



Modelling of Wax Deposition in Sub-sea Pipelines

Prepared by

Stephen Hugh Bryan

(1285145)

A Research Report submitted to the Faculty of Engineering and the Built Environment,
University of the Witwatersrand, in partial fulfilment of the requirements for the degree of
Master of Science in Engineering

Supervisors: Dr. Mulopo, Dr. Nkazi and Prof. Dandekar.

May, 2016

DECLARATION

I declare that this work is a result of my own research, except where specifically indicated to the contrary, and has not been submitted for any other degree or examination to any other University.

Signed: Stephan

Date: 23 May 2016

ACKNOWLEDGEMENTS

I would like to share my deepest gratitude to my Supervisors Dr. Jean Mulopo, Dr. Diakanua Nkazi and Prof. Abhijit Dandekar for their supervision and guidance throughout this year. I give my sincerest thanks to Susan, Sarah and Fiona for their support and encouragement and my utmost gratitude to Prof. Ian Barker for all of his time and insight. I dedicate this work to my parents David and Irene for all they have done for me.

ABSTRACT

Wax deposition in sub-sea pipelines is a major concern in the oil industry. Wax precipitates in a pipeline when the temperature falls below a certain temperature called the Wax Appearance Temperature. As wax precipitates and deposits in a pipeline, the wax deposit can cause oil flow problems and reduce production efficiency. At critical conditions wax deposits in pipelines may damage production equipment or cause a production stoppage.

In the past two or three decades, the challenge has been to develop and apply high-fidelity models for wax deposition. In this context, two promising models have recently been developed: that of Eskin et al. (2014) and that of Haj-Shafiei et al. (2014). This research report tries to solve and elucidate some of the key assumptions of these wax deposition models, through implementing them in the Python Programming Language.

In investigating the significance of the shape of the Solubility Curve on wax deposition, a concave shaped solubility curve was found to exhibit the highest average deposition and the convex shaped solubility curve the lowest average deposition. The shape of the solubility curve also drastically affects the peak wax height with a concave shaped solubility curve forming the greatest wax peak.

The results have shown that the description of the phenomenon of Wax deposition in undersea pipelines is difficult to grasp for anyone not familiar with the topic because of the complexity of the mechanisms involved. Often, the models are difficult to visualise owing to their multi-dimensional solution procedure and the interdependence of parameters. The resultant programs could be utilised to help illustrate the process of Wax deposition to Engineering students or professionals who do not have access to expensive proprietary software or who are less proficient with programming but interested in the field.

CONTENTS

DECLARATION	i
ACKNOWLEDGEMENTS	ii
ABSTRACT	iii
LIST OF TABLES	1
LIST OF FIGURES	2
NOMENCLATURE	4
1 INTRODUCTION	9
1.1 Wax Deposition	9
1.2 Wax Control	10
1.3 South Africa	12
1.4 Wax Deposition Modelling	13
1.5 Wax Solubility and Precipitation Curves	14
1.6 Python	16
1.7 Aim and Outline of the Report	17
2 LITERATURE REVIEW	19
2.1 Significant Factors for Wax Deposition Modelling	19
2.1.1 Deposition Mechanisms	19
2.1.2 Wax Ageing	20
2.2 The Michigan Wax Predictor	21
3 METHODOLOGIES	24
3.1 Mass-flux-based Implementation	24
3.1.1 Temperature	24
3.1.2 Concentration	28
3.1.3 Mass-flux	32
3.1.4 Shear Stripping	34
3.1.5 Deposit Model	36
3.2 Thermal-difference-based Implementation	39
3.2.1 Thermal Driving Force	40
3.2.2 Cold-flow	43
3.2.3 Deposit Ageing	45

4 RESULTS AND DISCUSSION	47
4.1 Results	47
4.1.1 The Effect of the Shape of the Solubility Curve on Wax Deposition	47
4.1.2 The Effect of Operating Temperature and Solubility on Wax Deposition	48
4.2 Discussion	50
4.2.1 Deposit Temperature Assumption	50
4.2.2 Main Driving Force Assumption	52
4.2.3 Model Implementations	55
4.2.3.1 Jupyter Notebook	55
4.2.3.2 Accessibility	57
4.2.3.3 Speed of Execution	58
4.2.3.4 Application	59
5 CONCLUSIONS	60
6 FUTURE WORK	62
7 BIBLIOGRAPHY	63
8 APPENDICES	67
8.1 Other significant factors for wax deposition	67
8.1.1 Wax Thermodynamics	67
8.1.2 Precipitation Kinetics	68
8.1.3 Multiphase Flow and Pour Point	68
8.1.4 Effects of the Oil Flow Rate on Wax Deposition	70
8.2 Mass-flux-based Implementation	70
8.2.1 Typical Input Data	70
8.2.2 Parameter Equations	72
8.2.3 Interface Temperature Equations	72
8.2.4 Mass-flux-based Implementation's Code	73
8.3 Thermal-difference-based Implementation	85
8.3.1 Typical Input data	85
8.3.2 Parameter Equations	87
8.3.3 Thermal-difference-based Implementation's Code	88
8.4 Effect of Changing the Operating Temperature	93

LIST OF TABLES

Table	Page
0.1 Michigan Wax Predictor	4
0.2 Mass-flux-based Implementation	5
0.3 Thermal-difference-based Implementation	7
4.1 Wax Thickness using Different Sloped Solubility Curves	48
4.2 Mass-flux-based Implementation's Changes in Average Thickness with Temperature	49
4.3 Mass-flux-based Implementation's Changes in Maximum Thickness with Temperature	50
4.4 Thermal-difference-based Implementation's Changes in Thickness with Temperature	50
8.1 Variable Input Values – Mass-flux-based Implementation	70
8.2 Variable Property Values - Mass-flux-based Implementation	71
8.3 Fixed Property Values - Mass-flux-based Implementation	71
8.4 Variable Input Values – Thermal-difference-based Implementation	85
8.5 Fixed Property Values – Thermal-difference-based Implementation	86
8.6 Maximum Thickness of Different Solubility Curves with Temperature	93
8.7 Average Thickness of Different Solubility Curves with Temperature	93
8.8 Thermal-difference-based Implementation's Thickness with Temperature	94

LIST OF FIGURES

Figure	Page
1.1 Wax deposit	11
1.2 Diagram of a Pig removing deposits	11
1.3 Dewaxing Pigs	12
1.4 Cost with pigging frequency	12
1.5 A typical graph of a wax thickness profile changing with time	14
1.6 Wax solubility curve	15
1.7 Wax precipitation curve	16
1.8 Diagram of the RedMonk Programming Language Rankings	17
2.1 Diffusion across a wax deposit	21
2.2 Deposit thickness profiles	23
3.1 Graph showing the resistance layer temperature profiles	25
3.2 Graph showing axial fluid temperature profiles	26
3.3 Graph showing a radial fluid temperature profile	26
3.4 Graph showing a 2D temperature segment	27
3.5 Graph showing a 2D temperature segment zoomed in near the wall and inlet	27
3.6 Graph showing the bulk temperature profile and the WAT	30
3.7 Graph showing the bulk concentration profile and solubility profile	31
3.8 Graph showing a 2D concentration segment	31
3.9 Graph showing a radial supersaturation profile	32
3.10 Graph showing a 2D supersaturation segment zoomed in near the wall and inlet	33
3.11 Graph showing a 2D solubility segment	34
3.12 Graph showing shear-removal and the resultant mass-flux	35
3.13 Graph showing the shear removal constant	36
3.14 Graph showing the average thermal conductivity	38
3.15 Graph showing a normalised porosity	38
3.16 Graph showing the deposit thickness varying with time (0-6 days)	39
3.17 Graph showing the heat loss profile	41
3.18 Graph showing the wax thickness profile	42

3.19 Graph showing the pressure drop profile	43
3.20 Graph showing precipitation	44
3.21 Phase Diagram	44
3.22 Cubical-Cage Representation for Modelling the Effect of Shear Stress	45
3.23 Graph showing the deposit volume fraction profile without shear induced ageing	46
3.24 Graph showing the deposit volume fraction profile with shear induced ageing	46
4.1 Solubility curves used in solubility trial	48
4.2 Solubility curves used in temperature trial	49
4.3 The temperature profile from the Thermal-difference-based Implementation	51
4.4 Graph showing the deposit thickness profiles with a straight line (0.1) solubility curve	53
4.5 Graph showing the deposit thickness profiles with a concave solubility curve	53
4.6 Graph showing the deposit thickness profiles with a convex solubility curve	54
4.7 The General User Interface for the Mass-flux-based Implementation	56
4.8 The General User Interface for the Thermal-difference-based Implementation	57
4.9 Static Website	58
8.1 Impact of flow pattern on wax deposit	69
8.2 A flow map of two phase oil/water flow	69
8.3 Non-isothermal velocity profile of oil/water flow	69

NOMENCLATURE

Table 0.1: Michigan Wax Predictor

Symbol	Description	Units
V_z	Velocity in the axial direction	m/s
r	Radial coordinate	m
T	Temperature	°C
ε_H	Eddy thermal diffusivity	m ² /s
α_T	Thermal diffusivity of the oil	m ² /s
C	Concentration of wax molecules	kg/m ³
ε_M	Eddy mass diffusivity	m ² /s
D_{wo}	Diffusivity of wax in oil	m ² /s
k_r	Rate constant for precipitation	s ⁻¹
C_{ws}	Solubility of wax	kg/m ³
$k_{r,cloud}$	Rate constant at the cloud point temperature	s ⁻¹
T_{cloud}	Cloud point temperature	°C
E	Activation Energy	J/mol
R	Ideal gas constant	J/(°K.mol)
V_a	Molecular volume of the paraffin	cm ³ /mol
$\rho_{deposit}$	Density of the deposit	kg/m ³
$\overline{F_w}$	Wax fraction in the deposit	No
δ	Deposit thickness	mm
t	Time	s
D_{wo}	Diffusivity of wax in oil	m ² /s
D_{eff}	Effective Diffusivity of wax in the deposit	No

Table 0.2: Mass-flux-based Implementation

Symbol	Description	Units
T_B	Bulk temperature	$^{\circ}\text{K}$
ξ	Axial coordinate	m
h_{bw}	Heat transfer coefficient between the bulk fluid and the deposit surface	$\text{W}/(\text{m}^2 \cdot ^{\circ}\text{K})$
T_d	Deposit Temperature	$^{\circ}\text{K}$
ρ_f	Fluid density	kg/m^3
C_{pf}	Fluid heat capacity	$\text{J}/(\text{kg} \cdot ^{\circ}\text{K})$
U	Mean flow velocity	m/s
D_p	Pipe diameter	m
T_{sf}	Surface Temperature	$^{\circ}\text{K}$
y	Coordinate determining distance from the deposit layer surface	No
$+$	Dimensionless variable	No
Pr, Pr_T	Prandtl number, Turbulent Prandtl number	No
q_t	Deposition rate to the deposit layer surface	$\text{kg}/(\text{m}^2 \cdot \text{s})$
χ	Von Karman constant	No
ε_t	Thermal eddy diffusivity	m^2/s
a_f	Fluid thermal diffusivity	m^2/s
T	Temperature	$^{\circ}\text{K}$
ε_M	Turbulent diffusivity	m^2/s
D	Molecular diffusivity of wax	m^2/s
k_r	Precipitation rate constant for wax solids in a hydrocarbon fluid	No
C	Wax mass concentration	kg/m^3
C_s	Saturation concentration	kg/m^3
μ_f	Dynamic viscosity	Pa.s
v_w	Molar volume of wax	cm^3/mol

Sc_m	Shmidt number	No
j	Axial node	No
u^*	Friction velocity	m/s
ν_f	Fluid kinematic viscosity	m^2/s
c_{wax}	Concentration of wax at the wax appearance temperature	kg/m^3
T_{wat}	Wax appearance temperature	$^{\circ}K$
κ	Shear removal constant	s^{-1}
q_d	Deposition rate through the deposit layer surface	kg/m^3
ϕ	Porosity	No
λ_f	Fluid thermal conductivity	$W/(m. ^{\circ}K)$
λ_d	Deposit thermal conductivity	$W/(m. ^{\circ}K)$
n	Diffusivity Exponent	No
υ	Dimensionless complex	No
Θ	Characteristic time	s
λ_{wax}	Wax thermal conductivity	$W/(m. ^{\circ}K)$
L_p	Pipe length	m
δ	Deposit layer thickness	m
ρ_{wax}	Wax density	kg/m^3
Re	Reynolds number	No
c_m	Mean wax content	kg/m^3
τ_w	Wall-shear stress	Pa
r_{wo}, A_{wo}	Wall outside radius, wall outside area	m, m^2
$\Delta t_{wall,}$	Wall thickness	m
r_{dep}, A_{dep}	Radial distance to deposit surface, deposit surface area	m, m^2
Δz	Axial element length	m
A_{wi}	Segment wall inside area	m^2
$A_{logmean_pipe}$	Segment log mean area of pipe	m^2
$A_{logmean_dep}$	Segment log mean area of deposit	m^2
R_{out}, R_{in}	Outer fluid thermal resistance, Inner fluid thermal	$^{\circ}K.s/J$

	resistance	
h_o, h_i	Outer heat transfer coefficient , Inner heat transfer coefficient	$W/(m^2 \cdot ^\circ K)$
R_{pipe}, R_{dep}	Pipe thermal resistance, deposit thermal resistance	$^\circ K.s/J$
k_{pipe}	Thermal conductivity of pipe	$W/(m.K)$
q	Segment heat loss	J/s
T_{sea}	Sea Temperature	$^\circ K$
T_{wo}, T_{wi}	Wall outside temperature, Wall inside temperature	$^\circ K$

Table 0.3: Thermal-difference-based Implementation

Symbol	Description	Units
q, q_h, q_d	Rate of heat transfer	W
$C_{p,l}^{mix}$	Wax-solvent mixture specific heat capacity	$J/(kg \cdot ^\circ K)$
T_{hin}	Inlet wax-solvent mixture temperature	$^\circ K$
T_{hout}	Outlet wax-solvent mixture temperature	$^\circ K$
$\dot{m}$	Mass flow rate of wax-solvent mixture	kg/s
h_h	Convective heat transfer coefficient of wax solvent mixture	$W/(m^2 \cdot ^\circ K)$
R	Inside pipe radius	m
δ	Deposit layer thickness	m
ΔL	Length of axial pipe element for steady state calculations	m
T_h	Average wax-solvent mixture temperature	$^\circ K$
T_d	Liquid-deposit interface temperature	$^\circ K$
T_c	Surrounding temperature	$^\circ K$
ΔP	Pressure drop	Pa
ρ_l^{mix}	Wax-solvent mixture density	kg/m^3
D	Pipe inside diameter	m
μ	Viscosity of wax solvent mixture	Pa.s

F	Volumetric flow rate	m^3/s
Δt	Residence time	s
χ	Rate of cooling	$^{\circ}\text{K}/\text{s}$
Δf	Change in solid wax phase mass fraction	No
ΔH_m	Latent heat of fusion for $\text{C}_{29}\text{H}_{60}$	J/kmol
w_{29}	Mass fraction of wax in wax-solvent mixture	No
w_{29}^*	Liquid phase mass fraction of wax	No
V_s, V_l	Solid phase volume fraction in cubical cage, Liquid phase volume fraction in cubical cage	No
ρ_i	Paraffin wax-solvent densities	kg/m^3
$V_{s\beta}$	Tilted Solid phase volume fraction in cubical cage	No
β	Tilting angle of cubical cage	degrees
Pr	Prandtl number	No
Re	Reynolds number	No
$C_{p,l}^{CH_3}, C_{p,l}^{CH_2}$	Specific heat capacity of CH_3 and CH_2	$\text{J}/(\text{kg} \cdot ^{\circ}\text{K})$
$C_{p,l}^{C_{13}}, C_{p,l}^{C_{29}}$	Specific heat capacity of $\text{C}_{13}\text{H}_{28}$ (wax) and $\text{C}_{29}\text{H}_{60}$ (solvent)	$\text{J}/(\text{kg} \cdot ^{\circ}\text{K})$
w_{13}, w_{29}	Mass fraction of wax and solvent	No
ρ_{13}, ρ_{29}	Density of wax and solvent	kg/m^3
ϕ_{13}, ϕ_{29}	Superficial volume fraction of wax and solvent	No
g_{13}, g_{29}	Mole fraction of wax and solvent	No
v_{13}, v_{29}	Molar volume of wax and solvent	m^3/kmol
k_{13}, k_{29}	Thermal conductivity of wax and solvent	$\text{W}/(\text{m} \cdot ^{\circ}\text{K})$
k_l^{mix}	Thermal conductivity of wax-solvent mixture	$\text{W}/(\text{m} \cdot ^{\circ}\text{K})$
k_d	Thermal conductivity of the deposit layer	$\text{W}/(\text{m} \cdot ^{\circ}\text{K})$
$k_{wax,s}$	Thermal conductivity of the solid wax	$\text{W}/(\text{m} \cdot ^{\circ}\text{K})$

1 INTRODUCTION

A major issue in upstream oil and gas production is the deposition of wax onto the pipe walls in undersea pipelines. Two promising wax deposition models have recently been developed: that of Eskin et al. (2014) and that of Haj-Shafiei et al. (2014). This research report, tries to solve and elucidate some of the key assumptions of these models through implementing them in the Python Programming Language.

1.1 Wax Deposition

Hot oil from sub-sea oil wells often flows along the sea floor through under water pipelines up to surface facilities. The typical reservoir temperatures range from between 70°C and 150°C while sea water temperatures near the sea floor are around 4°C (Huang et al., 2011b). As the oil flows, heat is lost through the pipe walls to the sea. The solubility of the heavier oil components in the bulk oil decreases as the oil cools causing wax precipitation to occur (Stubsjøen, 2012).

Wax depositing onto the pipe walls in these undersea pipelines is a major issue in upstream oil production. The accumulation of wax on the pipeline wall restricts the flow area and causes a pressure drop (Galta, 2014). Many oil wells are located far from shore in deep waters or in Arctic regions and thus suffer from wax deposition (Galta, 2014).

Due to the continued move towards deep-sea oil production, flow assurance is now a serious technical and economic problem for the Oil and Gas industry (Coutinho et al., 2006). New oil production in colder areas is likely to experience greater wax deposition (Aiyejina et al., 2011).

The worst case scenario of wax deposition is complete production stoppage resulting from wax build up and the blocked pipe section having to be replaced (Huang et al., 2011b). A platform was once completely abandoned in the North Sea due to severe frequent wax deposition (Huang et al., 2011b).

Flow assurance has recently progressed faster than the average applied engineering discipline (Gupta and Anirbid, 2015). In the last decade the wax deposition problem has led to the

development of flow advisory software. This software relies on rigorous, robust models and is still under development (Gupta and Anirbid, 2015).

1.2 Wax Control

The components of oil with the highest melting points can form a solid phase at temperatures occurring in sub-sea pipelines. Wax is composed mainly of linear alkanes (n-paraffin molecules) with more than 20 carbon atoms (Coutinho et al., 2006). A wax deposit is porous with oil held within its contents (Aiyejina et al., 2011). Figure 1.1 (Siljoberg, 2012) shows an example of severe wax deposition in a section of pipeline.

Wax deposits are usually removed by mechanical means such as “pigging” as illustrated by Figure 1.2 (Devaux, 2015). Figure 1.3 (Pigtek, 2015) is a photograph which shows examples of a type of De-waxing Pig. The application of this method is, however, restricted by the thickness and amount of wax (Eskin et al., 2014).

Pigging infrequently enough can result in the Pig becoming stuck; however pigging too frequently can result in losses of tens of millions of dollars from delayed production, as seen in Figure 1.4 (Huang et al., 2011a). An accurate model of wax deposition is needed to optimally schedule the pigging program (Eskin et al., 2014). The frequency varies from 2 to 3 days up to 3 to 4 months depending on the pipeline (Galta, 2014).

Other means of wax control are Thermal Methods which include insulation and active and periodic heating, but these are expensive (Deo et al., 2007). Chemical injection is another method used, however this is often constrained in its applicability (Deo et al., 2007). A method uses chemical additives to generate heat or gas inside the pipeline by means of exothermic reactions (Deo et al., 2007).

An interesting new means of wax control has been proposed, namely Cold-flow. In this method the oil is seeded and cooled. This diminishes the radial heat gradient and provides surface area on which wax can grow within the bulk oil instead of on the pipe wall (Deo et al., 2007). However oil can turn into a gel when its solid content is around 4 - 5%, this can change the

rheological properties of the oil resulting in it becoming a shear-thinning suspension (Ramirez-Jaramillo et al., 2004).

Other less common wax deposition prevention methods include: internal coatings, biodegradation or sonic methods (Deo et al., 2007). Wax control methods can also be used in conjunction with one another, a common example is Pigging and Thermal insulation.



Figure 1.1: Wax deposit (Siljubergh, 2012)

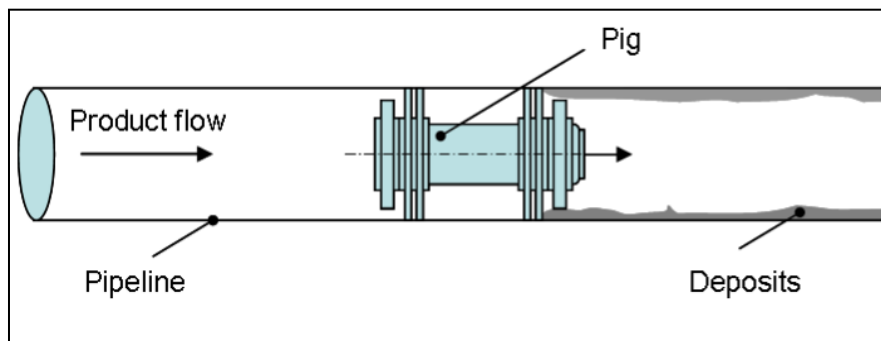


Figure 1.2: Diagram of a Pig removing deposits (Devaux)



Figure 1.3: Dewaxing Pigs (Pigtek)

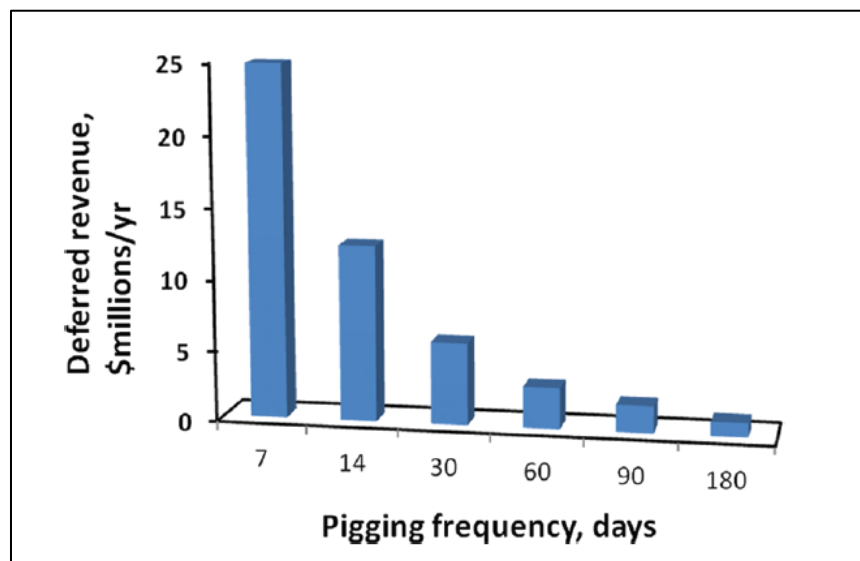


Figure 1.4: Cost with pigging frequency (Lu et al., 2012)

1.3 South Africa

The Sub-Saharan African region currently has a relatively small share of the world's proven oil reserves; however this is expected to change as exploration in the region has been confirming it's oil and gas potential (Saoga, 2015). The exploration off the coast of South Africa has been limited by the depth of these potential resources; however there have been major improvements in deep-sea technology and there is now a need for South Africa to diversify its energy supply (Saoga, 2015).

South Africa, as a result of its mature downstream industry as well as the fact that the country is geographically and politically well-positioned, has developed into a key hub for the region (Saoga, 2015). It is therefore important to develop South African expertise and to find models applicable to local reservoirs.

1.4 Wax Deposition Modelling

Aiyejina et al. (2011, p.672) argue that the “Accurate modelling of deposition in pipelines can be a complex and difficult undertaking, because while precipitation is mainly a function of thermodynamic variables such as composition, pressure and temperature, deposition is also dependent on flow hydrodynamics, heat and mass transfer, solid-solid and surface-solid interactions”.

Significant research has gone into the understanding and modelling of wax deposition over the previous decades as accurate prediction of the wax deposition rate provides invaluable information for the design and maintenance of a pipeline (Azevedo and Teixeira, 2003). The accurate prediction of the deposition rate can help to increase the efficiency of oil and gas production (Kamari et al., 2013).

Wax deposition models are used to predict the severity of wax deposition. A common objective of these models is to predict how the wax deposit thickness and wax fraction profiles change over time. However wax deposit models vary in their applicability depending on which factors they include, their different methods, and different assumptions. A typical graph of how the thickness of wax deposited along the pipeline length changes with time is shown in Figure 1.5 (Kjøraas, 2012).

Machine learning techniques (used for regression and classification problems) can provide a powerful tool for wax deposition prediction (Kamari, 2013). Artificial Neural Networks (ANNs) or Support Vector Machines (SVMs) as examples could, with sufficient field data, produce an accurate prediction of deposition. These methods however require a comprehensive training data set (enough data, spanning a wide enough range of conditions). These techniques, though

promising, do not help reveal the underlying physiochemical processes of wax deposition and will not be considered in this Research Report.

Aiyejina et al. (2011) found that numerous older models make the assumption that heat and mass transfer are linked through the chain-rule. They note that this assumption is not valid if the temperature and concentration are dependent (Aiyejina et al., 2011). Ventkatesan and Fogler (2004) state that heat transfer analogies based on this assumption are not applicable for turbulent flow (Aiyejina et al., 2011). Recently the Michigan Wax Predictor (MWP) applied a corrected heat and mass transfer analogy (Stubsjøen, 2012).

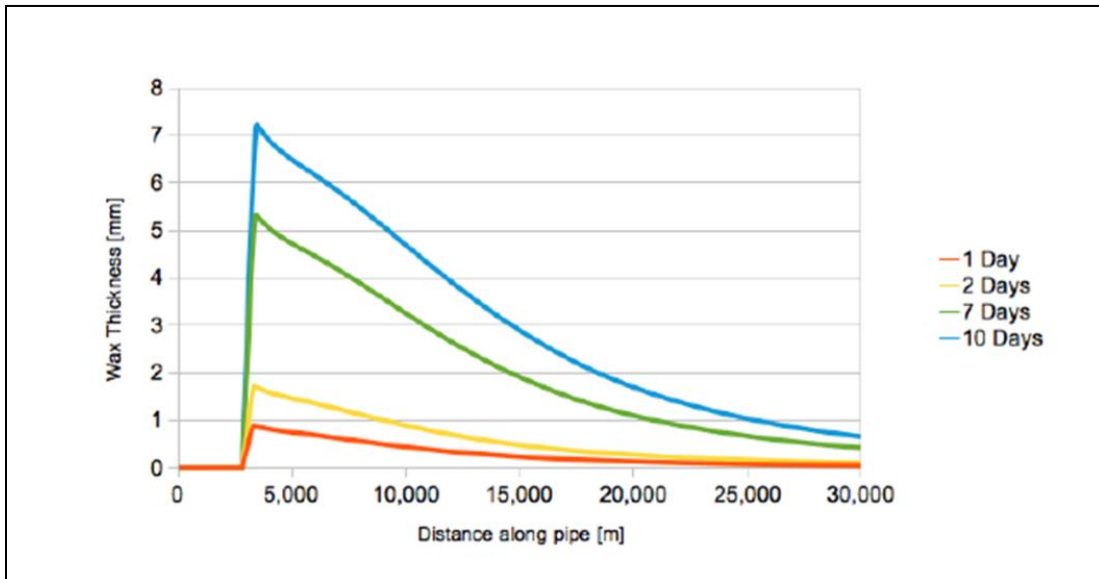


Figure 1.5: A typical graph of a wax thickness profile changing with time (Kjøraas, 2012)

1.5 Wax Solubility and Precipitation Curves

Modelling of wax precipitation and deposition is highly temperature sensitive. Wax precipitate first appears at a temperature known the Wax Appearance Temperature (WAT). It is the temperature when wax liquid starts to precipitate out of the solution as a solid state (Stubsjøen, 2012).

Wax deposition is thought to not only depend on the WAT but also the wax solubility curve or its closely related Wax Precipitation Curve (WPC) (Stubsj en, 2012). Oil itself is a mixture of molecules each with their own freezing point temperature, referring to this simply as solubility is a simplification (Siljuberg, 2012). The Solubility Curve expresses the maximum possible ‘wax in solution’ as a function of temperature. An example is shown in Figure 1.6 (Siljuberg, 2012). The wax precipitation curve expresses the amount of precipitated wax as a function of temperature, as shown in Figure 1.7 (Kj raas, 2012).

The expectation that wax precipitation should occur immediately on reaching its lowest soluble temperature is not correct. Normally a level of super-saturation is necessary to begin the crystallisation process (Azevedo and Teixeira, 2003).

The shapes of these curves vary for different crude oils based on their composition. The shape of the wax precipitation curve affects the equilibrium concentration of wax and varies the driving force of mass transfer (Stubsj en, 2012).

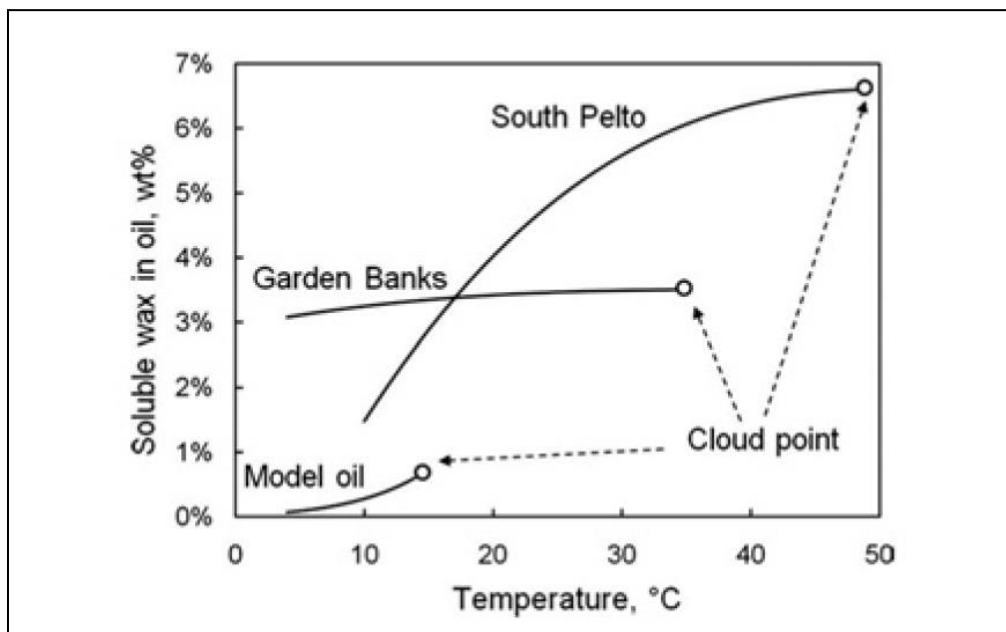


Figure 1.6: Wax solubility curve (Siljuberg, 2012)

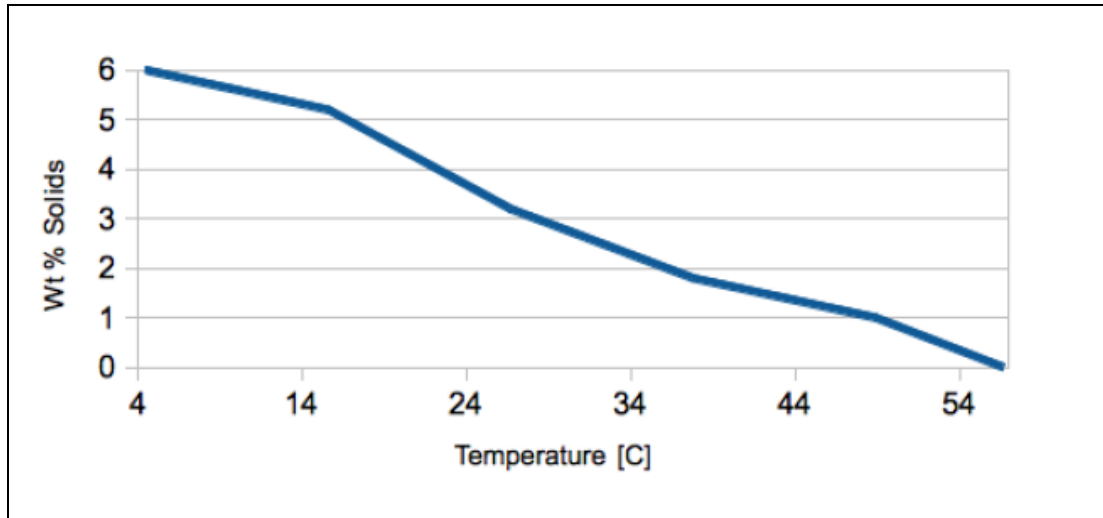


Figure 1.7: Wax precipitation curve (Kjøraas, 2012)

1.6 Python

Python is a non-proprietary general-purpose programming language and one of the most widely used computing languages as seen in Figure 1.8 (tecosytems, 2015). Python has a growing user base and is developing quickly. It is a language widely used for Scientific Computing capable of parallel distributed computations (Pérez and Granger, 2007). Python is used in a number of well-known commercial software projects such as Youtube and Dropbox.

Writing software using a non-proprietary programming language could allow for a greater audience to have access to it. Open-source software allows for the inspection of the inner workings of a program normally not possible with Proprietary Software.

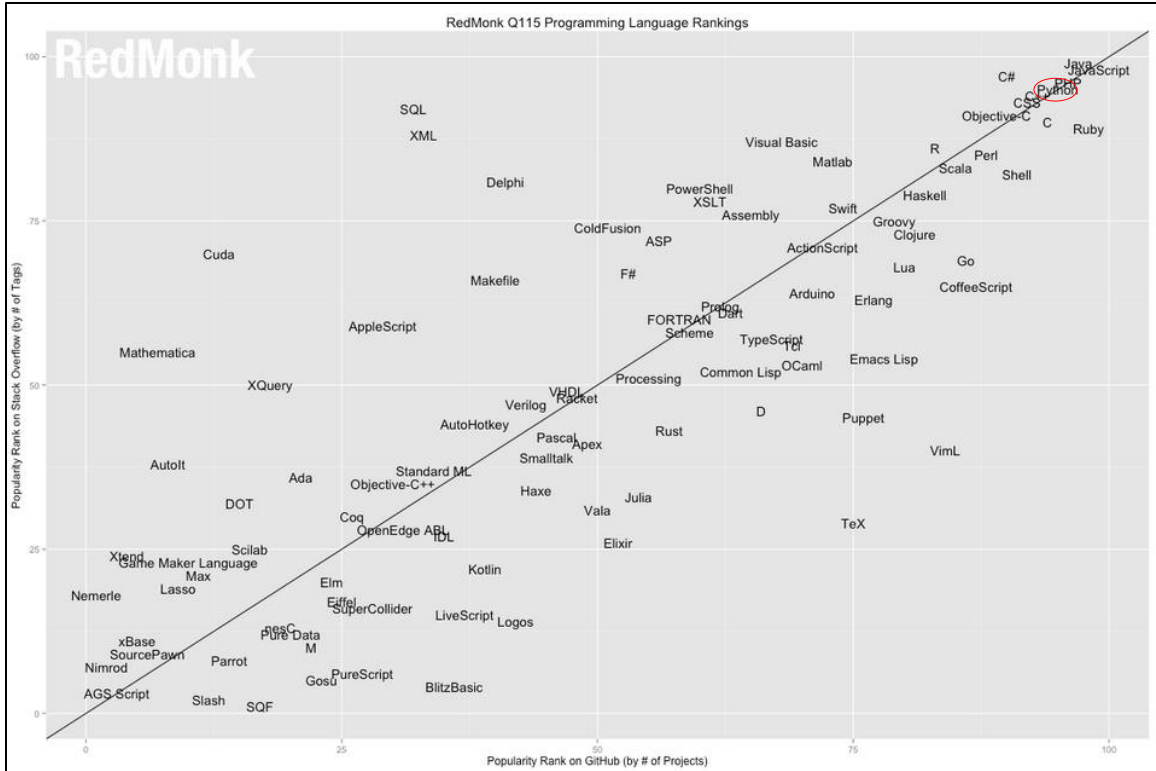


Figure 1.8: Diagram of the RedMonk Programming Language Rankings (tecosystems)

1.7 Aim and Outline of the Report

In the past two to three decades, the challenge has been the development and application of high-fidelity predictive models for wax deposition. Three recent noteworthy Wax Deposition Models are discussed namely the models of Eskin et al. (2014), Huang et al. (2011b) and Haj-Shafiei et al. (2014). The model equations of Eskin et al. (2014) and Haj-Shafiei et al. (2014) were solved using the Python Programming Language, from their publications. These models are recent and to our knowledge not part of current commercial software.

The model of Eskin et al. (2014) follows a mass-flux based approach, utilising a solubility curve, while the model of Haj-Shafiei et al. (2014) follows a thermal-difference based approach. These implementations were done in order to investigate both of these methods and the significance of the shape of the Solubility Curve on wax deposition. Two important differing assumptions are discussed. Different oils exhibit different solubility curves based on their composition. It is

therefore important to understand the significance of the shape of the solubility curve as it could affect the suitability of either methodology for certain oils.

The purpose of this study is:

- i. To solve and elucidate some of the key assumptions of the wax deposition models of Eskin et al. (2014) and Haj-Shatiei et al. (2014) as described in open publications and implement them in the Python Programming language.
- ii. To investigate the significance of the shape of the wax solubility curve on the deposition.

The aim while developing the model implementations was to reproduce the models as closely as possible to how the authors described them in their papers. However due to the limited scope of this Project, the Model developers were not involved. Where the methodology was not explicitly described it had to be assumed. Where this was necessary, any assumptions introduced were thought to represent what the authors implied or were thought reasonable to make.

2 LITERATURE REVIEW

Gupta and Anirbid (2015, p.48) point out that a “number of methods have [sic] been used to predict paraffin problems, but their standardization still eludes the scientific community”. With this in mind, a literature review was conducted to help provide an overview of the topic and the main factors affecting wax deposition modelling.

2.1 Significant Factors for Wax Deposition Modelling

2.1.1 Deposition Mechanisms

Models are created based on the properties of the mixture as well as the mechanisms of wax deposition and removal (Aiyejina et al., 2011). However there is still contention as to which of these mechanisms are relevant, therefore these mechanisms will be discussed briefly.

Burger et al. (1981), in a highly cited and comprehensive study on wax deposition mechanisms, found Molecular Diffusion, Brownian Diffusion, Shear Dispersion and Gravity Settling to be possible mechanisms for the deposition of wax (Stubsjøen, 2012).

Molecular diffusion results from the random collision of molecules. These random collisions result in a net flux of molecules in the direction of lower concentration. In sub-sea pipelines, where the wall temperature is below the WAT, a radial temperature gradient will form with wax crystallisation occurring in the cooler areas close to the wall. This causes the liquid phase close to the wall to have the lowest concentration of wax and wax to diffuse to the pipe wall (Stubsjøen, 2012). This establishes a concentration gradient.

Brownian Diffusion can occur below the Wax Appearance Temperature. Precipitated solid wax suspended in solution can move towards an area of lower concentration as a result of Brownian motion (Azevedo and Teixeira, 2003). This process can be described using Fick's law of diffusion characterised by a Brownian diffusion coefficient (Azevedo and Teixeira, 2003).

Though many authors have dismissed Brownian diffusion as a relevant mechanism, it still remains a possible contributing factor (Aiyejina et al., 2011).

Shear dispersion is the movement of suspended particles, cross-stream, in the direction of decreasing shear. In a pipe, this would result in movement towards the wall, where shear is at a minimum (Azevedo and Teixeira, 2003). However shear stripping or shear-removal of wax diminishes the extent of wax deposition.

Studies of deposition under no heat-flux conditions were carried out in flow loop experiments where only the flow-induced deposition mechanisms such as shear dispersion could apply (Azevedo and Teixeira, 2003). These experiments showed no deposition, which indicate that this mechanism is not significant for wax deposition (Azevedo and Teixeira, 2003). Gravity settling has also been shown to be not significant for typical pipeline flow conditions (Azevedo and Teixeira, 2003).

Singh in Stubbsj en (2012, p. 11), “suggested that the overall consensus is that the dominant wax deposition mechanism is molecular diffusion of n-paraffins in a viscous sub-layer driven by radial diffusion, following Fick’s law”.

2.1.2 Wax Ageing

The composition of a formed wax deposit changes over time (Aiyejina et al., 2011). This ‘Wax Ageing’ is another important factor for modelling wax deposition in pipelines (Aiyejina et al., 2011). Ageing also affects the hardness of the Wax deposit which is vitally important when deciding the “pigging” frequency.

Singh et al. (2000) developed a model for the calculation of wax deposit thickness and the deposit’s wax content along the pipe. Wax deposition begins with the precipitation of wax onto the wall forming a wax-crystal gel-like network containing a lot of liquid oil (Singh et al., 2001). Wax deposits are a porous three dimensional structure of wax crystals which contain a large fraction of oil (Singh et al., 2000).

As illustrated in Figure 2.1 (Siljuberg, 2012), the growth and ‘ageing’ of the gel-like deposit is a result of wax molecules diffusing in from the bulk solution (Singh et al., 2001). With time and an increased wax fraction the ‘ageing’ process hardens the wax deposits (Singh et al., 2000). Aiyejina et al. (2011, p.680) note “that the diffusion mechanism as used by Singh et al. (2000 and 2001a,b) is not the only possible mechanism for explaining the ageing process. In fact Continuo et al. (2003) found that ageing of wax deposits takes place even for samples under isothermal conditions”.

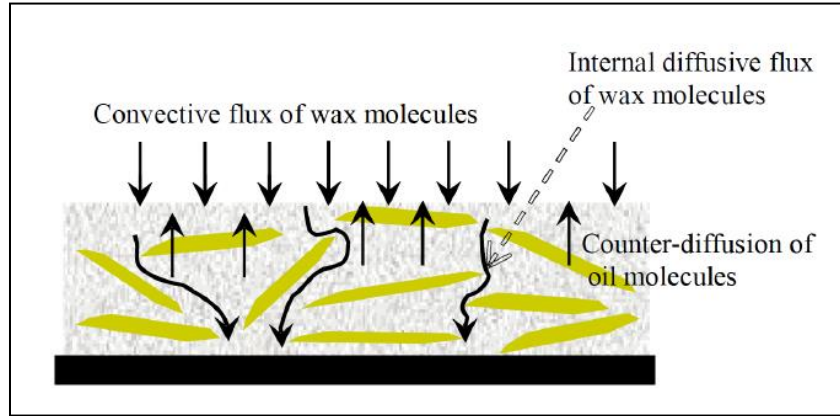


Figure 2.1: Diffusion across a wax deposit (Siljuberg, 2012)

2.2 The Michigan Wax Predictor

A brief description of the Michigan Wax Predictor (MWP) is necessary as the mass-flux based model that was solved is based on this model’s Mass Flux Analogy.

Rather than using empirical correlations as with previous models, the model solves the transport equations below (Huang et al., 2011b):

$$Vz \frac{\partial T}{\partial z} = \frac{1}{r} \frac{\partial}{\partial r} \left[r(\epsilon_H + \alpha_T) \frac{\partial T}{\partial r} \right] \quad (2.1)$$

$$Vz \frac{\partial C}{\partial z} = \frac{1}{r} \frac{\partial}{\partial r} \left[r(\epsilon_M + D_{wo}) \frac{\partial C}{\partial r} \right] - k_r(C - C_{ws}(T)) \quad (2.2)$$

Heat and mass transfer equations are solved together due to the assumed temperature-dependent precipitation kinetics (Huang et al., 2011b). Stubbsj en (2012, p.11) explains that “Molecular Diffusion is considered the dominant mechanism for wax deposition.” The MWP applies Fourier’s and Fick’s laws by using semi-empirical correlations to extend them to the turbulent flow regime (Stubbsj en, 2012). This is represented by the terms ε_H and, ε_M in the continuity equations above.

Bulk precipitation, represented by the term on the far right of the Mass Transfer equation (2.2), reduces wax deposition on the walls. Bulk precipitation decreases the amount of wax that can diffuse to the pipe-wall and deposit (Huang et al., 2011b).

As stated the Michigan Wax Predictor includes temperature dependent Wax Precipitation Kinetics. The MWP (Huang et al., 2011b) determines the precipitation rate constant (k_r) at temperature (T) from the following equations:

$$\frac{k_r}{k_{r,cloud}} = \left(\frac{T}{T_{cloud}}\right)^{1.47} \exp\left(\frac{\gamma E}{R} \left(\frac{1}{T} - \frac{1}{T_{cloud}}\right)\right) \quad (2.3)$$

$$\gamma = \frac{10.2}{V_a} - 0.791 \quad (2.4)$$

By changing the precipitation rate constant, the amount of deposition varies between two earlier models: the Independent Heat and Mass Transfer (IHMT) model and the Solubility model (Huang et al., 2011b). The IHMT model represents no bulk precipitation and the Solubility model represents immediate bulk precipitation (Huang et al., 2011b). Pressure is not included as it has been shown to be less significant for wax deposition (Stubbsj en, 2012).

How the deposit thickness changes with time is shown in Figure 2.2 (Stubbsj en, 2013). The deposit thickness is calculated as the difference between the convective mass-flux to the deposit surface, and that which diffuses into the deposit as below (Huang et al., 2011b):

$$2\pi r_i \rho_{deposit} \bar{F}_w \frac{d\delta}{dt} = 2\pi r_d \left(-D_{wo} \frac{dc}{dr} \Big|_{interface, convection} \right) - 2\pi r_d \left(-D_{eff} \frac{dc}{dr} \Big|_{interface, diffusive} \right) \quad (2.5)$$

The model also calculated the wax fraction in the deposit as described by the following equation (Huang et al., 2011b):

$$\pi \rho_{deposit} (r_i^2 - r_d^2) \frac{d\bar{F}_w}{dt} = 2\pi r_i \left(-D_{eff} \frac{dc}{dr} \Big|_{interface, diffusive} \right) \quad (2.6)$$

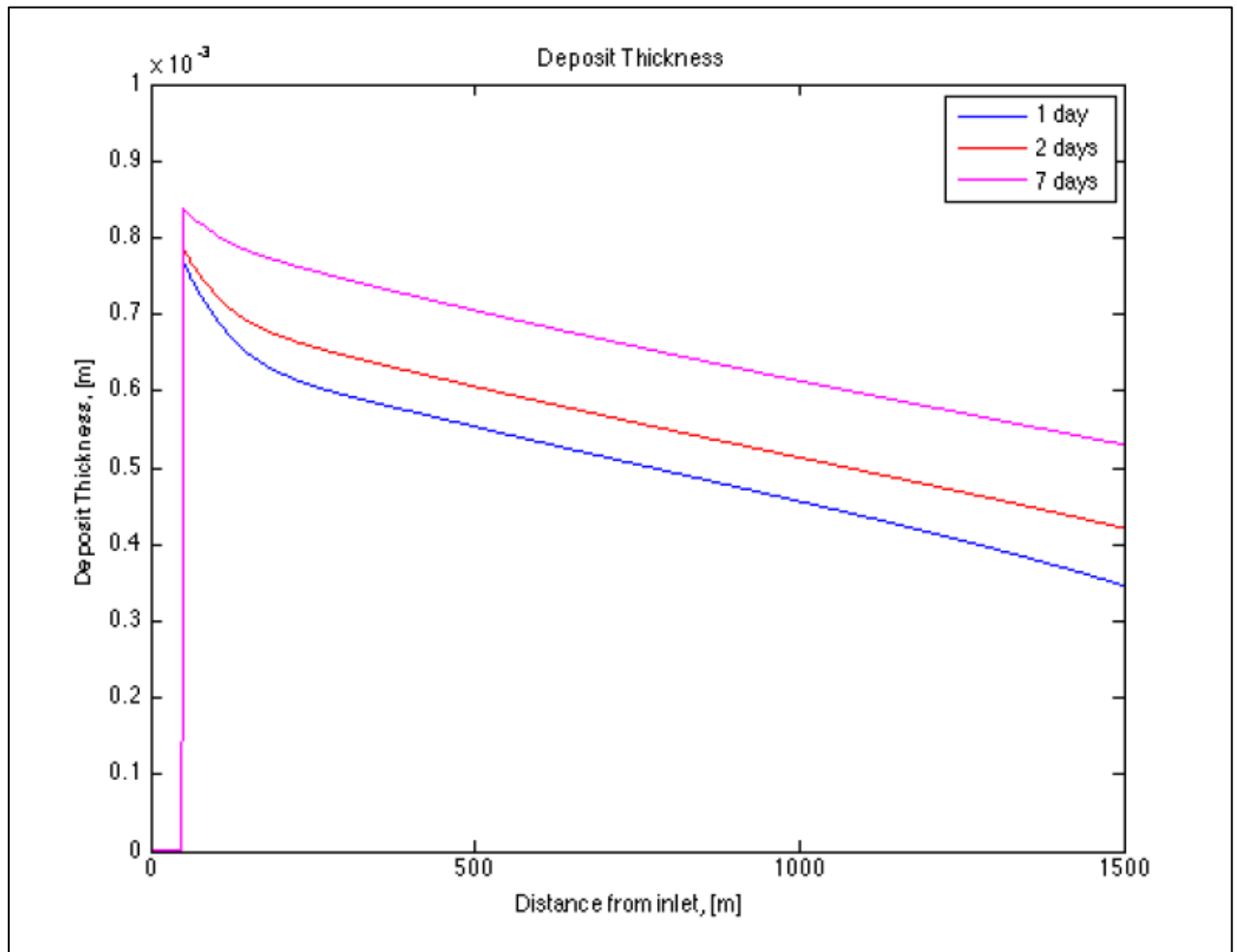


Figure 2.2: Deposit thickness profiles (Stubsj en, 2013)

3 METHODOLOGIES

3.1 Mass-flux-based Implementation

The model described in the paper ‘Modelling Wax Deposition in Oil Transport Pipelines.’ by Eskin, et al. (2014) was the model chosen to be implemented; referred to here as the Mass-flux-based Implementation. It is from this implementation that the graphs used in this section were generated. The input values for these graphs are provided in Appendix 8.2.1 with additional important property equations in Appendix 8.2.2. It should be noted that being an unsteady-state model the graphs vary with time as well as direction.

3.1.1 Temperature

The model equations were solved using a programming language due to its computational complexity. This model, as with most, does not follow a linear solution procedure. For example, in order to calculate the bulk temperature profile (Eqn. 3.1) the deposit temperature profile is required. However, this requires the deposit thickness profile to be known. This has to be calculated from the rate of deposition, which in turn is dependent on the bulk temperature profile.

$$\frac{dT_b}{d\xi} = - \frac{4h_{bw}(T_B - T_d)}{\rho_f C_{pf} U D_p} \quad (3.1)$$

The radial heat transfer equations described in Stubbsj en (2013) were used in order to introduce the added thermal resistance layers referred to in the paper; these equations are shown in Appendix 8.2.3. Notice in Figure 3.1 below, that the deposit thickness affects the temperature profiles across the different thermal resistance layers. The divergence in the deposit surface temperature from the inner wall temperature results from the deposit thickness increasing (increased insulation) in this segment of the pipeline.

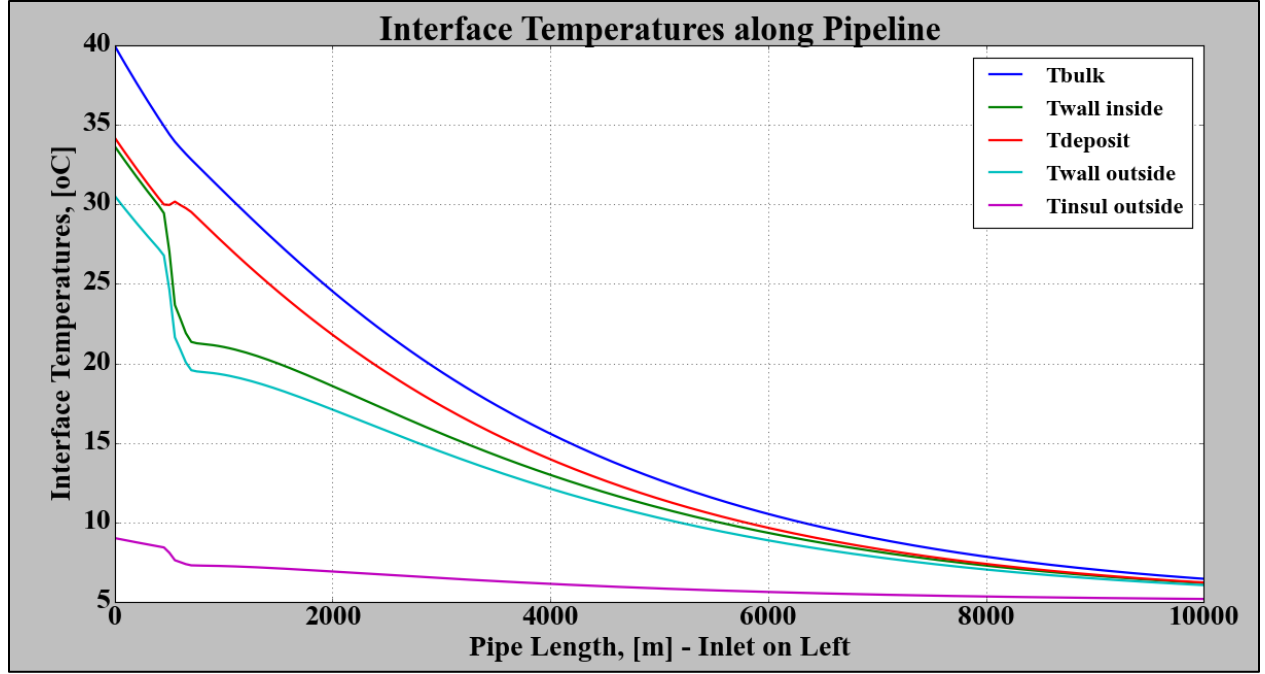


Figure 3.1: Graph showing the resistance layer temperature profiles

The bulk temperature allows the temperature distribution to be determined. This is done with the following correlation which depends on a dimensionless distance ($y^+ = \frac{y \cdot u^+}{\nu_f}$) from the deposit surface:

$$\frac{T - T_{sf}}{\Delta T} = \frac{y^+ Pr}{\Delta T_{200}^+} \quad y^+ < 5 \quad (3.2)$$

$$\frac{T - T_{sf}}{\Delta T} = \frac{5Pr + 5\ln(0.2y^+ Pr + 1 - Pr)}{\Delta T_{200}^+} \quad 5 \leq y^+ \leq 30 \quad (3.3)$$

$$\frac{T - T_{sf}}{\Delta T} = \frac{5Pr + 5\ln(1 + 5Pr) + 2.5\left(\frac{y^+}{30}\right)}{\Delta T_{200}^+} \quad 30 < y^+ < 200 \quad (3.4)$$

$$\Delta T_{200}^+ = 5Pr + 5\ln(1 + 5Pr) + 2.5\ln\left(\frac{200}{300}\right) \quad (3.5)$$

$$\Delta T = T_B - T_{dep} + \frac{q_t}{\left(\frac{\chi}{Pr_T}\right)\rho_f C_{p_f} u^*} \ln\left(\frac{400}{Dp^+}\right) \quad (3.6)$$

These equations result in the near wall temperature profiles shown in Figure 3.2 and Figure 3.3. Figure 3.4 and Figure 3.5 show two dimensional temperature segments that are assumed to be

symmetrical around the pipe centreline. Note that these profiles also vary with time as the deposit thickness grows with time.

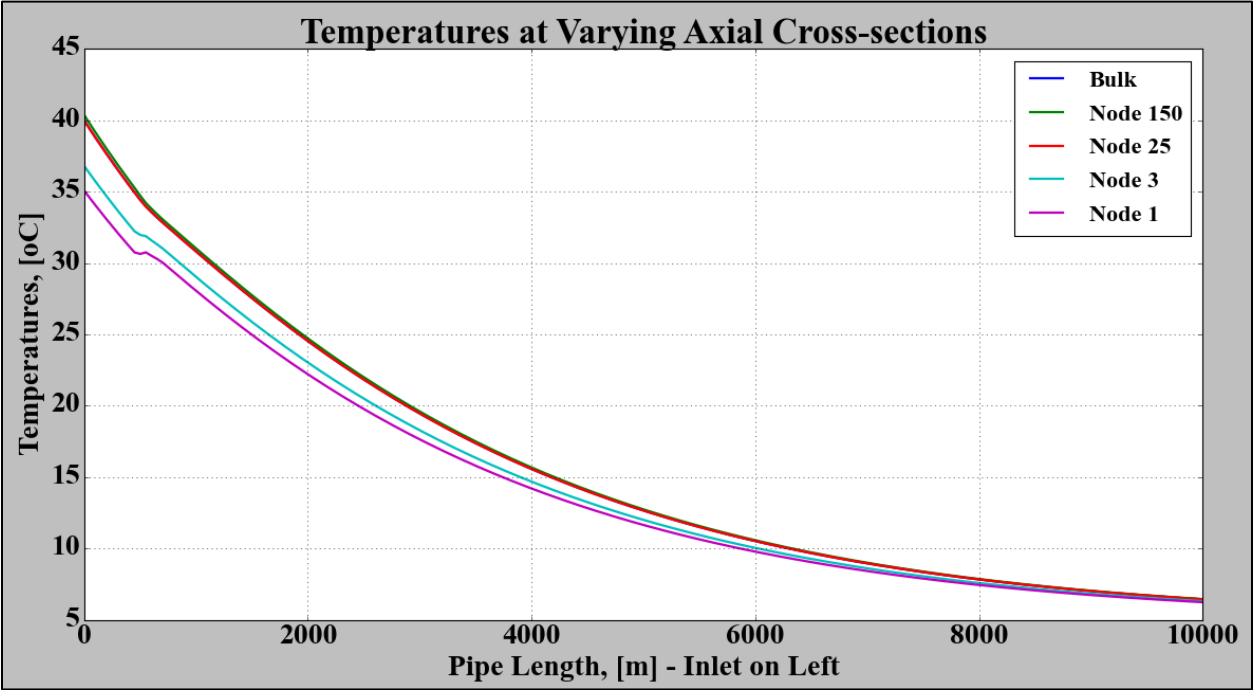


Figure 3.2: Graph showing axial fluid temperature profiles

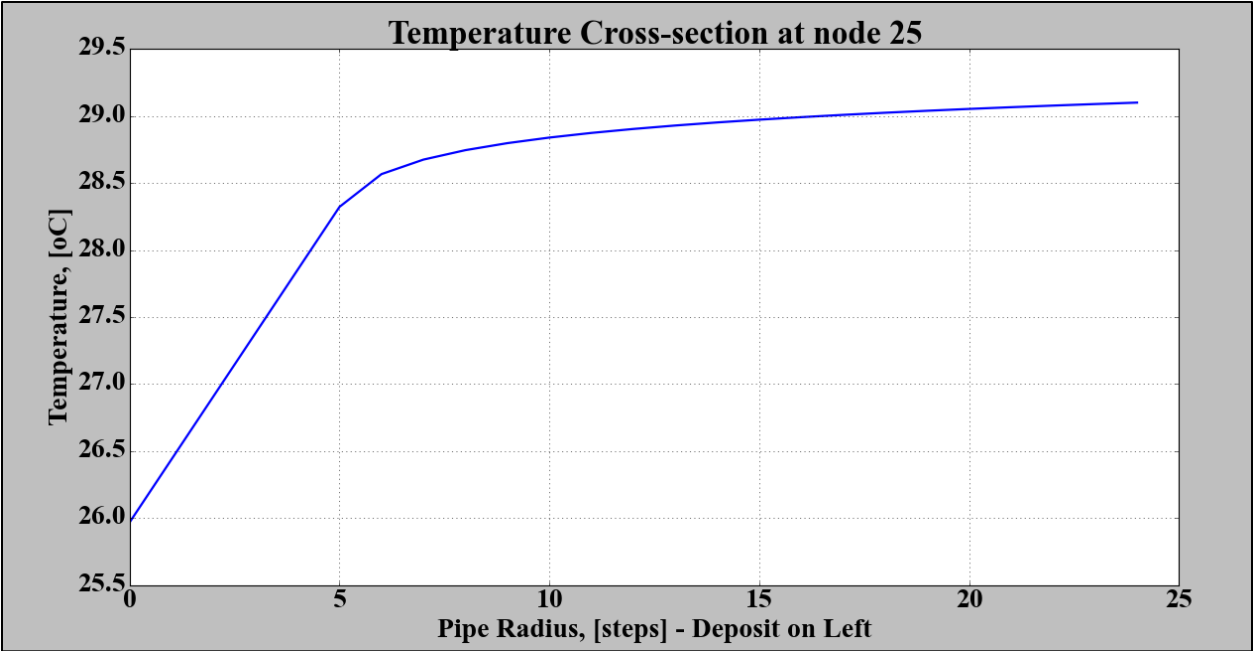


Figure 3.3: Graph showing a radial fluid temperature profile

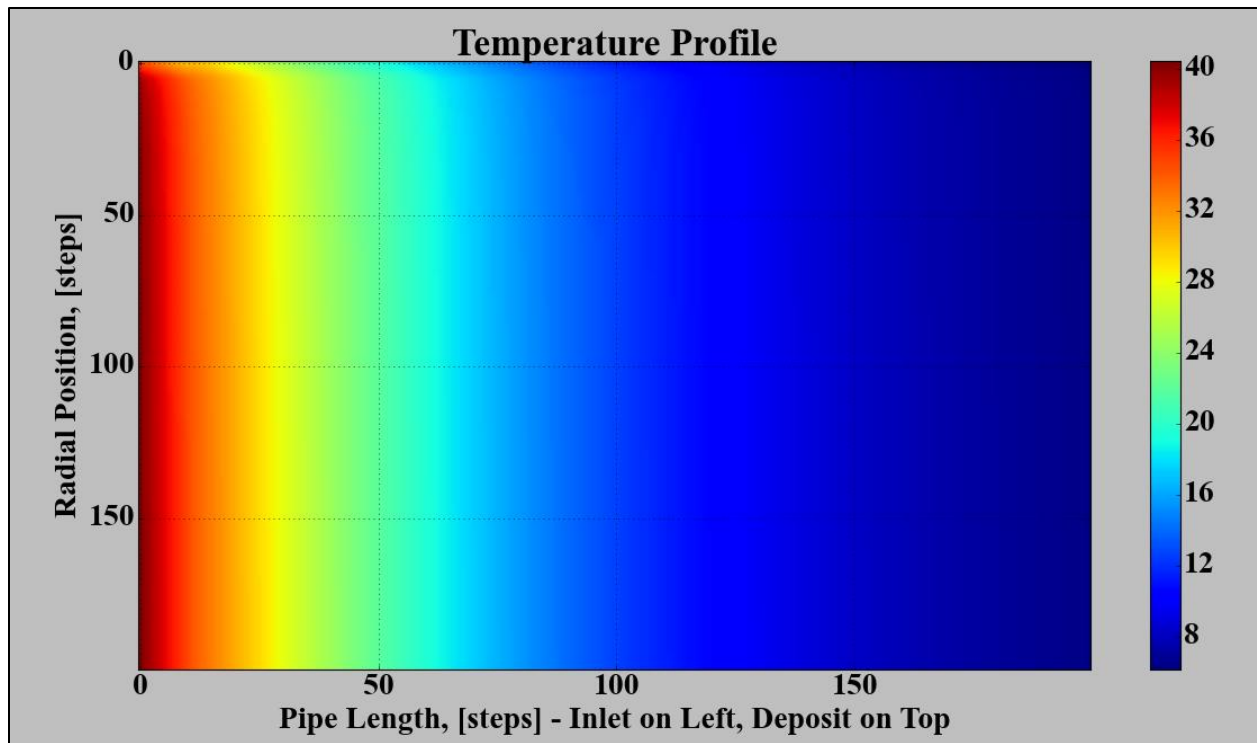


Figure 3.4: Graph showing a 2D temperature segment

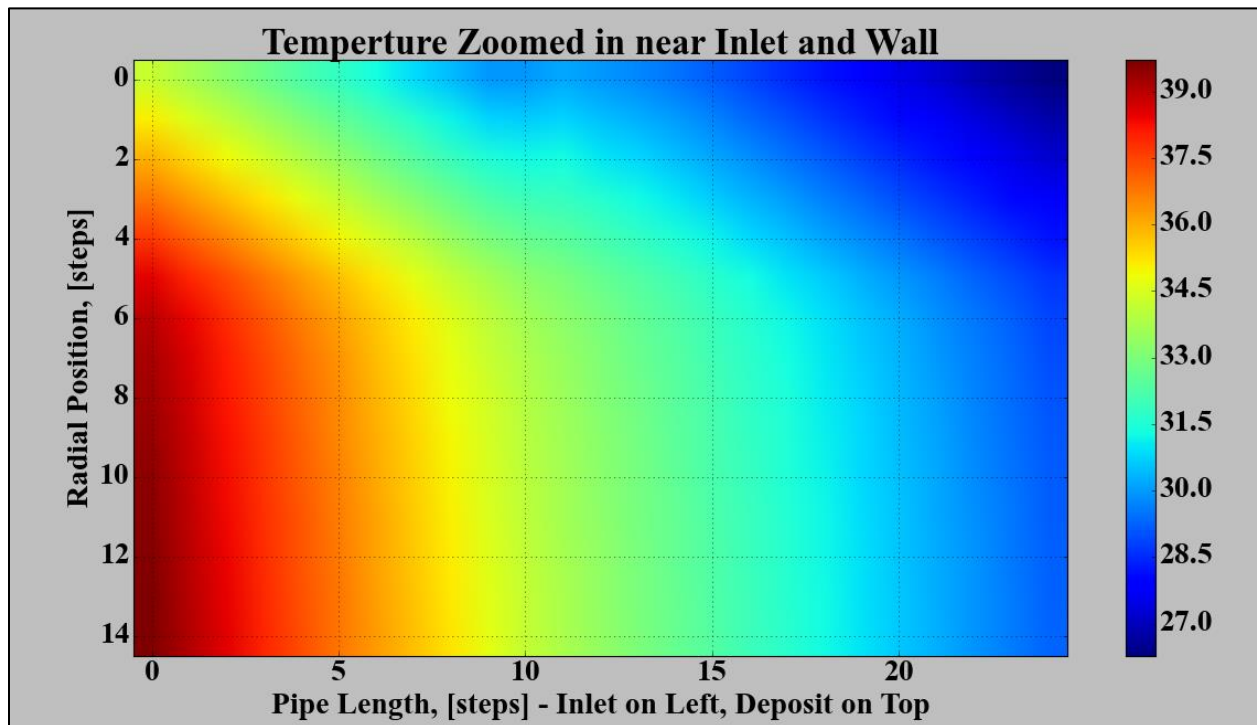


Figure 3.5: Graph showing a 2D temperature segment zoomed in near the wall and inlet

3.1.2 Concentration

This model simplifies the approach used by Huang et al. (2011b) improving computation speed which becomes significant for field pipelines (Eskin et al., 2014). The continuity equations (Eqn. 3.7 and Eqn. 3.8) as shown below, assume axial mixing is negligible. This assumption was justified in the paper and although the model equations were simplified Eskin et al. (2014) claim that the calculation accuracy was maintained. The plus (+) superscript represents a parameter made dimensionless.

$$\frac{1}{y} \frac{\partial}{\partial y} y(\varepsilon_t + a_f) \frac{\partial T}{\partial y} = 0 \quad (3.7)$$

$$\frac{\partial}{\partial y^+} (\varepsilon_m^+ + D^+) \frac{\partial C}{\partial y^+} - k_r \frac{v_f}{u_*^2} (c - c_s) = 0 \quad (3.8)$$

The dimensionless diffusivity is calculated from the correlation below:

$$D^+ = \frac{13.3 \cdot 10^{-12} \tau^{1.47} \mu \left(\frac{10.2}{v_w} - 0.791 \right) v_w^{-0.71}}{v_f} \quad (3.9)$$

The eddy diffusivities were calculated from the following correlation:

$$\varepsilon_t^+ = \left(\frac{y^+}{14.5} \right)^3 \quad y^+ \leq 5 \quad (3.10)$$

$$\varepsilon_t^+ = \frac{y^+}{5} - 0.959 \quad 5 < y^+ \leq 30 \quad (3.11)$$

$$\text{Where: } \varepsilon_m^+ = \frac{\varepsilon_t^+}{Sc_m} \quad (3.12)$$

In order to solve the continuity equation, for the concentration in the near wall region, a solution procedure was employed similar to the one used by Siljberg (2012) for the MWP. This model however, by neglecting axial mixing, results in the matrix calculation for each axial element to be independent of one other. The continuity equation was discretised as below:

$$D_{wo} \left(\frac{C_{j+1} - 2C_j + C_{j-1}}{\Delta y^{+2}} \right) + \frac{D_{wo}}{y_j^{+}} \left(\frac{C_{j+1} - C_j}{\Delta y^{+}} \right) + \frac{k_r v_f}{u^{*2}} (C_j - C_s) = 0 \quad (3.13)$$

Where:

$$D_{wo} = \varepsilon_m^{+} + \frac{D}{v_f} \quad (3.14)$$

In the form of: $AC_j + BC_{j+1} + C_{j-1} = D$, this simplifies to the following equation:

$$C_{j+1} \left(1 + \frac{\Delta y^{+}}{y^{+}} \right) + C_j \left(-2 - \frac{\Delta y^{+}}{y^{+}} - \frac{k_r v_f \Delta y^{+2}}{D_{wo} u^{*2}} \right) + C_{j-1} = \frac{k_r v_f \Delta y^{+2}}{D_{wo} u^{*2}} C_s \quad (3.15)$$

where:

$$A = \left(-2 - \frac{\Delta y^{+}}{y^{+}} - \frac{k_r v_f \Delta y^{+2}}{D_{wo} u^{*2}} \right) \quad (3.16)$$

$$B = \left(1 + \frac{\Delta y^{+}}{y^{+}} \right) \quad (3.17)$$

$$C = 1 \quad (3.18)$$

$$D = -\frac{k_r v_f \Delta y^{+2}}{D_{wo} u^{*2}} C_s \quad (3.19)$$

In order to solve this equation for each axial element a matrix calculation is performed as follows:

$$\begin{pmatrix} 1 & 0 & 0 & & 0 & 0 & 0 \\ C_2 & A_2 & B_2 & \cdots & 0 & 0 & 0 \\ 0 & C_3 & A_3 & & 0 & 0 & 0 \\ & \vdots & & \ddots & \vdots & & \\ 0 & 0 & 0 & & A_{N-2} & B_{N-2} & 0 \\ 0 & 0 & 0 & \cdots & C_{N-1} & A_{N-1} & B_{N-1} \\ 0 & 0 & 0 & & 0 & 0 & 1 \end{pmatrix} = \begin{bmatrix} C_{j,1} \\ C_{j,2} \\ C_{j,3} \\ \vdots \\ C_{j,N-1} \\ C_{j,N} \end{bmatrix} = \begin{bmatrix} C_{wall} \\ D_2 \\ D_3 \\ \vdots \\ D_{N-1} \\ C_{bulk} \end{bmatrix} \quad (3.20)$$

Where the boundary conditions are:

$$c = c_s(T(0)) \quad \text{if } T(0) < T_{wat} \text{ and } y^{+} = 0 \quad (3.21)$$

$$c = c_{wax} \quad \text{if } T(0) \geq T_{wat} \text{ and } y^{+} = 0 \quad (3.22)$$

$$c = c_{wax} \quad \text{if } T(30) \geq T_{wat} \text{ and } y^{+} = 30 \quad (3.23)$$

$$c = c_s(T(30)) \quad \text{if } T(30) < T_{wat} \text{ and } y^{+} = 30 \quad (3.24)$$

The boundary conditions are a set of equations because the Wax Appearance Temperature (shown in Figure 3.6) is not reached for all radial positions at the same axial location. The resulting concentration profiles are shown below in Figure 3.7 and Figure 3.8.

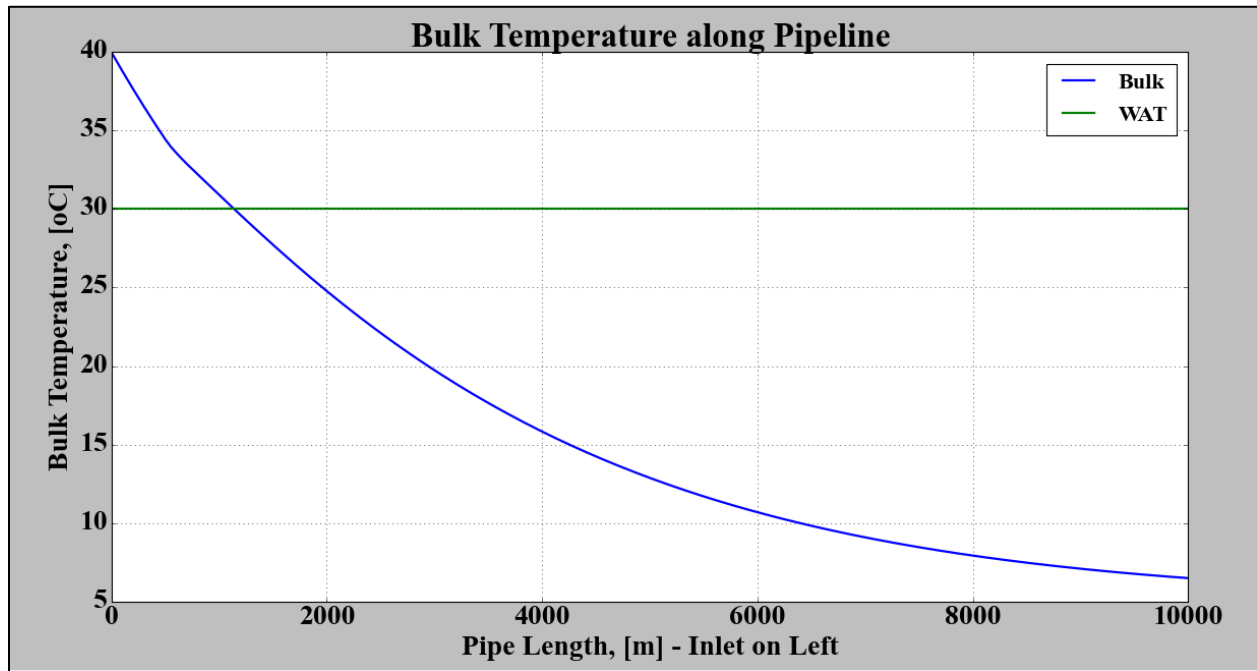


Figure 3.6: Graph showing the bulk temperature profile and the WAT

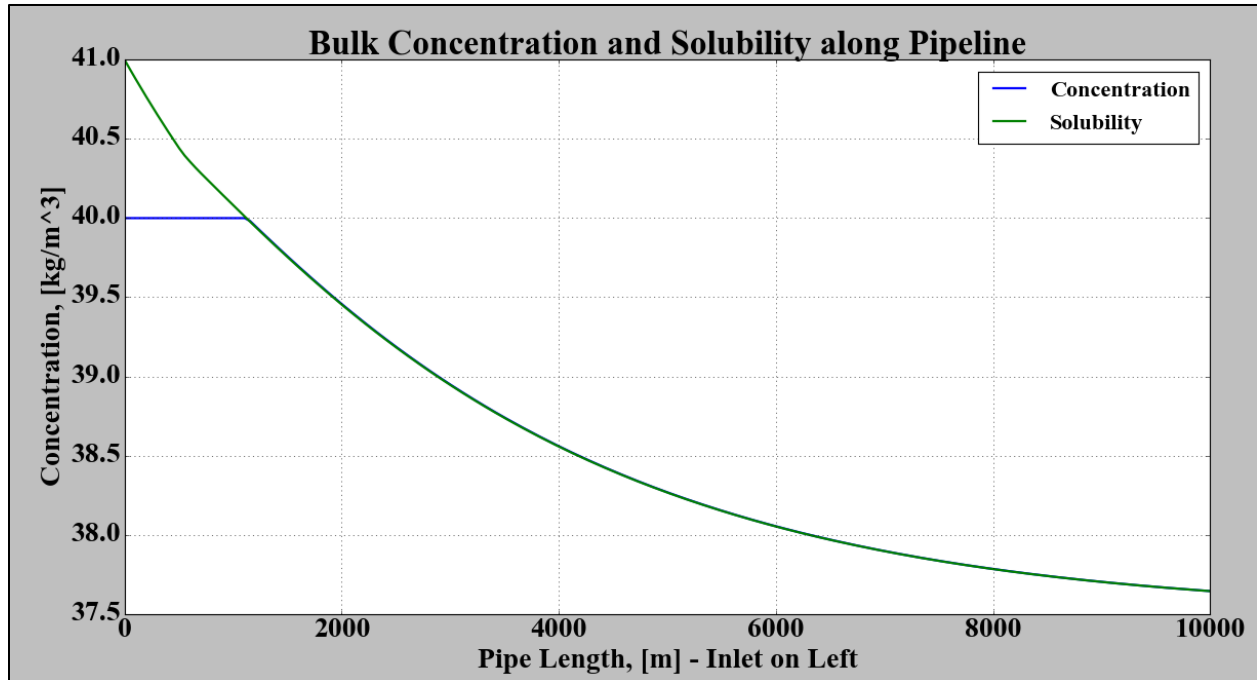


Figure 3.7: Graph showing the bulk concentration profile and solubility profile

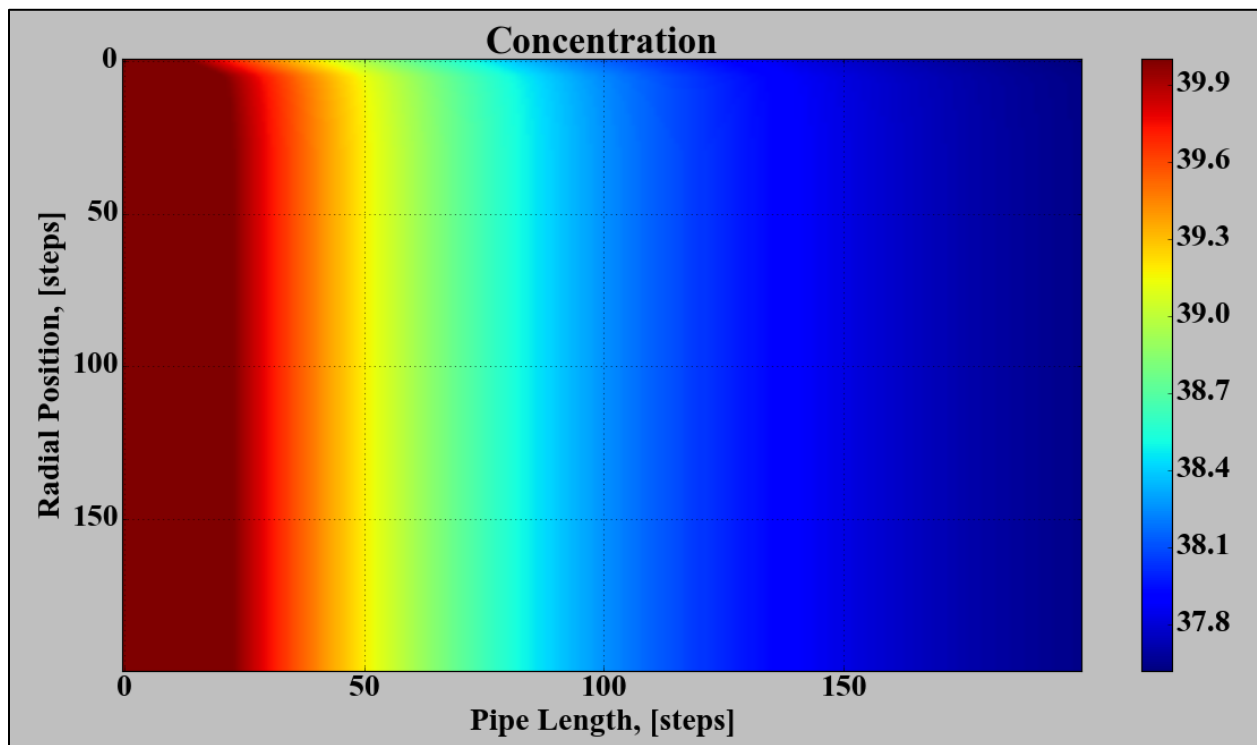


Figure 3.8: Graph showing a 2D concentration segment

3.1.3 Mass-flux

Similarly to Huang et al. (2011b), Eskin et al. (2014) assume wax precipitation occurs. However unlike the MWP's temperature dependent rate constant in their model the precipitation rate constant is fixed for a given deposition case. Figure 3.9 and Figure 3.10 below show the supersaturation resulting from precipitation kinetics:

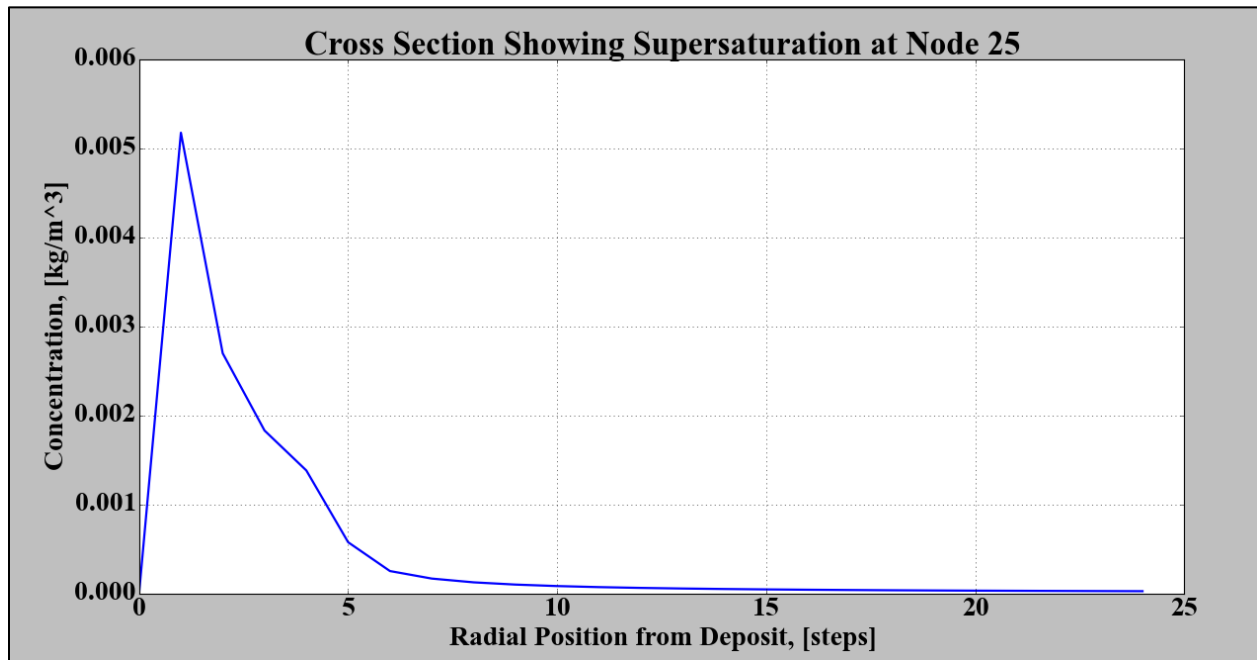


Figure 3.9: Graph showing a radial supersaturation profile

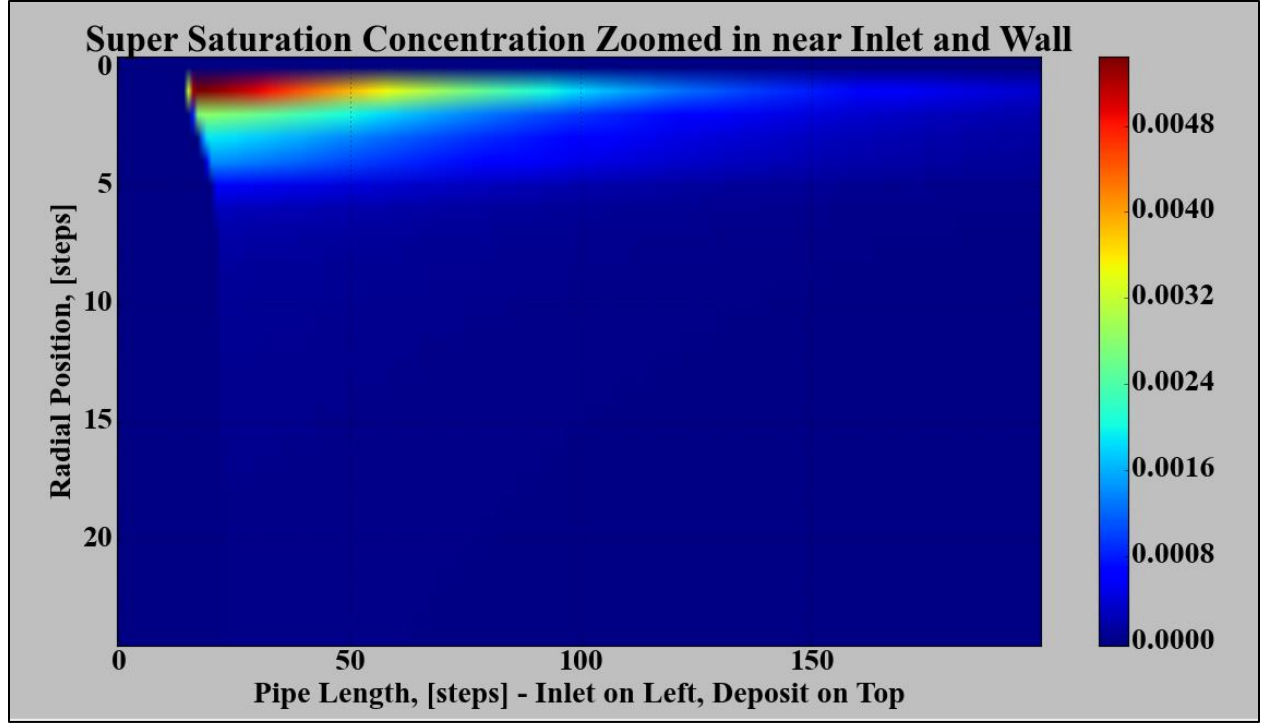


Figure 3.10: Graph showing a 2D supersaturation segment zoomed in near the wall and inlet

In the MWP's equation for the precipitation rate constant, an empirical parameter ($k_{r,cloud}$) was introduced which Eskin et al. (2014) believe is not justified. The Eskin et al. (2014) paper used the following straight line solubility curve:

$$c_s(T) = c_{wax} - \frac{dc}{dT}(T_{wat} - T) \quad (3.25)$$

This results in the following solubility profile in Figure 3.11. In the implementation, the ability to use different solubility curves was introduced by allowing the user to input solubility curve points which calculates the resultant solubility curve. This was needed in order to investigate the significance of shape of the solubility curve.

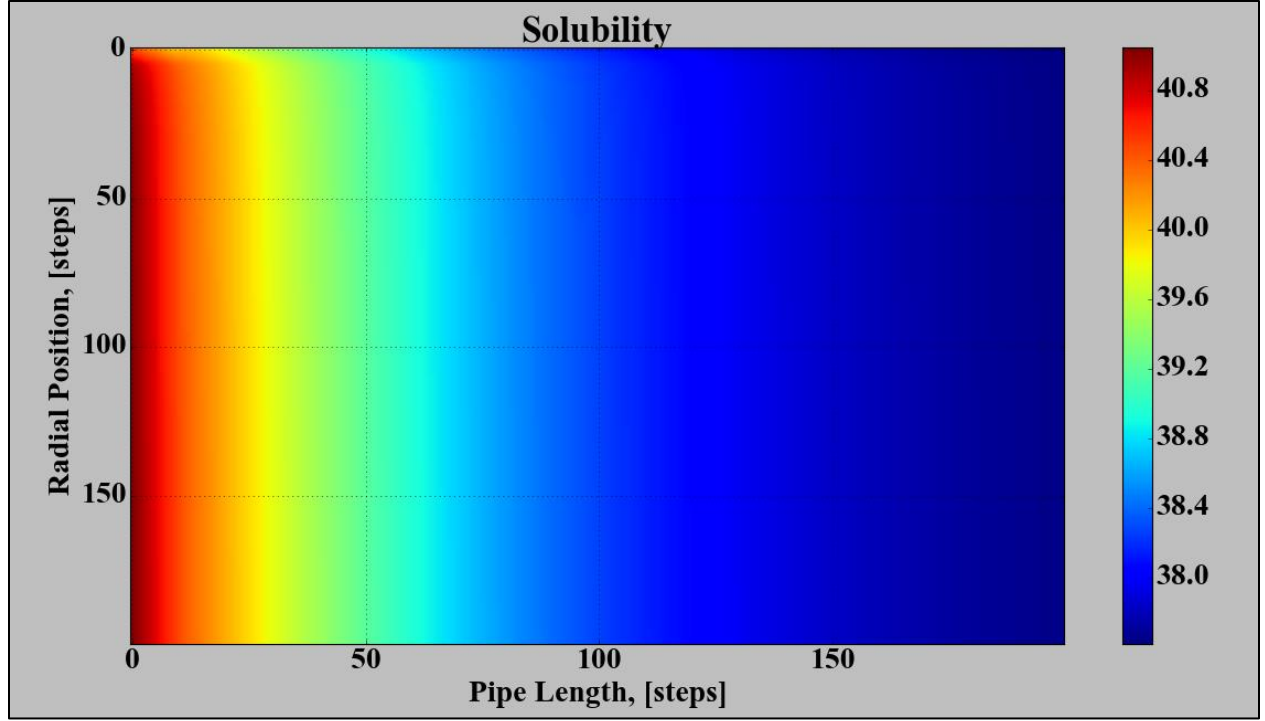


Figure 3.11: Graph showing a 2D solubility segment

The mass-flux (q_t) to the deposit is calculated from the concentration gradient as follows:

$$q_t = D^+ \left. \frac{dc}{dy^+} \right|_{y^+=0} u_* \quad (3.26)$$

3.1.4 Shear Stripping

Removal of wax deposits by shear stripping becomes important with turbulent flow as removal can be significant. It is therefore necessary to incorporate this effect in order to model wax deposition, particularly for high-flow velocities (Aiyejina et al., 2011).

In this model, Shear-stripping was taken into account as seen in Figure 3.12, where q_{sr} is the shear removal flux and q_d is the net deposit flux (Eskin et al., 2014). This is represented by a two-stage process where a long term shear-removal constant, determined from experiment, replaces the calculated constant (Eskin et al., 2014). κ , the shear removal constant, is calculated

from the equation below until it reaches a maximum value determined from experimentation (shown in Figure 3.13). Eskin et al. (2014) acknowledge that they over-simplified the Shear-removal process, as it is not directly dependent on porosity but believe this to only be important for thick deposits (Eskin et al., 2014).

$$\kappa = \left(\frac{\left(\frac{q_t}{q_d} \right) - \varphi(1)^n \left(\frac{\lambda_f}{\lambda_d(\varphi(1))} \right) + \varphi(0)^n \left(\frac{\lambda_f}{\lambda_d(\varphi(0))} \right) - n\varphi(1)^{n-1} \left(\frac{\lambda_f}{\lambda_d(\varphi(1))} \right) \left(1 - \left(\frac{\varphi(1)}{n} \right) \left(\frac{d \ln \lambda_d}{d \varphi} \right) \Big|_{z=1} \right) (v - \varphi(1))}{\Theta} \right) \quad (3.27)$$

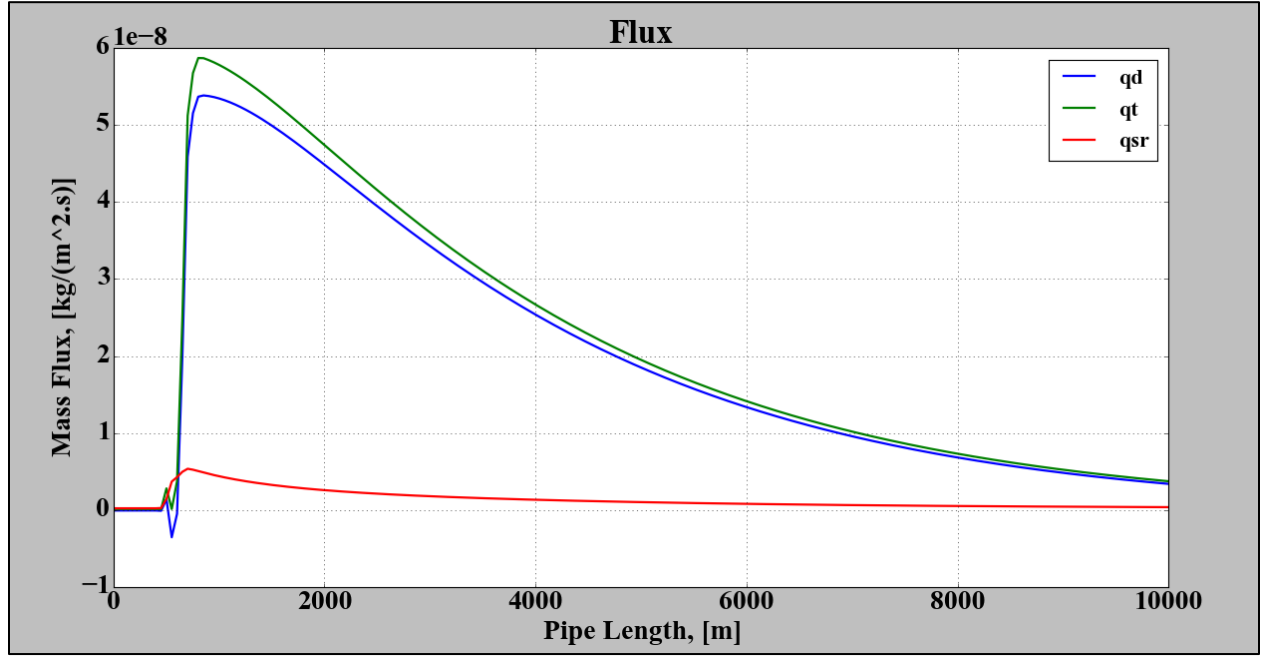


Figure 3.12: Graph showing shear-removal and the resultant mass-flux

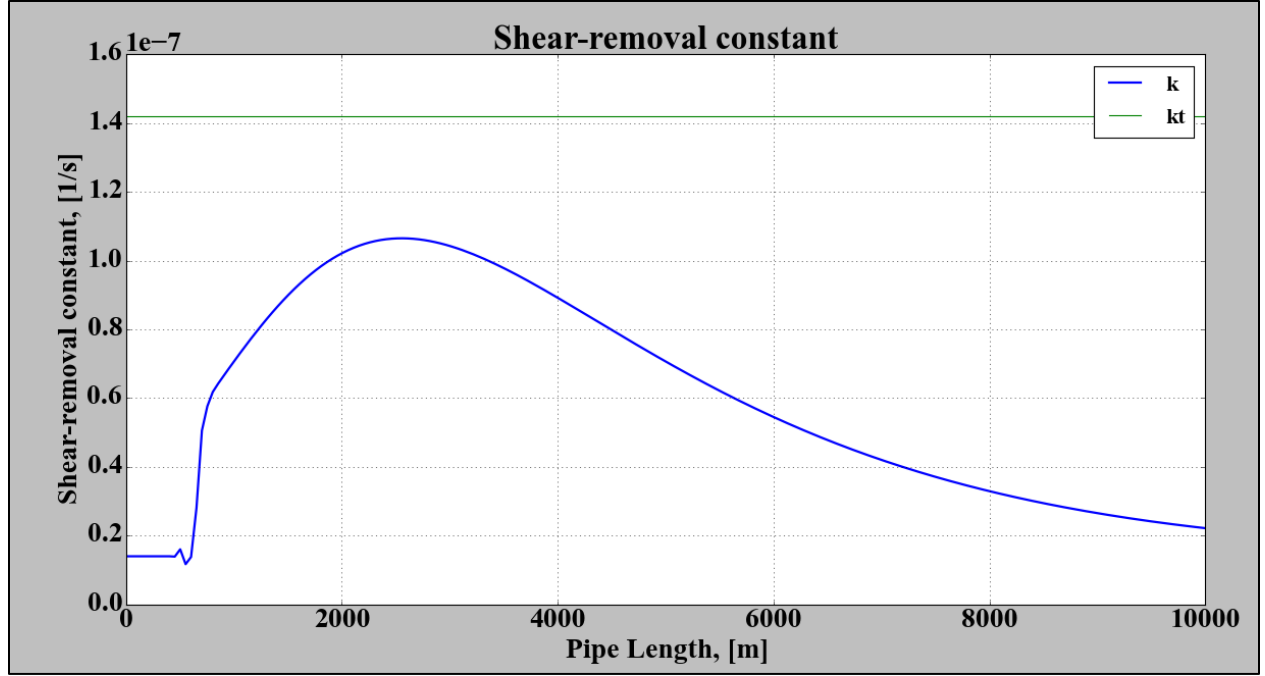


Figure 3.13: Graph showing the shear removal constant

3.1.5 Deposit Model

Eskin et al. (2014) focus strongly on modelling deposit ageing and developed their ageing model based on that of Singh et al. (Eskin et al., 2014). Deposit ageing was assumed as a result of molecular diffusion of wax through the deposit (Eskin et al., 2014). The deposit's porosity is altered by molecular diffusion of wax across the deposit (Eskin et al., 2014). The thermal conductivity of the deposit, as seen in Figure 3.14, is determined with the solid wax considered to be distinct from the hydrocarbon fluid it contains (Eskin et al., 2014). Eskin et al. used the Maxwell equation below to calculate the thermal conductivity; this equation is valid for dilute suspensions of spherical particles (2014):

$$\ln(\lambda_d) = \ln \left(\lambda_f \left(\frac{2\lambda_{wax} + \lambda_f + (\lambda_{wax} - \lambda_f)(1-\phi)}{\lambda_{wax} + \lambda_f - 2(\lambda_{wax} - \lambda_f)(1-\phi)} \right) \right) \quad (3.28)$$

The natural log of this equation was differentiated by parts for use in the following equations where the porosity at the bottom of the deposit is given by:

$$\frac{d\varphi(0,\xi)}{dt} = -(\varphi(1,\xi) - \varphi(0,\xi)) \frac{1}{\Theta} \left(n\varphi(0,\xi)^{n-1} \frac{\lambda_f}{\lambda_d(\varphi(0,\xi))} \left(1 - \frac{\varphi(0,\xi)}{n} \frac{d\ln\lambda_d(0,\xi)}{d\varphi} \right) \right) \quad (3.29)$$

While the porosity at the top of the deposit is given by:

$$\begin{aligned} \frac{d\varphi(1,\xi)}{dt} &= -(\varphi(1,\xi) - \varphi(0,\xi)) \frac{1}{\Theta} \left(n\varphi(1,\xi)^{n-1} \frac{\lambda_f}{\lambda_d(\varphi(1,\xi))} \left(1 - \frac{\varphi(1,\xi)}{n} \frac{d\ln\lambda_d(1,\xi)}{d\varphi} \right) - \Theta(\xi) \frac{d\ln\delta(\xi)}{dt} \right), \\ \xi &= [0, L_p] \end{aligned} \quad (3.30)$$

The porosity is shown in Figure 3.15 and the deposit thickness profile is shown in Figure 3.16 as it varies with time. The porosity allows the calculation of the deposit thickness (δ) from the equation below:

$$\frac{d\ln\delta(\xi)}{dt} = \frac{1}{\Theta(\xi)} \left(\frac{\left(\frac{q_t}{q_d} \right) - \left(\varphi(1,\xi)^n \left(\frac{\lambda_f}{\lambda_d(\varphi(1,\xi))} \right) \right) - \varphi(0,\xi)^n \left(\frac{\lambda_f}{\lambda_d(\varphi(0,\xi))} \right) - \kappa\Theta(\xi)}{v - \varphi(1,\xi)} \right) \quad (3.31)$$

Eskin et al. (2014) state that the MWP does not have a Shear-removal effect (refer to section 8.1.4 for more detail) and has an over simplified model of ageing. They say that the MWP assumes that the deposit porosity is constant and that the deposit flux consists of two parts; one which increases the layer's solid content and another which increases the layer thickness (Eskin et al., 2014).

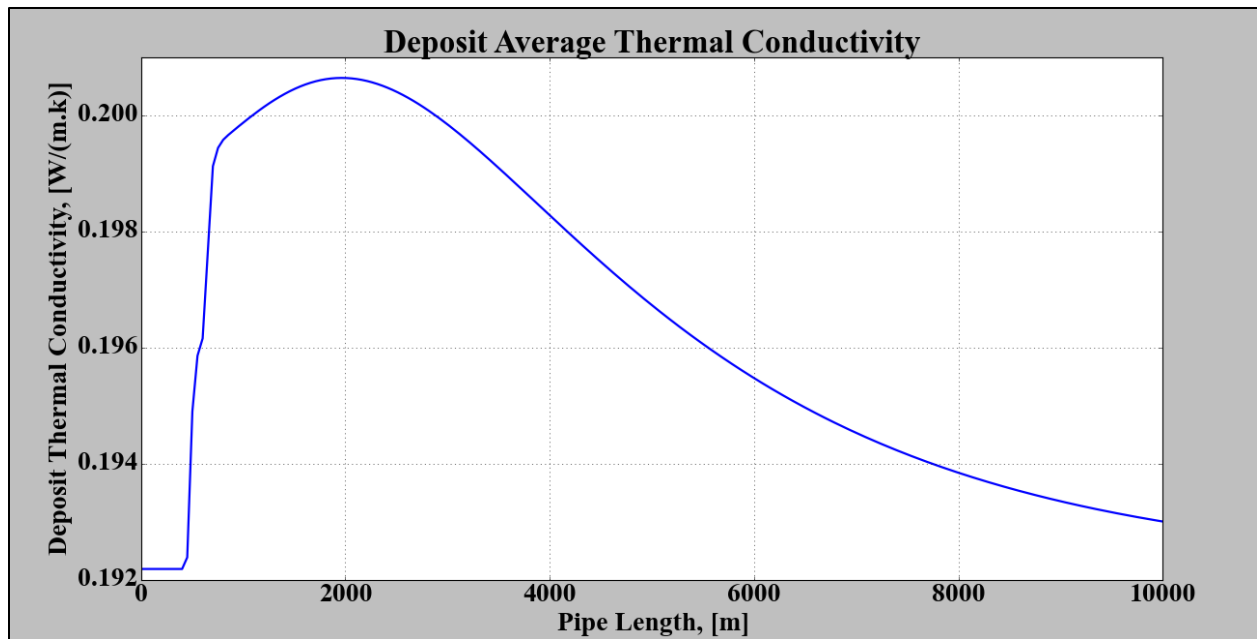


Figure 3.14: Graph showing the average thermal conductivity

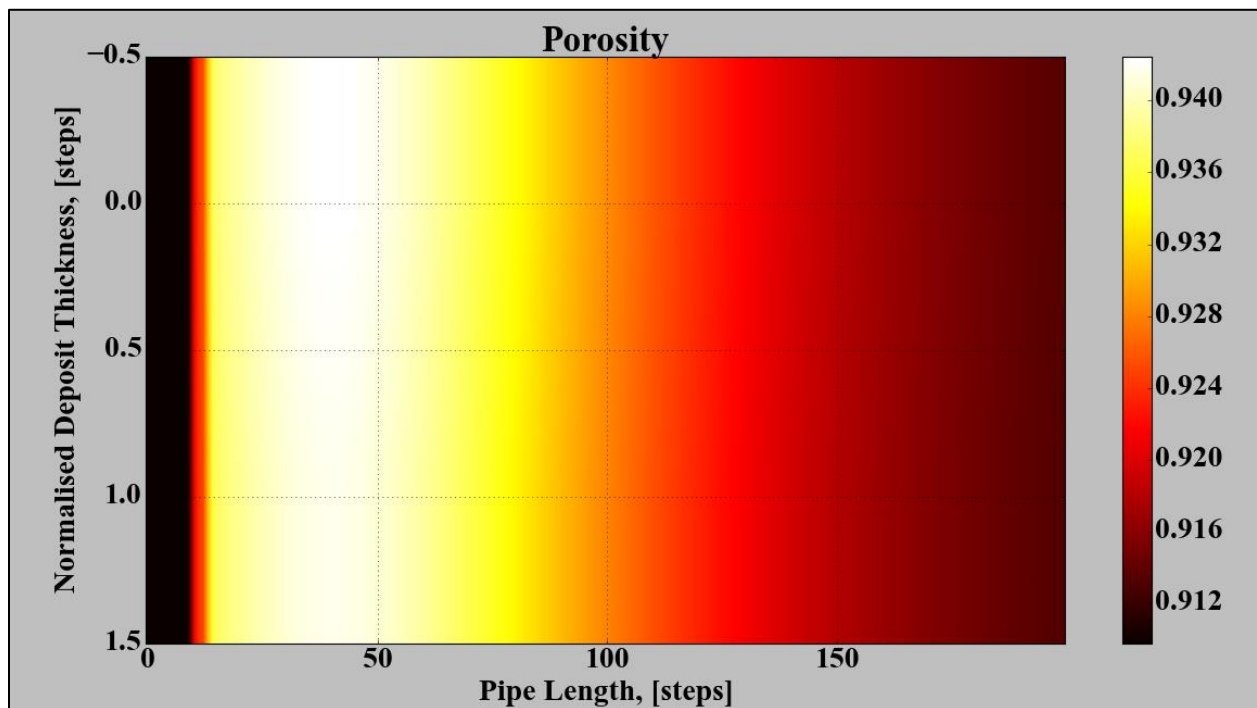


Figure 3.15: Graph showing a normalised porosity

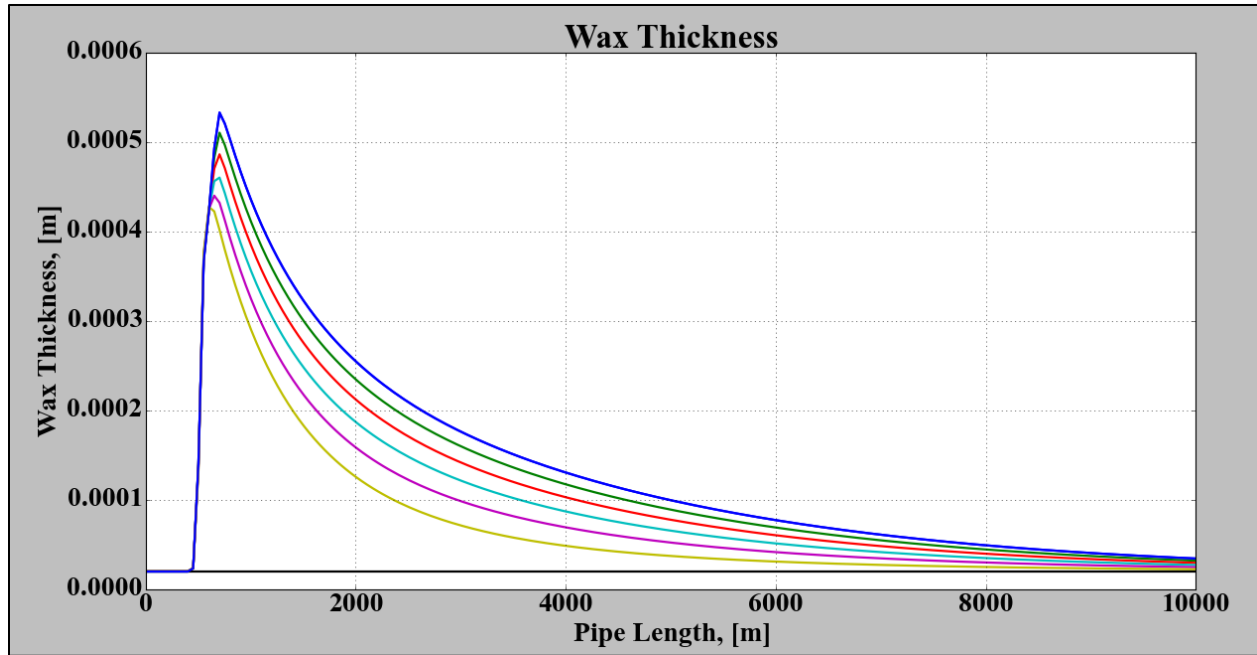


Figure 3.16: Graph showing the deposit thickness varying with time (0-6 days)

3.2 Thermal-difference-based Implementation

It was decided to also implement the model described in the paper ‘A steady-state heat-transfer model for solids deposition from waxy mixtures in a pipeline’ by Haj-Shafiei et al. (2014). This thermal-difference based model assumes deposition to be equivalent to partial freezing resulting from radial heat transfer (Haj-Shafiei et al., 2014). This additional implementation was attempted to better understand and demonstrate this conceptually different approach. The input values for these graphs are provided in Appendix 8.3.1 with additional equations in Appendix 8.3.2.

Oil was approximated, for simplicity, as a combination of $C_{13}H_{28}$ (light solvent) and $C_{29}H_{60}$ (heavy wax) (Haj-Shafiei et al., 2014). The mixture enters the pipe in a single-phase called the Hot-flow Regime, defined as the region where the average bulk temperature is above the WAT and, upon cooling, enters into a Cold-flow Regime (where the average bulk temperature is below the WAT) (Haj-Shafiei et al., 2014).

In the Hot-flow regime the deposit thickness increases along the pipe, as the thermal driving force decreases, until a maximum thickness is reached (Haj-Shafiei et al., 2014). Cold-flow is characterised by two-phase flow of a wax-solvent mixture (Haj-Shafiei et al., 2014). The amount of suspended wax increases axially down the pipe resulting in a decrease in wax layer thickness (Haj-Shafiei et al., 2014).

3.2.1 Thermal Driving Force

The overall thermal driving force influences the heat transfer rate but it is the driving force across each thermal resistive layer that influences the deposit thickness directly (Haj-Shafiei et al., 2014). Table 4.4 in the results, shows that increasing the coolant temperature does not have the same effect as decreasing the inlet temperature. Haj-Shafiei et al. (2014) also assume that the deposit interface temperature is equal to the WAT for the Hot-flow regime and that it decreases during the Cold-Flow regime (Haj-Shafiei et al., 2014).

The pipeline is discretised and a value for the segment outlet temperature is assumed, followed by the iterative solving of the following three heat transfer equations below:

$$q = \dot{m}C_{p,l}^{mix}(T_{hin} - T_{hout}) \quad (3.32)$$

$$q_h = h_h(R - \delta)(2\pi\Delta L)(T_h - T_d) \quad (3.33)$$

$$q_d = \frac{k_d(2\pi\Delta L)(T_d - T_c)}{\ln\left(\frac{R}{R-\delta}\right)} \quad (3.34)$$

T_h is the average between the bulk inlet (T_{hin}) and outlet bulk (T_{hout}) temperatures for a given segment. In the implementation, all temperature dependent parameters were recalculated for each new segment. It is vital to choose the correct order of iteration as the wrong order can result in convergence issues. The equations are rearranged as follows:

$$q = h_h(R - \delta)(2\pi\Delta L)(T_h - T_d) \quad (3.35)$$

$$\delta = R\left(1 - e^{k_d 2\pi\Delta L(T_h - T_d)/q}\right) \quad (3.36)$$

$$T_{hout} = T_{hin} - \frac{q}{mC} \quad (3.37)$$

This calculates the heat-loss (Figure 3.17), the deposit thickness (Figure 3.18), and the segment outlet temperature. The outlet temperature of the segment is assumed equal to the inlet temperature of the following segment; this procedure is followed until the Cold-flow regime is reached.

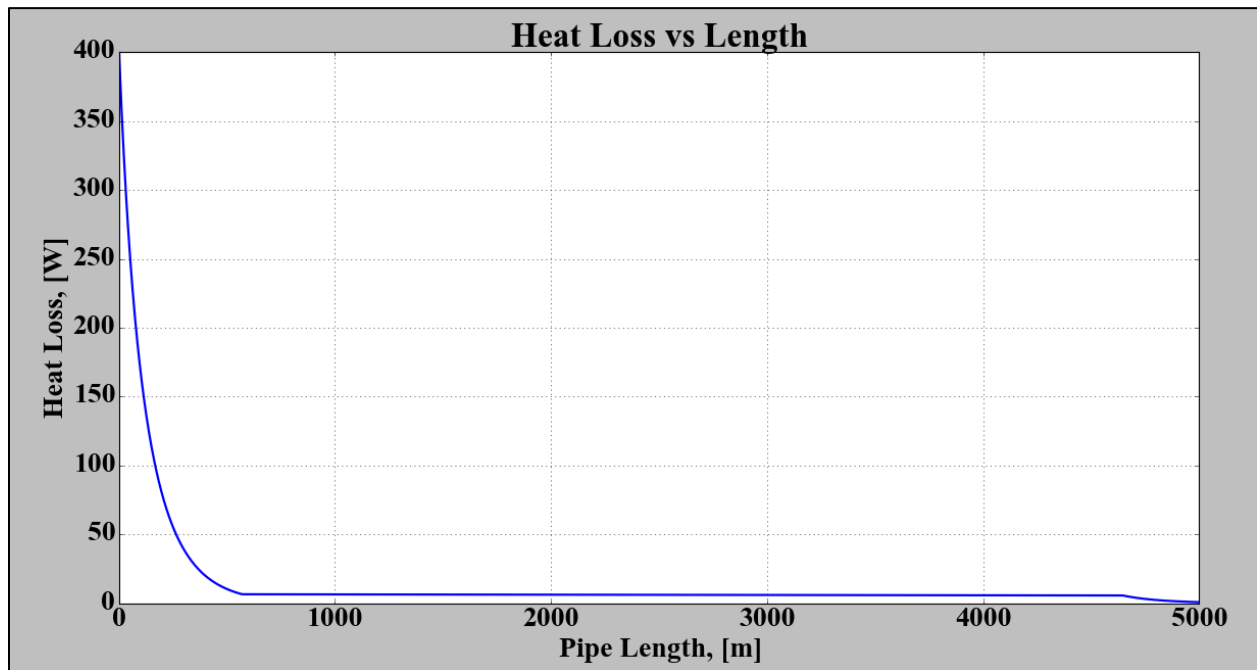


Figure 3.17: Graph showing the heat loss profile

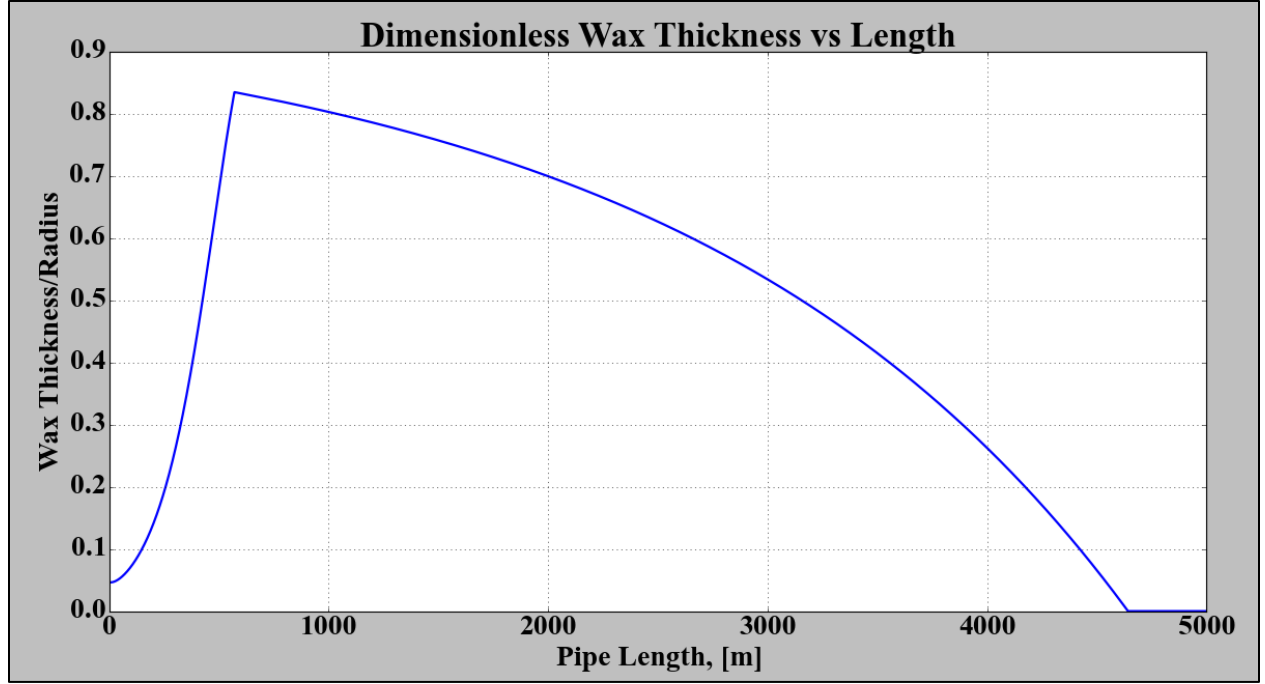


Figure 3.18: Graph showing the wax thickness profile

The pressure drop (Figure 3.19) and cooling rate are calculated as shown below:

$$\Delta p = \left[\frac{2(0.046)\rho_l^{mix}\Delta L}{(D-2\delta)^{5-0.2}} \right] \left[\frac{\mu}{\rho_l^{mix}} \right]^{0.2} \left[\frac{4F}{\pi} \right]^{2-0.2} \quad (3.38)$$

$$\Delta t = \frac{\pi(R-\delta)^2\Delta L}{F} \quad (3.39)$$

$$\chi = \frac{T_{hin}+T_{hout}}{\Delta t} \quad (3.40)$$

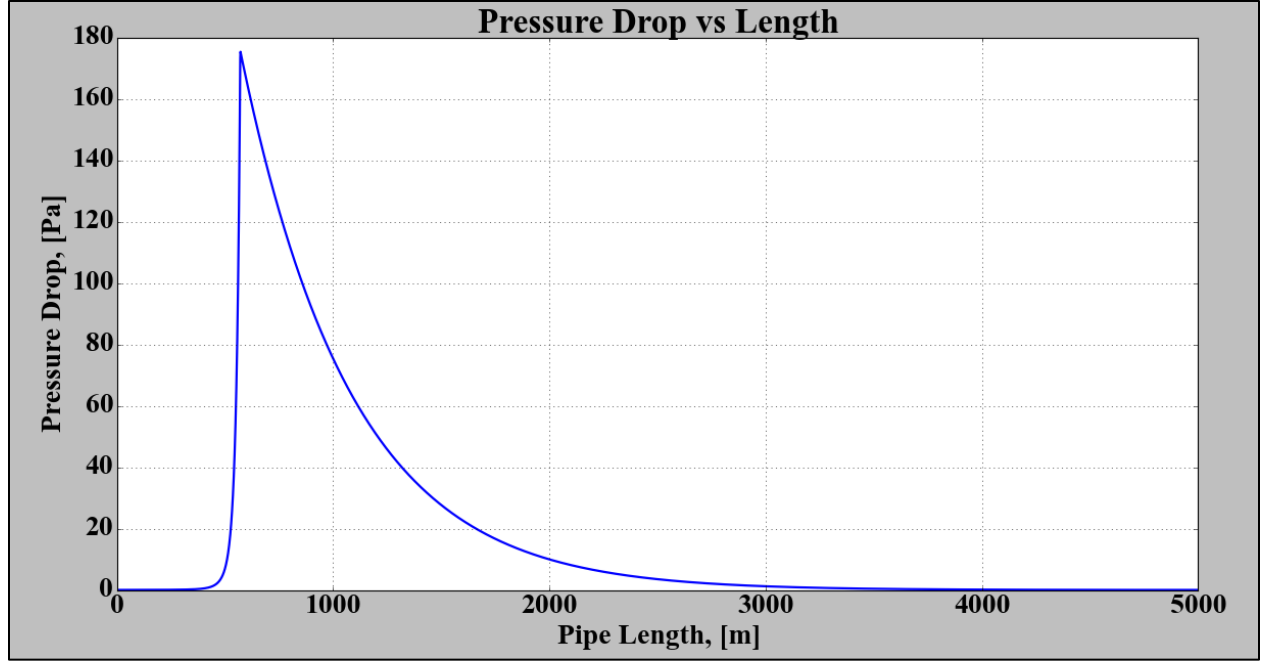


Figure 3.19: Graph showing the pressure drop profile

3.2.2 Cold-flow

When the cooling rate reaches a predefined value, the Cold-flow regime begins. This method of transitioning, besides having a scientific justification, allows the transition between regimes. The bulk temperature converges towards the WAT in the hot-flow regime and never equals it. The bulk temperature by coming within a predefined range of the WAT was used in the implementation for the transition.

The cold-flow regime follows a similar procedure to the hot-flow regime however the enthalpy balance is modified to include the bulk precipitation. The difference between the WAT (now considered equal to the Bulk Temperature) and the deposit surface is chosen, and set here as in the paper at 0.5°K. The fraction precipitated (Figure 3.20 (Mehrotra and Bhat, 2005)) is calculated using the lever-arm rule from the phase diagram (Figure 3.21) for the mixture:

$$q = \dot{m}C_{p,l}^{mix} (T_{hin} - T_{hout}) + \dot{m}\Delta f\Delta H_m \quad (3.40)$$

$$f = \frac{w_{29}(T_h) - w_{29}(*)}{1 - w_{29}(*)} \quad (3.41)$$

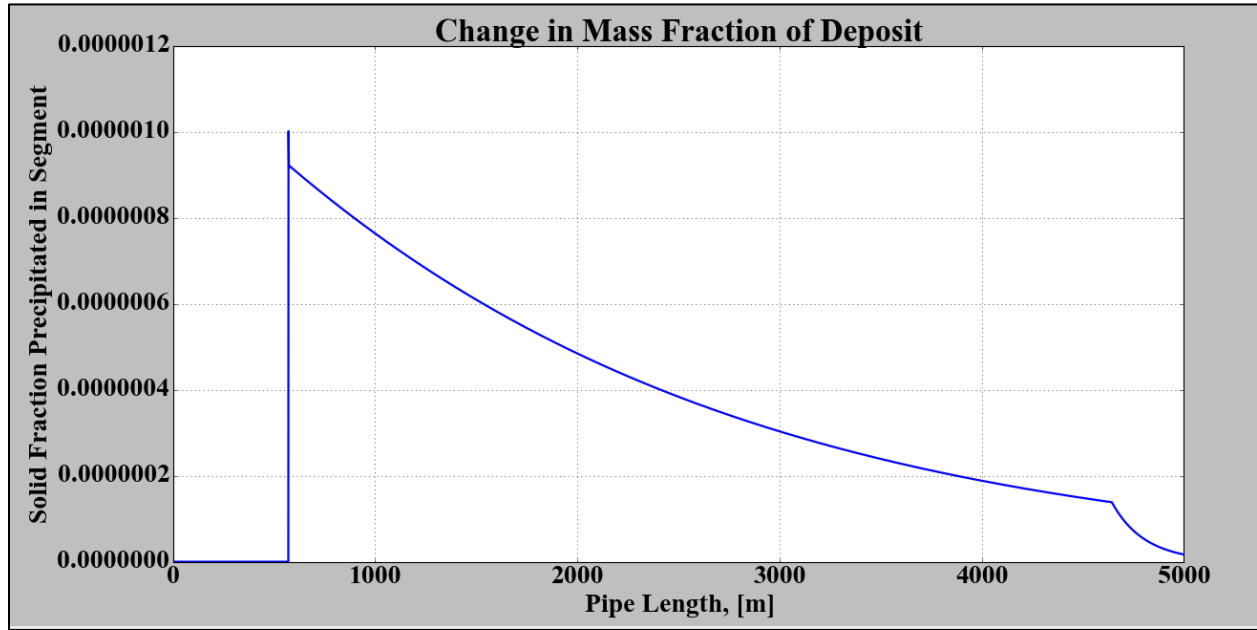


Figure 3.20: Graph showing precipitation

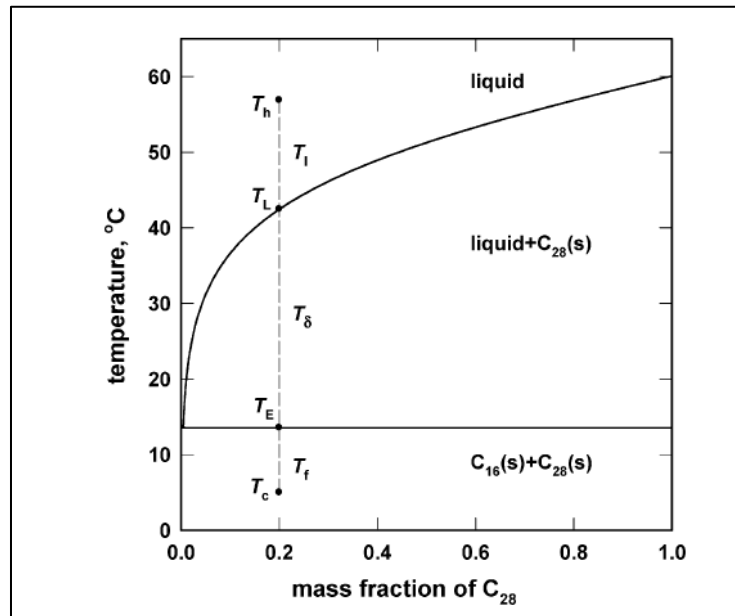


Figure 3.21: Phase diagram (Mehrotra and Bhat, 2005)

3.2.3 Deposit Ageing

Deposit ageing is incorporated into this model via a deformation approach resulting from deposit shear, as shown in Figure 3.22 (Haj-Shafiei et al., 2014). A cubic cage represents the solid wax of the deposit and the volume inside represents the lighter component contained within. Deposit shear results in an increase in solid volume fraction, modelled as the result of the tilting of this cubical cage (Haj-Shafiei et al., 2014). A volume fraction deposit profile without ageing is shown in Figure 3.23. Figure 3.24 shows this ageing which results in a stronger deposit layer (Haj-Shafiei et al., 2014). From their model Haj-Shafiei et al. (2014) also found that deposit ageing because of an increase in deposit thermal conductivity increases the deposit thickness.

$$V_s = \frac{f}{f + (1-f) \frac{\rho_{29}}{\rho_{13}}} \quad (3.42)$$

$$V_{s\beta} = \frac{V_s}{\cos\beta} \quad (3.43)$$

$$\beta = 79.394 + \frac{7.256}{\left[1 + \exp\left(-\frac{(\log Re - 4.411)}{0.220}\right)\right]} \quad (3.44)$$

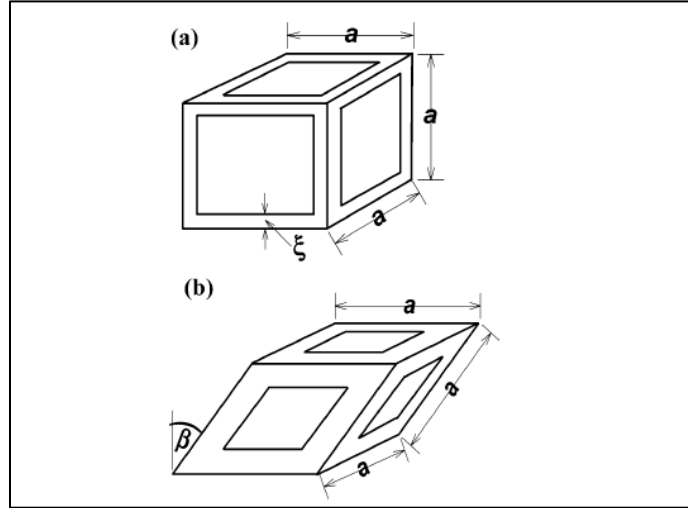


Figure 3.22: Cubical-Cage Representation for Modelling the Effect of Shear Stress (Mehrotra and Bhat, 2007)

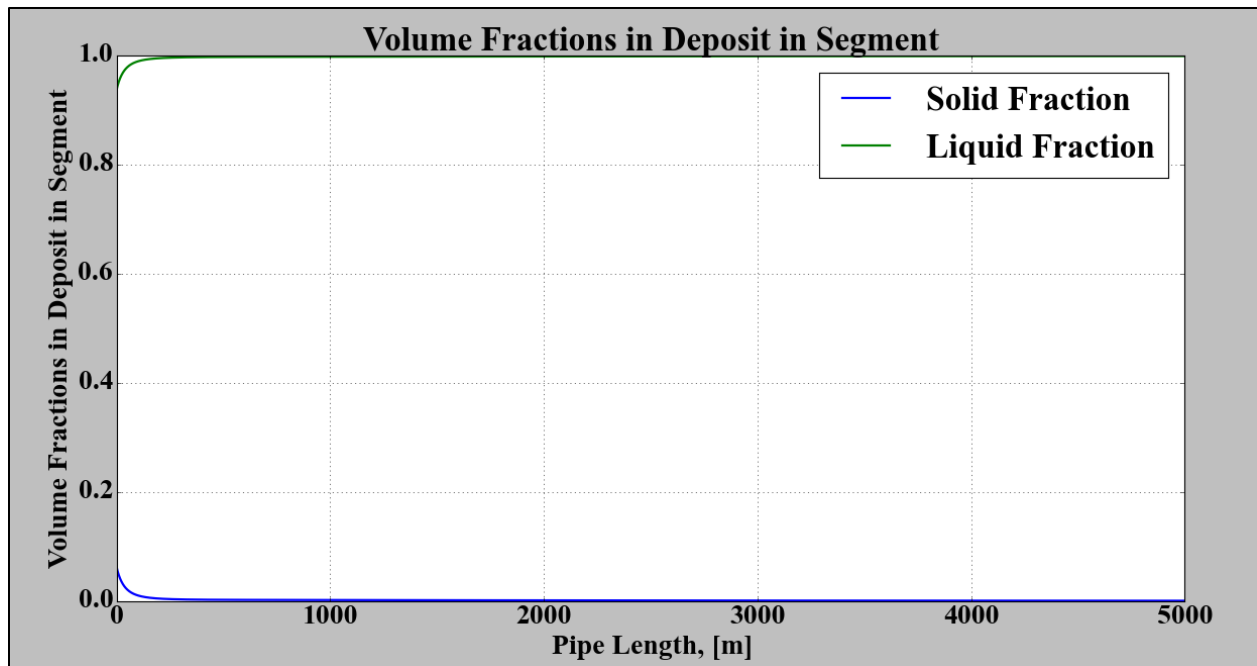


Figure 3.23: Graph showing the deposit volume fraction profile without shear induced ageing

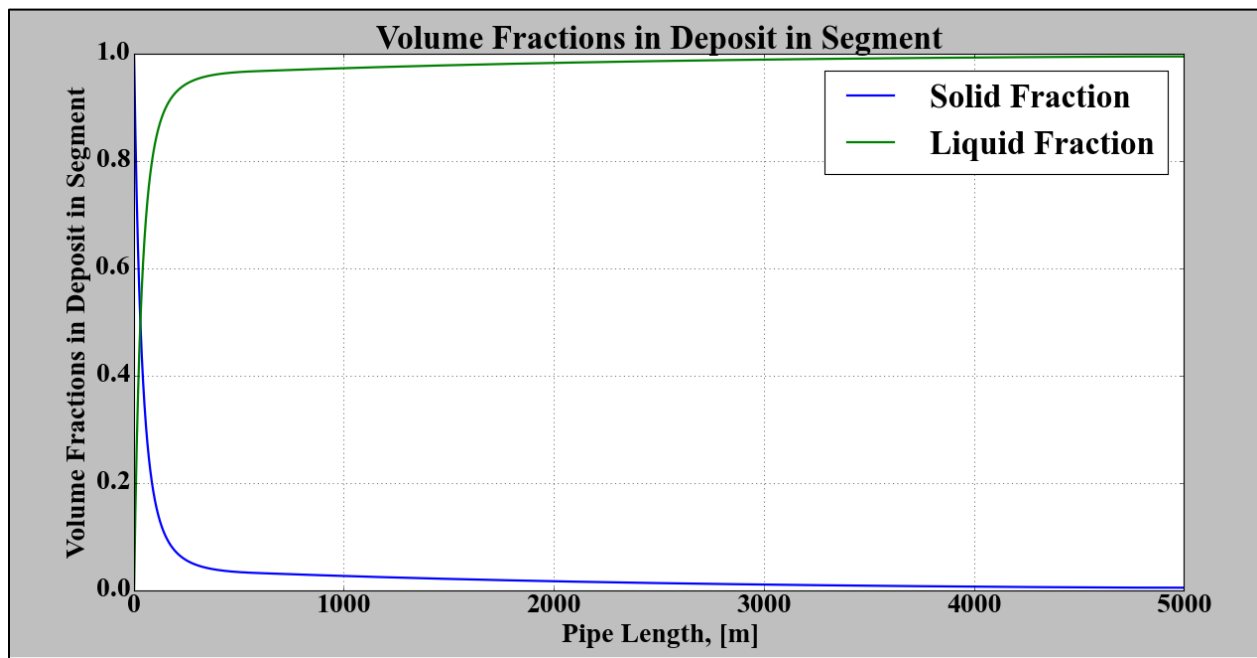


Figure 3.24: Graph showing the deposit volume fraction profile with shear induced ageing

4 RESULTS AND DISCUSSION

This chapter shows the results obtained from simulations performed using the model implementations. These results are then be used to discuss the significance of the shape of the solubility curve on wax deposition, and two important differing assumptions: that of the deposit temperature and the main driving force. The discussion ends with the model implementations being discussed. The Implementations' User Interfaces, accessibility, execution speeds and potential application are explained.

4.1 Results

4.1.1 The Effect of the Shape of the Solubility Curve on Wax Deposition.

In order to explore the effect of the shape of the solubility curve on wax deposition the Mass-flux-based Implementation was run using the default input values (Appendix 8.2.1) with different shaped solubility curves as shown in Figure 4.1 below. The bulk inlet temperature was set at 30°C and the sea temperature at 0°C. The resultant deposit thicknesses after six days are shown in Table 4.1 below.

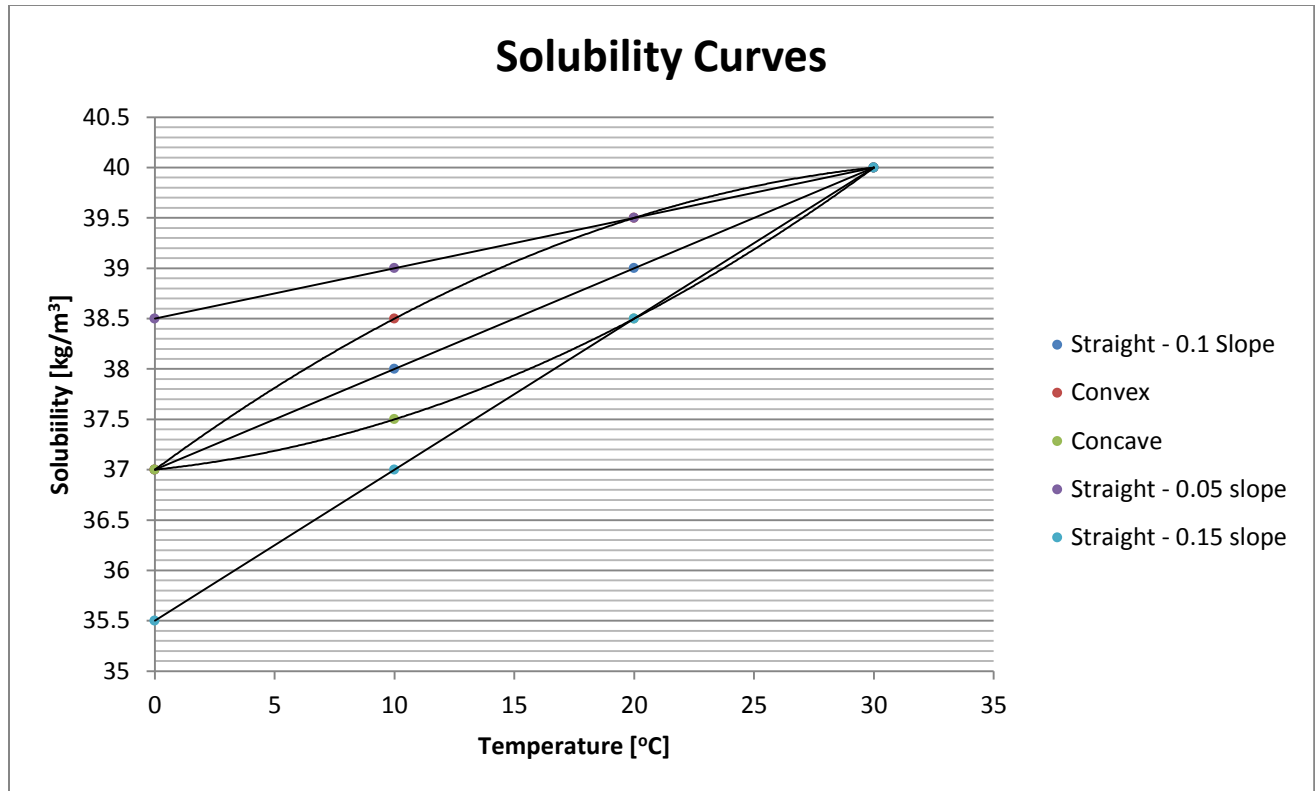


Figure 4.1: Solubility curves used in solubility trial

Table 4.1: Wax Thickness using Different Sloped Solubility Curves

Slope [kg/(m ³ .K)]/ Shape	Average Thickness After six days [mm]	Maximum Thickness after six days [mm]
0.15	0.342	3.015
0.1	0.158	0.735
0.05	0.077	0.241
Convex	0.149	0.282
Concave	0.224	3.268

4.1.2 The Effect of Operating Temperature and Solubility on Wax Deposition

The above trial was run again but with varying operating temperatures. The first additional run was done with a bulk inlet temperature of 20°C and a sea temperature of 0°C. The second additional run was done with a bulk inlet temperature of 30°C and a sea temperature of 10°C.

Two “Plateau” shaped curves were included (Figure 4.2) these curves were capped at the maximum concentration. These runs (Appendix 8.4: Table 8.6 and Table 8.7) allowed for the effect of varying the operating temperature to be investigated, as shown in Table 4.2 and Table 4.3. In these tables both columns represent a decrease in thermal driving force.

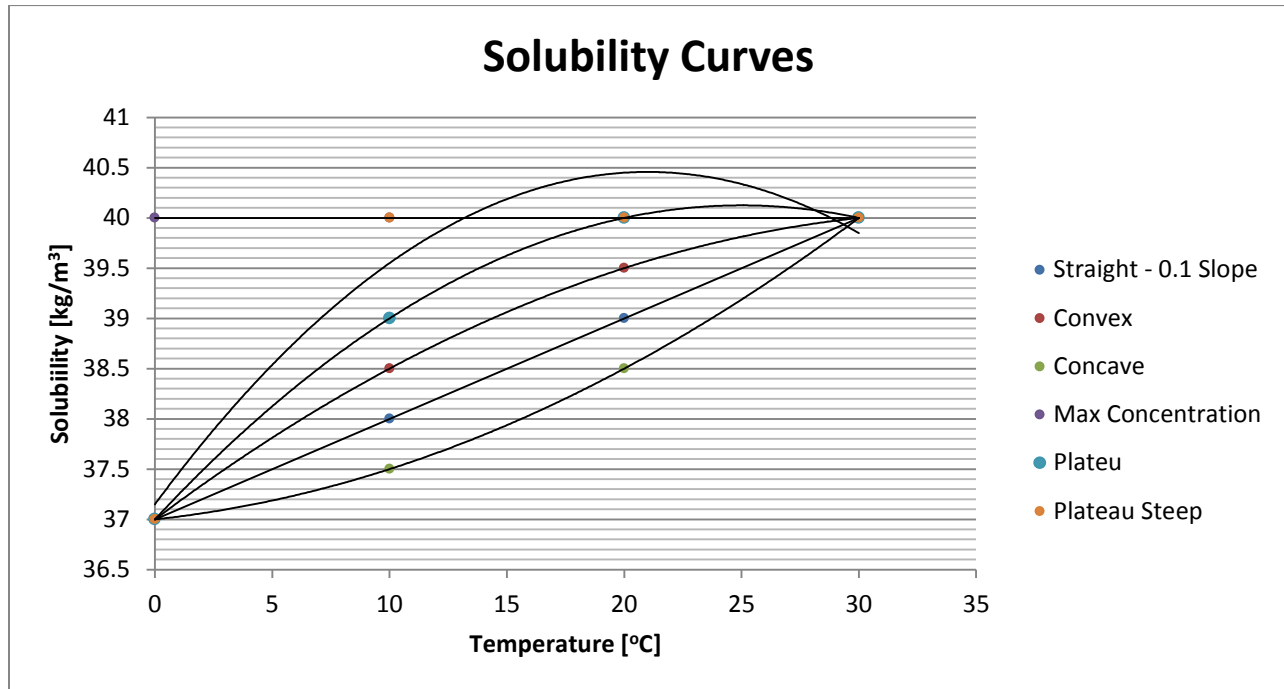


Figure 4.2: Solubility curves used in temperature trial

Table 4.2: Mass-flux-based Implementation’s Changes in Average Thickness with Temperature

Slope [kg/(m ³ .K)]/ Shape	Decrease in thickness due to decreasing inlet temperature from 30°C to 20°C [%]	Decrease in thickness due to increasing sea temperature from 0°C to 10°C [%]
0.1	38.61	35.44
Convex	19.46	44.97
Concave	66.52	38.39
Plateau	7.64	57.32
Plateau - steep	0.61	65.64

Table 4.3: Mass-flux-based Implementation's Changes in Maximum Thickness with Temperature

Slope [kg/(m³.K)]/ Shape	Decrease in thickness due to decreasing inlet temperature from 30°C to 20°C [%]	Decrease in thickness due to increasing sea temperature from 0°C to 10°C [%]
0.1	55.51	52.38
Convex	0.35	48.58
Concave	88.34	72.61
Plateau	0.00	61.65
Plateau - steep	0.00	69.83

Table 4.4: Thermal-difference-based Implementation's Changes in Thickness with Temperature

Thickness	Decrease in thickness due to decreasing inlet temperature from 30°C to 20°C [%]	Decrease in thickness due to increasing sea temperature from 0°C to 10°C [%]
Average	0.00	20.18
Maximum	0.00	87.18

4.2 Discussion

4.2.1 Deposit Temperature Assumption

The model of Haj-Shafiei et al. (2014) is based on the assumption that the deposit surface temperature stays constant (equal to the WAT) for the hot flow regime but decreases for the cold flow regime (Haj-Shafiei et al., 2014). This differs from the assumption of the previously discussed Mass-flux-based models. The mass-flux based models assume a variable temperature at this interface, calculated from an energy balance, which Haj-Shafiei et al. (2014) say has not been validated experimentally. The temperature profiles generated from both model implementations are shown in Figure 3.1 and Figure 4.3.

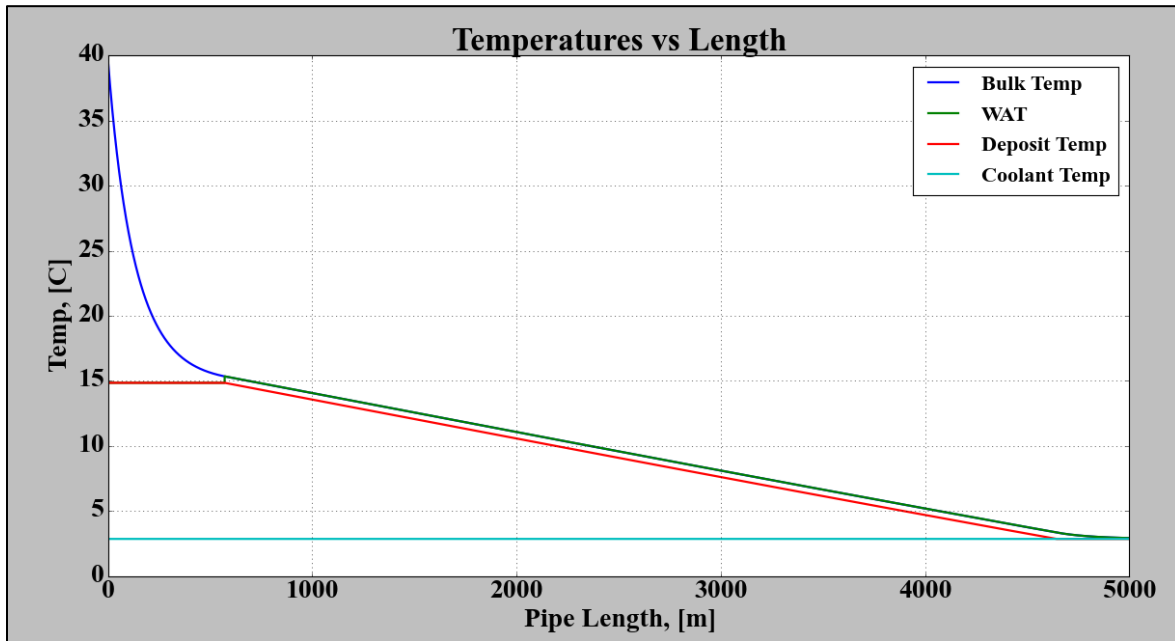


Figure 4.3: The temperature profile from the Thermal-difference-based Implementation

Huang and Fogler (2012) argue that it is not feasible to directly evaluate the deposit surface temperature in a deposition experiment because of the disruption it would cause to the flow and heat-transfer profiles. However an increase in outlet temperature has been found by numerous experiments, to be consistent with an increase in deposit interface temperature, resulting from the insulating effect of deposit build-up (Huang and Fogler, 2012). This effect can be seen with the Mass-flux-based Implementation in Figure 3.1. Huang and Fogler (2012) argue that the assumption that the wax surface temperature remains constant at the WAT violates heat-transfer principles.

Huang and Fogler (2012) claim that the approach of combining the solubility curve with mass-transfer and heat-transfer (mass-flux-based method) is a more fundamental and accurate method for describing the process of wax deposition, as the same governing equations are used for the “hot-flow” and “cold-flow” cases. Huang and Fogler (2012) further state that the lowering of the deposit temperature from the WAT used in the thermal difference studies has the same effect as a solubility curve.

The separate deposit temperature assumptions for either regime used in the Thermal-difference-based Implementation appear to have a similar effect to the Mass-flux-based Implementation's deposit surface's boundary conditions. In the Thermal-difference-based Implementation the wax deposit surface is held at the WAT for the Hot-flow regime and decreases in the Cold-flow regime. This is analogous to the deposit surface boundary condition used in the Mass-flux-based Implementation (Eqn. 3.22). The concentration at the deposit surface is constant at its maximum for deposit temperatures above the WAT; holding the deposit temperature at the WAT should have no effect on this boundary condition until the WAT is reached. In the Cold-flow regime the deposit temperature decreases as does the concentration (following solubility) in the mass-flux based method.

4.2.2 Main Driving Force Assumption

Table 4.1 shows that the greatest amount of wax deposition (highest average wax deposition) results from the steepest solubility curve. The smaller the solubility curves gradient the lower the average amount of wax deposition. The reason for this can be seen in Figure 4.1 of the shape of the solubility curves - the gradient affects the minimum solubility reached.

Interestingly, between the curves that cover the same solubility range (i.e the concave, convex and the 0.1 slope), the concave curve results in the highest average deposition and the convex the lowest average deposition.

The shape of the solubility curve also drastically affects the peak wax thickness. It is useful to explain this phenomenon using their respective deposit profiles below. With the straight line solubility curve (Figure 4.4) the amount of deposition decreases after the bulk temperature reaches the WAT (the bulk inlet temperature is equal to the WAT in these trials). Bulk precipitation diminishes the flux of wax to the deposit. A concave shaped solubility curve compounds the wax peak height by forcing the majority of the wax to come out of solution upon initial cooling, as seen in Figure 4.5. The convex shaped Solubility Curve 'delays' much of the wax deposition further down the pipeline resulting in a less severe wax deposit peak and shifts it

further down the pipeline (Figure 4.6). This is important to flow assurance as a thick wax peak could result in a Pig becoming stuck.

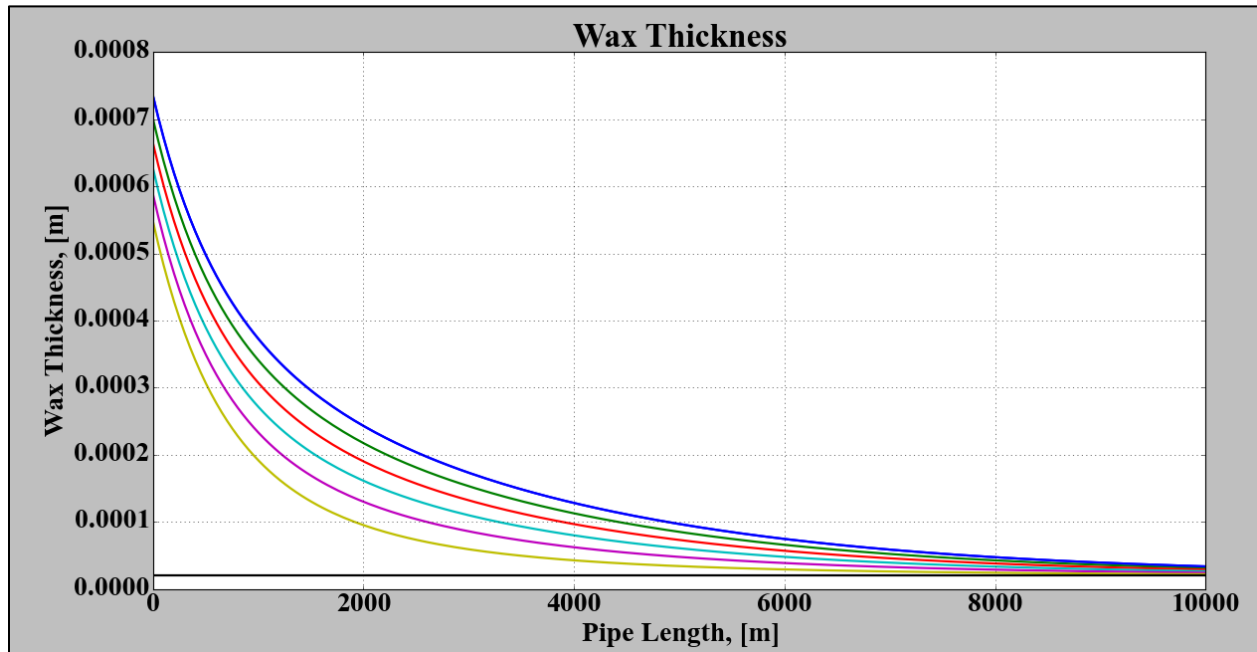


Figure 4.4: Graph showing the deposit thickness profiles with a straight line (0.1) solubility curve

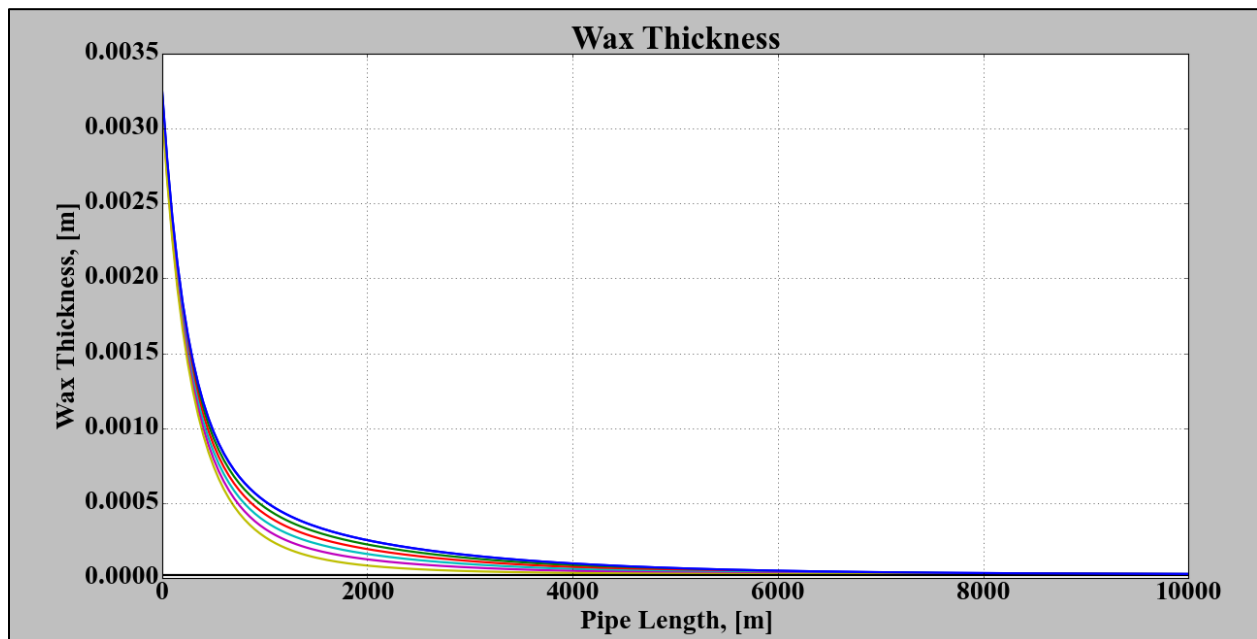


Figure 4.5: Graph showing the deposit thickness profiles with a concave solubility curve

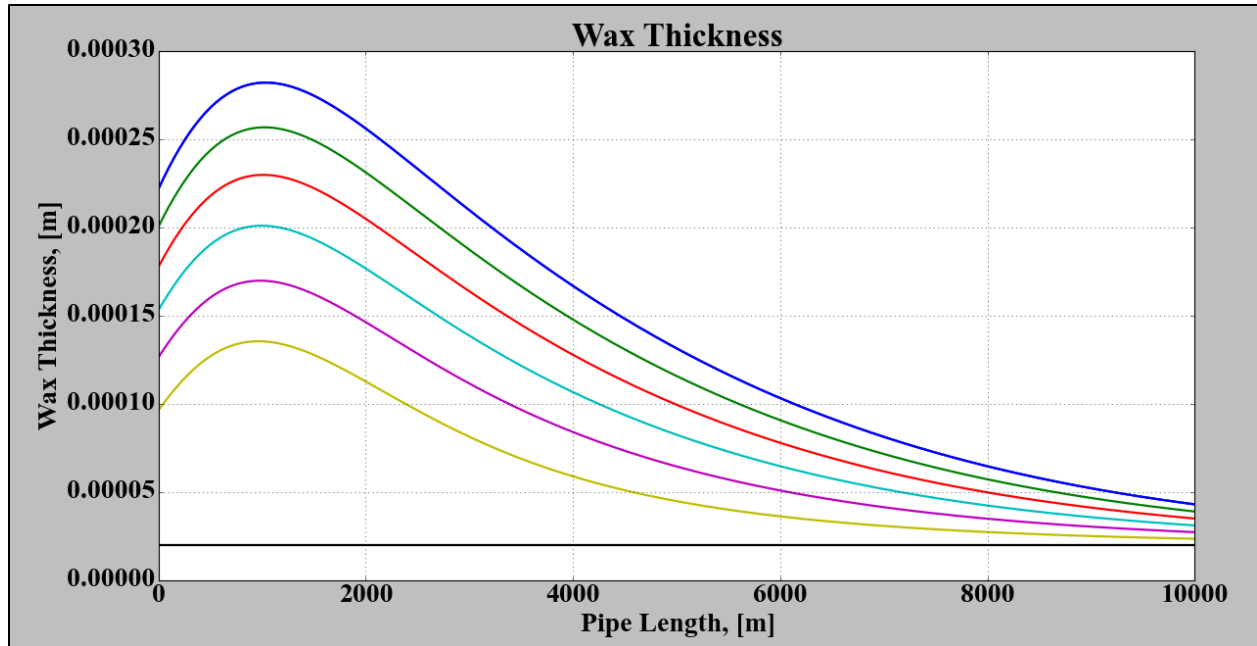


Figure 4.6: Graph showing the deposit thickness profiles with a convex solubility curve

Experiments under differing operating conditions have been conducted to study wax deposition (Huang et al., 2011a). A number of these studies show decreased deposition with a decrease in thermal difference between the bulk oil and the coolant, denoted as the thermal driving force (Huang et al., 2011a). A study by Paso and Fogler showed an opposite trend with more wax deposition occurring with a smaller thermal driving force (Huang et al., 2011a).

An experiment conducted by Huang et al. (2011a) demonstrated both of these trends with decreased deposition found for an increased oil temperature (increased thermal driving force) and increased coolant temperature (decreasing thermal driving force). They attribute these differing trends to the differing shape of the solubility curve of the oil used in the experiments (Huang et al., 2011a). A plateau shaped solubility curve could experience both these trends.

It can be seen in Tables 4.2 and 4.3 in the results section that the reversal of the trend in deposition found by Huang et al. (2011a) with regard to the effect of operating temperature was not found. This model implementation is, however, of a pipeline and not of a flow-loop as in the experiments conducted. This trial should be repeated with different oil and pipeline conditions in

future work in order to see if it is possible to reproduce this trend using this model implementation.

Huang et al. (2011a) claim that the mass driving force and the shape of the solubility curve, not the thermal driving forces are the most important parameters for deposition. They suggest that a change of gradient of the solubility curve can explain how both more and less wax can deposit for a decrease in operating temperature (Huang et al., 2011a). Due to the solubility gradient, for some oils, the thermal driving force can become less significant than that of the effect of the solubility curve (Huang et al., 2011a). Huang et al. (2011a) used the MWP to predict an increase or decrease in wax deposition and say that that using heat-transfer alone cannot predict both trends (Huang and Fogler, 2012).

The results have shown the significant effect of the shape of the solubility curve on wax deposition. Based on the discussion with regard to the deposit surface temperature and the main driving force, it is felt that the mass-flux based approach is an overall more promising methodology for wax deposition prediction.

4.2.3 Model Implementations

4.2.3.1 Jupyter Notebook

Python code can be given a user interface and converted to be a stand-alone program that can run without installing Python to execute it. Recent developments however with the Jupyter Notebook Integrated Development Environment (IDE) allowed for a different method for distribution to be tested as discussed below.

The Jupyter Notebook IDE was chosen because it uses the Web browser as part of its architecture. This also allows Web Technologies such as HTML, CSS and JavaScript to be used in conjunction with Python in order to create rich displays such as the use of different styling, embedding images and adding widget functionality.

The code was made to function as an application to make it user-friendly. It is possible to be used over the internet without the need for installing any additional software. The Jupyter Notebook allowed for the easy creation of a User Interface which allows the model to be used interactively with the code abstracted. The input to the code is through slider bars and input boxes as seen in Figure 4.7 and Figure 4.8 and the graphs are updated automatically.

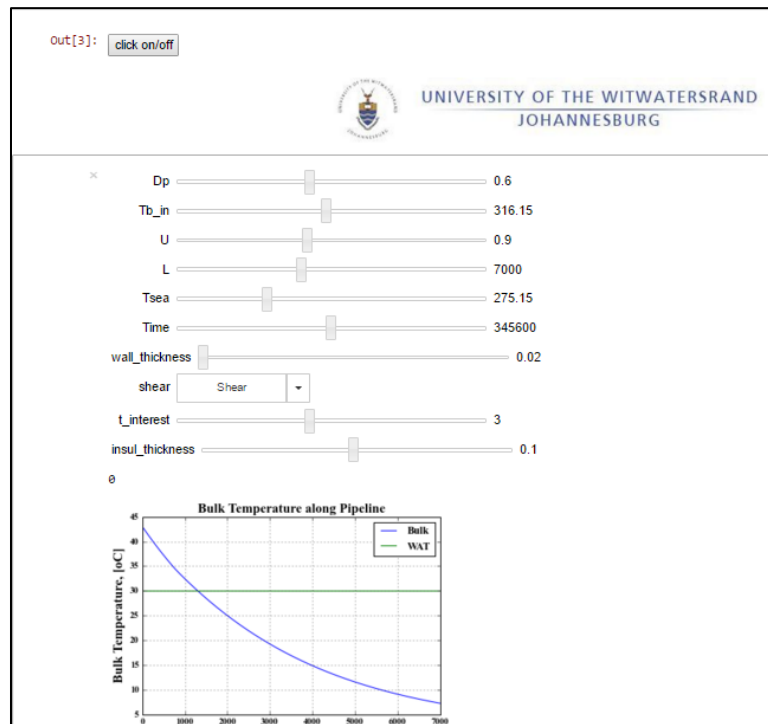


Figure 4.7: The General User Interface for the Mass-flux-based Implementation

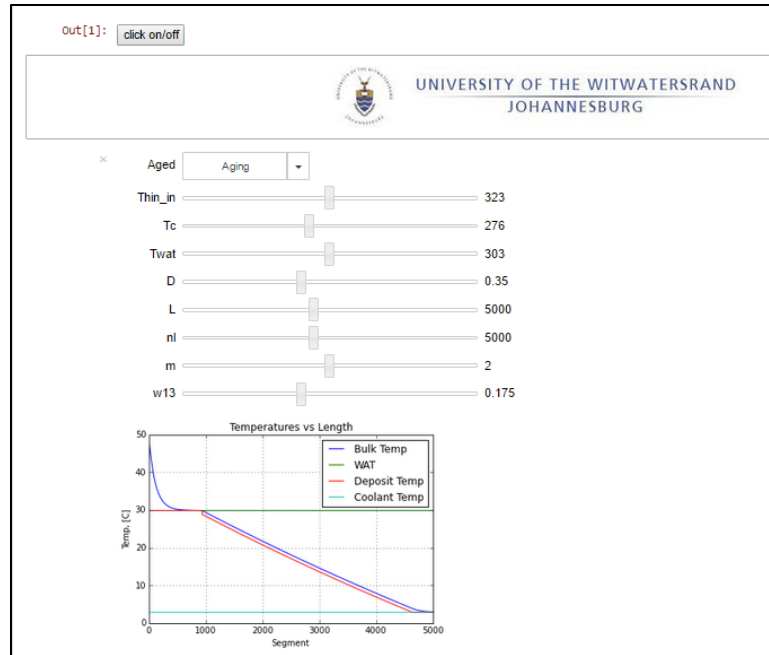


Figure 4.8: The General User Interface for the Thermal-difference-based Implementation

4.2.3.2 Accessibility

It is possible to host Python software with a number of Cloud Computing Services such as Amazon Web Services. These services however require a paid subscription. A personal computer was formatted and installed with an open-source distribution of Linux (Ubuntu) for use as a temporary test server. Ngrok reverse proxy software was used to deal with Port Forwarding and the issue of a dynamically changing local IP address.

Using a temporary server computer posed the problem of a non-fixed web address. To avoid needing to redistribute the web address when it changes, a static website with an easily updatable link to the server was created as seen in Figure 4.9. This website is hosted for free and can be found at <https://bryans01.github.io/wax/>. The link can easily be redirected if the program is moved to another server. The model implementations and server were given password protection but the link will not work until it has been decided to whom or whether to allow access. These

programs could be made openly available or could be used for the basis of an open-source wax deposition program with stake-holder permission.

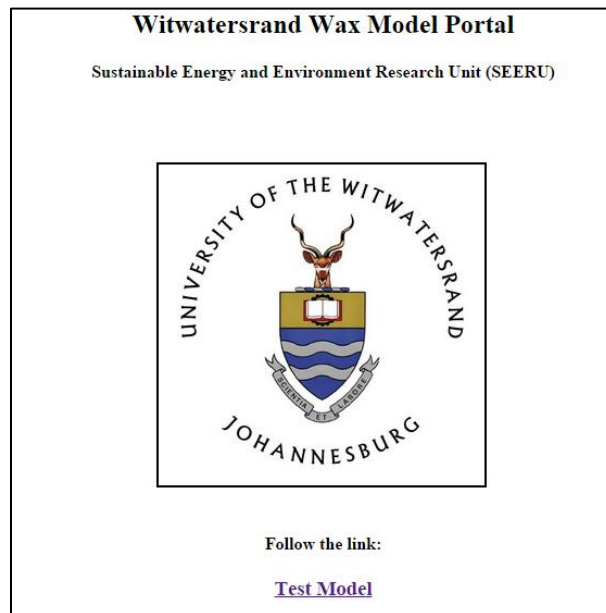


Figure 4.9: Static Website

4.2.3.3 Speed of Execution

Most of the computations are done as arrays (using the Numpy module) as opposed to using loops in order to minimise execution time. This drastically improves the performance as vectorised Numpy operations are performed in C, and array (matrix) calculations greatly reduce the number of calculations performed (Van der Walt et al., 2011). Numpy has found application in many industries (Van der Walt et al., 2011).

Python allows for computationally intensive segments of code to be compiled or substituted for other lower level languages. Compiling the code was attempted using the third party module Numba from Consortium Analytics. It did marginally improve the run time with little effort. Manually compiling code segments should result in better performance.

The implementation of Thermal-difference-based model is very fast as the model only discretises the pipeline in the axial direction. For twenty iterations per segment with a 6000 segments as in the model implementation, the run time is 1.7 seconds and 3.6 seconds with GUI and 9 graphs displayed in the browser.

The mass-flux based model was discretised into a two-dimensional grid pattern from the deposit surface, the temperature and concentration profiles are assumed symmetrical around the pipe's centre axis. With a grid pattern of 200 by 200 and 7 time steps, the code executes in 2.2 seconds without the graphs (13.5 seconds with the GUI and 35 graphs).

4.2.3.4 Application

The resultant implementations were made easy-to-use with general user interfaces and can be accessed through the internet with the University's permission. This could allow somebody interested in understanding the phenomenon of wax deposition to use these recent models.

The model implementations can be used to help illustrate the process of Wax deposition to Engineering Students or someone interested that does not have access to expensive proprietary software and who is not proficient with Programming. The implementations written in the Python Programming language could form the basis for a simple, but easy-to-use open-source wax deposition web application or model repository.

5 CONCLUSIONS

In their review paper, Azevedo and Teixeira (2003) accept molecular diffusion as the dominant paraffin deposition mechanism with gravity settling and shear dispersion not being significant to the process (Stubsjøen, 2012). They argued however that there is not enough evidence to exclude the contribution of Brownian diffusion of solid crystals (Stubsjøen, 2012). Shear forces are still thought to contribute to the removal of wax deposits with wax ageing having a significant effect on model accuracy (Aiyejina et al., 2011). Singh in Stubsjøen (2012) suggested that the overall agreement is that the dominant wax deposition mechanism is molecular diffusion.

This report demonstrates the significance that the shape of the solubility curve has on the wax deposition profile by solving the model equations of Eskin et al. (2014) and Haj-Shafiei et al. (2014). The influence that the solubility curve has on wax deposition was discussed as was the methodology of both models and key assumptions. It is important to understand the differences regarding the main driving force assumption as it results in models following very different solution strategies. Depending on the oil's solubility curve this assumption could result in very different predictions for the Pigging frequency. Based on the discussion with regard to the deposit surface temperature and the main driving force it is felt that the mass-flux based approach is an overall more promising method for wax deposition prediction.

Between the curves that cover the same solubility range, the concave shape exhibits the highest average deposition and the convex the lowest average deposition. The shape of the solubility curve also drastically affects the peak wax height with a concave shaped solubility curve forming the greatest wax peak.

Due to the solubility gradient, for some oils, the thermal driving force can become less significant than that of the effect of the solubility curve (Huang et al., 2011a). The reversal of the trend in deposition found by Huang et al. (2011a) with regard to the effect of operating temperature was not found. This phenomenon should be investigated further in future work.

The phenomenon of Wax Deposition in pipelines is difficult to understand due to the depth and breadth of this field of research. Models are hard to visualise because to their multi-dimensional

algorithms, the interdependence of parameters and the brevity of the explanations in journal articles.

It is hoped that these applications could provide a basis for future wax deposition studies at the University and that the model implementations may help reduce the learning curve for anyone not already familiar with the field. The resultant programs can be used to help illustrate the process of Wax deposition to Engineering Students or anyone interested that does not have access to expensive proprietary software and is not proficient with any Programming Language. Python has been proven to be a suitable language for Wax deposition modelling. The resultant implementations were made easy to use with General User Interfaces and can be accessed through the Internet with the University's permission. This could allow somebody interested in the phenomenon of wax deposition to use these recent models.

Flow assurance has recently progressed faster than the average applied engineering discipline (Gupta and Anirbid, 2015). In the last decade the wax deposition problem has resulted in the creation of flow advisory and monitoring software. These systems rely on robust models for predictions and these software systems are still being developed (Gupta and Anirbid, 2015).

Even with processes that are well understood, many factors affect these processes and their real-life effects cannot always be predicted with great certainty. The best decisions are usually made from experience of the Process. Production decisions, however, often involve millions of dollars. As such, even decisions with low probability of error still involve high risk. The better the tools Engineers have at their disposal the more confident they can be in their decision making. It is therefore important to continue developing better flow assurance software. However; it is felt that there is a need to standardise the methods used to predict paraffin deposition as suggested by Gupta and Anirbid (2015).

6 FUTURE WORK

- Further attempts should be made to reproduce the reversal in trend with the change in operating temperature reported by Huang et al. (2011a) using different operating conditions.
- Attempt to incorporate some of the other significant factors for wax deposition, as described in Appendix 8.1, (eg. multiphase flow) into the Mass-flux-based Implementation.
- More features should be added to the application such as varying the sea-temperature where the pipeline would rise to the surface.
- Different correlations could be used (eg. diffusivity correlations) to test their performance against laboratory or field data.

7 BIBLIOGRAPHY

- Aiyejina, A., Chakrabarti, D. P., Pilgrim, A. and Sastry, M. K. S. (2011). Wax Formation in Oil Pipelines: A Critical Review. *International Journal of Multiphase Flow*, 37, pp. 671-694.
- Azevedo, L. F. A. and Teixeira, A. M. (2003). A Critical Review of the Modeling of Wax Deposition Mechanisms. *Petroleum Science and Technology*, 21(3 and 4), pp. 393-408.
- Bidmus, H. O. and Mehrotra, A. K. (2012). Comments on ‘The Effect of Operating Temperatures on Wax Deposition’ by Huang et al. *Energy & Fuels*, 26, pp. 3963-3966.
- Coutinho, J. A. P., Edmonds, B., Moorwood, T., Szezepanski, R. and Zhang, Z. (2006). Reliable Wax Predictions for Flow Assurance. *Energy & Fuels*, 20, pp. 1081-1088.
- Deo, M., Magda, J. and Roehner, R. (2007). RPSEA 1201 Controlling Wax Deposition in the Presence of Hydrates. *Technology Evaluation Report*, University of Utah Salt Lake City.
- Devaux, T. (n.d.). *Diagram of Pig removing deposits*. [Diagram] Available at: <http://International.iteem.ec-lille.fr/Europe/how-amazing-pigs-are-in-the-oil-and-gas-industry> [Accessed 13 May, 2015].
- Eskin, D., Ratulowski, J. and Akbarzadeh, K. (2014). Modelling Wax Deposition in Oil Transport Pipelines. *The Canadian Journal of Chemical Engineering*, 92, pp. 973-988.
- Galta, T. (2014). *Bypass Pigging of Subsea Pipelines Suffering Wax Deposition*. MSc. Norwegian University of Science and Technology.
- Gupta, A. and Amirbid, S. (2015). Need of Flow Assurance for Crude Oil Pipelines: A Review. *International Journal of Multidisciplinary Sciences and Engineering*, 6(2), pp. 43-49.

- Haj-Shafiei, S., Serafini, D. and Mehrotra, A. K. (2014). A Steady-state Heat-transfer Model for Solids Deposition from Waxy Mixtures in a Pipeline. *Fuel*, 137, pp. 346-359.
- Huang, Z. and Fogler, S. (2012). Response to “Comments on ‘The Effect of Operating Temperatures on Wax Deposition’ by Huang et al”. *Energy and Fuels*, 26, pp. 7390-7392.
- Huang, Z., Lu, Y., Hoffmann, R., Amundsen, L. and Fogler, S. H. (2011a). The Effect of Operating Temperatures on Wax Deposition. *Energy and Fuels*, 25, pp. 5180-5188.
- Huang, Z., Su Lee, H., Senra, M., and Fogler, S. H. (2011b). A Fundamental Model of Wax Deposition in Subsea Oil Pipelines. *AIChE Journal*, 57(11), pp. 2955-2964.
- Huang, Z., Senra, M., Kapoor, R. and Fogler, S. H. (2011c). Wax Deposition Modelling of Oil/Water Stratified Channel Flow. *AIChE Journal*, 57(4), pp. 841-851.
- Kamari, A., Khaksar-Manshad, A., Gharagheizi, F., Mohammadi, A. H. and Ashoori, S. (2013). Robust Model for the Determination of Wax Deposition in Oil Systems. *I&EC Research*, 52, pp. 15664-15672.
- Kjøråas, M. (2012). *Modeling of Wax Deposition in Subsea Pipelines*. MSc. Norwegian University of Science and Technology.
- Lu, Y., Huang, Z., Hoffmann, R., Amundsen, L. and Fogler, H. S. (2012). Counterintuitive Effects of the Oil Flow Rate on Wax Deposition. *Energy & Fuels*, 26(7), pp. 4091-4097.
- Mehrotra, A. K. and Bhat, N. V. (2005). Modeling of Deposit Formation from “Waxy” Mixtures via Moving Boundary Formulation: Radial Heat Transfer under Static and Laminar Flow Conditions. *Industrial & Engineering Chemistry Research*, 17, pp. 6948-6962.

- Mehrotra, A. K. and Bhat, N. V. (2007). Modeling the Effect of Shear Stress on Deposition from “Waxy” Mixtures under Laminar Flow with Heat Transfer. *Energy & Fuels*, 21, p. 1227-1286.
- Peréz, F. and Granger, B. E. (2007). IPython: A System for Interactive Scientific Computing. *Computing in Science & Engineering*, 9(21), p. 21-29.
- Pigtek. (n.d.). *Dewaxing Pigs*. [Photograph] Available at: <http://www.pigtek.com/products/dewaxing-pigs> [Accessed 13 May 2015].
- Ramirez-Jaramillo, E., Lira-Galeana, C. and Manero, O. (2004). Modeling Wax Deposition in Pipelines. *Petroleum Science and Technology*, 22(7 and 8), pp. 821-861.
- Saoga. (2015). *Upstream Oil & Gas in South Africa | SAOGA*. [online] Available at: <http://www.saoga.org.za/oil-gas-hubs/upstream-oil-gas-south-africa> [Accessed 2 June 2015].
- Siljubergh, M. K. (2012). *Modelling of Paraffin Wax in Oil Pipelines*. MSc. Norwegian University of Science and Technology.
- Singh, P., Venkatesan, R. and Fogler, H. S. (2000). Formation and Ageing of Incipient Thin Film Wax-Oil Gels. *AIChE Journal*, 46(5), pp. 1059-1074.
- Singh, P., Venkatesan, R., Fogler, H. S. and Nagarajan, N. R. (2001). Morphological Evolution of Thick Wax Deposits During Ageing. *AIChE Journal*, 47(1), pp. 6-18.
- Van der Walt, S. S., Colbert, C. and Varoquaux, G. (2011). The Numpy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(22), pp. 22-30.

Stubsjøen, M. (2012). *Wax Modeling in Subsea Oil Pipelines*. MSc. Norwegian University of Science and Technology.

Stubsjøen, M. (2013). *Analytical Numerical Modeling of Paraffin Wax in Pipelines*. MSc. Norwegian University of Science and Technology.

Tecosystems. (2015). *The RedMonk Programming Language Rankings*. [online] Available at: <http://redmonk.com/sograde/2015/01/14/language-rankings-1-15/> [Accessed 13 Oct. 2015].

8 APPENDICES

8.1 Other Significant Factors for Wax Deposition

8.1.1 Wax Thermodynamics

Performing diffusion calculations requires the driving force towards equilibrium to be known. The calculated deposition rate will be inaccurate if this is not accurately known (Coutinho et al., 2006). The Wax Precipitation Curve has been shown to be of great importance among the input parameters.

Thermodynamic equilibrium between the oil and the wax is therefore an important factor for predicting wax deposition (Coutinho et al., 2006). The Wax Precipitation Curves can be obtained by laboratory analysis of the specific crude oil or calculated from thermodynamic models and equations of state.

Monger-McClure suggested that the uncertainty for good modern measurements of WAT may be around 3°C with older measurements being considerably higher (Coutinho et al., 2006).

A lot of research has been done in order to investigate the thermodynamics of paraffin wax deposition in order to create an accurate model of this process (Aiyejina et al., 2011). These thermodynamic models were developed to determine the WAT and equilibria between wax and oil (Azevedo and Teixeira, 2003).

Some models currently used in the oil industry, use parameters determined empirically from available oil data and this data is often not very precise (Coutinho et al., 2006). Thermodynamic models often incorrectly estimate the WAT and amount of wax precipitation (Kamari et al., 2013). These models can predict wax precipitation fairly well but the main question is how much deposition will occur (Azevedo and Teixeira, 2003).

8.1.2 Precipitation Kinetics

Pasco (2005) found three regimes for slow cooling crystallisation: the nucleation lag period, the supersaturated growth period and the equilibrium growth period (Aiyejina et al., 2011). Pasco substantiated the lack of transport limitations in the crystallisation rate and showed that nucleation acts as the main kinetic limitation at low cooling rates (Aiyejina et al., 2011). Nucleation sites such as solid impurities also play a role in the rate of crystal formation (Azevedo and Teixeira, 2003). Compared to the modelling of wax thermodynamics, the crystallisation kinetics of wax deposition has not been as extensively researched (Aiyejina et al., 2011).

8.1.3 Multiphase Flow and Pour Point

With multiphase flow, standard heat and mass transfer correlations are no longer reliable (Huang et al., 2011c). As illustrated in Figure 8.1 (Aiyejina et al., 2011), the thickness and hardness of the deposit in a pipe with gas-liquid flow varies around the circumference (Aiyejina et al., 2011). As shown in Figure 8.2 (Huang et al., 2011c) there exist different flow patterns depending on the flow rates and oil-water ratios. Huang et al. (2011c) explain that for stratified flow of oil and water under non-isothermal conditions, the change in position of the oil-water boundary must be taken into account, as shown in Figure 8.3 (Huang et al., 2011c).

Wax deposition in multiphase oil-water flow is considerably reduced due to water reducing heat loss of the oil and occupying space against the pipe wall (Huang et al., 2011c). There is however a significant risk to flow assurance in multiphase offshore production posed by the presence of water above a certain level, as emulsified water has an effect on crude oil gelation (Aiyejina et al., 2011). Precipitated wax crystals, through making the oil behave as a gel, may also cause major issues when re-starting production after a shut-down (Azevedo and Teixeira, 2003).

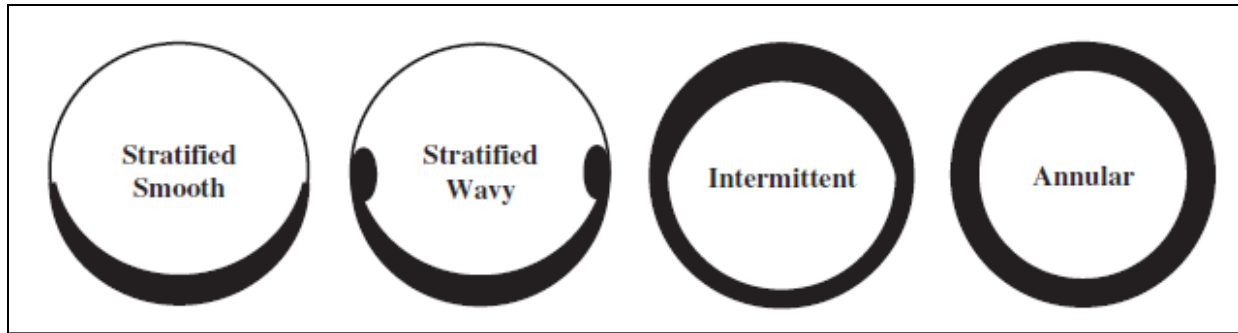


Figure 8.1: Impact of flow pattern on wax deposit (Aiyejina et al., 2011)

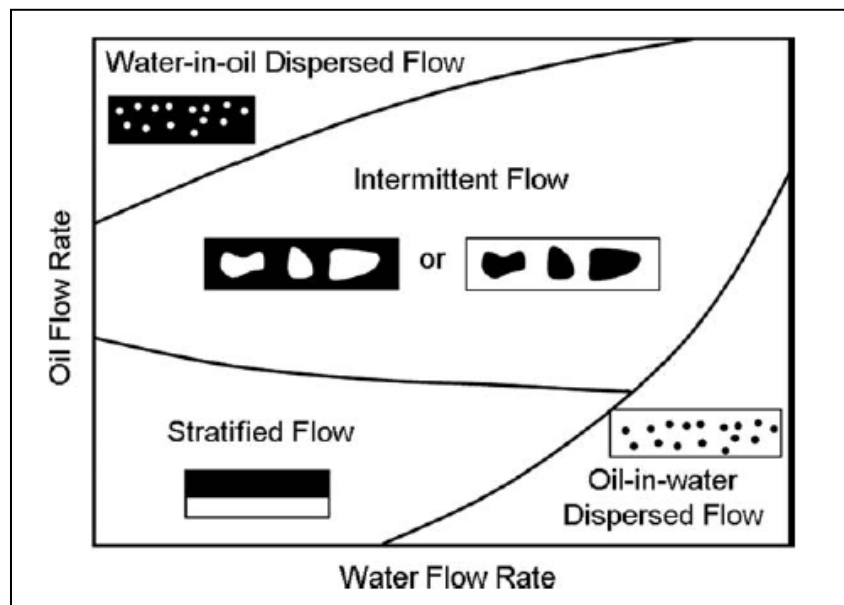


Figure 8.2: A flow map of two phase oil/water flow (Huang et al., 2011c)

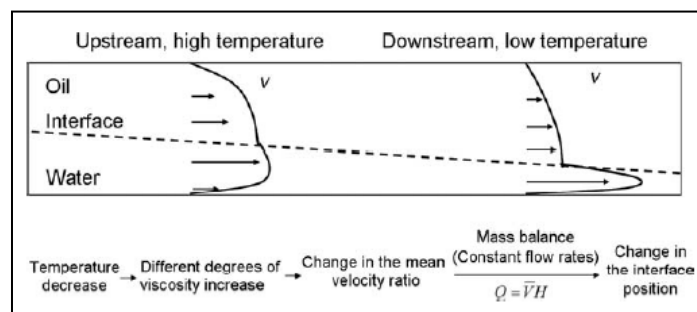


Figure 8.3: Non-isothermal velocity profile of oil/water flow (Huang et al., 2011c)

8.1.4 Effects of the Oil Flow Rate on Wax Deposition

A number of experiments have found decreased deposition for a higher oil flow rate and this is often attributed to the effect of shear stripping (Lu et al., 2012). Lu et al. (2012) through using the Michigan Wax Predictor (MWP) found three competing heat and mass transfer based causes for a change in deposition, resulting from a change in flow rate. These are the change in boundary layer thickness and the effects that the change in interfacial diffusivity and interfacial concentration have on mass transfer (Lu et al., 2012). These three effects compete throughout the deposition process of wax with their relative importance changing with time (Lu et al., 2012).

8.2 Mass-flux-based Implementation

8.2.1 Typical Input Data

The typical input variables which are input using slider bars or input boxes are listed below

Table 8.1: Variable Input Values - Mass-flux-based Implementation

Symbol	Value	Unit	Description
D_p	0.5	m	Inner pipe diameter
T_{b_in}	50+273.15	°K	Bulk inlet temperature
U	2.0	m/s	Mean flow velocity
L	10000	m	Pipe length
n_z	200	No	Number of nodes in the axial direction
n_r	200	No	Number of nodes in the radial direction
n_{tm}	7	No	Number of nodes in time
Time	7	days	Time
T_{sea}	5+273.15	°K	Outer wall temperature
wall_thickness	0.012	m	Wall thickness
Insul_thickness	0.1	m	Insulation thickness

The variable properties which can be input through input boxes.

Table 8.2: Variable Property Values - Mass-flux-based Implementation

Symbol	Value	Unit	Description
ρ_f	750	kg/m ³	Fluid density
ρ_{wax}	750	kg/m ³	Solid wax density
μ_f	0.0005	Pa.s	Dynamic viscosity
λ_f	0.1	W/(m.K)	Fluid thermal conductivity
C_{p_f}	2300	J/(kg.K)	Fluid specific heat capacity
k_r	1.4	W/(m.K)	Precipitation rate constant
k_{pipe}	20	W/(m.K)	Thermal conductivity of pipe
h_i	922.0	W/(m ² .K)	Inner heat transfer coefficient
λ_{wax}	0.25	W/(m.K)	Wax solid thermal conductivity
ε	0.00001	kg/(m ³ .s)	Proportionality coefficient for shear-removal
n	2	No	Diffusivity exponent
k_{sr}	$5 \cdot 10^{-8}$	1/(Pa.s)	Shear removal term
κ_t	$1.42 \cdot 10^{-7}$	1/s	Long term shear removal constant
h_o	922	W/(m ² .K)	Outer heat transfer coefficient
k_{insul}	20	W/(m.K)	Thermal conductivity of insulation

Table 8.3: Fixed Property Values - Mass-flux-based Implementation

Symbol	Value	Unit	Description
n_k	2	No	Number of steps in radial direction of deposit
v_w	430	cm ³ /mol	Molar volume of wax
Pr_T	0.87	No	Turbulent Prandlt number
χ	0.4	No	Von Karman constant
τ_w	7.73	Pa	Wall shear stress
δ_o	$20 \cdot 10^{-6}$	m	Initial inlet deposit thickness

8.2.2 Parameter Equations

$$v_f = \frac{\mu_f}{\rho_f} \quad (8.1)$$

$$a_f = \frac{\lambda_f}{(\rho_f C_{pf})} \quad (8.2)$$

$$Pr = \frac{v_f}{(a_f)} \quad (8.3)$$

$$Re = \frac{UD_p}{(v_f)} \quad (8.4)$$

$$u^+ = \left(\frac{\tau_w}{(\rho_f)} \right)^{0.5} \quad (8.5)$$

$$y^+ = \frac{y^* u^+}{v_f} \quad (8.6)$$

$$\varepsilon_t^+ = \frac{\varepsilon_t}{v_f} \quad (8.7)$$

$$\Theta = \left(\frac{(\rho_{wax} - c_m)}{q_d} \right) \delta \quad (8.8)$$

$$u = \left(\frac{\rho_{wax}}{(\rho_{wax} - c_m)} \right) \quad (8.9)$$

8.2.3 Interface Temperatures Equations

$$r_{wo} = R + \Delta t_{wall} \quad (8.10)$$

$$r_{dep} = R - \delta \quad (8.11)$$

$$A_{wo} = 2\pi r_{wo} \Delta z \quad (8.12)$$

$$A_{wi} = 2\pi R \Delta z \quad (8.13)$$

$$A_{dep} = 2\pi r_{dep} \Delta z \quad (8.14)$$

$$A_{logmean_pipe} = \left(\frac{A_{wo} - A_{wi}}{\ln\left(\frac{A_{wo}}{A_{wi}}\right)} \right) \quad (8.12)$$

$$A_{logmean_dep} = \left(\frac{A_{wi} - A_{dep}}{\ln\left(\frac{A_{wi}}{A_{dep}}\right)} \right) \quad (8.16)$$

$$R_{out} = \left(\frac{1}{h_o A_{wo}} \right) \quad (8.17)$$

$$R_{in} = \left(\frac{1}{h_i A_{dep}} \right) \quad (8.18)$$

$$R_{\text{pipe}} = \left(\frac{r_{\text{wo}} - r_{\text{wi}}}{k_{\text{pipe}} A_{\log\text{mean_pipe}}} \right) \quad (8.19)$$

$$R_{\text{dep}} = \left(\frac{r_{\text{wi}} - r_{\text{dep}}}{k_{\text{dep}} A_{\log\text{mean_dep}}} \right) \quad (8.20)$$

$$q = \frac{T_{\text{B}} - T_{\text{sea}}}{R_{\text{pipe}} + R_{\text{dep}} + R_{\text{in}} + R_{\text{out}}} \quad (8.21)$$

$$T_{\text{wo}} = T_{\text{sea}} + q R_{\text{out}} \quad (8.22)$$

$$T_{\text{wi}} = T_{\text{wo}} + q R_{\text{pipe}} \quad (8.23)$$

$$T_{\text{dep}} = T_{\text{wi}} + q R_{\text{dep}} \quad (8.24)$$

8.2.4 Mass-flux-based Implementation's Code



UNIVERSITY OF THE WITWATERSRAND
JOHANNESBURG

```

import numpy as np
import time
from ipywidgets import interact, fixed
import matplotlib
matplotlib.rc('font', size = 25, family = 'Times New Roman')
import matplotlib.pyplot as plt
%matplotlib inline

mode = "w"
mode = str(input('Widget [w] mode or input box mode [b] eg. [w/b]:'))

mode2 = "d"
mode2 = str(input('Input default chemical properties [d] or input properties manually [m] eg. [d/m]:'))

mode3 = "d"
mode3 = str(input('Input default solubility example [d] or solubility data [m] eg. [d/m]:'))

shear = "Shear"
t_interest = 6

```

```

def solubilty(mode3):
    if mode3 == "d":
        dc_dt = 0.1                # [kg/(m^3.K)] Wax solubility constant
        Cwax = 40.0                # [kg/m^3] Inlet concentration
        Twat = 30.0+ 273.15        # [K] Wax Appearance Temperature
        polynomial = 0
        coefficient = 0
    else:
        points = np.zeros(4)
        c1 = float(input('Input data point conc 1, [kg/m^3] eg. 37:'))
        t1 = float(input('Input data point temp 1, [K] eg. 273.15:'))
        c2 = float(input('Input data point conc 2, [kg/m^3] eg. 38:'))
        t2 = float(input('Input data point temp 2, [K] eg. 283.15:'))
        c3 = float(input('Input data point conc 3, [kg/m^3] eg. 39:'))
        t3 = float(input('Input data point temp 3, [K] eg. 293.15:'))
        c4 = float(input('Input data point conc 4, [kg/m^3] eg. 40:'))
        t4 = float(input('Input data point temp 4, [K] eg. 303.15:'))

        Cwax = float(input('Concentration of Wax at WAT: [kg/m^3] eg. 40:'))
        Twat = float(input('Wax appearance temperature: [oC] eg. 303.15:'))

        cvec = [c1,c2,c3,c4]
        tvec = [t1,t2,t3,t4]
        coefficient = np.polyfit(tvec,cvec, 2)
        polynomial = np.polyid(coefficient)
        print(polynomial)
        dc_dt = 0.1

    return dc_dt, Cwax, Twat, polynomial, coefficient

```

```

def parameters(mode2):
    if mode2 == 'm':
        pf = float(input('Fluid density, [kg/m^3] eg. 750:'))
        pwax = float(input('Wax solid density, [kg/m^3] eg. 750:'))
        μf = float(input('Dynamic Viscosity, [Pa.s] eg. 0.0005:'))
        λf = float(input('Fluid thermal conductivity, [W/(m.K)] eg. 0.1:'))
        Cpf = float(input('Fluid Specific heat capacity, [J/(kg.K)] eg. 2300:'))
        kr = float(input('Precipitation rate constant, [1/(s)] eg. 1.4:'))
        k_pipe = float(input('Thermal conductivity of pipe, [W/(m.K)] eg. 20:'))
        hi = float(input('Inner heat transfer coefficient of pipe, [W/(m^2.K)] eg. 922:'))
        λwax = float(input('Wax solid thermal conductivity, [W/(m.K)] eg. 0.25:'))
        ε = float(input('Porportionality coefficient for shear-removal term, [kg/(m^3.s)] eg. 0.00001:'))
        n = float(input('Diffusivity exponents, [No] eg. 2:'))
        ksr = float(input('Shear removal factor, [1/(Pa.s)] eg. 0.00000005:'))
        kt = float(input('Second stage shear removal constant from experiment, [(1/s) eg. 0.000000142:'))
        ho = float(input('Outer heat transfer coefficient of pipe, [W/(m^2.K)] eg. 922:'))
        k_insul = float(input('Thermal conductivity of insulation, [W/(m.K)] eg. 20:'))

    else:
        pf = 750.0                # [kg/m^3] FLuid density
        pwax = 750.0              # [kg/m^3] Wax solid density
        μf = 0.5*10**-3           # [Pa.s] Dynamic Viscosity
        λf = 0.1                  # [W/(m.K)] Fluid thermal conductivity
        Cpf = 2300.0              # [J/(kg.K)] Fluid specific heat capacity
        kr = 1.4                  # [(s^-1)] Precipitation rate constant
        k_pipe = 20               # [W/(m.K)] Thermal conductivity of pipe
        hi = 922.0                # [W/(m^2.K)] Inner heat transfer coefficient of pipe
        λwax = 0.25               # [W/m.K] Wax solid thermal conductivity
        ε = 0.00001              # [kg/(m^3.s)] Porportionality coefficient for shear-removal term
        n = 2.0                   # [No] Diffusivity exponent
        ksr = 5.0*10**-8          # [1/(Pa.s)] Shear removal factor
        kt = 1.42*10**-7          # [1/(s^-1)] Long term shear removal constant from experiment
        ho = 922                  # [W/(m^2.K)] Outer heat transfer coefficient of pipe
        k_insul = 20              # [W/(m.K)] Thermal conductivity of insulation

    return pf,pwax,μf,λf,Cpf,kr,k_pipe,hi,ε,n,ksr,λwax,kt,ho,k_insul

```



```

def select(mode):
    if mode == 'b':
        Dp = float(input('Input inner pipe diameter: [m] eg. 0.5:'))
        Tb_in = 273.15 + float(input('Input bulk inlet temperature: [oC] eg. 40:'))
        U = float(input('Input mean flow velocity: [m/s] eg. 2.0:'))
        L = float(input('Input pipe length: [m] eg. 10000:'))
        Tsea = 273.15 + float(input('Input sea temperature: [oC] eg. 5:'))
        nz = int(input('Input number of nodes in the axial direction [No] eg. 200:'))
        nr = int(input('Input number of nodes in the radial direction [No] eg. 200:'))
        ntm = int(input('Input number of nodes in time eg. [No] 7:'))
        Time = 24*3600*float(input('Time step : [days] eg. 7:'))
        wall_thickness = float(input('Wall thickness, [m] eg. 0.012:'))
        insul_thickness = float(input('Insulation thickness, [m] eg. 0.1:'))

        start_time = time.time() # Timing code run time
        model(Dp,Tb_in,U,L,Tsea,Twat,Cwax,nz,nr,ntm,Time,pf,pwax,mf,lamf,Cpf,kr,dc_dt,k_pipe,hi,wall_thickness,e,n,ksr,
            lamax,kt,shear,t_interest,ho,k_insul,insul_thickness,polynomial,mode3,coefficient)
        print("%f seconds" % (time.time() - start_time))

    else:
        Dp = 0.5 # [m] Inner pipe diameter
        Tb_in = 40.0+273.15 #41 # [K] Bulk inlet temperature
        U = 2.0 # [m/s] Mean flow velocity
        L = 10000.0 # [m] Pipe Length
        Tsea = 5.0 + 273.15 #278.15 # [K] Sea/Outer wall temperature
        nz = 200 # [No] Number of nodes in axial direction
        nr = 200 # [No] Number of nodes in radial direction
        ntm = 7 # [No] Number of nodes in time
        Time = 7*24*3600.0 # [s] Time
        wall_thickness = 0.012 # [m] Wall thickness
        insul_thickness = 0.1 # [m] Insulation thickness

```

```

        interact(model,shear=("Shear", "No Shear"),Dp=(0.3000,1,0.1),Tb_in=(303.15,330.15,1),U=(0.1,2,0.2),L=(5000,10000,1000),
            Tsea=(273.15,280.15,2),Time=(3600*24,3600*7*24,3600*24),wall_thickness=(0.005,0.035,0.005),
            insul_thickness=(0.005,0.2,0.005),t_interest=(0,ntm),Twat=fixed(Twat),Cwax=fixed(Cwax),nz=fixed(nz),
            nr=fixed(nr),ntm=fixed(ntm-1),pf=fixed(pf),pwax=fixed(pwax),mf=fixed(mf),lamf=fixed(lamf),Cpf=fixed(Cpf),
            kr=fixed(kr),dc_dt=fixed(dc_dt),k_pipe=fixed(k_pipe),hi=fixed(hi),e=fixed(e),n=fixed(n),ksr=fixed(ksr),
            lamax=fixed(lamax),kt=fixed(kt),ho=fixed(ho),k_insul=fixed(k_insul),polynomial=fixed(polynomial),
            mode3=fixed(mode3),coefficient=fixed(coefficient))

def model(Dp,Tb_in,U,L,Tsea,Twat,Cwax,nz,nr,ntm,Time,pf,pwax,mf,lamf,Cpf,kr,dc_dt,k_pipe,hi,wall_thickness,e,n,ksr,lamax,kt,shear,
    t_interest,ho,k_insul,insul_thickness,polynomial,mode3,coefficient):

    #Fixed Parameters
    Pr_T = 0.87 # [No] Turbulent Prandlt
    chi = 0.4 # [No] Von Karman constant
    tau_w = 7.73 # [N/m^2]
    vw = 430 # [cm^3/mol] Molar volume of wax
    delta_0 = 20*10**-6 # [m] Initial inlet deposit thickness
    nk = 2 # [No] Number of steps in radial direction in deposit

    # Parameter Calculations
    vf = mf/pf # [m^2/s] Kinematic viscosity
    af = lamf/(pf*Cpf) # [m^2/s] Fluid thermal diffusivity
    Pr = vf/af # [No] Prandlt number
    Re = U*Dp/vf # [No] Reynolds number
    u_star = (tau_w/pf)**0.5 # [m/s] Friction velocity
    y_end = int(Dp*u_star/(vf*2)) # [No] - int for rounding if nessisary

    # Initialising vectors
    z = np.linspace(0,L,nz)
    y_plus = np.arange(nr)

```

```

Tb = np.zeros([nz,ntm])
Tbstub = np.zeros([nz,ntm])
qt = np.zeros([nz,ntm])
ΔT = np.zeros([nz,ntm])
T = np.zeros([nr,nz,ntm])
et_plus = np.zeros([nr,nz,ntm])
Cbulk = np.zeros([nz,ntm])
Cs = np.zeros([nr,nz,ntm])
D = np.zeros([nr,nz,ntm])
Scm = np.zeros([nr,nz,ntm])
D_plus = np.zeros([nr,nz,ntm])
eM_plus = np.zeros([nr,nz,ntm])
Dtot = np.zeros([nr,nz,ntm])
C = np.zeros([nr,nz,ntm])
C_sol = np.zeros([nr,nz,ntm])
dc_dy_plus_0 = np.zeros([nz,ntm])
Csbulk = np.zeros([nz,ntm])
Cwall = np.zeros([nz,ntm])
Cswall = np.zeros([nz,ntm])

Cm = np.zeros([nz,ntm])
θ = np.zeros([nz,ntm])
qsr = np.zeros([nz,ntm])
qd = np.zeros([nz,ntm])
φ = np.zeros([nk,nz,ntm])
λd = np.zeros([nk,nz,ntm])
λd_top = np.zeros([nz,ntm])
λd_bot = np.zeros([nz,ntm])
λd_avg = np.zeros([nz,ntm])
dln_λd_dφ_top = np.zeros([nz,ntm])
dln_λd_dφ_bot = np.zeros([nz,ntm])
dln_δ_dt = np.zeros([nz,ntm])
δ = np.zeros([nz,ntm])
κ = np.zeros([nz,ntm])
u = np.zeros([nz,ntm])

```

```

where = np.zeros([nz,ntm])
R_lm_pipe = np.zeros([nz,ntm])
R_lm_dep = np.zeros([nz,ntm])
r_dep = np.zeros([nz,ntm])
A_dep = np.zeros([nz,ntm])
A_lm_dep = np.zeros([nz,ntm])
k_dep = np.zeros([nz,ntm])
R_pipe = np.zeros([nz,ntm])
R_dep = np.zeros([nz,ntm])
R_in = np.zeros([nz,ntm])
q = np.zeros([nz,ntm])
Twi = np.zeros([nz,ntm])
Tdep = np.zeros([nz,ntm])
Tio = np.zeros([nz,ntm])
Two = np.zeros([nz,ntm])
Tsf = np.zeros([nz,ntm])
R_out = np.zeros(nz)
av = np.zeros(nz)
mx = np.zeros(nz)

# Calculating step size
Δz = L/nz # [m] axial step size
Δr = Dp/(nr*2) # [m] radial step size
ΔTime = Time/ntm
Δt = ΔTime # [s]

# Initial Conditions
Tb[0,:] = Tb_in
φ[1,:,:] = 0.89 # [No] Porosity at [0,0]
φ[0,:,:] = 0.91 # [No] Porosity at [0,1] - 0.9-0.95
δ[:,0] = δ0
λd_avg[:] = 0.25 # [W/(m.K)] Initial inlet deposit thermal conductivity

y = y_plus*vf/u_star # [m] Same number of steps as nr
Δy_plus = Δr*u_star/vf # [No] ALL increments assumed equal
Dp_plus = Dp*u_star/vf # [No] Dimensionless Pipe Diameter

```

```

matrix_end = 30
matrix_A = np.zeros([matrix_end+1,matrix_end+1])
matrix_C = np.zeros([matrix_end+1,1])
matrix_D = np.zeros([matrix_end+1,1])

# Layer Temperature calculations
R = Dp/2 # [m] Radius

r_wo = R + wall_thickness # [m] Outer wall Radius
r_io = r_wo + insul_thickness # [m] Outer wall Radius
r_wi = R # [m] Inner wall Radius

A_io = 2*np.pi*r_io*Δz
A_wo = 2*np.pi*r_wo*Δz # [m^2] Outside wall segment area
A_wi = 2*np.pi*r_wi*Δz # [m^2] Inside wall segment area
A_lm_pipe = (A_wo - A_wi)/(np.log(A_wo/A_wi)) # Log mean pipe segment area

if insul_thickness < 0:
    R_insul = 0
else:
    A_lm_insul = (A_io - A_wo)/(np.log(A_io/A_wo)) # [m^2] Log mean pipe segment area
    R_insul = (r_io - r_wo)/(k_insul*A_lm_insul) # [m^2.K.s/J]

R_out = 1/(ho*A_io) # [K.s/J] Outer fluid thermal resistance
R_pipe = (r_wo - r_wi)/(k_pipe*A_lm_pipe) # [K.s/J] Pipe thermal resistance

ΔT_200 = (5*Pr+5*np.log(1+5*Pr)+2.5*np.log(200/300)) # [No]

```

```

for tm in range(n tm):

    # Calculating Flux to Deposit Surface
    for l in range(nz):
        k_dep[l,tm] = λd_avg[l,tm] # [J/(m.s.K)]

        r_dep[l,tm] = R - δ[l,tm] # [m] Deposit radius - changes

        A_dep[l,tm] = 2*np.pi*r_dep[l,tm]*Δz # [m^2] # Deposit area - changes

        if δ[l,tm] < 0: # Singularity for zero deposit
            R_dep[l,tm] = 0
            R_in[l,tm] = 1/(hi*A_dep[l,tm]) # Inner fluid thermal resistance
        elif δ[l,tm] >= R:
            δ[l,tm] = R
        else:
            A_lm_dep[l,tm] = (A_wi - A_dep[l,tm])/(np.log(A_wi/A_dep[l,tm])) # Log mean deposit area

            R_dep[l,tm] = (r_wi - r_dep[l,tm])/(k_dep[l,tm]*A_lm_dep[l,tm]) # Deposit thermal resistance
            R_in[l,tm] = 1/(hi*A_dep[l,tm]) # Inner fluid thermal resistance

        q[l,tm] = (Tb[l,tm] - Tsea)/(R_pipe+R_dep[l,tm]+R_in[l,tm]+R_out+R_insul) # [J/s] Segment heat Loss

        Tio[l,tm] = Tsea + q[l,tm]*R_out # [K]

        Two[l,tm] = Tio[l,tm] + q[l,tm]*R_insul

        Twi[l,tm] = Two[l,tm] + q[l,tm]*R_pipe

        Tdep[l,tm] = Twi[l,tm] + q[l,tm]*R_dep[l,tm]

        #Tsff[l,tm]= Tdep[l,tm] + q[l,tm]*R_in[l,tm]
        Tsff[l,tm]= Tdep[l,tm]

        ΔT[l,tm] = Tb[l,tm] - Tsff[l,tm] + ((qt[l,tm]/A_dep[l,tm])/((χ/Pr_T)*pf*Cpf*u_star))*np.log(400/Dp_plus) # [K]

```

```

if l < nz - 1: # Extra L in the following equation
    Tb[l+1,tm] = Tb[l,tm] - (Δz*(0.092*(Re**-0.2)*(Pr**-0.67)*(Tb[l,tm] - Tsf[l,tm])))/Dp

#Calculating Temperature
for r in range(200):

    #y_plus = r
    if r <= 5:
        T[r,l,tm] = ((r*Pr)/ΔT_200)*ΔT[l,tm] + Tsf[l,tm]

    elif (5.0 < r <= 30.0):
        #np.log is the natural log
        T[r,l,tm] = ((5*Pr + 5*np.log(0.2*r*Pr + 1 - Pr))/ΔT_200)*ΔT[l,tm] + Tsf[l,tm]

    elif 30 < r :
        T[r,l,tm] = ((5*Pr + 5*np.log(1 + 5*Pr)+2.5*np.log(r/30))/ΔT_200)*ΔT[l,tm] + Tsf[l,tm]

for r in range(30):
    # Calculating
    if r <= 5 :
        et_plus[r,l,tm] = (r/14.5)**3

    elif r > 5:
        et_plus[r,l,tm] = (r/5) - 0.959

    if mode3 == "d":
        if T[r,l,tm] < Twat:
            C_sol[r,l,tm] = Cwax - dc_dt*(Twat - T[r,l,tm]) # [kg/m^3]

        elif T[r,l,tm] >= Twat:
            C_sol[r,l,tm] = Cwax

    else:
        if T[r,l,tm] < Twat:
            C_sol[r,l,tm] = polynomial(T[r,l,tm])

```

```

        elif T[r,l,tm] >= Twat:
            C_sol[r,l,tm] = Cwax

#Calculating Solubility
if mode3 == "d":
    Cs[:, :,tm] = Cwax - dc_dt*(Twat - T[:, :,tm]) # Solubility Concentration Assuming dc/dt = constant
else:
    Cs[:, :,tm] = polynomial(T[:, :,tm])

#Calculating Molecular Diffusivity
D[:, :,tm] = 13.3*(10**-12)*(T[:, :,tm]**1.47)*((1000*μf)**(10.2/vw-0.791))*(vw**-0.71) # [m^2/s] Wax molecular diffusivity

D_plus[:, :,tm] = D[:, :,tm]/vf # [No] Making dimensionless

Scm[:, :,tm] = vf/D[:, :,tm] # [No] Schmidt number for turbulent mass transfer

eM_plus[:, :,tm] = et_plus[:, :,tm]/Scm[:, :,tm]

Dtot[:, :,tm] = (D_plus[:, :,tm] + eM_plus[:, :,tm])

# Calculating Solubility
if mode3 == "d":
    Csbulk[:,tm] = Cwax - dc_dt*(Twat - Tb[:,tm]) # For plotting
    Cswall[:,tm] = Cwax - dc_dt*(Twat - T[0[:,tm])
else:
    Csbulk[:,tm] = polynomial(Tb[:,tm])
    Cswall[:,tm] = polynomial(T[0[:,tm])

for l in range(nz):
    for r in range(matrix_end+1):

```

```

# Boundary conditions
if r == 0:
    matrix_A[0,0] = 1
    if T[r,l,tm] >= Twat: # if T[30,L] >= Twat:
        Cwall[l,tm] = Cwax

    elif T[r,l,tm] < Twat:
        Cwall[l,tm] = Cs[r,l,tm]

    matrix_D[0] = Cwall[l,tm]

if 0 < r < 30: # if A*C[r] + B*C[r+1] + C*C[r-1] = D - Derived from eqn 16 in Eskin et al. (2014) paper

    matrix_A[r,r-1] = 1
    matrix_A[r,r] = (-2*Δy_plus/r-(kr*vf*Δy_plus**2)/(Dtot[r,l,tm]*u_star**2)) # C
    matrix_A[r,r+1] = (1 + Δy_plus/r) # A
    matrix_D[r] = -(C_sol[r,l,tm]*(kr*vf*Δy_plus**2)/(Dtot[r,l,tm]*u_star**2)) # B
    matrix_D[r] = -(C_sol[r,l,tm]*(kr*vf*Δy_plus**2)/(Dtot[r,l,tm]*u_star**2)) # D

if r == matrix_end:

    matrix_A[r,r] = 1

    if T[r,l,tm] >= Twat: # if T[30,L] >= Twat:
        Cbulk[l,tm] = Cwax

    elif T[r,l,tm] < Twat:
        Cbulk[l,tm] = Cs[r,l,tm]

    matrix_D[r] = Cbulk[l,tm]

    matrix_C = np.linalg.solve(matrix_A,matrix_D)

    C[:,l,tm][0:matrix_end+1] = (matrix_C[:,0])

if r > matrix_end:
    C[r,l,tm] = Cbulk[l,tm]

```

```

dc_dy_plus_0[:,tm] = (C[1,:,tm]-Cswall[:,tm])/(Δy_plus) # [kg/(m^3)] Concentration gradient near wall

qt[:,tm] = D_plus[0,:,tm]*dc_dy_plus_0[:,tm]*u_star # [kg/(m^2.s)] Flux to deposit

for l in range(nz):
    if qt[l,tm] <= 0:
        qt[l,tm] = 1*10**-10

Cm[:,tm] = (C[0,:,tm]) # Mean wax content in a hydrocarbon fluid
u[:,tm] = (pwax/(pwax-Cm[:,tm])) # Dimensionless complex

if shear == "No Shear":
    qsr[:,tm] = 0

    qd[:,tm] = qt[:,tm]

else:
    qsr[:,tm] = ε*δ[:,tm] # Shear-removal rate

    qd[:,tm] = qt[:,tm] - qsr[:,tm]

θ[:,tm] = ((pwax-Cm[:,tm])/qd[:,tm])*δ[:,tm] # [s] Characteristic time

#Maxwell equation - Eskin
λd_top[:,tm] = λf*((2*λwax+λf+(λwax-λf)*(1-φ[1,:,tm]))/(λwax+λf-2*(λwax-λf)*(1-φ[1,:,tm]))) # [W/(m.K)]
λd_bot[:,tm] = λf*((2*λwax+λf+(λwax-λf)*(1-φ[0,:,tm]))/(λwax+λf-2*(λwax-λf)*(1-φ[0,:,tm]))) # [W/(m.K)]

#Differentiating Maxwell equation - using quotient and chain rule
dln_λd_dφ_top[:,tm] = (1/λd_top[:,tm])*((λf*(λf-λwax))*(λwax+λf-2*(λwax-λf)*(1-φ[1,:,tm]))-(λf*(2*λwax+λf+(λwax-λf)*(2*(λwax-λf)))))/((λwax+λf-2*(λwax-λf)*(1-φ[1,:,tm]))**2) # [No]
dln_λd_dφ_bot[:,tm] = (1/λd_bot[:,tm])*((λf*(λf-λwax))*(λwax+λf-2*(λwax-λf)*(1-φ[0,:,tm]))-(λf*(2*λwax+λf+(λwax-λf)*(2*(λwax-λf)))))/((λwax+λf-2*(λwax-λf)*(1-φ[0,:,tm]))**2) # [No]

λd_avg[:,tm] = (λd_top[:,tm]+λd_bot[:,tm])/2

```

```

    λd_avg[:,tm] = (λd_top[:,tm]+λd_bot[:,tm])/2

    if shear == "No Shear":
        κ[:,tm] = 0
    else:
        κ[:,tm] = ((qt[:,tm]/qd[:,tm])-(φ[1,:,tm]**n)*(λf/λd_top[:,tm])+(φ[0,:,tm]**n)*(λf/λd_bot[:,tm])-n*(φ[1,:,tm]**(n-1))*((λf/λd_top[:,tm]))*(1-(φ[1,:,tm]/n)*(dln_λd_dφ_top[:,tm])*(u[:,tm]-φ[1,:,tm])))/θ[:,tm] # [1/s]

        for l in range(nz):
            if κ[l,tm] > κt:
                κ[l,tm] = κt

    dln_δ_dt[:,tm] = (1/θ[:,tm])*((qt[:,tm]/qd[:,tm])-(φ[1,:,tm]**n)*(λf/λd_top[:,tm])+(φ[0,:,tm]**n)*(λf/λd_bot[:,tm])-(κ[:,tm]*θ[:,tm])/(u[:,tm]-φ[1,:,tm])) # [1/s]

    if tm < ntm-1:
        λd_avg[:,tm+1] = λd_avg[:,tm]

        qt[:,tm+1]=qt[:,tm]
        # Porosity evolution @ z = 1
        φ[1,:,tm+1] = φ[1,:,tm] - Δt*((φ[1,:,tm]-φ[0,:,tm])*(1/θ[:,tm])*(n*(φ[1,:,tm]**(n-1))*(λf/λd_top[:,tm])*(1-(φ[1,:,tm]/n)*(dln_λd_dφ_top[:,tm]))-(θ[1,tm]*dln_δ_dt[1,tm])))
        # Porosity evolution @ z = 0
        φ[0,:,tm+1] = φ[0,:,tm] - Δt*((φ[1,:,tm]-φ[0,:,tm])*(1/θ[:,tm])*(n*(φ[0,:,tm]**(n-1))*(λf/λd_bot[:,tm])*(1-(φ[0,:,tm]/n)*(dln_λd_dφ_bot[:,tm])))))

        δ[:,tm+1] = δ[:,tm]*np.e**(Δt*dln_δ_dt[:,tm])

    for r in range(30,nr):

        C[r, :, :] = Cbulk[:, :]
        if r > 200:
            T[r, :, :] = Tb[:, :]
            Cs[r, :, :] = Csbulk[:, :]

```

```

for r in range(1,nk):
    φ[r, :, :] = (φ[1, :, :]-φ[0, :, :])*r/nz + φ[0, :, :]

cross_section1 = 1
cross_section2 = 25
cross_section3 = 150

zoom3 = 15
zoom = 25
zoom2 = 100

ld = 2
C_sat = C-C_sol
κt = np.ones(nz)*κt
av[:] = np.mean(δ[:,t_interest])
mx[:] = np.amax(δ[:,t_interest])
tshow = np.linspace(277.15,60+273.15)
cwaxshow = Cwax*np.ones(len(tshow))

if mode3 == "d":
    cshow = Cwax - dc_dt*(Twat - tshow)
else:
    cshow = polynomial(tshow)

print(polynomial)
fontsize = 15

#1
plt.plot(z,Tb[:,t_interest]-273.15,label = "Bulk",linewidth=ld)
plt.plot(z,np.ones(nz)*Twat-273.15,label = "WAT",linewidth=ld)
plt.title('Bulk Temperature along Pipeline')
plt.ylabel('Bulk Temperature, [oC]')
plt.xlabel('Pipe Length, [m] - Inlet on Left')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

```

```

#2
plt.plot(z,Tb[:,ntm-1]-273.15,label = "Tbulk",linewidth=ld)
plt.plot(z,Twi[:,ntm-1]-273.15,label = "Twall inside",linewidth=ld)
plt.plot(z,Tdep[:,ntm-1]-273.15,label = "Tdeposit",linewidth=ld)
plt.plot(z,Two[:,ntm-1]-273.15,label = "Twall outside",linewidth=ld)
plt.plot(z,Tio[:,ntm-1]-273.15,label = "Tinsul outside",linewidth=ld)
plt.title('Interface Temperatures along Pipeline')
plt.ylabel('Interface Temperatures, [oC]')
plt.xlabel('Pipe Length, [m] - Inlet on Left')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#3
plt.plot(y[0:zoom],T[:,cross_section2,t_interest][0:zoom]-273.15,linewidth=ld)
plt.title('Temperature Cross-section at node %s' %cross_section2)
plt.ylabel('Temperature, [oC]')
plt.xlabel('Pipe Radius, [m] - Wall on Left')
plt.grid()
plt.show()

#4
plt.imshow(T[:,:,t_interest]-273.15, aspect = 'auto')
plt.colorbar()
plt.title('Temperature Profile')
plt.ylabel('Radial Position, [steps]')
plt.xlabel('Pipe Length, [steps] - Inlet on Left, Wall on Top')
plt.grid()
plt.show()

```

```

#5
plt.imshow(T[:,:,t_interest][0:zoom3,0:zoom]-273.15, aspect = 'auto')
plt.colorbar()
plt.title('Temperture Zoomed in near Inlet and Wall')
plt.xlabel('Pipe Length, [steps] - Inlet on Left, Wall on Top')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#6
plt.imshow(T[:,:,t_interest][0:zoom3,cross_section3:cross_section3+zoom]-273.15, aspect = 'auto')
plt.colorbar()
plt.title('Temperture Zoomed in from Cross-section at Node %s' %cross_section3)
plt.xlabel('Pipe Length, [steps] - Inlet on Left, Wall on Top')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#7
plt.plot(z,Tb[:,t_interest]-273.15,label = "Bulk",linewidth=ld)
plt.plot(z,T[150,:,t_interest]-273.15,label = "Node 150",linewidth=ld)
plt.plot(z,T[25,:,t_interest]-273.15,label = "Node 25",linewidth=ld)
plt.plot(z,T[3,:,t_interest]-273.15,label = "Node 3",linewidth=ld)
plt.plot(z,T[1,:,t_interest]-273.15,label = "Node 1",linewidth=ld)
plt.title('Temperatures at Varying Axial Cross-sections')
plt.ylabel('Temperatures, [oC]')
plt.xlabel('Pipe Length, [m] - Inlet on Left')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

```

```

#8
plt.plot(tshow,cshow,label = "Csol",linewidth=ld)
plt.plot(tshow,cwaxshow,label = "Cwax",linewidth=ld)
plt.title("Solubility Curve")
plt.xlabel('Temperature, [K]')
plt.ylabel('Concentration, [kg/m^3]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#9
plt.plot(z,Csbulk[:,t_interest], label = "Bulk Solubility",linewidth=ld)
plt.title('Bulk Solubility along Pipeline')
plt.ylabel('Concentration, [kg/m^3]')
plt.xlabel('Pipe Length, [m] - Inlet on Left')
plt.grid()
plt.show()

#10
plt.plot(y[0:zoom],Cs[:,cross_section2,t_interest][0:zoom],linewidth=ld)
plt.title('Solubility Cross Section')
plt.xlabel('Radial Position from Wall, [m]')
plt.ylabel('Solubility, [kg/m^3]')
plt.grid()
plt.show()

#11
plt.imshow(Cs[:,t_interest], aspect = 'auto')
plt.colorbar()
plt.title('Solubility')
plt.xlabel('Pipe Length, [steps]')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

```

```

#12
plt.imshow(Cs[:,t_interest][0:zoom3,0:zoom], aspect = 'auto')
plt.colorbar()
plt.title('Solubility Zoomed in near Inlet and Wall')
plt.xlabel('Pipe Length, [steps] - Inlet on Left, Wall on Top')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#13
plt.plot(z,Cbulk[:,t_interest],label = "Concentration",linewidth=ld)
plt.plot(z,Csbulk[:,t_interest], label = "Solubility",linewidth=ld)
plt.title('Bulk Concentration and Solubility along Pipeline')
plt.ylabel('Concentration, [kg/m^3]')
plt.xlabel('Pipe Length, [m] - Inlet on Left')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#14
plt.plot(y[0:zoom],Cs[:,cross_section2,t_interest][0:zoom], label = "Solubility Curve",linewidth=ld)
plt.plot(y[0:zoom],C[:,cross_section2,t_interest][0:zoom], label = "Concentration",linewidth=ld)
plt.plot(y[0:zoom],C_sol[:,cross_section2,t_interest][0:zoom],label = "Solubility Model")
plt.title('Concentration and Solubility Cross Section at Node %s' %cross_section2)
plt.xlabel('Radial Position from Wall, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#15
plt.plot(y[0:zoom],C_sat[:,cross_section2,t_interest][0:zoom],linewidth=ld)
plt.title('Cross Section Showing Supersaturation at Node %s' %cross_section2)
plt.xlabel('Radial Position from Wall, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.grid()
plt.show()

```



```

#16
plt.plot(y[0:zoom],Cs[:,cross_section3,t_interest][0:zoom], label = "Solubility",linewidth=ld)
plt.plot(y[0:zoom],C[:,cross_section3,t_interest][0:zoom], label = "Concentration",linewidth=ld)
plt.plot(y[0:zoom],C_sol[:,cross_section3,t_interest][0:zoom],label = "No kinetics",linewidth=ld)
plt.title('Concentration and Solubility Cross Section at Node %s' %cross_section3)
plt.xlabel('Radial Position from Wall, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#17
plt.plot(y[0:zoom],C_sat[:,cross_section3,t_interest][0:zoom],linewidth=ld)
plt.title('A Cross Section Showing Supersaturation at Node %s' %cross_section3)
plt.xlabel('Radial Position from Wall, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.grid()
plt.show()

#18
plt.imshow(C[:,:,t_interest], aspect = 'auto')
plt.colorbar()
plt.title('Concentration')
plt.xlabel('Pipe Length, [steps]')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#19
plt.imshow(C[:,:,t_interest][0:zoom3,0:zoom], aspect = 'auto')
plt.colorbar()
plt.title('Concentration Zoomed in near Inlet and Wall', size=fontsize)
plt.xlabel('Pipe Length, [steps] - Inlet on Right, Wall on Top', size=fontsize)
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#20
plt.imshow(C[:,:,t_interest][0:zoom3,cross_section3:cross_section3+zoom], aspect = 'auto')
plt.colorbar()
plt.title('Concentration Zoomed in from Node %s' %cross_section3)
plt.xlabel('Pipe Length, [steps] - Inlet on Left, Wall on Top')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#21
plt.plot(z,Cs[cross_section1,:,t_interest],label = "Sol",linewidth=ld)
plt.plot(z,C[cross_section1,:,t_interest],label = "Conc",linewidth=ld)
plt.title('Concentration and Solubility at Axial Cross-section %s ' %cross_section1, size=fontsize)
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Concentration at %s, [kg/m^3]' %cross_section1)
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#22
plt.plot(z,C_sat[cross_section1,:,t_interest],label = "Bulk",linewidth=ld)
plt.title('Super saturation at Axial Cross-section %s ' %cross_section1)
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.grid()
plt.show()

```

```

#23
plt.plot(z,C[cross_section2,:,t_interest],label = "Concentration",linewidth=ld)
plt.plot(z,Cs[cross_section2,:,t_interest],label = "Solubility",linewidth=ld)
plt.title('Concentration at Axial Cross-section %s ' %cross_section2)
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#24
plt.plot(z,C_sat[cross_section2,:,t_interest],label = "Bulk",linewidth=ld)
plt.title('Super saturation at Axial Cross-section %s ' %cross_section2)
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Concentration, [kg/m^3]')
plt.grid()
plt.show()

#25
plt.imshow(C_sat[:, :, t_interest][0:zoom, :], aspect = 'auto')
plt.colorbar()
plt.title('Super Saturation Concentration Zoomed in near Inlet and Wall')
plt.xlabel('Pipe Length, [steps] - Inlet on Left, Wall on Top')
plt.ylabel('Radial Position, [steps]')
plt.grid()
plt.show()

#26
plt.plot(z,qt[:,t_interest],linewidth=ld)
plt.title('Mass Flux to Deposit')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Mass Flux, [kg/(m^2.s)]')
plt.grid()
plt.show()

```

```

#27
plt.plot(z,Cm[:,t_interest])
plt.title('Concentration at Deposit')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Concentration at Deposit, [kg/(m^3)]')
plt.grid()
plt.show()

#28
plt.plot(z,theta[:,t_interest],linewidth=ld)
plt.title('Characteristic Time')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Characteristic Time, [s]')
plt.grid()
plt.show()

#29
plt.plot(z,qd[:,t_interest],label = "qd",linewidth=ld)
plt.plot(z,qt[:,t_interest],label = "qt",linewidth=ld)
plt.plot(z,qsr[:,t_interest],label = "qsr",linewidth=ld)
plt.title('Flux')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Mass Flux, [kg/(m^2.s)]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#30
plt.plot(z,k[:,t_interest], label = "k",linewidth=ld)
plt.plot(z,kt, label = "kt")
plt.title('Shear-removal constant')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Shear-removal constant, [1/s]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

```

```

#31
plt.imshow( $\phi[:, :, t\_interest]$ , aspect = 'auto', cmap = "hot")
plt.colorbar()
plt.title('Porosity')
plt.xlabel('Pipe Length, [steps]')
plt.ylabel('Normalised Deposit Thickness, [steps]')
plt.grid()
plt.show()

#32
plt.plot( $z, \lambda d\_avg[:, t\_interest]$ , linewidth=ld)
plt.title('Deposit Average Thermal Conductivity')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Deposit Thermal Conductivity, [W/(m.k)]')
plt.grid()
plt.show()

#33
plt.plot( $z, \delta[:, t\_interest]$ , label = "days = %f" %int( $\Delta Time * t\_interest / (3600 * 24)$ ), linewidth=ld)
plt.plot( $z, \delta[:, t\_interest-1]$ , label = "days = %f" %int( $\Delta Time * (t\_interest-1) / (3600 * 24)$ ), linewidth=ld)
plt.plot( $z, \delta[:, t\_interest-2]$ , label = "days = %f" %int( $\Delta Time * (t\_interest-2) / (3600 * 24)$ ), linewidth=ld)
plt.plot( $z, \delta[:, t\_interest-3]$ , label = "days = %f" %int( $\Delta Time * (t\_interest-3) / (3600 * 24)$ ), linewidth=ld)
plt.plot( $z, \delta[:, t\_interest-4]$ , label = "days = %f" %int( $\Delta Time * (t\_interest-4) / (3600 * 24)$ ), linewidth=ld)
plt.plot( $z, \delta[:, t\_interest-5]$ , label = "days = %f" %int( $\Delta Time * (t\_interest-5) / (3600 * 24)$ ), linewidth=ld)
plt.plot( $z, \delta[:, t\_interest-6]$ , label = "days = %f" %int( $\Delta Time * (t\_interest-6) / (3600 * 24)$ ), linewidth=ld)
plt.title('Wax Thickness')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Wax Thickness, [m]')
plt.grid()
plt.show()

```

```

#34
plt.plot( $z, \delta[:, t\_interest]$ , label = "t_interest", linewidth=ld)
plt.plot( $z, av[:, :]$ , label = "Average thickness = %f" %av[0], linewidth=ld)
plt.plot( $z, mx[:, :]$ , label = "Maximum thickness = %f" %mx[0], linewidth=ld)
plt.title('Wax Thickness')
plt.xlabel('Pipe Length, [m]')
plt.ylabel('Wax Thickness, [m]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

#35
plt.imshow( $\delta, T$ , aspect = 'auto', origin = 'lower', cmap = "hot" )
plt.colorbar()
plt.title('Deposit Thickness')
plt.xlabel('Pipe Length, [steps]')
plt.ylabel('Deposit Thickness, [steps]')
plt.grid()
plt.show()

dc_dt, Cwax, Twat, polynomial, coefficient = solubilty(mode3)
pf, pwax,  $\mu f$ ,  $\lambda f$ , Cpf, kr, k_pipe, hi,  $\epsilon, n$ , ksr,  $\lambda wax$ , kt, ho, k_insul = parameters(mode2)
select(mode)

```

8.3 Thermal-difference-based Implementation

8.3.1 Typical Input Data

Table 8.4: Variable Input Values – Thermal-difference-based Implementation

Symbol	Value	Unit	Description
Aged	No Ageing	No	No shear ageing included

T_{hin_in}	313	K	Bulk inlet temperature
T_c	276	K	Sea temperature
T_{watin}	288	K	Wax Appearance Temperature
D	0.35	m	Pipe inner diameter
L	5000	m	Pipe Length
n_l	5000	No	Number of axial elements
m	2	kg/s	Mass flowrate
w_{13in}	0.875	No	Inlet mass fraction of lighter component

Table 8.5: Fixed Property Values – Thermal-difference-based Implementation

Symbol	Value	Unit	Description
new_par	0.5	K	Parameter to force transition into cold-flow
iterations	20	No	Number of iterations of each axial element
ρ_{13}	756.0	kg/m ³	Density of light component
ρ_{29}	808.3	kg/m ³	Density of heavy component
MM ₁₃	184.361	kg/kmol	Molar Mass of light component
MM ₂₉	408.707	kg/kmol	Molar Mass of heavy component
R2	8.314/1000	MJ/(K.kmol)	Universal Gas Constant
ΔH_{m_C13}	28.5	MJ/kmol	Heat of melting of light component
ΔH_{m_C29}	106.6	MJ/kmol	Heat of melting of heavy component
T_{m_C13}	267.8	K	Melting temperature of light component
T_{m_C29}	335.4	K	Melting temperature of heavy component
k_{s_wax}	0.25	W/(m.K.)	Thermal conductivity of solid wax
$T_d - T_h$	0.5	No	Cold Flow temperature difference

8.3.2 Parameter Equations

$$C_{p,l}^{CH_3} = 17.33 + 0.0455T_h \quad (8.25)$$

$$C_{p,l}^{CH_2} = 30.41 + 0.01479T_h \quad (8.26)$$

$$C_{p,l}^{C13} = 2C_{p,l}^{CH_3} + (13 - 2) C_{p,l}^{CH_2} \quad (8.27)$$

$$C_{p,l}^{C29} = 2C_{p,l}^{CH_3} + (29 - 2) C_{p,l}^{CH_2} \quad (8.28)$$

$$C_{p,l}^{mix} = C_{p,l}^{C13} w_{13} + C_{p,l}^{C29} w_{29} \quad (8.29)$$

$$\rho_l^{mix} = \left(\frac{w_{13}}{\rho_{13}} + \frac{w_{29}}{\rho_{29}} \right)^{-1} \quad (8.30)$$

$$\varphi_{13} = \frac{g_{13}v_{13}}{g_{13}v_{13} + g_{29}v_{29}} \quad (8.31)$$

$$\varphi_{29} = \frac{g_{29}v_{29}}{g_{13}v_{13} + g_{29}v_{29}} \quad (8.32)$$

$$k_{13,29} = \frac{2}{\left(\frac{1}{k_{13}}\right) + \left(\frac{1}{k_{29}}\right)} \quad (8.33)$$

$$k_l^{mix} = k_{13}\varphi_{13}^2 + k_{13,29}\varphi_{13}\varphi_{29} + k_{29}\varphi_{29}^2 \quad (8.34)$$

$$k_d = k_l^{mix}V_l + k_{wax,s}V_s \quad (8.35)$$

$$\mu = 10^{-3} \exp\left(-4.75 + \frac{1720}{T_h}\right) \quad (8.36)$$

$$Pr = \frac{C_{p,l}^{mix} \mu}{k} \quad (8.37)$$

$$Re = \frac{4\rho_l^{mix}F}{\pi\mu(D-2\delta)} \quad (8.38)$$

$$h_h = \frac{0.023Re^{0.8}Pr^{0.3}k_l^{mix}}{D-2\delta} \quad (8.39)$$

8.3.3 Thermal-difference-based Implementation's Code



UNIVERSITY OF THE WITWATERSRAND
JOHANNESBURG

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.rc('font', size = 25, family = 'Times New Roman')
%matplotlib inline
from ipywidgets import interact, fixed
import time

start_time = time.time()                # Timing code run time
nl = 100                                # Number of increments

def model(Aged,Thin_in,Tc,Twatin,D,L,nl,m,w13in):
    new_par = 0.5
    iterations = 20
    interest = nl
    cold_flow = False
    Thout_in = Thin_in - 1                # [K] Initial guess of segment outlet temperature

    # Initialising Vectors
    Tc = np.ones(nl)*Tc                  # Constant for whole pipe
    Twat = np.ones(nl)*Twatin             # Constant for whole pipe
    z = np.linspace(0,L,nl)              # Vector for plotting
```

```
# Initialising Empty Vectors
Thin = np.zeros(nl)
Thout = np.zeros(nl)
Td = np.zeros(nl)
q = np.zeros(nl)
qd = np.zeros(nl)
qh = np.zeros(nl)
Cp_l_CH3 = np.zeros(nl)
Cp_l_CH2 = np.zeros(nl)
C13 = np.zeros(nl)
C29 = np.zeros(nl)
C = np.zeros(nl)
μ = np.zeros(nl)
Re = np.zeros(nl)
Pr = np.zeros(nl)
hh = np.zeros(nl)
δ = np.zeros(nl)
χ = np.zeros(nl)
Δt = np.zeros(nl)
ΔP = np.zeros(nl)
Tδ = np.zeros(nl)
k113 = np.zeros(nl)
k129 = np.zeros(nl)
k = np.zeros(nl)
k13_29 = np.zeros(nl)
kd = np.zeros(nl)
```

```

β = np.zeros(n1)
w29_star = np.zeros(n1)
w13_star = np.zeros(n1)
w29_Th = np.zeros(n1)
w13_Th = np.zeros(n1)
f = np.zeros(n1)
Δf = np.zeros(n1)
Vs = np.zeros(n1)
Vl = np.zeros(n1)
Vsβ = np.zeros(n1)
Vlβ = np.zeros(n1)
WPT = np.zeros(n1)
frac_change = np.zeros(n1)
w13 = np.zeros(n1)
g13 = np.zeros(n1)
g29 = np.zeros(n1)
φ13 = np.zeros(n1)
φ29 = np.zeros(n1)
p_l_mix = np.zeros(n1)
F = np.zeros(n1)

# Initial conditions
Thin[0] = Thin_in # [K] Inlet mixture temperature
Thout[0] = Thout_in # [K] Outlet mixture temperature - Initial guess
Td[0] = Twat[0]
w13[0] = w13in
p13 = 756.0 # [Kg/m^3] Density of C13H28
p29 = 808.3 # [Kg/m^3] Density of C29H60
WPT[0] = 300 # [K] Inlet wax precipitation temperature
χ[0] = 1
w29 = 1-w13 # [No]

```

```

MM13 = 184.361 # [Kg/Kmol] Molar mass of C13H28
MM29 = 408.707 # [Kg/Kmol] Molar mass of C29H60

N13 = 13 # [No] N = number of carbon atoms in the alkane
N29 = 29

c = 0.046 # [No] From Haj - for turbulent flow - pressure drop calc
n = 0.2 # [No] From Haj - for turbulent flow - pressure drop calc
R2 = 8.314/1000 # [MJ/(K.kmol)] Universal gas constant

ΔHm_C13 = 28.5 # [MJ/kmol]
ΔHm_C29 = 106.6 # [MJ/kmol]
Tm_C13 = 267.8 # [K]
Tm_C29 = 335.4 # [K]

Th = (Thin + Thout)/2 # [K]
ΔL = L/n1 # [m]

R = D/2 # [m]
Ax = np.pi*(R**2) # [m^2] Cross sectional area

Ai = np.pi*D*ΔL # [m^2] Pipe surface sectional area

v13 = (p13/MM13)**-1 # [m^3/kmol] Molar volume of C13H28
v29 = (p29/MM29)**-1 # [m^3/kmol] Molar volume of C29H60

for seg in range(interest):

    g13[seg] = (w13[seg]/MM13)/((w13[seg]/MM13)+(w29[seg]/MM29)) # [No] Mole fraction of C13H28
    g29[seg] = 1 - g13[seg] # [No] Mole fraction of C29H60

```

```

φ13[seg] = g13[seg]*v13/(g13[seg]*v13+g29[seg]*v29)      # [No] Superficial volume fraction of C13H28
φ29[seg] = g29[seg]*v29/(g13[seg]*v13+g29[seg]*v29)      # [No] Superficial volume fraction of C29H60

ρ_l_mix[seg] = (w13[seg]/p13+w29[seg]/p29)**-1             # [kg/m^3] Density

F[seg] = m/ρ_l_mix[seg]                                     # [m^3/s] Volume flow rate

if Th[seg] > Tc[seg]:
    Tδ[seg] = (Tc[seg] + Td[seg])/2                         # [K] Average deposit temp

    w29_star[seg] = np.e**(-(ΔHm_C29)/R2)*((1/Tδ[seg])-(1/Tm_C29)) # [No]
    w29_Th[seg] = np.e**(-(ΔHm_C29)/R2)*((1/Th[seg])-(1/Tm_C29))

    f[seg] = (w29_Th[seg]-w29_star[seg])/(1-w29_star[seg])    # [No] Solid mass fraction
    Vs[seg] = f[seg]/(f[seg]+(1-f[seg])*(p29/p13))           # [No] Solid Volume Fraction

    k113[seg] = (0.00018*(1121.4-Th[seg]))                   # For C16 [W/m.K]
    k129[seg] = (0.00014*(1371.4-Th[seg]))                   # For C28 [W/m.K]
    k_s_wax = 0.25                                           # [W/m.K] Solid wax thermal conductivity

    μ[seg] = (10**-3)*np.e**(-4.75+(1720/Th[seg]))           # [Pa.s]
    Re[seg] = (4*ρ_l_mix[seg]*F[seg])/(np.pi*μ[seg]*(R*2-2*δ[seg])) # [No]

    if Re[seg] > 0:
        β[seg] = 79.394 + 7.256/(1+np.e**(-(np.log(Re[seg])-4.411)/0.220)) # [Degrees]

    Vsβ[seg] = Vs[seg]/np.cos(β[seg]*np.pi/180)              # [No]

```

```

# If you want aged wax
if Aged == "Aging":
    Vs[seg] = Vsβ[seg]

    if Vs[seg] > 1:
        Vs[seg] = 1

Vl[seg] = 1 - Vs[seg]

# Mixture approximations
k13_29[seg] = 2/((1/k113[seg])+(1/k129[seg]))
# Thermal conductivity of wax-solvent mixture
k[seg] = k113[seg]*(φ13[seg]**2) + 2*k13_29[seg]*φ13[seg]*φ29[seg] + k129[seg]*(φ29[seg]**2)
kd[seg] = ((k[seg]*Vl[seg])+(k_s_wax*Vs[seg]))               # [W/(m.K)] Thermal conductivity of the deposit

Cp_1_CH3[seg] = 17.33 + 0.0455*Th[seg]
Cp_1_CH2[seg] = 30.41 + 0.01479*Th[seg]

C13[seg] = 2*Cp_1_CH3[seg] + (N13-2)*Cp_1_CH2[seg]          # [J/(kg.K)]
C29[seg] = 2*Cp_1_CH3[seg] + (N29-2)*Cp_1_CH2[seg]
C[seg] = ((C13[seg]*w13[seg])+(C29[seg]*w29[seg]))

Pr[seg] = μ[seg]*C[seg]/k[seg]                                # [No]

if χ[seg] < 0.0001:
    cold_flow = True
else:
    cold_flow = False

```



```

if cold_flow == False or (Th[seg] > Twat[seg] + new_par):      # Cold flow regime only
    Td[seg] = Twat[seg]
else:
    #Twat[seg] = WPT[seg]
    #Th[seg] = Twat[seg]
    Twat[seg] = Th[seg]
    Td[seg] = Th[seg] - new_par
    Δf[seg] = f[seg-1] - f[seg]
    w29[seg] = w29[seg] - w29[seg]*Δf[seg]

if Td[seg] < Tc[seg]:
    Td[seg] = Tc[seg]

for iter in range(0, iterations):

    hh[seg] = 0.023*(Re[seg]**0.8)*(Pr[seg]**0.3)*k[seg]/(R*2-2*δ[seg]) # [W/(m^2.K)] Dittus-Boelter correlation

    if Td[seg] == Twat[seg]:

        q[seg] = hh[seg]*(R-δ[seg])*(2*np.pi*ΔL)*(Th[seg]-Td[seg])      # [W]

        δ[seg] = R*(1-np.e**((kd[seg]*2*np.pi*ΔL*(Tc[seg]-Td[seg])/q[seg])))

        Thout[seg] = Thin[seg] - q[seg]/(m*C[seg])

    else:

        q[seg] = hh[seg]*(R-δ[seg])*(2*np.pi*ΔL)*(Th[seg]-Td[seg])

        δ[seg] = R*(1-np.e**((kd[seg]*2*np.pi*ΔL*(Tc[seg]-Td[seg])/q[seg])))

```

```

Thout[seg] = Thin[seg] - q[seg]/(m*C[seg]) + Δf[seg]*ΔHm_C29*1000000/(MM29*C[seg])

if seg < n1-1:
    Th[seg+1] = (Thin[seg] + Thout[seg])/2
    Thin[seg+1] = Thout[seg]
else:
    δ[seg-1] = 0
    q[seg-1] = 0
    Thout[seg] = Tc[seg]
    Thin[seg] = Tc[seg]
    Th[seg] = Tc[seg]
    Td[seg] = Tc[seg]
    Twat[seg] = Tc[seg]

ΔP[seg] = ((2*c*p_l_mix[seg]*ΔL)/((D-2*δ[seg])**5-n))**((μ[seg]/p_l_mix[seg])**n)*((4*F[seg]/np.pi)**(2-n))

Δt[seg] = (np.pi*((R-δ[seg])**2)*ΔL)/F[seg]      # Δt = average resistance time for the axial element

χ[seg] = (Thin[seg] - Thout[seg])/Δt[seg]      # Rate of cooling

WPT[seg] = Twat[seg] - 4.16*χ[seg] + 6.68*np.log(w29[seg])

if seg < n1 -1:
    WPT[seg+1] = WPT[seg]

ld = 2
av = np.mean(δ)
mx = np.amax(δ)

```

```

print(av)
print(mx)

plt.plot(z,Thout-273.15,label = 'Bulk Temp', linewidth=ld)
plt.plot(z,Twat-273.15,label = 'WAT', linewidth=ld)
plt.plot(z,Td-273.15,label = 'Deposit Temp', linewidth=ld)
plt.plot(z,Tc-273.15,label = 'Coolant Temp', linewidth=ld)
plt.title('Temperatures vs Length')
plt.ylabel('Temp, [C]')
plt.xlabel('Pipe Length, [m]')
plt.legend(prop={'size':20})
plt.grid()
plt.show()

plt.plot(z,q, linewidth=ld)
plt.title('Heat Loss vs Length')
plt.ylabel('Heat Loss, [W]')
plt.xlabel('Pipe Length, [m]')
plt.grid()
plt.show()

plt.plot(z,d/R, linewidth=ld)
plt.title('Dimensionless Wax Thickness vs Length')
plt.ylabel('Wax Thickness/Radius')
plt.xlabel('Pipe Length, [m]')
plt.grid()
plt.show()

```

```

plt.plot(z,Th-Twat, linewidth=ld)
plt.title('Th -Twat')
plt.ylabel('Th -Twat, [K]')
plt.xlabel('Pipe Length, [m]')
plt.grid()
plt.show()

plt.plot(z,w29_Th, label='w29_Th', linewidth=ld)
plt.plot(z,w29, label='w29')
plt.title('Heavy Component Mass Fraction vs Length')
plt.ylabel('Mass Fraction')
plt.xlabel('Pipe Length, [m]')
plt.legend()
plt.grid()
plt.show()

plt.plot(z,x, linewidth=ld)
plt.title('Rate of Cooling vs Length')
plt.ylabel('Rate of Cooling, [K/s]')
plt.xlabel('Pipe Length, [m]')
plt.grid()
plt.show()

plt.plot(z,dP, linewidth=ld)
plt.title('Pressure Drop vs Length')
plt.ylabel('Pressure Drop, [Pa]')
plt.xlabel('Pipe Length, [m]')
plt.grid()
plt.show()

```

```

plt.plot(z,Af, linewidth=ld)
plt.title('Change in Mass Fraction of Deposit')
plt.ylabel('Solid Fraction Precipitated in Segment')
plt.xlabel('Pipe Length, [m]')
plt.grid()
plt.show()

plt.plot(z,Vs,label = 'Solid Fraction', linewidth=ld)
plt.plot(z,Vl,label = 'Liquid Fraction', linewidth=ld)
plt.title('Volume Fractions in Deposit in Segment')
plt.ylabel('Volume Fractions in Deposit in Segment')
plt.xlabel('Pipe Length, [m]')
plt.legend()
plt.grid()
plt.show()

print("%f seconds" % (time.time() - start_time))

interact(model,Aged=("Aging", "No aging"),Thin_in=(303,323),Tc=(273,280),Twatin=(278,298),
        D=(0.1,0.6),L=(1000,10000,1000),nl=(1000,10000,1000),m=(1,3),w13in=(0.8,0.95))

```

8.4 Effect of Changing the Operating Temperature

Table 8.6: Maximum Thickness of Different Solubility Curves with Temperature

Slope [kg/(m ³ .K)]/ Shape	Thickness [mm], T _{in} = 20, T _{sea} = 0	Thickness [mm], T _{in} = 30, T _{sea} = 0	Thickness [mm], T _{in} = 30, T _{sea} = 10
0.1	0.327	0.735	0.35
Convex	0.281	0.282	0.145
Concave	0.381	3.268	0.895
Plateau	0.279	0.279	0.107
Plateau - steep	0.295	0.295	0.089

Table 8.7: Average Thickness of Different Solubility Curves with Temperature

Slope [kg/(m ³ .K)]/ Shape	Thickness [mm], T _{in} = 20°C, T _{sea} = 0 °C	Thickness [mm], T _{in} = 30 °C, T _{sea} = 0 °C	Thickness [mm], T _{in} = 30, T _{sea} = 10 °C
0.1	0.097	0.158	0.102
Convex	0.12	0.149	0.082
Concave	0.075	0.224	0.138
Plateau	0.145	0.157	0.067
Plateau - steep	0.162	0.163	0.056

Table 8.8: Thermal-difference-based Implementation's Thickness with Temperature

	Thickness [mm], $T_{in} = 20\text{ }^{\circ}\text{C}$, $T_{sea} = 0\text{ }^{\circ}\text{C}$	Thickness [mm], $T_{in} = 30\text{ }^{\circ}\text{C}$, $T_{sea} = 0\text{ }^{\circ}\text{C}$	Thickness [mm], $T_{in} = 30\text{ }^{\circ}\text{C}$, $T_{sea} = 10\text{ }^{\circ}\text{C}$
Average	0.114	0.114	0.091
Maximum	0.156	0.156	0.020