Master’s Thesis

Numerical Simulations of LNG Flow in a Two-Tank System using REFPROP

R.J. Bossier
June 2011

Advisors:
dr. ir. M.L. Beks
prof. dr. R.M.M. Mattheij
dr. ir. A.S. Tijsseling
Abstract

The supply and demand of liquefied natural gas (LNG) is growing at a high rate, leading to an increasing need for models analyzing the entire chain of production, storage and transport. In this report, we will consider a basic system consisting of two tanks connected by two pipes: one is intended for liquid flow and one for gas flow. We discuss several models describing the fluid flow and heat exchange between these tanks, making use of the software package REFPROP to calculate state variables. Simulations have to be fast, so that extensions to larger and more complex systems are still feasible. Therefore, spatial discretization of the tanks and pipes is limited. A system of differential-algebraic equations (DAEs) is formed by applying mass and enthalpy balances over the pipes. Numerical solutions of the DAEs using explicit Euler and Matlab’s ODE solvers ode23t and ode15s are compared, for both single-component and multi-component fluids. Depending on the behavior around discontinuities in the solutions, good results are found. The simulations for the single-component fluids are remarkably faster than for the multi-component fluids.
Acknowledgements

Before the start of this graduation project for the master Industrial and Applied Mathematics, I had a meeting with prof. dr. Bob Mattheij where we discussed possible subjects. I told him that I was looking for a project in the field of scientific computing, but that the topic not necessarily had to be solely about numerical methods. As I followed several courses in physics during my studies, mainly concerning fluid dynamics, he proposed the current subject, that seemed to match my background and interests very well. Indeed, it turned out to be a very challenging project, in which I learned a lot about numerical modeling as well as about the physics involved in the problem. Therefore, I am very grateful to prof. Mattheij for offering me this opportunity.

This project has its origin in work that has been done by the Laboratory of Industrial Mathematics Eindhoven BV (LIME), a spin-off of the Department of Mathematics and Computer Science of the Eindhoven University of Technology (TU/e). This consultancy company carried out some work regarding liquefied natural gas (LNG) for industrial partners, which gave rise to several unanswered questions that formed the start of this project. As I was working in the LIME room at the TU/e during this project, I had the opportunity to work in a research environment while at the same time getting an idea of day-to-day business of a company, which I found a very interesting combination. I especially would like to thank LIME consultant dr.ir. Mark Beks for his useful advice regarding the modeling, his help in solving some computer problems and the small talks at the coffee machine.

Furthermore, my supervisor dr.ir. Arris Tijsseling was always willing to help if I had any problems. He kept an eye on the progress of the project, but at the same time I had the freedom to try out many things myself. For me, this was a very pleasant and motivating way of working, for which I am very thankful. Moreover, I would like to thank him for the detailed feedback and advice, which contributed significantly to the quality of this report.

Many fantastic people from all over the world have worked in the LIME room during my stay there. I am very grateful to them for the nice time we had, motivating and helping each other, but of course also having time for some enjoyable breaks.

Finally, I owe the greatest of thanks to my parents, family and friends for their interest, support and the great time I had in my study as well as next to it.

Ruben Bossier,
## Contents

1 Introduction 1

1.1 Liquefied Natural Gas (LNG) 1

1.1.1 Boil-Off 1

1.1.2 Safety 3

1.1.3 Future Importance 4

1.1.4 Numerical Models 4

1.2 Project Goal 5

1.3 Outline 5

2 Governing Equations 7

2.1 Conservation Laws 7

2.1.1 Reynolds Transport Theorem 7

2.1.2 Conservation of Mass 7

2.1.3 Conservation of Linear Momentum 8

2.1.4 Conservation of Energy 9

2.2 Constitutive Equations 11

2.2.1 Stress Tensor 11

2.2.2 Equations of State 11

2.2.3 Heat Flux 12

2.3 Initial Conditions and Boundary Conditions 12

3 Numerical Methods for Initial Value Problems 13

3.1 Initial Value Problems 13

3.2 Differential-Algebraic Equations 13

3.3 Convergence and Stability 14

3.3.1 Error Tolerance 14

3.4 Time Integration Methods 15
B. REFPROP

B.1 General .......................................................... 113
B.2 Used Models ...................................................... 113
B.3 Incompleteness .................................................... 114

C. Program Code

C.1 Two-Tank System with Constant Temperature without Upper Pipe .......... 115
  C.1.1 Exact Solution for Two-Tank System with Constant Temperature without Upper Pipe ............................................ 115
  C.1.2 Explicit Euler for Two-Tank System with Constant Temperature without Upper Pipe ............................................. 119
C.2 Two-Tank System with Constant Temperature with Upper Pipe ............. 122
  C.2.1 Explicit Euler for Two-Tank System with Constant Temperature with Upper Pipe ............................................. 122
  C.2.2 Runge-Kutta and NDF for Two-Tank System with Upper Pipe with Constant Temperature .............................. 126
C.3 One-Tank System with Variable Temperature ...................................... 132
  C.3.1 Explicit Euler for One-Tank System ..................................... 132
  C.3.2 Runge-Kutta and NDF for One-Tank System ............................. 136
C.4 Two-Tank System with Variable Temperature ...................................... 148
  C.4.1 Explicit Euler for Two-Tank System with Upper Pipe with Variable Temperature ............................................. 148
  C.4.2 Runge-Kutta and NDF for Two-Tank System with Upper Pipe with Variable Temperature .............................. 154
C.5 Multi-Component Fluids ........................................................................ 160
  C.5.1 Explicit Euler for One-Tank System with Multi-Component Fluid ........ 160
  C.5.2 Runge-Kutta and NDF for One-Tank System with Multi-Component Fluid ............................................. 166
  C.5.3 Explicit Euler for Two-Tank System with Multi-Component Fluid ........ 180

Bibliography .............................................................................. 189

Index ..................................................................................... 193
List of Figures

1.1 LNG tanker at a loading terminal ....................... 2
1.2 Sketch of the two-tank system ........................ 2

4.1 Sketch of a possible initial situation of the open two-tank system .................. 18
4.2 Four possible initial situations of the two-tank system .......................... 21
4.3 Two points $a$ and $b$ along a streamline are defined in Bernoulli’s equation .... 21
4.4 Example of equilibrium state with unequal liquid levels in tank 1 and tank 2 .. 23
4.5 Two points $a$ and $b$ in the case that $H_{2,0} < H_{1,0}$ ....................... 25
4.6 Exact solution for an open two-tank system .................................. 27
4.7 Initial situation of the closed two-tank system .................................. 28
4.8 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 5m$, $H_{2,0} = 0m$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$ ................. 31
4.9 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 5m$, $H_{2,0} = 0m$ and $p_{1,0} = p_{\text{atm}}$, $p_{2,0} = 1.1p_{\text{atm}}$ ................. 32
4.10 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 5m$, $H_{2,0} = 0m$ and $p_{1,0} = 1.25p_{\text{atm}}$, $p_{2,0} = p_{\text{atm}}$ ................. 32
4.11 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 20m$, $H_{2,0} = 20m$ and $p_{1,0} = 2p_{\text{atm}}$, $p_{2,0} = p_{\text{atm}}$ ................. 33
4.12 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 20m$, $H_{2,0} = 20m$ and $p_{1,0} = 3p_{\text{atm}}$, $p_{2,0} = p_{\text{atm}}$ ................. 33
4.13 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 20m$, $H_{2,0} = 20m$ and $p_{1,0} = 5p_{\text{atm}}$, $p_{2,0} = p_{\text{atm}}$ ................. 34
4.14 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 15m$, $H_{2,0} = 10m$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$ ................. 34
4.15 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 20m$, $H_{2,0} = 15m$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$ ................. 35
4.16 Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 25m$, $H_{2,0} = 20m$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$ ................. 35
5.1 Point \(a\) is chosen in tank 1, point \(b\) is chosen at the right end of the upper pipe, just inside the pipe. ................................................. 38
5.2 Oscillations occur when using explicit Euler scheme with \(\Delta t = 1\) s. ................................................. 40
5.3 Oscillations occur when using explicit Euler scheme with \(\Delta t = 0.1\) s. ................................................. 41
5.4 Gas pressures and values of \(\Delta t\), with explicit Euler with a time step of 0.005 s. 42
5.5 Oscillations are smaller when using explicit Euler scheme with constraint \(5.13\). 43
5.6 Gas pressure history for a closed two-tank system with explicit Euler with constraint \(5.13\). ......................... 44
5.7 Explicit Euler scheme with constraint \(5.13\) with smaller step sizes. ................................................. 44
5.8 Solution with explicit Euler scheme with constraint \(5.16\). ................................................. 46
5.9 Gas pressure history for a closed two-tank system with explicit Euler with constraint \(5.16\). ......................... 46
5.10 Smooth solution with explicit Euler scheme for \(\Delta t = 1\) s. ................................................. 47
5.11 Oscillations are larger if one tank is almost filled for explicit Euler without additional constraints. ................................. 48
5.12 Oscillations are larger if one tank is almost filled, even if gas flow is limited by constraint \(5.13\). 48
5.13 Oscillations do not occur if we switch to the open-tanks system. ................................................. 49
5.14 Solution for a closed two-tank system by \textit{ode45} with a relative error of \(10^{-2}\) 50
5.15 Solution for a closed two-tank system by \textit{ode45} with a relative error of \(10^{-4}\). 50
5.16 Solution for a closed two-tank system by \textit{ode45} with a relative error of \(10^{-12}\). 51
5.17 Solution for a closed two-tank system by \textit{ode23t} with a relative error of \(10^{-2}\). 51
5.18 Solution for a closed two-tank system by \textit{ode23t} with a relative error of \(10^{-4}\). 52
5.19 Solution for a closed two-tank system by \textit{ode23t} with a relative error of \(10^{-6}\). 52
5.20 Solution for a closed two-tank system by \textit{ode15s} with a relative error of \(10^{-4}\). 53
5.21 Solution for a closed two-tank system by \textit{ode15s} with a relative error of \(10^{-6}\). 53
5.22 Solution for a closed two-tank system by \textit{ode45} with Events option and relative error \(10^{-6}\). ................................................. 54
5.23 Solution for a closed two-tank system by \textit{ode45} with Events option and relative error \(10^{-5}\). ................................................. 55
5.24 Solution for a closed two-tank system for \(D_U = 0.05\) m, \(H_{1,0} = 6\) m, \(H_{2,0} = 5\) m, \(p_1 = p_2 = 10^5\) Pa. ................................................. 56
5.25 Solution for a closed two-tank system for \(D_U = 0.005\) m, \(H_{1,0} = 6\) m, \(H_{2,0} = 5\) m, \(p_1 = p_2 = 10^5\) Pa. ................................................. 57
5.26 Solution for a closed two-tank system for \(D_U = 0.05\) m, \(H_{1,0} = H_{2,0} = 5\) m, \(p_1 = 1.5 \cdot 10^5\) Pa and \(p_2 = 10^5\) Pa. ................................................. 57
5.27 Solution for a closed two-tank system for \(D_U = 0.005\) m, \(H_{1,0} = H_{2,0} = 5\) m, \(p_1 = 1.5 \cdot 10^5\) Pa and \(p_2 = 10^5\) Pa. ................................................. 58
5.28 Solution for a closed two-tank system for $D_U = 0$ m, $H_{1,0} = H_{2,0} = 5$ m, $p_1 = 1.5 \cdot 10^5$ Pa and $p_2 = 10^5$ Pa. ............................................. 58
5.29 ode45 and ode15s give bad results for large requested tolerances. ............ 60

6.1 Sketch of one-tank problem. ............................................................. 63
6.2 Linear temperature profile in the tank wall ........................................... 64
6.3 Solution for a one-tank system by explicit Euler for $\Delta t = 1$ day, for $\xi_0 = 0.01$ and $T_0 = 111$ K. ................................................................. 71
6.4 Solution for a one-tank system by explicit Euler for $\Delta t = 1$ day, for $\xi_0 = 0.001$ and $T_0 = 111$ K. ................................................................. 72
6.5 Solution for one-tank system by ode23t with relative tolerance of $10^{-5}$, for $\xi_0 = 0.01$ and $T_0 = 111$ K. ................................................................. 73
6.6 The evolution of the gas pressure if (6.26) is used instead of (6.10). ............ 74
6.7 Solution for one-tank system with $\xi_0 = 0.01$, $T_0 = 111$ K, by ode15s with Event option. ................................................................. 76
6.8 Solution for one-tank system with $\xi_0 = 0.001$, $T_0 = 111$ K, by ode23t with Event option. ................................................................. 77

7.1 Solution of two-tank system with variable temperature, with explicit Euler with constraints. ................................................................. 82
7.2 Solution with ode23t and a relative tolerance of $10^{-4}$ .......................... 84
7.3 Solution with ode23t and a relative tolerance of $10^{-2}$ .......................... 85
7.4 Solution with ode23t and a relative tolerance of $10^{-3}$ .......................... 86
7.5 Solution with ode15s and a relative tolerance of $10^{-2}$ ......................... 87
7.6 Solution with ode15s and a relative tolerance of $10^{-2}$ ......................... 88

8.1 Solution of multi-component one-tank system by explicit Euler for $T_0 = 111$ K and $\xi_0 = 0.0311$. ................................................................. 94
8.2 Solution of multi-component one-tank system by explicit Euler for $T_0 = 111$ K and $\xi_0 = 0.0311$. ................................................................. 95
8.3 Solution of multi-component one-tank system by explicit Euler for $T_0 = 111$ K and $\xi_0 = 0.05$. ................................................................. 96
8.4 Solution of multi-component one-tank system by explicit Euler for $T_0 = 111$ K and $\xi_0 = 0.05$. ................................................................. 97
8.5 Solution of multi-component one-tank system by ode23t for $T_0 = 111$ K and $\xi_0 = 0.05$. ................................................................. 98
8.6 Solution of multi-component one-tank system by ode23t for $T_0 = 111$ K and $\xi_0 = 0.05$. ................................................................. 99
8.7 Solution of multi-component two-tank system by explicit Euler with constraint (5.10). ................................................................. 100
8.8 Solution of multi-component two-tank system by explicit Euler with constraint (5.16) .............................................................. 101

B.1 REFPROP is not complete .................................................. 114
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Parameters corresponding to the solution given by Figure 4.6</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Parameters corresponding to the solutions given by Figures 4.8-4.16</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Geometrical parameters corresponding to the solutions given in Figures 5.2-5.10</td>
<td>40</td>
</tr>
<tr>
<td>5.2</td>
<td>Speed and order of convergence for explicit Euler method with constraint (5.16)</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>Calculation times for Matlab’s ODE solvers</td>
<td>59</td>
</tr>
<tr>
<td>5.4</td>
<td>Calculation times for Matlab’s ODE solvers for the filling of a tank scenario</td>
<td>60</td>
</tr>
<tr>
<td>5.5</td>
<td>Speed and order of convergence for explicit Euler method with constraint (5.16) for $H_{1,0} = 12$ m and $H_{2,0} = 2$ m.</td>
<td>61</td>
</tr>
<tr>
<td>6.1</td>
<td>Parameters corresponding to the solution given in Figures 6.3-6.8</td>
<td>68</td>
</tr>
<tr>
<td>6.2</td>
<td>Speed and order of convergence for explicit Euler method for one-tank system with $\xi_0 = 0.01$ and $T_0 = 111$ K.</td>
<td>78</td>
</tr>
<tr>
<td>6.3</td>
<td>Calculation times for Matlab’s ODE solvers for the one-tank system</td>
<td>78</td>
</tr>
<tr>
<td>7.1</td>
<td>Speed and order of convergence for explicit Euler method with constraint (5.16) for the two-tank system with variable temperature.</td>
<td>89</td>
</tr>
<tr>
<td>7.2</td>
<td>Calculation times for ode15s and ode23t for two-tank system with variable temperature.</td>
<td>89</td>
</tr>
<tr>
<td>8.1</td>
<td>LNG composition as used in REFPROP calculations</td>
<td>91</td>
</tr>
<tr>
<td>8.2</td>
<td>LNG composition in liquid and gas phase for $p = 1$ bar and $T = 111$ K for the total composition of LNG of Table 8.1</td>
<td>92</td>
</tr>
<tr>
<td>8.3</td>
<td>Speed and order of convergence for explicit Euler method for the one-tank system with multi-component fluid.</td>
<td>102</td>
</tr>
<tr>
<td>8.4</td>
<td>Calculation times for ode15s and ode23t for one-tank system with multi-component fluids.</td>
<td>103</td>
</tr>
<tr>
<td>8.5</td>
<td>Speed and order of convergence for explicit Euler method with constraint (5.16) for the two-tank system with multi-component fluid.</td>
<td>103</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Thanks to technical and economical advantages, natural gas has been one of our major energy sources for already many years. It is clean and offers high possibilities to adapt to the fluctuations in demand. However, the pipelines used to transport natural gas have a few disadvantages: they cannot be laid at any location and they form permanent connections, giving less flexibility to control the supply. Therefore, alternatives are used by means of transportation by ships, trucks and trains. Since the costs are largely depending on the volume, the natural gas is either compressed or liquefied by cooling it to cryogenic temperatures. The first is known as CNG (compressed natural gas) and the latter is known as LNG (liquefied natural gas). We will focus on LNG in this report, although some modeling ideas might be extended to CNG as well.

1.1 Liquefied Natural Gas (LNG)

LNG is natural gas in liquid form, cooled to approximately 111 K (-162 °C) while keeping it around atmospheric pressure. Compared to natural gas, this gives a volume reduction by a factor 600 or even more, with a resulting energy density of about 60% of that of diesel fuel. Furthermore, LNG is colorless, odorless, non-corrosive, non-toxic and lighter than air when vaporized. It has the appearance of mineral water, but is less tasty.

The volume reduction makes storage and transport easier and cheaper. Transportation is mainly done by shipping it with specially designed carriers with strongly insulated storage tanks. In Figure 1.1 an LNG carrier and loading terminal in Australia are shown. The largest LNG carriers currently in use have capacities of 266000 m³. An overview of the active LNG fleet is found in [1].

1.1.1 Boil-Off

Although the storage tanks are strongly insulated, some heat will leak into the system, causing evaporation of the liquid (boil-off). Since the vaporization of the components of LNG occurs

The two should not be confused. CNG is a highly pressurized compressed gas, while LNG is a liquid with a very low temperature stored around atmospheric pressure.
Figure 1.1: LNG tanker with four interconnected vessels at the Karratha gas plant loading terminal, Western Australia [33].

Figure 1.2: Sketch of the two-tank system. The tanks are connected by two pipes. The lower pipe is intended for liquid flow and the upper pipe for gas flow. Throughout this report, we will assume that the upper pipe is connected at the top of the tanks.
at different rates, the composition of the liquid changes over time. This so-called weathering effect leads to a denser liquid, since nitrogen and methane evaporate at a faster rate than ethane and heavier alkanes. The evaporation results in an increasing pressure, and hence also an increasing vapor pressure, which causes slower evaporation compared to open tanks. Normally, extreme pressures are reduced by the use of (de)compressors, removing vapor and thereby keeping the liquid at a low temperature and pressure. Sometimes, this vapor is cooled and reliquefied and again inserted into the tank, but it can also be used as fuel for the compressing and cooling system, or in the case of transportation as a transport fuel. In onshore plants, it is often compressed and added to the regional pipeline network. Models describing different mechanisms of heat leak and methods for evaluating the daily boil-off in LNG tanks are found in, among others, [8, 12, 27].

1.1.2 Safety

Although LNG production, storage and transportation have proven to be relatively safe compared to other energy sources, several accidents have been recorded, caused by a variety of reasons. There exists plenty of literature concerning the safety of LNG. An extensive overview of possible dangers and assessments of the corresponding risks is presented in [34]. Here, we will first give a short description of two hazards that are typical for LNG and then also briefly mention some other dangers that are well known from other substances.

Rapid Phase Transition

One of the most important safety issues is the possibility of rapid phase transition (RPT). This occurs when the cryogenic liquid comes into contact with a hotter environment, for example in the case of a tank damage, when LNG leaks into the sea or the atmosphere. Due to the large temperature difference, a large amount of LNG can heat and evaporate in a very short time interval. Fluid leaking out of the tank also reduces the pressure inside the tank, causing boiling of the liquid, which rapidly releases a large vapor amount. The consequent volume increase may cause the effect of an explosion, in the sense that a large overpressure wave is created and the tank can be destroyed. It is also referred to as a boiling liquid expanding vapor explosion (BLEVE).

Rollover Effect

Another danger is the so-called rollover effect. This can occur when we have layers with different densities in the same tank, for example when a tank already contained older LNG and newer LNG has been loaded on top of it. Due to weathering, the older liquid is warmer and denser than the new liquid, so we obtain two layers in the liquid. At first sight, we might have a stable interface between these layers of LNG. However, due to heat leaking through the walls of the tank, the lower layer can eventually reach a temperature at which its density is reduced so much that the interface becomes unstable. This process is intensified by the movement of heavier components from the lower layer to the top layer and the result is a sudden release of the heat in the lower layer and an increase in vaporization. The consequent abrupt pressure increase can exceed the capacity of the relief valves, resulting in large spills or
damages. A well-reported incident caused by the rollover effect occurred in La Spezia, Italy, in 1971, discharging about 2000 tons of LNG vapor within a few hours time [3].

Other Hazards

Some other dangers or consequences of spills caused by the above-described effects are mentioned below [34].

- As the boil-off natural gas is flammable, fire is regarded as the main threat of LNG spills. Several models of the ignition and development of LNG fires are documented.

- Due to the low temperature, sudden releases of vapor close to public form a danger of freezer burns caused by direct skin contact.

- The risk of death caused by asphyxiation (choking) occurs when oxygen volume concentrations drop below 6%, which is unlikely outdoors thanks to variations in wind directions. Inside tanks, very strict safety rules apply.

- Depending on the composition of the LNG, flammability is a hazard when the natural gas volume concentration in the air is about 5%-15%.

- With an increasing attention for terrorism in the last decades, the danger of attacks on terminals and carriers has been investigated in a large number of studies. Because the large spills that could be caused by such attacks have never occurred in reality, the consequences have to be predicted by computer models.

1.1.3 Future Importance

Natural gas plays an important role in the supply of energy in this world with its ongoing increase in energy consumption and is forecasted to account for 25% of the total energy demand in 2030. Large cost reductions in the entire process of LNG production, storage, transportation and regasification have been accomplished in recent years, resulting in a fast increase in demand and the construction of new carriers and terminals, e.g. in Rotterdam. The LNG production is expected to grow at a higher rate than other treatments of gas with an annual increase of about 10% in the next 10 years [15]. Therefore, there is an increasing need for models analyzing the variously behaving different components of the LNG chain.

1.1.4 Numerical Models

To assess the safety of installations containing LNG and to optimize certain aspects of LNG systems and processes, numerical simulations are carried out. These simulations typically view the installation as an interconnected network of valves, pipes, pumps and containers. By applying mass, force and enthalpy balances, a system of differential algebraic equations is formed, see also Chapter 2. Various software packages such as Aspen HYSYS [2], EcoSimPro [4] and OpenModelica [5] are available to form the system of differential equations from definitions of the components and integrate them numerically. The solvers in these packages, however, are not always robust: they may break down or become unusably slow. In this
project, we will analyse the system of equations formed for certain test problems and build and discuss MATLAB simulations that are combined with the software package REFPROP. This program of the National Institute of Standards and Technology (NIST) provides accurate equations of state for a wide range of substances, including several mixtures [18], see Appendix B and Section 2.2.2 for details.

1.2 Project Goal

We will mainly focus on a basic benchmark example of LNG flow and storage, which is a system consisting of two tanks connected by two pipes. We assume that the lower pipe is intended for liquid flow and that the upper pipe, connected at the tops of the tanks, is for gas flow, see Figure 1.2. The goal of the project is to develop and compare mathematical models and numerical methods to describe the fluid flow and heat exchange, thereby using REFPROP. We will first consider a single-component fluid without heat flow, which is finally extended to a multi-component fluid with heat flow. Hereby, we keep in mind that the speed of the models is important, because it is of interest to add models simulating phenomena as the waterhammer effect, the effect of water waves on flexible hoses and sloshing in carriers in future work. Ideally, the model can also be extended to larger networks of tanks and pipes or other elements such as compressors and heaters.

Furthermore, we try to solve several known numerical problems, related to discontinuities in the solution, to be more specific: discontinuities in time derivatives of e.g. pressure, liquid height and vapor fraction. These can be caused by

- filling of a nearly full tank: the liquid height in one of the tanks reaches the top of the tank, which means that either the flow stops or liquid (instead of gas) starts flowing through the top pipe;
- transitions from a two-phase state to a single-phase state.

1.3 Outline

Before we introduce any models, we first give a brief overview of the governing equations in Chapter 2 and some background on numerical methods in Chapter 3. Then, we start with a two-tank system with single-component flow, constant temperature and only one pipe connection in Chapter 4 and we expand the complexity of the problem and the model step by step in Chapters 5-7. In Chapter 8, we end up with models for both a one-tank system and a two-tank system with multi-component fluids. Finally, a conclusion is given in Chapter 9. The appendices provide a list of symbols, some background information about REFPROP, and the developed program codes.

---

2The waterhammer effect is the propagation of a pressure wave in a liquid when the flow is suddenly forced to stop, for example when a valve is closed.
Chapter 2

Governing Equations

In this part we will describe general equations for fluid flow. To solve the problems in the next chapters, we will start from these equations and derive equations that are suitable for the problem. There exists plenty of literature discussing the general equations more extensively, see for example [20] and [24]. We will globally follow [20] in these derivations. For an overview of the meaning of the variables, see Appendix A.

2.1 Conservation Laws

2.1.1 Reynolds Transport Theorem

Before we derive the conservation laws, we first introduce the Reynolds transport theorem, which we will use in the following derivations. For some quantity per unit volume \( \psi(x, t) \), this theorem is given by

\[
\frac{d}{dt} \int_{V(t)} \psi \, dV = \int_{V(t)} \left( \frac{\partial \psi}{\partial t} + \nabla \cdot (\psi v) \right) \, dV. \tag{2.1}
\]

We refer to [20] for a short proof of this theorem.

2.1.2 Conservation of Mass

Consider a material volume \( V(t) \) of constant mass, i.e. the volume of some substance moving in space, with mass density \( \rho = \rho(x, t) \). Assuming that there are no mass sources, we have conservation of mass:

\[
\frac{d}{dt} \int_{V(t)} \rho \, dV = 0, \tag{2.2}
\]

where the derivative \( \frac{d}{dt} \) is the material derivative with respect to time,

\[
\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + v \cdot \nabla \phi, \tag{2.3}
\]
for any scalar field $\varphi(x, t)$, with $v$ the velocity of the fluid element. The derivative in (2.2) can be taken inside the integral by application of Reynolds transport theorem. Applying the transport theorem (2.1) to equation (2.2) gives
\[
\int_{V(t)} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) \right) dV = 0. \tag{2.4}
\]
This holds for any material volume $V(t)$ and therefore the integrand has to be zero, which gives the so-called continuity equation in conservative form:
\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0. \tag{2.5}
\]
In the case of an incompressible fluid, $\frac{d\rho}{dt} = 0$, equation (2.5) simplifies to
\[
\nabla \cdot v = 0. \tag{2.6}
\]

2.1.3 Conservation of Linear Momentum

Consider again the material volume $V(t)$. Conservation of momentum for this volume is derived from Newton’s second law:
\[
\frac{d}{dt} \int_{V(t)} \rho v \, dV = \int_{V(t)} \rho f_v \, dV + \int_{S(t)} f_s \, ds, \tag{2.7}
\]
where $S(t)$ is the surface of the volume at time $t$, $f_v$ denotes the specific volume force (per unit mass) and $f_s$ denotes the nonspecific surface force (per unit area). We assume that the only volume force in our model is gravity,
\[
f_v = g, \tag{2.8}
\]
with $g = (0, 0, -g)^T$, and that surface forces are due to pressure and viscosity. It is common to write
\[
f_s = \mathcal{T} n, \tag{2.9}
\]
where $\mathcal{T}$ denotes the so-called stress tensor, and $n$ the outer normal of the surface.

We now apply Reynolds transport theorem (2.1) to the left-hand side of equation (2.7) and Gauss’s divergence theorem to the surface integral to obtain
\[
\int_{V(t)} \left( \frac{\partial}{\partial t} (\rho v) + \nabla \cdot (\rho vv^T) \right) dV = \int_{V(t)} \rho g \, dV + \int_{V(t)} \nabla \cdot \mathcal{T}^T \, dV, \tag{2.10}
\]
where
\[
v v^T = \begin{pmatrix} v_1^2 & v_1 v_2 & v_1 v_3 \\ v_2 v_1 & v_2^2 & v_2 v_3 \\ v_3 v_1 & v_3 v_2 & v_3^2 \end{pmatrix} \text{ and } \nabla \cdot \mathcal{T}^T = \begin{pmatrix} \nabla \cdot \mathcal{T}_{1s} \\ \nabla \cdot \mathcal{T}_{2s} \\ \nabla \cdot \mathcal{T}_{3s} \end{pmatrix}.
\]
with \( \mathbf{T}_{i*} \) the \( i \)-th row of \( \mathbf{T} \). It is convenient to split the stress tensor in a pressure related part and a viscous part by writing

\[
\mathbf{T} = -p \mathbf{I} + \mathbf{\tau}.
\]  

(2.11)

The definition of the viscous part \( \mathbf{\tau} \) depends on the used model, we will discuss this in Section 2.2. Using the symmetry of \( \mathbf{T} \), see [20], and splitting the stress tensor in a pressure part and a viscous part, we find

\[
\int_{V(t)} \left( \frac{\partial}{\partial t} (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{vv}^T) \right) \, dV = \int_{V(t)} (\rho \mathbf{g} - \nabla p + \nabla \cdot \mathbf{\tau}) \, dV.
\]  

(2.12)

Since this holds for any material volume \( V(t) \), we can write equation (2.12) in differential form:

\[
\frac{\partial}{\partial t} (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{vv}^T) = \rho \mathbf{g} - \nabla p + \nabla \cdot \mathbf{\tau}.
\]  

(2.13)

First applying the product rule to the left-hand side of equation (2.13), and then using the continuity equation (2.5), we find

\[
\frac{\partial}{\partial t} (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{vv}^T) = \rho \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \frac{\partial \rho}{\partial t} + \mathbf{v} \nabla \cdot (\rho \mathbf{v}) + \rho \mathbf{v} \cdot \nabla \mathbf{v}
\]

\[
= \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v}.
\]  

(2.14)

The momentum equation (2.13) can finally be written as

\[
\rho \frac{d \mathbf{v}}{dt} = \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} = \rho \mathbf{g} - \nabla p + \nabla \cdot \mathbf{\tau}.
\]  

(2.15)

### 2.1.4 Conservation of Energy

In addition, we have conservation of the total energy \( \rho \left(u + \frac{1}{2} ||\mathbf{v}||^2\right) \) per unit volume, where \( u \) is the specific internal energy and \( \frac{1}{2} \rho ||\mathbf{v}||^2 \) is the kinetic energy and the norm \( || \cdot || \) denotes the Euclidean norm. The change in total energy is balanced by the work done by volume forces, surface forces and heat transfers:

\[
\frac{d}{dt} \int_{V(t)} \rho \left(u + \frac{1}{2} \mathbf{v} \cdot \mathbf{v}\right) \, dV = \int_{V(t)} \left(\rho (\mathbf{f}_v \cdot \mathbf{v} + r)\right) \, dV + \int_{S(t)} (\mathbf{f}_s \cdot \mathbf{n} - \mathbf{q} \cdot \mathbf{n}) \, ds,
\]  

(2.16)

where \( \mathbf{q} \) is the heat flux vector and \( \rho r \) the heat absorption per unit volume from internal heat sources. Again, we apply the transport theorem [21] to the left-hand side and Gauss’s divergence theorem to the surface integrals on the right-hand side, which gives with (2.8) and (2.9)

\[
\int_{V(t)} \left( \frac{\partial}{\partial t} \left(\rho \left(u + \frac{1}{2} \mathbf{v} \cdot \mathbf{v}\right)\right) + \nabla \cdot \left(\rho \left(u + \frac{1}{2} \mathbf{v} \cdot \mathbf{v}\right) \mathbf{v}\right) \right) \, dV
\]

\[
= \int_{V(t)} \left(\rho (\mathbf{g} \cdot \mathbf{v} + r) + \nabla \cdot (\mathbf{T} \mathbf{v}) - \nabla \cdot \mathbf{q}\right) \, dV.
\]  

(2.17)
In thermodynamic problems, it is common to use the specific enthalpy \( h := u + \frac{v}{\rho} \), and we will also use this variable later as input for the program REFPROP. Inserting this in the left-hand side of the conservation of energy equation (2.21) gives

\[
\frac{\partial}{\partial t} \left( \rho(u + \frac{1}{2}v \cdot v) \right) + \nabla \cdot (\rho(u + \frac{1}{2}v \cdot v)v) = \rho(g \cdot v + r) + \nabla \cdot (\mathbf{T} v) - \nabla \cdot \mathbf{q}.
\] (2.18)

The left-hand side can be simplified by the product rule and the continuity equation (2.22):

\[
\frac{\partial}{\partial t} \left( u + \frac{1}{2}v \cdot v \right) + \nabla \cdot \left( \rho u + \frac{1}{2}v \cdot v \right)
\]

\[
= \rho \frac{\partial}{\partial t} \left( u + \frac{1}{2}v \cdot v \right) + (u + \frac{1}{2}v \cdot v) \frac{\partial \rho}{\partial t} + (u + \frac{1}{2}v \cdot v) \nabla \cdot (\rho v) + \rho v \cdot \nabla (u + \frac{1}{2}v \cdot v)
\]

\[
= \rho \frac{\partial}{\partial t} \left( u + \frac{1}{2}v \cdot v \right) + \rho v \cdot \nabla \left( u + \frac{1}{2}v \cdot v \right).
\] (2.19)

Recall that \( \mathbf{T} = -p \mathbf{I} + \tau \). If we now take the inner product of the momentum equation (2.15) with \( v \), we find

\[
\rho \frac{\partial v}{\partial t} \cdot v + \rho (v \cdot \nabla) v = \rho g \cdot v - \nabla p \cdot v + (\nabla \cdot \tau) \cdot v
\]

\[
= \rho g \cdot v + (\nabla \cdot \mathbf{T}) \cdot v.
\] (2.20)

We can rewrite the left-hand side of (2.20) as follows:

\[
\rho \frac{\partial v}{\partial t} \cdot v + \rho (v \cdot \nabla) v = \rho \frac{\partial}{\partial t} \left( \frac{1}{2}v \cdot v \right) + \rho v \cdot \nabla \left( \frac{1}{2}v \cdot v \right),
\] (2.21)

and by combining equations (2.20) and (2.21), we obtain

\[
\rho \frac{\partial}{\partial t} \left( \frac{1}{2}v \cdot v \right) + \rho v \cdot \nabla \left( \frac{1}{2}v \cdot v \right) = \rho g \cdot v - \mathbf{T} : \nabla v + \nabla \cdot (\mathbf{T} v).
\] (2.22)

Here, \( \mathbf{T} : \nabla v \) is defined by the Frobenius inner product (or double inner product)

\[
\mathbf{T} : \nabla v := \sum_{i=1}^{3} \sum_{j=1}^{3} T_{ij} \frac{\partial v_i}{\partial x_j}.
\] (2.23)

If we finally subtract equation (2.22) from the energy equation (2.18), we end up with

\[
\frac{\partial}{\partial t} (\rho u) + \nabla \cdot (\rho u v) = \mathbf{T} : \nabla v - \nabla \cdot \mathbf{q} + \rho r.
\] (2.24)

Energy Conservation in Enthalpy Form

In thermodynamic problems, it is common to use the specific enthalpy \( h := u + \frac{v}{\rho} \), and we will also use this variable later as input for the program REFPROP. Inserting this in the left-hand side of the conservation of energy equation (2.21) gives

\[
\frac{\partial}{\partial t} \left( \rho (h - \frac{v}{\rho}) \right) + \nabla \cdot (v(\rho h - p))
\]

\[
= \rho \frac{\partial}{\partial t} h + h \frac{\partial \rho}{\partial t} - \frac{\partial p}{\partial t} + \rho v \cdot \nabla h + h \nabla \cdot (\rho v) - v \cdot \nabla p - p \nabla \cdot v
\]

\[
= h \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) \right) + \rho \frac{\partial h}{\partial t} + \rho v \cdot \nabla h - \frac{\partial p}{\partial t} - v \cdot \nabla p - p \nabla \cdot v
\]

\[
= h \cdot 0 + \rho \frac{dh}{dt} - \frac{dp}{dt} - p \nabla \cdot v
\]

\[
= \mathbf{T} : \nabla v - \nabla \cdot \mathbf{q} + \rho r.
\] (2.25)
With $\mathbf{T} = -p\mathbf{I} + \tau$, it easily follows that $\mathbf{T} : \nabla \mathbf{v} = -p \nabla \cdot \mathbf{v} + \tau : \nabla \mathbf{v}$. So equation (2.25) can be reduced to an equivalent form of the energy equation (2.24), in convective form:

$$\rho \frac{dh}{dt} = \frac{dp}{dt} - \nabla \cdot \mathbf{q} + \tau : \nabla \mathbf{v} + \rho r.$$  \hspace{1cm} (2.26)

### 2.2 Constitutive Equations

In general, the conservation laws contain more unknowns than equations, so we need some additional relations. Most of these relations depend on material properties, and are given by the constitutive equations. In this section, we will discuss the most important constitutive equations that will be used later.

#### 2.2.1 Stress Tensor

For incompressible Newtonian fluids, the viscous stress tensor $\tau$ is defined as [24]:

$$\tau = 2\mu \mathbf{D},$$  \hspace{1cm} (2.27)

where $\mu$ is the dynamic viscosity, and $\mathbf{D}$ is the deformation velocity tensor given by

$$\mathbf{D} = \frac{1}{2} \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right).$$  \hspace{1cm} (2.28)

Besides the dynamic viscosity $\mu$, we will also use the kinematic viscosity $\nu$, with

$$\nu = \frac{\mu}{\rho}.$$  \hspace{1cm} (2.29)

#### 2.2.2 Equations of State

In the case of a compressible fluid, assume a gas, we need an equation describing its state, i.e. a relation between properties as temperature, pressure and volume for the matter. Such a relation is usually called an equation of state, and one of the best known is the ideal gas law:

$$pV = nRT,$$  \hspace{1cm} (2.30)

where $p$ is the pressure, $V$ the volume, $n$ the number of moles of the substance, $R$ the molar gas constant ($R \approx 8.314 \text{ J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$) and $T$ the absolute temperature. Equivalently, this is written as

$$p = \rho RT,$$  \hspace{1cm} (2.31)

where $\mathcal{R}$ is the specific gas constant with dimension $[\mathcal{R}] = \text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$. If we assume that phase transitions or chemical reactions do not occur and that the temperature of the gas remains constant, then we are left with Boyle’s law:

$$pV = C$$  \hspace{1cm} (2.32)
for some constant $C$.

Although the ideal gas law is a good approximation in many cases, more sophisticated equations of state are often used. We will briefly discuss three important models that are used by the program REFPROP [18], see Appendix B. One of these is based on the Benedict-Webb-Rubin (BWR) equation of state:

$$p = \rho_{mol} RT + \left( B_0 RT - A_0 - \frac{C_0}{T^2} \right) \rho_{mol}^2 + (bRT - a) \rho_{mol}^3 + \alpha \rho_{mol}^6 + \frac{c \rho_{mol}^3}{T^2} (1 + \gamma \rho_{mol}^2) \exp(-\gamma \rho^2),$$

with $\rho_{mol}$ the molar density ($[\rho_{mol}] = \text{kg} \cdot \text{mol}^{-1}$) and $a, b, c, A_0, B_0, C_0, \alpha$ and $\gamma$ constants. The development of this equation is originated by a study of thermodynamic properties of light hydrocarbons (methane, ethane, propane, n-butane) and their mixtures [10]. REFPROP makes use of an extended version of this equation.

Other equations of state make explicit use of the Helmholtz energy $A$, given by

$$A = U - TS,$$

where $U$ is the internal energy and $S$ the entropy of the system [11].

Finally, REFPROP makes use of an extended corresponding states (ECS) model. This model is based on the principle of corresponding states for which we introduce the reduced variables of a gas:

$$p_r = \frac{p}{p_c}, \quad V_r = \frac{V_{mol}}{V_c}, \quad T_r = \frac{T}{T_c},$$

where $V_{mol}$ is the molar volume $V$, and the subscript $c$ corresponds to the critical constants, i.e. the pressure, molar volume and temperature at the critical point of a substance. The principle of corresponding states is that real gases at equal reduced volume and reduced temperature exert the same reduced pressure [7].

### 2.2.3 Heat Flux

A convenient way of modeling a heat flux through a conductor is Fourier’s Law of thermal conduction:

$$\mathbf{q} = -\kappa \nabla T,$$

where $\mathbf{q}$ denotes the heat flux per unit area, $[\mathbf{q}] = \text{W} \cdot \text{m}^{-2}$. The material or thermal conductivity $\kappa$ can depend on temperature or direction and has dimension $[\kappa] = \text{W} \cdot \text{m}^{-1} \text{K}^{-1}$.

### 2.3 Initial Conditions and Boundary Conditions

In order to find unique solutions to our differential equations, we need initial conditions and boundary conditions. This means that we assume known values of variables or derivatives for $t = 0$ (initial conditions) and for the boundary of the domain of interest (boundary conditions). Examples of boundary conditions are prescribed values of temperature or pressure at the boundary. Useful conditions for particular problems will be specified in the next chapters.
Chapter 3

Numerical Methods for Initial Value Problems

3.1 Initial Value Problems

We will use several numerical methods to solve our initial value problems (IVPs). Some of these are implemented manually, but we will also make use of standard Matlab functions. The reasons to choose certain methods will be explained later on. In this chapter, we will describe the methods in a general way. For a more extensive overview we refer to [6, 20, 26, 29]. To start with, consider the following IVP:

\[
\begin{align*}
\frac{du}{dt} &= f(u, v, t), \\
0 &= g(u, v, t), \\
u(0) &= u_0, \\
v(0) &= v_0.
\end{align*}
\]

Here, \(u \in \mathbb{R}^{m_1}\), \(v \in \mathbb{R}^{m_2}\), \(f : \mathbb{R}^{m_1+m_2+1} \to \mathbb{R}^{m_1}\) and \(g : \mathbb{R}^{m_1+m_2+1} \to \mathbb{R}^{m_2}\). The functions \(f(u, v, t)\) and \(g(u, v, t)\) may contain derivatives with respect to spatial variables (but not with respect to \(t\)). Let

\[
w = \begin{pmatrix} u \\ v \end{pmatrix},
\]

and let the approximation of \(w(t^n)\) at time \(t^n\) be given by \(w^n\).

3.2 Differential-Algebraic Equations

In the case that \(m_2 \geq 1\), the problem (IVP) is called a system of differential algebraic equations (DAEs). The difference with a system of ordinary differential equations (ODEs) is that some variables occur in the equations without their derivatives with respect to \(t\). These are given in the vector \(v\) in (IVP). A DAE can be written as

\[
M(t)\dot{w} = h(w, t),
\]

(3.2)
where $\mathbf{\dot{w}} = \frac{d\mathbf{w}}{dt}$ and $M(t)$ is called the mass matrix, which is singular. A DAE is called semi-explicit if it is of the form of equation (3.1). In this way, DAEs are divided into differential equations and algebraic equations. All other DAEs of the general form $f(\mathbf{u}, \mathbf{\dot{u}}, t) = 0$ are called fully-implicit DAEs [32].

Sometimes, a DAE can be reduced to an ODE by simple substitution, but in general this is not that easy. However, most of the standard ODE solvers cannot handle DAEs. One of the ways to reduce the system is the algorithm of Pantelides [25]. In Matlab, the functions `ode23t` and `ode15s` are the only ODE solvers that can handle DAEs [29, 30, 31].

### 3.3 Convergence and Stability

Two important properties of a numerical method are convergence and stability. Convergence means that taking smaller step sizes gives a better approximation of the solution, so that the numerical solution approaches the exact solution as the step size goes to zero. In this context, also the order of convergence is of interest. If $u_h$ is the numerical approximation with typical step size $h$, then the method is said to converge with order $p$ if

$$
\|u_h - u\| = O(h^p),
$$

where $u$ is the exact solution. If the exact solution is not known, one can estimate $p$ by calculating numerical solutions $u_{h_k}$ for several step sizes $h_k$, where we assume that $h_k = h_0 \gamma^k$, for $k = 1, 2, 3, \ldots$ and with $0 < \gamma < 1$. The order $p$ is then given by

$$
p = \frac{\log(\frac{u_{h_k}-u_{h_{k+1}}}{u_{h_{k+1}}-u_{h_k}})}{\log(\gamma)}. \quad (3.4)
$$

A method is called stable if errors in the numerical approximations do not grow in time. Of course, stability depends on the problem of interest. Some methods are unconditionally stable, while for other methods stability conditions can be derived.

### 3.3.1 Error Tolerance

To control the accuracy of a numerical solution, it is convenient to use a relative tolerance. Roughly speaking, this indicates the number of correct digits of the solution, i.e. a solution with relative tolerance $10^{-3}$ is correct in the first three digits. Next to the relative tolerance, the absolute tolerance is often used, which is a bound for the absolute error. In Matlab, the error is normally said to be small enough if it satisfies the following:

$$
|e(i)| \leq \max\{\text{RelTol} \cdot |y(i)|, \text{AbsTol}(i)\}, \quad (3.5)
$$

where $e(i)$ is the $i$-th component of the error and $y(i)$ is the $i$-th component of the solution [31].
3.4 Time Integration Methods

Many methods to solve differential equations numerically are in the Runge-Kutta family. Suppose we have an ODE with corresponding initial condition of the following form
\[
\begin{align*}
\text{(ODE)} \quad \frac{du}{dt} &= f(u,t), \\
u(0) &= u_0,
\end{align*}
\]
and let the numerical approximation of \( u \) at time \( t^n \) be denoted by \( u^n \). Then the general Runge-Kutta method with \( m \) stages using a time step of \( \Delta t \), is given by
\[
\frac{1}{\Delta t} \left( u^{n+1} - u^n \right) = \sum_{i=1}^{m} b_i k^n_i ,
\]
with
\[
\sum_{i=1}^{m} b_i = 1, \quad c_i = \sum_{j=1}^{m} a_{ij} \quad (i = 1, 2, \ldots, m).
\]

It is convenient to write the coefficients \( a_{ij}, b_i \) and \( c_i \) in a so-called Butcher tableau, named after the New Zealand mathematician John C. Butcher:
\[
\begin{array}{c|ccc}
& a_{11} & a_{12} & \cdots & a_{1m} \\
& a_{21} & a_{22} & \cdots & a_{2m} \\
& \vdots & \vdots & \ddots & \vdots \\
& a_{m1} & a_{m2} & \cdots & a_{mm} \\
\hline
& b_1 & b_2 & \cdots & b_m \\
\end{array} = \begin{bmatrix} c^T \end{bmatrix} \begin{bmatrix} A \\ b^T \end{bmatrix}.
\]

3.4.1 Euler Methods

The simplest method of this form is the explicit Euler method, also called forward Euler method, given by
\[
\frac{1}{\Delta t} \left( u^{n+1} - u^n \right) = f(u^n, t^n).
\]
It is called an explicit method, because at any time \( t^{n+1} \), given \( u^n \), the new approximate value of \( u, \) \( u^{n+1} \), can be calculated directly. For an explicit method, the coefficient matrix \( A \) in the Butcher tableau (3.8) is always lower triangular.

The backward method is analogous to the forward method, and relates \( u^n \) to \( u^{n+1} \) and \( f(u^{n+1}, t^{n+1}) \) by doing a backward step:
\[
\frac{1}{\Delta t} \left( u^{n+1} - u^n \right) = f(u^{n+1}, t^{n+1}).
\]
This is called an implicit method, since \( u^{n+1} \) cannot be calculated directly from the given information at \( t^n \). One first needs to solve an equation to find \( u^{n+1} \). Both methods are first-order accurate.
3.4.2 θ-Method

To combine the advantages of the forward and backward Euler method, it is convenient to use a linear combination of both methods. This is accomplished by the so-called θ method:

\[
\frac{1}{\Delta t} (u^{n+1} - u^n) = (1 - \theta)f(u^n, t^n) + \theta f(u^{n+1}, t^{n+1}), \quad \theta \in [0, 1].
\] (3.11)

As the θ-method is a linear combination of the two first-order Euler methods, it will in general be first order as well. The special case \(\theta = \frac{1}{2}\) is second order and is called the trapezoidal rule, which is used by the Matlab function \texttt{ode23t}:

\[
\frac{1}{\Delta t} (u^{n+1} - u^n) = \frac{1}{2}f(u^n, t^n) + \frac{1}{2} f(u^{n+1}, t^{n+1}).
\] (3.12)

3.4.3 Dormand-Prince Method

Matlab’s advised “first try” ODE solver \texttt{ode45} is based on the Dormand-Prince method, also called the Dormand-Prince pair, which is given by the following Butcher tableau [13]:

\[
\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{3} & 1 & & & & & \\
\frac{2}{5} & & 3 & 9 & 3 & 10 & 32 \\
\frac{3}{5} & & 44 & -56 & 32 & 9 & 24 \\
\frac{4}{5} & & 6972 & 25360 & 64448 & 212 & 712 \\
\frac{5}{6} & & 6096 & 2187 & 5127 & 176 & 5406 \\
1 & & 35 & 384 & 0 & 500 & 125 & 517 \\
1 & & 35 & 384 & 0 & 500 & 125 & 6784 & 81 \\
\hline
5129 & 0 & 7571 & 393 & 92097 & 187 & 40 \\
57609 & 0 & 16069 & 640 & 339200 & 2100 & 0 \\
384 & 0 & 500 & 125 & 2187 & 11 & 0 \\
384 & 0 & 1113 & 192 & 6784 & 81 & 0 \\
\end{array}
\] (3.13)

The two rows below the horizontal line are the \(b_i\) coefficients for the fifth-order (upper one) and fourth-order accurate schemes. The difference between the two solutions is regarded as the error of the fourth-order solution. The error estimate is then used to adapt the step sizes of the integration.

3.4.4 Backward and Numerical Differentiation

Finally, we will also make use of Matlab’s \texttt{ode15s} method in this report. This method is a variable order solver, i.e. the order of accuracy (between 1 and 5 for in this case) of the approximation at a certain time depends on the behavior of the numerical solution. It normally uses numerical differentiation formulas (NDFs), which are a modification of backward differentiation formulas (BDFs). In standard form, the BDF of order \(k\) is given by

\[
\sum_{j=0}^{k} \alpha_j u^{n+j-k+1} = \Delta t \beta_k f(u^{n+1}, t^{n+1}),
\] (3.14)

where \(\alpha_j\) and \(\beta_k\) are certain coefficients. For more details we refer to [6, 29].
Chapter 4

Two-Tank System with Constant Temperature without Upper Pipe

4.1 Problem Description

Consider a system existing of two upright standing cylindrical tanks, connected by two pipes, see Figure 4.1. The inner diameters of tank 1 and tank 2 are $D_1$ and $D_2$ respectively, and the inner diameters of the pipes are $D_L$ for the lower pipe and $D_U$ for the upper pipe. The tanks are filled with a combination of a liquid and a gas, with the gas located on top of the liquid (no mixing occurs). The initial levels of the liquid in the tanks are given by $H_{1,0}$ and $H_{2,0}$. The lower pipes are connected to tank 1 and tank 2 at heights $H_{L1}$ and $H_{L2}$, respectively, and the upper pipes are connected to the tanks at the tops of the tanks, so at heights $H_{T1}$ and $H_{T2}$. The heights of the pipes are given from the bottom of the tank to the center of the pipe at the pipe-tank connection. The tanks can be connected or disconnected by two valves in the pipes. We assume that the liquid is incompressible with density $\rho_l$ and the (compressible) gas has a density $\rho_g$ with $\rho_g \ll \rho_l$. We will discuss two possible scenarios in this chapter:

1. The tanks are open at the top. In this case, the gas in the tanks is simply air, and the pressure in the gas equals atmospheric pressure, $p_{atm}$. See the sketch in Figure 4.1.

2. The tanks are closed at the top. In this case, the gas is not necessarily air, and the initial pressures in the gas are $p_{1,0}$ and $p_{2,0}$ (not necessarily equal to the atmospheric pressure) for tank 1 and tank 2 respectively. See the sketch in Figure 4.7.

In both cases, we will consider quasi-steady flow. This means that we do not have to take inertial effects into account. To start with, we neglect the upper pipe, and we assume that liquid flows from tank 1 to tank 2. Throughout this chapter and the next chapter, we assume that temperature is constant and uniform in the system. We are interested in determining the average velocity $v$ in the lower pipe.

\footnote{When we are talking about pressure in this report, we mean absolute pressure.}
Figure 4.1: Possible initial situation of the two-tank system. The liquid levels in the tanks, relative to the tank bottom, are \( H_{1,0} \) and \( H_{2,0} \), and the heights of the pipes are \( H_{1L}, H_{2L}, H_{T1} \) and \( H_{T2} \), from bottom to the central axis of the pipe at the pipe-tank connection.

4.2 One-Dimensional Model

Before we consider specific initial situations or system properties (different initial liquid heights, open tanks or closed tanks, etc.), we will first discuss the general model for the two tank system. At time \( t = 0 \), the valve in the lower pipe is open and \( H_1 \) and \( H_2 \) are known. We assume that the flow is developed (no inertial effects due to start up are considered), incompressible and that viscous effects are negligible. Furthermore, we assume that no heat transfers occur. Therefore, we start from the continuity equation for incompressible fluids (2.6) and the momentum equation (2.15), in simplified form:

\[
\nabla \cdot \mathbf{v} = 0 \quad (4.1a)
\]

\[
\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \rho \mathbf{g} - \nabla p. \quad (4.1b)
\]

4.2.1 Bernoulli’s Equation

Derivation

The flow is assumed to be quasi-steady, so \( \frac{\partial \mathbf{v}}{\partial t} \) is negligible and equation (4.1b) becomes

\[
\rho (\mathbf{v} \cdot \nabla \mathbf{v}) = \rho \mathbf{g} - \nabla p. \quad (4.2)
\]

Using \( \mathbf{v} \cdot \nabla \mathbf{v} = \frac{1}{2} \nabla (\mathbf{v} \cdot \mathbf{v}) + (\nabla \times \mathbf{v}) \times \mathbf{v} \) and \( \mathbf{g} = (0, 0, -g) \), equation (4.2) can be rewritten in the following form:

\[
\rho \left( \frac{1}{2} \nabla (\mathbf{v} \cdot \mathbf{v}) + (\nabla \times \mathbf{v}) \times \mathbf{v} + \nabla (gz) + \frac{\nabla p}{\rho} \right) = 0. \quad (4.3)
\]
Note that $(\nabla \times \mathbf{v}) \times \mathbf{v}$ is perpendicular to $\mathbf{v}$, and hence its inner product with $\mathbf{v}$ is zero. Taking the inner product of equation (4.3) with $\mathbf{v}$ gives a special case of equation (2.22):

$$
\mathbf{v} \cdot \rho \left( \frac{1}{2} \nabla (\mathbf{v} \cdot \mathbf{v}) + (\nabla \times \mathbf{v}) \times \mathbf{v} + \nabla (g_z + \frac{\nabla p}{\rho}) \right) = \mathbf{v} \cdot \nabla \left( \frac{1}{2} \rho (\mathbf{v} \cdot \mathbf{v}) + \rho g_z + p \right) = 0.
$$

From this last expression, we conclude that along any streamline we have

$$
\frac{1}{2} \rho (\mathbf{v} \cdot \mathbf{v}) + \rho g_z + p = C, \quad (4.4)
$$

for some constant $C$. Clearly, for $\mathbf{v} = 0$ the pressures equal hydrostatic pressures. Equation (4.4) is known as Bernoulli’s equation. Since this equation holds along any streamline, it is useful to write

$$
\frac{1}{2} \rho v_a^2 + \rho g z_a + p_a = \frac{1}{2} \rho v_b^2 + \rho g z_b + p_b, \quad (4.5)
$$

which holds for any two points $a$ and $b$ along a streamline, where the subscripts indicate the position. Specific applications to the two-tank system are given in Section 4.3.

**Energy Loss**

Equation (4.5) can also be written as

$$
\frac{v_a^2}{2g} + z_a + \frac{p_a}{\rho g} = \frac{v_b^2}{2g} + z_b + \frac{p_b}{\rho g}, \quad (4.6)
$$

for any two points $a$ and $b$ along a streamline. The terms $\frac{v^2}{2g}$, $z$ and $\frac{p}{\rho g}$ are called the velocity head, elevation and pressure head respectively. In practice however, these losses can be very significant, so it is important to extend the model with possible energy losses. In systems consisting of several components (pipelines, tanks, valves, etc.), it is common to introduce these energy losses per component. We will only add a loss term for the pipeline (including possible valves). We therefore add a term $h_f$, the head loss, which can be divided in “general losses”, associated with energy loss per length of straight pipe and “local losses”, associated with energy loss caused by bends, valves, etc. The general head losses $h_{fg}$ can be described by the Darcy-Weisbach equation [11]:

$$
h_{fg} = f \frac{L}{D} \frac{v^2}{2g}, \quad (4.7)
$$

where $f$ is the Darcy-Weisbach friction factor, which can be found from empirical data, $L$ is the length of the pipe and $D$ is the diameter of the pipe. So energy losses are proportional to the velocity head $\frac{v^2}{2g}$. We will neglect energy losses due to friction and turbulence in the tanks, since we assume the velocity in the tanks to be very small compared to the fluid velocity in the pipes.

The minor losses should be estimated from physical experiments and we assume that these losses are proportional to $\frac{v^2}{2g}$. The minor losses are then equivalent with the loss over a certain length of straight pipe. The total head loss of the pipeline is written as

$$
h_f = \alpha \frac{v^2}{2g}, \quad (4.8)
$$
where \( \alpha \) is some constant depending on the physical and geometrical parameters of the pipe and the fluid. Adding head loss to the most downstream point in the Bernoulli equation yields the so-called extended Bernoulli equation [28]

\[
\frac{v_a^2}{2g} + z_a + \frac{p_a}{\rho g} = \frac{v_b^2}{2g} + z_b + \frac{p_b}{\rho g} + h_f
\]

\[
= (1 + \alpha) \frac{v_b^2}{2g} + z_b + \frac{p_b}{\rho g}.
\] (4.9)

Clearly, \( \alpha \to 0 \) corresponds to the limit case of no friction or local losses, and \( \alpha \to \infty \) corresponds to “infinite” friction or a closed valve, i.e. \( v_b = 0 \).

4.3 Solution

In this section, we will use the model described above to find a solution for several scenarios for the two-tank system, for both the open and closed system. We will distinguish between four initial situations:

- Open system
  - The tanks are open at the top and the initial liquid level in tank 2 is at equal height or above the center of the pipe at the connection to tank 2: \( H_{2,0} \geq H_{L2} \), see Figure 4.2(a).
  - The tanks are open at the top and the initial liquid level in tank 2 is below the center of the pipe at the connection to tank 2: \( H_{2,0} < H_{L2} \), see Figure 4.2(b).

- Closed system
  - The tanks are closed at the top and the initial liquid level in tank 2 is at equal height or above the center of the pipe at the connection to tank 2: \( H_{2,0} \geq H_{L2} \), see Figure 4.2(c).
  - The tanks are closed at the top and the initial liquid level in tank 2 is below the center of the pipe at the connection to tank 2: \( H_{2,0} < H_{L2} \), see Figure 4.2(d).

4.3.1 Open Two-Tank System

\( H_{2,0} \geq H_{L2} \)

Consider the open two-tank system with the initial liquid level in tank 2 above the center of the pipe connection at tank 2, \( H_{2,0} \geq H_{L2} \), see Figure 4.2(a). Point \( a \) is chosen at the free surface in tank 1, at \( z = H_1 - H_{L1} \), point \( b \) is chosen at the right end of the lower pipe, but just inside the pipe at \( z = H_{L2} - H_{L1} \), see Figure 4.3. Then we have by the extended Bernoulli equation [1.9]

\[
\frac{v_a^2}{2} + g(H_1 - H_{L1}) + \frac{p_{atm}}{\rho i} = \frac{v_b^2}{2} + g(H_{L2} - H_{L1}) + g(H_2 - H_{L2}) + \frac{p_{atm}}{\rho i}.
\] (4.10)
4.3 Solution

Figure 4.2: We distinguish between four initial situations: open tanks and $H_{2,0} \geq H_{L2}$ (a), open tanks and $H_{2,0} < H_{L2}$ (b), closed tanks and $H_{2,0} \geq H_{L2}$ (c) and closed tanks and $H_{2,0} < H_{L2}$ (d).

Let $A_1$ be the horizontal cross-sectional area of tank 1, and let $A_L$ be the cross-sectional area of the lower pipe. Then we have by mass conservation that

$$A_1 v_a = -A_1 \frac{dH_1}{dt} = A_L v_b = A_2 \frac{dH_2}{dt}. \quad (4.11)$$

Since $A_1 \gg A_L$, we have that $v_a \ll v_b$, and we therefore neglect the term with $v_a^2$ in equation (4.10).

Figure 4.3: Two points $a$ and $b$ along a streamline are defined in Bernoulli’s equation. Point $a$ is located at the surface of the liquid in tank 1 at $z = H_1 - H_{L1}$, and point $b$ is located at the right end of the lower pipe, but just inside the pipe at $z = H_{L2} - H_{L1}$. 

\[ \text{Figure 4.3: Two points } a \text{ and } b \text{ along a streamline are defined in Bernoulli's equation.} \]

\[ \text{Point } a \text{ is located at the surface of the liquid in tank 1 at } z = H_1 - H_{L1}, \text{ and point } b \text{ is located at the right end of the lower pipe, but just inside the pipe at } z = H_{L2} - H_{L1}. \]
We then find

\[(1 + \alpha) \frac{v_b^2}{2} = g(H_1 - H_2)\].

Clearly, \(v_b = 0 \) for \(H_1 = H_2\), \(v_b > 0 \) if \(H_1 > H_2\) and \(v_b < 0 \) if \(H_1 < H_2\). With \(v_b \geq 0\), this implies

\[v_b = \sqrt{\frac{2g}{1+\alpha}(H_1 - H_2)}, \quad (4.12)\]

which is Torricelli’s law for \(\alpha = 0\). Since \(v_a = -\frac{dH_1}{dt}\), we have by equations (4.11) and (4.12) that

\[-\frac{dH_1}{dt} = v_a = \frac{A_L}{A_1}v_b = \beta \sqrt{H_1 - H_2}, \quad (4.13)\]

where \(\beta = \frac{A_L}{A_1} \sqrt{\frac{2g}{1+\alpha}}\). Equations (4.11) and (4.13) form a system of two ODEs:

\[
\begin{pmatrix}
-1 & 0 \\
-\frac{A_2}{A_1} & 1
\end{pmatrix}
\begin{pmatrix}
\frac{d}{dt}(H_1) \\
\frac{d}{dt}(H_2)
\end{pmatrix}
= \begin{pmatrix}
\beta \sqrt{H_1 - H_2} \\
0
\end{pmatrix}.
\]

(4.14)

We can express \(H_2\) as a function of \(H_1\) by mass conservation:

\[H_2 = \frac{V_{TOT} - A_1H_1 - A_{LL}}{A_2}, \quad (4.15)\]

where \(V_{TOT}\) denotes the total liquid volume in the system, which can be calculated from the initial liquid heights:

\[V_{TOT} = A_1H_{1,0} + A_2H_{2,0} + A_{LL}.\]

Eliminating \(H_2\), equation (4.13) can be written as

\[-\frac{dH_1}{dt} = \beta \sqrt{H_1 - \frac{V_{TOT} - A_1H_1 - A_{LL}}{A_2}}
= \beta \sqrt{\frac{(A_2 + A_1)H_1 + A_{LL} - V_{TOT}}{A_2}}
= \beta \sqrt{\frac{(A_2 + A_1)H_1 - A_1H_{1,0} - A_2H_{2,0}}{A_2}}.
\]

(4.16)

Together with the initial condition \(H_1(0) = H_{1,0}\), this ODE has the following solution:

\[H_1 = \frac{(A_2 + A_1)\beta^2t^2}{4A_2} - \beta t \sqrt{H_{1,0} - H_{2,0}} + H_{1,0}. \quad (4.17)\]

At \(t = 0\), equation (4.16) becomes

\[-\frac{dH_1}{dt}(0) = \beta \sqrt{H_{1,0} - H_{2,0}},\]

so \(\frac{dH_1}{dt}(0) = 0\) in the case that the initial liquid heights are equal. Equation (4.17) gives \(H_1(0) = H_{1,0}\). Since (4.17) represents a parabola, we should note that the equation only holds as long as equilibrium is not reached. Let us denote the time at which equilibrium is reached by \(t_e\), \(H_1(t_e) = H_2(t_e)\). Then (4.17) is only valid for \(0 \leq t \leq t_e\).
Remark. Note that in the equilibrium state the liquid levels in the tanks are not necessarily equal. For example, if the pipe connection to tank 1 is much higher than the pipe connection to tank 2, and the initial liquid levels are just above the pipe connections, the liquid in tank 1 will reach $H_{L1}$ before $H_{1} = H_{2}$, see Figure 4.4. However, we will consider the “normal” scenario in which the liquid levels become equal in the calculation of $t_{e}$. We then have $H_{1} = H_{2}$ and $\frac{dH}{dt} = 0$ in equation (4.13) for $t \geq t_{e}$. We will now calculate the condition for reaching an equilibrium state with equal liquid heights in both tanks. We will assume that the pipe is always totally filled with liquid. In an equilibrium state with equal liquid heights, $H_{1}(t_{e})$ (and hence also $H_{2}(t_{e})$) should be larger than the maximum of $H_{L1}$ and $H_{L2}$, if we ignore the trivial scenarios $H_{1,0} = H_{2,0}$ with $H_{1,0}$ and $H_{2,0}$ below the highest pipe connection $\max\{H_{L1}, H_{L2}\}$; for example $H_{1,0} = H_{2,0} = 0$ is such a trivial scenario. So

$$A_{1}H_{1,0} + A_{2}H_{2,0} = A_{1}H_{1}(t_{e}) + A_{2}H_{2}(t_{e}) = (A_{1} + A_{2})H_{1}(t_{e}) \geq (A_{1} + A_{2})\max\{H_{L1}, H_{L2}\},$$

which is equivalent to saying that the weighted average of the initial liquid heights should be at least $\max\{H_{L1}, H_{L2}\}$.

Calculation of $t_{e}$

For a general quadratic function $f(x) = ax^{2} + bx + c$, the top is given by $x_{\text{top}} = -\frac{b}{2a}$, $f(x_{\text{top}}) = c - \frac{b^2}{4a}$. Only at the top of (4.17) we have $\frac{dH}{dt} = 0$, so the time at which equilibrium is reached
is given by
\[ t_c = \frac{\beta \sqrt{H_{1,0} - H_{2,0}}}{2(A_2 + A_1)\beta^2} \]
\[ = 2A_2 \frac{\sqrt{H_{1,0} - H_{2,0}}}{(A_2 + A_1)\beta}. \quad (4.19) \]

The corresponding height of the liquid in tank 1 is
\[ H_1(t_c) = H_{1,0} - \frac{A_2}{A_1 + A_2}(H_{1,0} - H_{2,0}) \]
\[ = \frac{A_1H_{1,0} + A_2H_{2,0}}{A_1 + A_2}. \quad (4.20) \]

This last expression is in correspondence with the mass conservation in equation (4.15) for \( H_1 = H_2 \), as one would expect.

**\( H_{2,0} < H_{L2} \)**

For an initial liquid height in tank 2 below the pipe-tank connection, we assume that the pipe is totally filled with liquid and that the flow is developed: no start-up effects due to inertia are taken into account. We take again two points along the streamline. Point \( a \) is located at the transition between liquid and gas in tank 1, at \( z = H_1 - H_{L1} \). Point \( b \) is located at the right end of the lower pipe, but just inside the pipe at \( z = H_{L1} - H_{L2} \), like in the case that \( H_{2,0} \leq H_{L2} \), see Figure 4.3. As long as \( H_2 \leq H_{L2} \), we have
\[ \frac{v_a^2}{2} + g(H_1 - H_{L1}) + \frac{p_{atm}}{\rho_l} = (1 + \alpha)\frac{v_b^2}{2} + g(H_{L2} - H_{L1}) + \frac{p_{atm}}{\rho_l}. \quad (4.21) \]

Analogous to the previous case, the solution is given by
\[ v_b = \sqrt{\frac{2g}{1+\alpha}(H_1 - H_{L2})}, \quad (4.22) \]

which holds as long as \( H_1 \geq \max\{H_{L1}, H_{L2}\} \) and \( H_2 \leq H_{L2} \). For \( \alpha = 0 \), this is known as Torricelli’s law.

Since \( v_a = -\frac{dH_1}{dt} \), we have by equations (4.11) and (4.22) that
\[ -\frac{dH_1}{dt} = v_a = \frac{A_\ell}{A_1}v_b = \beta\sqrt{H_1 - H_{L2}}, \quad (4.23) \]

where \( \beta = \frac{A_\ell}{A_1} \sqrt{\frac{2g}{1+\alpha}} \).

Together with the initial condition \( H_1(0) = H_{1,0} \), this ODE has the following solution:
\[ H_1 = \frac{\beta^2t^2}{4} - \beta t \sqrt{H_{1,0} - H_{L2}} + H_{1,0}. \quad (4.24) \]
Two points $a$ and $b$ along the streamline are defined in Bernoulli’s equation. Point $a$ is located at the surface of the liquid in tank 1 at $z = H_1 - H_{L1}$, and point $b$ is located at the right end of the lower pipe at $z = H_{L2}$.

We can calculate the time $t_{L2}$ at which the liquid level in tank 2 reaches the center of the pipe at the connection to tank 2, $H_2 = H_{L2}$. By mass conservation, we then have

$$A_1(H_{1,0} - H_1(t_{L2})) = A_2(H_{L2} - H_{2,0}),$$

so

$$H_1(t_{L2}) = H_{1,0} - \frac{A_2(H_{L2} - H_{2,0})}{A_1}. \tag{4.25}$$

Substitution of (4.25) in the solution (4.24) of the ODE gives

$$H_{1,0} - \frac{A_2(H_{L2} - H_{2,0})}{A_1} = \frac{\beta^2 t^2_{L2}}{4} - \beta t_{L2} \sqrt{H_{1,0} - H_{L2} + H_{1,0}},$$

which has solution

$$t_{L2} = \frac{2 \left( \sqrt{H_{1,0} - H_{L2}} - \sqrt{H_{1,0} - H_{L2} - \frac{A_2(H_{L2} - H_{2,0})}{A_1}} \right)}{\beta}. \tag{4.26}$$

At $t = t_{L2}$, we switch from solution (4.24) to the solution we found for $H_{2,0} > H_{L2}$ (4.17), with continuity condition $H_2(t_{L2}) = H_{L2}$. So for $H_{2,0} \lessgtr H_{L2}$, we finally find

$$H_1 = \begin{cases} \frac{\beta t^2}{4} - \beta t \sqrt{H_{1,0} - H_{L2} + H_{1,0}} & \text{if } 0 \leq H_2 \leq H_{L2}; \\ \frac{1}{4} \left(1 + \frac{A_1}{A_2}\right) \beta^2 (t - t_{L2})^2 + H_1(t_{L2}) - \beta (t - t_{L2}) \sqrt{H_1(t_{L2}) - H_{L2}} & \text{if } H_{L2} < H_2. \end{cases} \tag{4.27}$$

The solution is a piecewise polynomial of degree 2. The “switch” from the first polynomial to the other one occurs at $t = t_{L2}$, see equation (4.26).
Example

A solution of this problem is given in Figure 4.6, for the geometrical and physical parameters in Table 4.1, for initial liquid heights of $H_{1,0} = 25$ m and $H_{2,0} = 0$ m. The density $\rho_{\text{liq}}$ in this table is the approximate density of LNG at atmospheric pressure and just below boiling point ($T = 110$ K). From Table 4.1 we can calculate $V_{\text{TOT}}$ and $\beta$:

\[
V_{\text{TOT}} = \pi \frac{1}{4} D_1^2 H_{1,0} + \pi \frac{1}{4} D_L^2 L_L \approx 8247 \text{ m}^3,
\]

\[
\beta = \frac{A_L}{A_1} \sqrt{\frac{2g}{1+\alpha}} = \frac{1}{16} \frac{100\pi}{\sqrt{\frac{2.981}{3}}} \approx 1.6 \cdot 10^{-3} \text{ m}^2\text{s}^{-1}.
\]  

(4.28)

The program code is given in Appendix C. After approximately 130 seconds, the liquid level in tank 2 reaches the height of the lower pipe at 1 m. From then on, $p_{\text{out}}$ is not constant anymore and the pressure difference over the pipe decreases more rapidly. After approximately 3000 seconds, equilibrium is reached: $H_1 = H_2$. The value of $H_1$ and $H_2$ in the equilibrium state can be calculated by mass conservation. The pipe is totally filled in the equilibrium state, so we have

\[
V_{\text{TOT}} - A_L L_L = A_1 H_1 + A_2 H_2.
\]

Since $A_1 = A_2$ in this example and $H_1 = H_2$ for $t \to \infty$, we find

\[
H_1 = \frac{V_{\text{TOT}} - A_L L_L}{2 A_1} \approx \frac{8050 - \frac{2000\pi}{16}}{200\pi} \approx 12.5 \text{ m},
\]

as one would expect.

<table>
<thead>
<tr>
<th>$D_1$</th>
<th>20 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_2$</td>
<td>20 m</td>
</tr>
<tr>
<td>$H_{L,1}$</td>
<td>1 m</td>
</tr>
<tr>
<td>$H_{L,2}$</td>
<td>1 m</td>
</tr>
<tr>
<td>$D_L$</td>
<td>0.5 m</td>
</tr>
<tr>
<td>$L_L$</td>
<td>2000 m</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>2</td>
</tr>
<tr>
<td>$\rho_{\text{liq}}$</td>
<td>433 kg·m⁻³</td>
</tr>
<tr>
<td>$H_{1,0}$</td>
<td>25 m</td>
</tr>
<tr>
<td>$H_{2,0}$</td>
<td>0 m</td>
</tr>
</tbody>
</table>

Table 4.1: Parameters corresponding to the solution given by Figure 4.6
Figure 4.6: Figure (a) shows the evolution of the liquid heights and Figure (b) shows the corresponding pressure at $z = 0$ in the tanks for the parameters given in Table 4.1. The liquid level in tank 2 reaches the lower pipe connection at a height of 1 meter after approximately 130 seconds.
4.3.2 Closed Two-Tank System

We now consider tanks that do not have an opening at the top, so they form a closed system. Initially, the liquid levels are given by $H_{1,0}$ and $H_{2,0}$ respectively and the pressure of the gas in the left tank is $p_{1,0}$ and the pressure of the gas in the right tank is $p_{2,0}$. See Figure 4.7. Like in the open system, we assume that the pipe is always totally filled with liquid and that the flow is quasi steady: start-up effects are neglected.

The derivation of the solution is analogous to the open tanks scenario, except that the pressure in the gas is not constant anymore. At any time $t$ the pressure in the gas in tank $i$, $p_i$, can be calculated from Boyle’s law, given the initial pressure and the liquid heights. From equation (2.32), it follows that

$$p_i = p_{i,0} \frac{H_{Ti} - H_{i,0}}{H_{Ti} - H_i},$$  \hspace{1cm} (4.29)

where $H_{Ti}$ denotes the height of tank $i$. An important consequence of the variability of the gas pressure is that the liquid does not necessarily flow from the tank with the highest liquid level to the tank with the lowest liquid level.

![Figure 4.7](image-url)

**Figure 4.7:** Initial situation of the closed two-tank system. The liquid levels in the tanks are $H_{1,0}$ and $H_{2,0}$, the heights of the connections of the pipes to the tanks are $H_{L1}$ and $H_{L2}$ for the lower pipe and $H_{T1}$ and $H_{T2}$ for the upper pipe. The heights of the tanks are $H_{T1}$ and $H_{T2}$. The pressure of the gas in the left tank is $p_{1,0}$ and the pressure of the gas in the right tank is $p_{2,0}$. 
4.3 Solution

$H_{2,0} \geq H_{L2}$

We take two points $a$ and $b$, similar to the open tanks case, see Figure 4.3, and use the extended Bernoulli equation (4.9) to find

$$\frac{v_a^2}{2} + g(H_1 - H_{L1}) + \frac{p_1}{\rho_l} = (1 + \alpha)\frac{v_b^2}{2} + g(H_{L2} - H_{L1}) + g(H_2 - H_{L2}) + \frac{p_2}{\rho_l}, \quad (4.30)$$

which, after substitution of equation (4.29) for $p_1$ and $p_2$ and neglecting $\frac{v_a^2}{2}$, gives

$$v_b = \sqrt{\frac{2}{1 + \alpha} \left\{ g(H_1 - H_2) + \frac{p_{1,0}}{\rho_l} \frac{H_{T1} - H_{1,0}}{H_{T1} - H_1} - \frac{p_{2,0}}{\rho_l} \frac{H_{T2} - H_{2,0}}{H_{T2} - H_2} \right\}}. \quad (4.31)$$

$H_{2,0} < H_{L2}$

In the case that $H_{2,0} < H_{L2}$, the application of the extended Bernoulli equation (4.9) is analogous to the open tanks scenario. We assume again that the pipe will be totally filled with liquid at $t = 0$. Points $a$ and $b$ are chosen as in Figure 4.3. We then find

$$\frac{v_a^2}{2} + g(H_1 - H_{L1}) + \frac{p_1}{\rho_l} = (1 + \alpha)\frac{v_b^2}{2} + g(H_{L2} - H_{L1}) + \frac{p_2}{\rho_l}, \quad (4.32)$$

which, after substitution of equation (4.29) for $p_1$ and $p_2$ and neglecting $\frac{v_a^2}{2}$, gives

$$v_b = \sqrt{\frac{2}{1 + \alpha} \left\{ g(H_1 - H_{L2}) + \frac{p_{1,0}}{\rho_l} \frac{H_{T1} - H_{1,0}}{H_{T1} - H_1} - \frac{p_{2,0}}{\rho_l} \frac{H_{T2} - H_{2,0}}{H_{T2} - H_2} \right\}}. \quad (4.33)$$

Numerical Solution

Using $v_b = \frac{\Delta t}{A_p}v_a$, $\frac{dH_1}{dt} = -v_a$ and mass conservation (4.15), we can write equations (4.31) and (4.33) in the ODE form

$$\frac{dH_1}{dt} = f(H_1). \quad (4.34)$$

Unfortunately, we are unable to find an exact solution for this ODE, due to the complexity of the gas pressure terms. Therefore, we will solve this ODE numerically. To start with, we choose a method that is easy to understand and implement and that is quickly adapted to expansions of our model, namely the explicit Euler method, see Chapter 3:

$$\frac{1}{\Delta t} (H_1^{n+1} - H_1^n) = f(H_1^n). \quad (4.35)$$

Here, $\Delta t$ is the time step of the discretization, $H_1^n = H_1(t^n)$, and $f(H_1^n)$ is the discretized right hand side of (4.34). The implementation of this method is straightforward, see Appendix C.1.2. However, we have to be careful with special situations:
• Liquid can only flow from a tank to the pipe if the liquid level is higher than the pipe-tank connection.

• In contrast to the open tanks system, it is possible that liquid flows from the tank with lower liquid height to the tank with higher liquid height.

Examples with Explicit Euler

The parameters in Table 4.2 are used for the examples in Figures 4.8-4.16. In the cases that $H_{2,0} < H_{L2}$, we assume that the pipe is totally filled with liquid at $t = 0$. Like in the open tanks example, $\beta \approx 1.6 \cdot 10^{-3}$ m$^1$s$^{-1}$, see (4.28).

Numerical solutions are computed using explicit Euler with a step size $\Delta t$ of 1 second. Calculation takes approximately $10^{-3}$ seconds on a 1.86 Ghz processor\(^2\). In each figure, the graph on the left shows the evolution of $H_1$ and $H_2$, and the graph on the right shows the evolution of the pressures in the left and right tank at $z = 0$, $p_{in}$ and $p_{out}$ respectively. Except for special cases, $p_{in}$ and $p_{out}$ become equal for $t \to \infty$. Below we give some other observations on the examples:

• Opposite to the open tanks case, the liquid heights in tank 1 and tank 2 will generally not be equal for $t \to \infty$. Consider for example Figure 4.8 where $H_{1,0} = 5$ m and $H_{2,0} = 0$ m and $p_{1,0} = p_{2,0}$. After $t = 0$, the gas volume in tank 1 increases, while the gas volume in tank 2 decreases due to the liquid flow from tank 1 to tank 2. In the equilibrium state, the liquid heights in the tanks should also differ to “compensate” the difference in the gas pressures. Depending on the initial values of the liquid heights and gas pressures, the system is in equilibrium for $H_1 > H_2$ or $H_1 < H_2$ (or for $H_1 = H_2$ in a special case), see Figures 4.9 and 4.10.

• As illustrated by Figures 4.11 and 4.12, liquid will flow from tank 1 to tank 2 if the pressures in the tank are different, even when the initial liquid heights in the tanks are equal. It may appear strange from first sight that it takes longer to reach equilibrium when the initial pressure difference is largest. This can be explained by the fact that more liquid has to flow from tank 1 to tank 2 if the pressure difference is large before $p_{in}$ and $p_{out}$ are equal. Note that in the first figure equilibrium is reached after approximately 620 seconds with $H_2$ close to 22.7 m, while in the second one $H_2$ exceeds 22.7 m within 400 seconds. If the initial pressure difference is further increased to $p_{1,0} = 5p_{atm}$ and $p_{2,0} = p_{atm}$, then equilibrium is reached faster, as shown in Figure 4.13.

• In these examples, the time $t_{L2}$ at which $H_2$ reaches $H_{L2}$ is easily found in the height graphs: the value of $t$ that corresponds to $H_2 = H_{L2}$ (=1 m in the examples). For $t < t_{L2}$, the pressure in tank 2 at $z = 0$, $p_{out}$, increases very slowly (it increases because the gas in the tank is compressed). For $t > t_{L2}$, it increases much faster since the liquid level rises above $H_{L2}$. In Figure 4.13 for instance, $t_{L2}$ is approximately 400 seconds. So the discontinuities in the derivatives of the pressure correspond to $t = t_{L2}$. Note that this is only valid if $H_{L1} = H_{L2}$, since we defined $z = 0$ at the pipe-tank 1 connection.

\(^2\)All computations in this report are performed on an Intel Pentium M 750 processor (1.86 Ghz).
• Figures 4.14-4.16 illustrate the effect of the closedness of the tanks. In these three figures, \( p_{1,0} = p_{2,0} = p_{atm} \) and \( H_{1,0} = H_{2,0} + 5 \) m with \( H_{1,0} = 15, 20, 25 \) m respectively. If the tanks were open, the three figures of the liquid height would have been exactly the same, but shifted 5 meters upwards, because the pressure difference over the pipe remains the same for all three initial situations for \( t \geq 0 \). In the case of closed tanks however, we observe a small difference in the final liquid heights. For larger initial values of \( H_{1,0} \) and \( H_{2,0} \) the gas in the left tank is expanding relatively more and the gas in the right tank is relatively more compressed. As a consequence, the liquid levels in tank 1 and tank 2 differ relatively more, since the difference in the gas pressures in tank 1 and tank 2 is also larger.

| \( H_{T1} \)  | 30 m |
| \( H_{T2} \)  | 30 m |
| \( D_1 \)    | 20 m |
| \( D_2 \)    | 20 m |
| \( H_{L1} \) | 1 m  |
| \( H_{L2} \) | 1 m  |
| \( D_L \)    | 0.5 m|
| \( L_L \)    | 2000 m|
| \( \alpha \)  | 2    |
| \( \rho_l \)  | 433 kg·m\(^{-3}\) |

Table 4.2: Parameters corresponding to the solutions given by Figures 4.8-4.16

![Graph of liquid levels](image1)

Figure 4.8: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and \( H_{1,0} = 5m, H_{2,0} = 0m \) and \( p_{1,0} = p_{2,0} = p_{atm} \).
Figure 4.9: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_1,0 = 5m$, $H_2,0 = 0m$ and $p_1,0 = p_{atm}$, $p_2,0 = 1.1p_{atm}$.

Figure 4.10: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_1,0 = 5m$, $H_2,0 = 0m$ and $p_1,0 = 1.25p_{atm}$, $p_2,0 = p_{atm}$.
Figure 4.11: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and \( H_1,0 = 20 \text{m}, \ H_2,0 = 20 \text{m} \) and \( p_{1,0} = 2p_{atm}, \ p_{2,0} = p_{atm} \).

Figure 4.12: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and \( H_1,0 = 20 \text{m}, \ H_2,0 = 20 \text{m} \) and \( p_{1,0} = 3p_{atm}, \ p_{2,0} = p_{atm} \).
Figure 4.13: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 20m$, $H_{2,0} = 20m$ and $p_{1,0} = 5\text{ atm}$, $p_{2,0} = p_{\text{atm}}$.

Figure 4.14: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 15m$, $H_{2,0} = 10m$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$. 
4.3 Solution

Figure 4.15: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 20\text{m}, H_{2,0} = 15\text{m}$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$.

Figure 4.16: Numerical solution of the closed two-tank system for the parameters in Table 4.2 and $H_{1,0} = 25\text{m}, H_{2,0} = 20\text{m}$ and $p_{1,0} = p_{2,0} = p_{\text{atm}}$. 
Chapter 5

Two-Tank System with Constant Temperature with Upper Pipe

5.1 Equations for Coupled System with Gas Flow in Upper Pipe

In a more realistic scenario, the two tanks are connected by two pipes instead of one pipe. The upper pipe is intended for gas flow, so that the gas from the tank that is being filled can escape to the other tank. To avoid liquid levels higher than the upper pipe connections, we assume that the pipe is connected to the tops of the tanks, see Figure 5.1.

Of course, this scenario is only of interest in the case that the system of the two tanks is closed. Similar to the liquid flow in the lower pipe, we apply the extended Bernoulli equation to the upper pipe. Suppose that the gas pressure in tank 1 is larger than the gas pressure in tank 2, and that the pressure per tank is uniform in the gas. Then we choose point a close to the upper pipe, in tank 1, and point b at the other end of the pipe, but just inside the pipe, see Figure 5.1. We then have

\[
\frac{1}{2}\rho_a v_a^2 + p_a = \frac{1}{2}\rho_b (1 + \alpha_U) v_b^2 + p_b,
\]

with \(\alpha_U\) the friction factor for gas flow in the upper pipe, see (4.8). Assume that \(v_a \ll v_b\), \(p_a \approx p_1\) and \(p_b \approx p_2\), so the energy loss from point b to tank 2 is assumed to be negligible. If this energy loss has to be taken into account, we can add a constant to \(\alpha_U\). So we obtain

\[
v_b = \begin{cases} 
\sqrt{\frac{2}{\rho_b(1+\alpha_U)}(p_1 - p_2)} & \text{if } p_1 > p_2; \\
-\sqrt{\frac{2}{\rho_b(1+\alpha_U)}(p_2 - p_1)} & \text{if } p_1 < p_2; \\
0 & \text{if } p_1 = p_2.
\end{cases}
\]

The liquid flow is again covered by the extended Bernoulli equation, see Chapter 4.

\[
\frac{dH_1}{dt} = \begin{cases} 
\frac{-\Delta L}{A_L} \sqrt{\frac{2}{1+\alpha_L}} \left( g(H_1 - \max\{H_2, H_{L1}\}) + \frac{p_1 - p_2}{\rho_l} \right) & \text{if } p_{in} > p_{out}; \\
\frac{\Delta L}{A_L} \sqrt{\frac{2}{1+\alpha_L}} \left( g(H_2 - \max\{H_1, H_{L2}\}) + \frac{p_2 - p_1}{\rho_l} \right) & \text{if } p_{in} < p_{out}; \\
0 & \text{if } p_{in} = p_{out}.
\end{cases}
\]
And by mass conservation we have again

$$\frac{dH_2}{dt} = -\frac{A_1}{A_2} \frac{dH_1}{dt}. \quad (5.4)$$

We assume that the total gas mass in the upper pipe remains constant, with a mass density equal to the average density $\bar{\rho}_g$ of the gas in tank 1 and 2. Given the initial liquid heights $H_{1,0}$ and $H_{2,0}$, the initial gas pressures $p_{1,0}$ and $p_{2,0}$ in the tanks, the constant temperature $T$ and the dimensions of the tanks, the initial gas densities in the tanks are given by the ideal gas law, see Chapter 2. And since we assume a constant temperature, the gas pressure is proportional to the gas density in time,

$$p_i = \rho_{gi}RT, \quad (5.5)$$

with

$$\rho_{gi} = \frac{m_{gi}}{A_i(H_{Ti} - H_i)} \quad (5.6)$$

and conservation of the gas masses is given by

$$-\frac{dm_{g1}}{dt} = \frac{dm_{g2}}{dt} = \varphi_g, \quad (5.7)$$

where $\varphi_g$ is the gas mass flux,

$$\varphi_g = \begin{cases} 
\rho_{g2}A_Uv_b & \text{if } p_1 > p_2; \\
\rho_{g1}A_Uv_b & \text{if } p_1 < p_2; \\
0 & \text{if } p_1 = p_2.
\end{cases}$$

Note that $\varphi_g$ has the same sign as $v_b$. The average gas density in the system is then given by

$$\bar{\rho}_g = \frac{\rho_{g1,0}A_1(H_{T1} - H_{1,0}) + \rho_{g2,0}A_2(H_{T2} - H_{2,0})}{A_1(H_{T1} - H_{1,0}) + A_2(H_{T2} - H_{2,0})}, \quad (5.8)$$
where $H_{1,0}$ and $H_{2,0}$ are not both equal to $H_{T1}$ and $H_{T2}$, respectively. Given initial conditions for liquid height and gas pressure, we can calculate numerical solutions of this problem, since we have nine equations (5.2)-(5.7) for the nine unknowns $H_i, p_i, m_{gi}, \rho_{gi}$ and $v_b$. In the normal scenario that liquid does not reach the height of the upper pipe and we do not have gas flow in the lower pipe, see condition (4.18), equilibrium is reached if $p_1 = p_2$ and $p_{in} = p_{out}$. Clearly, $H_1 = H_2$ in that case.

5.2 Numerical Results with Explicit Euler

We will first solve some of examples of this problem using explicit Euler, although this method is certainly not the best in the sense of accuracy and stability. However, this method is easy to implement and fast, which is a great advantage if the model is extended to complex scenarios or larger networks. Furthermore, it gives the possibility for a quick comparison with other solvers. Especially if we use standard Matlab solvers, as we will do later, numerical problems are recognized and solved with the explicit Euler method, for instance by adding constraints to the model.

5.2.1 Solution

The set of equations (5.2)-(5.7) is solved using the explicit Euler method, see Chapter 3. Table 5.1 shows the geometrical parameters corresponding to the solutions in Figures 5.2-5.10. Figures 5.2 and 5.3 are numerical solutions using the explicit Euler scheme with a direct implementation of equations (5.2)-(5.5), for $H_{1,0} = 6$ m, $H_{2,0} = 5$ m and $\Delta t$ is 1 s and 0.1 s respectively. The corresponding program code is given in Appendix C.2.1. As one can expect, this method causes oscillations in the liquid height and pressure. The reason for these oscillations is that for any discrete time point $t^n$, the time step is large compared to the time for the gas to reach equilibrium. This is a direct consequence of the explicit Euler method: for the calculation of properties at the next time point $t^{n+1}$, the fluxes at $t^n$ are used over the whole interval $(t^n, t^{n+1})$, although the physical fluxes are continuously decreasing in time for this example.

The figure for $\Delta t = 0.1$ s shows that decreasing the time step size leads to a better solution than the solution for $\Delta t = 1$ s. Note that the larger oscillations in Figure 5.2 also act as a kind of damping: because approximately half of the time the gas pressure in tank 2 is too large, this slows down the flow process. Furthermore, the liquid heights do not converge as equilibrium is reached, which is most visible in Figure 5.2. Instead, the liquid heights oscillate about an equilibrium with a difference between the liquid height of tank 1 and tank 2. This difference is also caused by the large gas fluxes at each time step. The liquid height and gas pressure of a tank are in phase with each other, while the liquid height and gas pressure of the other tank are in opposite phase. The gas pressure in tank 2 is oscillating about a smaller equilibrium than the gas pressure in tank 1, which compensates the difference in the liquid heights.

Assuming that the results for $\Delta t = 0.1$ s are still not acceptable, we need to use even smaller step sizes, which leads to relatively large computational costs. Before we try to solve the problem with a more advanced method or an implicit method, we will make some
modifications to our model to get reasonable results even with the explicit Euler method. To get an idea of how small the step sizes need to be to avoid the oscillations with this method, we first derive a stability condition.

<table>
<thead>
<tr>
<th>parameter</th>
<th>tank 1</th>
<th>tank 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_T$</td>
<td>20 m</td>
<td>10 m</td>
</tr>
<tr>
<td>$H_L$</td>
<td>1 m</td>
<td>1 m</td>
</tr>
<tr>
<td>$H_U$</td>
<td>20 m</td>
<td>10 m</td>
</tr>
<tr>
<td>$D_T$</td>
<td>5 m</td>
<td>2 m</td>
</tr>
<tr>
<td>$L_L$</td>
<td>10 m</td>
<td></td>
</tr>
<tr>
<td>$L_U$</td>
<td>10 m</td>
<td></td>
</tr>
<tr>
<td>$D_L$</td>
<td>0.2 m</td>
<td></td>
</tr>
<tr>
<td>$D_U$</td>
<td>0.2 m</td>
<td></td>
</tr>
<tr>
<td>$\alpha_L$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$\alpha_U$</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Geometrical parameters corresponding to the solutions given in Figures 5.2-5.10.

Figure 5.2: Liquid height and a close-up of the gas pressure for the two-tank system with constant temperature, solved by the explicit Euler method with a time step of 1 s.

5.2.2 Stability Condition for Explicit Euler Method

At time $t = t^n$, the gas mass per tank in the next time step $t = t^{n+1}$ is given by the discrete form of (5.7):

$$m_{g1}^{n+1} = m_{g1}^n - \Delta t \varphi_g^n,$$

$$m_{g2}^{n+1} = m_{g2}^n + \Delta t \varphi_g^n,$$

where the gas mass in tank $i$ and the mass flux at $t^n$ are denoted by $m_{gi}^n$ and $\varphi_g^n$, respectively. The equilibrium gas mass $m_{i,eq}^n$ in tank $i$ at $t = t^n$ is given by

$$m_{i,eq}^n = \overline{p}_g A_T (H_T - H_i^n). \quad (5.9)$$
5.2 Numerical Results with Explicit Euler

Figure 5.3: Liquid height and pressure at $z = 0$ for two-tank system with constant temperature, solved by explicit Euler with a time step of 0.1 s. $p_{\text{in}}$ denotes the pressure in tank 1 at $z = 0$ and $p_{\text{out}}$ denotes the pressure in tank 2 at $z = 0$.

Now assume that $p_{1}^{n} > p_{2}^{n}$, then stability (in the sense of avoiding $p_{1}^{n+1} < p_{2}^{n+1}$) is guaranteed if

$$p_{g1}^{n+1} \geq \bar{p}g \iff m_{g1}^{n+1} \geq \bar{p}gA_{1}(H_{T1} - H_{1}^{n+1}).$$

Since $m_{g1}^{n+1} = \rho_{g1}^{n+1}A_{1}(H_{T1} - H_{1}^{n+1})$ and $m_{g1}^{n+1} = m_{g1}^{n} - \Delta t\varphi_{g}^{n}$, we have

$$m_{g1}^{n} - \Delta t\varphi_{g}^{n} \geq \bar{p}gA_{1}(H_{T1} - H_{1}^{n+1}) \iff \Delta t \leq \frac{m_{g1}^{n} - \bar{p}gA_{1}(H_{T1} - H_{1}^{n+1})}{\varphi_{g}^{n}}.$$ (5.10)

As the inequality (5.10) shows, the time step $\Delta t$ can be large if the gas mass at $t = t^{n}$ is large compared to the gas mass in equilibrium state or if the gas mass flux is relatively small compared to the numerator.

Since $H_{1}^{n+1}$ appears in the right hand side of condition (5.10), it is not in an explicit form yet. So we will use an equation for the liquid flow to express $H_{1}^{n+1}$ in terms of $H_{1}^{n}$. In the worst case, not only the gas but also the liquid in the lower pipe flows from tank 1 to tank 2 ($p_{\text{in}} > p_{\text{out}}$), because the gas volume in tank 1 is increasing and the gas volume in tank 2 is decreasing then. This means that $p_{1}^{n+1} < p_{2}^{n+1}$ can occur with a smaller time step.

Assuming this worst case scenario, $H_{1}^{n+1}$ is given by the discrete (explicit Euler) form of equation (5.3):

$$H_{1}^{n+1} = H_{1}^{n} - \Delta t\frac{A_{L}}{A_{1}}\sqrt{\frac{2}{1 + \alpha_{L}} g(H_{1}^{n} - \max\{H_{2}^{n}, H_{L2}\}) + \frac{p_{1}^{n} - p_{2}^{n}}{\rho_{l}}}. $$ (5.11)

As this worst case scenario is not always the case ($H_{1}^{n}$ might be smaller than $\max\{H_{2}^{n}, H_{L2}\}$), we insert an absolute value in the square root before substituting (5.11) in (5.10). We then
find the following upper bound for $\Delta t$:

$$
\Delta t \leq \frac{m_{g1} - \overline{ho}_g A_1 \left\{ H_{T1} - \left( H^n_1 - \Delta t A_1 \sqrt{\frac{2}{1+\alpha_L} \left( g(H^n_1 - \max\{H^n_2, H_{L2}\}) + \frac{p^n_1 - p^n_2}{\rho_l} \right)} \right\} \right\}}{\varphi^n_g} \\
\Leftrightarrow \Delta t \leq \frac{1 + \left( A_L \overline{\rho}_g \sqrt{\frac{2}{1+\alpha_L} \left( g(H^n_1 - \max\{H^n_2, H_{L2}\}) + \frac{p^n_1 - p^n_2}{\rho_l} \right)} \right)}{\varphi^n_g} \leq \frac{m_{g1} - \overline{ho}_g A_1 (H_{T1} - H^n_1)}{\varphi^n_g}.
$$

(5.12)

For $p^n_1 < p^n_2$, an analogous stability condition can be derived. Let us denote the right hand side of the final inequality (5.12) by $\Delta t_s$. A graph of $\Delta t_s$ and the corresponding gas pressures is shown in Figure 5.4, calculated with explicit Euler with a step size of 0.005 s. We observe that $\Delta t_s$ goes to zero as the gas pressure difference goes to zero. From $t \approx 66$ s, the gas pressures are alternately largest in tank 1 and tank 2 like in Figures 5.2 and 5.3. Therefore, the calculated value of $\Delta t_s$ remains constant (and somewhat larger than zero) for $t > 66$ s.

![Graph of gas pressures and $\Delta t_s$](image)

**Figure 5.4:** Gas pressures and values of $\Delta t_s$ for $H_{1,0} = 6$ m, $H_{2,0} = 5$ m, for explicit Euler with a step size of 0.005 s. $\Delta t_s$ goes to zero as the gas pressure difference goes to zero.

### 5.2.3 Limiting the Oscillations in the Numerical Gas Flux

The oscillations in the liquid heights in Figures 5.2 and 5.3 are relatively small compared to the oscillations in the pressures, because the liquid height depends on more terms than only the gas pressure, see equation (5.3).

As mentioned, the oscillations are inherent to the usage of the explicit Euler method with too large time steps. Of course, the results are improved by using a smaller step size. But by imposing some additional constraints in our numerical model, we can still improve the results of the explicit Euler method. Doing this, we still have a fast and easy method, which hopefully gives reasonable results as well.
5.2 Numerical Results with Explicit Euler

Since the oscillations are caused by the small gas pressure differences, we add the condition that at \( t = t^n \) the gas flux over a time step \( \Delta t \) is smaller or equal to the gas flux needed to reach equilibrium \((p_1 = p_2)\) at \( t = t^{n+1} \). At every time point \( t^n \), we first calculate the liquid heights at \( t^{n+1} \), \( H_{1}^{n+1} \) and \( H_{2}^{n+1} \), by applying the explicit Euler method to equation (5.3). In the equilibrium situation, the gas pressures are equal and so are the gas densities equal to the average gas density (5.8). So the gas mass in tank 1 at \( t = t^{n+1} \) is calculated by

\[
m_{g1}^{n+1} = \begin{cases} 
\max\{m_{g1}^{n} - \Delta t \varphi_g^n, m_{1,eq}^{n+1}\} & \text{if } p_1 > p_2; \\
\min\{m_{g1}^{n} - \Delta t \varphi_g^n, m_{1,eq}^{n+1}\} & \text{if } p_1 < p_2.
\end{cases}
\]  

(5.13)

The gas mass in tank 2 is then found by mass conservation (5.7):

\[
m_{g2}^{n+1} = m_{g2}^{n} + m_{g1}^{n} - m_{g1}^{n+1}.
\]  

(5.14)

Results with these additional constraints are shown in Figures 5.5-5.6 for a time step of 1 s. In Figure 5.7, the same computations are performed, with step sizes of 0.1 s and 0.01 s respectively. As expected, the oscillations are smaller than in Figures 5.2-5.3. Moreover, the oscillations are decreasing in time, until the equilibrium is reached. Since the gas volume in tank 2 is smaller than the gas volume in tank 1 for all \( t \geq 0 \) in Figures 5.5-5.7, the oscillations in gas pressure are largest in tank 2. Although this constraint (5.13) on the gas mass gives us smaller oscillations, the results still seem not satisfactory. Since we did not encounter such problems in the previous chapter, we try to make a switch to the open two-tanks system, which is the first one of the following limit scenarios.

**Limit Case of no Friction.** In the case that the gas flow through the pipe is assumed to be frictionless and the diameter of the pipe is large, i.e. the gas can move freely (without opposing forces) from one tank to the other tank, the system behaves like the open two-tank system in Section 4.3 when \( p_1 = p_2 \) holds. Note that we consider quasi-steady flow again. Then, liquid and gas volume flow are equal (but in opposite direction), and hence the gas
Two-Tank System with Constant Temperature with Upper Pipe

Figure 5.6: Gas pressure history for two-tank system with constant temperature, solved by explicit Euler with additional constraint (5.13) and a time step of 1 s.

Figure 5.7: The pressures at the end of the lower pipes for explicit Euler with additional constraint (5.13) and step sizes of 0.1 s and 0.01 s respectively.
5.2 Numerical Results with Explicit Euler

Pressures remain equal. At any time $t$, assuming a positive pressure difference $p_{in} - p_{out}$, the liquid volume flux $\Phi$ is given by

$$\Phi = A_1 v_a = A_L v_b = A_L \sqrt{\frac{2}{(1 + \alpha L) \rho_l}} (p_{in} - p_{out}).$$  \hspace{1cm} (5.15)

**Limit Case of Infinite Friction.** By making the diameter of the upper pipe smaller and smaller or by considering an increasing friction factor, we come closer to the limit case of “infinite friction” or a closed valve. The system is then equal to the closed two-tank system without upper pipe, see Section 4.3.

5.2.4 Switching to Open System

In the example of Figures 5.2-5.7 with a relatively large upper pipe diameter and small differences in the gas pressure, the situation is almost the same as in the first limit case above. So even with the additional constraint that the gas flow per time interval should be at most equal to the amount of flow that is needed to reach equal pressures $\left(5.13\right)$, oscillations occur: if $p_{1n} = p_{2n}$, the gas masses in the tanks do not change, but the liquid heights change. So at $t = t^{n+1}$, the gas pressures are unequal again. Compare Figures 5.2 and 5.6. In the first one, the gas pressures in tank 1 and 2 both oscillate around the equilibrium state of 1 bar, while in the latter a gas pressure of 1 bar is a maximum in tank 1 and a minimum in tank 2. If we would like to avoid these oscillations close to the point at which $p_{1} = p_{2}$, we could change the system to the open tanks case as soon as we are close to this equilibrium. In the limit case of no friction, this is indeed valid. To do so, we extend equation $\left(5.13\right)$ to

$$m_{g1}^{n+1} = \begin{cases} \max\{m_{g1}^n - \Delta t \rho_g, m_{1,eq}^n\} & \text{if } \frac{p_1 - p_2}{p_1} > \varepsilon; \\ \min\{m_{g1}^n - \Delta t \rho_g, m_{1,eq}^n\} & \text{if } \frac{p_1 - p_2}{p_1} < -\varepsilon; \\ m_{1,eq}^{n+1} & \text{otherwise.} \end{cases}$$  \hspace{1cm} (5.16)

Here, $\varepsilon$ is a small positive number between, say, $\varepsilon = 10^{-2}$, where $\varepsilon$ is the used machine precision ($\varepsilon \approx 2.2 \cdot 10^{-16}$). This is done, since for $\varepsilon = 0$ oscillations due to rounding errors in the calculation of the gas pressures can occur, which is shown in Figures 5.8-5.9 for $\Delta t = 0.1 \text{ s}$. So $\varepsilon$ should be chosen large enough to avoid these oscillations and, if one wants to avoid oscillations around the equilibrium state, also large enough to make sure that we will eventually reach the point that $-\varepsilon < \frac{p_1 - p_2}{p_1} < \varepsilon$. If we indeed arrive in this equilibrium region, which is the case every time that $m_{g1}^{n+1} = m_{1,eq}^{n+1}$, the system behaves as the open tank system and the gas pressures remain equal from then on. Now, we finally obtain smooth results with the explicit Euler method, even for $\Delta t = 1 \text{ s}$, see Figure 5.10. A value for $\varepsilon$ of $10^{-8}$ is used here.

\footnote{Analogously, this is done for the liquid flow in the lower pipe, but we will not discuss it into detail here.}
Two-Tank System with Constant Temperature with Upper Pipe

Figure 5.8: Liquid height and pressure at $z = 0$ for two-tank system with constant temperature, solved by explicit Euler with constraints (5.16) and a time step of 0.1 s.

Figure 5.9: Gas pressure history for two-tank system with constant temperature, solved by explicit Euler with constraint (5.16) and a time step of 0.1 s.

Runge-Kutta Methods

We also solved the problem using Matlab’s ODE solver `ode45`, based on the Dormand-Prince pair (3.13), with a relative tolerance of $10^{-8}$. We did not impose the additional constraints as discussed above, but solved the set of equations (5.2)–(5.7). We then obtain figures that look the same as in Figures 5.10. In Section 5.5 at the end of this chapter, we compare the speed and accuracy of the used methods.
5.3 Filling of a Tank

Another numerical problem is encountered when one of the tanks gets totally filled with liquid. This problem is one of the motivations of this project, as mentioned in the Introduction. We will again consider an example with the parameters in Table 5.1. Assume that the liquid level in tank 1 is higher than the lower tank-pipe connections and that the initial pressure difference over the lower pipe is large enough to make sure that tank 2 will be totally filled with liquid.

In the next examples, we will consider an initial situation of $H_1,0 = 12$ m, $H_2,0 = 2$ m and $p_{1,0} = p_{2,0} = 10^5$ Pa.

5.3.1 Explicit Euler Method

Using the explicit Euler method to solve equations (5.2)-(5.5) without any additional constraints gives bad results again, as we could expect, see Figure 5.11. This picture nicely illustrates that the oscillations become larger as the gas volume in tank 2 decreases. In equation (5.6), we divide by $H_{T_i} - H_i$, so errors in $H_i$ will be magnified as $H_{T_i} - H_i$ goes to zero. If we add the constraint (5.13) that at each time step the gas flow is limited to the gas flow needed to reach equilibrium, like in Section 5.2.3, we still obtain problems with the tank filling, see the pressure graph in Figure 5.12 although the graph of the liquid heights looks fine at first sight.

In the case of our model that switches to the open-tanks system using 5.16, the problem with the filling of the tank does not occur. Obviously, the gas pressures are not calculated anymore then, so the method does not “recognize” the tank to be filled when $H_2(t) = 10$ m. See Figure 5.13 for the corresponding solution for $\Delta t = 0.1$ s.

Remark. It is important to notice that if the liquid height comes close to the tank height, the gas volume in this tank becomes very small. So it is questionable to what extent our model is still accurate then, since the gas volume in the pipe is not negligible anymore compared to...
the gas volume in the tank. In this report, we only try to solve the numerical problems that arise when the liquid level in the tank comes close to the tank height.

Figure 5.11: Liquid height and pressure at $z = 0$ for two-tank system with constant temperature, solved by explicit Euler without any additional constraints with a time step of 0.1 s. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.

Figure 5.12: Liquid height and pressure at $z = 0$ for two-tank system with constant temperature, solved by explicit Euler with constraint (5.13) and a time step of 0.1 s. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.
5.3 Filling of a Tank

5.3.2 More Advanced Runge-Kutta and NDF Methods

The same computations as above are performed with Matlab’s ODE solvers ode45, which we already used before, ode15s and ode23t [31]. The methods ode23t and ode15s are the only methods that can solve DAEs, which we will encounter in the next chapters, so it is of interest to investigate their performance.

These methods can give acceptable results, but one has to be careful with the requested tolerance. In case that this tolerance is not feasible using the smallest possible step size, the method breaks down automatically. However, if the method does not break down, it does not guarantee realistic results, as will become clear in the following.

Using ode45 we obtain good results, but only if the requested tolerance is small enough. For a relative tolerance of $10^{-2}$, $10^{-4}$ and $10^{-12}$ the corresponding graphs are given in Figures 5.14-5.16.

The method ode23t, based on the trapezoidal method, gives already quite good results for a relative tolerance of $10^{-2}$. But as the liquid level in tank 2 reaches the height of the tank, the method becomes unreliable. In contrast to the ode45 method, it does not break down early for a relative tolerance of $10^{-4}$. Results are given for relative tolerances of $10^{-2}$, $10^{-4}$ and $10^{-6}$ in Figures 5.17-5.19.

Finally, results with ode15s are shown in Figures 5.20-5.21 for relative tolerances of $10^{-4}$ and $10^{-6}$. It is interesting to see that this method simply seems to “stop” the flow if tank 2 is filled.
5.3.3 Switching to Filled-Tank Situation

To avoid that Matlab’s standard methods break down when the liquid level in one of the tanks comes close to the tank height, we can stop the integration using the Events option in the ODE solvers [31]. The integration is then restarted keeping the variables constant. Results for ode45 are shown in Figure 5.22 for a relative tolerance of $10^{-6}$. Using ode23t and ode15s, we obtain similar pictures. Still, we need to be careful with the requested tolerance for the ode45 method as oscillations before the tank is filled, as in Figure 5.14, are not excluded by the Events option. With these initial values, for a relative tolerance of $10^{-5}$ or larger, there is a jump in the solution of the gas pressure in tank 2 at the switching point, see Figure 5.23. However, this is not very relevant at this moment, since there is no gas left in tank 2 after this tank becomes filled with liquid.
5.3 Filling of a Tank

Figure 5.16: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode45 with relative error $10^{-12}$. Initial values are $H_{1,0} = 12 \text{ m}$, $H_{2,0} = 2 \text{ m}$ and $p_{1,0} = p_{2,0} = 1 \text{ bar}$.

Figure 5.17: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode23t with relative error $10^{-2}$. Initial values are $H_{1,0} = 12 \text{ m}$, $H_{2,0} = 2 \text{ m}$ and $p_{1,0} = p_{2,0} = 1 \text{ bar}$. 
Figure 5.18: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by `ode23t` with relative error $10^{-4}$. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.

Figure 5.19: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by `ode23t` with relative error $10^{-6}$. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.
5.3 Filling of a Tank

Figure 5.20: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode15s with relative error $10^{-4}$. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.

Figure 5.21: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode15s with relative error $10^{-6}$. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.
Two-Tank System with Constant Temperature with Upper Pipe

Figure 5.22: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode45 with Events option and relative error $10^{-6}$. Initial values are $H_{1,0} = 12$ m, $H_{2,0} = 2$ m and $p_{1,0} = p_{2,0} = 1$ bar.
Figure 5.23: Liquid height, pressure at \( z = 0 \) and gas pressure for two-tank system with constant temperature, solved by \texttt{ode45} with \texttt{Events} option and relative error \( 10^{-5} \). Initial values are \( H_{1,0} = 12 \) m, \( H_{2,0} = 2 \) m and \( p_{1,0} = p_{2,0} = 1 \) bar.
5.4 Upper Pipe Effects

Apart from numerical problems arising from the gas flow in the upper pipe, the system is also essentially different from the system with only a lower pipe. This is shown in the following examples. In Figures 5.24 and 5.25 solutions are given for initial liquid heights $H_{1,0} = 6 \text{ m}$ and $H_{2,0} = 5 \text{ m}$, initial gas pressures of $10^5 \text{ Pa}$ and an upper pipe diameter of 0.05 m and 0.005 m respectively. The other parameters are the same as in Table 5.1. Compared to Figure 5.10, the decrease of $D_U$ to 0.05 m slows down the flow by approximately 20 s. We could say that from the point where the pressure difference over the upper pipe is maximum, the gas flow starts “dominating” the process. In Figure 5.24 this occurs after roughly 10 s. It then takes approximately another 60 s to reach the equilibrium state. The gas volume flux $\Phi_{gas}$ is given by $\Phi_{gas} = A_U v_b$, where $v_b$ is given by equation (5.2). Therefore, we expect that decreasing the upper pipe diameter by a factor 10 leads to a gas volume flux decrease by a factor 100, since $A_U \sim D_U^2$. Indeed, the time to reach equilibrium in Figure 5.25 is approximately 6000 s, whereas the time to reach the maximum gas pressure difference is very small, relatively.

![Figure 5.24: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode15s with relative error $10^{-8}$ for $D_U = 0.05 \text{ m}$, $H_{1,0} = 6 \text{ m}$, $H_{2,0} = 5 \text{ m}$, $p_1 = p_2 = 10^5 \text{ Pa.}$.](image-url)
5.4 Upper Pipe Effects

Figure 5.25: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode15s with relative error $10^{-8}$ for $D_U = 0.005$ m, $H_{1,0} = 6$ m, $H_{2,0} = 5$ m, $p_1 = p_2 = 10^5$ Pa.

In Figures 5.26 and 5.27 similar computations are performed for initial liquid $H_{1,0} = H_{2,0} = 5$ m and initial gas pressures of $p_{1,0} = 1.5 \cdot 10^5$ Pa and $p_{2,0} = 10^5$ Pa for $D_U = 0.05$ m and $D_U = 0.005$ m, respectively. Other parameters are again equal to the parameters in Table 5.1. From the moment that the gas pressure difference dominates the flow, the time scale to reach equilibrium in both graphs differs by approximately a factor 100. Note that the maximum difference in liquid height is larger for $D_U = 0.005$ m than for $D_U = 0.05$ m, as one would expect. In Figure 5.28 a solution is given for $D_U = 0$ m, which is similar to Figures 4.11-4.13 without upper pipe.

Figure 5.26: Liquid height, pressure at $z = 0$ and gas pressure for two-tank system with constant temperature, solved by ode15s with relative error $10^{-8}$ for $D_U = 0.05$ m, $H_{1,0} = H_{2,0} = 5$ m, $p_1 = 1.5 \cdot 10^5$ Pa and $p_2 = 10^5$ Pa.
5.5 Speed and Accuracy Comparison

As we have used several numerical methods to solve the problem, which is more complex now than in the previous chapter, it is of interest to know how the speed and accuracy of the methods compare. For the explicit Euler method with constraint (5.16), speed and order of convergence are shown in Table 5.2. The order of convergence \( p \) is estimated with equation (3.4) of Chapter 3. Calculations are done with parameters in Table 5.1, for initial liquid heights \( H_{1,0} = 6 \text{ m} \) and \( H_{2,0} = 5 \text{ m} \) and initial gas pressures of 1 bar, as in Figure 5.10. As a reference point we take \( t = 50 \text{ s} \); let us denote the numerical solution of \( H_1 \) at \( t = 50 \text{ s} \) by \( H_k \), for \( 10^k \) time steps (from 0 to 100 s). In Table 5.2 we observe that the method is indeed first-order accurate. Furthermore, we encounter a dramatic increase in calculation time for \( k = 5 \), much larger than a factor 10. Since the liquid heights are somewhere between 5 and 6 m, the relative error for \( k = 5 \) is about \( 10^{-7} \), which is good.
In Table 5.3, calculation times are shown for the solution of the same problem solved by the Matlab solvers ode45, ode23t and ode15s. Clearly, for relative tolerances of $10^{-6}$ or smaller, the Matlab solvers are much faster than explicit Euler with constraint (5.16). For larger tolerances, the speed of the methods is comparable. However, the results of ode45 and ode15s should not be trusted blindly then. In Figure 5.29 results are shown for ode45 and ode15s, both with a relative tolerance of $10^{-2}$. In the first case, we observe a similar problem as with explicit Euler without constraints, although its graph looks quite smooth: due to oscillations, it takes much more time for the liquid heights to become equal, approximately 200 s instead of 66 s. This could be expected, since ode45 is also an explicit method. For ode15s, the solution behaves well up to 50 s, but then the liquid heights diverge again.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Calculation time (s)</th>
<th>$H_{k+1} - H_k$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>7.27e-3</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>6.51e-4</td>
<td>1.048</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>6.44e-5</td>
<td>1.004</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>6.44e-6</td>
<td>1.000</td>
</tr>
<tr>
<td>5</td>
<td>8e1</td>
<td>6.44e-7</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>1e4</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.2: Speed and order of convergence $p$ for explicit Euler method with constraint (5.16). $H_k$ is the solution of the liquid height in tank 1 at $t = 50$ s for $10^k$ time steps.

<table>
<thead>
<tr>
<th>$k$</th>
<th>ode45</th>
<th>ode23t</th>
<th>ode15s</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 5.3: Calculation times in seconds for Matlab’s ODE solvers, for relative tolerances of $10^{-k}$.

We now perform a similar comparison to the “filling of a tank” scenario, corresponding to initial liquid heights of $H_{1,0} = 12$ m and $H_{2,0} = 2$ m, see Table 5.4. We compare solutions at $t = 150$ s, which is close to the time at which liquid in tank 2 reaches the top of the tank. Let $H$ denote the liquid height in tank 1 obtained by ode15s at $t = 150$ s with a relative tolerance of $10^{-12}$. Then $\frac{H_{1,0} - H_1}{H_{1,0}}$ is used as a relative error estimate of $H_k$, which denotes the numerical approximation of $H_1$ at $t = 150$ s with a relative tolerance of $10^{-k}$. Analogously, $\Delta p_k$ denotes the numerical approximation of the gas pressure difference $p_1 - p_2$ at $t = 150$ s, calculated with a relative tolerance of $10^{-k}$. Again, we compare these values with the numerical solution of the gas pressure difference calculated by ode15s with a relative tolerance of $10^{-12}$.

Like in the case of initial liquid heights of $H_{1,0} = 6$ m and $H_{2,0} = 5$ m, we observe a bad performance of the ode45 method for large tolerances. Although the (absolute) relative error in liquid height is small compared to the requested tolerance, the (absolute) relative error in the pressure difference is relatively large, especially for $k \leq 4$. For smaller tolerances, ode45 seems to be accurate, but it is then very slow compared to the other two methods ode23t
Two-Tank System with Constant Temperature with Upper Pipe

Figure 5.29: Liquid heights calculated by ode45 (a) and ode15s (b) with a relative tolerance of $10^{-2}$.

and ode15s.

In Table 5.4 convergence order and calculation times are shown for the explicit Euler method with constraint (5.16). As the relative error is approximately 10 times smaller than $H_{k+1} - H_k$, we observe that the calculation time of this method is comparable to the calculation times of Matlab's ODE solvers for $k \leq 4$. Not very surprisingly, the convergence order is equal to 1 again. Moreover, both in Table 5.2 and in Table 5.5 the difference $H_k - H_{k-1}$ is always positive, as expected: the Euler method is overestimating the liquid flux over a whole time step, since the flux of the previous time point is used. Making the step size smaller, the overestimation also becomes smaller.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$t_c$ (s)</th>
<th>$H_k - H_{k-1}$</th>
<th>$\Delta p_k - \Delta p_{k-1}$</th>
<th>$t_c$ (s)</th>
<th>$H_k - H_{k-1}$</th>
<th>$\Delta p_k - \Delta p_{k-1}$</th>
<th>$t_c$ (s)</th>
<th>$H_k - H_{k-1}$</th>
<th>$\Delta p_k - \Delta p_{k-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.0</td>
<td>-6.2e-4</td>
<td>2.5e2</td>
<td>0.60</td>
<td>3.7e-4</td>
<td>3.5e-2</td>
<td>0.58</td>
<td>2.6e-4</td>
<td>2.6e-2</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>-1.8e-5</td>
<td>-5.8e-1</td>
<td>0.64</td>
<td>1.5e-6</td>
<td>1.2e-4</td>
<td>0.66</td>
<td>3.6e-6</td>
<td>3.0e-4</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>-5.3e-10</td>
<td>-6.5e-4</td>
<td>0.67</td>
<td>-2.5e-10</td>
<td>-1.7e-5</td>
<td>0.64</td>
<td>6.1e-10</td>
<td>3.0e-7</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>2.7e-11</td>
<td>-3.5e-5</td>
<td>0.71</td>
<td>1.9e-11</td>
<td>-2.4e-6</td>
<td>0.70</td>
<td>-5.0e-12</td>
<td>-1.6e-7</td>
</tr>
<tr>
<td>10</td>
<td>38</td>
<td>-2.2e-13</td>
<td>1.7e-7</td>
<td>1.1</td>
<td>3.7e-14</td>
<td>-2.6e-7</td>
<td>0.79</td>
<td>-2.6e-12</td>
<td>-3.7e-8</td>
</tr>
<tr>
<td>12</td>
<td>1.1e2</td>
<td>-2.9e-13</td>
<td>-3.5e-8</td>
<td>2.7</td>
<td>-4.4e-13</td>
<td>-9.5e-8</td>
<td>4.1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.4: Calculation times $t_c$ in seconds, error estimates of the liquid height in tank 1 and relative error estimate of the gas pressure difference at $t = 150$ s and for Matlab’s ODE solvers, for relative tolerances of $10^{-k}$. Initial liquid heights are $H_{1,0} = 12$ m and $H_{2,0} = 2$ m.
5.5 Speed and Accuracy Comparison

Table 5.5: Speed and order of convergence $p$ for explicit Euler method with constraint (5.16). $H_k$ is the solution of the liquid height in tank 1 at $t = 150$ s for $10^k$ time steps. Initial liquid heights are $H_{1,0} = 12\text{m}$ and $H_{2,0} = 2\text{m}$. 

<table>
<thead>
<tr>
<th>$k$</th>
<th>Calculation time (s)</th>
<th>$H_{k+1} - H_k$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>5.77e-2</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>5.29e-3</td>
<td>1.037</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>5.25e-4</td>
<td>1.003</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>5.25e-5</td>
<td>1.000</td>
</tr>
<tr>
<td>5</td>
<td>5e1</td>
<td>5.25e-6</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>1e4</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Chapter 6

One-Tank System with Variable Temperature

6.1 Problem Description

To get insight in the effect of heating of the system and to get used to REFPROP, we will consider this in a one-tank system first. Suppose we have a fixed single tank, filled with a substance consisting of one or more components in liquid-gas phase. We have a uniform heat flux per unit area $q$ through the wall of the tank. Furthermore, we assume that the temperature $T_i$ inside the tank is uniform (which is explained in Section 6.2.2) but varying in time. The ambient temperature $T_{amb}$ is both uniform and constant in time, see Figure 6.1.

![Figure 6.1](image)

**Figure 6.1:** Tank containing a substance in liquid-gas phase, with a uniform heat flux $q$ per unit area through the wall of the tank. Ambient temperature as well as inside temperature are uniform.
We are interested in the evolution of the quality $\xi$, which is the mass vapor fraction, defined as

$$\xi = \frac{m_g}{m_g + m_l} = \frac{m_g}{m}, \quad (6.1)$$

where $m_g$ and $m_l$ are the total gas and liquid mass in the tank, respectively, and $m$ is the total mass of the substance in the tank. We assume that the saturated liquid and gas are strictly separated by a horizontal interface, so that we have $\xi_l = 0$ in the lower part of the tank and $\xi_g = 1$ in the upper part. This means that no gas is dissolved in the liquid and there is no liquid in the gas or against the wall above the liquid-gas interface. For multi-component substances, the composition of liquid and gas will change, since components evaporate at different rates. This will be discussed in Chapter 8. For now, we consider the pure single-component substance methane.

6.2 Heat

6.2.1 Heat Flux

The heat flux through the tank walls is modeled by Fourier’s law:

$$q = -\kappa \nabla T, \quad (6.2)$$

where $\kappa$ is the coefficient of heat conduction, which is supposed to be constant. We assume a linear temperature profile over the tank wall as shown in Figure 6.2.

Figure 6.2: The temperature profile in the tank wall is assumed to be linear.

The heat flux then becomes uniform:

$$q = -k(T_i - T_{amb}). \quad (6.3)$$

The (positive) coefficient $k$ denotes the heat flux per unit area per unit temperature: $[k] = \text{W} \cdot \text{m}^{-2} \text{K}^{-1}$. If $T_{amb} > T_i$, then $q > 0$ and if $T_{amb} < T_i$, then $q < 0$.

If we assume $k$ to be constant and uniform over the walls of the tank, we can write the total heat transfer rate $Q$ as

$$Q = -K(T_i - T_{amb}), \quad (6.4)$$

where $K$ is the heat flux coefficient of the total tank per unit temperature difference, $K = S\kappa$, with $S$ the heat transferring surface area of the tank.

---

1 This assumption is not necessary for the one-tank model, but we will use it in the two-tank model of Chapter 7. It is therefore of interest to see how properties like gas and liquid density, gas pressure, liquid and gas enthalpy vary in time.
6.2.2 Biot Number

Since we consider a highly insulated system, we assume that heat conduction inside the tank is much faster than heat exchange with the exterior. This is described by the Biot number, named after the French physicist Jean-Baptiste Biot, which is given by

$$Bi = \frac{kL}{k_b}.$$  \hspace{1cm} (6.5)

Here, $k$ denotes the heat transfer coefficient as in equation (6.3), $L$ denotes a characteristic length and $k_b$ denotes the thermal conductivity of the body. The characteristic length is often defined as body volume divided by surface area: $L = \frac{V_b}{S}$. Clearly, if $V_b$ is large compared to the surface area $S$, this means that a relatively large part of the body is farther away from the exterior. Hence, it takes more time for this part to reach the temperature of the exterior.

In our tank, the fluid forms the body and $k$ describes the heat exchange with the exterior. The Biot number can then be seen as the ratio between the thermal resistance inside the fluid and the thermal resistance between the exterior and the tank interior. If $k_b$ is large compared to $kL$, the conduction inside the tank is much faster than the heat transfer rate between the fluid and the exterior. Due to the strong insulation, we assume $Bi \ll 1$, so that added heat is spread instantaneously over the liquid and the gas in such a way that we have a uniform temperature field inside the tank. This implies that we assume quasi-equilibrium states. Considering the heat transfer of the tank as a whole is also called a lumped-capacity model. This is convenient for describing larger networks of “lumps”, each with its own thermal resistance, which is very similar to modeling electrical circuits. More information about the Biot number and lumped-capacity models can be found in [19].

6.3 Equations and Variables

6.3.1 Equations for Two-Phase System

In addition to the dimensions of the tank, the ambient temperature, and the material properties of the substance, we need to know two independent variables to describe the state of the fluid, for which we make use of REFRPROP [15], see Appendix [13]. We assume that the initial quality $\xi_0$ (6.1) and the initial (uniform) temperature in the tank $T_0$ are given.\footnote{Instead of quality, it is maybe more intuitive to start with liquid height, since this is easier to measure. However, quality is an input variable of REFRPROP, so it is easier implemented in the model. This choice does not affect the results.} Then, assuming a quasi-equilibrium state, we can calculate the average density $\rho = \frac{m}{V}$ of the substance in the tank as a function of $\xi_0$, $T_0$ using REFRPROP. Let $f_z(x,y)$ denote the REFRPROP output value for $z$ if $x$ and $y$ are used as input. Then

$$\rho = \frac{m}{V} = f_\rho(T_0, \xi_0),$$  \hspace{1cm} (6.6)

where $m$ is the total mass in the tank and $V = A_T H_T$ the tank volume. Since we have a closed tank, we assume conservation of the total mass $m$ in the tank, $\frac{dm}{dt} = 0$. Furthermore,
the process is isochoric, i.e. the volume of the fluid remains constant, and hence also the density is constant:
\[
\frac{d\rho}{dt} = 0.
\] (6.7)

Conservation of energy in enthalpy form is given by [226]:
\[
\rho \frac{dh}{dt} = \frac{dp}{dt} - \nabla \cdot q + \tau : \nabla v + \rho r.
\] (6.8)

For the moment we assume a uniform pressure in the tank, in order to derive a relation between the change of enthalpy \( I \) of the total tank and the heat transfer rate \( Q \). Neglecting the friction term \( \tau : \nabla v \) and the internal heat sources \( \rho r \) in (6.8), we obtain the following change in enthalpy for the total tank:
\[
\frac{dI}{dt} = \frac{d}{dt}(mh) = m \frac{dh}{dt} = \rho V \frac{dh}{dt} = V \frac{dp}{dt} + Q.
\] (6.9)

Since \( I = U + pV \), we have \( \frac{dI}{dt} = \frac{dU}{dt} + p \frac{dV}{dt} + V \frac{dp}{dt} \). The volume is constant, \( \frac{dV}{dt} = 0 \), and the change in total internal energy \( U \) is therefore equal to the added heat,
\[
\frac{dU}{dt} = Q = -K(T - T_{amb}).
\] (6.10)

Clearly, equilibrium is reached when \( T = T_{amb} \).

Remark. Note that we assume that the heat flux only depends on the temperature difference between the inside of the tank and the exterior. So although the state of the substance changes, the heat flux coefficient \( K \) in (6.10) remains constant in our model.

Equations (6.7) and (6.10) are the two differential equations of interest, where the first one has a trivial solution. The system is further described by a set of algebraic equations, for which we use REFPROP.

Since we assume \( Bi \ll 1 \), supplied heat is spread instantaneously and hence we assume a quasi-equilibrium state. The quality \( \xi \) can be calculated by REFPROP, given two thermodynamic state variables. We will use the specific internal energy \( u = \frac{U}{m} \) and the density \( \rho \):
\[
\xi = f_\xi(u, \rho).
\] (6.11)

Note that we use the average density and average specific energy over the tank in this calculation (6.11). So here we implicitly assume that the increasing hydrostatic pressure towards the bottom of the tank is negligible. The initial value of \( U \) can also be found by this relation:
\[
U_0 = m \cdot u_0 = m \cdot f_u(\xi_0, \rho_0).
\]

Also the temperature can be calculated as a function of the specific internal energy and density:
\[
T = f_T(u, \rho).
\] (6.12)
Equations (6.7), (6.10), (6.11) and (6.12) form a set of four differential-algebraic equations for the four unknowns $T, \xi, u$ and $\rho$, for which we can calculate numerical solutions. Keeping the two-tank model in mind, it is of interest to investigate the behavior of some other quantities as well, although we do not necessarily need them in the one-tank model. We calculate these quantities in the following way.

The liquid density $\rho_l$ can be calculated as a function of $T$, assuming saturated liquid (recall that we assumed separation of liquid and gas with $\xi_l = 0$ in the liquid):

$$\rho_l = f_{\rho_l}(T, \xi_l).$$

(6.13)

The liquid height $H$ is related to the quality as

$$H = \frac{(1 - \xi)m}{\rho_l A_T}.$$  

(6.14)

The gas density is given by

$$\rho_g = \frac{\xi \cdot m}{A_T (H_T - H)}$$

(6.15)

and the gas pressure can be calculated as a function of the gas density and the temperature:

$$p_g = f_{\rho_g}(T, \rho_g).$$

(6.16)

The specific liquid enthalpy as a function of $T$ and $\rho_l$ is given by

$$h_l = f_{h_l}(T, \rho_l)$$

(6.17)

and the specific gas enthalpy as a function of $T$ and $\rho_g$ by

$$h_g = f_{h_g}(T, \rho_g).$$

(6.18)

### 6.3.2 Switching to One-Phase System

Depending on the heat transfer rate and the state of the system, the added heat either causes vaporization of the liquid or condensation of the gas, so we may eventually reach a point at which we have a one-phase system: $\xi = 0$ or $\xi = 1$. Let us denote this time by $t_{one}$. From then on, it is not very interesting anymore to calculate specific density or enthalpy for both liquid and gas.

Even for positive $Q$ it is possible to have a net condensation of vapor, because the heating of the closed system results in an increasing pressure. Although the temperature increases, the compression resulting from the larger pressure can “force” the fluid to be in liquid phase, this is a so-called subcooled liquid. Although the temperature might be higher than its standard boiling point, the fluid is forced to be a liquid since the temperature is lower than the saturation temperature at the higher pressure.

---

4The standard boiling point is the saturation temperature at $p = 1$ bar.
If all liquid turns into vapor and heating continues, the temperature will rise above the boiling point temperature and the fluid is referred to as a superheated vapor. If the fluid is heated so much that pressure and temperature become above its critical point, we have a supercritical state: there is no longer a distinction between liquid and gas, so \( \rho_l = \rho_g \), \( h_l = h_g \). More information about subcooling, superheating, supercritical states, etcetera, is found in e.g. [7, 19, 35].

With the change from a two-phase system to a one-phase system near saturated conditions, we have to make sure that no numerical problems arise. Several variables will have discontinuous time derivatives at \( t = t_{one} \). The liquid height \( H \) will either remain constant at 0 or at \( H_T \). The quality output value of \texttt{REFPROP} for superheated and subcooled states is 998 and -998, respectively. Quality values outside the interval \([0, 1]\) cannot be used as input of \texttt{REFPROP} calls. For \( t \geq t_{one} \), equations (6.13)-(6.15) are therefore replaced by

\[
\begin{align*}
\frac{d\rho_l}{dt} &= 0, \\
\frac{dH}{dt} &= 0, \\
\frac{d\rho_g}{dt} &= 0.
\end{align*}
\] (6.19)

(6.20)

(6.21)

For heating beyond the critical point, we replace equations (6.19) and (6.21) by

\[
\rho_l = \rho_g = \frac{m}{V}.
\] (6.22)

Note that consequently specific gas and liquid enthalpy are also equal, see (6.17)-(6.18).

### 6.4 Numerical Results

Combining the linearized heat flux model (6.3) with the assumption of a small Biot number, we can use some experimental data to estimate realistic values of the heat conduction factor \( K \). Typical heat flux densities of 4 W \( \cdot \) m\(^{-2}\) for tanks and 30 W \( \cdot \) m\(^{-2}\) for pipes of LNG installations are measured, according to [8] and [27], respectively.

Now consider the following tank with initial temperature \( T_0 \) and initial liquid height \( H_{1,0} \), see Table 6.1. The surface area \( S \) of this tank is equal to

\[
S = 2\pi \frac{1}{2} D H_T + 2\pi (\frac{1}{2} D)^2 = 600\pi + 200\pi = 800\pi \approx 2513 \text{ m}^2.
\] (6.23)

| \( H_T \) | 30 m |
| \( D \) | 20 m |
| \( T_{amb} \) | 293 K |
| fluid | pure methane |

Table 6.1: Parameters corresponding to the solution given in Figures 6.3-6.8.
6.4 Numerical Results

Assuming a constant ambient temperature of 293 K, we will use a heat conduction factor $K$ of $55 \text{ W m}^{-1} \text{K}^{-1}$, so $k = \frac{K}{2513} \approx 2.19 \cdot 10^{-2} \text{ W m}^{-3} \text{K}^{-1}$. For a temperature difference of 182 K, according to the linearized heat flux model (6.3), this gives a total heat flow $Q$ of

$$Q = -K(T - T_{\text{amb}}) = 55 \cdot 182 = 10010 \text{ W},$$

(6.24)

corresponding to a heat flux density $q$ of $3.98 \text{ W m}^{-2}$, which is close to the $4 \text{ W m}^{-2}$ mentioned earlier.

6.4.1 Explicit Euler

A numerical solution of the set of equations (6.7), (6.10)-(6.12) is calculated, where the explicit Euler scheme is used for (6.10):

$$U^{n+1} = U^n - \Delta t \cdot K(T^n - T_{\text{amb}}),$$

(6.25)

for given initial values $T_0$ and $\xi_0$ and the parameters given in Table 6.1. The quantities $\rho_l, H, \rho_g, h_l, h_g$ and $p_g$ are calculated as well, using equations (6.13)-(6.18) for the two-phase system and the replacement of (6.13)-(6.15) by (6.19)-(6.21) for the one-phase system. The Matlab implementation is found in Appendix C.3.1.

For an initial quality $\xi_0 = 0.01$, initial temperature $T_0 = 111 \text{ K}$ and a step size $\Delta t$ of 1 day, the corresponding graphs are given in Figure 6.3.5. Neglecting the question whether the tank is resistant to such high pressures for large $t$, we discuss the evolution of the variables in two parts: before and after the point where all liquid is vaporized, $t = t_{\text{one}}$. For $t < t_{\text{one}}$, it is remarkable that the liquid height increases, while the liquid mass decreases. For an observer of the liquid height this is misleading, because he might think that gas is condensing to liquid (more precise: more gas is condensing than liquid is evaporating). However, the other way around is true, but at the same time the liquid density decreases so much that the net liquid volume is increasing. From the liquid mass graph, we can see that the rate of evaporation is increasing in time, which corresponds to the enthalpy graph: the difference between gas and liquid enthalpy is decreasing, so less heat is needed to turn liquid into vapor. For $t > t_{\text{one}}$, temperature can increase faster, as no heat is used anymore to turn liquid into gas. Quickly after all liquid is vaporized, we are left with a supercritical fluid: the liquid and gas enthalpy are equal and there is no difference anymore between liquid and gas. Furthermore, the pressure increases more proportional with the temperature compared to before this point, because the gas volume and mass changed then. The temperature converges to the ambient temperature, as we would expect. Note that properties of the liquid are not relevant anymore if we are left with only gas, and vice versa.

In Figure 6.4, the results are shown for initial values $\xi_0 = 0.001$ and $T_0 = 111 \text{ K}$, solved by explicit Euler with a time step of 1 day. Instead of a superheated vapor, we now obtain a subcooled liquid, after approximately 110 weeks. Soon afterwards, the critical point is

\footnote{Note that the time scale of the quality graph is smaller than the scale of the other graphs, since for $t > t_{\text{one}}$ we have qualities of 998 and 999, which is not very interesting to plot in the same graph.}
reached. Then, pressure increases much faster than for $\xi_0 = 0.01$ while it takes more time for
the temperature to reach the ambient temperature. When we have a subcooled liquid, there
is no gas in the tank, which explains the strange shape of the specific gas enthalpy graph.

Remark. Sometimes it is automatically assumed that $Q$ equals the enthalpy change $\frac{dI}{dt}$
in (6.9), which is indeed valid if $V \frac{dp}{dt} \ll Q$. However, $p$ does change significantly in time. Suppose that we used this assumption,
\[
\frac{dI}{dt} = Q = -K(T - T_{amb}),
\]
(6.26)

instead of (6.10). For the example of Figure 6.3 with $\xi_0 = 0.01$ and $T_0 = 111$ K, we then
obtain a similar behavior, but with a different timescale. This is illustrated by Figure 6.6.

6.4.2 Trapezoidal Rule

We now perform similar calculations using Matlab’s ode23t, a method that uses the trape-
zoidal rule, see Section 3.4. First of all, numerical solutions are calculated for $\xi_0 = 0.01$ and
$T_0 = 111$ K, with a relative tolerance of $10^{-5}$. Tank dimensions are given in Table 6.1 and the
solutions are shown in Figure 6.5. The method breaks down close to the point that all the
liquid is vaporized. The result looks similar to the result found with explicit Euler, Figure 6.3. If we use a relative tolerance of $10^{-6}$ or smaller, the method breaks down much earlier
(typically within a few days of simulated time), because the tolerance cannot be achieved with the
smallest allowed step size. It is not clear whether this is caused by numerical errors, errors in REFPROP or by the assumptions we made, or a combination of these things. Probably,
the inconsistency of the assumption that increasing hydrostatic pressure to the bottom of the
tank is negligible leads to this breakdown. Like in Chapter 5 for explicit Euler, numerical
errors are larger if the quality is close to 0 or 1.

For a larger relative tolerance of $10^{-2}$, the method does not break down because of errors
larger than the requested tolerance. However, after the switch from the two-phase to the
single-phase state, the method fails to converge, because of the discontinuities in derivatives
of some variables and the discontinuity in the quality output value of REFPROP. Hence, no
output is given at all. So we have to set the maximum simulated time small enough to avoid
a superheated vapor or supercritical state.

We observe similar problems for initial values of $\xi_0 = 0.001$ and $T_0 = 111$ K. Again, the
graphs look the same as for explicit Euler, and a breakdown close to $t = t_{one}$ is found for
requested tolerances in the range of $10^{-5}$ to $10^{-2}$. For larger tolerances, the method does not
break down because of tolerance problems, but fails to converge. For smaller tolerances, the
breakdown occurs already much earlier.
6.4 Numerical Results

Figure 6.3: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy and liquid mass in a one-tank system are shown, solved by explicit Euler for $\Delta t = 1$ day, for $\xi_0 = 0.01$ and $T_0 = 111$ K. After approximately 110 weeks, all liquid is vaporized and temperature increases faster, since no energy is needed anymore to vaporize liquid. Note that the quality graph has a different time axis than the other graphs.
Figure 6.4: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy and liquid mass in a one-tank system are shown, solved by explicit Euler for $\Delta t = 1$ day, for $\xi_0 = 0.001$ and $T_0 = 111$ K. After approximately 110 weeks, all vapor is condensed. Note that the quality graph has a different time axis than the other graphs.
Figure 6.5: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy, liquid mass and quality in a one-tank system are shown, solved by Matlab’s trapezoidal method ode23t with relative tolerance of $10^{-5}$, for $\xi_0 = 0.01$ and $T_0 = 111$ K.
Figure 6.6: The evolution of the gas pressure if (6.26) is used instead of (6.10). The graph looks similar to the gas pressure graph in Figure 6.3, but the timescale differs by about 5%.

6.4.3 NDF Method *ode15s*

Investigating the results for $T_0 = 111$ K and $\xi_0 = 0.01$ or $\xi_0 = 0.001$ with Matlab function *ode15s*, we can draw the same conclusions as for *ode23t*: the method gives similar results to the explicit Euler method and *ode23t* as long as $t$ is not too close to $t_{\text{one}}$. Depending on the used relative tolerance,

- the method breaks down close to $t = t_{\text{one}}$ (typically for tolerances between $10^{-1}$ and $10^{-5}$ in this case),
- the method fails to converge for $t > t_{\text{one}}$ (typically for tolerances larger than $10^{-1}$), and no output is given,
- or the method breaks down very early, i.e. within a few days of simulated time (for tolerances smaller than $10^{-5}$).

Changing some built-in options (using backward differentiation instead of numerical differentiation formulas, or using other values for the maximum order of the methods), does not solve the problem around $t = t_{\text{one}}$. Therefore, we make use of the *Event* option, like we did in the previous chapter for the tank filling scenario.

6.4.4 Event Option

Using the *Event* option, we stop the integration as close as possible to $t = t_{\text{one}}$, and restart with equations for the one-phase system. So the events that cause a switch are $\xi = 0$, $\xi = 1$ and a breakdown of the current method. The initial variables for the restarted integration are equal to the variables for the stopped integration. Only the quality is set equal to the quality of the new single-phase we are in (superheated vapor, subcooled liquid or supercritical state). We repeat this restart if we switch from subcooled liquid or superheated vapor to the
supercritical state.
Since we want to keep the error in making these switches as small as possible, we need to choose appropriate tolerances. As we have seen for ode23t and ode15s without the Event option, the tolerances should not be chosen too small, because the method will break down very early then. However, choosing a larger relative tolerance has the obvious drawback of giving less accurate solutions. Ideally, an adaptive approach is used, starting with very strict tolerances, and making them gradually larger to get as close as possible to $\xi = 0$ or $\xi = 1$. We will not implement this though, but simply choose tolerances in the range of $10^{-2}$ to $10^{-5}$ as suggested above. The corresponding program code is found in Appendix C.3.2.

Figures 6.7 and 6.8 show the results for initial values of $T_0 = 111$ K and $\xi_0 = 0.01$, using ode15s, and $\xi_0 = 0.001$, using ode23t, respectively. Both calculations are performed with relative tolerances of $10^{-3}$. Indeed, the results look the same as for explicit Euler, see Figures 6.3 and 6.4.

6.4.5 Speed and Accuracy Comparison

To make a fair comparison of the speed and accuracy of the three methods, we use initial values $\xi_0 = 0.01$ and $T_0 = 111$ K as in Figures 6.3-6.7 as an example, and run the simulations for $t = 100$ weeks. Doing this way, we avoid the switch to the one-phase state, which can have large influences on the calculation times and accuracy, depending on the relative tolerance. To compare the accuracy of the methods, we take the calculated liquid height at $t = 100$ weeks as a reference.

For the explicit Euler method, speed and order of convergence are shown in Table 6.2. The liquid height at $t = 700$ days is denoted by $H_k$ for a step size of $10^{3-k}$ days. The method is first-order convergent and the calculation time is proportional to the number of time steps, as expected, see Chapter 5.

In Table 6.3 calculation times and error estimates are given for ode15s and ode23t. The calculated liquid height at $t = 100$ weeks is denoted by $H_l$ for a relative tolerance of $10^{-l}$. Error estimates are found by comparing $H_l$ with the liquid height calculated by explicit Euler with a time step of $10^{-5}$ days, denoted by $H$.

Both ode15s and ode23t are unable to find a solution for relative tolerances of $10^{-6}$ or smaller for this example. The speed of both methods is comparable for tolerances between $10^{-4}$ and $10^{-2}$, but ode23t is significantly faster for tolerances of $10^{-1}$ and $10^{-5}$.

Remark. Note that the calculations in this chapter could have been accelerated by leaving out the equations that are not essential to solving the model, (6.13)-(6.18). Then, we would have to calculate 6 equations less per time point, from which 4 are calls to REFPROP, giving an expected speedup of 67%.

Assuming that the REFPROP calls are by far the most time consuming equations, and that all REFPROP calls are equally time consuming.

---

6 Assuming that the REFPROP calls are by far the most time consuming equations, and that all REFPROP calls are equally time consuming.
Figure 6.7: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy, liquid mass and quality in a one-tank system are shown, solved by ode15s with Event option and relative tolerances of $10^{-3}$. Initial values are $\xi_0 = 0.01$ and $T_0 = 111$ K. Note the different time axis of the quality graph.
Figure 6.8: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy, liquid mass and quality in a one-tank system are shown, solved by ode23t with Event option and relative tolerances of $10^{-3}$. Initial values are $\xi_0 = 0.001$ and $T_0 = 111$ K. Note the different time axis of the quality graph.
Table 6.2: Speed and order of convergence for explicit Euler method for one-tank system with $\xi_0 = 0.01$ and $T_0 = 111$ K. $H_k$ is the solution of the liquid height in tank 1 at $t = 700$ days for $\Delta t = 10^{3-k}$ days.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Calculation time (s)</th>
<th>$H_{k+1} - H_k$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.9e-1</td>
<td>1.12e0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>6.1e-1</td>
<td>7.98e-2</td>
<td>1.146</td>
</tr>
<tr>
<td>3</td>
<td>4.4e0</td>
<td>7.76e-3</td>
<td>1.012</td>
</tr>
<tr>
<td>4</td>
<td>4.3e1</td>
<td>7.73e-4</td>
<td>1.001</td>
</tr>
<tr>
<td>5</td>
<td>4.4e2</td>
<td>7.73e-5</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>4.5e3</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.3: Calculation times $t_c$ in seconds, error estimates of the liquid height at $t = 700$ days for Matlab’s ODE solvers, for relative tolerances of $10^{-l}$. Initial values are $\xi_0 = 0.01$ and $T_0 = 111$ K. In case the method does not converge, this is denoted by NC.

<table>
<thead>
<tr>
<th></th>
<th>ode23t</th>
<th></th>
<th>ode15s</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$l$</td>
<td>$t_c$(s)</td>
<td>$H_l - H$</td>
<td>$t_c$(s)</td>
<td>$H_l - H$</td>
</tr>
<tr>
<td>1</td>
<td>1.6e0</td>
<td>-2.57e-3</td>
<td>9.5e0</td>
<td>-1.09e-2</td>
</tr>
<tr>
<td>2</td>
<td>3.7e0</td>
<td>-6.10e-3</td>
<td>2.5e0</td>
<td>-1.87e-2</td>
</tr>
<tr>
<td>3</td>
<td>4.2e0</td>
<td>-4.97e-3</td>
<td>3.2e0</td>
<td>-2.88e-3</td>
</tr>
<tr>
<td>4</td>
<td>5.2e0</td>
<td>-9.11e-4</td>
<td>4.8e0</td>
<td>-9.02e-5</td>
</tr>
<tr>
<td>5</td>
<td>1.3e0</td>
<td>-1.80e-4</td>
<td>4.6e1</td>
<td>-5.68e-5</td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
</tbody>
</table>

6.5 Conclusion

In this chapter, we discussed a model for the one-tank system. Good numerical solutions were found with both the explicit Euler method and the Matlab solvers ode15s and ode23t, especially as long as we did not get close to the switch to a one-phase state. Therefore, we are confident to find good results for the two-tank system with variable temperature as well, by combining the equations of state for the one-tank model with equations for fluid flow of Chapter 5 extended with heat exchange via the pipes. Nevertheless, we need to be careful, as we have seen that requested tolerances can not always be achieved. On the other hand, using large tolerances does not guarantee convergence because the errors are also larger than for small tolerances, which might lead to large inconsistencies.
Chapter 7

Two-Tank System with Variable Temperature

In this chapter, we add the ideas of modeling variable temperature of Chapter 6 to the model of the two-tank system of Chapters 4 and 5. An important assumption of Chapter 6 is that heat is spread instantaneously and evenly over a tank by assuming a small Biot number. Given the mass of liquid and gas per tank, we can calculate other properties of interest, like gas density, vapor pressure, etcetera, with the help of REFPROP [18]. For simplicity, we now assume that the boundary of the system is adiabatic, so we do not have any heat exchange with the surroundings. However, the temperatures of the fluids can differ per tank now, so heat is exchanged via the pipes connecting the two tanks.

7.1 Péclet Number

We assume that advection is dominating this heat transfer in the pipes, so that diffusive heat flow is negligible. This is often described by the Péclet number $Pe$, which is given by the ratio of advection and diffusion:

$$Pe = \frac{VL}{\alpha},$$

(7.1)

where $V$ is the velocity, $L$ a characteristic length scale (pipe length in this case) and $\alpha$ the thermal diffusivity, given by

$$\alpha = \frac{\kappa}{\rho c_p},$$

(7.2)

with $\kappa$ the thermal conductivity, which we already encountered in Fourier’s law (2.35), $\rho$ the density and $c_p$ the specific heat capacity, [$c_p$] = J·kg$^{-1}$K$^{-1}$. For methane, $c_p$ is typically in the range of 2000-3500 J·kg$^{-1}$K$^{-1}$ and $\kappa$ is between 0.01 and 0.2 W·m$^{-1}$K$^{-1}$. Since $\rho$ is between 1 and 440 kg·m$^{-3}$ for temperatures between 100 and 180 K, at atmospheric pressure, we find a thermal diffusivity of the order of $10^{-4}$ to $10^{-7}$ m$^2$s$^{-1}$. For pipe lengths of the order of 1 m, we have a large Péclet number, even for flow velocities of $10^{-2}$ m·s$^{-1}$. So we conclude that $Pe \gg 1$, and heat transport is indeed dominated by advection.
Remark. Note that if we get close to the equilibrium state of the two-tank system, the
.typical velocity in the pipes will go to zero. The assumption does not hold anymore then, so
we have to be careful with this situation in our model and in the interpretation of the results.

7.2 Equations and Variables

As in the previous chapter, we use REFPROP to describe the equations of state per tank.
In fact, the set of equations (6.11)-(6.18) is extended to two tanks. Furthermore, we have
conservation of the total mass (liquid and gas) per tank,
\[
\frac{dm_1}{dt} = \varphi_g + \varphi_l; \quad (7.3a)
\]
\[
\frac{dm_2}{dt} = -\frac{dm_1}{dt} = -\varphi_g - \varphi_l. \quad (7.3b)
\]
Here, \(\varphi_g\) and \(\varphi_l\) denote the gas and liquid mass flux, respectively. We assume that only liquid
flows through the lower pipe and only gas flows through the upper pipe. Since in general we
have \(\varphi = \rho A v\), the mass fluxes \(\varphi_g\) and \(\varphi_l\) are determined from Bernoulli’s equations for the
flow in lower and upper pipe:
\[
v_L = \begin{cases}
\sqrt{\frac{2}{(1+\alpha_L)\rho_l,1}(p_{in} - p_{out})} & \text{if } p_{in} > p_{out}; \\
-\sqrt{\frac{2}{(1+\alpha_L)\rho_l,1}(p_{out} - p_{in})} & \text{if } p_{in} < p_{out}; \\
0 & \text{if } p_{in} = p_{out}.
\end{cases} \quad (7.4a)
\]
\[
v_U = \begin{cases}
\sqrt{\frac{2}{(1+\alpha_U)\rho_g,2}(p_1 - p_2)} & \text{if } p_1 > p_2; \\
-\sqrt{\frac{2}{(1+\alpha_U)\rho_g,1}(p_2 - p_1)} & \text{if } p_1 < p_2; \\
0 & \text{if } p_1 = p_2.
\end{cases} \quad (7.4b)
\]
In (7.4a), the pressure difference \(p_{in} - p_{out}\) over the lower pipe is given by
\[
p_{in} - p_{out} = p_1 - p_2 + \rho_{l,1}g(\max\{H_1 - H_{L1,1}, 0\}) - \rho_{g,1}g(\max\{H_2 - H_{L1,1}, 0\}).
\]
As mentioned, the heat transport between the tanks is dominated by advection in our model,
and we will therefore neglect possible diffusive heat transport between the tanks. Inside the
tanks, we assume instantaneous distribution of heat over the tank, corresponding to a very
small Biot number, \(Bi \ll 1\), which means that we have a quasi-equilibrium state in each
tank. The flow is considered as quasi-steady\(^1\) and adiabatic, and therefore it is isenthalpic:
specific enthalpy of both liquid and gas are conserved when flowing through the pipe and valve\(^2\). Mass flow through the pipes is modeled by Bernoulli’s equation\(^2\), so by calculating the
specific liquid and gas enthalpy \(h_{l,i}\) and \(h_{g,i}\) in each tank \(i\), the enthalpy changes \(dI/dt\) per tank are
\[
\frac{dI_1}{dt} = h_{l,i}\rho_{l,i}A_L v_L + h_{g,i}\rho_{g,i}A_U v_U; \quad (7.5a)
\]
\[
\frac{dI_2}{dt} = \frac{dI_1}{dt}. \quad (7.5b)
\]
\(^1\)See the derivation of Bernoulli’s equation in Chapter 4.
\(^2\)We assume no heat flow over the walls of the pipe, so Bernoulli’s equation can still be applied.
In (7.5a), \( i \) and \( j \) depend on the direction of the flow: \( i = 1, j = 1 \) if the flow is from tank 2 to tank 1 and \( i = 2, j = 2 \) if the flow is from tank 1 to tank 2. Calculation of specific enthalpy is done in the same way as in Chapter 6, using REFPROP as in equations (6.17)-(6.18). We assume that specific enthalpy is uniform throughout the liquid per tank and throughout the gas per tank. Furthermore, we assume that the gas density is uniform in the gas per tank, and the liquid density is uniform in the liquid per tank. Like in the previous chapter, we neglect increasing hydrostatic pressure to the bottom of the tanks in the calculation of the quality.

7.3 Numerical Solution

We will compute and compare numerical solutions of this model using the explicit Euler method (with additional constraints), \( \text{ode23t} \) and \( \text{ode15s} \). The geometrical parameters of the tanks are the same as in Table 5.1 and the fluid is pure methane.

7.3.1 Explicit Euler with Constraints

First, we try to solve the problem using the explicit Euler method for equations (7.3) and (7.5), with the additional constraint (5.16). We start with two thermodynamic quantities per tank and calculate the other quantities of interest with equations (6.11)-(6.18). So for each tank we have eight algebraic equations plus two differential equations, making a total of twenty equations for twenty unknowns. In our examples, we start with initial quality and initial temperature, but the program is easily adapted to the usage of other initial variables. See Appendix C.4.1 for the program code.

In Figure 7.1 results are shown for the explicit Euler method with constraints and initial values \( \xi_1 = \xi_2 = 0.02 \) and \( T_1 = 112 \) K and \( T_2 = 125 \) K, for a time step \( \Delta t = 0.1 \) s. We observe that gas pressures are more or less proportional to the temperatures in the tank. Gas pressures and temperatures in the tanks become (almost) equal at \( t = 25 \) s, while it takes about 125 s to reach the equilibrium state. This is not very surprising, as we also saw in Chapter 5 that gas flow was much faster than liquid flow, for the same tank dimensions. It is interesting to see that directly after starting the simulation, the quality in both tank 1 and tank 2 increases. In the beginning, liquid as well as gas flows from tank 2 to tank 1 and \( \xi_2 \) increases fast compared to \( \xi_1 \). The gas and liquid flowing from tank 2 to tank 1 have a higher temperature than the fluid already present in tank 1, making it possible that \( \xi_1 \) is also increasing.

\(^{3}p_i, H_i, I_i, p_{g,i}, p_{l,i}, \xi_i, m_i, h_{g,i}, h_{l,i}, T_i \) for \( i \in \{1, 2\} \)
Figure 7.1: Liquid height, pressure at \( z = 0 \), gas pressure, temperature and quality history for two-tank system with variable temperature, solved by explicit Euler with constraint (5.16) for a step size of \( \Delta t = 0.1 \) s. Initial qualities are \( \xi_{1,0} = \xi_{2,0} = 0.02 \) and initial temperatures are \( T_{1,0} = 112 \) K and \( T_{2,0} = 125 \) K.
7.3 Numerical Solution

7.3.2 Matlab Solvers

Again, we also investigate the performance of \texttt{ode15s} and \texttt{ode23t}. To avoid oscillations when equilibrium is reached, and to speed up the method from that point, the Event option is used to detect when equilibrium is reached. The event “occurs” when both of the following conditions hold for the first time:

\[
\frac{|p_1 - p_2|}{p_1} < \text{reltol} \quad \text{AND} \quad \frac{|p_{in} - p_{out}|}{p_{in}} < \text{reltol}.
\]

After this event, all variables are kept constant. The program code for these methods is given in Appendix C.4.2.

Numerical solutions using \texttt{ode23t} are given in Figures 7.2-7.4. In the first two figures, initial values of \(\xi_{1,0} = 0.01, \xi_{2,0} = 0.02, T_{1,0} = 111 \text{ K}\) are used with an initial temperature in tank 2 of \(T_{2,0} = 135 \text{ K}\) and \(T_{2,0} = 111 \text{ K}\) and relative tolerances of \(10^{-4}\) and \(10^{-2}\), respectively. In the first case, we observe a similar behavior as in Figure 7.1: temperature and gas pressures are proportional and converge significantly faster than the liquid heights. The qualities in both tanks are increasing again for \(t < 20 \text{ s}\). In Figure 7.3, where initial temperatures are equal, the graphs of liquid height, pressure at \(z = 0\) and quality look fine. The graphs of the temperature and gas pressure however, do not look very smooth. Oscillations occur just within the margin of the relative tolerances. Therefore, it is not very surprising that the method breaks down for this example if relative tolerances of \(10^{-3}\) or smaller are used. In the case of equal initial temperatures, we could as well use the methods presented in Chapter 5, where a constant temperature is assumed.

In Figure 7.4, for initial values of \(\xi_{1,0} = 0.008, \xi_{2,0} = 0.004, T_{1,0} = 120 \text{ K}\) and \(T_{2,0} = 110 \text{ K}\), the pressure difference \(p_{in} - p_{out}\) is always positive, in contrast with the pressure difference in Figure 7.2 but in the graphs of liquid height and pressure at \(z = 0\) one can still clearly see when gas pressures and temperatures are equal.

Similar conclusions can be drawn if \texttt{ode15s} is used. Depending on the initial values of the temperature, relative tolerances of \(10^{-2}\) (for equal or almost equal temperatures) to \(10^{-4}\) can be achieved. Figures 7.5-7.6 show solutions using \texttt{ode15s}, for the same initial values as in Figures 7.2-7.4. A relative tolerance of \(10^{-3}\) is used. For relative tolerances of \(10^{-4}\) and \(10^{-5}\), we found the same results, but with a breakdown close to the equilibrium point. For a tolerance of \(10^{-6}\) or smaller, the method breaks down within a few seconds of simulated time.
Figure 7.2: Liquid height, pressure at $z = 0$, gas pressure, temperature and quality history for two-tank system with variable temperature, solved by ode23t with a relative tolerance of $10^{-4}$. Initial qualities are $\xi_{1,0} = 0.01$, $\xi_{2,0} = 0.02$ and initial temperatures are $T_{1,0} = 111$ K and $T_{2,0} = 135$ K.
Figure 7.3: Liquid height, pressure at $z = 0$, gas pressure, temperature and quality for two-tank system with variable temperature, solved by ode23t with a relative tolerance of $10^{-2}$. Initial qualities are $\xi_{1,0} = 0.01$, $\xi_{2,0} = 0.02$ and initial temperatures are $T_{1,0} = T_{2,0} = 111$ K.
Figure 7.4: Liquid height, pressure at $z = 0$, gas pressure, temperature and quality for two-tank system with variable temperature, solved by ode23t with a relative tolerance of $10^{-3}$. Initial qualities are $\xi_1,0 = 0.008$, $\xi_2,0 = 0.004$ and initial temperatures are $T_1,0 = 120$ K and $T_2,0 = 110$ K.
Figure 7.5: Liquid height, pressure at \( z = 0 \), gas pressure, temperature and quality for two-tank system with variable temperature, solved by \texttt{ode15s} with a relative tolerance of \( 10^{-2} \). Initial qualities are \( \xi_{1,0} = 0.01 \), \( \xi_{2,0} = 0.02 \) and initial temperatures are \( T_{1,0} = 111 \text{ K} \) and \( T_{2,0} = 135 \text{ K} \).
Figure 7.6: Liquid height, pressure at $z = 0$, gas pressure, temperature and quality for two-tank system with variable temperature, solved by \texttt{ode15s} with a relative tolerance of $10^{-2}$. Initial qualities are $\xi_{1,0} = 0.01$, $\xi_{2,0} = 0.02$ and initial temperatures are $T_{1,0} = T_{2,0} = 111 \text{ K}$.
7.3 Numerical Solution

7.3.3 Speed and Accuracy Comparison

For the example with $T_{1,0} = 112$ K, $T_{2,0} = 125$ K and $\xi_{1,0} = \xi_{2,0} = 0.02$, speed and order of convergence for the explicit Euler method with constraints are given in Table 7.1. We compare the calculated liquid height in tank 1 at $t = 20$ s, which we denote by $H_k$, for a used step size of $10^{-k}$ s. We observe that the calculation time is more or less linear, and that the convergence order is equal to 1. In Table 7.2 calculation times are shown for the same example, now solved by ode23t and ode15s. Unfortunately, the methods do not converge for relative tolerances of $10^{-5}$ or smaller. For tolerances of $10^{-3}$ and larger, the explicit Euler method with constraint is as fast as the Matlab solvers or even faster.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Calculation time (s)</th>
<th>$\frac{H_{k+1}-H_k}{H_{k+1}}$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>4.1e-1</td>
<td>1.19e-2</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>2.6e0</td>
<td>4.30e-4</td>
<td>1.444</td>
</tr>
<tr>
<td>1</td>
<td>2.6e1</td>
<td>4.40e-5</td>
<td>0.989</td>
</tr>
<tr>
<td>2</td>
<td>2.5e2</td>
<td>4.41e-6</td>
<td>0.999</td>
</tr>
<tr>
<td>3</td>
<td>2.6e3</td>
<td>4.41e-7</td>
<td>1.000</td>
</tr>
<tr>
<td>4</td>
<td>2.7e4</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.1: Speed and order of convergence $p$ for explicit Euler method with constraint (5.16) for the two-tank system with variable temperature. Initial conditions are $T_{1,0} = 112$ K, $T_{2,0} = 125$ K and $\xi_{1,0} = \xi_{2,0} = 0.02$. $H_k$ is the solution of the liquid height in tank 1 at $t = 20$ s for a step size of $10^{-k}$ s.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\text{ode23t}$</th>
<th>$\text{ode15s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.3</td>
<td>5.4</td>
</tr>
<tr>
<td>2</td>
<td>6.1</td>
<td>7.0</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>9.4</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
<td>NC</td>
</tr>
</tbody>
</table>

Table 7.2: Calculation times in seconds using $\text{ode15s}$ and $\text{ode23t}$, with relative tolerances of $10^{-l}$. Initial conditions are $T_{1,0} = 112$ K, $T_{2,0} = 125$ K and $\xi_{1,0} = \xi_{2,0} = 0.02$. In case that the method cannot find a solution satisfying the requested tolerance, we denote this by NC.
Chapter 8

Multi-Component Fluids

In the previous chapters, we only considered systems containing single-component fluids. In reality, fluids are often a mixture of several components. This has large effects on the modeling of the problem, especially in situations around the boiling point of a fluid. In the case of multi-component fluids, we may have different evaporation or condensation rates per component, which changes the composition of the liquid and gas parts of the substance. Depending on the production and storage process, LNG is mainly composed of three substances: methane, ethane and nitrogen. Sometimes also heavier alkanes are part of LNG, but in our examples we will only consider the three mentioned components. The mass fractions of methane, ethane and nitrogen in LNG as we will use in our calculations are 0.9, 0.09 and 0.01 respectively. The corresponding molar fractions as calculated by REFPROP are given in Table 8.1. These calculations are based on the so-called GERG model [16].

<table>
<thead>
<tr>
<th>component</th>
<th>molar fraction (mol/mol)</th>
<th>mass fraction (kg/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>methane</td>
<td>0.943648747220886</td>
<td>0.9</td>
</tr>
<tr>
<td>ethane</td>
<td>5.0346958769393e-02</td>
<td>0.09</td>
</tr>
<tr>
<td>nitrogen</td>
<td>6.0045569021752e-03</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 8.1: LNG composition as used in our REFPROP calculations of this chapter.

8.1 Numerical Results

Performing calculations with mixtures instead of single-component fluids using REFPROP leads to an increase of the calculation time. Since REFPROP is designed for accuracy and completeness rather than speed, this effect is large. Some comparisons are given in Section 8.2. For further speed-up, we can think of a meta-model by interpolating REFPROP results as has been done in [17].
8.1.1 One-Tank Model

In the one-tank model of Chapter 6, we can simply change the single-component substance into a multi-component substance. We therefore assume the liquid and gas to be mixtures of several components. More specifically, we start with LNG as given in Table 8.1 and tank dimensions and ambient temperature as in Table 6.1. Given two independent state variables (for example pressure and temperature), REFPROP calculates the mass fractions of the components in the liquid and the gas. Let us denote these fractions by

\[
\begin{align*}
X_{\text{liq}} &= \left( X_{\text{CH}_4 \text{liq}}, X_{\text{C}_2\text{H}_6 \text{liq}}, X_{\text{N}_2 \text{liq}} \right), \\
X_{\text{gas}} &= \left( X_{\text{CH}_4 \text{gas}}, X_{\text{C}_2\text{H}_6 \text{gas}}, X_{\text{N}_2 \text{gas}} \right). \\
\end{align*}
\]

Clearly, the sum of both the mass fractions in the liquid and the mass fractions in the gas should be equal to 1: \( \|X_{\text{liq}}\|_1 = \|X_{\text{gas}}\|_1 = 1 \), where \( \| \cdot \|_1 \) denotes the 1-norm. Keeping track of the compositions of the liquid and gas separately is of particular interest in the two-tank system, in order to examine the componentwise mass flux in the upper pipe and lower pipe by using the gas compositions and liquid compositions, respectively.

For instance, starting with a pressure of 1 bar and a temperature of 111 K, REFPROP gives the following compositions of the liquid and gas, see Table 8.2, with a corresponding vapor fraction \( \xi \) of approximately 0.0311. The boiling points of methane, ethane and nitrogen for a pressure of 1 bar are approximately 112 K, 184 K and 77 K respectively. Therefore, the result in Table 8.2 is as expected: the mass fraction of ethane in the liquid is much larger than in the gas, and the mass fraction of nitrogen is much larger in the gas than in the liquid phase.

<table>
<thead>
<tr>
<th>component</th>
<th>mass fraction in liquid (kg/kg)</th>
<th>mass fraction in gas (kg/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>methane</td>
<td>9.015265683755455e-001</td>
<td>8.524744450897586e-001</td>
</tr>
<tr>
<td>ethane</td>
<td>9.288407101275754e-002</td>
<td>1.60385865077948e-004</td>
</tr>
<tr>
<td>nitrogen</td>
<td>5.590242611696787e-003</td>
<td>1.473651690451636e-001</td>
</tr>
</tbody>
</table>

Table 8.2: LNG composition in liquid and gas phase for \( p = 1 \text{ bar} \) and \( T = 111 \text{ K} \) for the total composition of LNG of Table 8.1

Explicit Euler method

If we take initial values for temperature and quality of \( T_0 = 111 \text{ K} \) and \( \xi_0 = 0.0311 \), and perform similar simulations as in Chapter 6 using the explicit Euler method with a time step of 0.1 week, we obtain the results given in Figures 8.1-8.2. The corresponding program code is given in Appendix C.5.1. Unfortunately, REFPROP fails to give output values of the quality after 37 weeks, so we have to stop our simulation there. This is caused by incompleteness of REFPROP, see Appendix B.3, and whether or not this occurs depends on the chosen initial values. For initial values of \( T_0 = 111 \text{ K} \) and \( \xi_0 = 0.05 \) for example, we luckily do not have to stop or simulation early, see Figures 8.3-8.4. Again, a time step of 0.1 weeks is used.

---

1The 1-norm, also known as the taxicab norm is defined by \( \|a\|_1 = \sum_i |a_i| \).
The resulting compositions in Figure 8.2 are as one can expect. With the temperature increasing from 111 K to 170 K, methane will have the highest rate of evaporation in the beginning since its normal boiling point is around 112 K. Ethane, which has a higher boiling point, will become more likely to evaporate if temperature and pressure are further increased. Nitrogen has a boiling point far below the initial temperature, so a further increase in temperature and pressure does not result in a high evaporation or condensation rate: most of the nitrogen is already in gaseous state.

Similar conclusions can be drawn from Figures 8.3-8.4. When the superheated vapor state is reached, the gas compositions are equal to the initial composition of the entire LNG, \( X_0 = (0.9 \ 0.09 \ 0.01)^T \), as one should expect.

**Matlab Solvers**

Using ode15s and ode23t, we obtain alike results. As we have seen in the previous chapters, we need to be careful with the choice of our relative tolerances. For these multi-component fluids, the methods are more fragile, because there is a decent chance of encountering a gap in REFPROP, as mentioned above, see Appendix B. However, for requested tolerances larger than \( 10^{-4} \), we are in many cases able to find solutions. An example is given in Figures 8.5-8.6 where initial values of \( T_0 = 111 \) K and \( \xi_0 = 0.05 \) are used, solved by ode23t with a relative tolerance of \( 10^{-3} \). Indeed, these graphs look the same as the graphs in Figures 8.3-8.4, where explicit Euler is used for the same initial conditions.

8.1.2 Two-Tank Model

For the two-tank model, we have to reconsider the flow in the pipes, as the compositions of the gas and liquid can differ per tank now. This makes the problem much more complicated, but the results can be of particular interest, e.g. if the fraction of certain components have to be of some minimum or maximum amount. Therefore, we have to keep track of the methane, ethane and nitrogen fractions in both the liquid and the gas of each tank, denoted by the vectors \( X_{\text{liq},i} \) and \( X_{\text{gas},i} \) for \( i \in \{1, 2\} \), see (8.1). Analogous to the enthalpy conservation equations (7.5) in Chapter 7, we have conservation of mass per component \( c \):

\[
\frac{dm_i^c}{dt} = X_{\text{liq},i}^c \rho_{\text{liq},i} A_L v_L + X_{\text{gas},j}^c \rho_{\text{gas},j} A_U v_U; \quad (8.2a)
\]

\[
\frac{d m_i^c}{dt} = - \frac{d m_j^c}{dt}. \quad (8.2b)
\]

Here, \( m_i^c \) denotes the mass of component \( c \) in tank \( i \). In (8.2a), \( i \) and \( j \) depend on the direction of the flow: \( i = 1, j = 1 \) if the flow is from tank 2 to tank 1 and \( i = 2, j = 2 \) if the flow is from tank 1 to tank 2. So we have 6 additional variables \( m_i^c \) for the 6 equations (8.2) and 4 additional vectors consisting of 3 elements with the compositions in liquid and gas in both tank 1 and tank 2, together with their corresponding equations of state given by REFPROP.

**Explicit Euler**

The implementation of the additional variables and equations is quite straightforward, see Appendix C.5.3. The remarks made in previous chapters about stability and accuracy if the
Figure 8.1: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy, liquid mass and quality in a one-tank system, solved by explicit Euler for $\Delta t = 0.1$ week. Initial values are $T_0 = 111$ K and $\xi_0 = 0.0311$.

Pressure differences become small still hold here. Results with explicit Euler with constraint (5.16) are shown in Figures 8.7-8.8, for $T_1,0 = 112$ K, $T_2,0 = 125$ K and $\xi_1,0 = \xi_2,0 = 0.02$ and a time step of 1 s. We observe several similarities with the solution for the single-component
system with the same initial values, see Figures 7.1. The timescales for the pressures to become equal are almost the same. An important difference is seen from the moment that the pressures are equal. The liquid heights and temperatures are unequal in this case. The reason is that the compositions in the tanks are not the same anymore. Therefore, the pressures over the lower and upper pipe can be equal, while liquid heights and temperatures are unequal. Since we only have convective heat transfer between the tanks in our model, the dynamic equilibrium causes the temperatures to remain constant. A heat conduction model can be added to describe the heat transfer when $Pe \gg 1$ does not hold anymore.

**Matlab Solvers**

Unfortunately, we were unable to build properly working simulations using `ode15s` and `ode23t` for the multi-component two-tank system at the time of writing this report. Therefore, we cannot compare the explicit Euler method with these methods in this case. Hence, we should be reserved with drawing conclusions about the performance of the explicit Euler method here.

![Graphs showing mass fractions of LNG components](image)

**Figure 8.2:** The evolution of the mass fractions of the three LNG components in both the liquid and gas phase in a one-tank system, solved by explicit Euler for $\Delta t = 0.1$ week. Initial values are $T_0 = 111$ K and $\xi_0 = 0.0311$. 
Figure 8.3: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy, liquid mass and quality in a one-tank system, solved by explicit Euler for $\Delta t = 0.1$ week. Initial values are $T_0 = 111$ K and $\xi_0 = 0.05$. Note that the time axis of the quality graph is different from the other graphs.
Figure 8.4: The evolution of the mass fractions of the three LNG components in both the liquid and gas phase in a one-tank system, solved by explicit Euler for $\Delta t = 0.1$ week. Initial values are $T_0 = 111$ K and $\xi_0 = 0.05$. 

\[ X(t) \text{ (kg/kg)} \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]

\[ X_{\text{gas}}(t) \]
Figure 8.5: The evolution of the pressure, temperature, liquid height, liquid and gas enthalpy, liquid mass and quality in a one-tank system, solved by ode23t with a relative tolerance of $10^{-3}$. Initial values are $T_0 = 111$ K and $\xi_0 = 0.05$. 
8.1 Numerical Results

Figure 8.6: The evolution of the mass fractions of the three LNG components in both the liquid and gas phase in a one-tank system, solved by `ode23t` with a relative tolerance of $10^{-3}$. Initial values are $T_0 = 111$ K and $\xi_0 = 0.05$. 
Figure 8.7: The evolution of the pressure, temperature, liquid height, total mass per tank and quality in a two-tank system, solved by explicit Euler with constraint (5.16) with a time step of 1 s. Initial values are \( T_{1,0} = 112 \) K, \( T_{2,0} = 125 \) K and \( \xi_{1,0} = \xi_{2,0} = 0.02 \).
Figure 8.8: The evolution of the mass fractions of the three LNG components in the liquid and gas phase in both tanks of a two-tank system, solved by explicit Euler with constraint (5.16) with a time step of 1 s. Initial values are $T_{1,0} = 112$ K, $T_{2,0} = 125$ K and $\xi_{1,0} = \xi_{2,0} = 0.02$. 
8.2 Speed and Accuracy Comparison

8.2.1 One-Tank System

To compare the speed and accuracy of the explicit Euler method, \texttt{ode23t} and \texttt{ode15s} for the multi-component one-tank system, we use the same initial values as in Figures 8.3-8.6: \( T_0 = 111 \) K and \( \xi_0 = 0.05 \). The simulated time is 35 weeks and values of the liquid height at the end of the simulations are compared. For the explicit Euler method, these values are denoted by \( H_k \), for a corresponding number of time steps of \( 7 \cdot 2^k \). Resulting calculation times, error estimates \( H_{k+1} - H_k \) and order of convergence \( p \) are given in Table 8.3. As expected, the calculation times are proportional to the number of time steps and the method is first-order convergent.

In Table 8.3, calculation times are shown for the same example, solved by \texttt{ode23t} and \texttt{ode15s} for relative tolerances of \( 10^{-l} \). The latter method is faster, especially for stricter tolerances. Unfortunately, both methods are unable to find a solution for \( l \geq 4 \). The solution for \texttt{ode15s} for \( l = 3 \) is denoted by \( H \) and is used to compute error estimates \( H_l - H \). Since the liquid height at \( t = 35 \) weeks is about 0.04 m, the relative error is approximately 25 times larger than \( H_l - H \). \texttt{Matlab}'s ODE solvers are significantly better than the explicit Euler methods for this example.

<table>
<thead>
<tr>
<th>( k )</th>
<th>Calculation time (s)</th>
<th>( H_{k+1} - H_k )</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5e1</td>
<td>1.91e-2</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>3.1e1</td>
<td>1.05e-2</td>
<td>0.862</td>
</tr>
<tr>
<td>3</td>
<td>6.2e1</td>
<td>5.21e-3</td>
<td>1.014</td>
</tr>
<tr>
<td>4</td>
<td>1.2e2</td>
<td>2.60e-3</td>
<td>1.007</td>
</tr>
<tr>
<td>5</td>
<td>2.5e2</td>
<td>1.29e-3</td>
<td>1.004</td>
</tr>
<tr>
<td>6</td>
<td>5.3e2</td>
<td>6.47e-4</td>
<td>1.001</td>
</tr>
<tr>
<td>7</td>
<td>1.1e3</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 8.3: Speed and order of convergence \( p \) for explicit Euler method with constraint (5.16) for the one-tank system with multi-component fluid. Initial conditions are \( T_0 = 111 \) K and \( \xi_0 = 0.05 \). \( H_k \) is the solution of the liquid height in the tank at \( t = 35 \) weeks for \( 7 \cdot 2^k \) time steps.

8.2.2 Two-Tank System

To investigate the speed and accuracy for the multi-component two-tank system with the explicit Euler method with constraint (5.16), we run simulations for \( T_{1,0} = 112 \) K, \( T_{2,0} = 125 \) K and \( \xi_{1,0} = \xi_{2,0} = 0.02 \) like in Figures 8.7-8.8. To obtain results within a reasonable time, we use a simulated time of 10 seconds. The liquid height in tank 1 is used as a reference and is denoted by \( H_k \) for a number of time steps of \( 5 \cdot 2^k \). The results are given in Table 8.5. The calculation time is proportional to the number of time steps, and the method is first-order accurate. With a liquid height in tank 1 at \( t = 10 \) s of 4 m, the relative errors in the table are approximately between 1e-3 for \( k = 0 \) and 2.5e-5 for \( k = 5 \), which is good.
8.3 Conclusion

We observed a large increase in the calculation times for multi-component fluids compared to single-component fluids. Furthermore, REFPROP seems less accurate and less complete in this case. Nevertheless, comparisons of the different methods and comparisons with results in the previous chapters make it plausible that good results were found.

To improve speed, completeness and accuracy, an interpolation model of REFPROP can be helpful. Also less sophisticated equations of state can be considered to replace REFPROP in order to find a good balance between accuracy and speed.

### Table 8.4: Calculation times in seconds using ode15s and ode23t, with relative tolerances of $10^{-l}$.

<table>
<thead>
<tr>
<th>$l$</th>
<th>$t_c(s)$</th>
<th>$H_l - H_l^*$</th>
<th>$t_c(s)$</th>
<th>$H_l - H_l^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2e2</td>
<td>-1.77e-4</td>
<td>1.1e2</td>
<td>-1.74e-4</td>
</tr>
<tr>
<td>2</td>
<td>1.8e2</td>
<td>-9.02e-6</td>
<td>1.4e2</td>
<td>3.16e-4</td>
</tr>
<tr>
<td>3</td>
<td>3.3e2</td>
<td>-4.57e-5</td>
<td>2.2e2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>5</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
</tbody>
</table>

Initial conditions are $T_0 = 111$ K and $\xi_0 = 0.05$. In case that the method cannot find a solution satisfying the requested tolerance, we denote this by NC.

### Table 8.5: Speed and order of convergence $p$ for explicit Euler method with constraint (5.16) for the two-tank system with multi-component fluid. Initial values are $T_{1,0} = 112$ K, $T_{2,0} = 125$ K and $\xi_{1,0} = \xi_{2,0} = 0.02$. $H_k$ is the solution of the liquid height in tank 1 at $t = 10$ s for $5 \cdot 2^k$ time steps.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Calculation time (s)</th>
<th>$\frac{H_{k+1} - H_k}{H_{k+1}}$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.1e1</td>
<td>8.99e-4</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>8.9e1</td>
<td>4.62e-4</td>
<td>0.959</td>
</tr>
<tr>
<td>2</td>
<td>1.5e2</td>
<td>2.58e-4</td>
<td>0.839</td>
</tr>
<tr>
<td>3</td>
<td>3.2e2</td>
<td>1.03e-4</td>
<td>1.327</td>
</tr>
<tr>
<td>4</td>
<td>4.7e2</td>
<td>5.52e-5</td>
<td>0.897</td>
</tr>
<tr>
<td>5</td>
<td>9.0e2</td>
<td>2.55e-5</td>
<td>1.118</td>
</tr>
<tr>
<td>6</td>
<td>1.9e3</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

8.3 Conclusion
Chapter 9

Conclusion

9.1 Summary

In this report, we presented numerical solutions to some examples of LNG systems. For the open two-tank system with only a lower pipe connection, a single-component fluid and constant temperature, we were able to find an exact solution describing the fluid flow, based on the extended Bernoulli equation. Investigating the same situation for a closed system, we added Boyle’s law to describe the gas pressures in the tanks, which resulted in a set of equations that we could only solve numerically.

In Chapter 5, the upper pipe is added to the closed two-tank system, leading to an additional equation governing the gas flow. Using the explicit Euler method can give oscillations in the solution, especially when the liquid level in one of the tanks gets close to the tank height. Modifications to the method in the form of some constraints lead to good and fast results, as well as using some higher order Runge-Kutta and NDF methods. We should note however, that close to such points the presented models are less accurate. For example, the gas volume in a pipe cannot be neglected anymore compared to the gas volume in the tank.

If we allow temperature variations in our model, together with vaporization and condensation of the fluid, we use REFPROP to calculate our state variables. In Chapter 6 this is first done for a one-tank system with external heat flux, assuming a small Biot number. Good results were obtained with both explicit Euler and with the Matlab solvers based on trapezoidal rule (ode23t) and numerical differentiation (ode15s). However, one needs to be careful with the requested relative tolerances to make sure that the methods converge in case of the Matlab solvers, especially near the transition to the single-phase state.

In the extension to the two-tank system, we assumed a large Péclet number, meaning that heat transport in the pipes is dominated by advection. Good results were found with both the explicit Euler method with constraints and ode23t and ode15s, for relative tolerances of at least (roughly) $10^{-4}$. For smaller tolerances, the method can break down early with Matlab’s solvers.
In the case of multi-component fluids, we encounter a large increase of the calculation time. Good results are found for the one-tank system. For the two-tank system, we assume that the exchange of components occurs similar to the heat exchange for a large Péclet number. The results for explicit Euler seem acceptable. Unfortunately, it is harder to obtain convergence, which is also caused by the incompleteness of REFPROP.

9.2 Discussion

In this project, we focused on numerical properties of the model rather than trying to find a very precise solution for specific physical values. For example, the used values of the friction factors $\alpha$ are assumed to have a realistic order of magnitude, but we did not have the intention to use very precise values for a particular situation. So in further usage, these values should be estimated from the Moody diagram or other experimental data, depending on the presence of valves, bends, etcetera [21]. Below, we give a short list of the most important assumptions and simplifications regarding the physical part of the models we used.

- The usage of the extended Bernoulli equation is based on the assumption of quasi-steady flow, so inertia effects are ignored.
- We assumed incompressible flow in the pipes, so one needs to be careful with the results if large pressure differences exist over the upper pipe.
- We assumed $Bi \ll 1$ and $Pe \gg 1$. As we have seen in the discussion of the rollover effect in the Introduction, heat is not always spread instantaneously over the tank. So with the assumption of a small Biot number, we cannot describe all phenomena that may occur in reality. The latter assumption of a small Péclet number is not valid if pressures are almost equal: flow velocity is very small then.
- We did not discretize the tanks, which means that an increasing hydrostatic pressure towards the bottom of the tank is not taken into account in determining the quality per tank, and density in the liquid is assumed to be uniform. If we would discretize the tanks and pipes, boundary conditions will play a more important role. Moreover, we assumed that liquid and gas are strictly separated by a horizontal surface. This is not necessarily true, as for instance vapor may be solved in the liquid, or liquid might be condensed against the tank walls, but is in most cases a good approximation.
- We used several physical parameters (friction factors, heat conduction coefficients), geometrical parameters (tank dimensions, pipe dimensions) and initial values of simulations, for which we estimated realistic values in our examples. We tried to give an extensive overview of the performance of the methods and to evaluate the results depending on the chosen parameters and initial values. Nevertheless, it is impossible to calculate and discuss all imaginable combinations of these values in this report.

Next, we give a brief overview of the most important points of discussion regarding numerical issues.

- Especially for the one-tank model with variable temperature, where we considered the transition from a two-phase to a one-phase state, a correct choice of relative tolerances is essential in order to obtain good results with the Matlab solvers.
• The implementation of REFPROP calls within Mat1ab is easy and REFPROP can give different output values for a large number of combinations of two independent input values and a substance. The user does not have to care about which equation of state would be the best to describe his situation of interest, this is all done by REFPROP. However, this is also one of the major drawbacks of REFPROP in combination with Mat1ab: the user cannot influence the speed of the solution by choosing appropriate equations, and the results should be trusted quite blindly. Moreover, if no convergence of REFPROP is achieved, it is often hard to find out the reason. This is in particular the case if the program is used with Mat1ab’s ODE solvers, since the error may be caused by inaccurate actions done before.

• Since our model is a simplified version of reality, we did not expect to find extremely precise results. However, REFPROP is designed to be very precise, so it is of interest to investigate if this accuracy for solving equations of state is always necessary. Moreover, using REFPROP can also slow down the calculations, especially for mixtures.

9.3 Further Research

Besides ideas for further research that would naturally arise from the discussion points above, we give some possible extensions of our model:

• For the one-tank model, we assumed a constant and uniform heat flux coefficient $k$. In reality however, this coefficient depends on temperature, so it is of interest to make this coefficient temperature dependent. Moreover, the coefficient for the gas phase of the fluid may differ from the value for the liquid phase.

• In order to keep the two-tank model apprehensive, we did not include an external heat flux yet as is done for the one-tank model. Doing this way, we could also keep a distinction between the effects of external heating and heat transfer within the two-tank system. A natural extension would be to include the external heating in the two-tank model. We expect that implementation is straightforward. However, total vaporization or liquefaction in a tank can be harder to predict and to manage.

• One of the things to keep in mind during the development of the models was the importance of the possibility for extensions to larger networks. The main idea would remain the same as for the two-tank system: connections are described by the Bernoulli’s equation, and also heat exchanges and componentwise mass exchanges via the connections are described in this way, using specific liquid and gas enthalpy and the liquid and gas compositions, respectively.

• It would be interesting to investigate a meta-model of REFPROP as has been done in [9] in combination with the programs presented in this report. Furthermore one can try to find out when more simplified equations are good enough to replace REFPROP. This hopefully leads to a speedup of the model, which is particularly important for the multi-component simulations.

\footnote{Fortran source code is accessible, but is very unapprehensive.}
• Eventually, it should be possible to include models describing for instance the water-hammer effect, sloshing of tanks [17], multi-component two-phase flow in the pipes [17, 28].
## Appendix A

### List of Symbols

#### A.1 General

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>friction factor</td>
<td>dimensionless</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>thermal diffusivity</td>
<td>$m^2 s^{-1}$</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>coefficient of heat conduction</td>
<td></td>
</tr>
<tr>
<td>$\mu$</td>
<td>dynamic viscosity</td>
<td></td>
</tr>
<tr>
<td>$\nu$</td>
<td>kinematic viscosity</td>
<td></td>
</tr>
<tr>
<td>$\xi$</td>
<td>quality (mass fraction of vapor)</td>
<td></td>
</tr>
<tr>
<td>$\rho$</td>
<td>density</td>
<td></td>
</tr>
<tr>
<td>$\varphi$</td>
<td>mass flux</td>
<td></td>
</tr>
<tr>
<td>$\Phi$</td>
<td>volume flux</td>
<td></td>
</tr>
<tr>
<td>$\tau$</td>
<td>viscous stress tensor</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>(cross-sectional) area</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>Helmholtz energy</td>
<td></td>
</tr>
<tr>
<td>$Bi$</td>
<td>Biot number</td>
<td>dimensionless</td>
</tr>
<tr>
<td>$c$</td>
<td>specific heat capacity</td>
<td>$m$</td>
</tr>
<tr>
<td>$D$</td>
<td>diameter</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{D}$</td>
<td>deformation velocity tensor</td>
<td>dimensionless</td>
</tr>
<tr>
<td>$f$</td>
<td>(Darcy) friction factor</td>
<td></td>
</tr>
<tr>
<td>$f$</td>
<td>specific or nonspecific surface or volume force</td>
<td></td>
</tr>
<tr>
<td>$h$</td>
<td>specific enthalpy</td>
<td></td>
</tr>
<tr>
<td>$h$</td>
<td>hydraulic head</td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td>height</td>
<td></td>
</tr>
<tr>
<td>$I$</td>
<td>(total) enthalpy</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>gravitational acceleration</td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>heat flux coefficient per unit area</td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>heat flux coefficient of total tank</td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>length</td>
<td></td>
</tr>
</tbody>
</table>

*Continued on next page*
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>mass</td>
<td>kg</td>
</tr>
<tr>
<td>$n$</td>
<td>amount of substance</td>
<td>mol</td>
</tr>
<tr>
<td>$p$</td>
<td>pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>$Pe$</td>
<td>Pécel number</td>
<td>dimensionless</td>
</tr>
<tr>
<td>$q$</td>
<td>conductive heat flux</td>
<td>W·m$^{-2}$</td>
</tr>
<tr>
<td>$Q$</td>
<td>heat transfer rate</td>
<td>W</td>
</tr>
<tr>
<td>$r$</td>
<td>specific heat absorption from internal sources</td>
<td>W·kg$^{-1}$</td>
</tr>
<tr>
<td>$R$</td>
<td>molar gas constant</td>
<td>J·K$^{-1}$·mol$^{-1}$</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>specific gas constant</td>
<td>J·kg$^{-1}$·K$^{-1}$</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number</td>
<td>dimensionless</td>
</tr>
<tr>
<td>$S$</td>
<td>surface area</td>
<td>m$^2$</td>
</tr>
<tr>
<td>$S$</td>
<td>(thermodynamic) entropy</td>
<td>J·K$^{-1}$</td>
</tr>
<tr>
<td>$t$</td>
<td>time</td>
<td>s</td>
</tr>
<tr>
<td>$T$</td>
<td>(thermodynamic) temperature</td>
<td>K</td>
</tr>
<tr>
<td>$\mathcal{T}$</td>
<td>stress tensor</td>
<td>Pa</td>
</tr>
<tr>
<td>$u$</td>
<td>specific internal energy</td>
<td>J·kg$^{-1}$</td>
</tr>
<tr>
<td>$U$</td>
<td>(total) internal energy</td>
<td>J</td>
</tr>
<tr>
<td>$v$</td>
<td>velocity</td>
<td>m·s$^{-1}$</td>
</tr>
<tr>
<td>$V$</td>
<td>volume</td>
<td>m$^3$</td>
</tr>
<tr>
<td>$x$</td>
<td>position</td>
<td>m</td>
</tr>
<tr>
<td>$X$</td>
<td>composition (mass fractions)</td>
<td>kg·kg$^{-1}$</td>
</tr>
<tr>
<td>$z$</td>
<td>height</td>
<td>m</td>
</tr>
</tbody>
</table>
### A.2 Subscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Corresponds to</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>point $a$</td>
</tr>
<tr>
<td>$amb$</td>
<td>ambient</td>
</tr>
<tr>
<td>$atm$</td>
<td>atmospheric</td>
</tr>
<tr>
<td>$b$</td>
<td>point $b$ or body</td>
</tr>
<tr>
<td>$c$</td>
<td>point $c$, calculation or critical</td>
</tr>
<tr>
<td>$d$</td>
<td>point $d$</td>
</tr>
<tr>
<td>$e$</td>
<td>equilibrium</td>
</tr>
<tr>
<td>$f$</td>
<td>friction</td>
</tr>
<tr>
<td>$fm$</td>
<td>straight pipe friction</td>
</tr>
<tr>
<td>$g$</td>
<td>gas</td>
</tr>
<tr>
<td>$gas$</td>
<td>gas</td>
</tr>
<tr>
<td>$i$</td>
<td>interior</td>
</tr>
<tr>
<td>$in$</td>
<td>$z = 0$ in tank 1</td>
</tr>
<tr>
<td>$l$</td>
<td>liquid</td>
</tr>
<tr>
<td>$liq$</td>
<td>liquid</td>
</tr>
<tr>
<td>$m$</td>
<td>mass</td>
</tr>
<tr>
<td>$mol$</td>
<td>molar</td>
</tr>
<tr>
<td>$p$</td>
<td>constant pressure</td>
</tr>
<tr>
<td>$L$</td>
<td>lower pipe</td>
</tr>
<tr>
<td>$L1$</td>
<td>lower pipe and tank 1</td>
</tr>
<tr>
<td>$L2$</td>
<td>lower pipe and tank 2</td>
</tr>
<tr>
<td>$one$</td>
<td>one-phase</td>
</tr>
<tr>
<td>$out$</td>
<td>$z = 0$ in tank 2</td>
</tr>
<tr>
<td>$r$</td>
<td>reduced</td>
</tr>
<tr>
<td>$s$</td>
<td>surface or stability</td>
</tr>
<tr>
<td>$T$</td>
<td>tank</td>
</tr>
<tr>
<td>$TOT$</td>
<td>total system</td>
</tr>
<tr>
<td>$T1$</td>
<td>tank 1</td>
</tr>
<tr>
<td>$T2$</td>
<td>tank 2</td>
</tr>
<tr>
<td>$U$</td>
<td>upper pipe</td>
</tr>
<tr>
<td>$v$</td>
<td>volume</td>
</tr>
<tr>
<td>$0$</td>
<td>$t = 0$</td>
</tr>
<tr>
<td>$1$</td>
<td>tank 1</td>
</tr>
<tr>
<td>$2$</td>
<td>tank 2</td>
</tr>
</tbody>
</table>
List of Symbols
Appendix B

REFPROP

B.1 General

In this report we make use of the software package REFPROP \[18\] to calculate state variables. REFPROP, acronym for REFerence fluid PROPerties, is a program written by NIST, the National Institute of Standards and Technology. The program provides tables and plots of the thermodynamic and transport properties of industrially important fluids and their mixtures with an emphasis on refrigerants and hydrocarbons. Integrated in Matlab, REFPROP can be used to calculate requested state variables, given two independent state variables and a substance, possibly a mixture. Moreover, the substance can be in the two-phase region. The package comes with several options for including it in other programs or programming codes as Excel, Python and Matlab. The source code is written in Fortran. The following description is from the website of NIST \[3\]:

"The REFPROP "database" is actually a program and does not contain any experimental information, aside from the critical and triple points of the pure fluids. The program uses equations for the thermodynamic and transport properties to calculate the state points of the fluid or mixture. These equations are the most accurate equations available worldwide. [...]
Their high accuracy is obtained through many coefficients in the equations, and thus the calculation speed will be slower than other equations such as the Peng-Robinson cubic equations. The equations are generally valid over the entire vapor and liquid regions of the fluid, including supercritical states; the upper temperature limit is usually near the point of decomposition of the fluid, and the upper pressure (or density) limit is defined by the melting line of the substance."

B.2 Used Models

A little more background of the models is given in the help function of the program:

"REFPROP is based on the most accurate pure fluid and mixture models currently available. It implements three models for the thermodynamic properties of pure fluids: equations of state explicit in Helmholtz energy, the modified Benedict-Webb-Rubin equation of state, and an extended corresponding states (ECS) model. Mixture calculations employ a model that applies
mixing rules to the Helmholtz energy of the mixture components; it uses a departure function to account for the departure from ideal mixing. Viscosity and thermal conductivity are modeled with either fluid-specific correlations, an ECS method, or in some cases the friction theory method.

The mixture models used in REFPROP vary by type of mixture (i.e., refrigerant vs. hydrocarbon mixtures). Constituents typically found in natural gas are modeled by the latest GERG model (finished in 2004). The mixing part of the model (the excess contribution) is similar to that used for the refrigerant mixtures. The pure fluid equations of state have been shortened and differ from the default equations in REFPROP. The calculations using these shorter equations are somewhat faster than the default equations, but are slightly less accurate. If the user wishes to use the shorter equations specified by the GERG-2004 model, the ‘Use full GERG-2004 natural gas mixture model’ option should be selected. This will result in slightly different values than the default values. In either situation, the mixing interaction parameters remain the same.” Documentation for this model is found in [16].

B.3 Incompleteness

Although REFPROP is in general fast and accurate, it sometimes fails to converge. It can be hard to find the true reasons for these failures without spending a lot of time searching in REFPROP’s source code. Probably, inconsistencies are occurring in our simulations due to numerical errors. Then, there is no reason to blame REFPROP. However, it turns out that REFPROP might also fail to give an output value if we do expect one. It seems that REFPROP is especially incomplete if the substance of interest is a mixture. An example is given in Figure B.1. Here, REFPROP is able to return a value for the quality, given the density $\rho = 100 \text{ kg} \cdot \text{m}^{-3}$ and the specific internal energy $u$, for $u < 3.6e5 \text{ J} \cdot \text{kg}^{-1}$ and $u > 4.04e5 \text{ J} \cdot \text{kg}^{-1}$, where we the used substance is LNG: a methane, ethane and nitrogen mixture, with corresponding mass fractions of 0.9, 0.09 and 0.01, respectively.

Figure B.1: REFPROP fails to give an output value of the quality $\xi$ for $3.6e5 \text{ J} \cdot \text{kg}^{-1} < u < 4.04e5 \text{ J} \cdot \text{kg}^{-1}$, with a density of $100 \text{ kg} \cdot \text{m}^{-3}$ and a methane, ethane and nitrogen mixture with mass fractions of 0.9, 0.09 and 0.01, respectively.

\[ \text{An error with the message “no convergence” is given without any output value.} \]
Appendix C

Program Code

The Matlab code of the discussed models is given in this appendix.

C.1 Two-Tank System with Constant Temperature without Upper Pipe

C.1.1 Exact Solution for Two-Tank System with Constant Temperature without Upper Pipe

```matlab
%( This script plots an exact solution of the 1D open two tank system

% Meaning of the variables:
% rhol liquid density
% rhog gas density
% HL1 height of lower pipe connection to tank 1
% HL2 height of lower pipe connection to tank 2
% HU1 height of upper pipe connection to tank 1
% HU2 height of upper pipe connection to tank 2
% H1 liquid height in left tank
% H2 liquid height in right tank
% HT1 total height of right tank
% HT2 total height of left tank
% p01 initial gas pressure in left tank
% p02 initial gas pressure in right tank
% pin pressure at left end of pipe
% pout pressure at right end of pipe
% DL diameter of lower pipe
% DU diameter of upper pipe
% D1 diameter of left tank
% D2 diameter of right tank
% LL length of lower pipe
% LU length of upper pipe
% alpha friction factor
% thetal fraction of lower pipe filled with liquid
% C total liquid volume

% 'methane','ethane','nitrogen',
% [ 0.943648747220886 5.03466958769393e-02 6.0045569021752e-03]```

```
116 Program Code

clear all; format long e;
tic
% Geometric and physical parameters

g = 9.81; % gravitation constant
rhol = refpropm('D','T',111,'P',100,'methane');
rhol = 433;
rv = 0; % assume rhog<<rhol
HL1 = 1;
HL2 = 1;
HU1 = 20;
HU2 = 20;
HT1 = 30;
HT2 = 30;
p01 = 1; % initial pressure of gas in tank 1, np01=1 corresponds to atmospheric pressure
p02 = 1; % initial pressure of gas in tank 2, np01=1 corresponds to atmospheric pressure
DL = .5; % assume Dp<<Dt
DU = .5;
D1 = 20;
D2 = 20;
LL = 2000;
LU = 2000;

open = 1; % if open=1, then system is open, closed=0;
closed = abs(open-1); % if open=0, then system is closed, closed=1;
alpha = 2; % friction factor, alpha=0: no friction
AL = pi*(.5*DL)^2; % cross-sectional area lower pipe
AU = pi*(.5*DU)^2; % cross-sectional area upper pipe
A1 = pi*(.5*D1)^2; % cross-sectional area left tank
A2 = pi*(.5*D2)^2; % cross-sectional area right tank

% initial conditions
H10 = 25; % for open case, H1{1}>=H2{1}
H20 = 0;
if H20<HL2
    thetal = 1;
else
    thetal = 1;
end
thetau = 0; % no upper pipe
C = A1+H10+A2+H20+thetal*AL+LL+thetau*AU+LU;

% calculate beta
beta = (AL/A1)*sqrt(2*g/(1+alpha));

if open==1
    % compute solution for open system
    if H20>HL2
        % plot solution
        figure(1);
        t=[0:1000:20000];
        [H1,H2]=H1openH20larger(t,A1,A2,beta,H10,C,AL,LL,HL1,HL2);
        [pin] = p01+rhol*g*max(H1-HL1,0);
        [pout] = p02+rhol*g*max(H2-HL1,0);
        plot( t,H1,t,H2,'red','LineWidth',1.8 );
        grid;
        xlabel('t(s)');
        ylabel('$H(m)$');
        h = legend('$H_1(t)$','$H_2(t)$');
        figure(2);
        plot( t,pin,t,pout,'red','LineWidth',1.8 );
        grid;
        xlabel('t(s)');
        ylabel('$H(m)$');
        h = legend('$p_{in}(t)$','$p_{out}(t)$');
    end
else
    % plot solution
    figure(1);
    t=[0:1000:20000];
    [H1,H2]=H1openH20larger(t,A1,A2,beta,H10,C,AL,LL,HL1,HL2);
    [pin] = p01+rhol*g*max(H1-HL1,0);
    [pout] = p02+rhol*g*max(H2-HL1,0);
    plot( t,H1,t,H2,'red','LineWidth',1.8 );
    grid;
    xlabel('t(s)');
    ylabel('$H(m)$');
    h = legend('$H_1(t)$','$H_2(t)$');
end
C.1 Two-Tank System with Constant Temperature without Upper Pipe

% calculates liquid heights for two-tank system without upper pipe with constant
% temperature for the case that H20<HL2 (exact solution)

function [H1,H2] = H1openH20smaller(t,A1,H10,HL1,HL2,beta,C,A2,AL,LL,H20,DL)

H1=[];
H2=[];

for i=1:length(t)
    if t(i)<2*(sqrt(H10-HL2)-sqrt(-HL2+(C-A2*H20-AL*LL)/A1))/beta
        HH1 = 0.25*beta^2*t(i).^2-beta*t(i)*sqrt(H10-HL2)+H10;
        if HH1<=max(HL1,HL2)
            H1 = [H1 max(HL1,HL2)];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        else
            H1 = [H1 HH1];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        end
    elseif t(i)<tL2
        HH1 = 0.25*beta^2*t(i).^2-beta*t(i)*sqrt(H10-HL2)+H10;
        if HH1<max(HL1,HL2)
            H1 = [H1 max(HL1,HL2)];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        else
            H1 = [H1 HH1];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        end
    elseif t(i)<tL3
        HH1 = (1+A1/A2)*0.25*beta^2*(t(i)-tL2).^2-beta*(t(i)-tL2).*
        sqrt((1+A1/A2)*((C-A2*H20+AL+LL)/A1)-(C-AL+LL)/A2)+((C-A2*H2+AL+LL)/A1);
        if HH1<max(HL1,HL2)
            H1 = [H1 max(HL1,HL2)];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        else
            H1 = [H1 HH1];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        end
    else
        HH1 = (1+A1/A2)*0.25*beta^2*(t(i)-tL2).^2-beta*(t(i)-tL2).*
        sqrt((1+A1/A2)*((C-A2*H20+AL+LL)/A1)-(C-AL+LL)/A2)+((C-A2*H2+AL+LL)/A1);
        if HH1<max(HL1,HL2)
            H1 = [H1 max(HL1,HL2)];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        else
            H1 = [H1 HH1];
            H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
        end
    end
end
end
H1 = [H1 HH1];
H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
end
end

for i=2:length(t)
    if H1(i)>H1(i-1)
        H1(i)=H1(i-1);
    end
    if H2(i)<H2(i-1)
        H2(i)=H2(i-1);
    end
end
end

% calculates liquid heights for two-tank system without upper pipe with
% constant temperature for the case that H20>=HL2 (exact solution)
function [H1,H2] = H1openH20larger(t,A1,A2,beta,H10,C,AL,LL,HL1,HL2)
H1=[];
H2=[];
for i=1:length(t)
    HH1 = (1+A1/A2)*0.25*beta^2*t(i).^2-beta.*t(i).*sqrt((1+A1/A2)*H10-(C-AL*LL)/(A2))... +H10;
    if HH1<=max(HL1,HL2)
        H1 = [H1 max(HL1,HL2)];
        H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
    else
        H1 = [H1 HH1];
        H2 = [H2 (C-A1*H1(i)-AL*LL)/A2];
    end
end
for i=2:length(t)
    if H1(i)>H1(i-1)
        H1(i)=H1(i-1);
    end
    if H2(i)<H2(i-1)
        H2(i)=H2(i-1);
    end
end
end
C.1 Two-Tank System with Constant Temperature without Upper Pipe

C.1.2 Explicit Euler for Two-Tank System with Constant Temperature without Upper Pipe

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script computes numerical solutions of the 1D two-tanks model without upper pipe
% time integration: explicit Euler
% Meaning of the variables:
% rhol liquid density
% rhog gas density
% Hl1 height of lower pipe-tank 1 connection
% Hl2 height of lower pipe-tank 2 connection
% Hu1 height of upper pipe-tank 1 connection
% Hu2 height of upper pipe-tank 2 connection
% H1 liquid height in left tank
% H2 liquid height in right tank
% H1T total height of right tank
% H2T total height of left tank
% p01 initial gas pressure in left tank
% p02 initial gas pressure in right tank
% pin pressure tank 1 at z=0
% pout pressure tank 2 at z=0
% (z=0 at lower pipe-tank 1 connection)
% Dpl diameter of lower pipe
% Dpu diameter of upper pipe
% Dt1 diameter of left tank
% Dt2 diameter of right tank
% Vtot total liquid volume
% alpha friction factor
% theta parameter in the theta-method
% Nt number of time steps
% Tmax final time
% dltt time step

%'methane', 'ethane', 'nitrogen', [.943648747220886 5.0 3466958769393e-02 6.0045569021752e-03]
clear all; format long e;
tic
% Geometric and physical parameters
g = 9.81;  % gravitation constant
rhol = 433;
rhog = 0;  % assume rhog<<rhol
Hl1 = 1;
Hl2 = 1;
H1T = 30;
H2T = 30;
p01 = 1e5;  % initial pressure of gas in tank 1 (Pa)
p02 = 1e5;  % initial pressure of gas in tank 2 (Pa)
Dpl = 0.5;  % diameter of lower pipe
Dpu = << Dt
Dt1 = 20;
Dt2 = 20;
open = 0;  % if open=1, then system is open, closed=0;
closed = abs(open-1);  % if open=0, then system is closed, closed=1;
alpha = 2;  % friction factor, alpha=0: no friction
Apl = pi*(Dpl/2)^2;  % cross-sectional area lower pipe
At1 = pi*(Dt1/2)^2;  % cross-sectional area left tank
At2 = pi*(Dt2/2)^2;  % cross-sectional area right tank
LL = 2000;  % length of lower pipe
fractionL0 = 1;  % initial fraction of lower pipe filled with liquid
H10 = 25;  % initial liquid height tank 1
```

```
H20 = 20; % initial liquid height tank 2

% grid
Tmax = 1000;
dltt = 1;
Nt = round(Tmax/dltt);

% variables
H1 = zeros(Nt+1,1);
H2 = zeros(Nt+1,1);
pin = zeros(Nt+1,1);
pout = zeros(Nt+1,1);
fractionL = zeros(Nt+1,1);

% initial conditions
H1(1) = H10;
H2(1) = H20;

pin(1) = p01+rh1*g*max(H1(1)-Hl1,0);
pout(1) = p02+rh1*g*max(H2(1)-Hl1,0);
fractionL(1) = fractionL0;
Vtot = At1*H1(1)+At2*H2(1)+fractionL(1)*Ap1*LL;

if open==1

% compute solution for open system
for n=1:Nt
    if (H1(n)>H2(n)) && (H1(n)>max(Hl1,Hl2))
        H1(n+1) = H1(n)-dltt*(Ap1/At1)*sqrt((2/(1+alpha))*g*(H1(n)-H2(n)));
    else
        H1(n+1) = H1(n)-dltt*(Ap1/At1)*sqrt((2/(1+alpha))*g*(H1(n)-Hl2));
    end
else
    if (H1(n)<H2(n)) && (H2(n)>max(Hl2,Hl1))
        H1(n+1) = H1(n)+dltt*(Ap1/At1)*sqrt((2/(1+alpha))*g*(H2(n)-H1(n)));
    else
        H1(n+1) = H1(n)+dltt*(Ap1/At1)*sqrt((2/(1+alpha))*g*(H2(n)-Hl1));
    end
end

% update H2 and fractionL
if fractionL(n)=1
    fractionL(n+1) = min(1,(Vtot-At1*H1(n+1)-At2*H2(n+1))/(Ap1*LL));
else
    fractionL(n+1)=fractionL(n);
end
H2(n+1) = (Vtot-At1*H1(n+1)-fractionL(n+1)*Ap1*LL)/At2;
pin(n+1) = p01+rh1*g*max(Hl1-n+1,0);
pout(n+1) = p02+rh1*g*max(H2(n+1)-Hl1,0);

end
else
% compute solution for closed system
for n=1:Nt
    if (pin(n)>pout(n)) && (H1(n)>max(Hl1,Hl2))
        H1(n+1) = H1(n)-dltt*(Ap1/At1)*sqrt(abs((2/(1+alpha))*g*(H1(n)-H2(n)))+...
            (p01/(rh1))*(H1T-H1(1))/(H1T-H1(n))-(p02/(rh1))*((H2T-H2(1))/(H2T...
            -H2(n))));
    else
        H1(n+1) = H1(n)-dltt*(Ap1/At1)*sqrt(abs((2/(1+alpha))*g*(H1(n)-Hl2)+...
            (p01/(rh1)))*(H1T-H1(1))/(H1T-H1(n))-(p02/(rh1))*(H2T-H2(1))/H2T...
            -H2(n))));
    end
end

% update H2 and fractionL
if fractionL(n)<1
    fractionL(n+1) = min(1,(fractionL0*Ap1*LL+At1*(H1(n)-H1(n+1)))/(Ap1*LL));
end

end

%%H2(n+1) not specified yet!!!
else
    fractionL(n+1)=fractionL(n);
end
H2(n+1) = (Vtot-At1*H1(n+1)-fractionL(n+1)*Apl*LL)/At2;
if H1(n+1)<Hl1
    H1(n+1) = Hl1;
    H2(n+1) = (Vtot-At1*Hl1(n+1)-fractionL(n+1)*Apl*LL)/At2;
end
elseif pin(n)<pout(n) && (H2(n)>max(Hl2,Hl1))
if H1(n)>Hl1
    H2(n+1)=H2(n)-dltt*(Apl/At2)*sqrt(\( \frac{2}{1+\alpha} \) \((g*(H2(n)-H1(n))+...\)
    (p02/(rhol))*((H2T-H2(1))/(H2T-H2(n)))-(p01/(rhol))*((H1T-H1(1))/(H1T...-
    H1(n))))));
else
    H2(n+1)=H2(n)-dltt*(Apl/At2)*sqrt(\( \frac{2}{1+\alpha} \) \((g*(H2(n)-Hl1)+...\)
    (p02/(rhol))*((H2T-H2(1))/(H2T-H2(n)))-(p01/(rhol))*((H1T-H1(1))/(H1T...-
    H1(n))))));
end
%update H1 and fractionL
if fractionL(n)<1
    fractionL(n+1)=min(1,(fractionL0*Apl*LL+At1*(H2(1)-H2(n+1)))/(Apl*LL));
else
    fractionL(n+1)=fractionL(n);
end
H1(n+1) = (Vtot-At2*H2(n+1)-fractionL(n+1)*Apl*LL)/At1;
if H2(n+1)<Hl2
    H2(n+1) = Hl2;
    H1(n+1) = (Vtot-At2*Hl2(n+1)-fractionL(n+1)*Apl*LL)/At1;
end
else
    H1(n+1)=H1(n);
    H2(n+1)=H2(n);
end
pin(n+1) = p01*((H1T-H1(1))/(H1T-H1(n)))+rhol*g*max(H1(n+1)-Hl1,0);
pout(n+1) = p02*((H2T-H2(1))/(H2T-H2(n)))+rhol*g*max(H2(n+1)-Hl1,0);
end
end
toc

% plot solution
figure(1);
plot( [0:dltt:Tmax],H1,[0:dltt:Tmax],H2,'--red','LineWidth',1.8 );
grid;
xlabel('StS(s)');
ylabel('$SH_{(t)}$');
h = legend('SH_1(t)$','$SH_2(t)$',0);

figure(2);
plot( [0:dltt:Tmax],pin,[0:dltt:Tmax],pout,'--red','LineWidth',1.8 );
grid;
xlabel('StS(s)');
ylabel('$Sp'(Pa)');
h = legend('$Sp_{(in)}(t)$','$Sp_{(out)}(t)$',0);

laprint(1,'252011hc','width',7.375,'figcopy','off');
laprint(2,'252011pc','width',7.375,'figcopy','off');
# Program Code

## C.2 Two-Tank System with Constant Temperature with Upper Pipe

### C.2.1 Explicit Euler for Two-Tank System with Constant Temperature with Upper Pipe

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script computes numerical solutions of the 1D two tank model
% with lower and upper pipe
% with constant temperature, Explicit Euler
% time integration: explicit Euler

% Meaning of the variables:
% HL    height of lower pipe-tank connection
% HU    height of upper pipe-tank connection
% H     liquid height in tanks
% HT    total height of tanks
% p0    initial gas pressure in tanks
% pin   pressure tank 1 at z=0
% pout  pressure tank 2 at z=0
%(z=0 at lower pipe-tank 1 connection)
% Dpl   diameter of lower pipe
% Dpu   diameter of upper pipe
% Dt1   diameter of left tank
% Dt2   diameter of right tank
% Vtot  total liquid volume
% alpha friction factor
% theta parameter in the theta-method
% Nt    number of time steps
% Tmax  final time
% delt  time step
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% methane', 'ethane', 'nitrogen', [.943648747220886 5.03466958769393e-02 6.00455569021752e-03]

clear all;

function [values50] = twotanksupperpipeeuler(delta)
format long e;
tic

% Geometric and physical parameters
g = 9.81;            % gravitation constant
HL = [1 1];          % lower pipe connection
HT = [20 10];        % tank heights
DP = [0.2 0.2];      % assume Dp<<Dt
DT = [5 2];          % tank diameters

alpha = 2;           % friction factor for liquid flow in pipe,...
alpha1 = 1;          % friction factor for gas flow in pipe
AP = pi.*((DP./2).^2);  % cross-sectional area pipes
AT = pi.*((DT./2).^2);  % cross-sectional area tanks
LP = [10 10];        % length of pipe
gastemp = 111;
specificgasconstant = 8.314472/(16.042e-3);  % specific gas constant methane=gas ...
                    % constant/molar mass (kg/mol)
constraints = 2;     % use constraints nr 1 for gas flow:1,
```
C.2 Two-Tank System with Constant Temperature with Upper Pipe

% use constraints nr 1 and 2:2, 0 ...
otherwise

CT = [0];

% rhoL = refpropm('D','T',liquidtemp,'P',100,'methane');
rhol = 433;

% parameters for euler method
theta = 0;
theta1 = 1-theta;

% grid (time)
Tmax = 300;
dltt = delta;
Nt = round(Tmax/dltt);

% variables
H = zeros(Nt+1,2);
pin = zeros(Nt+1,1);
pout = zeros(Nt+1,1);
pgas = zeros(Nt+1,2);
rhog = zeros(Nt+1,2);

% initial conditions
H(1,1) = 12;
H(1,2) = 2;

fractionL = 1;
fractionU = 0;

pgas(1,1) = 1e5;
pgas(1,2) = 1e5;

pin(1) = pgas(1,1)+rhol*g*max(H(1,1)-HL(1),0);
pout(1) = pgas(1,2)+rhol*g*max(H(1,2)-HL(1),0);
rhog(1,1) = pgas(1,1)/(specificgasconstant*gastemp);
rhog(1,2) = pgas(1,2)/(specificgasconstant*gastemp);

Vliq = AT(1)*H(1,1)+AT(2)*H(1,2)+fractionL*AP(1)*LP(1)+fractionU*AP(2)*LP(2);
Mliq = rhoL*Vliq;

Vgas1 = AT(1)*(HT(1)-H(1,1));
Vgas2 = AT(2)*(HT(2)-H(1,2));

Vgas = Vgas1+Vgas2+((1-fractionL)*AP(1)*LP(1)+((1-fractionU)*AP(2)*LP(2);
Mgas = rhoL*Mgas1+Mgas2;

Vgasaverage = Mgas/Vgas;

Hequil = (AT(1)*H(1,1)+AT(2)*H(1,2))/(AT(1)+AT(2));

% compute solution
for n=1:Nt

if (pin(n)-pout(n))/pin(n)>1e-4&&((H(n,1)-max(HL(1),HL(2)))<1e-13)
H(n+1,1) = max(H(n,1)-dltt*(AP(1)/AT(1))*sqrt((2/(1+alpha))*(g*(H(n,1)-max(H(n,2)...,
,HL(2))))+...;
(pgas(n,1)-pgas(n,2))/(rhoL)),HL(1));

if (n+1,2) = (Vliq*AT(1)*H(n+1,1)-fractionL*AP(1)*LP(1)-fractionU*AP(2)*LP(2))/AT...
(2);
else (pin(n)-pout(n))/pin(n)<1e-4&&((H(n,2)-max(HL(1),HL(2)))<1e-13)
H(n+1,2) = max(H(n,2)-dltt*(AP(1)/AT(2))*sqrt((2/(1+alpha)))*g*(H(n,2)-max(H(n,1)...,
,HL(2))))+...;
(pgas(n,2)-pgas(n,1))/(rhoL)),HL(2));

if (n+1,1) = (Vliq*AT(2)*H(n+1,2)-fractionL*AP(1)*LP(1)-fractionU*AP(2)*LP(2))/AT...
(2);
else (H(n,1)-max(HL(1),HL(2)))>0&&((H(n,2)-max(HL(1),HL(2)))>0)
H(n+1,1) = Hequil;

end

end
H(n+1,2) = H_{equil};
else
H(n+1,1) = H(n,1);
H(n+1,2) = H(n,2);
end

V_{gas1} = A_T(1) \times (HT(1)-H(n+1,1));
V_{gas2} = A_T(2) \times (HT(2)-H(n+1,2));
M_{gasequilibrium1} = \rho_{gasaverage} \times V_{gas1};
M_{gasequilibrium2} = \rho_{gasaverage} \times V_{gas2};

if \left( \frac{p_{gas}(n,1)-p_{gas}(n,2)}{p_{gas}(n,1)} \right) > 1e-13
\frac{gasvelocity = \sqrt{2 \times (p_{gas}(n,1)-p_{gas}(n,2)) / ((1+\alpha_{1}) \times \rho_{g}(n,2))}}{massflux = \rho_{g}(n,2) \times A_P(2) \times gasvelocity;}
\frac{CTtemp = (M_{gas1} - \rho_{gasaverage} \times A_P(1) \times (HT(1)-H(n,1))) / (massflux \times A_P(1) \times \rho_{g}(n,1) \times \sqrt{2 \times (p_{gas}(n,1)-p_{gas}(n,2)) / (p_{gas}(n,2))}))}{}
\frac{CT = [CT CTtemp];}
if constraints==1||constraints==2
M_{gas1} = \max(M_{gas1}-dltt \times massflux, M_{gasequilibrium1});
M_{gas2} = \min(M_{gas2}+dltt \times massflux, M_{gasequilibrium2});
else
M_{gas1} = M_{gas1}-dltt \times massflux;
M_{gas2} = M_{gas2}+dltt \times massflux;
end
else if \left( \frac{p_{gas}(n,1)-p_{gas}(n,2)}{p_{gas}(n,1)} \right) < -1e-13
\frac{gasvelocity = \sqrt{2 \times (p_{gas}(n,2)-p_{gas}(n,1)) / ((1+\alpha_{1}) \times \rho_{g}(n,1))}}{massflux = \rho_{g}(n,1) \times A_P(2) \times gasvelocity;}
\frac{CTtemp = (M_{gas2} - \rho_{gasaverage} \times A_P(2) \times (HT(2)-H(n,2))) / (massflux \times A_P(1) \times \rho_{g}(n,1) \times \sqrt{2 \times (p_{gas}(n,2)-p_{gas}(n,1)) / (p_{gas}(n,1))}))}{}
\frac{CT = [CT CTtemp];}
if constraints==1||constraints==2
M_{gas1} = \min(M_{gas1}+dltt \times massflux, M_{gasequilibrium1});
M_{gas2} = \max(M_{gas2}-dltt \times massflux, M_{gasequilibrium2});
else
M_{gas1} = M_{gas1}+dltt \times massflux;
M_{gas2} = M_{gas2}-dltt \times massflux;
end

else if constraints==2
M_{gas1} = M_{gasequilibrium1};
M_{gas2} = M_{gasequilibrium2};
end
CTtemp = 0;
CT = [CT CTtemp];
end
\frac{\rho_{g}(n+1,1) = M_{gas1} / (A_T(1) \times (HT(1)-H(n+1,1)))}{\rho_{g}(n+1,2) = M_{gas2} / (A_T(2) \times (HT(2)-H(n+1,2)))};
p_{gas}(n+1,1) = \rho_{g}(n+1,1) \times \text{specificgasconstant} \times gastemp;
p_{gas}(n+1,2) = \rho_{g}(n+1,2) \times \text{specificgasconstant} \times gastemp;
p_{in}(n+1) = p_{gas}(n+1,1) + r_{hol} \times g \times \max(H(n+1,1)-HL(1),0);
p_{out}(n+1) = p_{gas}(n+1,2) + r_{hol} \times g \times \max(H(n+1,2)-HL(1),0);
end
toc

%% plot solution
figure(1);
plot([0:dltt:Tmax],H(:,1),[0:dltt:Tmax],H(:,2),’-r’,’LineWidth’,1.8);
grid;
xlabel(’$t$(s)’);
ylabel(’$H$(m)’);
h = legend(’$H_1(t)$’,’$H_2(t)$’,4);
C.2 Two-Tank System with Constant Temperature with Upper Pipe

```matlab
figure(2);
plot([0:dltt:Tmax],pin,[0:dltt:Tmax],pout,'--red','LineWidth',1.8);
grid;
xlabel('
\textit{t}\text{\( (s)\}'));
ylabel('
\textit{p}\text{\( (Pa)\)}');
h = legend('
\textit{p}_{in}(t)$','$p_{out}(t)$','location','southeast');

figure(3);
plot([0:dltt:Tmax],pgas(:,1),[0:dltt:Tmax],pgas(:,2),'-red','LineWidth',1.8);
grid;
xlabel('
\textit{t}\text{\( (s)\}'));
ylabel('
\textit{p}\text{\( (Pa)\)}');
h = legend('
\textit{p}_{gas1}(t)$','$p_{gas2}(t)$','location','SouthEast');

figure(4);
plot([0:dltt:Tmax],CT(:),'--red','LineWidth',1.8);
grid;
xlabel('
\textit{t}\text{\( (s)\}'));
ylabel('
\Delta t_s\text{\( (s)\)}');
h = legend('
\Delta t_s(t)$','location','northEast');

values50 = zeros(6,1);
values50(1) = H(150/delta+1,1);
values50(2) = H(150/delta+1,2);
values50(3) = pin(150/delta+1);
values50(4) = pout(150/delta+1);
values50(5) = pgas(150/delta+1,1);
values50(6) = pgas(150/delta+1,2);

lprintf(1,'2tanksconstraints12H01c','figcopy','off','width',7.375);
lprintf(2,'2tanksconstraints12P01c','figcopy','off','width',7.375);
lprintf(3,'2tanksconstraints12Pgas01c','figcopy','off','width',7.375);
lprintf(4,'2tankswithchange65Ts1c','figcopy','off','width',7.375);
```
C.2.2 Runge-Kutta and NDF for Two-Tank System with Upper Pipe with Constant Temperature

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script computes numerical solutions of the 1D closed two-tank model with lower and upper pipe, with constant temperature. Runge-Kutta methods

% Meaning of the variables:
% HL height of lower pipe-tank connection
% HU height of upper pipe-tank connection
% H liquid height in tanks
% HT total height of tanks
% p0 initial gas pressure in tanks
% pin pressure tank 1 at z=0
% pout pressure tank 2 at z=0
% (z=0 at lower pipe-tank 1 connection)
% Dpl diameter of lower pipe
% Dpu diameter of upper pipe
% Dt1 diameter of left tank
% Dt2 diameter of right tank
% Vtot total liquid volume
% alpha friction factor

% theta parameter in the theta-method
% Nt number of time steps
% Tmax final time
% dlTT time step

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [y50] = rkberoulli1(rtol)

tic

HL = [1 1];
HT = [20 10];
DT = [5 2];
AT = pi.*((DT./2).ˆ2);
rhol = 433;
g = 9.81;
specificgasconstant = 8.314472/(16.042e-3); %specific gas constant methane=gas constant/molar mass (kg/mol)

% Initial conditions
H1 = 12;
H2 = 2;
p01 = 1e5;
p02 = 1e5;
reltol = rtol; %<1e-4 for H0=[12 2] to prevent oscillations in pressure if ode45
events = 0; %if events=1: use events

pin0 = p01+rhol*g*max(H1-HL(1),0);
pout0 = p02+rhol*g*max(H2-HL(1),0);
gastemp = 111;

% Initialization
y0 = zeros(10,1);
y50 = zeros(10,1);

y0(1) = H1;
y0(2) = H2;
```

---

The code above is designed to simulate the behavior of a closed two-tank system with lower and upper pipes connected to two tanks filled with liquid. The system's behavior is influenced by various parameters such as the heights of the tanks, pipe diameters, initial gas pressure, and friction factors. The script computes numerical solutions using Runge-Kutta methods, ensuring accurate simulation of the system's dynamics, particularly in scenarios where the pressure inside the tanks might oscillate due to the system's configuration. The code includes specific variables and parameters that are crucial for modeling the system's behavior under constant temperature conditions.
C.2 Two-Tank System with Constant Temperature with Upper Pipe

\[ y_0(3) = \text{pin}_0; \]
\[ y_0(4) = \text{pout}_0; \]
\[ y_0(5) = \rho_\text{ol}; \]
\[ y_0(6) = \rho_1; \]
\[ y_0(7) = \rho_2; \]
\[ y_0(8) = \left(\frac{\rho_1}{\text{specificgasconstant}\times\text{gastemp}}\right)\times(\text{HT}(1)-\text{HT}_1)\times\Delta T(1); \]
\[ y_0(9) = \left(\frac{\rho_2}{\text{specificgasconstant}\times\text{gastemp}}\right)\times(\text{HT}(2)-\text{HT}_2)\times\Delta T(2); \]
\[ y_0(10) = \text{reltol}; \]

\[ \text{tstart} = 0; \]
\[ \text{tfinal} = 300; \]
\[ \text{tout} = \text{tstart}; \]
\[ \text{yout} = y_0; \]
\[ \text{teout} = []; \]
\[ \text{yeout} = []; \]
\[ \text{ieout} = []; \]

\[ \text{if} \ events==1 \]
\[ \text{options} = \text{odeset}(\text{'RelTol'},\text{reltol}); \]
\[ [t,y] = \text{ode23t}(@rkbern,[\text{tstart} 150],y_0,\text{options}); \]
\[ \]$\text{Accumulate output. This could be passed out as output arguments.}$
\[ \text{nt} = \text{length}(t); \]
\[ \text{tout} = [\text{tout}; t(2:nt)]; \]
\[ \text{yout} = [\text{yout}; y(2:nt,:)]; \]
\[ \] % Events at tstart are never reported.
\[ \text{y0}(:,1) = y(nt,:); \]
\[ y_{50} = y0(:,1); \]
\[ \text{options} = \text{odeset}(\text{'Events'},@events3,'\text{'RelTol'},\text{reltol}); \]
\[ [t,y,\text{te},y\text{e},\text{ie}] = \text{ode23t}(@rkbern,\text{[150 tfinal]},y_0,\text{options}); \]
\[ \]$\text{Accumulate output. This could be passed out as output arguments.}$
\[ \text{nt} = \text{length}(t); \]
\[ \text{tout} = [\text{tout}; t(2:nt)]; \]
\[ \text{yout} = [\text{yout}; y(2:nt,:)]; \]
\[ \] % Events at tstart are never reported.
\[ \text{teout} = [\text{teout}; te]; \]
\[ \text{yeout} = [\text{yeout}; ye]; \]
\[ \text{ieout} = [\text{ieout}; ie]; \]

\[ \text{else} \]
\[ \text{options} = \text{odeset}(\text{'RelTol'},\text{reltol}); \]
\[ [t,y] = \text{ode23t}(@rkbern,[\text{tstart tfinal}],y_0,\text{options}); \]
\[ \]$\text{Accumulate output. This could be passed out as output arguments.}$
\[ \text{nt} = \text{length}(t); \]
\[ \text{tout} = [\text{tout}; t(2:nt)]; \]
\[ \text{yout} = [\text{yout}; y(2:nt,:)]; \]
\[ \] % Events at tstart are never reported.
\[ \] % if \ t(nt)<tfinal
\[ \text{fprintf('filled');} \]
\[ \text{tstart} = t(nt); \]
\[ \text{options3} = \text{odeset}(\text{'RelTol'},\text{reltol}); \]
\[ \]$\text{Solve until the first terminal event.}$
% y0(:) = y(nt,:);
% [t,y] = ode15s(@rkbernconstant,[tstart tfinal],y0,options3);

% nt = length(t);
% tout = [tout; t(2:nt)];
% yout = [yout; y(2:nt,:)];

end

% plot solution
figure(1);
plot( tout,yout(:,1),'-',tout,yout(:,2),'--r', ...
     'LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$H$(m)');
h = legend( '$H_1(t)$','$H_2(t)$','location','southeast');

figure(2);
plot( tout,yout(:,3),'-',tout,yout(:,4),'--r',tout,yout(:,6),'-g',tout,yout(:,7),'-',...
     'LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$Sp(Pa)$');
h = legend( '$p_{in}(t)$','$p_{out}(t)$','$p_{gas1}(t)$','$p_{gas2}(t)$','location','east'... 
                     );

figure(3);
plot( tout,yout(:,6),'-',tout,yout(:,7),'--r',...
     'LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$m_{gas}(kg)$');
h = legend( '$m_{gas1}(t)$','$m_{gas2}(t)$',0);

figure(4);
plot( tout,yout(:,8),'-',tout,yout(:,9),'--r',...
     'LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$p$(Pa)');
h = legend( '$p_{in}(t)$','$p_{out}(t)$','location','southeast');
toc

laprint(1,'2tanksode23tH2c','width',7.375,'figcopy','off');
laprint(2,'2tanksode23tP2','width',7.375,'figcopy','off');
laprint(3,'2tanksode23tPgass2c','width',7.375,'figcopy','off');
laprint(5,'2tanksode23tm2c','width',7.375,'figcopy','off');

%-------------------------------------------------- ------------------------
function dy = rkbern(t,y)
dy = zeros(10,1);  % a column vector

%--------------------------------------------------

% Geometric and physical parameters

$ g = 9.81; \quad \text{# gravitation constant}$
$ HL = [1 1]; \quad \text{# lower pipe connection}$
$ HT = [20 10]; \quad \text{# upper pipe connection}$
$ DP = [0.2 0.2]; \quad \text{# assume Dp<Dt}$
$ DT = [5 2]; \quad \text{# tank diameters}$
C.2 Two-Tank System with Constant Temperature with Upper Pipe

\[
\begin{align*}
\text{alpha} &= 2; \quad \text{\texttt{alpha=0: no friction}} \\
\text{alpha1} &= 1; \quad \text{\texttt{friction factor for gas flow in pipe}} \\
\text{AP} &= \pi \cdot (\text{DP}/2)^2; \quad \text{\texttt{cross-sectional area pipes}} \\
\text{AT} &= \pi \cdot (\text{DT}/2)^2; \quad \text{\texttt{cross-sectional area tanks}}
\end{align*}
\]

gastemp = 111;

especificgasconstant = 8.314472/(16.042e-3); %specific gas constant methane=gas constant/molar mass (kg/mol)

\[
\begin{align*}
\text{dy}(10) &= 0; \\
\text{dy}(5) &= 0;
\end{align*}
\]

\[
\begin{align*}
\text{if} \quad (y(3)>y(4)) && (y(1)>\text{max}(HL(1),HL(2))) && (y(2)+HT(2))/HT(2)<-y(10) \\
\quad \text{dy}(1) &= -\left(\frac{\text{AP}(1)}{\text{AT}(1)}\right) \cdot \text{sqrt} \left(\frac{\text{abs}}{2/(1+\text{alpha})} \cdot \left(\text{y}(1)-\text{y}(2)\right) \cdot \left(\text{y}(6)-\text{y}(7)\right) / \text{y}(5)\right); \\
\text{else} \quad \text{dy}(1) &= -\left(\frac{\text{AP}(1)}{\text{AT}(1)}\right) \cdot \text{sqrt} \left(\frac{\text{abs}}{2/(1+\text{alpha})} \cdot \left(\text{y}(2)-\text{HL}(2)\right) \cdot \left(\text{y}(6)-\text{y}(7)\right) / \text{y}(5)\right);
\end{align*}
\]

\[
\begin{align*}
\text{%update H2 and fractionL} \\
\text{if fractionL(n)<1} \\
\text{fractionL(n+1) = min(1,(Vtot-At1*H1(n+1)-At2*H2(n+1))/(Ap1*LL));} \\
\text{else} \\
\text{fractionL(n+1)=fractionL(n);} \\
\text{end}
\end{align*}
\]

\[
\begin{align*}
\text{dy}(2) &= -(\text{AT}(1)/\text{AT}(2)) \cdot \text{dy}(1); \quad \text{elseif} \quad (y(3)<y(4)) && (y(2)+\text{max}(HL(1),HL(2))) && (y(2)+\text{-HT}(1))/\text{HT}(1)<0 \\
\quad \text{dy}(1) &= -(\text{AP}(1)/\text{AT}(1)) \cdot \text{sqrt} \left(\frac{\text{abs}}{2/(1+\text{alpha})} \cdot \left(\text{y}(2)-\text{y}(1)\right) \cdot \left(\text{y}(7)-\text{y}(6)\right) / \text{y}(5)\right); \\
\text{else} \quad \text{dy}(1) &= -(\text{AP}(1)/\text{AT}(1)) \cdot \text{sqrt} \left(\frac{\text{abs}}{2/(1+\text{alpha})} \cdot \left(\text{y}(2)-\text{HL}(1)\right) \cdot \left(\text{y}(7)-\text{y}(6)\right) / \text{y}(5)\right);
\end{align*}
\]

\[
\begin{align*}
\text{dy}(2) &= -(\text{AT}(1)/\text{AT}(2)) \cdot \text{dy}(1); \\
\text{end}
\end{align*}
\]

\[
\begin{align*}
\text{VGas1} &= \text{AT}(1) \cdot (\text{HT}(1)-\text{y}(1)); \\
\text{VGas2} &= \text{AT}(2) \cdot (\text{HT}(2)-y(2)); \\
\text{Mgasequilibrium1} &= \text{rhogasaverage} \cdot \text{VGas1}; \\
\text{Mgasequilibrium2} &= \text{rhogasaverage} \cdot \text{VGas2}; \quad \text{if} \quad (y(1)-\text{HT}(1))/\text{HT}(1)<-y(10) && (y(2)-\text{HT}(2))/\text{HT}(2)<-y(10) \quad \text{gasvelocity} = \text{sqrt} \left(\frac{2 \times \text{y}(6) - \text{y}(7)}{(1 + \text{alpha}) \times \text{y}(7) / (\text{specificgasconstant} \times \text{gastemp})} \right); \\
\text{massflux} &= \text{y}(7) / (\text{specificgasconstant} \times \text{gastemp}) \cdot \text{AP}(2) \cdot \text{gasvelocity}; \\
\text{dy}(8) &= -\text{massflux}; \\
\text{dy}(9) &= \text{massflux}.
\end{align*}
\]
```matlab
else if y(6)-y(7)<-y(10)
gasvelocity = sqrt(2*(y(7)-y(6))/(1+alpha1)*(y(6)/specificgasconstant*gastemp)...)
masstransfer = (y(6)/(specificgasconstant*gastemp))*AP(2)*gasvelocity;
dy(8) = masstransfer;
dy(9) = -masstransfer;
end
else
dy(8) = 0;
dy(9) = 0;
end
else
dy(8) = 0;
dy(9) = 0;
end

% rho1 = Mgas1/(AT(1)*(HT(1)-y(1)));
% rho2 = Mgas2/(AT(2)*(HT(2)-y(2)));
if (y(1)-HT(1))/HT(1)<-y(10)
dy(6) = (((AT(1)*(HT(1)-y(1))))*dy(8)+y(8)*(AT(1)*dy(1)))/(((AT(1)*(HT(1)-y(1))))^2))...*gastemp*specificgasconstant;
else
dy(6) = 0;
end
if (y(2)-HT(2))/HT(2)<-y(10)
dy(7) = (((AT(2)*(HT(2)-y(2))))*dy(9)+y(9)*(AT(2)*dy(2)))/(((AT(2)*(HT(2)-y(2))))^2)))...*gastemp*specificgasconstant;
else
dy(7) = 0;
end
if y(1)>HL(1)
dy(3) = dy(6)+y(5)*g*dy(1);
else
dy(3) = dy(6);
end
if y(2)>HL(1)
dy(4) = dy(7)+y(5)*g*dy(2);
else
dy(4) = dy(7);
end

function dy = rkberncanstant(t,y)
dy = zeros(10,1); % a column vector
% Geometric and physical parameters
g = 9.81; % gravitation constant
HL = [1 1]; % lower pipe connection
HU = [20 10]; % upper pipe connection
HT = [20 10]; % tank heights
DP = [0.2 0.2]; % assume Dp<<Dt
DT = [5 2]; % tank diameters
alpha = 2; % friction factor for liquid flow in pipe,...
alpha = 0; % no friction
AP = pi.*((DP./2).^2); % cross-sectional area pipes
AT = pi.*((DT./2).^2); % cross-sectional area tanks
gastemp = 111;
specificgasconstant = 8.314472/(16.042e-3); % specific gas constant methane = gas ...
constant/molar mass (kg/mol)
dy(1) = 0;
```
dy(2) = 0;
dy(3) = 0;
dy(4) = 0;
dy(5) = 0;
dy(6) = 0;
dy(7) = 0;
dy(8) = 0;
dy(9) = 0;
dy(10) = 0;

%--------------------------------------------------
function [value,isterminal,direction] = events3(t,y)
HT = [20 10]; % tank heights
% Locate the time when quality passes through zero in a decreasing direction
value(1) = y(1) - HT(1); % detect H1=HT1
isterminal(1) = 1; % stop the integration
direction(1) = 1; % positive direction
value(2) = y(2) - HT(2); % detect H2=HT2
isterminal(2) = 1; % stop the integration
direction(2) = 1; % positive direction
C.3 One-Tank System with Variable Temperature

C.3.1 Explicit Euler for One-Tank System

```matlab
function [values] = onetankeuler1(delta)

% One-tank model with heat flux from environment,
% solves set of differential algebraic equations (DAEs) by explicit Euler

% Enthalpy equations are based on equilibrium, no z-discretisation, i.e.
% specific gas and liquid enthalpy are constant in space per tank
% internal energy instead of enthalpy

% LNG = 'methane','ethane','nitrogen', [.94364874720886 5.03466958769393e-02 ... 6.0045569021752e-03]

tic

% Geometric and physical parameters
g = 9.81;  % gravitation constant
HT = 30;  % tank height
DT = 20;  % tank diameter

AT = pi*(DT/2)^2;  % cross-sectional area tank
ST = 2*pi*(DT/2)*HT+2*pi*(DT/2)^2;  % surface area tank

temperatureamb1 = 293;
K = 55;  % heat coefficient

Nt = round(Tmax/dltt);

% Initial conditions
y = zeros(10,Nt);

y(9,1) = quality;
y(6,1) = temperature;

% mass
mass = AT(1)*HT(1)*refpropm('D','T',y(6,1),'Q',y(9,1),'methane');

% enthalpy
y(1,1) = mass*refpropm('H','T',y(6,1),'Q',y(9,1),'methane');

% grid
Tmax = 100*7*24*3600;
dltt = delta*24*3600;

% specific gas constant methane=gas constant/molar mass (kg/mol)
specificgasconstant = 8.314472/(16.042e-3);

% fprintf T = %g K
fprintf('T = %g K',temperature);

% fprintf q = %g
fprintf('q = %g',quality);

% superheated = 0;
% supercritical = 0;
% subcooled = 0;

% tcrit = refpropm('T','D',10^10,'Q',1,'methane');
% specific gas constant methane=gas constant/molar mass (kg/mol)
specificgasconstant = 8.314472/(16.042e-3);

% grid
Tmax = 100*7*24*3600;
dltt = delta*24*3600;
Nt = round(Tmax/dltt);

% Initial conditions
y = zeros(10,Nt);

y(9,1) = quality;
y(6,1) = temperature;
```

% internal energy
y(10,1) = mass * refpropm('U', 'T', y(6,1), 'Q', y(9,1), 'methane');

% liquid density
y(4,1) = refpropm('D', 'T', y(6,1), 'Q', 0, 'methane');
% y0(4) = refpropm('D', 'T', y0(6), 'Q', 0, 'methane', 'ethane', ...
% 'nitrogen', [ .943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

% specific liquid enthalpy
y(2,1) = refpropm('H', 'T', y(6,1), 'D', y(4,1), 'methane');
% y(2,1) = refpropm('H', 'T', y(6,1), 'Q', 0, 'methane');
% y0(2) = refpropm('H', 'T', y0(6), 'D', y0(4), 'methane', 'ethane', ...
% 'nitrogen', [ .943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

% liquid height
y(8,1) = (((1-y(9,1)) * mass)/y(4,1))/AT;

% gas density
y(5,1) = (y(9,1) * mass)/(AT * (HT-y(8,1)));

% specific gas enthalpy
y(3,1) = refpropm('H', 'T', y(6,1), 'D', y(5,1), 'methane');
% y(3,1) = refpropm('H', 'T', y(6,1), 'Q', 1, 'methane');
% y0(3) = refpropm('H', 'T', y0(6), 'D', y0(5), 'methane', 'ethane', ...
% 'nitrogen', [ .943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

% gas pressure
y(7,1) = 1000 * refpropm('P', 'T', y(6,1), 'D', y(5,1), 'methane');
% y0(7) = 1000 * refpropm('P', 'T', y0(6), 'D', y0(5), 'methane', 'ethane', ...
% 'nitrogen', [ .943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

for t=1:Nt
% internal energy update
y(10,t+1) = y(10,t) - K * dltt * (y(6,t)-temperatureamb1);
% enthalpy update
y(1,t+1) = y(1,t) - K * dltt * (y(6,t)-temperatureamb1);
% quality update
y(9,t+1) = refpropm('Q', 'D', mass/(AT*HT), 'U', y(10,t+1)/mass, 'methane');
if (y(9,t+1)==998)
    superheated<1
    fprintf('superheated vapor, but quality not defined (T > Tc)\n');
    fprintf('\%g
', (t-1) * dltt/(24 * 3600 * 7));
end
superheated = 1;
elseif (y(9,t+1)==999)
    supercritical<1
    fprintf('supercritical state (T > Tc) and (p > Pc)\n');
    fprintf('\%g
', (t-1) * dltt/(24 * 3600 * 7));
end
supercritical = 1;
elseif (y(9,t+1)<0)
    y(9,t+1) = 0;
    subcooled<1
    fprintf('\subcooled liquid\n');
    fprintf('\%g
', (t-1) * dltt/(24 * 3600 * 7));
end
subcooled = 1;
elseif (y(9,t+1)>1)
    y(9,t+1) = 1;
end
% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q = 1 indicates saturated vapor
134 Program Code

% q > 1 indicates superheated vapor
% q = 998 superheated vapor, but quality not defined (t > Tc)
% q = 999 indicates supercritical state (t > Tc) and (p > Pc)

% dy(9) = refpropm('Q','D',mass/(AT * HT),'H',y(1)/mass,'methane','ethane','...
% 'nitrogen',{ .943648747220886 5.03466958769393e-02 6.0045569021752e-03}) - y(9);

% temperature
y(6,t+1) = refpropm('T','D',mass/(AT(1) * HT(1)),'U',y(10,t+1)/mass,'methane','ethane','...
% 'nitrogen',{ .943648747220886 5.03466958769393e-02 6.0045569021752e-03}) - y(6);

% liquid density per tank
if (y(9,t+1)>=1)&&(y(9,t+1)<998.5)
  y(4,t+1) = y(4,t);
elseif (y(9,t+1)<=0)||(y(6,t+1)>=tcrit)||(y(9,t+1)>998.5)
  y(4,t+1) = mass/(AT * HT);
else
  y(4,t+1) = refpropm('D','T',y(6,t+1),'Q',0,'methane');
end
% dy(4) = refpropm('D','T',y(6),'Q',0,'methane','ethan e','nitrogen',
% [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(4);

% liquid height per tank
y(8,t+1) = min(HT,max(0,((1-y(9,t+1)) * mass/y(4,t+1))/AT));

% gas density per tank
if (y(9,t+1)>=1)||(y(6,t+1)>tcrit)
  y(5,t+1) = mass/(AT * HT);
elseif (y(9,t+1)<=0)
  y(5,t+1) = y(5,t);
else
  y(5,t+1) = max(0,(y(9,t+1) * mass)/(AT(1) * (HT(1)-y(8,t+1))));
end
% gas pressure per tank
if y(9,t+1)<=0
  y(7,t+1) = 1000 * refpropm('P','T',y(6,t+1),'D',mass/(AT+HT),'methane');
else
  y(7,t+1) = 1000 * refpropm('P','T',y(6,t+1),'D',y(5,t+1),'methane');
end
% dy(7) = 1000 * refpropm('P','T',y(6),'D',y(5),'methane') - y(7);

% specific liquid enthalpy per tank
% if (y(9,t+1)>998.5) | | (y(6,t+1)>tcrit)
%  y(2,t+1) = refpropm('H','T',y(6,t+1),'D',y(5,t+1),'methane');
% elseif (y(9,t+1)==998)
%  y(2,t+1) = refpropm('H','T',y(6,t+1),'Q',0,'methane');
% else
%  y(2,t+1) = refpropm('H','T',y(6,t+1),'D',y(4,t+1),'methane');
% end
% dy(2) = refpropm('H','T',y(6),'D',y(4),'methane','ethane','nitrogen',
% [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(2);

% specific gas enthalpy per tank
% dy(3) = refpropm('H','T',y(6),'D',y(5),'methane') - y(3);
y(3,t+1) = refpropm('H','T',y(6,t+1),'D',y(5,t+1),'methane');
% dy(3) = refpropm('H','T',y(6),'D',y(5),'methane','ethane','nitrogen',
% [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(3);
end

toc
C.3 One-Tank System with Variable Temperature

\[ T = [0:dt:t_{\text{max}}]; \]

\begin{verbatim}
figure (1);
plot( T/(3600*24*7),y(8,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$H$(m)');
h = legend('$H(t)$', 'location', 'northeast');
figure (2);
plot( T/(3600*24*7),y(7,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$p$(Pa)');
h = legend('$p_{\text{gas}}$', 'location', 'southeast');
figure (3);
plot( T/(3600*24*7),y(2,:), T/(3600*24*7),y(3,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$h$(J)');
h = legend('$h_{\text{liq}}(t)$', '$h_{\text{gas}}(t)$', 'location', 'southeast');
figure (4);
plot( T/(3600*24*7),y(6,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$T$(K)');
h = legend('$T(t)$', 'location', 'southeast');
figure (5);
plot( T/(3600*24*7),y(1,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$H$(J)');
h = legend('$H(t)$', 'location', 'southeast');
figure (6);
plot( T/(3600*24*7),AT.*y(8,:).*y(4,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$m_{\text{liq}}$(kg)');
h = legend('$m_{\text{liq}}(t)$', 'location', 'northeast');
figure (7);
plot( T/(3600*24*7),y(9,:), 'LineWidth', 1.8);
grid;
xlabel( '$t$(weeks)');
ylabel( '$\xi$(kg/kg)');
h = legend('$\xi(t)$', 'location', 'southeast');
values(:)=y(:,700/delta+1);
\end{verbatim
function [values]=onetankrk1(rtol)

% One-tank model with heat flux from environment,
% solves set of differential algebraic equations (DAEs) by ode15s or ode23t
% Enthalpy equations are based on equilibrium, no z-discretisation, i.e.
% specific gas and liquid enthalpy are constant in space per tank
% Using internal energy
% LNG = 'methane','ethane','nitrogen',[ .943648747220886 5.03466958769393e-02 ...
% 6.0045569021752e-03]
tic

% Geometric and physical parameters
% gravitation constant
g = 9.81;
% tank height
HT = 30;
% tank diameter
DT = 20;
AT = pi.*((DT/2).^2); % cross-sectional area tank

% a constant singular mass matrix
M = zeros(12,12);
M(1,1) = 1; % enthalpy
M(10,10) = 1; % initial quality
M(11,11) = 1; % initial temperature
M(12,12) = 1; % internal energy

% Initial conditions
y0 = zeros(12,1);
values = zeros(12,1);
y0(10) = .01; % initial quality=vapor fraction kg/kg
y0(11) = 111; % initial temperature
y0(9) = y0(10); % quality
y0(6) = y0(11); % temperature

% mass
mass = AT*HT*refpropm('D','T',y0(6),'Q',y0(9),'methane');

% enthalpy
y0(1) = mass*refpropm('H','T',y0(6),'Q',y0(9),'methane');

% internal energy
y0(12) = mass*refpropm('U','T',y0(6),'Q',y0(9),'methane');

% liquid density
y0(4) = refpropm('D','T',y0(6),'Q',0,'methane');

% specific liquid enthalpy
y0(2) = refpropm('H','T',y0(6),'D',y0(4),'methane')
y0(2) = refpropm('H','T',y0(6),'Q',0,'methane'); % more precise?

% liquid height
y0(8) = (((1-y0(9))*mass)/y0(4))/AT;

% gas density
y0(5) = (y0(9)*mass)/(AT*(HT-y0(8)));

% specific gas enthalpy
y0(3) = refpropm('H','T',y0(6),'D',y0(5),'methane')
y0(3) = refpropm('H','T',y0(6),'Q',1,'methane');
C.3 One-Tank System with Variable Temperature

gas pressure
\[
y_0(7) = 1000 \cdot \text{refpropm}('P', 'T', y_0(6), 'D', y_0(5), 'methane');
\]
\[
t_{\text{start}} = 0;
\]
\[
t_{\text{final}} = 3600 \times 24 \times 7 \times 100;
\]
\[
t_{\text{out}} = t_{\text{start}};
\]
\[
y_{\text{out}} = y_0.;
\]
\[
t_{\text{eout}} = [];
\]
\[
y_{\text{eout}} = [];
\]
\[
i_{\text{eout}} = [];
\]
\[
\text{events} = 0;
\]
\[
\text{if events>1}
\]
\[
options = \text{odeset('Events', @events1, 'Mass', M, 'RelTol', rtol, 'MaxOrder', 5);
\]
\[
\text{Solve until the first terminal event.}
\]
\[
[t, y, t_{\text{e}}, y_{\text{e}}, i_{\text{e}}] = \text{ode23t}(@\text{onetanktwoPhase}, [t_{\text{start}} t_{\text{final}}], y_0, \text{options});
\]
\[
\text{Accumulate output. This could be passed out as output arguments.}
\]
\[
\text{nt} = \text{length}(t);
\]
\[
t_{\text{out}} = [t_{\text{out}}; t(2:nt)];
\]
\[
y_{\text{out}} = [y_{\text{out}}; y(2:nt,:)];
\]
\[
t_{\text{eout}} = [t_{\text{eout}}; t_{\text{e}}];
\]
\[
y_{\text{eout}} = [y_{\text{eout}}; y_{\text{e}}];
\]
\[
i_{\text{eout}} = [i_{\text{eout}}; i_{\text{e}}];
\]
\[
\text{while } (\text{abs}(y(n,9) - 1) > 1e-3) \&\& (\text{abs}(y(n,9) - 1e-3) > 0) \&\& (t(n) < t_{\text{final}})
\]
\[
\text{fprintf('run loop');}
\]
\[
\text{t}_{\text{start}} = t(n);
\]
\[
y_0(:, 1) = y(n, :);
\]
\[
\text{options = odeset('Events', @events2, 'Mass', M, 'RelTol', rtol, 'MaxOrder', 5);
\]
\[
\text{Solve until the first terminal event.}
\]
\[
[t, y, t_{\text{e}}, y_{\text{e}}, i_{\text{e}}] = \text{ode15s}(@\text{onetanktwoPhase}, [t_{\text{start}} t_{\text{final}}], y_0, \text{options});
\]
\[
\text{Accumulate output. This could be passed out as output arguments.}
\]
\[
\text{nt} = \text{length}(t);
\]
\[
t_{\text{out}} = [t_{\text{out}}; t(2:nt)];
\]
\[
y_{\text{out}} = [y_{\text{out}}; y(2:nt,:)];
\]
\[
t_{\text{eout}} = [t_{\text{eout}}; t_{\text{e}}];
\]
\[
y_{\text{eout}} = [y_{\text{eout}}; y_{\text{e}}];
\]
\[
i_{\text{eout}} = [i_{\text{eout}}; i_{\text{e}}];
\]
\[
\text{end}
\]
\[
\text{if } (\text{abs}(y(n,9) - 1) < 1e-2) \&\& (y(n,9) < 900) \&\& (t(n) < t_{\text{final}})
\]
\[
\text{fprintf('quality 1');}
\]
\[
a constant singular mass matrix
\]
\[
M_2 = \text{zeros}(12, 12);
\]
\[
M_2(1, 1) = 1; \quad \% \text{enthalpy}
\]
\[
M_2(2, 2) = 1; \quad \% \text{specific liquid enthalpy}
\]
\[
M_2(4, 4) = 1; \quad \% \text{liquid density}
\]
\[
M_2(8, 8) = 1; \quad \% \text{liquid height}
\]
\[
M_2(10, 10) = 1; \quad \% \text{initial quality}
\]
\[
M_2(11, 11) = 1; \quad \% \text{initial temperature}
\]
\[
M_2(12, 12) = 1; \quad \% \text{internal energy}
\]
\[
\text{Set the new initial conditions, with quality 1}
\]
\[
y_0(10) = y(n,10);
\]
\[
y_0(11) = y(n,11);
\]
\[
y_0(9) = 1;
\]
\[
y_0(6) = y(n,6);
\]
\[
\text{mass}
\]
\[
\text{mass} = A T(1) \cdot \text{HT}(1) \cdot \text{refpropm}('D', 'T', y_0(11), 'Q', y_0(10), 'methane');
\]
138 Program Code

% enthalpy
y0(1) = mass*refpropm('H','T',y0(6),'Q',y0(9),'methane');

% internal energy
y0(12) = mass*refpropm('U','T',y0(6),'Q',y0(9),'methane');

% liquid density
y0(4) = refpropm('D','T',y0(6),'Q',0,'methane');

% specific liquid enthalpy
y0(2) = refpropm('H','T',y0(6),'D',y0(4),'methane');

% liquid height
y0(8) = 0;

% gas density
y0(5) = mass/(AT*HT);

% specific gas enthalpy
y0(3) = refpropm('H','T',y0(6),'D',y0(5),'methane');

% gas pressure
y0(7) = 1000*refpropm('P','T',y0(6),'D',y0(5),'methane');

145 % A good guess of a valid first timestep is the length of the last valid
% timestep, so use it for faster computation. 'refine' is 4 by default.

150 tstart = t(nt);
options3 = odeset('Events',@events2,'Mass',M2,'RelTol',rtol,'MaxOrder',5);
% Solve until the first terminal event.
[t,y,te,ye,ie] = ode23t(@onetankonephase1,[tstart tfinal],y0,options3);

155 elseif t(nt)<tfinal
  fprintf('quality 0');

158 M2 = zeros(11,11);
M2(1,1) = 1;  % mass
M2(3,3) = 1;  % specific gas enthalpy
M2(5,5) = 1;  % density
M2(7,7) = 1;  % gas pressure
M2(8,8) = 1;  % liquid height
M2(9,9) = 1;  % quality
M2(10,10) = 1; % initial quality
M2(11,11) = 1; % initial temperature
M2(12,12) = 1; % internal energy

163 % Set the new initial conditions, with quality 0
y0(10) = y(nt,10);
y0(11) = y(nt,11);
y0(9) = 0;
y0(6) = y(nt,6);

168 mass = AT(1)*HT(1)*refpropm('D','T',y(nt,11),'Q',y(nt,10),'methane');

172 if t(nt)<tfinal
  fprintf('quality 0');

175 tstart = t(nt);
options3 = odeset('Events',@events2,'Mass',M2,'RelTol',rtol,'MaxOrder',5);
% Solve until the first terminal event.
[t,y,te,ye,ie] = ode23t(@onetankonephase1,[tstart tfinal],y0,options3);

180 elseif t(nt)<tfinal
  fprintf('quality 0');

185 M2 = zeros(11,11);
M2(1,1) = 1;  % mass
M2(3,3) = 1;  % specific gas enthalpy
M2(5,5) = 1;  % density
M2(7,7) = 1;  % gas pressure
M2(8,8) = 1;  % liquid height
M2(9,9) = 1;  % quality
M2(10,10) = 1; % initial quality
M2(11,11) = 1; % initial temperature
M2(12,12) = 1; % internal energy

190 % Set the new initial conditions, with quality 0
y0(10) = y(nt,10);
y0(11) = y(nt,11);
y0(9) = 0;
y0(6) = y(nt,6);

195 mass = AT(1)*HT(1)*refpropm('D','T',y(nt,11),'Q',y(nt,10),'methane');
C.3 One-Tank System with Variable Temperature

```matlab
% enthalpy
\$ \hat{y}_0(1) = mass * refpropm('H','T',y_0(6),'Q',y_0(9),'methane'); \$
\$ y_0(1) = y(nt,1); \$
% internal energy
\$ \hat{y}_0(12) = mass * refpropm('U','T',y_0(6),'Q',y_0(9),'methane'); \$
\$ y_0(12) = y(nt,12); \$
% liquid density
\$ y_0(4) = mass / (AT * HT); \$
% specific liquid enthalpy
\$ y_0(2) = refpropm('H','T',y_0(6),'D',y_0(4),'methane'); \$
\$ y_0(2) = refpropm('H','T',y_0(6),'Q',y_0(9),'methane'); \$
% liquid height
\$ y_0(8) = HT; \$
% gas density
\$ y_0(5) = y(nt,5); \$
% specific gas enthalpy
\$ y_0(3) = refpropm('H','T',y_0(6),'D',y_0(5),'methane'); \$
\$ y_0(3) = y(nt,3); \$
% gas pressure
\$ y_0(7) = 1000 * refpropm('P','T',y_0(6),'D',y_0(5),'methane'); \$
% A good guess of a valid first timestep is the length of the last valid
timestep, so use it for faster computation. ‘refine’ is 4 by default.
t0start = t(nt);
options2 = odeset('Mass',M2,'RelTol',rtol);
[t,y] = ode23t(@onetankonephase2,[tstart tfinal],y0,options2);
nt = length(t);
tout = [tout; t(2:nt)];
yout = [yout; y(2:nt,:)];
teout = [teout; te]; \% Events at tstart are never reported.
yeout = [yeout; ye];
ieout = [ieout; ie];
end

if (y(nt,9)>998)&&(t(nt)<tfinal) \%supercritical
    fprintf('quality 999');
    \% a constant singular mass matrix
    M2 = zeros(12,12);
\$ M2(1,1) = 1; \% enthalpy
\$ M2(2,2) = 1; \% liquid enthalpy
\$ M2(4,4) = 1; \% liquid density
\$ M2(8,8) = 1; \% liquid height
\$ M2(10,10) = 1; \% initial quality
\$ M2(11,11) = 1; \% initial temperature
\$ M2(12,12) = 1; \% internal energy
\$ Set the new initial conditions, with quality 999
\$ y0(9) = 999;
\$ y0(6) = y(nt,6);
\$ mass
\$ mass = AT(1) * HT(1) * refpropm('D','T',y(nt,11),'Q',y(nt,10),'methane'); \$
\$ enthalpy
\$ y0(1) = y(nt,1); \$
```
% internal energy
y0(12) = y(nt,12);

% liquid density
y0(4) = y(nt,4);

% specific liquid enthalpy
% y0(2) = refpropm('H','T',y0(6),'D',y0(4),'methane')
y0(2) = y(nt,2);

% liquid height
y0(8) = 0;

% gas density
y0(5) = mass/(AT*HT);

% specific gas enthalpy
% y0(3) = refpropm('H','T',y0(6),'D',y0(5),'methane')
y0(3) = y(nt,3);

% gas pressure
y0(7) = y(nt,7);

% A good guess of a valid first timestep is the length of the last valid
timestep, so use it for faster computation. ‘refine’ is 4 by default.
tstart = t(nt);
options2 = odeset('Mass',M2,'RelTol',rtol);
[t,y] = ode23t(@onetankonephase3,[tstart tfinal],y0,options2);

nt = length(t);
tout = [tou; t(2:nt)];
yout = [yout; y(2:nt,:)];
% Events at tstart are never reported.
yeout = [yeout; ye];
ieout = [ieout; ie];

else
options = odeset('Mass',M,'RelTol',rtol,'MaxOrder',5);
[t,y] = ode15s(@onetanktwophase,[tstart 100*7*24*3600],y0,options);

% Accumulate output. This could be passed out as output arguments.
t = length(t);
tout = [tout; t(2:nt)];
yout = [yout; y(2:nt,:)];
values(:) = y(nt,:);
y0(:) = y(nt,:);
end

toc

circlesingraphs = 0;
if circlesingraphs == 1
figure(1);
plot( tout/(3600*24*7),yout(:,8),'-o','LineWidth',1.8);
grid;
xlabel('t (weeks)');
C.3 One-Tank System with Variable Temperature

```matlab
ylabel('$H$(m)');

h = legend('$H(t)$','location','southwest');

figure(2);
plot(tout/(3600*24*7),yout(:,7),'-o','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$p$(Pa)');
h = legend('$p_{gas}(t)$','location','southeast');

figure(3);
plot(tout/(3600*24*7),yout(:,2),'-o',tout/(3600*24*7),yout(:,3),'-o','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$h$(J)');
h = legend('$h_{liq}(t)$','$h_{gas}(t)$','location','southeast');

figure(4);
plot(tout/(3600*24*7),yout(:,6),'-o','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$T$(K)');
h = legend('$T(t)$','location','southeast');

figure(5);
plot(tout/(3600*24*7),yout(:,1),'-o','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$H$(J)');
h = legend('$H(t)$','location','southwest');

figure(6);
plot(tout/(3600*24*7),AT.*yout(:,8).*yout(:,4),'-o','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$m_{liq}$(kg)');
h = legend('$m_{liq}(t)$','location','southeast');

figure(7);
plot(tout/(3600*24*7),yout(:,5),'-o','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$\rho_{gas}$(kg/m$^3$)');
h = legend('$\rho_{gas}(t)$','location','northeast');

else
    figure(1);
    plot(tout/(3600*24*7),yout(:,8),'-','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$H$(m)');
h = legend('$H(t)$','location','southwest');

figure(2);
plot(tout/(3600*24*7),yout(:,7),'-','LineWidth',1.8);
grid;
xlabel('$t$(weeks)');
ylabel('$p_{gas}(t)$');
```

\[ h = \text{legend}('h_{\text{gas}}(t)', 'location', 'southeast'); \]

\[
\text{figure}(3); \\
\text{plot}(\text{tout}/(3600*24*7),\text{yout}(;2),'-',\text{tout}/(3600*24*7),\text{yout}(;3),'-', 'LineWidth', 1.8 )... \\
\text{grid}; \\
\text{xlabel}('t$(\text{weeks})'); \\
\text{ylabel}('h$(\text{J})'); \\
\text{h = legend}('h_{\text{liq}}(t)', 'h_{\text{gas}}(t)', 'location', 'southeast'); \\
\text{figure}(4); \\
\text{plot}(\text{tout}/(3600*24*7),\text{yout}(;6),'-', 'LineWidth', 1.8 ); \\
\text{grid}; \\
\text{xlabel}('t$(\text{weeks})'); \\
\text{ylabel}('T$(\text{K})'); \\
\text{h = legend}('T(t)', 'location', 'southeast'); \\
\text{figure}(5); \\
\text{plot}(\text{tout}/(3600*24*7),\text{yout}(;1),'-', 'LineWidth', 1.8 ); \\
\text{grid}; \\
\text{xlabel}('t$(\text{weeks})'); \\
\text{ylabel}('H$(\text{J})'); \\
\text{h = legend}('H(t)', 0); \\
\text{figure}(6); \\
\text{plot}(\text{tout}/(3600*24*7),\text{AT}.*\text{yout}(;8).*\text{yout}(;4),'-', 'LineWidth', 1.8 ); \\
\text{grid}; \\
\text{xlabel}('t$(\text{weeks})'); \\
\text{ylabel}('m_{\text{liq}}$(\text{kg})'); \\
\text{h = legend}('m_{\text{liq}}(t)', 'location', 'southeast'); \\
\text{figure}(7); \\
\text{plot}(\text{tout}/(3600*24*7),\text{yout}(;5),'-', 'LineWidth', 1.8 ); \\
\text{grid}; \\
\text{xlabel}('t$(\text{weeks})'); \\
\text{ylabel}('\rho_{\text{gas}}$(\text{kg/m}^3)'); \\
\text{h = legend}('\rho_{\text{gas}}(t)', 0); \\
\text{figure}(8); \\
\text{plot}(\text{tout}/(3600*24*7),\text{yout}(;9),'-', 'LineWidth', 1.8 ); \\
\text{grid}; \\
\text{xlabel}('t$(\text{weeks})'); \\
\text{ylabel}('\xi$(\text{kg/kg})'); \\
\text{h = legend}('\xi(t)', 'location', 'southwest'); \\
\text{end} \\
\]

%-------------------------------------------------- -----------------------% 
\[
\text{function} dy = \text{onetanktwophase}(t,y) \\
% Geometric and physical parameters \\
g = 9.81; \quad \% gravitation constant \\
\text{HT} = 30; \quad \% tank height \\
\text{DT} = 20; \quad \% tank diameter \\
\text{AT} = \pi.*\text{(DT}/2).^2; \quad \% cross-sectional area tanks \\
\]
temperatureamb1 = 293;

K = 55;
tcrit = refpropm('T', 'D', 10^10, 'Q', 1, 'methane');

mass = AT*HT*refpropm('D', 'T', y(11), 'Q', y(10), 'methane');

% mass = AT*HT*refpropm('D', 'T', temperature, 'Q', quality, 'methane', 'ethane', ...

% 'nitrogen', [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

dy = zeros(12,1);
dy(10) = 0;
dy(11) = 0;

% enthalpy update
dy(1) = -K*(y(6)-temperatureamb1);
% internal energy update
dy(12) = -K*(y(6)-temperatureamb1);
% quality update

% enthalpy update
dy(9) = refpropm('Q', 'D', mass/(AT*HT), 'U', y(12)/mass, 'methane') - y(9);

% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q = 1 indicates saturated vapor
% q > 1 indicates superheated vapor
% q = 998 superheated vapor, but quality not defined (t > Tc)
% q = 999 indicates supercritical state (t > Tc) and (p > Pc)

% temperature
dy(6) = refpropm('T', 'D', mass/(AT*HT), 'U', y(12)/mass, 'methane') - y(6);

% liquid density per tank
if y(6)<tcrit
    dy(4) = refpropm('D', 'T', y(6), 'Q', 0, 'methane') - y(4);
else
    dy(4) = mass/(AT*HT) - y(4);
end

% quality update

% liquid height per tank
if (y(9)>=1)
    dy(8) = max(0, ((1-y(9))*mass/y(4))/AT) - y(8);
else
    dy(8) = (max((1-y(9)),0)*mass/y(4))/(AT) - y(8);
end

% gas density per tank
if (y(9)>=1)
    dy(5) = mass/(AT*HT) - y(5);
else
    dy(5) = max(0, (y(9)*mass)/(AT*(HT-y(8)))) - y(5);
end

% gas pressure per tank
if (y(9)>=1)
    dy(7) = 1000*refpropm('P', 'T', y(6), 'D', mass/(AT*HT), 'methane') - y(7);
else
    dy(7) = 1000*refpropm('P', 'T', y(6), 'D', y(5), 'methane') - y(7);
end

% specific liquid enthalpy per tank
if (abs(y(9)-999)<0.1)^
    dy(2) = refpropm('H', 'T', y(6), 'D', mass/(AT*HT), 'methane') - y(2);
else
    dy(2) = refpropm('H', 'T', y(6), 'Q', 0, 'methane') - y(2);
end
Specific gas enthalpy per tank

```matlab
if (y(9)>=1)
    dy(3) = refpropm('H', 'T', y(6), 'D', mass/(AT*HT), 'methane') - y(3);
else
    dy(3) = refpropm('H', 'T', y(6), 'Q', 1, 'methane') - y(3);
end
```

Function dy = onetankonephase1(t,y) % if we are left with superheated vapor

```matlab
function dy = onetankonephase1(t,y)
% if we are left with superheated vapor
% Geometric and physical parameters
g = 9.81;  % gravitation constant
HT = 30;  % tank heights
DT = 20;  % tank diameters
AT = pi.*((DT./2).^2);  % cross-sectional area tanks

mass = AT*HT*refpropm('D','T',y(11),'Q',y(10),'methane');
% mass = AT*HT*refpropm('D','T',temperature,'Q',quality,'methane','ethane','...
% 'nitrogen',[ .94364874720886 5.03466958769393e-02 6.0045569021752e-03]);
dy = zeros(12,1);

% enthalpy update
dy(1) = -K*(y(6)-temperatureamb1);
% internal energy update
dy(12) = -K*(y(6)-temperatureamb1);

% quality update
if (y(9,t+1) == 998)
    % if superheated vapor, but quality not defined (T > Tc)
    fprintf('superheated vapor, but quality not defined (T > Tc)
');
    fprintf('%g
',(t-1) * dltt/(24 * 3600 * 7));
else
    % if supercritical
    fprintf('supercritical state (T > Tc) and (p > Pc)
');
    fprintf('%g
',(t-1) * dltt/(24 * 3600 * 7));
end

% subcooled liquid
if (y(9,t+1) == 999)
    % if supercritical
    fprintf('subcooled liquid
');
    fprintf('%g
',(t-1) * dltt/(24 * 3600 * 7));
% end
% supercritical
else
    % if subcooled liquid
    fprintf('subcooled liquid
');
    fprintf('%g
',(t-1) * dltt/(24 * 3600 * 7));
end
```

Quality update

```matlab
if (refpropm('Q', 'D', mass/(AT*HT), 'U', y(12)/mass, 'methane')>1) && refpropm('Q', 'D', mass/(AT... + HT), 'U', y(12)/mass, 'methane')<998
    dy(9) = 1 - y(9)
else
    dy(9) = refpropm('Q', 'D', mass/(AT*HT), 'U', y(12)/mass, 'methane') - y(9);
end
```

% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
C.3 One-Tank System with Variable Temperature

590 \[ q = 0 \] indicates saturated liquid
591 \[ q = 1 \] indicates saturated vapor
592 \[ q > 1 \] indicates superheated vapor
593 \[ q = 998 \] superheated vapor, but quality not defined \( (t > T_c) \)
594 \[ q = 999 \] indicates supercritical state \( (t > T_c) \) and \( (p > P_c) \)

595 \textbf{temperature}
596 \[ dy(6) = \text{refpropm}'T', 'D', \text{mass}/(\text{AT*HT}), 'U', y(12)/\text{mass}, 'methane' \) - \( y(6) \);
597 \textbf{liquid density per tank}
598 \[ dy(4) = 0; \]
599 \textbf{liquid height per tank}
600 \[ dy(8) = 0; \]
601 \textbf{gas density per tank}
602 \[ dy(5) = \text{mass}/(\text{AT*HT}) - y(5); \]
603 \textbf{gas pressure per tank}
604 \[ dy(7) = 1000*\text{refpropm}'P', 'T', y(6), 'D', \text{mass}/(\text{AT*HT}), 'methane' \) - \( y(7) \);
605 \textbf{specific gas enthalpy per tank}
606 \[ dy(3) = \text{refpropm}'H', 'T', y(6), 'D', y(5), 'methane' \) - \( y(3) \);
607 \textbf{specific liquid enthalpy per tank}
608 \[ dy(2) = \begin{cases} \text{refpropm}'H', 'T', y(6), 'D', y(4), 'methane' \) - \( y(2) \); & if \( |y(9) - 999| < 0.1 \) or \( y(6) > tcrit \) \\ \text{refpropm}'H', 'T', y(6), 'Q', 0, 'methane' \) - \( y(2) \); & if \( |y(9) - 998| < 5e-1 \) \\ \text{refpropm}'H', 'T', y(6), 'D', y(4), 'methane' \) - \( y(2) \); & else \end{cases} \]
609 \[ dy(2) = 0; \]
610 \textbf{gravitational constant}
611 \[ g = 9.81; \]
612 \textbf{gravitational constant}
613 \[ HT = 30; \]
614 \textbf{tank heights}
615 \[ DT = 20; \]
616 \textbf{tank diameters}
617 \[ AT = \pi.(DT./2).^2; \]
618 \textbf{cross-sectional area tanks}
619 \[ \text{temperatureamb1} = 293; \]
620 \[ \text{tcrit} = \text{refpropm}'T', 'D', 10^-10, 'Q', 1, 'methane' \);
621 \[ dy(10) = 0; \]
622 \[ dy(11) = 0; \]
623 \textbf{mass}
624 \[ \text{mass} = \text{AT*HT*refpropm}'D', 'T', y(11), 'Q', y(10), 'methane' \);
625 \[ \text{mass} = \text{AT*HT*refpropm}'D', 'T', \text{temperature}, 'Q', \text{quality}, 'methane', 'ethane', 'nitrogen', ' \); \\
626 \[ \text{mass} = \text{AT*HT*refpropm}'D', 'T', \text{temperature}, 'Q', \text{quality}, 'methane', 'ethane', 'nitrogen', ' \); \\
627 \textbf{enthalpy update}
628 \[ dy(1) = -K*\{y(6) - \text{temperatureamb1}\}; \]
629 \textbf{enthalpy update}
630 \[ dy(12) = -K*\{y(6) - \text{temperatureamb1}\}; \]
631 \textbf{quality update}
632 \[ dy(9) = \text{refpropm}'Q', 'D', \text{mass}/(\text{AT+HT}), 'U', y(12)/\text{mass}, 'methane' \) - \( y(9) \);
633 \textbf{quality update}
634 \[ dy(9) = \text{refpropm}'Q', 'D', \text{mass}/(\text{AT+HT}), 'U', y(12)/\text{mass}, 'methane' \) - \( y(9) \);
635 \textbf{quality update}
636 \[ \text{if} \ q < 0 \ \text{indicates subcooled (compressed) liquid} \]
637 \textbf{quality update}
638 \[ \text{if} \ q < 0 \ \text{indicates subcooled (compressed) liquid} \]
639 \textbf{quality update}
640 \[ \text{if} \ q = 0 \ \text{indicates saturated liquid} \]
641 \textbf{quality update}
642 \[ \text{if} \ q = 0 \ \text{indicates saturated liquid} \]
643 \textbf{quality update}
644 \[ \text{if} \ q = 0 \ \text{indicates saturated liquid} \]
% q > 1 indicates superheated vapor
% q = 998 superheated vapor, but quality not defined (t > Tc)
% q = 999 indicates supercritical state (t > Tc) and (p > Pc)

% temperature
dy(6) = refpropm('T', 'D', mass/(AT * HT), 'U', y(12)/mass, 'methane') - y(6);

% liquid density per tank
dy(4) = 0;

% dy(4) = refpropm('D', 'T', y(6), 'Q', 0, 'methane', 'ethane', 'nitrogen', ...
% [ .943648747220886 5.0346695876933e-02 6.0045569021752e-03]) - y(4);

% liquid height per tank
dy(8) = 0;

% gas density per tank
dy(5) = mass/(AT * HT) - y(5);

% gas pressure per tank
dy(7) = 1000 * refpropm('P', 'T', y(6), 'D', mass/(AT * HT), 'methane') - y(7);

% specific gas enthalpy per tank
dy(3) = refpropm('H', 'T', y(6), 'D', mass/(AT * HT), 'methane') - y(3);

% specific liquid enthalpy per tank
if (abs(y(9)-999)<0.1)||(y(6)>tcrit)
   dy(2) = refpropm('H', 'T', y(6), 'D', mass/(AT * HT), 'methane') - y(2);
elseif abs(y(9)-998)<5e-1
   dy(2) = refpropm('H', 'T', y(6), 'Q', 0, 'methane') - y(2);
else
   dy(2) = refpropm('H', 'T', y(6), 'D', y(4), 'methane') - y(2);
end

dy(2) = y(3) - y(2);

function dy = onetankonephase2(t,y)  % if we are left with only (subcooled) liquid
% Geometric and physical parameters
g = 9.81;  % gravitation constant
HT = 30;  % tank heights
DT = 20;  % tank diameters
AT = pi.*((DT./2).^2);  % cross-sectional area tanks
temperatureamb1 = 293;
K = 55;
tcrit = refpropm('T', 'D', 10^10, 'Q', 1, 'methane');
mass = AT * HT * refpropm('D', 'T', y(11), 'Q', y(10), 'methane');
% mass = AT*HT*refpropm('D', 'T', temperature, 'Q', quality, 'methane', 'ethane', ...
% 'nitrogen', [.943648747220886 5.0346695876933e-02 6.0045569021752e-03]);
dy = zeros(12,1);
dy(10) = 0;
dy(11) = 0;

% enthalpy update
dy(1) = -K * (y(6) - temperatureamb1);
% internal energy update
dy(12) = -K * (y(6) - temperatureamb1);
% quality update
dy(9) = 0;

% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q = 1 indicates saturated vapor
% q > 1 indicates superheated vapor
C.3 One-Tank System with Variable Temperature

$q = 998$ superheated vapor, but quality not defined ($t > T_c$)
$q = 999$ indicates supercritical state ($t > T_c$) and ($p > P_c$)

\[
\text{temperature}
\]
\[
d y(6) = \text{refpropm}\left('T', 'D', \text{mass}/(AT*HT), 'U', y(12)/\text{mass}, 'methane'\right) - y(6);
\]

\[
\text{liquid density per tank}
\]
\[
d y(4) = \text{mass}/(AT*HT) - y(4);
\]

\[
\text{dy(4) = refpropm('D', 'T', y(6), 'Q', 0, 'methane', 'ethane', 'nitrogen', ...}
\]
\[
\text{[.943648747220886 5.03466958769393e-02 6.0045569021752e-03]} - y(4);
\]

\[
\text{liquid height per tank}
\]
\[
d y(8) = 0;
\]

\[
\text{gas density per tank}
\]
\[
d y(5) = 0;
\]

\[
\text{gas pressure per tank}
\]
\[
d y(7) = 0;
\]

\[
\text{specific liquid enthalpy per tank}
\]
\[
d y(2) = \text{refpropm('H', 'T', y(6), 'D', \text{mass}/(AT*HT), 'methane') - y(2)};
\]

\[
\text{specific gas enthalpy per tank}
\]
\[
\text{if } y(6) < tcrit
\]
\[
\text{dy(3) = refpropm('H', 'T', y(6), 'D', y(5), 'methane') - y(3)};
\]
\[
\text{else}
\]
\[
\text{dy(3) = y(2) - y(3)};
\]

\[
\text{--------------------------------------------------}
\]
\[
\text{function [value, isterminal, direction] = events1(t, y)}
\]
\[
\text{value(1) = y(9) - 1; \hspace{1cm} \% detect quality = 1}
\]
\[
\text{isterminal(1) = 1; \hspace{1cm} \% stop the integration}
\]
\[
direction(1) = 1; \hspace{1cm} \% positive direction
\]
\[
\text{value(2) = y(9); \hspace{1cm} \% detect quality = 0}
\]
\[
\text{isterminal(2) = 1; \hspace{1cm} \% stop the integration}
\]
\[
direction(2) = -1; \hspace{1cm} \% negative direction
\]

\[
\text{--------------------------------------------------}
\]
\[
\text{function [value, isterminal, direction] = events2(t, y)}
\]
\[
\text{value(1) = y(9) - 1; \hspace{1cm} \% detect quality = 1}
\]
\[
\text{isterminal(1) = 1; \hspace{1cm} \% stop the integration}
\]
\[
direction(1) = 1; \hspace{1cm} \% positive direction
\]
\[
\text{value(2) = y(9); \hspace{1cm} \% detect quality = 0}
\]
\[
\text{isterminal(2) = 1; \hspace{1cm} \% stop the integration}
\]
\[
direction(2) = -1; \hspace{1cm} \% negative direction
\]
\[
\text{value(3) = y(9) - 999; \hspace{1cm} \% detect supercritical state}
\]
\[
\text{isterminal(3) = 1; \hspace{1cm} \% stop the integration}
\]
\[
direction(3) = 1; \hspace{1cm} \% positive direction
C.4 Two-Tank System with Variable Temperature

C.4.1 Explicit Euler for Two-Tank System with Upper Pipe with Variable Temperature

```matlab
function [values20] = twotanksenthalpyeuler(delta)

% This script computes numerical solutions of the 1D two-tank model
% with lower and upper pipe
% with enthalpy based on equilibrium per tank
% time integration: explicit Euler

% Meaning of the variables:
% HL height of lower pipe-tank connection
% HU height of upper pipe-tank connection
% H liquid height in tanks
% HT total height of tanks
% p0 initial gas pressure in tanks
% pin pressure tank 1 at z=0
% pout pressure tank 2 at z=0
% (z=0 at lower pipe-tank 1 connection)
% Dpl diameter of lower pipe
% Dpu diameter of upper pipe
% Dt1 diameter of left tank
% Dt2 diameter of right tank
% Vtot total liquid volume
% alpha friction factor

% theta parameter in the theta-method
% Nt number of time steps
% Tmax final time
% dt time step

% 'methane','ethane','nitrogen',[ .943648747220886 5.0 3466958769393e-02 6.0045569021752e-03]

format long e;
tic

% Geometric and physical parameters

% gravitation constant
g = 9.81;

% lower pipe connection
HL = [1 1];

% tank heights
HT = [20 10];

% assume Dp<<Dt
DP = [0.2 0.2];

% tank diameters
DT = [5 2];

alpha = 2; % friction factor for liquid flow in pipe,...
alpha1 = 1; % friction factor for gas flow in pipe
AP = pi.*((DP./2).^2);
AT = pi.*((DT./2).^2);
LP = [10 10]; % length of pipe
constraints = 2; % use constraints nr 1 for gas flow:1,
                 % otherwise

% grid (time)
Tmax = 200;
dltt = delta;
Nt = round(Tmax/dltt);
```

C.4 Two-Tank System with Variable Temperature

% variables
H = zeros(Nt+1,2);
pin = zeros(Nt+1,1);
pout = zeros(Nt+1,1);
pout = zeros(Nt+1,2);
rhog = zeros(Nt+1,2);
quality = zeros(Nt+1,2);

% initial conditions
quality(1,1) = 0.02;
quality(1,2) = 0.02;

% temperature
temperature(1,1) = 112;
temperature(1,2) = 125;

% total mass per tank
totaltankmass(1,1) = AT(1) * HT(1) * refpropm('D', 'T', temperature(1,1), 'Q', quality(1,1), ...
methane');
totaltankmass(1,2) = AT(2) * HT(2) * refpropm('D', 'T', temperature(1,2), 'Q', quality(1,2), ...
methane');

% total enthalpy per tank
totaltankenthalpy(1,1) = totaltankmass(1,1) * refpropm('H', 'T', temperature(1,1), 'D', rholiq(1,1), 'methane');
totaltankenthalpy(1,2) = totaltankmass(1,2) * refpropm('H', 'T', temperature(1,2), 'D', rholiq(1,2), 'methane');

% liquid density per tank
rholiq(1,1) = refpropm('D', 'T', temperature(1,1), 'Q', 0, 'methane');
rholiq(1,2) = refpropm('D', 'T', temperature(1,2), 'Q', 0, 'methane');

% specific liquid enthalpy per tank
specliqenthalpy(1,1) = refpropm('H', 'T', temperature(1,1), 'D', rholiq(1,1), 'methane');
specliqenthalpy(1,2) = refpropm('H', 'T', temperature(1,2), 'D', rholiq(1,2), 'methane');

% liquid height per tank
H(1,1) = ((1-quality(1,1)) * totaltankmass(1,1))/(rholiq(1,1) * AT(1));
H(1,2) = ((1-quality(1,2)) * totaltankmass(1,2))/(rholiq(1,2) * AT(2));

% gas density per tank
rhog(1,1) = (quality(1,1) * totaltankmass(1,1))/(AT(1) * (HT(1)-H(1,1)));
rhog(1,2) = (quality(1,2) * totaltankmass(1,2))/(AT(2) * (HT(2)-H(1,2)));

% specific gas enthalpy per tank
specgasenthalpy(1,1) = refpropm('H', 'T', temperature(1,1), 'D', rhog(1,1), 'methane');
specgasenthalpy(1,2) = refpropm('H', 'T', temperature(1,2), 'D', rhog(1,2), 'methane');

% gas pressure per tank
pgas(1,1) = 1000 * refpropm('P', 'T', temperature(1,1), 'D', rhog(1,1), 'methane');
pgas(1,2) = 1000 * refpropm('P', 'T', temperature(1,2), 'D', rhog(1,2), 'methane');

% pressure at end of lower pipe
pin(1) = pgas(1,1) + rholiq(1,1) * g * (max(H(1,1)-HL(1),0));
pout(1) = pgas(1,2) + rholiq(1,2) * g * (max(H(1,2)-HL(1),0));

fractionL = 1;
fractionU = 0;
150 Program Code

compute solution for closed system

for n=1:Nt
  Vliq1 = AT(1) * H(n,1);
  Vliq2 = AT(2) * H(n,2);
  Vliq = Vliq1+Vliq2 + fractionL * AP(1) * LP(1) +
        fractionU * AP(2) * LP(2);
  Mliq1 = rholiq(n,1) * AT(1) * H(n,1);
  Mliq2 = rholiq(n,2) * AT(2) * H(n,2);
  Mliq = Mliq1+Mliq2 + ((Mliq1+Mliq2)/(Vliq1+Vliq2)) * (
    fractionL * AP(1) * LP(1) + fractionU * AP(2) * LP(2));
  rholiqaverage = Mliq/Vliq;
  Vgas1 = AT(1) * (HT(1)-H(n,1));
  Vgas2 = AT(2) * (HT(2)-H(n,2));
  Vgas = Vgas1+Vgas2 + (1-fractionL) * AP(1) * LP(1) +
        (1-fractionU) * AP(2) * LP(2);
  Mgas1 = rhog(n,1) * AT(1) * (HT(1)-H(n,1));
  Mgas2 = rhog(n,2) * AT(2) * (HT(2)-H(n,2));
  Mgas = Mgas1+Mgas2 + ((Mgas1+Mgas2)/(Vgas1+Vgas2)) * ((1-fractionL) * AP(1) * LP(1) +
        (1-fractionU) * AP(2) * LP(2));
  rhogasaverage = Mgas/Vgas;

if (pin(n)-pout(n))/pin(n)>1e-13&&(H(n,1)-max(HL(1),HL(2))>1e-13)
  newH1 = max(H(n,1)-dltt * (AP(1)/AT(1)) *
             sqrt(abs(2/((1+alpha) * rholiq(n,2))) * (pin(n...)
             -pout(n))))/HL(1));
  newH2 = H(n,2)+AT(1) * (H(n,1)-newH1)/AT(2);
  ligenthalpyflux = (H(n,1)-newH1) * rholiq(2,1) * AT(1);
  ligenthalpyflux = ligenthalpyflux * specligenthalpy(n,1); % is positive
else
  lqmgasflux = 0;
  lqenthalpyflux = 0;
  newH1 = H(n,1);
  newH2 = H(n,2);
end

Vgas1 = AT(1) * (HT(1)-newH1);
Vgas2 = AT(2) * (HT(2)-newH2);
Mgasequilibrium1 = rhogasaverage*Vgas1;
Mgasequilibrium2 = rhogasaverage*Vgas2;

if (pgas(n,1)-pgas(n,2))/pgas(n,1)>1e-12
  gasvelocity = sqrt(2*(pgas(n,1)-pgas(n,2))/((1+alpha1)*rhog(n,2))) ;
  massflux = rhog(n,2)*AP(2)+gasvelocity;
  if constraints==1||constraints==2
    if Mgas1-dltt*massflux<Mgasequilibrium1
      gasmassflux = abs(Mgas1-Mgasequilibrium1); % is positive
    else
      gasmassflux = abs(dltt*massflux); % is positive
    end
else
  gasmassflux = dltt*massflux; % is positive
  Mgas1 = Mgas1-dltt*massflux;
  Mgas2 = Mgas2+dltt*massflux;
end

gasenthalpyflux = gasmassflux*specgasenthalpy(n,1);

else if (pgas(n,1)-pgas(n,2))/pgas(n,1)<-1e-12
  gasvelocity = sqrt(2*(pgas(n,2)-pgas(n,1))/((1+alpha1)*rhog(n,1))) ;
  massflux = rhog(n,1)*AP(2)+gasvelocity;
  if constraints==1||constraints==2
    if Mgas1+dltt*massflux<Mgasequilibrium1
      gasmassflux = -abs(dltt*massflux); % is negative
else
gasmassflux = -\text{abs}(\text{Mgas1} - \text{Mgasequilibrium1}); \text{ is negative}
end

\text{Mgas1} = \min(\text{Mgas1} + \text{dltt} \cdot \text{gasmassflux}, \text{Mgasequilibrium1});
\text{Mgas2} = \max(\text{Mgas2} - \text{dltt} \cdot \text{gasmassflux}, \text{Mgasequilibrium2});
else
gasmassflux = -\text{abs}(\text{dltt} \cdot \text{massflux}); \text{ is negative}
Mgas1 = \text{Mgas1} + \text{dltt} \cdot \text{gasmassflux};
Mgas2 = \text{Mgas2} - \text{dltt} \cdot \text{gasmassflux};
end
else
gasmassflux = \text{abs}(\text{Mgas1} - \text{Mgasequilibrium1}); \text{ is positive}
gasenthalpyflux = \text{gasmassflux} \cdot \text{specgasenthalpy}(n,2);
else
\text{gasenthalpyflux} = 0;
gasenthalpyflux = 0;
end
end
\text{totaltankmass}(n+1,1) = \text{totaltankmass}(n,1) - \text{gasmassflux} - \text{liqmassflux};
\text{totaltankmass}(n+1,2) = \text{totaltankmass}(n,2) + \text{gasmassflux} + \text{liqmassflux};
\text{totaltankenthalpy}(n+1,1) = \text{totaltankenthalpy}(n,1) - \text{liqenthalpyflux} - \text{gasenthalpyflux};
\text{totaltankenthalpy}(n+1,2) = \text{totaltankenthalpy}(n,2) + \text{liqenthalpyflux} + \text{gasenthalpyflux};
% quality update
\text{quality}(n+1,1) = \text{refpropm}('Q', 'D', \text{totaltankmass}(n+1,1) / (\text{AT}(1) \cdot \text{HT}(1)), 'H', ...
\text{totaltankenthalpy}(n+1,1) / \text{totaltankmass}(n+1,1), 'methane');
\text{quality}(n+1,2) = \text{refpropm}('Q', 'D', \text{totaltankmass}(n+1,2) / (\text{AT}(2) \cdot \text{HT}(2)), 'H', ...
\text{totaltankenthalpy}(n+1,2) / \text{totaltankmass}(n+1,2), 'methane');
% temperature per tank
\text{temperature}(n+1,1) = \text{refpropm}('T', 'D', \text{temperature}(n+1,1), 'Q', 0, 'methane');
\text{temperature}(n+1,2) = \text{refpropm}('T', 'D', \text{temperature}(n+1,2), 'Q', 0, 'methane');
% liquid density per tank
\text{rholiq}(n+1,1) = \text{refpropm}('D', 'T', \text{temperature}(n+1,1), 'Q', 0, 'methane');
\text{rholiq}(n+1,2) = \text{refpropm}('D', 'T', \text{temperature}(n+1,2), 'Q', 0, 'methane');
% liquid height per tank
\text{H}(n+1,1) = ((1 - \text{quality}(n+1,1)) \cdot \text{totaltankmass}(n+1,1) / \text{rholiq}(n+1,1)) / (\text{AT}(1));
\text{H}(n+1,2) = ((1 - \text{quality}(n+1,2)) \cdot \text{totaltankmass}(n+1,2) / \text{rholiq}(n+1,2)) / (\text{AT}(2));
% specific liquid enthalpy per tank
\text{specliqenthalpy}(n+1,1) = \text{refpropm}('H', 'T', \text{temperature}(n+1,1), 'D', \text{rholiq}(n+1,1), 'methane');
\text{specliqenthalpy}(n+1,2) = \text{refpropm}('H', 'T', \text{temperature}(n+1,2), 'D', \text{rholiq}(n+1,2), 'methane');
% gas density per tank
\text{rhog}(n+1,1) = (\text{quality}(n+1,1) \cdot \text{totaltankmass}(n+1,1)) / (\text{AT}(1) \cdot (\text{HT}(1) - \text{H}(n+1,1)));
\text{rhog}(n+1,2) = (\text{quality}(n+1,2) \cdot \text{totaltankmass}(n+1,2)) / (\text{AT}(2) \cdot (\text{HT}(2) - \text{H}(n+1,2)));
specgasenthalpy(n+1,1) = refpropm('H', 'T', temperature(n+1,1), 'D', rhog(n+1,1), 'methane...');
specgasenthalpy(n+1,2) = refpropm('H', 'T', temperature(n+1,2), 'D', rhog(n+1,2), 'methane...');

% gas pressure per tank
pgas(n+1,1) = 1000 * refpropm('P', 'T', temperature(n+1,1), 'D', rhog(n+1,1), 'methane');
pgas(n+1,2) = 1000 * refpropm('P', 'T', temperature(n+1,2), 'D', rhog(n+1,2), 'methane');

% pressure at end of lower pipe
pin(n+1) = pgas(n+1,1) + rholiq(n+1,1) * g * (max(H(n+1,1) - HL(1), 0));
pout(n+1) = pgas(n+1,2) + rholiq(n+1,2) * g * (max(H(n+1,2) - HL(1), 0));

end
toc

% plot solution
figure(1);
plot([0:dltt:Tmax], H(:,1), [0:dltt:Tmax], H(:,2), 'red', 'LineWidth', 1.8);
grid;
xlabel('t(s)');
ylabel('H(m)');
h = legend('H_1(t)', 'H_2(t)', 4);

figure(2);
plot([0:dltt:Tmax], pin, [0:dltt:Tmax], pout, 'red', 'LineWidth', 1.8);
grid;
xlabel('t(s)');
ylabel('p(Pa)');
h = legend('p_{in}(t)', 'p_{out}(t)', 'location', 'northeast');

figure(3);
plot([0:dltt:Tmax], pgas(:,1), [0:dltt:Tmax], pgas(:,2), 'red', 'LineWidth', 1.8);
grid;
xlabel('t(s)');
ylabel('p(Pa)');
h = legend('p_{gas1}(t)', 'p_{gas2}(t)', 'location', 'northEast');

figure(4);
plot([0:dltt:Tmax], temperature(:,1), [0:dltt:Tmax], temperature(:,2), 'red', 'LineWidth', 1.8);
grid;
xlabel('t(s)');
ylabel('T(K)');
h = legend('T_1(t)', 'T_2(t)', 'location', 'northEast');

figure(5);
plot([0:dltt:Tmax], totaltankmass(:,1), [0:dltt:Tmax], totaltankmass(:,2), 'red', 'LineWidth', 1.8);
grid;
xlabel('t(s)');
ylabel('m(kg)');
h = legend('m_1(t)', 'm_2(t)', 'location', 'SouthEast');

figure(5);
plot([0:dltt:Tmax], quality(:,1), [0:dltt:Tmax], quality(:,2), 'red', 'LineWidth', 1.8);
grid;
xlabel('t(s)');
ylabel('\xi(kg)');
h = legend('\xi_1(t)', '\xi_2(t)', 'location', 'SouthEast');

values20 = zeros(8,1);
values20(1) = H(20/delta+1,1);
values20(2) = H(20/delta+1,2);
values20(3) = pin(20/delta+1);
values20(4) = pout(20/delta+1);
values20(5) = pgas(20/delta+1,1);
values20(6) = pgas(20/delta+1,2);
values20(7) = temperature(20/delta+1,1);
values20(8) = temperature(20/delta+1,2);

% lprintf(1,'ch70202112125H01','figcopy','off','width',7.375);
% lprintf(2,'ch70202112125P01','figcopy','off','width',7.375);
% lprintf(3,'ch70202112125Pgas01','figcopy','off','width',7.375);
% lprintf(4,'ch70202112125T01','figcopy','off','width',7.375);
% lprintf(5,'ch70202112125quality01','figcopy','off','width',7.375);
C.4.2 Runge-Kutta and NDF for Two-Tank System with Upper Pipe with Variable Temperature

function [values]=lngtanksrun(reltol)

% Two-tank model with gas flow in upper pipe and liquid flow in lower pipe,
% solves set of differential algebraic equations (DAEs) by ode15s or ode23t
% Enthalpy equations are based on equilibrium, no z-discretisation, i.e.
% specific gas and liquid enthalpy are constant in space per tank
% 'methane','ethane','nitrogen',[ .943648747220886 5.03466958769393e-02 6.0045569021752e...-03]

tic
% Geometric and physical parameters
g = 9.81; % gravitation constant
HL = [1 1]; % lower pipe connection
HT = [20 10]; % tank heights
DT = [5 2]; % tank diameters
AT = pi.*((DT./2).ˆ2); % cross-sectional area tanks

% a constant singular mass matrix
M = zeros(22,22);
M(1,1) = 1;
M(2,2) = 1;
M(3,3) = 1;
M(4,4) = 1;

% Initial conditions
y0 = zeros(22,1);
values = zeros(22,1);
tstart = 0;
tfinal = 200;
% yp0 = zeros(22,1);

% quality
y0(21) = 0.01;
y0(22) = 0.02;
% temperature
y0(13) = 111;
y0(14) = 111;
% mass per tank
y0(1) = 1e5;
y0(2) = 1.2e4;
y0(1) = AT(1)*HT(1)*refpropm('D','T',y0(13),'Q',y0(21),'methane');
y0(2) = AT(2)*HT(2)*refpropm('D','T',y0(14),'Q',y0(22),'methane');

% total enthalpy per tank
% y0(1) = 1e8;
% y0(2) = 4e3;
y0(3) = y0(1)*refpropm('H','T',y0(13),'Q',y0(21),'methane');
y0(4) = y0(2)*refpropm('H','T',y0(14),'Q',y0(22),'methane');

% quality per tank
% y0(21) = refpropm('Q','D',y0(1)/(AT(1)*HT(1)),'H',y0(3)/y0(1),'methane');
% y0(22) = refpropm('Q','D',y0(2)/(AT(2)*HT(2)),'H',y0(4)/y0(2),'methane');

% temperature per tank
% y0(13) = refpropm('T','D',y0(1)/(AT(1)*HT(1)),'Q',y0(21),'methane');

end
C.4 Two-Tank System with Variable Temperature 155

\[ y(14) = \text{refpropm('T', 'D', y0(2)/(AT(2) * HT(2)), 'Q', y0(22), 'methane');} \]

\[ y0(9) = \text{refpropm('D', 'T', y0(13), 'D', y0(9), 'methane');} \]

\[ y0(10) = \text{refpropm('D', 'T', y0(14), 'Q', 0, 'methane');} \]

\[ y0(5) = \text{refpropm('H', 'T', y0(13), 'D', y0(9), 'methane');} \]

\[ y0(6) = \text{refpropm('H', 'T', y0(14), 'D', y0(10), 'methane');} \]

\[ y0(19) = (((1-y0(21)) * y0(1))/y0(9))/AT(1); \]

\[ y0(20) = (((1-y0(22)) * y0(2))/y0(10))/AT(2); \]

\[ y0(11) = (y0(21)*y0(11))/(AT(1) * (HT(1)-y0(19))); \]

\[ y0(12) = (y0(22)*y0(21))/(AT(2) * (HT(2)-y0(20))); \]

\[ y0(7) = \text{refpropm('H', 'T', y0(13), 'D', y0(11), 'methane');} \]

\[ y0(8) = \text{refpropm('H', 'T', y0(14), 'D', y0(12), 'methane');} \]

\[ y0(15) = 1000 * \text{refpropm('P', 'T', y0(13), 'D', y0(11), 'methane');} \]

\[ y0(16) = 1000 * \text{refpropm('P', 'T', y0(14), 'D', y0(12), 'methane');} \]

\[ y0(17) = y0(15) + y0(9) * g * \max(y0(19)-HL(1),0)); \]

\[ y0(18) = y0(16) + y0(10) * g * \max(y0(20)-HL(1),0)); \]

\[ tout = tstart; \]
\[ yout = y0.; \]

\[ teout = []; \]
\[ yeout = []; \]
\[ ieout = []; \]

\[ if \text{ getvalues==1}\]
\[ \text{options = odeset('Mass',M,'RelTol',reltol);} \]
\[ [t,y] = \text{ode23t(@lngtanks,[tstart 20],y0,options);} \]

\[ % Accumulate output. This could be passed out as output arguments. \]
\[ nt = \text{length(t);} \]
\[ tout = [tout; t(2:int)]; \]
\[ yout = [yout; y(2:int,:)]; \]
\[ y0(:) = y(nt,:); \]
\[ values = y0(:); \]

\[ \text{options = odeset('Events',@events4,'Mass',M,'RelTol',reltol);} \]
\[ [t,y,te,ye,ie] = \text{ode23t(@lngtanks,[20 tfinal],y0,options);} \]

\[ % Accumulate output. This could be passed out as output arguments. \]
\[ nt = \text{length(t);} \]
\[ tout = [tout; t(2:int)]; \]
\[ yout = [yout; y(2:int,:)]; \]
\[ teout = [teout; te]; \]
\[ yeout = [yeout; ye]; \]
\[ ieout = [ieout; ie]; \]

\[ else \]
\[ \text{options = odeset('Events',@events4,'Mass',M,'RelTol',reltol);} \]
\[ [t,y,te,ye,ie] = \text{ode15s(@lngtanks,[tstart tfinal],y0,options);} \]

\[ % Accumulate output. This could be passed out as output arguments. \]
\[ nt = \text{length(t);} \]
\[ tout = [tout; t(2:int)]; \]

\[ end \]

\[ getvalues=0; \]
yout = [yout; y(2:nt,:)]; yeout = [yeout; ye]; iout = [iout; iout];
end

if t(nt)<tfinal
    fprintf('event');
    t(nt) = y(nt,:); tstart = t(nt);
    options = odeset('RelTol',1e-8);
    [t,y] = ode45 @(ntanksconstant,tstart tfinal,y0,options);
    nt = length(t);
    tout = [tout; t(nt)];
yout = [yout; y(2:nt,:)];
end
toc

figure(1);
plot( tout,yout(:,19),tout,yout(:,20),'--red','LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$H$(m)');

figure(2);
plot( tout,yout(:,17),tout,yout(:,18),'--red','LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$p$(Pa)');

figure(3);
plot( tout,yout(:,15),tout,yout(:,16),'--red','LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$p$(Pa)');

figure(4);
plot( tout,yout(:,13),tout,yout(:,14),'--red','LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$T$(K)');

figure(5);
plot( tout,yout(:,17),tout,yout(:,18),tout,yout(:,15),tout,yout(:,16),'--red','LineWidth',1.8 );
grid;
xlabel( '$t$(s)');
ylabel( '$p$(Pa)');

figure(6);
C.4 Two-Tank System with Variable Temperature

```matlab
plot( tout,yout(:,21),tout,yout(:,22),’--red’,’LineWidth’,1.8 );
grid;
xlabel( ’$t$(s)’);
ylabel( ’$\xi$(Pa)’);
h = legend(’$\xi_{1}(t)$’,’$\xi_2(t)$’,’location’,’northeast’);

laprint(1,’Q12T110110H’,’width’,7.375,’figcopy’,’off’);
laprint(2,’Q12T110110P’,’width’,7.375,’figcopy’,’off’);
laprint(3,’Q12T110110P’,’width’,7.375,’figcopy’,’off’);
laprint(4,’Q12T110110T’,’width’,7.375,’figcopy’,’off’);
laprint(5,’Q12T110110P1’,’width’,7.375,’figcopy’,’off’);
laprint(6,’Q12T110110Q’,’width’,7.375,’figcopy’,’off’)

%-------------------------------------------------- -----------------------%
function dy = lngtanksconstant(t,y)
dy = zeros(22,1);
%-------------------------------------------------- -----------------------%
function dy = lngtanks(t,y)
% Geometric and physical parameters
g = 9.81; % gravitation constant
HL = [1 1]; % lower pipe connection
HT = [20 10]; % tank heights
DP = [0.2 0.2]; % assume Dp<<Dt
DT = [5 2]; % tank diameters
alpha = 2; % friction factor for liquid flow
AP = pi.)*(DP./2).^2; % in pipe, alpha=0: no friction
AT = pi.)*(DT./2).^2; % cross-sectional area tanks
LP = [10 10]; % length of pipe

specificgasconstant = 8.314472/(16.042e-3);

% specific gas constant methane=gas constant/molar mass (kg/mol)
dy = zeros(22,1);

% quality per tank
dy(21) = refpropm(’Q’,’D’,y(1)/(AT(1)*HT(1)),’H’,y(3)/y(1),’methane’) - y(21);
dy(22) = refpropm(’Q’,’D’,y(2)/(AT(2)*HT(2)),’H’,y(4)/y(2),’methane’) - y(22);

% temperature per tank
dy(13) = refpropm(’T’,’D’,y(1)/(AT(1)*HT(1)),’Q’,y(21),’methane’) - y(13);
dy(14) = refpropm(’T’,’D’,y(2)/(AT(2)*HT(2)),’Q’,y(22),’methane’) - y(14);

% liquid density per tank
dy(9) = refpropm(’D’,’T’,y(13),’Q’,0,’methane’) - y(9);
dy(10) = refpropm(’D’,’T’,y(14),’Q’,0,’methane’) - y(10);

% liquid height per tank
dy(19) = ((1-y(21)) * y(1)/y(9))/(AT(1)) - y(19);
dy(20) = ((1-y(22)) * y(2)/y(10))/(AT(2)) - y(20);

% specific liquid enthalpy per tank
dy(5) = refpropm(’H’,’T’,y(13),’D’,y(9),’methane’) - y(5);
dy(6) = refpropm(’H’,’T’,y(14),’D’,y(10),’methane’) - y(6);

% gas density per tank
dy(11) = y(21) * y(1)/(AT(1)*(HT(1)-y(19))) - y(11);
dy(12) = y(22) * y(2)/(AT(2)*(HT(2)-y(20))) - y(12);
```
\[ \text{specific gas enthalpy per tank} \]
\[ dy(7) = \text{refpropm}(\text{'H'}, \text{'T'}, y(13), \text{'D'}, y(11), \text{'methane'}) - y(7); \]
\[ dy(8) = \text{refpropm}(\text{'H'}, \text{'T'}, y(14), \text{'D'}, y(12), \text{'methane'}) - y(8); \]

\[ \text{gas pressure per tank} \]
\[ dy(15) = 1000 \times \text{refpropm}(\text{'P'}, \text{'T'}, y(13), \text{'D'}, y(11), \text{'methane'}) - y(15); \]
\[ dy(16) = 1000 \times \text{refpropm}(\text{'P'}, \text{'T'}, y(14), \text{'D'}, y(12), \text{'methane'}) - y(16); \]

\[ \text{pressure at end of lower pipe} \]
\[ dy(17) = y(15) + y(9) \times g \times \max(y(19) - HL(1), 0)) - y(17); \]
\[ dy(18) = y(16) + y(10) \times g \times \max(y(20) - HL(1), 0)) - y(18); \]

\[ \text{eps}=1e-8; \]

\[ \text{mass and enthalpy change per tank} \]
\[ \text{if} (y(17) - y(18) > \text{eps} \&\& (y(19) - \max(\text{HL}(1), \text{HL}(2)) > \text{eps})) \]
\[ \text{dy}(1) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(10)); \]
\[ \text{dy}(3) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(7)); y(9)); \]
\[ \text{else} \ y(15) - y(16) < -\text{eps} \]
\[ \text{dy}(1) = -y(12) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) < -\text{eps} \]
\[ \text{dy}(1) = -y(12) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) > \text{eps} \]
\[ \text{dy}(1) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) < -\text{eps} \]
\[ \text{dy}(1) = -y(12) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) > \text{eps} \]
\[ \text{dy}(1) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8));
\[ \text{elseif} \ y(20) - \max(\text{HL}(1), \text{HL}(2)) > \text{eps} \]
\[ \text{dy}(1) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(10)); \]
\[ \text{dy}(3) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(7)); y(9)); \]
\[ \text{else} \ y(15) - y(16) < -\text{eps} \]
\[ \text{dy}(1) = -y(12) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) > \text{eps} \]
\[ \text{dy}(1) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) < -\text{eps} \]
\[ \text{dy}(1) = -y(12) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) > \text{eps} \]
\[ \text{dy}(1) = -y(11) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
\[ \text{else} \ y(15) - y(16) < -\text{eps} \]
\[ \text{dy}(1) = -y(12) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + y(10) \times \sqrt{(2/(1+alpha))) \times g \times \max(y(20), \text{HL}(2))) + \}
\[ (y(16) - y(15)) / y(9)); y(8)); \]
else
    dy(1) = 0;
    dy(3) = 0;
end
end
dy(2) = -dy(1);  % mass conservation
dy(4) = -dy(3);  % enthalpy conservation

function [value,isterminal,direction] = events4(t,y)
reltol=1e-6;
if ((abs(y(15)-y(16)))/y(15)<reltol)&&(abs(y(17)-y(18)))/y(17)<reltol)
    value(1) = 1;  % detect pressures are (almost) equal
    isterminal(1) = 1;  % stop the integration
    direction(1) = 1;
else
    value(1) = 1;
    isterminal(1) = 1;
    direction(1) = 1;
end

HT = [20 10];  %tank heights
% Locate the time when quality passes through zero in a decreasing direction
% and stop integration.
value(2) = y(19) - HT(1);  % detect H1=HT1
isterminal(2) = 1;  % stop the integration
direction(2) = 1;  % positive direction

value(3) = y(20) - HT(2);  % detect H2=HT2
isterminal(3) = 1;  % stop the integration
direction(3) = 1;  % positive direction
C.5 Multi-Component Fluids

C.5.1 Explicit Euler for One-Tank System with Multi-Component Fluid

```matlab
function [values]=multicomponetankeuler(timestep)

% One-tank model with heat flux from environment,
% solves set of differential algebraic equations (DAEs) by explicit Euler

% Enthalpy equations are based on equilibrium, no z-discretisation, i.e.
% specific gas and liquid enthalpy are constant in space per tank
tic

% Geometric and physical parameters
% gravitation constant
g = 9.81;
% tank height
HT = 30;
% tank diameter
DT = 20;
% cross-sectional area tank
AT = pi.*(DT/2).^2;
% surface area tank
ST = 2*pi.*(DT/2).*HT+2*pi.*(DT/2).^2;
% heat coefficient
K = 55;
% temperature
temperatureamb1 = 293;
% vapor fraction (kg/kg)
quality = .05;

fprintf('T = %g K
',temperature);
fprintf('q = %g
',quality);

superheated = 0;
supercritical = 0;
% molarmassmethane = 16.042e-3; %(kg/mol)
% molarmassethane = 30.07e-3; %(kg/mol)
% molarmassnitrogen = 14.0067e-3; %(kg/mol)
% specific gas constant methane=gas constant/molar mass (kg/mol)
% specificgasconstant = 8.314472/(molarmassmethane);

% grid
Tmax = 100*7*24*3600;
dltt = timestep*7*24*3600;
Nt = round(Tmax/dltt);

% parameters for theta method
theta = 0;
theta1 = 1-theta;

% Initial conditions
y = zeros(16,Nt+1);
values = zeros(16,1);
y(9,1) = quality;
y(6,1) = temperature;

% composition (both input and output are mass fractions!)
[x1 x2] = refpropm('X','T',y(6,1),'Q',y(9,1),'methane','....
'ethane','nitrogen',[
0.9 0.09 0.01]);
y(10,1) = x1(1);
y(11,1) = x1(2);
y(12,1) = x1(3);
y(13,1) = x2(1);
y(14,1) = x2(2);
y(15,1) = x2(3);
```

% mass
mass = AT(1) * HT(1) * refpropm('D', 'T', y(6,1), 'Q', y(9,1), 'methane', 'ethane', 'nitrogen', [ .9 ... 0.91]);
% mass = AT(1) * HT(1) * refpropm('D', 'T', y0(6), 'Q', y0(9), 'methane', ... 'ethane', 'nitrogen', [ .93468474220886 5.03466958769393e-02 6.0045569021752e-03]);

% enthalpy
y(1,1) = mass * refpropm('H', 'T', y(6,1), 'Q', y(9,1), 'methane', 'ethane', 'nitrogen', [ .943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

% internal energy
y(16,1) = mass * refpropm('U', 'T', y(6,1), 'Q', y(9,1), 'methane', 'ethane', 'nitrogen', [ .943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

% liquid density
y(4,1) = refpropm('D', 'T', y(6,1), 'Q', 0, 'methane', 'ethane', 'nitrogen', [y(10,1) y(11,1) y(12,1)]);

% y0(4) = refpropm('D', 'T', y0(6), 'Q', 0, 'methane', 'ethane', ... 'nitrogen', [ .943684747220886 5.03466958769393e-02 6.0045569021752e-03]);

% specific liquid enthalpy
y(2,1) = refpropm('H', 'T', y(6,1), 'D', y(4,1), 'methane', 'ethane', 'nitrogen', [y(10,1) y(11,1) y(12,1)]);

% y(2,1) = refpropm('H', 'T', y0(6), 'D', y0(4), 'methane', 'ethane', ... 'nitrogen', [ .943684747220886 5.03466958769393e-02 6.0045569021752e-03]);

% liquid height
y(8,1) = ((1-y(9,1)) * mass)/y(4,1))/AT;

% gas density
y(5,1) = (y(9,1) * mass)/(AT * (HT-y(8,1)));

% specific gas enthalpy
y(3,1) = refpropm('H', 'T', y(6,1), 'D', y(5,1), 'methane', 'ethane', ... 'nitrogen', [ y(13,1) y(14,1) y(15,1)]);

% y0(3) = refpropm('H', 'T', y0(6), 'D', y0(5), 'methane', 'ethane', ... 'nitrogen', [ .943684747220886 5.03466958769393e-02 6.0045569021752e-03]);

% gas pressure
y(7,1) = 1000 * refpropm('P', 'T', y(6,1), 'D', y(5,1), 'methane', 'ethane', 'nitrogen', [y(13,1) y(14,1) y(15,1)]);

% y0(7) = 1000 * refpropm('P', 'T', y0(6), 'D', y0(5), 'methane', 'ethane', ... 'nitrogen', [ .943684747220886 5.03466958769393e-02 6.0045569021752e-03]);

for t=1:Nt
% enthalpy update
y(1,t+1) = y(1,t) - K * dltt * (y(6,t)-temperatureamb1);

% internal energy update
y(16,t+1) = y(16,t) - K * dltt * (y(6,t)-temperatureamb1);

% quality update
y(9,t+1) = refpropm('Q', 'D', mass/(AT*HT), 'U', y(16,t+1)/mass, 'methane', 'ethane', ... 'nitrogen', [ .9 .99 .01]);

if (y(9,t+1)==998)
if superheated<1
fprintf('%s
', (t-1) * dltt/(24*3600*7));
end
end

superheated = 1;
else if (y(9,t+1)==999)
  if supercritical<1
    fprintf('supercritical state (T > Tc) and (p > Pc')
    fprintf('%.4g\n', (t-1) * dltt/(24*3600*7));
  end
  supercritical = 1;
else if (y(9,t+1)<0)
  y(9,t+1)==0;
  fprintf('subcooled liquid\n');
  fprintf('%.4g\n', (t-1) * dltt/(24*3600*7));
else if (y(9,t+1)>1)
  y(9,t+1)=1;
end

% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q = 1 indicates saturated vapor
% q > 1 indicates superheated vapor
% q = 998 superheated vapor, but quality not defined (t > Tc)
% q = 999 indicates supercritical state (t > Tc) and (p > Pc)

% dy(9) = refpropm('Q','D',mass/(AT * HT),'H',y(1)/mass,'methane','ethane',...
% 'nitrogen', [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(9);

% temperature
y(6,t+1) = refpropm('T','D',mass/(AT(1) * HT(1)),'U',y(16,t+1)/mass,'methane','ethane',...
'nitrogen', [.9 .09 .01]);
% dy(6) = refpropm('T','D',mass/(AT(1) * HT(1)),'Q',y(9),'methane','ethane',...
% 'nitrogen', [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(6);

% composition (mass fractions!)
if (y(9,t+1)>1)
  y(10,t+1) = y(10,t);
  y(11,t+1) = y(11,t);
  y(12,t+1) = y(12,t);
  y(13,t+1) = .9;
  y(14,t+1) = .09;
  y(15,t+1) = .01;
else if (y(9,t+1)<=0)
  y(10,t+1) = .9;
  y(11,t+1) = .09;
  y(12,t+1) = .01;
  y(13,t+1) = y(13,t);
  y(14,t+1) = y(14,t);
  y(15,t+1) = y(15,t);
else
  [x1 x2] = refpropm('X','T',y(6,t+1),'Q',y(9,t+1),'methane','ethane','nitrogen',[...
  .9 .09 .01]);
  y(10,t+1) = x1(1);
  y(11,t+1) = x1(2);
  y(12,t+1) = x1(3);
  y(13,t+1) = x2(1);
  y(14,t+1) = x2(2);
  y(15,t+1) = x2(3);
end

% liquid density per tank
if (y(9,t+1)>1)
  y(4,t+1) = y(4,t);
else
C.5 Multi-Component Fluids

\[
y(4,t+1) = \text{refpropm}('D', 'T', y(6,t+1), 'Q', 0, \text{methane}, \text{ethane}, \text{nitrogen}, [y(10,t+1), y(11,t+1), y(12,t+1)]);
\]
end

dy(4) = \text{refpropm}('D', 'T', y(6), 'Q', 0, \text{methane}, \text{ethane}, \text{nitrogen},...
\[
\{( .943648747220886 5.03466958769393e-02 6.0045569021752e-03)} - y(4);
\]

liquid height per tank
\[
y(8,t+1) = \max(0, ((1-y(9,t+1)) * \text{mass}/y(4,t+1))/AT);
\]
if (y(9,t+1)>1)
\[
y(8,t+1)=0;
\]
end

gas density per tank
\[
\text{if} (y(9,t+1)>1)
y(5,t+1) = y(5,t);
\text{else}
y(5,t+1) = \max(0, (y(9,t+1) * \text{mass})/(AT(1) * (HT(1)-y(8,t+1))));
\]

gas pressure per tank
\[
y(7,t+1) = 1000 * \text{refpropm}('P', 'T', y(6,t+1), 'Q', 1, \text{methane}, \text{ethane}, \text{nitrogen},[y(13,t+1), y(14,t+1), y(15,t+1)]);
\]
\[
y(7,t+1) = 1000 * \text{refpropm}('P', 'T', y(6,t+1), 'D', y(5,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(13,t+1), y(14,t+1), y(15,t+1)]);
\]
\[
\text{if} (y(9,t+1)>1)
y(7,t+1) = 1000 * \text{refpropm}('P', 'T', y(6,t+1), 'D', y(5,t+1), \text{methane},
\text{nitrogen},[\text{gasmethanemoles}/\text{sumgas gasethanemoles}/\text{sumgas gasnitrogenmoles}/\text{sumgas}]);
\]
\[
\text{if} (y(9,t+1)>1)
y(7,t+1) = \text{refpropm}('P', 'T', y(6), 'D', y(5), \text{methane}, \text{ethane}, \text{nitrogen},
\text{[(y(9,t+1)==999)\text{refpropm}('H', 'T', y(6,t+1), 'D', y(5,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(10,t+1), y(11,t+1), y(12,t+1)]);\text{else if} (y(9,t+1)==998)y(2,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'Q', 0, \text{methane}, \text{ethane}, \text{nitrogen},[y(10,t+1), y(11,t+1), y(12,t+1)]);\text{else}y(2,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'D', y(4,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(10,t+1), y(11,t+1), y(12,t+1)]);\end{\text{if}}\)
\]

specific liquid enthalpy per tank
\[
\text{if} (y(9,t+1)==999)y(2,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'D', y(5,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(10,t+1), y(11,t+1), y(12,t+1)]);\text{else if} (y(9,t+1)==998)y(2,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'Q', 0, \text{methane}, \text{ethane}, \text{nitrogen},[y(10,t+1), y(11,t+1), y(12,t+1)]);\text{else}y(2,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'D', y(4,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(10,t+1), y(11,t+1), y(12,t+1)]);\end{\text{if}}\)
\]

specific gas enthalpy per tank
\[
\text{if} (y(9,t+1)==999)y(3,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'Q', 1, \text{methane}, \text{ethane}, \text{nitrogen},[y(13,t+1), y(14,t+1), y(15,t+1)]);\text{else if} (y(9,t+1)==998)y(3,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'D', y(5,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(13,t+1), y(14,t+1), y(15,t+1)]);\text{else}y(3,t+1) = \text{refpropm}('H', 'T', y(6,t+1), 'D', y(5,t+1), \text{methane}, \text{ethane}, \text{nitrogen},[y(13,t+1), y(14,t+1), y(15,t+1)]);\end{\text{if}}\)
\]

end

T=[0:dltt:Tmax];
values(:) = y(:,Nt+1);
figure(1);
plot(T/(3600*24*7),y(8,:),'LineWidth',1.8);
grid;
xlabel('t$(weeks)$');
ylabel('\$H(m)$');

figure(2);
plot( T/(3600*24*7),y(7,:), 'LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$H(t)$');

h = legend('\$H_{liq}(t)$','\$H_{gas}(t)$','location','southeast');

figure(3);
plot( T/(3600*24*7),y(2,:),T/(3600*24*7),y(3,:),'--r','LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$p$(Pa)');

h = legend('\$p_{gas}(t)$','location','southeast');

figure(4);
plot( T/(3600*24*7),y(6,:), 'LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$h$(J)');

h = legend('\$h_{liq}(t)$','\$h_{gas}(t)$','location','southeast');

figure(5);
plot( T/(3600*24*7),y(1,:), 'LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$T$(K)');

h = legend('\$T(t)$','location','southeast');

figure(6);
plot( T/(3600*24*7),AT.*y(8,:).*y(4,:), 'LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$m_{liq}(t)$');

h = legend('\$m_{liq}(t)$','location','southeast');

figure(7);
plot( T/(3600*24*7),y(9,:), 'LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$\xi$(kg/kg)');

h = legend('\$\xi(t)$','location','southeast');

figure(8);
plot( T/(3600*24*7),y(10,:),T/(3600*24*7),y(13,:),'--r','LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$X$(kg/kg)');

h = legend('\$X_{liq}^{CH_4}(t)$','\$X_{gas}^{CH_4}(t)$','location','east');

figure(9);
plot( T/(3600*24*7),y(11,:),T/(3600*24*7),y(14,:),'--r','LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$X$(kg/kg)');

h = legend('\$X_{liq}^{C_2H_6}(t)$','\$X_{gas}^{C_2H_6}(t)$','location','east');

figure(10);
plot( T/(3600*24*7),y(12,:),T/(3600*24*7),y(15,:),'--r','LineWidth',1.8);
grid;
xlabel('\$t$(weeks)');
ylabel('\$X$(kg/kg)');

h = legend('\$X_{liq}^{N_2}(t)$','\$X_{gas}^{N_2}(t)$','location','northeast');
C.5 Multi-Component Fluids

lprint(1,'multicomponentankelert111q005W','width',7.375,'figcopy','off');
lprint(2,'multicomponentankelert111q005pgas','width',7.375,'figcopy','off');
lprint(3,'multicomponentankelert111q005enthalpy','width',7.375,'figcopy','off');
lprint(4,'multicomponentankelert111q005T','width',7.375,'figcopy','off');
lprint(5,'multicomponentankelert111q005enthalpytot','width',7.375,'figcopy','off');
lprint(6,'multicomponentankelert111q005mliq','width',7.375,'figcopy','off');
% lprint(7,'multicomponentankelert111q005quality','width',7.375,'figcopy','off');
lprint(8,'multicomponentankelert111q005methane','width',7.375,'figcopy','off');
lprint(9,'multicomponentankelert111q005ethane','width',7.375,'figcopy','off');
lprint(10,'multicomponentankelert111q005nitrogen','width',7.375,'figcopy','off');
C.5.2 Runge-Kutta and NDF for One-Tank System with Multi-Component Fluid

```matlab
function [values]=multicomponenttankrk1(rtol)

% One-tank model with heat flux from environment,
% solves set of differential algebraic equations (DAEs) by ode15s or ode23t
% Enthalpy equations are based on equilibrium, no z-discretisation, i.e.
% specific gas and liquid enthalpy are constant in space per tank
% Using internal energy

% LNG = 'methane','ethane','nitrogen', [ .943648747220886 5.03466958769393e-02 ...
% 6.0045569021752e-03]
tic

% Geometric and physical parameters
g = 9.81; %gravitation constant
HT = 30; %tank height
DT = 20; %tank diameter
AT = pi.*((DT/2).^2); %cross-sectional area tank

% a constant singular mass matrix
M = zeros(18,18);
M(1,1) = 1; %enthalpy
M(10,10) = 1; %initial quality
M(11,11) = 1; %initial temperature
M(12,12) = 1; %internal energy

% Initial conditions
y0 = zeros(18,1);
values = zeros(18,1);
y0(10) = .05; % initial quality=vapor fraction kg/kg
y0(11) = 111; % initial temperature
y0(9) = y0(10); % quality
y0(6) = y0(11); % temperature

% compositions (mass fractions!)
[x1 x2] = refpropm('X','T',y0(6), 'Q', y0(9), 'methane','ethane','nitrogen',[.9 .09 .01]);
y0(13) = x1(1);
y0(14) = x1(2);
y0(15) = x1(3);
y0(16) = x2(1);
y0(17) = x2(2);
y0(18) = x2(3);

% mass
mass = AT*HT*refpropm('D','T',y0(6), 'Q', y0(9), 'methane','ethane','nitrogen',[.9 .09 .01])...

% enthalpy
y0(1) = mass*refpropm('H','T',y0(6), 'Q', y0(9), 'methane','ethane','nitrogen',[.9 .09 .01])...

% internal energy
y0(12) = mass*refpropm('U','T',y0(6), 'Q', y0(9), 'methane','ethane','nitrogen',[.9 .09 ... .01]);

% liquid density
```

166 Program Code
\begin{verbatim}
C.5 Multi-Component Fluids

y0(4) = refpropm('D', 'T', y0(6), 'Q', 0, 'methane', 'ethane', 'nitrogen', [y0(13) y0(14) y0(15) ...]);

% specific liquid enthalpy
y0(2) = refpropm('H', 'T', y0(6), 'D', y0(4), 'methane');

% liquid height
y0(8) = (((1-y0(9)) * mass) / y0(4)) / AT;

% gas density
y0(5) = (y0(9) * mass) / (AT * (HT-y0(8)));

% specific gas enthalpy
y0(3) = refpropm('H', 'T', y0(6), 'D', y0(5), 'methane');

% gas pressure
y0(7) = 1000 * refpropm('P', 'T', y0(6), 'D', y0(5), 'methane', 'ethane', 'nitrogen', [y0(16) y0(17) y0(18) ...]);

if events==1
  options = odeset('Events', @events1, 'Mass', M, 'RelTol', rtol, 'MaxOrder', 5);
  [t, y, te, ye, ie] = ode23t(@onetanktwophase, [tstart tfinal], y0, options);
  % Solve until the first terminal event.
  % Accumulate output. This could be passed out as output arguments.
  nt = length(t);
  tout = [t; t(2:nt)];
  yout = [y; y(2:nt,:)];
  teout = [te; te]
  yeout = [ye; ye]
  ieout = [ie; ie]
  % while (((abs(y(nt,9)-1)>1e-3)&&(abs(y(nt,9))>1e-3))&&(t(nt)<tfinal))
  % fprintf('run loop');
  % tstart = t(nt);
  % y0(:) = y(nt,:);
  % options = odeset('Events', @events1, 'Mass', M, 'RelTol', 1e-8);
  % Solve until the first terminal event.
  % Accumulate output. This could be passed out as output arguments.
  % nt = length(t);
  % tout = [t; t(2:nt)];
  % yout = [y; y(2:nt,:)]
  % teout = [te; te]
  % yeout = [ye; ye]
  % ieout = [ie; ie]
  end
end
if (abs(y(nt,9)-1)<1e-2)&&(y(nt,9)<900)&&(t(nt)<tfinal) % if quality has reached 1
  fprintf('quality 1');
end
\end{verbatim}
% a constant singular mass matrix
M2 = zeros(12,12);

M2(1,1) = 1; % enthalpy
M2(2,2) = 1; % specific liquid enthalpy
M2(4,4) = 1; % liquid density
M2(8,8) = 1; % liquid height
M2(10,10) = 1; % initial quality
M2(11,11) = 1; % initial temperature
M2(12,12) = 1; % internal energy

M2(13,13) = 1; % compositions
M2(14,14) = 1; % compositions
M2(15,15) = 1; % compositions
M2(16,16) = 1; % compositions
M2(17,17) = 1; % compositions
M2(18,18) = 1; % compositions

% Set the new initial conditions, with quality 1
y0(10) = y(nt,10);
y0(11) = y(nt,11);
y0(9) = 1;
y0(6) = y(nt,6);

% compositions (mass fractions!)
[x1 x2] = refpropm('X','T',y0(6),'Q',y0(9),'methane','ethane','nitrogen',[.9 .09 ... .01]);

y0(13) = x1(1);
y0(14) = x1(2);
y0(15) = x1(3);
y0(16) = x2(1);
y0(17) = x2(2);
y0(18) = x2(3);

% mass
mass = AT(1) * HT(1) * refpropm('D','T',y0(11),'Q',y0(10),'methane','ethane','nitrogen',[.9 .09 .01]);

% enthalpy
y0(1) = mass * refpropm('H','T',y0(6),'D',y0(4),'methane','ethane','nitrogen',[.9 ... .09 .01]);

% internal energy
y0(12) = mass * refpropm('U','T',y0(6),'D',y0(5),'methane','ethane','nitrogen',[.9 ... .09 .01]);

% liquid density
y0(4) = refpropm('D','T',y0(6),'Q',0,'methane','ethane','nitrogen',[y0(13) y0(14) ...
y0(15)]);

% specific liquid enthalpy
y0(2) = refpropm('H','T',y0(6),'D',y0(4),'methane')
y0(2) = refpropm('H','T',y0(6),'Q',0,'methane','ethane','nitrogen',[y0(13) y0(14) ...
y0(15)]);

% liquid height
y0(8) = 0;

% gas density
y0(5) = mass/(AT*HT);

% specific gas enthalpy
y0(3) = refpropm('H','T',y0(6),'D',y0(5),'methane')
C.5 Multi-Component Fluids

\[ y_0(3) = \text{refpropm}('H', 'T', y_0(6), 'Q', 1, 'methane', 'ethane', 'nitrogen', [y_0(16) \ldots y_0(18)]); \]

\% gas pressure
\[ y_0(7) = 1000 \times \text{refpropm}('P', 'T', y_0(6), 'Q', 1, 'methane', 'ethane', 'nitrogen', [y_0(16) \ldots y_0(18)]); \]

\% A good guess of a valid first timestep is the length of the last valid
\% timestep, so use it for faster computation. ‘refine’ is 4 by default.
\[ t_{\text{start}} = t(n_1); \]

\[ \text{options}_3 = \text{odeset}('Events', @events2, 'Mass', M2, 'RelTol', rtol', 'MaxOrder', 5); \]

\% Solve until the first terminal event.
\[ [t, y, te, ye, ie] = \text{ode23t}(@onetankonephase1, [t_{\text{start}} t_{\text{final}}], y_0, \text{options}_3); \]

\% \text{options}_3 = \text{odeset}('Mass', M2, 'RelTol', 1e-2');
\[ [t, y] = \text{ode15s}(@onetankonephase1, [t_{\text{start}} t_{\text{final}}], y_0, \text{options}_3); \]

\[ nt = \text{length}(t); \]
\[ \text{tout} = [\text{tout}; t(2:nt)]; \]
\[ \text{yout} = [\text{yout}; y(2:nt,:)]; \]

\[ t_{\text{eout}} = [t_{\text{eout}}; te]; \]
\[ y_{\text{eout}} = [y_{\text{eout}}; ye]; \]
\[ i_{\text{eout}} = [i_{\text{eout}}; ie]; \]

\% Events at t\text{start} are never reported.
\% quality has reached 0
\[ \text{fprintf}('\text{quality} 0'); \]
\% a constant singular mass matrix
\[ M2 = \text{zeros}(11, 11); \]
\% \text{mass}
\[ M2(1, 1) = 1; \]
\% \text{specific gas enthalpy}
\[ M2(3, 3) = 1; \]
\% \text{gas density}
\[ M2(5, 5) = 1; \]
\% \text{gas pressure}
\[ M2(7, 7) = 1; \]
\% \text{liquid height}
\[ M2(8, 8) = 1; \]
\% \text{quality}
\[ M2(9, 9) = 1; \]
\% \text{initial quality}
\[ M2(10, 10) = 1; \]
\% \text{initial temperature}
\[ M2(11, 11) = 1; \]
\% \text{internal energy}
\[ M2(12, 12) = 1; \]
\% \text{compositions}
\[ M2(13, 13) = 1; \]
\% \text{compositions}
\[ M2(14, 14) = 1; \]
\% \text{compositions}
\[ M2(15, 15) = 1; \]
\% \text{compositions}
\[ M2(16, 16) = 1; \]
\% \text{compositions}
\[ M2(17, 17) = 1; \]
\% \text{compositions}
\[ M2(18, 18) = 1; \]
\% \text{compositions}

\% Set the new initial conditions, with quality 0
\[ y_0(10) = y(n_1, 10); \]
\[ y_0(11) = y(n_1, 11); \]
\[ y_0(9) = 0; \]
\[ y_0(6) = y(n_1, 6); \]

\% \text{compositions (mass fractions!)}
\[ [x_1 \ldots x_2] = \text{refpropm}('X', 'T', y_0(6), 'Q', y_0(9), 'methane', 'ethane', 'nitrogen', [.9 .09 ... .01]); \]

\[ y_0(13) = x_1(1); \]
\[ y_0(14) = x_1(2); \]
\[ y_0(15) = x_1(3); \]
\[ y_0(16) = x_2(1); \]
\[ y_0(17) = x_2(2); \]
\[ y_0(18) = x_2(3); \]

\% \text{mass}
mass = AT(1) * HT(1) * refpropm('D', 'T', y(nt, 11), 'Q', y(nt, 10), 'methane', 'ethane', ...
    'nitrogen', [.9 .09 .01]);

% enthalpy
% y0(1) = mass * refpropm('H', 'T', y0(6), 'Q', y0(9), 'methane');
y0(1) = y(nt, 1);

% internal energy
% y0(12) = mass * refpropm('U', 'T', y0(6), 'Q', y0(9), 'methane');
y0(12) = y(nt, 12);

% liquid density
y0(4) = mass/(AT * HT);

% specific liquid enthalpy
% y0(2) = refpropm('H', 'T', y0(6), 'D', y0(4), 'methane')
y0(2) = refpropm('H', 'T', y0(6), 'Q', 0, 'methane', 'ethane', 'nitrogen', [y0(13) y0(14) ... 
    y0(15)]);

% liquid height
y0(8) = HT;

% gas density
y0(5) = y(nt, 5);

% specific gas enthalpy
% y0(3) = refpropm('H', 'T', y0(6), 'D', y0(5), 'methane')
y0(3) = y(nt, 3);

% gas pressure
y0(7) = 1000 * refpropm('P', 'T', y0(6), 'D', y0(5), 'methane', 'ethane', 'nitrogen', [.9 ... 
    .09 .01]);

% A good guess of a valid first timestep is the length of the last valid
% timestep, so use it for faster computation. 'refine' is 4 by default.
tstart = t(nt);
options2 = odeset('Mass', M2, 'RelTol', rtol);
[t, y] = ode23t(@onetankonephase2, [tstart tfinal], y0, options2);
nt = length(t);
tout = [tout; t(2:nt)];
yout = [yout; y(2:nt,:)];
teout = [teout; te];
% Events at tstart are never reported.
% yeout = [yeout; ye];
ieout = [ieout; ie];

if (y(nt,9)>998) && (t(nt)<tfinal) % supercritical
    fprintf('quality 999');
    M2 = zeros(12, 12);
    M2(1,1) = 1; % enthalpy
    M2(2,2) = 1; % liquid enthalpy
    M2(4,4) = 1; % liquid density
    M2(8,8) = 1; % liquid height
    M2(10,10) = 1; % initial quality
    M2(11,11) = 1; % initial temperature
    M2(12,12) = 1; % internal energy
    M2(13,13) = 1; % compositions
    M2(14,14) = 1; % compositions
    M2(15,15) = 1; % compositions
    M2(16,16) = 1; % compositions
    M2(17,17) = 1; % compositions
    M2(18,18) = 1; % compositions
% Set the new initial conditions, with quality 999
y0(9) = 999;
y0(6) = y(nt,6);

% compositions (mass fractions!)
y0(13) = y(nt,13);
y0(14) = y(nt,14);
y0(15) = y(nt,15);
y0(16) = y(nt,16);
y0(17) = y(nt,17);
y0(18) = y(nt,18);

% mass
mass = AT(1) * HT(1) * refpropm('D', 'T', y(nt,11), 'Q', y(nt,10), 'methane', 'ethane', ...
     'nitrogen', [.9 .09 .01]);

% enthalpy
y0(1) = y(nt,1);

% internal energy
y0(12) = y(nt,12);

% liquid density
y0(4) = y(nt,4);

% specific liquid enthalpy
y0(2) = refpropm('H','T',y0(6),'D',y0(4),'methane')
y0(2) = y(nt,2);

% liquid height
y0(8) = 0;

% gas density
y0(5) = mass/(AT*HT);

% specific gas enthalpy
y0(3) = refpropm('H','T',y0(6),'D',y0(5),'methane')
y0(3) = y(nt,3);

% gas pressure
y0(7) = y(nt,7);

% A good guess of a valid first timestep is the length of the last valid
timestep, so use it for faster computation. ‘refine’ is 4 by default.
tstart = t(nt);
options2 = odeset('Mass',M2,'RelTol',rtol);
[t,y] = ode23t(@onetankonephase3,[tstart tfinal],y0,options2);

nt = length(t);
tout = [t(1); t(2:nt)];
yout = [y(1); y(2:nt,:)];
yeout = [ye(1); ye(2:nt,:)];
teout = [te(1); te(2:nt)];

if options2 == 'refine'

else

options = odeset('Mass',M,'RelTol',rtol,'MaxOrder',5);
[t,y] = ode23t(@onetanktwophase,[tstart tfinal],y0,options);

nt = length(t);
tout = [tout; t(2:nt)];
yout = [yout; y(2:nt,:)];
values(:,:) = y(nt,:);
y0(:,1) = y(1,:);

% options = odeset('Mass',M,'RelTol',rtol,'MaxOrder',5);
% [t,y] = ode15s(@onetanktwophase,[100*7*24+360 tfinal],y0,options);
% % Accumulate output. This could be passed out as output arguments.
% nt = length(t);
% tout = [tout; t(2:nt)];
% yout = [yout; y(2:nt,:)];

end

toc

figure(1);
plot( tout/(3600*24*7),yout(:,8),’-’,’LineWidth’,1.8);
grid;
xlabel(’$t$(weeks)’);
ylabel(’$H$(m)’);
h = legend(’$H(t)$’,’location’,’southwest’);

figure(2);
plot( tout/(3600*24*7),yout(:,7),’-’,’LineWidth’,1.8);
grid;
xlabel(’$t$(weeks)’);
ylabel(’$p$(Pa)’);

figure(3);
plot( tout/(3600*24*7),yout(:,2),’-’,tout/(3600*24*7),yout(:,3),’--r’,
’LineWidth’,1.8);
grid;
xlabel(’$t$(weeks)’);
ylabel(’$h$(J)’);

figure(4);
plot( tout/(3600*24*7),yout(:,1),’-’,’LineWidth’,1.8);
grid;
xlabel(’$t$(weeks)’);
ylabel(’$H$(J)’);

figure(5);
plot( tout/(3600*24*7),AT.*yout(:,8).*yout(:,4),’-’,’LineWidth’,1.8);
grid;
xlabel(’$t$(weeks)’);
ylabel(’$m_{liq}$(kg)’);

figure(6);
plot( tout/(3600*24*7),yout(:,5),’-’,’LineWidth’,1.8);
grid;
xlabel(’$t$(weeks)’);
ylabel(’$\rho_{gas}$(kg/m$^3$)’);
C.5 Multi-Component Fluids

grid;
xlabel ( '$t$ (weeks)');
ylabel ( '$\xi$ (kg/kg)');

h = legend ( '$\xi(t)$', 'location', 'southeast');

figure (9);
plot ( tout/(3600*24*7),yout(:,13),tout/(3600*24*7),yout(:,16), '--r', 'LineWidth', 1.8 );
grid;
xlabel ( '$t$ (weeks)');
ylabel ( '$X$ (kg/kg)');
h = legend ( '$X_{liq}^{CH_4}(t)$', '$X_{gas}^{CH_4}(t)$', 'location', 'southwest');

figure (10);
plot ( tout/(3600*24*7),yout(:,14),tout/(3600*24*7),yout(:,17), '--r', 'LineWidth', 1.8 );
grid;
xlabel ( '$t$ (weeks)');
ylabel ( '$X$ (kg/kg)');
h = legend ( '$X_{liq}^{C_2H_6}(t)$', '$X_{gas}^{C_2H_6}(t)$', 'location', 'northwest');

figure (11);
plot ( tout/(3600*24*7),yout(:,15),tout/(3600*24*7),yout(:,18), '--r', 'LineWidth', 1.8 );
grid;
xlabel ( '$t$ (weeks)');
ylabel ( '$X$ (kg/kg)');
h = legend ( '$X_{liq}^{N_2}(t)$', '$X_{gas}^{N_2}(t)$', 'location', 'northeast');

%-------------------------------------------------- -----------------------%
function dy = onetanktwophase(t,y)

% Geometric and physical parameters
g = 9.81; % gravitation constant
HT = 30; % tank heights
DT = 20; % tank diameters
AT = pi.*((DT./2).^2); % cross-sectional area tanks

temperatureamb1 = 293;
K = 55;
dy = zeros(18,1);
mass = AT*HT*refpropm('D','T',y(11),'Q',y(10),'methane','ethane','nitrogen',[.9 .09 .01])...
     - y(9);

% mass = AT*HT*refpropm('D','T',temperature,'Q',quality,'methane','ethane','nitrogen', ...
%  'nitrogen', [ .94364874720886 5.03466958769393e-02 6.0045569021752e-03]);
dy(10) = 0;
dy(11) = 0;

% enthalpy update
dy(1) = -K*(y(6)-temperatureamb1);
% internal energy update
dy(12) = -K*(y(6)-temperatureamb1);
% quality update

dy(9) = refpropm('Q','D',mass/(AT*HT),'U',y(12)/mass,'methane','ethane','nitrogen',[.9 ... .09 .01]) - y(9);
% q -- vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q = 1 indicates saturated vapor
% q > 1 indicates superheated vapor
% q = 998 superheated vapor, but quality not defined (t > Tc)
% q = 999 indicates supercritical state (t > Tc) and (p > Pc)

% temperature
dy(6) = refpropm('T','D',mass/(AT*HT),'U',y(12)/mass,'methane','ethane','nitrogen',[.9 ... .09 .01]) - y(6);

% composition (mass fractions!)
if (y(9)<=1)&&(y(9)>=0)
  [x1 x2] = refpropm('X','T',y(6),'Q',y(9),'methane','ethane','nitrogen',[.9 .09 .01])
  dy(13) = x1(1) - y(13);
  dy(14) = x1(2) - y(14);
  dy(15) = x1(3) - y(15);
else if y(9)>1
  [x1 x2] = refpropm('X','T',y(6),'Q',1,'methane','ethane','nitrogen',[.9 .09 .01]);
  dy(13) = x1(1) - y(13);
  dy(14) = x1(2) - y(14);
  dy(15) = x1(3) - y(15);
endif

% liquid density per tank
if (y(9)<=1)&&(y(9)>=0)
  dy(4) = refpropm('D','T',y(6),'Q',0,'methane','ethane','nitrogen',[y(13) y(14) y(15)...]) - y(4);
else if y(9)>1
  dy(4) = (max((1-y(9)),0) * mass/y(4)) / AT - y(4);
else
  dy(4) = mass/(AT*HT) - y(4);
endif

% liquid height per tank
if (y(9)>=1)
  dy(8) = max(0,((1-y(9)) * mass/y(4)) / AT) - y(8);
else
  dy(8) = (max(1-y(9),0) * mass/y(4)) / (AT) - y(8);
endif

% gas density per tank
C.5 Multi-Component Fluids

if (y(9)>=1)
    dy(5) = mass/(AT*HT) - y(5);
else
    dy(5) = max(0, (y(9)*mass)/(AT*(HT-y(8)))) - y(5);
end

% gas pressure per tank
if (y(9)>=1)
    dy(7) = 1000*refpropm('P','T',y(6),'D',mass/(AT*HT),'methane','ethane','nitrogen',[.9...
        .09 .01]) - y(7);
else
    dy(7) = 1000*refpropm('P','T',y(6),'D',y(5),'methane','ethane','nitrogen',[y(16) y...
        {17} y(18)]) - y(7);
end

% specific liquid enthalpy per tank
if (abs(y(9)-999)<0.1)
    dy(2) = refpropm('H','T',y(6),'D',mass/(AT*HT),'methane','ethane','nitrogen',[.9 .09 ...
        .01]) - y(2);
elseif (y(9)>=0)&&(y(9)<=1)
    dy(2) = refpropm('H','T',y(6),'Q',0,'methane','ethane','nitrogen',[y(13) y(14) y(15)...
        ]) - y(2);
else
    dy(2) = refpropm('H','T',y(6),'Q',1,'methane','ethane','nitrogen',[.9 .09 .01]) - y...
        (2);
end

% specific gas enthalpy per tank
if (y(9)>=1)
    dy(3) = refpropm('H','T',y(6),'D',mass/(AT*HT),'methane','ethane','nitrogen',[.9 .09 ...}
        .01]) - y(3);
else
    dy(3) = refpropm('H','T',y(6),'Q',1,'methane','ethane','nitrogen',[y(16) y(17) y(18)...
        ]) - y(3);
end

function dy = onetankonephase1(t,y)
% if we are left with superheated vapor
% Geometric and physical parameters
q = 9.81; % gravitation constant
HT = 30; % tank heights
DT = 20; % tank diameters
AT = pi.*((DT/2).^2); % cross-sectional area tanks
temperatureamb1 = 293;
K = 55;

mass = AT*HT*refpropm('D','T',y(11),'Q',y(10),'methane','ethane','nitrogen',[.9 .09 .01]) ...}
    ;
% mass = AT*HT*refpropm('D','T',temperature,'Q',quality,'methane','ethane',...}
% 'nitrogen',[.94364874720886 5.03466958769393e-02 6.0045569021752e-03]);
dy = zeros(18,1);
dy(10) = 0;
dy(11) = 0;
% compositions remain constant
dy(13) = 0;
dy(14) = 0;
dy(15) = 0;
dy(16) = 0;
dy(17) = 0;
dy(18) = 0;
% enthalpy update
dy(1) = -K*(y(6)-temperatureamb1);
% internal energy update
dy(12) = -K * (y(6) - temperatureamb1);

% quality update
% y(9,t+1) = refpropm(‘Q’,’D’,mass/(AT+HT),’U’,y(1,t+1)/mass,’methane’);
if (y(9,t+1)==998)
  if superheated<1
    fprintf(‘superheated vapor, but quality not defined (T > Tc)\n’);
    fprintf(‘%g\n’,(-1)*dltt/(24*3600*7));
  end
  superheated = 1;
elseif (y(9,t+1)==999)
  if supercritical<1
    fprintf(‘supercritical state (T > Tc) and \n’);
    fprintf(‘%g\n’,(-1)*dltt/(24*3600*7));
  end
  supercritical = 1;
elseif (y(9,t+1)<0)
  y(9,t+1)=0;
  fprintf(‘subcooled liquid\n’);
  fprintf(‘%g\n’,(-1)*dltt/(24*3600*7));
elseif (y(9,t+1)>1)
  y(9,t+1)=1;
end

% quality update
if (refpropm(‘Q’,’D’,mass/(AT+HT),’U’,y(12)/mass,’methane’,’ethane’,’nitrogen’,[.9 .09 ...
  .01])>1) && (refpropm(‘Q’,’D’,mass/(AT+HT),’U’,y(12)/mass,’methane’,’ethane’,’nitrogen’...
  ,[.9 .09 .01])<998)
  dy(9) = 1 - y(9)
else
dy(9) = refpropm(‘Q’,’D’,mass/(AT+HT),’U’,y(12)/mass,’methane’,’ethane’,’nitrogen’...
  ,[.9 .09 .01]) - y(9);
end

% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q > 1 indicates saturated vapor
% q = 998 superheated vapor, but quality not defined (T > Tc)
% q = 999 indicates supercritical state (T > Tc) and (p > Pc)

% temperature
dy(6) = refpropm(‘T’,’D’,mass/(AT+HT),’U’,y(6),mass,’methane’,’ethane’,’nitrogen’,[.9 ...
  .09 .01]) - y(6);

% liquid density per tank
dy(4) = 0;

% liquid height per tank
dy(8) = 0;

% gas density per tank
dy(5) = mass/(AT+HT) - y(5);

% gas pressure per tank
dy(7) = 1000 * refpropm(‘P’,’T’,y(6),’D’,mass/(AT+HT),’methane’,’ethane’,’nitrogen’,[.9 .09...
  .01]) - y(7);

% specific gas enthalpy per tank
dy(3) = refpropm(‘H’,’T’,y(6),’D’,y(5),’methane’,’ethane’,’nitrogen’,[.9 .09 .01]) - y(3)...
% if (abs(y(9)-999)<0.1)||(y(6)>tcrit)
% dy(2) = refpropm('H','T',y(6),'D',mass/(AT * HT),'methane') - y(2);
elseif abs(y(9)-998)<5e-1
% dy(2) = refpropm('H','T',y(6),'Q',0,'methane') - y(2);
else
% dy(2) = refpropm('H','T',y(6),'D',y(4),'methane') - y(2);
end
dy(2) = 0;

%------------------------------------------------- -----------------------%
function dy = onetankonephase3(t,y) %if we are left with supercritical fluid
% Geometric and physical parameters

%gravitation constant
HT = 30; %tank heights
DT = 20; %tank diameters
AT = pi * (DT./2).^2; %cross-sectional area tanks
temperatureamb1 = 293;
K = 55;
dy = zeros(18,1);
dy(10) = 0;
dy(11) = 0;
% compositions remain constant
dy(13) = 0;
dy(14) = 0;
dy(15) = 0;
dy(16) = 0;
dy(17) = 0;
dy(18) = 0;

mass = AT*HT*refpropm('D','T',y(11),'Q',y(10),',methane','ethane','nitrogen',[.9 .09 .01])...

% mass = AT*HT*refpropm('D','T',temperature,'Q',quality,'methane','ethane',...%'
%nitrogen',[.943648747220886 5.03466958769393e-02 6.0045569021752e-03]);

% enthalpy update
dy(1) = -K * (y(6)-temperatureamb1);
% internal energy update
dy(12) = -K * (y(6)-temperatureamb1);
% quality update
dy(9) = refpropm('Q','D',mass/(AT*HT),'U',y(12)/mass,','methane','ethane','nitrogen',[.9 ...
% 0.9 .01]) - y(9);
% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q = 1 indicates saturated vapor
% q > 1 indicates superheated vapor
% q = 998 superheated vapor, but quality not defined (t > Tc)
% q = 999 indicates supercritical state (t > Tc) and (p > Pc)
% temperature
dy(6) = refpropm('T','D',mass/(AT*HT),'U',y(12)/mass,','methane','ethane','nitrogen',[.9 ...
% 0.9 .01]) - y(6);
% liquid density per tank
dy(4) = 0;
% dy(4) = refpropm('D','T',y(6),'Q',0,'methane','ethane','nitrogen',...
% [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(4);

% liquid height per tank
dy(8) = 0;
% gas density per tank
dy(5) = mass/(AT*HT) - y(5);
% gas pressure per tank
dy(7) = 1000 * refpropm('P', 'T', y(6), 'D', mass/(AT*HT), 'methane', 'ethane', 'nitrogen', [.9 .09 ... .01]) - y(7);

% specific gas enthalpy per tank
dy(3) = refpropm('H', 'T', y(6), 'D', mass/(AT*HT), 'methane', 'ethane', 'nitrogen', [.9 .09 ... .01]) - y(3);

% specific liquid enthalpy per tank
% if (abs(y(9)-999)<0.1)||(y(6)>tcrit)
% dy(2) = refpropm('H','T',y(6),'D',mass/(AT*HT),'methane') - y(2);
% elseif abs(y(9)-998)<5e-1
% dy(2) = refpropm('H','T',y(6),Q',0,'methane') - y(2);
% else
% dy(2) = refpropm('H','T',y(6),'D',y(4),'methane') - y(2);
% end
dy(2) = y(3) - y(2);

%-------------------------------------------------- %
function dy = onetankonephase2(t,y) %if we are left with only (subcooled) liquid

% Geometric and physical parameters
g = 9.81; % gravitation constant
HT = 30; % tank heights
DT = 20; % tank diameters

AT = pi.*((DT./2).^2); % cross-sectional area tanks

temperatureamb1 = 293;
K = 55;
mass = AT*HT*refpropm('D','T',temperature,'Q',y(10), 'methane', 'ethane', 'nitrogen', [.9 .09 .01])
% mass = AT*HT*refpropm('D','T',y(11),'Q',y(10), 'methane', 'ethane', 'nitrogen',
% [.94364874720886 5.03466958769393e-02 6.0045569021752e-03]);
dy = zeros(18,1);
dy(10) = 0; % compositions remain constant
dy(11) = 0;
dy(13) = 0;
dy(14) = 0;
dy(15) = 0;
dy(16) = 0;
dy(17) = 0;
dy(18) = 0;
% enthalpy update
dy(1) = -K*(y(6)-temperatureamb1);
% internal energy update
dy(12) = -K*(y(6)-temperatureamb1);
% quality update
dy(9) = 0;
% q--vapor quality on a MOLAR basis [moles vapor/total moles]
% q < 0 indicates subcooled (compressed) liquid
% q = 0 indicates saturated liquid
% q > 1 indicates superheated vapor
% q > 1 indicates superheated vapor, but quality not defined (t > Tc)
% q > 1 indicates supercritical state (t > Tc) and (p > Pc)

dy(6) = refpropm('T', 'D', mass/(AT*HT), 'U', y(12)/mass, 'methane', 'ethane', 'nitrogen', [.9 ... .09 .01]) - y(6);
dy(4) = mass/(AT*HT)-y(4);
% dy(4) = refpropm('D','T',y(6),'Q',0,'methane','ethane','nitrogen',...
% [.943648747220886 5.03466958769393e-02 6.0045569021752e-03]) - y(4);

% liquid height per tank
dy(8) = 0;
% gas density per tank
dy(5) = 0;
% gas pressure per tank
dy(7) = 0;
% specific liquid enthalpy per tank
dy(2) = refpropm('H','T',y(6),'D',mass/(AT*HT),'methane','ethane','nitrogen',[.9 .09 ... .01]) - y(2);
% specific gas enthalpy per tank, should be edited if y(9)<999
else
dy(3) = refpropm('H','T',y(6),'D',y(5),'methane','ethane','nitrogen',[.9 .09 .01]) - y(3);
dy(3) = y(2) - y(3);
end

%-------------------------------------------------- -----------------------%
function [value,isterminal,direction] = events1(t,y)
value(1) = y(9) - 1;  % detect quality = 1
isterminal(1) = 1;    % stop the integration
direction(1) = 1;     % positive direction

value(2) = y(9);      % detect quality = 0
isterminal(2) = 1;    % stop the integration
direction(2) = -1;    % negative direction

%-------------------------------------------------- -----------------------%
function [value,isterminal,direction] = events2(t,y)
value(1) = y(9) - 1;  % detect quality = 1
isterminal(1) = 1;    % stop the integration
direction(1) = 1;     % positive direction

value(2) = y(9);      % detect quality = 0
isterminal(2) = 1;    % stop the integration
direction(2) = -1;    % negative direction

value(3) = y(9) - 999; % detect supercritical state
isterminal(3) = 1;    % stop the integration
direction(3) = 1;     % positive direction
C.5.3 Explicit Euler for Two-Tank System with Multi-Component Fluid

```matlab
function [values]=multicomptwotankseuler1(timestep)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This script computes numerical solutions of the 1D two-tank model
%% with lower and upper pipe
%% with enthalpy based on equilibrium per tank
%%
%% space discretisation: N/A
%% time integration: explicit Euler
%%
%% Meaning of the variables:
%% HL height of lower pipe-tank connection
%% HU height of upper pipe-tank connection
%% H liquid height in tanks
%% HT total height of tanks
%% p0 initial gas pressure in tanks
%% pin pressure tank 1 at z=0
%% pout pressure tank 2 at z=0
%% (z=0 at lower pipe-tank 1 connection)
%% Dpl diameter of lower pipe
%% Dpu diameter of upper pipe
%% Dt1 diameter of left tank
%% Dt2 diameter of right tank
%% Vtot total liquid volume
%% alpha friction factor

%% theta parameter in the theta-method
%% Nt number of time steps
%% Tmax final time
%% dltt time step

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%'methane','ethane','nitrogen',[ .943648747220886 5.0 3466958769393e-02 6.0045569021752e...
-03]

format long e;
 tic

% Geometric and physical parameters

g = 9.81; %gravitation constant
HL = [1 1]; %lower pipe connection
HU = [20 10]; %upper pipe connection
HT = [20 10]; %tank heights
DP = [0.2 0.2]; %assume Dp<<Dt
DT = [5 2];

alpha = 2; %friction factor for liquid flow in pipe,...
alpha1 = 1; %friction factor for gas flow in pipe

AP = pi.*((DP./2).^2); %cross-sectional area pipes
AT = pi.*((DT./2).^2); %cross-sectional area tanks
specificgasconstant = 8.314472/(16.042e-3); %specific gas constant methane=gas ...
constant/molar mass (kg/mol)
constraints = 2; %use constraints nr 1 for gas flow:1,
otherwise
molarmassmethane = 16.042e-3; %kg/mol
molarmassethane = 30.07e-3; %kg/mol
molarmassnitrogen = 14.0067e-3; %kg/mol

% rhol = refpropm('D','T',liquidtemp,'P',100,'methane');
```

\[
rhol = 433;
\]
\[
% parameters for euler method
theta = 0;
\]
\[
% grid (time)
Tmax = 200;
dltt = timestep;
\]
\[
Nt = round(Tmax/dltt);
\]
\[
% variables
H = zeros(Nt+1,2);
pin = zeros(Nt+1,1);
\]
\[
pgas = zeros(Nt+1,2);
rhoq = zeros(Nt+1,2);
quality = zeros(Nt+1,2);
temperature = zeros(Nt+1,2);
totaltankmass = zeros(Nt+1,2);
totaltankmethanemass = zeros(Nt+1,2);
totaltankethanemass = zeros(Nt+1,2);
totaltanknitrogenmass = zeros(Nt+1,2);
totalktankenthalpy = zeros(Nt+1,2);
\]
\[
rholiq = zeros(Nt+1,2);
specliqenthalpy = zeros(Nt+1,2);
specgasenthalpy = zeros(Nt+1,2);
values = zeros(2,1);
\]
\[
compositions = zeros(Nt+1,3,2,2);
% 1st entry time
% 2nd entry fraction methane ethane nitrogen
% 3rd entry liquid (1) gas (2)
% 4th entry tank 1 or 2
\]
\[
% initial conditions
% quality
quality(1,1) = 0.02;
quality(1,2) = 0.02;
\]
\[
% temperature
temperature(1,1) = 112;
temperature(1,2) = 125;
\]
\[
% composition (mass fractions!)
[x1 x2] = refpropm('X','T',temperature(1,1), 'Q', quality(1,1), 'methane',... 'ethane', 'nitrogen', [.9 .09 .01]);
\]
\[
[x3 x4] = refpropm('X','T',temperature(1,2), 'Q', quality(1,2), 'methane',... 'ethane', 'nitrogen', [.9 .09 .01]);
\]
\[
% liquid tank 1
compositions(1,1,1,1) = x1(1);
compositions(1,2,1,1) = x1(2);
compositions(1,3,1,1) = x1(3);
\]
\[
% gas tank 1
compositions(1,1,2,1) = x2(1);
compositions(1,2,2,1) = x2(2);
compositions(1,3,2,1) = x2(3);
\]
\[
% liquid tank 2
compositions(1,1,1,2) = x3(1);
compositions(1,2,1,2) = x3(2);
compositions(1,3,1,2) = x3(3);
% gas tank 2
compositions(1,1,2,2) = x4(1);
compositions(1,2,2,2) = x4(2);
compositions(1,3,2,2) = x4(3);

% total mass per tank
totaltankmass(1,1) = AT(1)*HT(1)*refpropm('D','T',temperature(1,1),'Q',quality(1,1),'
methane','ethane','nitrogen',[0.9 0.09 0.01]);
totaltankmass(1,2) = AT(2)*HT(2)*refpropm('D','T',temperature(1,2),'Q',quality(1,2),'
methane','ethane','nitrogen',[0.9 0.09 0.01]);
totaltankmethanemass(1,1) = 0.9*totaltankmass(1,1);
totaltankmethanemass(1,2) = 0.9*totaltankmass(1,2);
totaltankethanemass(1,1) = 0.09*totaltankmass(1,1);
totaltankethanemass(1,2) = 0.09*totaltankmass(1,2);
totaltanknitrogenmass(1,1) = 0.01*totaltankmass(1,1);
totaltanknitrogenmass(1,2) = 0.01*totaltankmass(1,2);

% total enthalpy per tank
totaltankenthalpy(1,1) = totaltankmass(1,1)*refpropm('H','T',temperature(1,1),'Q',quality...
(1,1),'methane','ethane','nitrogen',[0.9 0.09 0.01]);
totaltankenthalpy(1,2) = totaltankmass(1,2)*refpropm('H','T',temperature(1,2),'Q',quality...
(1,2),'methane','ethane','nitrogen',[0.9 0.09 0.01]);

% liquid density per tank
rholiq(1,1) = refpropm('D','T',temperature(1,1),'Q',0,'methane','ethane','nitrogen',
compositions(1,1,1,1) compositions(1,2,1,1) compositions... 
(1,3,1,1));
rholiq(1,2) = refpropm('D','T',temperature(1,2),'Q',0,'methane','ethane','nitrogen',
compositions(1,1,1,2) compositions(1,2,1,2) compositions...
(1,3,1,2));

% specific liquid enthalpy per tank
specliqenthalpy(1,1) = refpropm('H','T',temperature(1,1),'D',rholiq(1,1),'methane','ethane','nitrogen',
compositions(1,1,2,1) compositions(1,2,2,1) compositions...
(1,3,2,1));
specliqenthalpy(1,2) = refpropm('H','T',temperature(1,2),'D',rholiq(1,2),'methane','ethane','nitrogen',
compositions(1,1,2,2) compositions(1,2,2,2) compositions...
(1,3,2,2));

% liquid height per tank
H(1,1) = ((1-quality(1,1))*totaltankmass(1,1))/(rholiq(1,1)*AT(1));
H(1,2) = ((1-quality(1,2))*totaltankmass(1,2))/(rholiq(1,2)*AT(2));

% gas density per tank
rhog(1,1) = (quality(1,1)*totaltankmass(1,1))/(AT(1)*(HT(1)-H(1,1)));
rhog(1,2) = (quality(1,2)*totaltankmass(1,2))/(AT(2)*(HT(2)-H(1,2)));

% specific gas enthalpy per tank
specgasenthalpy(1,1) = refpropm('H','T',temperature(1,1),'D',rhog(1,1),'methane','ethane','nitrogen',
compositions(1,1,2,1) compositions(1,2,2,1) compositions...
(1,3,2,1));
specgasenthalpy(1,2) = refpropm('H','T',temperature(1,2),'D',rhog(1,2),'methane','ethane','nitrogen',
compositions(1,1,2,2) compositions(1,2,2,2) compositions...
(1,3,2,2));

% gas pressure per tank
pgas(1,1) = 1000*refpropm('P','T',temperature(1,1),'D',rhog(1,1),'methane','ethane','nitrogen',
compositions(1,1,2,1) compositions(1,2,2,1) compositions...
(1,3,2,1));
pgas(1,2) = 1000*refpropm('P', 'T', temperature(1,2), 'D', rhog(1,2), 'methane', 'ethane', 'nitrogen', ... compositions(1,1,2,2) compositions(1,2,2,2) compositions ... (1,3,2,2));

% pressure at end of lower pipe
pin(1) = pgas(1,1)+rholiq(1,1)*g*(max(H(1,1)-HL(1),0));
pout(1) = pgas(1,2)+rholiq(1,2)*g*(max(H(1,2)-HL(1),0));

fractionL = 1;
fractionU = 0;

Vliq = AT(1)*H(1,1)+AT(2)*H(1,2)+fractionL*AP(1)*LP(1)+fractionU*AP(2)*LP(2);
Mliq = rhol*Vliq;
Vgas1 = AT(1)*(HT(1)-H(1,1));
Vgas2 = AT(2)*(HT(2)-H(1,2));
Vgas = Vgas1+Vgas2+(1-fractionL)*AP(1)*LP(1)+(1-fractionU)*AP(2)*LP(2);
Mgas1 = rhog(1,1)*AT(1)*(HT(1)-H(1,1));
Mgas2 = rhog(1,2)*AT(2)*(HT(2)-H(1,2));
Mgas = Mgas1+Mgas2+(Mgas1+Mgas2)/(Vgas1+Vgas2)*((1-fractionL)*AP(1)*LP(1)+(1-fractionU)*AP(2)*LP(2));

rhogasaverage = Mgas/Vgas;

% compute solution for closed system
for n=1:Nt
    if (pin(n)-pout(n)>1e-4)&&(H(n,1)-max(HL(1),HL(2))>1e-13)
        newH1 = max(H(n,1)-dltt*(AP(1)/AT(1))*sqrt(abs((2/(1+alpha)) * (g*(H(n,1)-max(H(n... ,2),HL(2))))+((pgas(n,1)-pgas(n,2))/(rholiq(n,1))))),HL(1));
        newH2 = H(n,2)+AT(2)*(H(n,2)-newH2)/AT(1);
        liqmassflux = (H(n,2)-newH2)*rholiq(n,2)*AT(2);
        liqenthalpyflux = liqmassflux*specliqenthalpy(n,2);
        liqmethaneflux = liqmassflux*compositions(n,1,2);
        liqethaneflux = liqmassflux*compositions(n,2,2);
        liqnitrogenflux = liqmassflux*compositions(n,3,2);
    elseif (pin(n)-pout(n)<-1e-4)&&(H(n,2)-max(HL(1),HL(2))>1e-13)
        newH2 = max(H(n,2)-dltt*(AP(1)/AT(2))*sqrt(abs((2/(1+alpha)) * (g*(H(n,2)-max(H(n... ,1),HL(2))))+((pgas(n,2)-pgas(n,1))/(rholiq(n,2))))),HL(2));
        newH1 = H(n,1)+AT(1)*(H(n,1)-newH1)/AT(2);
        liqmassflux = -((H(n,2)-newH2)*rholiq(n,2)*AT(2));
        liqenthalpyflux = liqmassflux*specliqenthalpy(n,2);
        liqmethaneflux = liqmassflux*compositions(n,1,1);
        liqethaneflux = liqmassflux*compositions(n,2,1);
        liqnitrogenflux = liqmassflux*compositions(n,3,1);
    else
        liqmassflux = 0;
        liqenthalpyflux = 0;
        liqmethaneflux = 0;
        liqethaneflux = 0;
        liqnitrogenflux = 0;
        newH1 = H(n,1);
    end

Vgas1 = AT(1)*H(1,1)-newH1;
Vgas2 = AT(2)*H(1,2)-newH2;
Mgasequilibrium1 = rhogasaverage*Vgas1;
Mgasequilibrium2 = rhogasaverage*Vgas2;

if ((pgas(n,1)-pgas(n,2))/pgas(n,1)>1e-2)
    gasvelocity = sqrt(abs(2*(pgas(n,1)-pgas(n,2))/((1+alpha1)*rhog(n,2))));
    massflux = rhog(n,2)*AP(2)*gasvelocity;
endif constraints==1||constraints==2
if Mgaseq-dltt*massflux<Mgasequilibrium1
    gasmassflux = abs(Mgaseq-Mgasequilibrium1); %is positive
else
    \texttt{gasmassflux = abs(dltt \times massflux); \% is positive}
end
\texttt{Mgas1 = max(Mgas1 - dltt \times massflux, Mgasequilibrium1);}
\texttt{Mgas2 = min(Mgas2 + dltt \times massflux, Mgasequilibrium2);}
else
    \texttt{gasmassflux = dltt \times massflux; \% is positive}
    \texttt{Mgas1 = Mgas1 - dltt \times massflux;}
    \texttt{Mgas2 = Mgas2 + dltt \times massflux;}
end
\texttt{gashenthalpyflux = gasmassflux \times specgasenthalpy(n,1); \% is positive}
gasmethaneflux = gasmassflux \times compositions(n,1,2,1);
gasethaneflux = gasmassflux \times compositions(n,2,2,1);
gasnitrogenflux = gasmassflux \times compositions(n,3,2,1);
elseif (p_{gas}(n,1) - p_{gas}(n,2))/p_{gas}(n,1) < -1e-2
gasvelocity = \sqrt{\frac{2 \times (p_{gas}(n,2) - p_{gas}(n,1))}{(1 + \alpha1) \times \rho_{g}(n,1)}};
\texttt{massflux = \rho_{g}(n,1) \times \sqrt{2 \times (p_{gas}(n,2) - p_{gas}(n,1)) / ((1 + \alpha1) \times \rho_{g}(n,1))};}
\texttt{if constraints==1 || constraints==2}
\texttt{if Mgas1+dltt \times massflux<Mgasequilibrium1}
\texttt{gasmassflux = -abs(dltt \times massflux); \% is negative}
\texttt{else}
\texttt{Mgas1 = min(Mgas1+dltt \times gasmassflux, Mgasequilibrium1);}
\texttt{Mgas2 = max(Mgas2-dltt \times gasmassflux, Mgasequilibrium2);}
\texttt{else}
\texttt{gasmassflux = -abs(dltt \times massflux); \% is negative}
    \texttt{Mgas1 = Mgas1+dltt \times gasmassflux;}
    \texttt{Mgas2 = Mgas2-dltt \times gasmassflux;}
end
\texttt{gashenthalpyflux = gasmassflux \times specgasenthalpy(n,2);}
gasmethaneflux = gasmassflux \times compositions(n,1,2,2);
gasethaneflux = gasmassflux \times compositions(n,2,2,2);
gasnitrogenflux = gasmassflux \times compositions(n,3,2,2);
else
    \texttt{if constraints==2}
    \texttt{Mgas1 = Mgasequilibrium1;}
    \texttt{Mgas2 = Mgasequilibrium2;}
    \texttt{if pin(n)>pout(n)}
\texttt{gasmassflux = -abs(Mgas1-Mgasequilibrium1); \% is negative}
gasenthalpyflux = gasmassflux \times specgasenthalpy(n,2);
gasmethaneflux = gasmassflux \times compositions(n,1,2,2);
gasethaneflux = gasmassflux \times compositions(n,2,2,2);
gasnitrogenflux = gasmassflux \times compositions(n,3,2,2);
\texttt{else}
\texttt{gasmassflux = abs(Mgas1-Mgasequilibrium1); \% is positive}
gasenthalpyflux = gasmassflux \times specgasenthalpy(n,1);
gasmethaneflux = gasmassflux \times compositions(n,1,2,1);
gasethaneflux = gasmassflux \times compositions(n,2,2,1);
gasnitrogenflux = gasmassflux \times compositions(n,3,2,1);
\texttt{end}
\texttt{else}
\texttt{gasmassflux = 0;}
gasenthalpyflux = 0;
gasmethaneflux = 0;
gasethaneflux = 0;
gasnitrogenflux = 0;
\texttt{end}
end
\texttt{totaltankmass(n+1,1) = totaltankmass(n,1) - gasmassflux - liqmassflux;}
\texttt{totaltankmass(n+1,2) = totaltankmass(n,2) + gasmassflux + liqmassflux;}
\texttt{totaltankmethanemass(n+1,1) = totaltankmethanemass(n,1) - gasmethaneflux - liqmethaneflux...}
\[
\text{totaltankmethanemass}(n+1,2) = \text{totaltankmethanemass}(n,2) + \text{gasmethaneflux} + \text{liqmethaneflux};
\]

\[
\text{totaltankethanemass}(n+1,1) = \text{totaltankethanemass}(n,1) - \text{gasethaneflux} - \text{liqethaneflux};
\]

\[
\text{totaltankethanemass}(n+1,2) = \text{totaltankethanemass}(n,2) + \text{gasethaneflux} + \text{liqethaneflux};
\]

\[
\text{totaltanknitrogenmass}(n+1,1) = \text{totaltanknitrogenmass}(n,1) - \text{gasnitrogenflux} - \text{liqnitrogenflux};
\]

\[
\text{totaltanknitrogenmass}(n+1,2) = \text{totaltanknitrogenmass}(n,2) + \text{gasnitrogenflux} + \text{liqnitrogenflux};
\]

\[
\text{totaltankenthalpy}(n+1,1) = \text{totaltankenthalpy}(n,1) - \text{liqenthalpyflux} - \text{gasthalpyflux};
\]

\[
\text{totaltankenthalpy}(n+1,2) = \text{totaltankenthalpy}(n,2) + \text{liqenthalpyflux} + \text{gasthalpyflux};
\]

\[
\text{quality}(n+1,1) = \text{refpropm('~Q', ~'D', \text{totaltankmass}(n+1,1)/(\text{AT}(1) + \text{HT}(1)), ~'H', \text{totaltankenthalpy}(n+1,1)/\text{totaltankmass}(n+1,1), \text{methane'}, \text{ethane'}, \text{nitrogen'}, \text{totaltankmethanemass}(n+1,1)/\text{totaltankmass}(n+1,1), \text{totaltankethanemass}(n+1,1)/\text{totaltankmass}(n+1,1), \text{totaltanknitrogenmass}(n+1,1)/\text{totaltankmass}(n+1,1));}
\]

\[
\text{quality}(n+1,2) = \text{refpropm('~Q', ~'D', \text{totaltankmass}(n+1,2)/(\text{AT}(2) + \text{HT}(2)), ~'H', \text{totaltankenthalpy}(n+1,2)/\text{totaltankmass}(n+1,2), \text{methane'}, \text{ethane'}, \text{nitrogen'}, \text{totaltankmethanemass}(n+1,2)/\text{totaltankmass}(n+1,2), \text{totaltankethanemass}(n+1,2)/\text{totaltankmass}(n+1,2), \text{totaltanknitrogenmass}(n+1,2)/\text{totaltankmass}(n+1,2));}
\]

\[
\text{temperature}(n+1,1) = \text{refpropm('~T', ~'D', \text{totaltankmass}(n+1,1)/(\text{AT}(1) + \text{HT}(1)), ~'H', \text{totaltankenthalpy}(n+1,1)/\text{totaltankmass}(n+1,1), \text{methane'}, \text{ethane'}, \text{nitrogen'}, \text{totaltankmethanemass}(n+1,1)/\text{totaltankmass}(n+1,1), \text{totaltankethanemass}(n+1,1)/\text{totaltankmass}(n+1,1), \text{totaltanknitrogenmass}(n+1,1)/\text{totaltankmass}(n+1,1));}
\]

\[
\text{temperature}(n+1,2) = \text{refpropm('~T', ~'D', \text{totaltankmass}(n+1,2)/(\text{AT}(2) + \text{HT}(2)), ~'H', \text{totaltankenthalpy}(n+1,2)/\text{totaltankmass}(n+1,2), \text{methane'}, \text{ethane'}, \text{nitrogen'}, \text{totaltankmethanemass}(n+1,2)/\text{totaltankmass}(n+1,2), \text{totaltankethanemass}(n+1,2)/\text{totaltankmass}(n+1,2), \text{totaltanknitrogenmass}(n+1,2)/\text{totaltankmass}(n+1,2));}
\]

\[
[x1 \times 2] = \text{refpropm('~X', ~'T', \text{temperature}(1,1), ~'Q', \text{quality}(1,1), \text{methane'}, \text{ethane'}, \text{nitrogen'}, \text{totaltankmethanemass}(n+1,1)/\text{totaltankmass}(n+1,1), \text{totaltankethanemass}(n+1,1)/\text{totaltankmass}(n+1,1), \text{totaltanknitrogenmass}(n+1,1)/\text{totaltankmass}(n+1,1));}
\]

\[
[x3 \times 4] = \text{refpropm('~X', ~'T', \text{temperature}(1,2), ~'Q', \text{quality}(1,2), \text{methane'}, \text{ethane'}, \text{nitrogen'}, \text{totaltankmethanemass}(n+1,2)/\text{totaltankmass}(n+1,2), \text{totaltankethanemass}(n+1,2)/\text{totaltankmass}(n+1,2), \text{totaltanknitrogenmass}(n+1,2)/\text{totaltankmass}(n+1,2));}
\]

\[
\text{compositions}(n+1,1,1,1) = x1(1);
\]

\[
\text{compositions}(n+1,2,1,1) = x1(2);
\]

\[
\text{compositions}(n+1,3,1,1) = x1(3);
\]

\[
\text{compositions}(n+1,1,2,1) = x2(1);
\]

\[
\text{compositions}(n+1,2,2,1) = x2(2);
\]

\[
\text{compositions}(n+1,3,2,1) = x2(3);
\]

\[
\text{compositions}(n+1,1,1,2) = x3(1);
\]

\[
\text{compositions}(n+1,2,1,2) = x3(2);
\]

\[
\text{compositions}(n+1,3,1,2) = x3(3);
\]

\[
\text{compositions}(n+1,1,2,2) = x4(1);
\]

\[
\text{compositions}(n+1,2,2,2) = x4(2);
\]

\[
\text{compositions}(n+1,3,2,2) = x4(3);
\]

\[
\text{liquid density per tank}
\]
rholiq(n+1,1) = refpropm(‘D’,’T’,temperature(n+1,1),’Q’,0,’methane’,...
’ethane’,’nitrogen’,[ compositions(n+1,1,1) compositions(n+1,2,1,1) compositions(n+
+1,3,1,1)]);
rholiq(n+1,2) = refpropm(‘D’,’T’,temperature(n+1,2),’Q’,0,’methane’,...
’ethane’,’nitrogen’,[ compositions(n+1,1,1,2) compositions(n+1,2,1,2) compositions(n+
+1,3,1,2)]);

% liquid height per tank
H(n+1,1) = ((1-quality(n+1,1)) * totaltankmass(n+1,1)/rholiq(n+1,1))/(AT(1));
H(n+1,2) = ((1-quality(n+1,2)) * totaltankmass(n+1,2)/rholiq(n+1,2))/(AT(2));

% specific liquid enthalpy per tank
specliqenthalpy(n+1,1) = refpropm(‘H’,’T’,temperature(n+1,1),’D’,rholiq(n+1,1),’.....
’methane’,’nitrogen’,[ compositions(n+1,1,1,1) compositions(n+1,2,1,1) compositions(n+
+1,3,1,1)]);
specliqenthalpy(n+1,2) = refpropm(‘H’,’T’,temperature(n+1,2),’D’,rholiq(n+1,2),’.....
’methane’,’nitrogen’,[ compositions(n+1,1,1,2) compositions(n+1,2,1,2) compositions(n+
+1,3,1,2)]);

% gas density per tank
rhog(n+1,1) = (quality(n+1,1) * totaltankmass(n+1,1))/(AT(1) * (HT(1)-H(n+1,1)));
rhog(n+1,2) = (quality(n+1,2) * totaltankmass(n+1,2))/(AT(2) * (HT(2)-H(n+1,2)));

% specific gas enthalpy per tank
specgasenthalpy(n+1,1) = refpropm(‘H’,’T’,temperature(n+1,1),’Q’,1,’methane’,...
’ethane’,’nitrogen’,[ gasmethanemoles1/sumgas1 gasethanemoles1/sumgas1 ...
gasnitrogenmoles1/sumgas1]);
specgasenthalpy(n+1,2) = refpropm(‘H’,’T’,temperature(n+1,2),’Q’,1,’methane’,...
’ethane’,’nitrogen’,[ gasmethanemoles2/sumgas2 gasethanemoles2/sumgas2 ...
gasnitrogenmoles2/sumgas2]);

% gas pressure per tank
pgas(n+1,1) = 1000 * refpropm(‘P’,’T’,temperature(n+1,1),’D’,rhog(n+1,1),’.....
’ethane’,’nitrogen’,[ gasmethanemoles1/sumgas1 gasethanemoles1/sumgas1 ...
gasnitrogenmoles1/sumgas1]);
pgas(n+1,2) = 1000 * refpropm(‘P’,’T’,temperature(n+1,2),’D’,rhog(n+1,2),’.....
’ethane’,’nitrogen’,[ gasmethanemoles2/sumgas2 gasethanemoles2/sumgas2 ...
gasnitrogenmoles2/sumgas2]);

% pressure at end of lower pipe
pin(n+1) = pgas(n+1,1)+rholiq(n+1,1) * g * (max(H(n+1,1)-HL(1),0));
pout(n+1) = pgas(n+1,2)+rholiq(n+1,2) * g * (max(H(n+1,2)-HL(1),0));
values(:, :) = H(Nt+1,:);

figure(1);
plot([0:dltt:Tmax],H(:,1),[0:dltt:Tmax],H(:,2),'
--red','LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$H$(m)');
h = legend('\$H_1(t)$','\$H_2(t)$',4);

figure(2);
plot([0:dltt:Tmax],pin,[0:dltt:Tmax],pout,'
--red','LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$p$(Pa)');
h = legend('\$p_{in}(t)$','\$p_{out}(t)$','location','northeast');

figure(3);
plot([0:dltt:Tmax],pgas(:,1),[0:dltt:Tmax],pgas(:,2),'
--red','LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$p$(Pa)');
h = legend('\$p_{gas1}(t)$','\$p_{gas2}(t)$','location','northEast');

figure(4);
plot([0:dltt:Tmax],temperature(:,1),[0:dltt:Tmax],temperature(:,2),'
--red','LineWidth'...
1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$T$(K)');
h = legend('\$T_{1}(t)$','\$T_{2}(t)$','location','northEast');

figure(5);
plot([0:dltt:Tmax],totaltankmass(:,1),[0:dltt:Tmax],totaltankmass(:,2),'
--red','...'
LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$m$(kg)');
h = legend('\$m_{1}(t)$','\$m_{2}(t)$','location','east');

figure(6);
plot([0:dltt:Tmax],quality(:,1),[0:dltt:Tmax],quality(:,2),'
--red','LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$\xi$(kg)');
h = legend('\$\xi_{1}(t)$','\$\xi_{2}(t)$','location','northEast');

figure(7);
plot([0:dltt:Tmax],compositions(:,1,1,1),[0:dltt:Tmax],compositions(:,1,1,2),'
--red','...'
LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$X$(kg/kg)');
h = legend('\$X_{liq,1}^{-CH_4}(t)$','\$X_{gas,1}^{-CH_4}(t)$','location','east');

figure(8);
plot([0:dltt:Tmax],compositions(:,1,2,1),[0:dltt:Tmax],compositions(:,1,2,2),'
--red','...'
LineWidth',1.8);
grid;
xlabel('\$t$(s)');
ylabel('\$X$(kg/kg)');
h = legend('\$X_{liq,2}^{-CH_4}(t)$','\$X_{gas,2}^{-CH_4}(t)$','location','SouthEast');

figure(9);
plot([0:dltt:Tmax],compositions(:,2,1,1),[0:dltt:Tmax],compositions(:,2,1,2),'
--red','...'
LineWidth',1.8);
grid; xlabel ('$t$ (s)'); ylabel ('$X$ (kg/kg)');
h = legend ('$X_{liq,1}^{C_2H_6}(t)$', '$X_{gas,1}^{C_2H_6}(t)$', 'location', 'east');
figure(10); plot([0:dltt:Tmax], compositions(:,2,1,2), [0:dltt:Tmax], compositions(:,2,2,2), '--red', 'LineWidth', 1.8);
grid; xlabel ('$t$ (s)'); ylabel ('$X$ (kg/kg)');
h = legend ('$X_{liq,2}^{C_2H_6}(t)$', '$X_{gas,2}^{C_2H_6}(t)$', 'location', 'east');
figure(11);
plot([0:dltt:Tmax], compositions(:,3,1,1), [0:dltt:Tmax], compositions(:,3,2,1), '--red', 'LineWidth', 1.8);
grid; xlabel ('$t$ (s)'); ylabel ('$X$ (kg/kg)');
h = legend ('$X_{liq,1}^{N_2}(t)$', '$X_{gas,1}^{N_2}(t)$', 'location', 'east');
figure(12);
plot([0:dltt:Tmax], compositions(:,3,1,2), [0:dltt:Tmax], compositions(:,3,2,2), '--red', 'LineWidth', 1.8);
grid; xlabel ('$t$ (s)'); ylabel ('$X$ (kg/kg)');
h = legend ('$X_{liq,2}^{N_2}(t)$', '$X_{gas,2}^{N_2}(t)$', 'location', 'northEast');
laprint(1,'mc2H1','figcopy','off','width',7.375);
laprint(2,'mc2p1','figcopy','off','width',7.375);
laprint(3,'mc2pgas1','figcopy','off','width',7.375);
laprint(4,'mc2m1','figcopy','off','width',7.375);
laprint(5,'mc2methane11','figcopy','off','width',7.375);
laprint(6,'mc2methane21','figcopy','off','width',7.375);
laprint(7,'mc2ethane11','figcopy','off','width',7.375);
laprint(8,'mc2ethane21','figcopy','off','width',7.375);
laprint(9,'mc2nitrogen11','figcopy','off','width',7.375);
laprint(10,'mc2nitrogen21','figcopy','off','width',7.375);
Bibliography


Index

θ-method, 16
absolute pressure, 17
adiabatic boundary, 79
backward differentiation formula (BDF), 16
Bernoulli’s equation, 18
extended, 20 24 28 37 103 106
Biot number, 26 68 79 106
BLEVE, 3
boil-off, 1 68
boiling point, 67
ethane, 92
methane, 92
nitrogen, 92
Boyle’s law, 11 28 105
Butcher tableau, 15 16
compressed natural gas (CNG), 1
conductivity coefficient, 12
conservation of
  energy, 9
  energy, in enthalpy form, 10 60
  linear momentum, 8
  mass, 7 22 63 93
constitutive equations, 11
continuity equation, 8 18
  corresponding states, 12
critical point, 68
Darcy friction factor, 19 57 106
Darcy-Weisbach equation, 19
differential-algebraic equation (DAE), 13 60
  fully-implicit, 14
  semi-explicit, 14
Dormand-Prince pair, 16
energy loss, 19
enthalpy, 10 60 71 80
equation of state, 11 66 113
  Benedict-Webb-Rubin, 12
  Boyle’s law, 11 28 105
  ideal gas law, 11
equilibrium, 22 66
  condition, 23
explicit Euler method, 15 29 59 11 17 69
  accuracy, 39 58 102
  constraint, 43 45 105
  convergence, 68 58 102
  speed, 58 89 102
  stability, 59 17
  stability condition, 10 12
Fourier’s law of thermal conduction, 12 64
  79
Frobenius inner product, 12
gas flow, 57
GERG model, 91
head, 19
heat capacity, 79
heat flux, 12 63 64 88 73
  coefficient, 64 107
Helmholtz energy, 12
ideal gas law, 11
implicit Euler method, 15
initial value problem (IVP), 13
internal energy, 9 60
isenthalpic process, 80
isochoric process, 60
liquefied natural gas (LNG)
  carrier, 1
  composition, 1 91
  density, 26
properties, 1, 79
safety, 3
supply and demand, 4
lumped-capacity model, 65

machine precision, 45
Matlab, 13, 14, 107, 113
Event, 50, 74, 83
ode15s, 16, 49, 50, 72, 81, 93, 105
ode23t, 16, 49, 50, 72, 81, 105
ode45, 16, 49, 50, 50
accuracy, 68, 69, 89, 102
breakdown, 69, 50, 72, 74, 83, 105
speed, 58, 75, 89, 102
momentum equation, 18
Moody diagram, 106
multi-component fluid, 91, 106

numerical differentiation formula (NDF), 16, 49, 105
numerical methods, 15, 75
convergence, 14, 95
speed, 102
stability, 14

ordinary differential equation (ODE), 13, 22, 24, 29
Péclet number, 74, 105, 110
Pantelides’ algorithm, 14
pressure
absolute, 17

quality, 63, 68, 106

rapid phase transition (RPT), 5
REFPROP, 5, 12, 63, 65, 66, 72, 79, 91, 105, 107, 113
incompleteness, 92, 106, 114
speed, 75
Reynolds transport theorem, 7
rollover effect, 3
Runge-Kutta method, 15, 46, 106

saturation temperature, 67
subcooled liquid, 67, 69, 74
supercritical fluid, 68, 69, 72, 74
superheated vapor, 66, 68

tensor
deformation velocity, 11
stress, 8
viscous stress, 9, 11
time integration, 15
tolerance, 14, 16, 49, 50, 72, 75, 83, 93, 105, 106
absolute, 14
relative, 14
Torricelli’s law, 22
transport theorem, 7
trapezoidal rule, 16, 72

viscosity
dynamic, 11
kinematic, 11

waterhammer effect, 4, 108
weathering, 3