Transient Flow Modelling of Carbon Dioxide (CO_2) Injection into Depleted Gas Fields

A thesis submitted to the University College London for the degree of Doctor of Philosophy

By

Revelation Jacob Samuel

Department of Chemical Engineering
University College London
Torrington Place
London WC1E 7JE
United Kingdom

October 2019

I, Revelation Jacob Samuel confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Name Signature Date

Abstract

The internationally agreed global climate deal reached at the Paris Climate Conference in 2015 is intended to limit the increase in global average temperatures to 'well below' 2°C above pre-industrial levels. This comes in addition to the European Union ambition for 80% to 95% reduction in the 1990 greenhouse gas emissions by 2050 in order to avoid dangerous climate change. Most scenario studies indicate that Carbon Capture and Storage (CCS) is essential for achieving such ambitious reductions.

In CCS operations, depleted gas fields represent prime targets for large-scale storage of the captured CO_2. Considering the relatively low wellhead pressure of such fields, the uncontrolled injection of the high-pressure dense phase CO_2 will result in its rapid, quasi-adiabatic Joule-Thomson expansion leading to significant temperature drops. This could pose several risks, including blockage due to hydrate and ice formation following contact of the cold sub-zero CO_2 with the interstitial water around the wellbore and the formation water in the perforations at the near well zone, thermal stress shocking and fracture of the wellbore casing steel and over-pressurisation accompanied by CO_2 backflow into the injection system due to the violent evaporation of the superheated liquid CO_2 upon entry into the wellbore.

In order to minimise the above risks and develop best-practice guidelines for the injection of CO_2, the accurate prediction of the CO_2 pressure and temperature along the well during the injection process is of paramount importance.

This thesis deals with the development and verification of a Homogeneous Equilibrium Mixture (HEM) model and a Homogenous Equilibrium Relaxation Mixture (HERM) model for simulating the transient flow phenomena taking place during the injection of dense phase CO_2 into depleted gas fields. The HEM model assumes instantaneous interface mass, momentum and energy exchange between the constituent CO_2 liquid and vapour phases. As such they remain at the same pressure, temperature and velocity, whence the corresponding fluid-flow may be described using a single set of mass, momentum and energy conservation equations. The HERM on the other hand presents an additional equation which accounts for the thermodynamic non-equilibrium thorough the introduction of a relaxation time. It also accounts for phase and flow dependent fluid/wall friction and heat transfer, variable well cross sectional area as well as deviation of the well from the vertical. At the well inlet, the opening of the upstream flow regulator valve is modelled as an isenthalpic expansion process; whilst at the well outlet, a formation-specific pressure-mass flow rate correlation is adopted to characterise the storage site injectivity.

The testing of the models is based on their application to CO_2 injection into the depleted 2582 m deep Goldeneye Gas Reservoir at Hewett field in the North Sea for which the required design and operational data are publically available. Varying injection scenarios involving the rapid (5 mins), medium (30 mins) and slow (2 hrs) linear ramping up of the injected CO_2 flow rate to the peak nominal value of 33.5 kg/s are simulated. In each case, the simulated pressure and temperature transients at the top and bottom of the well are used to ascertain the risks of well-bore thermal shocking or interstitial ice formation leading to well blockage due to the rapid cooling of the CO_2. Detailed sensitivity analysis of the most important parameters affecting the CO_2 in-well flow behaviour, including the wellbore diameter variations, well inclination, upstream temperature, pressure and time variant injection mass flow rate are conducted. The simulation results obtained for a slow (2 hrs) flowrate ramp-up case using the HEM model produce a minimum wellhead temperature of - 11 ^oC. The corresponding minimum temperature using the HERM model on the other hand is - 21 ^oC, demonstrating the importance of accounting for non-equilibrium effects and the model's usefulness as a tool for the development of optimal injection strategies for minimising the risks associated with the injection of CO_2 into depleted gas fields.

Impact Statement

This thesis investigates the risks associated with the rapid, quasi-adiabatic Joule-Thomson expansion that leads to significant temperature drops during geological sequestration of high-pressure dense phase CO_2. The thesis specifically deals with the development and verification of two mathematical models for simulating the transient flow phenomena taking place during the injection of CO_2 into depleted gas fields.

As Carbon Capture and Storage (CCS) is widely considered to be a key technology for mitigating the continuous rise in global temperatures, depleted gas fields are potential targets for the large-scale storage of the captured CO_2. Thus, the safe injection of the captured CO_2 is of paramount importance. The injection models developed in this thesis can serve as powerful tools for the development of optimal injection strategies and best-practice guidelines for minimising the risks associated with the injection of CO_2 into depleted gas fields.

In addition, the models can easily be extended to account for the consequences of CO_2 mixture with other important process fluids such as hydrogen and hydrocarbons.

Finally, the numerical scheme introduced in this thesis for solving the flow model governing equations have shown improved accuracy and computational efficiency over the conventional methods, which is beneficial to the further development of computational fluid dynamics in chemical process engineering.

Nomenclature

A – Cross-sectional area of injection tube, m^2

a – Speed of sound, ms^{-1}

$\tilde{A}$ – The minimum pressure required for the flow to start from the well into the reservoir, Pa^2

B – Threshold parameter

$\tilde{B}$ – Site-specific dimensional constants, $Pa^2.s/kg$

C_0 – Profile parameter

c – Characteristics wave speed, ms^{-1}

$\tilde{C}$ – Site-specific dimensional constants, $Pa^2.s\ ^2/kg^2$

D – Diameter of injection tube, m

E – Specific energy, J

e – Internal energy, J

F – Force applied, N or Fluxes

f – Frictional force, N

f^c – Convective flux, $mol.cm^{-2}s^{-1}$

f^p – Pressure flux, $Pa.m^{-1}$

f_w– Fanning friction factor

H – Total enthalpy, $J.kg^{-1}$

h – Specific enthalpy, $J.kg^{-1}$

$\bar{h}$ – Convective heat transfer coefficient, $W.m^{-2}K^{-1}$

J - Joules

K – Kelvin

K_u – Kutateladze number

L – Wellbore depth, m

M – Constant mass flux, $kg.\ s^{-1}m^{-2}$

$\dot{m}$ – Variable mass flux, $kg.\ s^{-1}m^{-2}$

N – Newton

N_B – Bond number

Nu – Nusselt number

P – Pressure, bara

P_{BHP} – The instantaneous bottom-hole pressure, bara

Pr – Prandtl number

P_{res} – The reservoir static pressure, bara.

Q – Heat flux, W.m^{-2} K^{-1}

R – Ideal gas constant

Re – Reynolds number

r – Radius of injection tube, m

T – Temperature, K

S – Source terms

s – Specific entropy, J.K^{-1}

t – Time, s

U – Conserved variables

u – Velocity, ms^{-1}

x – Vapour quality

z – Distance between grid cells, m

Z – Compressibility factor

Subscripts

c - Characteristics

d – Drift

eq - Equivalent

f – Fluid

g – Gravity, ms^{-2}

ig – ideal gas

l – Liquid

m – Mixture

ml – meta-stable liquid phase

p – Pressure, bara

sgf – Gas superficial velocity, ms^{-1}

sv – saturated vapour

v - Vapour

w – Wellbore or wall

Superscripts

e – Energy

L - Left

m - Mass

mom - Momentum

n – Number of cell

R - Right

Special characters

ρ – Density of fluid, kg.m^{-3}

θ – Angle of inclination

β – Gravitational term

∂ – Differential

γ – slip parameter

σ – Surface tension, N.m^{-1}

η – Heat transfer coefficient, W.m^{-2}K-$^{-1}$

$\mathcal{M}$ – Mach number

λ – Thermal conductivity, W.m^{-1}K^{-1}

$\mathcal{P}$ - Polynomials

μ – Viscosity of fluid, m^{2}.s^{-1}

$\emptyset$ – Relaxation time

ω – Acentric factor

∞ – Infinity

ζ , $\mathcal{L}$ – Waves amplitude

κ – Heat conductivity, W.m^{-1}K^{-1}

v – Specific volume, m^3.kg^{-1}

Abbreviations

CCS – Carbon Capture and Storage

CFL – Courant–Friedrichs–Lewy condition

EoS – Equation-of-State

FVM – Finite Volume Method

HEM – Homogeneous Equilibrium Mixture

HERM – Homogeneous Equilibrium Relaxation Mixture

HLL – Harten, Lax, van Leer solver

MoC – Method of Characteristics

MD – Measured Depth

TVD – True Vertical Depth

Acknowledgements

I give all glory and honour to God Almighty the giver of life and Jesus Christ the source of all wisdom, knowledge and understanding.

I would like to convey my enormous gratitude to Inain, my beloved wife, for her support and encouragement all the way. To our beautiful daughters Chloe, Cherish and Rhema I love you all and many thanks for making the experience more interesting. To Pst Victor Patrick, Pst Henry Onyirioha, Pst Philip Obih and Pst Frank Oyibo, I remain grateful for your prayers and moral support. And I say thank you to all my beloved brethren at DLBC Clapham and DLBC Luton and the Holy Ghost Warriors prayer team.

My profound gratitude goes to my supervisor Professor Haroun Mahgerefteh for the fatherly love and excellent guidance throughout my PhD studies at UCL.

I would also like to thank Dr. Andrea Sacconi for his excellent advice and contributions toward the success of this thesis. Also, my sincere gratitude to Dr. Solomon Brown, Dr. Sergey Martynov, Dr RJT Porter, Dr. Wentian Zheng, Dr. Nor Binti Daud, Mr. Yu Jianhao and Mr Fidal Bashir for their encouragement, patience and for providing an excellent work place.

Also, I will like to thank all my fellow scholars and friends at UCL and beyond who supported me during the course of this PhD, particularly Sunny Emmanuel, Michael Akpan, Jude Majasan, Jude Ejeh, Ayoola Shoneye, Babatunde Badmos, Massud Baba, Emmanuel Agunyole, Anthony Futughe and many more.

Finally, I want say a big thank you to the Government of the Federal Republic of Nigeria for funding my PhD studies at UCL.

List of Publications

1. Revelation J. Samuel and Haroun Mahgerefteh, (2017). Transient Flow Modelling of Start-up CO_2 Injection into Highly-depleted Oil/Gas Fields. *International Journal of Chemical Engineering Applications*, Vol. 8, No. 5, pp. 319 – 326

2. Revelation Samuel and Haroun Mahgerefteh, (2017). Transient Flow Modelling of Start-up CO_2 Injection into deep geological formations. Poster at UKCCSRC Autumn 2017 Biannual. https://ukccsrc.ac.uk/sites/default/files/documents/event/revelation-samuel.pdf

3. Revelation J. Samuel and Haroun Mahgerefteh, (2018). Transient Flow Modelling of the Impact of CO_2 Stream Impurities During Geological Sequestration. *The European Conference on Sustainability, Energy & the Environment 2018 Official Conference Proceedings*. https://papers.iafor.org/submission41420/

4. Revelation J. Samuel and Haroun Mahgerefteh, (2019). Investigating the impact of flow rate ramp-up on Carbon Dioxide start-up injection. *International Journal of Greenhouse Gas Control*, 88 (2019) 482–490

5. Andrea Sacconi, Revelation Samuel and Haroun Mahgerefteh, (2019). Modelling start-up injection of CO_2 into highly-depleted gas fields. *Journal of Energy*, (under 2nd review for publication).

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 The carbon dioxide problem

The impact of global warming of 1.5 °C temperature rise above pre-industrial levels and related global greenhouse gas emission pathways was reported by the Intergovernmental Panel on Climate Change IPCC, (2018) identifying CO_2 emission as a major contributor. The Panel in its previous publication IPCC, (2014) concluded that "warming of the climate system is unequivocal and that increased emission in greenhouse gases, particularly CO_2 in the atmosphere have been accompanied by warming of the atmosphere and oceans, reducing snow and ice, ocean acidification and sea level rise". The Panel concludes that "there is an immediate need for implementation of various actions to reduce CO_2 emissions to mitigate these changes, including increased energy supply from renewable and nuclear sources, increased energy efficiency and moving to fossil-fuel based power with carbon capture and storage (CCS)" (IPPC, 2014; Fisher et al., 2007; Blackford et al., 2010).

Also in 2015, 198 Nations met in Paris for the Global Climate Change Summit and agreed on a global temperature control regulation called the "Paris Agreement" (United Nations, 2015). Its aim as described in the UN Paris Agreement (2015) is to enhance the implementation of the United Nations Framework Convention on Climate Change (COP21 Paris Agreement) through;

"(a) holding the increase in the global average temperature to well below 2 °C above pre-industrial levels and to pursue efforts to limit the temperature increase to 1.5 °C above pre-industrial levels, recognizing that this would significantly reduce the risks and impacts of climate change;

(b) increasing the ability to adapt to the adverse impacts of climate change and foster climate resilience and low greenhouse gas emissions development, in a manner that does not threaten food production;

(c) making finance flows consistent with a pathway towards low greenhouse gas emissions and climate-resilient development."

On the other hand, the Global Energy Perspective (GEP) forecasts that due to the continuous increase in the world population the world primary energy demand will also rise per year (IEA, 2011). Figure 1.1 shows the relationship between the world population and the rate of energy consumption.

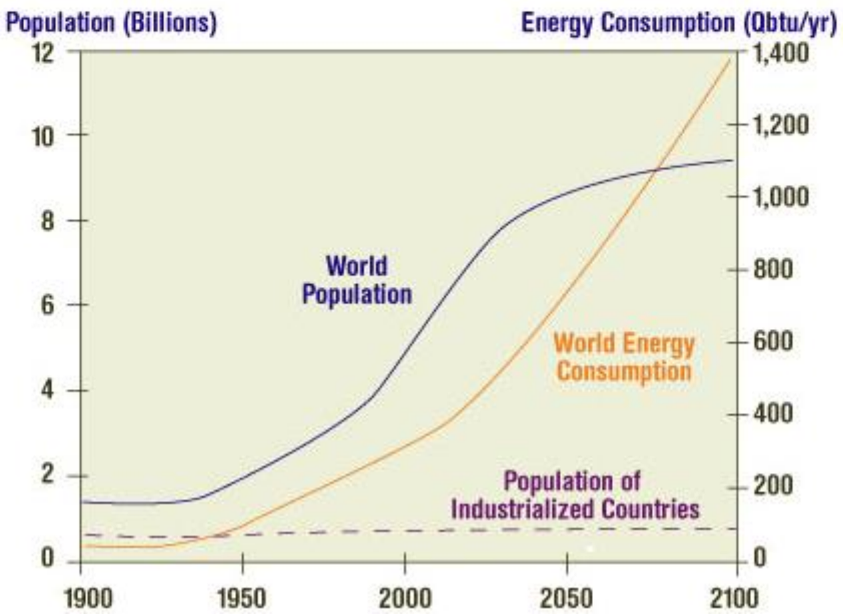


Figure 1.1: World population and energy demand growth (IEA, 2011)

Given the increasing demand in the world primary energy, and in the absence of sufficient amount of renewable energy to meet such need in the near term (10 to 20 years), the demand for fossil fuels is expected to increase, leading to increase in greenhouse gas emissions.

1.2 CCS and CO_2 storage options

In the search for a possible large-scale mitigation strategy for the anthropogenic CO_2 emissions, Carbon Capture and Storage (CCS) technology has emerged as a highly viable option. The concept of CCS is simple: capturing the CO_2 produced from the use of fossil fuels and other industrial emitters followed by its transportation, usually using high pressure pipelines for permanent storage.

The most suitable location for large-scale sequestration is deep geological formations, such as highly-depleted oil/gas reservoirs that would be (Nordbotten and Celia, 2011):

i. "sufficiently permeable to accept large quantities of CO_2

and

ii. overlain by very low-permeability formations that will keep the injected buoyant CO_2 in place for the foreseeable future".

The existences of all the needed technology is the main advantage of the application of CCS technology. As such, the need for new technological development is not a major limitation of its large-scale implementation, although improved efficiencies and low cost capture technologies are being researched. Also, past experience in the oil industry on the injection of both gasses and liquids into deep subsurface formations is an added advantage. The oil industry has extensive experience in CO_2 injection for enhanced oil recovery. However, the permanent injection of CO_2 into deep geological sites such as saline aquifers, unmineable coal seams or highly-depleted oil/gas fields is a different experience.

CO_2 storage is performed by pumping the fluid into a carefully selected reservoir. Such reservoirs are of different types: some contains undrinkable salt water such as deep saline aquifers and others are mostly depleted oil/gas fields. According to Arts et al. (2008) all reservoirs have "the same key geological features for storing CO_2 safely and are usually sandstone or limestone" having to meet the following basic criteria (Arts et al., 2008):

- "a layer of porous rock at the correct depth to hold the CO_2 (anywhere from 700 metres to 5,000 metres deep);
- sufficient capacity;
- and an impermeable layer of "cap" rock to seal the porous layer of rock underneath"

Amongst these available CO_2 storage options, highly-depleted oil/gas fields represent prime potential targets for the large-scale storage of captured CO_2 emitted from industrial sources and fossil-fuel power plants (Pale Blue Dot Energy, 2016). In the case of the UK for example, the Southern North Sea and the East Irish Sea depleted gas reservoirs provide 3.8 of the total 4-billion-tonne storage capacity required to meet UK's CO_2 reduction commitments for the period 2020-2050 (Hughes, 2009). As opposed to saline aquifers, "depleted gas fields are better characterised given the availability of geological data, such as pressure, porosity and permeability, derived from years of gas production, as well as seals that have successfully retained hydrocarbon gas for millions of years, and may offer a shorter route to practical implementation for early CCS projects" (Hughes, 2009; Sanchez Fernandez et al., 2016).

The UK has three large FEED study projects for offshore CO_2 storage at Hewett, Golden Eye and Endurance sites (ETI, 2016). However, CO_2 storage at full industrial scale has only been demonstrated at a small number of sites around the world: Sleipner, In Salah, Snøvit (Eiken et al., 2011), Ordos (Jiang et al., 2014), and the Quest project (Shell Canada Limited, 2017). In order to boost the confidence of both investors and the public and thus facilitate the full-scale deployment of CCS, it is of paramount importance to guarantee that the storage site will be of high-quality and operate in a safe manner. However given the low reservoir pressures along with the unique thermodynamic properties of CO_2, the start-up injection of CO_2 into depleted low pressure gas fields presents significant safety and operational challenges.

1.3 Statement of the problem and relevance of the study

The most cost effective way of transporting the captured CO_2 for subsequent storage is in its dense phase using high pressure pipelines (Brown et al., 2014; Mahgerefteh et al., 2012; Mahgerefteh et al., 2006). This means that the CO_2 arriving via the sub-sea pipeline to the injection well will typically be at pressures greater than 70 bara and temperature between 4 to 8 ^{o}C. Given the substantially lower pressure at the well head, the uncontrolled injection of the CO_2 will result in its rapid quasi-adiabatic expansion commonly known as "Joule-Thomson expansion" leading to temperatures as low as -70 ^{o}C (Oldenburg, 2007). In practice, this process could pose several risks, namely:

- blockage due to hydrate and ice formation following contact of the cold CO_2 with the interstitial water around the wellbore and the formation water in the perforations at the near well zone;
- thermal stress shocking of the wellbore casing steel due to the tube inner and outer temperature gradient leading to its fracture and escape of CO_2;
- over-pressurisation accompanied by CO_2 backflow into the injection system due to the violent evaporation of the superheated liquid CO_2 upon entry into the 'low pressure' wellbore.

As such developing appropriate start-up injection strategies for avoiding the above risks is of paramount importance. Key to this is the availability of reliable mathematical models for predicting the behaviour of the injected CO_2, in particular its variation of pressure and temperature along the well and ultimately at the point of entry into the depleted reservoir. The alternative is the heating of the CO_2 stream prior to injection which is highly costly given the significant volumes involved.

Figure 1.2 illustrates the main features and components of an injection well connecting the upstream CO_2 to the storage reservoir. As the CO_2 is injected into the formation, depending on the magnitude of the pressure difference between incoming fluid and the pressure at the top of the well, the CO_2 may undergo drastic expansion and the temperature drops. This could lead to serious risks such as thermal shocking of the steel wall and backflow due to the superheating of CO_2. In the event of sufficient amount of water being present, this may result in ice or hydrate formation, both leading to the risk of well blockage.

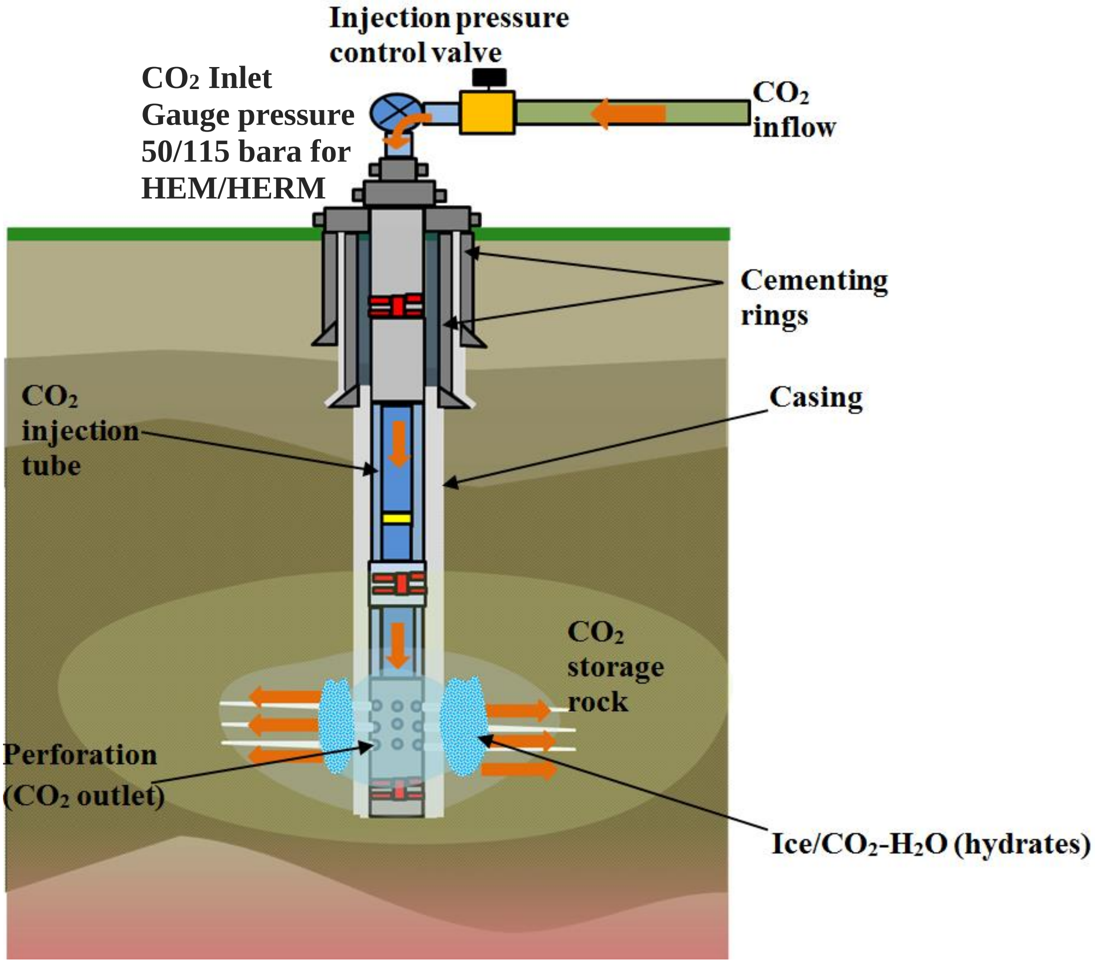

Injection pressure control valve
CO_2 Inlet Gauge pressure 50/115 bara for HEM/HERM
CO_2 inflow
Cementing rings
Casing
CO_2 injection tube
CO_2 storage rock
Perforation (CO_2 outlet)
Ice/CO_2-H_2O (hydrates)

Figure 1.2: Schematic CO_2 injection process diagram

Some work has already been devoted to the analysis of the injection of CO_2 from wells into underground reservoirs (see for example André ety al., 2007; Goodarzi et al., 2010; Nordbotten et al., 2005). However, as noted by Linga and Lund (2016) and Munkejord et al. (2016), the modelling of the transient behaviour of the CO_2 flowing inside the injection wells has received limited attention. Lu & Connell (2008) for example presented a steady-state model for the evaluation the non-isothermal flow of CO_2 and its mixtures in wells. Paterson et al. (2008) considered the CO_2 liquid-gas phase change in wells that are static, and also performed a simulation of the pressure and temperature profiles by neglecting changes in kinetic energy and assuming a quasi-steady state. Lindeberg (2011) proposed a model combining Bernoulli's equation for the pressure drop along the well and a simple heat transfer mechanism between the fluid and the surrounding rock. The model was applied to the Sleipner CO_2 injection well. Pan et al. (2011) presented analytical solutions for steady-state, compressible two-phase flow model through a wellbore under isothermal conditions using a drift flux model. Lu and Connell (2014) investigated a non-isothermal and unsteady wellbore flow model for multispecies mixtures. Ruan et al. (2013) and Jiang et al. (2014) studied the impact of CO_2 injection on the rock temperature in both axial and radial directions and also developed a 2-D radial numerical model to study the mechanisms of CO_2 temperature increase in the well tubing. Li et al. (2015) developed a model to account for the dynamics of CO_2 into injection wells subject to highly-transient operations, such as start-up and shut-in. Linga and Lund (2016) discussed a two-fluid model for vertical flow applied to CO_2 injection wells, predicting flow regimes along the well and computing friction and heat transfer accordingly. Li et al. (2017) presented "a unified model for wellbore flow and heat transfer in pure CO_2 injection for geological sequestration, EOR and fracturing operations".

However, despite the mathematical rigour of the more recent models, none are designed for handling the controlled 'ramping up' of the feed pressure and hence the mass flow rate at the point of injection into the well. In practice, this important conditioning step may be carried out using in-line pressure control valves to gradually expose the well to the full pipeline pressure in order to minimise the risks associated with the rapid decompression of the feed CO_2. Also, on top of the uncertainty and complexity involved in modelling the transient flow scenario during CO_2 injection, the CO_2 stream impurities on the other hand further complicate issues given their profound impact on the CO_2 phase equilibrium behaviour even if present is small quantities (Wang et al., 2015). Previous works (see for example Li et al. (2015) and Linga and Lung (2016)) have modelled the process with no consideration for the impact of CO_2 stream impurities.

In this thesis, the development and verification of a Homogeneous Equilibrium Mixture (HEM) model and a Homogenous Equilibrium Relaxation Mixture (HERM) model for simulating the transient flow phenomena taking place during the injection of CO_2 into highly depleted gas fields is presented. The HEM model considers the two-phase mixture being injected to be at homogeneous equilibrium, which means instantaneous interface mass, momentum and energy exchange. As such, the constituent gas and liquid phases are assumed to remain at the same pressure, temperature and velocity. The corresponding fluid flow can therefore be described using a single set of the mass, momentum and energy conservation equations. For fluid/wall friction and heat exchange, empirical correlations are adopted. The HERM on the other hand presents an additional equation which accounts for the thermodynamic non-equilibrium relaxation time in two-phase mixtures.

1.4 Aim and objectives of the study

The main aim of this study is to offer an environmentally sustainable solution option to the challenge of global temperature increase otherwise called “global warning” through CCS. This study is therefore focussed on modelling the transient flow behaviour of carbon dioxide (CO_2) during geological sequestration. The main objectives are to:

- Develop and verify transient flow models for the injection of CO_2 into depleted gas fields
- Demonstrate the usefulness of the models developed by applying data from a real CO_2 injection system as test cases.
- Employ the findings to predict and recommend optimum CO_2 injection strategies.

The thesis is divided into 7 chapters:

Chapter 2 presents a literature review covering the current challenges in the large-scale deployment of CCS and recent CO_2 injection related publications. The Joule-Thomson expansion effect which describes the flow of the expanding CO_2 into the injection well is reviewed in detail. This is followed by a detailed description of the general conservation equations for fluid dynamics, and the constituent relations for fluid/wall and fluid/fluid interface interactions, the Equation-of-State (EoS) for predicting the thermodynamics and transport properties and equilibrium data. The chapter concludes with a review of previous applications and performance evaluation of the Homogeneous Equilibrium Mixture (HEM) and Homogeneous Relaxation Mixture (HERM) models for simulating pressured pipeline ruptures.

Chapter 3 presents the Finite Volume Method numerical solution to the HEM flow model presented in chapter 2. The application software (CLAWPACK) is presented with the applied boundary conditions for the start-up injection CO_2 into depleted gas fields. The FVM numerical solution follows the flux-difference splitting scheme which solves a numerical inviscid flux function for a general system of conservation equation. The start-up injection of CO_2 at the top of the well is described by the constant mass flux subsonic inflow boundary condition for a one-dimensional hyperbolic system. The model results obtained show the impact of CO_2 start-up injection operations on wellhead pressure and temperature, and the consequent flow of the

CO_2 stream down the injection well. The HEM model is deployed to simulate the start-up injection process with focus on the CO_2 pressure, temperature and density behaviour at the top and bottom of the well.

In Chapter 4, the HEM flow model is employed to simulate sudden blowout from the injection well following an overpressure and backflow as a result of the rapid expansion and superheating of the dense phase CO_2. The following two case studies are considered:

- blowout prior to start-up injection, and
- blowout after 5 mins of start-up injection

A detailed account of the start-up and blowout boundary conditions for inflow or outflow at the wellhead and inflow or outflow at the bottom of the well is presented. The first case study compares pressure and temperature changes during start-up injection and blowout prior to injection whilst the second case study compares blowout prior to injection and blowout after 5 mins of injection. The boundary conditions and resulting pressure and temperature profiles for the wellhead and bottom of the well are presented.

In Chapter 5 the HEM flow model is extended to account for the injection flow rates ramping-up times. This is driven by the need for the development of optimal injection strategies and best-practice guidelines for minimising of the risks associated with the CO_2 injection process. The appropriate time-dependent boundary conditions are employed to simulate three injection flow rate ramping up durations corresponding to slow, medium and fast. Finally, a case study investigating the impact of varying range of impurities expected from pre-combustion, post-combustion and oxy-fuel the main CO_2 capture technologies used with fossil-fuelled power plants on the pressure and temperature profiles at the top and bottom of the well during CO_2 injection is investigated.

Chapter 6 presents the development and verification of a Homogeneous Equilibrium Relaxation Model (HERM), where mass, momentum, and energy conservation equations are considered for two-phase flow. It accounts for phase and flow dependent fluid/wall friction and heat transfer, variable well cross sectional area as well as deviation of the well from the vertical. The HERM contrary to the HEM presents an additional equation which accounts for the thermodynamic non-equilibrium relaxation time in two-phase multi-component mixtures. The model developed is also used to perform a detailed sensitivity analysis of the most important parameters affecting the CO_2 flow behaviour, including the wellbore diameter variations, well inclination, upstream temperature, pressure and time variant mass flow rate; the latter being representative of the feed pressure ramping up process.

Chapter 7 provides a summary of the main conclusions of this thesis followed by recommendations for future work.

Chapter 2: Literature Review

2.1 Introduction

In relation to chapter 1, this chapter presents a literature review covering the current challenges in the large-scale deployment of CCS and detailed review of recent CO_2 injection related publications. The Joule-Thomson expansion effect which describes the flow of the expanding CO_2 into the injection well is reviewed in detail. This is followed by a detailed description of the general conservation equations for fluid dynamics, pipe flows and the constituent relations for fluid/wall and fluid/fluid interface interactions, the equation of state (EoS) for predicting the thermodynamics and transport properties and equilibrium data of the overall formulations. The chapter concludes with a review of previous applications and performance evaluation of the Homogeneous Equilibrium Mixture (HEM) and Homogeneous Relaxation Mixture (HERM) models for simulating pressured pipeline ruptures.

2.1.1 CO_2 storage

After CO_2 is captured and compressed, it is then transported to a selected geological reservoir for long-term storage. As can be seen in Figure 2.1, the geological reservoir for storing CO_2 may be a depleted oil or gas field (or one that is undergoing depletion), a deep saline aquifer, or unmineable coal seams.

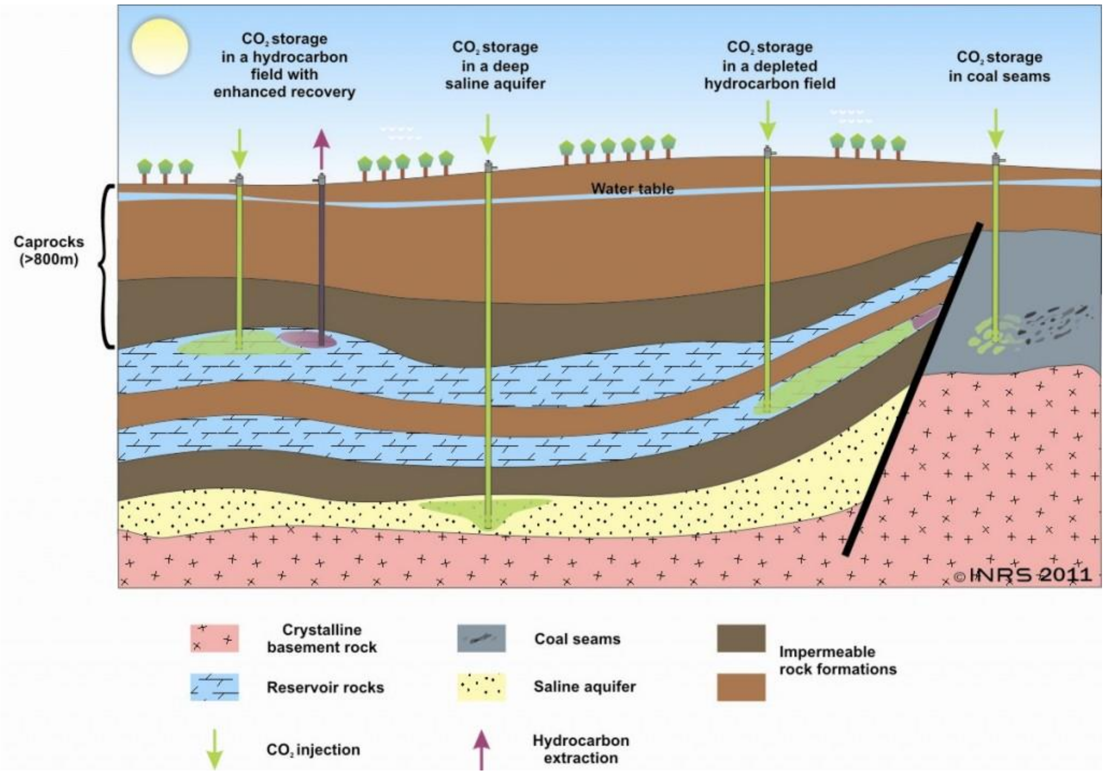

CO₂ storage in a hydrocarbon field with enhanced recovery
CO₂ storage in a deep saline aquifer
CO₂ storage in a depleted hydrocarbon field
CO₂ storage in coal seams
Water table
Caprocks (>800m)
©INRS 2011
Crystalline basement rock
Coal seams
Impermeable rock formations
Reservoir rocks
Saline aquifer
CO₂ injection
Hydrocarbon extraction

Figure 2.1: The different types of CO_2 storage reservoirs (IEA, 2011)

At the injection point CO_2 is delivered to the geological reservoir at pressures greater than 74 baras and depths around 800 to 5,000 metres. The CO_2 at pressure above 74 baras and temperature above 31.1 ^oC is in a “supercritical” state; thus, it is as dense as a liquid and viscos as a gas. At depths around 800 metres and more, one tonne of CO_2 is expected to occupy a volume of 0.1 to 0.2 cubic metres, this allows an enormous quantities of CO_2 can be stored in a relatively restricted space.

It is important to realise that these stable natural geological reservoirs intended for the injection of CO_2 have existed for millions of years (IEA, 2011). However, in order to determine their suitability for geological CO_2 storage, it is important to analyse whether they will display the same characteristics devoid of the injected CO_2 as CO_2-containing natural reservoirs, and if they are suitable for the intended purpose.

The geological storage of CO_2 represents “a new method for reducing greenhouse gas (GHG) emissions” (IEA, 2011). But injection of CO_2 into oil or gas fields is not in itself not new since Enhanced Oil Recovery (EOR) that uses CO_2 injection to increase the yield from hydrocarbon fields has been widely practiced since the 1970s (IEA, 2011). However, injection of CO_2 into a highly-depleted low pressure oil or gas field presents a new challenge different from the EOR practice.

2.1.2 CO_2 storage reservoirs

Geological reservoirs consist of "porous rocks (sandstone and some carbonate rocks) containing pores that can be filled with CO_2" (Ehrenberg & Nadeau, 2005). The CO_2 is stopped from escaping to the surface by a impermeable and thick cap-rock (fine-grained or rock salt) which is located above the reservoir (Raza et al., 2016). The cap-rock helps to keep the injected CO_2 at a suitable depth and also essential to keeping the reservoir sealed.

2.1.2.1 Geological storage of CO_2 in highly-depleted oil and natural gas fields or those undergoing depletion

Injection of CO_2 into geological storage of oil and gas fields may be for enhanced oil recovery purpose or permanent storage. In EOR as illustrated in Figure 2.1 (a), CO_2 is injected into an oil or gas field to increase the pressure in the reservoir and facilitate oil or gas extraction (Juho, 2015). The emptied pores that once contained hydrocarbons are then filled with the injected CO_2. As illustrated in Figure 2.1 (b), the application of CO_2 injection into depleted gas field as an environmental intervention measure is a relatively new practice thus, requires further research and development.

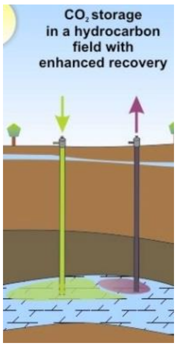


Figure 2.1 (a)

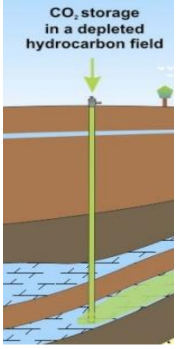


Figure 2.1 (b)

Amongst the different types of reservoirs, depleted oil and gas fields are better characterised given the availability of geological data, such as pressure, porosity and permeability, derived from years of gas production, as well as "seals that have successfully retained hydrocarbon gas for millions of years, and may offer a shorter route to practical implementation for early CCS projects" (Hughes, 2009; Sanchez Fernandez et al., 2016)

2.1.2.2 Geological storage of CO_2 in deep saline aquifers

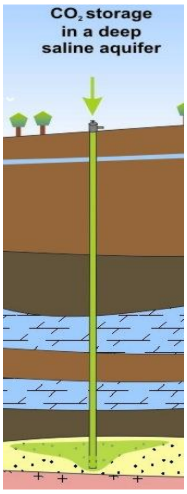


Figure 2.1 (c)

As illustrated in Figure 2.1 (c), "deep saline aquifers are rock formations that are porous, permeable and saturated with salt water (much saltier than seawater) that is not fit for consumption" (IEA, 2011). The impermeable layer surrounding the aquifer allows the injected CO_2 into these formations to gradually migrates but restrict the plume movement to the lateral and vertical directions.

The "deep saline aquifers represent the most promising potential for geological CO_2 storage amongst these reservoirs due to the large capacity potential available for CO_2 to be stored" (IEA, 2011). However, the possibility of CO_2 leakage from deep saline aquifers is a major issue affecting their deployment for large-scale application.

2.1.2.3 CO_2 storage in unmineable coal seams

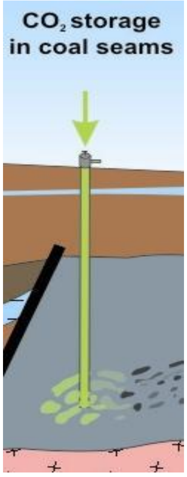


Figure 2.1 (d)

Geologically, the storage of CO_2 in "coal seams that are not economically mineable due to their depth can be based on the recovery of methane from coal seams (ECBM: Enhanced Coal Bed Methane)" (IEA, 2011). As shown in Figure 2.1 (d), the idea is to inject CO_2 into the pores of the coal seam to displace the methane gas present thus, allowing the methane to be recovered and leaving the CO_2 trapped in the coal's pores.

Table 2.1 provides a summary of the advantages and disadvantages of the above three types of potential reservoirs for CO_2 geological storage.

Table 2.1: Potential storage reservoir with their merits and demerits

Storage reservoir type	Advantages	Disadvantages
Oil/gas fields	They have cap-rocks and traps that are leak-proof to non-reactive gasses.	They are generally far from large CO_2 emission sites. Storage capacity may be

	Very well-known and characterised reservoir systems. They are economically proven to be viable via EOR.	mostly limited
Deep saline aquifers	They are widespread with great potential for CO_2 storage. Storage site can be easily found for emitters. Water in saline aquifers is not potable.	Very few have been characterised to date. There is potential threat of CO_2 leakage after injection
Unmineable coal seams	Very near CO_2 emission sites. They are economically proven to be viable via methane recovery	Difficulties with injection due to coal's low permeability. Very limited storage capacity

2.2 Issues associated with CCS

One of the key advantages of CCS is the existence of all of the required technology for large-scale implementation (Nordbotten & Celia, 2011). Since the oil industry and the waste disposal industries have extensive experience in injection of both gases and liquids into deep subsurface formations, as such their wide spread application is not limited by the need for new technological developments (Nordbotten & Celia, 2011). Despite the previous experience, CCS will require an improvement in process efficiency, low cost capture techniques, and new technologies for long-term monitoring of injection system. Thus, the major challenges associated with large-scale implementation of CCS are classified into three categories (Nordbotten & Celia, 2011):

1) “Regulatory challenges

- The lack of regulatory framework within which large-scale injection operations would be permitted
- The lack of international agreements that would lead to effective global solutions.

2) Economic challenges

- The lack of economic system where the cost of emitting CO_2 is internalised within an individual operator such as a power plant.

3) Technical challenges

- The uncertainties surrounding the safety of personnel, equipment and injection start-up and shut down.
- The lack of understanding of the subsurface system and ability to predict the fate of the injected CO_2.

- The uncertainties surrounding the possibility of leakage of injected CO_2 and the concern for significant environmental damage especially on underground water".

Notably, finding a solution to the technical challenges will pave way for regulatory framework and economic credit systems to be implemented. Having a comprehensive understanding of a safe and reliable injection process will provide investors with confidence and facilitate international agreements that would lead to effective global solutions. This study addresses key technical challenge of the injection process safety concerns at start-up injection by modelling the transient changes during the process.

Considering both regulations and economic credit systems, the key question focuses on the safety of injecting CO_2, and on a general note, what are the likely effects of large-scale injection of CO_2 on the subsurface system. If there is a significant temperature drop during start-up injection, the CO_2 may fracture the formation casing due to thermal shock and block the injector due to hydrates or ice formation with the interstitial water molecules. Then "the operation may be viewed as a failure from an environmental, regulatory, and economic-credit perspective". Nordbotten & Celia (2011) considered a "broader analysis of the overall system and suggested that leakage of either the injected CO_2, or brine that is displaced by the injected CO_2, is another issue associated with geological storage of CO_2. They therefore proposed a comprehensive analysis of the overall subsurface system, including the movement and ultimate fate of the injected CO_2 and other subsurface fluids affected by the injection" Nordbotten & Celia (2011).

The CO_2 storage reservoirs are most likely under significant fluid pressure as a result of the weight of the fluid in the overlying geology (Lu & Connell, 2014a). This means that an exceeding greater CO_2 pressure to that of the reservoir will be required at the bottom of the injection well. As such, compressing the CO_2 to higher pressures before injecting it into the reservoir becomes necessary. At such high pressures, CO_2 becomes dense, and may contribute significant pressure increase to the bottom-hole of the well. Notably, the CO_2 may undergo phase transitions as it exchanges heat with the surrounding geology during the flow into the well. Therefore, there are highly variable flow scenarios in the well leading to transient pressure and temperature profiles along the well depth (Lu & Connell, 2014). Again, there is the challenge of highly depleted oil and gas reservoirs at very low pressure in which the wellhead pressure and temperature are much lower under hydrostatic conditions. The behaviour of such low pressure injection well becomes complex during CO_2 injection especially at well start-up and shut-in times.

This study stresses that assessing and managing the flow control aspects of CO_2 injection, especially when the reservoirs are at very low pressure poses a major technical challenge (Hughes, 2009). This may include; a control device with high-tech flow at the base of the tubing to allow controlled expansion of the dense phase CO_2. Also, conventional software applied in the design and modelling of hydrocarbon flow may not be sufficiently rigorous enough for the modelling of complex fluid like CO_2 with peculiar thermodynamic behaviour. Understanding and hence mitigating the Joule-Thomson effects and hence avoiding the possible formation of ice and gas hydrates requires the development of an appropriate mathematical model that accurately captures CO_2 flow system including the modelling of the thermodynamic effects to realise the potential that depleted gas field storage offers (Hughes, 2009 and Lu and Connell, 2014b).

2.3 Previous works on CO_2 injection

Investigation of the flow of CO_2 in wellbores was first studied for oil and gas problems, specifically for enhanced oil recovery (Lu and Connell, 2014b; Cronshaw and Bolling, 1982). The authors utilised an approximate thermodynamic treatment based on a simplified flow model (Lu and Connell, 2014b). The wellbores CO_2 flow study presented by Cronshaw and Bolling (1982) was a simplified scheme that deal with the problem of quasi-steady flows. Recently, Lu and Connell, (2008); Battistelli et al., (2011); Wiese et al., (2010); Lindeberg, (2011) and Sasaki et al., (2009) all studied the geological storage phenomenon of pure CO_2 flow in wellbore employing various sophisticated numerical schemes and thermodynamic models. However, they are mostly concerned with the pattern of steady-state flow which partly or fully neglects the transient effects during start-up injection. While the steady-state flow model is considered as a good approximation for the behaviour of the flow in wellbore, it becomes inadequate when significant unsteady-state behaviour such as ramp-up injection is performed (Lu and Connell, 2014b; Michael et al., 2011).

Practically in some situations (e.g. Michael et al., 2011), the flow maybe highly transient especially during start-up injection, and thus assuming a steady-state will not represent the system correctly. “A typical example is the ramp-up injection during start-up, in which an estimated characteristic time-scale may likely be in hours or minutes. In such instances, the variation of the CO_2 injection pressure and temperature with time and the rate of injection rate may be rapid, leading to transient flow scenarios within the well" (Lu and Connell, 2014b). Oldenburg and Pan, (2013); Pan and Oldenburg, (2014) also developed a wellbore-reservoir flow model recently for transient operations, focussed on the simulation of CO_2 injection dynamics and wellbores leakage scenarios. However, their model fails to take into account possible phase transition and thermal effects during the injection flow process. When performing the simulation of the flow model, a challenging problem in the flow equation may be the inclusion of the thermal effects and transient behaviour. The conditions of numerical stability may be greatly restricted by the non-linearity of the Equation-of-state and the phase transition behaviour for real gases, along with the nonlinear partial differential flow equations. Some recent models were developed to study the CO_2 flow behaviour during instances such as blowout or rapid depressurisation (see for example de Koeijer et al., 2009; Michael et al., 2011; Munkejord et al., 2010). These models were however not aimed at the transient flow behaviour that may occur during ramp-up injection, but at unsteady flow-over extremely short times, such as a second or a fraction of second (Lu & Connell, 2014b). These models do not account for the characteristics propagation of wave and as such may not be reliable for longer duration practical transient problems.

Lu & Connell, (2014b) proposed a "novel modelling scheme to deal with transient CO_2 wellbore hydraulics during geological storage". Their model represents the major wellbore flow characteristics involved in the flow process, and the computational efficiency of the scheme. They modelled the rigorous flows which follow the thermodynamic principle in particular for "the isothermal flash process both by stability analyses (e.g. Baker et al., 1982; Harding & Floudas, 2000; Michelsen, 1982) and phase split calculations (e.g. Ammar & Renon, 1987; Henderson et al., 2001; Michelsen, 1982)". During two-phase fluid flow, the equilibrium of the relevant "gas and liquid chemical potential is also rigorously followed for each phase" (Lu & Connell, 2014b). The above CO_2 flow process is considered based on the relevant wellbore flow equations. Their model was tested for numerical stability with a field application for a trial CO_2 injection case study into a "560-m-deep coal seam for enhanced coalbed methane recovery (ECBM)" (Connell et al., 2011). Lu and Connell (2014b) extended the model test based on the 560-m-deep coal seam field trial for the injection of pure CO_2. In the field test, the CO_2 was delivered periodically by a tanker truck in liquid form, while the injection was also performed periodically with the well shut-in breaks between each injection (Lu & Connell, 2014b). During each period of continuous injection, "the pressure and temperature of the injecting fluid at the well head may vary relatively slowly over time, the injection rates can change quite dramatically, leading to highly transient flows" (Lu & Connell, 2014b).

Afanasyev (2013) proposed a "new compositional modelling approach for sub- and supercritical three-phase flows of water, liquid CO_2 and gaseous CO_2". The "new approach is based on the calculation of the thermodynamic potential of the mixture as a function of pressure, total enthalpy and mixture composition and storing it values as a spline table, which is then used for the hydrodynamic simulation" (Afanasyev, 2013). A "three-parametric generalisation of the Peng-Robinson Equation of state is used to fit the experimental data on CO_2-H_2O mixture properties" (Afanasyev, 2013). The temperature variations in the reservoir were analysed based on the evaporation of liquid CO_2 and the dissolution of CO_2 in water. The interplay of these processes results in a complicated non-monotonic temperature distribution (Paterson et al., 2008a). The CO_2 fluid temperature with respect to the reservoir temperature before injection can either decrease or increase, when considering varying distances from the point of injection. The evaporation of the liquid CO_2 is considered as the main phenomenon responsible for the considerable temperature decline (otherwise known as Joule-Thomson cooling effect) around the CO_2 injection point.

There are still conflicting results in the literature on "the estimation of pressure build-up, the resulting number of injection wells required for large-scale CO_2 geological storage and storage efficiency despite the advanced understanding of subsurface flow processes and development of modelling tools" (Michael et al., 2011 and Wildgust et al., 2010). For these issues, "there seems to be inadequate conclusion, particularly with respect to the large CO_2 injection rates (> 10 Mt/year) anticipated for industrial scale CCS operations" (Michael et al., 2011). The results of the CO_2 injection studies have been limited to more or less generic analytical and numerical modelling exercises due to the unavailability of CO_2 injection and storage experimental data. Due to the relative permeability or residual saturation being uncertainties in model parameters, there is need testing the model results against field data in order to achieve conclusions that are closely applicable beyond generic studies.

Li et al. (2015) analysed the "influence of various parameters on transient CO_2 operations based on the design of wells in the Goldeneye CCS Project in the UK (Peterhead CCS Project)". The CO_2 flow rate, injection pressure, and temperature for the Peterhead CCS project and some assumptions formed the basis for selection of the simulation data. "A 1D+1D model using the OLGA (OLGA 7 User manual, 2010) software for wellbore dynamics, in which the mass, momentum and energy conversation equations of fluid in the tubing were 1D along the direction of flow and heat transfer between fluid and surrounding rock was calculated radially (+1D), was developed to predict the decrease in the temperature of CO_2 and of every well component, as well as CO_2 phase behaviour in the wellbore during well transient operations" used by Li et al., (2015).

Li et al. (2015) performed their numerical simulations "using the commercial dynamic multiphase model OLGA 7.0.0. OLGA software is widely utilised in the oil and gas industry to simulate various flow processes using a finite-element method for solving coupled continuity, momentum, and energy equations (OLGA 7 user manual, 2010)". The governing conservation equations were those outlined by (Bendiksen et al., 1991). Heat transfer between the well components and the surrounding rock was considered to be by conduction only in the radial direction. The radial transient heat conduction equation neglecting the axial heat conduction is given as (Li et al, 2015);

$$\rho_W c_{pW} \frac{\partial T_W}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(k_W r \frac{\partial T_W}{\partial t}\right) \qquad (2.1)$$

where the subscript, *W* represents wall layer, c_p is specific heat capacity, *T* is temperature, *r* is a radial coordinate, and *k* is heat conductivity.

Furthermore, the following quadratic inflow equation describing the pressure difference between the bottom-hole and the reservoir was employed within the OLGA Well Module (Li et al., 2015);

$$A + B \times M + C \times M^2 = P_{BHF}^2 - P_{res}^2 \qquad (2.2)$$

where, "*A* is the minimum pressure required for flow to start from the well into the reservoir, *B* and *C* are constants, *M* is mass flow rate at bottom-hole, P_{BHF} is bottom-hole flowing pressure, P_{res} is reservoir static pressure (17.2 MPa in the model). The constants, *A*, *B* and *C*, were 0, 1.3478 × 1012 $Pa^2 s/kg$ and 2.1592 × 1010 $Pa^2\ s^2/kg^2$ respectively in the model" (Li et al., 2015).

As opposed to Peng-Robinson EoS used in the study, Li et al. (2015) employed the Span & Wagner EoS (Span & Wagner, 1996) for the calculation of the density and the specific heat of CO_2 and the viscosity and thermal conductivity of CO_2 are calculated using the K.S.P EoS (Pedersen et al., 1989).

An offshore injection well as depicted in Figure 2.2 was simulated "based on the reservoir pressure, surrounding geological conditions, and the configuration of the injection well in the Goldeneye CCS project" (Li et al., 2015). In the simulation, "CO_2 was injected into the wellbore at 4 ^oC with an initially constant mass flow rate of 38 kg/s. During transient operations, the wellhead mass flow rate varied with time, resulting in variation of wellhead pressure, wellhead temperature, downhole and upward pressure propagation, and axial and radial temperature propagation. The total depth of the injection well was 2500 m and the reservoir static pressure was estimated as 17.2 MPa" by Li et al., (2015).

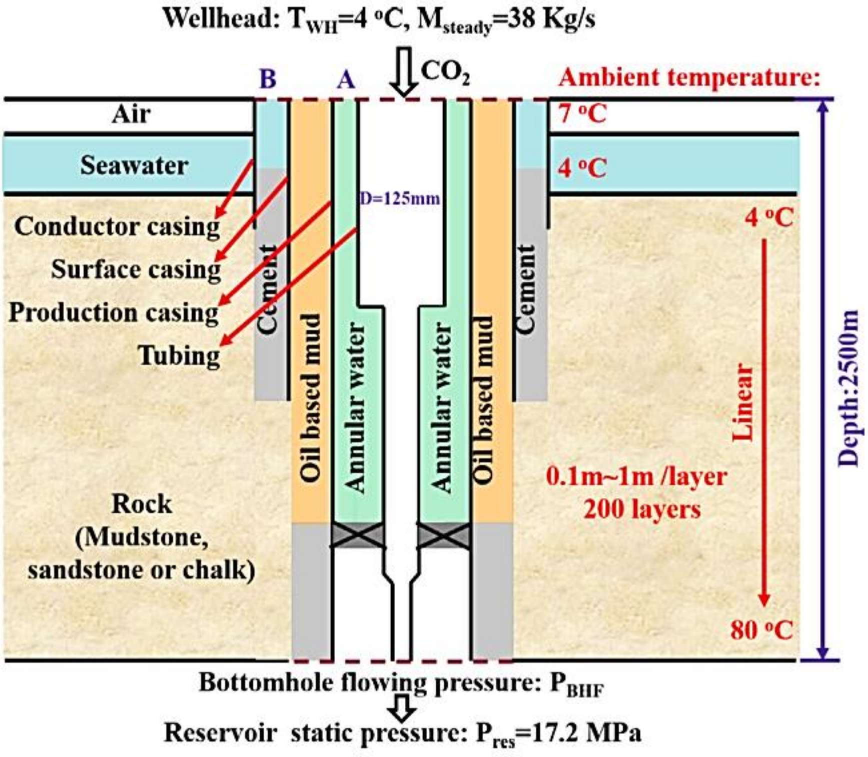


Figure 2.2: Schematic diagram of the CO_2 injection well (Li et al, 2015)

Cases of well shut-in (sudden termination of injection) and start-up injection were considered in the results presented in their work. However, greater attention was given to well shut-in with less attention on start-up injection. Conditions of the Goldeneye CCS Project corresponding to "injection rate and injection temperature of 38 kg/s of 4^{o}C respectively were selected to investigate the changes in wellhead pressure and temperature and assess the influence of shut-in time" (Li et al., 2015).

The following conclusions were drawn from the study (Li et al, 2015):

- “For the case of well shut-in operations, larger injection flow rates and larger injection temperatures led to larger wellhead temperature drops for comparatively similar injection pressures and shut-in periods.

- For the case of well start-up operation, the decrease in wellhead pressure was more pronounced than in well shut-in due to the increase of frictional pressure loss and increase of the bottom-hole pressure”.

Thus, the study recommended an optimal shut-in period of 10 mins at which the CO_2 temperature and the tubing wall temperatures never dropped below −14.1 ^oC and −3 ^oC respectively and a rapid well start-up to avoid very low temperatures that could impact well integrity. However, due to the unavailability of CO_2 injection experimental data, the simulation results could not be validated.

Linga et al. (2016) proposed a more sophisticated model for CO_2 injection. Here the authors described the flow “by a physically consistent two-fluid model, with the Span-Wagner reference Equation-of-State (Span & Wagner, 1996) to describe the thermodynamics of CO_2. Friction and heat transfer in the flow are modelled specifically for each flow regime, such as bubbly, annular and mist flow. The flow model is coupled to a model for heat conduction through the various layers of the well, such as tubing, packer fluid, casing, cement and rock. The performance of the model is demonstrated by using it to simulate transient well operations, in particular critical incidents such as sudden shut-in and blowout”. Linga et al. (2016) also applied their model to simulate sudden blowout and shut-in cases. They employed well conditions utilised in the Sleipner project where the CO_2 injection well depth, reservoir pressure and temperature and geothermal gradient are known (Linga & Lund, 2016; Thu, 2013). The result obtained for the blowout case predicted temperatures well above dry ice formation condition in the well. The assumption that outflow from the reservoir was at the reservoir temperature led to some steady-state temperature profile at the bottom of the well. However, such assumption may have disregarded any possible Joule-Thomson effect that might have occurred in the reservoir. For the shut-in case on the other hand, “they predicted a water hammer effect resulting from simultaneously closing well-head and bottom-hole valves when there is a steady downward flow” (Linga & Lund, 2016).

Acevedo & Chopra, (2017) conducted a study on transients and start-up injection for the Goldeneye injection well. They studied the influence of phase behaviour in the well design of CO_2 injectors. The study shows that “during transient operations (closing-in and re-starting injection operations), a temperature drop is observed at the top of the well for a short period of time. This is due to the reduction of friction caused by a lower injection rate. The duration of these operations dictates the extent to which the various well elements are affected. The sequence of steady state injection, closing-in operation (30 minutes), closed-in time (10minutes), and starting-up operation (30minutes) was simulated for the low reservoir pressure case using OLGA" (Acevedo & Chopra, 2017).

2.4 Adiabatic Joule-Thomson cooling effect

The process of injecting CO_2 with higher pressure into a well with lower pressure at the wellhead is similar to the expansion of a real gas which is released from a high pressure region to a low pressure region. Such process is characterised by an inevitable cooling of the gas upon entering the lower pressure domain and it is controlled by Joule-Thomson cooling effect. It is therefore important to fully understand its concept. Looking at the CO_2 phase diagram in Figure 2.3 (Oldenburg, 2007) the supercritical region is described as the most appropriate for geological storage.

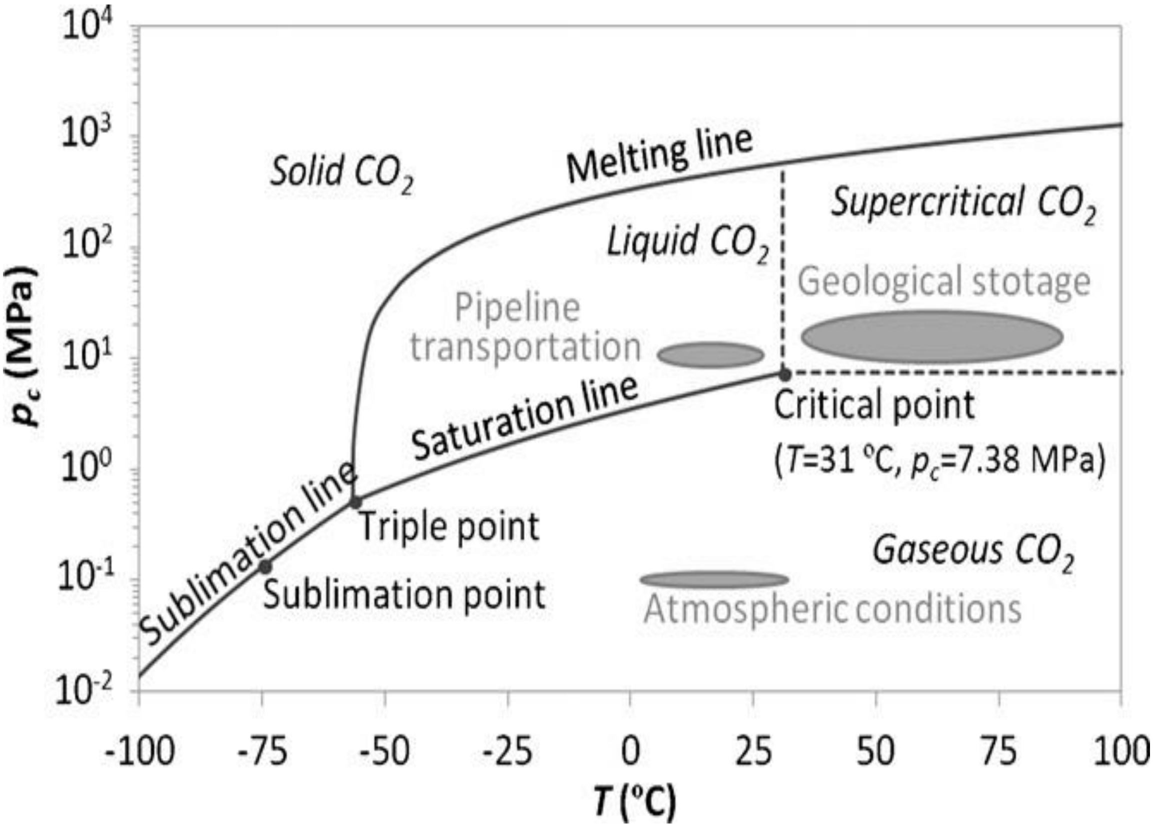


Figure 2.3: CO_2 phase diagram showing various regions (Oldenburg, 2007)

The "decrease in temperature that takes place when a real gas, such as CO_2, expands from high pressure to low pressure during an isenthalpic process is called Joule-Thomson cooling" (Li et al., 2015). This phenomenon accompanies "CO_2 injection from the injection well to the reservoir, especially in the two of the most common situations, a decrease in wellhead temperature during well start-up operations and a decrease in temperature at the inlet of the reservoir in cases of CO_2 storage in depleted gas fields with low pressure" (Li et al., 2015; Mathias et al., 2010). The "Joule-Thomson coefficient in this process is defined by equation (2.3), whereby this coefficient is an inherent thermodynamic property of real gases and can be derived using the thermodynamic relationship using the basic thermodynamic parameters of *P*, *V*, and *T* and basic property C_P as shown in the second part of equation (2.3)" by Li et al., (2015).

$$\mu_{JT} = \left(\frac{\partial T}{\partial P}\right)_H = \frac{T(\partial V/\partial T)_P - V}{C_P} \qquad (2.3)$$

Where μ_{JT} is Joule Thomson coefficient, C_P is heat capacity at constant pressure, H is enthalpy, P is pressure, T is temperature and V is volume of the fluid.

The enthalpy change of CO_2 during injection in the wellbore at a certain time can be described as follows:

$$H_{out} - H_{in} = \frac{Q}{M} + g(h_{in} - h_{out}) \qquad (2.4)$$

where "*out*" denotes the outer boundary of the section in the downward flow direction, "*in*" denotes the inner boundary of the section, and *M* denotes the mass flow rate, Q denotes heat transfer coefficient and g is gravity. As shown in equation (2.4), changes in CO_2 enthalpy are related to heat exchange with the surrounding rocks and potential energy loss. Consequently, "CO_2 expands from a high pressure to a low pressure at the wellhead under isenthalpic conditions in adiabatic tubing during transient operations" (Li et al., 2015). This mechanism determines the temperature drop of CO_2. The adiabatic change in temperature can be calculated from equation (2.5):

$$\Delta T_{adiabatic} = \int \left[\mu_{JT}(P,T)dP\right] \qquad (2.5)$$

2.5 Background Theory

2.5.1 Development of transient flow model

The development of a transient flow model for CO_2 geological sequestration comprises three major steps:

1. Formulating the basic governing equations of the flow, thermodynamics, flow-dependent closure equations and the setting up pf the initial and boundary conditions.

2. Selecting and implementing an efficient and accurate method that resolves or simplifies the model equations (such as method of characteristics or finite volume methods).

3. In the case where experimental data is available, validating the model.

4. In the absence of experimental data, testing the model's efficacy based on its application to a real system and performing sensitivity parametric studies.

Figure 2.4 shows a schematic flow diagram of an injection tube. A control volume of a section of the tube is considered for analysis and derivation of model governing equations.

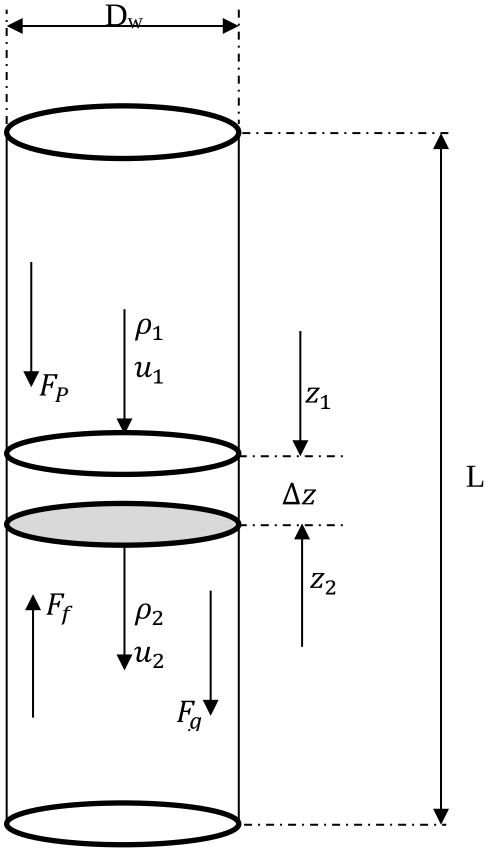


Figure 2.4: Schematic representation of a control volume within a vertical pipe and the forces acting on it

where F_P, F_f, F_g, ρ and u are pressure force, frictional force, gravitational force, fluid density and velocity respectively. L, D_w and Δz are well depth, diameter and differential control volume.

In order to derive the conservation equations some important factors should be considered to represent the transient behaviour of the flow as follows:

- The fluid flow: the flow regimes may include fully dispersed, stratified, bubbly, and annular or churn two-phase flow. However, when two phase flow modelling is required where each phase flows at different velocity, the conservation laws should then be applied separately to each phase. The mass and heat exchanges between the phases should also be considered.

- Thermodynamic state of the fluid, indicating if the fluid is vapour, liquid or a two-phase mixture and the pertinent physical properties of the fluid.

- Dissipative fluid parameters such as frictional losses and heat losses: Dissipative fluid/wall interactions represent the effects caused by heat transfer to and from the surrounding formation and friction losses to the pipe wall.

- For a flow governed by gravitational effects the pipeline angle of inclination should be considered. This study considers a vertical injection well, hence pipe inclination is not accounted for.

2.5.2 The Homogeneous Equilibrium Mixture (HEM) model

In order to provide complete formulation for modelling fluid flow, Navier-Stokes Equations are used. These equations consider the variation of the fluid properties with respect to three spatial dimensions and time. Three dimensional modelling is not required only in engineering applications but also for solving most complex systems that are computationally very expensive. Therefore, by employing reasonable assumptions, one can still maintain the accuracy of the model and the relevant solution. To this end, the following simplifying assumptions are applied:

- One-dimensional flow in the pipe

- Homogeneous equilibrium fluid flow

- Negligible fluid structure interaction through vibrations
- Constant cross section area of pipe

The assumption of homogeneous equilibrium flow requires that all phases are at mechanical and thermal equilibrium (i.e. phases are flowing with same velocity and temperature) hence the three conservation equations may be applied for the fluid mixture. Although, in practice usually the vapour phase travels faster than the liquid phase, the HEM model has proven to provide acceptable accuracy in many practical applications (see (Bilicki & Kestin, 1990; Brown et al, 2015; Denton, 2009).

The HEM model assumes instantaneous mass, momentum and energy exchange between fluid phases. As a result, all fluid phases share the same velocity, temperature and pressure. The mass, momentum, and energy conservation equations for a homogeneous two-phase flow model in a pipeline are rewritten in a differential form for numerical solution scheme (Zucrow and Hoffman, 1975) respectively as:

Mass conservation

$$\frac{\partial \rho_m}{\partial t} + \frac{\partial \rho_m u_m}{\partial z} = 0 \tag{2.6}$$

Momentum conservation

$$\frac{\partial \rho_m u_m}{\partial t} + \frac{\partial}{\partial z}(\rho_m u_m^2) + \frac{\partial P}{\partial z} = \frac{f_w \rho_m u_m^2}{D_w} - \rho_m g \sin\theta \tag{2.7}$$

Energy conservation

$$\frac{\partial E}{\partial t} + \frac{\partial u_m(E+P)}{\partial z} = \frac{f_w \rho_m u_m^3}{D_w} - \rho_m u_m g \sin\theta + \frac{Q}{\pi r_w^2} \tag{2.8}$$

where ρ_m, u_m, P, D_w, r_w, f_w, g, θ are mixture density, mixture velocity, pressure, wellbore diameter, wellbore inner radius, wall friction coefficient, gravity and inclination angle of the wall respectively. The subscripts m and w denote mixture and pipe wall. It should be noted that the solutions of equations 2.6 to 2.8 represent the time-averaged flow (over finite, Δt) within the control volume of interest.

The wall friction between the fluid and pipe wall is described by the friction factor for pipes with rough walls f_w, as defined by Chen's correlation (Chen, 1979):

$$\frac{1}{\sqrt{f_w}} = -2\log\left[\frac{\varepsilon/D_w}{3.7065} - \frac{5.0452}{Re}\log\left(\frac{1}{2.8257}\left(\frac{\varepsilon}{D_w}\right)^{1.1098} + \frac{5.8506}{Re^{0.8981}}\right)\right]$$

Q is the heat exchange between the fluid and its surrounding wall and formation.

$$Q = \frac{4}{D_{eq}}h_f\left(T_w - T_f\right) \qquad (2.9)$$

where the wellbore equivalent diameter is given as $D_{eq} = \sqrt{\frac{4A}{\pi}}$

E in Equation (2.8) represents the total mixture energy defined as:

$$E = \rho_m(e + \frac{1}{2}u_m^2) \qquad (2.10)$$

Based on the homogeneous equilibrium flow assumption e is the mixture specific internal energy:$e = x_{eq}e_v + (1 - x_{eq})e_l$ (2.11)

and, x_{eq} is the equilibrium vapour quality. The mixture density, ρ_m is given by:

$$\frac{1}{\rho_m} = \frac{x_{eq}}{\rho_v} + \frac{(1-x_{eq})}{\rho_l} \qquad (2.12)$$

The subscripts *v* and *l* refer to the vapour and liquid phase respectively which is considered as a mixture in this case.

2.5.3 Two-phase flow modelling and constitutive relations

During CO_2 injection, two-phase flow may occur mainly as a result of evaporation of the liquid phase, or expansion of the compressed vapour phase. This may lead to complex interface interactions that may completely change the flow characteristics. For example, with very rapid rates of interphase mass, momentum and energy exchange, the constituent fluid phases are at thermodynamic and mechanical equilibrium (homogeneous equilibrium), that is, no differences can be observed between the temperatures, pressures or velocities of the phases. When interphase momentum exchange rate is not sufficiently high, mechanical equilibrium cannot be retained, and heterogeneous flow (e.g. slug flow, annular flow and stratified flow) occurs. In the case of delayed interface mass and energy exchange, thermodynamic non-equilibrium is respectively manifested in fluid metastable states (e.g. superheated liquid phase) and distinctive phasic temperatures (thermal stratification).

The accurate simulation of the above phenomena has been the focus of many studies, involving different modelling approaches for dealing with the interface interactions. For instance Arzanfudi & Al-khoury (2015) modified the momentum conservation equation (2.7) to include the phase slip parameter, γ as:

$$\frac{\partial \rho_m u_m}{\partial t} + \frac{\partial}{\partial z}(\rho_m u_m^2 + \gamma) + \frac{\partial P}{\partial z} = -\frac{f_w \rho_m u_m^2}{D_p} - \rho_m g \sin\theta \tag{2.13}$$

The slip parameter, γ is defined as (Arzanfudi & Al-khoury, 2015)

$$\gamma(P, u_m, \rho_m) = \frac{f_v}{1-f_v}\frac{\rho_v \rho_l \rho_m}{\rho_m^{*2}}[(C_0 - 1)u_m + u_d]^2 \tag{2.14}$$

in which f_v is the vapour volume fraction, u_d is the drift velocity, and

$$\rho_m^* = f_v C_0 \rho_v + (1 - f_v C_0)\rho_l \tag{2.15}$$

The drift velocity describes the variance in velocities between the phases of a mixture and it is given as (Shi et al., 2005)

$$u_d(P, u_m, \rho_m) = \frac{(1-f_v C_0)u_c K(f_v, K_u, C_0) m(\theta)}{f_v C_0 \sqrt{\rho_v/\rho_l} + 1 - f_v C_0} \tag{2.16}$$

where

u_c is the characteristics velocity, given by

$$u_c = \left[\frac{g\sigma_{vl}(\rho_l - \rho_v)}{\rho_l^2}\right]^{\frac{1}{4}} \tag{2.17}$$

where σ_{vl} is the vapour-liquid surface tension.

K_u is the Kutateladze number, described as

$$K_u = \left[\frac{C_{ku}}{\sqrt{N_B}}\left(\sqrt{1 + \frac{N_B}{C_{ku}^2 C_w}} - 1\right)\right]^{\frac{1}{2}} \tag{2.18}$$

in which $C_w = 0.008$ and $C_{ku} = 142$ (Oldenburg & Pan, 2013), and N_B is Bond number, defined as

$$N_B = 4r_i^2\left[\frac{g(\rho_l - \rho_v)}{\sigma_{vl}}\right] \tag{2.19}$$

$m(\theta)$ is an inclination adjusting function, described as

$$m(\theta) = m_0(\cos\theta)^{n_1}(1 - \sin\theta)^{n_2} \tag{2.20}$$

in which m_0, n_1 and n_2 are fitting parameters.

K is a smooth transition function, introduced to make a smooth transition of the drift velocity between the bubble rise and the film flooding flow regimes, described as(Oldenburg & Pan, 2013)

$$K = \begin{cases} 1.53 & f_v \leq a_1 \\ 1.53 + \frac{C_0K_u - 1.53}{2}\left[1 - \cos\left(\pi \frac{f_v - a_1}{a_2 - a_1}\right)\right] & a_1 \leq f_v \leq a_2 \\ C_0K_u & f_v \geq a_2 \end{cases} \qquad (2.21)$$

in which a_1 and a_2 are two transitional gas volume fractions corresponding to the bubble rise and the film flooding flow regimes, respectively.

C_0 is a profile parameter, calculated as (Shi et al., 2005)

$$C_0 = \frac{C_{max}}{1 + (C_{max} - 1)\eta^2} \qquad (2.22)$$

in which C_{max} is the profile parameter for low gas fraction and η is a parameter reflecting the effect of the flow status on the profile parameter given by

$$\eta = \frac{\beta - B}{1 - B} \qquad (2.23)$$

where B is the threshold parameter above which C_0 starts to drop below C_{max} described as (Oldenburg & Pan, 2013)

$$B = \frac{2}{C_{max}} - 1.0667 \qquad (2.24)$$

and β is calculated as

$$\beta = max\left(f_v, \frac{f_v|u_m|}{u_{sgf}}\right), 0 \leq \beta \leq 1 \qquad (2.25)$$

in which u_{sgf} is the gas superficial velocity at which flooding occurs and it is calculated using

$$u_{sgf} = K_u \left(\frac{\rho_l}{\rho_v}\right)^{\frac{1}{2}} u_c \qquad (2.26)$$

The heat exchange between the well and surrounding formation can be described as (Arzanfudi & Al-khoury, 2015)

$$Q = 2\pi r_w U\big(T - T_e(z)\big) \qquad (2.27)$$

in which $T_e(z)$ is the formation temperature, and U is the overall thermal interaction coefficient of the wellbore, which can be described as (Shao et al., 2014)

$$U = \frac{1}{R_{conv} + R_{cond}} \qquad (2.28)$$

where R_{conv} and R_{cond} are thermal resistances of the fluid and casing material, respectively, described as

$$R_{conv} = r_o/\left(r_w\bar{h}\right) \tag{2.29}$$

$$R_{cond} = r_o/\ln(r_o/r_w)\,/\lambda_p \tag{2.30}$$

in which r_o is the outer radius of the wellbore, λ_p is the thermal conductivity of the casing material, and $\bar{h}$ is the convective heat transfer coefficient, described as

$$\bar{h} = \mathrm{Nu}\lambda_m/(2r_i) \tag{2.31}$$

where Nu is Nusselt number, defined as

$$\mathrm{Nu} = \begin{cases} 0.664\mathrm{Re}^{1/2}\mathrm{Pr}^{1/3} & \mathrm{Re} \leq 2000 \\ 0.023\mathrm{Re}^{0.8}\mathrm{Pr}^{0.4} & \mathrm{Re} \leq 2000 \end{cases} \tag{2.32}$$

In which Re is Reynolds number and Pr is Prandtl number given respectively as:

$$Re = \frac{\rho\mu_m D_p}{\mu_m} \tag{2.34}$$

$$\mathrm{Pr} = \frac{\mu_m c_{pm}}{\lambda_m} \tag{2.35}$$

where c_{pm} and λ_m are the specific isobaraic heat capacity and thermal conductivity of the mixture respectively.

2.5.4 The Homogeneous Equilibrium Relaxation Mixture (HERM) model

The HERM model is developed to account for thermodynamic non-equilibrium caused by delayed phase transition during decompression. For a vapour-liquid two-phase mixture, as the phase fraction can no longer be assumed at thermodynamic equilibrium values, an additional equation describing its evolution with the flow needs to be introduced. The non-equilibrium liquid-vapour transition is accounted for by a relaxation to thermodynamic equilibrium through Equation (2.36) for the vapour mass fraction. Following Bilicki & Kestin (1990), Downar-Zapolski & Bilicki (1996), Brown et al. (2013) and Nouri-Borujerdi & Shafiei Ghazani (2017), such an Equation is essentially a scalar transport Equation given by:

$$\frac{\partial x}{\partial t} + u_m\frac{\partial x}{\partial z} = \frac{x_{eq}-x}{\emptyset} \tag{2.36}$$

where x is the dynamic vapour quality and $\emptyset$ is a relaxation time accounting for the delay in the phase change transition. The mixture density, ρ_m, is defined as (Brown et al., 2013):

$$\frac{1}{\rho_m} = \frac{x}{\rho_v(P)} + \frac{(1-x)}{\rho_{ml}(P,e_{ml})} \tag{2.37}$$

and the mixture internal energy:

$$e = xe_{sv}(P) + (1-x)e_{ml} \tag{2.38}$$

where the "subscripts *sv* and *ml* respectively refer to the saturated vapour and meta-stable liquid phases, both of which may be at different temperatures" (Brown et al., 2013).

Thus for the HERM an additional relaxation time dependent Equation can be rewritten as:

$$\frac{\partial \rho_m x}{\partial t} + \frac{\partial \rho_m u_m x}{\partial z} = \rho_m \frac{x_{eq} - x}{\emptyset} \tag{2.39}$$

Based on "tests involving the steady flow of CO$_2$ through a nozzle, (Angielczyk et al., 2010) proposed the following empirically determined correlation for the relaxation time", $\emptyset$ (Brown et al., 2013):

$$\emptyset = 2.15 \times 10^{-7} \left(x \frac{\rho_m}{\rho_{sv}}\right)^{-0.54} \left(\frac{P_s(T_{in}) - P}{P_c - P_s(T_{in})}\right)^{-1.76}, \tag{2.40}$$

where T_{in} is feed temperature, P_c is critical pressure and P_s is saturation pressure at the given temperature.

2.6 Thermodynamics

2.6.1 The Equation-of-State (EoS)

To determine the fluid flow thermal properties (e.g. pressure and temperature) and phase equilibrium data, an Equation-of-State (EoS) can be applied. Generally for pipeline decompression modelling (as also in CO$_2$ injection modelling), the most commonly adopted EoS are the cubic EoS which are computationally efficient (Munkejord et al., 2016). Examples include (Mahgerefteh et al. 2008; Teng et al. 2016; Nouri-Borujerdi & Shafiei Ghazani 2017). Among these, the most widely validated for CO$_2$ is the Peng-Robinson EoS (PR RoS; Peng & Robinson 1976) and its variations (e.g. volume-translated Peng-Robinson EoS (Abudour et al., 2013)). Its general form and key parameters are presented below:

$$P = \frac{R}{v-b} - \frac{c\delta}{v^2 + 2bv + - b^2} \tag{2.41}$$

where c, b and δ are respectively given by:

$$c = 0.45724 \frac{(RT_c)^2}{P_c} \tag{2.42}$$

$$b = 0.0778 \frac{RT_c}{P_c} \tag{2.43}$$

$$\delta = \left[1 + \left(0.37464 + 1.54226\omega - 0.2699\omega^2\right)\left(1 - \sqrt{\frac{T}{T_c}}\right)\right] \tag{2.44}$$

where P_c, T_c and ω are respectively the critical pressure, the critical temperature and the acentric factor for CO_2.

From Equation (13), the internal energy may be calculated using the identity (Poling, Prausnitz, & O'Connell, 2001):

$$e - e_{ig} = \int_{-\infty}^{v} \left[T\left(\frac{\partial P}{\partial T}\right)_v - P\right] dv \tag{2.45}$$

where the subscript, ig refers to ideal gas. The mixture speed of sound, c is defined through the relation (Brown et al., 2013):

$$\frac{1}{\rho c^2} = \frac{x}{\rho_v c_v^2} + \frac{1-x}{\rho_l c_l^2} \tag{2.46}$$

Recently, more advanced multi-parameter EoS have become popular especially for multi-component mixtures, due to their superior accuracy (see (Elshahomi et al. 2015; Munkejord & Hammer 2015), for example).

2.6.2 Predictions of Key Thermal Properties

Following the above, the calculation of the fluid pressure, specific enthalpy and specific entropy based on an EoS is presented (Lemmon & Tillner-Roth, 1999):

$$p = \rho RT + \frac{\rho^2}{\rho_r}\left(\frac{\partial a^r}{\partial(\rho/\rho_r)}\right) \tag{2.47}$$

$$h = RT + RT_r\left(\left(\frac{\partial a^o}{\partial(T_r/T)}\right) + \left(\frac{\partial a^r}{\partial(T_r/T)}\right)\right) + \frac{\rho RT}{\rho_r}\left(\frac{\partial a^r}{\partial(\rho/\rho_r)}\right) \tag{2.48}$$

$$s = \frac{RT_r}{T^2}\left(\left(\frac{\partial a^o}{\partial(T_r/T)}\right) + \left(\frac{\partial a^r}{\partial(T_r/T)}\right)\right) - \frac{R\left(a^o + a^r\right)}{T} \tag{2.49}$$

Note that the above Equations are all explicit in Helmholtz free energy and its derivatives.

Alternatively for PR EoS (Equation 2.41), the departure Equation can be applied as shown in the following. Writing PR EoS in terms of compressibility factor, *Z*:

$$Z = 1 + \frac{\tilde{b}\rho}{1-\tilde{b}\rho} - \frac{\tilde{a}}{\tilde{b}RT}\left(\frac{\tilde{b}\rho}{1+2\tilde{b}\rho-\tilde{b}^2\rho^2}\right) \tag{2.50}$$

the fluid specific enthalpy and entropy are respectively given by:

$$h - h^o = RT\left[\int_0^\rho -T\left(\frac{\partial Z}{\partial T}\right)\frac{d\rho}{\rho} + Z - 1\right] \tag{2.51}$$

$$s - s^o = R\left[\int_0^\rho \left(-T\left(\frac{\partial Z}{\partial T}\right) - (Z-1)\right)\frac{d\rho}{\rho} + \ln(Z)\right] \tag{2.52}$$

where the value zero in the integration limit represents the ideal gas limit (Z = 1). h^o and s^o are respectively the ideal gas specific enthalpy and entropy at a given state (p and T).

Another important thermodynamic variable is the fluid speed of sound defined as:

$$c = \sqrt{\left(\frac{\partial \rho}{\partial p}\right)_s} \tag{2.53}$$

2.7 Applications of the HEM model

2.7.1 Mahgerefteh et al. (1999)

The HEM model applied for pipeline decompression by Mahgerefteh et al. (1999) is validated and reported in the open literature. The model was originally developed for the quantitative failure consequence assessment of high-pressure hydrocarbon and CO_2 pipeline Full Bore Rupture (FBR) failures (Mahgerefteh et al. 1999; Mahgerefteh et al. 2007; Mahgerefteh et al. 2008) and later extended to puncture failures (Oke et al. 2003). Other important features were reported in subsequent studies, including the simulations of emergency dynamic valve responses (Mahgerefteh et al., 2000), ductile fracture propagation of the pipe wall (Mahgerefteh et al. 2011; Mahgerefteh et al. 2012) and decompression of pipeline networks (Mahgerefteh et al., 2006).

Following their methodology, the HEM model (equations 2.6 to 2.8) is used to describe the fluid flow dynamics. Fluid/wall heat transfer and friction are modelled using standard correlations (following Steiner & Taborek (1992) and Techo et al. (1965), respectively). Real fluid behaviour is also accounted for using PR EoS (Peng & Robinson, 1976).

In solving the resulting PDE system, the conservation Equations in the HEM model are firstly converted into wave form and the set of three ODEs are identified along the relevant waves given by:

$$dp + \rho c du = \left(cS_F - c^2 \left(\frac{\partial \rho}{\partial s} \right)_p \frac{S_E}{T} \right) dt \tag{2.54}$$

$$dp - \rho c du = \left(-cS_F - c^2 \left(\frac{\partial \rho}{\partial s} \right)_p \frac{S_E}{T} \right) dt \tag{2.55}$$

$$ds = \frac{S_E}{T} dt \tag{2.56}$$

where the wall/fluid friction and volumetric heat source terms, S_F and S_E are given by:

$$S_F = \frac{2 f \rho |u| u}{D_{w,in}} \tag{2.57}$$

$$S_E = \frac{4\dot{q}_w}{D_{w,in}} = \frac{4U\left(T_w - T\right)}{D_{w,in}} \tag{2.58}$$

The Method of Characteristics (MoC) (Zucrow & Hoffman, 1976) is then applied to obtain numerical solutions of Equations (2.54) to (2.56). The flow model has been validated against multiple large-scale high-pressure pipeline decompression experiments for both hydrocarbons and CO_2.

The following presents the results relating to the testing of the model covering the experimental set-up, relevant measurements taken and model validation by comparison to the field data.

National Grid COOLTRANS CO_2 pipeline decompression tests 6 and 7 (Cosham, Jones, Armstrong, Allason, & Baranett, 2012)

As part of the COOTRANS project undertaken by National Grid U.K., a number pipeline Full Bore Rupture (FBR) decompression tests were carried out for CO_2 and CO_2-rich mixtures in order to simulate the decompression wave behaviour in single- and two-phase flows. Among these, tests 6 and 7 consisted of a 144 m length, 150 mm i.d., 11 mm pipe wall thickness heavily insulated ASTM A333 Grade 6 low carbon steel seamless pipe filled with gaseous mixtures of CO_2, N_2 and SO_2. The pertinent test conditions are summarised in table 2.2.

Table 2.2: Relevant initial conditions for test 6 and 7

Test	Feed compositions	Feed temperature (°C)	Feed pressure (bara)	Ambient temperature (°C)
6	95.97 mol% CO_2 + 4.03 mol% N_2	5.3	37.9	20.4
7	99.14 mol% CO_2 + 0.86 mol% SO_2	9.9	38.0	13.9

The test pipe was instrumented with multiple pressure and temperature transducers along its entire length to monitor transient pressure and temperature variations. In addition, by comparing the recoded pressure-time profiles from adjacent pressure transducers (with known distance apart), decompression wave speed was also reported in the form of a fan diagram (pressure versus decompression wave speed plot).

In validating the flow model, the predicted decompression wave speed was compared to the corresponding measurement. The results are presented in Figures 2.5 and 2.6.

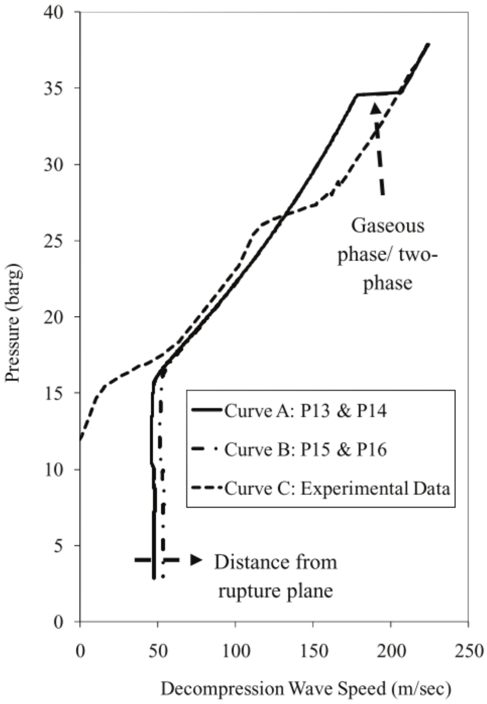


Figure 2.5: Fan diagram at pressure transducers P13 to P16 following FBR for test 6. Curves A and B: model predictions; Curve C: measured data (Mahgerefteh et al., 2012).

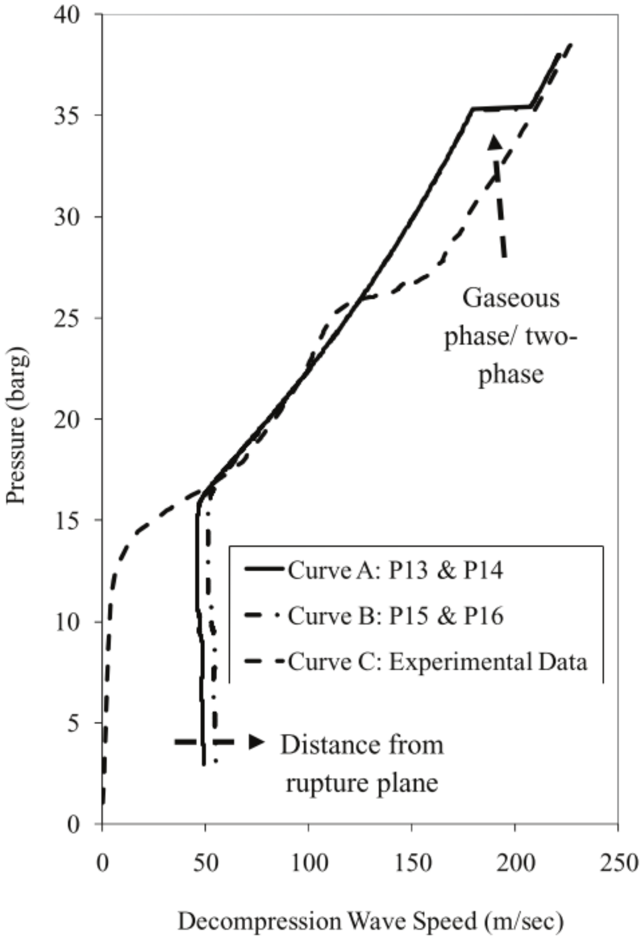


Figure 2.6: Fan diagram at pressure transducers P13 to P16 following FBR for test 7. Curves A and B: model predictions; Curve C: measured data (Mahgerefteh et al., 2012).

Focussing on the measured data in both above figures, a pressure plateau can be observed (each at 34 bara for test 6 in Figure 2.5 and at 36 bara for test 7 in Figure 2.6). Such a pressure plateau corresponds to condensation of vapour, where the speed of sound and hence the decompression wave speed significantly decrease (by ca. 50 m/s in both cases). The wave speed eventually reaches zero when the local flow (between transducer P13 to P16) becomes sonic (choked).

With regards to the model predictions in both cases, despite demonstrating a general agreement with the data, over-predictions can be observed for the pressure plateaux at the vapour-liquid phase transition boundaries and the decompression wave speeds at the later stages of the decompression process. According to the authors, the former is due to thermodynamic non-equilibrium between the vapour and liquid phases, which is not accounted for in their model. With regards to the over-prediction for the decompression wave speed, this is attributed to the friction correlation adopted (Mahgerefteh et al., 2012).

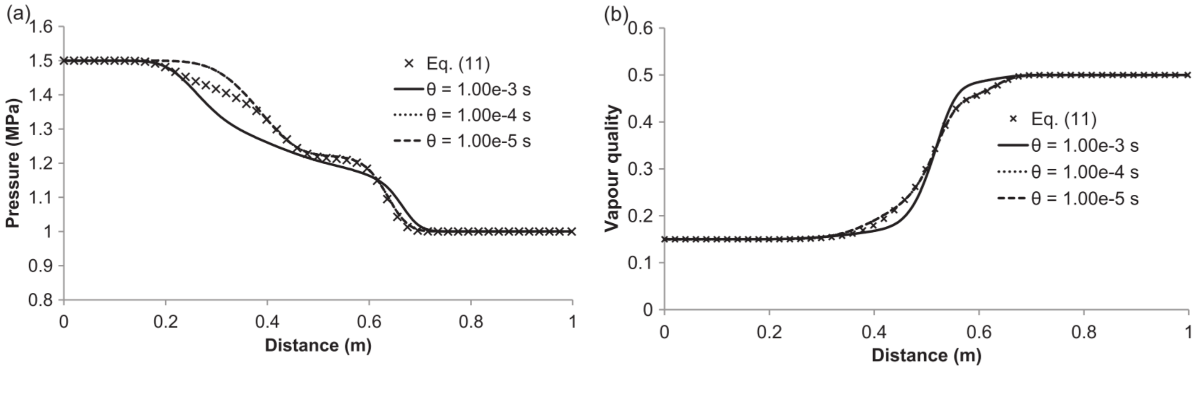
2.8 An application of the HERM

2.8.1 Brown et al. (2013)

In a recent study by Brown et al. (2013), the HERM model was applied to investigate the transient decompression of a high-pressure CO_2 pipeline. The neglected time-dependent terms such as the vapour quality which are not accounted for in the HEM were accounted for in the HERM. Fluid/wall friction was computed using Chen's correlation (Ning Hsing Chen, 1979), and fluid/wall heat transfer was argued to be insignificant given the very short decompression duration.

The resulting PDE system was solved numerically using the Finite Volume Model (FVM; Randall J LeVeque, 2002) in conjunction with the Harten-Lax-Van Leer (HLL) scheme (Toro, 2009).

The model developed was first verified by solving a specified Riemann problem on a computational domain of 1 m length filled with vapour-liquid two-phase CO_2. The initial conditions for the Riemann problem were given by a step function each in pressure (15 bara to 1 bara) and vapour quality (0.15 to 0.5). Both constant relaxation times (ranging from 0.01 to 1 ms) and those computed using the correlation were tested. The results in terms of pressure and density profiles along the length of the computational domain are plotted in Figure 2.7.



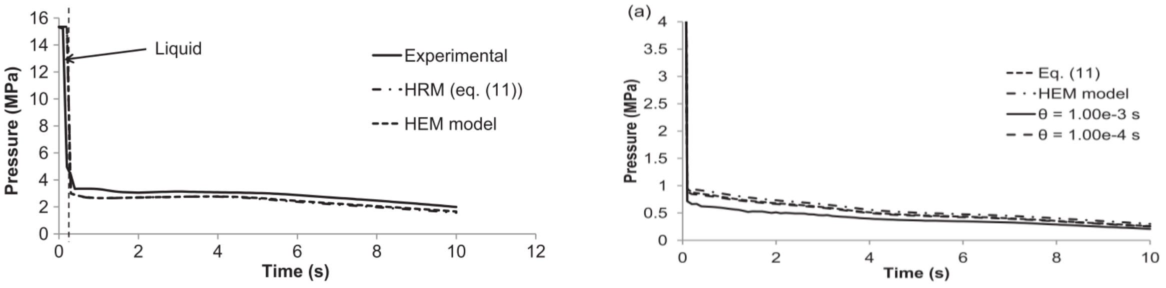
(a) (b)

Figure 2.7: Predicted pressure (a) and vapour quality (b) profiles along the computational domain based on the computed relaxation time (denoted as Eq. (11) in the Figure legend) and different constant relaxation times (denoted as θ) at 2 ms from the start of the simulation (Brown, Martynov, et al., 2013).

From Figure 2.7 (a), it is clear that a larger relaxation time is observed to produce a faster decompression wave front. The slowest and the fastest fronts are 0.1 m apart, which gives a wave speed difference of ca. 50 m/s. This is attributed to the corresponding increased delay in vapour-liquid phase transition (see Figure 2.7 (b)).

The authors went on to simulate a large-scale CO_2 pipeline FBR decompression test performed during the course of the National Grid UK COOLTRANS project (Cosham et al., 2012). The test rig was the same as that presented in Section 2.7.1. The pipe was initially filled with pure CO_2 at 153.35 bara and 278.35 K, and FBR was initiated at one end using an explosive charge. Pressure and temperature were measured at multiple points along the pipe length during decompression.

Figure 2.8 presents the predictions from the HERM and HEM models for the variations of pressure as a function of time at both pipe intact and release ends. Also included in Figure 2.8 (a) are the measurements at the pipe intact end for comparison.



(a) (b)

Figure 2.8: Predicted pressure-time profiles from the HERM and HEM models at the pipe intact end (a) and the release end (b) during decompression. The measurement at the pipe intact end is included in (a) (Brown, Martynov, et al., 2013).

Referring to Figure 2.8 (a), there is a marginal difference between the HERM and HEM model predictions; they are both in good agreement with the measured data, showing a rapid drop in the decompression rate at vapour-liquid phase transition boundary (marked by the vertical dashed line in Figure 2.8 (a)). Turning to Figure 2.8 (b), at the pipe release end, the

HERM model predicts a faster decompression rate prior to phase transition. This corresponds to the faster decompression waves as a result of delayed phase transition. Unfortunately, no experimental data is available at the release end, and thus there is no conclusive evidence to support the validity of the simulated results.

2.8.2 Concluding remarks

From the above review of the HERM model application, where thermodynamic non-equilibrium is relevant, the model has shown to produce a significant improvement over the HEM model. However, as compared to the HEM model, the HERM model validation against appropriate experimental data is lacking. The following is a summary of the key findings:

- Where thermodynamic non-equilibrium is relevant, the HERM model is shown to produce improved accuracy compared to the HEM model;
- The sensitivity of the HERM model predictions is dependent on the specified relaxation time;
- For pipeline decompression, a higher decompression rate (prior to phase transition) is predicted by the HERM model as compared to the HEM model. However, this needs to be verified against experimental data.

Chapter 3: Modelling of CO_2 Start-up Injection into Depleted Gas Fields

3.1 Background

As pointed out in chapter 2, the behaviour of CO_2 during well injection operations is strongly dependent on the CO_2 injection rate, injection temperature, and well configuration. Different parameters have been identified to have influence on the well transient behaviour by various demonstration projects. Although, the analysis of these projects suggests a cooling effect but the main challenge facing them is whether the prediction of such cooling effect can be done simply or not, thus, the primary parameter that causes the effect remains unclear. As such, the phenomenon underlying the CO_2 behaviour during transient well injection operation will require a proactive approach that employs a rigorous numerical model for the evaluation the process. This approach will involve specific challenges with the transient behaviour of CO_2 during well injection. Some commercial software like OLGA are useful and had been used for pipeline multi-phase flow simulation but are limited in most cases. The advantages and disadvantages of such software are discussed below.

3.1.1 Advantages and Disadvantages of Commercial Software

As discussed earlier in chapter 2, Li et al., (2015) studied the influence of various parameters on transient CO_2 operations based on the design of wells in the Goldeneye CCS Project. They employed OLGA commercial software in their analysis, but the use of such commercial simulators causes difficulty in model validation, since there is often little public data available about the details of the model. Also, such software is implemented for specific purpose which may not accommodate further modifications. Notably, the source of the work published by Li et al, (2015) being a commercial software package contains less details of the model and can be interpreted as a black-box or in-house experimental solution. In other words the available commercial software for pipeline consequence simulation such as CFD, Fluent, OLGA, Aspen Hysys, Pipephase, Pipesim, etc. are advantageous is some instances and not applicable in others. The underlying reason for their inapplicability is the suitability of the software for the specific purpose required by the user. The key advantages and disadvantages of these commercial software are summarised below.

The main advantages are

- Easy to acquire and use

- Programmed for specific tasks
- Mostly good interface
- Simulation results obtained are mostly reliable

The disadvantages are

- Limited multiplatform programming
- Cannot copy due to lack of model details
- Mostly difficult to modify
- Restrictions of usage
- Higher cost required for purchase of commercial software
- Technical support issues
- Proprietary software

In the light of these limitations, there is an undisputed benefits of developing independent models such the HEM or HERM suitable for the purpose of the specific case studies to be examined.

As CO_2 approaches its supercritical state 31.1 ^{o}C, 73.8 bara (Blunt, 2010), which is likely to occur somewhere along the wellbore, sharp changes in the properties of CO_2 are induced (Jiang et al., 2014; Li et al., 2015). These have the ability to render any numerical model unstable. The “transient operations involve multiple physical processes in and around the wellbore, such as liquid/gas two-phase flow of CO_2, significant Joule-Thomson cooling of CO_2, and transient heat conduction between fluids and the wellbore and the surrounding rock, which can induce significant non-linearity in the fluid properties via the equation-of-state, rendering numerical solutions difficult” (Li et al., 2015).

The transient operations and pseudo steady-state numerical simulation of wellbore dynamics are both reported various in the literatures. Hasan and Kabir, (1991, 2002); Paterson et al., (2010) presented “analytical solutions based on diffusivity coupled to the energy balance to predict the temperature profile of single-phase or gas/liquid flow in the wellbore. Pan, Webb, and Oldenburg, (2011) later presented an analytical solution to predict steady-state, compressible and isothermal two-phase flow through a wellbore using a drift flux model” (Li et al., 2015). Also, recently Li et al, (2015); Ruan et al., (2013) and (2013b) studied the CO_2 temperature increase phenomenon in the tubing, and the rock temperature behaviour in both axial and radial directions using various two-dimensional radial numerical models. Also, “Pan and Oldenburg, (2014) successfully developed a coupled wellbore and reservoir model, called T2Well, to simulate the dynamics of CO_2 flow in the wellbore and reservoir. In that reservoir model, the standard multiphase Darcy flow approach was used and a drift-flux model and the conservation equations for describing transient two-phase non-isothermal wellbore flow of CO_2-water mixtures were employed" (Li et al., 2015). Paterson et al., (2008) "numerically simulated the temperature and pressure profile considering CO_2 liquid-gas phase change in static wells, producing wells and blowout situations, with assumptions of quasi-steady state and neglecting changes in kinetic energy" (Xiaolu Li et al., 2015). The observation of low temperatures in the wellbores can be attributed to the expanding CO_2 as the pressure decreases and followed by a phase change. Paterson et al., (2008) discussed "potential two-phase behaviour during shut-in of injection wells and stated that a reservoir model is required to address the coupled wellbore-reservoir response. Recently, this approach has been extended to gas mixtures including CO_2 and impurities, and the phase behaviour of mixed gas injection was investigated in cases involving a cyclical injection rate" (Lu & Connell, 2014).

Notably, none of these models have focused on the time sensitivity or injection condition sensitivity of transient operations such as ramping up on start-up injection of CO_2. Consequently, "none of the recommended transient operations involved an optimal start-up period and no procedure suitable for different injection conditions has yet been presented for the complex CO_2 dynamic behaviour that takes place during transient operations, actual data are required for CO_2 injection design" (Li et al., 2015).

The Peterhead CCS project, also known as the Goldeneye CCS project was planned, designed and implemented at a gas power station in Aberdeenshire, UK (Shell UK, 2015). The data used in this present study is based on the conditions of operation for the proposed Goldeneye CCS project. The project comprises capturing of the CO_2 from a Combined Cycle Gas Turbine (CCGT) power station at Peterhead being transported offshore to depleted gas fields 2.5 km beneath the North Sea for storage it in the Goldeneye reservoir (Li et al., 2015; Shell UK, 2015).

This chapter presents the Finite Volume Method numerical solution to the HEM flow model presented in chapter 2. The application software (CLAWPACK) is presented with the applied boundary conditions for the start-up injection CO_2 into depleted gas fields. The FVM numerical solution follows the flux-difference splitting scheme which solves a numerical inviscid flux function for a general system of conservation equation. The start-up injection of CO_2 at the top of the well is described by the constant mass flux subsonic inflow boundary condition for a one-dimensional hyperbolic system. The HEM model is deployed to simulate the start-up injection process with focus on the CO_2 pressure, temperature and density behaviour at the top and bottom of the well.

3.1.2 Wells and injection scenarios

Wells are an essential component of any CO_2 storage project. They are the only way by which CO_2 can be discharged into underground reservoirs in the timeframes required. It is important to recognise that the injected CO_2 is not stored in the wells. They simply represent the transportation route through which the CO_2 will be released into reservoirs and then remain safely trapped. Wells are drilled for a range of purposes such as appraisal, exploration, injection, monitoring or production. Well objectives strongly influence its design, depth, size and cost. The information gathered from existing wells drilled and used by oil and gas operators proves to be very useful in characterising the subsurface geology of a site (ETI, 2016). Figure 3.1 shows a typical well for the injection of CO_2 into a depleted gas field. The well is composed of various layers (typically filled with fluids different from CO_2) and surrounded by a formation, whose composition varies from site to site. Wells have to be carefully analysed in order to assess their suitability for CO_2 injection in depleted oil and gas fields (Raza et al., 2016).

To the begin steady-state injection of CO_2 into a preselected the reservoir, it is necessary to perform time-dependent operations to estimate important reservoir properties, e.g. permeability and pressure. Such operations include start-up, shut-in and emergency shut-down, which prove to be critical in the overall design of the well (Böser & Belfroid, 2013). It is possible that after steady injection condition is attained the well may be subjected to shut in and started up for maintenance of the upstream transportation system or other routine checks. It is therefore of paramount importance to be able to predict the behaviour of the CO_2, in terms of pressure and temperature along the full length of the well in order to characterise and quantify potential risks (Li et al., 2015).

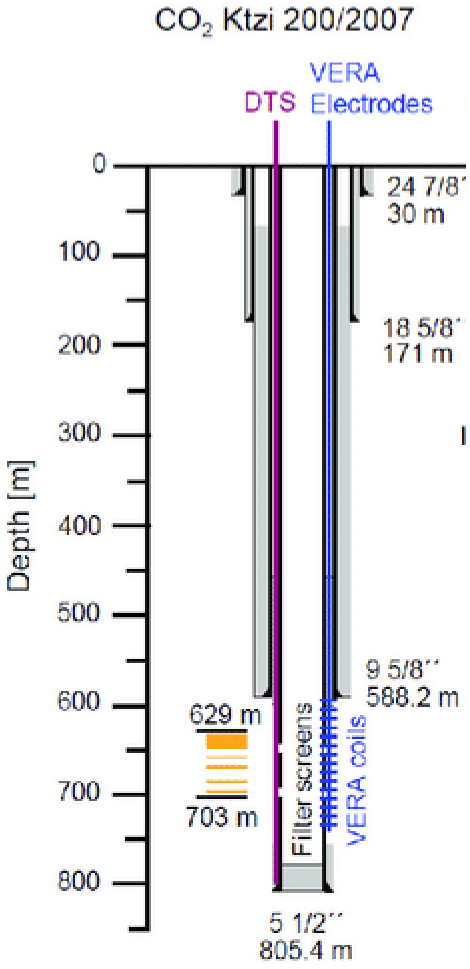


Figure 3.1: Schematic representation of a deep well CO_2 injection well (Martens et al., 2011)

3.2 Numerical solution scheme

In order to describe an outflow of a CO_2 mixture from a long injection pipe, a mathematical model is developed using several assumptions as expressed in section 2.5. In particular, if the ratio of the pipe length is much larger than the pipe diameter; "the flow is described as one-dimensional". In order to describe the wave propagation in a pipe, the compressible and multiphase nature of the fluid is taken into account in the model. The effects of turbulent motion, friction and heat transfer at the pipe wall and between the phases will be taken into account in sub-models using empirical correlations for steady-state flows. The flow model is developed within a general framework of Eulerian interpenetrating continua multi-fluid concept (see A. Prosperetti & G. Tryggvason, 2007). The concept allows the fluid to be treated as a continuous multiphase fluid where both phases co-exist everywhere in the domain. Also, the conservation equations for each phase can be solved separately with provision made to include the source terms in the conservation equations using constitutive equations

The HEM flow is modelled using a concept of interpenetrating continua (Joseph & Lundgren, 1989). The set of governing equations describing one-dimensional flow in a vertical pipe as presented in Equations (2.6 – 2.8) can be written in numerical solution scheme as:

Mass conservation:

$$\frac{\partial\rho}{\partial t}+\frac{\partial\rho u}{\partial z}=S^{m} \tag{3.1}$$

Momentum Conservation:

$$\frac{\partial\rho u}{\partial t}+\frac{\partial\rho u^{2}}{\partial z}=-\frac{\partial P}{\partial z}+\rho g+S^{mom} \tag{3.2}$$

Energy Conservation:

$$\frac{\partial\rho e}{\partial t}+\frac{\partial\rho u^{3}e}{\partial z}+\frac{\partial Pu}{\partial z}=-\frac{\partial uP}{\partial z}+\rho g+S^{e} \tag{3.3}$$

where S^{m}, $S^{mom}$and S^{e} are the source terms for mass, momentum and energy conservation.

The source terms S^{m}, $S^{mom}$and S^{e} describe the effects of mass, momentum and heat exchange between the fluid and its surrounding respectively, as well as friction and heat exchange at the pipe wall. Various correlations (such as the ones presented in Cheng et al, 2008) are available in the literature for modelling the source terms depending on the nature of fluid phase and the flow regime in a pipe. For practical calculations, the closure relationship for the source terms must be provided.

The aforementioned equations describing the transient release from a long pipe consists of quasi-linear hyperbolic partial differential equations. These equations can only be solved numerically. Previous attempts include a finite difference method (Bendiksen et al. 1991), a finite element method (Lang 1991) and the method of characteristics (MOC) (Mahgerefteh et al. 2008), (Oke et al. 2003), (Mahgerefteh et al. 2006). The MOC has been used to solve equations describing homogeneous equilibrium flow (Oke et al. 2003). However, it is incapable of handling highly transient flows. In this study, a more effective model based on the Finite Volume Method (FVM), incorporating a conservative Godunov type finite-difference scheme (Godunov 1959, Radvogin et al. 2011, Cumber et al. 1994) is used. The

FVM is well-established and thoroughly validated CFD technique. In essence, the methodology involves the integration of the fluid flow equations over the entire control volumes of the solution domain and then accurate calculation of the fluxes through the boundaries of the computed cells.

For the purpose of numerical solution of the governing equations they are written in a vector form (Toro 2010):

$$\frac{\partial\vec{Q}}{\partial t}+\frac{\partial\vec{f}}{\partial z}=\vec{S}, \tag{3.4}$$

where

$$\vec{Q}=(\rho\,,\ \rho u\,,\ \rho e)^T, \tag{3.5}$$

$$\vec{f}=\left(\rho u\,,\quad(\rho u^2+P)\,,\ u(\,\rho ue+\rho u^2+P)\right)^T \tag{3.6}$$

$$\vec{S}=(S^m,\ S^{mom},\ S^e)^T \tag{3.7}$$

$\vec{Q}$, $\vec{f}$ and $\vec{S}$ are the vectors of conserved variables, fluxes and source terms respectively.

Equations (3.5 – 3.7) form a set of quasi-linear hyperbolic equations, provided that they have distinct and real eigenvalues. Equations of such kind can be solved numerically using methods developed in computational gas dynamics (Toro 2010, Leveque, 2002). One of these methods is the finite volume method largely used for computation of transient compressible flows.

Applying the finite volume methodology, the spatial domain is discretised into a finite number of cells (control volumes or grid cells) and keeping track of an approximation of the integral of the flux over these volumes. In each time step the approximation of the flux through the endpoints of the interval is updated.

Denote the *i*-th grid cell by

$$C_i=\left(z_{i-1/2},z_{i+1/2}\right), \tag{3.8}$$

as shown in Figure 3.2. The value Q_i^n will approximate the average value over the *i*-th interval at time t_n:

$$Q_i^n\approx\frac{1}{\Delta x}\int_{z_{i-1/2}}^{z_{i+1/2}}q(z,t_n)dx\equiv\frac{1}{\Delta z}\int_{C_i}^{C_n}q(z,t_n)dx \tag{3.9}$$

where $\Delta x = z_{i+1/2} - z_{i-1/2}$ is the length of the cell.

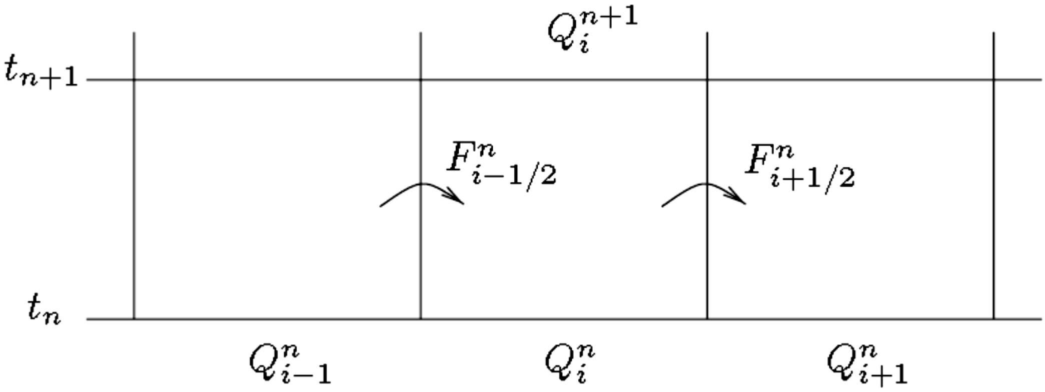


Figure 3.2: Cell variables and inter-cell fluxes in finite-volume discretisation of the special and time domains.

Then conservation equations are integrated over a control volume as can be seen in Figure 3.2 to transform the differential equations to a finite set of algebraic equations. Integrating and rearranging equation (3.9) gives:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta z}\left(F_{i+1/2}^n - F_{i-1/2}^n\right) \qquad (3.10)$$

where $F_{i-1/2}^n$ is the approximation to the average flux along $\Delta z = z_{i-1/2}$. Rewriting equation (3.10) becomes:

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \frac{\left(F_{i+1/2}^n - F_{i-1/2}^n\right)}{\Delta z} = S_i \qquad (3.11)$$

where Q_i^n is an average value for a piece wise constant in each cell for the i-th control volume, the n-th time step. Equation (3.11) uses explicit time integration scheme.

3.2.1 The CLAWPACK software

The finite volume method used in this study has been implemented through the software package CLAWPACK (see Appendix). The software was authored by R. J. LeVeque in his book titled “Finite Volume Methods for Hyperbolic Problems” published in the Cambridge University Press, 2002 {http://www.clawpack.org/book.html}. The software allows the application of “wide variety of hyperbolic systems simply by providing the appropriate Riemann solver, along with initial data and boundary conditions” (R.J. LeVeque, 2002).

In one-dimensional space, the CLAWPACK routine “claw1” or the simplified version “claw1ez” can be used to solve a system of equations of the form (R.J. LeVeque, 2002):

$$\kappa(z)q_t + f(q)_x = \psi(q, z, t), \qquad (3.12)$$

where $q = q(z, t)$, and the standard case of homogeneous conservation law has $\kappa \equiv 1$ and $\psi \equiv 0$, thus equation (3.12) becomes;

$$q_t + f(q)_z = 0 \qquad (3.13)$$

The basic requirement on the homogeneous system is that it be hyperbolic in the sense that a Riemann solver (HLL, Harten, Lax, van Leer solver) can be specified which approximate the solution based on the integral form of the conservation equations and the velocities at the interface.. For instance, if the input data consist of two arrays ql and qr, the value $ql(i,:)$ is the value Q_i^L at the left edge of the i-th cell, while $qr(i,:)$ is the value Q_i^R at the right edge of the i-th cell as indicated in Figure 3.3.

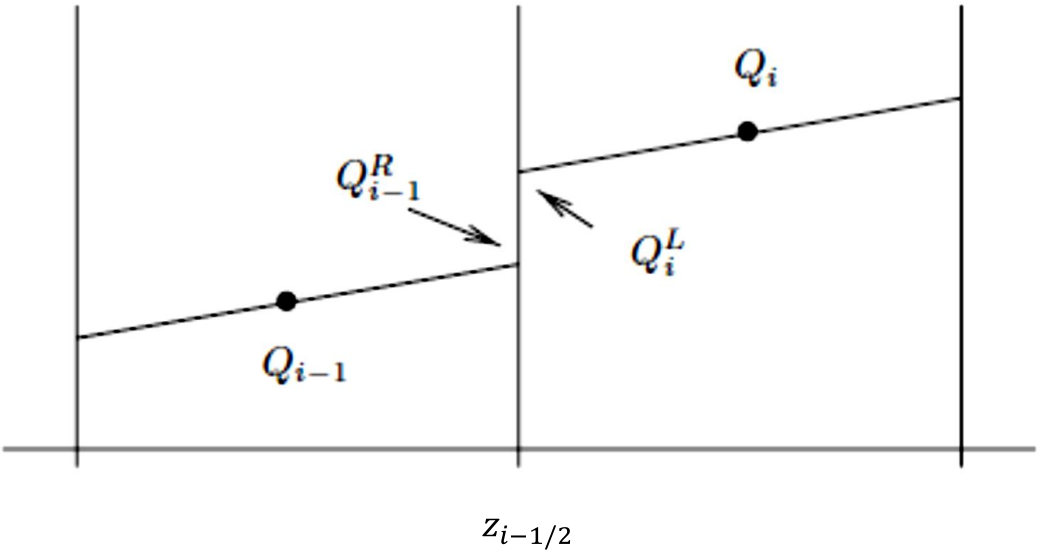


Figure 3.3: The states used in solving the Riemann problem at the cell interface, $\boldsymbol{z_{i-1/2}}$ (R. J. LeVeque, 2002).

A “Riemann problem in the theory of hyperbolic equations is a problem in which the initial state of the system is defined” (Dullemond & Johansen, 2007) and having a piecewise constant with a single discontinuity. In the context of numerical analysis, the Riemann problem mostly appear in a natural way in the form of finite volume methods for the solution of conservation law equations as can be seen in Figure 3.2 due to the discreteness of the grid. In the present work the HLL solver developed by Harten, Lax and van Leer (Harten et al.

1983, Toro, 2010) is employed. As can been seen in Figure 3.3 the cell fluxes $F_{i+1/2}^{n+1/2}$ are expressed in terms of the left going and the right going fluctuations $A^{-}\Delta Q_{i-1/2}$ and $A^{+}\Delta Q_{i+1/2}$:

$$F_{i-1/2} = F_i - A^{+}\Delta Q_{i-1/2} \tag{3.13}$$

$$F_{i-1/2} = F_{i-1} + A^{-}\Delta Q_{i+1/2} \tag{3.14}$$

where $\Delta Q_{i-1/2} = Q_i^L - Q_i^R$.

Substitution of these approximations into equation (3.11) gives:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(A^{-}\Delta Q_{i+1/2} + A^{+}\Delta Q_{i-1/2}\right) \tag{3.14}$$

The terms in brackets on the right-hand side in Equation (3.14) correspond to so-called flux-difference splitting which solves a numerical inviscid flux function for a general system of conservation equation. These terms can be calculated using the HLLC (Harten-Lax-van Leer-Contact) Riemann solver, see (Steiner, 2006; Dullemond & Johansen, 2007; Toro, 2012) for detailed explanation.

For any two given states Q_{i-1} and Q_i the Riemann solver returns the waves $W_{i-1/2}^{p}$ and their propagation speeds $s_{i-1/2}^{p}$ ($p = 1...m$) (LeVeque 2002). The waves $W_{i-1/2}^{p}$ are forced to satisfy the condition:

$$\sum_{p=1}^{m} W_{i-1/2}^{p} = Q_i - Q_{i-1} \equiv \Delta Q_{i-1/2} \tag{3.15}$$

Then, knowing the waves $W_{i-1/2}^{p}$ and the speeds $s_{i-1/2}^{p}$, the left going and the right going fluctuations $A^{-}\Delta Q_{i-1/2}$ and $A^{+}\Delta Q_{i+1/2}$ are calculated as:

$$A^{-}\Delta Q_{i-1/2} = \sum_{p}\left(s_{i-1/2}^{p}\right)^{-} W_{i-1/2}^{p} \tag{3.16}$$

$$A^{+}\Delta Q_{i-1/2} = \sum_{p}\left(s_{i-1/2}^{p}\right)^{+} W_{i-1/2}^{p} \tag{3.17}$$

satisfying Equations (3.13) and (3.14):

$$A^{-}\Delta Q_{i+1/2} + A^{+}\Delta Q_{i-1/2} = F_i - F_{i-1}\ , \qquad (3.18)$$

In Equations (3.16) and (3.17) $s_{i-1/2}^{p-}$ and $s_{i-1/2}^{p+}$ are the speeds of the left- and right- going fluctuations defined as:

$$s_{i-1/2}^{p-} = \min(\ s_{i-1/2}^{p}, 0)$$

$$s_{i-1/2}^{p+} = \max(\ s_{i-1/2}^{p}, 0)$$

In the exact Riemann solver, originally proposed by Godunov (1959), solution to the Riemann problem is employed to calculate $Q_{i-1/2}$ and $Q_{i+1/2}$. However, this requires solution of a nonlinear algebraic equation for each cell interface that makes the method very computationally expensive (R.J. LeVeque, 2002b). Therefore, approximate Riemann solvers as HLLC are being developed (Toro, 2010).

HLL solver builds a numerical approximation for the inter-cell value $Q_{i-1/2}$. A generalised Rankine-Hugoniot conditions (see Lin, 2000 for details) relate the changes in the state and corresponding fluxes through the sub-waves decomposing the jump $Q_l - Q_r$:

$$F_k - F_{k-1} = \sum_{p} s_{k-1/2}^{p}(Q_k - Q_{k-1})\ , \qquad (3.19)$$

where p is the index of sub-waves.

In the original version of HLL solver (Harten et al. 1983), used in the present work, the wave structure is approximated by two sound waves with speeds $s_{k-1/2}^{1} = u - c$ and $s_{k+1/2}^{2} = u + c$. Then equation (3.20) is resolved for the inter-cell value $Q_{i-1/2}$:

$$Q_{i-1/2} = \frac{F_i - F_{i-1} - s_{i-1/2}^{2} Q_i + s_{i-1/2}^{1} Q_{i-1}}{s_{i-1/2}^{1} - s_{i-1/2}^{2}} \qquad (3.20)$$

3.2.2 Boundary conditions and computational algorithm

Boundary conditions are employed to supply information needed at the boundaries of computational volumes for complete definition of the system behaviour. The physics of the problem determine the number of boundary conditions to be imposed. As stated in equation (3.15 – 3.17), at a given boundary, the number of incoming waves at that point must be equal to the number of boundary conditions to be specified.

For the start-up injection case, the constant mass flux subsonic inflow boundary condition for a one-dimensional hyperbolic system considered in this study. Equation (3.4) describes the "conservation properties of the system; such that, it relates the rate of change of the integral of a field over a small volume to the flux of that field across the volume boundaries" (Thompson, 1990). An alternate form for equation (3.4) is the primitive system, with a vector of dependent variables **U**, which satisfies

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{A}\frac{\partial \mathbf{U}}{\partial z} + \mathbf{C} = 0, \qquad 0 < z < 1, \quad t > 0 \tag{3.21}$$

where **A** is *n* x *n* matrix. **C** is a term containing all the terms that are not derivatives of **U**.

In other to satisfy the requirement of the "boundary conditions that the fluid equations be put in characteristic form at the boundaries, it is important to start the boundary specification for fluid dynamics problems by finding the characteristic form for the fluid equations" (Thompson, 1990). For one dimensional flow, the fluid equations can be written in diagonal matrix form from Equation (3.21);

$$\mathbf{U} = \begin{bmatrix} \rho \\ u \\ s \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} u & \rho & 0 \\ \frac{c^2}{\rho} & u & \frac{p}{\rho s} \\ 0 & 0 & u \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \rho u \\ 0 \\ 0 \end{bmatrix}, \tag{3.22}$$

where *s* is a measure of the entropy

Setting the dissipative terms to zero at the boundaries, the eigenvalues of **A** are

$$\lambda_1 = u - c, \qquad \lambda_2 = u, \qquad \lambda_3 = u + c, \tag{3.23}$$

and the left eigenvectors are

$$\mathbf{I_1} = \begin{bmatrix} -\frac{\rho}{sc} \\ \rho \\ -c \end{bmatrix}, \qquad \mathbf{I_2} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{I_3} = \begin{bmatrix} \frac{p}{sc} \\ \rho \\ c \end{bmatrix}, \tag{3.24}$$

The eigenvalue calculation is simplified by taking s as a primitive variable, but it is not suitable for numerical solution. Therefore, s is eliminated using Equation (3.21) in favour of ρ and p and the characteristic equations are presented as (Thompson, 1987)

$$\frac{\partial p}{\partial t} - \rho c \frac{\partial u}{\partial t} + \lambda_1 \left(\frac{\partial p}{\partial r} - \rho c \frac{\partial u}{\partial r} \right) = 0, \tag{3.25}$$

$$\frac{\partial p}{\partial t} - c^2 \frac{\partial \rho}{\partial t} + \lambda_2 \left(\frac{\partial p}{\partial r} - c^2 \frac{\partial \rho}{\partial r} \right) = 0, \tag{3.26}$$

$$\frac{\partial p}{\partial t} - \rho c \frac{\partial u}{\partial t} + \lambda_3 \left(\frac{\partial p}{\partial r} - \rho c \frac{\partial u}{\partial r} \right) = 0 \tag{3.27}$$

where the primitive variable vector **U** has components ρ, u, and p.

For the one-dimensional case considered in the present study, “the interior algorithm requires data at the boundary points which are obtained by solving the combined characteristic and non-reflecting Equations (3.25–3.27) for the boundary calculations.” (Thomson, 1987)

The boundary equations for each cell i are written as

$$\frac{dp_i}{dt} - \rho_i c_i \frac{du_i}{dt} + \mathcal{L}_{1i} = 0, \tag{3.28}$$

$$\frac{dp_i}{dt} - c_i^2 \frac{d\rho_i}{dt} + \mathcal{L}_{2i} = 0, \tag{3.29}$$

$$\frac{dp_i}{dt} - \rho_i c_i \frac{du_i}{dt} + \mathcal{L}_{3i} = 0 \tag{3.30}$$

where each $\mathcal{L}_{ki}$ is set to zero if λ_k at right boundary is directed inward (i.e. non-reflecting condition), otherwise λ_k is directed outward and computed according to the characteristic wave equations:

$$\mathcal{L}_{1i} = (u_i - c_i)[p_{i+1} - p_i - \rho_i c_i (u_{i+1} - u_i)], \quad i < 0, \; u_i - c_i < 0, \tag{3.31}$$

$$= (u_i - c_i)[p_i - p_{i-1} - \rho_i c_i (u_i - u_{i-1})], \quad i > I, \; u_i - c_i > 0, \tag{3.32}$$

$$\mathcal{L}_{2i} = u_i[p_{i+1} - p_i - c_i^2 (\rho_{i+1} - \rho_i)], \quad i < 0, \; u_i < 0, \tag{3.33}$$

$$= u_i[p_i - p_{i-1} - c_i^2 (\rho_i - \rho_{i-1})], \quad i < I, \; u_i > 0, \tag{3.34}$$

$$\mathcal{L}_{3i} = (u_i + c_i)[p_{i+1} - p_i - \rho_i c_i (u_{i+1} - u_i)], \quad i < 0, \; u_i + c_i < 0, \tag{3.35}$$

$$= (u_i + c_i)[p_i - p_{i-1} - \rho_i c_i (u_i - u_{i-1})], \quad i > I, \; u_i + c_i > 0. \tag{3.36}$$

Combining equations (3.25–3.27) and equations (3.31–3.36), the time derivatives of the primitive variables are obtained as

$$\frac{dp_i}{dt} = -\frac{1}{2}(\mathcal{L}_{3i} + \mathcal{L}_{1i}) \, , \tag{3.37}$$

$$\frac{du_i}{dt} = -\frac{1}{2\rho_i c_i}(\mathcal{L}_{3i} - \mathcal{L}_{2i}) \, , \tag{3.38}$$

$$\frac{d\rho_i}{dt} = \frac{1}{c_i^2}\left(\frac{dp_i}{dt} + \mathcal{L}_{2i}\right) . \tag{3.39}$$

Considering a constant mass flux subsonic inflow at the boundary which describes the inflow of CO2 into the injection well, then

$$\frac{d(\rho u)}{dt} = \rho\frac{\partial u}{\partial t} + u\frac{\partial \rho}{\partial t}. \tag{3.40}$$

Substituting equations (3.37 – 3.39) in (3.40) gives

$$2u\mathcal{L}_2 - \mathcal{L}_3(u-c) - \mathcal{L}_1(u+c) = 2\rho c^2 u \tag{3.41}$$

If the second wave $\mathcal{L}_2$ is specified, then

$$\mathcal{L}_3 = \frac{1}{(u+c)}[2\rho c^2 u - 2u\mathcal{L}_2 - \mathcal{L}_3(u-c)], \quad 0 < u < c \tag{3.42}$$

and

$$\mathcal{L}_3 = \frac{1}{(u-c)}[2\rho c^2 u - 2u\mathcal{L}_2 + \mathcal{L}_1(u+c)], \quad -c < u < 0 \tag{3.43}$$

Figure 3.4 shows the flow chart of algorithm used to solve numerically the set of discretised conservation equations, using an explicit method of time integration. For practical computations, the algorithm was coded using FORTRAN 1995/2003 language see Chapman, (2003). Due to unavailability of experimental data the results obtained were compared with those presented by Li et al., (2015) from their 1D+1D model and Shell, (2015) using OLGA (OLGA 7 User manual, 2010) software for wellbore dynamics.

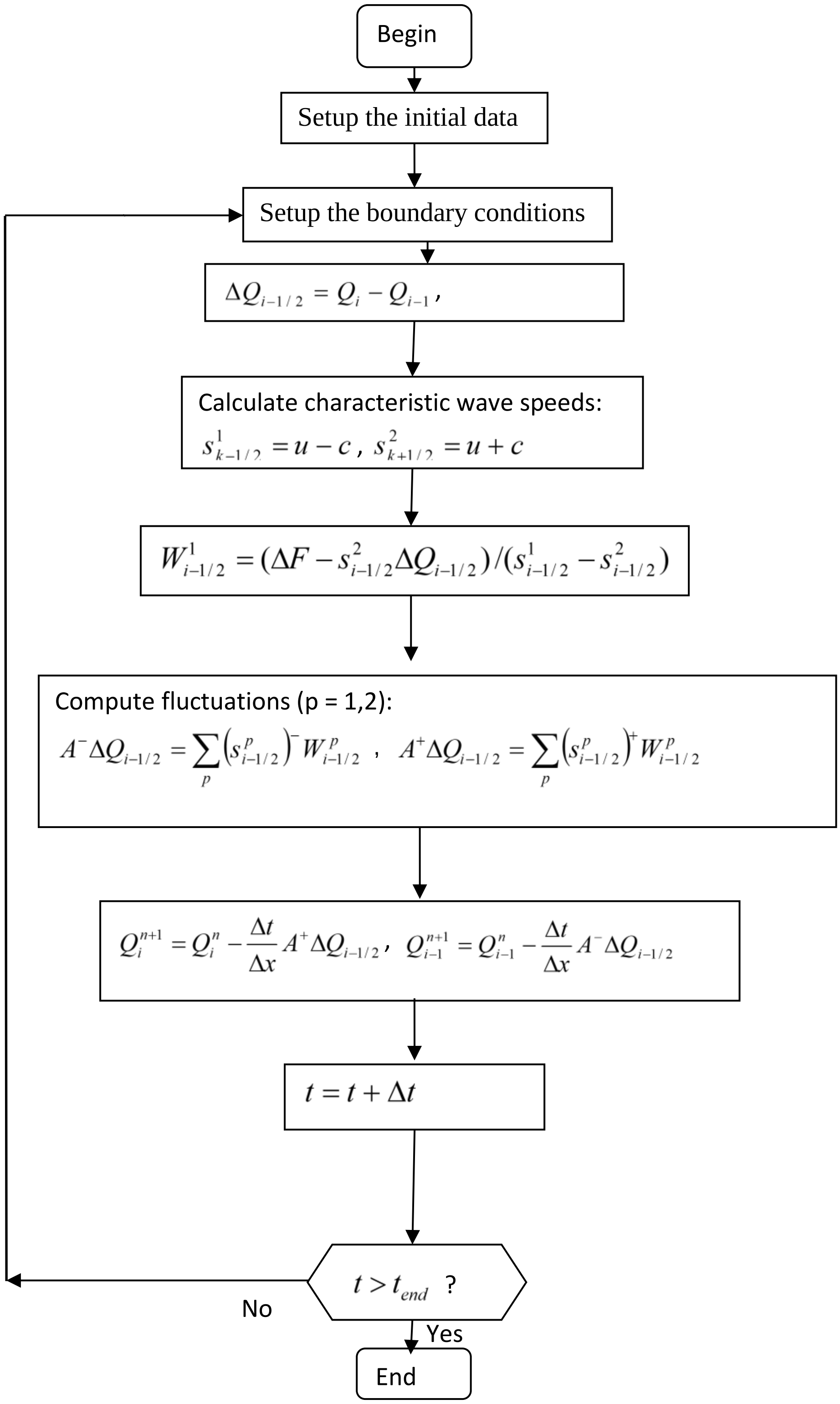

Begin
Setup the initial data
Setup the boundary conditions
$\Delta Q_{i-1/2} = Q_i - Q_{i-1}$,
Calculate characteristic wave speeds:
$s^1_{k-1/2} = u - c$, $s^2_{k+1/2} = u + c$
$W^1_{i-1/2} = (\Delta F - s^2_{i-1/2}\Delta Q_{i-1/2})/(s^1_{i-1/2} - s^2_{i-1/2})$
Compute fluctuations (p = 1,2):
$A^-\Delta Q_{i-1/2} = \sum_p (s^p_{i-1/2})^- W^p_{i-1/2}$, $A^+\Delta Q_{i-1/2} = \sum_p (s^p_{i-1/2})^+ W^p_{i-1/2}$
$Q^{n+1}_i = Q^n_i - \frac{\Delta t}{\Delta x} A^+\Delta Q_{i-1/2}$, $Q^{n+1}_{i-1} = Q^n_{i-1} - \frac{\Delta t}{\Delta x} A^-\Delta Q_{i-1/2}$
$t = t + \Delta t$
$t > t_{end}$?
No
Yes
End

Figure 3.4: Flowchart of an algorithm used to solve the flow equations using the HLL scheme by Harten et al. (1983) and Toro, (2010).

3.3 Simulation results and discussion

The results obtained from the simulation of the start-up CO_2 injection study are described in the following to show the importance of understanding the transient Joule-Thomson cooling effect behaviour of CO_2 during start-up injection. For the case considered, the well geometry and parameters for the Goldeneye CCS project presented in Li et al, (2015) were used as basis for simulation and comparison.

The HEM model results presented below show the impact of CO_2 start-up injection operations on wellhead pressure and temperature, and the flow of the CO_2 stream down the injection well. In a real CCS project however, the transient behaviour of CO_2 during well start-up will be observed based on the differences in the injection pressure, the injection temperature, wellbore depth, the reservoir pressure and the injection flow rate. The start-up CO_2 injection analysis is vital to predicting an optimum injection strategy for large-scale CO_2 sequestration.

3.3.1 Start-up CO_2 injection case study

The model input parameters are adapted to mimic those of Goldeneye CCS project injection well conditions and outlined in Table 3.1. However, for simplicity and initial test run, the input conditions are mainly assumed at this point for purposes of testing the model performance. The full Goldeneye injection well conditions will be applied in subsequent chapters. Based on the assumed conditions in Table 3.1, the simulation results for pressure, temperature and density profiles of CO_2 in the tubing at different depth and times are respectively presented below.

Table 3. 1: Goldeneye injection well and CO_2 inlet conditions (Shell UK, 2015)

Input parameter	Value
Wellhead pressure, bara	36.5
Wellhead temperature, K	280
Bottom-hole pressure, bara	82
Bottom-hole temperature, K	296
Well depth, m	2500
CO_2 injection rate, kg/s	38
Injection tube diameter, m	0.125

CO_2 inlet pressure, bara	50
CO_2 inlet temperature, K	277

Based on the assumed data the wellhead pressure is maintained at 36.5 bara, and the wellhead temperature is at thermal equilibrium with its surroundings at 7 ^{o}C. The interface between the liquid-gaseous CO_2 along the injection well during start-up depends on the reservoir pressure, well depth, and the temperature conditions of the surrounding formation. Since the hydrostatic pressure condition is the differential factor between the variations in wellhead pressure and bottom-hole pressure, it is likely that with relatively low reservoir pressure at an injection well, the wellhead pressure at start-up injection may be less than the saturation pressure corresponding to it under ambient temperature. As a result of the lower pressure condition at the wellhead, more gaseous CO_2 is present near the wellhead in the tubing during well start-up injection.

The stability of this numerical scheme was tested using CFL condition ranging from 0.2 to 0.8, along with discretisation distance Δz, 50 to 400 mm and time steps Δt between $1x10^{-8}$ to 0.05 seconds. The **Courant–Friedrichs–Lewy (CFL) condition** is a condition necessary for solving certain partial differential equations (usually hyperbolic PDEs) to attain numerical convergence (Mahgerefteh, et al., 2009). This condition is usually employed for the numerical solution of explicit time integration schemes and numerical analysis. As a result, a certain time step is maintained to allow an explicit time-marching computer simulation in order to avoid incorrect simulation results. Hence, the results obtained using CFL condition 0.2 to 0.4, Δz 100 to 200 mm and Δt of $1x10^{-8}$ to $1x10^{-3}$ seconds show better accuracy and numerical stability.

Figures 3.5 and 3.6 respectively show the injection well pressure and temperature profiles at 0, 100, 300 and 500 simulation seconds. As can be seen in Figure 3.5, the bottom-hole pressure gradually increases from the reservoir static pressure (82 bara) to about 110 bara after well start-up for 100 seconds. This means that the bottom-hole pressure increases with time as more CO_2 is injected into the well resulting to a corresponding increase in the wellhead pressure due to hydrostatic pressure build-up. Consequently, there is a possibility of back-flow or blowout over time due to the continuous increase in the bottom-hole pressure as the predicted well pressure profiles at 100, 300 and 500 seconds show a consistent rise in the bottom-hole pressure. Hence, there is need for corresponding increase in the injection pressure with time in order maintain continuous well injection. Consequently, this means that a pressure sensor at the wellhead should be enabled to send signal to the compressor or pump control valve to increase the inlet pressure accordingly.

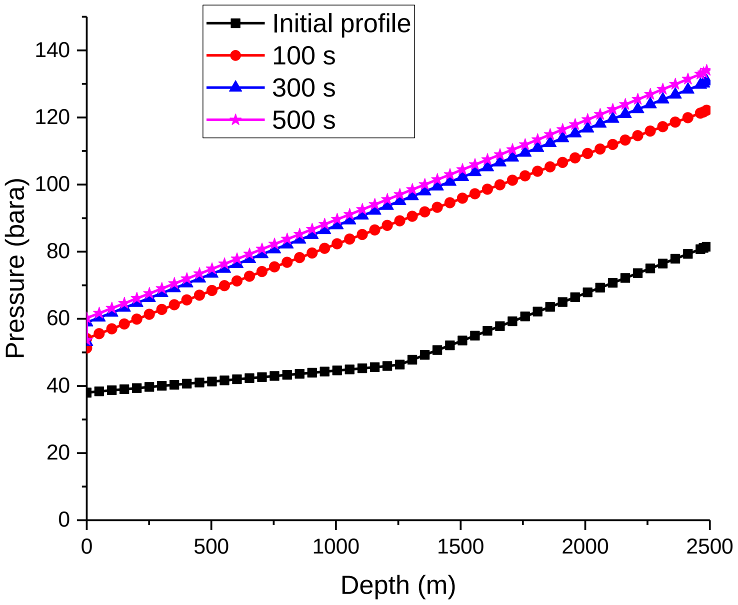


Figure 3.5: Wellbore pressure profiles of CO_2 stream at different simulation times

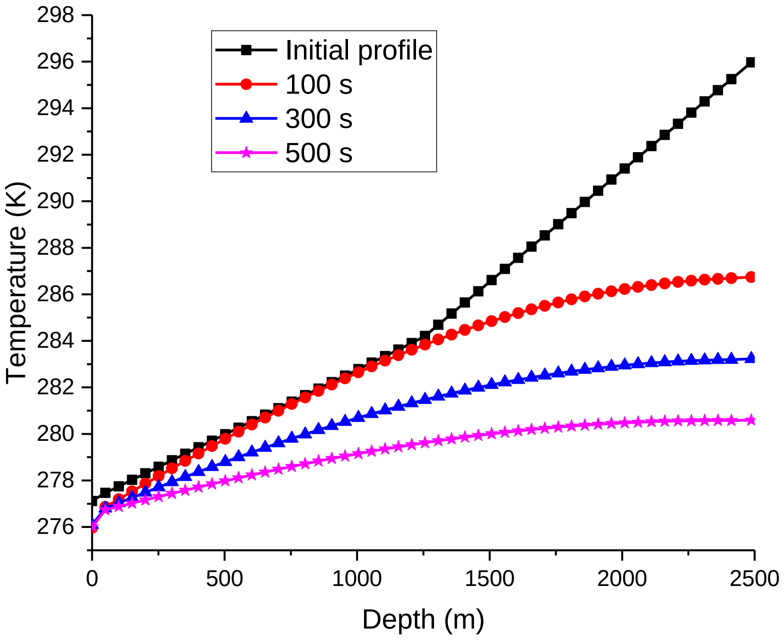


Figure 3.6: Wellbore temperature profiles of CO_2 stream at different simulation times

Figure 3.7 shows the injection well density profiles at 5, 20 and 100 simulation seconds. As can be seen in Figure 3.6, the bottom-hole temperature gradually decreases during start-up injection from initial 296 K to about 286 K after 100 seconds. Also, the temperature at the wellhead (at 0 m depth) drops significantly within the first 100 sec due to the Joule-Thomson cooling effect of the expanding CO_2. Notably, there is a continuous decrease in temperature at the reservoir end from 296 K to 286 K, 282.5 and 281 K after 100, 300 and 500 sec respectively. This decrease can be attributed to the heat exchange between the surrounding formation and the incoming CO_2 stream. However, as CO_2 pressure increases along the wellbore (see Figure 3.5 pressure profile) it tends to get colder and denser with such increasing pressure (see Figure 3.6 temperature profile), as heat is lost to the surrounding formation and then the temperature drops as it approaches the reservoir end. There is therefore an indication that the phenomenon of heat exchange and frictional losses may have significant influence on the pressure and temperature profiles as can be seen in Figure 3.5 and 3.6.

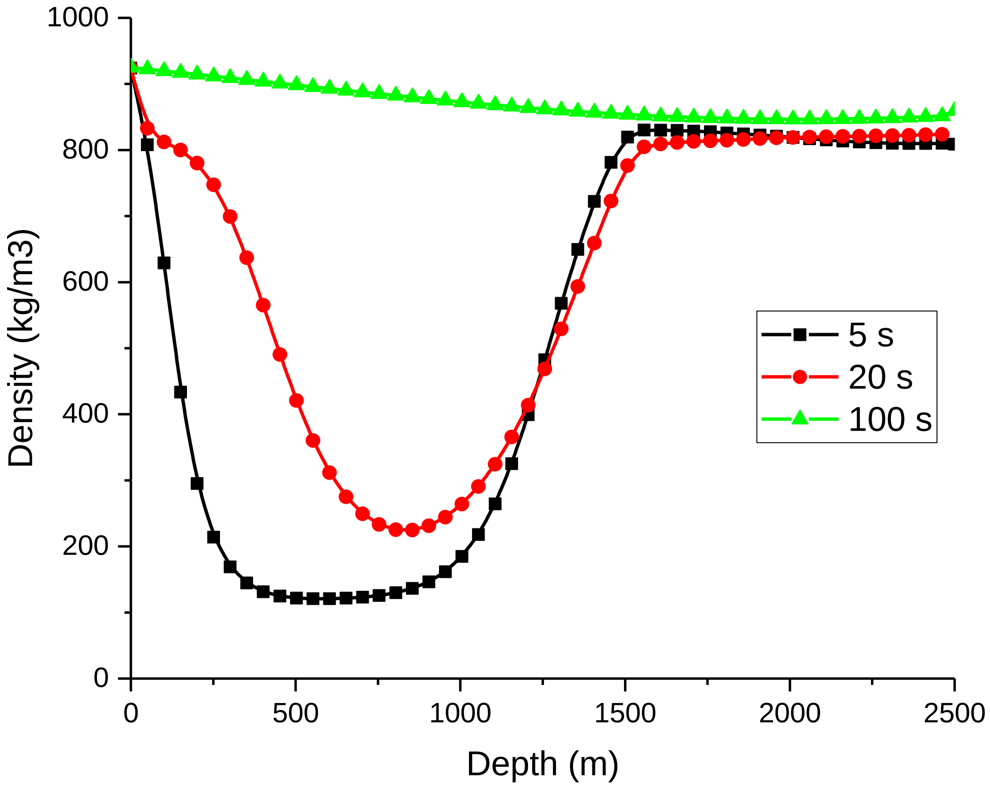


Figure 3.7: Wellbore density profiles of CO_2 stream at different simulation times

The density profile in Figure 3.7 shows a clearer description of the CO_2 stream behaviour going down the injection well. The profiles after 5 and 20 seconds of simulation show a sudden decrease at about 400 m and later increases at about 1200 m down the injection well which likely corresponds to a possible phase transition. As can be seen in Figure 3.7, CO_2 stream at the start of injection dropped into lower density phase and stabilises after 100 sec. The CO_2 density rises significantly with well depth showing higher composition of denser stream composition down the well. Due to the higher pressures and temperatures at the bottom-hole, the CO_2 stream is likely to arrive the injection well (reservoir) as a supercritical (dense) or liquid-phase fluid with high density. Figure 3.8 shows CO_2 phase diagram for density profile of various regions where the liquid region has higher densities followed by the supercritical region. At higher densities above 900 kg.m^{-3} and temperatures above 273 K the CO_2 stream is likely to be in the supercritical or liquid phase.

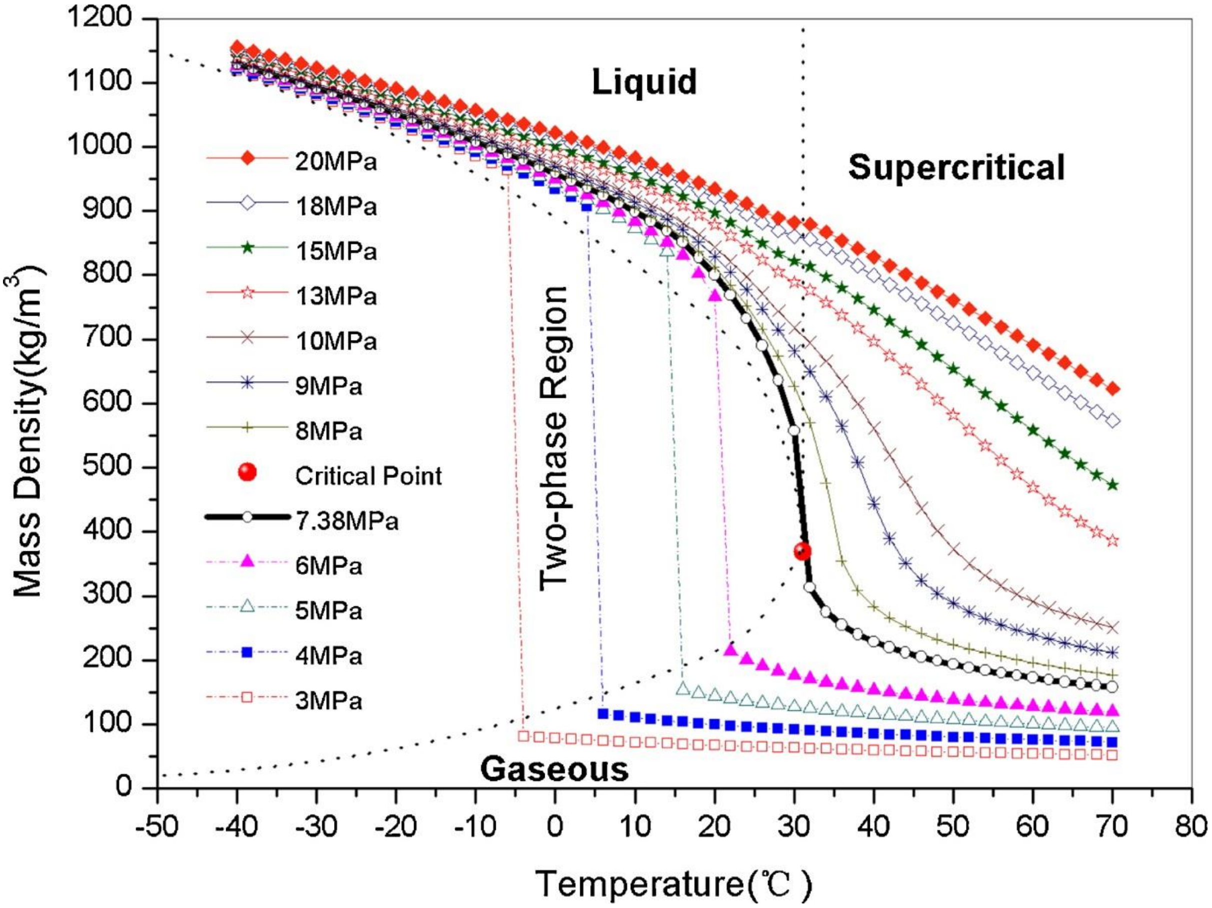


Figure 3.8: CO_2 phase diagram showing densities at different regions (Zhao & Li, 2014)

Furthermore, the well pressure, temperature, density and vapour mass fraction profiles at different locations; wellhead, 400 m and 800 m depth are plotted in Figure 3.9 to 3.11 respectively.

Figure 3.9 shows the injected CO_2 stream pressure profiles at 0, 400 and 800 m well depth. The data shows a sharp depressurisation at the start of injection due to the pressure difference between the incoming CO_2 and the wellhead pressure. The incoming CO_2 (at 50 bara) expands upon arriving at the lower pressure (at 36.5 bara) wellhead. The result predicted shows a drop in pressure within the first 10 to 50 seconds after which the pressure starts building up due to the hydrostatic condition and possible minimal frictional losses encountered by the fluid. The start-up injection test case at the wellhead showed a significantly low pressure was near 30 bara from the initial inlet pressure of 50 bara. The major factor responsible for the sharp pressure drop of CO_2 at the wellhead upon injection is the Joule-Thomson expansion effect.

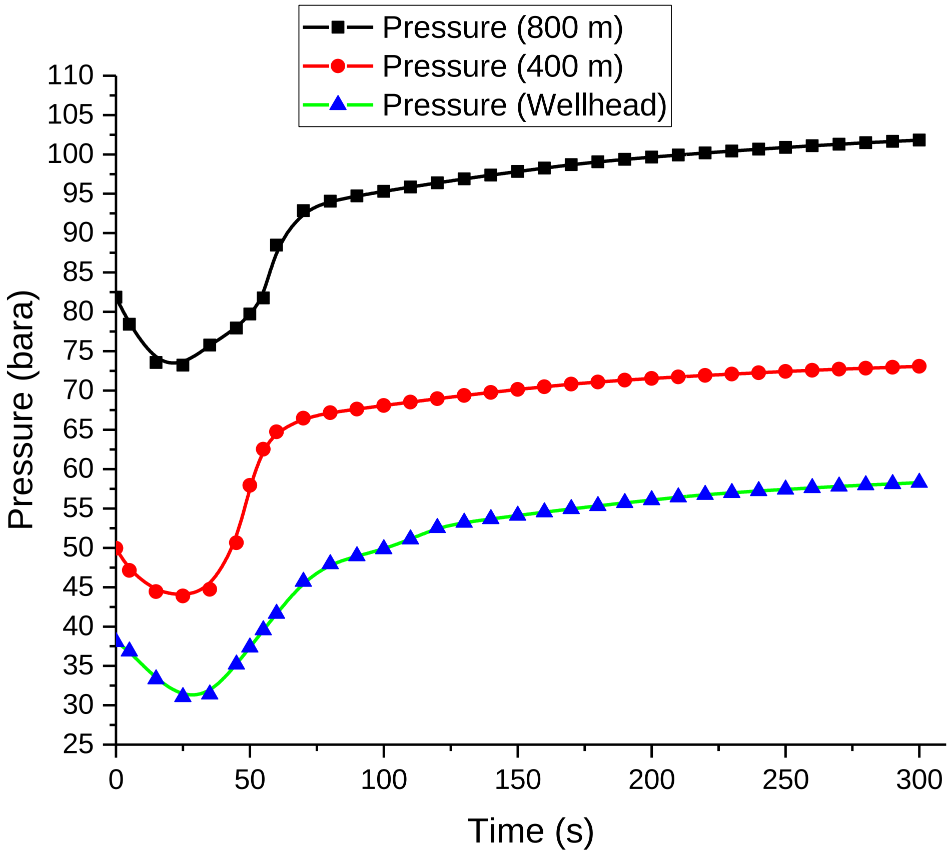


Figure 3.9: CO_2 stream pressure profiles at different well depths

Figure 3.10 shows the injected CO_2 stream temperature profiles at 0, 400 and 800 m well depth. The temperature profiles also follow a similar pattern like that of the pressure as can be seen in Figures 3.9 and 3.10. However, greater temperature drop is predicted at the wellhead where the CO_2 stream experienced a drastic expansion upon arriving at a lower pressure region. Such expansion is accompanied by a significant temperature drop induced by Joule-Thomson cooling effect on an expanding gas as the dense-phase CO_2 enters the injection well. Hence, the most significant cooling takes place at and near the wellhead during start-up operations compared with 400 and 800 m down the well. Notably, the results show a possibility of a significant drop in temperature below the freezing point of water (i.e. 273 K) which poses serious safety concerns for large-scale CO_2 sequestration projects. The presence of interstitial water molecules in the wellbore that come in contact with the injected CO_2 may form hydrates or ice which could block the injector inlet and cause severe operational challenges.

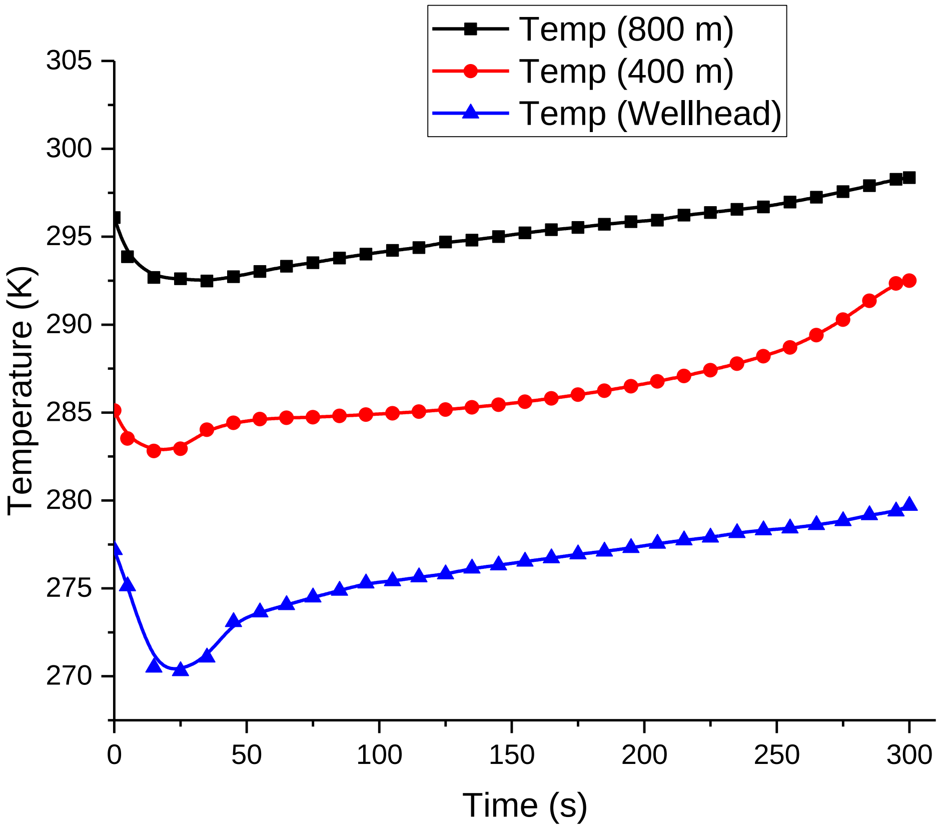


Figure 3.10: CO_2 stream temperature profiles at different well depths

Figure 3.11 shows the injected CO_2 stream density as function of time at varying well depths (0, 400 and 800 m). At the wellhead (0 m) , the stream density drops significantly within the first 20 sec due to sudden decrease in pressure and temperature (as can be seen in Figures 3.9 and 3.10) of the CO_2 stream. This is a consequence of a definite phase transition from the dense to the 2-pahse region as can be seen in Figure 3.8. The density later increases down the injection well and at 400 m the stream density is higher than at the wellhead as shown in Figure 3.11. Also, as more CO_2 is injected into the well, the density profile stabilises and attains somewhat steady-state. The CO_2 density at depth up of 800 m only marginally decreases with injection time in a relatively constant manner.

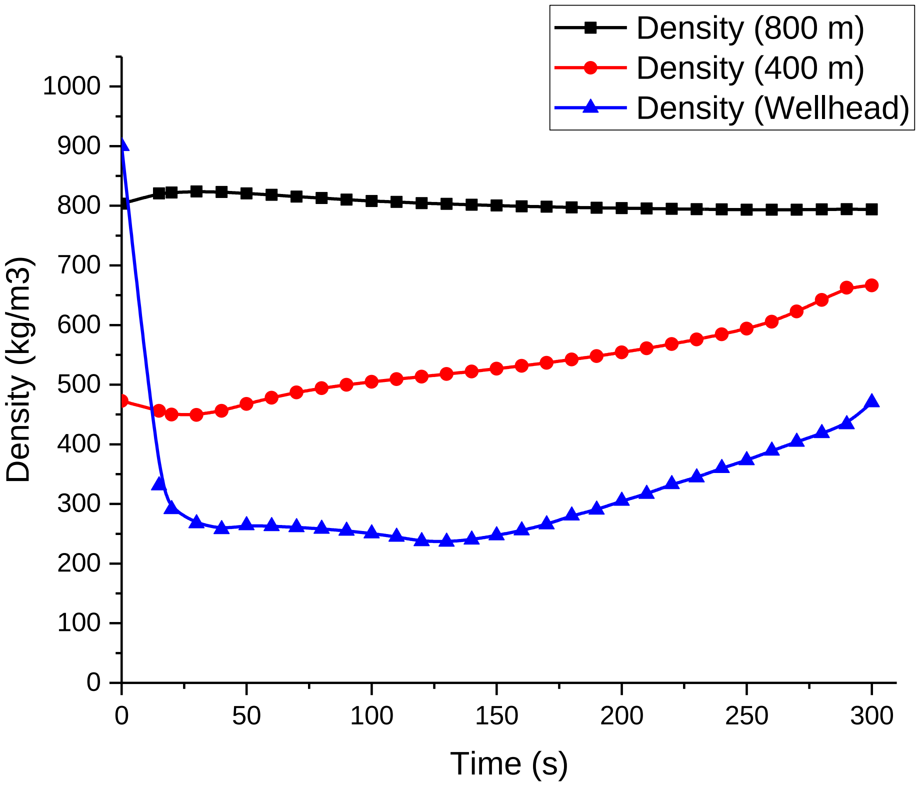


Figure 3.11: CO_2 stream density profiles at different well depths

Figure 3.12 shows the injected CO_2 stream vapour mass fraction as a function of well depth at 5, 20 and 100 simulation time. Notably, the vapour mass fraction over time shows CO_2 expansion upon arriving at the wellhead indicated by a rise in vapour mass fraction followed by a rapid drop as more fluid is injected. Such expansion is followed by rapid cooling which reduces the vapour composition quickly and this behaviour agrees with the previous observation (in Figure 3.10) of rapid temperature drop at the wellhead. The vapour fraction profile after 20 s of simulation shows lower vapour composition in the CO_2 stream compared with the profile after 5 s of simulation. However, it is noteworthy that the vapour mass fraction from wellhead to the bottom of the well remains constant after 100 s simulation time. This means that after 100 s of simulation time the CO_2 stream pressure and temperature tends to be in the liquid or dense phase where the vapour phase slowly disappears as can be seen in the phase diagram (Figure 3.8).

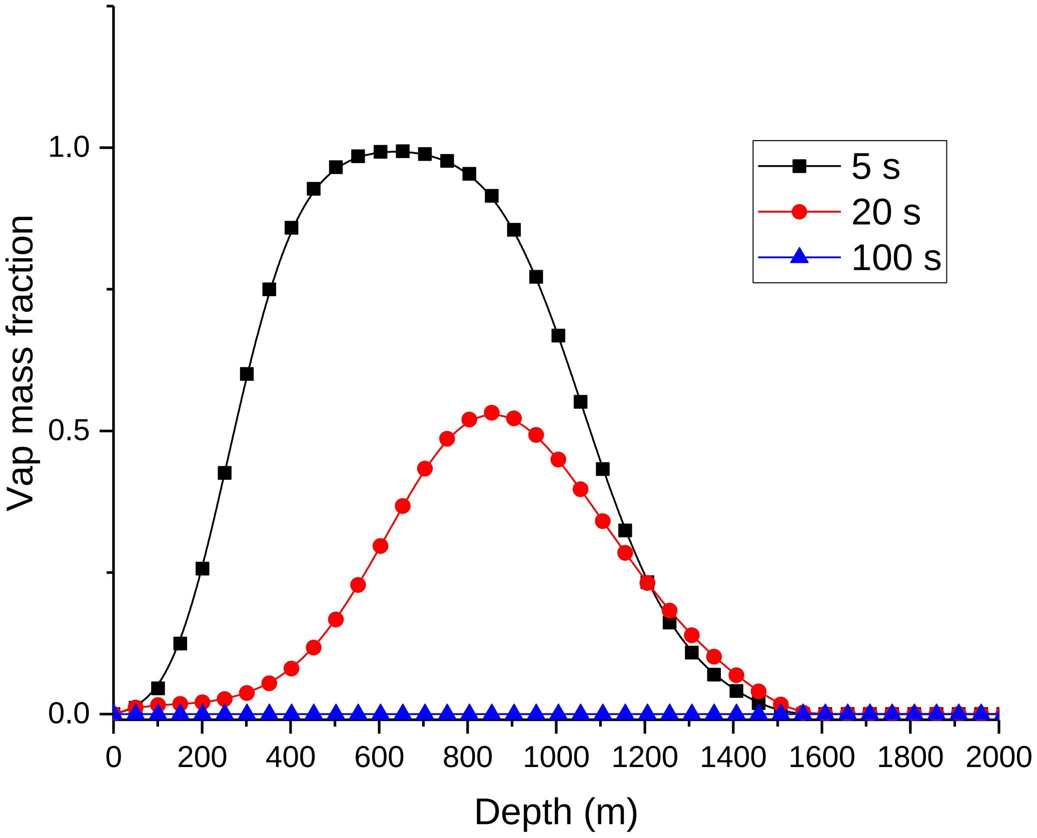


Figure 3.12: Vapour mass fraction of CO_2 stream at different well depths

3.3.2 Simulation results comparison with published data

The simulation results of the current HEM model show trends similar to those published in the literature. As mentioned in section 2.3 of chapter 2, Li et al, (2015) developed a 1D+1D OLGA based simulation model for wellbore dynamics. The model was employed to predict the decrease in pressure and temperature of CO_2 at the wellhead, as well as CO_2 phase behaviour in the wellbore during well transient operation. The CO_2 injection and well conditions used by Li et al, (2015) are constant injection rate 38 kg/s and injection inlet pressure and temperature 50 bara and 4°C respectively and well bore id 0.125 m. Based on these parameters, the current HEM model and Li et al, (2015) OLGA model simulation results are compared.

Figure 3.13 shows the comparison of CO_2 stream pressure profiles of Li et al, (2015) OLGA model and the current HEM model simulations. As can be seen in Figure 3.13, the wellhead pressure profile obtained using the current HEM model simulation shows very similar trend with the profile obtained by Li et al, (2015) after 300 seconds. Both results show an initial depressurisation from the inlet pressure (50 bara) to a value below the wellhead pressure (36.5 bara) before increasing due to continuous injection and hydrostatic pressure build-up from the bottom-hole. Notably, the wellhead pressure remained at a value below the threshold injection pressure (50 bara) in the current HEM model simulation whereas Li et al, (2015) predicted a value higher (up to 12 bara) than the injection inlet pressure showing the need for continuous increase in the injection pressure.

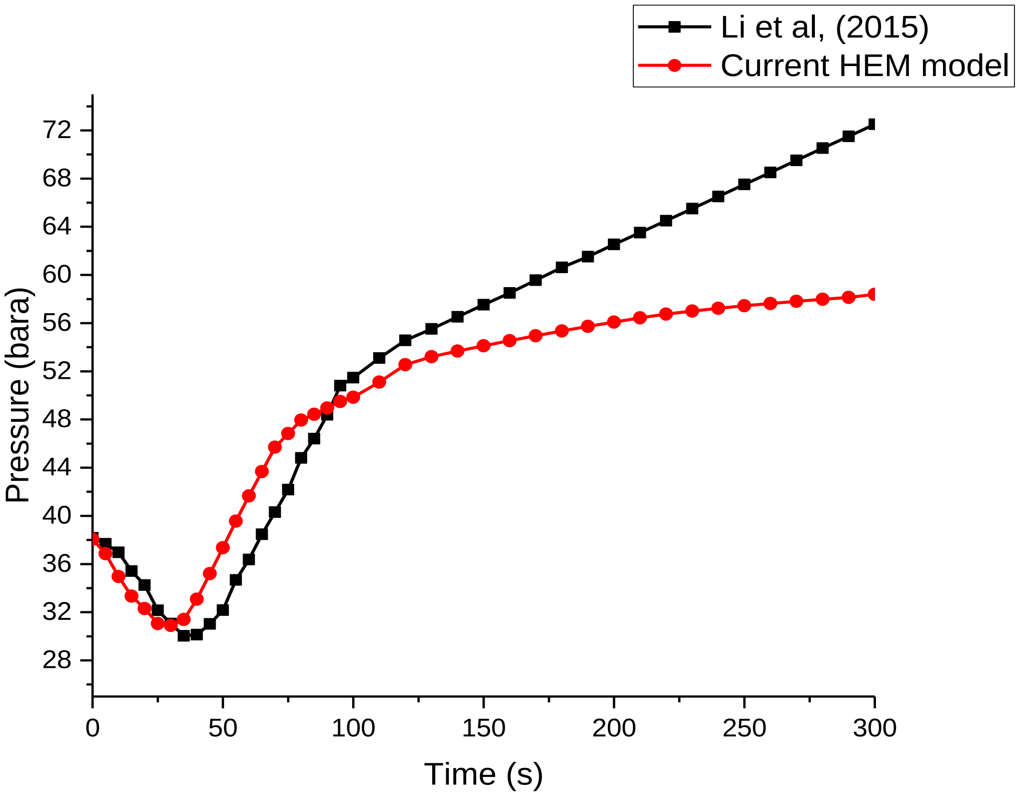


Figure 3.13: Comparison of CO_2 stream wellhead pressure profiles

Figure 3.14 shows the comparison of CO_2 stream temperature profiles of Li et al, (2015) OLGA model and the current HEM model simulations. Comparing the temperature profiles of the current HEM model simulation and Li et al, (2015) OLGA model simulation, the trend is very similar as both predict significant temperature drops. The gaseous CO_2 expansion induced temperature drop at the wellhead due to Joule-Thomson cooling effect is observed in Figure 3.14. Notably, as can be seen in Figure 3.14 Li et al, (2015) predicts a lower temperature drop (about 3 ^{o}C lower) than that obtained using the current HEM model. This variation may be due to the different EoS employed in predicting the thermodynamic properties of CO_2. Li et al., (2015) used Span and Wagner (Span and Wagner, 1996) EoS to calculate the density and the specific heat of CO_2, and Soave-Redlich-Kwong (Pedersen et al., 1989) equation was used for calculating the viscosity and thermal conductivity of CO_2. In the current model, the Peng-Robinson EoS (Peng & Robinson, 1976) is employed to determine the phase equilibrium and all the thermodynamic properties for CO_2. The different EoS used may over-predict or under-predict some properties leading to high or low cooling effects and corresponding temperature drops. However, accuracy and right conclusions can only be established when experimental data are available for model validation.

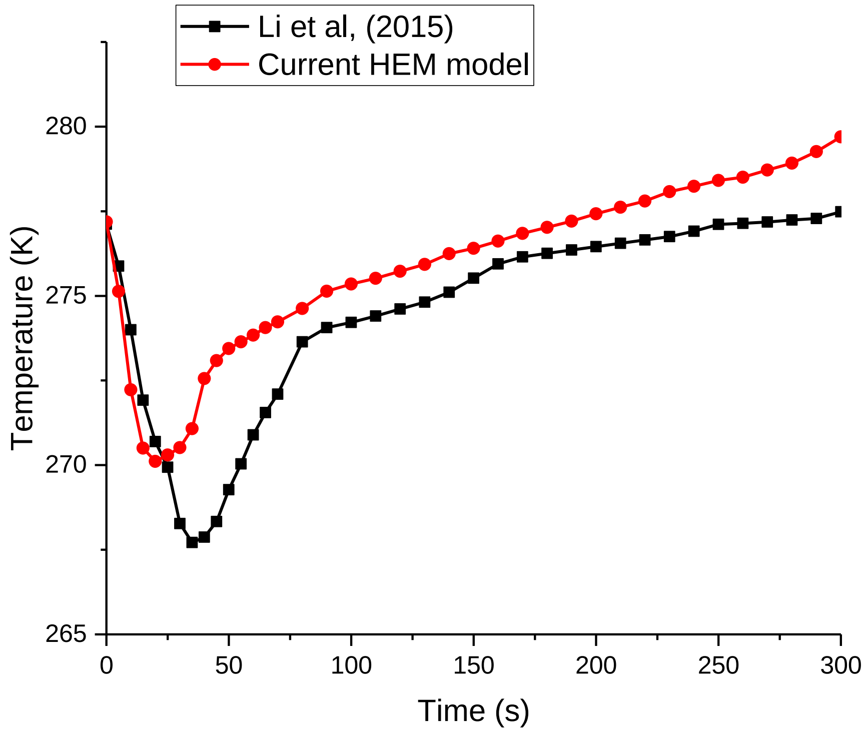


Figure 3.14: Comparison of CO_2 stream wellhead temperature profiles

3.4 Model Validation

The model relevance and applicability to real-world injection project is validated using Ketzin pilot site experimental data. The model predictions are closely in agreement with the real-life CO_2 injection scenario.

CO_2 injection well initial and boundary conditions for Ketzin pilot site Brandenburg, Germany obtained from Möller et al, (2014) employed for the model validation are:

- CO_2 inlet pressure 57 bar, temperatures 10 ^{o}C and 20 ^{o}C, and injection mass flow rate 0.41 kg/s
- Initial wellhead pressure 48 bar and temperature 10 ^{o}C,
- Total well depth 550m; 0.0889m internal diameter
- Initial bottom-hole pressure 68 bar and temperature 33 ^{o}C,

As can be seen in Figs. 3 and 4, the simulation results for 10^{o}C and 20^{o}C injection inlet temperature condition all showed good agreement with the experimental data. The performance of our model in relation to the Ketzin pilot real-world experimental injection project shows it reliability and applicability.

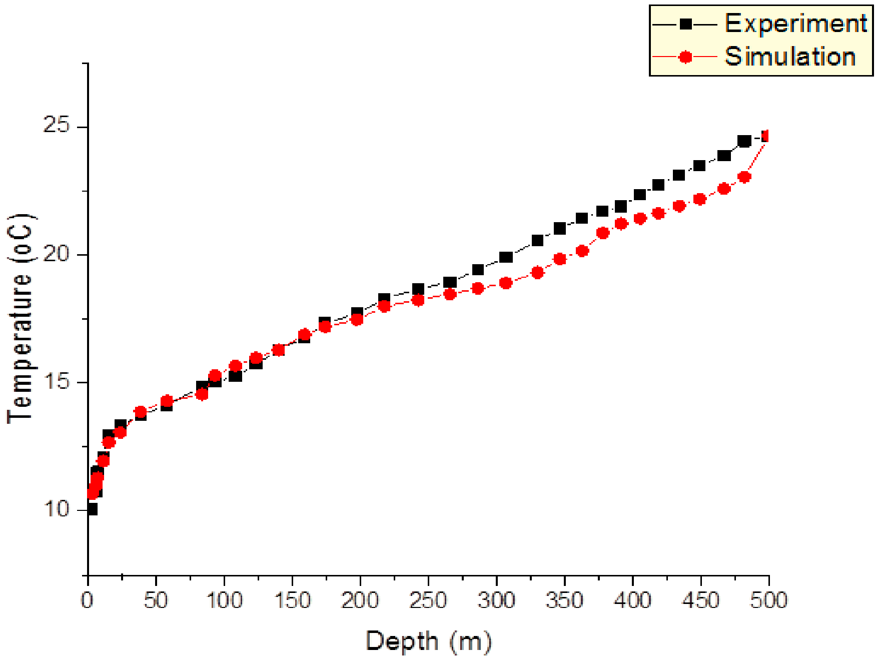


Figure 3.15: Well temperature profile for 10^{o}C inlet CO_2 injection temperature

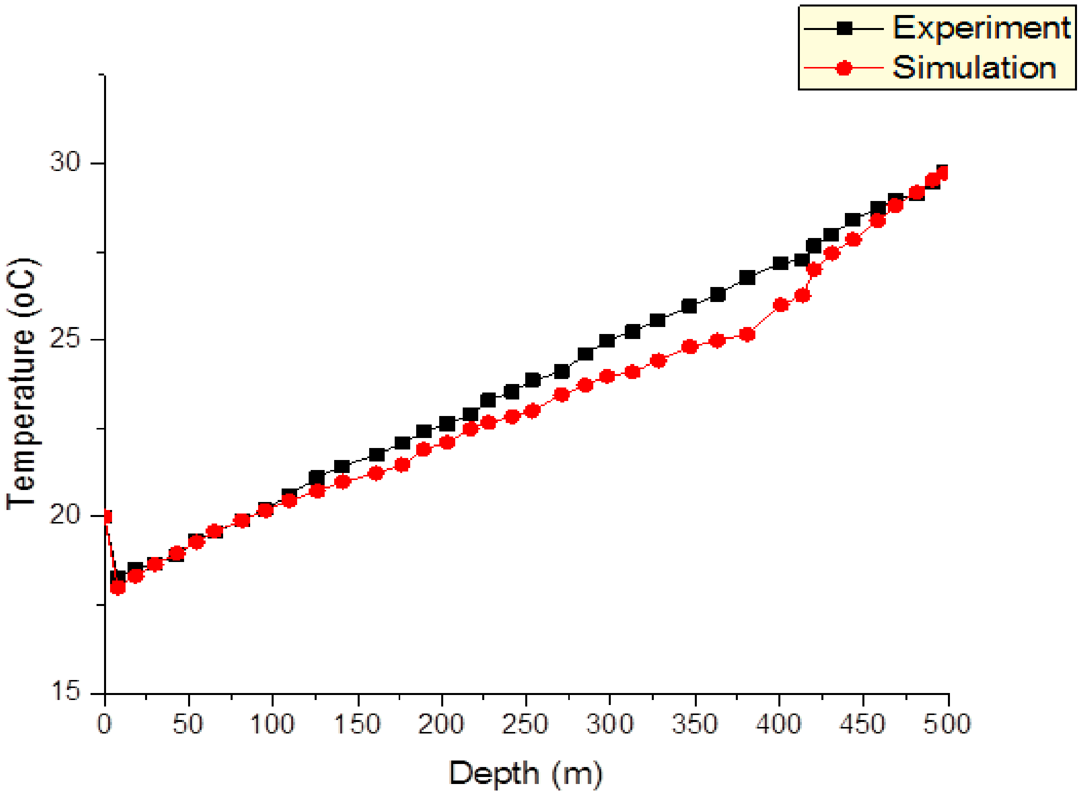


Figure 3.16: Well temperature profile for 20°C inlet CO_2 injection temperature

The model was further tested to establish an optimum start-up injection condition by alternating some parameters. The key parameter monitored was the lowest wellhead temperature while varying the injection flow rate, injection inlet pressure and temperature conditions. Table 3.2 shows the lowest wellhead temperatures obtained by Li et al, (2015) OLGA model and the current HEM model at different injection flow rate, inlet pressure and temperature during start-up. As can be seen in Table 3.2, the analysis shows that the wellhead temperature is lower when the injection flow rate and injection inlet temperature is decrease. On the hand, increasing the injection inlet pressure also results in lower wellhead temperature.

Table 3. 2: Lowest wellhead temperature at different injection rate and inlet pressure during start-up

Case	Injection pressure (bara)	Injection rate (kg/s)	Injection temperature (◦C)	Lowest wellhead temperature (◦C)	
				Li et al, (2015)	Current Simulation
1	50	38	4	-6.8	-3.2
2	50	38	-4	-10.2	-7.3
3	50	26	4	-9.4	-5.8
4	60	38	4	--	-4.2
5	60	38	-4	--	-8.6

3.5 Concluding remarks

This chapter first presented the Finite Volume Method (FVM) numerical solution to the HEM model. The application software (CLAWPACK) was presented with the applied boundary conditions for the start-up injection CO_2 into depleted gas fields. The FVM numerical solution followed the flux-difference splitting scheme which solves a numerical inviscid flux function for a general system of conservation equation. The start-up injection of CO_2 at the top of the well was described by the constant mass flux subsonic inflow boundary condition for a one-dimensional hyperbolic system. The model results obtained showed the impact of CO_2 start-up injection operations on wellhead pressure and temperature, and the consequent flow of the CO_2 stream down the injection well. The simulation results of the current HEM model show good agreement with those published in literature. The start-up CO_2 injection analysis is vital to predicting an optimum injection strategy for large-scale CO_2 sequestration.

Based on the findings from the test case presented in this chapter, the key findings are summarised as follows:

- the degree of cooling along the injection well becomes more severe with increasing injection duration. Following the rapid Joule-Thomson cooling effect at the wellhead when the higher pressure CO_2 is introduced, the cooling rate then decreases with well depth;

- the formation of ice is likely during the injection process given that the minimum fluid temperature falls well below 0 ^oC. This poses the risk of well blockage in the event that sufficient quantity of water is present;

- the minimum simulated CO_2 temperature and the corresponding pressures predicted along the well during the start-up injection process are well outside the ranges where CO_2 hydrate or CO_2 solid formation would be expected; The ideal hydrate formation conditions are pressures and temperatures below 273.15 K and 12.56 bara however, tiny molecules of CO_2 hydrates begin formation at pressures and temperatures just below 283 K and 49.99 bara (Circone et al., 2003; Yang et al., 2012).

- given the observed relatively modest drop in temperature along the well, its failure due to thermal shocking during the injection process is highly unlikely;

- also the wellhead pressure and temperature profiles obtained in the current simulation show good agreement with those published in the literature.

Chapter 4: Modelling Well Start-up Injection and Blowout of CO_2

4.1 Introduction

In this chapter, the HEM model for the start-up injection presented in the chapter 3 is adapted to simulate for both start-up and sudden blowout scenarios leading to outflow at the wellhead and inflow at the bottom of the well. An outflow at the wellhead also referred to as blowout from an injection well can occur under varying circumstances such as underground fluid movement resulting to pressure build-up due to the hydrostatic condition. In the initial state, the CO_2 injection well is in contact with the bottom of the reservoir and the top is assumed to be closed. The initial condition in the well usually consists of a dense phase fluid in the lower part of the well, and vapour phase in the upper part of the well. The well pressure is assumed to be under hydrostatic conditions, however for the blowout case the well pressure is extrapolated from the reservoir pressure at the bottom of the well. At the time of injection, the well head is assumed to be opened to atmospheric pressure. At such, should the flow reach sonic velocity, it becomes choked and may impose the choke pressure at the outlet, rather than the atmospheric pressure which will cause the CO_2 to flow from the reservoir into the injection well. In this chapter, two realistic case studies are considered. The first case study compares pressure and temperature changes during start-up injection and blowout prior to injection while the second case study compares blowout prior to injection and blowout after 5 mins of injection. The boundary conditions are first presented and are next implemented as part of the mathematical modelling for the closure equations for the HEM model presented in chapter 3. The resulting pressure and temperature profiles for the wellhead and bottom of the well are also presented and discussed.

4.2 Start-up injection and blowout boundary conditions

Setting appropriate boundary conditions at the top and bottom of the well as part of the finite set of grid cells covering the computational domain is important given that these govern the flow profiles during injection and blowout. The first and the last numerical discretisation cells will not have the required neighbouring information on the left and right respectively. Therefore, in order to close the flow equations, relevant boundary conditions are added using a ghost cell at either end of the well. Figure 4.1 shows a schematic representation of the isenthalpic inflow condition in the ghost cell at the wellhead. Based on Figure 4.1, the pressure, enthalpy and mass flowrate in the ghost cell will equate those in the first computational cell at time, $t - \Delta t$. This follows the analysis of time-dependent boundary conditions for subsonic inflows described by Thompson, (1987, 1990).

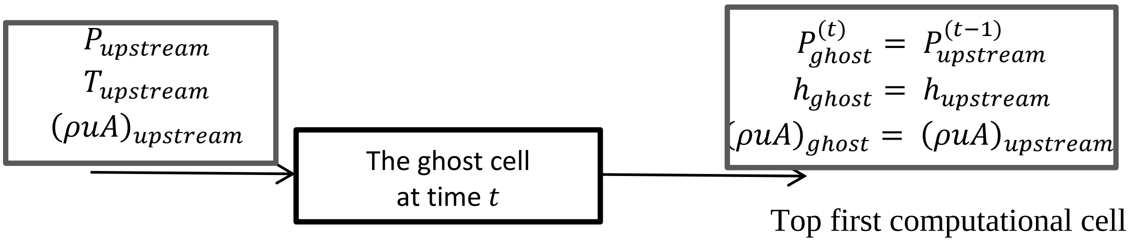

$P_{upstream}$
$T_{upstream}$
$(\rho uA)_{upstream}$
The ghost cell at time t
$P_{ghost}^{(t)} = P_{upstream}^{(t-1)}$
$h_{ghost} = h_{upstream}$
$(\rho uA)_{ghost} = (\rho uA)_{upstream}$
Top first computational cell

Figure 4.1: Schematic representation of the isenthalpic inflow condition in the ghost cell at the wellhead.

At the bottom of the well an empirical pressure-flow relationship derived from reservoir properties (Li et al, 2015; Shell, 2015) is employed:

$$\tilde{A} + \tilde{B} \times M + \tilde{C} \times M^2 = P_{BHP}^2 - P_{res}^2 \tag{4.1}$$

where

$\tilde{A}$ is the minimum pressure required for the flow to start from the well into the reservoir, Pa^2

$\tilde{B}$ and $\tilde{C}$ are site-specific dimensional constants, Pa^2.s/kg and Pa^2.s ^{2}/kg^2 respectively

M is the instantaneous mass flow rate at the bottom-hole, kg/s

P_{BHP} is the instantaneous bottom-hole pressure, bara and

P_{res} is the reservoir static pressure, bara.

The right hand side of Equation (4.1) represent the pressure differential between the bottom of the well and the reservoir. Significantly, when $P_{BHP}^2 > P_{res}^2$ there is injection into the reservoir whereas when $P_{BHP}^2 < P_{res}^2$ there is a backflow or blowout. In other words, the boundary condition at the bottom of the well can be discretely utilised to satisfy both injection and blowout cases. The above is in contrast to Linga & Lund, (2016) who used a standard linear correlation called "*injectivity index*" in describing the relationship between the reservoir and the bottom of the well. Equation (4.1) represents a more sophisticated condition than a standard, linear correlation between the bottom-hole pressure and the flow rate given by an injectivity index.

During blowout, the boundary condition at the top of the well follows the analysis of time-dependent boundary conditions for supersonic outflows described by Thompson, (1990). The above equates the pressure, enthalpy and mass flowrate in the last computational cell at the top of the well to those at the ghost cell at time $t + \Delta t$. As such, in order to close the flow equations the appropriate boundary condition at the wellhead located at (z = 0) requires setting the initial velocity, u prior to blowout equals to zero.

Prior to blowout at the reservoir end, i.e. z = L, where L is the length of the wellbore, the fluid is exposed to the higher reservoir static pressure, P_{res}. This means that the reservoir static pressure is greater than the instantaneous bottom-hole pressure during blowout ($P_{BHP}^2 < P_{res}^2$).

Figure 4.2 shows a schematic representation of the isenthalpic outflow condition in the ghost cell (R.J. LeVeque, 2002a) at the bottom-hole into the first computational cell at the bottom of the well. At such, in order to close the flow equations, the appropriate boundary condition at the bottom of the well is:

$$^{dP}/_{dt} = 0 \tag{4.2}$$

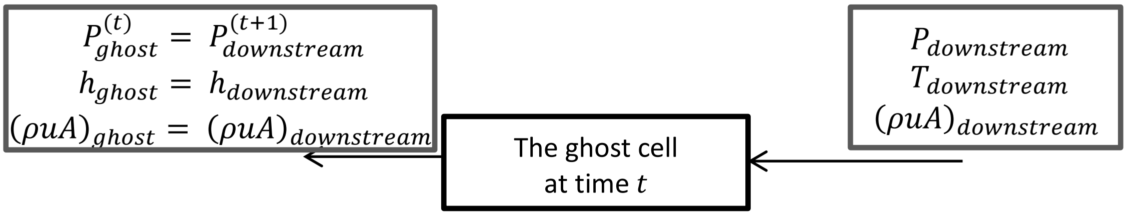

$P_{ghost}^{(t)} = P_{downstream}^{(t+1)}$
$h_{ghost} = h_{downstream}$
$(\rho uA)_{ghost} = (\rho uA)_{downstream}$
The ghost cell at time t
$P_{downstream}$
$T_{downstream}$
$(\rho uA)_{downstream}$

Figure 4.2: Schematic representation of the isenthalpic outflow condition in the ghost cell at the bottom of the well

4.3 Case studies

Considered below are two realistic case studies covering blowout prior to injection and blowout during injection. Case study 1 analyses and contrasts the pressure and temperature profiles of CO_2 at the wellhead and bottom of the well '*during start-up injection'* and '*blowout prior to injection*'. Here, the pressure and temperature profiles of CO_2 during start-up injection (downward flow) are compared against those of blowout (upward flow) under same initial conditions at the top and bottom of the well. On the other hand, case study 2 considers a comparison of CO_2 pressure and temperature profiles at the wellhead and bottom of the well during '*blowout prior to injection'* and '*blowout after 5 mins of injection'*. In other words, the initial conditions for blowout prior to injection may differ from those of blowout after 5 mins of injection.

The data used in this study obtained from the Peterhead CCS project include the well depth and pressure and temperature profiles, along with the surrounding formation characteristics as presented in Li et al., 2015 and Shell UK, 2015 and reproduced in Table 4.1. The "Peterhead CCS project was aimed to capture 1.2 million tonnes of CO_2 per annum for 15 years from an existing combined cycle gas turbine located at Peterhead Power Station in Aberdeenshire, Scotland" (Shell UK, 2015). In the project, the CO_2 captured from the Peterhead Power Station would have been transported by pipeline and then injected into the depleted Goldeneye reservoirs. Despite the cancellation of the project funding, useful information was already available, given that the Goldeneye reservoir had been used for extraction of natural gas for many years.

Table 4. 1: Goldeneye injection well and CO_2 inlet conditions (Li et al., 2015 and Shell UK, 2015)

Input parameter	**Value**
Wellhead pressure, bara	36.5
Wellhead temperature, K	280
Bottom-hole temperature, K	353.15
Well depth, m	2580
CO_2 injection rate, kg/s	38
Injection tube diameter, m	0.125m
CO_2 inlet pressure, bara	115
CO_2 inlet temperature, K	277.15
CO_2 mass flux constants (see Equation 4.1)	$\tilde{A} = 0$ $\tilde{B} = 1.3478 \times 10^{12}$ Pa^{2}s/kg $\tilde{C} = 2.1592 \times 10^{10}$ Pa^{2}s^2/kg^2 $P_{res} = 177$ bara

Figure 4.3 shows the initial distribution of the pressure and the temperature profiles along the well. Comparing the above with the phase diagram in Figure 4.4, the CO_2 is in the gaseous

phase in the first 400 m along the well and dense phase along the rest of the well depth. This initial condition is common to both start-up injection and sudden blowout considered. In addition, a CFL condition of 0.3, $\Delta t = 1 \times 10^{-4}$s and 300 computational discretisation cells were employed. These ensured numerical stability and convergence. Varying range of mass flow rates at the inlet with time can be assessed. However, for purpose of this analysis a constant mass flow rate of 38 kg/s injected for 300 s at CO_2 feed pressure of 115 bara is employed. Also, for simplicity an overall heat transfer coefficient of 0.01 kWm^{-2}K^{-1} is assumed representing outer well layers such as casing, cementing and other well surrounding materials.

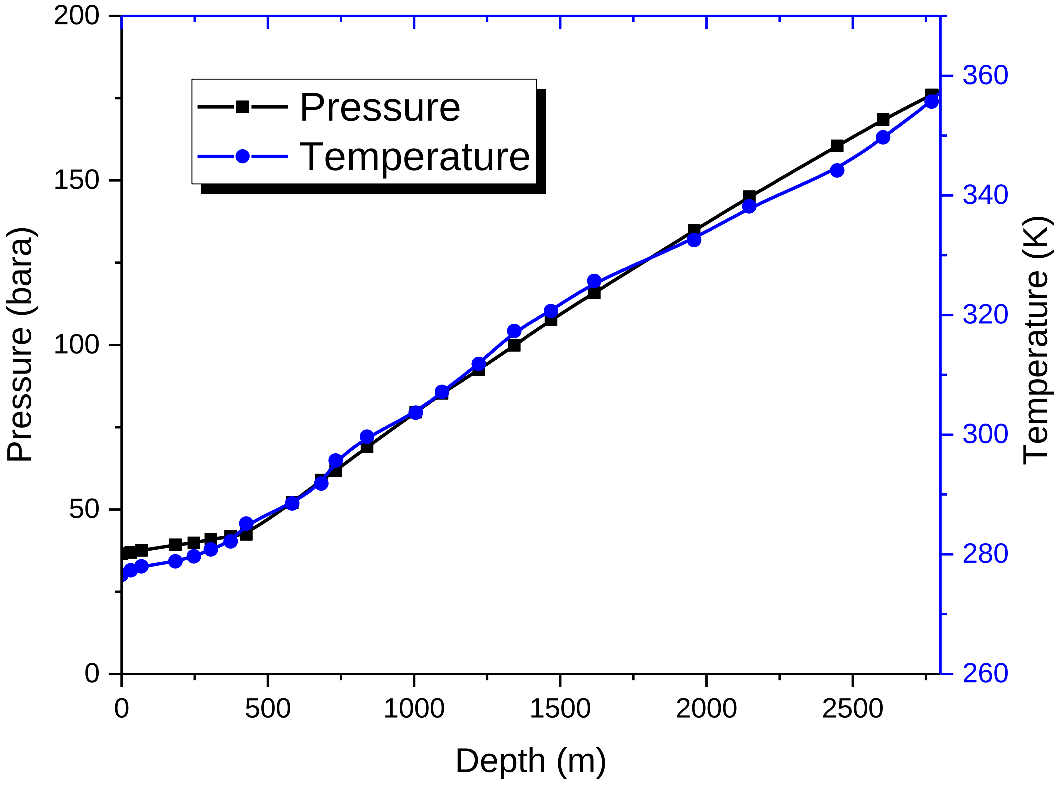


Figure 4.3: Initial distribution of the pressure and the temperature along the well

4.3.1 Case study 1: CO_2 start-up injection versus sudden blowout prior to injection

In this case study, CO_2 start-up injection pressure and temperature transient profiles are compared with those following sudden blowout prior to injection. The comparison is very significant in understanding the similarities and differences between the transient pressure and temperature profiles of the two scenarios at the wellhead and bottom-hole. The resulting pressure and temperature transients at both the wellhead and bottom of the well are presented in Figures 4.5 to 4.13.

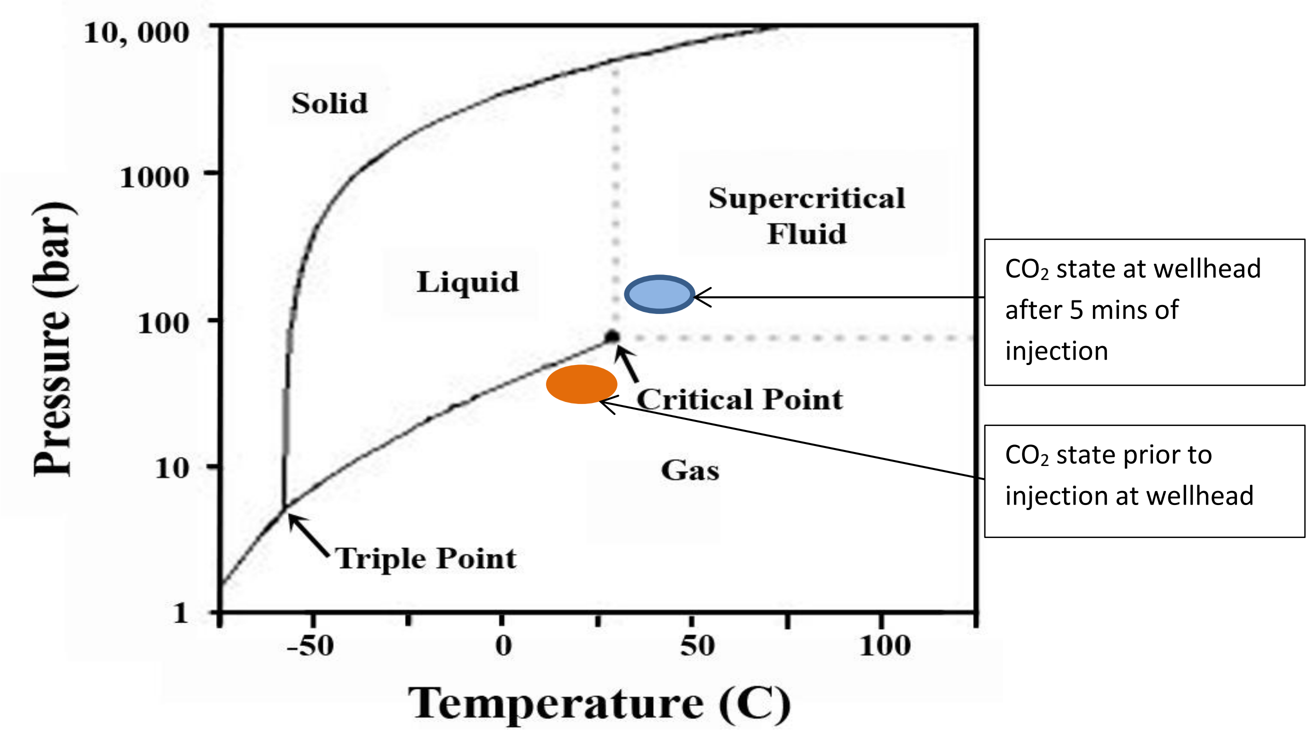


Figure 4.4: CO_2 P-T phase diagram (adapted from Krogh, Nilsen, & Henningsen, 2012)

Figure 4.5 shows the variations in pressure with time at the top of the well during start-up injection and blowout prior to injection spanning the first 300 s. The start-up injection data shows an initial relatively gradual drop in the wellhead pressure from 36.5 bara to 30 bara within the first 50 s. This is followed by a rapid pressure build-up to a maximum of 121 bara after 70 s, followed by a gradual drop reaching a marginal rate of drop after 200 s. In a similar way, the blowout data also shows a drop in the wellhead pressure from 36.5 bara to 28.5 bara in the first 50 s followed by a rapid but modest pressure build-up to a maximum of 41 bara after 50 s, attaining a marginal rate of drop after 100 s. The trend observed in both cases appear similar, but there is a significant pressure difference between them after the first

50 s. The wellhead pressure after 50 s of start-up injection increased to 106 bara whilst for the sudden blowout case the wellhead pressure remained below the initial wellhead pressure of 36.5 bara corresponding to the choke pressure condition during sudden blowout.

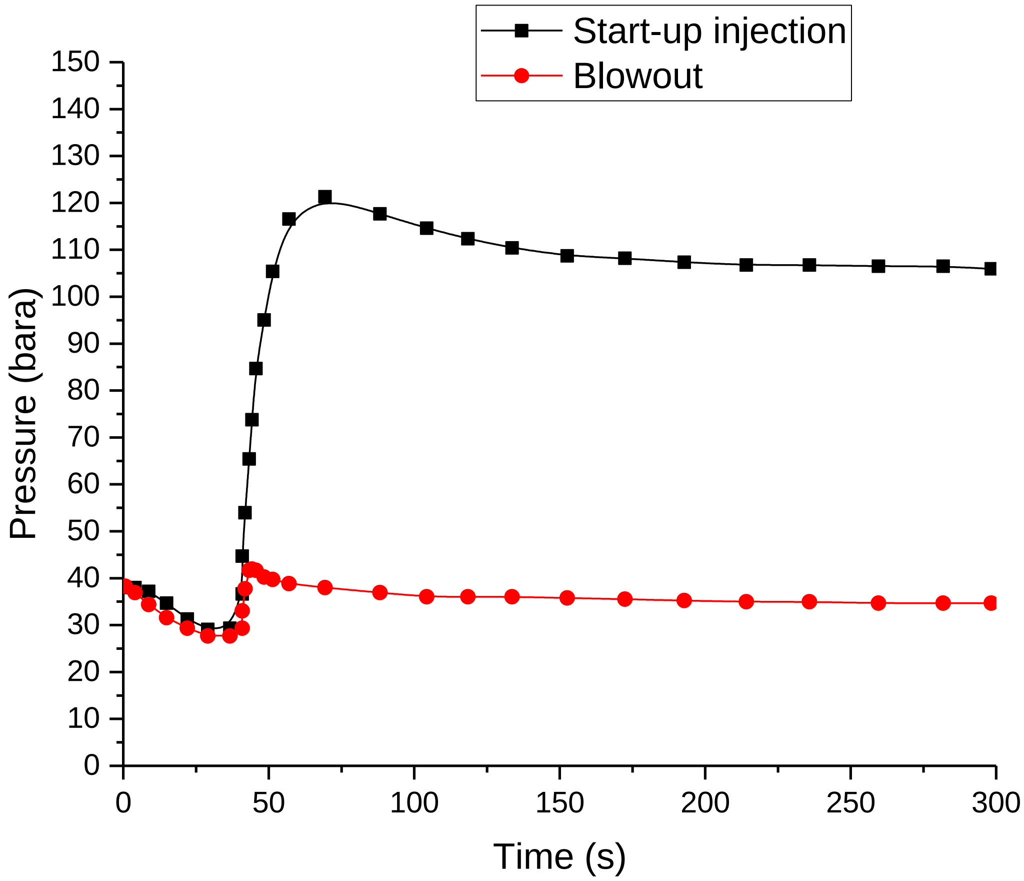


Figure 4.5: Variation in wellhead pressure with time for start-up injection and blowout prior to injection

Figure 4.6 shows the corresponding variations in temperature with time at the top of the well during start-up injection and blowout prior to injection. The start-up injection case data indicates a sharp drop in initial wellhead temperature due to Joule-Thomson cooling effect to ca. 266 K within the first 50 s followed by a rapid increase to 279 K and finally attaining a relatively marginal linear rate of drop after 100 s. The blowout simulation result predicts a larger drop in the wellhead temperature down to 265 K followed by a sharp build-up to a maximum of 274 K and finally reaches a marginal linear decrease. Notably, the temperature of CO_2 at the wellhead reaches sub-zero within the first 50 s of injection; this may result in ice formation in the presence interstitial water molecules and hence the risk of well blockage. Also, sub-zero temperature poses a serious risk of thermal stress shocking of the well bore casing steel leading to its fracture and eventual escape of CO_2. A similar trend is observed in both instances however, there is a 5 K temperature difference between them after 300 s simulation time. The wellhead temperature after 300 s of simulation stays below the initial 277 K in the case of blowout prior to injection.

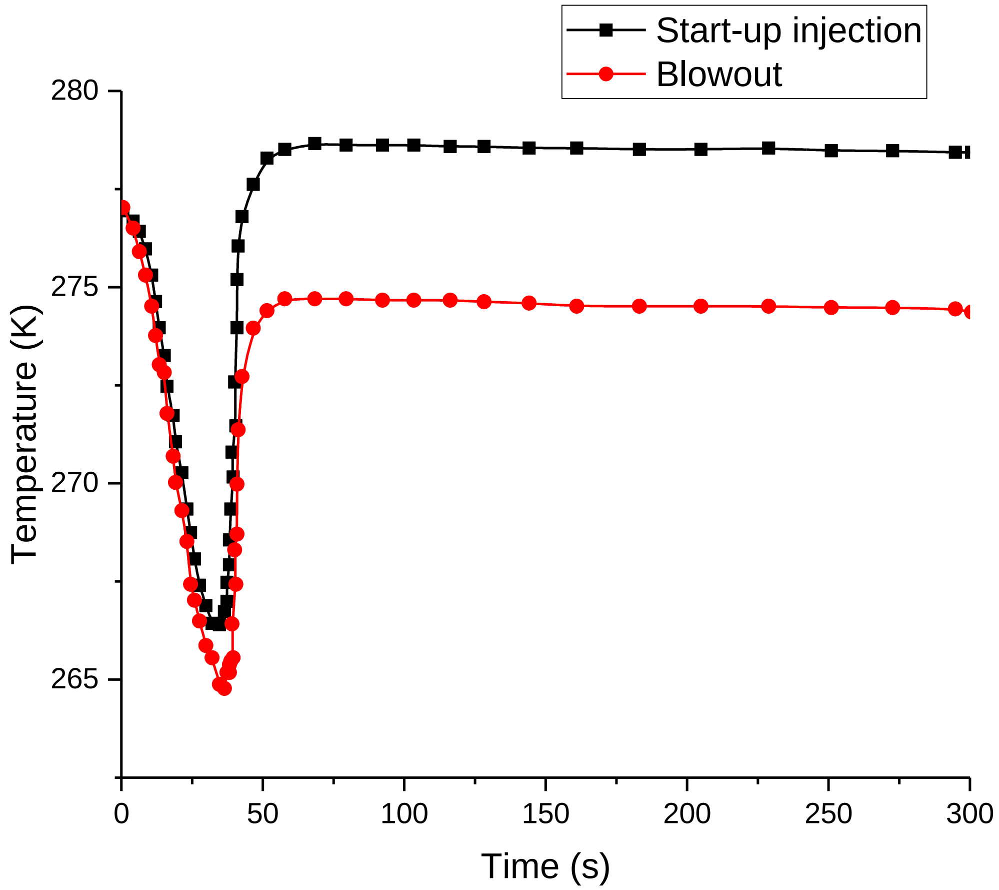


Figure 4.6: Variation in wellhead temperature with time for start-up injection and blowout prior to injection

Figure 4.7 shows the bottom-hole pressure profiles with time for start-up injection and blowout prior to injection. As expected, the results show significantly different pressure profiles. A rapid pressure build-up in 100 s reaching a maximum value of 195 bara from 177 bara at the bottom of the well during injection. A relatively constant pressure thereafter is observed. On the other hand, a decline in pressure from the starting pressure of 177 bara to 171 bara is predicted at the bottom of the well during blowout.

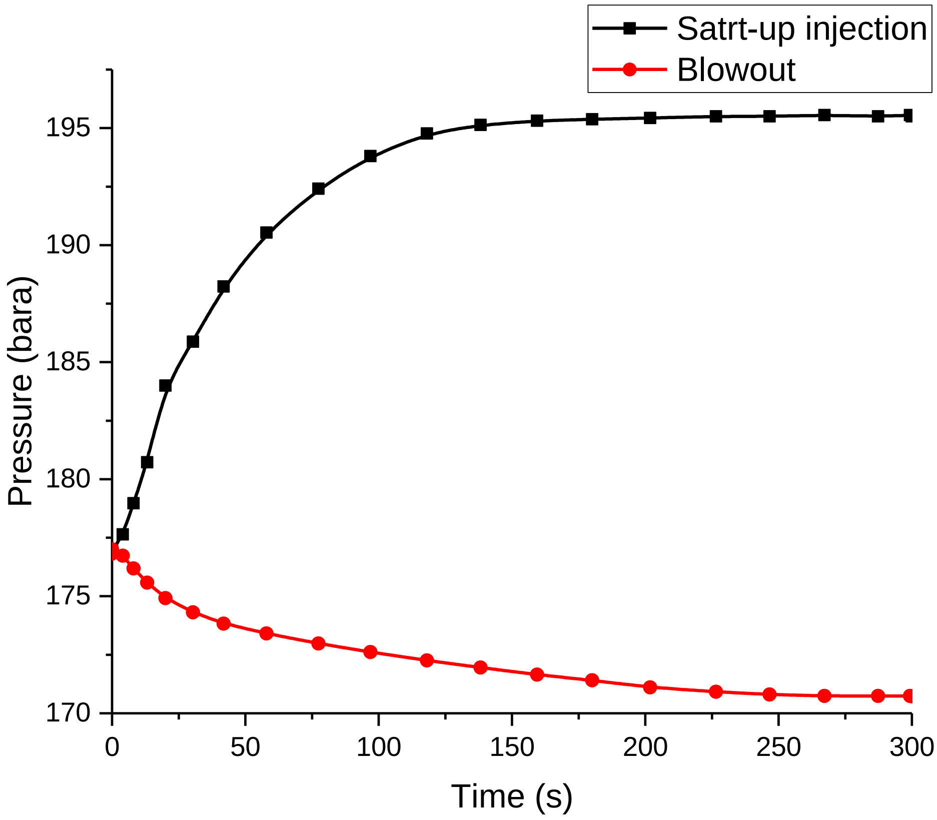


Figure 4.7: Variation in bottom-hole pressure with time for start-up injection and blowout prior to injection

Figure 4.8 shows the corresponding variations in bottom-hole temperatures with time for start-up injection and blowout prior to injection. A significant decrease in temperature at the bottom of the well starting from ca 352 K down to ca 290 K is predicted during start-up injection. This drop is likely to be as a result of the expanding CO_2 which cools upon arriving at the reservoir. In contrast however, the sudden blowout prior to injection simulation results in a relatively marginal drop in the bottom-hole temperature.

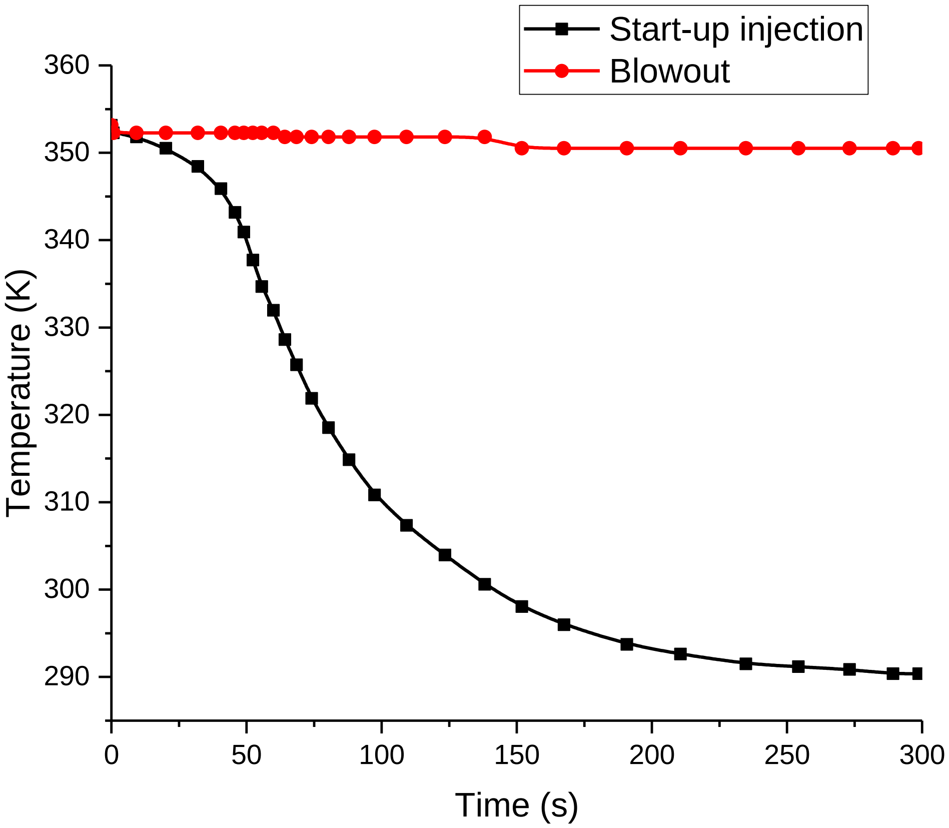


Figure 4.8: Variations in bottom-hole temperature with time for start-up injection and blowout prior to injection

4.3.2 Case study 2: Blowout prior to injection versus blowout after 5 mins of injection

This case study compares the simulation results of CO_2 blowout from the injection well prior to start of injection and blowout after 5 mins of injection. This analysis is conducted to study the effect of varying wellhead and bottom-hole initial conditions on the pressure and temperature transients during blowout. As can be seen from the previous case study and Figure 4.9, the initial wellhead pressure and temperature after 5 mins of injection are 106 bara and 278 K respectively compared with the previous initial 36.5 bara and 277 K. At the bottom of the well, the previous initial pressure and temperature was 177 bara and 353.15 K respectively but after 5 mins of injection the pressure has reach 195 bara while the temperature has reduced to 292 K. The pressure and temperature profile at the wellhead and bottom of the well for both cases are presented in Figures 4.10 to 4.13.

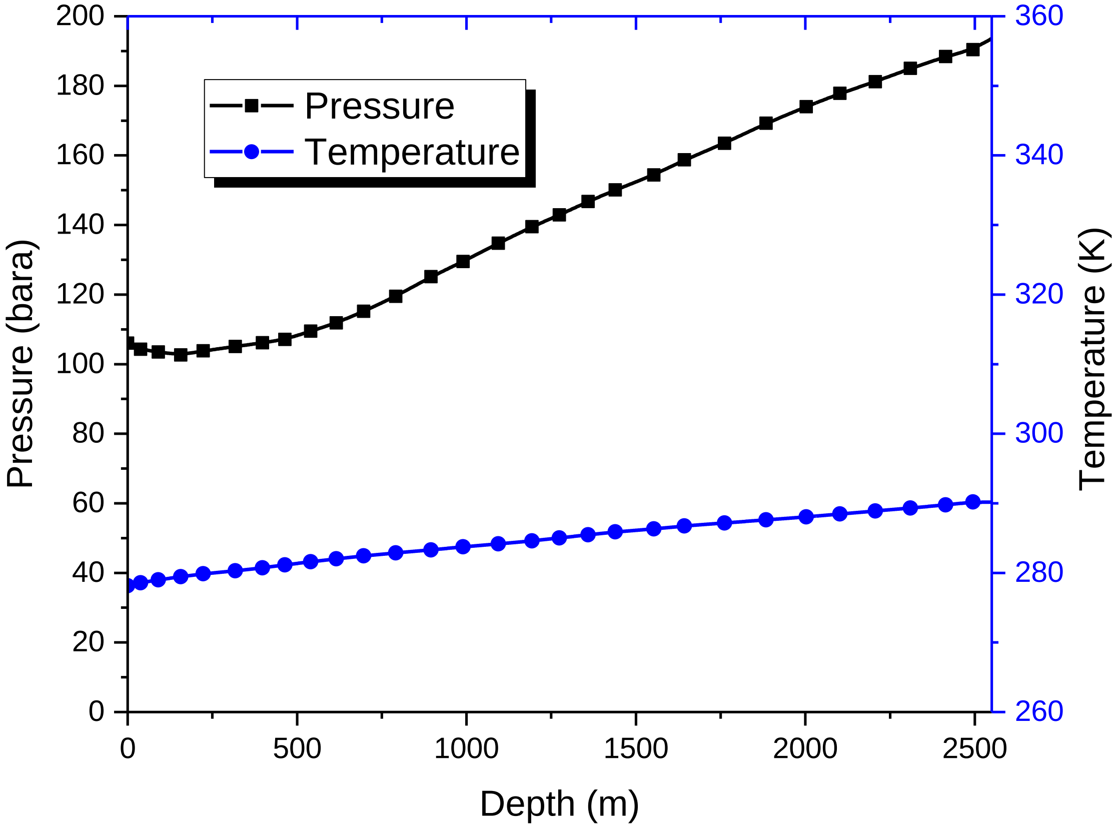


Figure 4.9: Pressure and temperature initial distribution along the well after 5 mins of injection

Figure 4.10 shows the variation in pressure with time at the wellhead for blowout prior to start-up injection and blowout after 5 mins (300 s) of injection. After 5 mins of injection the wellhead pressure reaches 106 bara, at which point blowout occurs. A sharp drop in pressure to 98 bara is observed within the first 30 s followed by brief pressure build-up to 105 bara and finally reaching steady state at 102 bara. This analysis shows that the pressure drop at the top of the well is relatively modest (106 to 98 bara) at higher wellhead pressure. The significance of this study is connected to understanding the fluid properties behaviour of CO_2 during scenario such as start-up injection and blowout. Looking at Figure 4.4 and 4.9 it is clear that CO_2 wellhead condition prior to injection and after 5 mins of injection significantly defers. Prior to injection CO_2 is in the vapour phase at the top of the well but after 5 mins of injection the wellhead pressure has significantly increased and CO_2 is now in the dense phase.

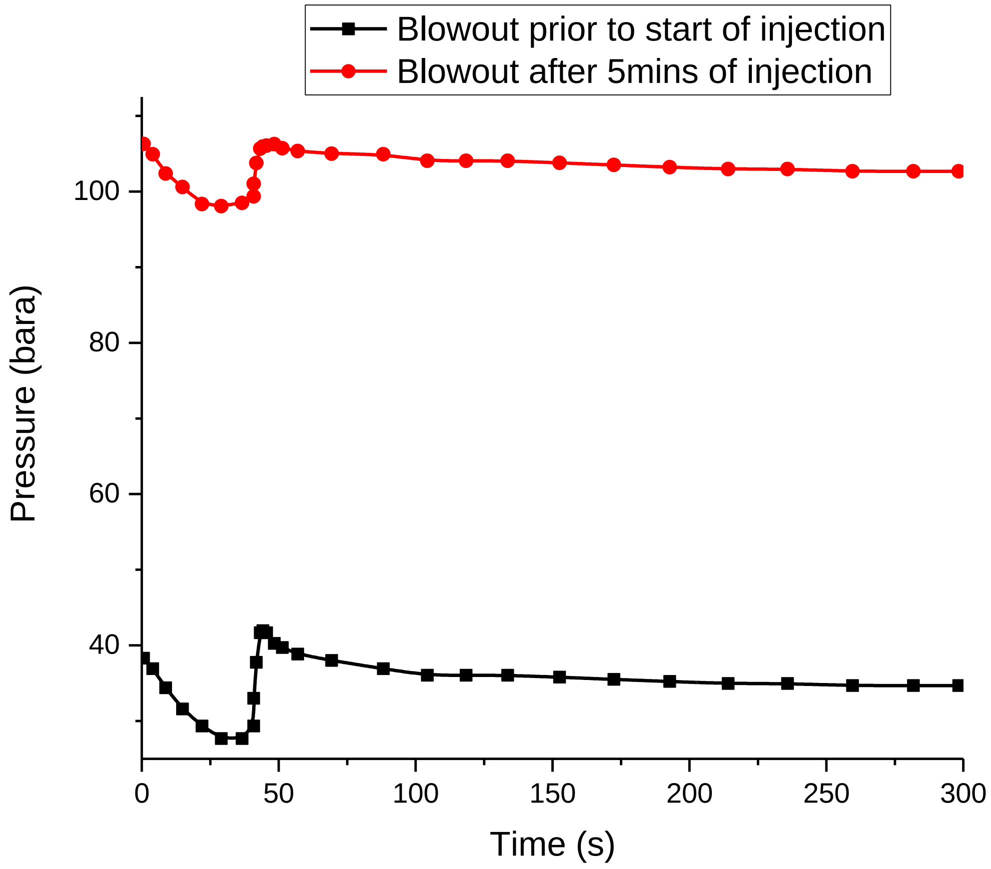


Figure 4.10: Variation in pressure with time at the wellhead for blowout prior to start-up injection and blowout after 5 mins of injection

Figure 4.11 shows the variation in temperature with time at the wellhead for blowout prior to start-up injection and blowout after 5 mins of injection. Notably, after 5 mins of injection the wellhead temperature has reached 278 K, at which point the sudden blowout is assumed to occur. A rapid drop in temperature from 278 K to 270 K is observed within the first 30 s followed by a build-up to 277 K and finally reaching a steady state at 276 K. This analysis for the two blowout scenarios show that during sudden blowout, a higher wellhead temperature will minimise the overall temperature drop compared with lower wellhead temperatures. Also, the minimum temperature at the wellhead for the case of blowout after 5 mins of injection is below the freezing point of water. This means there is a possibility of ice formation should the CO_2 come in contact with interstitial water molecules.

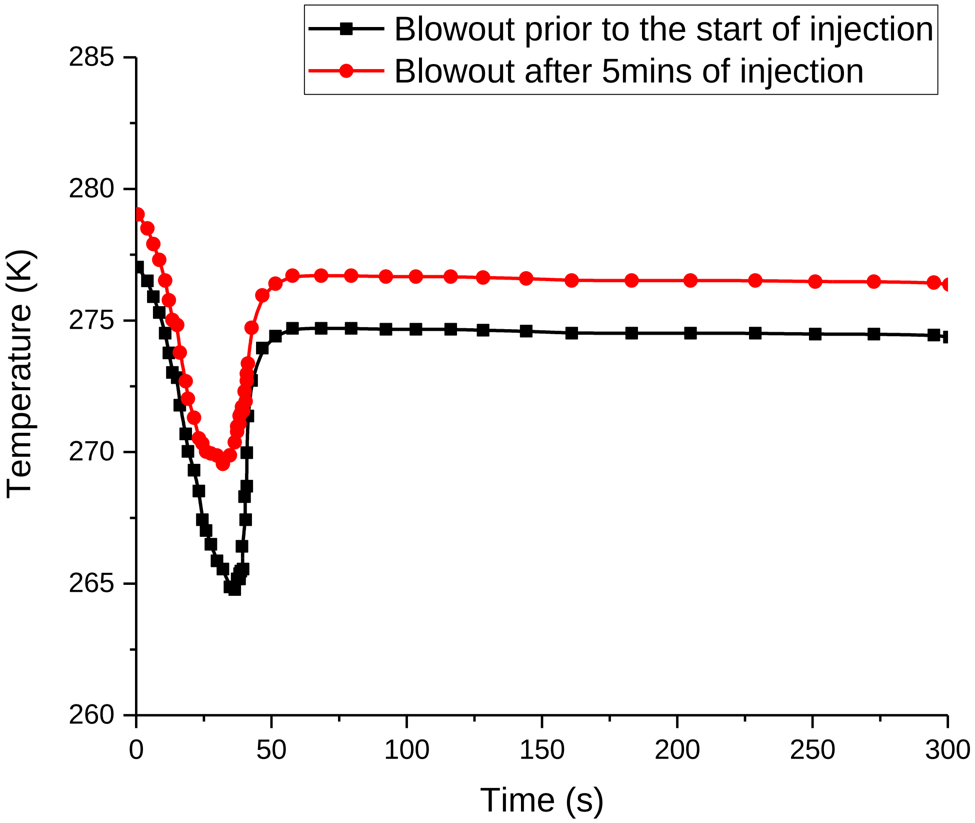


Figure 4.11: Variation in temperature with time at the wellhead for blowout prior to start-up injection and blowout after 5 mins of injection

Figure 4.12 shows the variation in pressure with time at the bottom of the well for blowout prior to start-up injection and blowout after 5 mins of injection. Both pressure profiles at the bottom of the well show a gradual decline in pressure during blowout. After 5 mins of injection the pressure at the bottom of the well has significantly risen from 177 bara to 196 bara before the blowout. Consequently, after 5 mins of outflow during blowout the pressure at the bottom of the well slowly decreases to 190 bara. This shows a slower decline in pressure compared with the rate of increase during start-up injection for same time frame as can be seen when comparing Figures 4.7 and 4.12.

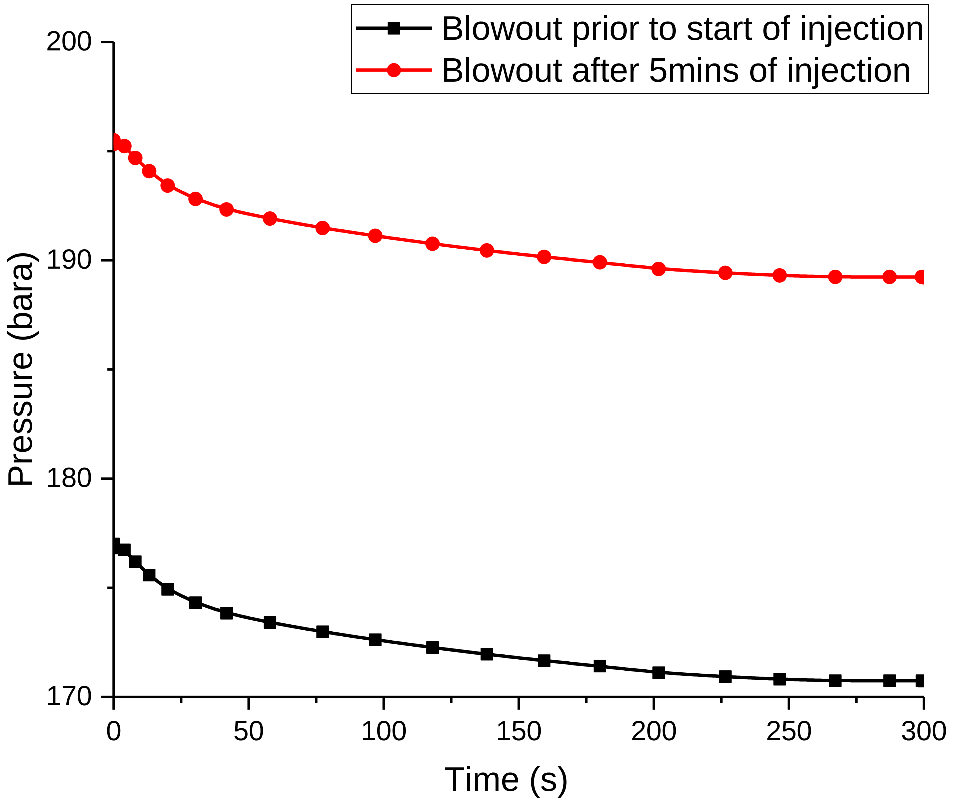


Figure 4.12: Variation in bottom-hole pressure with time for blowout prior to start-up injection and blowout after 5 mins of injection

Figure 4.13 shows the variation in temperature with time at the bottom of the well for blowout prior to start-up injection and blowout after 5 mins of injection. The profiles show very slow and marginal decline in temperature at the bottom of the well during blowout. After 5 mins of injection the bottom-hole temperature has significantly dropped from 353 K to 290 K before the blowout. Consequently, after 5 mins of outflow the bottom-hole temperature slowly drop further to 287 K showing a slower decline compared to decline during injection for same time frame as can be seen in Figure 4.6 and 4.13. The temperature variation at the bottom of the well is near constant for the blowout cases.

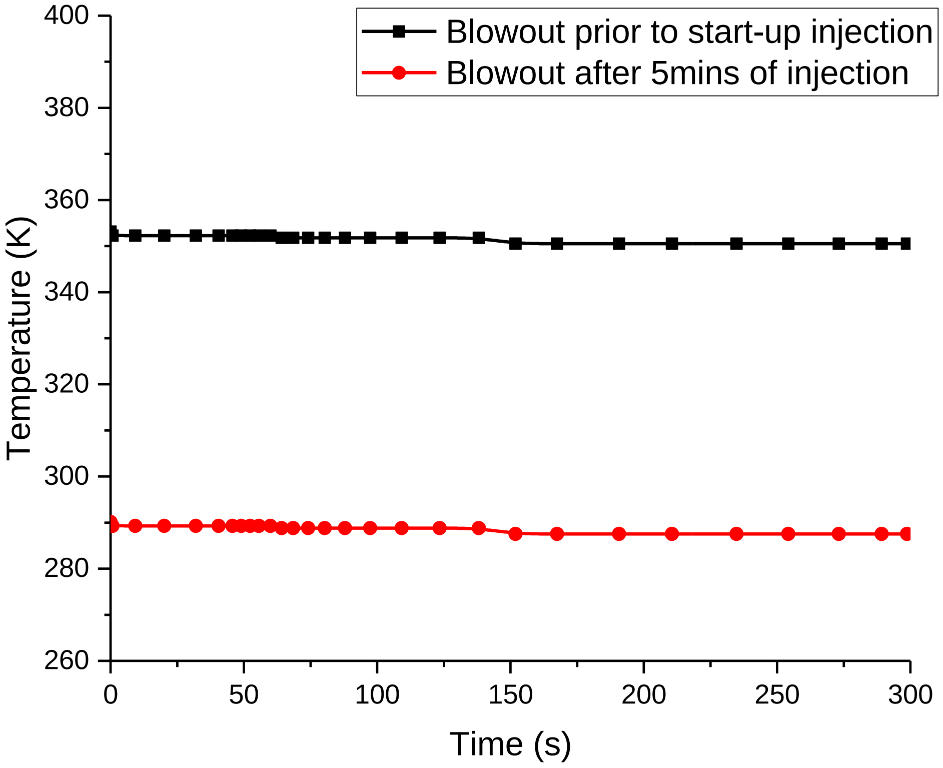


Figure 4.13: Variation in bottom-hole temperature with time for blowout prior to start-up injection and blowout after 5 mins of injection

4.4 Concluding remarks

This chapter presented the formulation of the boundary conditions for start-up injection and blowout from the well. The bounding conditions are applied as closure equations for the HEM model presented in chapter 3. The model was tested based on two realistic case studies involving the analysis of CO_2 start-up injection and blowout from the injection well. As explained in the chapter introduction, blowout from injection well can occur under varying circumstances such as underground fluid movement resulting to pressure build-up due to the hydrostatic condition. The first case study compared the simulated pressure and temperature changes during start-up injection with those of blowout prior to injection. The second case study compared the simulated pressure and temperature changes of blowout prior to injection with those of blowout after 5 mins of injection. Based on these analyses the following conclusions can be drawn:

- At the wellhead, there is significant drop ca. 266 K (– 7 ^{o}C) in temperature during injection. This poses the risk of ice formation in the presence of interstitial water molecules and hence the danger of well blockage. Also, sub-zero temperature poses a serious risk of thermal stress shocking of the well bore casing steel (as metals are likely taking out of their stress limit) leading to its fracture and eventual escape of CO_2. The CO_2 pressure and temperature ca. 28 bara and 264 K (– 9 ^{o}C) predicted respectively at the wellhead shows lower values for the blowout case compared with the injection case. More importantly based on the case studies conducted, a phase change from gas to dense phase CO_2 occurs after 50 s of injection at the top of the well. This transition is important as dense phase CO_2 injection is highly recommended due to its high density and low viscosity. However, there is danger of over-pressurisation accompanied by CO_2 backflow into the injection system due to the violent evaporation of the superheated liquid CO_2 upon entry into the well bore.

- The analysis for the two blowout scenarios show that during sudden blowout, a higher wellhead temperature will minimise the overall temperature drop compared with lower wellhead temperatures. Also, the minimum temperature at the wellhead for the case of blowout after 5 mins of injection is below the freezing point of water. This means there is a possibility of ice formation should the CO_2 come in contact with interstitial water molecules.

- At the bottom of the well, a rapid pressure build-up during injection occurs. After 5 mins of injection the pressure at the bottom of the well raises from 177 bara to 196 bara before the blowout. Consequently, after 5 mins of outflow during blowout the pressure at the bottom of the well slowly decrease to 190 bara. Also at the bottom of the well, a rapid decline in temperature from 353 K to 290 K during injection occurs but a slow decline from 290 K to 287 K during blowout. The predicted results at the bottom of the well gives a clear indication of increase in pressure due to the expanding dense phase CO_2. Also, a decrease in temperature is observed indicating a cooling CO_2 stream at the bottom of the well during injection. However, the behaviour was different during blowout, as a slow decrease in pressure and temperature at the bottom of the well is observed. Significantly at the bottom of the well, the temperature is well above the freezing point of water, thus there is no risk of ice or hydrate formation.

Chapter 5: Application of the HEM Model for Injection Flow-rate Ramp-up and CO_2 Impurities Impact Assessment

5.1 Introduction

In this chapter, the HEM model presented in chapters 3 and 4 is extended to simulate CO_2 injection flow rates ramping up times corresponding to fast (5 mins), medium (30 mins) and slow (2 hrs) ramp-ups. The above time durations were chosen in accordance with industrial practice for general fluid ramp-up (see Black and Veatch, 2012 for further details). The importance of investigating the injection flow rates ramping up times is driven by the need for the development of optimal injection strategies and best-practice guidelines for the minimisation of the risks associated with the injection process. Investigating injection flowrates ramping up is essential for understanding the rate of rapid quasi-adiabatic Joule-Thomson cooling effect when high pressure CO_2 is injected into a low pressure injection well. As shown in chapter 4, rapid depressurisation if not monitored and controlled carefully may lead to significant temperature drops posing several risks, including blockage due to hydrate and ice formation with interstitial water molecules or the formation water in the perforations at the near well zone; thermal stress shocking of the wellbore casing steel leading to its fracture and ultimate escape of CO_2.

The chapter is divided into two main parts. The first presents the formulation and deployment of the CO_2 injection ramping up inlet boundary condition and variable mass flow rates closure equations.. The second part of the chapter investigates the impact of typical CO_2 impurities (such as H_2O, CH_4, N_2, CO, H_2S, Ar, etc) in the injected CO_2 stream captured from fossil fuel power plants and other industries fitted with CCS on the pressure and temperature profiles at the wellhead during CO_2 injection. The analysis presented is with respect to Porter et al. (2015) “the range of impurities present in CO_2 streams captured using pre-combustion, post-combustion and oxy-fuel technologies”.

5.2 CO_2 ramping up injection fluid dynamics

The HEM model developed in chapter 3 is employed in the following for the simulation of the time-dependent mass flow or ramping up of CO_2 into injection wells (schematic representation of the injection well and its various component was presented in Figure 1.2). To account for such inlet condition, a suitable boundary condition must be specified at the top of the well. The HEM assumption is applied in the ghost cell at the well inlet and the flow is assumed to be isentropic. As a result, the conservation equations following Thompson, (1987) and (1990) are rewritten as:

$$\frac{\partial \rho}{\partial t} + \frac{1}{c^2}\left[\zeta_2 + \frac{1}{2}(\zeta_3 + \zeta_1)\right] = 0 \qquad (5.1)$$

$$\frac{\partial p}{\partial t} + \frac{1}{2}[\zeta_3 + \zeta_1] = 0 \qquad (5.2)$$

$$\frac{\partial u}{\partial t} + \frac{1}{2\rho c}[\zeta_3 - \zeta_1] - g = 0 \qquad (5.3)$$

where, $\zeta$is referred to as the wave amplitude, for which the generic expressions are given by:

$$\zeta_1 = (u - c)\left\{\frac{\partial p}{\partial x} - \rho c \frac{\partial u}{\partial x}\right\} \qquad (5.4)$$

$$\zeta_2 = 0 \qquad (5.5)$$

$$\zeta_3 = (u + c)\left\{\frac{\partial p}{\partial x} + \rho c \frac{\partial u}{\partial x}\right\} \qquad (5.6)$$

ζ_2 describes the inflow entropy, so setting $\zeta_2 = 0$ states that the inflow entropy is constant in the x-direction. Following (K W Thompson, 1990), modifications to Equations (5.4) and (5.6) may be required at the flow boundaries. In this case, ζ_1 and ζ_3 have to be specified in accordance with the flow acceleration through the variable flow areas (i.e. from the injection pipeline into the wellhead).

From Equations (5.1) and (5.3), the following can be derived for ζ_1 and ζ_3:

$$\zeta_1 = \frac{1}{(u-c)}\left\{2\rho c^2 g \frac{\partial \dot{m}}{\partial t} - (u + c)\zeta_3\right\} \qquad (5.7)$$

$$\zeta_3 = \frac{1}{(u+c)}\left\{2\rho c^2 g \frac{\partial \dot{m}}{\partial t} - (u - c)\zeta_1\right\} \qquad (5.8)$$

where $\dot{m} = \rho u$ and its derivative with respect to time is approximated by:

$$\frac{\partial \dot{m}}{\partial t} = \frac{1}{\Delta t}[\dot{m}|_n - \dot{m}|_{n-1}] \qquad (5.9)$$

where the subscripts, $n - 1$ and n respectively denote the previous and current time-steps.

Finally, Equations (5.1) to (5.3) can now be readily solved numerically for a complete description of the flow conditions in the ghost cell at the wellhead. Refer to section 3.2 of chapter 3 where the detailed numerical solution is described.

The boundary condition at the bottom of the well as described in section 4.2 of the previous chapter applies in this case.

In the following section considers the linearly ramped-up injection mass flow rates from 0 to 38 kg/s in 5 mins, 30 mins and 2 hrs in order to elucidate the impact of injection ramp-up duration.

5.2.1 Results and discussion

The results obtained following the simulation of the transient flow model for the ramping up injection of CO_2 into the above depleted gas field are presented and discussed. Figures 5.1 to 5.3 respectively show the variations in CO_2 pressure with time at the top of the well for 5 mins, 30 mins and 2 hrs ramping up injection times.

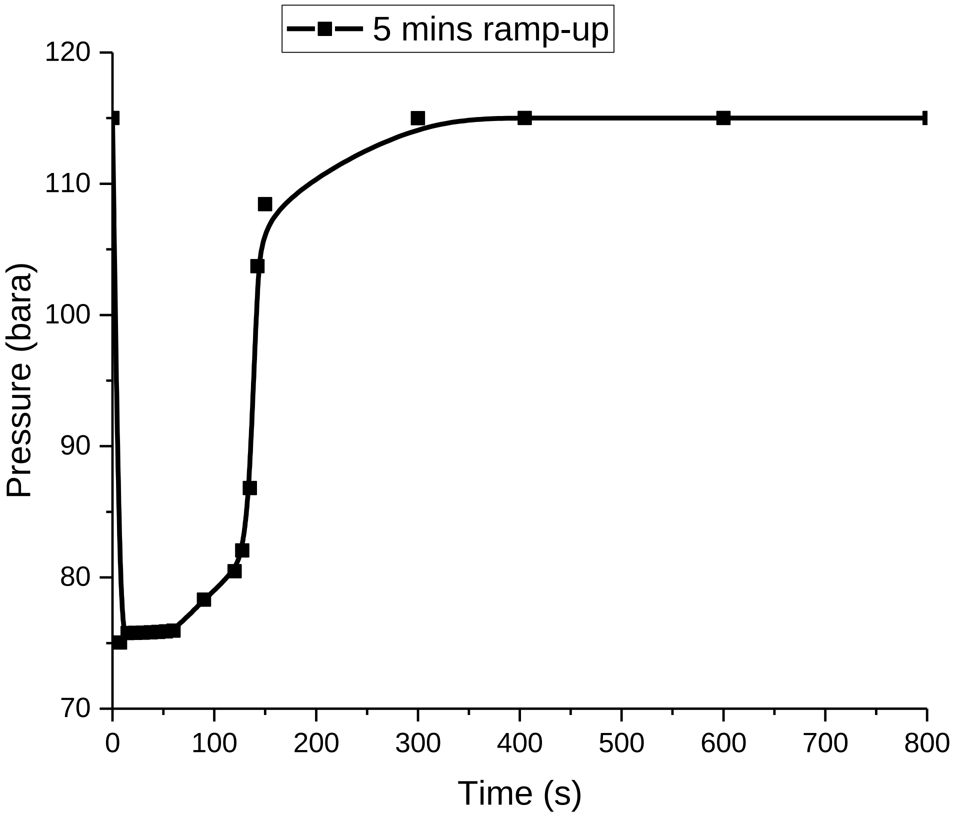


Figure 5.1: CO_2 wellhead pressure variation with time for 5 mins ramp-up injection case. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 38 kg/s

The ramping up CO_2 injection pressure profiles for all cases shows significant rapid drop in pressure of the incoming CO_2. The CO_2 inlet pressure at 115 bara drops to ca. 74, 66 and 36.5 bara, corresponding to injection ramping up period of 5 mins, 30 mins and 2hrs respectively. This is followed by a rapid recovery until a constant pressure is attained in each case. The pressure recovery rate and the magnitude of the drop in pressure decrease with increase in ramping-up duration.

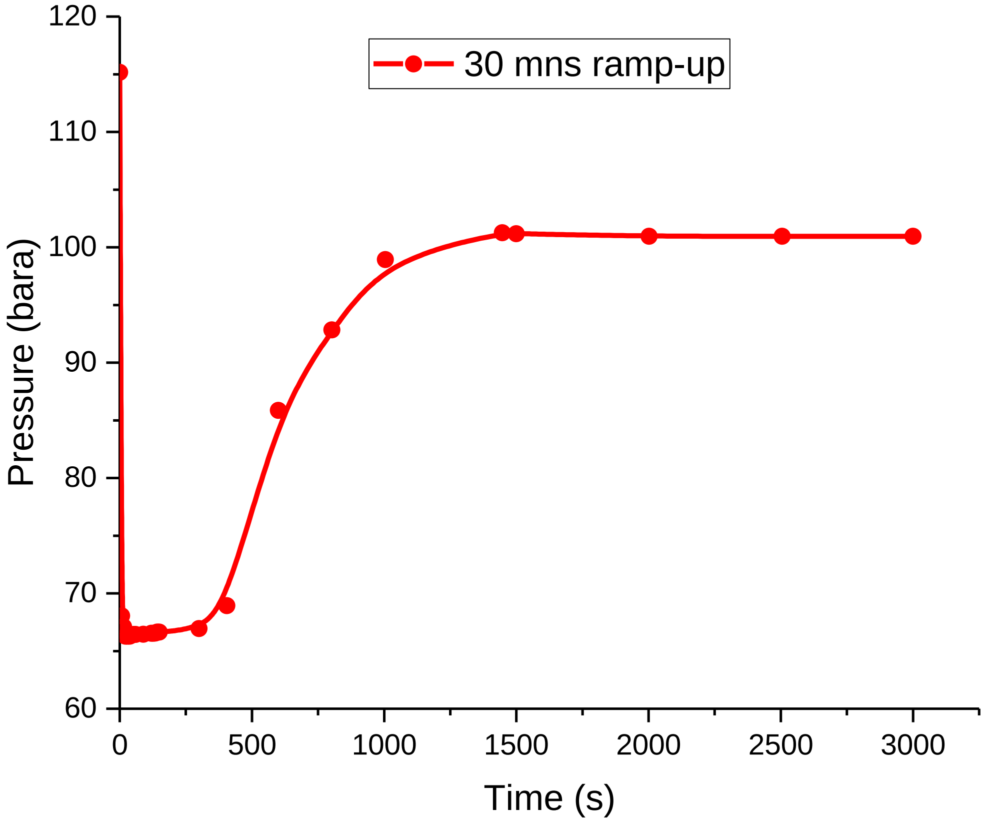


Figure 5.2: CO_2 wellhead pressure variation with time for 30 mins ramp-up injection case. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (1800 s) at which the injection flow rate reaches its peak value of 38 kg/s

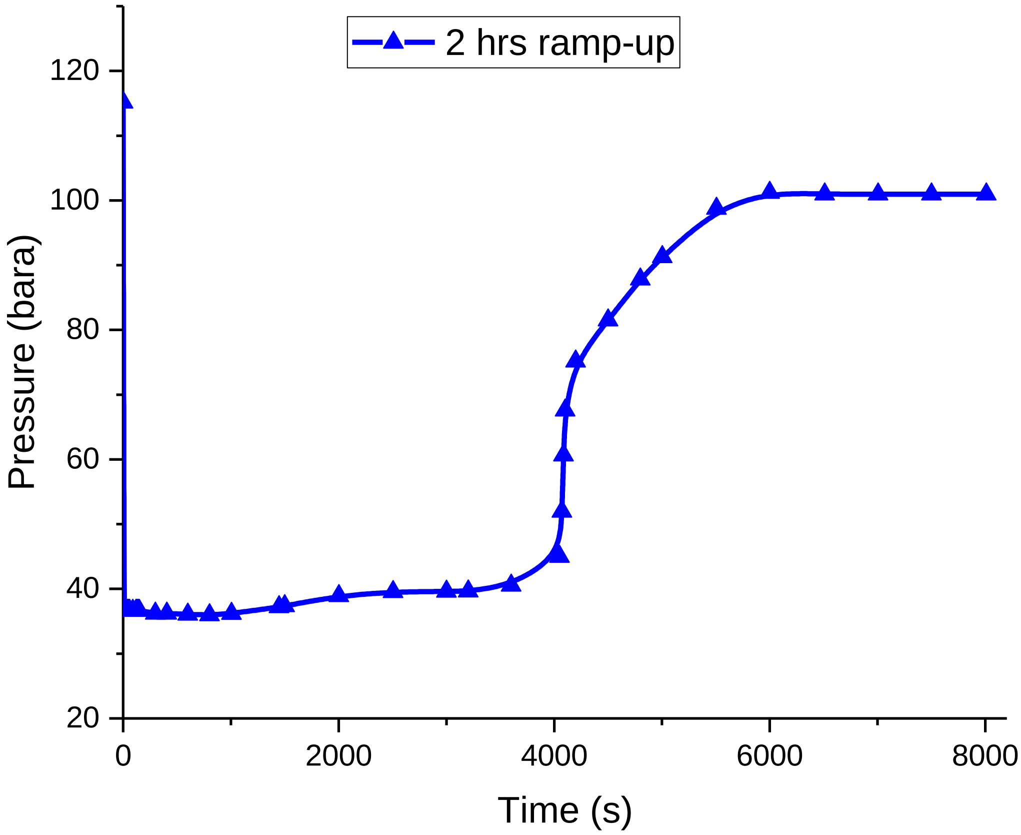


Figure 5.3: CO_2 wellhead pressure variation with time for 2 hrs ramp-up injection case. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (7200 s) at which the injection flow rate reaches its peak value of 38 kg/s

Figure 5.4 shows all three ramping up cases on the same plot to provide a clearer perspective of the difference in pressure drop and recovery behaviour for the various injections ramping up times.

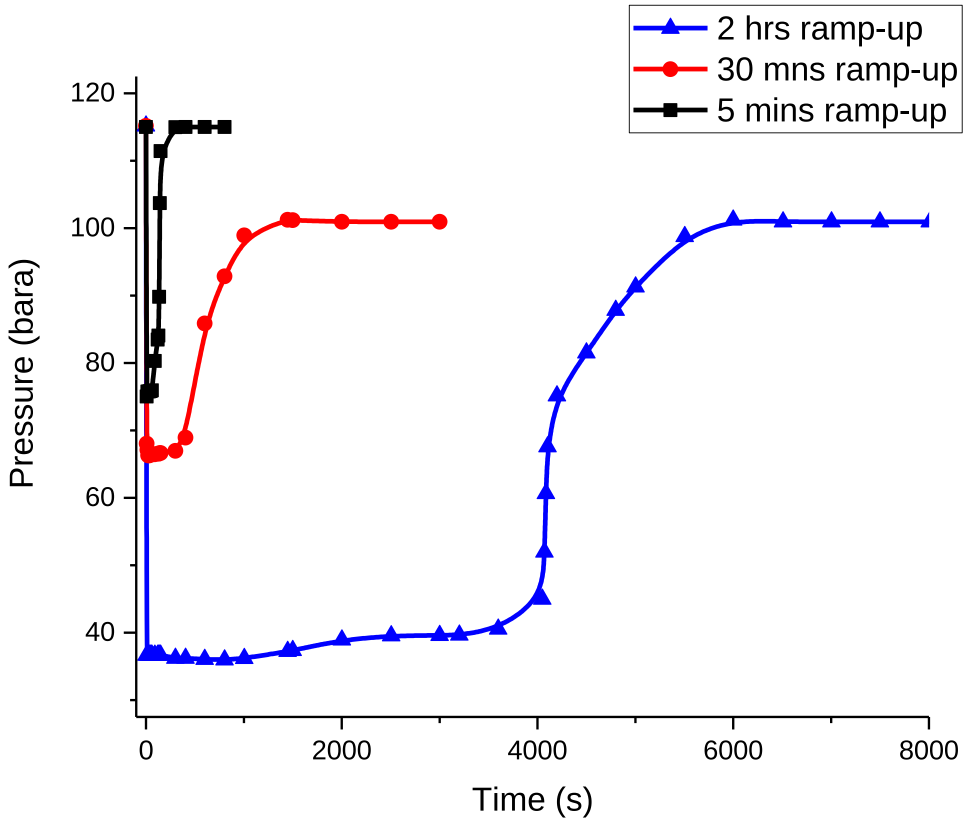


Figure 5.4: CO_2 wellhead pressure variation with time for all three ramp-up injection cases. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara).

Figures 5.5 to 5.7 show the corresponding variations in the injected CO2 temperature with time at the top of the well for 5 mins, 30 mins and 2 hrs ramping up injection times. Figure 5.8 shows corresponding superimposed plots. The ramping up injection temperature profiles for all three cases shows a rapid drop in temperature of the incoming CO_2 (277 K) to as low as 226 K. The CO_2 inlet temperature at 277 K dropped rapidly to 252, 238 and 226 K corresponding to injection ramping up duration of 5 mins, 30 mins and 2hrs respectively. The initial temperature drop is followed by a recovery over the ramp-up duration until steady state is attained in each case. The temperature recovery is quicker for the fast 5 mins ramp-up case and slower for the 30 mins or 2 hrs ramp-up cases. Consequently, the analysis clearly suggests that to minimise the initial temperature drop due to Joule-Thomson cooling effect on incoming CO_2 and allow a faster recovery after the rapid temperature drop. A faster ramping up duration is important since the fast injection ramp-up will mean less exposure time to the refrigerating effect imposed by the Joule-Thomson expansion.

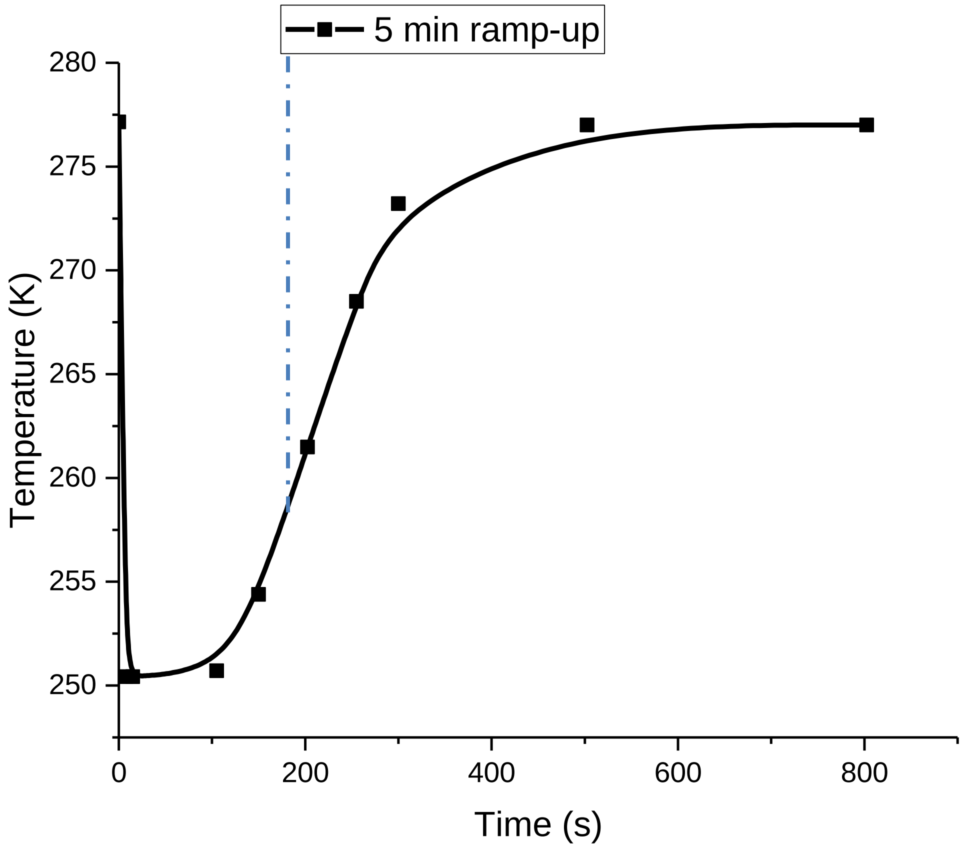


Figure 5.5: CO_2 wellhead temperature variation with time for 5 mins ramp-up injection case. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 38 kg/s

+

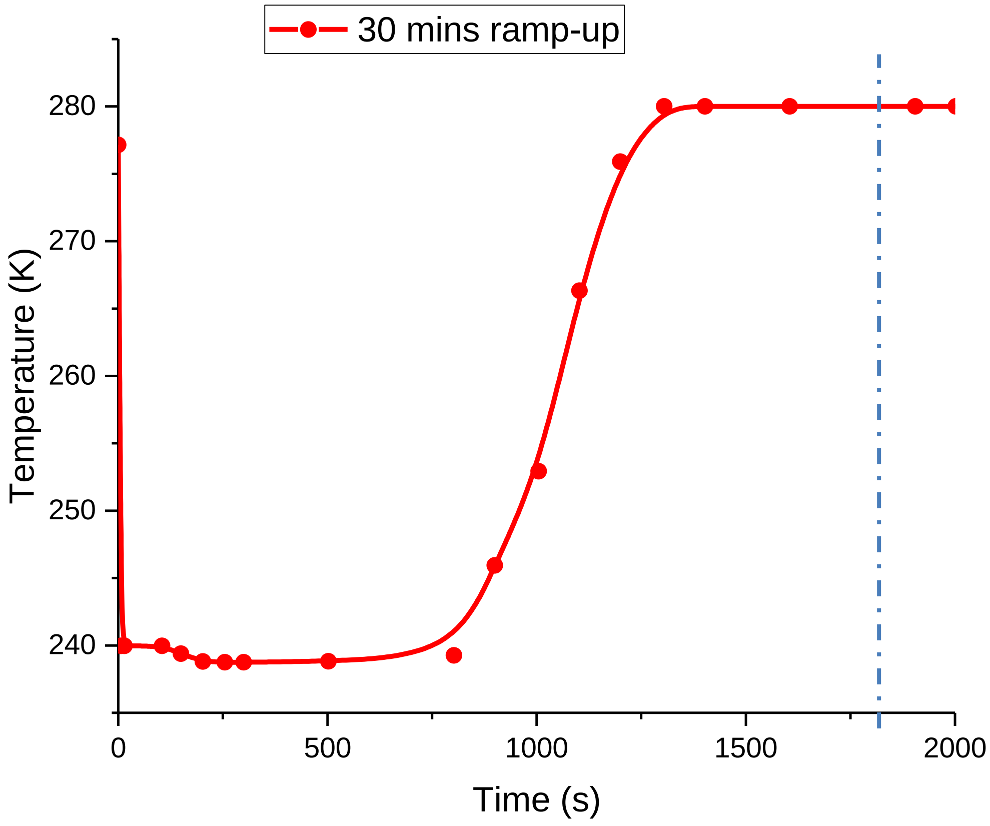


Figure 5.6: CO_2 wellhead temperature variation with time for 30 mins ramp-up injection case. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (1800 s) at which the injection flow rate reaches its peak value of 38 kg/s

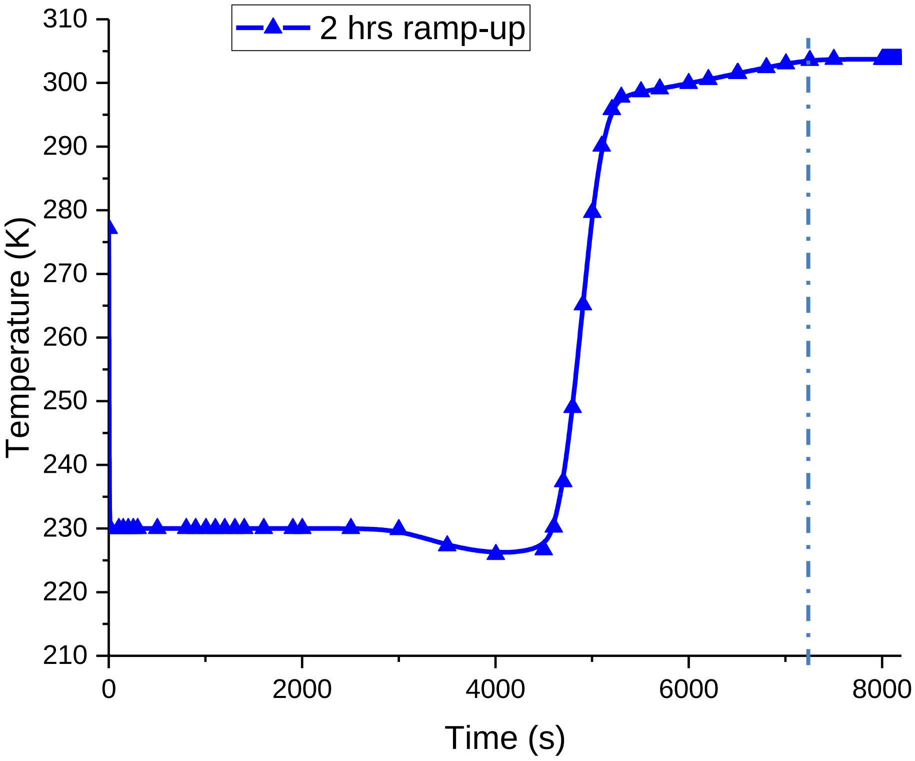


Figure 5.7: CO_2 wellhead temperature variation with time for 2 hrs ramp-up injection case. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (7200 s) at which the injection flow rate reaches its peak value of 38 kg/s

Figure 5.8 shows CO_2 wellhead temperature variation with time for all three ramp-up injection cases. In terms of the minimum temperature in each case, it is clear that the formation of ice during the ramp-up injection process is likely, given that in all three cases simulated, the minimum fluid temperature falls well below 0 ^oC at the top of the wellbore. Such condition poses the risk of ice formation and ultimately well blockage in present of a sufficient amount of water molecules.

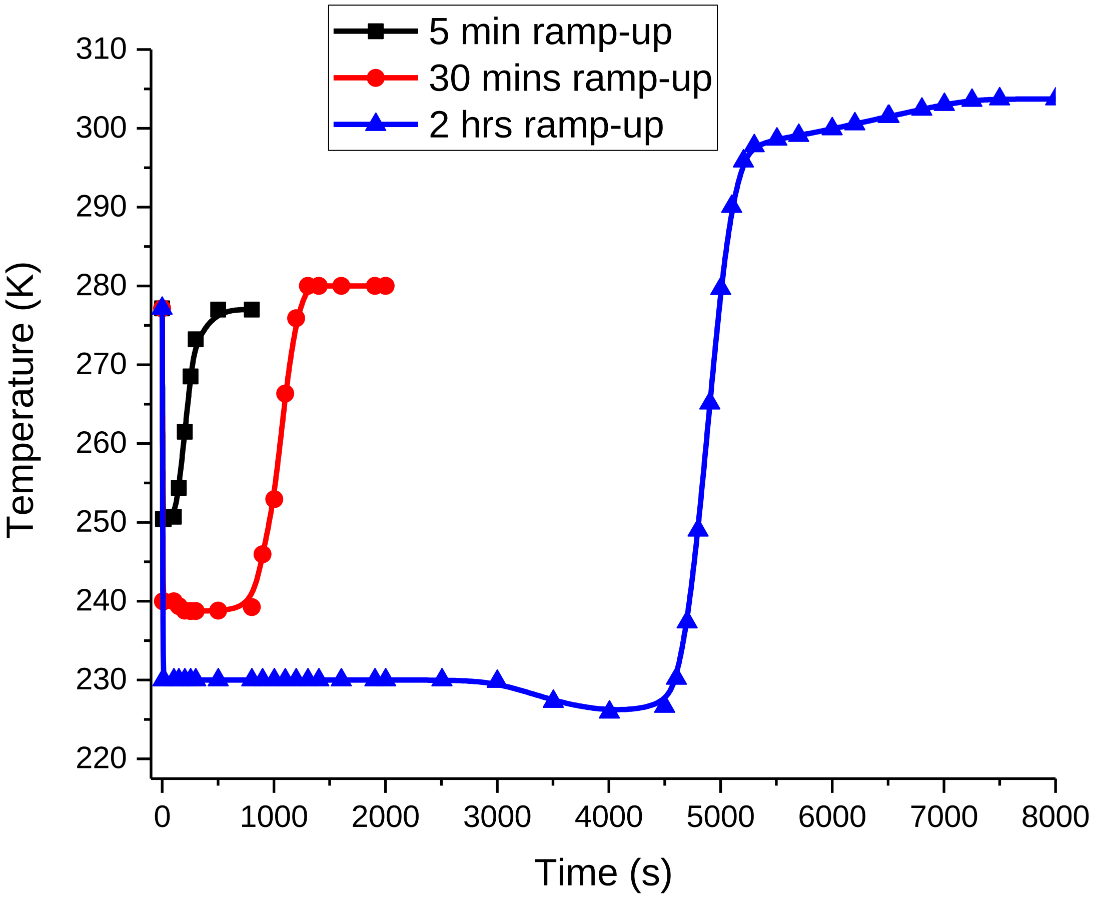


Figure 5.8: CO_2 wellhead temperature variation with time for all three ramp-up injection cases. Following Table 4.1 (feed temperature = 277.15 K, feed pressure = 115 bara)

Figures 5.9 and 5.10 respectively show the corresponding results for the temperature and pressure profiles along the length of the well during the 5 mins ramping up process at various selected time intervals of 10, 100, 200 and 300 seconds. As can be seen in Figure 5.9, the pressure profiles show continues pressure build-up along the wellbore during the 5 mins injection ramp-up. The temperature profiles show a significant temperature drop for the 10, 100 and 200 s well profiles in comparison with the initial well temperature profile at 0 s. The well temperature profile at 300 s shows significant recovery approaching the initial well temperature profile.

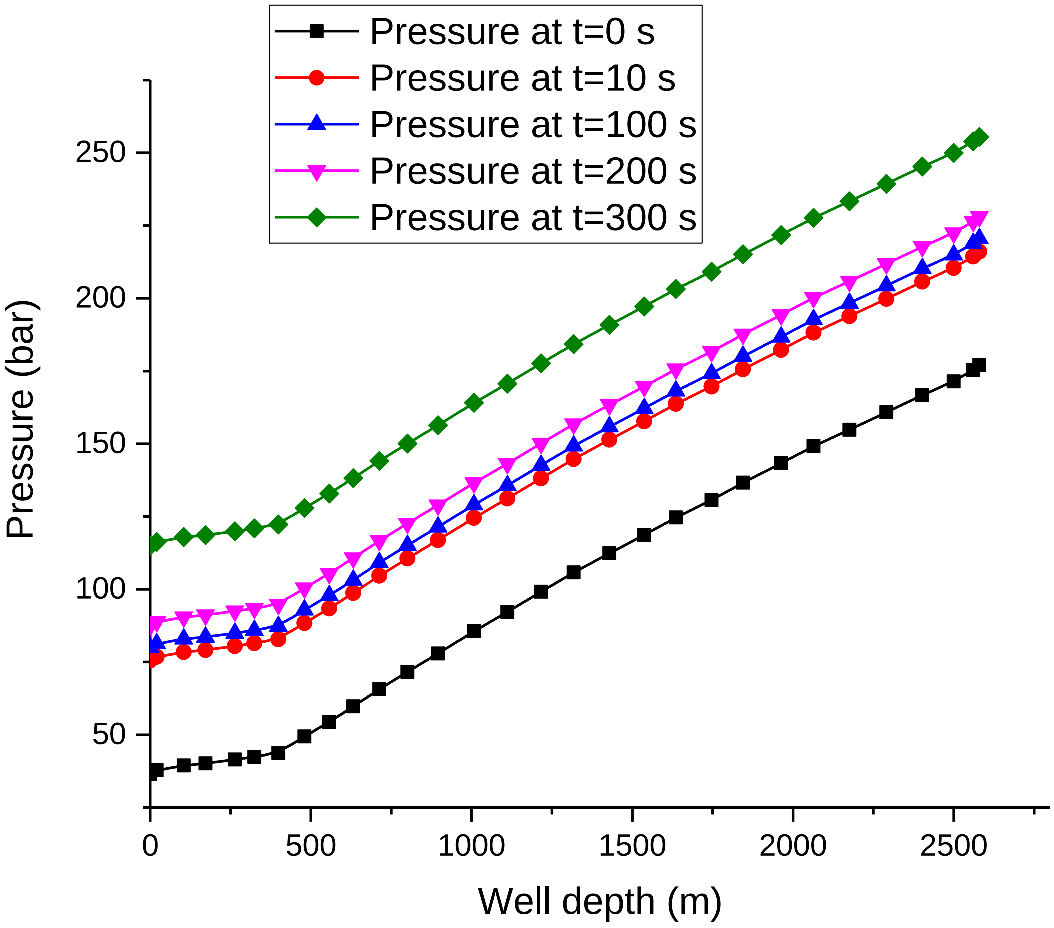


Figure 5.9: Five minutes ramping up injection well pressure profiles at varying intervals

As can been seen in Figure 5.10, the 300 s ramp-up duration temperature profile approaches the initial temperature profile signifying a recovery from the rapid drop when compared with the 10, 100 and 200 s cases. In other words, operating a fast injection ramp-up is recommended to rapidly increase the injection flowrate with time and consequently minimise the drop in temperature along the wellbore.

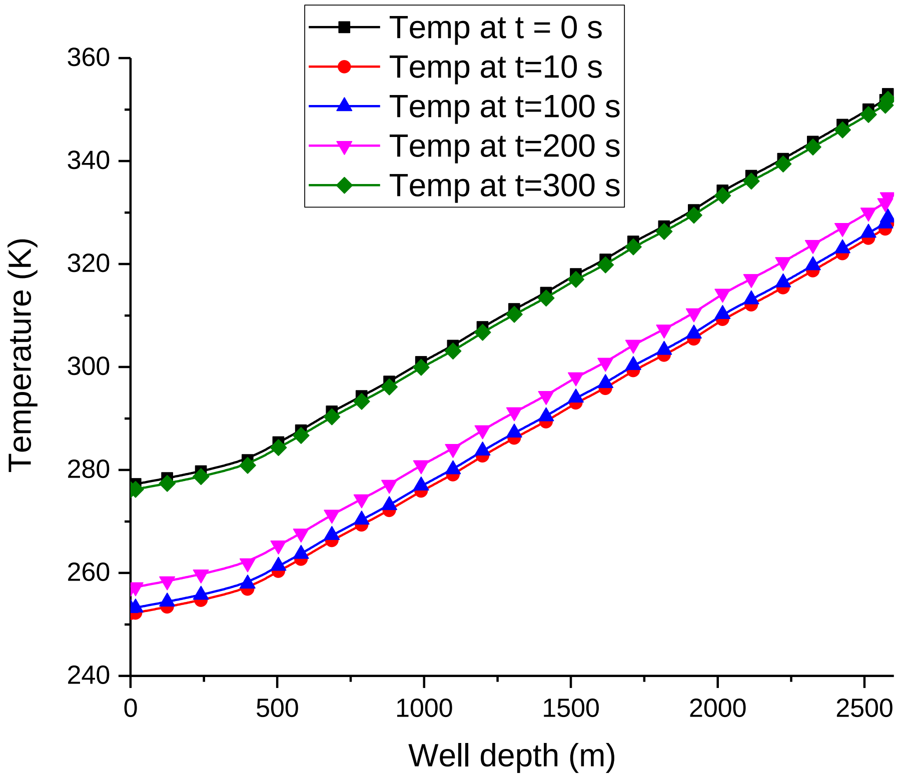


Figure 5.10: Five minutes ramping up injection well temperature profiles at varying intervals

5.2.4 Concluding remarks

In this section the HEM model was applied to investigate the impact of flow rates ramping up times corresponding to fast (5 mins), medium (30 mins) and slow (2 hrs) feed flow ramp-up on the injected CO_2 stream pressure and temperature profiles. In practice, the model can serve as a valuable tool for the development of optimal injection strategies and best-practice guidelines for the minimisation of the risks associated with the ramping up injection of CO_2 into depleted gas fields. Hence, the importance of investigating the injection flow rates ramp-up times. Despite this, varying injection ramping up times considered in this study has never been studied in details in any previous publications. As such, this study introduces a new area of consideration and gives a clearer insight on the impact of CO_2 injection flow rates ramping up times on the wellbore pressure and temperature profiles.

Based on the application of the HEM model to realistic test cases involving the ramping up of CO_2 injection flowrates from 0 to 38 kg/s into the Goldeneye depleted reservoir in the North

Sea following fast (5 mins), medium (30 mins) and slow (2 hrs) injection ramp-up times, the key findings of this sub-section can be summarised as follows:

- The degree of cooling along the injection well becomes less severe with a decrease in the injection ramping up duration. In other words, operating a fast ramping up injection is recommended (i.e. between 5 and 30 mins) rather than a much slower one (over 2 hrs);

- The formation of ice during the ramp-up injection process is likely, given that in all three cases considered the minimum fluid temperature falls well below 0 ^{o}C at the top of the well. This poses the risk of ice formation and ultimately well blockage should a sufficient quantity of water be present. Thus, for this case study injecting CO_2 at temperatures above 290 K at flowrate ramp-up duration of 5 mins will most likely minimise this risk.

- The minimum simulated CO_2 temperature and the corresponding pressures predicted at the top of well during the start-up injection flowrate ramp-up process are well close to the ranges where CO_2 hydrates formation would be expected. The ideal hydrate formation conditions are pressures and temperatures below 12.56 bara and 273.15 K however, tiny molecules of CO_2 hydrates begin formation at pressures and temperatures just below 49.99 bara and 283 K (Circone et al., 2003; Yang et al., 2012).

- Given that the temperature drops as low as -50 ^{o}C at the top of the well especially for the 2 hrs injection flow rate ramping up case, this could pose a system failure risk due to thermal shocking during the injection process. Being that such huge drop in temperature creates a sharp gradient with the outer temperature of the steel casing. Hence, a fast injection flow rate ramp-up operation is recommended in order to minimise the risk of injection system failure due to thermal shocking.

5.3 Modelling the impact of CO_2 stream impurities

Generally, “impurities in CO_2 captured from combustion based power generation with CCS can arise in a number of ways (Porter et al., 2015). Water is a major combustion product and is considered an impurity in the CO_2 stream. The elements inherently present in a fuel such as coal include sulphur, chlorine and mercury, and are released upon complete or incomplete combustion and form compounds in the gas phase which may remain to some extent as impurities in the CO_2 after it is captured and compressed (Porter et al., 2015). The oxidising agent used for combustion such as air may result in residual impurities of N_2, O_2 and Ar" (Porter et al., 2015).

The impacts of various impurities on CO_2 injection and storage can be categorised into two sub-sections: "physical and chemical. Physical impacts are those concerning phase behaviour, storage capacity, permeation flux, buoyancy, etc. Chemical impacts are those concerning rock-porosity related injectivity, caprock integrity, corrosion of well materials, hazardousness in the event of leakage, etc" (IEA, 2011) thus, "impurities in the CO_2 stream have the potential to affect the efficiency and safety of the storage systems. The presence of impurities is likely to have a significant effect on the phase behaviour, with implications for the design and operation of pipelines and injection wells" (IEA, 2011). Therefore, this study tests the impact of varying types and levels of impurities in the CO_2 stream on the pressure and temperature profiles at the wellhead. The analysis conducted shows the corresponding effect of the Joule-Thomson expansion on the varying stream impurities combination.

5.3.1 Literatures on CO_2 stream impurities

Much literature (see for eaxample Alex, 2016; Hajiw et al, 2018; IEAGHG, 2011; Mahgerefteh et al, 2012; Wang et al, 2011; Wang et al, 2015; Wetenhall et al, 2014) on the impacts of CO_2 stream impurities are available and mostly focussed on the rate of CO_2 injectivity, phase behaviour, storage capacity, transportation, compression and pipeline transportation. Wang et al, (2011) developed a simple formula to enable quick determination of the effects of impurities in CO_2 streams on geological storage. The study highlighted that "non-condensable impurities such as N_2, O_2 and Ar greatly reduce CO_2 storage capacity of geological formations, and there is a maximum reduction of the storage capacity at a certain pressure under a given temperature" (Wang et al., 2011). Their study showed those stream "impurities which are more condensable than CO_2, such as SO_2, can increase the storage capacity, and there is a maximum increase at a certain pressure under a given temperature; change of density caused by non-condensable gas impurities results in lower injectivity of impure CO_2 into geological formations" (Wang et al., 2011). The "importance of CO_2 impurities on the range of operation, safety considerations, fracture, cracking, corrosion control, dispersion in the event of a release, fluid density, operating pressure and temperature and the quantity of CO_2 that can be transported" was studied by Wetenhall et al., (2014). The study summarises the "impact of CO_2 impurity on CO_2 compression, liquefaction and transportation" following various capture technologies. Porter et al., (2015) studied available CO_2 purification technologies for the removal of impurities such as Hg and non-condensable compounds from raw oxy-fuel flue gas. They concluded that over "99% CO_2 purity levels are achievable using post-combustion capture technologies with low levels of the main impurities such as N_2, Ar and O_2". However, "CO_2 capture from oxy-fuel combustion and integrated gasification combined cycle power plants will need to take into consideration the removal of non-condensable acid gas species, and other contaminants. The actual level of CO_2 purity required will be dictated by a combination of transport and storage requirements, and process economics" (Porter et al., 2015).

The "presence of impurities in the CO_2 stream is therefore a well-established study and depending on the capture technology various amounts of impurities can be present in the captured CO_2 stream" (Wetenhall et al., 2014). Notably during CO_2 injection, the impact of varying types and levels of impurities in the CO_2 stream on the pressure and temperature profiles at the top of the well has not been clearly studied in any previous publication. At such, this study considers an entirely new area that is yet to be account for in any prior publication.

Table 5.1 shows the various CO_2 stream compositions according to post-combustion, pre-combustion and oxy-fuel capture technologies presented by Porter et al, (2015) is employed in this study CO_2 impurities simulations.

Table 5. 1: CO_2 stream % compositions based on the various capture technologies (Porter et al., 2015)

Component	**Post-combustion**	**Pre-combustion**	**Oxy-fuel**
CO_2 %v/v	(99.75)	(97.95)	(85)
Ar %v/v	(0.02)	(0.03)	(4.47)
N_2 %v/v	(0.09)	(0.9)	(5.80)
O_2 %v/v	(0.03)	-	(4.70)
CO ppmv	-	(400)	(50)
H_2S ppmv	-	(100)	-
H_2 ppmv	-	(20)	-
SO_2 ppmv	(20)	-	(50)
H_2O ppmv	(600)	(600)	(100)

NOx ppmv	(20)	-	(100)
CH_4 ppmv	-	(100)	-

These impurities are mainly identified to have significant impact on the behaviour of CO_2 during pipeline transportation and very significant changes in the thermodynamic properties of CO_2 stream are previously reported (see Brown et al, 2013; Martynov et al , 2013).

5.3.2 Impacts of impurities on phase behaviour of CO_2 streams

Figure 5.11 shows the CO_2 streams phase envelopes calculated using the Peng-Robinson Equation-of-State for the following cases (Wang et al., 2011): the CO_2 stream from oxyfuel combustion in a fluidized bed pilot plant combustor in Canmet Energy, containing 5.2 vol % O_2, 221 ppm CO, 1431 ppm SO_2 and 243 ppm NO; while the CO_2 stream from a zero-emissions process proposed by Canmet Energy, containing 1.05% CO, 1.7% SO_2, 0.32% H_2 and 690 ppm H_2S;

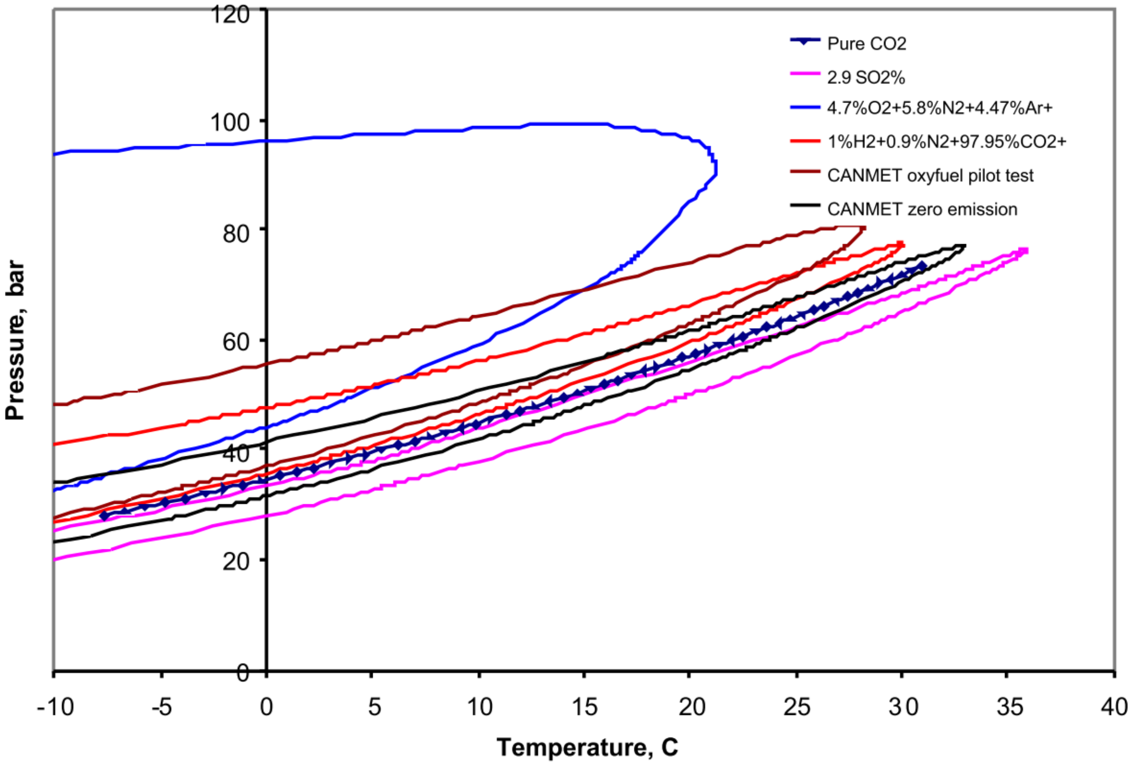


Figure 5.11: Calculated phase envelopes for pure CO_2 and CO_2 mixtures (Wang et al., 2011)

From Figure 5.11, it is observed that “the critical temperature and pressure of the mixtures different from that of the pure CO_2. N_2, O_2, Ar and H_2 show the greatest effect of increasing the saturation pressure of the liquid and decreasing the critical temperature. One extreme case is the mixture from oxyfuel combustion containing 5.8% N_2, 4.7% O_2, 4.47% Ar and other impurities at ppm level" (Wang et al., 2011). The "critical temperature decreases by about 10°C in comparison with that of pure CO_2, and the liquefaction pressure increases by over 50 bara" (Wang et al., 2011). On the other hand, the presence SO_2 in the mixture shows a "decrease in the saturation pressure and an increase of the critical temperature, as is expected from the high critical temperature of pure SO_2 (157.6°C)" (Wang et al., 2015). It can also be seen that low-concentration impurities, such as CO and NOx would not significantly affect the phase. Therefore, before the steady injection of CO_2 into the reservoir can be started, it is necessary to perform time-dependent operations to estimate important wellbore pressure and temperature behaviours of the injection of various CO_2 stream impurities combination. It is therefore of paramount importance to be able to predict the impact of the presence of these CO_2 stream impurities in terms of pressure and temperature behaviour in the well, to characterise and quantify potential risks.

5.3.3 Results and discussion

The results obtained following the simulation of the transient flow model for the injection of CO_2 with various stream impurities into highly depleted oil/gas fields are presented and discussed in detail. The pressure and temperature profiles corresponding to the varying impurities combinations at the top of the well are shown in Figures 5.12 and 5.13 respectively. The 5 mins injection flow rate ramping up case for pure CO_2 (Figures 5.1 and 5.5) is regarded as the reference case for the study of the various CO_2 impurities effect..

Figure 5.1 shows the pure CO_2 pressure variation with time at the top of the well for 5 mins ramping up injection time up to a maximum flow rate of 38 kg/s. The pressure profile for pure CO_2 injection shows a rapid drop in pressure of the incoming CO_2 from 115 bara to as low as 74 bara. Following the initial pressure drop is a recovery over the ramping up duration until some steady state is attained. Figure 5.1 therefore forms the bases for the CO_2 stream impurities impact assessment presented in Figure 5.12.

Figure 5.12 shows the variations in pressure with time at the top of the well for pure CO_2 and varying stream impurities combination. As can be seen, the pressure profiles of the mixtures are marginally different from that of the pure CO_2. Oxyfuel and pre-combustion captured CO_2 mixtures show the greatest effect of pressure drop at the start of injection. The mixture from oxyfuel combustion containing 5.8% N_2, 4.7% O_2, 4.47% Ar and other impurities at ppm level recorded the highest pressure drop of 69 bara from the inlet pressure of 115 bara. On the other hand, pre-combustion captured CO_2 stream containing 0.09 % N_2, 0.03 % O_2, 0.02 % Ar and other impurities at ppm level showed a minimal drop in pressure.

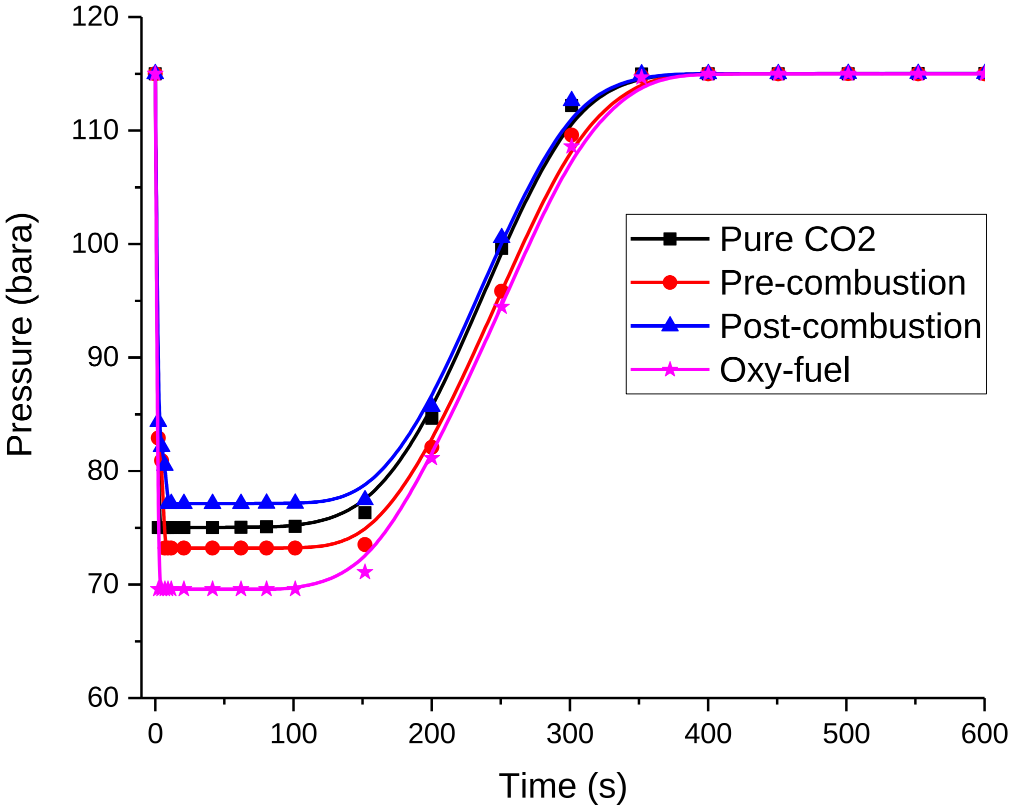


Figure 5.12: Pressure variation with time at the top of the well for pure CO_2 and varying stream impurities mass flow rate

Also, Figure 5.5 shows the corresponding pure CO_2 temperature variation with time at the top of the well for 5 mins ramping up injection time up to a maximum injection rate of 38 kg/s. The temperature profile for pure CO_2 injection shows a rapid drop in temperature of the incoming CO_2 from 277 K to as low as 251 K. The temperature dropped rapidly from 277 K to 251 K within the first 120 sec. The initial temperature drop is followed by a recovery over the ramp-up duration until steady state is attained in all cases. Figure 5.5 therefore forms the bases for the CO_2 stream impurities impact assessment on the wellhead injection temperature presented in Figure 5.13.

Figure 5.13 shows the corresponding variations in the CO_2 temperature with time at the top of the well for pure CO_2 and varying stream impurities combination. As can be seen, the temperature profiles for the mixtures are quite different from that of the pure CO_2. Oxyfuel and pre-combustion captured CO_2 mixtures show the greatest impact on temperature drop at the start of injection. The mixture from oxyfuel combustion containing 5.8% N_2, 4.7% O_2, 4.47% Ar and other impurities at ppm level recorded the highest temperature drop of 226 K from the inlet temperature of 277 K. On the other hand, pre-combustion captured CO_2 stream containing 0.09 % N_2, 0.03 % O_2, 0.02 % Ar and other impurities at ppm level showed a minimal drop in temperature. The results clearly show that the presence of impurities in the CO_2 stream will adversely impact the temperature drop along the wellbore. For instance, the oxyfuel captured CO_2 stream with higher levels of impurities may likely impose temperatures below 0 ^{o}C all the way down to the bottom of the well. The effect of such low temperatures on the injection system is highly significant and accounted as a critical factor for safe injection of CO_2. Based on the fact that there is a possibility of interstitial water molecules present within the wellbore during injection and at the reservoir inlet, the formation of ice is likely and may pose significant safety risks.

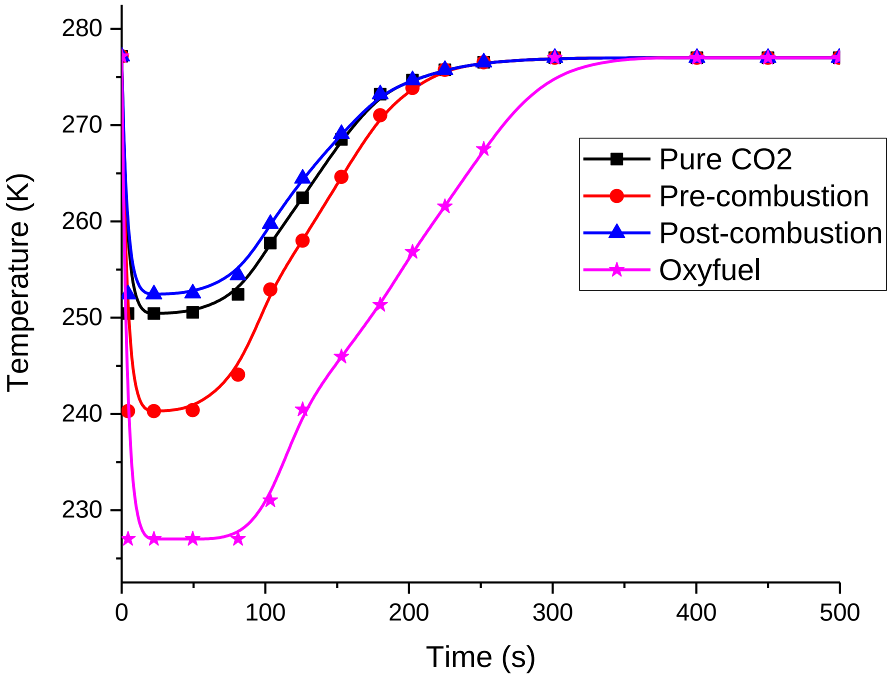


Figure 5.13: Temperature variation with time at the top of the well for pure CO_2 and varying stream impurities combination

5.3.4 Concluding remarks

In this section, the study has shown the impact of CO_2 stream impurities on the wellhead pressure and temperature profiles during the injection of CO_2 into depleted gas fields. Pre-combustion, post-combustion and oxyfuel captured CO_2 streams are considered. The following conclusions can be summarised based on the results presented above:

- The wellhead pressure profiles of the CO_2 mixtures showed marginal differences from that of the pure CO_2. Oxyfuel and pre-combustion captured CO_2 mixtures showed the greatest effect of pressure drop at the start of injection. The CO_2 captured by oxyfuel combustion containing 5.8% N_2, 4.7% O_2, 4.47% Ar and other impurities at ppm level recorded the highest pressure drop of 69 bara from the inlet pressure of 115 bara. On the other hand, pre-combustion captured CO_2 stream containing 0.09 % N_2, 0.03 % O_2, 0.02 % Ar and other impurities at ppm level showed minimal pressure drop compared with that of post-combustion and oxyfuel.
- The presence of impurities in the CO_2 stream showed significant effect on the temperature profile as can be seen in Figure 5.13. The Joule-Thomson cooling effect at the start of the injection process showed great impact on the CO_2 stream impurities. As seen in the results, oxyfuel and pre-combustion captured methods corresponding to high proportion of CO_2 stream impurities showed the greatest impact on temperature drop at the start of injection. The mixture from oxyfuel combustion containing 5.8% N_2, 4.7% O_2, 4.47% Ar and other impurities at ppm level recorded the highest temperature drop of 226 K (–47 ^{o}C) from the inlet temperature of 277 K. On the other hand, pre-combustion captured CO_2 stream containing 0.09 % N_2, 0.03 % O_2, 0.02 % Ar and other impurities at ppm level showed a minimal drop in temperature to 240 K (–33 ^{o}C). Based on these low temperatures and the fact that there is a possibility of interstitial water molecules within the wellbore during injection and at the reservoir inlet, the formation of ice is likely and may pose significant safety risks.

Chapter 6: Homogeneous Relaxation Modelling of CO_2 Injection into depleted Gas Fields

6.1 Introduction

In the previous chapters the Homogeneous Equilibrium Mixture (HEM) model for simulating the transient flow phenomena taking place during the injection of CO_2 into depleted gas fields was presented. The HEM model is characterised by a two-phase mixture being injected at homogeneous equilibrium, which means instantaneous interface mass, momentum and energy exchange. At such, the constituent gas and liquid phases are assumed to remain at the same pressure, temperature and velocity. The corresponding fluid flow was described using a single set of the mass, momentum and energy conservation equations.

This chapter on the other hand, presents the development and verification of a Homogeneous Equilibrium Relaxation Mixture (HERM) model, where mass, momentum, and energy conservation equations are considered for two-phase flow. It accounts for phase and flow dependent fluid/wall friction and heat transfer, variable well cross sectional area as well as deviation of the well from the vertical. The HERM contrary to the HEM presents an additional equation which accounts for the thermodynamic non-equilibrium relaxation time in two-phase multi-component mixtures. The opening of the upstream flow regulator valve into the injection well (see Figure 1.2) is modelled as an isenthalpic expansion process; whilst at the well outlet, a pre-defined site-characteristic pressure-mass flow rate correlation is used to simulate the migration of the CO_2 into the geological substrate. Furthermore to the application of the HEM model, the HERM model developed is also used to perform a detailed sensitivity analysis of the most important parameters affecting the CO_2 flow behaviour, including the wellbore diameter variations, well inclination, upstream temperature, pressure and time variant mass flow rate; the latter being representative of the feed pressure ramping up process. These results are intended to demonstrate the efficacy of the injection model presented as a powerful tool for the development of optimal injection strategies and best-practice guidelines for minimising the risks associated with the start-up injection of CO_2 into depleted gas fields.

6.2 HERM model formulations

6.2.1 Fluid dynamics

The following gives a detailed account of the HERM employed for the simulation of the time-dependent flow of CO_2 in injection wells. Given that this model extends those presented in Brown et al., (2013) and Brown et al., (2015), the latter reference is closely followed for the presentation of all the quantities of interest. The system of four partial differential equations for the CO_2 liquid/gas mixture, to be solved in the well tubing, can be written in the well-known conservative form as follows:

$$\frac{\partial}{\partial t}\boldsymbol{U} + \frac{\partial}{\partial z}\boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S_1} + \boldsymbol{S_2} \tag{6.1}$$

where

$$\frac{\partial}{\partial t}\rho A + \frac{\partial}{\partial z}\rho u A = 0 \quad \dots\dots\dots\dots \; Mass\ conservation$$

$$\frac{\partial}{\partial t}\rho u A + \frac{\partial}{\partial z}(\rho u^2 A + AP) = P\frac{\partial A}{\partial z} + A(F + \rho\beta g) \dots\dots\dots. \; Momentum\ conservation$$

$$\frac{\partial}{\partial t}\rho E A + \frac{\partial}{\partial z}\rho u H A = P\frac{\partial A}{\partial z} + A(Fu + \rho u\beta g + Q) \dots\dots\dots \; Energy\ conservation$$

(6.2)

In the above, the set of three equations correspond to mass, momentum, and energy conservation, respectively. The fourth equation describes the fact that the cross-sectional area A is, at any location along the well, constant in time, but might vary along the depth of the well. Moreover, u and ρ are the mixture velocity and density, respectively. P is the mixture pressure, while E and H represent the specific total energy and total enthalpy of the mixture, respectively. They are defined as:

$$E = e + \frac{1}{2}u^2 \tag{6.3}$$

$$H = E + \frac{P}{\rho} \tag{6.4}$$

where e is the specific internal energy. In addition, z denotes the space coordinate, t the time, F the viscous friction force, Q the heat flux, and g the gravitational acceleration. In the case of the HERM, “the same assumption of mechanical equilibrium with no phase slip, is retained. However, non-equilibrium liquid-vapour transition is accounted for by a relaxation to thermodynamic equilibrium through the following equation for the vapour mass fraction” (Brown et al., 2013; Brown et al., 2015; Downar-Zapolski et al., 1996):

$$\frac{\partial x}{\partial t} + u\,\frac{\partial x}{\partial z} = \frac{x_e - x}{\theta} \tag{6.5}$$

where x is the vapour quality of the mixture and θ is a relaxation time accounting for the delay in the phase change transition. The mixture density ρ is defined as (Brown et al., 2013; Brown et al., 2015):

$$\frac{1}{\rho} = \frac{x}{\rho_{sv}(P)} + \frac{1-x}{\rho_{ml}(P, e_{ml})} \tag{6.6}$$

while the mixture internal energy is defined as:

$$e = x e_{sv}(P) + (1-x) e_{ml} \tag{6.7}$$

where the subscripts sv and ml refer to the saturated vapour and meta-stable (super-heated) liquid phases, respectively(Brown et al., 2013)

The system of partial differential equation (6.2) is an extension of the work previously done in, (Brown et al., 2013, 2015), by accounting for both a variable cross-sectional area and additional source terms. Analysing in more detail the various source terms appearing on the right-hand side of (6.2), the frictional loss, F in equation (6.2) can be expressed as

$$F = -f_w \frac{\rho u^2}{D_p} \tag{6.8}$$

where f_w is the Fanning friction factor, calculated using Chen’s correlation (N H Chen, 1979), and D_p is the internal diameter of the pipe.

The gravitational term includes

$$\beta = \beta(z) = \rho g \sin\theta \tag{6.9}$$

which accounts for the possible well deviation. In practice, a well is often not drilled in the perfectly vertical direction, as a deviation can occur caused by geological conditions, specifics of the drilling technology and the process conditions (Chabook et al., 2015; Wood 2016). The well deviation, and consequently the correct gravitational term in Equation (6.9), is usually expressed through two different quantities: True Vertical Depth (TVD) and Measured Depth (MD). The former denotes the vertical distance from the wellhead to a point in the well path. Such quantity is calculated from directional survey data. The latter denotes the actual length of the well, and it is always greater than the corresponding TVD, given the wellbore curvature.

In Equation (6.2) the source term, Q accounts for the heat exchange between the fluid and the well wall. The corresponding heat transfer coefficient, η is calculated using the well-known Dittus-Boelter correlation (Dittus & Boelter, 1930):

$$\eta \ = 0.023 \, Re^{0.8} Pr^{0.4} \frac{k}{D_p} \tag{6.10}$$

where k, Re and Pr are the thermal conductivity, Reynold’s number and Prandtl’s number for the fluid. The heat exchanged between the fluid and the wall is given by:

$$Q = \frac{4}{D_p} \eta (T_w - T) \tag{6.11}$$

where T_w and T are the fluid and wall temperatures respectively. Note that $T_w = T_w(z,t)$, i.e. T_w is not assumed constant, but variable with time and space. The transient heat conduction in the coordinate r (the radial direction from the axis of the well towards the well layers and the formation surrounding the well) is solved using the following equation (Lund, Torsæter, & Munkejord, 2015):

$$\rho(r) c_p(r) \frac{\partial}{\partial t} T(r,t) = \frac{1}{r} \frac{\partial}{\partial r} \left(r\kappa(r) \frac{\partial}{\partial r} T(r,t) \right) \tag{6.12}$$

where T, ρ, c_p and κ represent temperature, density, heat capacity and heat conductivity calculated at the radial coordinate r.

In order to close the HERM model system of equations Equation (6.1) and (6.2), the pertinent thermodynamic properties for vapour and liquid phases are calculated with the aid of the Peng–Robinson Equation-of-State (EoS) (Peng & Robinson, 1976). It should be noted that while other more accurate EoS exist for pure CO_2 exist e.g. (Diamantonis & Economou, 2011; Span & Wagner, 1996), the computational workload required to use them in conjunction with the HERM model is currently prohibitive. The PR Equation-of-State is given by

$$P = \frac{R}{v - b} - \frac{c\delta}{v^2 + 2bv + \ - b^2} \tag{6.13}$$

where v, T, and R are respectively the specific volume, temperature, and the gas constant. c, b and δ are given by Brown et al., (2013):

$$c = 0.45724\, \frac{(RT_c)^2}{P_c} \tag{6.14}$$

$$b = 0.0778\, \frac{RT_c}{P_c} \tag{6.15}$$

$$\delta = \left[1 + \left(0.37464 + 1.54226\omega - 0.2699\omega^2\right)\left(1 - \sqrt{\frac{T}{T_c}}\right)\right] \tag{6.16}$$

where P_c, T_c and ω are respectively the critical pressure, the critical temperature and the acentric factor for CO_2 given by Brown et al., (2013) and Brown et al., (2015) as:

$P_c = 72.44016$ bara, $T_c = 304.35$ K and $\omega = 0.2236$.

From Equation (6.13), the internal energy may be calculated by (Poling et al., 2001):

$$e - e_{ig} = \int_{-\infty}^{v} \left[T\left(\frac{\partial P}{\partial T}\right)_v - P\right] dv \tag{6.17}$$

where the subscript ig refers to ideal gas. The mixture speed of sound c is defined through the relation given by Brown et al., (2013):

$$\frac{1}{\rho c^2} = \frac{x}{\rho_v c_v^2} + \frac{1-x}{\rho_l c_l^2} \tag{6.18}$$

6.3 Numerical method

As the HERM system of Equations (6.2) cannot be solved analytically, a suitable numerical method has been employed within the Finite Volume framework. More specifically, in order to accurately capture the rich flow dynamics, the AUSM+-up Flux Vector Splitting scheme (Liou, 2006) has been applied to the fluid problem. The solution domain is first divided into N cells and integrated over the i-th computational cell $\left[z_{i-\frac{1}{2}}, z_{i+\frac{1}{2}}\right], i = 1, \ldots, N$, to yield the semi-discrete formulation:

$$\frac{d\boldsymbol{U}_i}{dt} = -\frac{1}{\Delta z}\left(\boldsymbol{F}_{i+\frac{1}{2}} - \boldsymbol{F}_{i-\frac{1}{2}}\right) + \boldsymbol{S}_{1,i} + \boldsymbol{S}_{2,i} \tag{6.19}$$

where $i+\frac{1}{2}$ denotes the interface between cells i and $i+1$, at which the inter-cell flux $\boldsymbol{F}_{i+\frac{1}{2}}$ has to be computed. Following the standard AUSM method (Liou, 2006), the conservative flux vector is split into convective and pressure fluxes (Brown et al., 2013):

$$\boldsymbol{F}_{i+\frac{1}{2}} = \boldsymbol{f^c}_{i+\frac{1}{2}} + \boldsymbol{f^p}_{i+\frac{1}{2}} \tag{6.20}$$

where

$$\boldsymbol{f^c}_{i+\frac{1}{2}} = \begin{pmatrix} \rho u A \\ \rho u^2 A \\ \rho u H A \\ 0 \end{pmatrix}_{i+\frac{1}{2}}, \quad \boldsymbol{f^p}_{i+\frac{1}{2}} = \begin{pmatrix} 0 \\ AP \\ 0 \\ 0 \end{pmatrix}_{i+\frac{1}{2}} \tag{6.21}$$

6.3.1 Convective flux discretisation

From Equation (621) the convective flux can be written explicitly as

$$\boldsymbol{f^c} = \dot{m}\begin{pmatrix} 1 \\ u \\ H \\ 0 \end{pmatrix} = \dot{m}\boldsymbol{\Psi} \tag{6.22}$$

where $\dot{m}$ is the area-weighted mass flux:

$$\dot{m} = \rho u A \tag{6.23}$$

The numerical flux at cell interface $i+\frac{1}{2}$ is then defined as:

$$\boldsymbol{f^c}_{i+\frac{1}{2}} = \dot{m}^*\boldsymbol{\Psi}^* = \dot{m}^*\frac{1}{2}(\boldsymbol{\Psi}_i + \boldsymbol{\Psi}_{i+1}) + \frac{1}{2}|\dot{m}^*|(\boldsymbol{\Psi}_i - \boldsymbol{\Psi}_{i+1}) \tag{6.24}$$

In order to express $\dot{m}$, the interface speed of sound $a_{i+\frac{1}{2}}$ and the left and right Mach numbers are defined as follows:

$$a_{i+\frac{1}{2}} = \frac{a_i + a_{i+1}}{2}, \qquad M_i = \frac{u_i}{a_{i+\frac{1}{2}}}, \qquad M_{i+1} = \frac{u_{i+1}}{a_{i+\frac{1}{2}}} \tag{6.25}$$

The interface Mach number is next defined as:

$$\widetilde{M}_{i+\frac{1}{2}} = \mathcal{M}_4^+(M_i) + \mathcal{M}_4^-(M_{i+1}) \tag{6.26}$$

where $\mathcal{M}^+$ and $\mathcal{M}^-$ are the polynomials introduced by Liou (1996) and the subscripts indicating the order of the polynomial used:

$$\mathcal{M}_1^{\pm} = \frac{1}{2}(M \pm |M|) \tag{6.27}$$

$$\mathcal{M}_2^{\pm} = \pm\frac{1}{4}(M \pm 1)^2 \tag{6.28}$$

$$\mathcal{M}_4^{\pm} = \begin{cases} \dfrac{\mathcal{M}_1^{\pm}}{M}, & \text{where } |M| \geq 1 \\ \pm\,\mathcal{M}_1^{\pm}\left(1 \mp 16B\mathcal{M}_2^{\mp}\right) \end{cases} \tag{6.29}$$

with $B = \frac{1}{8}$.

In order to approximate numerically the area-weighted mass flux introduced in Equation (6.23), it is defined by:

$$\dot{m}^* = a_{i+\frac{1}{2}}\left[\frac{(\rho A)_i}{2}\left(M_{i+\frac{1}{2}} + \left|M_{i+\frac{1}{2}}\right|\right) + \frac{(\rho A)_{i+1}}{2}\left(M_{i+\frac{1}{2}} - \left|M_{i+\frac{1}{2}}\right|\right)\right] \tag{6.30}$$

where $M_{i+\frac{1}{2}} = \widetilde{M}_{i+\frac{1}{2}}$. However, at low Mach numbers this approximation approaches a central difference scheme and can suffer from odd-even decoupling (Liou, 1996). In order to avoid poor numerical solutions, a velocity-based dissipation is added to $\widetilde{M}_{i+\frac{1}{2}}$:

$$M_{i+\frac{1}{2}} = \widetilde{M}_{i+\frac{1}{2}} - K_p \ \max(1 - \bar{M}^2, 0)\frac{P_{i+1} - P_i}{\bar{\rho}\bar{a}^2} \tag{6.31}$$

where $\bar{M}$, $\bar{\rho}$ and $\bar{a}$ represent the arithmetic averages of the values attained by Mach number, density and speed of sound in cells i and $i+1$. K_p is a constant, which is set to unity in this study.

While the above scheme has proved to be remarkably robust, in order to avoid instabilities at discontinuities in the cross-sectional area following Niu et al., (2008) and replacing $\widetilde{M}_{i+\frac{1}{2}}$ in Equation (6.26) with

$$\widetilde{M}_{i+\frac{1}{2}} = \mathcal{M}_1^{+}(M_i) + \mathcal{M}_1^{-}(M_{i+1}) \tag{6.32}$$

6.3.2 Pressure flux discretisation

In Equation (6.21) the pressure flux $\boldsymbol{f}^p{}_{i+\frac{1}{2}}$ was introduced. According to the AUSM+ splitting, its non-zero element is defined as:

$$(AP)_{i+\frac{1}{2}} = A_{i+\frac{1}{2}}(\mathcal{P}_5^+(M_i) \cdot P_i + \mathcal{P}_5^-(M_{i+1}) \cdot P_{i+1}) \tag{6.33}$$

where the polynomials $\mathcal{P}_5^{\pm}$ are those introduced in Liou (1996) and the subscript indicates the order of the polynomial used:

$$\mathcal{P}_5^{\pm} = \begin{cases} \frac{\mathcal{M}_1^{\pm}}{M}, \text{ where } |M| \geq 1 \\ \pm\, \mathcal{M}_2^{\pm}\left(2 \mp M - 16CM\mathcal{M}_2^{\mp}\right) \end{cases} \tag{6.34}$$

with $C = \frac{3}{16}$.

In Equation (6.33) $A_{i+\frac{1}{2}}$ can attain two different values in the two cells separated by the interface $i + \frac{1}{2}$. Specifically, for the i-th cell, $A_{i+\frac{1}{2}}$ takes the value of the area on the left of the interface, i.e. $A_{i+\frac{1}{2},i}$, while in the case of the $(i+1)$-th cell, the value on the right of the interface is set to $A_{i+\frac{1}{2}}$, i.e. $A_{i+\frac{1}{2},i+1}$. As such, the strategy will permit the discontinuous variations in the area at the interface $i + \frac{1}{2}$.

Then, "a dissipation final term $P_{i+\frac{1}{2}}$ is added. The form of the dissipation term developed by Liou, (2006) has been widely used for both single- and two-phase flows (Paillère et al., 2003; Robbins et al., 2013) and is given" by Brown et al., (2013):

$$\begin{aligned} P_{i+\frac{1}{2}} = {} & \mathcal{P}_5^+(M_i)(P)_i + \mathcal{P}_5^-(M_{i+1})(P)_{i+1} \\ & - K_u\mathcal{P}_5^+(M_i)\mathcal{P}_5^-(M_{i+1})(\rho_i + \rho_{i+1})a_{i+\frac{1}{2}}(u_{i+1} - u_i) \end{aligned} \tag{6.35}$$

where K_u is constant and set to unity in most cases. Otherwise, to maintain stability at the cross-sectional areas discontinuities, Equation (6.35) is modified to incorporate the cross-sectional area dependency term:

$$\begin{aligned} P_{i+\frac{1}{2}} = {} & \mathcal{P}_5^+(M_i)(P)_i + \mathcal{P}_5^-(M_{i+1})(P)_{i+1} \\ & - K_u\mathcal{P}_5^+(M_i)\mathcal{P}_5^-(M_{i+1})(\rho_i + \rho_{i+1})a_{i+\frac{1}{2}}\frac{A_{i+\frac{1}{2},i+1}u_{i+1} - A_{i+\frac{1}{2},i}u_i}{\max\left(A_{i+\frac{1}{2},i+1}, A_{i+\frac{1}{2},i}\right)} \end{aligned} \tag{6.36}$$

6.3.3 Non-conservative fluxes

The “discretisation of the source term, $\boldsymbol{S_1}$ containing the non-conservative derivative requires special attention to ensure numerical stability”, the source term is given by Brown et al., (2013) and Brown et al., (2015):

$$\boldsymbol{S}_{1,i} = \frac{P_i}{\Delta \mathrm{z}} \begin{pmatrix} 0 \\ \Delta_i A \\ 0 \\ 0 \end{pmatrix} \tag{6.37}$$

Liou et al., (2008) by analogy studied the multi-fluid equations with the non-conservative terms and discretised the non-zero term in Equation (6.37) as follows:

$$\Delta_i A = A_{i+\frac{1}{2},i} - A_{i-\frac{1}{2},i} \tag{6.38}$$

where the cell within is taken with the areas at the interfaces. Liou et al., (2008) holds also that it is easy to show analogously the non-disturbance relation, i.e. following the discretisation steady-state conditions with $u = 0$ and $P = const$ to produce the relation:

$$\frac{\partial AP}{\partial z} = P\frac{\partial A}{\partial z} \tag{6.39}$$

6.3.4 Temporal discretisation

Equation (6.19) is integrated over the time interval, Δt using an Explicit Euler method. The vapour quality relaxation equation (6.5) is then solved as shown in Brown et al., (2013).

6.3.5 Boundary conditions

So far the methodology for updating the cell value $\boldsymbol{U}_i$ has been presented assuming that the neighbouring cell values $\boldsymbol{U}_{i-1}$ and $\boldsymbol{U}_{i+1}$ in order to compute the inter-cell fluxes $\boldsymbol{F}_{i+\frac{1}{2}}$ and $\boldsymbol{F}_{i-\frac{1}{2}}$. Given that in practice a finite set of grid cells covering the computational domain exists, in the first and last cells, will not have the required neighbouring information. In order to close the flow equations, the relevant boundary conditions are imposed by adding a ghost cell at either end of the well (Randall J LeVeque, 2002). All the thermodynamic properties in the ghost cells are set at the beginning of each time step according to the appropriate boundary condition. It is noted that the boundary conditions explicitly depend on time, given the unsteady nature of the start-up injection of CO_2.

The following gives a detailed account of both inflow and outflow boundary conditions.

The mass flow rate at the well injector is determined by the well operator according to the upstream conditions of the transportation system that supplies CO_2 to the storage site (Böser & Belfroid, 2013). The CO_2 arrives at a certain pressure and temperature and undergoes an isenthalpic process through a choke valve. The implementation of the inflow boundary condition is represented in **Error! Reference source not found.**1. Where the temporal e volution of the thermodynamic variables in the ghost cell at the top of the computational domain is analysed. At every time step t, a system of three nonlinear equations represented by the following three conditions using the Matlab DASSL solver are resolved:

1. the pressure in the ghost cell will equate the pressure in the first computational cell at time $t - \Delta t$;
2. the CO_2 feed stream, arriving with predefined upstream pressure and temperature, undergoes an isenthalpic process. The enthalpy in the ghost cell will therefore equate the enthalpy of the feed stream.
3. the mass flow rate, as imposed by the well operator, is preserved.

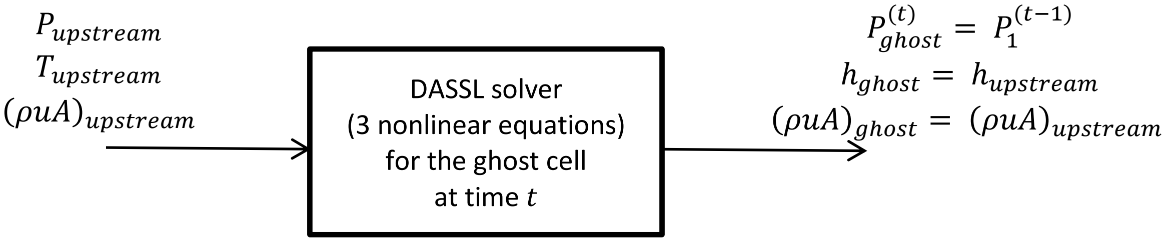


Figure 6.1: Schematic representation of the isenthalpic inflow condition in the ghost cell at the top of the computational domain.

Note that two of the above conditions carry information coming from outside the domain (namely, the conservation of enthalpy and the mass flow rate prescribed by the well operator), while only one piece of information is provided from the interior of the domain. This is in line with the analysis of time-dependent boundary conditions for subsonic inflows (Thompson, 1987 and 1990).

The outflow boundary condition has been modelled according to an empirical pressure-flow relationship derived from reservoir properties (Li et al., 2015; Shell, 2015):

$$\tilde{A} + \tilde{B} \times M + \tilde{C} \times M^2 = P_{BHF}^2 - P_{res}^2 \qquad (41)$$

where $\tilde{A}$ is the minimum pressure required for the flow to start from the well into the reservoir, $\tilde{B}$ and $\tilde{C}$ are site-specific dimensional constants, M is the instantaneous mass flow rate at the bottom-hole, P_{BHF}^2 is the instantaneous bottom-hole pressure, and P_{res}^2 is the reservoir static pressure. It represents a more sophisticated condition than a standard, linear relationship between the bottom-hole pressure and the flow rate given by the so-called injectivity index (Linga & Lund, 2016).

6.3.6 Heat exchange with the well layers

To reflect reality, eleven outer well layers are taken into account for the 1D radial heat transfer calculations (Equations (10) to (12)). These include the tubing, A-annulus water, production casing, oil-based mud, surface casing, B-annulus water, cement, conductor casing, mudstone, sandstone, and chalk. The various layers are discretised into a number of points in the radial direction from the well axis, and at every point the numerical finite volume heat transfer model computes the instantaneous temperature. Two boundary conditions, namely at the wall where the first well layer is in contact with the CO_2 mixture, and that for the outer layer in contact with the surrounding formation are employed. For the inner wall, the heat flux presented by Equation (6.11) is prescribed for the outer wall, the formation temperature is assumed to be known from geological surveys.

6.4 Results and discussion

Given the availability of the pertinent data for modelling purposes, the abandoned Peterhead CCS project (Shell, 2015) is used as a case study in this work. The project would have involved capturing one million tonnes of CO_2 per annum for 15 years from an existing combined cycle gas turbine located at Peterhead Power Station in Aberdeenshire, Scotland followed by pipeline transportation and injection into the depleted Goldeneye reservoir. However, given the decision of the UK Government to withdraw the capital budget for the Carbon Capture and Storage Competition (Cotton et al., 2017), the project was cancelled.

The data used for the case study include the well depth and geometry, pressure and temperature profiles, along with the surrounding formation characteristics as presented in Li et al., (2015) and Shell, (2015) and summarised in Table 6.1.

Table 6.1: Main CO_2 injection simulation parameters (Li et al., 2015; Shell, 2015)

Parameter	Value
Well length	2582m
Internal diameter	0 – 800m depth: 0.125m 800 – 2582m depth: 4 different cases: 0.076, 0.07, 0.065 and 0.0625m
Well deviation	2 different cases: perfectly vertical configuration or deviation as in (Li et al., 2015)
Upstream pressure	115 bara
Upstream temperature	Case A: 278.15 K Case B: 283.15 K
Inlet mass flow rate	Case 1: linearly ramped-up to 33.5 kg/s in 5 mins Case 2: linearly ramped-up to 33.5 kg/s in 30 mins Case 3: linearly ramped-up to 33.5 kg/s in 2 hours
Outflow (see equation 41)	$\tilde{A} = 0$ $\tilde{B} = 1.3478 \times 10^{12}$ Pa^2s/kg $\tilde{C} = 2.1592 \times 10^{10}$ Pa^2s^2/kg^2 $P_{res} = 177$ bara
Formation temperature	Varying linearly with depth from 277.15 K to 353.15 K
Well layers	See (Li et al., 2015) for thickness, heat capacity, heat conductivity and density for the well outer layers

6.4.1 Case studies part I

Six realistic CO_2 injection scenarios are considered in this part of the study and the resulting transient pressure and temperature profiles along the well are simulated. The injection scenarios include linear ramping up to a maximum injections flow rate of 33.5 kg/s in 5 minutes (case 1), 30 minutes (case 2) and 2 hours (case 3) each at two starting injection temperatures of 278.15 K (case A) and 283.15 K (case B) for a perfectly vertical configuration. For reference purposes, each of the six scenarios will be named after the values of the two parameters, e.g. case 1-A refers to a linear ramp-up dense-phase CO_2 injection from 0 to 33.5 kg/s in 5 minutes at a feed temperature of 278.15 K.

To ensure numerical stability and convergence, a CFL condition of 0.3, a relaxation time of 10^{-4}s and 300 computational cells are employed for conducting the injection simulations. For the numerical approximation of the heat exchanges with the well layers, the discretisation parameters given by Li et al., (2015) are employed. For the heat transfer analysis, thickness, heat capacity, heat conductivity, and density for the eleven well layers are taken the same as those given in (Li et al., 2015).

The initial pressure and temperature profiles along the tapered well are given in Figures 6.2 and 6.3 respectively. The relatively low pressure at the bottom of the well (38 bara) means that the CO_2 remains in the gaseous phase in the first 400 m along the well, following which transition to the dense or liquid phase takes place (as indicated by the vertical dashed lines).

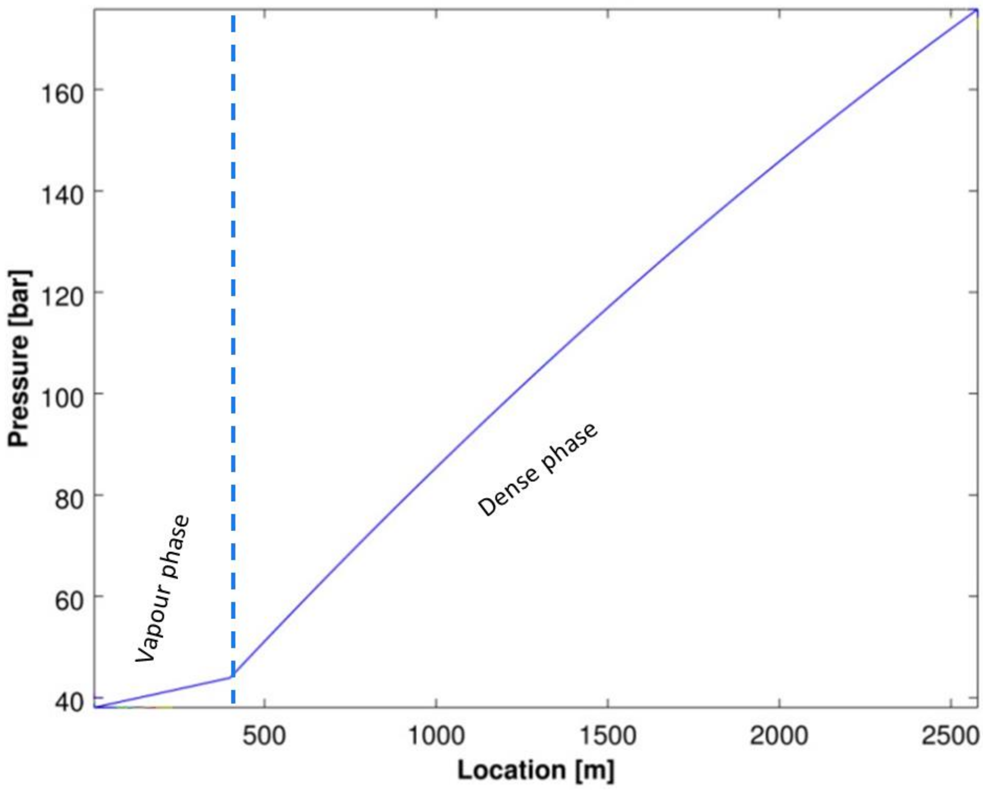


Figure 6.2: The initial pressure profile along the well

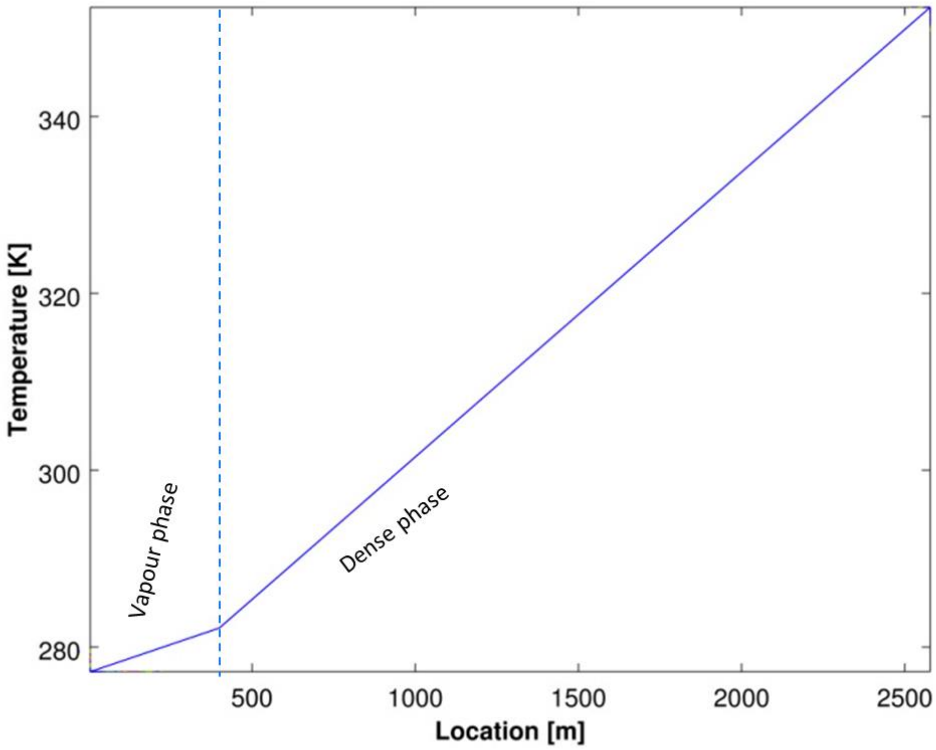


Figure 6.3: The initial temperature profile along the well

Figure 6.4 shows the variations pressure and temperature at the top of the well as a function of time during the linear ramping up injection of CO_2 from 0 to 33.5 kg/s over 5 minutes (case 1A). The feed CO_2 stream pressure and temperature are 115 bara and 278.15 K respectively. Figure 6.5 on the other hand shows the same transient profiles but for a higher feed temperature of 283.15 K (case 1B). For both of the above initial conditions, the injected CO_2 is in the dense phase and remains so for the entire time frame (500 s) under consideration. Also, in both cases an initial rapid depressurisation and cooling is followed by a recovery coinciding with the time (300 s) at which the injection rate (33.5 kg/s) reaches its peak value. Whilst the temperature remains relatively constant thereafter, a secondary modest (ca. 5 bara) drop in pressure is obtained.

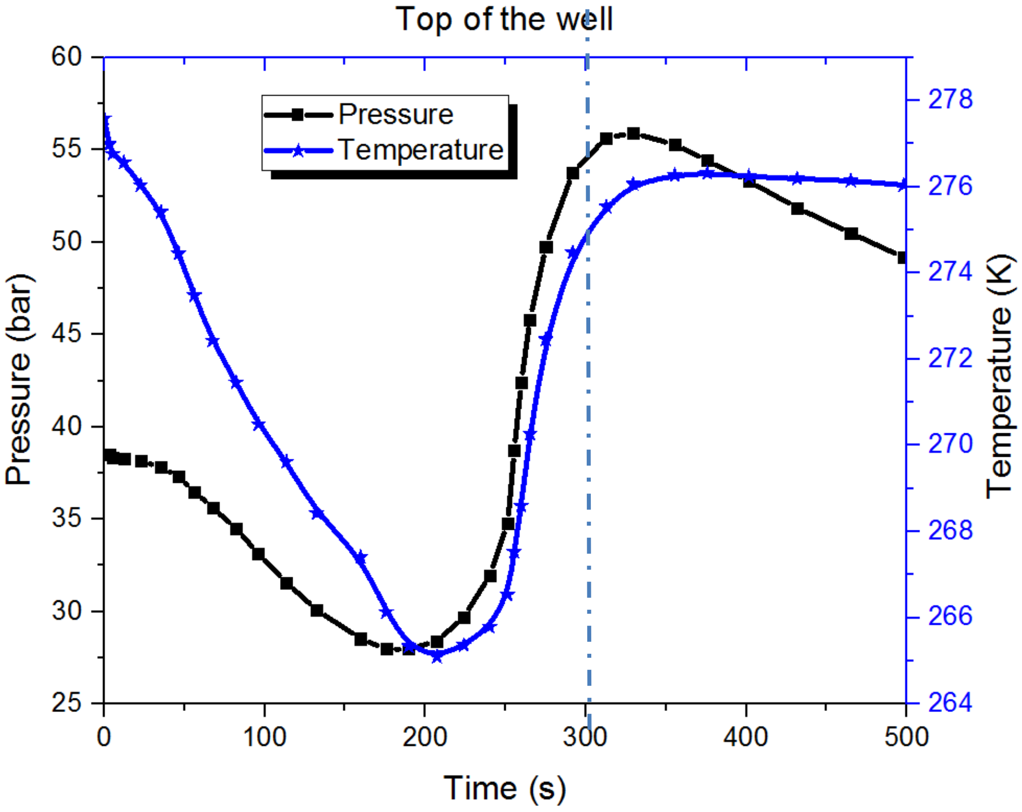


Figure 6.4: The transient pressure and temperature profiles at the top of the well for the fast injection ramping rate, case 1A (Table 6.1 feed temperature = 278.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 33.5 kg/s.

Additionally as expected, the lower the feed temperature, the lower is minimum temperature attained. In the case of the feed temperature of 278.15 K, the lowest temperature reached at the top of the well is 265 K (case 1A). This compares with the slightly higher minimum temperature of 267.5 K for the feed temperature of 283.15 K (case 1B). Given that both of these minimum temperatures are below the freezing point for water, the presence of an appreciable amount of water either in the CO_2 stream or the well would pose the risk of blockage due to ice formation at the wellhead some 180 s after the injection process has commenced.

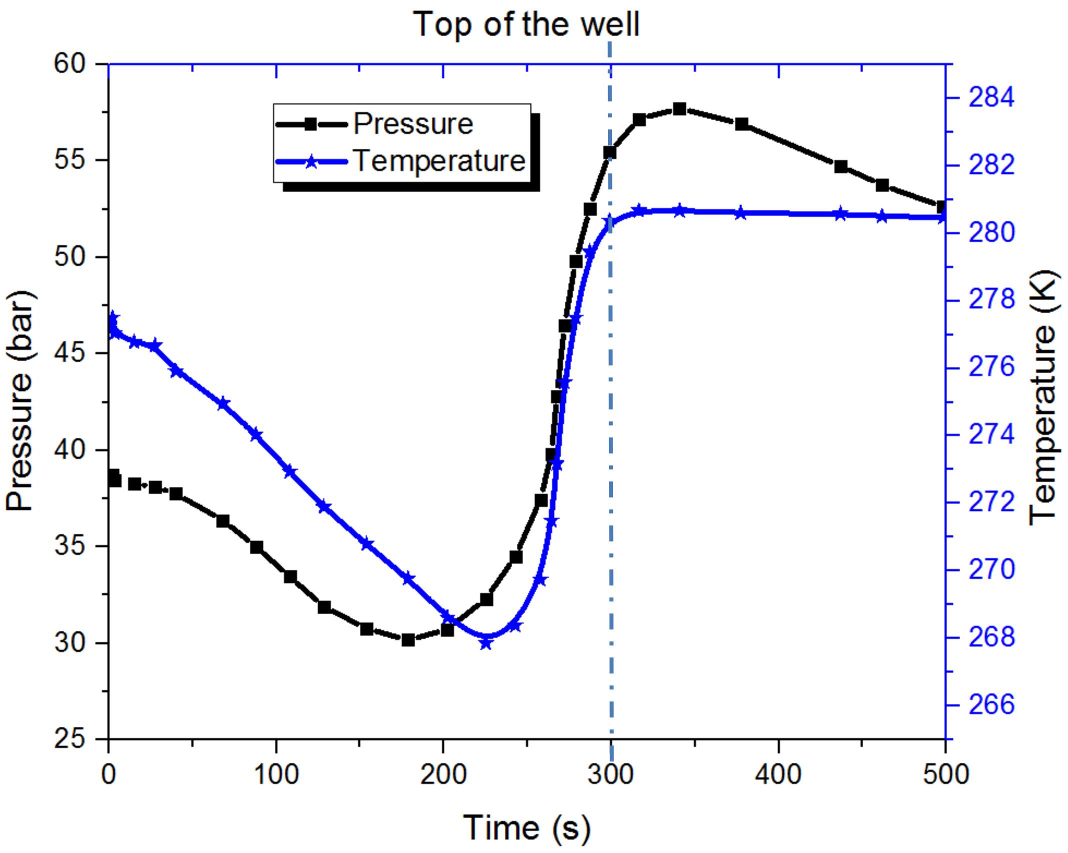


Figure 6.5: The transient pressure and temperature profiles at the wellhead for the fast injection ramping rate, case 1B (Table 6.1: feed temperature = 283.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 33.5 kg/s.

Figure 6.6 to 6.9 present the wellhead pressure and pressure transient profiles for the higher injection ramping up durations of 30 minutes and 2 hours each for the starting temperatures of 278.15 K and 283.15 K. With the expectation of the different magnitude of the changes and their respective durations, the observed trends are very similar to those for the shorter ramping up period of 5 minutes described above.

In all cases, the termination of the initial rapid drops in temperature closely follows the times at which the peak injection rates are reached.

More specifically, the rates of drop in pressure and temperature and their subsequent recovery decrease with the increase in the injection ramping up duration. In addition, and most importantly, the longer the injection ramping up period, the lower is the minimum CO_2 temperature reached at the wellhead maximising the risk of wellhead blockage due to ice formation. The minimum temperature of 252 K (some 21 °C below the freezing point for water) is observed for case 3-A (Figure 6.8) corresponding to an injection ramping up period of 3800 s following the feed injection temperature of 278.15 K.

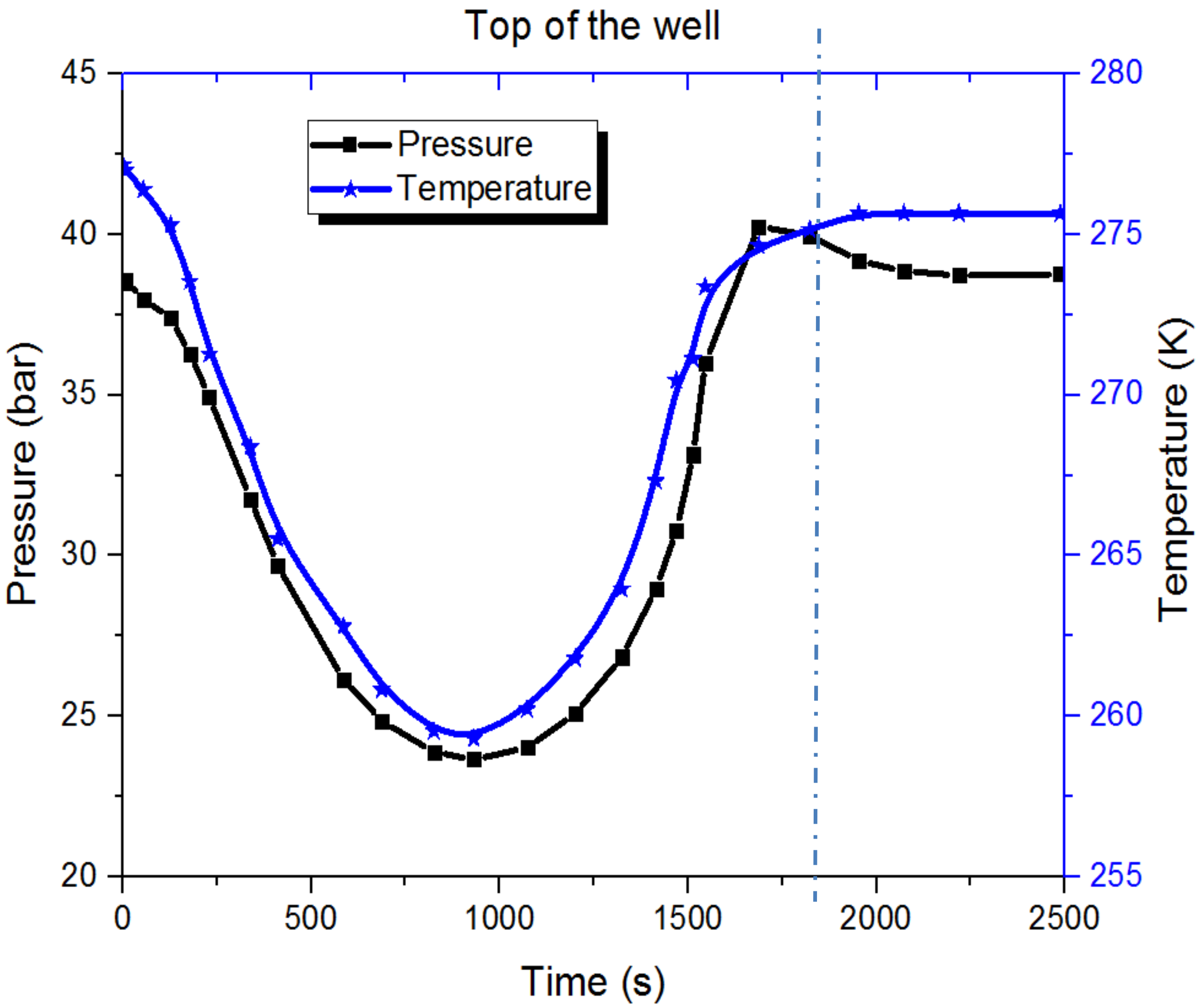


Figure 6.6: The transient pressure and temperature profiles at the top of the well for the medium injection ramping rate, case 2-A (see Table 6.1: feed temperature = 278.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (1800 s) at which the injection flow rate reaches its peak value of 33.5 kg/s.

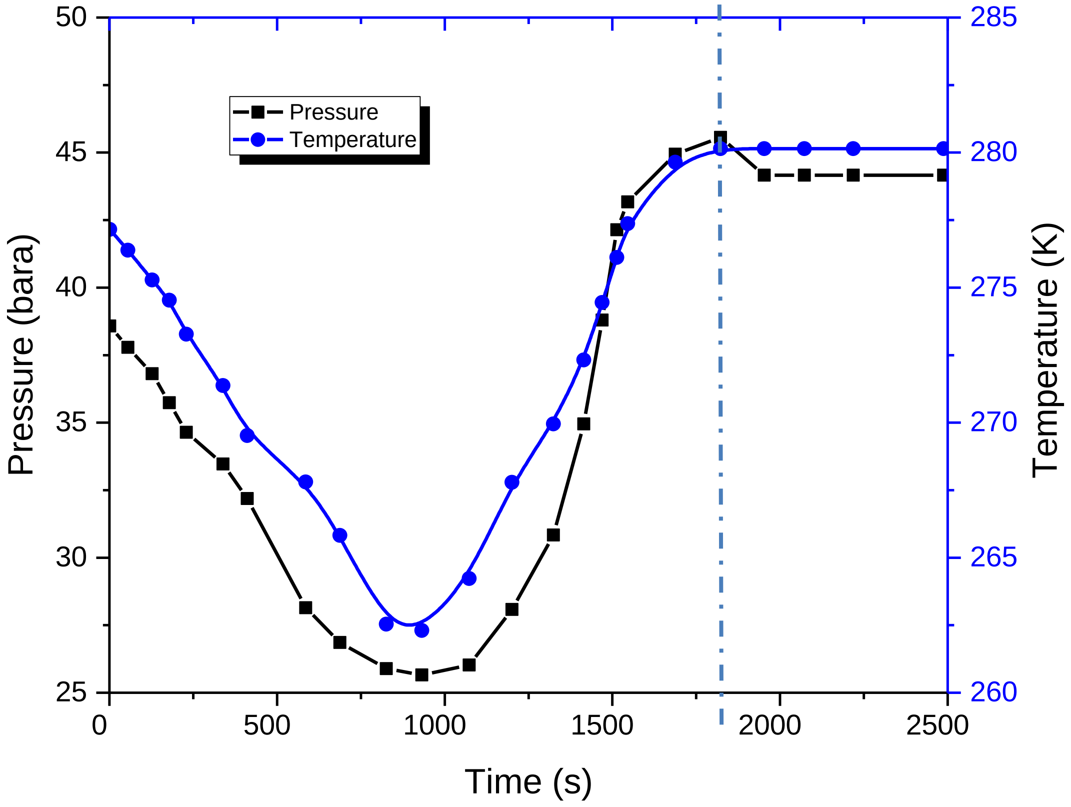


Figure 6.7: The transient pressure and temperature profiles at the top of the well for the medium injection ramping rate, case 2-B (Table 6.1: feed temperature = 283.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (1800 s) at which the injection flow rate reaches its peak value of 33.5 kg/s.

Figure 6.8 and Figure 6.9 present the results for the slow injection ramping 3-A and 3-B. The drops in both pressure (minimum at 20 and 22 bara, respectively) and temperature (minimum at 252 K and 256 K, respectively) are more severe. It is noted that at 7200s, when the linear ramp-up injection reaches its maximum value, for case 3-A neither the pressure nor the temperature immediately reaches a steady state. Simulations were terminated at 9000s, given their computational demanding nature.

As it can clearly be observed by comparing cases 1-A, 2-A, 3-A with 1-B, 2-B, 3-B, the longer the ramp-up operations for the mass flow rate, the more severe the degree of cooling and depressurisation along the well. Moreover, both pressure and temperature attain slightly higher minimum values when a higher feed temperature is chosen, for all pairs 1-A/1-B, 2-

A/2-B and 3-A/3-B. Based on the above data, in order to avoid a prolonged period of time during which pressure and temperature could remain low at the top of the well, thus leading to the risk of ice formation and thermal shocks, it is therefore necessary to ramp-up the injection rate to the maximum desired value over a shorter period of time.

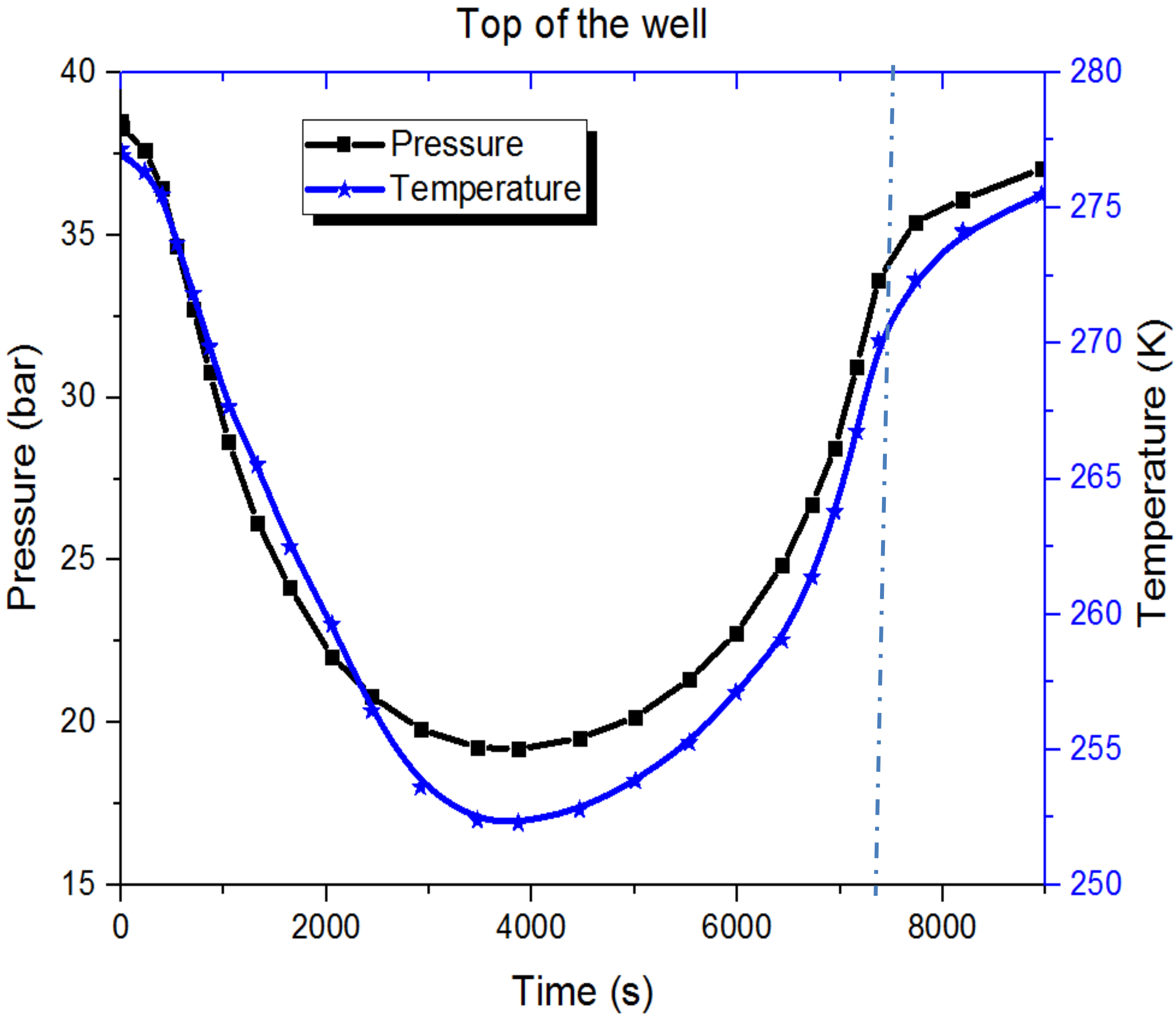


Figure 6.8: The transient pressure and temperature profiles at the top of the well for the slow injection ramping rate, case 3-A, Table 6.1 (feed temperature = 278.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (7200 s) at which the injection flow rate reaches its peak value of 33.5 kg/s.

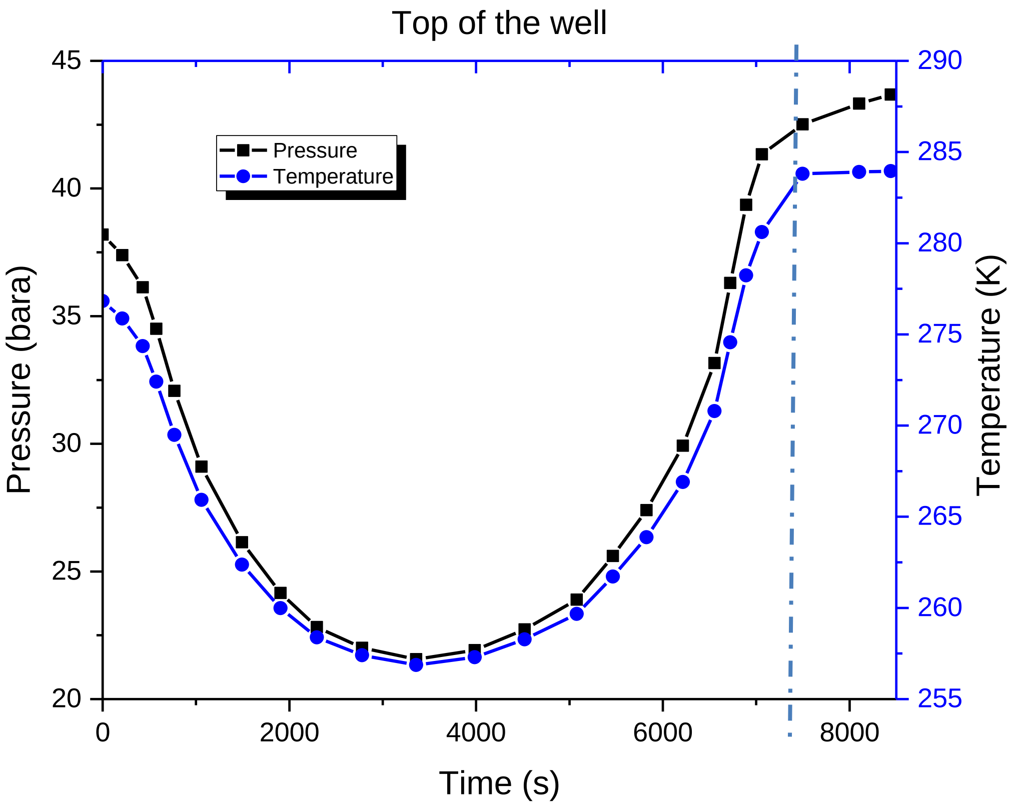


Figure 6.9: The transient pressure and temperature profiles at the top of the well for the slow injection ramping rate, case 3-B, Table 6.1 (feed temperature = 283.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (7200 s) at which the injection flow rate reaches its peak value of 33.5 kg/s.

Figures 6.10 to 6.12 respectively present the variations of pressure and temperature at the bottom of the well as a function of time for the slow (case 1-A), medium (case 2-A) and slow (case 3-A) injection ramping strategies. The corresponding plots for the higher inlet temperature of 383.15 K (cases 1-B, 2-B and 3-B) are not presented given that they are very similar to those in Figures 6.10 to 6.12. It is observed that in all the three cases the pressure increases from an initial value of 177 bara to a final value of ca. 195 bara. It is worth mentioning that for the slow-injection cases 2-A and 3-A, the pressure build-up reaches its steady state as the injection threshold is reached (1800s and 7200s, respectively), while for the faster injection case 1-A, a transient behaviour is still observed at the end of the simulation.

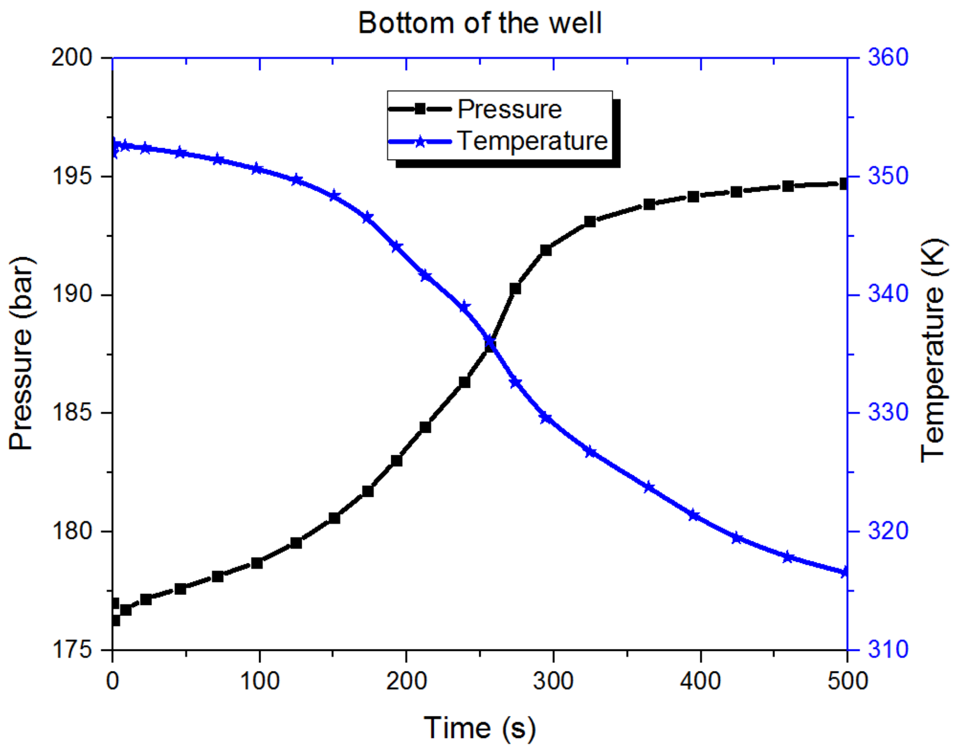


Figure 6.10: Transient pressure and temperature profiles at the bottom of the well for case 1-A (see Table 6.1).

Regarding the temperature, we observe that for all the cases considered the temperature drops at the bottom of the well. Moreover, the most severe decrease is observed for the slowest-injection case 3-A (ca. 65 K less than the initial value). This behaviour is in line with the results of the simulations performed in Li et al., (2015) and was already noticed in Linga & Lund, (2016). In this work, a perfectly insulated bottom of the well is assumed in order to derive a rigorous, but easy to implement Finite Volume method for radial heat conduction, which is employed in this study. The severity of the temperature drop observed in the simulations is most certainly due to this assumption, and will require further investigation in the future, especially a direct comparison with field data, should they become available.

The data above are also helpful in assessing the risk of CO_2 hydrates forming around the wellbore, when the CO_2 is in contact with the interstitial water. According the CO_2 hydrate phase diagram reported in (Mahgerefteh et al., 2006), CO_2 hydrate formation is unlikely at the given pressure and temperature ranges.

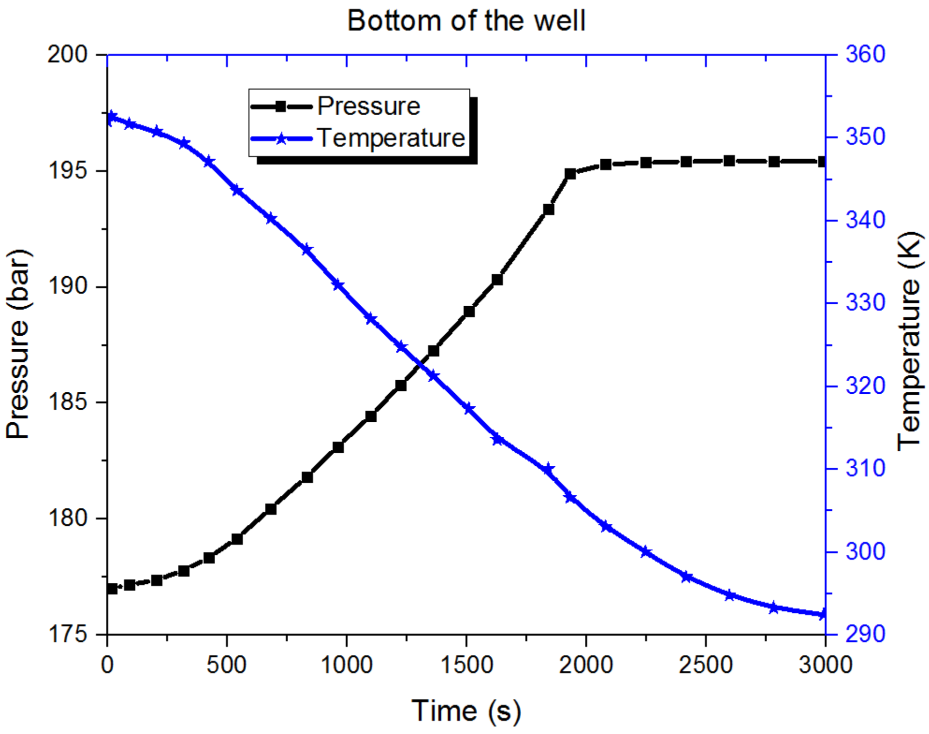


Figure 6.11: Transient pressure and temperature profiles at the bottom of the well for case 2-A (see Table 6.1)

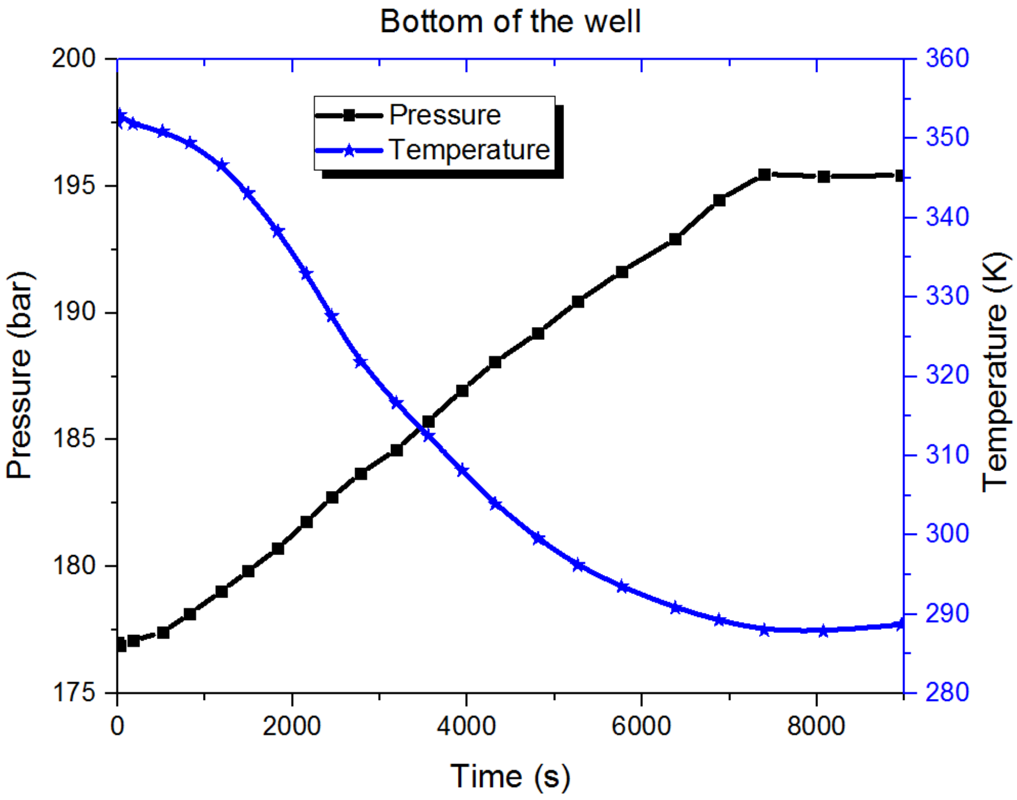


Figure 6.12: Transient pressure and temperature profiles at the bottom of the well for case 3-A (see Table 6.1)

It is worth noting that pressure, temperature, and the mass flow rate are not the only factors to which this time-dependent model is very sensitive. In particular, it is noticed that the application of accurate inflow and outflow boundary conditions is of paramount importance. More specifically, at the beginning a generic subsonic outflow boundary conditions is implemented according to Thompson (1987, 1990), but the results of the simulations were not physically acceptable. A CO_2 backflow at the bottom of the well at the beginning of the simulation, coupled with a significant pressure drop, indicated that a more accurate boundary condition had to be specified. As indicated in Thompson (1987, 1990), the generic subsonic outflow boundary condition is meant to be used where no additional information is available about the reservoir or the chamber into which the fluid is to be discharged. For injection problems, as the one analysed in this project, it is recommended to prescribe a site-specific outflow condition, which can take into account the properties (pressure, porosity, permeability, etc.) of the particular reservoir into which the CO_2 is discharged. Regarding the inflow modelling, it is noted that the boundary condition accounting for the isenthalpic process undergone by the pressurised CO_2 at the well injector is robust and results in degrees of depressurisation and cooling in very good agreement with the simulations conducted in Li et al., (2015) and Shell UK, (2015).

6.4.2 Case studies part II

It is now worth investigating, for each of the three mass flow rate choices considered above, the impact of the well configuration (as can be seen in Figure 6.13) on the transient behaviour of the CO_2 flowing in the injection well. Firstly, considering the 5-minute injection strategy and modify the reference case by varying the cross-sectional area of the well. As indicated in Table 6.1, the internal diameter exhibits a discontinuity at 800m down the well, from an upper value of 0.125m to possible lower values of 0.076m, 0.07m, 0.065m and 0.0625m.

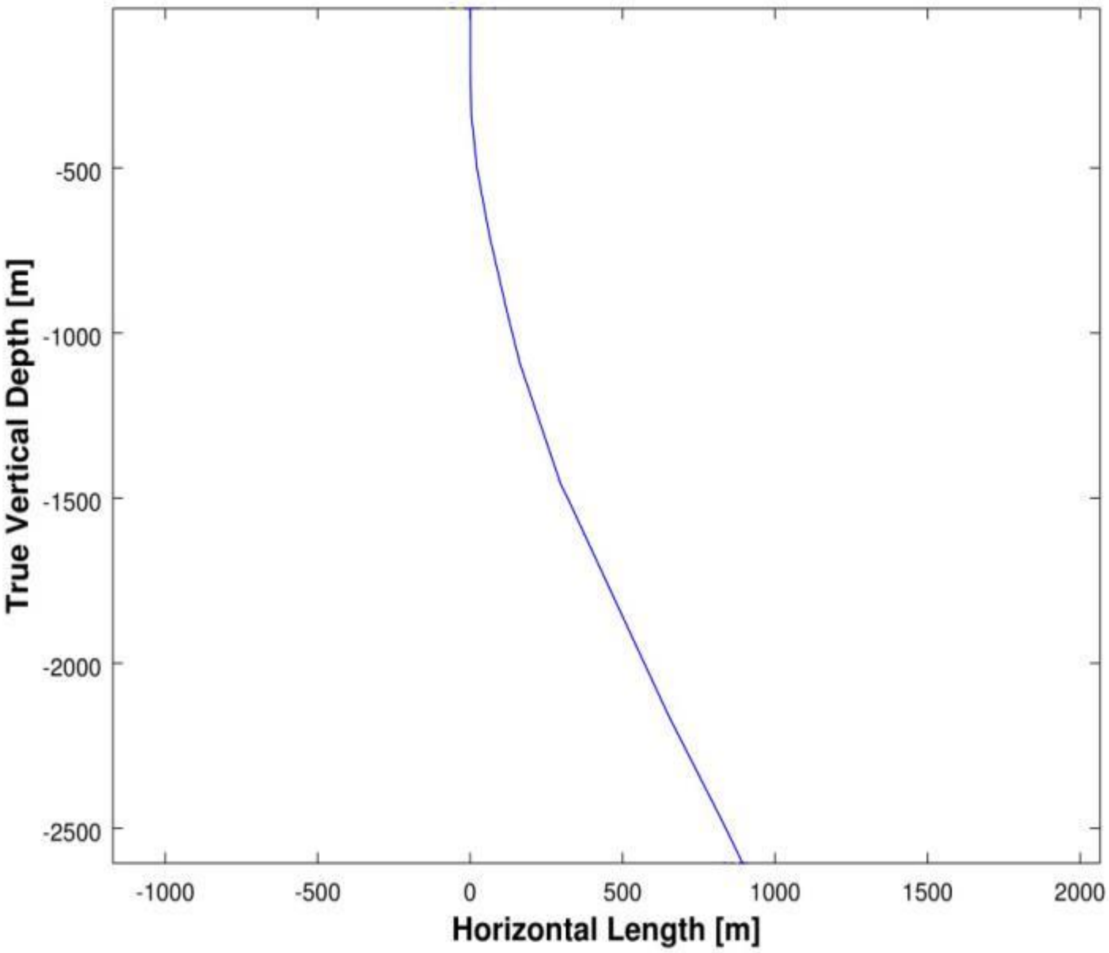


Figure 6.13: Well deviation from a perfectly vertical configuration

Figure 6.14 shows the variation of the temperature at the top of the well as a function of time and internal diameter restriction. The less pronounced the tapering at 800m down the well, the lower the temperature at any given time. In particular, the lowest value is attained for the case with i.d. 0.076m (265.11 K), while the highest recovery temperature is achieved for the reference case, with a steady-state value of ca. 278.54 K.

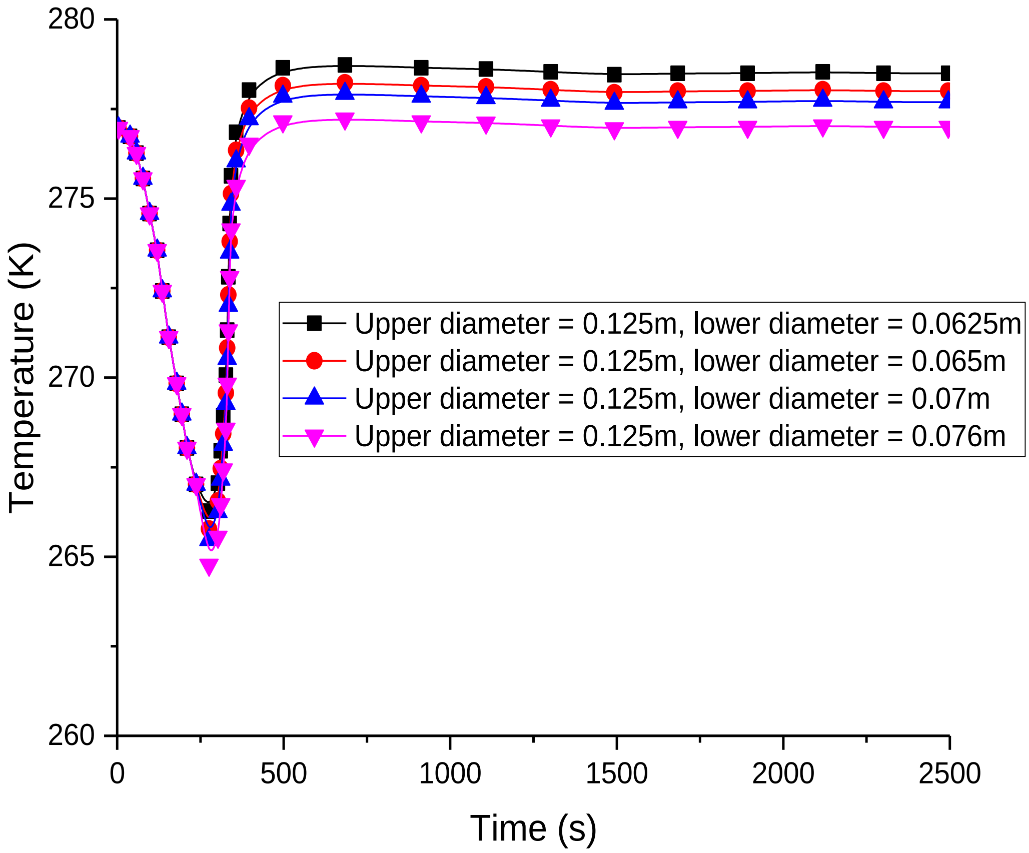


Figure 6.14: Transient temperature profile at the top of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

In Figure 6.15 the temporal variation of the pressure at the top of the well is analysed. For this pressure transient, the impact of the variation in the cross-sectional area is even more visible. For the reference case, the pressure attains its minimum at ca. 29 bara and then exhibits a very consistent build-up, with a maximum of ca. 147 bara and a steady-state value of ca. 132 bara. Conversely, for the least pronounced restriction at 800m, the minimum is attained at a slightly lower value (ca. 28 bara), followed by a recovery up to 56 bara and then another drop to a steady-state value of ca. 40 bara. In Figures 6.16 and 6.17 the variation of density and velocity at the top of the well, as a function of time and internal diameter is plotted.

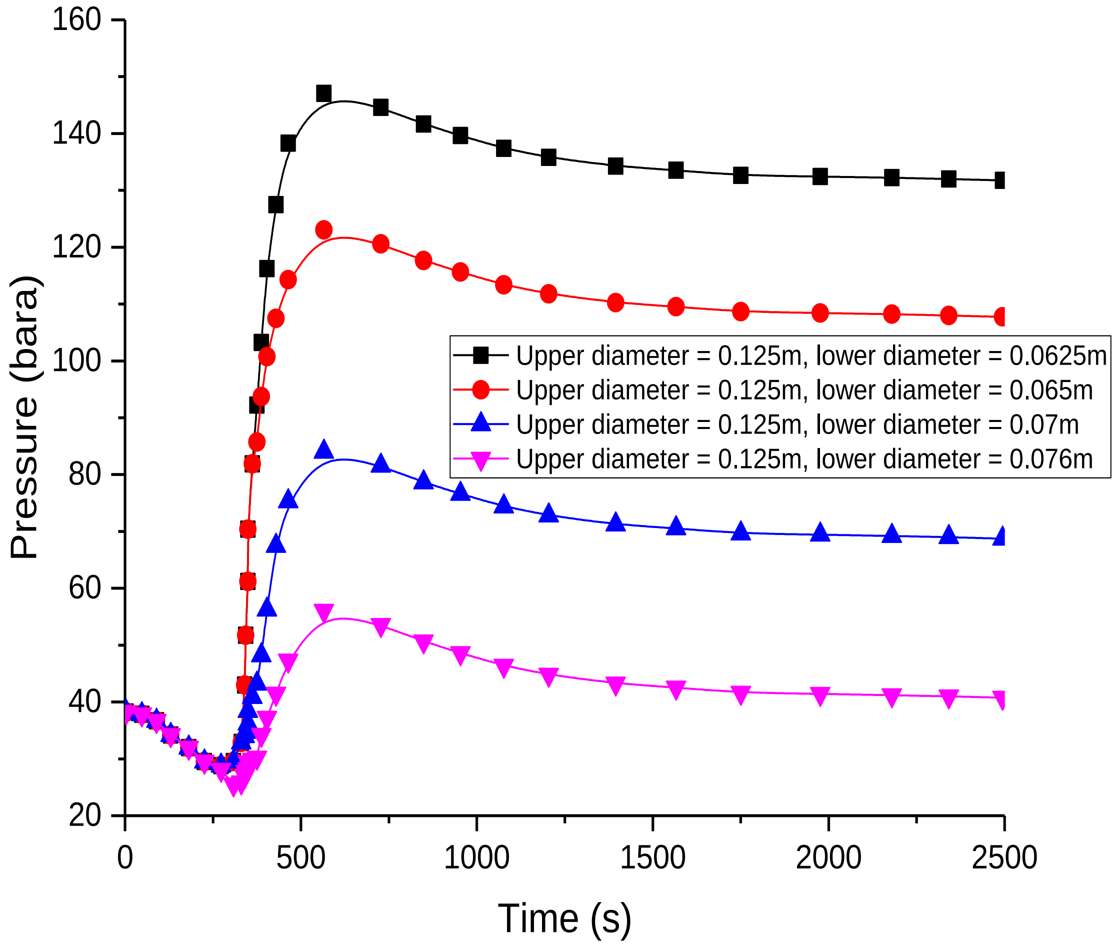


Figure 6.15: Transient pressure profile at the top of the well for the 5-minute injection case, with well deviation as in Table 6.1and upstream temperature equal to 278.15 K

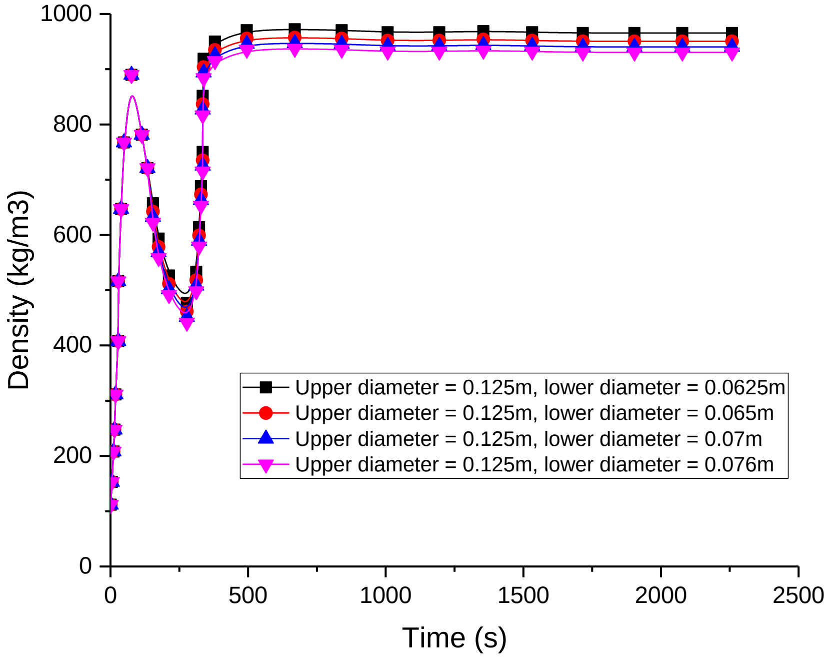


Figure 6.16: Transient density profile at the top of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

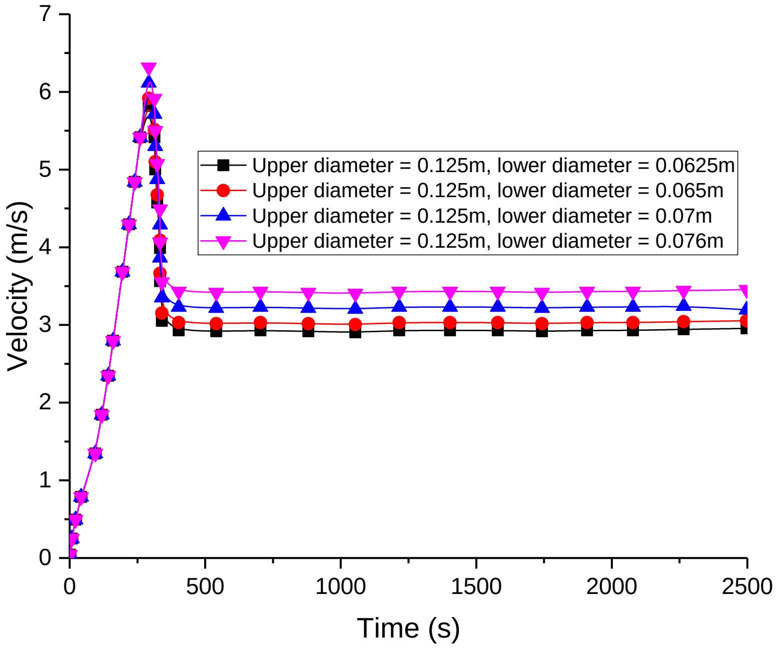


Figure 6.17: Transient velocity profile at the top of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

Given that the inlet mass flow rate is prescribed, a higher density for the 0.0625m lower diameter case (corresponding to the temperature and the pressure discussed above) is associated with a lower velocity. In continuation, what happens at the bottom of the well will now be investigated.

As can be seen from Figure 6.18 and 6.20, the steady-state pressures and temperatures are practically the same for all the diameter restrictions. This is most certainly due to the presence of the reservoir with its specific permeability and porosity. Remembering that a pressure-mass flow rate outflow boundary condition was imposed, it naturally follows that the outflow mass fluxes will be almost indistinguishable. However, in Figure 6.19 the highest outflow velocity and the lowest density are associated with the most pronounced lower restriction (lower diameter 0.0625m).

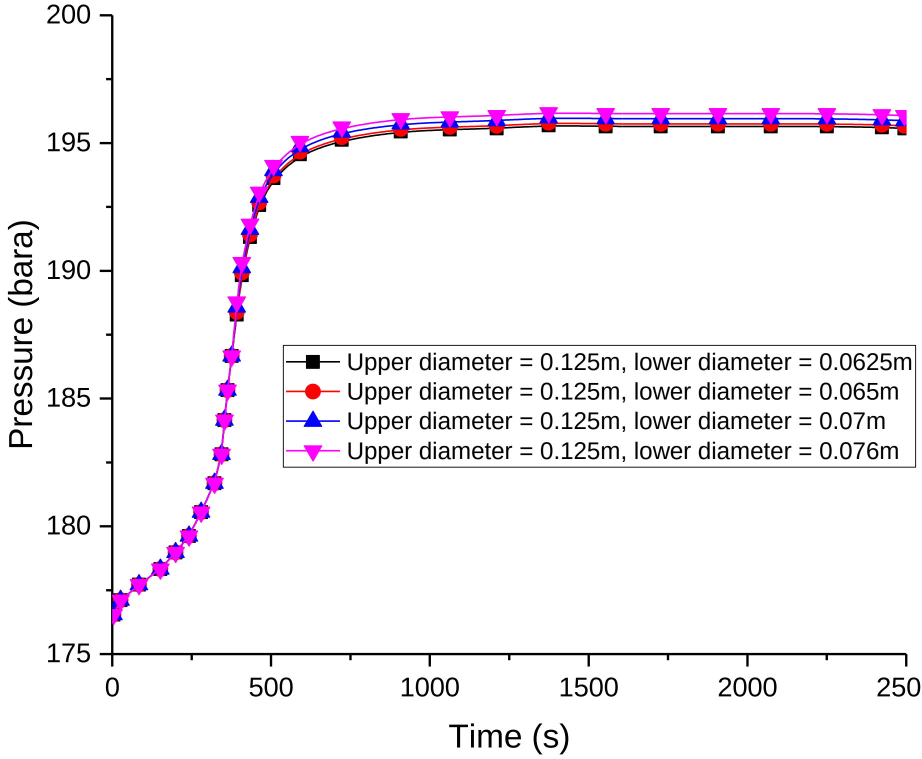


Figure 6.18: Transient pressure profile at the bottom of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

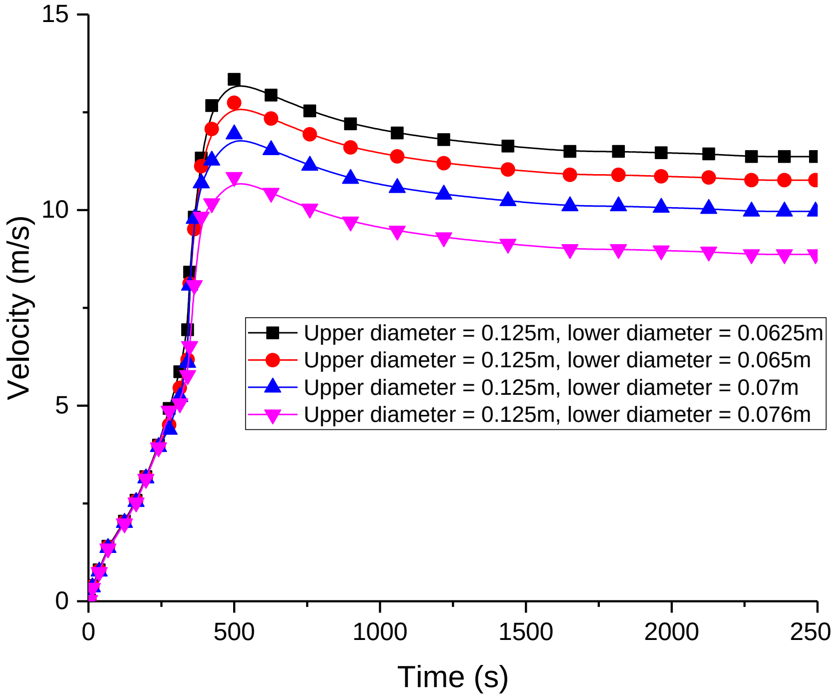


Figure 6.19: Transient velocity profile at the bottom of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

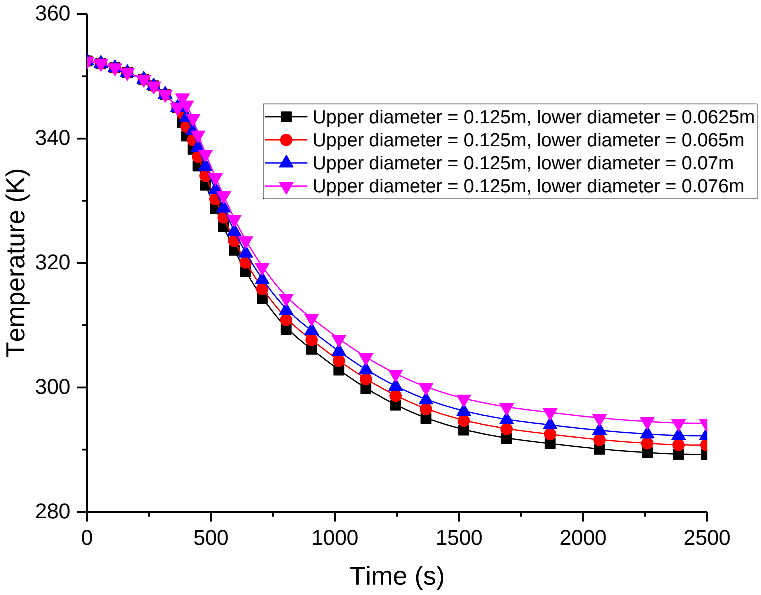


Figure 6.20: Transient temperature profile at the bottom of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

In continuation of the sensitivity analysis by keeping all the parameters as in the previous test the same with the exception of the well configuration, which is now set to be perfectly vertical. First, from Figure 6.21 it can be seen that the temperature drop at the top of the well is more severe (with a 262.61 K minimum reached with the least pronounced internal diameter restriction, 0.076m). In addition, in such case the temperature recovers up to 275.42 K but then starts decreasing again. This behaviour is associated with two-phase flow, as it can be seen from the corresponding density, plotted in Figure 6.22. For further reductions of the cross-sectional area, no two-phase flow is encountered at the top of the well.

In order to investigate the impact of the well configuration, the reference case is modified with the second mass flow rate strategy (ramp-up injection from 0 to 33.5 kg/s in 30 minutes).

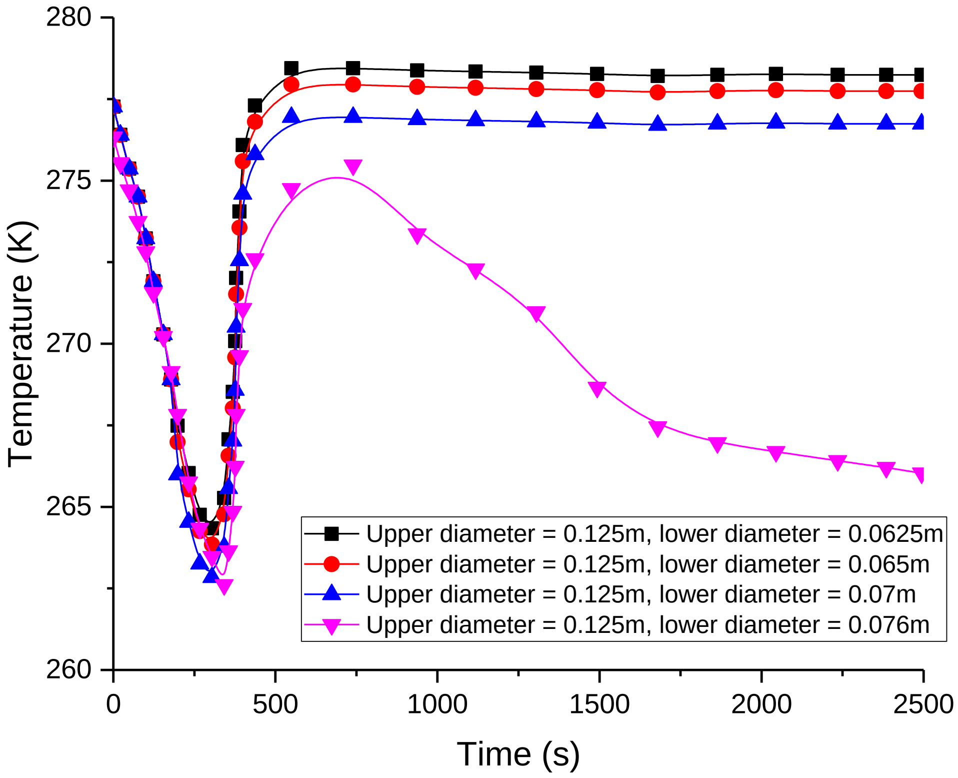


Figure 6.21: Transient temperature profile at the top of the well for the 5-minute injection case, with perfectly vertical well configuration and upstream temperature equal to 278.15 K

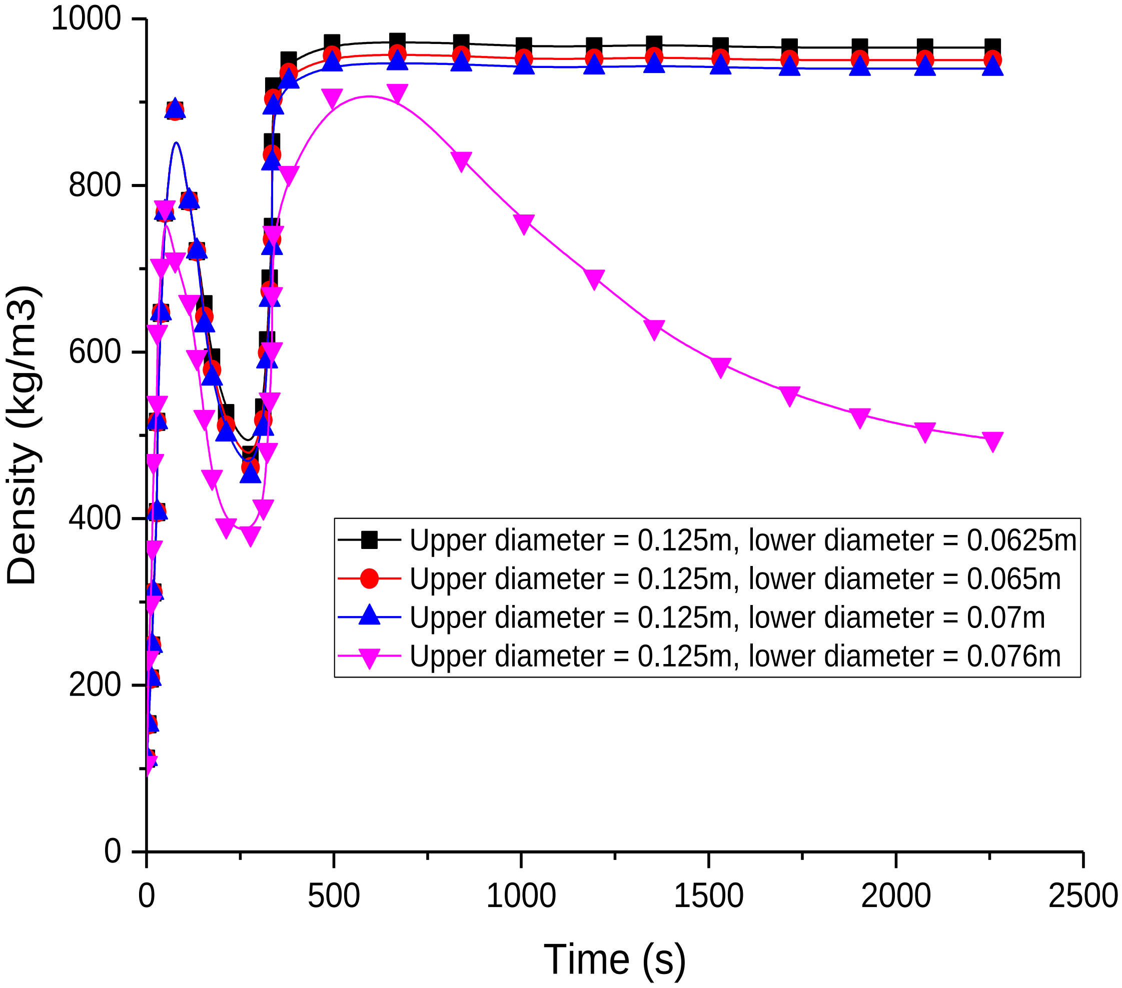


Figure 6.22: Transient density profile at the top of the well for the 5-minute injection case, with perfectly vertical well configuration and upstream temperature equal to 278.15 K

Figure 6.23 shows the variation of the temperature at the top of the well as a function of time and well internal diameter restriction. Similar to the 5-minute ramp-up strategy considered above, the less pronounced restrictions in the cross-sectional area at 800m down the well correspond to lower temperature drop, at any given time. In particular, the temperature attains a minimum of ca. 259 K (6 K lower than the minimum for the 5-minute ramp-up strategy), while the highest recovery temperature is achieved for the reference case, with a steady-state value of ca. 278.5 K.

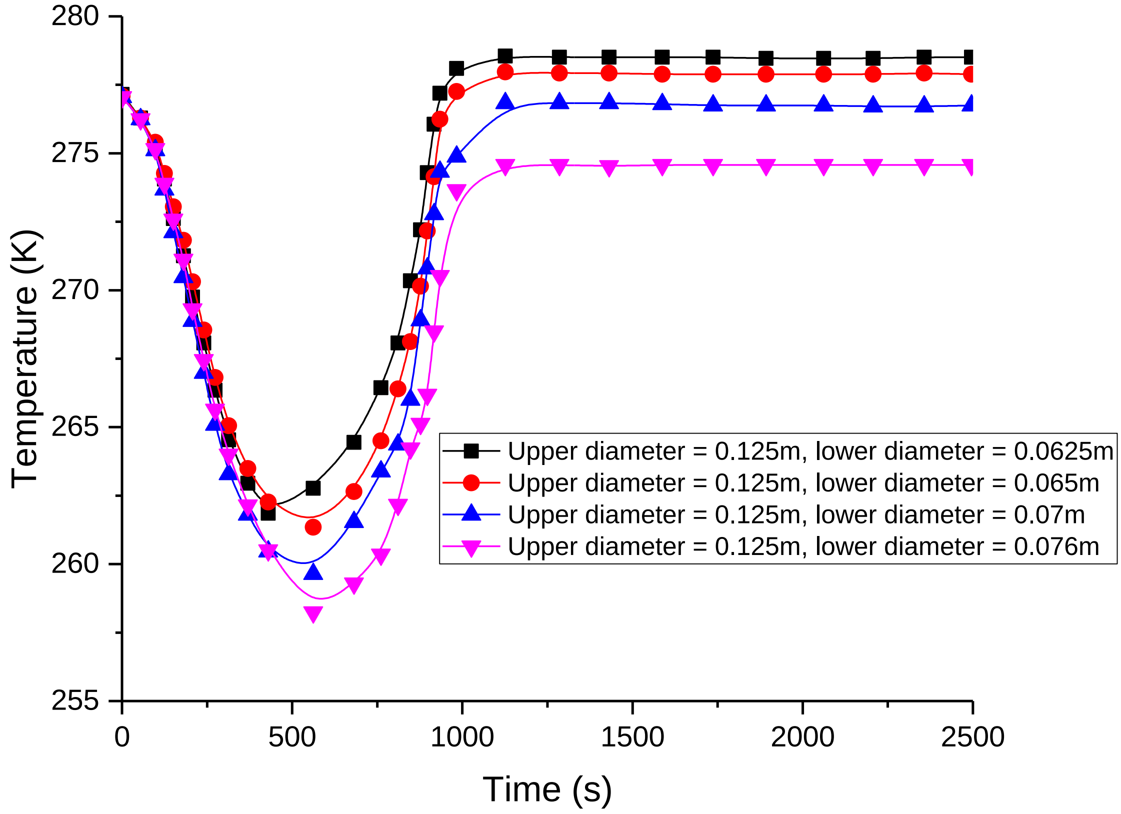


Figure 6.23: Transient temperature profile at the top of the well for the 30-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

In Figure 6.24**Error! Reference source not found.** the temporal variation of the pressure at t he top of the well is analysed. For the reference case, the pressure attains a minimum of 25.5 bara, followed by a recovery up to ca. 135 bara, with a final, steady-state value of ca. 131 bara. Conversely, the least pronounced restriction (i.e. the highest internal diameter 0.076 m imposed) at 800m downward, the minimum pressure is attained at a lower value (23.7 bara) followed by a recovery up to 41.5 bara and then another drop to a steady-state value of 38.5 bara. Considerations regarding density and velocity at the top and the bottom of the well can be made in a similar fashion as in the previous analysis for the 5-minute ramp-up strategy.

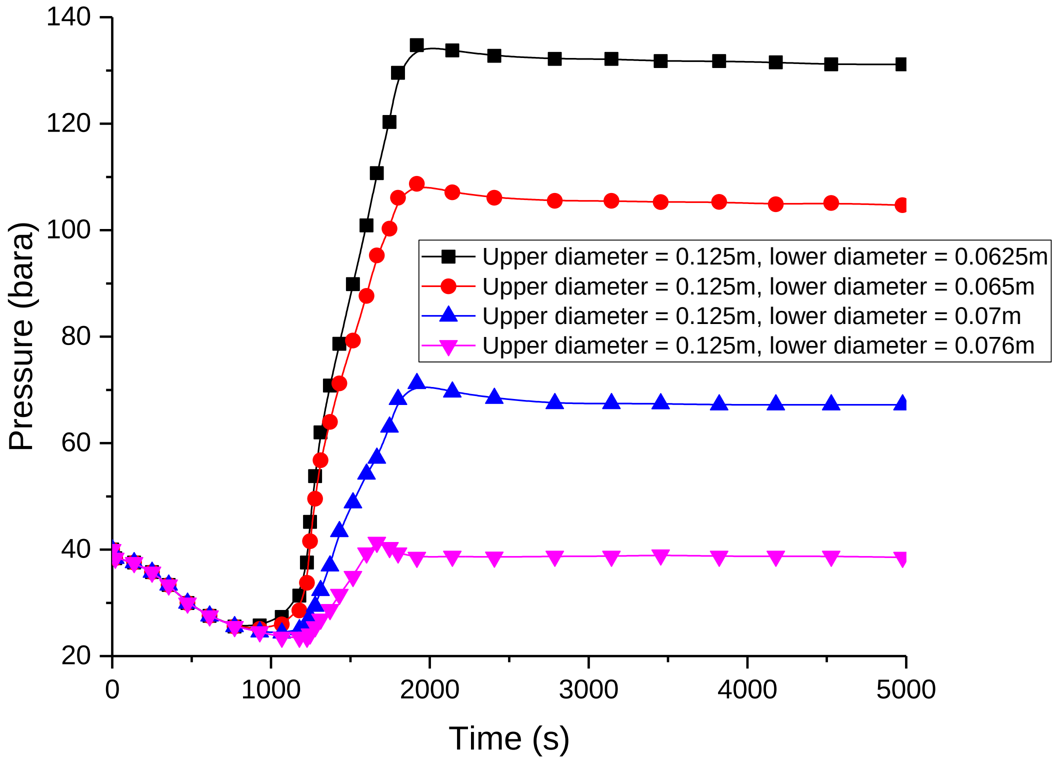


Figure 6.24: Transient pressure profile at the top of the well for the 30-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 278.15 K

Considering a perfectly vertical injection wellbore configuration and keeping all the other parameters the same as in the previous case, the predictions in Figure 6.25 show that the minimum temperature of 256.15 K at the top of the well is more severe for the least pronounced diameter restriction case. As can be seen in Figure 6.25, in such this case the temperature recovers up to ca. 265 K but then slowly decreases again. This behaviour is associated with two-phase flow, as it can be seen from the corresponding density, plotted in Figure 6.26. For the more pronounced area restrictions (i.e. for small internal diameter 0.0625 m case), no two-phase flow is encountered at the top of the well.

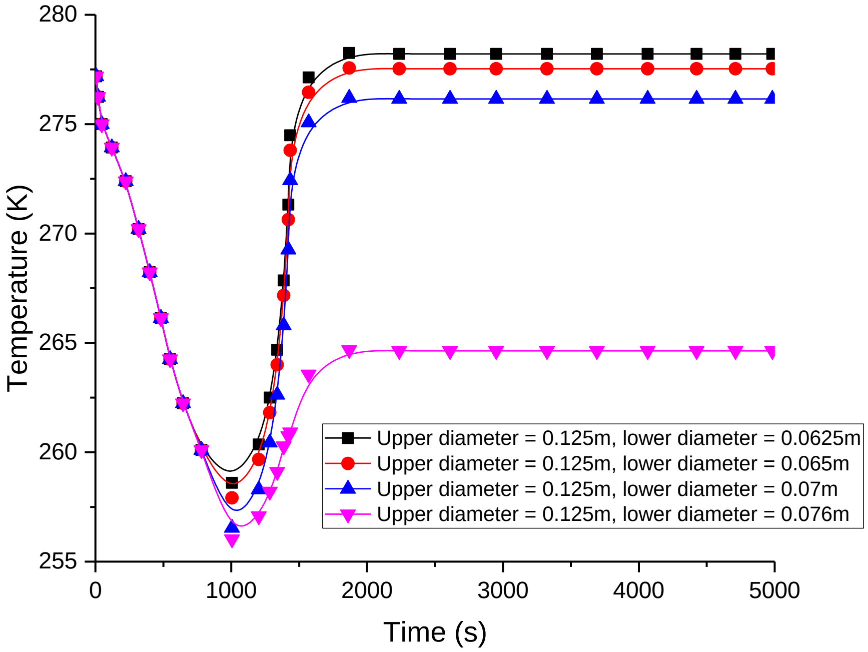


Figure 6.25: Transient temperature profile at the top of the well for the 30-minute injection case, with perfectly vertical well configuration and upstream temperature equal to 278.15 K

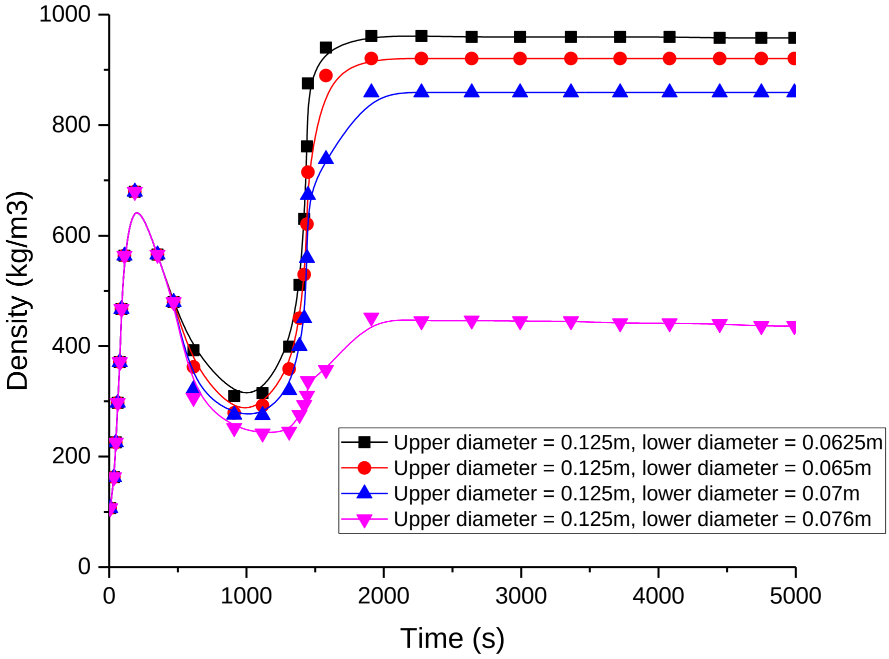


Figure 6.26: Transient density profile at the top of the well for the 30-minute injection case, with perfectly vertical well configuration and upstream temperature equal to 278.15 K.

The impact of the CO_2 feed stream temperature on the CO_2 flow behaviour is discussed next. Given that the simulations are computationally demanding, the analysis is limited to the 5-minute injection only. Considering again the reference case and modifying only the upstream temperature from 278.15 K to 283.15 K, Figure 6.27 shows the variation of the CO_2 temperature at the top of the well as a function of time. It is noted that the temperature drop is less severe when the internal diameter restriction at 800m downward is more pronounced (i.e. 0.0625 m). The minimum temperature attained is ca. 267 K, about 2 K higher than in the reference case plotted in Figure 6.14. Also noted is that the temperature reaches higher steady-state values, within a range from 280.14 K (least pronounced area restriction) to 283.15 K (reference case).

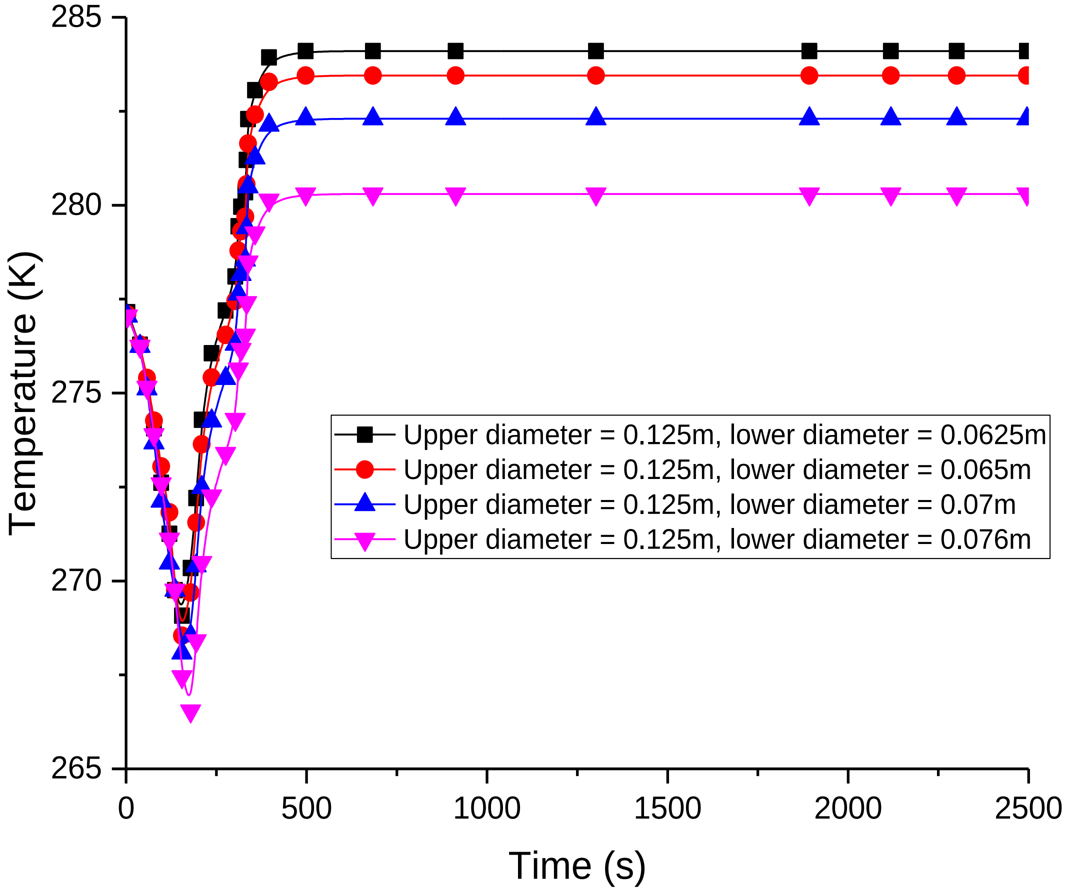


Figure 6.27: Transient temperature profile at the top of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 283.15 K

In Figure 6.28 the temporal variation of the pressure at the top of the well is predicted. Once again it is noted that the variation in the cross-sectional area impacts the results. The pressure drop is less severe (minimum of 30.3 bara for the least pronounced restriction), while the build-up in pressure reaches maxima higher than those in the 278.15 K feed stream case, plotted in Figure 6.15. In particular, a maximum of 150.8 bara is observed for the reference case, which eventually drops to a steady-state value of 139.1 bara.

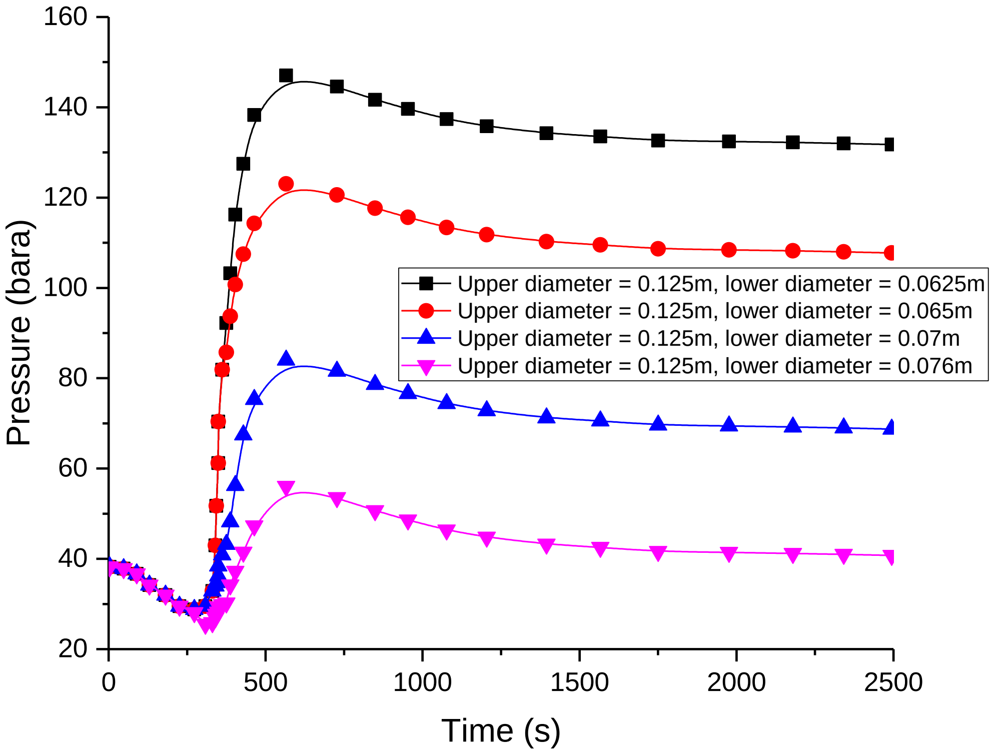


Figure 6.28: Transient pressure profile at the top of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 283.15 K

As can be observed in Figures 6.29 and 6.30, pressure and velocity trends are very similar to the 278.15 K feed stream case (Figures 6.18 and 6.19). In Figure 6.31 the temperature profiles for the case of well deviation as in Table 6.1 and upstream temperature equal to 283.15 K is plotted. The temperature values at the end of the simulations (2500s) are slightly higher than those obtained for the lower feed temperature case plotted in Figure 6.20. For instance, for the most pronounced internal diameter restriction (i.e. the smallest internal diameter 0.0625 m case), the final temperature is 292.77 K, compared a lower value of 289.5 K shown in Figure 6.31 and 3.20 respectively.

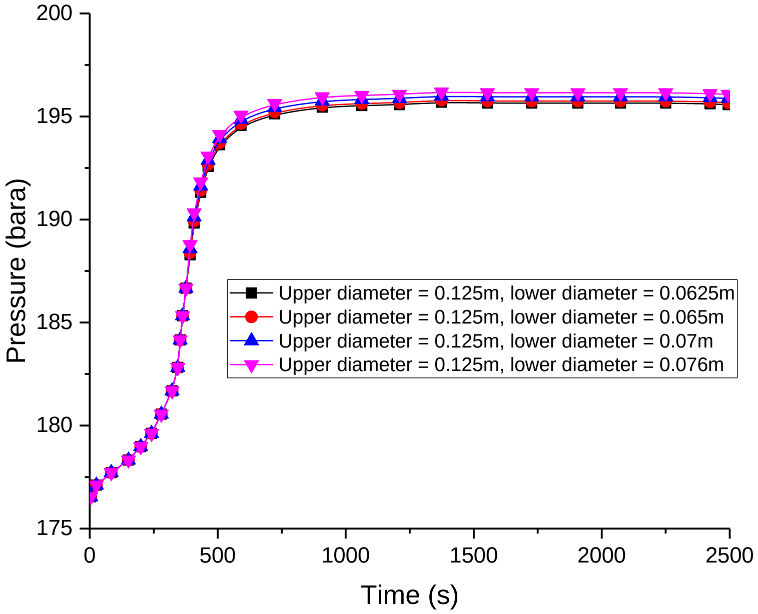


Figure 6.29: Transient pressure profile at the bottom of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 283.15 K

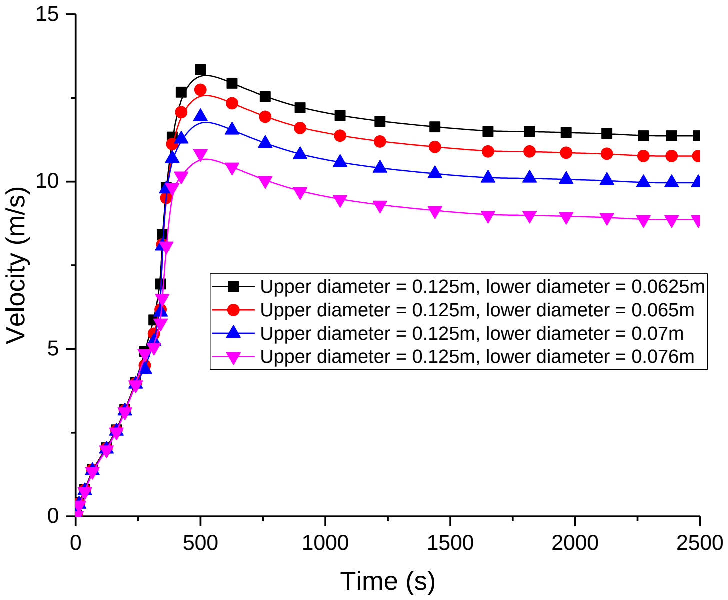


Figure 6.30: Transient velocity profile at the bottom of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 283.15 K

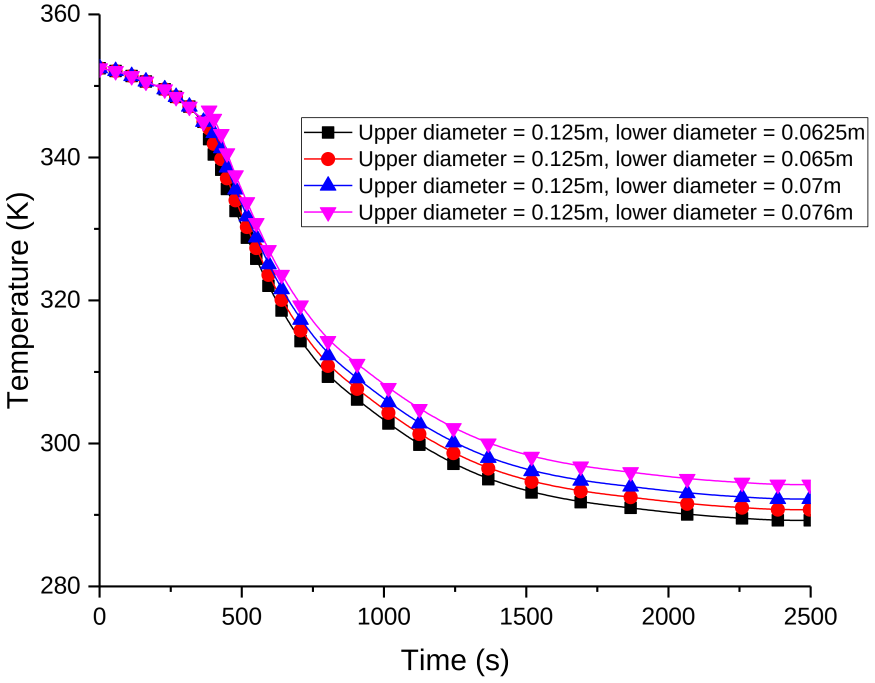


Figure 6.31: Transient temperature profile at the bottom of the well for the 5-minute injection case, with well deviation as in Table 6.1 and upstream temperature equal to 283.15 K

6.5 Comparison of HEM and HERM models performance

A comparison between the HEM and HERM simulation results for the transient pressure and temperature profiles at the top of the well for the fast injection ramping up case 1A (Table 6.1: feed temperature = 278.15 K, feed pressure = 115 bara) chosen as an example is presented next. The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 33.5 kg/s. As can be seen Figures 6.32 and 6.33, the two models show very similar trends for the pressure and temperature profile at the top of the well. However, the HERM predicts a much lower CO_2 pressure and temperatures compared to the HEM. In the case of the HEM, the minimum predicted temperature and pressure are 269 K (– 4 ^{o}C) and 36.5 bara respectively. The corresponding values in the case of the HERM are 265 K (– 8 ^{o}C) and 27.5 bara respectively. These findings demonstrate the significant impact of the thermodynamic non-equilibrium relaxation time accounted for in the HERM model.

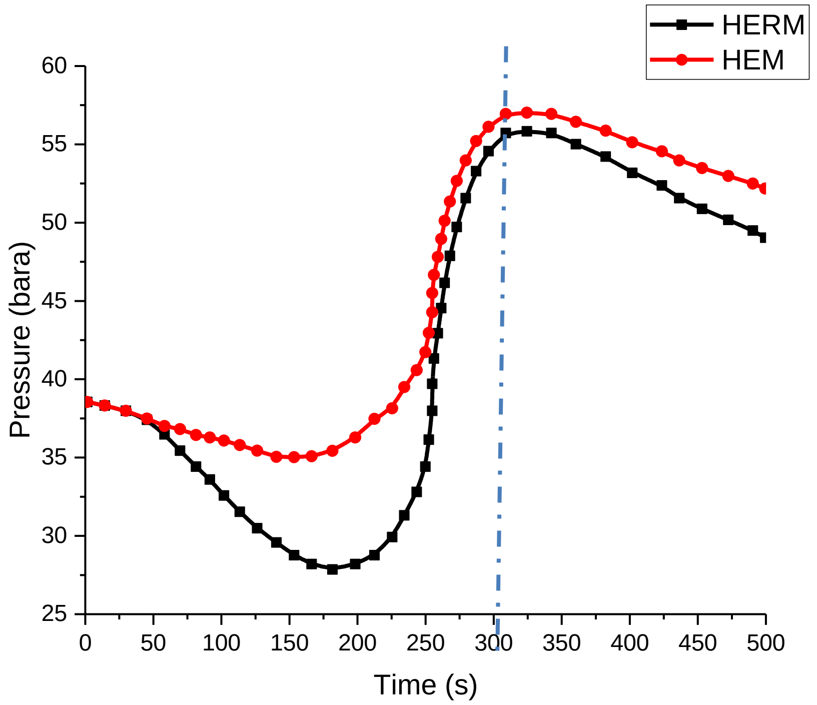


Figure 6.32: HEM and HERM transient pressure profiles at the top of the well for the fast injection ramping rate, case 1A (Table 6.1 feed temp = 278.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 33.5 kg/s

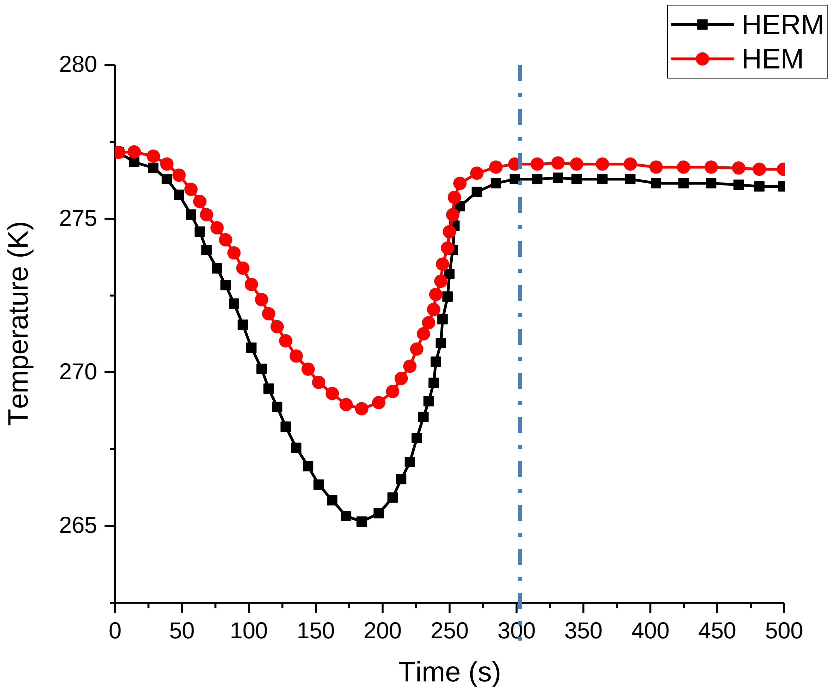


Figure 6.33: HEM and HERM transient temperature profiles at the top of the well for the fast injection ramping rate, case 1A (Table 6.1 feed temp = 278.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (300 s) at which the injection flow rate reaches its peak value of 33.5 kg/s

Figure 6.34 shows the HEM and HERM transient temperature profiles at the top of the well for the slow (2 hrs) injection flow rate ramp-up case 3A (Table 6.1 feed temp = 278.15 K, feed pressure = 115 bara). The simulation results obtained for a slow (2 hrs) flowrate ramp-up case using the HEM model produce a minimum wellhead temperature of 262 K (- 11 ^{o}C). The corresponding minimum temperature using the HERM model on the other hand is 252 K (-21 ^{o}C), demonstrating the importance of accounting for non-equilibrium effects and the model's usefulness as a tool for the development of optimal injection strategies for minimising the risks associated with the injection of CO_2 into depleted gas fields.

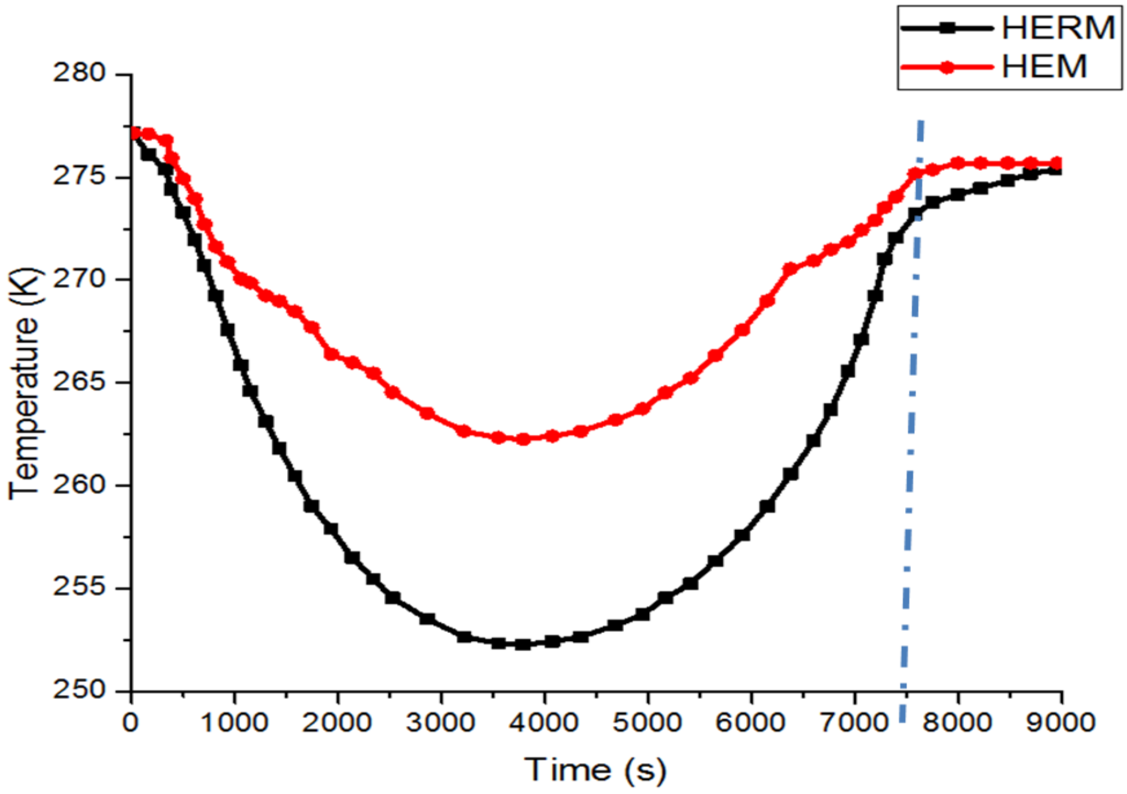


Figure 6.34: HEM and HERM transient temperature profiles at the top of the well for the slow (2 hrs) injection flow rate ramp-up case 3A (Table 6.1 feed temp = 278.15 K, feed pressure = 115 bara). The vertical dashed line indicates the time (7200 s) at which the injection flow rate reaches its peak value of 33.5 kg/s

6.6 Concluding remarks

This chapter has led to the development and testing of a rigorous HERM model for the simulation of the highly-transient multi-phase flow phenomena taking place in wellbores during the start-up injection of high pressure CO_2 into depleted gas fields. Similar to the HEM, the HERM model developed can serve as a valuable tool for the development of optimal injection strategies and best-practice guidelines for the minimisation of the risks associated with the start-up injection of CO_2 into highly-depleted gas fields. In order to assess which parameters have the highest impact on the transient flow of CO_2 in the injection well, a sensitivity analysis was performed over fours parameters not directly assessed in previous publications. Specifically, the study considered different restrictions of the cross-sectional area at 800m down the well, the deviation of the well from a perfectly vertical configuration, different durations for the ramp-up injection and different CO_2 feed stream temperatures.

Based on the application of the HERM model to realistic test cases involving the injection of CO_2 into the Goldeneye depleted reservoir in the North Sea, the main findings of this work can be summarised as follows:

- the degree of cooling along the injection well becomes less severe with a decrease in the injection duration. In other words, operating a fast start-up injection (between 5 and 30 minutes) is recommended rather than a slower one (over 2 hours);
- the degree of cooling along the injection well becomes less severe with a more pronounced internal diameter restriction at 800m along the well. Moreover, such restriction plays a paramount role in the build-up of pressure and temperature at the top of the well: both quantities attain higher values when the diameter restriction is more pronounced;
- when a perfectly vertical well is considered, for any given ramp-up injection strategy both temperature and pressure drops are more severe than those exhibited in the case of a deviated well. Moreover, for the least pronounced internal diameter restriction, no build-up in pressure or temperature is observed, and an undesired two-phase flow is reached as steady-state condition at the top of the well;
- the minimum pressure and temperature at the wellhead are higher when the CO_2 stream is injected at a higher temperature;

- the formation of ice during the injection process is likely, given that the minimum fluid temperature falls well below 0 ^{o}C. This poses the risk of well blockage should a sufficient quantity of water be present;
- the minimum simulated CO_2 temperature and the corresponding pressures predicted along the well during the start-up injection process are well outside the ranges where CO_2 hydrate or CO_2 solid formation would be expected;
- given the observed relatively modest drop in temperature along the well, its failure due to thermal shocking during the injection process is highly unlikely.
- finally, a comparison between the HEM and HERM simulation results for the transient pressure and temperature profiles at the top of the well for the fast injection ramping rate case shows that the HERM predicted twice lower temperature drop than the HEM. The simulation results obtained for a slow (2 hrs) flowrate ramp-up case using the HEM model produce a minimum wellhead temperature of 262 K (- 11 ^{o}C) while the corresponding minimum temperature using the HERM model on the other hand is 252 K (- 21 ^{o}C),

Chapter 7: Conclusions and recommendations for future work

7.1 Conclusions

This thesis presented the development and verification of a Homogeneous Equilibrium Mixture (HEM) model and a Homogenous Equilibrium Relaxation Mixture (HERM) model for simulating the transient flow phenomena taking place during the injection of CO_2 into highly depleted gas fields. The HEM model considers the two-phase mixture being injected to be at homogeneous equilibrium, which means instantaneous interface mass, momentum and energy exchange. As such, the constituent gas and liquid phases are assumed to remain at the same pressure, temperature and velocity. The corresponding fluid flow can be described using a single set of the mass, momentum and energy conservation equations. For fluid/wall friction and heat exchange, empirical correlations were utilised. The HERM on the other hand presented an additional equation which accounted for the thermodynamic non-equilibrium relaxation time in two-phase multi-component mixtures.

The models presented in this thesis have showed to serve as valuable tools for the development of optimal injection strategies and best-practice guidelines for the minimisation of the risks associated with the injection of CO_2 into highly-depleted gas fields. In this thesis, the major CO_2 injection process risks identified to be minimised are:

- blockage due to hydrate and ice formation following contact of the cold CO_2 with the interstitial water around the wellbore and the formation water in the perforations at the near well zone;
- thermal stress shocking of the wellbore casing steel due to the tube inner and outer temperature gradient leading to its fracture and escape of CO_2;
- over-pressurisation accompanied by CO_2 backflow into the injection system due to the violent evaporation of the superheated liquid CO_2 upon entry into the wellbore.

As such developing appropriate start-up injection strategies for avoiding the above risks is of paramount importance.

In chapter 2, a literature review covering the current challenges in the large-scale deployment of CCS and recent CO_2 injection related publications was presented. The chapter covers important areas such as:

- the Joule-Thomson expansion effect which describes the flow of the expanding CO_2 into the injection well
- the description of the general conservation equations for fluid dynamics, pipe flows and the constituent relations for fluid/wall and fluid/fluid interface interactions
- the Equation-of-State (EoS) for predicting the thermodynamics and transport properties and equilibrium data of the overall formulations.
- a review of previous application and performance of the Homogeneous Equilibrium Mixture (HEM) and Homogeneous Equilibrium Relaxation Mixture (HERM) models on pipeline transportation.

In chapter 3, the Finite Volume Method numerical solution to the HEM flow model was presented. The simulation results of the HEM model show good agreement with experimental data and those published in literature. The conclusions drawn from the findings of the test case presented in chapter 3 are:

- the degree of cooling along the injection well becomes more severe with increasing injection duration. Following the rapid Joule-Thomson cooling effect at the wellhead when the higher pressure CO_2 is introduced, the cooling rate then decreases with well depth;

- the formation of ice is likely during the injection process given that the minimum fluid temperature falls well below 0 ^oC. This poses the risk of well blockage in the event that sufficient quantity of water is present;

- the minimum simulated CO_2 temperature and the corresponding pressures predicted along the well during the start-up injection process were well outside the ranges where CO_2 hydrate or CO_2 solid formation would be expected;

- given the observed relatively modest drop in temperature along the well, its failure due to thermal shocking during the injection process is highly unlikely;

- also the wellhead pressure and temperature profiles obtained in the current simulation showed very good agreement with experiment and those published in the literature.

In Chapter 4, the HEM flow model is employed to simulate sudden blowout from the CO_2 injection well following an overpressure and backflow following the rapid expansion and superheating of the dense phase CO_2. The chapter clearly showed an application of the boundary condition closure equations for the HEM model. The results were produced by using the boundary conditions presented in section 4.2 as the closure equations for the HEM model in chapter 3. The first case study compares the simulated pressure and temperature changes during start-up injection and blowout prior to injection while the second case study compares blowout prior to injection and blowout after 5 mins of injection. The concluding remarks from the results obtained are outlined below.

- At the wellhead, there is significant drop ca. 266 K (– 7 ^{o}C) in temperature during injection. This poses the risk of ice formation in the presence of interstitial water molecules and hence the danger of well blockage. The CO_2 pressure and temperature ca. 28 bara and 264 K (– 9 ^{o}C) predicted respectively at the wellhead shows lower values for the blowout case compared with the injection case. More importantly based on the case studies conducted, a phase change from gas to dense phase CO_2 occurs after 50 s of injection at the top of the well. This transition is important as dense phase CO_2 injection is highly recommended due to its high density and low viscosity. However, there is danger of over-pressurisation accompanied by CO_2 backflow into the injection system due to the violent evaporation of the superheated liquid CO_2 upon entry into the well bore.

- The analysis for the two blowout scenarios show that during sudden blowout, a higher wellhead temperature will minimise the overall temperature drop compared with lower wellhead temperatures. Also, the minimum temperature at the wellhead for the case of blowout after 5 mins of injection is below the freezing point of water. This means there is a possibility of ice formation should the CO_2 come in contact with interstitial water molecules.

- At the bottom of the well, a rapid pressure build-up during injection occurs. After 5 mins of injection the pressure at the bottom of the well raises from 177 bara to 196 bara before the blowout. Consequently, after 5 mins of outflow during blowout the pressure at the bottom of the well slowly decrease to 190 bara. Also at the bottom of the well, a rapid decline in temperature from 353 K to 290 K during injection occurs but a slow decline from 290 K to 287 K during blowout. The predicted results at the bottom of the well gives a clear indication of increase in pressure due to the expanding dense phase CO_2. Also, a decrease in temperature is observed indicating a cooling CO_2 stream at the bottom of the well during injection. However, the behaviour was different during blowout, as a slow decrease in pressure and temperature at the bottom of the well is observed. Significantly at the bottom of the well, the temperature is well above the freezing point of water, thus there is no risk of ice or hydrate formation.

In Chapter 5, the HEM flow model was extended to account for the injection flow rates ramping up times and CO_2 stream impurities impact. Three injection flow rate ramping up durations corresponding to slow, medium and fast were simulated. A case study on the impact of varying range of impurities expected from pre-combustion, post-combustion and oxy-fuel captured CO_2 on the pressure and temperature profiles at the top and bottom of the well during CO_2 injection was investigated. The chapter was divided into two main sections.

In the first section of chapter 5 the HEM model was applied to investigate the impact of flow rates ramping up times corresponding to fast (5 mins), medium (30 mins) and slow (2 hrs) ramp-up. Noting that varying injection ramping up times considered in this study has never been studied in details in previous publications. As such, this study introduces a new area of consideration and gives a clearer insight on the impact of CO_2 injection flow rates ramping up times on the wellbore pressure and temperature profiles.

Based on the application of the HEM model to realistic test cases involving the ramping up of CO_2 injection flowrates from 0 to 38 kg/s into the Goldeneye depleted gas reservoir in the North Sea following fast (5 mins), medium (30 mins) and slow (2 hrs) injection ramp-up times, the key findings of the sub-section are summarised as follows:

- The degree of cooling along the injection well becomes less severe with a decrease in the injection ramp-up duration. In other words, operating a fast ramping up injection is recommended (i.e. between 5 and 30 mins) rather than a much slower one (over 2 hrs);

- The formation of ice during the ramp-up injection process is likely, given that in all three cases considered the minimum fluid temperature falls well below 0 ^{o}C at the top of the well. This poses the risk of ice formation and ultimately well blockage should a sufficient quantity of water be present. Thus, for this case study injecting CO_2 at temperatures above 290 K at flowrate ramp-up duration of 5 mins will most likely minimise this risk of ice formation.

- The minimum simulated CO_2 temperature and the corresponding pressures predicted at the top of well during the start-up injection flowrate ramp-up process are well close to the ranges where CO_2 hydrates formation would be expected.

- Given the observed relatively large drop in temperature (up to – 50 ^oC) at the top of the well especially for the 2 hrs injection flow rate ramping up case, this poses a risk of system failure.

In the later part of chapter 5, an analysis of the impact of CO_2 stream impurities on the wellhead pressure and temperature profiles during the injection of CO_2 into depleted gas fields was performed. Pre-combustion, post-combustion and oxyfuel captured CO_2 streams with varying impurities combinations were considered. The simulation results showed that Oxyfuel and pre-combustion captured CO_2 mixtures imposed the greatest effect of pressure drop at the start of injection. The CO_2 stream captured by oxyfuel combustion containing 5.8% N_2, 4.7% O_2, 4.47% Ar and other impurities at ppm level recorded the highest pressure drop of 69 bara from the inlet pressure of 115 bara. On the other hand, pre-combustion captured CO_2 stream containing 0.09 % N_2, 0.03 % O_2, 0.02 % Ar and other impurities at ppm level showed a minimal drop in pressure. Also, the effect on the temperature drop as seen in the results, oxyfuel and pre-combustion captured methods with higher level of CO_2 stream impurities mixtures showed the greatest impact on temperature drop at the start of injection. The oxyfuel captured CO_2 stream with higher levels of impurities imposed much lower temperatures on the wellbore down to the bottom of the well. The effect of such low temperatures on the injection system is highly significant and accounted as a critical factor for safe injection of CO_2. Based on the fact that there is a possibility of interstitial water molecules present within the wellbore during injection, the formation of ice is likely and may pose significant safety risks.

Chapter 6 presented the development and testing of varying area and inlet temperature sensitivity analysis of the HERM model capable of predicting thermodynamic non-equilibrium in two-phase flows. Using a series of case studies, the injection flow rates ramping up for slow, medium and fast ramping up cases at varying inlet temperature are simulated to produce relevant pressure, temperature and density profiles. Specifically, the study considered different restrictions of the cross-sectional area at 800m down the well, the deviation of the well from a perfectly vertical configuration, different durations for the ramp-up injection and different CO_2 feed stream temperatures.

Based on the application of the HERM model to realistic test cases involving the injection of CO_2 into the Goldeneye depleted reservoir in the North Sea, a comparison between the HEM and HERM simulation results for the transient pressure and temperature profiles at the top of the well for the fast (5 mins) injection ramping up case was conducted. The HERM predicts a much lower CO_2 pressure and temperature compared to the HEM. The simulation results obtained for a slow (2 hrs) flowrate ramp-up case using the HEM model produce a minimum wellhead temperature of - 11 ^{o}C. The corresponding minimum temperature using the HERM model on the other hand is - 21 ^{o}C, demonstrating the importance of accounting for non-equilibrium effects and the model's usefulness as a tool for the development of optimal injection strategies for minimising the risks associated with the injection of CO_2 into depleted gas fields.

Finally, the work of mathematical and computational models presented in this study provide a basis for the reliable assessment of the injection of the CO_2 and the consequences associated with its rapid quasi-adiabatic expansion commonly known as "Joule-Thomson expansion" leading to large temperature drop. The study by allowing the analysis of ramping-up effect, thermodynamic non-equilibrium, and stream impurities impact has gone beyond the current state of the art and tackled some of the theoretical drawbacks established before now. The models developed are also directly applicable to systems of hydrocarbons.

Since the predicted data from the flow models act as the input in determining the expected performance and safety of the CO_2 injection system, the developments in this work will help to quantify the corresponding hazard profiles with improved certainty. This in turn yields cost benefits in view of the design and implementation of relevant protection and mitigation systems. Thus, the fundamental nature of this work will benefit CO_2 injection process safety and the large-scale implementation of CCS by protecting life, property and the environment.

Finally, it is critically important however to bear in mind that the above conclusions are not universal. On the contrary, they are only based on the case studies investigated. Each injection scenario must be individually examined in order to determine the likely risks. Most importantly, the computational tool needed to make such assessment has been developed in this thesis.

7.2 Suggestions for future works

The experience gained during this work has opened new interesting areas of research, which would be worth investigating. Two main potential areas of developments are highlighted here.

7.2.1 Extended CO_2 stream impurities impact study

Significantly, all the previous work on well injection has been confined to pure CO_2. In practice the injected stream will contain a varying range of different impurities, which will have a profound impact on the CO_2 fluid phase behaviour and hence its injectivity. Although, this study has opened an interesting line of research that would be to closely examine how the presence of stream impurities can affect the drop and the recovery in pressure and temperature. There is need for further research using the HERM or a more robust heterogeneous model that accounts for the properties of each component in the mixture. Also, different injection strategies (varying pressure, temperature and ramp-up operations) can be tested as well, in order to understand the impact produced by the impurities and, in particular, their compositions, which can vary among different CO_2 emitters. An appropriate multi-fluid model with flow-regime-dependent correlations will have to be developed and analysed for a wider range of CO_2 stream impurities.

7.2.2 Coupling with a reservoir model

An extended flow model will have to be enhanced with additional boundary conditions, in order to take into consideration different scenarios not considered in this thesis. Specifically, regarding the time-dependent discharge of the CO_2 into the depleted reservoir, the use of a site-specific outflow boundary condition is an acceptable approximation in the case of depleted gas reservoirs whose properties are known. However, for the case of an Aquafer, a coupling between the HERM in the well and an appropriate Darcy solver for all the reservoir variables would be worth considering. In so doing, the system would be able to take into account more directly those properties of the reservoir (pressure, porosity, permeability, etc.) which vary in time as well and depend on the actual exploitation history of the reservoir under investigation.

References

A. Prosperetti & G. Tryggvason. (2007). *Computational Methods for Multiphase Flow*. *Journal of Fluid Mechanics* (Vol. 603). Cambridge University Press. http://doi.org/10.1017/S0022112008001250

Abudour, A. M., Mohammad, S. A., Robinson, R. L., & Gasem, K. A. M. (2013). Volume-translated Peng-Robinson equation of state for liquid densities of diverse binary mixtures. *Fluid Phase Equilibria*, *349*, 37–55. http://doi.org/10.1016/j.fluid.2013.04.002

Acevedo, L., & Chopra, A. (2017). Influence of Phase Behaviour in the Well Design of CO2Injectors. *Energy Procedia*, *114*(November 2016), 5083–5099. http://doi.org/10.1016/j.egypro.2017.03.1663

Afanasyev, A. a. (2013). Multiphase compositional modelling of CO2 injection under subcritical conditions: The impact of dissolution and phase transitions between liquid and gaseous CO2 on reservoir temperature. *International Journal of Greenhouse Gas Control*, *19*, 731–742. http://doi.org/10.1016/j.ijggc.2013.01.042

Alex, R. (IEAGHG). (2016). Impurities in the CO2 stream, (2015), 2–5.

Ammar, M. N., & Renon, H. (1987). The isothermal flash problem: New methods for phase split calculations. *AIChE Journal*, *33*(6), 926–939. http://doi.org/10.1002/aic.690330606

André, L., Audigane, P., Azaroual, M., & Menjoz, A. (2007). Numerical modeling of fluid-rock chemical interactions at the supercritical CO2-liquid interface during CO2 injection into a carbonate reservoir, the Dogger aquifer (Paris Basin, France). *Energy Conversion and Management*, *48*(6), 1782–1797. http://doi.org/10.1016/j.enconman.2007.01.006

Arts, R., Beaubien, S., Benedictus, T., Czernichowski-lauriol, I., Fabriol, H., Gundogan, O., … Vercelli, S. (2008). What does CO2 Geological Storage really mean? *CO2 GeoNet European Network of Excellence*. Retrieved from www.zeroemissionsplatform.eu//index.php?option=com_downloads&id=23

Arzanfudi, M. M., & Al-khoury, R. (2015). A compressible two-fluid multiphase model for CO2 leakage through a wellbore, (January), 477–507.

Baker, L. E., Pierce, A. C., & Luks, K. D. (1982). Gibbs Energy Analysis of Phase Equilibria. http://doi.org/10.2118/9806-PA

Battistelli, A., Ceragioli, P., & Marcolini, M. (2011). Injection of Acid Gas Mixtures in Sour Oil Reservoirs: Analysis of Near-Wellbore Processes with Coupled Modelling of Well and Reservoir Flow. *Transport in Porous Media*, *90*(1), 233–251. http://doi.org/10.1007/s11242-010-9685-6

Bilicki, Z., & Kestin, J. (1990). Physical Aspects of the Relaxation Model in Two-Phase Flow. In *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* (Vol. 428, pp. 379–397). The Royal Society. http://doi.org/10.1098/rspa.1990.0040

Black and Veatch. (2012). Cost and Performance Data for Power Generation Technologies. *Prepared for the National Renewable Energy Laboratory (NREL), U.S. Department of Energy*, (February).

Blackford, J., Beaubien, S., Foekema, E., Gemeni, V., Gwosdz, S., Jones, D., … Ziogou, F. (2010). A guide to potential impacts of leakage from CO2 storage, 61. http://doi.org/10.1016/j.egypro.2014.11.351

Blunt, M. (2010). Carbon dioxide storage. *Imperial College London Grantham Institute for Climate Change*, (4).

Böser, W., & Belfroid, S. (2013). Flow Assurance Study. *Energy Procedia*, *37*, 3018–3030. http://doi.org/10.1016/j.egypro.2013.06.188

Brown, S., Beck, J., Mahgerefteh, H., & Fraga, E. S. (2013). Global sensitivity analysis of the impact of impurities on CO2 pipeline failure. *Reliability Engineering and System Safety*, *115*, 43–54. http://doi.org/10.1016/j.ress.2013.02.006

Brown, S., Martynov, S., & Mahgerefteh, H. (2015). Simulation of two-phase flow through ducts with discontinuous cross-section. *Computers & Fluids*, *120*, 46–56. http://doi.org/10.1016/j.compfluid.2015.07.018

Brown, S., Martynov, S., Mahgerefteh, H., Chen, S., & Zhang, Y. (2014). Modelling the non-equilibrium two-phase flow during depressurisation of CO2 pipelines. *International Journal of Greenhouse Gas Control*, *In review*, 9–18. http://doi.org/10.1016/j.ijggc.2014.08.013

Brown, S., Martynov, S., Mahgerefteh, H., & Proust, C. (2013). A homogeneous relaxation flow model for the full bore rupture of dense phase CO2 pipelines. *International Journal of Greenhouse Gas Control*, *17*, 349–356. http://doi.org/10.1016/j.ijggc.2013.05.020

Chabook, M., Al-Ajmi, A., & Isaev, V. (2015). The role of rock strength criteria in wellbore stability and trajectory optimization. *International Journal of Rock Mechanics and Mining Sciences*, *80*, 373–378. http://doi.org/10.1016/j.ijrmms.2015.10.003

Chapman, S. J. (2003). Fortran 95/2003 for Scientists and Engineers Third Edition.

Chen, N. H. (1979). An Explicit Equation for Friction Factor in Pipe. *Industrial and Engineering Chemistry Fundamentals*, *18*(3), 296–297. http://doi.org/10.1021/i160071a019

Chen, N. H. (1979). An Explicit Equation for Friction Factor in Pipe. *Industrial & Engineering Chemistry Fundamentals*, *18*(3), 296–297. http://doi.org/10.1021/i160071a019

Circone, S., Stern, L. a., Kirby, S. H., Durham, W. B., Chakoumakos, B. C., Rawn, C. J., … Ishii, Y. (2003). CO $_2$ Hydrate: Synthesis, Composition, Structure, Dissociation Behavior, and a Comparison to Structure I CH $_4$ Hydrate. *The Journal of Physical Chemistry B*, *107*(23), 5529–5539. http://doi.org/10.1021/jp027391j

Cosham, A., Jones, D. G., Armstrong, K., Allason, D., & Barnett, J. (2012). The Decompression Behaviour of Carbon Dioxide in the Dense Phase. In *Proceedings of the 2012 9th International Pipeline Conference* (pp. 1–18). Calgary: ASME. http://doi.org/10.1115/IPC2012-90461

Cotton, A., Gray, L., & Maas, W. (2017). Learnings from the Shell Peterhead CCS Project Front End Engineering Design. *Energy Procedia, 114*, 5663–5670. http://doi.org/10.1016/J.EGYPRO.2017.03.1705

De Koeijer, G., Borch, J. H., Jakobsenb, J., & Drescher, M. (2009). Experiments and modeling of two-phase transient flow during CO2 pipeline depressurization. *Energy Procedia*, *1*(1), 1683–1689. http://doi.org/10.1016/j.egypro.2009.01.220

Denton, G. S. (2009). CFD simulation of highly transient flows, (July). Retrieved from http://eprints.ucl.ac.uk/18693/

Diamantonis, N. I., & Economou, I. G. (2011). Evaluation of statistical associating fluid theory (SAFT) and perturbed chain-SAFT equations of state for the calculation of thermodynamic derivative properties of fluids related to carbon capture and sequestration. *Energy and Fuels*, *25*(7), 3334–3343. http://doi.org/10.1021/ef200387p

Dittus, F. W., & Boelter, L. M. K. (1930). Heat Transfer in Automobile Radiators, *2*, 443.

Downar-Zapolski, P., & Bilicki, Z. (1996). The non-equilibrium relaxation model for one-dimensional flashing liquid flow. *International Journal of Multiphase Flow*, *22*(3), 473–483.

Downar-Zapolski, P., Bilicki, Z., Bolle, L., & Franco, J. (1996). The non-equilibrium relaxation model for one-dimensional flashing liquid flow. *International Journal of Multiphase Flow*, *22*(3), 473–483. http://doi.org/10.1016/0301-9322(95)00078-X

Dullemond, C. P. (University of H., & Johansen, a. (University of H. (2007). Chapter 6 Riemann solvers, *M*, 99–119. Retrieved from http://www.mpia-hd.mpg.de/~dullemon/lectures/hydrodynamicsII/

Ehrenberg, S. N., & Nadeau, P. H. (2005). Sandstone vs. carbonate petroleum reservoirs: A global perspective on porosity-depth and porosity-permeability relationships. *AAPG Bulletin*, *89*(4), 435–445. http://doi.org/10.1306/11230404071

Eiken, O., Ringrose, P., Hermanrud, C., Nazarian, B., Torp, T. A., & H??ier, L. (2011). Lessons Learned from 14 years of CCS Operations: Sleipner, In Salah and Sn??hvit. *Energy Procedia*, *4*, 5541–5548. http://doi.org/10.1016/j.egypro.2011.02.541

Elshahomi, A., Lu, C., Michal, G., Liu, X., Godbole, A., & Venton, P. (2015). Decompression wave speed in CO2 mixtures: CFD modelling with the GERG-2008 equation of state. *Applied Energy*, *140*, 20–32. http://doi.org/10.1016/j.apenergy.2014.11.054

ETI. (2016). *Reducing the cost of CCS developments in capture plant technology*.

Fisher, B. S., Nakicenovic, N., Alfsen, K., Morlot, J. C., Chesnaye, F. D. La, Hourcade, J.-C., … Warren, R. (2007). Issues related to mitigation in the long-term context. *Climate Change 2007: Mitigation. Contribution of Working Group III to the Fourth Assessment Report of the Inter-Governmental Panel on Climate Change*, 196–250.

Goodarzi, S., Settari, A., Zoback, M. D., & Keith, D. (2010). Thermal Aspects of Geomechanics and Induced Fracturing in CO2 Injection With Application to CO2 Sequestration in Ohio River Valley. *SPE International Conference on CO2 Capture, Storage, and Utilization*. Society of Petroleum Engineers. http://doi.org/10.2118/139706-MS

Hajiw, M., Corvisier, J., Ahmar, E. El, & Coquelet, C. (2018). Impact of impurities on CO2 storage in saline aquifers: Modelling of gases solubility in water. *International Journal of Greenhouse Gas Control*, *68*, 247–255. http://doi.org/10.1016/J.IJGGC.2017.11.017

Harding, S. T., & Floudas, C. a. (2000). Phase stability with cubic equations of state: Global optimization approach. *AIChE Journal*, *46*(7), 1422–1440. http://doi.org/10.1002/aic.690460715

Hasan, A. R., & Kabir, C. S. (1991). Heat Transfer During Two-Phase Flow in Wellbores; Part I--Formation Temperature. Society of Petroleum Engineers. http://doi.org/10.2118/22866-MS

Hasan, A. R., & Kabir, C. S. (2002). Fluid Flow and Heat Transfer in Wellbores. *Society of Petroleum Engineers*.

Henderson, N., de Oliveira, J. R., Souto, H. P. A., & Marques, R. P. (2001). Modeling and Analysis of the Isothermal Flash Problem and Its Calculation with the Simulated Annealing Algorithm. *Industrial & Engineering Chemistry Research*, *40*(25), 6028–6038. http://doi.org/10.1021/ie001151d

Hughes, D. S. (2009). Carbon storage in depleted gas fields: Key challenges. *Energy Procedia*, *1*(1), 3007–3014. http://doi.org/10.1016/j.egypro.2009.02.078

IEA. (2011). IEA Newsletter – June 2009.

IEAGHG (International Energy Agency greenhouse gas emissions). (2011). Effects of Impurities on Geological Storage of CO2, (June), Report: 2011/04.

IPCC. (2014). Climate Change 2014 Synthesis Report. *Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, 1–151.

IPCC, R. (2018). Global Warming of 1.5°C. Summary for Policymakers, (October 2018), 1 – 34. Retrieved from http://report.ipcc.ch/sr15/pdf/sr15_spm_final.pdf

Jiang, P., Li, X., Xu, R., Wang, Y., Chen, M., Wang, H., & Ruan, B. (2014). Thermal modeling of CO2 in the injection well and reservoir at the Ordos CCS demonstration project, China. *International Journal of Greenhouse Gas Control*, *23*, 135–146. http://doi.org/10.1016/j.ijggc.2014.01.011

Joseph, D. D., & Lundgren, T. S. (1989). Ensemble averaged and mixture theory equations. *IMA Preprint Series*, (April).

Juho, L. (2015). Storing CO 2 through Enhanced Oil Recovery. *OECD/IEA*.

Krogh, E., Nilsen, R., & Henningsen, R. (2012). Liquefied CO2 injection modelling. *Energy Procedia*, *23*(1876), 527–555. http://doi.org/10.1016/j.egypro.2012.06.022

Lemmon, E. W., & Tillner-Roth, R. (1999). A Helmholtz energy equation of state for calculating the thermodynamic properties of fluid mixtures. *Fluid Phase Equilibria*, *165*, 1–21. http://doi.org/10.1016/S0378-3812(99)00262-9

LeVeque, R. J. (2002a). Finite Volume Methods for Hyperbolic Problems. *Cambridge University Press*, *54*, 258. http://doi.org/10.1017/CBO9780511791253

LeVeque, R. J. (2002b). Finite Volume Methods for Hyperbolic Problems. *Cambridge University Press*, *54*, 258. http://doi.org/10.1017/CBO9780511791253

LeVeque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems:* Cambridge: Cambridge University Press. http://doi.org/10.1017/CBO9780511791253

Li, X., Li, G., Wang, H., Tian, S., Song, X., Lu, P., & Wang, M. (2017). International Journal of Greenhouse Gas Control A unified model for wellbore flow and heat transfer in pure CO 2 injection for geological sequestration , EOR and fracturing operations. *International Journal of Greenhouse Gas Control*, *57*, 102–115. http://doi.org/10.1016/j.ijggc.2016.11.030

Li, X., Xu, R., Wei, L., & Jiang, P. (2015). Modeling of wellbore dynamics of a CO2 injector during transient well shut-in and start-up operations. *International Journal of Greenhouse Gas Control*, *42*, 602–614. http://doi.org/10.1016/j.ijggc.2015.09.016

Lin, X.-B. (2000). Generalized Rankine–Hugoniot Condition and Shock Solutions for Quasilinear Hyperbolic Systems. *Journal of Differential Equations*, *168*(2), 321–354. http://doi.org/10.1006/jdeq.2000.3889

Lindeberg, E. (2011). Modelling pressure and temperature profile in a CO2 injection well. *Energy Procedia*, *4*, 3935–3941. http://doi.org/10.1016/j.egypro.2011.02.332

Linga, G., & Lund, H. (2016). A two-fluid model for vertical flow applied to CO2 injection wells. *International Journal of Greenhouse Gas Control*, *51*, 71–80. http://doi.org/10.1016/j.ijggc.2016.05.009

Liou, M. (1996). A Sequel to AUSM: AUSM. *Journal of Computational Physics*, *129*, 364–382. http://doi.org/10.1006/jcph.1996.0256

Liou, M. M.-S., Nguyen, L., Theofanous, T. G., Chang, C. C.-H., Nguyen, L., & Theofanous, T. G. (2008). How to Solve Compressible Multifluid Equations: a Simple, Robust, and Accurate Method. *AIAA Journal*, *46*(9), 2345–2356. http://doi.org/10.2514/1.34793

Liou, M. S. (2006). A sequel to AUSM, Part II: AUSM+-up for all speeds. *Journal of Computational Physics*, *214*(1), 137–170. http://doi.org/10.1016/j.jcp.2005.09.020

Lu, M., & Connell, L. D. (2008). Non-isothermal flow of carbon dioxide in injection wells during geological storage. *International Journal of Greenhouse Gas Control*, *2*(2), 248–258. http://doi.org/10.1016/S1750-5836(07)00114-4

Lu, M., & Connell, L. D. (2014a). The transient behaviour of CO2 flow with phase transition in injection wells during geological storage – Application to a case study. *Journal of Petroleum Science and Engineering*, *124*, 7–18. http://doi.org/10.1016/j.petrol.2014.09.024

Lu, M., & Connell, L. D. (2014b). Transient, thermal wellbore flow of multispecies carbon dioxide mixtures with phase transition during geological storage. *International Journal of Multiphase Flow*, *63*, 82–92. http://doi.org/10.1016/j.ijmultiphaseflow.2014.04.002

Lund, H., Torsæter, M., & Munkejord, S. T. (2015). Study of thermal variations in wells during CO<inf>2</Study of thermal variationsinf> injection. *Society of Petroleum Engineers - SPE Bergen One Day Seminar 2015*, *31*(June), 327–337. http://doi.org/10.2118/173864-PA

Mahgerefteh, H., Atti, O., & Denton, G. (2007). An Interpolation Technique for Rapid CFD Simulation of Turbulent Two-Phase Flows. *Process Safety and Environmental Protection*, *85*(B1), 45–50. http://doi.org/10.1205/psep.05118

Mahgerefteh, H., Brown, S., & Denton, G. (2012). Modelling the impact of stream impurities on ductile fractures in CO2 pipelines. *Chemical Engineering Science*, *74*, 200–210. http://doi.org/10.1016/j.ces.2012.02.037

Mahgerefteh, H., Brown, S., & Denton, G. (2012). Modelling the impact of stream impurities on ductile fractures in CO2 pipelines. *Chemical Engineering Science*, *74*, 200–210. http://doi.org/10.1016/j.ces.2012.02.037

Mahgerefteh, H., Brown, S., & Martynov, S. (2012). A study of the effects of friction, heat transfer, and stream impurities on the decompression behavior in CO2 pipelines. In *Greenhouse Gases: Science and Technology* (Vol. 2, pp. 369–379). http://doi.org/10.1002/ghg.1302

Mahgerefteh, H., Brown, S., & Zhang, P. (2011). A dynamic boundary ductile-fracture-propagation model for CO2 pipelines. *Journal of Pipeline Engineering*, *9*(4), 265–276.

Mahgerefteh, H., Denton, G., & Rykov, Y. (2008). A hybrid multiphase flow model. *AIChE Journal*, *54*(9), 2261–2268. http://doi.org/10.1002/aic

Mahgerefteh, H., Oke, A., & Atti, O. (2006). Modelling outflow following rupture in pipeline networks. *Chemical Engineering Science*, *61*, 1811–1818. http://doi.org/10.1016/j.ces.2005.10.013

Mahgerefteh, H., Oke, A. O., & Rykov, Y. (2006). Efficient numerical solution for highly transient flows. *Chemical Engineering Science*, *61*(15), 5049–5056. http://doi.org/10.1016/j.ces.2006.03.012

Mahgerefteh, H., Rykov, Y., & Denton, G. (2009). Courant, Friedrichs and Lewy (CFL) impact on numerical convergence of highly transient flows. *Chemical Engineering Science*, *64*(23), 4969–4975. http://doi.org/10.1016/J.CES.2009.08.002

Mahgerefteh, H., Saha, P., & Economou, I. G. (1999). Fast numerical simulation for full-bore rupture of pressurized pipelines. *AIChE Journal*, *45*(6), 1191–1201.

Mahgerefteh, H., Saha, P., & Economou, I. G. (2000). Modeling fluid phase transition effects on dynamic behavior of ESDV. *AIChE Journal*, *46*(5), 997–1006. http://doi.org/10.1002/aic.690460512

Martens, S., Liebscher, A., Möller, F., Würdemann, H., Schilling, F., & Kühn, M. (2011). Progress report on the first european on-shore CO2 storage site at ketzin (Germany) — Second year of injection. *Energy Procedia*, *4*, 3246–3253. http://doi.org/10.1016/j.egypro.2011.02.243

Martynov, S., Brown, S., Mahgerefteh, H., & Sundara, V. (2013). Modelling choked flow for CO2 from the dense phase to below the triple point. *International Journal of Greenhouse Gas Control*, *19*, 552–558. http://doi.org/10.1016/j.ijggc.2013.10.005

Mathias, S. a., Gluyas, J. G., Oldenburg, C. M., & Tsang, C. F. (2010). Analytical solution for Joule-Thomson cooling during CO2 geo-sequestration in depleted oil and gas reservoirs. *International Journal of Greenhouse Gas Control*, *4*(5), 806–810. http://doi.org/10.1016/j.ijggc.2010.05.008

Michael, K., Neal, P. R., Allinson, G., Ennis-King, J., Hou, W., Paterson, L., … Aiken, T. (2011). Injection strategies for large-scale CO2 storage sites. *Energy Procedia*, *4*, 4267–4274. http://doi.org/10.1016/j.egypro.2011.02.376

Michelsen, L. M. (1982). THE ISOTHERMAL FLASH PROBLEM. PART I, *9*.

Michelsen, M. L. (1982). The isothermal flash problem. Part II. Stability. *Fluid Phase Equilibria*, *9*(1), 1–19. http://doi.org/10.1016/0378-3812(82)85001-2

Möller, F., Liebscher, A., Martens, S., Schmidt-Hattenberger, C., & Streibel, M. (2014). Injection of CO2 at ambient temperature conditions - Pressure and temperature results of the “cold injection” experiment at the Ketzin pilot site. In *Energy Procedia* (Vol. 63, pp. 6289–6297). http://doi.org/10.1016/j.egypro.2014.11.660

Munkejord, S. T., & Hammer, M. (2015). Depressurization of CO2-rich mixtures in pipes: Two-phase flow modelling and comparison with experiments. *International Journal of Greenhouse Gas Control*, *37*, 398–411. http://doi.org/10.1016/j.ijggc.2015.03.029

Munkejord, S. T., Hammer, M., & L??vseth, S. W. (2016). CO2 transport: Data and models - A review. *Applied Energy*, *169*, 499–523. http://doi.org/10.1016/j.apenergy.2016.01.100

Munkejord, S. T., Hammer, M., & Løvseth, S. W. (2016). CO2 transport: Data and models – A review. *Applied Energy*, *169*, 499–523. http://doi.org/10.1016/j.apenergy.2016.01.100

Munkejord, S. T., Jakobsen, J. P., Austegard, A., & Mølnvik, M. J. (2010). Thermo- and fluid-dynamical modelling of two-phase multi-component carbon dioxide mixtures. *International Journal of Greenhouse Gas Control*, *4*(7465), 589–596. http://doi.org/10.1016/j.ijggc.2010.02.003

Niu, Y.-Y., Lin, Y.-C., & Chang, C.-H. (2008). A further work on multi-phase two-fluid approach for compressible multi-phase flows. *International Journal for Numerical Methods in Fluids*, *58*, 879–896.

Nordbotten, J. M., & Celia, M. a. (2011). Geological Storage of CO2: The Carbon Problem. *Geological Storage of CO2*, 1–15. http://doi.org/10.1002/9781118137086.ch1

Nordbotten, J. M., Celia, M. A., & Bachu, S. (2005). Injection and storage of CO2 in deep saline aquifers: Analytical solution for CO2 plume evolution during injection. *Transport in Porous Media*, *58*(3), 339–360. http://doi.org/10.1007/s11242-004-0670-9

Nouri-Borujerdi, A., & Shafiei Ghazani, A. (2017). Equilibrium and non-equilibrium gas–liquid two phase flow in long and short pipelines following a rupture. *AIChE Journal*, *63*(7), 3214–3223. http://doi.org/10.1002/aic.15675

Oke, A., Mahgerefteh, H., Economou, I., & Rykov, Y. (2003). A transient outflow model for pipeline puncture. *Chemical Engineering Science*, *58*, 4591–4604. http://doi.org/10.1016/s0009-2509(03)00338-5

Oldenburg, C. M. (2007). Joule-Thomson cooling due to CO2 injection into natural gas reservoirs. *Energy Conversion and Management*, *48*(6), 1808–1815. http://doi.org/10.1016/j.enconman.2007.01.010

Oldenburg, C. M., & Pan, L. (2013). Porous Media Compressed-Air Energy Storage (PM-CAES): Theory and Simulation of the Coupled Wellbore-Reservoir System. *Transport in Porous Media*, *97*(m), 201–221. http://doi.org/10.1007/s11242-012-0118-6

Paillère, H., Corre, C., & García Cascales, J. R. (2003). On the extension of the AUSM+ scheme to compressible two-fluid models. *Computers and Fluids*, *32*(6), 891–916. http://doi.org/10.1016/S0045-7930(02)00021-X

Pale Blue Dot Energy. (2016). Progressing development of the UK's Strategic Carbon Dioxide Storage Resource: A Summary of Results from the Strategic UK CO2 Storage Appraisal Project, (April 2016). Retrieved from http://www.eti.co.uk/project/strategic-uk-ccs-storage-appraisal/?utm_medium=email&utm_campaign=Strategic UK CCS Storage Appraisal announcement&utm_content=Strategic UK CCS Storage Appraisal announcement+CID_3258184913365b7308725bbdb1076a42&utm_source=Emai

Pan, L., & Oldenburg, C. M. (2014). T2Well-An integrated wellbore-reservoir simulator. *Computers and Geosciences*, *65*, 46–55. http://doi.org/10.1016/j.cageo.2013.06.005

Pan, L., Webb, S. W., & Oldenburg, C. M. (2011). Analytical solution for two-phase flow in a wellbore using the drift-flux model. *Advances in Water Resources*, *34*(12), 1656–1665. http://doi.org/10.1016/j.advwatres.2011.08.009

Paterson, L., Ennis-King, J. P., & Sharma, S. (2010). Observations of Thermal and Pressure Transients in Carbon Dioxide Wells. Society of Petroleum Engineers. http://doi.org/10.2118/134881-MS

Paterson, L., Lu, M., Connell, L., & Ennis-King, J. (2008a). Numerical Modeling of Pressure and Temperature Profiles Including Phase Transitions in Carbon Dioxide Wells. *SPE Annual Technical Conference and Exhibition, Denver, Colorado USA, Sept 21-24*. Society of Petroleum Engineers. http://doi.org/10.2118/115946-MS

Paterson, L., Lu, M., Connell, L., & Ennis-King, J. P. (2008b). Numerical Modeling of Pressure and Temperature Profiles Including Phase Transitions in Carbon Dioxide Wells. Society of Petroleum Engineers. http://doi.org/10.2118/115946-MS

Peng, D. Y., & Robinson, D. B. (1976). A new two-constant equation of state. *Industrial Engineering Chemistry Fundamentals*, *15*, 59–64.

Peng, D.-Y., & Robinson, D. B. (1976). A New Two-Constant Equation of State. *Industrial & Engineering Chemistry Fundamentals*, *15*(1), 59–64. http://doi.org/10.1021/i160057a011

Poling, B. E., Prausnitz, J. M., & O'Connell, J. P. (2001). *The properties of gases and liquids*. McGraw-Hill.

Porter, R. T. J., Fairweather, M., Pourkashanian, M., & Woolley, R. M. (2015). The range and level of impurities in CO2 streams from different carbon capture sources. *International Journal of Greenhouse Gas Control*, *36*, 161–174. http://doi.org/10.1016/j.ijggc.2015.02.016

Raza, A., Gholami, R., Rezaee, R., Bing, C. H., Nagarajan, R., & Hamid, M. A. (2016). Well selection in depleted oil and gas fields for a safe CO2 storage practice: a case study from Malaysia. *Petroleum*. http://doi.org/10.1016/j.petlm.2016.10.003

Robbins, D. J., Cant, R. S., & Gladden, L. F. (2013). Development of accurate, robust liquid equations of state for multi-phase CFD simulations with a modified AUSM+-up scheme. *Computers and Fluids*, *77*, 166–180. http://doi.org/10.1016/j.compfluid.2013.01.031

Ruan, B., Xu, R., Wei, L., & Jiang, P. (2013). Thermal modelling in and around a CO2 injector. *Energy Procedia*, *37*, 3283–3290. http://doi.org/10.1016/j.egypro.2013.06.216

Ruan, B., Xu, R., Wei, L., Ouyang, X., Luo, F., & Jiang, P. (2013). Flow and thermal modeling of CO2 in injection well during geological sequestration. *International Journal of Greenhouse Gas Control*, *19*, 271–280. http://doi.org/10.1016/j.ijggc.2013.09.006

Sanchez Fernandez, E., Naylor, M., Lucquiaud, M., Wetenhall, B., Aghajani, H., Race, J., & Chalmers, H. (2016). Impacts of geological store uncertainties on the design and operation of flexible CCS offshore pipeline infrastructure. *International Journal of Greenhouse Gas Control*, *52*, 139–154. http://doi.org/10.1016/j.ijggc.2016.06.005

Sasaki, K., Yasunami, T., & Sugaia, Y. (2009). Prediction model of bottom hole temperature and pressure at deep injector for CO2 sequestration to recover injection rate. *Energy Procedia*, *1*(1), 2999–3006. http://doi.org/10.1016/j.egypro.2009.02.077

Shell. (2015). *Peterhead CCS Project*.

Shell Canada Limited. (2017). Quest Carbon Capture and Storage Project.

Shell UK, T. R. (2015). Peterhead CCS Project.

Span, R., & Wagner, W. (1996). A New Equation of State for Carbon Dioxide Covering the Fluid Region from the Triple-Point Temperature to 1100 K at Pressures up to 800 MPa. *Journal of Physical and Chemical Reference Data*, *25*(6), 1509. http://doi.org/10.1063/1.555991

Steiner, D., & Taborek, J. (1992). Flow boiling heat transfer in vertical tubes correlated by an asymptotic method. *Heat Transfer Engineering*, *13*(2), 43–69.

Steiner, O. (2006). Introduction to Riemann Solvers 1 . Three major advancements in the numerical treatment of the hydrodynamic equations, 12–14.

Techo, R., Tickner, R. R., & James, R. E. (1965). An accurate equation for the computation of the friction factor for smooth pipes from the Reynolds number. *Journal of Applied Mechanics*, *32*(2), 443–443.

Teng, L., Li, Y., Zhao, Q., Wang, W., Hu, Q., Ye, X., & Zhang, D. (2016). Decompression characteristics of CO2 pipelines following rupture. *Journal of Natural Gas Science and Engineering*, *36*, 213–223. http://doi.org/10.1016/j.jngse.2016.10.026

Thompson, K. W. (1987). Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, *68*(1), 1–24. http://doi.org/10.1016/0021-9991(87)90041-6

Thompson, K. W. (1990). Time-dependent boundary conditions for hyperbolic systems, {II}. *Journal of Computational Physics*, *89*, 439–461. http://doi.org/10.1016/0021-9991(90)90152-Q

Thu, E. S. (2013). Modeling of Transient CO2 Flow in Pipelines and Wells.

Toro, E. (2012). The HLLC Riemann Solver Abstract : This lecture is about a method to solve approximately.

Toro, E. F. (2009). The HLL and HLLC Riemann Solvers. In *Riemann Solvers and Numerical Methods for Fluid Dynamics* (3rd ed., pp. 293–311). Heidelberg: Springer. http://doi.org/10.1007/978-3-662-03490-3_10

United Nation, F. (2015). COP21 Agreement: Framework Convention on Climate Change, *21930*(December). http://doi.org/FCCC/CP/2015/L.9

Wang, J., Ryan, D., Anthony, E. J., Wildgust, N., & Aiken, T. (2011). Effects of impurities on CO2transport, injection and storage. *Energy Procedia*, *4*, 3071–3078. http://doi.org/10.1016/j.egypro.2011.02.219

Wang, J., Wang, Z., Ryan, D., & Lan, C. (2015). A study of the effect of impurities on CO2 storage capacity in geological formations. *International Journal of Greenhouse Gas Control*, *42*, 132–137. http://doi.org/10.1016/J.IJGGC.2015.08.002

Wetenhall, B., Aghajani, H., Chalmers, H., Benson, S. D., Ferrari, M.-C., Li, J., … Davison, J. (2014). Impact of CO2 impurity on CO2 compression, liquefaction and transportation. *Energy Procedia*, *63*, 2764–2778. http://doi.org/10.1016/j.egypro.2014.11.299

Wetenhall, B., Race, J. M., & Downie, M. J. (2014). The Effect of CO2 Purity on the Development of Pipeline Networks for Carbon Capture and Storage Schemes. *International Journal of Greenhouse Gas Control*, *30*, 197–211. http://doi.org/10.1016/J.IJGGC.2014.09.016

Wiese, B., Nimtz, M., Klatt, M., & K??hn, M. (2010). Sensitivities of injection rates for single well CO2 injection into saline aquifers. *Chemie Der Erde - Geochemistry*, *70*(SUPPL. 3), 165–172. http://doi.org/10.1016/j.chemer.2010.05.009

Wildgust, N., Aiken, T., & Capture, C. (2010). Injection Strategies for Large- Scale. *Carbon*, (May).

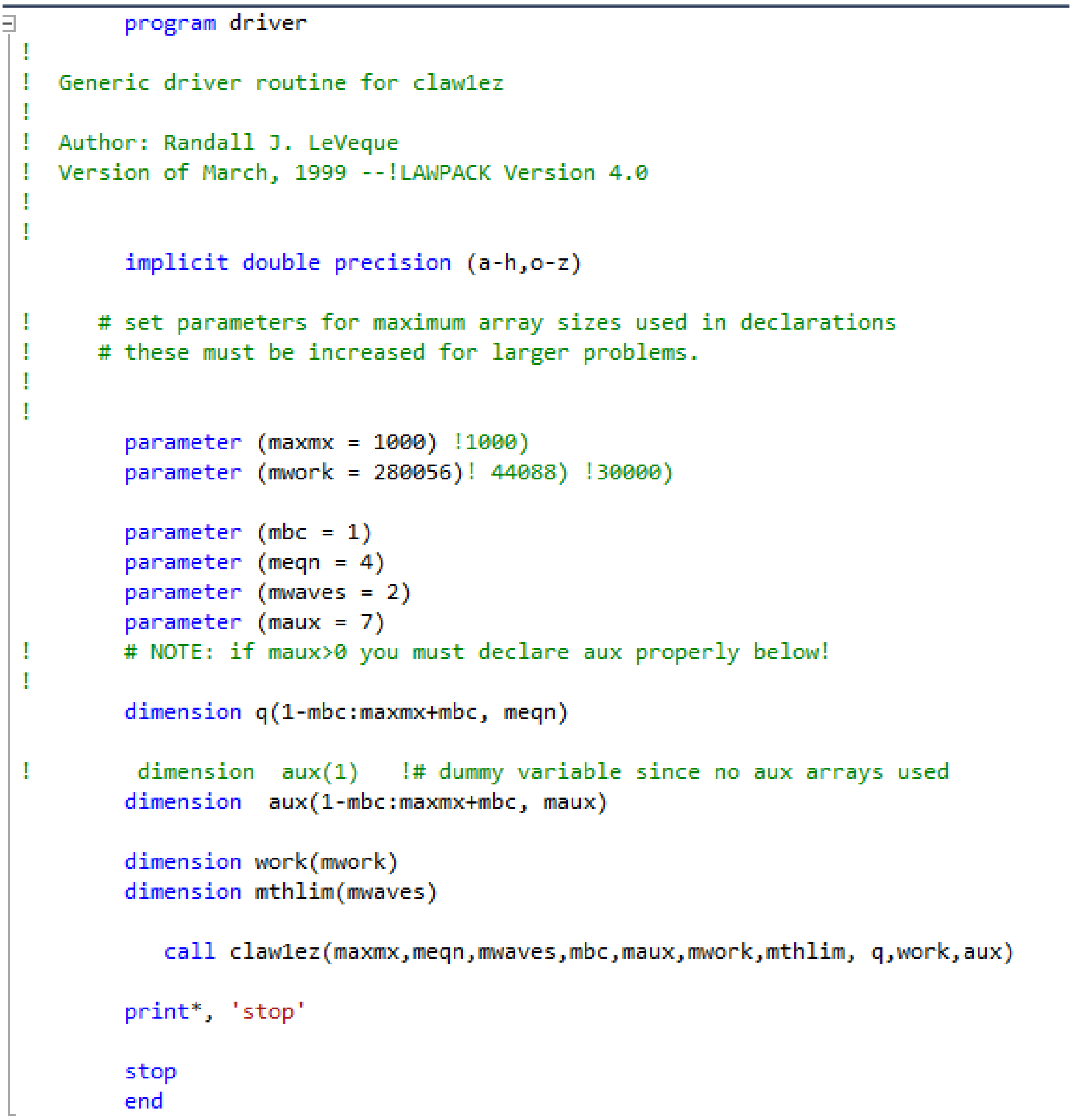
Wood, D. A. (2016). Metaheuristic profiling to assess performance of hybrid evolutionary optimization algorithms applied to complex wellbore trajectories. *Journal of Natural Gas Science and Engineering*, *33*, 751–768. http://doi.org/10.1016/j.jngse.2016.05.041

Yang, M., Song, Y., Ruan, X., Liu, Y., Zhao, J., & Li, Q. (2012). Characteristics of CO2 hydrate formation and dissociation in glass beads and silica gel. *Energies*, *5*(4), 925–937. http://doi.org/10.3390/en5040925

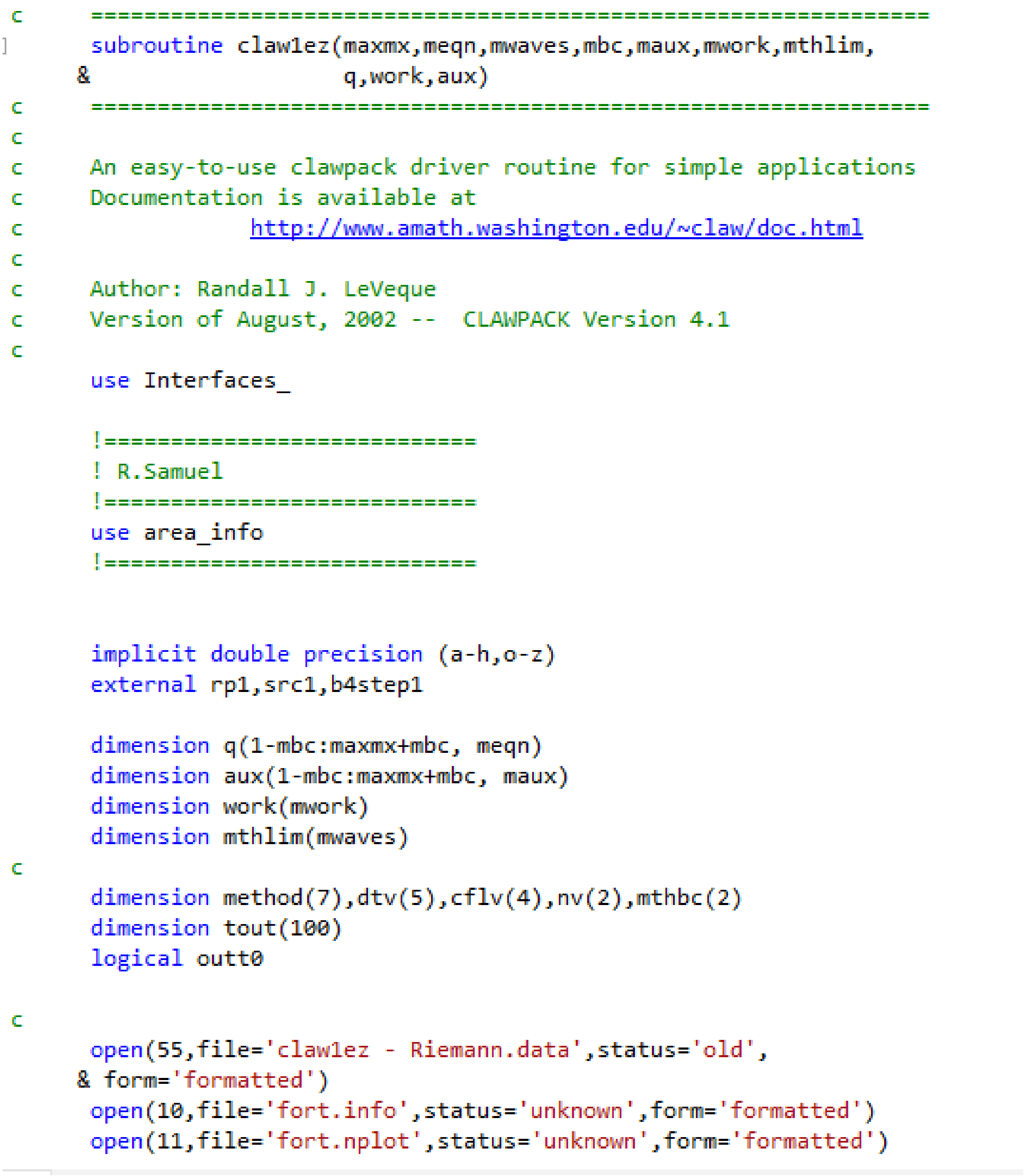
Zhao, Q., & Li, Y.-X. (2014). The influence of impurities on the transportation safety of an anthropogenic CO2 pipeline. *Process Safety and Environmental Protection*, *92*(1), 80–92. http://doi.org/10.1016/j.psep.2013.10.007

Zucrow, M. J., & Hoffman, J. D. (1976). *Gas Dynamics*. New York: Wiley.

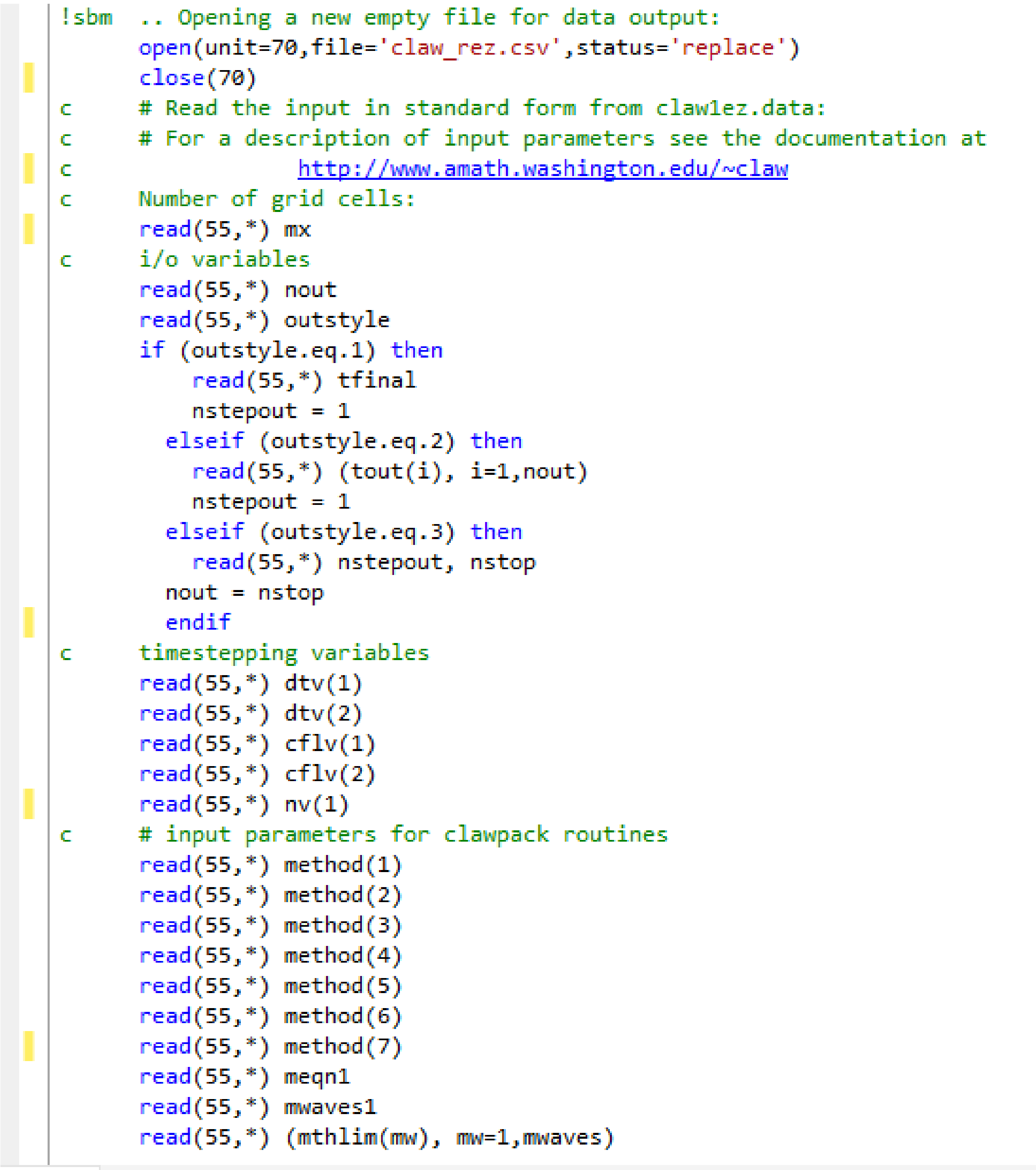
Appendix



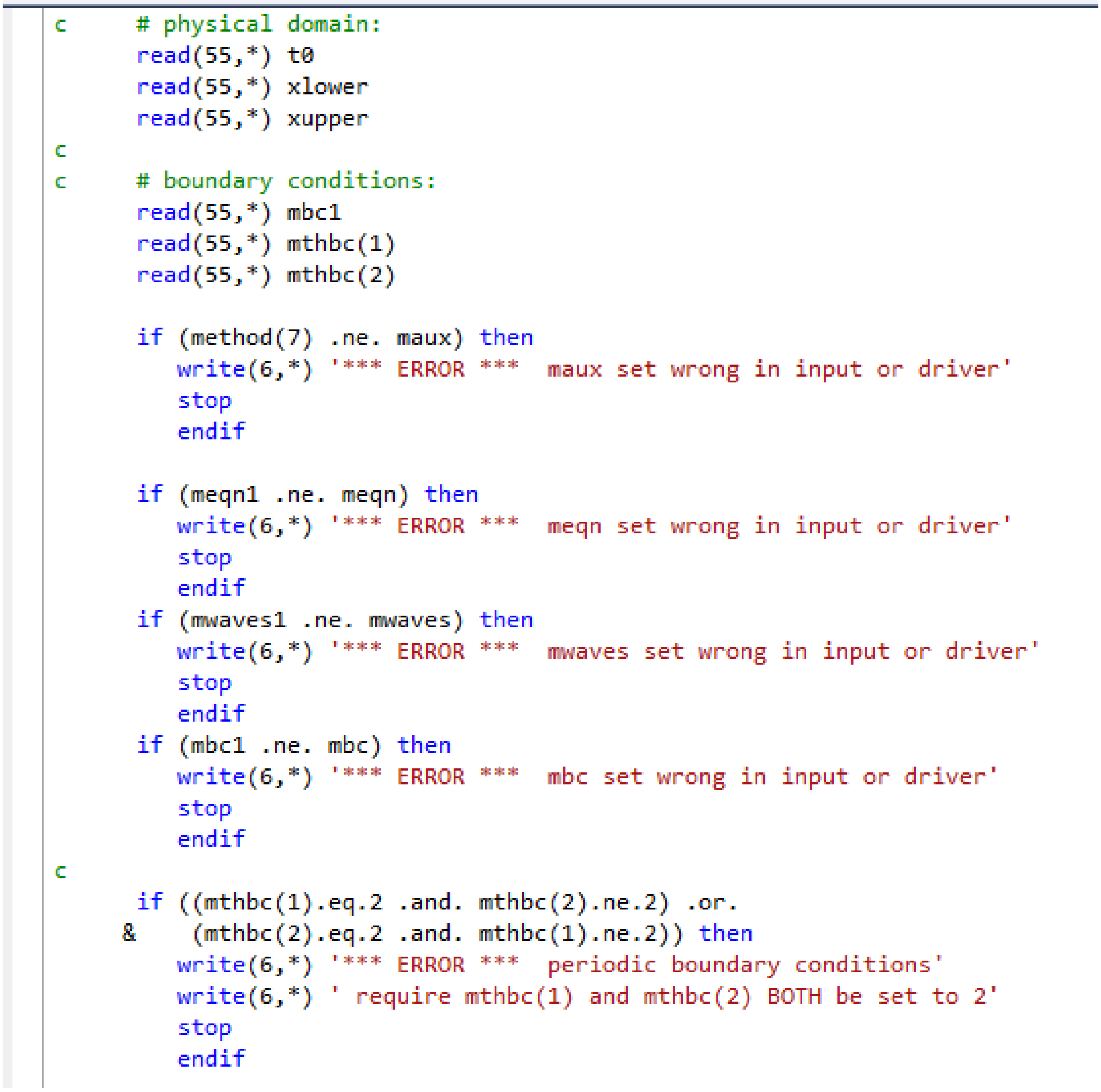
```
        program driver
!
!  Generic driver routine for claw1ez
!
!  Author: Randall J. LeVeque
!  Version of March, 1999 --!LAWPACK Version 4.0
!
!

        implicit double precision (a-h,o-z)

!     # set parameters for maximum array sizes used in declarations
!     # these must be increased for larger problems.
!
!

        parameter (maxmx = 1000) !1000)
        parameter (mwork = 280056)! 44088) !30000)

        parameter (mbc = 1)
        parameter (meqn = 4)
        parameter (mwaves = 2)
        parameter (maux = 7)
!       # NOTE: if maux>0 you must declare aux properly below!
!

        dimension q(1-mbc:maxmx+mbc, meqn)

!        dimension  aux(1)   !# dummy variable since no aux arrays used
        dimension  aux(1-mbc:maxmx+mbc, maux)

        dimension work(mwork)
        dimension mthlim(mwaves)

           call claw1ez(maxmx,meqn,mwaves,mbc,maux,mwork,mthlim, q,work,aux)

        print*, 'stop'

        stop
        end
```



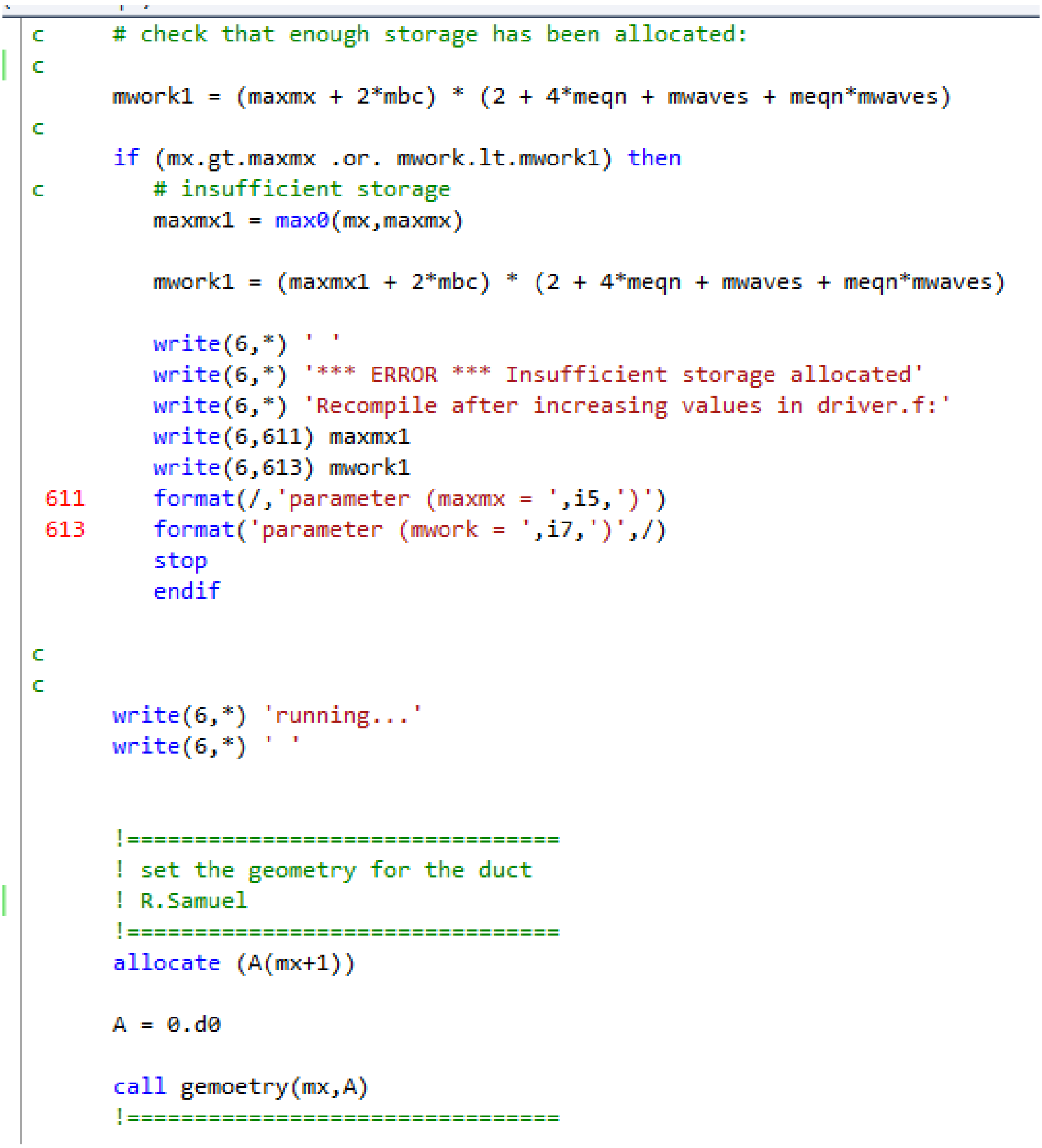
```
c     ==================================================================
      subroutine claw1ez(maxmx,meqn,mwaves,mbc,maux,mwork,mthlim,
     &                   q,work,aux)
c     ==================================================================
c
c     An easy-to-use clawpack driver routine for simple applications
c     Documentation is available at
c                 http://www.amath.washington.edu/~claw/doc.html
c
c     Author: Randall J. LeVeque
c     Version of August, 2002 --  CLAWPACK Version 4.1
c
      use Interfaces_

      !=============================
      ! R.Samuel
      !=============================
      use area_info
      !=============================


      implicit double precision (a-h,o-z)
      external rp1,src1,b4step1

      dimension q(1-mbc:maxmx+mbc, meqn)
      dimension aux(1-mbc:maxmx+mbc, maux)
      dimension work(mwork)
      dimension mthlim(mwaves)
c
      dimension method(7),dtv(5),cflv(4),nv(2),mthbc(2)
      dimension tout(100)
      logical outt0

c
      open(55,file='claw1ez - Riemann.data',status='old',
     & form='formatted')
      open(10,file='fort.info',status='unknown',form='formatted')
      open(11,file='fort.nplot',status='unknown',form='formatted')
```



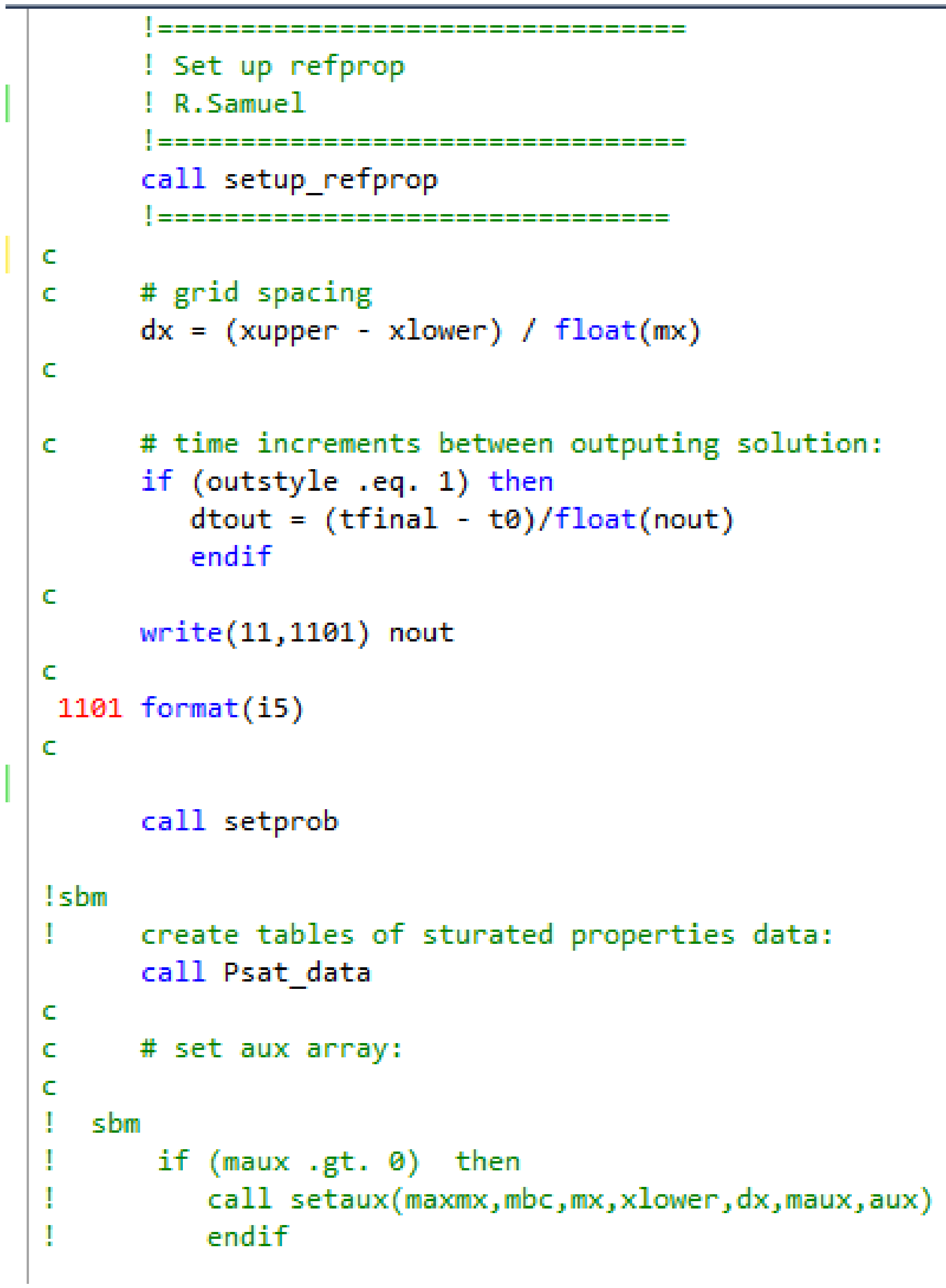
```
!sbm  .. Opening a new empty file for data output:
      open(unit=70,file='claw_rez.csv',status='replace')
      close(70)
c     # Read the input in standard form from claw1ez.data:
c     # For a description of input parameters see the documentation at
c                 http://www.amath.washington.edu/~claw
c     Number of grid cells:
      read(55,*) mx
c     i/o variables
      read(55,*) nout
      read(55,*) outstyle
      if (outstyle.eq.1) then
          read(55,*) tfinal
          nstepout = 1
        elseif (outstyle.eq.2) then
          read(55,*) (tout(i), i=1,nout)
          nstepout = 1
        elseif (outstyle.eq.3) then
          read(55,*) nstepout, nstop
        nout = nstop
        endif
c     timestepping variables
      read(55,*) dtv(1)
      read(55,*) dtv(2)
      read(55,*) cflv(1)
      read(55,*) cflv(2)
      read(55,*) nv(1)
c     # input parameters for clawpack routines
      read(55,*) method(1)
      read(55,*) method(2)
      read(55,*) method(3)
      read(55,*) method(4)
      read(55,*) method(5)
      read(55,*) method(6)
      read(55,*) method(7)
      read(55,*) meqn1
      read(55,*) mwaves1
      read(55,*) (mthlim(mw), mw=1,mwaves)
```



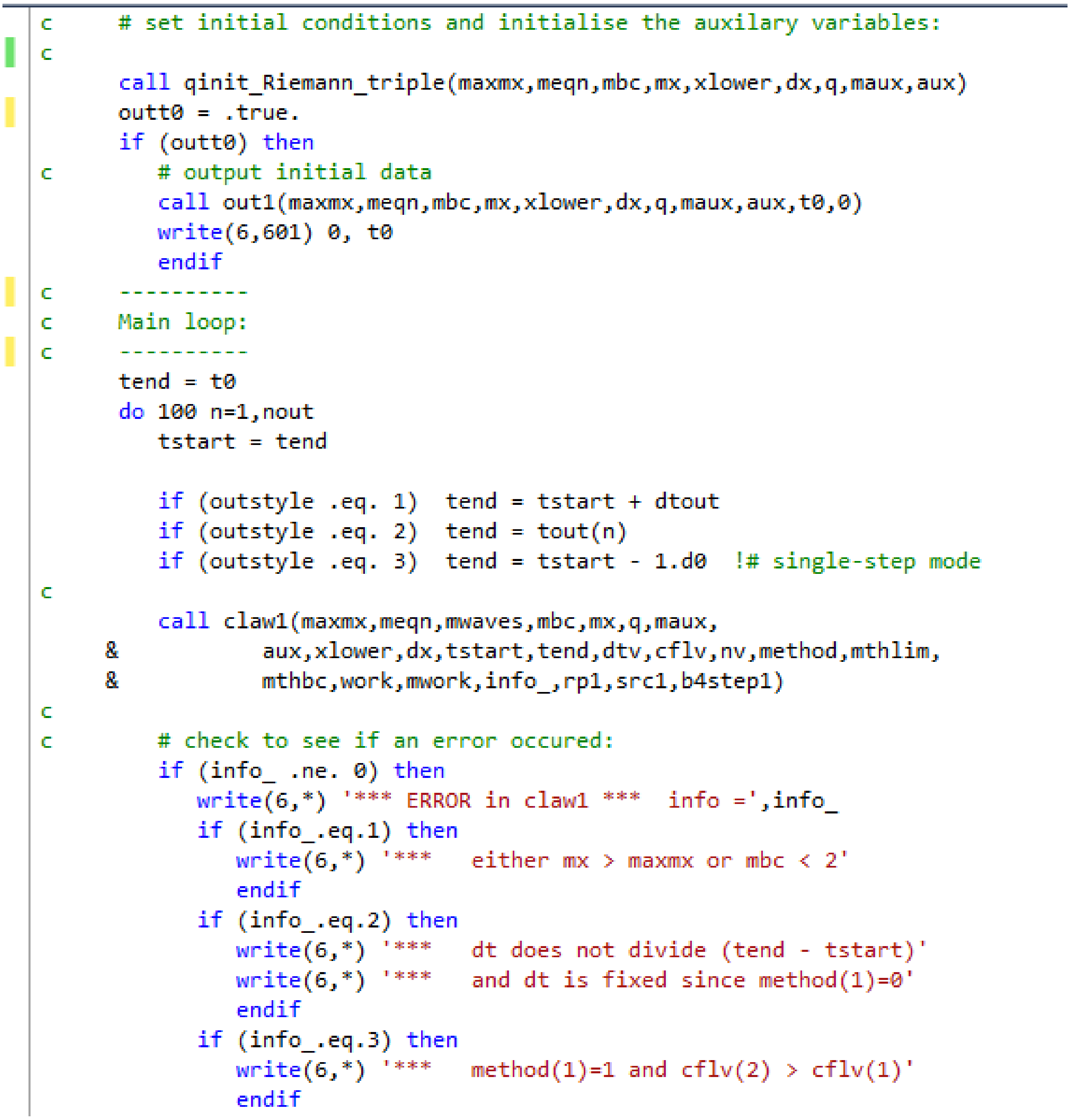
```
c     # physical domain:
      read(55,*) t0
      read(55,*) xlower
      read(55,*) xupper
c
c     # boundary conditions:
      read(55,*) mbc1
      read(55,*) mthbc(1)
      read(55,*) mthbc(2)

      if (method(7) .ne. maux) then
         write(6,*) '*** ERROR ***  maux set wrong in input or driver'
         stop
         endif

      if (meqn1 .ne. meqn) then
         write(6,*) '*** ERROR ***  meqn set wrong in input or driver'
         stop
         endif
      if (mwaves1 .ne. mwaves) then
         write(6,*) '*** ERROR ***  mwaves set wrong in input or driver'
         stop
         endif
      if (mbc1 .ne. mbc) then
         write(6,*) '*** ERROR ***  mbc set wrong in input or driver'
         stop
         endif
c
      if ((mthbc(1).eq.2 .and. mthbc(2).ne.2) .or.
     &    (mthbc(2).eq.2 .and. mthbc(1).ne.2)) then
         write(6,*) '*** ERROR ***  periodic boundary conditions'
         write(6,*) ' require mthbc(1) and mthbc(2) BOTH be set to 2'
         stop
         endif
```



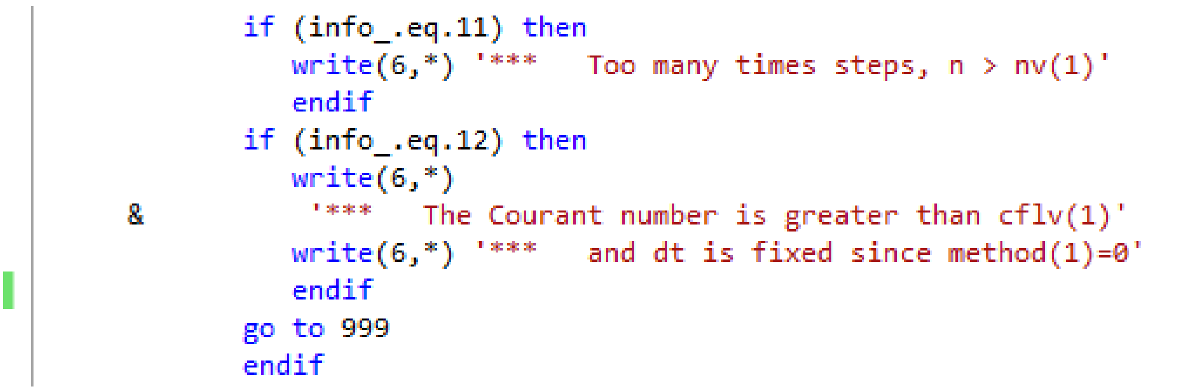
```
c     # check that enough storage has been allocated:
c
      mwork1 = (maxmx + 2*mbc) * (2 + 4*meqn + mwaves + meqn*mwaves)
c
      if (mx.gt.maxmx .or. mwork.lt.mwork1) then
c        # insufficient storage
         maxmx1 = max0(mx,maxmx)

         mwork1 = (maxmx1 + 2*mbc) * (2 + 4*meqn + mwaves + meqn*mwaves)

         write(6,*) ' '
         write(6,*) '*** ERROR *** Insufficient storage allocated'
         write(6,*) 'Recompile after increasing values in driver.f:'
         write(6,611) maxmx1
         write(6,613) mwork1
 611     format(/,'parameter (maxmx = ',i5,')')
 613     format('parameter (mwork = ',i7,')',/)
         stop
         endif

c
c
      write(6,*) 'running...'
      write(6,*) ' '


      !=================================
      ! set the geometry for the duct
      ! R.Samuel
      !=================================
      allocate (A(mx+1))

      A = 0.d0

      call gemoetry(mx,A)
      !=================================
```



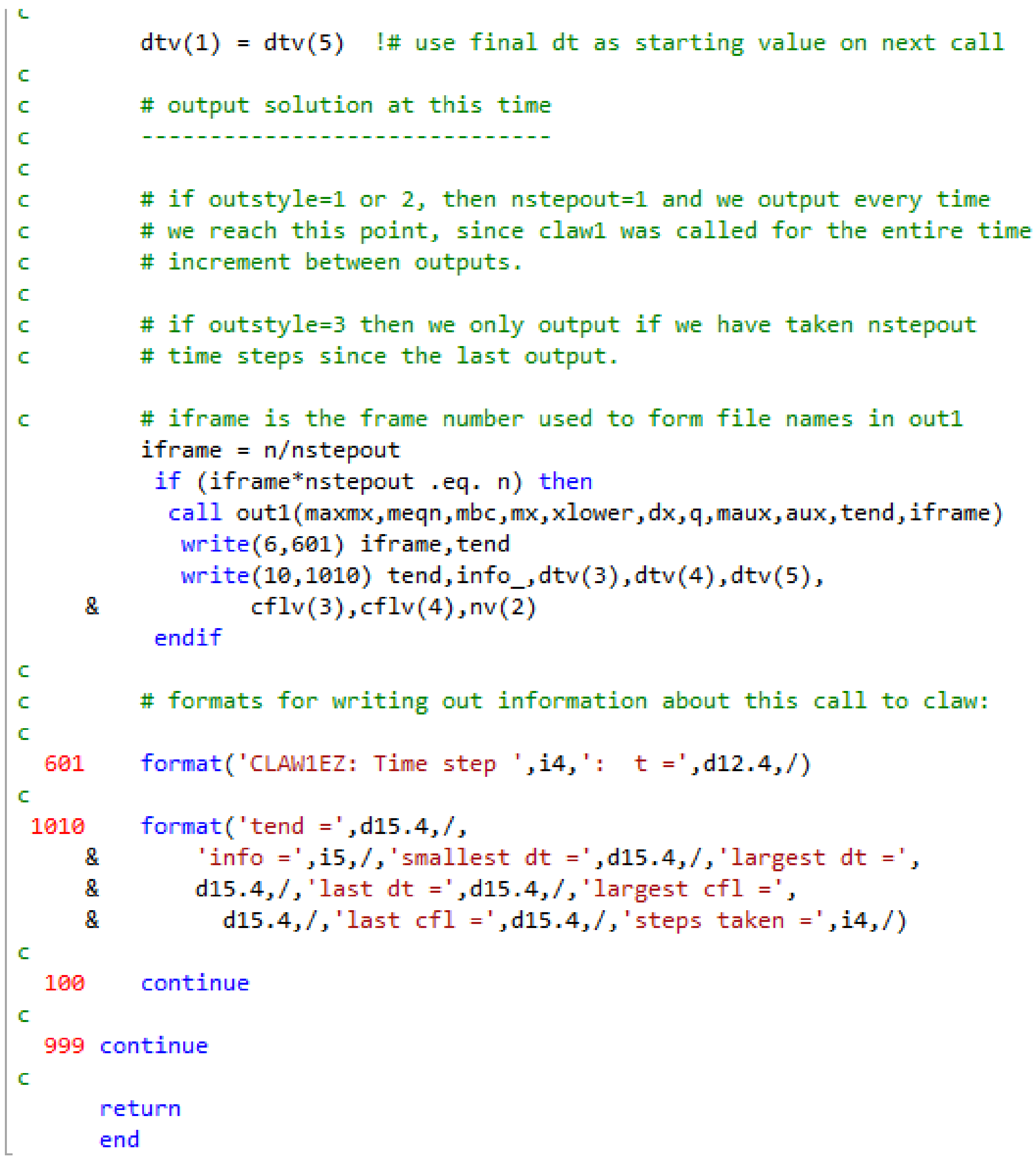
```
      !================================
      ! Set up refprop
      ! R.Samuel
      !================================
      call setup_refprop
      !===============================
c
c     # grid spacing
      dx = (xupper - xlower) / float(mx)
c

c     # time increments between outputing solution:
      if (outstyle .eq. 1) then
         dtout = (tfinal - t0)/float(nout)
         endif
c
      write(11,1101) nout
c
 1101 format(i5)
c

      call setprob

!sbm
!     create tables of sturated properties data:
      call Psat_data
c
c     # set aux array:
c
!  sbm
!      if (maux .gt. 0)  then
!         call setaux(maxmx,mbc,mx,xlower,dx,maux,aux)
!         endif
```



```
c     # set initial conditions and initialise the auxilary variables:
c
      call qinit_Riemann_triple(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux)
      outt0 = .true.
      if (outt0) then
c        # output initial data
         call out1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,t0,0)
         write(6,601) 0, t0
         endif
c     ----------
c     Main loop:
c     ----------
      tend = t0
      do 100 n=1,nout
         tstart = tend

         if (outstyle .eq. 1)  tend = tstart + dtout
         if (outstyle .eq. 2)  tend = tout(n)
         if (outstyle .eq. 3)  tend = tstart - 1.d0  !# single-step mode
c
         call claw1(maxmx,meqn,mwaves,mbc,mx,q,maux,
     &           aux,xlower,dx,tstart,tend,dtv,cflv,nv,method,mthlim,
     &           mthbc,work,mwork,info_,rp1,src1,b4step1)
c
c        # check to see if an error occured:
         if (info_ .ne. 0) then
            write(6,*) '*** ERROR in claw1 ***  info =',info_
            if (info_.eq.1) then
               write(6,*) '***   either mx > maxmx or mbc < 2'
               endif
            if (info_.eq.2) then
               write(6,*) '***   dt does not divide (tend - tstart)'
               write(6,*) '***   and dt is fixed since method(1)=0'
               endif
            if (info_.eq.3) then
               write(6,*) '***   method(1)=1 and cflv(2) > cflv(1)'
               endif
```



```
            if (info_.eq.11) then
               write(6,*) '***   Too many times steps, n > nv(1)'
               endif
            if (info_.eq.12) then
               write(6,*)
     &          '***   The Courant number is greater than cflv(1)'
               write(6,*) '***   and dt is fixed since method(1)=0'
               endif
            go to 999
            endif
```



```
c
         dtv(1) = dtv(5)  !# use final dt as starting value on next call
c
c        # output solution at this time
c        ------------------------------
c
c        # if outstyle=1 or 2, then nstepout=1 and we output every time
c        # we reach this point, since claw1 was called for the entire time
c        # increment between outputs.
c
c        # if outstyle=3 then we only output if we have taken nstepout
c        # time steps since the last output.

c        # iframe is the frame number used to form file names in out1
         iframe = n/nstepout
          if (iframe*nstepout .eq. n) then
           call out1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,tend,iframe)
            write(6,601) iframe,tend
            write(10,1010) tend,info_,dtv(3),dtv(4),dtv(5),
     &           cflv(3),cflv(4),nv(2)
          endif
c
c        # formats for writing out information about this call to claw:
c
  601    format('CLAW1EZ: Time step ',i4,':  t =',d12.4,/)
c
 1010    format('tend =',d15.4,/,
     &       'info =',i5,/,'smallest dt =',d15.4,/,'largest dt =',
     &       d15.4,/,'last dt =',d15.4,/,'largest cfl =',
     &         d15.4,/,'last cfl =',d15.4,/,'steps taken =',i4,/)
c
  100    continue
c
  999 continue
c
      return
      end
```

```
subroutine gemoetry(mx, area)
    use constants_module
    implicit none
    integer,intent(in) :: mx                  ! number of the grids within the physical
domain
    real(kind=8),intent(inout) :: area(mx+1) ! number of the surfaces of the grids

    integer :: i                              ! local variables

    !do i = 1,mx+1
    !    if (i <= 20) then
    !        area(i) = 1.0d0
    !    else if (i > 20 .and. i <= 40) then
    !        area(i) = 1.0d0+1.d0*3.0d0*tanh(float(i-20)/10.0d0)
    !    else if (i > 40 .and. i <= 60) then
    !        area(i) = area(40)+area(40)*5.25d0*tanh(float(i-40)/10.0d0)
    !    else if (i > 60) then
    !        area(i) = area(60)+area(60)*5.25d0*tanh(float(i-60)/50.0d0)
    !    end if
    !
    !end do


    do i = 1, mx+1
        if (i <= 450) then
            area(i) = 21.7d0
        else if (i > 450) then
            area(i) = area(450)-area(450)*tanh(float(i-450)/23.5d0)
        end if
    end do

    area = 1.d0

    area_ht = 3.1415926*id*length
    volume_ht = 3.1415926*(id/2.d0)*(id/2.d0)*length
    area_ht_sp = area_ht/volume_ht

end subroutine




!================================================
! this subroutine is used to set up REFPROP
! 1. reference state
! 2. fluids
! 3. number of component
! 4. file path
!================================================

subroutine setup_refprop
    !use shared_const
    !use shared_var
    use constants_module

    implicit none
    !character(len=255), dimension(20) :: component
    !character(len=255), dimension(20) :: hf
    !character(len=255) :: hfmix,hrf
    !character(len=255) :: hcode,hcite,herr
    !integer :: iflag
    !integer :: icomp
    !integer :: nc
    !integer :: ierr

    !======================
    ! Define the system
    !======================
    component(1) = 'CO2.fld'
    !======================


    ! header files
    !include 'var.fi'
    !
    !setup the path of mixture and fluid file
    !

    call SETPATH('C:\Users\Revelation\Desktop\REFPROP_lib') !D:\Program Files
(x86)\REFPROP

    !
    !initialise the mixture with default reference state

    !iflag = 1
    !call GERG04(nc,iflag,ierr,herr)

    !htype = 'EOS'
    !hmix = 'HMX'
    !hcomp = 'NBS'
    !
    !call SETMOD(nc,htype,hmix,hcomp,ierr,herr)

    hf = component
    hfmix = 'hmx.bnc'
    hrf='DEF'

    call SETUP(nc,hf,hfmix,hrf,ierr,herr)
    if(ierr /= 0) print*, herr
    !call PREOS(2)                            ! turns on if full PR-EOS is to be used

    !
    !get the EoS and mixing rule used
    !
    !icomp = 2
    !htype = 'EOS'
    !call GETMOD(icomp,htype,hcode,hcite)   ! check the equation of state of each
component with icomp being the component number

    !
    !return successful information
    !
    if (ierr == 0) print*, 'REFPROP initialisation successful'

end subroutine setup_refprop


      Subroutine setprob
      Use PPL_interface

      implicit double precision (a-h,o-z)
!      double precision mw_liq

      common /vapour/ gam_vap,po_vap,cv_vap,ho_vap,ro_vap,so_vap

      common /pipe_diam/ D_p,pipeThickness
      common /HRM/ index_HRM, theta
      common /TOLS/ tol_nms,x_tol


      ire = MAIN_INTERFACE_INIT(ipure,kind_eos)
      call main_interface_init_set_ref_only_solid(ipure)

!   Diameters of the pipe and the rupture/orifice (m):
!     D_p = 0.233   ! DUT
      D_p = 0.125   ! COOLTRANS
```

```
!   Parameters of the HRM model:
      index_HRM = 0      ! 1 - HRM, 0 - HEM
      theta = 1.d-6;     !   relaxation time (s)

      tol_nms = 1.d-9
      x_tol = 1.d-8

!     Parameters of the phases for CO2 described using stiff EoS
!    (Lund, Flåtten, Tollak Munkejord - 2011 - Depressurization of
!     carbon dioxide in pipelines models and methods)
!

!  Ideal-gas parameters for the vapour phase.
!   Using this set of parameters gives better accuracy for vapour phase than
!   the stiff gas constants. However, the equilibrium condition  g_vap = g_liq
!   at the saturation line is violated.

!  h_vap and s_vap adjusted to the saturated properties at 50 bar (21 Apr 2016)

      gam_vap= 1.28;
      po_vap = 73.d5*(13.21/12.25-1.) !0.d0;
      cv_vap = 8314./44./(gam_vap-1.);  !Cv = R/(gam-1)
      ho_vap = 1.d5 +  734533. - 248736. -35.d3 ! <- set larger than -5.077d5, to get
better
                    !    agreement for the latent heat of evaporation
      ro_vap = 13.21 !159.9;
      so_vap = 1750. - 4269.5 - 883.1;


      return
      end

! ==============================================================
  Subroutine psat_data
! ==============================================================
!     Calculate and tabulate the saturated properties
!
! ==============================================================
!     R.Samuel- 21 April 2016
! ==============================================================

        Use Properties_tables
        Use Triple_Point_Smoothing
        Use triple_point
        Use PPL_interface

        implicit double precision (a-h,o-z)

        !============================
        ! R.Samuel
        !============================
        double precision :: z(20),xliq(20),xvap(20)
        integer :: kph, ierr
        character (len=255) :: herr

        !============================

        common /vapour/ gam_vap,po_vap,cv_vap,ho_vap,ro_vap,so_vap

! ==============================================================
!
```

```
!    Adjust the reference value for the energy/ enthalpy
!
         T_ref = 260.
         p_ref  = interface_compute_Psat_EOS(icomp, T_ref)
         r_ref  = den_liq_PPL(p_ref,T_ref)
         e_ref_PPL = e_liq_PPL(r_ref,T_ref)

         p_bar = p_ref/1.d5
         r_x   = rho_liq(p_bar)
         h_x   =   h_liq(p_bar)
         e_x   = h_x - p_ref/r_x

         E_ref = e_x - e_ref_PPL             !  E_ref = E_Anvar - E_PPL

         !print*,' E_PPL = ',e_ref_PPL
         !print*,' E_x   = ',e_x
         !print*,' E_ref = ',E_ref
         !read*

!----------------------------------------------------------
!
!    Adjust the reference value for the vapour entropy
!
         !==================================================
         !T_ref = 217.
         !p_ref = interface_compute_Psat_EOS(icomp, T_ref)
         !r_ref = den_vap(p_ref,T_ref)
         !
         !S_vap_ref = s_vap(r_ref,T_ref)
         !S_sat_ref = s_v(p_ref/1.d5)
         !
         !deltaS = S_sat_ref - S_vap_ref          !  deltaS = S_sat_ref - S_vap_ref
         !so_vap_ = so_vap - deltaS

         !print*
         !print*,' S_vap = ',S_vap_ref
         !print*,' S_sat = ',S_sat_ref
         !print*,' S_ref = ',so_vap,so_vap_
         !
         !so_vap = so_vap_
         !S_vap_ref_ = s_vap(r_ref,T_ref)
         !
         !deltaS = S_sat_ref - S_vap_ref_         !  deltaS = S_sat_ref - S_vap_ref
         !====================================================
         read*

!===========================================================
! adjust the liquid phase entropy (reference entropy)
!===========================================================
         !r_l_ref = den_liq_PPL(p_ref,T_ref)
         !s_l_ref = s_liq_PPL(r_l_ref,T_ref)
         !s_l_sat = s_c(p_ref/(1.d5));
         !
         !S_ref = s_l_sat-s_l_ref

!===========================================================
! adjust the vapour phase enthalpy (reference enthalpy)
!===========================================================
         r_v_ref = den_vap(p_ref,T_ref)
         e_v_ref = e_vap(r_v_ref,T_ref)
         r_v_sat = d_v(p_bar)
         e_v_sat = h_v(p_ref/(1.d5))- p_ref/r_v_sat
         E_vap_ref = e_v_sat-e_v_ref

!===========================================================
! adjust the vapour phase entropy (reference entropy)
!===========================================================
         s_v_ref = s_vap(r_v_ref,T_ref)
         s_v_sat = s_v(p_ref/(1.d5))

         S_vap_ref = s_v_sat - s_v_ref

!===========================================================
! adjust the 'refprop' reference state for the liquid phase
!===========================================================
        p_ref = p_ref
        T_ref = T_ref

        p_bar = p_ref/1.d5
        r_x   = rho_liq(p_bar)
        h_x   =   h_liq(p_bar)
        s_x = s_c(p_ref/(1.d5))
        e_x   = h_x - p_ref/r_x

        z = 0.d0
        z(1) = 1.d0
        p = p_ref/1000.d0
        kph = 1
        call SATP (p,z,kph,t,rhol,rhov,xliq,xvap,ierr,herr)

        T = T
        rho = rhol
        call THERM (T,rho,z,p,u,h,s,cv,cp,w,hjt)
        s_refprop = s/(mw_liq/1000.d0)

        e_refprop = u/(mw_liq/1000.d0)
        E_refprop_ref = e_x - e_refprop
        S_refprop_ref = s_x - s_refprop

!===========================================================

!----------------------------------------------------------
!    Define coefficients for smoothing functions that
!    will be used to calculate saturated properties
!    near the critical point

         p_tr = 73.3 !d2   ! bar
         dp_tr = 0.001 ! 0.1d-3  !0.02 !0.2 !0.8   ! bar  <-  the pressure interval
for smoothing around p_tr

         !=====================================
         ! R. Samuel  critical point info
         !=====================================
         dp_smoothing = dp_tr*1.d5 !pa

         p_3p = 73.31d0
         T_3p = T_subl (p_3p)
         rhov_3p = d_v(p_3p)
         rhol_3p = rho_liq(p_3p)
         rhos_3p = rho_sol(p_3p)
         hv_3p = h_v(p_3p)
         hl_3p = h_liq(p_3p)
         hs_3p = h_sol(p_3p)
         ev_3p = hv_3p - p_3p*1.d5/rhov_3p
         el_3p = hl_3p - p_3p*1.d5/rhol_3p
         es_3p = hs_3p - p_3p*1.d5/rhos_3p
         sv_3p = s_v(p_3p)
         sl_3p = s_liq(p_3p)
         ss_3p = s_sol(p_3p)

         Csv_3p = a_v(p_3p)
         Csl_3p = a_sol(p_3p)
         Css_3p = a_liq(p_3p)

         mul_3p = visc_liq(rhol_3p,T_3p)
         mus_3p = visc_sol(rhos_3p,T_3p)
         !===================================

         !===================================
         ! smoothing around triple point
         !===================================



         p_tr_s = p_tr - dp_tr/2.  ! braketing point to the left of p_tr
         p_tr_l = p_tr + dp_tr/2.  ! braketing point to the left of p_tr

         p_bar_s = p_tr_s
         p_bar_l = p_tr_l

         T_tr_l = T_satur(p_bar_l)
         T_tr_s = T_subl (p_bar_s)
         dT_tr  = T_tr_l - T_tr_s

         rho_tr_v = d_v(p_bar_l)
         rho_tr_l = rho_liq(p_bar_l)
         rho_tr_s = rho_sol(p_bar_s)
         drho_tr = rho_tr_l - rho_tr_s

         h_tr_v = h_v(p_bar_l)
         h_tr_l = h_liq(p_bar_l)
         h_tr_s = h_sol(p_bar_s)
         dh_tr  = h_tr_l - h_tr_s

         e_tr_v = h_tr_v - p_tr_l*1.d5/rho_tr_v
         e_tr_l = h_tr_l - p_tr_l*1.d5/rho_tr_l
         e_tr_s = h_tr_s - p_tr_s*1.d5/rho_tr_s
         de_tr  = e_tr_l - e_tr_s

         s_tr_l = s_liq(p_bar_l)
         s_tr_s = s_sol(p_bar_s)
         ds_tr  = s_tr_l - s_tr_s

         Cs_tr_s = a_sol(p_bar_s)
         Cs_tr_l = a_liq(p_bar_l)
         dCs_tr  = Cs_tr_l - Cs_tr_s

         mu_tr_s = visc_sol(rho_tr_s,T_tr_s)
         mu_tr_l = visc_liq(rho_tr_l,T_tr_l)
         dmu_tr  = mu_tr_l - mu_tr_s


! =============================================================
!
!    Specify the range of pressures to cover:
        p_min = 50.d5  !Pa
        p_max = 200.d5  !Pa  <- set around the critical pressure for CO2 (73 bar)

        dp = (p_max-p_min-1.d5)/(k_max - 1)

        p=p_min

        p_tr_flag = 0.

        print*, ' k         p           T          rs_v         rs_l       hs_v(k)
hs_l(k)'
        print*


        do k=1,k_max

                  p_bar = p*1.d-5

            P_s(k) = p
            T = T_sat(p)
            T_s(k) = T

            rs_v(k) = d_v(p_bar)
            rs_l(k) = d_c(p_bar)
            ss_v(k) = s_v(p_bar)
            ss_l(k) = s_c(p_bar)
            hs_v(k) = h_v(p_bar)
            hs_l(k) = h_c(p_bar)
            es_v(k) = hs_v(k) - p/rs_v(k)
            es_l(k) = hs_l(k) - p/rs_l(k)

            P_s(k) = p;   T_s(k) = T
            Ps_(k) = p;   Ts_(k) = T

!------------------------------------------------
!  testing...

            den_gas = rs_v(k)
            e_gas = e_vap_(den_gas,p)
            T_gas = T_vap(den_gas,e_gas)
            s_gas = s_vap(den_gas,T_gas)
!           h_gas = e_gas + p/d_gas
            e_g2 = e_vap(den_gas,T)
            den_g2 = den_vap(p,T)

            p_1 = p_vap(den_gas,e_gas)
            p_2 = p_vap_(den_gas,T)
            T_1 = T_vap_(den_gas,s_gas)
            T_2 = T_vap_ps(p,s_gas)

!------------------------------------------------


! End points extrapolated to p->0 and T->0,
            if (k.eq.1) then
              px=p*1.d-5;  Tx=T*1.d-5;

              px = 0.d0;   Tx = 0.d0;
              P_s(k) = px;  Ps_(k) = px
              T_s(k) = Tx;  Ts_(k) = Tx
              rs_v(k) = rs_v(k)*1.d-5;
            endif

            j=2*k_max+1-k

!sbm 27 Sep 2013
!           P_s(j) = p;       T_s(j) = T
            P_s(j) = P_s(k);  T_s(j) = T_s(k)

            rs_(k) = rs_v(k); rs_(j)=rs_l(k)
            ss_(k) = ss_v(k); ss_(j)=ss_l(k)
            hs_(k) = hs_v(k); hs_(j)=hs_l(k)
            es_(k) = es_v(k); es_(j)=es_l(k)

            p = p + dp

            if (p_tr_flag.eq.1) then
              p=p_tr_l*1.d5
              p_tr_flag = 0.
            else
            if ( ((p-dp).lt.p_tr_s*1.d5) .and. (p.gt.(p_tr_s*1.d5+1.d-5)) ) then
              p=p_tr_s*1.d5
              p_tr_flag = 1.
            endif
            endif

            if (p.gt.p_max) p=p_max


            write(*,'(i3,4F11.1,2F13.1)')
k,P_s(k),T_s(k),rs_v(k),rs_l(k),es_v(k),es_l(k)


        enddo


        print*,' ----- Psat_data completed ------------------------------------------'
        read*

  End Subroutine psat_data


!===================================================================================
       subroutine qinit_Riemann_triple(maxmx,meqn,mbc,mx,xlower,dx,q,
     &           maux,aux)
!===================================================================================
c
c     # Set initial conditions for q.
c     # Riemann problem
c
      USE PPL_interface

      implicit double precision (a-h,o-z)
      dimension q(1-mbc:maxmx+mbc, meqn)
      dimension aux(1-mbc:maxmx+mbc, maux)

      !======================
      !Written by: R.Samuel
      !======================
      double precision :: z(20),xvap(20),xliq(20)
      double precision :: rhobulk, rholiq, rhovap
      double precision :: quality,u_internal,h,s,cv,cp,w,T,p
      integer :: ierr
      character (len=255) :: herr
      !======================


      common /problem/ p_out, p_in, T_in, rho_in, x_in, Cs_in

!===================================================================================
!     Riemann problem states:
!       Left state  - liquid-vapour at p > 5.18 bar and T = Tsat(p);
!       Right state - solid-vapour at  p < 5.18 bar and T = Tsub(p);
!       Interface (few cells) - at p and Tsat(p).
!
!     u = 0 in the entire domain.
!===================================================================================

      x_o = 400.d0   !xlower + dx*mx/2.   ! X coordinate of the phase transition
boundary

!===================================================================================
!sbm: right state:
!===================================================================================

!===================================================================================
! initialise the well with desired profile
!===================================================================================
! R.Samuel
!===================================================================================


      do 150 i=1-mbc,mx+mbc

         xcell = xlower + (i-0.5d0)*dx



        if (xcell <x_o) then

            !==================================================
            z = 0.d0
            z(1) = 1.d0
            temp = (284.15d0-277.15d0)/(400.d0)*xcell+277.15d0
            pres = ((46.191683-38.d0)/(400.d0)*xcell+38.d0)*100000.d0
            t = temp
            p = pres/1000.d0
            call TPRHO (T,p,z,-2,0,rhobulk,ierr,herr)
            call THERM (t,rhobulk,z,p,u_internal,h,s,cv,cp,w,hjt)
            !==================================================
            rs_l = rhobulk*mw_liq
            es_l = u_internal/(mw_liq/1000.d0)+E_refprop_ref
            s_l_ = s/(mw_liq/1000.d0)+S_refprop_ref
            Cs_l = w
            !==================================================
            rho = rs_l
            u = 0.d0
            e = es_l
            x_left = 1.0d0
            Cs = Cs_l
            Tliq = temp
            !===================================================

            q(i,1) = rho;
            q(i,2) = q(i,1)*u;
            q(i,3) = q(i,1)*(e+0.5d0*u*u);
            q(i,4) = x_left*rho;

            aux(i,1) = pres;
            aux(i,2) = Temp;
            aux(i,3) = x_left;
            aux(i,4) = Cs;
            aux(i,5) = Tliq;

        elseif (xcell >= (x_o)) then
            !===================================================
            z = 0.d0
            z(1) = 1.d0
            temp_ = (353.15-284.15)/(400.d0)*(xcell-400.0)+284.15
            pres_ = ((172.d0-46.191683)/(400.d0)*(xcell-400.0)+
     &   46.191683)*100000.d0
            t = temp_
            p = pres_/1000.d0
            call TPRHO (T,p,z,-1,0,rhobulk,ierr,herr)
            call THERM (t,rhobulk,z,p,u_internal,h,s,cv,cp,w,hjt)
            !===================================================
            rs_l_ = rhobulk*mw_liq
            es_l_ = u_internal/(mw_liq/1000.d0)+E_refprop_ref
            s_l_ = s/(mw_liq/1000.d0)+S_refprop_ref
            Cs_l_ = w

            !===================================================
            rho_ = rs_l_
            u_ = 0.d0
            e_ = es_l_
            x_right = 0.0d0
            Cs_ = Cs_l_
            Tliq_ = temp_
            !====================================================

            q(i,1) = rho_;
            q(i,2) = q(i,1)*u_;
            q(i,3) = q(i,1)*(e_+0.5d0*u_*u_);
            q(i,4) = x_right*rho_;

            aux(i,1) = pres_
            aux(i,2) = Temp_
            aux(i,3) = x_right
            aux(i,4) = Cs_
            aux(i,5) = Tliq_

        end if


  150 continue
c
      p_out = pres_    ! dummy > 0 (not used in the Riemann problem calculations)



!==========================================================
! if 'outflow' boundary condition has been selected
! ghost cell (i=mx+mbc) should be initialised with
! ambient condition p = p_out
!
! R.Samuel
```

```
!==========================================================

        if(i.gt.mx) then
            p_out = 172.d5

            u_   = u;

            !=======================
            !R.Samuel
            !=======================
            pres_= p_out
            Temp_ = T_sat(pres_) !T_out !273. !T_sat(pres_)
            Tliq_ = Temp_

            p_bar =   pres_/1.d5;
            rs_v =  d_v(p_bar);
            rs_l =  d_c(p_bar);
            hs_v =  h_v(p_bar);
            hs_l =  h_c(p_bar);

            s_v_ =  s_v(p_bar);
            s_l_ =  s_c(p_bar);

            es_v = hs_v - pres_/rs_v;
            es_l = hs_l - pres_/rs_l;

            Cs_v  =   a_v(p_bar);
            Cs_l  =   a_c(p_bar);


            !x_out = (s_left - s_l_)/(s_v_ - s_l_)        ! initialise the ghost cell
vapour quality isentropically
            x_ = 1.0                                     !x_out
!1.!-1.d-5!      sbm: 15-Sept-2013

            if (x_.lt.1.d-6) then
            !   special case: pure liquid:
            !rs_l = den_liq_PPL(pres_,Temp_)
            !es_l =   e_liq_PPL(rs_l,Temp_)  + E_ref ! E_Anvar = E_PPL + E_ref
            !s_l_ =   s_liq_PPL(rs_l,Temp_)
            !Cs_l =  Cs_liq_PPL(rs_l,Temp_)

            !================================
            ! using refprop to calculate the density
            ! R.Samuel
            !================================
                z = 0.d0
                z(1) = 1.d0
                T = Temp_
                p = pres_/1000.d0
                call TPFLSH (t,p,z,rhobulk,rholiq,rhovap,xliq,xvap,
     &                       quality,u_internal,h,s,cv,cp,w,ierr,herr)
            !================================
                rs_l = rhobulk*mw_liq
                es_l = u_internal/(mw_liq/1000.d0)+E_refprop_ref
                s_l_ = s/(mw_liq/1000.d0)+S_refprop_ref
                Cs_l = w

                !============================
                ! R.Samuel
                !============================
                !set vapour mass fraction to zero
                x_ = 0.d0
                T_liq_ = Temp_

                rho_=rs_l
                e_ =es_l
                vf = 1.d0/rs_l
                Cs = Cs_l

            else if (x_ > 1.d0) then
              !================================
              !R.Samuel
              !================================
                rho_ = den_vap(pres_,Temp_)
                e_ = e_vap(rho_, Temp_) + E_vap_ref
                Cs_ = Cs_vap(rho_,e_)

                x_ = 1.d0
                Tliq_ = Temp_

            else
                !=================================
                ! R.Samuel
                !=================================
                rho_=1./(x_/rs_v + (1.-x_)/rs_l)
                e_ = x_*es_v + (1.-x_)*es_l
                vf = vf_mix(x_,rs_v,rs_l)
                Cs_ = Cs_mix(vf,pres_,Temp_,rho_,rs_v,rs_l,Cs_v,Cs_l)
            endif

            q(i,1) = rho_;
            q(i,2) = q(i,1)*u_;
            q(i,3) = q(i,1)*(e_+0.5d0*u_*u_);
            q(i,4) = x_*rho_;

            aux(i,1) = pres_
            aux(i,2) = Temp_
            aux(i,3) = x_
            aux(i,4) = Cs_
            aux(i,5) = Tliq_
            aux(i,6) = s_mix_

            print*,' qinit: ',Cs_
            print*,'p,T,x = ',pres_,Temp_,x_

        endif
!===================================================
      return
      end

c
c     ==============================================================
      subroutine claw1(maxmx,meqn,mwaves,mbc,mx,q,
     &         maux,aux,xlower,dx,tstart,tend,dtv,cflv,nv,method,mthlim,
     &         mthbc,work,mwork,info,rp1,src1,b4step1)
c     ==============================================================
c
c  Solves a hyperbolic system of conservation laws in one space dimension
c  of the general form
c
c     capa * q_t + A q_x = psi
c
c  The "capacity function" capa(x) and source term psi are optional
```

```
c  (see below).
c
c  For a more complete description see the documentation at
c      http://www.amath.washington.edu/~claw
c
c  Sample driver programs and user-supplied subroutines are available.
c  See the the directories claw/clawpack/1d/example* for some examples, and
c  codes in claw/applications for more extensive examples.
c
c  --------------------------------------------------------
c
c  The user must supply the following subroutines:
c
c    bc1, rp1        subroutines specifying the boundary conditions and
c                    Riemann solver.
c                    These are described in greater detail below.
c
c    b4step1            The routine b4step1 is called each time step and
c                       can be supplied by the user in order to perform
c                       other operations that are necessary every time
c                       step.  For example, if the variables stored in
c                       the aux arrays are time-dependent then these
c                       values can be set.
c
c  In addition, if the equation contains source terms psi, then the user
c  must provide:
c
c    src1               subroutine that solves capa * q_t = psi
c                       over a single time step.
c
c  These routines must be declared EXTERNAL in the main program.
c  For description of the calling sequences, see below.
c
c  Dummy routines b4step1.f and src1.f are available in
c       claw/clawpack/1d/lib
c
c
c
c  Description of parameters...
c  ----------------------------
c
c
c    maxmx is the maximum number of interior grid points in x,
c          and is used in declarations of the array q.
c
c    meqn is the number of equations in the system of
c         conservation laws.
c
c    mwaves is the number of waves that result from the
c           solution of each Riemann problem.  Often mwaves = meqn but
c           for some problems these may be different.
c
c    mbc is the number of "ghost cells" that must be added on to each
c       side of the domain to handle boundary conditions.  The cells
c       actually in the physical domain are labelled from 1 to mx in x.
c       The arrays are dimensioned actually indexed from 1-mbc to mx+mbc.
c       For the methods currently implemented, mbc = 2 should be used.
c       If the user implements another method that has a larger stencil and
c       hence requires more ghost cells, a larger value of mbc could be used.
c       q is extended from the physical domain to the ghost cells by the
c       user-supplied routine bc1.
c
```

```
c    mx is the number of grid cells in the x-direction, in the
c       physical domain.  In addition there are mbc grid cells
c       along each edge of the grid that are used for boundary
c       conditions.
c       Must have mx .le. maxmx
c
c    q(1-mbc:maxmx+mbc, meqn)
c        On input:  initial data at time tstart.
c        On output: final solution at time tend.
c        q(i,m) = value of mth component in the i'th cell.
c        Values within the physical domain are in q(i,m)
c                for i = 1,2,...,mx
c        mbc extra cells on each end are needed for boundary conditions
c        as specified in the routine bc1.
c
c    aux(1-mbc:maxmx+mbc, maux)
c        Array of auxiliary variables that are used in specifying the problem.
c        If method(7) = 0 then there are no auxiliary variables and aux
c                         can be a dummy variable.
c        If method(7) = maux > 0 then there are maux auxiliary variables
c                         and aux must be dimensioned as above.
c
c        Capacity functions are one particular form of auxiliary variable.
c        These arise in some applications, e.g. variable coefficients in
c        advection or acoustics problems.
c        See Clawpack Note # 5 for examples.
c
c        If method(6) = 0 then there is no capacity function.
c        If method(6) = mcapa > 0  then there is a capacity function and
c            capa(i), the "capacity" of the i'th cell, is assumed to be
c            stored in aux(i,mcapa).
c            In this case we require method(7).ge.mcapa.
c
c    dx = grid spacing in x.
c         (for a computation in ax <= x <= bx,  set dx = (bx-ax)/mx.)
c
c    tstart = initial time.
c
c    tend = Desired final time (on input).
c              If tend<tstart, then claw1 returns after a single successful
c                 time step has been taken (single-step mode).
c              Otherwise, as many steps are taken as needed to reach tend,
c                 up to a maximum of nv(1).
c         = Actual time reached (on output).
c
c    dtv(1:5) = array of values related to the time step:
c               (Note: method(1)=1 indicates variable size time steps)
c         dtv(1) = value of dt to be used in all steps if method(1) = 0
c                = value of dt to use in first step if method(1) = 1
c         dtv(2) = unused if method(1) = 0.
c                = maximum dt allowed if method(1) = 1.
c         dtv(3) = smallest dt used (on output)
c         dtv(4) = largest dt used (on output)
c         dtv(5) = dt used in last step (on output)
c
c    cflv(1:4) = array of values related to Courant number:
c         cflv(1) = maximum Courant number to be allowed.  With variable
c                   time steps the step is repeated if the Courant
c                   number is larger than this value.  With fixed time
c                   steps the routine aborts.  Usually cflv(1)=1.0
c                   should work.
c         cflv(2) = unused if method(1) = 0.
```

```
c                 = desired Courant number if method(1) = 1.
c                   Should be somewhat less than cflv(1), e.g. 0.9
c         cflv(3) = largest Courant number observed (on output).
c         cflv(4) = Courant number in last step (on output).
c
c    nv(1:2) = array of values related to the number of time steps:
c         nv(1) = unused if method(1) = 0
c               = maximum number of time steps allowed if method(1) = 1
c         nv(2) = number of time steps taken (on output).
c
c    method(1:7) = array of values specifying the numerical method to use
c         method(1) = 0 if fixed size time steps are to be taken.
c                       In this case, dt = dtv(1) in all steps.
c                   = 1 if variable time steps are to be used.
c                       In this case, dt = dtv(1) in the first step and
c                       thereafter the value cflv(2) is used to choose the
c                       next time step based on the maximum wave speed seen
c                       in the previous step.  Note that since this value
c                       comes from the previous step, the Courant number will
c                       not in general be exactly equal to the desired value
c                       If the actual Courant number in the next step is
c                       greater than 1, then this step is redone with a
c                       smaller dt.
c
c         method(2) = 1 if Godunov's method is to be used, with no 2nd order
c                       corrections.
c                   = 2 if second order correction terms are to be added, with
c                       a flux limiter as specified by mthlim.
c
c         method(3)  is not used in one-dimension.
c
c         method(4) = 0 to suppress printing
c                   = 1 to print dt and Courant number every time step
c
c         method(5) = 0 if there is no source term psi.  In this case
c                       the subroutine src1 is never called so a dummy
c                       parameter can be given.
c                   = 1 if there is a source term.  In this case
c                       the subroutine src1 must be provided and a
c                       fractional step method is used.
c                       In each time step the following sequence is followed:
c                            call bc to extend data to ghost cells
c                            call step1 to advance hyperbolic eqn by dt
c                            call src1 to advance source terms by dt
c                   = 2 if there is a source term and Strang splitting is to
c                       be used instead of the Godunov splitting above.
c                       In each time step the following sequence is followed:
c                            call bc to extend data to ghost cells
c                            call src1 to advance source terms by dt/2
c                            call step1 to advance hyperbolic equation by dt
c                            call src1 to advance source terms by dt/2
c                       For most problems 1 is recommended rather than 2
c                       since it is less expensive and works essentially as
c                       well on most problems.

c
c         method(6) = 0 if there is no capacity function capa.
c                   = mcapa > 0 if there is a capacity function.  In this case
c                       aux(i,mcapa) is the capacity of the i'th cell and you
c                       must also specify method(7) .ge. mcapa and set aux.
c
c         method(7) = 0 if there is no aux array used.
```

```
c                   = maux > 0  if there are maux auxiliary variables.
c
c
c         The recommended choice of methods for most problems is
c            method(1) = 1,  method(2) = 2.
c
c    mthlim(1:mwaves) = array of values specifying the flux limiter to be used
c                     in each wave family mw.  Often the same value will be used
c                     for each value of mw, but in some cases it may be
c                     desirable to use different limiters.  For example,
c                     for the Euler equations the superbee limiter might be
c                     used for the contact discontinuity (mw=2) while another
c                     limiter is used for the nonlinear waves.  Several limiters
c                     are built in and others can be added by modifying the
c                     subroutine philim.
c
c        mthlim(mw) = 0 for no limiter
c                   = 1 for minmod
c                   = 2 for superbee
c                   = 3 for van Leer
c                   = 4 for monotonized centered
c
c
c    work(mwork) = double precision work array of length at least mwork
c
c    mwork = length of work array.  Must be at least
c               (maxmx + 2*mbc) * (2 + 4*meqn + mwaves + meqn*mwaves)
c            If mwork is too small then the program returns with info = 4
c            and prints the necessary value of mwork to unit 6.
c
c
c    info = output value yielding error information:
c         = 0 if normal return.
c         = 1 if mx.gt.maxmx   or  mbc.lt.2
c         = 2 if method(1)=0 and dt doesn't divide (tend - tstart).
c         = 3 if method(1)=1 and cflv(2) > cflv(1).
c         = 4 if mwork is too small.
c         = 11 if the code attempted to take too many time steps, n > nv(1).
c              This could only happen if method(1) = 1 (variable time steps).
c         = 12 if the method(1)=0 and the Courant number is greater than 1
c              in some time step.
c
c           Note: if info.ne.0, then tend is reset to the value of t actually
c           reached and q contains the value of the solution at this time.
c
c    User-supplied subroutines
c    -------------------------
c
c    bc1 = subroutine that specifies the boundary conditions.
c         This subroutine should extend the values of q from cells
c         1:mx to the mbc ghost cells along each edge of the domain.
c
c          The form of this subroutine is
c  -------------------------------------------------
c     subroutine bc1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,t,mthbc)
c     implicit double precision (a-h,o-z)
c     dimension   q(1-mbc:maxmx+mbc, meqn)
c     dimension aux(1-mbc:maxmx+mbc, *)
c     dimension mthbc(2)
c  -------------------------------------------------
c
c    The routine claw/clawpack/1d/lib/bc1.f can be used to specify
```

```
c    various standard boundary conditions.
c
c
c    rp1 = user-supplied subroutine that implements the Riemann solver
c
c          The form of this subroutine is
c  -------------------------------------------------
c     subroutine rp1(maxmx,meqn,mwaves,mbc,mx,ql,qr,auxl,auxr,wave,s,amdq,apdq)
c     implicit double precision (a-h,o-z)
c     dimension   ql(1-mbc:maxmx+mbc, meqn)
c     dimension   qr(1-mbc:maxmx+mbc, meqn)
c     dimension auxl(1-mbc:maxmx+mbc, *)
c     dimension auxr(1-mbc:maxmx+mbc, *)
c     dimension wave(1-mbc:maxmx+mbc, meqn, mwaves)
c     dimension    s(1-mbc:maxmx+mbc, mwaves)
c     dimension amdq(1-mbc:maxmx+mbc, meqn)
c     dimension apdq(1-mbc:maxmx+mbc, meqn)
c  -------------------------------------------------
c
c        On input, ql contains the state vector at the left edge of each cell
c                  qr contains the state vector at the right edge of each cell
c                auxl contains auxiliary values at the left edge of each cell
c                auxr contains auxiliary values at the right edge of each cell
c
c        Note that the i'th Riemann problem has left state qr(i-1,:)
c                                            and right state ql(i,:)
c        In the standard clawpack routines, this Riemann solver is
c        called with ql=qr=q along this slice.  More flexibility is allowed
c        in case the user wishes to implement another solution method
c        that requires left and rate states at each interface.

c        If method(7)=maux > 0 then the auxiliary variables along this slice
c        are passed in using auxl and auxr.  Again, in the standard routines
c        auxl=auxr=aux in the call to rp1.
c
c         On output,
c             wave(i,m,mw) is the m'th component of the jump across
c                             wave number mw in the ith Riemann problem.
c             s(i,mw) is the wave speed of wave number mw in the
c                             ith Riemann problem.
c             amdq(i,m) = m'th component of A^- Delta q,
c             apdq(i,m) = m'th component of A^+ Delta q,
c                    the decomposition of the flux difference
c                        f(qr(i-1)) - f(ql(i))
c                    into leftgoing and rightgoing parts respectively.
c
c           It is assumed that each wave consists of a jump discontinuity
c           propagating at a single speed, as results, for example, from a
c           Roe approximate Riemann solver.  An entropy fix can be included
c           into the specification of amdq and apdq.
c
c    src1 = subroutine for the source terms that solves the equation
c               capa * q_t = psi
c           over time dt.
c
c           If method(5)=0 then the equation does not contain a source
c           term and this routine is never called.  A dummy argument can
c           be used with many compilers, or provide a dummy subroutine that
c           does nothing (such a subroutine can be found in
c           claw/clawpack/1d/lib/src1.f)
c
c          The form of this subroutine is
```

```
c  -------------------------------------------------
c     subroutine src1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,t,dt)
c     implicit double precision (a-h,o-z)
c     dimension   q(1-mbc:maxmx+mbc, meqn)
c     dimension aux(1-mbc:maxmx+mbc, *)
c  -------------------------------------------------
c      If method(7)=0  or the auxiliary variables are not needed in this solver,
c      then the latter dimension statement can be omitted, but aux should
c      still appear in the argument list.
c
c      On input, q(i,m) contains the data for solving the
c                source term equation.
c      On output, q(i,m) should have been replaced by the solution to
c                 the source term equation after a step of length dt.
c
c
c      b4step1 = subroutine that is called from claw1 before each call to
c                step1.  Use to set time-dependent aux arrays or perform
c                other tasks which must be done every time step.
c
c          The form of this subroutine is
c
c  -------------------------------------------------
c      subroutine b4step1(maxmx,mbc,mx,meqn,q,xlower,dx,time,dt,maux,aux)
c      implicit double precision (a-h,o-z)
c      dimension   q(1-mbc:maxmx+mbc, meqn)
c      dimension aux(1-mbc:maxmx+mbc, *)
c  -------------------------------------------------
c
c
c
c
c =========================================================================
c
c  Copyright 1994 -- 2002 R. J. LeVeque
c
c  This software is made available for research and instructional use only.
c  You may copy and use this software without charge for these non-commercial
c  purposes, provided that the copyright notice and associated text is
c  reproduced on all copies.  For all other uses (including distribution of
c  modified versions), please contact the author at the address given below.
c
c  *** This software is made available "as is" without any assurance that it
c  *** will work for your purposes.  The software may in fact have defects, so
c  *** use the software at your own risk.
c
c  --------------------------------------
c    CLAWPACK Version 4.1,  August, 2002
c    Webpage: http://www.amath.washington.edu/~claw
c  --------------------------------------
c    Author:  Randall J. LeVeque
c             Applied Mathematics
c             Box 352420
c             University of Washington,
c             Seattle, WA 98195-2420
c             rjl@amath.washington.edu
c =========================================================================
c
c
c
c
c    ======================================================================
c    Beginning of claw1 code
c    ======================================================================
c
      USE Boundary_conditions
      implicit double precision (a-h,o-z)
      external rp1,src1,b4step1
      dimension q(1-mbc:maxmx+mbc, meqn)
      dimension aux(1-mbc:maxmx+mbc, maux)
      dimension work(mwork)
      dimension mthlim(mwaves),method(7),dtv(5),cflv(4),nv(2)
      dimension mthbc(2)
      common /comxt/ dtcom,dxcom,tcom
c
c
      info = 0
      t = tstart
      maxn = nv(1)
      dt = dtv(1)   !# initial dt
      cflmax = 0.d0
      dtmin = dt
      dtmax = dt
      nv(2) = 0
!sbm
!      maux = method(7)

c
c     # check for errors in data:
c
      if (mx .gt. maxmx) then
         info = 1
         go to 900
         endif
c
      if (method(1) .eq. 0) then
c        # fixed size time steps.  Compute the number of steps:
         if (tend .lt. tstart) then
c             # single step mode
             maxn = 1
           else
              maxn = (tend - tstart + 1d-10) / dt
              if (dabs(maxn*dt - (tend-tstart)) .gt.
     &                          1d-5*(tend-tstart)) then
c                # dt doesn't divide time interval integer number of times
                 info = 2
                 go to 900
                 endif
          endif
         endif
c
      if (method(1).eq.1 .and. cflv(2).gt.cflv(1)) then
         info = 3
         go to 900
         endif
c
c     # partition work array into pieces for passing into step1:
      i0f = 1
      i0wave = i0f + (maxmx + 2*mbc) * meqn
      i0s = i0wave + (maxmx + 2*mbc) * meqn * mwaves
      i0dtdx = i0s + (maxmx + 2*mbc) * mwaves
      i0qwork = i0dtdx + (maxmx + 2*mbc)
      i0amdq = i0qwork + (maxmx + 2*mbc) * meqn
      i0apdq = i0amdq + (maxmx + 2*mbc) * meqn
      i0dtdx = i0apdq + (maxmx + 2*mbc) * meqn
      i0end = i0dtdx + (maxmx + 2*mbc) - 1
c
      if (mwork .lt. i0end) then
         write(6,*) 'mwork must be increased to ',i0end
         info = 4
         go to 900
         endif
c
c     -----------
c     # main loop
c     -----------
c
      if (maxn.eq.0) go to 900
      do 100 n=1,maxn
         told = t   !# time at beginning of time step.

!         print*,'Time = ',t,' s'


c        # adjust dt to hit tend exactly if we're near end of computation
c        #  (unless tend < tstart, which is a flag to take only a single step)
         if (told+dt.gt.tend .and. tstart.lt.tend) dt = tend - told

         if (method(1).eq.1) then
c           # save old q in case we need to retake step with smaller dt:
            call copyq1(maxmx,meqn,mbc,mx,q,work(i0qwork))
            endif
c
   40    continue
         dt2 = dt / 2.d0
         thalf = t + dt2  !# midpoint in time for Strang splitting
         t = told + dt    !# time at end of step

c        # store dt and t in the common block comxt in case they are needed
c        # in the Riemann solvers (for variable coefficients)
         tcom = told
         dtcom = dt
         dxcom = dx

c_sbm
c         write(*,'(" Time = ",E10.2)')t

c
c        ------------------------------------------------------------------
c        # main steps in algorithm:
c        ------------------------------------------------------------------
c


c        # extend data from grid to bordering boundary cells:

         call bc1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,told,dt,mthbc)

! The following has been moved in the b4step1 subroutine in order to simplify the code
c_sbm    # calculate the auxilary variables
!         call aux1(maxmx,meqn,mbc,mx,q,maux,aux)

c
c        # call user-supplied routine which might set aux arrays
c        # for this time step, for example.
```

```
        ! print*," b4step1 ..."

         call b4step1(maxmx,mbc,mx,meqn,q,
     &                xlower,dx,told,dt,maux,aux)
c
c

!sbm
!         write(*,*)'claw1';
         if (method(5).eq.2) then
c            # with Strang splitting for source term:
             call src1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,told,dt2)
             endif
c
c        # take a step on the homogeneous conservation law:
         call step1(maxmx,meqn,mwaves,mbc,mx,q,maux,aux,dx,dt,
     &             method,mthlim,cfl,work(i0f),work(i0wave),
     &             work(i0s),work(i0amdq),work(i0apdq),work(i0dtdx),
     &             rp1)
c
         if (method(5).eq.2) then
c            # source terms over a second half time step for Strang splitting:
c            # Note it is not so clear what time t should be used here if
c            # the source terms are time-dependent!
             call src1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,thalf,dt2)
             endif

         if (method(5).eq.1) then
c            # source terms over a full time step:
             call src1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,t,dt)
             endif
c

c
c        -------------------------------------------------------------------
c
         if (method(4) .eq. 1) write(6,601) n,cfl,dt,t
  601    format('CLAW1... Step',i4,
     &                   '   Courant number =',f6.3,'  dt =',d12.4,
     &                   '  t =',d12.4)
c
         if (method(1) .eq. 1) then
c           # choose new time step if variable time step
            if (cfl .gt. 0.d0) then
                dt = dmin1(dtv(2), dt * cflv(2)/cfl)
                dtmin = dmin1(dt,dtmin)
                dtmax = dmax1(dt,dtmax)
              else
                dt = dtv(2)
              endif
            endif
c
c        # check to see if the Courant number was too large:
c
         if ((cfl-d1mach(4)) < cflv(1)) then
c               # accept this step
                cflmax = dmax1(cfl,cflmax)
              else
c               # reject this step
                t = told
                call copyq1(maxmx,meqn,mbc,mx,work(i0qwork),q)
c
```

```
                if (method(4) .eq. 1) then
                   write(6,602)
  602              format('CLAW1 rejecting step... ',
     &                         'Courant number too large')
                   endif
                if (method(1).eq.1) then
c                   # if variable dt, go back and take a smaller step
                    go to 40
                  else
c                   # if fixed dt, give up and return
                    cflmax = dmax1(cfl,cflmax)
                    go to 900
                  endif
               endif
c
c        # see if we are done:
         nv(2) = nv(2) + 1
         if (t .ge. tend) go to 900
c
  100    continue
c
  900  continue
c
c      # return information
c
       if (method(1).eq.1 .and. t.lt.tend .and. nv(2) .eq. maxn) then
c         # too many timesteps
          info = 11
          endif
c
       if (method(1).eq.0 .and. cflmax .gt. cflv(1)) then
c         # Courant number too large with fixed dt
          info = 12
          endif
       tend = t
       cflv(3) = cflmax
       cflv(4) = cfl
       dtv(3) = dtmin
       dtv(4) = dtmax
       dtv(5) = dt
       return
       end


module constants_module

!Written by: R.Samuel


    implicit none
    save area_ht, volume_ht, area_ht_sp, rho_wall, cp_wall, &
         mu_l, mu_v, k_l, k_v, &
         rho_l, rho_v, latent_heat, void, TWALL

    !=================================
    ! Fluid properties
    ! Written by: R.Samuel
    !=================================

    integer, parameter :: dp_t = selected_real_kind(15,307)
    integer, parameter :: nc = 1
    integer, parameter :: ncmax = 20
    character(len=255), dimension(ncmax) :: component
    character(len=255), dimension(ncmax) :: hf
    character(len=255) :: hfmix,hrf
    character(len=255) :: hcode,hcite,herr
    integer :: iflag
    integer :: icomp
    integer :: ierr

    real(dp_t),dimension(ncmax) :: z
    real(dp_t),allocatable :: mu_l(:),mu_v(:),k_l(:),k_v(:)
    real(dp_t),allocatable :: cp_l(:),cp_v(:)
    real(dp_t),allocatable :: rho_l(:), rho_v(:)
    real(dp_t),allocatable :: latent_heat(:)
    real(dp_t),allocatable :: void(:)

    !=================================
    ! Pipeline dimensions
    !=================================
    real(dp_t),parameter :: id = 0.125_dp_t          ! 233 mm
    real(dp_t),parameter :: PipeThickness = 0.020_dp_t ! 20 mm
    real(dp_t),parameter :: length = 2500._dp_t         ! 256 m
    real(dp_t) :: area_ht, volume_ht, area_ht_sp       ! heat transfer area and volume
of the pipe
    real(dp_t), public :: rho_wall = 7700.d0           ! kg/m3
    real(dp_t), public :: cp_wall = 500.d0             ! J/kg
    real(dp_t),allocatable :: TWALL(:)

end module


Module Heat_Transfer_Coeff
!use bl_types
!use bl_constants_module

!Written by: R.Samuel

use constants_module
implicit none

real(kind=dp_t), private :: InsulationThickness = 0.05      ! m - thermal insulation
thickness  (5 cm: Zhang et al, 2006)
real(kind=dp_t), private :: ThermalCondInsulation  = 0.058  ! Watt/m/K  - thermal
conductivity of insulation material (0.058: cellular galss)
real(kind=dp_t), private :: ThermalCondWall  = 31.15        ! Watt/m/K  - thermal
conductivity of pipe wall
real(kind=dp_t), private :: htc_amb = 10.                   ! W/m2/K - coefficient of
convective heat transfer to air
real(kind=dp_t), public ::  TempAmb = 285.d0

    contains

    function htc_DB(Re,Pr,k,D,alpha) result(htc)
    real(kind=dp_t),intent(in)::Re,Pr
    real(kind=dp_t),intent(in)::k,D,alpha
    real(kind=dp_t) :: htc
    htc = 0.023 * abs(Re)**(0.8) * abs(Pr)**(0.4) *k/(alpha*D)
    return
    end function

    function htc_Droplet(Re,Pr,k,D,alpha) result(htc)
    real(kind=dp_t),intent(in)::Re,Pr
    real(kind=dp_t),intent(in)::k,D,alpha
    real(kind=dp_t) :: htc

    htc = 10._dp_t*k/(alpha*D)
    return
    end function

    elemental function htc(htc_f,id,PipeThickness) result(htc_overall)
        real(kind=dp_t), intent(in) :: htc_f, id, PipeThickness
        !!Internal Variables
        real(kind=dp_t) :: htc_w, htc_i, htc_s, htc_a, htc_, htc_overall, od
        ! Heat transfer coefficient

        od = id + PipeThickness

        htc_w = 1./(0.5*id/ThermalCondWall*DLOG(od/id))

        htc_i = 1./(0.5*id/ThermalCondInsulation*DLOG((od+InsulationThickness)/id))

        !We're going to ignore the soil heat conductivity
        !htc_s = 1./(0.5*id/ThermalCondSoil*DLOG(PipeDepth/id))

        !htc_a = htc_amb * 2.* PipeDepth / id

        htc_i = 10000000.d0

        htc_ = 1./(1./(htc_f+epsilon(htc_)) + 1./htc_w + 1./htc_i)  !+ 1./htc_amb)

        htc_overall = htc_
        return
    end function

    real(kind=dp_t) function q_nucleate
(cp_l,rho_l,rho_v,latent_heat,Pr,TBULK,TWALL,mu_l)
        implicit none
        save SUR

        real(dp_t),intent(in) :: cp_l,rho_l,rho_v,latent_heat
        real(dp_t),intent(in) :: PR,TBULK,TWALL,mu_l

        !
        !local variables
        !
        real(dp_t) :: x(ncmax),y(ncmax)
        real(dp_t),parameter :: G = 9.81 !m/s^2
        real(dp_t) :: CP,DENL,DENV,HL
        real(dp_t) :: WMOL
        real(dp_t) :: SUR
        real(dp_t) :: TE,T
        real(dp_t) :: C,D

        integer :: ierr
        character(len=255) :: herr

        x = 0.d0
        y = 0.d0
        x(1) = 1.d0
        y(1) = 1.d0

        TE = TWALL - TBULK
        T = TBULK
        if (T > 220.d0) then
            call SURTEN(T,rho_l,rho_v,X,Y,SUR,ierr,herr)

            if (ierr /= 0) then
                print*, 'surface tension calculation fails'
            end if

        else
            SUR = SUR
        end if

        CP = cp_l                 !J/kg-K
        DENL = rho_l              !kg/m3
        DENV = rho_v
        HL = latent_heat          !J/kg

        C = 0.006_dp_t
        D = 1._dp_t

        q_nucleate = mu_l*SQRT(G*(DENL-DENV)/SUR)/(HL**2)
        q_nucleate = q_nucleate*((CP*TE/(C*(Pr**D)))**3)

        return
    end function

end module

Module Transport_Properties
!use array_size
!Use bl_types
!use bl_constants_module
!use FluidProperties, only: ncmax,herr

use constants_module
Implicit none

real(kind=dp_t),external :: wmol

Contains

    real(kind=dp_t) function Prandtl(Thermal_Cond,cp,viscosity)
        real(kind=dp_t),intent(in)::Thermal_Cond
        real(kind=dp_t),intent(in)::cp,viscosity
        Prandtl = cp*viscosity/Thermal_Cond
    return
    end function

    real(kind=dp_t) function Reynolds(rho,velocity,alpha,diameter,viscosity)
        real(kind=dp_t),intent(in)::rho
        real(kind=dp_t),intent(in)::velocity,viscosity
        real(kind=dp_t),intent(in)::diameter
        real(kind=dp_t),intent(in)::alpha
        Reynolds = alpha*rho*abs(velocity)*Diameter/viscosity
    return
    end function

    real(kind=dp_t) function Thermal_Conductivity(rho_,T,x)

      real(kind=dp_t),intent(in)::rho_                   ! density
      real(kind=dp_t),intent(in)::T                      ! temperature
      real(kind=dp_t),intent(in)::x(ncmax)               ! composition
      real(kind=dp_t)::rho, Thermal_Conductivity_
      !!
      integer :: ierr
      real(kind=dp_t)::eta,tcx,mw_fluid,t_,y(ncmax)

      y = 0._dp_t
      y(nc) = 1._dp_t
      mw_fluid = wmol(y)
      rho = rho_ !/ mw_fluid
      t_ = t
      if (t<3000.d0) t_ = 3000.D0
      call TRNPRP (t_,rho,y,eta,tcx,ierr,herr)
      if (ierr/=0) then
         print*,'Thermal Conductivity'
         print*,ierr,herr
         print*,rho,T_,tcx
         !read*
      end if
      Thermal_Conductivity = tcx
      return
    end function

    real(kind=dp_t) function Viscosity(rho_,T,x)

      real(kind=dp_t),intent(in)::rho_
      real(kind=dp_t),intent(in)::T
      real(kind=dp_t),intent(in)::x(ncmax)
      real(kind=dp_t)::rho
      !!
      integer :: ierr
      real(kind=dp_t)::eta,tcx, mw_fluid, t_, y(ncmax)

      y = 0._dp_t
      y(nc) = 1._dp_t
      mw_fluid = wmol(y)
      rho = rho_ !/ mw_fluid
      t_ = t
      if (t<3000.d0) t_ = 3000.D0
      call TRNPRP (t_,rho,y,eta,tcx,ierr,herr)
      if (ierr/=0) then
         print*,'Viscosity'
         print*,ierr
         print*,rho,T_,eta,mw_fluid
         print*,'Composition: ',x(:nc)
         !read*
      end if
      Viscosity = eta*1.d-6
      return
    end function
end module

c     ============================================
      subroutine b4step1(maxmx,mbc,mx,meqn,q,
     &            xlower,dx,t,dt,maux,aux)
c     ============================================
c
c     # called from claw1 before each call to step1.
c     # use to set time-dependent aux arrays or perform other tasks
c     # which must be done every time step.
c
c     # dummy routine
c
c
```

```
      Use Failure_type
      USE Boundary_conditions
      use fluxToQData
      implicit None
      integer::maxmx
      integer::mbc
      integer::mx
      integer::meqn
      double precision::xlower
      double precision::dx
      double precision::dt
      double precision::t
      integer::maux
      double precision,dimension(1-mbc:maxmx+mbc, meqn):: q
      double precision,dimension(1-mbc:maxmx+mbc, maux) :: aux
      !internal variables
      double precision, dimension(meqn) ::S_punc,qriemann
      integer :: i
      integer :: j
      double precision :: D_p
      double precision :: pipeThickness
      double precision :: told
      double precision :: ratio
      integer , parameter :: n = 2
      integer , parameter :: lw = 120 !(3*n**2+13*n)/2.
      external fluxToQ
      double precision fvec(n),tol_nms,w(lw),x(n)
      integer  iflag,  info, iopt, nprint
      double precision x_tol
      integer :: index_HRM
      double precision :: theta
      double precision f(1-mbc:maxmx_p+mbc,meqn)
      !commons (these may be removed later)
      common /pipe_diam/ D_p,pipeThickness
      common /TOLS/ tol_nms,x_tol
      common /HRM/ index_HRM, theta

      if (nozzle) then
          ! In this case some alteration to the boundary condition is required
      i = mx+1
          if (.not.allocated(q_p)) then
              ! We have to allocate the arrays in module Failure_Type
              open(unit=70,file='Puncture.csv',status='replace')
              dx_p = D_p/real(mx_p)
              Allocate(q_p(1-mbc:maxmx_p+mbc,meqn),
     & aux_p(1-mbc:maxmx_p+mbc,maux))
              Allocate(amdq_p(1-mbc:maxmx_p+mbc,meqn),
     & apdq_p(1-mbc:maxmx_p+mbc,meqn))
              Allocate(wave_p(1-mbc:maxmx_p+mbc,meqn,mwaves_p))
              Allocate(s_p(1-mbc:maxmx_p+mbc, mwaves_p))
              do j = 1-mbc,maxmx_p+mbc
                  ! Initialising arrays for the flow through the puncture
                            q_p(j,:) = q(i,:)
                            aux_p(j,:) = aux(i,:)
                     end do
          else
             ! The space has been allocated and so we only require to open the output
file
              open(unit=70,file='Puncture.csv',access='append')
              q_p(1,:)= q(i,:)
              aux_p(1,:) = aux(i,:)
          end if
```

```
!! Calculating the flow through the puncture
          call bc1(maxmx_p,meqn,mbc,mx_p,xlower,dx_p,q_p,maux,aux_p,t
     &         ,dt,mthbc_p)
          call rp1(maxmx_p,meqn,mwaves_p,mbc,mx_p,q_p,q_p,maux
     & ,aux_p,wave_p,s_p,amdq_p,apdq_p)
!! Collecting the flux through the puncture as a source term
          ratio = R_punc**2/((D_p/2.)**2*dx)
          S_punc(1) = amdq_p(2,1)*ratio
          S_punc(2) = amdq_p(2,2)*ratio
          S_punc(3) = amdq_p(2,3)*ratio
          S_punc(4) = amdq_p(2,4)*ratio

          qriemann(1) = amdq_p(2,1)*ratio/s_p(2,1) + q_p(1,1)
          qriemann(2) = amdq_p(2,2)*ratio/s_p(2,1) + q_p(1,2)
          qriemann(3) = amdq_p(2,3)*ratio/s_p(2,1) + q_p(1,3)
          qriemann(4) = amdq_p(2,4)*ratio/s_p(2,1) + q_p(1,4)

!Calculate state of exiting fluid
        !iopt = 2 ! approximate Jacobian
!       iopt = 1 ! Jacobian is evaluated in external function
      !call ffcn(maxmx_p,1000,meqn,mbc,mx_p,q_p,maux,aux_p,f)
        !flux1 = amdq_p(2,1)*ratio*dx
        !flux2 = amdq_p(2,2)*ratio*dx
        !flux3 = amdq_p(2,3)*ratio*dx
        !flux4 = amdq_p(2,4)*ratio*dx

!        if(index_HRM.eq.1) then
!        x_dynamic = amdq_p(2,4)/amdq_p(2,1)
!        end if
!        nprint = 0
!!        tol_nms = 1.0d-12
!        x(1) = (q_p(2,1)+q_p(1,1))/2.
!        x(2) = (aux_p(2,1)+aux_p(1,1))/2.
!
!        call dnsqe ( fluxToQ, Jac_, iopt, n, x,
!     &   fvec, tol_nms, nprint, info, w, lw )
          !! outputing outflow rate

          print*,R_punc,amdq_p(2,1),q_p(1,:)
          print*,aux_p(1,:)
          read(*,*)
          write(70,1000) t,amdq_p(2,1)*R_punc**2*3.14159,
     & qriemann(2)*R_punc**2*3.14159
          close(unit=70)
!-------------------------------------------------------------------------------------
--------
!        Source term for injection:
!-------------------------------------------------------------------------------------
--------
         q(i,:) = q(i,:) - dt*S_punc(:)

      end if
c_sbm    # calculate the auxilary variables
         call aux1(maxmx,meqn,mbc,mx,q,maux,aux)

1000     format(e16.8,20(' , ',e16.8))
      return
          end


c
```

```
c
c ===================================================================
      subroutine step1(maxmx,meqn,mwaves,mbc,mx,q,maux,aux,dx,dt,
     &              method,mthlim,cfl,f,wave,s,amdq,apdq,dtdx,rp1)
c ===================================================================
c
c     # Take one time step, updating q.
c
c     method(1) = 1   ==>  Godunov method
c     method(1) = 2   ==>  Slope limiter method
c     mthlim(p)  controls what limiter is used in the pth family
c
c
c     amdq, apdq, wave, s, and f are used locally:
c
c     amdq(1-mbc:maxmx+mbc, meqn) = left-going flux-differences
c     apdq(1-mbc:maxmx+mbc, meqn) = right-going flux-differences
c        e.g. amdq(i,m) = m'th component of A^- \Delta q from i'th Riemann
c                         problem (between cells i-1 and i).
c
c     wave(1-mbc:maxmx+mbc, meqn, mwaves) = waves from solution of
c                                           Riemann problems,
c            wave(i,m,mw) = mth component of jump in q across
c                           wave in family mw in Riemann problem between
c                           states i-1 and i.
c
c     s(1-mbc:maxmx+mbc, mwaves) = wave speeds,
c            s(i,mw) = speed of wave in family mw in Riemann problem between
c                      states i-1 and i.
c
c     f(1-mbc:maxmx+mbc, meqn) = correction fluxes for second order method
c            f(i,m) = mth component of flux at left edge of ith cell
c     -------------------------------------------------------------------
c
      implicit double precision (a-h,o-z)
      dimension    q(1-mbc:maxmx+mbc, meqn)
      dimension  aux(1-mbc:maxmx+mbc, maux)
      dimension    f(1-mbc:maxmx+mbc, meqn)
      dimension    s(1-mbc:maxmx+mbc, mwaves)
      dimension wave(1-mbc:maxmx+mbc, meqn, mwaves)
      dimension amdq(1-mbc:maxmx+mbc, meqn)
      dimension apdq(1-mbc:maxmx+mbc, meqn)
      dimension dtdx(1-mbc:maxmx+mbc)
      dimension method(7),mthlim(mwaves)
      logical limit
!sbm
!         write(*,*)'step1 .. begin';
c
c     # check if any limiters are used:
      limit = .false.
      do 5 mw=1,mwaves
         if (mthlim(mw) .gt. 0) limit = .true.
   5     continue
c
      mcapa = method(6)
      do 10 i=1-mbc,mx+mbc
         if (mcapa.gt.0) then
             if (aux(i,mcapa) .le. 0.d0) then
                write(6,*) 'Error -- capa must be positive'
                stop
                endif
             dtdx(i) = dt / (dx*aux(i,mcapa))
```

```
            else
             dtdx(i) = dt/dx
            endif
   10    continue
c
c
c
c     # solve Riemann problem at each interface
c     -----------------------------------------
c
      call rp1(maxmx,meqn,mwaves,mbc,mx,q,q,
     & maux,aux,wave,s,amdq,apdq)
c
c     # Modify q for Godunov update:
c     # Note this may not correspond to a conservative flux-differencing
c     # for equations not in conservation form.  It is conservative if
c     # amdq + apdq = f(q(i)) - f(q(i-1)).
c

      do 40 i=1,mx+1
         do 40 m=1,meqn

!sbm: Sol's correction:   apdq(i,m) = 0 at mx+1
!            if (i.gt.mx) print*,apdq(i,m),i

            q(i,m) = q(i,m) - dtdx(i)*apdq(i,m)
            q(i-1,m) = q(i-1,m) - dtdx(i-1)*amdq(i,m)
   40       continue

c
c     # compute maximum wave speed:
      cfl = 0.d0
      do 50 mw=1,mwaves
         do 45 i=1,mx+1
c          # if s>0 use dtdx(i) to compute CFL,
c          # if s<0 use dtdx(i-1) to compute CFL:
           cfl = dmax1(cfl, dtdx(i)*s(i,mw), -dtdx(i-1)*s(i,mw))
   45      continue
   50    continue
c
      if (method(2) .eq. 1) go to 900
c
c     # compute correction fluxes for second order q_{xx} terms:
c     ----------------------------------------------------------
c
      do 100 m = 1, meqn
            do 100 i = 1-mbc, mx+mbc
               f(i,m) = 0.d0
  100          continue
c
c      # apply limiter to waves:
      if (limit) call limiter(maxmx,meqn,mwaves,mbc,mx,wave,s,mthlim)
c
      do 120 i=1,mx+1
         do 120 m=1,meqn
            do 110 mw=1,mwaves
               dtdxave = 0.5d0 * (dtdx(i-1) + dtdx(i))
               f(i,m) = f(i,m) + 0.5d0 * dabs(s(i,mw))
     &             * (1.d0 - dabs(s(i,mw))*dtdxave) * wave(i,m,mw)
  110          continue
  120       continue
c
```

```
c
c_sbm  140 continue

c
c     # update q by differencing correction fluxes
c     ============================================
c
c     # (Note:  Godunov update has already been performed above)
c
      do 150 m=1,meqn
         do 150 i=1,mx
            q(i,m) = q(i,m) - dtdx(i) * (f(i+1,m) - f(i,m))
  150       continue
c
  900 continue
      return
      end



      Module Boundary_conditions
      contains
c
c     =================================================================
      subroutine bc1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,t,
     & dt,mthbc)
c     =================================================================
c
c     # Standard boundary condition choices for claw2
c
c     # At each boundary  k = 1 (left),  2 (right):
c     #   mthbc(k) =  0  for user-supplied BC's (must be inserted!)
c     #            =  1  for zero-order extrapolation
c     #            =  2  for periodic boundary coniditions
c     #            =  3  for solid walls, assuming this can be implemented
c     #                  by reflecting the data about the boundary and then
c     #                  negating the 2'nd component of q.
c     ------------------------------------------------
c
c     # Extend the data from the computational region
c     #      i = 1, 2, ..., mx2
c     # to the virtual cells outside the region, with
c     #      i = 1-ibc  and   i = mx+ibc   for ibc=1,...,mbc
c
      USE PPL_interface
      USE Failure_type
      implicit double precision (a-h,o-z)
      integer, INTENT(IN)::maxmx
      integer, INTENT(IN)::meqn
      integer, INTENT(IN)::mbc
      integer, INTENT(IN)::mx
      double precision, INTENT(IN)::xlower
      double precision, INTENT(IN)::dx
      integer, INTENT(IN):: maux
      double precision, INTENT(IN)::t
      double precision, INTENT(IN)::dt
      double precision,INTENT(INOUT),dimension(1-mbc:maxmx+mbc, meqn)::q
      double precision,INTENT(INOUT),dimension(1-mbc:maxmx+mbc,maux)::
     & aux
      integer,INTENT(in),dimension(2) :: mthbc

      integer :: i
      ! Internal variables
      double precision :: u,Cs,pres_,pres,rho,u_
      double precision :: rho_new,u_new_p_new,T_new
      double precision :: wave_1,wave_2,wave_3,wave_4

      !=====================
      !refprop variables
      !Written by: R.Samuel
      !=====================
      double precision :: z(20),xliq(20),xvap(20)
      integer :: ierr
      character(255) :: herr
      !=====================


      common /problem/ p_out, t_out,p_in,T_in, rho_in, x_in, Cs_in
      common /pipe_diam/ D_p,pipeThickness
      common /HRM/ index_HRM, theta

c
c
c-------------------------------------------------------
c     # left boundary:
c-------------------------------------------------------
      go to (100,110,120,130,140) mthbc(1)+1
c
  100 continue

c     # user-specified boundary conditions go here in place of error output
C      write(6,*) '*** ERROR *** mthbc(1)=0 and no BCs specified in bc1'
C      stop


c_sbm ------------------------------------------------------------------------
c     Constant pressure at the inlet, and
c     zero-order extrapolation for the rest of the variables.
c
!    do 101 m=1,meqn
!       do 101 ibc=1,mbc
!             q(1-ibc,m) = q(1,m)
!101 continue

!
!     Expressions for BCs for the Euler equations are taken from Thompson (1987)
!
!        Flow properties for the first internal cell (i=1)
!        u_, pres_ needed for approximation of the spatial derivatives.


!===========================================================================
! subsonic inflow boundary condition
! Written by: R.Samuel 22/03/2016
!===========================================================================
! It is a modification of the original subsonic inlet boundary condition
!===========================================================================

      i=1;            ! previous timestep cell information at i = 1 (first interior
cell)
         u = q(i,2)/q(i,1);
         rho = q(i,1);
               e = q(i,3)/rho-0.5*u*u;
         if (index_HRM.eq.1) then
            x_mf  = q(i,4)/rho  ! HRM: non-eq vapour mass fraction
         else
            x_mf = x_eq         ! HEM: equilibrium mass fraction
         endif

         v_ = 1./rho;

         pres = aux(i,1)
         Temp = aux(i,2)
         x_eq = aux(i,3)
         Cs   = aux(i,4)
         Tliq = aux(i,5)

         u_=u;
         rho_=rho;
         pres_=pres;
         x_mf_=x_mf
        rho_x_ = rho;  ! .. for the relax eqn source term ..
        Tliq_ = Tliq


      do i=0,1-mbc,-1

         rho = q(i,1);
         u = q(i,2)/q(i,1);
               e = q(i,3)/rho-0.5*u*u;
         v_ = 1./rho;

         pres = aux(i,1)
         Temp = aux(i,2)
         x_eq = aux(i,3)   ! eq vapour mass fraction
         Cs   = aux(i,4)
         Tliq = aux(i,5)

         if (index_HRM.eq.1) then
            x_mf  = q(i,4)/rho  ! HRM: non-eq vapour mass fraction
         else
            x_mf = x_eq         ! HEM: equilibrium mass fraction
         endif

        if (i==0)then

        !======================
        !Written by: R.Samuel
        ! Initialise the ghost cell
        !======================
         if (t >= 0.d0) then
             rho = 924.62;
             u = 0.d0;
             e = 207080.d0+E_refprop_ref;
             v_ = 1./rho;

             pres = 50.0d5
             Temp = 277.d0
             x_eq = 0.d0   ! eq vapour mass fraction
             Cs   = 670.42
             Tliq = 277.d0

             q(i,1)  = rho
             q(i,2)  = 0.d0
             q(i,3)  = 0.d0
             q(i,4)  = rho*e
```

```
        else if (t>100.d0) then

            rho = 0.19402;
            u = 0.d0;
            e = 48578.d0+E_refprop_ref;
            v_ = 0;

            pres = 1.0d5
            Temp = 273.d0
            x_eq = 0.d0   ! eq vapour mass fraction
            Cs    = 670.42
            Tliq = 273.d0

            q(i,1) = rho
            q(i,2) = 0.d0
            q(i,3) = 0.d0
            q(i,4) = rho*e

        end if


       !=====================
!========================================================================
!
!=========================================================================
!     subsonic inflow boundary condition
!=========================================================================
!    Written by: R.Samuel
!=========================================================================
! the information (amplitude carried by, in the order of appearence,
! left running, material wave (entropy wave) and right running characteristics
!=========================================================================
           if (t <= 3000.d0) then
               drhoudt = 2.58d0 ! gradient of inflow mass flux
           else
               drhoudt = 0.d0
           end if

           wave1 = (u-Cs)/dx*(pres_-pres - rho*Cs*(u_-u))
           wave2 = 0.0
           wave3 = (1.d0/(u+Cs))*(-2.d0*u*wave2-2.d0*drhoudt*Cs*Cs
    &                  -wave1*(u-Cs))

           dpdt = -0.5d0*(wave3+wave1)
           dudt = -0.5d0/(2.*rho*Cs)*(wave3-wave1)
           drhodt = 1.0d0/(Cs*Cs)*(dpdt+wave2)
           dmfdt = 0.d0

!==========================================================================

!         x_mf = x_mf_!0.5*(x_mf+x_mf_);  !!!!!!!!!!!!!!!!

!             Source terms for the governing Eqs for conservative variables:
!       (Eqs (65-67) in Thompson (1987))
!
!              Continuity Eq:
        !print*,'Before i= ',i,q(i,1),q(i,2)
        q(i,1)= q(i,1) + dt*(drhodt);

!              Momentum Eq:
        q(i,2)= q(i,2) + dt*(u*drhodt + rho*dudt);
```

```
!        q(i,2)= q(i,2) + dt*(u*drhodt + q(i,1)*dudt);
         !print*,'After i= ',i,q(i,1),q(i,2)

!              Mass fraction Eq:
         q(i,4)= q(i,4) + dt*(x_mf*drhodt+rho*dmfdt)
      !&     + (dt/dx)**2*(q(i+1,4)-2*q(i,4)+q(i-1,4));

!         print*,' bc1 ...'; pause


!------------------------------------------------------------------------------
!  Update the internal energy using the new values of pressure and density:

         rho_new = q(i,1);
         p_new   = pres + dt*dpdt;
         u_new   =   u  + dt*dudt;

!         x_new_   = x_mf + dt*dmfdt;
         x_new_  = q(i,4) / rho_new;

!  Ensure the limits for the mass fraction = [0;1]:
         if (x_new_.gt.1.) x_new_=1.
         if (x_new_.lt.0.) x_new_=0.
         q(i,4) = x_new_* rho_new;

        if (index_HRM.eq.1) then
!!             HRM:
!         v_new = 1./rho_new;
!         x_new = x_new_;
!         T_new = Temp;
!         T_liq = Temp;  ! Tliq;
!         e_new = e;
!         Call DP_flash_neq(v_new,e_new,x_new,p_new,T_new,T_liq);
!
!
!         if (T_new.gt.T_liq) then
!!           write(*,"(' T_new > T_liq, ',4F12.3)")T_new,T_liq,p_new,x_new
!!           read*
!!           T_new=T_liq   !!!!!!!! 18May2012
!         endif
!
!!         if (T_liq.gt.(T_new+100.)) T_liq=Tnew+100.  ! 18May2012
!
        else
!              HEM:
         v_new = 1./rho_new;
         x_new = x_mf;
         T_new = Temp;
         e_new = e;
!         Call DP_flash(v_new,e_new,p_new,T_new,x_new)

         !=====================================================
         !
         !=====================================================
         !Call DP_flash_sat (v_new,h_new,s_new,p_new,T_new,x_new)
         !e_new = h_new - p_new*v_new
         !=====================================================
         z = 0.d0
         z(1) = 1.d0
         call PDFLSH(p_new/1000.d0,rho_new/44.01,z,T_new,rholiq,rhovap,
     &               xliq,xvap,x_new,e_new,h,s,cv,cp,w,ierr,herr)
```

```
         if (x_new >= 1.d0) then
             x_new = 1.0
         else if (x_new <= 0.d0) then
             x_new = 0.0
         end if

         e_new = e_new/(44.01/1000.d0)+E_refprop_ref

         T_liq    = T_new;
        endif


        q(i,3) = rho_new*(e_new + 0.5d0*u_new*u_new);



        if (index_HRM.eq.1) then
!!------------------------------------------------------------------------------
!!             HRM:
!!------------------------------------------------------------------------------
!!   Update the equilibrium mass fraction x_new_eq:
!         T_new_eq = T_new
!         p_new_eq = p_new      ! 11 June 2012
!         x_equi = x_new_
!         Call DP_flash(v_new,e_new_eq,p_new,T_new_eq,x_equi)
!!         Call DE_flash(v_new,e_new,p_new_eq,T_new_eq,x_equi)
!
!!   Ensure the limits for the mass fraction = [0;1]:
!         if (x_equi.gt.1.) x_equi=1.
!         if (x_equi.lt.0.) x_equi=0.
!
!!------------------------------------------------------------------------------
!!    Add the relaxation equation source term and update the mass fraction:
!!
!!        rho_x = rho_x_;    ! use the upwind cell value
!!        rho_x = rho;       ! use the current cell old value
!!        rho_x = rho_new;   ! use the current cell new value
!
!         q(i,4) = rho_new*x_equi +
!     &            (q(i,4)-rho_new*x_equi)*dexp(-dt/theta);
!
!         x_new_ = q(i,4)/rho_new
!
!!   Ensure the limits for the mass fraction = [0;1]:
!         if (x_new_.gt.1.) x_new_ = 1.
!         if (x_new_.lt.0.) x_new_ = 0.
!         q(i,4)=x_new_ *rho_new
!
!!------------------------------------------------------------------------------
!!        T_new = T_new_eq
!!        T_liq = T_new_eq
!!         Call DE_flash_neq (v_new,e_new,x_new_,p_new,T_new,T_liq);
!          Call DP_flash_neq(v_new,e_new,x_new_,p_new,T_new,T_liq);
!         if (T_new.lt.1.) T_new = 1.
!         if (T_liq.lt.1.) T_liq = 1.
!         if (p_new.lt.1.) p_new = 1.
        else
!------------------------------------------------------------------------------
!              HEM:
!------------------------------------------------------------------------------
         x_equi = x_new
         x_new_ = x_new
```

```
      endif

      !=========
      !Written by: R.Samuel
      !=========
         if (x_new > 0.d0 .and. x_new < 1.d0) then
         p_bar = p_new/1.d5
         rho_v = d_v(p_bar);
         h_v_  = h_v(p_bar);
         e_v   = h_v_- p_new/rho_v;
         Cs_v  = a_v(p_bar);

         rho_l = d_c(p_bar);
         h_l_  = h_c(p_bar);
         e_l   = h_l_- p_new/rho_l;
         Cs_l  = a_c(p_bar);                       ! sbm 24-09-2013

         !if (x_new_.lt.1.d-6) then
         !!   special case: pure liquid:
         !       rho_l= den_liq_PPL(p_new,T_new)
         !       e_l  =   e_liq_PPL(rho_l,T_new) + E_ref ! E_Anvar = E_PPL + E_ref
         !       Cs_l =  Cs_liq_PPL(rho_l,T_new)
         !       h_l_ =  e_l + p_new/rho_l;
         !endif

         vf = vf_mix(x_new_,rho_v,rho_l);
         Cs1 = Cs_mix(vf,p_new,T_new,rho_new,rho_v,rho_l,Cs_v,Cs_l);
         else
             Cs1 = w
         end if

         aux(i,1) = p_new
         aux(i,2) = T_new
         aux(i,3) = x_equi   ! x_new_
         aux(i,4) = Cs1
         aux(i,5) = T_liq
         !print*,'High pressure end:',p_new,T_new,u_new
         !read(*,*)

              else
        q(i,:) = q(i+1,:)
        aux(i,:) = aux(i+1,:)
      end if

      enddo
      go to 199


c
  110 continue
c     # zero-order extrapolation:
      do 115 m=1,meqn
         do 115 ibc=1,mbc
               q(1-ibc,m) = q(1,m)
  115       continue
      go to 199

  120 continue
c     # periodic:
      do 125 m=1,meqn
         do 125 ibc=1,mbc
               q(1-ibc,m) = q(mx+1-ibc,m)
  125       continue
      go to 199


  130 continue




c     # solid wall (assumes 2'nd component is velocity or momentum in x):
      do 135 m=1,meqn
         do 135 ibc=1,mbc
               q(1-ibc,m) = q(ibc,m)
  135       continue
c     # negate the normal velocity:
      do 136 ibc=1,mbc
            q(1-ibc,2) = -q(ibc,2)
  136    continue
      go to 199

  140 continue
!-------------------------------------------------------------------------
!     Constant inlet flow rate
!-------------------------------------------------------------------------
       i=1;

         u = q(i,2)/q(i,1);
         rho = q(i,1);
               e = q(i,3)/rho-0.5*u*u;
         if (index_HRM.eq.1) then
            x_mf  = q(i,4)/rho  ! HRM: non-eq vapour mass fraction
         else
            x_mf = x_eq         ! HEM: equilibrium mass fraction
         endif

         v_ = 1./rho;

         pres = aux(i,1)
         Temp = aux(i,2)
         x_eq = aux(i,3)
         Cs   = aux(i,4)
         Tliq = aux(i,5)

         u_=u;
         rho_=rho;
         pres_=pres;
         x_mf_=x_mf
        rho_x_ = rho;  ! .. for the relax eqn source term ..
        Tliq_ = Tliq


      do i=0,1-mbc,-1
!      i=mx;

!      print*,' -------- BC1 orifice calculations ----------------------'
!      print*
!      print*,' -------- i = m+1  ---------------------------------------'
!      print*

         rho = q(i,1);
         u = q(i,2)/q(i,1);
               e = q(i,3)/rho-0.5*u*u;
         v_ = 1./rho;

         pres = aux(i,1)
         Temp = aux(i,2)
         x_eq = aux(i,3)   ! eq vapour mass fraction
         Cs   = aux(i,4)
         Tliq = aux(i,5)

         if (index_HRM.eq.1) then
            x_mf  = q(i,4)/rho  ! HRM: non-eq vapour mass fraction
         else
            x_mf = x_eq         ! HEM: equilibrium mass fraction
         endif

        if (i==0)then




!             Amplitude variations of characteristic waves
!             (Thompson' notation - handwritten capital "L"):
!       (Eqs (83,85,87) in Thompson (1987))
               if(u.gt.Cs) then
!            u=Cs;
              wave_1 = (u-Cs)/dx*(pres_-pres - rho*Cs*(u_-u));
         else
              wave_1 = 0.;  !this is OK for chocked flow (sonic release: u=Cs), but is
only crude
                          !approximation for subsonic release (u<Cs)
         endif;

               if(u.gt.0.) then
              wave_2 = u/dx*(pres_-pres - Cs*Cs*(rho_-rho));
         else
              wave_2 = 0.;
         endif;

               if(u.gt.(-Cs)) then
              wave_3 = (u+Cs)/dx*(pres_-pres + rho*Cs*(u_-u));
         else
              wave_3 = 0.;
         endif;

               if(u.gt.(0.)) then
              wave_4 = u/dx*(x_mf_ - x_mf);
         else
              wave_4 = 0.;
         endif;

!             Time derivatives of the primitive variables:
!       (Eqs (88-90) in Thompson (1987))

!         The following conditions are applied in this case:
!         1. L2 = 0 : i.e. entropy is constant
!         2. L3 given by equation (70) in Thompson (1990)


           wave_2 = 0.
           wave_3 = (-2.*u*wave_2 - wave_1*(u-Cs))/(u+Cs)
           wave_4 = 0.0;  !!!!!!!!!   sbm - 14May 2012

           dpdt = -0.5*(wave_3 + wave_1)
           dudt = -0.5/rho/Cs*(wave_3-wave_1);
           drhodt = (dpdt + wave_2)/Cs**2;
           dmfdt = -wave_4


!         x_mf = x_mf_!0.5*(x_mf+x_mf_);  !!!!!!!!!!!!!!!!

!             Source terms for the governing Eqs for conservative variables:
!       (Eqs (65-67) in Thompson (1987))
!
!              Continuity Eq:
         q(i,1)= q(i,1) + dt*(drhodt);

!              Momentum Eq:
         q(i,2)= q(i,2) + dt*(u*drhodt + rho*dudt);
!        q(i,2)= q(i,2) + dt*(u*drhodt + q(i,1)*dudt);

!              Mass fraction Eq:
         q(i,4)= q(i,4) + dt*(x_mf*drhodt+rho*dmfdt)
      !&     + (dt/dx)**2*(q(i+1,4)-2*q(i,4)+q(i-1,4));

!         print*,' bc1 ...'; pause


!------------------------------------------------------------------------------
!  Update the internal energy using the new values of pressure and density:

         rho_new = q(i,1);
         p_new    = pres + dt*dpdt;
         u_new    =   u  + dt*dudt;

!         x_new_   = x_mf + dt*dmfdt;
         x_new_  = q(i,4) / rho_new;

!  Ensure the limits for the mass fraction = [0;1]:
         if (x_new_.gt.1.) x_new_=1.
         if (x_new_.lt.0.) x_new_=0.
         q(i,4) = x_new_* rho_new;

        if (index_HRM.eq.1) then
!!             HRM:
!         v_new = 1./rho_new;
!         x_new = x_new_;
!         T_new = Temp;
!         T_liq = Temp;  ! Tliq;
!         e_new = e;
!         Call DP_flash_neq(v_new,e_new,x_new,p_new,T_new,T_liq);
!
!
!         if (T_new.gt.T_liq) then
!!           write(*,"(' T_new > T_liq, ',4F12.3)")T_new,T_liq,p_new,x_new
!!           read*
!!           T_new=T_liq   !!!!!!!! 18May2012
!         endif
!
!!          if (T_liq.gt.(T_new+100.)) T_liq=Tnew+100.  ! 18May2012

        else
!              HEM:
         v_new = 1./rho_new;
         x_new = x_mf;
         T_new = Temp;
         e_new = e;
!         Call DP_flash(v_new,e_new,p_new,T_new,x_new)
         Call DP_flash_sat (v_new,h_new,s_new,p_new,T_new,x_new)
         e_new = h_new - p_new*v_new

         T_liq   = T_new;
        endif


        q(i,3) = rho_new*(e_new + 0.5d0*u_new*u_new);



        if (index_HRM.eq.1) then
!!------------------------------------------------------------------------------
!!             HRM:
!!------------------------------------------------------------------------------
!!   Update the equilibrium mass fraction x_new_eq:
!         T_new_eq = T_new
!         p_new_eq = p_new      ! 11 June 2012
!         x_equi = x_new_
!         Call DP_flash(v_new,e_new_eq,p_new,T_new_eq,x_equi)
!!         Call DE_flash(v_new,e_new,p_new_eq,T_new_eq,x_equi)
!
!!   Ensure the limits for the mass fraction = [0;1]:
!         if (x_equi.gt.1.) x_equi=1.
!         if (x_equi.lt.0.) x_equi=0.
!
!!------------------------------------------------------------------------------
!!    Add the relaxation equation source term and update the mass fraction:
!!
!!        rho_x = rho_x_;   ! use the upwind cell value
!!        rho_x = rho;      ! use the current cell old value
!!        rho_x = rho_new;  ! use the current cell new value
!
!         q(i,4) = rho_new*x_equi +
!     &            (q(i,4)-rho_new*x_equi)*dexp(-dt/theta);
!
!         x_new_ = q(i,4)/rho_new
!
!!   Ensure the limits for the mass fraction = [0;1]:
!         if (x_new_.gt.1.) x_new_ = 1.
!         if (x_new_.lt.0.) x_new_ = 0.
!         q(i,4)=x_new_ *rho_new
!
!!------------------------------------------------------------------------------
!!        T_new = T_new_eq
!!        T_liq = T_new_eq
!!         Call DE_flash_neq (v_new,e_new,x_new_,p_new,T_new,T_liq);
!          Call DP_flash_neq(v_new,e_new,x_new_,p_new,T_new,T_liq);
!         if (T_new.lt.1.) T_new = 1.
!         if (T_liq.lt.1.) T_liq = 1.
!         if (p_new.lt.1.) p_new = 1.
        else
!------------------------------------------------------------------------------
!              HEM:
!------------------------------------------------------------------------------
         x_equi = x_new
         x_new_ = x_new
        endif
```

```
         p_bar = p_new/1.d5
         rho_v = d_v(p_bar);
         h_v_  = h_v(p_bar);
         e_v   = h_v_- p_new/rho_v;
         Cs_v  = a_v(p_bar);

         rho_l = d_c(p_bar);
         h_l_  = h_c(p_bar);
         e_l   = h_l_- p_new/rho_l;
         Cs_l  = a_c(p_bar);                    ! sbm 24-09-2013

         if (x_new_.lt.1.d-6) then
         !   special case: pure liquid:
                rho_l= den_liq_PPL(p_new,T_new)
                e_l  =   e_liq_PPL(rho_l,T_new)  + E_ref ! E_Anvar = E_PPL + E_ref
                Cs_l =  Cs_liq_PPL(rho_l,T_new)
                h_l_ =  e_l + p_new/rho_l;
         endif

         vf = vf_mix(x_new_,rho_v,rho_l);
         Cs1 = Cs_mix(vf,p_new,T_new,rho_new,rho_v,rho_l,Cs_v,Cs_l);


         aux(i,1) = p_new
         aux(i,2) = T_new
         aux(i,3) = x_equi   ! x_new_
         aux(i,4) = Cs1
         aux(i,5) = T_liq

              else
          q(i,:) = q(i-1,:)
          aux(i,:) = aux(i-1,:)
        end if

      enddo
      go to 199

  199 continue

c
c
!---------------------------------------------------------------------------
c     # right boundary:
!---------------------------------------------------------------------------
      go to (200,210,220,230,240) mthbc(2)+1
c
  200 continue
c     # user-specified boundary conditions go here in place of error output
c      write(6,*) '*** ERROR *** mthbc(2)=0 and no BCs specified in bc2'
c      stop

c_sbm -------------------------------------------------------------------------
c     Constant pressure at the outlet, and
c     zero-order extrapolation for the rest of the variables.
c

!sbm   17May 2012:
      ! do 201 m=1,meqn
      !   do 201 ibc=1,mbc

      ! Adding here an outflow correlation based on the paper:
      ! "Modeling of wellbore dynamics of a CO2 injector during transient
      ! well shut-in and start-up operations"
      ! IJGGC 42, 2015, 602-614

      coeffA = 0.d0
      coeffB = 1.3478d12
      coeffC = 2.1592d10

      ! Information from the last computational cell (i.e. i = mx)

      i = mx;
      rho = q(i,1);
      u = q(i,2) / q(i,1);
      e = q(i,3) / rho - 0.5d0 * u**2;
      v_ = 1.d0 / rho;

      pres = aux(i,1)
      Temp = aux(i,2)
      x_eq = aux(i,3)   ! eq vapour mass fraction
      Cs   = aux(i,4)
      Tliq = aux(i,5)

      ! Assigning the new values in the ghost cell according to the
      ! outflow correlation mentioned in the aforementioned paper

      p_static = 198.d5 ! 172.d5 ! CHECK IT and make it consistent with the INITIAL
PROFILE

      ! Same density
      q(i + 1, 1) = rho;
      ! Same velocity
      q(i + 1, 2) = rho * u;
      ! We need to take into account the smaller diameter after the discontinuity
      flowrate_outflow = rho*u*3.14159265d0*0.25d0 * id**2 / 2.5d0;
      p_bhf = dsqrt(p_static**2 + coeffA + coeffB * flowrate_outflow +
     &        coeffC * flowrate_outflow**2)

      z = zero
      z(1) = one
      call PDFLSH(p_bhf/1000.d0, rho/44.01, z , T_new, rholiq, rhovap,
     &               xliq, xvap, x_new, e_new, h,s,cv,cp, w, ierr,herr)

      e_new = e_new / (44.01 / 1000.d0) + E_refprop_ref
      q(i + 1, 3) = rho * (e_new + 0.5d0 * u**2);
      q(i + 1, 4) = 0.d0;

      aux(i + 1, 1) = p_bhf;
      aux(i + 1, 2) = T_new;
      aux(i + 1, 3) = 1.d0;
      aux(i + 1, 4) = w;
      aux(i + 1, 5) = T_new;

      go to 299

C$$$$$$                 print*,' BC: m =',m,'; q =',q(mx+ibc,m), q(mx,m)
C$$$$$$                 read*

C$$$$$$                 q(mx+ibc,m) = q(mx,m)

 201    continue

!
!     Expressions for BCs for the Euler equations are taken from Thompson (1987)
```

```
!
!        Flow properties for last internal cell (i=mx)
!        u_, pres_ needed for approximation of the spatial derivatives.
      i=mx;

!      print*,' -------- BC1 orifice calculations ----------------------'
!      print*
!      print*,' -------- i = m+1  --------------------------------------'
!      print*

        rho = q(i,1);
        u = q(i,2)/q(i,1);
              e = q(i,3)/rho-0.5*u*u;
        v_ = 1./rho;

        pres = aux(i,1)
        Temp = aux(i,2)
        x_eq = aux(i,3)   ! eq vapour mass fraction
        Cs   = aux(i,4)
        Tliq = aux(i,5)

        if (index_HRM.eq.1) then
           x_mf  = q(i,4)/rho  ! HRM: non-eq vapour mass fraction
        else
           x_mf = x_eq         ! HEM: equilibrium mass fraction
        endif

!!        write(*,'(" e-p:",6F11.2)')rho,u,pres/1.d5,e,x_mf,Temp

       u_=u;
       rho_=rho;
       pres_=pres;
       x_mf_=x_mf;
       x_eq_=x_eq;

       rho_x_ = rho;  ! .. for the relax eqn source term ..
       Tliq_ = Tliq


      do i=mx+1,mx+mbc
        if (i==mx+1)then
!          Flow properties at the ghost cells:
!             The outflow boundary condition is implemented here:
!              -> subsonic release: p_ghost = p_out (external pressure),
!              -> sonic and supersonic release: p_ghost doesn't affect the flow.


        rho = q(i,1);
        u = q(i,2)/q(i,1);
        e = q(i,3)/rho-0.5*u*u;
        v_ = 1./rho;

        pres = aux(i,1)
        Temp = aux(i,2)
        x_eq = aux(i,3)   ! eq vapour mass fraction
        Cs   = aux(i,4)
        Tliq = aux(i,5)

        if (index_HRM.eq.1) then
           x_mf  = q(i,4)/rho  ! HRM: non-eq vapour mass fraction
        else
           x_mf = x_eq         ! HEM: equilibrium mass fraction
```

```
         endif


!             Amplitude variations of characteristic waves
!             (Thompson' notation - handwritten capital "L"):
!       (Eqs (83,85,87) in Thompson (1987))
               if(u.gt.Cs) then
!            u=Cs;
              wave_1 = (u-Cs)/dx*(pres-pres_ - rho*Cs*(u-u_));
         else
              wave_1 = 0.;  !this is OK for chocked flow (sonic release: u=Cs), but is 
only crude
                          !approximation for subsonic release (u<Cs)
         endif;

               if(u.gt.0.) then
              wave_2 = u/dx*(pres-pres_ - Cs*Cs*(rho-rho_));
         else
              wave_2 = 0.;
         endif;

               if(u.gt.(-Cs)) then
              wave_3 = (u+Cs)/dx*(pres-pres_ + rho*Cs*(u-u_));
         else
              wave_3 = 0.;
         endif;

               if(u.gt.(0.)) then
              wave_4 = u/dx*(x_mf - x_mf_);
         else
              wave_4 = 0.;
         endif;

!             Time derivatives of the primitive variables:
!       (Eqs (88-90) in Thompson (1987))
         dpdt = -0.5*(wave_3+wave_1);
         dudt = -0.5/rho/Cs*(wave_3-wave_1);
         drhodt = 1./Cs/Cs*(dpdt+wave_2);
         dmfdt = -wave_4;


!>>   Subsonic release:
         !============================================
         ! in the case the flow is no longer choked
         ! Written by: R.Samuel
         !===========================================

         if (u.le.Cs) then
!           print*, ' ============ u < Cs ================='
!          read*

           pres = p_out; ! atmospheric pressure

           wave_3 = (u+Cs)/dx*(pres-pres_ + rho*Cs*(u-u_));
           wave_1 = - wave_3;
           wave_2 = 0.;
           wave_4 = u/dx*(x_mf - x_mf_);  !!!!!!!!!   sbm - 14May 2012

           dpdt = 0.
           dudt = -0.5/rho/Cs*(wave_3-wave_1);
           drhodt = wave_2/Cs**2;
           dmfdt = -wave_4
         end if
!<<   Subsonic release.


!             Source terms for the governing Eqs for conservative variables:
!       (Eqs (65-67) in Thompson (1987))
!
!              Continuity Eq:
         q(i,1)= q(i,1) + dt*(drhodt);

!              Momentum Eq:
         q(i,2)= q(i,2) + dt*(u*drhodt + rho*dudt);
!        q(i,2)= q(i,2) + dt*(u*drhodt + q(i,1)*dudt);

!              Mass fraction Eq:
         q(i,4)= q(i,4) + dt*(x_mf*drhodt+rho*dmfdt)
      !&     + (dt/dx)**2*(q(i+1,4)-2*q(i,4)+q(i-1,4));

!         print*,' bc1 ...'; pause


!------------------------------------------------------------------------------
!  Update the internal energy using the new values of pressure and density:

         rho_new = q(i,1);
         p_new   = pres + dt*dpdt;
         u_new   =   u  + dt*dudt;

!         x_new_   = x_mf + dt*dmfdt;
         x_new_  = q(i,4) / rho_new;

!  Ensure the limits for the mass fraction = [0;1]:
         if (x_new_.gt.1.) x_new_=1.
         if (x_new_.lt.0.) x_new_=0.
         q(i,4) = x_new_* rho_new;

        if (index_HRM.eq.1) then
!!             HRM:
!         v_new = 1./rho_new;
!         x_new = x_new_;
!         T_new = Temp;
!         T_liq = Temp;  ! Tliq;
!         e_new = e;
!         Call DP_flash_neq(v_new,e_new,x_new,p_new,T_new,T_liq);
!


        else
!              HEM:
         v_new = 1./rho_new;
         x_new = x_mf;
         T_new = Temp;
         e_new = e;
! sbm 18 Sept 2013
!         Call DP_flash(v_new,e_new,p_new,T_new,x_new)
         Call DP_flash_sat (v_new,h_new,s_new,p_new,T_new,x_new)
         e_new = h_new - p_new*v_new

         T_liq   = T_new;
        endif
        q(i,3) = rho_new*(e_new + 0.5d0*u_new*u_new);



        if (index_HRM.eq.1) then
!!-------------------------------------------------------------------------------
!!             HRM:
!!-------------------------------------------------------------------------------
!!   Update the equilibrium mass fraction x_new_eq:
!         T_new_eq = T_new
!         p_new_eq = p_new          ! 11 June 2012
!         x_equi = x_new_
!         Call DP_flash(v_new,e_new_eq,p_new,T_new_eq,x_equi)
!
!!   Ensure the limits for the mass fraction = [0;1]:
!         if (x_equi.gt.1.) x_equi=1.
!         if (x_equi.lt.0.) x_equi=0.
!
!!-------------------------------------------------------------------------------
!!    Add the relaxation equation source term and update the mass fraction:
!!
!!        rho_x = rho_x_;    ! use the upwind cell value
!!        rho_x = rho;       ! use the current cell old value
!!        rho_x = rho_new;   ! use the current cell new value
!
!         q(i,4) = rho_new*x_equi +
!     &            (q(i,4)-rho_new*x_equi)*dexp(-dt/theta);
!
!         x_new_ = q(i,4)/rho_new
!
!!   Ensure the limits for the mass fraction = [0;1]:
!         if (x_new_.gt.1.) x_new_ = 1.
!         if (x_new_.lt.0.) x_new_ = 0.
!         q(i,4)=x_new_ *rho_new
!
!!-------------------------------------------------------------------------------
!!        T_new = T_new_eq
!!        T_liq = T_new_eq
!!         Call DE_flash_neq (v_new,e_new,x_new_,p_new,T_new,T_liq);
!          Call DP_flash_neq(v_new,e_new,x_new_,p_new,T_new,T_liq);
!         if (T_new.lt.1.) T_new = 1.
!         if (T_liq.lt.1.) T_liq = 1.
!         if (p_new.lt.1.) p_new = 1.
!        else
!-------------------------------------------------------------------------------
!              HEM:
!-------------------------------------------------------------------------------
         x_equi = x_new
         x_new_ = x_new
        endif


         p_bar = p_new/1.d5
         rho_v = d_v(p_bar);
         h_v_  = h_v(p_bar);
         e_v   = h_v_- p_new/rho_v;
         Cs_v  = a_v(p_bar);

         rho_l = d_c(p_bar);
         h_l_  = h_c(p_bar);
         e_l   = h_l_- p_new/rho_l;
         Cs_l  = a_c(p_bar);
         if (x_new_.lt.1.d-6) then
         !   special case: pure liquid:
                rho_l= den_liq_PPL(p_new,T_new)
                e_l  =   e_liq_PPL(rho_l,T_new) + E_ref ! E_Anvar = E_PPL + E_ref
                Cs_l =  Cs_liq_PPL(rho_l,T_new)
                h_l_ =  e_l + p_new/rho_l;
         endif

         vf = vf_mix(x_new_,rho_v,rho_l);
         Cs1 = Cs_mix(vf,p_new,T_new,rho_new,rho_v,rho_l,Cs_v,Cs_l);

!sbm
!        Cs1=1000.


         aux(i,1) = p_new
         aux(i,2) = T_new
         aux(i,3) = x_new_ !x_equi   ! x_new_   ! smb 03-10-2013
         aux(i,4) = Cs1
         aux(i,5) = T_liq


              else
          q(i,:) = q(i-1,:)
          aux(i,:) = aux(i-1,:)
        end if

      enddo


      go to 299

c_sbm ------------------------------------------------------------------------


  210 continue
c     # zero-order extrapolation:
      do 215 m=1,meqn
         do 215 ibc=1,mbc
               q(mx+ibc,m) = q(mx,m)
  215       continue
      go to 299

  220 continue
c     # periodic:
      do 225 m=1,meqn
         do 225 ibc=1,mbc
               q(mx+ibc,m) = q(ibc,m)
  225       continue
      go to 299

  230 continue
c     # solid wall (assumes 2'nd component is velocity or momentum in x):
      do 235 m=1,meqn
         do 235 ibc=1,mbc
               q(mx+ibc,m) = q(mx+1-ibc,m)
  235       continue
      do 236 ibc=1,mbc
            q(mx+ibc,2) = -q(mx+1-ibc,2)
  236    continue
      go to 299
  240 continue
c     # solid wall (assumes 2'nd component is velocity or momentum in x):
      do 245 m=1,meqn
         do 245 ibc=1,mbc
               q(mx+ibc,m) = q(mx+1-ibc,m)
  245       continue
      do 246 ibc=1,mbc
            q(mx+ibc,2) = -q(mx+1-ibc,2)
  246 continue
      nozzle = .True.
      go to 299

  299 continue
c
      return
      end subroutine
      end Module Boundary_conditions

c =========================================================
      subroutine rp1(maxmx,meqn,mwaves,mbc,mx,ql,qr
     &         ,maux,aux,wave,s,amdq,apdq)
c =========================================================
c
c     # solve Riemann problem using a variant of the HLLE solver.
c
c     # The user must specify the flux function using ffcn below, and
c     # also provide a subroutine sminmax that returns estimates of
c     # min and max wave speeds.
c
c     # On input, ql contains the state vector at the left edge of each cell
c     #           qr contains the state vector at the right edge of each cell
c     # On output, wave contains the waves,
c     #            s the speeds,
c     #            amdq the  left-going flux difference  A^- \Delta q
c     #            apdq the right-going flux difference  A^+ \Delta q
c
c     # Note that the i'th Riemann problem has left state qr(i-1,:)
c     #                                    and right state ql(i,:)
c     # From the basic clawpack routine step1, rp is called with ql = qr = q.
c
c
      implicit double precision (a-h,o-z)
      dimension   ql(1-mbc:maxmx+mbc, meqn)
      dimension   qr(1-mbc:maxmx+mbc, meqn)
      dimension    s(1-mbc:maxmx+mbc, mwaves)
      dimension wave(1-mbc:maxmx+mbc, meqn, mwaves)
      dimension amdq(1-mbc:maxmx+mbc, meqn)
      dimension apdq(1-mbc:maxmx+mbc, meqn)
      dimension  aux(1-mbc:maxmx+mbc, maux)
      integer:: i,m


!      common /param/ gamma,gamma1
c     Written by: R.Samuel
c     # local storage
c     ---------------
      parameter (max2 = 10002)  !# assumes at most 4000 grid points with mbc=2
      dimension s1(1-mbc:maxmx+mbc), s2(1-mbc:maxmx+mbc)
!sbm      dimension f(-1:max2, 3)
      dimension   f(1-mbc:maxmx+mbc, meqn)
```

```
c
      if (mx+mbc .gt. max2) then
         write(6,*) '*** Error *** need to increase max2 in rp1eux'
         stop
         endif
c
      if (mwaves .ne. 2) then
         write(6,*) '*** Error ***  Set mwaves = 2 to use rp1hlle'
         stop
         endif
c
c     # evaluate flux function at each cell average
c     # (assuming ql=qr = cell average):
      call ffcn(maxmx,max2,meqn,mbc,mx,ql,maux,aux,f)
c
c     # estimate largest and smallest wave speed at each interface:
      call sminmax(maxmx,max2,meqn,mbc,mx,ql,maux,aux,s1,s2)
c
c     # 1-wave and 2-wave are chosen to span all characteristics

!sbm
      do 100 i=2-mbc,mx+mbc
         do 90 m=1,meqn
            dq = ql(i,m) - qr(i-1,m)
            df = f(i,m) - f(i-1,m)
            s(i,1) = s1(i)
            s(i,2) = s2(i)

            wave(i,m,1) = (df - s(i,2)*dq) / (s(i,1)-s(i,2))
            wave(i,m,2) = (df - s(i,1)*dq) / (s(i,2)-s(i,1))

   90    continue
  100 continue
c
c     # Compute fluctuations:
c     # amdq = SUM s*wave   over left-going waves
c     # apdq = SUM s*wave   over right-going waves
c
      do 200 m=1,meqn
!sbm
         do 180 i=2-mbc, mx+mbc
            amdq(i,m) = 0.d0
            apdq(i,m) = 0.d0
            do 150 mw=1,mwaves
               if (s(i,mw) .lt. 0.d0) then
!sol              amdq(i,m) = amdq(i,m) + s(i,mw)*wave(i,m,mw)
!                 if (i>2) amdq(i,m) = amdq(i,m) + s(i,mw)*wave(i,m,mw)  ! sol
                  if (i>1) amdq(i,m) = amdq(i,m) + s(i,mw)*wave(i,m,mw)  ! sbm
                 else
!sol              apdq(i,m) = apdq(i,m) + s(i,mw)*wave(i,m,mw)
                  if(i<=mx) apdq(i,m) = apdq(i,m) + s(i,mw)*wave(i,m,mw) ! sol
                 endif


  150       continue
  180    continue
  200 continue

      return
      end
```

```
c =========================================================
      subroutine ffcn(maxmx,max2,meqn,mbc,mx,q,maux,aux,f)
c =========================================================
      implicit double precision (a-h,o-z)
      dimension   q(1-mbc:maxmx+mbc, meqn)
!sbm      dimension   f(-1:max2, meqn)
      dimension  aux(1-mbc:maxmx+mbc, maux)

      dimension   f(1-mbc:maxmx+mbc, meqn)

!      common /param/ gamma,gamma1
      common /bubble/ fourthirdpi,rho_bub,R_bub, cp_bub


c     # user-supplied function to evaluate the flux function f(q)
c
c
csbm  #  3 Euler equations * (1-alfa)
csbm  #  1 eqn. for alfa*rho_bub (mass fraction of bubbles)
csbm  #  1 momentum conservation eqn. for the dispersed phase
csbm  #  1 energy conservation eqn. for the dispersed phase

      do i=1-mbc,mx+mbc
         u   = q(i,2)/q(i,1);
         rho = q(i,1);
               e = q(i,3)/q(i,1)-0.5d0*u*u;
!eq4
         x_ = q(i,4)/rho

         u_bub=0.!q(i,5)/q(i,4);

         v_ = 1./rho;
!eq4         call DE_flash_neq(v_,e,x_,pres,Temp,Tliq)


! This is the general call for two-phase mixture,
c         call DE_flash(v_,e,pres,Temp,x_)
! which is commented for simplicity here
! and vapour-phase subroutines are used instead:
!
!         x_=1.
!         Temp = T_vap(rho,e);
!         pres = p_vap(rho,e);

         pres = aux(i,1)
         Temp = aux(i,2)
!         x_eq = aux(i,3)
!         Cs   = aux(i,4)

         Tliq = aux(i,5)


!         print*,'   rp1hlle: x =',x_mf
!         read*

!         pres = p_vap(rho,e);
!         Temp = T_vap(rho,e);

          f(i,1) = q(i,2);
          f(i,2) = q(i,2)*u + pres;
          f(i,3) = q(i,3)*u + pres*u;
!
```

```
!eq4
          f(i,4) = q(i,4)*u;





      enddo

      return
      end

c ==========================================================
      subroutine sminmax(maxmx,max2,meqn,mbc,mx,q,
     &           maux,aux,smin,smax)
c ==========================================================
      USE PPL_interface
      implicit double precision (a-h,o-z)
      dimension   q(1-mbc:maxmx+mbc, meqn)
      dimension  aux(1-mbc:maxmx+mbc, maux)

      dimension   smin(1-mbc:maxmx+mbc),smax(1-mbc:maxmx+mbc)
c     # local variable
      dimension   sqmin(1-mbc:maxmx+mbc),sqmax(1-mbc:maxmx+mbc)

c
c_sbm
      common /bubble/ fourthirdpi,rho_bub,R_bub, cp_bub
      common /vapour/ gam_vap,po_vap,cv_vap,ho_vap,ro_vap,so_vap
      common /liquid/ gam_liq,po_liq,cv_liq,ho_liq,ro_liq,so_liq



c     # user-supplied function to estimate the mininum and
c     # maximum wave speeds across interface between cells i-1 and i
c
c     # HLLE method for the Euler equations:
c     # use characteristic speeds u-c and u+c and compare to Roe averages.

!      write (*,'(" HLL ")')
      do i=1-mbc,mx+mbc
c        # compute min and max characteristic speed in i'th cell:
c

c_sbm
         u   = q(i,2)/q(i,1);
         c    = aux(i,4)

         if (i .lt. mx+mbc)  sqmin(i) = u-c
         sqmax(i) = u+c
      enddo


      do i=2-mbc,mx+mbc
!sbm:  temporary
         smin(i) = sqmin(i-1)
         smax(i) = sqmax(i)

      enddo

      return
      end
```