV*transpose(f);
    temp2 = U1_INV*temp1;
    temp3 = transpose(temp2);

```

```

X2 = X1 - temp3;
% Checking for convergence. If checkConvergence finds that the method
% converges it will set flag to false and the program will stop.
[flag, test] = checkConvergence(X1,X2,num_var,threshold);
% Convergence value of each iteration
converge(sentinelCount+1) = test;
%Setting Xn = Xn+1 for the next iteration
X1 = X2;
%-----RECALCULATE WELLBORE FLUID PROPERTIES BEGIN-----
% Calculate pressure-dependent gas solubilities
Rs = generateRs(X1,pres,Rs_res,pref,pb,N,num_var,Nodes);
% Calculate pressure-dependent oil formation volume factors
Bo = generateBo(X1,pres,Bo_res,pref,pb,N,num_var,Nodes);
% Calculate pressure-dependent gas formation volume factors
Bg = generateBg(X1,pres,Bg_res,pref,pb,N,num_var,Nodes);
% Calculate pressure/temperature dependent viscosities for each phase
% and two-phase (TP) fluid
[mu2P mu] = generatemu(T_temp,X1,Tres,pref,pb,N,num_var,num_varT,...
    Nodes,bridges,Rs,mu2P_res,mu_res);
% Calculate pressure-dependent densities for each phase and two-phase
% (TP) fluid
[rho2P rho] = generaterho(X1,pres,rho2P_res,rho_res,pref,pb,N,...
    num_var,Nodes,bridges);
%-----RECALCULATE WELLBORE FLUID PROPERTIES END-----
% This if statement makes sure that the iteration stops if
% the method does not converge within a number of iteration input by
% the user in input_data.m
if(sentinelCount == stop)
    disp(' ');
    disp('Did not converge within the limitations given!');
    break;
end
% Update iteration index (sentinelCount)
sentinelCount = sentinelCount + 1;
end
% Check for imaginary numbers
for i=1:num_var
    % If the imaginary part of the solution is less than a value,
    % it is negligible
    if imag(X1)<1e-12
        X1=real(X1);
    % If the imaginary part is large, display "imag"
    else
        disp('imag');
    end
end
% Conversions of the converged variables back to their appropriate units
% and result display
displayoutput;

```

Functions to generate governing equations in isothermal network solver

```

% Generate function matrix for Segment 1
%Input:
%
%X1      : Unknown parameters at each iteration
%I       : Pre-calculated coefficient for inflow equations

```

```

%pres      : Reservoir pressures
%beta      : Pre-calculated coefficient for tubing flow calculations
%alpha     : Pre-calculated coefficient for annular flow calculations
%B         : Pre-calculated coefficient for slot/valve flow calculations
%Bo,Bg,Rs  : Black-oil properties
%mu2P      : Two-phase viscosities
%rho2P     : Two-phase densities
%alpha_res : Liquid holdups in reservoir
%f1        : Generated zero function matrix
%pref      : Reference pressure
%qref      : Reference flow rate
%
%Return:
%Function matrix for Segment 1
function func = f1Generator(X1,I,pres,beta,alpha,B,Bo,Bg,Rs,mu2P,...
    rho2P,alpha_res,f1,pref,qref)
% Liquid-phase material balance
f1(1) = X1(4)*X1(8)/Bo(2) - X1(3)*X1(7)/Bo(1); % At node 1
f1(2) = X1(6)*alpha_res(1)/Bo(3)...
    - X1(4)*X1(8)/Bo(2)...
    - X1(5)*X1(9)/Bo(2); % At node 2
% Inflow equation
f1(3) = X1(6)...
    - I(1)*(pres(1)/pref - X1(2))*pref/qref;
% Momentum balance for tubing bridge
f1(4) = X1(1) - X1(10)...
    - beta(1)*(X1(3)^1.75)*rho2P(1)^0.75*mu2P(1)^0.25;
% Flow equation for annular-to-tubing bridge
f1(5) = X1(2) - X1(1)...
    - B(1)*(X1(4)^2)*rho2P(2);
% Momentum balance for annular bridge
f1(6) = X1(2) - X1(11)...
    - alpha(1)*(X1(5)^1.75)*rho2P(3)^0.75*mu2P(3)^0.25;
% Gas-phase material balance
% At node 1
f1(7) = ((1-X1(8))*X1(4)/Bg(2) + X1(8)*Rs(2)*X1(4)/Bo(2))...
    - ((1-X1(7))*X1(3)/Bg(1) + X1(7)*Rs(1)*X1(3)/Bo(1));
% At node 2
f1(8) = ((1-alpha_res(1))*X1(6)/Bg(3) + ...
    alpha_res(1)*Rs(3)*X1(6)/Bo(3)) ...
    - ((1-X1(8))*X1(4)/Bg(2) + X1(8)*Rs(2)*X1(4)/Bo(2)) ...
    - ((1-X1(9))*X1(5)/Bg(2) + X1(9)*Rs(2)*X1(5)/Bo(2));
% Split equation
f1(9) = X1(9) - X1(8);
func = f1;

% Generate function matrix for Segment 2 to N-1
%Input:
%
%X1        : Unknown parameters at each iteration
%beta      : Pre-calculated coefficient for tubing flow calculations
%alpha     : Pre-calculated coefficient for annular flow calculations
%B         : Pre-calculated coefficient for slot/valve flow calculations
%I         : Pre-calculated coefficient for inflow equations
%pres      : Reservoir pressures
%Bo,Bg,Rs  : Black-oil properties
%mu2P      : Two-phase viscosities
%rho2P     : Two-phase densities
%alpha_res : Liquid holdups in reservoir
%f2        : Generated zero function matrix
%pref      : Reference pressure

```

```

%qref      : Reference flow rate
%N         : Number of segments
%b         : Bridge indexes
%
%Return:
%Function matrix for Segment 2 to N-1
function func = f2Generator(X1,beta,alpha,B,I,pres,Bo,Bg,Rs,mu2P,...
    rho2P,alpha_res,f2,pref,qref,N,b)
for i=0:N-3
    var = i*9;
    % Liquid-phase material balance
    % At tubing node
    f2(1+var) = X1(13+var)*X1(17+var)/Bo(3*i+5)*b(4*i+6)...
        + X1(3+var)*X1(7+var)/Bo(3*i+1)*b(4*i+1)...
        - X1(12+var)*X1(16+var)/Bo(3*i+4)*b(4*i+5);
    % At annular node
    f2(2+var) = X1(5+var)*X1(9+var)/Bo(3*i+2)*b(4*i+3)...
        + X1(15+var)*alpha_res(i+2)/Bo(3*i+6)*b(4*i+8)...
        - X1(14+var)*X1(18+var)/Bo(3*i+5)*b(4*i+7)...
        - X1(13+var)*X1(17+var)/Bo(3*i+5)*b(4*i+6);
    % Inflow equation
    % Inlet flow exists
    if b(4*i+8) ~= 0
        f2(3+var) = X1(15+var)...
            - (I(i+2)*(pres(i+2)/pref - X1(11+var)))*pref/qref;
    % No inlet flow
    else
        f2(3+var) = 0;
    end
    % Momentum balance for tubing bridge
    f2(4+var) = X1(10+var) - X1(19+var)...
        - beta(i+2)*(X1(12+var)^1.75)*rho2P(4*i+5)^0.75*...
        mu2P(4*i+5)^0.25;
    % Flow equation for annular-to-tubing bridge
    if b(4*i+6) ~= 0 % Annular-to-tubing flow exists
        f2(5+var) = X1(11+var) - X1(10+var)...
            - B(i+2)*(X1(13+var)^2)*rho2P(4*i+6);
    else % No annular-to-tubing flow
        f2(5+var) = 0;
        f2(9+var) = 0; % No split equation
    end
    % Momentum balance for annular bridge
    if b(4*i+7) ~= 0 % Annular flow exists
        f2(6+var) = X1(11+var) - X1(20+var)...
            - alpha(i+2)*(X1(14+var)^1.75)*rho2P(4*i+7)^0.75*...
            mu2P(4*i+7)^0.25*b(4*i+7);
        if b(4*i+7) == 1
            % If there is both annular and annular-to-tubing flows
            if b(4*i+6) ~= 0
                % Split equation
                f2(9+var) = X1(17+var) - X1(18+var);
            end
            % If flow in annulus is toward toe of well
            elseif b(4*i+7) == -1
                % No split equation
                f2(9+var) = 0;
            end
        % No annular flow
        else
            f2(6+var) = 0;
            f2(9+var) = 0;
        end
    % Gas-phase material balance
    % At tubing node
    f2(7+var) = ((1-X1(17+var))*X1(13+var)/Bg(3*i+5) + X1(17+var)*...
        Rs(3*i+5)*X1(13+var)/Bo(3*i+5))*b(4*i+6) ...
        + ((1-X1(7+var))*X1(3+var)/Bg(3*i+1) + X1(7+var)*...
        Rs(3*i+1)*X1(3+var)/Bo(3*i+1))*b(4*i+1) ...

```

```

- ((1-X1(16+var))*X1(12+var)/Bg(3*i+4) + X1(16+var)*...
  Rs(3*i+4)*X1(12+var)/Bo(3*i+4))*b(4*i+5);
% At annular node
f2(8+var) = ((1-X1(9+var))*X1(5+var)/Bg(3*i+2) + X1(9+var)*...
  Rs(3*i+2)*X1(5+var)/Bo(3*i+2))*b(4*i+3) ...
+ ((1-alpha_res(i+2))*X1(15+var)/Bg(3*i+6) +
  alpha_res(i+2)*...
  Rs(3*i+6)*X1(15+var)/Bo(3*i+6))*b(4*i+8)...
- ((1-X1(18+var))*X1(14+var)/Bg(3*i+5) + X1(18+var)*...
  Rs(3*i+5)*X1(14+var)/Bo(3*i+5))*b(4*i+7)...
- ((1-X1(17+var))*X1(13+var)/Bg(3*i+5) + X1(17+var)*...
  Rs(3*i+5)*X1(13+var)/Bo(3*i+5))*b(4*i+6);

end
func = f2;

% Generate function matrix for Segment N
%Input:
%
%X1      : Unknown parameters at each iteration
%beta    : Pre-calculated coefficient for tubing flow calculations
%B       : Pre-calculated coefficient for slot/valve flow calculations
%Bo,Bg,Rs : Black-oil properties
%mu2P    : Two-phase viscosities
%rho2P   : Two-phase densities
%f3      : Generated zero function matrix
%pref    : Reference pressure
%pbh     : Bottomhole pressure
%N       : Number of segments
%num_var : Number of unknowns
%Nodes   : Number of nodes
%bridges : Number of bridges
%b       : Bridge indexes
%
%Return:
%Function matrix for Segment N
function func = f3Generator(X1,beta,B,Bo,Bg,Rs,mu2P,rho2P,f3,pref,pbh,...
  N,num_var,Nodes,bridges,b)
% Liquid-phase material balance
% At tubing node
f3(1) = X1(num_var-12)*X1(num_var-8)/Bo(Nodes-4)*b(bridges-5)...
  + X1(num_var-2)*X1(num_var)/Bo(Nodes)*b(bridges)...
  - X1(num_var-3)*X1(num_var-1)/Bo(Nodes-1)*b(bridges-1);
% At annular node
f3(2) = X1(num_var-10)*X1(num_var-6)/Bo(Nodes-3)*b(bridges-3)...
  - X1(num_var-2)*X1(num_var)/Bo(Nodes)*b(bridges);
% Momentum balance for tubing bridge
f3(3) = X1(num_var-5) - pbh/pref...
  - beta(N)*(X1(num_var-3)^1.75)*rho2P(bridges-1)^0.75*...
  mu2P(bridges-1)^0.25;
% Flow equation for annular-to-tubing bridge
% Annular-to-tubing flow exists
if b(bridges) ~= 0
  f3(4) = X1(num_var-4) - X1(num_var-5) - B(N)*(X1(num_var-2)^2)*...
    rho2P(bridges);
% No annular-to-tubing flow
else
  f3(4) = 0;
end
% Gas-phase material balance
% At tubing node
f3(5) = ((1-X1(num_var-8))*X1(num_var-12)/Bg(Nodes-4) + ...
  X1(num_var-8)*Rs(Nodes-4)*X1(num_var-12)/Bo(Nodes-4))*...

```

```

        b(bridges-5) ...
    + ((1-X1(num_var))*X1(num_var-2)/Bg(Nodes) + ...
      X1(num_var)*Rs(Nodes)*X1(num_var-2)/Bo(Nodes))...
      *b(bridges)...
    - ((1-X1(num_var-1))*X1(num_var-3)/Bg(Nodes-1) + ...
      X1(num_var-1)*Rs(Nodes-1)*X1(num_var-3)/Bo(Nodes-1))*...
      b(bridges-1);
    % At annular node
    f3(6) = ((1-X1(num_var-6))*X1(num_var-10)/Bg(Nodes-3) + ...
      X1(num_var-6)*Rs(Nodes-3)*X1(num_var-10)/Bo(Nodes-3))*...
      b(bridges-3) ...
    - ((1-X1(num_var))*X1(num_var-2)/Bg(Nodes) + ...
      X1(num_var)*Rs(Nodes)*X1(num_var-2)/Bo(Nodes))*b(bridges);

func = f3;

```

Functions to generate jacobian matrices in isothermal network solver

```

% Generate jacobian matrix for Segment 1
%Input:
%
%I      : Pre-calculated coefficient for inflow equations
%X1     : Unknown parameters at each iteration
%beta   : Pre-calculated coefficient for tubing flow calculations
%alpha  : Pre-calculated coefficient for annular flow calculations
%B      : Pre-calculated coefficient for slot/valve flow calculations
%j1     : Generated zero jacobian matrix
%Bo,Bg,Rs : Black-oil properties
%mu2P   : Two-phase viscosities
%rho2P  : Two-phase densities
%alpha_res : Liquid holdups in reservoir
%pref   : Reference pressure
%qref   : Reference flow rate
%
%Return:
%Jacobian matrix for Segment 1
function func = j1Generator(I,X1,beta,alpha,B,j1,Bo,Bg,Rs,rho2P,...
    mu2P,alpha_res,pref,qref)
j1(1,3) = -X1(7)/Bo(1);
j1(1,4) = X1(8)/Bo(2);
j1(1,7) = -X1(3)/Bo(1);
j1(1,8) = X1(4)/Bo(2);
j1(2,4) = -X1(8)/Bo(2);
j1(2,5) = -X1(9)/Bo(2);
j1(2,6) = alpha_res(1)/Bo(3);
j1(2,8) = -X1(4)/Bo(2);
j1(2,9) = -X1(5)/Bo(2);
j1(3,2) = I(1)*(pref/qref);
j1(3,6) = 1;
j1(4,1) = 1;
j1(4,3) = -1.75*beta(1)*(X1(3)^0.75)*rho2P(1)^0.75*mu2P(1)^0.25;
j1(4,10) = -1;
j1(5,1) = -1;
j1(5,2) = 1;
j1(5,4) = -2*B(1)*X1(4)*rho2P(2);
j1(6,2) = 1;
j1(6,5) = -1.75*alpha(1)*(X1(5)^0.75)*rho2P(3)^0.75*mu2P(3)^0.25;
j1(6,11) = -1;
j1(7,3) = -((1-X1(7))/Bg(1) + X1(7)*Rs(1)/Bo(1));

```

```

j1(7,4) = ((1-X1(8))/Bg(2) + X1(8)*Rs(2)/Bo(2));
j1(7,7) = -(-X1(3)/Bg(1) + Rs(1)*X1(3)/Bo(1));
j1(7,8) = (-X1(4)/Bg(2) + Rs(2)*X1(4)/Bo(2));
j1(8,4) = -((1-X1(8))/Bg(2) + X1(8)*Rs(2)/Bo(2));
j1(8,5) = -((1-X1(9))/Bg(2) + X1(9)*Rs(2)/Bo(2));
j1(8,6) = ((1-alpha_res(1))/Bg(3) + alpha_res(1)*Rs(3)/Bo(3));
j1(8,8) = -(-X1(4)/Bg(2) + Rs(2)*X1(4)/Bo(2));
j1(8,9) = -(-X1(5)/Bg(2) + Rs(2)*X1(5)/Bo(2));
j1(9,8) = -1;
j1(9,9) = 1;
func = j1;

% Generate jacobian matrix for Segment 2 to N-1
%Input:
%
%X1      : Unknown parameters at each iteration
%beta    : Pre-calculated coefficient for tubing flow calculations
%alpha   : Pre-calculated coefficient for annular flow calculations
%I       : Pre-calculated coefficient for inflow equations
%B       : Pre-calculated coefficient for slot/valve flow calculations
%j2      : Generated zero jacobian matrix
%Bo,Bg,Rs : Black-oil properties
%mu2P    : Two-phase viscosities
%rho2P   : Two-phase densities
%alpha_res : Liquid holdups in reservoir
%pref    : Reference pressure
%qref    : Reference flow rate
%N       : Number of segments
%b       : Bridge indexes
%
%Return:
%Jacobian matrix for Segment 2 to N-1
function func = j2Generator(X1,beta,alpha,I,B,j2,Bo,Bg,Rs,rho2P,mu2P,...
    alpha_res,pref,qref,N,b)
for i=0:N-3
    var = 9*i;
    j2(1+var,3+var) = X1(7+var)/Bo(3*i+1)*b(4*i+1);
    j2(1+var,7+var) = X1(3+var)/Bo(3*i+1)*b(4*i+1);
    j2(1+var,12+var) = -X1(16+var)/Bo(3*i+4)*b(4*i+5);
    j2(1+var,13+var) = X1(17+var)/Bo(3*i+5)*b(4*i+6);
    j2(1+var,16+var) = -X1(12+var)/Bo(3*i+4)*b(4*i+5);
    j2(1+var,17+var) = X1(13+var)/Bo(3*i+5)*b(4*i+6);
    j2(2+var,5+var) = X1(9+var)/Bo(3*i+2)*b(4*i+3);
    j2(2+var,9+var) = X1(5+var)/Bo(3*i+2)*b(4*i+3);
    j2(2+var,13+var) = -X1(17+var)/Bo(3*i+5)*b(4*i+6);
    j2(2+var,14+var) = -X1(18+var)/Bo(3*i+5)*b(4*i+7);
    j2(2+var,15+var) = alpha_res(i+2)/Bo(3*i+6)*b(4*i+8);
    j2(2+var,17+var) = -X1(13+var)/Bo(3*i+5)*b(4*i+6);
    j2(2+var,18+var) = -X1(14+var)/Bo(3*i+5)*b(4*i+7);
    j2(7+var,3+var) = ((1-X1(7+var))/Bg(3*i+1) + X1(7+var)*Rs(3*i+1)/...
        Bo(3*i+1))*b(4*i+1);
    j2(7+var,7+var) = (-X1(3+var)/Bg(3*i+1) + Rs(3*i+1)*X1(3+var)/...
        Bo(3*i+1))*b(4*i+1);
    j2(7+var,12+var) = +((1-X1(16+var))/Bg(3*i+4) + X1(16+var)*...
        Rs(3*i+4)/Bo(3*i+4))*b(4*i+5);
    j2(7+var,13+var) = ((1-X1(17+var))/Bg(3*i+5) + X1(17+var)*...
        Rs(3*i+5)/Bo(3*i+5))*b(4*i+6);
    j2(7+var,16+var) = -(-X1(12+var)/Bg(3*i+4) + Rs(3*i+4)*X1(12+var)/...

```

```

        Bo(3*i+4))*b(4*i+5);
j2(7+var,17+var) = (-X1(13+var)/Bg(3*i+5) + Rs(3*i+5)*X1(13+var)/...
        Bo(3*i+5))*b(4*i+6);
j2(8+var,5+var) = ((1-X1(9+var))/Bg(3*i+2) + X1(9+var)*Rs(3*i+2)/...
        Bo(3*i+2))*b(4*i+3);
j2(8+var,9+var) = (-X1(5+var)/Bg(3*i+2) + Rs(3*i+2)*X1(5+var)/...
        Bo(3*i+2))*b(4*i+3);
j2(8+var,13+var) = -((1-X1(17+var))/Bg(3*i+5) + X1(17+var)*...
        Rs(3*i+5)/Bo(3*i+5))*b(4*i+6);
j2(8+var,14+var) = -((1-X1(18+var))/Bg(3*i+5) + X1(18+var)*...
        Rs(3*i+5)/Bo(3*i+5))*b(4*i+7);
j2(8+var,15+var) = ((1-alpha_res(i+2))/Bg(3*i+6) + alpha_res(i+2)*...
        Rs(3*i+6)/Bo(3*i+6))*b(4*i+8);
j2(8+var,17+var) = -(-X1(13+var)/Bg(3*i+5) + Rs(3*i+5)*X1(13+var)/...
        Bo(3*i+5))*b(4*i+6);
j2(8+var,18+var) = -(-X1(14+var)/Bg(3*i+5) + Rs(3*i+5)*X1(14+var)/...
        Bo(3*i+5))*b(4*i+7);

if b(4*i+8) ~= 0 % There is inflow equation
    j2(3+var,11+var) = I(i+2)*pref/qref;
    j2(3+var,15+var) = 1;
else % There is no inflow equation
    j2(3+var,15+var) = 1;
end

j2(4+var,10+var) = 1;
j2(4+var,12+var) = -1.75*beta(i+2)*(X1(12+var)^(0.75))*...
    rho2P(4*i+5)^0.75*mu2P(4*i+5)^0.25;
j2(4+var,19+var) = -1;

if b(4*i+6) ~= 0 % There is annular-to-tubing flow equation
    j2(5+var,11+var) = 1;
    j2(5+var,10+var) = -1;
    j2(5+var,13+var) = -2*B(i+2)*X1(13+var)*rho2P(4*i+6);
else % There is no annular-to-tubing flow equation
    j2(5+var,13+var) = 1;
    j2(9+var,17+var) = 1;
    j2(9+var,18+var) = 1;
end

if b(4*i+7) ~= 0 % There is annular flow equation
    j2(6+var,11+var) = 1;
    j2(6+var,14+var) = -alpha(i+2)*1.75*(X1(14+var)^0.75)*...
        rho2P(4*i+7)^0.75*mu2P(4*i+7)^0.25;
    j2(6+var,20+var) = -1;
    % There is tubing flow equation -- there is split equation
    if b(4*i+6) ~= 0
        j2(9+var,17+var) = 1;
        j2(9+var,18+var) = -1;
    end
    % If flow in annulus is toward toe of well
    if b(4*i+7) == -1
        j2(6+var,11+var) = -1;
        j2(6+var,20+var) = 1;
    end
end

% There is no annular flow equation
elseif b(4*i+7) == 0
    % The value "1" does not affect the results
    j2(6+var,14+var) = 1;
    j2(9+var,17+var) = 1;
    j2(9+var,18+var) = 1;
    if b(4*i+6) == 0
        % To avoid singularity
        j2(6+var,17+var) = 1e-20;
        j2(2+var,14+var) = 1e-20;
        j2(2+var,17+var) = 1e-20;
        j2(8+var,14+var) = 1e-20;
    end
end

```



```

        j2(8+var,17+var) = 1e-20;
    end
end
end
func = j2;

```

```

% Generate jacobian matrix for Segment N

```

```

%Input:

```

```

%
%X1      :   Unknown parameters at each iteration
%beta    :   Pre-calculated coefficient for tubing flow calculations
%B       :   Pre-calculated coefficient for slot/valve flow calculations
%j3      :   Generated zero jacobian matrix
%Bo,Bg,Rs :   Black-oil properties
%mu2P    :   Two-phase viscosities
%rho2P   :   Two-phase densities
%pref    :   Reference pressure
%N       :   Number of segments
%num_var  :   Number of unknowns
%Nodes   :   Number of nodes
%bridges  :   Number of bridges
%b       :   Bridge indexes
%

```

```

%Return:

```

```

%Jacobian matrix for Segment N

```

```

function func = j3Generator(X1,beta,B,j3,Bo,Bg,Rs,rho2P,mu2P,pref,...
    N,num_var,Nodes,bridges,b)

```

```

j3(1,num_var-12) = X1(num_var-8)/Bo(Nodes-4)*b(bridges-5);
j3(1,num_var-8) = X1(num_var-12)/Bo(Nodes-4)*b(bridges-5);
j3(1,num_var-3) = -X1(num_var-1)/Bo(Nodes-1)*b(bridges-1);
j3(1,num_var-2) = X1(num_var)/Bo(Nodes)*b(bridges);
j3(1,num_var-1) = -X1(num_var-3)/Bo(Nodes-1)*b(bridges-1);
j3(1,num_var) = X1(num_var-2)/Bo(Nodes)*b(bridges);

j3(2,num_var-10) = X1(num_var-6)/Bo(Nodes-3)*b(bridges-3);
j3(2,num_var-6) = X1(num_var-10)/Bo(Nodes-3)*b(bridges-3);
j3(2,num_var-2) = -X1(num_var)/Bo(Nodes)*b(bridges);
j3(2,num_var) = -X1(num_var-2)/Bo(Nodes)*b(bridges);

j3(5,num_var-12) = ((1-X1(num_var-8))/Bg(Nodes-4) + ...
    X1(num_var-8)*Rs(Nodes-4)/Bo(Nodes-4))*b(bridges-5);
j3(5,num_var-8) = (-X1(num_var-12)/Bg(Nodes-4) + ...
    Rs(Nodes-4)*X1(num_var-12)/Bo(Nodes-4))*b(bridges-5);
j3(5,num_var-3) = -((1-X1(num_var-1))/Bg(Nodes-1) + ...
    X1(num_var-1)*Rs(Nodes-1)/Bo(Nodes-1))*b(bridges-1);
j3(5,num_var-2) = ((1-X1(num_var))/Bg(Nodes) + ...
    X1(num_var)*Rs(Nodes)/Bo(Nodes))*b(bridges);
j3(5,num_var-1) = -(-X1(num_var-3)/Bg(Nodes-1) + ...
    Rs(Nodes-1)*X1(num_var-3)/Bo(Nodes-1))*b(bridges-1);
j3(5,num_var) = (-X1(num_var-2)/Bg(Nodes) + ...
    Rs(Nodes)*X1(num_var-2)/Bo(Nodes))*b(bridges);

j3(6,num_var-10) = ((1-X1(num_var-6))/Bg(Nodes-3) + ...
    X1(num_var-6)*Rs(Nodes-3)/Bo(Nodes-3))*b(bridges-3);
j3(6,num_var-6) = (-X1(num_var-10)/Bg(Nodes-3) + ...
    Rs(Nodes-3)*X1(num_var-10)/Bo(Nodes-3))*b(bridges-3);
j3(6,num_var-2) = -((1-X1(num_var))/Bg(Nodes) + ...
    X1(num_var)*Rs(Nodes)/Bo(Nodes))*b(bridges);
j3(6,num_var) = -(-X1(num_var-2)/Bg(Nodes) + ...
    Rs(Nodes)*X1(num_var-2)/Bo(Nodes))*b(bridges);

j3(3,num_var-5) = 1;
j3(3,num_var-3) = -1.75*beta(N)*(X1(num_var-3)^0.75)*...

```

```

        rho2P(bridges-1)^0.75*mu2P(bridges-1)^0.25;
% There is annular-to-tubing flow equation
if b(bridges) ~= 0
    j3(4,num_var-5) = -1;
    j3(4,num_var-4) = 1;
    j3(4,num_var-2) = -2*B(N)*X1(num_var-2)*rho2P(bridges);
% There is no annular-to-tubing flow equation
else
    % The value "1" does not affect the results
    j3(4,num_var-5) = 1;
    j3(4,num_var-4) = 1;
    j3(4,num_var-2) = 1;
end
func = j3;

```

Function to check for convergence in isothermal network solver

```

% Checks for convergence
%Input:
%
%X1      :   Unknown parameters at nth iteration
%X2      :   Unknown parameters at n+1th iteration
%N       :   Number of unknowns
%threshold :   Tolerance value for checking for convergence
%
%Return:
%flag     :   Convergence status (false = convergence)
%epsilon  :   Convergence value
function [func, func1] = checkConvergence(X1,X2,N,threshold)
flag = true;
epsilon = 0;
temp = 0;
for i=1:N
    if X1(i)~=0
        temp = abs(X1(i)-X2(i))/(X1(i)*N);
    else
        if X2==0
            temp=0;
        else
            X1(i) = 1e-20;
        end
    end
    epsilon = epsilon + temp;
end
if(epsilon < threshold)
    flag = false;
end
func = flag;
func1 = epsilon;

```

Network temperature solver

```

%-----DATA LOADING BEGIN-----
input_dataT;           % Input data file for temperature calculations
generatepdrop;         % Generate pressure drop between nodes
%-----DATA LOADING END-----

```

```

clc;
%-----PRE-CALCULATIONS BEGIN-----
XT_temp = zeros(1,num_varT);           % Initial guessed unknown temperatures
XT_temp = guessGeneratorT(Tres,L,Tref,Tbh,N);
XT = XT_temp;
%-----PRE-CALCULATIONS END-----
%-----ITERATIVE PROCESS BEGIN-----
% SentinelCount counts how many iterations that has been done, so that the
% program stops when the desired number of iterations are reached.
sentinelCount = 0;
% Flag determines whether the iteration should stop (solutions converged)
flagT = true;
% Iterate function calculations to solve for temperature unknowns using
% Newton-Raphson method
iterationT
%-----ITERATIVE PROCESS END-----
% Network model for temperature calculations end

```

Data input file for temperature network solver

```

% Input data file for temperature calculations
Tref      = 1;           % Reference temperature
Tbh       = 100.5;       % Guessed bottomhole temperature
MWg       = 0.1;         % Input molecular weight of the gas phase
MWO       = 171.35;      % Input molecular weight of the oil phase
Co        = 368.93/(MWO*1e-3); % Heat capacity of the oil phase (J/kg C)
Cg        = 30/(MWg*1e-3); % Heat capacity of the gas phase (J/kg C)
KJT       = -0.0321;     % Joule-Thompson coefficient
h_fl      = 0.269;       % Thermal conductivity of fluid
x_cem     = 0.05;        % Cement thickness
x_case    = 0.03;        % Casing thickness
x_tube    = 0.01;        % Tubing wall thickness
gam_ann   = 0.8;         % Non-perforated fraction of outer
                        % surface area of the annulus
gam_tube  = 0.4;         % Impermeable fraction of tubing wall
                        % surface area
k_cem     = 7.0337;      % Thermal conductivity of cement
k_case    = 12.1912;     % Thermal conductivity of casing
k_tube    = 50;          % Thermal conductivity of tubing wall
generateKappa;           % Generate overall heat transfer
                        % coefficients
rhoo_ST   = 954.7;       % Oil density at standard conditions
                        % (kg/m3)
rhog_ST   = 1;           % Gas density at standard conditions
                        % (kg/m3)
Doil      = Co*rhoo_ST*Tref; % Precalculated coefficient for the oil
                        % phase
Dgas      = Cg*rhog_ST*Tref; % Precalculated coefficient for the gas
                        % phase

% Generate overall heat transfer coefficients
% Calculate dimensionless variables
for i=1:N-1               % Segment 1 to N-1
% Reynolds numbers of fluid in tubing
Re_No(i) = tubingFlowrates(i)/60/60/24/(pi*ri(i)^2)*(2*ri(i))*...

```

```

    rho2P((i-1)*4+2)/mu2P((i-1)*4+2);
% Reynolds numbers of fluid in annulus
Re_ann(i) = annularFlowrates(i)/60/60/24/(pi*ro(i)^2-pi*ri(i)^2)*...
    (4*(pi*ro(i)^2-pi*ri(i)^2)/(2*pi*ro(i)+2*pi*ri(i)))*...
    rho2P((i-1)*4+2)/mu2P((i-1)*4+2);
% Prandtl numbers
Pr_No(i) = Co*mu2P((i-1)*4+2)/h_fl;
end
% Segment N
Re_No(N) = tubingFlowrates(N)/60/60/24/(pi*ri(N)^2*(2*ri(N))*...
    rho2P((N-1)*4+2)/mu2P((N-1)*4+2);
Pr_No(N) = Co*mu2P((N-1)*4+2)/h_fl;
% Calculate overall heat transfer coefficients
for i=1:N-1 % For Segment 1 to N-1
    if Re_ann(i) < 3000 % For laminar flows
        % Heat transfer coefficient of fluid in annulus
        h_ann(i) = 3.656*h_fl/(ro(i)-ri(i));
    else % For turbulence flows
        h_ann(i) = 0.023*Re_ann(i)^0.8*Pr_No(i)^0.33*h_fl/(ro(i)-ri(i));
    end
    % Overall heat transfer coefficient of fluid in annulus
    U_ann(i) = (x_cem/k_cem+x_case/k_case+1/h_ann(i))^( -1);
    if Re_No(i) < 3000 % For laminar flows
        % Heat transfer coefficient of fluid in tubing
        h_fluid(i) = 3.656*h_fl/2/ri(i);
    else % For turbulence flows
        h_fluid(i) = 0.023*Re_No(i)^0.8*Pr_No(i)^0.33*h_fl/2/ri(i);
    end
    % Overall heat transfer coefficient of fluid in tubing
    U_tube(i) = (x_tube/k_tube+1/h_fluid(i))^( -1);
% Calculate coefficients over the area for radial heat transfer
if bindex(4*(i-1)+2)~=0 % Flow through slots
    % Heat transfer coefficient of fluid in tubing
    Kappa_t(i) = gam_tube*U_tube(i)*2*pi*ri(i)*L(i)*Tref;
else % No slots -- no annular-to-tubing flow
    Kappa_t(i) = 1*U_tube(i)*2*pi*ri(i)*L(i)*Tref;
end
% Heat transfer coefficient of fluid in annulus
Kappa_a(i) = gam_ann*U_ann(i)*2*pi*ro(i)*L(i)*Tref;
end
% For Segment N
% Heat transfer coefficient of fluid in tubing
h_fluid(N) = 0.023*Re_No(N)^0.8*Pr_No(i)^0.33*h_fl/2/ri(N);
U_tube(N) = (x_tube/k_tube+1/h_fluid(N))^( -1);
Kappa_t(N) = U_tube(N)*2*pi*ri(N)*L(i)*Tref;
% To exclude heat transfer between fluid and surroundings, type:
% Kappa_a = Kappa_a*0;
% Kappa_t = Kappa_t*0;

```

Iterative method to solve for unknown parameters in network temperature solver

```

% Iterate function calculations to solve for temperature unknowns using
% Newton-Raphson method
f1 = zeros(1,3); % Function matrix for Segment 1
f2 = zeros(1,num_varT-(4)); % Function matrix for Segment 2 to N-1
f3 = zeros(1,1); % Function matrix for Segment N
j1 = zeros(3,num_varT); % Jacobian matrix for Segment 1
j2 = zeros(num_varT-(4),num_varT); % Jacobian matrix for Segment 2 to N-1
j3 = zeros(1,num_varT); % Jacobian matrix for Segment N

```

```

% Generate flow rates for use in temperature calculations and convert from
% m^3/d to m^3/s
generateFlow;
% Generate liquid holdups for use in temperature calculations
generateFractions;
% Convergence value to be compared with the tolerance value
converge = 0;
while(flagT)

    % Generating function matrices
    f1 = f1T(Doil,Dgas,Tres,Tref,XT,q,Lfrac,Bo,Bg,Rs,Kappa_t,Kappa_a,...
        f1,deltaP_a,deltaP_t,KJT);
    f2 = f2T(Doil,Dgas,Tres,Tref,XT,q,Lfrac,Bo,Bg,Rs,N,Kappa_t,Kappa_a,...
        f2,bindex,deltaP_a,deltaP_t,KJT);
    f3 = f3T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,N,f3,num_varT,Nodes,bridges,...
        Kappa_t,Kappa_a,bindex,deltaP_a,deltaP_t,KJT);
    f = [f1 f2 f3]; % Combine the matrices

    % Generating jacobian matrices
    j1 = j1T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,Kappa_t,Kappa_a,j1);
    j2 = j2T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,N,Kappa_t,Kappa_a,j2,bindex);
    j3 = j3T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,N,j3,num_varT,Nodes,bridges,...
        Kappa_t,Kappa_a,bindex);
    jac = [j1;j2;j3]; % Combine the matrices

    % LU factorization -- Inversion of the jacobian matrix
    [L1 U1] = lu(jac);
    L1_INV = inv(L1);
    U1_INV = inv(U1);
    temp1 = L1_INV*transpose(f);
    temp2 = U1_INV*temp1;
    temp3 = transpose(temp2);
    XT1_temp = XT;
    XT2 = XT - temp3;

    % Checking for convergence. If checkConvergence finds that the method
    % converges it will set flag to false and the program will stop.
    [flagT, test] = checkConvT(XT,XT2,num_varT,threshold);
    % Convergence value of each iteration
    converge(sentinelCount+1) = test
    %Setting Xn = Xn+1 for the next iteration
    XT = XT2;

    % This if statement makes sure that the iteration stops if
    % the method does not converge within a number of iteration input
    % by the user in input_data.m
    if(sentinelCount == stop)
        disp(' ');
        disp('Did not converge within the limitations given!');
        break;
    end

    % Update iteration index (sentinelCount)
    sentinelCount = sentinelCount + 1;
end

% Categorize converged temperature variables and
% result display
displayoutputT;

```

Functions to generate governing equations in network temperature solver

```

% Generate function matrix for Segment 1
%Input:
%
%Doil, Dgas :   Precalculated coefficients
%Tres       :   Reservoir temperatures
%Tref       :   Reference temperature
%XT         :   Unknown temperatures at each iteration
%q          :   Flow rates
%Lfrac      :   Liquid holdups
%Bo,Bg,Rs   :   Black-oil properties
%Kappa_t    :   Overall heat transfer coefficients for fluid in tubing
%Kappa_a    :   Overall heat transfer coefficients for fluid in annulus
%f1         :   Generated zero function matrix
%deltaP_t   :   Pressure drop between nodes for fluid in tubing
%deltaP_a   :   Pressure drop between nodes for fluid in annulus
%KJT        :   Joule-Thompson coefficient
%
%Return:
%Function matrix for Segment 1
function func = f1T(Doil,Dgas,Tres,Tref,XT,q,Lfrac,Bo,Bg,Rs,Kappa_t,...
    Kappa_a,f1,deltaP_a,deltaP_t,KJT)
Ttoe = Tres(1);
% Assign temperature in tubing of Segment one equal reservoir temperature
f1(1) = XT(1) - Ttoe/Tref;
% Energy balance at tubing node
f1(2) = -(Doil*q(2)*Lfrac(2)/Bo(2)...
    + Dgas*(q(2)*(1-Lfrac(2))/Bg(2)...
    + q(2)*Lfrac(2)*Rs(2)/Bo(2))*((XT(1)-Ttoe/Tref))...
    - (Doil*q(1)*Lfrac(1)/Bo(1)...
    + Dgas*(q(1)*(1-Lfrac(1))/Bg(1)...
    + q(1)*Lfrac(1)*Rs(1)/Bo(1))*((XT(3)-XT(1))...
    - KJT*deltaP_t(1))...
    - Kappa_t(1)*(XT(1)-Ttoe/Tref);
% Energy balance at annular node
f1(3) = - (Doil*q(3)*Lfrac(3)/Bo(2)...
    + Dgas*(q(3)*(1-Lfrac(3))/Bg(2)...
    + q(3)*Lfrac(3)*Rs(2)/Bo(2))*((XT(2)-Ttoe/Tref)-...
    KJT*deltaP_a(1));
func = f1;

% Generate function matrix for Segment 2 to N-1
%Input:
%
%Doil, Dgas :   Precalculated coefficients
%Tres       :   Reservoir temperatures
%Tref       :   Reference temperature
%XT         :   Unknown temperatures at each iteration
%q          :   Flow rates
%Lfrac      :   Liquid holdups
%Bo,Bg,Rs   :   Black-oil properties
%N          :   Number of segments
%Kappa_t    :   Overall heat transfer coefficients for fluid in tubing
%Kappa_a    :   Overall heat transfer coefficients for fluid in annulus
%f2         :   Generated zero function matrix
%b          :   Bridge indexes

```

```

%deltaP_t : Pressure drop between nodes for fluid in tubing
%deltaP_a : Pressure drop between nodes for fluid in annulus
%KJT : Joule-Thompson coefficient
%
%Return:
%Function matrix for Segment 2 to N-1
function func = f2T(Doil,Dgas,Tres,Tref,XT,q,Lfrac,Bo,Bg,Rs,N,Kappa_t,...
    Kappa_a,f2,b,deltaP_a,deltaP_t,KJT)
for i=0:N-3
    var = i*4;
    % Energy balance at tubing node
    f2(1+2*i) = -(Doil*q(var+6)*Lfrac(var+6)/Bo(3*i+5)...
        + Dgas*(q(var+6)*(1-Lfrac(var+6))/Bg(3*i+5)...
        + q(var+6)*Lfrac(var+6)*Rs(3*i+5)/Bo(3*i+5))*...
        ((XT(2*i+3)-XT(2*i+2))*b(var+6)...
        - (Doil*q(var+5)*Lfrac(var+5)/Bo(3*i+4)...
        + Dgas*(q(var+5)*(1-Lfrac(var+5))/Bg(3*i+4)...
        + q(var+5)*Lfrac(var+5)*Rs(3*i+4)/Bo(3*i+4))*...
        ((XT(2*i+5)-XT(2*i+3))-KJT*deltaP_t(i+2))*b(var+5)...
        - Kappa_t(i+2)*(XT(2*i+3)-XT(2*i+2)));
    % Energy balance at annular node
    f2(2+2*i) = -(Doil*q(var+8)*Lfrac(var+8)/Bo(3*i+6)...
        + Dgas*(q(var+8)*(1-Lfrac(var+8))/Bg(3*i+6)...
        + q(var+8)*Lfrac(var+8)*Rs(3*i+6)/Bo(3*i+6))*...
        ((XT(2*i+2)-Tres(i+2)/Tref))*b(var+8)...
        - Kappa_a(i+2)*(XT(2*i+2)-Tres(i+2)/Tref)...
        - (Doil*q(var+7)*Lfrac(var+7)/Bo(3*i+5)...
        + Dgas*(q(var+7)*(1-Lfrac(var+7))/Bg(3*i+5)...
        + q(var+7)*Lfrac(var+7)*Rs(3*i+5)/Bo(3*i+5))*...
        ((XT(2*i+4)-XT(2*i+2))-KJT*deltaP_a(i+2))*b(var+7);
    % If there is discontinuity in annular flow the temperature at the first
    % annular node is assumed to have the temperature of the inflow from the
    % reservoir
    if b(var+7)==0
        f2(2+2*i) = XT(i*2+4)-Tres(i+2)/Tref;
    end
end
func = f2;
% For case 3 (well with multiple inflow control valves), add
% c = b; % c is bridge indices
% Create all-positive bridge indeces so that the temperature change due to
% pressure drop is dependent on the flow direction but only on the pressure
% drop (either positive or negative change along the flow direction)
% for i = 1:(4*N-2)
%     if b(i)<0
%         b(i)=-b(i);
%     end
% end
% Flow directions can still be determined using "c". At the annular node
% where the reversed flow first starts, the temperature is fixed so that it
% equals to the temperature of the reservoir inflow.
% if c(var+3)==-1
%     if c(var+7)==1
%         f2(2+2*i) = XT(i*2+2)-Tres(i+2)/Tref;
%         f2(2+2*(i-1)) = XT((i-1)*2+4)-Tres(i+2)/Tref;
%     end
% end
% end

% Generate function matrix for Segment N
%Input:

```

```

%
%Doil, Dgas :   Precalculated coefficients
%XT          :   Unknown temperatures at each iteration
%q           :   Flow rates
%Lfrac       :   Liquid holdups
%Bo,Bg,Rs    :   Black-oil properties
%N           :   Number of segments
%f3          :   Generated zero function matrix
%num_varT    :   Number of unknown temperatures
%Nodes       :   Number of node
%bridges     :   Number of bridges
%Kappa_t     :   Overall heat transfer coefficients for fluid in tubing
%Kappa_a     :   Overall heat transfer coefficients for fluid in annulus
%b           :   Bridge indexes
%deltaP_a    :   Pressure drop between nodes for fluid in annulus
%deltaP_t    :   Pressure drop between nodes for fluid in tubing
%KJT         :   Joule-Thompson coefficient
%
%Return:
%Function matrix for Segment N
function func = f3T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,N,f3,num_varT,Nodes,...
    bridges,Kappa_t,Kappa_a,b,deltaP_a,deltaP_t,KJT)
% Energy balance at tubing node
f3(1) = -(Doil*q(bridges)*Lfrac(bridges)/Bo(Nodes)...
    + Dgas*(q(bridges)*(1-Lfrac(bridges))/Bg(Nodes)...
    + q(bridges)*Lfrac(bridges)*Rs(Nodes)/Bo(Nodes))*...
    ((XT(num_varT-1)-XT(num_varT-2))*b(bridges)...
    - (Doil*q(bridges-1)*Lfrac(bridges-1)/Bo(Nodes-1)...
    + Dgas*(q(bridges-1)*(1-Lfrac(bridges-1))/Bg(Nodes-1)...
    + q(bridges-1)*Lfrac(bridges-1)*Rs(Nodes-1)/...
    Bo(Nodes-1))*((XT(num_varT)-XT(num_varT-1))*...
    - KJT*deltaP_t(N))*b(bridges-1)...
    - Kappa_t(N)*(XT(num_varT-1)-XT(num_varT-2)));
func = f3;

```

Functions to generate jacobian matrices in network temperature solver

```

% Generate jacobian matrix for Segment 1
%Input:
%
%Doil, Dgas :   Precalculated coefficients
%XT          :   Unknown temperatures at each iteration
%q           :   Flow rates
%Lfrac       :   Liquid holdups
%Bo,Bg,Rs    :   Black-oil properties
%Kappa_t     :   Overall heat transfer coefficients for fluid in tubing
%Kappa_a     :   Overall heat transfer coefficients for fluid in annulus
%j1          :   Generated zero jacobian matrix
%
%Return:
%Jacobian matrix for Segment 1
function func = j1T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,Kappa_t,Kappa_a,j1)
j1(1,1) = 1;
j1(2,1) = -(Doil*q(2)*Lfrac(2)/Bo(2)...
    + Dgas*(q(2)*(1-Lfrac(2))/Bg(2)...
    + q(2)*Lfrac(2)*Rs(2)/Bo(2))*(+1)...
    - (Doil*q(1)*Lfrac(1)/Bo(1)...
    + Dgas*(q(1)*(1-Lfrac(1))/Bg(1)...
    + q(1)*Lfrac(1)*Rs(1)/Bo(1))*(-1)...
    - Kappa_t(1)*(+1);

```



```

j1(2,3) = - (Doil*q(1)*Lfrac(1)/Bo(1)...
            + Dgas*(q(1)*(1-Lfrac(1))/Bg(1)...
            + q(1)*Lfrac(1)*Rs(1)/Bo(1))*(+1);
j1(3,1) = - 0*(Doil*q(2)*Lfrac(2)/Bo(2)...
            + Dgas*(q(2)*(1-Lfrac(2))/Bg(2)...
            + q(2)*Lfrac(2)*Rs(2)/Bo(2))*(+1)...
            + 0*Kappa_t(1)*(+1);
j1(3,2) = - (Doil*q(3)*Lfrac(3)/Bo(2)...
            + Dgas*(q(3)*(1-Lfrac(3))/Bg(2)...
            + q(3)*Lfrac(3)*Rs(2)/Bo(2))*(+1);

func = j1;

% Generate jacobian matrix for Segment 2 to N-1
%Input:
%
%Doil, Dgas :   Precalculated coefficients
%XT         :   Unknown temperatures at each iteration
%q          :   Flow rates
%Lfrac      :   Liquid holdups
%Bo,Bg,Rs   :   Black-oil properties
%N          :   Number of segments
%Kappa_t    :   Overall heat transfer coefficients for fluid in tubing
%Kappa_a    :   Overall heat transfer coefficients for fluid in annulus
%j2         :   Generated zero jacobian matrix
%b          :   Bridge indexes
%
%Return:
%Jacobian matrix for Segment 2 to N-1
function func = j2T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,N,Kappa_t,Kappa_a,j2,b)
for i=0:N-3
    var = 4*i;
    j2(1+2*i,2*i+3) = -(Doil*q(var+6)*Lfrac(var+6)/Bo(3*i+5)...
                        + Dgas*(q(var+6)*(1-Lfrac(var+6))/Bg(3*i+5)...
                        + q(var+6)*Lfrac(var+6)*Rs(3*i+5)/...
                        Bo(3*i+5))*(+1)*b(var+6)...
                        + (Doil*q(var+5)*Lfrac(var+5)/Bo(3*i+4)...
                        + Dgas*(q(var+5)*(1-Lfrac(var+5))/Bg(3*i+4)...
                        + q(var+5)*Lfrac(var+5)*Rs(3*i+4)/...
                        Bo(3*i+4))*(-1)*b(var+5)...
                        - Kappa_t(i+2)*(+1);
    j2(1+2*i,2*i+2) = - (Doil*q(var+6)*Lfrac(var+6)/Bo(3*i+5)...
                        + Dgas*(q(var+6)*(1-Lfrac(var+6))/Bg(3*i+5)...
                        + q(var+6)*Lfrac(var+6)*Rs(3*i+5)/...
                        Bo(3*i+5))*(-1)*b(var+6)...
                        - Kappa_t(i+2)*(-1);
    j2(1+2*i,2*i+5) = - (Doil*q(var+5)*Lfrac(var+5)/Bo(3*i+4)...
                        + Dgas*(q(var+5)*(1-Lfrac(var+5))/Bg(3*i+4)...
                        + q(var+5)*Lfrac(var+5)*Rs(3*i+4)/...
                        Bo(3*i+4))*(+1)*b(var+5);
    j2(2+2*i,2*i+2) = -(Doil*q(var+8)*Lfrac(var+8)/Bo(3*i+6)...
                        + Dgas*(q(var+8)*(1-Lfrac(var+8))/Bg(3*i+6)...
                        + q(var+8)*Lfrac(var+8)*Rs(3*i+6)/...
                        Bo(3*i+6))*(+1)*b(var+8) ...
                        - Kappa_a(i+2)*(+1)...
                        - (Doil*q(var+7)*Lfrac(var+7)/Bo(3*i+5)...
                        + Dgas*(q(var+7)*(1-Lfrac(var+7))/Bg(3*i+5)...
                        + q(var+7)*Lfrac(var+7)*Rs(3*i+5)/...
                        Bo(3*i+5))*(-1)*b(var+7);

```

```

j2(2+2*i,2*i+4) = - (Doil*q(var+7)*Lfrac(var+7)/Bo(3*i+5)...
                    + Dgas*(q(var+7)*(1-Lfrac(var+7))/Bg(3*i+5)...
                    + q(var+7)*Lfrac(var+7)*Rs(3*i+5)/...
                    Bo(3*i+5))*(+1)*b(var+7);

if b(4*i+7) == 0
    j2(2+2*i,2*i+4) = 1;
end

end

func = j2;

% For case 3 (well with multiple inflow control valves), add
% c = b; % c is bridge indices
% Create all-positive bridge indeces so that the temperature change due to
% pressure drop is dependent on the flow direction but only on the pressure
% drop (either positive or negative change along the flow direction)
% for i = 1:(4*N-2)
%     if b(i)<0
%         b(i)=-b(i);
%     end
% end
% Flow directions can still be determined using "c". At the annular node
% where the reversed flow first starts, the temperature is fixed so that it
% equals to the temperature of the reservoir inflow.
% if c(var+3)==-1
%     if c(var+7)==1
%         j2(2+2*i,2*i+2) = 1;
%         j2(2+2*(i-1),2*(i-1)+4) = 1;
%     end
% end

% Generate jacobian matrix for Segment N
%Input:
%
%Doil, Dgas : Precalculated coefficients
%XT : Unknown temperatures at each iteration
%q : Flow rates
%Lfrac : Liquid holdups
%Bo,Bg,Rs : Black-oil properties
%N : Number of segments
%j3 : Generated zero jacobian matrix
%num_varT : Number of unknown temperatures
%Nodes : Number of nodes
%bridges : Number of bridges
%Kappa_t : Overall heat transfer coefficients for fluid in tubing
%Kappa_a : Overall heat transfer coefficients for fluid in annulus
%b : Bridge indexes
%
%Return:
%Jacobian matrix for Segment N
function func = j3T(Doil,Dgas,XT,q,Lfrac,Bo,Bg,Rs,N,j3,num_varT,Nodes,...
    bridges,Kappa_t,Kappa_a,b)
j3(1,num_varT-1) = -(Doil*q(bridges)*Lfrac(bridges)/Bo(Nodes)...
                    + Dgas*(q(bridges)*(1-Lfrac(bridges))/Bg(Nodes)...
                    + q(bridges)*Lfrac(bridges)*Rs(Nodes)/...
                    Bo(Nodes))*(+1)*b(bridges)...
                    - (Doil*q(bridges-1)*Lfrac(bridges-1)/Bo(Nodes-1)...
                    + Dgas*(q(bridges-1)*(1-Lfrac(bridges-1))/...
                    Bg(Nodes-1) + q(bridges-1)*Lfrac(bridges-1)*...
                    Rs(Nodes-1)/Bo(Nodes-1))*(-1)*b(bridges-1)...
                    - Kappa_t(N)*(+1);
j3(1,num_varT-2) = -(Doil*q(bridges)*Lfrac(bridges)/Bo(Nodes)...

```

```

        + Dgas*(q(bridges)*(1-Lfrac(bridges))/Bg(Nodes)...
        + q(bridges)*Lfrac(bridges)*Rs(Nodes)/...
        Bo(Nodes)))*(-1)*b(bridges)...
        - Kappa_t(N)*(-1);
j3(1,num_varT) = - (Doil*q(bridges-1)*Lfrac(bridges-1)/Bo(Nodes-1)...
        + Dgas*(q(bridges-1)*(1-Lfrac(bridges-1))/...
        Bg(Nodes-1) + q(bridges-1)*Lfrac(bridges-1)*...
        Rs(Nodes-1)/Bo(Nodes-1)))*(+1)*b(bridges-1);

func = j3;

```

Function to check for convergence in network temperature solver

```

% Checking for convergence
%Input:
%
%X1      :   Unknown temperatures at nth iteration
%X2      :   Unknown temperatures at n+1th iteration
%N       :   Number of unknown temperatures
%threshold :   Tolerance value for checking for convergence
%
%Return:
%flag     :   Convergence status (false = convergence)
%epsilon  :   Convergence value
function [func, func1] = checkConvT(X1,X2,N,threshold)
flag = true;
epsilon = 0;
temp = 0;
for i=1:N
    if X1(i)~=0
        temp = abs(X1(i)-X2(i))/(X1(i)*N);
    else
        if X2==0
            temp=0;
        else
            X1(i) = 1e-20;
        end
    end
    epsilon = epsilon + temp;
end
if(epsilon < threshold)
    flag = false;
end
func = flag;
func1 = epsilon;

```

Oil viscosity update

```

% Recalculate pressure/temperature dependent viscosities
%Input:
%
%XT      :   Unknown temperatures (converged)
%X1      :   Array containing pressure parameters
%Tres    :   Reservoir temperatures at reservoir nodes
%pref    :   Reference pressure
%pb      :   Bubblepoint pressure
%N       :   Number of segments
%num_var :   Number of unknowns in isothermal calculations

```

```

%num_varT : Number of unknown temperatures
%Nodes : Number of nodes
%bridges : Number of bridges
%Rs : Gas solubilities
%mu2P_res : Two-phase viscosities at reservoir conditions
%mu_res : Viscosities of both phases at reservoir conditions
%
%Return:
%mu2P : Two-phase viscosities in every bridge in the network
%mu : Viscosities of both phases at every node in the network
function func = updatemu(XT,X1,Tres,pres,pref,pb,N,num_var,num_varT,...
    Nodes,bridges,Rs,mu2P_res,mu_res)

pb_temp=pb; % Bubblepoint pressure in bara
pb = pb*1e2; % Bubblepoint pressure in kPa
XT_toe = Tres(1); % Temperature at toe of well

mu = zeros(2,Nodes); % Viscosities of both phases (1=liquid, 2=gas)
mu_od = zeros(1,Nodes); % Dead-oil viscosities
mu_sat = zeros(1,Nodes); % Saturated oil viscosities

for i=0:N-2
    var = i*9;
    % Change unit of pressure from Pa to kPa
    X1(1+var) = X1(1+var)*pref/10^3;
    % Change unit of pressure from Pa to kPa
    X1(2+var) = X1(2+var)*pref/10^3;

    % Calculate viscosity for the gas phase (from curve-fit to values from EOS)
    % Change unit of pressure from kPa to bara
    X_temp(1) = X1(1+var)/10^2;
    % Change unit of pressure from kPa to bara
    X_temp(2) = X1(2+var)/10^2;
    % Tubing node
    if X_temp(1)<pb_temp
        mu(2,3*i+1) = 3e-10*X_temp(1)^3 - 8e-09*X_temp(1)^2 + ...
            3e-05*X_temp(1) + 0.0136;
        mu(2,3*i+1) = mu(2,3*i+1)*10^(-3);
    end
    % Annular node
    if X_temp(2)<pb_temp
        mu(2,3*i+2) = 3e-10*X_temp(2)^3 - 8e-09*X_temp(2)^2 + ...
            3e-05*X_temp(2) + 0.0136;
        mu(2,3*i+2) = mu(2,3*i+2)*10^(-3);
    end
    % Reservoir node
    mu(2,3*i+3) = mu_res(2,i+1);

    % Calculate viscosity for the liquid phase (from correlations)
    % Tubing node
    % Calculate coefficient A
    A = 10.715*(5.61*Rs(3*i+1) + 100)^(-0.515);
    % Calculate coefficient B
    B = 5.44*(5.61*Rs(3*i+1) + 150)^(-0.338);
    % Calculate dead-oil viscosity in Pa.s, Tres in C
    mu_od(3*i+1) = 58.885*exp(-0.018*XT(i*2+1))*10^(-3);
    % Calculate saturated-oil viscosity
    mu_sat(3*i+1) = (A*(mu_od(3*i+1)*1e3)^B)*1e-3;
    if X_temp(1)<pb_temp
        % Oil viscosity = saturated-oil viscosity for pressures below
        % the bubblepoint
        mu(1,3*i+1) = mu_sat(3*i+1);
    else
        % Calculate undersaturated-oil viscosity for pressures above
        % the bubblepoint
        m = 0.263*X1(1+var)^1.187*exp(-11.513-1.302e-5*X1(1+var));
        mu(1,3*i+1) = mu_sat(3*i+1)*(X1(1+var)/pb)^m;
    end
end

```

```

end
% Annular node
% Calculate coefficient A
A = 10.715*(5.61*Rs(3*i+2) + 100)^(-0.515);
% Calculate coefficient B
B = 5.44*(5.61*Rs(3*i+2) + 150)^(-0.338);
if i==0 % Segment 1
% Calculate dead-oil viscosity in Pa.s, Tres in C
mu_od(3*i+2) = 58.885*exp(-0.018*XT_toe)*10^(-3);
% Calculate saturated-oil viscosity
mu_sat(3*i+2) = (A*(mu_od(3*i+2)*1e3)^B)*1e-3;
if X_temp(2)<pb_temp
% Oil viscosity = saturated-oil viscosity for pressures
% below the bubblepoint
mu(1,3*i+2) = mu_sat(3*i+2);
else
% Calculate undersaturated-oil viscosity for pressures
% above the bubblepoint
m = 0.263*X1(2+var)^1.187*exp(-11.513-1.302e-5*X1(2+var));
mu(1,3*i+2) = mu_sat(3*i+2)*(X1(2+var)/pb)^m;
end
else % Segment 2 to N-1
% Calculate dead-oil viscosity in Pa.s, Tres in C
mu_od(3*i+2) = 58.885*exp(-0.018*XT(i*2))*10^(-3);
% Calculate saturated-oil viscosity
mu_sat(3*i+2) = (A*(mu_od(3*i+2)*1e3)^B)*1e-3;
if X_temp(2)<pb_temp
% Oil viscosity = saturated-oil viscosity for pressures
% below the bubblepoint
mu(1,3*i+2) = mu_sat(3*i+2);
else
% Calculate undersaturated-oil viscosity for pressures
% above the bubblepoint
m = 0.263*X1(2+var)^1.187*exp(-11.513-1.302e-5*X1(2+var));
mu(1,3*i+2) = mu_sat(3*i+2)*(X1(2+var)/pb)^m;
end
end
% Reservoir node
mu(2,3*i+3) = mu_res(2,i+1);
end
% Change unit of pressure from Pa to kPa
X1(num_var-5) = X1(num_var-5)*pref/10^3;
% Change unit of pressure from Pa to kPa
X1(num_var-4) = X1(num_var-4)*pref/10^3;
% Calculate viscosity for the gas phase (from curve-fit to values from EOS)
% Change unit of pressure from kPa to bara
X_temp(1) = X1(num_var-5)/10^2;
% Change unit of pressure from kPa to bara
X_temp(2) = X1(num_var-4)/10^2;
% Tubing node
if X_temp(1)<pb_temp
mu(2,Nodes-1) = 3e-10*X_temp(1)^3 - 8e-09*X_temp(1)^2 + ...
3e-05*X_temp(1) + 0.0136;
mu(2,Nodes-1) = mu(2,Nodes-1)*10^(-3);
end
% Annular node
if X_temp(2)<pb_temp
mu(2,Nodes) = 3e-10*X_temp(2)^3 - 8e-09*X_temp(2)^2 + ...
3e-05*X_temp(2) + 0.0136;
mu(2,Nodes) = mu(2,Nodes)*10^(-3);
end
% Calculate viscosity for the liquid phase (from correlations)

```

```

% Tubing node
% Calculate coefficient A
A = 10.715*(5.61*Rs(Nodes-1) + 100)^(-0.515);
% Calculate coefficient B
B = 5.44*(5.61*Rs(Nodes-1) + 150)^(-0.338);
% Calculate dead-oil viscosity in Pa.s, Tres in C
mu_od(Nodes-1) = 58.885*exp(-0.018*XT(num_varT-1))*10^(-3);
% Calculate saturated-oil viscosity
mu_sat(Nodes-1) = (A*(mu_od(Nodes-1)*1e3)^B)*1e-3;
if X_temp(1)<pb_temp
    % Oil viscosity = saturated-oil viscosity for pressures below
    % the bubblepoint
    mu(1,Nodes-1) = mu_sat(Nodes-1);
else
    % Calculate undersaturated-oil viscosity for pressures above
    % the bubblepoint
    m = 0.263*X1(num_var-5)^1.187*exp(-11.513-1.302e-5...
        *X1(num_var-5));
    mu(1,Nodes-1) = mu_sat(Nodes-1)*(X1(num_var-5)/pb)^m;
end

% Annular node
% Calculate coefficient A
A = 10.715*(5.61*Rs(Nodes) + 100)^(-0.515);
% Calculate coefficient B
B = 5.44*(5.61*Rs(Nodes) + 150)^(-0.338);
% Calculate dead-oil viscosity in Pa.s, Tres in C
mu_od(Nodes) = 58.885*exp(-0.018*XT(num_varT-2))*10^(-3);
% Calculate saturated-oil viscosity
mu_sat(Nodes) = (A*(mu_od(Nodes)*1e3)^B)*1e-3;
if X_temp(2)<pb_temp
    % Oil viscosity = saturated-oil viscosity for pressures below
    % the bubblepoint
    mu(1,Nodes) = mu_sat(Nodes);
else
    % Calculate undersaturated-oil viscosity for pressures above
    % the bubblepoint
    m = 0.263*X1(num_var-4)^1.187*exp(-11.513-1.302e-5...
        *X1(num_var-4));
    mu(1,Nodes) = mu_sat(Nodes)*(X1(num_var-4)/pb)^m;
end

% Generate two-phase viscosities in every bridge
mu2P = zeros(1,bridges);
for i=0:N-2
    var = i*9;
    % Tubing bridge of Segment i+1
    mu2P(i*4+1) = mu(1,3*i+1)*X1(var+7) + mu(2,3*i+1)*(1-X1(var+7));
    % Annulus-to-tubing bridge of Segment i+1
    mu2P(i*4+2) = mu(1,3*i+2)*X1(var+8) + mu(2,3*i+2)*(1-X1(var+8));
    % Annular bridge of Segment i+1
    mu2P(i*4+3) = mu(1,3*i+2)*X1(var+9) + mu(2,3*i+2)*(1-X1(var+9));
    % Inlet bridge of Segment i+1
    mu2P(i*4+4) = mu2P_res(i+1);
end

% Tubing bridge of Segment N
mu2P(bridges-1) = mu(1,Nodes-1)*X1(num_var-1) + ...
    mu(2,Nodes-1)*(1-X1(num_var-1));
% Annulus-to-tubing bridge of Segment N
mu2P(bridges) = mu(1,Nodes)*X1(num_var) + mu(2,Nodes)*(1-X1(num_var));
func = mu2P;

```

Determine flow parameters to be used in detailed asphaltene precipitation analysis

```
% Get flow parameters at the location where asphaltene precipitation is
% suspected to be used in the detailed analysis
% Indicate the point in the well e.g. Segment 200 where asphaltene is a
% risk
locationindex = 200;

% Find flow parameters at the location at risk, e.g. for flow through ICV
% from annulus into tubing, type:
% Pressure in annulus of the segment at risk (upstream of the ICV)
riskp_1 = p_annulus(locationindex);
% Pressure in tubing of the segment at risk (downstream of the ICV)
riskp_2 = p_tubing(locationindex);
% Distance of the segment at risk from toe of well
risklength = lengths(locationindex);
% Flow rate through the ICV
riskflowrate = slotFlowRates(locationindex);
% Temperature in the annulus (upstream of the ICV)
riskT_1 = T_annulus(locationindex);
% Fluid density of the flow through ICV (annular-to-tubing bridge of
% Segment 200)
rho_1 = rho2P((locationindex-1)*4+2);

disp('Location where asphaltene might occur in the network');
disp(num2str(risklength));
disp('Upstream pressure');
disp(num2str(riskp_1));
disp('Downstream pressure');
disp(num2str(riskp_2));
disp('Flow rate through the restriction');
disp(num2str(riskflowrate));
disp('Upstream temperature');
disp(num2str(riskT_1));

% For Case 2-C (well with two temperature zones),
% Pressure in tubing at the mixing point of the two streams
% riskp_1 = p_tubing(locationindex);
% Pressure in tubing after the mixing of the two streams
% riskT_1 = T_tubing(locationindex+1);

% For case 3 (well with multiple inflow control valves), find flow
% parameters as in Case 1. Use locationindex = 95 for ICV 2 and
% locationindex = 100 for ICV 1.

% For Case 4 (well with restricted flow in annulus),
% locationindex = 170;
% Pressure in annulus of the segment at risk (upstream of the restriction)
% riskp_1 = p_annulus(locationindex);
% Pressure in annulus of the segment downstream of the restriction
% riskp_2 = p_annulus(locationindex+1);
% Distance of the segment at risk from toe of well
% risklength = lengths(locationindex);
% Flow rate through the restriction
% riskflowrate = annularFlowrates(locationindex);
% Temperature in the annulus upstream of the restriction
% riskT_1 = T_annulus(locationindex);
% Fluid density of the flow through restriction (annular bridge of Segment
% 170)
% rho_1 = rho2P((locationindex-1)*4+3);
```

Main programme of valve pressure drop model

```
inputvalue;          % Input values for solving valve pressure drop
if p0*1e-5 > pb      % For liquid (incompressible) flows, solve analytically:
    solvepdrop_incomp;
else
    solvepdrop; % For two-phase (compressible) flows, solve numerically:
end
% Plot results
plotp_restriction
```

Data input file for valve pressure drop model

```
% Input values for solving valve pressure drop
p0 = riskp_1;          % Input initial pressure (bara)
T = riskT_1;           % Input temperature (C)
MW = 171.35;           % Input molecular weight of fluid (g/mol)
R = 8.31447;           % Input Universal Gas Constant
                        % (m^3 Pa mol^-1 K^-1)
Z_pressure = generateZ(p0,pb); % Generated Z factor at the upstream
                        % conditions
MolVol = generateMolVol(T,pb); % Generated mole volume of fluid at the
                        % upstream conditions
fFactor = 0.043;       % Input friction factor
Lvalve = 0.14;         % Input length of restriction (m)
r1 = 0.03;             % Input radius upstream of the valve (m)
r2 = 0.01;             % Input the radius of the valve throat (m)
r3 = 0.08;             % Input radius downstream of the valve (m)
AA = 0.03;             % Distance upstream of the contraction
BB = AA+0.01;          % Distance at the beginning of valve throat
CC = BB+0.01;          % Distance at the end of valve throat
DD = CC+0.01;          % Distance at the end of the contraction

p0 = p0*1e5;           % Change unit of pressure from bara to Pa
MW = MW*1e-3;          % Change unit of molecular weight from g/mol to kg/mol
T = T+273.15;          % Change unit of temperature from C to K

% Calculate mass rate through the restriction
mRate = riskflowrate/(24*60*60)/(MolVol*1e-6)*MW;

% For Case 4 (well with restricted flow in annulus),
% r4 = ro(170);        % Outer radius of the annulus (m)
% r1 = ri(169);        % Radius upstream of the restriction (m)
% r2 = r_temp(170);    % Increased radius of the tubing (m)
% r3 = ri(171);        % Radius downstream of the restriction (m)
% Lvalve = 10.02;
% AA = 0.01;
% BB = AA+(Lvalve-0.02)/30;
% CC = BB+(Lvalve-0.02-2*(Lvalve-0.02)/30);
% DD = CC+(Lvalve-0.02)/30;
```

Valve pressure drop model for single-phase liquids

```
% Calculate analytically pressures in Section 1 of the valve (before
% contraction) for incompressible fluids
```



```

%Input:
%
%r1      :   Radius upstream of the valve
% mRate   :   Mass rate through the restriction
% fFactor :   Friction factor
% z1      :   Upstream position of the section
% z2      :   Downstream position of the section
% p0      :   Upstream pressure of the section
% rho     :   Fluid density
%
%Return:
%P        :   Pressures at different positions
%T1       :   Positions corresponding to the pressures
function [func1, func2] = pdrop_1(r1,mRate,fFactor,z1,z2,p0,rho)
A = pi*r1^2;           % Calculate cross-sectional area
Dh = 2*r1;             % Calculate hydraulic diameter
m = mRate;
f = fFactor;
h = (z2-z1)/40;        % Divide the section into 40 length intervals
T1(1) = z1;            % Start with the upstream position
p(1)=p0;               % Start with the upstream pressure
for i=1:40
    % Calculate postition at the end of the lenght step
    T1(i+1) = z1+h*i;
    % Calculate pressure at the specified position
    p(i+1) = p(i)-f*m^2/2/rho/A^2/Dh*h;
end
func2 = p;
func1 = T1;

% Calculate analytically pressures in Section 2 of the valve (tapered
% section until the valve throat) for incompressible fluids
%Input:
%
%r1      :   Radius upstream of the valve
%r2      :   Radius of the valve throat
% mRate   :   Mass rate through the restriction
% fFactor :   Friction factor
% AA      :   Upstream position of the section
% BB      :   Downstream position of the section
% p0      :   Upstream pressure of the section
% rho     :   Fluid density
%
%Return:
%P        :   Pressures at different positions
%T1       :   Positions corresponding to the pressures
function [func1, func2] = pdrop_2(r1,r2,mRate,fFactor,AA,BB,p0,rho)
m = mRate;
f=fFactor;
h = (BB-AA)/40;        % Divide the section into 40 length intervals
T1(1) = AA;            % Start with the upstream position
p(1)=p0;               % Start with the upstream pressure
for i=1:40
    T1(i+1) = AA+h*i;   % Calculate postition of the next lenght step
    T_int = T1(i);      % Assign initial condition for integration
    T_end = T1(i+1);    % Assign final condition for integration
    % Calculate pressure drop across the length step
    intterm1(i) = (-1/2*m^2/pi^2/rho/((r2-r1)/(BB-AA)*T_end-(r2-r1)*...
        AA/(BB-AA)+r1)^4)-(-1/2*m^2/pi^2/rho/((r2-r1)/(BB-AA)*T_int-(r2-r1)*...

```

```

AA/(BB-AA)+r1)^4);
intterm2(i) = (-1/16*f*m^2/rho/pi^2*((-r2*T_end+r2*AA+r1*T_end-r1*BB)^2/...
(-BB+AA)^2)^(1/2)/(-r2*T_end+r2*AA+r1*T_end-r1*BB)^5*(-BB+AA)^6/...
(-r2+r1))-(-1/16*f*m^2/rho/pi^2*((-r2*T_int+r2*AA+r1*T_int-r1*BB)^2/...
(-BB+AA)^2)^(1/2)/(-r2*T_int+r2*AA+r1*T_int-r1*BB)^5*(-BB+AA)^6/...
(-r2+r1));
pdrop = intterm1(i)-intterm2(i);
p(i+1) = p(i)+pdrop; % Calculate pressure at the specified position
end

func2 = p;
func1 = T1;

```

```

% Calculate analytically pressures in Section 3 of the valve (valve throat)
% for incompressible fluids

```

```

%Input:
%
%r2      : Radius of the valve throat
%Rate    : Mass rate through the restriction
%fFactor : Friction factor
%z1      : Upstream position of the section
%z2      : Downstream position of the section
%p0      : Upstream pressure of the section
%rho     : Fluid density
%
%Return:
%P        : Pressures at different positions
%T1       : Positions corresponding to the pressures
function [func1, func2] = pdrop_3(r2,mRate,fFactor,z1,z2,p0,rho)
A = pi*r2^2; % Calculate cross-sectional area
Dh = 2*r2; % Calculate hydraulic diameter
m = mRate;
f = fFactor;
h = (z2-z1)/40; % Divide the section into 40 length intervals
T1(1) = z1; % Start with the upstream position
p(1)=p0; % Start with the upstream pressure
for i=1:40
% Calculate position at the end of the length step
T1(i+1) = z1+h*i;
% Calculate pressure at the specified position
p(i+1) = p(i)-f*m^2/2/rho/A^2/Dh*h;
end
func2 = p;
func1 = T1;

```

```

% Calculate analytically pressures in Section 4 of the valve (expanding
% section from the valve throat) for incompressible fluids

```

```

%Input:
%
%r1      : Radius downstream of the valve
%r2      : Radius of the valve throat
%Rate    : Mass rate through the restriction
%fFactor : Friction factor
%CC      : Upstream position of the section
%DD      : Downstream position of the section
%p0      : Upstream pressure of the section
%rho     : Fluid density
%
%Return:

```

```

%P      : Pressures at different positions
%T1     : Positions corresponding to the pressures
function [func1, func2] = pdrop_4(r1,r2,mRate,fFactor,CC,DD,p0,rho)
m = mRate;
f = fFactor;
h = (DD-CC)/40;      % Divide the section into 40 length intervals
T1(1) = CC;          % Start with the upstream position
p(1)=p0;             % Start with the upstream pressure
for i=1:40
    T1(i+1) = CC+h*i; % Calculate position of the next length step
    T_int = T1(i);    % Assign initial condition for integration
    T_end = T1(i+1);  % Assign final condition for integration
    % Calculate pressure drop across the length step
    interm1 = (-1/2*m^2/pi^2/rho/((-r2+r1)/(DD-CC)*T_end-(-r2+r1)*CC/...
        (DD-CC)+r2)^4)-(-1/2*m^2/pi^2/rho/((-r2+r1)/(DD-CC)*T_int-(-r2+r1)*...
        CC/(DD-CC)+r2)^4);
    interm2 = (-1/16*f*m^2/rho/pi^2*((-r2*T_end+r1*T_end-r1*CC+r2*DD)^2/...
        (-DD+CC)^2)^(1/2)/(-r2*T_end+r1*T_end-r1*CC+r2*DD)^5*(-DD+CC)^6/...
        (-r2+r1))-(-1/16*f*m^2/rho/pi^2*((-r2*T_int+r1*T_int-r1*CC+r2*DD)^2/...
        (-DD+CC)^2)^(1/2)/(-r2*T_int+r1*T_int-r1*CC+r2*DD)^5*(-DD+CC)^6/...
        (-r2+r1));
    pdrop = interm1-interm2;
    p(i+1) = p(i)+pdrop; % Calculate pressure at the specified position
end
func2 = p;
func1 = T1;

% Calculate analytically pressures in Section 5 of the valve (downstream
% after the contraction) for incompressible fluids
%Input:
%
%r1      : Radius downstream of the valve
%mRate   : Mass rate through the restriction
%fFactor : Friction factor
%z1      : Upstream position of the section
%z2      : Downstream position of the section
%p0      : Upstream pressure of the section
%rho     : Fluid density
%
%Return:
%P      : Pressures at different positions
%T1     : Positions corresponding to the pressures
function [func1, func2] = pdrop_5(r1,mRate,fFactor,z1,z2,p0,rho)
A = pi*r1^2; % Calculate cross-sectional area
Dh = 2*r1; % Calculate hydraulic diameter
m = mRate;
f = fFactor;
h = (z2-z1)/40; % Divide the section into 40 length intervals
T1(1) = z1; % Start with the upstream position
p(1)=p0; % Start with the upstream pressure
for i=1:40
    % Calculate position at the end of the length step
    T1(i+1) = z1+h*i;
    % Calculate pressure at the specified position
    p(i+1) = p(i)-f*m^2/2/rho/A^2/Dh*h;
end
func2 = p;
func1 = T1;

```

Valve pressure drop model for two-phase fluids

```

% Solve for valve pressure drop numerically for compressible fluids
% J's are arrays containing distances (of the corresponding pressures)
% from the upstream location
J1 = 0;
J2 = 0;
J3 = 0;
J4 = 0;
J5 = 0;
% Y's are arrays containing pressures at different distances from the
% upstream location
Y1 = 0;
Y2 = 0;
Y3 = 0;
Y4 = 0;
Y5 = 0;

step = 50;          % Assign number of length step

% Calculate pressures in Section 1 of the valve (before contraction)
% The first column is z, 2nd = y(z) = p(z)
[J1, Y1] = pdrop1_comp(0,AA,p0,r1,fFactor,MW,R,T,Z_pressure,mRate,step);

% Calculate pressures in Section 2 of the valve (tapered section until the
% valve throat)
[J2, Y2] = pdrop2_comp(AA,BB,Y1(end),r1,r2,fFactor,MW,R,T,Z_pressure,...
    mRate,step);

if BB~=CC
    % Calculate pressures in Section 3 of the valve (valve throat)
    [J3, Y3] = pdrop3_comp(BB,CC,Y2(end),r2,fFactor,MW,R,T,Z_pressure,...
        mRate,step);
    % Calculate pressures in Section 4 of the valve (expanding section from
    % the valve throat)
    [J4, Y4] = pdrop4_comp(CC,DD,Y3(end),r3,r2,fFactor,MW,R,T,...
        Z_pressure,mRate,step);
% If the valve throat has zero thickness (V-shaped valve)
elseif BB==CC
    % Calculate pressures in Section 3 of the valve (valve throat)
    J3 = J2(end);
    Y3 = Y2(end);
    % Calculate pressures in Section 4 of the valve (expanding section from
    % the valve throat)
    [J4, Y4] = pdrop4_comp(CC,DD,Y2(end),r3,r2,fFactor,MW,R,T,...
        Z_pressure,mRate,step);
end

% Calculate pressures in Section 5 of the valve (downstream after the
% contraction)
[J5, Y5] = pdrop5_comp(DD,Lvalve,Y4(end),r3,fFactor,MW,R,T,Z_pressure,...
    mRate,step);

% Calculate pressure drop across the valve
valvedrop = p0*1e-5-Y5(end)*1e-5

% Display upstream pressure
disp('Pressure at point 1 (bara)');
disp(num2str(p0*1e-5));
% Display downstream pressure
disp('Pressure at point 2 (bara)');
disp(num2str(Y5(end)*1e-5));
% Display valve pressure drop
disp('Pressure at point 2 (bara)');
disp(num2str(valvedrop));

% Combine pressure results from all sections
% Y will be used as an input pressures for asphaltene precipitation

```

```

% calculations
Y = 0;
Jtot = 0;
Y = [Y1; Y2; Y3; Y4; Y5];
Jtot = [J1; J2; J3; J4; J5];

% Calculate numerically pressures in Section 1 of the valve (before
% contraction) for incompressible fluids
%Input:
%
%z0      : Upstream position of the section
%z1      : Downstream position of the section
%P0      : Upstream pressure of the section
%r1      : Radius upstream of the valve
%f       : Friction factor
%MW      : Fluid molecular weight
%R       : Universal gas constant
%T       : Temperature (upstream)
%Z       : Z factor (upstream)
%m       : Mass rate through the restriction
%step    : Number of length steps
%
%Return:
%z_temp  : Positions corresponding to the pressures
%P       : Pressures at different positions
function [func1 func2] = pdrop1_2(z0,z1,P0,r1,f,MW,R,T,Z,m,step)
% Generate positions at the end of each length step
z_temp = z0:(z1-z0)/step:z1;
% Size of the length step
h = (z1-z0)/step;
sz = size(z_temp);
P = zeros(1,sz(2));
P(1) = P0; % Start with the upstream pressure
A = pi*r1^2; % Calculate cross-sectional area
Dh = 2*r1; % Calculate hydraulic diameter
dA_dz = 0; % Calculate derivative of area with
% respect to distance dA/dz

% Calculate coefficients for numerical pressure drop calculations
C1 = (m^2/A^3*dA_dz-f*m^2/2/A^2/Dh);
C2 = (MW/R/T/Z);
C3 = (m^2/A^2);
for i = 2:sz(2)
    k1 = h*C1/P(i-1)/(C2-C3/P(i-1)^2);
    k2 = h*C1/(P(i-1)-k1/2)/(C2-C3/(P(i-1)-k1/2)^2);
    k3 = h*C1/(P(i-1)-k2/2)/(C2-C3/(P(i-1)-k2/2)^2);
    k4 = h*C1/(P(i-1)-k3)/(C2-C3/(P(i-1)-k3)^2);
    % Calculate pressure at each position
    P(i) = P(i-1) + 1/6*k1 + 1/3*k2 + 1/3*k3 + 1/6*k4;
end
func1 = transpose(z_temp);
func2 = transpose(P);

% Calculate numerically pressures in Section 2 of the valve (tapered
% section until the valve throat) for incompressible fluids
%Input:
%
%z0      : Upstream position of the section

```

```

%z1      : Downstream position of the section
%P0      : Upstream pressure of the section
%r1      : Radius upstream of the valve
%r2      : Radius of the valve throat
%f       : Friction factor
%MW      : Fluid molecular weight
%R       : Universal gas constant
%T       : Temperature (upstream)
%Z       : Z factor (upstream)
%m       : Mass rate through the restriction
%step    : Number of length steps
%
%Return:
%z_temp  : Positions corresponding to the pressures
%P       : Pressures at different positions
function [func1 func2] = pdrop1_2(z0,z1,P0,r1,r2,f,MW,R,T,Z,m,step)
% Generate positions at the end of each length step
z_temp = z0:(z1-z0)/step:z1;
% Size of the length step
h = (z1-z0)/step;
sz = size(z_temp);
P = zeros(1,sz(2));
P(1) = P0;                                % Start with the upstream pressure
AA = z0;                                  % Assign upstream position
BB = z1;                                  % Assign downstream position
for i = 2:sz(2)
    for j = 1:4
        % Generate parameters for numerical pressure drop calculations
        if j == 1
            z = z_temp(i-1);
        elseif j == 4
            z = z_temp(i-1)+h;
        else
            z = z_temp(i-1)+1/2*h;
        end
        % Cross-sectional area
        A = pi*((r2-r1)*(z-AA)/(BB-AA)+r1)^2;
        % Hydraulic diameter
        Dh = 2*((r2-r1)*(z-AA)/(BB-AA)+r1)^2/(((r2-r1)*(z-AA)/...
            (BB-AA)+r1)^2)^(1/2);
        % Derivative of area with respect to distance dA/dz
        dA_dz = 2*pi*((r2-r1)*(z-AA)/(BB-AA)+r1)*(r2-r1)/(BB-AA);
        % Calculate coefficients for numerical pressure drop calculations
        C1 = (m^2/A^3*dA_dz-f*m^2/2/A^2/Dh);
        C2 = (MW/R/T/Z);
        C3 = (m^2/A^2);
        if j == 1
            k(j) = h*C1/P(i-1)/(C2-C3/P(i-1)^2);
        elseif j == 2
            k(j) = h*C1/(P(i-1)-k(j-1)/2)/(C2-C3/(P(i-1)-k(j-1)/2)^2);
        elseif j == 3
            k(j) = h*C1/(P(i-1)-k(j-1)/2)/(C2-C3/(P(i-1)-k(j-1)/2)^2);
        elseif j == 4
            k(j) = h*C1/(P(i-1)-k(j-1))/(C2-C3/(P(i-1)-k(j-1))^2);
        end
    end
    % Calculate pressure at each position
    P(i) = P(i-1) + 1/6*k(1) + 1/3*k(2) + 1/3*k(3) + 1/6*k(4);
end
func1 = transpose(z_temp);
func2 = transpose(P);

```

```
% Calculate numerically pressures in Section 3 of the valve (valve throat)
% for incompressible fluids
```

```
%Input:
```

```
%
%z0      : Upstream position of the section
%z1      : Downstream position of the section
%P0      : Upstream pressure of the section
%r2      : Radius of the valve throat
%f       : Friction factor
%MW      : Fluid molecular weight
%R       : Universal gas constant
%T       : Temperature (upstream)
%Z       : Z factor (upstream)
%m       : Mass rate through the restriction
%step    : Number of length steps
```

```
%
%Return:
```

```
%z_temp  : Positions corresponding to the pressures
%P       : Pressures at different positions
```

```
function [func1 func2] = pdrop3_2(z0,z1,P0,r2,f,MW,R,T,Z,m,step)
```

```
% Generate positions at the end of each length step
```

```
z_temp = z0:(z1-z0)/step:z1;
```

```
% Size of the length step
```

```
h = (z1-z0)/step;
```

```
sz = size(z_temp);
```

```
P = zeros(1,sz(2));
```

```
P(1) = P0;
```

```
% Start with the upstream pressure
```

```
A = pi*r2^2;
```

```
% Calculate cross-sectional area
```

```
Dh = 2*r2;
```

```
% Calculate hydraulic diameter
```

```
dA_dz = 0;
```

```
% Calculate derivative of area with
% respect to distance dA/dz
```

```
% Calculate coefficients for numerical pressure drop calculations
```

```
C1 = (m^2/A^3*dA_dz-f*m^2/2/A^2/Dh);
```

```
C2 = (MW/R/T/Z);
```

```
C3 = (m^2/A^2);
```

```
for i = 2:sz(2)
```

```
    k1 = h*C1/P(i-1)/(C2-C3/P(i-1)^2);
```

```
    k2 = h*C1/(P(i-1)-k1/2)/(C2-C3/(P(i-1)-k1/2)^2);
```

```
    k3 = h*C1/(P(i-1)-k2/2)/(C2-C3/(P(i-1)-k2/2)^2);
```

```
    k4 = h*C1/(P(i-1)-k3)/(C2-C3/(P(i-1)-k3)^2);
```

```
    % Calculate pressure at each position
```

```
    P(i) = P(i-1) + 1/6*k1 + 1/3*k2 + 1/3*k3 + 1/6*k4;
```

```
end
```

```
func1 = transpose(z_temp);
```

```
func2 = transpose(P);
```

```
% Calculate numerically pressures in Section 4 of the valve (expanding
% section from the valve throat) for incompressible fluids
```

```
%Input:
```

```
%
%z0      : Upstream position of the section
%z1      : Downstream position of the section
%p0      : Upstream pressure of the section
%r1      : Radius upstream of the valve
%r2      : Radius of the valve throat
%f       : Friction factor
%MW      : Fluid molecular weight
```

```

%R      :   Universal gas constant
%T      :   Temperature (upstream)
%Z      :   Z factor (upstream)
%m      :   Mass rate through the restriction
%step   :   Number of length steps
%
%Return:
%z_temp :   Positions corresponding to the pressures
%P       :   Pressures at different positions
function [func1 func2] = pdrop4_2(z0,z1,P0,r1,r2,f,MW,R,T,Z,m,step,testfactor)
% Generate positions at the end of each length step
z_temp = z0:(z1-z0)/step:z1;
% Size of the length step
h = (z1-z0)/step;
sz = size(z_temp);
P = zeros(1,sz(2));
P(1) = P0;                                % Start with the upstream pressure
CC = z0;                                  % Assign upstream position
DD = z1;                                  % Assign downstream position
for i = 2:sz(2)
    for j = 1:4
        % Generate parameters for numerical pressure drop calculations
        if j == 1
            z = z_temp(i-1);
        elseif j == 4
            z = z_temp(i-1)+h;
        else
            z = z_temp(i-1)+1/2*h;
        end
        % Cross-sectional area
        A = pi*((r1-r2)*(z-CC)/(DD-CC)+r2)^2;
        % Hydraulic diameter
        Dh = 2*((r1-r2)*(z-CC)/(DD-CC)+r2)^2/(((r1-r2)*(z-CC)/...
            (DD-CC)+r2)^2)^(1/2);
        % Derivative of area with respect to distance dA/dz
        dA_dz = 2*pi*((r1-r2)*(z-CC)/(DD-CC)+r2)*(r1-r2)/(DD-CC);
        % Calculate coefficients for numerical pressure drop calculations
        C1 = (m^2/A^3*dA_dz-f*m^2/2/A^2/Dh);
        C2 = (MW/R/T/Z);
        C3 = (m^2/A^2);
        if j == 1
            k(j) = h*C1/P(i-1)/(C2-C3/P(i-1)^2);
        elseif j == 2
            k(j) = h*C1/(P(i-1)-k(j-1)/2)/(C2-C3/(P(i-1)-k(j-1)/2)^2);
        elseif j == 3
            k(j) = h*C1/(P(i-1)-k(j-1)/2)/(C2-C3/(P(i-1)-k(j-1)/2)^2);
        elseif j == 4
            k(j) = h*C1/(P(i-1)-k(j-1))/(C2-C3/(P(i-1)-k(j-1))^2);
        end
    end
    % Calculate pressure at each position
    P(i) = P(i-1) + 1/6*k(1) + 1/3*k(2) + 1/3*k(3) + 1/6*k(4);
end
func1 = transpose(z_temp);
func2 = transpose(P);

% Calculate numerically pressures in Section 5 of the valve (downstream
% after the contraction) for incompressible fluids

```



```

%Input:
%
%z0      : Upstream position of the section
%z1      : Downstream position of the section
%P0      : Upstream pressure of the section
%r1      : Radius downstream of the valve
%f       : Friction factor
%MW      : Fluid molecular weight
%R       : Universal gas constant
%T       : Temperature (upstream)
%Z       : Z factor (upstream)
%m       : Mass rate through the restriction
%step    : Number of length steps
%
%Return:
%z_temp  : Positions corresponding to the pressures
%P       : Pressures at different positions
function [func1 func2] = pdrop5_2(z0,z1,P0,r1,f,MW,R,T,Z,m,step)
% Generate positions at the end of each length step
z_temp = z0:(z1-z0)/step:z1;
% Size of the length step
h = (z1-z0)/step;
sz = size(z_temp);
P = zeros(1,sz(2));
P(1) = P0; % Start with the upstream pressure
A = pi*r1^2; % Calculate cross-sectional area
Dh = 2*r1; % Calculate hydraulic diameter
dA_dz = 0; % Calculate derivative of area with respect to distance dA/dz

% Calculate coefficients for numerical pressure drop calculations
C1 = (m^2/A^3*dA_dz-f*m^2/2/A^2/Dh);
C2 = (MW/R/T/Z);
C3 = (m^2/A^2);
for i = 2:sz(2)
    k1 = h*C1/P(i-1)/(C2-C3/P(i-1)^2);
    k2 = h*C1/(P(i-1)-k1/2)/(C2-C3/(P(i-1)-k1/2)^2);
    k3 = h*C1/(P(i-1)-k2/2)/(C2-C3/(P(i-1)-k2/2)^2);
    k4 = h*C1/(P(i-1)-k3)/(C2-C3/(P(i-1)-k3)^2);
    % Calculate pressure at each position
    P(i) = P(i-1) + 1/6*k1 + 1/3*k2 + 1/3*k3 + 1/6*k4;
end
func1 = transpose(z_temp);
func2 = transpose(P);

```

Main programme of asphaltene model

```

format long e
inputdata; % Input values for asphaltene precipitation calculations
solver; % Solve for fluid fugacities and predict precipitation
% Display onset conditions
disp('Asphaltene onset pressure');
disp(num2str(p_onset));
disp('Asphaltene onset temperature');
disp(num2str(T_onset-273.15));
disp('Second asphaltene onset pressure (redispersed)');
disp(num2str(p_onset_2));
disp('Second asphaltene onset temperature (redispersed)');
disp(num2str(T_onset_2-273.15));

```

```

% plot results
plotfa;                % Plot fugacities
figure(2)
plotT2;                % Plot temperature profile

```

Data input file for asphaltene model

```

% Input values for asphaltene precipitation calculations
Ncom = 13;              % Number of components
% Input mole% of components in the feed
z = 0.01*[2.46 36.94 3.47 4.05 1.93 1.57 4.35 ...
13.4093 9.4338 5.6864 4.9512 7.6812 4.0681];
% Critical pressures (bara)
Pc = [73.76 42.46 45.68 48.84 37.53 33.79 36.94 ...
28.94 20.27 17.67 16.4 15.05 15.05];
% Critical temperatures (C)
Tc = [31.05 96.65 -84.265 32.25 146.823 192.114 262.433 343.462 ...
433.821 497.463 554.268 741.575 741.575];
Tc = Tc+273.15;        % Change unit of temperature from C to K
% Critical volume (cm^3/mol)
vc = [94 148 203 98.75 257.45 304.94 353.81 523.32 907.81 ...
1270.78 1656.99 3285.15 3285.15];
% Acentric factors
w = [0.225 0.152 8.85E-03 0.098 0.1878 0.2397 0.4057 ...
0.5998 0.8231 0.9933 1.1371 1.2575 1.2575];
% Coefficients for heat capacity correlations
CP = [ 1.98E+01 1.96E+01 5.41E+00 -4.22E+00 6.16E+00 ...
-6.41E+00 -1.51E+01 -1.86E+01 -1.65E+01 -1.55E+01 ...
-1.43E+01 -8.49E+00 -8.49E+00;
7.34E-02 5.04E-02 1.78E-01 3.06E-01 3.48E-01 ...
4.96E-01 5.44E-01 7.66E-01 1.23E+00 1.65E+00 ...
2.07E+00 3.73E+00 3.73E+00;
-5.60E-05 1.24E-05 -6.94E-05 -1.59E-04 -1.33E-04 ...
-2.65E-04 -2.49E-04 -3.21E-04 -5.18E-04 -6.89E-04 ...
-8.63E-04 -1.54E-03 -1.54E-03;
1.72E-08 -1.13E-08 8.71E-09 3.21E-08 6.89E-09 ...
5.50E-08 2.37E-08 0.00E+00 0.00E+00 0.00E+00 ...
0.00E+00 0.00E+00 0.00E+00 ];
% Asphaltene molar volume (cm^3/mol)
Va = 1040;
Psat = pb;              % Saturation pressure (bara)
Pref = 356.6933736;    % Pressure at reference conditions (bara) for
                        % calculating asphaltene fugacity
Tref = 100+273.15;     % Temperature at reference conditions (K) for
                        % calculating asphaltene fugacity
P1 = riskp_1;           % Pressure at upstream location (bara)
T1 = riskT_1+273.15;   % Temperature at upstream location (K)
kcalculation;          % Generate binary interaction parameters kij

% Generate binary interaction parameters (BIP)
% Assign one fixed value for the BIP between CO2 and other hydrocarbons
% (kCO2).
% Assign one fixed value for the BIP between CnB+ and solvents.
% BIP between hydrocarbons are calculated based on critical volumes.
kij = zeros(Ncom);
% Assign BIP between CO2 (component 1) and other components.
kCO2 = 0.12;
kij(1,:) = kCO2;
kij(:,1) = kCO2;
% Assign BIP between CnB+ and solvents (Component 2 to 6).

```

```

% This value was tuned to match the precipitate amount from the experiments
kCnB = 0.22;
kCnB_temp = [kCO2, kCnB, kCnB, kCnB, kCnB, kCnB, 0, 0, 0, 0, 0, 0];
% Exponent e for calculating BIP between hydrocarbons. This value was
% tuned to match the saturation pressure from the experiments
e = 1.5807;
% Calculate BIP between hydrocarbons
for i=1:Ncom
    for j=1:Ncom
        if i==j
            kij(i,j) = 0;
        else
            if i~=1
                if j~=1
                    kHC(i,j) = 1-((2*vc(i)^(1/6)*vc(j)^(1/6))/...
                        (vc(i)^(1/3)+vc(j)^(1/3)))^e;
                    kij(i,j) = kHC(i,j);
                    if i==Ncom
                        kij(i,j) = kCnB_temp(j);
                    end
                    if j==Ncom
                        kij(i,j) = kCnB_temp(i);
                    end
                end
            end
        end
    end
end
end

```

Asphaltene precipitation solver

```

% Solve for fluid fugacities and predict precipitation
% Calculate reference asphaltene fugacity
fa;
% Evaluate precipitation at point 1
% Calculate pure asphaltene fugacity at the specified conditions
faref = calculatefa(P1, Pref, Tref, Va, fa_ref);
% Asphaltene fugacity at the upstream location
faref1 = faref;
% Perform isothermal flash calculation at the upstream location
Point1;
% Fugacity of precipitating component in the liquid phase at the
% upstream location
fa1 = f(1,Ncom);
% Check for precipitation using precip.m, if fugacity in the liquid phase
% is less than the calculated asphaltene fugacity, there is no
% precipitation and precipindex remains zero.
precipindex = 0;
precip;
if precipindex==0
    % If there is no precipitation, assess precipitation based on the pressure
    % profile generated using the valve pressure pdrop model
    % Y contains pressures from the valve pressure drop model
    P2_temp = Y*1e-5;
    % Calculate initial enthalpy at the upstream location for isenthalpic
    % flash calculation
    initialenthalpy;
    % Predict asphaltene precipitation based on the pressure inside the
    % restriction
    sizeP = size(P2_temp);
    % Asphaltene fugacity in the solid phase
    faref_pt = zeros(1,sizeP(1));
    % Asphaltene fugacity in the liquid phase at each point

```

```

fa_pt = zeros(1,sizeP(1));
faref_pt(1) = faref1;
fa_pt(1) = fa1;
% Calculated temperature at each point obtained from isenthalpic flash
T_pt(1) = T1;

precipindex = 0;    % Indicator for precipitation
pindex = 2;        % Index for each local point along the restriction
                    % starting from the upstream location (pindex = 1)
count = 0;         % Indicator for onset asphaltene precipitation

% Iterate precipitation prediction from the upstream point to the
% downstream point of the restrictions
for pindex = 2:sizeP(1)
    % Pressure condition at the point of interest
    P2 = P2_temp(pindex);
    % Calculate pure asphaltene fugacity at the specified conditions
    faref = calculatefa(P2,Pref,Tref,Va,fa_ref);
    faref_pt(pindex) = faref;
    % Perform isenthalpic flash.
    % First isenthalpic flash iteration
    firstrun;
    % isen is the indicator to check for enthalpy conservation
    isen = 0;
    % Isenthalpic flash algorithms to solve for fugacities (f) and
    % temperature (T2).
    % Perform isothermal flash at different temperatures until the
    % energy conservation is satisfied.
    while isen == 0
        main;
    end
    % Record fugacity of the last component in the liquid phase
    fa_pt(pindex) = f(1,Ncom);
    % Record calculated temperature
    T_pt(pindex) = T2;
    % Check for precipitation using precip.m, if fugacity in the
    % liquid phase is less than the calculated asphaltene fugacity,
    % there is no precipitation and precipindex remains zero.
    precip;
    % If it is the first time that the precipitation is detected,
    % the conditions are onset asphaltene conditions
    if precipindex == 1
        if count == 1
            p_onset = P2          % Asphaltene onset pressure
            T_onset = T2          % Asphaltene onset temperature
        end
    end
    % If it is the first time that the precipitation is detected,
    % the conditions are onset asphaltene conditions
    if precipindex == 0
        if count > 1
            % Asphaltene onset pressure
            % (second detected -- redispersed)
            p_onset_2 = P2_temp(pindex-1)
            % Asphaltene onset temperature
            % (second detected -- redispersed)
            T_onset_2 = T_pt(pindex-1)
            count = 0;
        end
    end
end
end
end
end

```

Calculate pure asphaltene fugacity

```

% Calculate reference asphaltene fugacity
Wasp = 0; % Percent weight of asphaltene at reference conditions
Moil = 171.343; % Molecular weight of oil e.g. 202.4 g/mol
Masp = 665.627; % Molecular weight of asphaltene e.g. 617.6 g/mol
% Calculate mole fraction of precipitate at reference conditions
precipitate = Wasp*Moil/Masp/100;
% Calculate fugacity of precipitating component in the liquid phase when
% the precipitate was removed from the mixture
% Recalculate mole fractions of feed when precipitate was removed.
% Feed mole fractions for performing flash.
zref = z;
% The precipitate is deducted from the mole fraction of the precipitating
% component (last component).
zref(Ncom) = z(Ncom)-precipitate;
% Normalize zref so that the total mole fraction is one
sumzref = 0;
for i=1:Ncom
    sumzref = sumzref + zref(i);
end
zref = zref/sumzref;
% Check if the total of mole fraction is 1, sumzref must equal 1
sumzref = 0;
for i=1:Ncom
    sumzref = sumzref + zref(i);
end
ns = 0; % No precipitation
% (precipitate has been removed from the system)
% Perform flash calculations at the reference pressure and temperature
% Above bubblepoint pressure, there is only liquid.
if Pref>=Psat;
    x(1,:) = zref; % Component mole fractions in liquid phase =
    % feed mole fractions
    x(2,:) = 0; % No gas phase
    phase = 1; % Number of phase = 1
    n(1) = 1; % Liquid phase fraction = 1
    n(2) = 0; % Gas phase fraction = 0
    n(3) = 0; % Solid phase fraction = 0
else % Below bubblepoint pressure, there are two phases
    % Calculate initial K values using Wilson equation
    K = initialK(Pc, Pref, w, Tc, Tref, Ncom);
    % result is the indicator for equilibrium
    result = 0;
    while result == 0
        % Calculate vapor mole fraction
        n = Findnv(zref, K, ns, Ncom)
        % Calculate mole fractions of each component in liquid and
        % gas phases
        x = xi(zref, n, K, Ncom);
        % Calculate EOS parameters for liquid and gas phases using
        % random mixing rules
        [aij, a, A, ac, alph, m] = coeff_a(w, Tc, Pc, Tref, Pref, kij,...
            x, Ncom);
        [bi, b, B] = coeff_b(w, Tc, Pc, Tref, Pref, kij, x, Ncom);
        % Solve Peng-Robinson EOS's for Z factors
        Z = PR(A, B);
        % Calculate component fugacities
        f = fugacity_z(bi, b, B, aij, a, A, Z, x, Pref, Ncom);
        % Check for equilibrium -- equal fugacities in both phases
        [result ch] = check(f, Ncom);
        % If the equilibrium criteria are not satisfied, update K values
        for i = 1:Ncom-1
            f_ratio(i) = f(1,i)/f(2,i);
            % Calculate new K values --
            % K value of the last component is always zero

```

```

        K(i) = K(i)*f_ratio(i);
    end
    end
    % Check for number of phases again
    [n, x, phase] = nv_boundary(n, zref, x, Ncom)
end
% If there is only one phase (pressure > saturation pressure or
% vapor phase fraction > 1 or < 0)
if phase == 1
    % Calculate fugacities knowing that there is only one phase
    % Calculate EOS parameters for liquid and gas phases using random
    % mixing rules
    [aij, a, A, ac, alph, m] = coeff_a(w, Tc, Pc, Tref, Pref, kij, x,...
        Ncom);
    [bi, b, B] = coeff_b(w, Tc, Pc, Tref, Pref, kij, x, Ncom);
    % Solve Peng-Robinson EOS's for Z factors
    Z = PR(A, B);
    % Calculate component fugacities
    f = fugacity_z(bi, b, B, aij, a, A, Z, x, Pref, Ncom);
end
% Reference asphaltene fugacity equals fugacity of the last component in
% the liquid phase
fa_ref = f(1, Ncom);

```

```

% Calculate pure asphaltene fugacity at the specified conditions

```

```

%Input:
%
%P          :   Pressure to perform flash
%Pref       :   Pressure at reference conditions
%Tref       :   Temperture at reference conditions
%Va         :   Asphaltene molar volume
%fa_ref     :   Asphaltene fugacity at reference conditions
%Ncom       :   Number of components
%
%Return:
%Asphaltene fugacity at conditions of interest
function func = calculatefa(P,Pref,Tref,Va,fa_ref)
faref = exp(log(fa_ref)+Va*(P-Pref)/(83.1447215*Tref));
func = faref;

```

Calculate system enthalpy

```

% Calculate initial enthalpy at the upstream location for isenthalpic
% flash calculation
% Calculate derivative of fugacity coefficients respective to temperature
% for calculating residual enthalpy
[dlndphi_i] = diffphi(bi, b, B, aij, a, A, ac, alph, m, Z, x1, P1, T1,...
    Tc, kij, Ncom);
% Calculate total enthalpy
[H, Hid_i] = enthalpy(dlnphi_i, x1, n1, CP, z, T1, Ncom);
H1 = H;          % Enthalpy of the fluid at point 1
Hsys = H1;       % Assign system enthalpy equal to fluid at point 1

```

```

% Calculate derivative of fugacity coefficients respective to temperature

```

```

% for calculating residual enthalpy
%Input:
%
%bi      : "bi" parameters
%b       : "b" parameters
%B       : "B" parameters
%aij     : "aij" parameters
%a       : "a" parameters
%A       : "A" parameters
%ac      : "ac" parameter
%alpha   : Temperature-dependent parameter in attractive term
%m       : "m" parameter for calculating "alpha"
%Z       : Z factors from EOS's
%x       : Component mole fractions
%P       : Pressure to perform flash
%T       : Temperature to perform flash
%Tc      : Critical temperature
%kij     : Binary interaction parameters
%Ncom    : Number of components
%
%Return:
%Derivative of ln(phi) with respect to temperature
function func1 = diffphi(bi, b, B, aij, a, A, ac, alpha, m, Z, x, P, T,...
    Tc, kij, Ncom)
% The expression for ln(phi) was divided into 4 terms for simplicity.
% The differentiation was done on each term separately.
% 1st term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0 % When there are 2 phases b~=0
            dterm1(k,i) = -bi(i)*Z(k)/(T*b(k));
        else % To avoid division by zero when b = 0
            dterm1(k,i) = 0;
        end
    end
end
% 2nd term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0 % To avoid division by zero
            dterm2(k,i) = (-Z(k)/T+b(k)*(P/(83.1447215*T))/T)/(Z(k) ...
                - b(k)*(P/(83.1447215*T)));
        else
            dterm2(k,i) = 0;
        end
    end
end
% 3rd term
for k = 1:2
    if b(k) ~= 0 % To avoid division by zero
        q(k) = sqrt(2)/(2*83.1447215*b(k));
        bPRminus(k) = (1-sqrt(2))*b(k)*P/83.1447215;
        bPRplus(k) = (1+sqrt(2))*b(k)*P/83.1447215;
        term3_2(k) = log((Z(k)+bPRminus(k)/T)/(Z(k)+bPRplus(k)/T));
        dterm3_2(k) = ((-Z(k)/T-bPRminus(k)/T^2)/(Z(k) ...
            + bPRplus(k)/T)-(Z(k)+bPRminus(k)/T)*...
            (-Z(k)/T-bPRplus(k)/T^2)/(Z(k)+bPRplus(k)/T)^2)*(Z(k) ...
            + bPRplus(k)/T)/(Z(k)+bPRminus(k)/T);
        for i = 1:Ncom
            term3(k,i) = 0;
            dterm3(k,i) = 0;
            for j = 1:Ncom
                if k == 1
                    q2liq(i,j) = q(k)*x(k,j)*(1-kij(i,j))*...
                        sqrt(ac(i)*ac(j));
                    term3liq_1(i,j) = q2liq(i,j)*...

```

```

        sqrt(alpha(i)*alpha(j))/T;
        term3liq(i,j) = term3liq_1(i,j)*term3_2(k);
        term3(k,i) = term3(k,i) + term3liq(i,j);
        dterm3liq_1(i,j) = 1/2*q2liq(i,j)*...
            (-sqrt(alpha(i))*alpha(j)*m(i)/...
            ((T/Tc(i))^5*Tc(i))-alpha(i)*...
            sqrt(alpha(j))*m(j)/((T/Tc(j))^5*Tc(j)))/...
            (sqrt(alpha(i)*alpha(j))*T)-q2liq(i,j)*...
            sqrt(alpha(i)*alpha(j))/T^2;
        dterm3liq(i,j) = term3liq_1(i,j)*dterm3_2(k) ...
            + term3_2(k)*dterm3liq_1(i,j);
        dterm3(k,i) = dterm3(k,i) + dterm3liq(i,j);
    elseif k == 2
        q2vap(i,j) = q(k)*x(k,j)*(1-kij(i,j))*...
            sqrt(ac(i)*ac(j));
        term3vap_1(i,j) = q2vap(i,j)*sqrt(alpha(i)*...
            alpha(j))/T;
        term3vap(i,j) = term3vap_1(i,j)*term3_2(k);
        term3(k,i) = term3(k,i) + term3vap(i,j);
        dterm3vap_1(i,j) = 1/2*q2vap(i,j)*...
            (-sqrt(alpha(i))*alpha(j)*m(i)/...
            ((T/Tc(i))^5*Tc(i))-alpha(i)*...
            sqrt(alpha(j))*m(j)/((T/Tc(j))^5*Tc(j)))/...
            (sqrt(alpha(i)*alpha(j))*T)-q2vap(i,j)*...
            sqrt(alpha(i)*alpha(j))/T^2;
        dterm3vap(i,j) = term3vap_1(i,j)*dterm3_2(k) + ...
            term3_2(k)*dterm3vap_1(i,j);
        dterm3(k,i) = dterm3(k,i) + dterm3vap(i,j);
    end
end
else
    term3(k,i) = 0;
    dterm3(k,i) = 0;
end
end

% 4th term
for k = 1:2
    if b(k) ~= 0 % To avoid division by zero
        r(k) = sqrt(2)/(4*83.1447215*b(k)^2);
        term4_2(k) = term3_2(k);
        dterm4_2(k) = dterm3_2(k);
        for h = 1:Ncom
            term4(k,h) = 0;
            dterm4(k,h) = 0;
            for i = 1:Ncom
                for j = 1:Ncom
                    if k == 1
                        q3liq(i,j) = x(k,i)*x(k,j)*(1-kij(i,j))*...
                            r(k)*sqrt(ac(i)*ac(j));
                        term4liq_1(i,j) = q3liq(i,j)*bi(h)*(alpha(i)*...
                            alpha(j))^(1/2)/T;
                        term4liq(i,j) = term4liq_1(i,j)*term4_2(k);
                        term4(k,h) = term4(k,h) + term4liq(i,j);
                        dterm4liq_1(i,j) = 1/2*q3liq(i,j)*bi(h)*...
                            (-alpha(i)^(1/2)*alpha(j)*m(i)/...
                            ((T/Tc(i))^5*Tc(i))-alpha(i)*...
                            alpha(j)^(1/2)*m(j)/((T/Tc(j))^5*Tc(j)))/...
                            ((alpha(i)*alpha(j))^(1/2)*T)-q3liq(i,j)*...
                            bi(h)*(alpha(i)*alpha(j))^(1/2)/T^2;
                        dterm4liq(i,j) = term4liq_1(i,j)*dterm4_2(k) ...
                            + term4_2(k)*dterm4liq_1(i,j);
                        dterm4(k,h) = dterm4(k,h) + dterm4liq(i,j);
                    else
                        q3vap(i,j) = x(k,i)*x(k,j)*(1-kij(i,j))*r(k)*...

```



```

        sqrt(ac(i)*ac(j));
        term4vap_1(i,j) = q3vap(i,j)*bi(h)*(alpha(i)*...
            alpha(j))^(1/2)/T;
        term4vap(i,j) = term4vap_1(i,j)*term4_2(k);
        term4(k,h) = term4(k,h) + term4vap(i,j);
        dterm4vap_1(i,j) = 1/2*q3vap(i,j)*bi(h)*...
            (-alpha(i)^(1/2)*alpha(j)*m(i)/...
            ((T/Tc(i))^5*Tc(i))-alpha(i)*...
            alpha(j)^(1/2)*m(j)/((T/Tc(j))^5*Tc(j)))/...
            ((alpha(i)*alpha(j))^(1/2)*T)-q3vap(i,j)*...
            bi(h)*(alpha(i)*alpha(j))^(1/2)/T^2;
        dterm4vap(i,j) = term4vap_1(i,j)*dterm4_2(k) ...
            + term4_2(k)*dterm4vap_1(i,j);
        dterm4(k,h) = dterm4(k,h) + dterm4vap(i,j);
    end
end
end
else
    term4(k,h) = 0;
    dterm4(k,h) = 0;
end
end
% Calculate diff(ln(phi)) of each component i in each phase k
for k = 1:2
    for i = 1:Ncom
        dlnphi_i(k,i) = dterm1(k,i)-dterm2(k,i)+dterm3(k,i)-dterm4(k,i);
    end
end
func1 = dlnphi_i;

% Calculate total enthalpy
%Input:
%
% dlnphi_i : Derivative of ln(phi) with respect to temperature of
%            each component i in each phase k
% x         : Component mole fractions
% n         : Phase mole fractions
% CP        : Coefficients for heat capacity correlations
% z         : Feed mole fractions
% T         : Temperature of interest
% Ncom      : Number of components
%
%Return:
% H         : Total enthalpy
% Hid_i     : Ideal-gas enthalpy of each component
function [func1, func2] = enthalpy(dlnphi_i, x, n, CP, z, T, Ncom)
% Calculate residual enthalpy
Hres = 0;
dlnphi_k = [0, 0];
for k = 1:2
    for i = 1:Ncom
        dlnphi_k(k) = dlnphi_k(k) + x(k,i)*dlnphi_i(k,i);
    end
    % Calculate residual enthalpy of each phase
    Hres_k(k) = -8.31447215*T^2*dlnphi_k(k);
    % Calculate total residual enthalpy
    Hres = Hres + n(k)*Hres_k(k);
end
% Calculate enthalpy of ideal gas
Hid = 0;
for i = 1:Ncom
    % Calculate ideal-gas enthalpy for each component using correlations

```

```

        Hid_i(i) = CP(1,i)*(T-273.15) + 1/2*CP(2,i)*(T^2-273.15^2) ...
        + 1/3*CP(3,i)*(T^3-273.15^3) + 1/4*CP(4,i)*(T^4-273.15^4);
        % Calculate total ideal-gase enthalpy
        Hid = Hid + z(i)*Hid_i(i);
    end
    % Calculate total enthalpy
    H = Hid + Hres;
    func1 = H;
    func2 = Hid_i;

```

Isoenthalpic flash algorithm

```

% First isenthalpic flash iteration
% Evaluate at local conditions
ns = 0; % Assume no precipitation
T2 = T1; % Assign T2 with intermediate temperature (T1)
flash % Isothermal flash calculations at local pressure (P2)
      % and temperature (T2)

% Calculate enthalpy
% Calculate first and second derivatives of fugacity coefficients
% respective to temperature for calculating residual enthalpy and
% heat capacity
[dlndphi_i, ddlnphi_i] = diffdiffphi(bi, b, B, aij, a, A, ac, alph, m, Z,...
    x, P2, T2, Tc, kij, Ncom);
% Calculate total enthalpy
[H, Hid_i] = enthalpy(dlnphi_i, x, n, CP, z, T2, Ncom);
% Calculate the difference between the calculated enthalpy and the
% system enthalpy at the upstream point
g_0 = H - Hsys;
% Calculate total specific heat capacity -- Cp = sum(n(k)Cp(k))
Cp = Heatcapacity(dlnphi_i, ddlnphi_i, CP, T2, z, x, n, Ncom);
% Update temperature for the first isenthalpic flash iteration
T_temp = Initialtemp(Cp, T2, g_0);
% Update local temperature (T2) with the updated temperature
T2 = T_temp(1);
% Index for isenthalpic flash iterations
iter = 1;
Hnew = 0;

```

Isothermal flash algorithm

```

% Isothermal flash calculations at local pressure (P2) and temperature (T2)
% Above bubblepoint pressure, there is only liquid
if P2>=Psat;
    x(1,:) = z; % Component mole fractions in liquid phase =
                % feed mole fractions
    x(2,:) = 0; % No gas phase
    phase = 1; % Number of phase = 1
    n(2) = 0; % Gas phase fraction = 0
    n(1) = 1-n(2)-ns; % Liquid phase fraction = 1
    n(3) = ns; % Solid phase fraction = 0
% Below bubblepoint pressure, there are two phases
else
    % Calculate initial K values using Wilson equation
    K = initialK(Pc, P2, w, Tc, T2, Ncom);
    % result is the indicator for equilibrium
    result = 0;

```

```

while result == 0
    % Calculate vapor mole fraction
    n = Findnv(z, K, ns, Ncom);
    % Calculate mole fractions of each component in liquid and
    % gas phases
    x = xi(z, n, ns, K, Ncom);
    % Calculate EOS parameters for liquid and gas phases using random
    % mixing rules
    [aij, a, A, ac, alph, m] = coeff_a(w, Tc, Pc, T2, P2, kij, x,...
        Ncom);
    [bi, b, B] = coeff_b(w, Tc, Pc, T2, P2, kij, x, Ncom);
    % Solve Peng-Robinson EOS's for Z factors
    Z = PR(A, B);
    % Calculate component fugacities
    f = fugacity_z(bi, b, B, aij, a, A, Z, x, P2, Ncom);
    % Check for equilibrium -- equal fugacities in both phases
    result = check(f, Ncom);
    % If the equilibrium criteria are not satisfied, update K values
    for i = 1:Ncom-1
        f_ratio(i) = f(1,i)/f(2,i);
        % Calculate new K values -- K value of the last component is
        % always zero
        K(i) = K(i)*f_ratio(i);
    end
    end
    % Check for number of phases again
    [n, x, phase] = nv_boundary(n, z, x, Ncom);
end

% If there is only one phase
% (pressure > saturation pressure or vapor phase fraction > 1 or < 0)
if phase == 1
    % Calculate fugacities knowing that there is only one phase
    % Calculate EOS parameters for liquid and gas phases using random
    % mixing rules
    [aij, a, A, ac, alph, m] = coeff_a(w, Tc, Pc, T2, P2, kij, x, Ncom);
    [bi, b, B] = coeff_b(w, Tc, Pc, T2, P2, kij, x, Ncom);
    % Solve Peng-Robinson EOS's for Z factors
    Z = PR(A, B);
    % Calculate component fugacities
    f = fugacity_z(bi, b, B, aij, a, A, Z, x, P2, Ncom);
end

end

% Calculate initial K values using Wilson equation
%Input:
%
%Pc      : Critical pressures
%P       : Pressure to perform flash
%w       : Acentric factors
%Tc      : Critical temperature
%T       : Temperature to perform flash
%Ncom    : Number of components
%
%Return:
%Initial K values for all components
function func = initialK(Pc, P, w, Tc, T, Ncom)
for i = 1:Ncom-1
    % Calculate K values
    K(i) = (Pc(i)/P)*exp(5.37*(1+w(i))*(1-Tc(i)/T));
end
% K value of the last component is always zero (no asphaltene in gas phase)
K(Ncom) = 0;
func = K;

```

```

% Calculate vapor mole fraction using Newton-Raphson algorithm
%Input:
%
%z      :   Feed mole fractions
%K      :   K values
%ns     :   Solid mole fraction
%Ncom   :   Number of components
%
%Return:
%Phase mole fractions
function func = Findnv(z, K, ns, Ncom)
% Equation relating nv and K-values -- Rachford-Rice equation
nvfunc = 0;
% Derivative of the nv function with respect to nv
diff_nvfunc = 0;
% Initial guessed value of nv
nv_temp(1) = 0.4;
% For checking for convergence
test = 1;
% Repeat the calculation if nv_temp(n) is not equal to nv_temp(n+1)
% in iteration n+1
while test >= 1.0e-12
    % Calculate nvfunc using nv_temp(n)
    for i = 1:Ncom-1
        nvfunc_i(i) = z(i)*(K(i)-1)/(nv_temp(1)*(K(i)-1)-ns+1);
        nvfunc = nvfunc + nvfunc_i(i);
    end
    nvfunc = nvfunc - z(Ncom)/((1-nv_temp(1))+ns*((1-nv_temp(1)-ns)/...
        (z(Ncom)-ns)-1));
    % Calculate diff_nvfunc using nv_temp(n)
    for i = 1:Ncom-1
        diff_nvfunc_i(i) = -z(i)*(K(i)-1)^2/(nv_temp(1)*(K(i)-1)-ns+1)^2;
        diff_nvfunc = diff_nvfunc + diff_nvfunc_i(i);
    end
    diff_Ncom = (-z(Ncom)*(-1-ns/(z(Ncom)-ns)))/(1-nv_temp(1)+ns*...
        ((1-nv_temp(1)-ns)/(z(Ncom)-ns)-1))^2);
    diff_nvfunc = diff_nvfunc - diff_Ncom;
    % Calculate nv_temp(n+1)
    % To avoid division by zero
    if diff_nvfunc == 0.0
        nv_temp(2) = nv_temp(1);
    else
        nv_temp(2) = nv_temp(1) - nvfunc/diff_nvfunc;
    end
    % Check if nv_temp(n) equals nv_temp(n+1)
    test = (1-nv_temp(1)/nv_temp(2))^2;
    % If not converged, replace nv_temp(n) with nv_temp(n+1)
    % for the next iteration
    nv_temp(1) = nv_temp(2);
    nvfunc = 0;
    diff_nvfunc = 0;
end
nv = nv_temp(1);
n(1) = 1-nv-ns;
n(2) = nv;
n(3) = ns;
func = n;
% Obtain converged value
% Liquid phase mole fraction
% Vapor phase mole fraction
% Solid phase mole fraction

```

```

% Calculate mole fractions of each component in liquid and gas phases
%Input:

```

```

%
%z      :   Feed mole fractions
%n      :   Phase mole fractions
%K      :   K values
%Ncom   :   Number of components
%
%Return:
%Component mole fractions
function func = xi(z, n, K, Ncom)
for k = 1:2
    if k == 1                                % For components in liquid phase
        for i = 1:Ncom-1
            x(k,i) = z(i)/(n(2)*(K(i)-1)-n(3)+1);
        end
        x(k,Ncom) = (z(Ncom)-n(3))/n(1);
    else                                      % For components in gas phase
        for i = 1:Ncom
            x(k,i) = z(i)*K(i)/(n(2)*(K(i)-1)-n(3)+1);
        end
    end
end
func = x;

% Calculate EOS "a" parameters for liquid and gas phases using random
% mixing rules
%Input:
%
%w      :   Acentric factors
%Tc     :   Critical temperatures
%Pc     :   Critical pressures
%T      :   Temperature to perform flash
%P      :   Pressure to perform flash
%kij    :   Binary interaction parameters
%x      :   Component mole fractions
%Ncom   :   Number of components
%
%Return:
%aij    :   Attraction parameters for component i and j
%a      :   Mixture "a" parameters
%A      :   Mixture "A" parameters
%ac     :   "ac" parameter
%alpha  :   Temperature-dependent parameter
%m      :   "m" parameter for calculating "alpha"
function [func1, func2, func3, func4, func5, func6] = coeff(w, Tc, Pc,...
    T, P, kij, x, Ncom)
% Calculate "a" parameters for each component
for i = 1:Ncom
    if w(i) <= 0.49
        m(i) = 0.37464 + 1.5422*w(i) - 0.26992*w(i)^2;
    else
        m(i) = 0.3796 + 1.485*w(i) - 0.1644*w(i)^2 + 0.01667*w(i)^3;
    end
    % Temperature-dependent parameter
    alpha(i) = (1+m(i)*(1-(T/Tc(i))^0.5))^2;
    % Omega a = 0.457235
    ac(i) = 0.457235*(83.1447215*Tc(i))^2/Pc(i);
    ai(i) = alpha(i)*ac(i);
end
% Calculate "aij" and "a" using the random mixing rules
for k = 1:2
    a(k) = 0;
    for i = 1:Ncom
        sumj(k,i) = 0;
        for j = 1:Ncom

```

```

        aij(i,j) = (ai(i)*ai(j))^0.5*(1-kij(i,j));
        sumj(k,i) = sumj(k,i) + x(k,j)*aij(i,j);
    end
    a(k) = a(k) + x(k,i)*sumj(k,i);
end
end
% Calculate "A" parameters for use in EOS
for k = 1:2          % 1 for the liquid phase and 2 for the vapor phase
    A(k) = a(k)*P/(83.1447215*T)^2;
end
func1 = aij;
func2 = a;
func3 = A;
func4 = ac;
func5 = alpha;
func6 = m;

% Calculate EOS "b" parameters for liquid and gas phases using random
% mixing rules
%Input:
%
%w      : Acentric factors
%Tc     : Critical temperatures
%Pc     : Critical pressures
%T      : Temperature to perform flash
%P      : Pressure to perform flash
%kij    : Binary interaction parameters
%x      : Component mole fractions
%Ncom   : Number of components
%
%Return:
%bi     : "b" parameters for component i
%b      : Mixture "b" parameters
%B      : Mixture "B" parameters
function [func1, func2, func3] = coeff_b(w, Tc, Pc, T, P, kij, x, Ncom)
% Calculate "bi" and "b" parameters using the random mixing rules
for k = 1:2
    b(k) = 0;
    for i = 1:Ncom
        bi(i) = 0.077796*83.1447215*Tc(i)/Pc(i);          % Omega b = 0.077796
        b(k) = b(k) + x(k,i)*bi(i);
    end
end
% Calculate "B" parameters for use in EOS
for k = 1:2          % 1 for the liquid phase and 2 for the vapor phase
    B(k) = b(k)*P/(83.1447215*T);
end
func1 = bi;
func2 = b;
func3 = B;

% Solve Peng-Robinson EOS's for Z factors
%Input:
%
%A      : EOS "A" parameters
%B      : EOS "B" parameters
%
%Return:
%Roots from solving Peng-Robinson EOS's for liquid and gas phases

```

```

function func = PR(A, B)
for k = 1:2
    EOS = [1 -(1-B(k)) +(A(k)-2*B(k)-3*B(k)^2) -(A(k)*B(k)-B(k)^2-B(k)^3)];
    r = roots(EOS); % Solve EOS
    % Check the number of real roots
    imagcount = 0;
    if abs(imag(r(1))) < 10e-4
        imagcount = imagcount;
    else
        imagcount = imagcount + 1;
    end
    if abs(imag(r(2))) < 10e-4
        imagcount = imagcount;
    else
        imagcount = imagcount + 1;
    end
    if abs(imag(r(3))) < 10e-4
        imagcount = imagcount;
    else
        imagcount = imagcount + 1;
    end
    % If there is only one real root, select the real root
    if imagcount == 2
        if abs(imag(r(1))) < 10e-4
            Z(k) = r(1);
        elseif abs(imag(r(2))) < 10e-4
            Z(k) = r(2);
        elseif abs(imag(r(3))) < 10e-4
            Z(k) = r(3);
        end
    elseif imagcount == 0 % If there are three real roots
        if k == 1 % For liquid phase, select the smallest root
            if r(1) < r(2)
                if r(1) < r(3)
                    Z(k) = r(1);
                else
                    Z(k) = r(3);
                end
            else
                if r(2) < r(3)
                    Z(k) = r(2);
                else
                    Z(k) = r(3);
                end
            end
        else % For gas phase, select the largest root
            if r(1) > r(2)
                if r(1) > r(3)
                    Z(k) = r(1);
                else
                    Z(k) = r(3);
                end
            else
                if r(2) > r(3)
                    Z(k) = r(2);
                else
                    Z(k) = r(3);
                end
            end
        end
    end
end
end
r = 0;
end
func = Z;

% Calculate component fugacities
%Input:
%
```

```

%bi      : "bi" parameters
%b       : "b" parameters
%B       : "B" parameters
%aij     : "aij" parameters
%a       : "a" parameters
%A       : "A" parameters
%Z       : Z factors from EOS's
%x       : Component mole fractions
%P       : Pressure to perform flash
%Ncom    : Number of components
%
%Return:
%Component fugacities
function func = fugacity_z(bi, b, B, aij, a, A, Z, x, P, Ncom)
% Calculate fugacity coefficients (ln phi). The expression for fugacity
% coefficient was divided into 4 terms for simplicity.
% 1st term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0 % To avoid division by zero
            fterm1(k,i) = bi(i)*(Z(k)-1)/b(k);
        else
            fterm1(k,i) = 0;
        end
    end
end
% 2nd term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0 % To avoid division by zero
            fterm2(k,i) = log(Z(k)-B(k));
        else
            fterm2(k,i) = 0;
        end
    end
end
% 3rd term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0 % To avoid division by zero
            fterm3(k,i) = 0;
            temp = 0;
            for j = 1:Ncom
                temp = A(k)/a(k)*1/2*2^0.5*x(k,j)*aij(i,j)*log((Z(k)...
                    + (1-2^(1/2))*B(k))/(Z(k)+(1+2^(1/2))*B(k)));
            end
            fterm3(k,i) = fterm3(k,i) + temp;
        else
            fterm3(k,i) = 0;
        end
    end
end
% 4th term
for k = 1:2
    for h = 1:Ncom
        fterm4(k,h) = 0;
        temp = 0;
        if b(k) ~= 0 % To avoid division by zero
            temp = 1/4*A(k)/b(k)/B(k)*bi(h)*2^0.5*log((Z(k)...
                + (1-2^(1/2))*B(k))/(Z(k)+(1+2^(1/2))*B(k)));
            fterm4(k,h) = fterm4(k,h) + temp;
        else
            term4(k,h) = 0;
        end
    end
end
% Calculate fugacity coefficients (ln_phi) and fugacities (f)
ln_phi = 0;
for k = 1:2
    for i = 1:Ncom

```



```

        ln_phi(k,i) = fterm1(k,i)-fterm2(k,i)+fterm3(k,i)-fterm4(k,i);
        % Calculate fugacity coefficients of all components
        phi(k,i) = exp(ln_phi(k,i));
        % Calculate fugacities -- f = phi*x*P
        f(k,i) = phi(k,i)*x(k,i)*P;
    end
end
func = f;

% Check for equilibrium -- equal fugacities in both phases
%Input:
%
%f1      :   Component fugacities
%Ncom    :   Number of components
%
%Return:
%result  :   Indication of equilibrium
%ch      :   Convergence value
function [func1 func2] = check(f1, Ncom)
ch = 0;
result = 0;
for i = 1:Ncom-1
    % To avoid division by zero
    if f1(2,i) == 0
        if f1(1,i) == 0
            ch0 = 0;
        else
            f1(2,i) = 1e-100;
        end
    else
        % Compare fugacities in vapor and liquid phases
        ch0 = (1-f1(1,i)/f1(2,i))^2;
    end
    ch = ch + ch0;
end
% If the total difference in fugacities is less than a tolerance number,
% equilibrium criteria are satisfied
if ch < 1.0e-12
    result = 1;
end
func1 = result;
func2 = ch;

% Check for number of phases from the calculated nv
%Input:
%
%n       :   Phase mole fractions
%z       :   Feed mole fractions
%x       :   Component mole fractions
%Ncom    :   Number of components
%
%Return:
%n       :   Phase mole fractions
%x       :   Component mole fractions
%phase   :   Number of phases
function [func1, func2, func3] = nv_boundary(n, z, x, Ncom)
phase = 2;
if n(2) > 0
    % If nv > 0
    if n(2) < 1
        % If nv < 1, there is gas phase
        n(2) = n(2);
    end
end

```

```

else
    phase = 1;
    n(2) = 1;
    x(2,:) = z;
    x(1,:) = 0;
    % If nv > 1, there is only gas phase
    % Number of phase = 1
    % Gas phase fraction = 1
    % Component mole fractions in gas phase =
    % feed mole fractions
    % Component mole fractions in liquid phase
    % = 0
end
else
    phase = 1;
    n(2) = 0;
    x(1,:) = z;
    x(2,:) = 0;
    % If nv < 0, there is only liquid phase
    % Number of phase = 1
    % Gas phase fraction = 0
    % Component mole fractions in liquid phase
    % = feed mole fractions
    % Component mole fractions in gas phase = 0
end
n(1) = 1-n(2)-ns;
n(2) = n(2);
n(3) = n(3);
func1 = n;
func2 = x;
func3 = phase;
% Liquid mole fraction
% Gas mole fraction
% Solid mole fraction

```

Calculate enthalpy and first temperature update

```

% Calculate first and second derivatives of fugacity coefficients
% respective to temperature for calculating residual enthalpy and
% heat capacity
%Input:
%
%bi      : "bi" parameters
%b       : "b" parameters
%B       : "B" parameters
%aij     : "aij" parameters
%a       : "a" parameters
%A       : "A" parameters
%ac      : "ac" parameter
%alpha   : Temperature-dependent parameter in attractive term
%m       : "m" parameter for calculating "alpha"
%Z       : Z factors from EOS's
%x       : Component mole fractions
%P       : Pressure to perform flash
%T       : Temperature to perform flash
%Tc      : Critical temperature
%kij     : Binary interaction parameters
%Ncom    : Number of components
%
%Return:
%dlndphi_i : First derivative of ln(phi) with respect to temperature
%ddlnphi_i : Second derivative of ln(phi) with respect to temperature
function [func1, func2] = diffdiffphi(bi, b, B, aij, a, A, ac, alpha, m,...
    Z, x, P, T, Tc, kij, Ncom)
% The expression for ln(phi) was divided into 4 terms for simplicity.
% The differentiation was done on each term separately.
% 1st term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0
            % When there are 2 phases
            dterm1(k,i) = -bi(i)*Z(k)/(T*b(k));
            ddterm1(k,i) = 2*bi(i)*Z(k)/(T^2*b(k));
        else
            % To avoid division by zero when b = 0
            dterm1(k,i) = 0;
            ddterm1(k,i) = 0;
        end
    end
end

```

```

end
end
end
% 2nd term
for k = 1:2
    for i = 1:Ncom
        if b(k) ~= 0 % To avoid division by zero
            dterm2(k,i) = (-Z(k)/T+b(k)*(P/(83.1447215*T))/T)/...
                (Z(k)-b(k)*(P/(83.1447215*T)));
            ddterm2(k,i) = (2*Z(k)/T^2-2*b(k)*(P/(83.1447215*T))/T^2)/...
                (Z(k)-b(k)*(P/(83.1447215*T)))-(-Z(k)/T+b(k)*...
                (P/(83.1447215*T))/T^2/(Z(k)-b(k)*(P/(83.1447215*T)))^2;
        else
            dterm2(k,i) = 0;
            ddterm2(k,i) = 0;
        end
    end
end
% 3rd term
for k = 1:2
    if b(k) ~= 0 % To avoid division by zero
        q(k) = sqrt(2)/(2*83.1447215*b(k));
        bPRminus(k) = (1-sqrt(2))*b(k)*P/83.1447215;
        bPRplus(k) = (1+sqrt(2))*b(k)*P/83.1447215;
        term3_2(k) = log((Z(k)+bPRminus(k)/T)/(Z(k)+bPRplus(k)/T));
        dterm3_2(k) = ((-Z(k)/T-bPRminus(k)/T^2)/(Z(k)+bPRplus(k)/T)...
            -(Z(k)+bPRminus(k)/T)*(-Z(k)/T-bPRplus(k)/T^2)/...
            (Z(k)+bPRplus(k)/T^2)*(Z(k)+bPRplus(k)/T)/...
            (Z(k)+bPRminus(k)/T);
        ddterm3_2(k) = ((2*Z(k)/T^2+2*bPRminus(k)/T^3)/...
            (Z(k)+bPRplus(k)/T)-2*(-Z(k)/T-bPRminus(k)/T^2)*...
            (-Z(k)/T-bPRplus(k)/T^2)/(Z(k)+bPRplus(k)/T^2+...
            (Z(k)+bPRminus(k)/T)*(-Z(k)/T-bPRplus(k)/T^2)/(Z(k)+...
            bPRplus(k)/T^3-(Z(k)+bPRminus(k)/T)*(2*Z(k)/T^2+...
            2*bPRplus(k)/T^3)/(Z(k)+bPRplus(k)/T^2)*(Z(k)+...
            bPRplus(k)/T)/(Z(k)+bPRminus(k)/T)-((-Z(k)/...
            T-bPRminus(k)/T^2)/(Z(k)+bPRplus(k)/T)-(-Z(k)+...
            bPRminus(k)/T)*(-Z(k)/T-bPRplus(k)/T^2)/(Z(k)+...
            bPRplus(k)/T^2)*(Z(k)+bPRplus(k)/T)*(-Z(k)/...
            T-bPRminus(k)/T^2)/(Z(k)+bPRminus(k)/T^2+...
            ((-Z(k)/T-bPRminus(k)/T^2)/(Z(k)+bPRplus(k)/T)...
            -(Z(k)+bPRminus(k)/T)*(-Z(k)/T-bPRplus(k)/T^2)/...
            (Z(k)+bPRplus(k)/T^2)*(-Z(k)/T-bPRplus(k)/T^2)/(Z(k)+...
            bPRminus(k)/T);
        for i = 1:Ncom
            term3(k,i) = 0;
            dterm3(k,i) = 0;
            ddterm3(k,i) = 0;
            for j = 1:Ncom
                if k == 1
                    q2liq(i,j) = q(k)*x(k,j)*(1-kij(i,j))*...
                        sqrt(ac(i)*ac(j));
                    term3liq_1(i,j) = q2liq(i,j)*sqrt(alpha(i)*...
                        alpha(j))/T;
                    term3liq(i,j) = term3liq_1(i,j)*term3_2(k);
                    term3(k,i) = term3(k,i) + term3liq(i,j);
                    dterm3liq_1(i,j) = 1/2*q2liq(i,j)*...
                        (-sqrt(alpha(i))*alpha(j)*m(i)/...
                        ((T/Tc(i))^0.5*Tc(i))-alpha(i)*...
                        sqrt(alpha(j))*m(j)/((T/Tc(j))^0.5*Tc(j)))/...
                        (sqrt(alpha(i)*alpha(j))*T)-q2liq(i,j)*...
                        sqrt(alpha(i)*alpha(j))/T^2;
                    dterm3liq(i,j) = term3liq_1(i,j)*dterm3_2(k) ...
                        + term3_2(k)*dterm3liq_1(i,j);
                    dterm3(k,i) = dterm3(k,i) + dterm3liq(i,j);
                end
            end
        end
    end
end

```

```

ddterm3liq_1(i,j) = -1/4*q2liq(i,j)*...
    (-sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i)).5*Tc(i))-alpha(i)*...
    sqrt(alpha(j))*m(j)/((T/Tc(j)).5*Tc(j)))2/...
    ((alpha(i)*alpha(j))(3/2)*T-q2liq(i,j)*...
    (-sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i)).5*Tc(i))-alpha(i)*...
    sqrt(alpha(j))*m(j)/((T/Tc(j)).5*Tc(j)))/...
    (sqrt(alpha(i)*alpha(j))*T2+...
    1/2*q2liq(i,j)*(.50*m(i)2*alpha(j)/...
    ((T/Tc(i))1.0*Tc(i)2+2.00*sqrt(alpha(i))*...
    sqrt(alpha(j))*m(i)*m(j)/((T/Tc(i)).5*...
    Tc(i)*(T/Tc(j)).5*Tc(j))+...
    .50*sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i))1.5*Tc(i)2+50*alpha(i))*...
    m(j)2/((T/Tc(j))1.0*Tc(j)2+50*alpha(i))*...
    sqrt(alpha(j))*m(j)/...
    ((T/Tc(j))1.5*Tc(j)2)/(sqrt(alpha(i))*...
    alpha(j))*T)+2*q2liq(i,j)*sqrt(alpha(i))*...
    alpha(j))/T3;
ddterm3liq(i,j) = term3liq_1(i,j)*ddterm3_2(k) ...
    + term3_2(k)*ddterm3liq_1(i,j) ...
    + 2*dterm3liq_1(i,j)*dterm3_2(k);
ddterm3(k,i) = ddterm3(k,i) + ddterm3liq(i,j);
elseif k == 2
q2vap(i,j) = q(k)*x(k,j)*(1-kij(i,j))*...
    sqrt(ac(i)*ac(j));
term3vap_1(i,j) = q2vap(i,j)*sqrt(alpha(i))*...
    alpha(j))/T;
term3vap(i,j) = term3vap_1(i,j)*term3_2(k);
term3(k,i) = term3(k,i) + term3vap(i,j);
dterm3vap_1(i,j) = 1/2*q2vap(i,j)*...
    (-sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i)).5*Tc(i))-alpha(i)*...
    sqrt(alpha(j))*m(j)/((T/Tc(j)).5*Tc(j)))/...
    (sqrt(alpha(i)*alpha(j))*T)-q2vap(i,j)*...
    sqrt(alpha(i)*alpha(j))/T2;
dterm3vap(i,j) = term3vap_1(i,j)*dterm3_2(k) ...
    + term3_2(k)*dterm3vap_1(i,j);
dterm3(k,i) = dterm3(k,i) + dterm3vap(i,j);
ddterm3vap_1(i,j) = -1/4*q2vap(i,j)*...
    (-sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i)).5*Tc(i))-alpha(i)*...
    sqrt(alpha(j))*m(j)/((T/Tc(j)).5*...
    Tc(j)))2/((alpha(i)*alpha(j))(3/2)*T...
    -q2vap(i,j)*(-sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i)).5*Tc(i))-alpha(i)*...
    sqrt(alpha(j))*m(j)/((T/Tc(j)).5*Tc(j)))/...
    (sqrt(alpha(i)*alpha(j))*T2+...
    1/2*q2vap(i,j)*(.50*m(i)2*alpha(j)/...
    ((T/Tc(i))1.0*Tc(i)2+2.00*sqrt(alpha(i))*...
    sqrt(alpha(j))*m(i)*m(j)/((T/Tc(i)).5*...
    Tc(i)*(T/Tc(j)).5*Tc(j))+50*...
    sqrt(alpha(i))*alpha(j)*m(i)/...
    ((T/Tc(i))1.5*Tc(i)2+50*alpha(i))*...
    m(j)2/((T/Tc(j))1.0*Tc(j)2+50*...
    alpha(i)*sqrt(alpha(j))*m(j)/...
    ((T/Tc(j))1.5*Tc(j)2)/(sqrt(alpha(i))*...
    alpha(j))*T)+2*q2vap(i,j)*sqrt(alpha(i))*...
    alpha(j))/T3;
ddterm3vap(i,j) = term3vap_1(i,j)*ddterm3_2(k) ...
    + term3_2(k)*ddterm3vap_1(i,j) ...

```

```

        + 2*dterm3vap_1(i,j)*dterm3_2(k);
        dterm3(k,i) = dterm3(k,i) + ddterm3vap(i,j);
    end
end
else
    term3(k,i) = 0;
    dterm3(k,i) = 0;
    ddterm3(k,i) = 0;
end
end

% 4th term
for k = 1:2
    if b(k) ~= 0 % To avoid division by zero
        r(k) = sqrt(2)/(4*83.1447215*b(k)^2);
        term4_2(k) = term3_2(k);
        dterm4_2(k) = dterm3_2(k);
        ddterm4_2(k) = ddterm3_2(k);
        for h = 1:Ncom
            term4(k,h) = 0;
            dterm4(k,h) = 0;
            ddterm4(k,h) = 0;
            for i = 1:Ncom
                for j = 1:Ncom
                    if k == 1
                        q3liq(i,j) = x(k,i)*x(k,j)*(1-kij(i,j))*r(k)*...
                            sqrt(ac(i)*ac(j));
                        term4liq_1(i,j) = q3liq(i,j)*bi(h)*(alpha(i)*...
                            alpha(j))^(1/2)/T;
                        term4liq(i,j) = term4liq_1(i,j)*term4_2(k);
                        term4(k,h) = term4(k,h) + term4liq(i,j);
                        dterm4liq_1(i,j) = 1/2*q3liq(i,j)*bi(h)*...
                            (-alpha(i)^(1/2)*alpha(j)*m(i)/...
                                ((T/Tc(i))^0.5*Tc(i))-alpha(i)*...
                                alpha(j)^(1/2)*m(j)/((T/Tc(j))^0.5*Tc(j)))/...
                                ((alpha(i)*alpha(j))^(1/2)*T)-q3liq(i,j)*...
                                bi(h)*(alpha(i)*alpha(j))^(1/2)/T^2;
                        dterm4liq(i,j) = term4liq_1(i,j)*dterm4_2(k) ...
                            + term4_2(k)*dterm4liq_1(i,j);
                        dterm4(k,h) = dterm4(k,h) + dterm4liq(i,j);
                        ddterm4liq_1(i,j) = -1/4*q3liq(i,j)*bi(h)*...
                            (-alpha(i)^(1/2)*alpha(j)*m(i)/...
                                ((T/Tc(i))^0.5*Tc(i))-alpha(i)*...
                                alpha(j)^(1/2)*m(j)/((T/Tc(j))^0.5*Tc(j))-...
                                Tc(j))^2/((alpha(i)*alpha(j))^(3/2)*T)-...
                                q3liq(i,j)*bi(h)*(-alpha(i)^(1/2)*alpha(j)*...
                                m(i)/((T/Tc(i))^0.5*Tc(i))-alpha(i)*...
                                alpha(j)^(1/2)*m(j)/((T/Tc(j))^0.5*Tc(j)))/...
                                ((alpha(i)*alpha(j))^(1/2)*T^2)+1/2*...
                                q3liq(i,j)*bi(h)*(.50*m(i)^2*alpha(j)/...
                                ((T/Tc(i))^1.0*Tc(i)^2)+2.00*alpha(i)^(1/2)*...
                                alpha(j)^(1/2)*m(i)*m(j)/((T/Tc(i))^1.5*Tc(i)^2) ...
                                + .50*alpha(i)*m(j)^2/((T/Tc(j))^1.0*...
                                Tc(j)^2)+.50*alpha(i)*alpha(j)^(1/2)*m(j)/...
                                ((T/Tc(j))^1.5*Tc(j)^2))/((alpha(i)*...
                                alpha(j))^(1/2)*T)+2*q3liq(i,j)*bi(h)*...
                                (alpha(i)*alpha(j))^(1/2)/T^3;
                        ddterm4liq(i,j) = term4liq_1(i,j)*ddterm4_2(k) ...
                            + term4_2(k)*ddterm4liq_1(i,j) ...
                            + 2*dterm4liq_1(i,j)*dterm4_2(k);
                        ddterm4(k,h) = ddterm4(k,h) + ddterm4liq(i,j);
                    else
                        q3vap(i,j) = x(k,i)*x(k,j)*(1-kij(i,j))*r(k)*...
                            sqrt(ac(i)*ac(j));

```

```

term4vap_1(i,j) = q3vap(i,j)*bi(h)*(alpha(i)*...
    alpha(j))^(1/2)/T;
term4vap(i,j) = term4vap_1(i,j)*term4_2(k);
term4(k,h) = term4(k,h) + term4vap(i,j);
dterm4vap_1(i,j) = 1/2*q3vap(i,j)*bi(h)*...
    (-alpha(i)^(1/2)*alpha(j)*m(i)/...
    ((T/Tc(i))^5*Tc(i))-alpha(i)*...
    alpha(j)^(1/2)*m(j)/((T/Tc(j))^5*Tc(j)))/...
    ((alpha(i)*alpha(j))^(1/2)*T)-q3vap(i,j)*...
    bi(h)*(alpha(i)*alpha(j))^(1/2)/T^2;
dterm4vap(i,j) = term4vap_1(i,j)*dterm4_2(k) ...
    + term4_2(k)*dterm4vap_1(i,j);
dterm4(k,h) = dterm4(k,h) + dterm4vap(i,j);
ddterm4vap_1(i,j) = -1/4*q3vap(i,j)*bi(h)*...
    (-alpha(i)^(1/2)*alpha(j)*m(i)/...
    ((T/Tc(i))^5*Tc(i))-alpha(i)*...
    alpha(j)^(1/2)*m(j)/((T/Tc(j))^5*...
    Tc(j)))^2/((alpha(i)*alpha(j))^(3/2)*T)...
    - q3vap(i,j)*bi(h)*(-alpha(i)^(1/2)*...
    alpha(j)*m(i)/((T/Tc(i))^5*Tc(i))...
    - alpha(i)*alpha(j)^(1/2)*m(j)/...
    ((T/Tc(j))^5*Tc(j)))/((alpha(i)*...
    alpha(j))^(1/2)*T^2)+1/2*q3vap(i,j)*bi(h)*...
    (.50*m(i)^2*alpha(j)/((T/Tc(i))^1.0*Tc(i)^2)...
    + 2.00*alpha(i)^(1/2)*alpha(j)^(1/2)*m(i)*...
    m(j)/((T/Tc(i))^5*Tc(i)*(T/Tc(j))^5*Tc(j))...
    + .50*alpha(i)*alpha(j)^(1/2)*m(i)/...
    ((T/Tc(i))^1.5*Tc(i)^2)+.50*alpha(i)*m(j)^2/...
    ((T/Tc(j))^1.0*Tc(j)^2)+.50*alpha(i)*...
    alpha(j)^(1/2)*m(j)/...
    ((T/Tc(j))^1.5*Tc(j)^2))/((alpha(i)*...
    alpha(j))^(1/2)*T)+2*q3vap(i,j)*bi(h)*...
    (alpha(i)*alpha(j))^(1/2)/T^3;
ddterm4vap(i,j) = term4vap_1(i,j)*ddterm4_2(k) ...
    + term4_2(k)*ddterm4vap_1(i,j) ...
    + 2*dterm4vap_1(i,j)*dterm4_2(k);
ddterm4(k,h) = ddterm4(k,h) + ddterm4vap(i,j);
end
end
end
else
    term4(k,h) = 0;
    dterm4(k,h) = 0;
    ddterm4(k,h) = 0;
end
end
% Calculate diff(ln(phi)) and diffdiff(ln(phi)) of each component i in
% each phase k
for k = 1:2
    for i = 1:Ncom
        dlnphi_i(k,i) = dterm1(k,i)-dterm2(k,i)+dterm3(k,i)-dterm4(k,i);
        ddlnphi_i(k,i) = ddterm1(k,i)-ddterm2(k,i)+ddterm3(k,i)...
            - ddterm4(k,i);
    end
end
func1 = dlnphi_i;
func2 = ddlnphi_i;

% Calculate total enthalpy
%Input:

```

```

%
%dlndphi_i : Derivative of ln(phi) with respect to temperature of
%            each component i in each phase k
%x          : Component mole fractions
%n          : Phase mole fractions
%CP         : Coefficients for heat capacity correlations
%z          : Feed mole fractions
%T          : Temperature of interest
%Ncom       : Number of components
%
%Return:
%H          : Total enthalpy
%Hid_i      : Ideal-gas enthalpy of each component
function [func1, func2] = enthalpy(dlnphi_i, x, n, CP, z, T, Ncom)
% Calculate residual enthalpy
Hres = 0;
dlnphi_k = [0, 0];
for k = 1:2
    for i = 1:Ncom
        dlnphi_k(k) = dlnphi_k(k) + x(k,i)*dlnphi_i(k,i);
    end
    % Calculate residual enthalpy of each phase
    Hres_k(k) = -8.31447215*T^2*dlnphi_k(k);
    % Calculate total residual enthalpy
    Hres = Hres + n(k)*Hres_k(k);
end
% Calculate enthalpy of ideal gas
Hid = 0;
for i = 1:Ncom
    % Calculate ideal-gas enthalpy for each component using correlations
    Hid_i(i) = CP(1,i)*(T-273.15) + 1/2*CP(2,i)*(T^2-273.15^2) ...
        + 1/3*CP(3,i)*(T^3-273.15^3) + 1/4*CP(4,i)*(T^4-273.15^4);
    % Calculate total ideal-gas enthalpy
    Hid = Hid + z(i)*Hid_i(i);
end
% Calculate total enthalpy
H = Hid + Hres;
func1 = H;
func2 = Hid_i;

% Calculate total specific heat capacity
%Input:
%
%dlndphi_i : First derivative of ln(phi) with respect to temperature
%ddlnphi_i : Second derivative of ln(phi) with respect to temperature
%CP         : Coefficients for calculating heat capacity
%T          : System temperature
%z          : Feed mole fractions
%x          : Component mole fractions
%n          : Phase mole fractions
%Ncom       : Number of components
%
%Return:
%Total specific heat capacity
function func = Heatcapacity(dlnphi_i, ddlnphi_i, CP, T, z, x, n, Ncom)
Cp = 0;
Cpres_k = [0, 0];
Cpres = 0;           % Residual heat capacity
Cpid = 0;            % Ideal-gas heat capacity
for k = 1:2
    for i = 1:Ncom
        Cpres_i(k,i) = -83.1447215*T^2*ddlnphi_i(k,i)-2*83.1447215*T*...

```

```

        dlnphi_i(k,i);
        Cpid_i(i) = z(i)*(CP(1,i) + CP(2,i)*(T-273.15) + CP(3,i)*...
            (T-273.15)^2 + CP(4,i)*(T-273.15)^3);
        Cpres_k(k) = Cpres_k(k) + x(k,i)*Cpres_i(k,i);
        Cpid = Cpid + Cpid_i(i);
    end
    Cpres = Cpres + n(k)*Cpres_k(k);
end
Cp = Cpres + Cpid;
func = Cp;

% Update temperature for the first isenthalpic flash iteration
%Input:
%
% Cp      : Calculated total specific heat capacity
% T       : Temperature of the system
% g_0     : Difference in enthalpy from the upstream system enthalpy
%
%Return:
% T_temp  : Updated temperature
function func = Initialtemp(Cp, T, g_0)
T_temp(1) = T - g_0/Cp;          % Calculate new temperature
func = T_temp;

```

Following isenthalpic flash iterations

```

% Isenthalpic flash calculations to solve for fugacities (f)
% and temperature (T2)

% This set of matlab codes includes isothermal flash calculation at
% the system pressure and a temperature T2.
% When phase equilibrium is reached, the results are used to calculate the
% enthalpy at the specified pressure and temperature. The calculated
% enthalpy is compared with the system enthalpy (at the upstream point) to
% check for constant enthalpy (isenthalpic). If it is not satisfied the
% temperature T2 is updated. % The process is repeated until the energy
% conservation is satisfied (isen = 1)

% Isothermal flash calculations at local pressure (P2) and temperature (T2)
flash

% Calculate derivative of fugacity coefficients respective to temperature
% for calculating residual enthalpy
dlnphi_i = diffphi(bi, b, B, aij, a, A, ac, alph, m, Z, x, P2, T2, Tc, ...
    kij, Ncom);
% Calculate total enthalpy
[H, Hid_i] = enthalpy(dlnphi_i, x, n, CP, z, T2, Ncom);
% Check constant enthalpy.
% Record calculated enthalpy of each isenthalpic flash iteration.
Hnew(iter) = H;
% Calculate the difference between the calculated enthalpy and
% the system enthalpy at the upstream point
g(iter) = Hnew(iter) - Hsys;
% Compare the calculated enthalpy with the upstream system enthalpy
checkg = (1-Hnew(iter)/Hsys)^2;
% If the difference is less than a tolerance value
if checkg < 1e-12
    % Stop this iterating process
    isen = 1;

```



```

% If the difference is more than the tolerance value
else
    % Update temperature using the energy balance equation
    T_temp = Updatetemp(T1, g_0, g, iter, T_temp);
    % Update local temperature (T2) with the updated temperature
    T2 = T_temp(iter+1);
    % Update isenthalpic flash iteration index
    iter = iter + 1;
end

% Update temperature using the energy balance equation
%Input:
%
% Cp      : Calculated total specific heat capacity
% T       : Temperature of the system
% g_0     : Difference in enthalpy from the upstream system enthalpy
%
%Return:
% T_temp  : Updated temperature
function func = Updatetemp(T1, g_0, g, iter, T_temp);
T_temp_0 = T1;
T_temp(iter+1) = 0;
if iter == 1 % For the second iteration
    T_temp_minus = T_temp_0;
    g_minus = g_0;
else
    T_temp_minus = T_temp(iter-1);
    g_minus = g(iter-1);
end
% Calculate new temperature
T_temp(iter+1) = T_temp(iter) - g(iter)*(T_temp(iter)-T_temp_minus)/...
    (g(iter)-g_minus);
func = T_temp;

```

Determine asphaltene precipitation

```

% Check for precipitation. If fugacity in the liquid phase is less than
% the calculated asphaltene fugacity, there is no precipitation and
% precipindex remains zero.
check0 = 0;
% Compare the fugacity of the last component in the liquid phase with
% the precalculated asphaltene fugacity
checkfa = f(1,Ncom)-faref;
% If checkfa is negative, there is no precipitation
if checkfa <= 0
    disp('There is no precipitation');
    precipindex = 0;
% If checkfa is positive, there is precipitation
elseif checkfa > 0
    disp('There is precipitation');
    precipindex = 1;
    count = count+1;
end

```

