NTNU – Trondheim
Norwegian University of
Science and Technology

Numerical methods for solving complex heat exchanger models in transient operation

Bjørnar Gundersen Skåre

Master of Science in Physics and Mathematics
Submission date: June 2014
Supervisor: Alex Hansen, IFY
Co-supervisor: Geir Skaugen, SINTEF Energi AS - Gassteknologi

Norwegian University of Science and Technology
Department of Physics

Master thesis work

for

student Bjørnar Skåre

Spring 2014

Numerical methods for solving complex heat exchanger models in transient operation.

Numeriske metoder for løsing av komplekse varmevekslermodeller for transiente forhold.

Background and objective

Heat exchangers play a key role in many industrial processes, and correct dimensioning and design is essential for the plant to operate as wanted. Tools that correctly predict the performance of heat exchangers are therefore very important.

Heat exchanger often consists of complex geometry with multiple streams. The descriptions of heat transfer and pressure drop are often highly non-linear, which makes the models numerically difficult to solve. It is often also desirable to use the heat exchanger models in optimisation routines for both process and geometrical design. Calculation speed is therefore also an important factor.

At SINTEF Energy Research, a heat exchanger modelling framework has been developed for solving several types of industrial heat exchangers. Several numerical methods (ODE solvers and equation solvers) are available in the library.

In the project work entitled "Numerical methods for solving complex heat exchanger models" an analysis and comparison using different numerical methods for solving a thermally complex heat exchanger under steady state condition was conducted.

An extension to this work is to implement a numerical solver for transient operation. The solver should be used both as an alternative to the steady-state solver for reaching steady conditions and to study the effects of process disturbances relevant for a selected industrial case.

The following tasks are to be considered:

1. Do a literature review of numerical methods and strategies for solving transient heat exchanger models where phase change can occur.
2. Implement a transient heat exchanger solver in the existing heat exchanger framework.

3. Compare and discuss the use such methods as an alternative to the earlier implemented steady-state solvers.
4. Implement the heat exchanger model in a process environment and test the effects of process disturbances.
5. Propose further work.

Abstract

Heat exchangers are important in many industrial systems. In order to maximize heat transfer while fulfilling requirements to size and weight, it is crucial to have good numerical tools available. Sintef has developed a modelling framework for solving a variety of different heat exchangers in steady-state. This thesis contains the work done to extend the framework with a transient solver. Dynamical aspects of heat exchangers can not be investigated by steady-state solvers, and a transient formulation is needed to explore these phenomena.

The transient formulation that has been attempted has proven to form a set of stiff equations. The stiffness is related to the size of the control volumes used by the solver. Furthermore, the transient solver is unable to solve a "base case" for the reference heat exchanger that has been investigated. In order for the solution to converge, the tube diameters need to be multiplied by a factor of 30. Increasing the tube diameters does not only violate potential size requirements, but it also results in a lower heat transfer. Thus, the transient formulation that is implemented is not robust enough to solve compact heat exchangers.

Sammendrag

Varmevekslere har en nøkkelrolle i mange industrielle system. For å maksimere varmeoverføring samtidig som krav til størrelse og vekt er oppfylt, er det viktig å ha gode numeriske verktøy til rådighet. Sintef har utviklet et rammeverk for modellering av ulike varmevekslere som kan løses i likevekt. Denne oppgaven inneholder arbeidet som er gjort for å utvide rammeverket med en transient løser. Dynamiske sider av varmevekslere kan ikke undersøkes av en likevektsløser, og en transient problemstilling er nødvendig for å utforske disse fenomenene.

Den transiente problemstillingen som er benyttet har vist seg å utgjøre et sett av stive ligninger. Stivheten er relatert til størrelsen av kontrollvolumene som er brukt av løseren. Videre er ikke den transiente løseren i stand til å løse "standardtilfellet" til varmeveksleren som har blitt undersøkt. For å få løsningen til å konvergere, må diameterene på rørene multipliseres med en faktor på 30. Økning av rørdiameterene motsier ikke bare eventuelle krav til størrelse, men det resulterer også i lavere varmeoverføring. Den transiente problemstillingen som er implementert er derfor ikke robust nok til å løse kompakte varmevekslere.

Preface

This work is a result of TFY4900 - Master Thesis (30 sp). My background is in Physics and Mathematics, and relevant higher level courses include: TEP4165 - Viscous Flows and Boundary Layers, TEP4112 - Turbulent Flows , TMA4215 - Numerical Mathematics, TFY4235 - Numerical Physics and TMA4220 - Numerical Solution of PDEs using Element Methods.

The thesis has been performed for SINTEF Energy AS - Gas technology as a part of their work in developing a numerical modelling framework for solving flexible heat exchangers. The aim of the thesis was to develop and test a transient solver in the modelling framework. In TFY4510 Specialization Project (15 sp) the existing steady-state solvers were investigated and slightly expanded.

As the work in this thesis is similar to the project work, some parts in the introduction and theory are very similar to that of the project report. Even though this thesis is not a directly continuation of the project, the earlier familiarization of the framework has been invaluable.

Upon the beginning of the thesis, I was given a derivation of the system of equations as they would appear in the modelling framework. These were proven to contain an error, which I unfortunately did not discover before the beginning of May. In order to account for the entire work process, I have chosen to include the description of some alternative solution approaches that have been implemented during debugging, even though they might have been avoided with the correct problem formulation.

I would like to thank my supervisors, Geir Skaugen (SINTEF) and Morten Hammer (SINTEF), for their help and guidance whenever needed. In addition, I would like to thank Alex Hansen (NTNU) for being my internal supervisor at NTNU.

Trondheim, 16.06.14

Bjørnar Skåre

Contents

Chapter 1

Introduction

1.1 Numerical modelling of heat transfer

Heat exchangers are used in many different kinds of applications; both for cooling and heating purposes. Due to the increased focus of environmental sustainability, it has also become important to look at possibilities of utilizing the energy in "waste heat" or background heat.

There are many demands on the heat exchanging systems such as weight and size. Maximizing profits are also usually desirable. Therefore, it is important to be able to design heat exchangers efficiently. This requires a good understanding of the mechanisms of heat transfer.

Due to the complexity of the heat transfer mechanisms, it has been important to investigate the field of numerical modelling of heat transfer. There are many types of heat exchangers which can be characterized by a number of factors due to geometries (plate exchangers, tubular exchangers, fins), flow natures (cross flows, co- and counter-flows) and thermodynamic working conditions (multi-phase, multi-mediums). Examples of different heat exchanger types are shown in figure 1.1. Skaugen et al. (2012) have implemented a framework for designing robust heat exchangers, which has been the study of this thesis. This modelling framework is called FlexHX, as an abbreviation for Flexible Heat eXchanger.

The framework contains a number of steady-state solvers. This thesis includes the work that has been done in order to extend the framework to transient solvers.

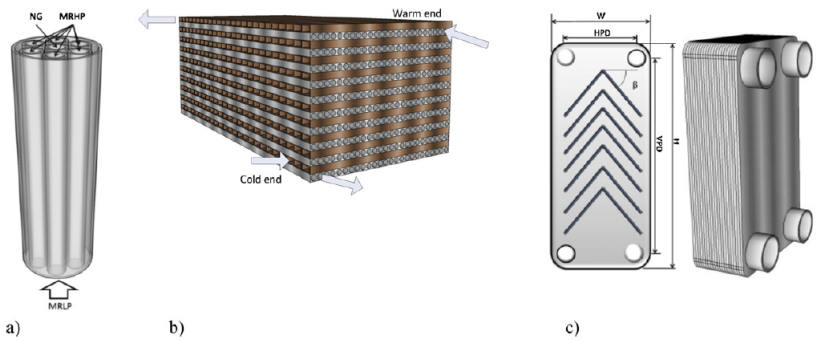

Figure 1.1: *Different heat exchanger types that have been implemented in FlexHX. a) Tube-in-shell exchanger. b) Multi-dimensional cross stream. c) Plate heat exchanger. Figure is taken from Skaugen et al. (2012).*

1.2 Motivation for transient problem formulation

Transient solvers are able to provide more information than steady-state solvers. Interesting phenomena to investigate could be perturbations of steady-state solutions, transient behaviour between two steady-state solutions or transient behaviour during startup and shutdown. This information could be useful in design stages of heat exchangers, i.e. when considering requirements to material properties.

The working conditions of a heat exchanger is often controlled in an outer system according to the total pressure drop within the heat exchanger. Ledinegg (1938) described how unsteady flows could have different mass flows that could lead to the same total pressure drop. Skaugen et al. (2010) related the Ledinegg instability to a two-phase fluid in a heat exchanger, which was expected to be similar to the illustration in figure 1.2. This has also been confirmed by steady-state simulations done by Skaugen et al. (2013). These mass flows represent quite different pressure profiles and heat transfers. Thus, if the heat exchanger is controlled by the total pressure drop, there could be several solutions. In such cases, transient simulations can provide information on whether the solutions are stable and likely obtainable or not.

In general, the pressure drop is increasing with the mass flow, as can be seen in 3.12. Yet, when dealing with complex cases such as phase transitions, this might not be the case. Phase transitions are also very relevant to heat transfer, as they either require large energy uptakes or releases. Therefore, operating heat exchangers under these conditions are desirable. Transient solvers are clearly relevant for designing efficient heat exchangers.

A transient solver must be very robust, as it can be initialized unsteady and required to handle difficult conditions. One of the computationally most demanding calculations is the integration of state variables in a two-phase fluid. Under these circumstances, the derivatives of the state variables will be very high, requiring

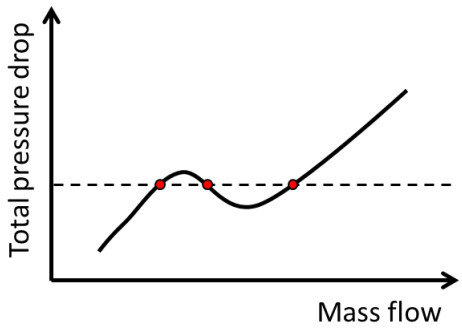

Figure 1.2: *Illustrative figure of the total pressure drop in a two-phase fluid as a function of the mass flow.*

small steps in order to keep the solution accurate and stable.

1.3 Structure of thesis

In this report I will begin by presenting some theory on fluid dynamics, heat transfer mechanisms and differential algebraic equations (DAEs). This will lay the foundation of the numerical problems that need to be solved. Then I will establish the system of equations that are attempted solved, and the reasons for solving the problem in this manner.

Furthermore, I will look at some difficulties I have encountered while developing the solver. Here, some redevelopment of the initial formulation of the problem will be performed. Then, some interesting simulation results that have been obtained are presented. Afterwards, the problem definition and the simulation results are discussed.

Finally I will conclude whether or not the implementation of a good transient solver has been achieved, and propose some further work that needs to be done.

1.4 Nomenclature

Abbreviation	
ABM	Adams-Bashworth Moulton
BDF	Backward Difference Formulae
DAE	Differential-Algebraic Equation
DASSL	Differential-Algebraic System Solver
FlexHX	Flexible Heat eXchanger
MRHP	Mixed Refrigerant High Pressure
MRLP	Mixed Refrigerant Low Pressure

NG	Natural gas
ODE	Ordinary Differential Equation

Symbol	**Units**	**Description**
A	[m^2]	Cross sectional area
A_w	[m^2]	Surface area
c	[m/s]	Velocity of sound
c_f	[]	Skin-friction coefficient
c_p	[J/(kg K)]	Heat capacity, solid
d_h	[m]	Hydraulic diameter, fluid
e	[J/kg]	Specific internal energy
E	[J/kg]	Total energy per unit mass
f_f	[]	Friction factor
g	[m/s^2]	Gravitational acceleration constant
h	[J/kg]	Specific enthalpy
h_c	[$W/(m^2K)$]	Heat transfer coefficient
k	[W/(m K)]	Thermal conductivity
m	[kg]	Mass, fluid
m_w	[kg]	Mass, solid
$\dot{m}$	[kg/s]	Mass flow rate
M_r	[]	Relative molecular mass
q	[J/s]	Heat flux
$\dot{Q}$	[J/s]	Heat rate
p	[Pa]	Pressure
R	[J/(K mol)]	Universal gas constant
R_t	[K/W]	Thermal resistance
t	[s]	Time
T	[K]	Temperature, fluid
T_{sur}	[K]	Surrounding temperature
T_w	[K]	Temperature, solid
u	[m/s]	Velocity, fluid
$\bar{u}$	[m/s]	Mean velocity, fluid
v	[m^3/kg]	Specific volume
V	[m^3]	Volume, fluid
x	[m]	Position
μ	[kg/(m s)]	Dynamic viscosity
ρ	[kg/m^3]	Mass density
σ	[$W/(m^2\ K^4)$]	Stefan-Boltzmann constant
τ_0	[N/m^2]	Shear stress at the wall

Chapter 2

Overview of literature

Bendapudi and Braun (2002) summarize the literature on dynamic heat transfer modelling up until 2002. Investigations of the field of transient modelling began in the late 1970s with Wedekind et al. (1978) as one of the first. The early transient models used very simplified dynamics in order to investigate the dominant transient behaviour. In particular, the momentum equation was replaced by algebraic equations and lumped system models were made. Wedekind et al. (1978, p.98) explains a lumped system as *"when a particular mechanism influencing the phenomenon is very complex, or is not completely understood, it is sometimes possible to formulate a simplified model of the phenomenon which lumps the mechanism's effects into a single, determinable parameter"*. It is also related to the use of averaged quantities, as opposed to the distributed method, and it is still being investigated, i.e. by Tulapurkar and Khandelwal (2010).

Solving the transient momentum equation directly does not seem to have been a priority, though Chi and Didion's (1982) worked on a moving boundary lumped parameter formulation using this. Moving boundary approaches, where the control volumes are divided by the fluid states, seem to be a popular choice also used by Nyers and Stoyan (1994), Xiandong He et al. (1998) and Williatzen et al. (1998). Alternatively, it is possible to use a finite volume method (FVM) as done by Rossi and Braun (1999).

Comparison between the moving boundaries approach and the finite volume method was made by Bendapudi et al. (2004). The conclusion was that the moving boundaries approach was found to be at least twice as fast as the finite volume method. However, there were no remarks on the accuracy of the two methods.

Xuan et al. (2006) developed a transient and steady-state simulation tool for different types of heat exchanger, similar to the framework at hand. They used a combination of finite volume and moving boundaries approach, where the heat exchanger is divided in equal-sized segments with the opportunity of using smaller control volumes when needed. The model would approach pure moving boundary when the segmentation size increases and finite volume if the tolerances for further division of segments are increased.

Chapter 3

Theory

In FlexHX, the heat transfer in question is concerning both solids and fluids in both liquid and vapour phase.

3.1 Transient heat transfer in solids

Incropera et al. (2013) state that heat transfer when considering a control volume, V, in a solid is governed by

$$\frac{dT_w}{dt} = \frac{1}{m_w c_p}\dot{Q}, \tag{3.1}$$

where $\dot{Q}$ is the sum of the heat rates flowing into the control volume. The temperature of the solid, T_w, is a natural observable and state variable. The following subsections will present the different processes of heat exchange or modes which might occur in (3.1), as described by Incropera et al. (2013). As shown by figure 3.1, the contribution to the heat rates may be very different.

3.1.1 Conduction

Conduction is heat transfer in a stationary medium with a temperature gradient. This can be related to high energetic particles colliding with low energetic particles, causing a net energy transfer from the high energetic side to the low energetic side. The heat transfer rate can then be written as

$$\dot{Q}_{cond} = -kA_w\frac{dT_w}{dx}. \tag{3.2}$$

3.1.2 Convection

Convection is heat transfer between two mediums in movement to one another, and is very important in the interactions between a moving fluid and a stationary

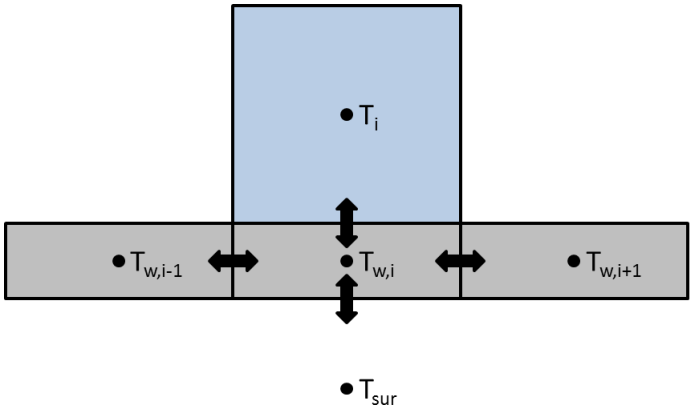

Figure 3.1: *Example of heat rates entering a control volume in a solid.*

solid at different temperatures. In convection processes, there might also occur latent heat exchanges. Latent heat exchange is associated to phase changes, such as condensation of vapour and boiling of fluids, which in some cases are very important in a heat exchanger.

The heat transfer rate for convection is given as

$$\dot{Q}_{conv} = -h_c A_w (T_w - T). \tag{3.3}$$

The heat transfer coefficient, h_c, is dependent on a lot of parameters such as geometry, fluid properties and flow dynamics. Due to the complexity of this quantity, there are a lot of different heat transfer correlations based on different cases. Hesselgreaves (2001) provides a selection of such correlation models.

3.1.3 Thermal radiation

Conduction and convection transfer heat through physical interactions between particles. However, even though there are no physical connection between two mediums, we can still have thermal radiation. This is emitted by any matter with a non-zero temperature. The heat transfer follows the Stefan-Boltzmann law and can be written as

$$\dot{Q}_{rad} = \epsilon \sigma A (T_w^4 - T_{sur}^4), \tag{3.4}$$

where $0 \leq \epsilon \leq 1$ is a factor relating the efficiency to a black body.

Compared to the other modes, radiation will be negligible in a heat exchanger, and is usually not taken into account.

3.1.4 Heat losses

A heat exchanger could also have non-negligible heat losses to the surroundings, also known as ambient losses. This could in general be related to the thermal

resistance concept in Incropera et al. (2013) through

$$\dot{Q}_{amb} = -\frac{T_w - T_{sur}}{R_t}. \tag{3.5}$$

These heat losses are represented in this general way, as they could be due to conduction, convection or radiation depending on the heat exchanger type and the surroundings of the heat exchanger.

3.2 Fluid dynamics and heat transfer

The governing conservation equations for the fluid are, as given in Welty et al. (2008), the Navier-Stokes equations which for a one-dimensional flow are given as

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\frac{\partial \rho u}{\partial x} \\ \frac{\partial \rho u}{\partial t} &= -\frac{\partial}{\partial x}\big(\rho u^2 + p - \frac{4}{3}\mu\frac{\partial u}{\partial x}\big) \\ \frac{\partial \rho E}{\partial t} &= -\frac{\partial}{\partial x}\big(\rho u E + up + q - \frac{4}{3}\mu u\frac{\partial u}{\partial x}\big). \end{aligned} \tag{3.6}$$

If inviscid flow ($\mu = 0$) and zero heat conductivity ($q = 0$) is assumed, the Navier-Stokes equations are simplified to the Euler equations which read

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\frac{\partial \rho u}{\partial x} \\ \frac{\partial \rho u}{\partial t} &= -\frac{\partial}{\partial x}\big(\rho u^2 + p\big) \\ \frac{\partial \rho E}{\partial t} &= -\frac{\partial}{\partial x}\big(\rho u E + up\big). \end{aligned} \tag{3.7}$$

In order to avoid rapid transient behaviour due to pressure wave propagation, it is possible to replace the momentum equation with an algebraic equation related to the pressure drop. The differential variable, u, can be replaced by the algebraic variable $\dot{m}$ through

$$\dot{m} = \rho A u. \tag{3.8}$$

By applying (3.8) and letting (3.7) be valid for a control volume, V, it is possible to do some further investigation of these equations. This will be performed in the following subsections.

3.2.1 Conservation of mass

The conservation of mass for a control volume is given as

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= -\frac{\partial \rho u}{\partial x} \\ &= -\frac{1}{A}\frac{\partial \dot{m}}{\partial x} \\ &= -\frac{1}{A}\frac{\dot{m}_{out} - \dot{m}_{in}}{l} \\ &= \frac{1}{V}(\dot{m}_{in} - \dot{m}_{out}).\end{aligned} \tag{3.9}$$

The cross-sectional area, A, is assumed to be constant.

3.2.2 Conservation of momentum

The pressure drop correlation is expressed by

$$0 = f(\dot{m}, \rho)\Delta x - (p_{out} - p_{in}), \tag{3.10}$$

where f is a function determining the pressure drop $\frac{dp}{dx}$. Incropera et al. (2013) give the function f for a pipe flow as

$$f(u, \rho) = f_D \frac{1}{d_h}\frac{\rho u^2}{2}, \tag{3.11}$$

which also is known as the Darcy-Weisbach equation. This is a phenomenological equation and the Darcy friction factor, f_D, could have further dependencies on the velocity through the Reynolds number. (3.11) could be rewritten as

$$f(\dot{m}, \rho) = f_D \frac{1}{A^2 d_h}\frac{\dot{m}^2}{2\rho} \tag{3.12}$$

in order to relate it to the algebraic mass flow.

The Darcy friction factor is often based on experimental results or validations, and there are a lot of studies devoted to this area. An example is Næss (2010) who investigates the effect of having finned tubes to increase heat transfer.

3.2.3 Conservation of energy

Incropera et al. (2013) defines the total energy, E, per unit mass is defined as the sum of the internal, kinetic and potential energy or

$$E = e + \frac{1}{2}u^2 + gz. \tag{3.13}$$

Thus, the conservation equation for energy can also be expressed through the specific internal energy, e, and putting (3.13) into the energy equation in (3.7) we get

$$\begin{aligned}\frac{\partial \rho(e+\frac{1}{2}u^2+gz)}{\partial t} &= -\frac{\partial}{\partial x}\big(\rho u(e+\frac{1}{2}u^2+gz)+up\big)\\ &= -\frac{\partial}{\partial x}\big(\rho u(h+\frac{1}{2}u^2+gz)\big)\end{aligned} \tag{3.14}$$

where specific enthalpy has been introduced through $h = e + p/\rho$. Including convective heat rates from the walls we obtain

$$\frac{\partial \rho(e+\frac{1}{2}u^2+gz)}{\partial t} = \frac{\partial}{\partial x}\big(-\rho u(h+\frac{1}{2}u^2+gz)+\frac{1}{A}\Sigma_i \dot{Q}_{conv,i}\big) \tag{3.15}$$

By including both kinetic and potential energy in the expressions for specific internal energy and enthalpy and applying the equation on a control volume, V, a differential equation for the specific internal energy can be found as

$$\begin{aligned}\rho\frac{\partial e^*}{\partial t} &= \frac{\partial}{\partial x}(-\rho u h^* + \frac{1}{A}\Sigma_i \dot{Q}_{conv,i}) - e^*\frac{\partial \rho}{\partial t}\\ &= \frac{1}{A}\frac{\partial}{\partial x}(-\dot{m}h^* + \Sigma_i \dot{Q}_{conv,i}) + e^*\frac{1}{A}\frac{\partial \dot{m}}{\partial x}\\ &= \frac{1}{V}\big(\dot{m}_{in}h^*_{in} - \dot{m}_{out}h^*_{out} + \dot{Q}_{conv} + e^*(\dot{m}_{out}-\dot{m}_{in})\big).\end{aligned} \tag{3.16}$$

Here, $\dot{Q}_{conv}$ represents the sum of all heat rates from the walls.

Per definition, e^* is the same as the total energy. However, as this variable is referred to as the internal energy in FlexHX, this report will continue to use this name. For simplicity, e^* and h^* will get the notations e and h, respectively. The reason for this is that the original definitions are no longer used, as it is the modified ones which are implemented in FlexHX.

Hence, the conservation of specific internal energy will be given as

$$\frac{\partial e}{\partial t} = \frac{1}{\rho V}\big(\dot{m}_{in}h_{in} - \dot{m}_{out}h_{out} + \dot{Q}_{conv} + e(\dot{m}_{out}-\dot{m}_{in})\big). \tag{3.17}$$

3.3 Differential- and algebraic equations

3.3.1 Theory

Problem formulations for dynamic processes often contain both differential- and algebraic equations (DAEs). In physics, the differential equations often express the conservation of quantities, while the algebraic equations consist of constitutive equations and rate equations. Examples of these are dependent thermodynamic properties (constitutive) and mass- or heat rates (rate).

In general, a DAE system can be written on the form

$$\mathbf{F}(\mathbf{z}', \mathbf{z}, t) = 0, \quad \mathbf{z}(t_0) = \mathbf{z}_0. \tag{3.18}$$

However, in most physical equations, the differential term, $\mathbf{z}'$, appears explicitly in the equations. This means that we can rewrite (3.18) into

$$\begin{aligned} \mathbf{x}' &= \mathbf{f}(\mathbf{x}, \mathbf{y}, t) \quad \mathbf{x}(t_0) = \mathbf{x}_0 \\ 0 &= \mathbf{g}(\mathbf{x}, \mathbf{y}, t), \end{aligned} \tag{3.19}$$

which is a class known as semi-explicit DAEs. Compared to (3.18), $\mathbf{z} = [\mathbf{x}\ \mathbf{y}]^T$. As we can see from the equations in the previous section, this is exactly the problem that needs to be dealt with. It is also worth noticing that if the algebraic equations in (3.19) were such that they could be solved explicitly for $\mathbf{y}$, i.e.

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, t), \tag{3.20}$$

they could simply be inserted in the differential equations. Then, a set of ordinary differential equations (ODEs) would be obtained. This is often not the case as $\mathbf{g}(\mathbf{x}, \mathbf{y}, t)$ could contain non-linear terms of $\mathbf{y}$.

Systems of DAEs are in general not trivial to solve. In order to find proper solution methods, it is important to investigate the problem thoroughly to see what kind of DAEs it consists of. One way to classify sets of DAEs is to use the concept of index, as done by Moe (1995). The meaning of this concept will be elaborated in the following.

The concept of index is an important classification in terms of solvability. Moe (1995) describes the index, v, as the number of differentiations needed to transform the system into an ODE. An ODE could therefore be classified as an index zero DAE. It could be cumbersome to figure out the index of a set of DAEs through differentiations, but insight in the problem could be of great help.

Moe (1995, p. 22) also provides a heuristic rule which reads: *"If one or several algebraic variables are absent from the algebraic equations, then the model index is two or higher."* The effect of having a set of DAEs of higher index is described as being comparable to stiff differential equations.

Stiffness is a numerical characterization which is difficult to explain. From a programming point of view, Press et al. (2007) describe it as the sensitivity of the step-size in the numerical scheme. An ODE is characterized as stiff if it requires a very small step-size to remain stable. Therefore, it is important to treat higher order index problems in a proper way. One way of doing this is by reducing the index of the problem.

For doing this, Moe (1995) presents some index reducing algorithms. One of them is removing algebraic equations and this algorithm will be explained through

an example. Lets investigate the following system:

$$\begin{aligned} x_1' &= -x_1 + x_2 + y \quad x_1(t_0) = x_{1,0} \\ x_2' &= 2x_1 - x_2 + 2y \quad x_2(t_0) = x_{2,0} \\ 0 &= x_1 + 2x_2. \end{aligned} \tag{3.21}$$

Clearly, this system can be written in the form of (3.19), with $\dim(\mathbf{x}) = 2$ and $\dim(\mathbf{y}) = 1$. Furthermore, the algebraic equation is independent of y from which we deduct that the index of the problem must be higher than one, according to the heuristic rule above.

The procedure is then to differentiate the algebraic equations, i.e.

$$0 = x_1' + 2x_2', \tag{3.22}$$

where we can replace the two terms by the expressions in (3.21), such that we obtain

$$\begin{aligned} 0 &= (-x_1 + x_2 + y) + 2(2x_1 - x_2 + 2y), \\ 0 &= 3x_1 - 2x_2 + 5y. \end{aligned} \tag{3.23}$$

The system has now been differentiated once, and the index has been reduced by one. However, as we can see from (3.23), our algebraic variable y still has not become a differential variable. By repeating the algorithm and differentiating the algebraic equations, (3.23) becomes

$$\begin{aligned} 0 &= 3x_1' - 2x_2' + 5y' \\ 0 &= 3(x_1 + x_2 + y) - 2(2x_1 - x_2 + 2y) + 5y' \\ y' &= \frac{1}{5}(7x_1 - 5x_2 + y). \end{aligned} \tag{3.24}$$

The overall problem can now be represented as an ODE. This means that our original system, (3.21), can be classified as a semi-explicit index two set of DAEs.

It is also possible to see how little it takes for the index to increase. Only by changing the algebraic equation in (3.21) to

$$0 = 2x_1 - x_2,$$

we see that the differentiation and insertion of derivatives will lead to

$$\begin{aligned} 0 &= 2(-x_1 + x_2 + y) - (2x_1 - x_2 + 2y), \\ 0 &= -4x_1 + 3x_2, \end{aligned}$$

which is another algebraic equation independent of y. Thus, the system is at least index three. By continuing the index reduction routine, it can be seen that it in

fact is a semi-explicit index three system. Thus, even though the systems look very much alike, they have different index and the index three system would be a lot stiffer than the index two.

Reducing the index is in general not as straightforward as these simple examples. Furthermore, other complications occur when reducing the index. First of all, it is important that the variables are initialized such that the original system (3.21) is satisfied. If this is not the case, the reduced system is not mathematically equivalent. Secondly, when the system is solved numerically, the solution may drift away from the original algebraic constraints. This is due to error in the numerical integration. Thus, it is important to monitor the original algebraic equations in order to ensure the validity of the solution.

3.3.2 Solution of DAEs

In the previous section some properties of DAEs were introduced. In particular, index and index reduction were presented. The next question is whether or not index reduction is necessary and/or advantageous. Moe (1995, p. 36) presented a figure to illustrate various problems related to the different classes of DAEs, which has been included in figure 3.2.

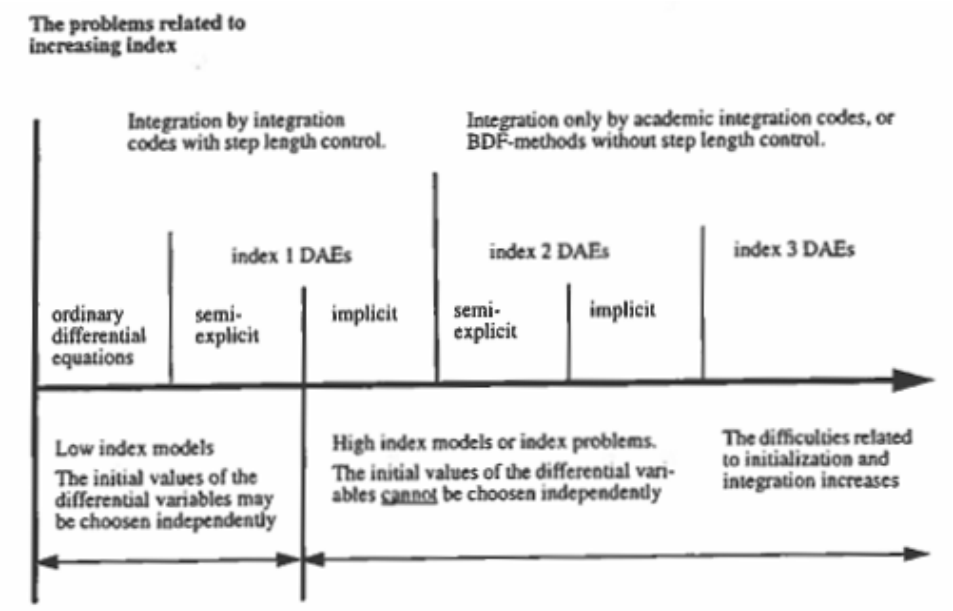

Figure 3.2: *Problems related to increasing index. Copied from Moe (1995).*

Here, we can see the benefits of lower index models. When working with a transient model under difficult conditions, the required step length for stable integration will vary a lot from initialization to steady-state. Thus, being able to use step length control is crucial for computational costs. This means that even

though index reduction may be difficult to perform, it is wise to at least reduce the system of DAEs to an index one problem.

If the problem is reduced to an index zero set of DAEs, basically any ODE solver can be used to perform the integration, as long as the initial conditions are consistent. Index one problems need a solver which is able to solve sets of equations formulated as (3.18). One of the routines available in FlexHX is DASSL written by Petzold (1982). According to Brenan et al. (1989) this routine handles index zero and one, even though it is possible to modify it to solve higher order index problems.

When dealing with semi-explicit index one problems, there is an additional solution algorithm, described by Moe (1995). The origin of this algorithm was to provide modelers with a methodology to formulate and solve dynamic process models with a semi-explicit index one system. The algorithm can be represented by the flow chart in figure 3.3.

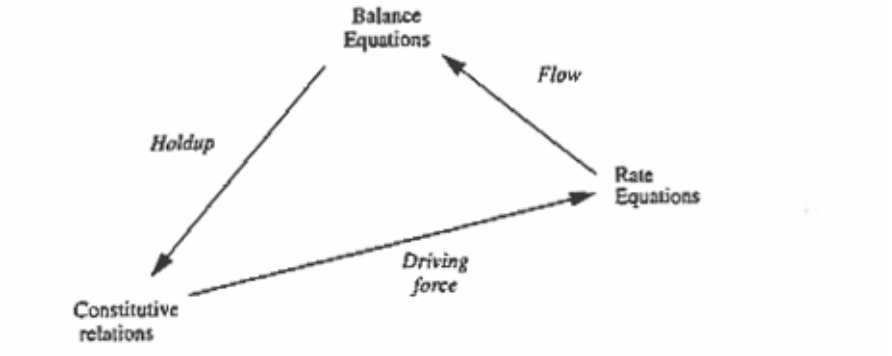

Figure 3.3: *Flow chart explaining an algorithm for solving semi-explicit one index problems. Copied from Moe (1995).*

The algorithm begins with the initialization of the differential variables in the balancing equations, for example the fluid densities from (3.9). After this, all constitutive relations are computed, which could be all thermodynamic properties as function of the differential variables. Then, all the rate equations are computed, in order to determine all heat- and mass fluxes, before the balancing equations are solved and a time step is made.

By using this algorithm, the algebraic and the differential equations are solved separately, meaning that any ODE solver could be used for the integration. Separating the problem in this manner will make the computation of each time step more efficient as we are solving two small systems instead of one large. However, an important simplification has been made by doing this, namely that all constitutive variables and flow rates are considered constant over the integration of the balancing equations. This means that in cases where these quantities are rapidly changing, the time step in the integration must be small. The advantage of solving two systems instead of one may be reduced or even turn out to be a disadvantage.

Therefore it is important to investigate different solution methods.

Chapter 4

FlexHX

4.1 Framework

The modelling framework builds up the heat exchanger geometry by connecting building blocks or elements. Schematically, this looks like figure 4.1.

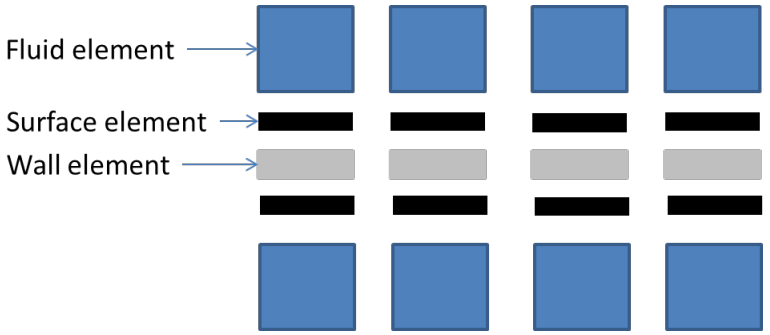

Figure 4.1: *Building blocks in FlexHX.*

In this way, it is possible to create different kinds of heat exchangers whose performances could be investigated.

The heat exchanger that has been the subject of development and testing in this master thesis, is a tube-in-shell with a three-stream flow as shown in figure 4.2. In the small, concentric tube enters natural gas (NG) from above. The other small tubes contain a mixed refrigerant at high pressure and temperature (MRHP) also entering from above, while the outer tube contains a mixed refrigerant at low pressure and temperature (MRLP) coming from below.

This heat exchanger was chosen due to its simple structure, for being thoroughly tested by Skaugen et al. (2012) and for including two-phase fluid states which are demanding when considering computational cost and stability.

A schematic of this heat exchanger as it is represented in FlexHX is shown in figure 4.3. The arrows represents the three flows and the dots are representing

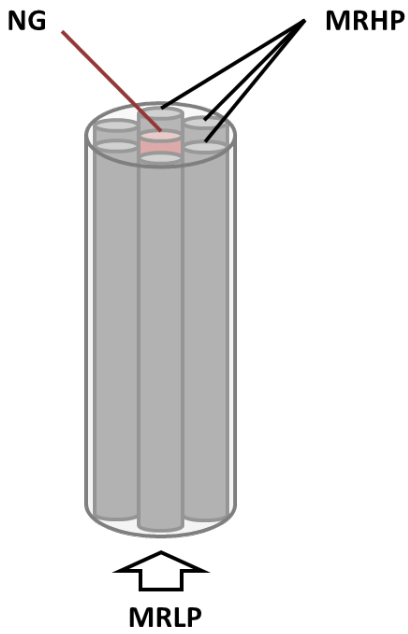

Figure 4.2: *A multi-stream tube-in-shell heat exchanger*

nodes in which the state variables are calculated. Thus, the green nodes are the inlet nodes.

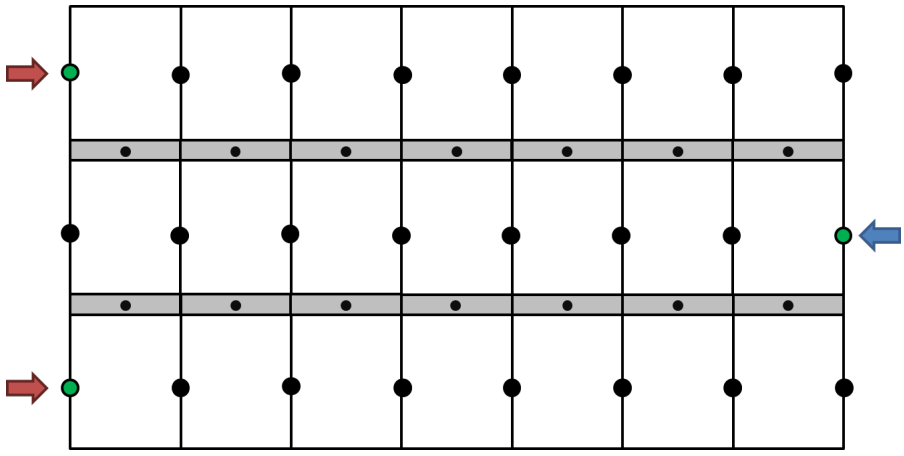

Figure 4.3: *Element division and nodal positioning in FlexHX.*

As we can see, the solid nodes within the walls are centred, while the fluid nodes are positioned at the boundaries.

4.2 Problem formulation

The heat transfer problem can, as we have seen, be formulated in different manners depending on the level of accuracy required. The steady-state solver in FlexHX has been using the algebraic pressure drop correlation. Thus, this approach has been attempted solved in a transient formulation, as it was the least invasive method to incorporate in the framework.

4.2.1 System of equations

Based on the current framework in FlexHX, the natural fluid state variables are either pressure and enthalpy or specific volume, v, and specific internal energy. As the specific volume is the inverse of the mass density, modelling equations were based on these. Thus, the differential equations were chosen to be

$$\begin{aligned}
\frac{dv_i}{dt} &= \frac{d\frac{1}{\rho_i}}{dt} = -\frac{v_i^2}{V}(\dot{m}_{i,in} - \dot{m}_{i,out}) \\
\frac{de_i}{dt} &= \frac{v_i}{V}\big(-\dot{Q}_{conv} + \\
&\quad \dot{m}_{i,in}h_{i-1} - \dot{m}_{i,out}h_i + e_i(\dot{m}_{i,out} - \dot{m}_{i,in})\big) \\
\frac{dT_{w,i}}{dt} &= \frac{1}{m_{w,i}c_p}\dot{Q}_i,
\end{aligned} \tag{4.1}$$

where subscript i represents element number i. The convective heat rate, $\dot{Q}_{conv}$, is defined positive for heat flowing into the walls, hence the minus sign in the energy equation. Finally, the volume of the elements is considered to be constant. Moving boundaries will be discussed later on.

In addition we have the algebraic pressure drop correlation given in (3.10). Clearly, this system of equations is a set of DAEs, where the variables can be classified as differential variables $\mathbf{x} = [\mathbf{v}\ \mathbf{e}\ \mathbf{T}_w]^T$ and algebraic variables $\mathbf{y} = \dot{\mathbf{m}}$.

Furthermore, we notice that it is a semi-explicit set of DAEs. In order to determine the index of the problem, it is also necessary to look at the boundary conditions.

4.2.2 Boundary conditions

Depending on the boundary conditions, this system could be index one or index two. Two boundary condition options and their advantages and disadvantages will be discussed.

Option 1

A natural imposition of boundary conditions is to define all inlet conditions of the streams, i.e inlet mass flow, specific volume and energy. The inlet state of the fluid could of course be given in terms of pressure and temperature, which are more

common variables to observe and control. v and e could then be determined by using constitutive relations.

If all inlet conditions are specified, the result of splitting the heat exchanger into N number of elements is that N outlet mass flows are becoming variables constrained by N algebraic pressure drop constraints. Only by investigating figure 4.3, we could argue that this system is an index one set of DAEs. As the inlet mass flows at the green nodes are given, it is clear that the pressure drop in the first element will be a function of the outlet mass flow and that this is the only algebraic variable it will be dependent of. Thus, by differentiating (3.10) for this element, one would obtain

$$0 = \Delta z \frac{\partial f}{\partial \dot{m}_{1,out}} \dot{m}'_{1,out} + \left(\Delta z \frac{\partial f}{\partial v_1} - \frac{\partial p_{1,out}}{\partial v_1} \right) v'_1 + \left(\Delta z \frac{\partial f}{\partial e_1} - \frac{\partial p_{1,out}}{\partial e_1} \right) e'_1. \quad (4.2)$$

After inserting the continuity equations for v and e from (4.1), it is obvious that (4.2) becomes a differential expression for the mass flow out of the first element. The same differentiation for the second element will give a differential expression for the second mass flow. The second element will also contain the inlet mass flow as a differential variable, but now we can use (4.2) for this expression. Obviously,

$$\dot{m}'_{i,out} = \dot{m}'_{i+1,in}. \quad (4.3)$$

After differentiating all of the algebraic equations, all algebraic variables can be expressed as differential variables. Therefore, the index of the system is one.

Option 2

However, heat exchangers often do not have a constant inlet mass flow. When put into a system, the inlet mass flow is regulated in order to maintain a constant pressure drop throughout the heat exchanger. In other words, the inlet mass flow becomes an algebraic variable and the constraint

$$0 = p_N - p(v_N, e_N), \quad (4.4)$$

is added.

The crucial point of this algebraic constraint is that it is independent of all of the algebraic variables. By using the heuristic rule presented by Moe (1995), the index should be larger than one.

By differentiating (4.4) we obtain

$$\begin{aligned}
0 =& \frac{\partial p}{\partial v_N} v'_N + \frac{\partial p}{\partial e_N} e'_N \\
=& -\frac{\partial p}{\partial v_N} \frac{v_N^2}{V} (\dot{m}_{N,in} - \dot{m}_{N,out}) + \\
& \frac{\partial p}{\partial e_N} \frac{v_N}{V} \big(-\dot{Q}_{N,conv} + h_{N-1}\dot{m}_{N,in} - h_N \dot{m}_{N,out} + e_N(\dot{m}_{N,out} - \dot{m}_{N,in}) \big) \qquad (4.5) \\
=& -\frac{\partial p}{\partial v_N} v_N (\dot{m}_{N,in} - \dot{m}_{N,out}) + \\
& \frac{\partial p}{\partial e_N} \big(-\dot{Q}_{N,conv} + h_{N-1}\dot{m}_{N,in} - h_N \dot{m}_{N,out} + e_N(\dot{m}_{N,out} - \dot{m}_{N,in}) \big),
\end{aligned}$$

which is another algebraic equation and the index is definitely larger than one. This algebraic equation is however dependent on the inlet and outlet mass flow of the last element. Without doing any thorough investigation and proof, it is quite safe to assume that a combination of the differentiation of (4.5) and the pressure drop for the last element would provide two differential equations for these mass flows.

Thus, with the rest of the argument being the same as for option 1 boundary conditions, one additional differentiation would suffice, and this would be a semi-explicit index two set of DAEs.

As an extra conviction, the pressure drop in an element could be assumed to be a function of the inlet mass flow only. The differentiations of the algebraic pressure drops are then related to differential equations for all of the inlet mass flows. The differential equations for the outlet mass flows of the last elements is the given by the differentiation of (4.5).

As option 2 is the most realistic set of boundary conditions, this was attempted solved. The problem formulation is then a semi-explicit index two set of DAEs.

Chapter 5

Proposed solution algorithm

High index DAEs (index ≥ 2) can, as shown in figure 3.2, not be solved by integrators with step length control, such as DASSL, without modification and need to be index-reduced in order to be used. When solving the proposed problem in section 4.2, the algebraic constraint in (4.5) was used instead of (4.4). The problem is then formulated as a semi-explicit index one system. Thus, it could also be possible that the outlet pressure drifts away from its set value, p_N.

5.1 Alternative 1: Semi-explicit algorithm

When the problem is described by a semi-explicit index one model, the algorithm described in section 3.3.2 proposed by Moe (1995) can be used.

5.1.1 Initialization

After performing an index reduction it is important to ensure that the problem is initialized consistent. In the present case, this means that the initial v_N and e_N must fulfil p_N. The inlet condition is usually given in terms of p and T or h, so in addition of fixing v_0 and e_0 such that these are true, the outlet condition is the only requirement of the initialization.

As the constraints on the initialization are on p and T or h, it seems natural to initialize profiles on these quantities for the fluid, in addition to temperature profiles within the wall. After determining these initial profiles, it is possible to use thermodynamic relations to find v and e which are the differential variables that are needed.

The initial profiles that were attempted were a linear profile for the pressure between p_0 and p_N combined with either a linear profile for the temperature or the enthalpy.

As the framework already contained several steady-state solvers, the steady-state solutions were known. This means that the outlet value of T or h could be appropriately set, in order not to have the problem too difficult to solve numerically.

In addition, an initialization where the profiles were very close to steady-state were implemented, in order to see if the solver could maintain the steady-state solution.

The initial wall temperature profile could either be set linearly between two pre-set temperatures, or as the mean of the surrounding temperatures. Examples of the possible initializations are given in figure 5.1 and 5.2.

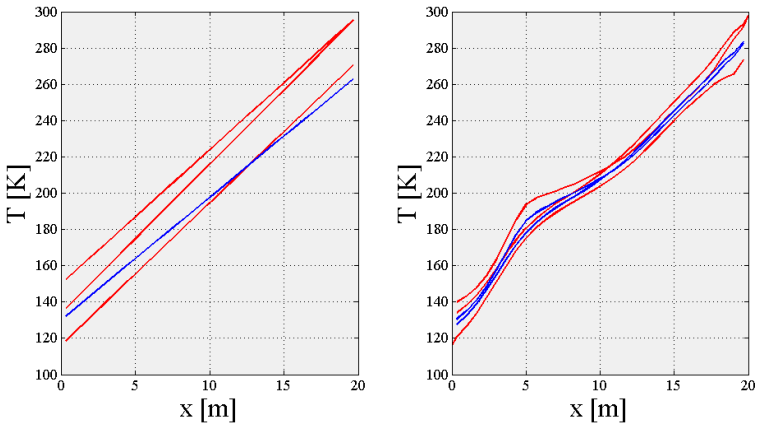

Figure 5.1: *Two possible initializations of the temperature profiles. To the left is both the fluid temperature profile in red and wall temperature profile in blue linear. To the right is the fluid temperature initialized near steady-state and wall temperature is the mean between its surrounding fluids.*

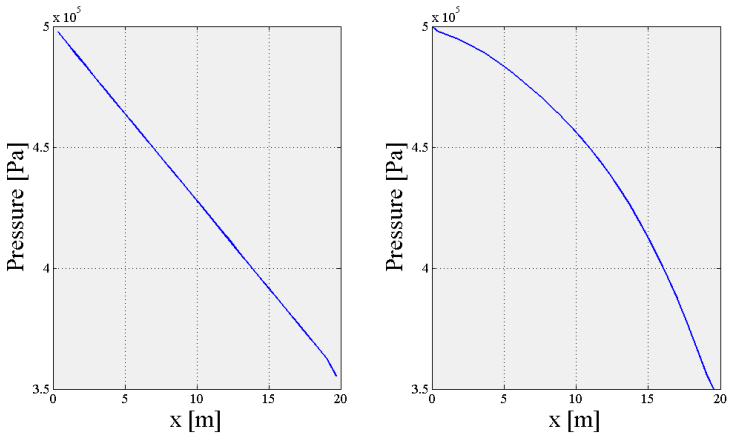

Figure 5.2: *Possible initializations of the pressure profile for MRLP. To the left is a linear pressure profile, while the steady-state profile is given to the right.*

5.1.2 Solution method

After the initial profiles have been set, the solver iterates through the circuits in order to calculate the constitutive relations and rate equations according to figure 3.3. These two steps are rather interconnected, as the necessary constitutive relations to determine the rates are calculated as the solver step through the circuits.

The circuits are calculated from outlet to inlet, opposite of the stream direction. This is due to the algebraic constraint on the outlet pressure. Thus, a set of two implicit equations must be solved to determine the inlet and the outlet mass flow of the last element, namely

$$
\begin{aligned}
0 =& f(\dot{m}_{N,in}, \dot{m}_{N,out}, \rho)\Delta x - (p_N - p_{N-1}) \quad \text{and} \\
0 =& -\frac{\partial p}{\partial v_N} v_N \big(\dot{m}_{N,in} - \dot{m}_{N,out}\big) + \\
& \frac{\partial p}{\partial e_N}\big(-\dot{Q}_{N,conv} + h_{N-1}\dot{m}_{N,in} - h_N \dot{m}_{N,out} + e_N(\dot{m}_{N,out} - \dot{m}_{N,in})\big).
\end{aligned}
$$

The algebraic variables, which are solved implicitly, are also shown in figure 5.3.

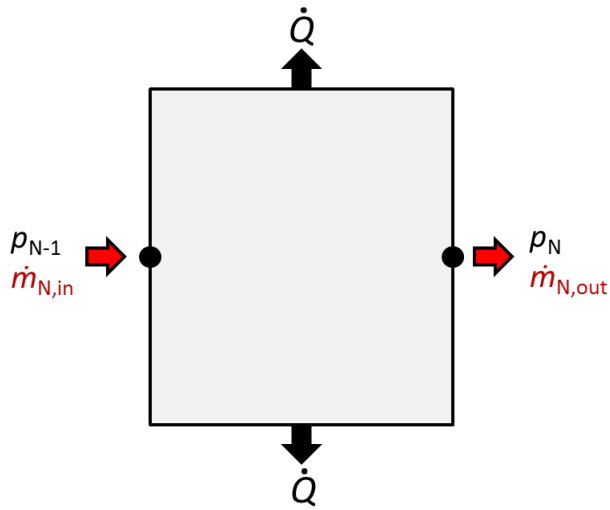

Figure 5.3: *Solving the last element.*

When the rate equations of the last element is known, the previous elements are implicitly solved only for the inlet mass flows. The outlet mass flow is known, as shown in figure 5.4, as $\dot{m}_{N-1,out} = \dot{m}_{N,in}$.

The convective heat rates, $\dot{Q}$, are also calculated while iterating through the circuits. However, these are solved explicitly of the differential and algebraic variables, i.e. $\dot{Q} = f(\dot{m}, v, e, T_w)$, and do not pose any numerical challenges. The pressure drop function in (3.12) has a non-linear dependency on the mass flow. In addition,

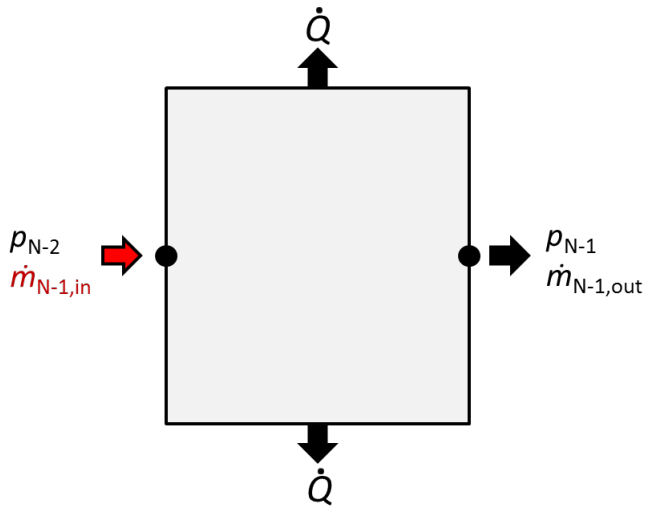

Figure 5.4: *Solving the next to last element.*

the Darcy friction factor, f_D, is usually dependent on the Reynolds number. This causes the non-linearity to be dependent on the form of the friction factor used. A non-linear equation solver is then needed to calculate the mass flows.

When all mass flows and heat rates have been determined, the balancing equations in (4.1) might be evaluated by any ODE solver. Afterwards, a step is made and the evaluation of all flows are repeated.

By dividing each circuit in N number of elements, the number of differential equations for the investigated heat exchanger with three circuits and two walls will be

v:		$3N$
e:	$+$	$3N$
T_w:	$+$	$2N$
Total:	$=$	$8N$.

For each time step, it is also required to solve the $3(N+1)$ implicit, algebraic equations due to the pressure drops and boundary conditions. Thus, the differential and algebraic equations are decoupled.

5.2 Alternative 2: DAE solver

An alternative to the semi-explicit algorithm is to use a commercial DAE solver. As shown in figure 3.2, index zero and one sets of DAEs can be solved by solvers with step length control. If the original semi-explicit index two problem is to be attempted, a solver with fixed step length must be used.

DASSL, that was mentioned in section 3.3.2, is a solver that could be used to solve the index reduced problem with (4.5) as the boundary condition constraint.

In order to use DASSL, all mass flows need to be included as variables in the solver. Generally, DASSL solves the system

$$0 = \mathbf{g}(\mathbf{y}', \mathbf{y}, t), \tag{5.1}$$

where $\mathbf{y} = [\mathbf{v}\ \mathbf{e}\ \mathbf{T_w}\ \dot{\mathbf{m}}]^T$ will be the variables solving the DAE system

$$\begin{aligned}
0 =& \frac{dv_i}{dt} + \frac{v_i^2}{V}(\dot{m}_{i,in} - \dot{m}_{i,out}) \\
0 =& \frac{de_i}{dt} - \frac{v_i}{V}\big(-\dot{Q}_{conv} + h_{i-1}\dot{m}_{i,in} - h_i\dot{m}_{i,out} + e_i(\dot{m}_{i,out} - \dot{m}_{i,in})\big) \\
0 =& \frac{dT_{w,i}}{dt} - \frac{1}{m_{w,i}c_p}\dot{Q}_i \\
0 =& f(\dot{m}, \rho, e)\Delta x - (p_i - p_{i-1}) \\
0 =& \frac{\partial p}{\partial v_N} v_N(\dot{m}_{N,in} - \dot{m}_{N,out}) + \\
& \frac{\partial p}{\partial e_N}\big(-\dot{Q}_{N,conv} + h_{N-1}\dot{m}_{N,in} - h_N\dot{m}_{N,out} + e_N(\dot{m}_{N,out} - \dot{m}_{N,in})\big).
\end{aligned}$$

The total number of equations that are required to be solved simultaneously is

v:		$3N$
e:	+	$3N$
T_w:	+	$2N$
$\dot{m}$:	+	$3(N+1)$
Total:	=	$11N+3$.

5.3 Comparison

Solving the set of DAEs directly will increase the size of the system. Hence, it will be more demanding to solve it compared to the decoupled problem in alternative 1 computationally. In particular, an explicit ODE solver without the need of solving a large set of non-linear equations will be a lot more efficient, but also less stable.

The gain of handling the set of DAEs directly, is that it is more accurate and stable. A challenge with alternative 1 is that when the problem is decoupled, the mass- and heat flows may become inconsistent. During the ODE integration, the mass- and heat flows will be considered constant, which may be a harsh approximation under unsteady conditions. Furthermore, the allowed time steps of the integration should at least not exceed the time it will take for the fluid to cross half the element, i.e.

$$\Delta t_{max} = \frac{1}{2}\frac{l}{u} = \frac{1}{2}\frac{lA\rho}{\dot{m}} = \frac{1}{2}\frac{\rho V}{\dot{m}}. \tag{5.2}$$

This constraint is based on the Courant-Friedrichs-Lewy condition described by Smith (2005).

An important factor for early implementations and debugging is to have a high level of control, and not leave too much of the calculations to black box routines. Therefore, alternative 1 is preferred as a starting point.

Chapter 6

Troubleshooting

During the development process, a lot of numerical difficulties have been encountered. Even though some of these may be directly related to the original, false derivation of the system of equations, they are very descriptive of the work process and alternative methods that have been attempted. Thus, a chapter has been devoted to these problems and the solutions that were implemented.

6.1 Consistency of solution

6.1.1 Grid

As we can see from figure 4.3, the fluid nodes are placed at the edge of the elements. This creates some concerns regarding the consistency of the solution.

FlexHX is based on solving the circuits element-wise, and the convective heat rates are also summed element-wise. However, the control volume approach in the transient problem formulation in (4.1) requires the nodes to be centred. Thus, in order to have consistency, the control volume would not coincide with the element volume, as shown in figure 6.1.

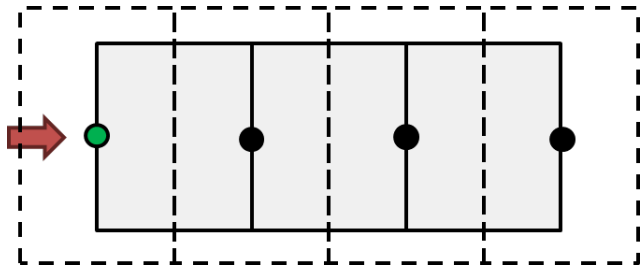

Figure 6.1: *Consistent control volumes represented by dashed lines.*

Unless the internal structure of the heat calculations is changed, the convective heat rates would have to be the mean of the two neighbouring elements, such that

a fine integration of these would be wasted. However, the mass flows are calculated from the pressure drop and the pressure is known by a profile only determined by the values at the nodes. Therefore, it would seem that dividing the fluid elements would lead to inaccuracies. In order to improve the accuracy of the solution, one would have to increase the number of elements instead of dividing the elements further.

Due to more pressing issues with the code, the consistency of the solution has not been investigated thoroughly. The implementation has assumed the control volume to be the element volume, such that the position of the fluid nodes in effect are centred in the fluid elements. This is shown in figure 6.2.

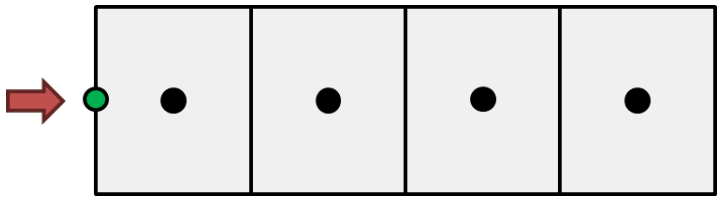

Figure 6.2: *Nodal positioning in the transient solver.*

6.1.2 Mass flow

When the mass flow is to be determined in order to fulfil the pressure drop in an element, it would be preferable to evaluate the pressure drop in (3.10) in the center of the element as shown in figure 6.3.

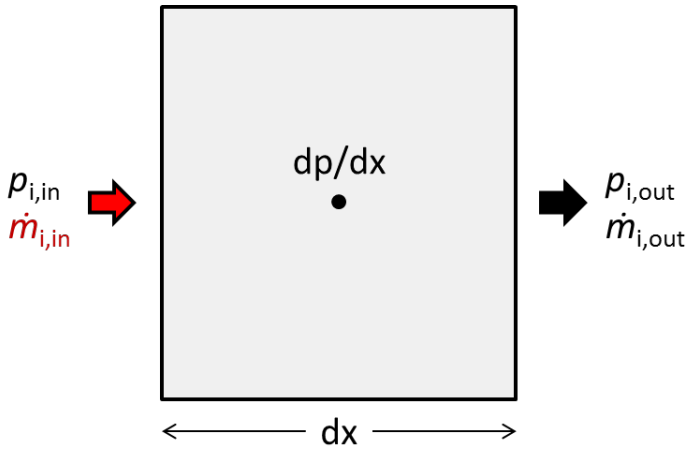

Figure 6.3: *Consistent pressure drop evaluation within an element.*

This evaluation proved to be problematic, as it resulted in large fluctuations in the mass flow. This was due to the boundary condition evaluation of the last

element. When maintaining a constant outlet pressure during very unsteady conditions, the inlet and outlet mass flow of the last element may differ a lot. This difference will then be transferred through the circuit as shown in figure 6.4.

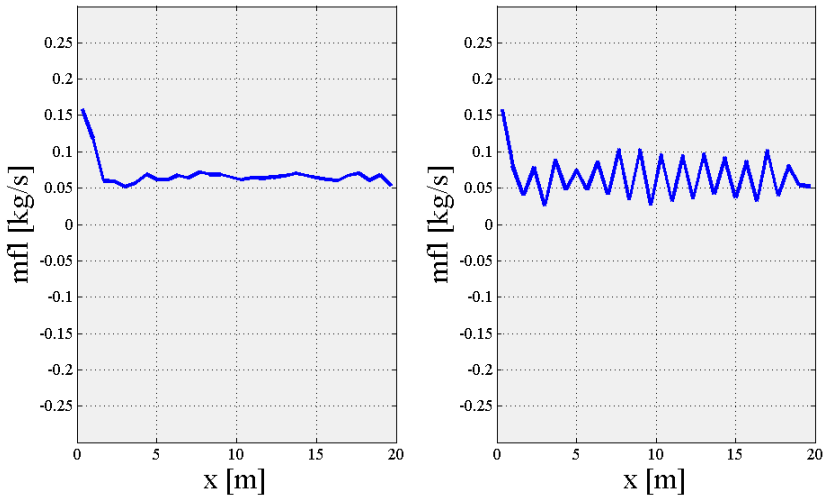

Figure 6.4: *Example of unstable mass flow profile. The mean mass flows in the elements are shown to the left, while the mass flows at inlets and outlets are plotted to the right.*

The mean values of the mass flows which are evaluated in the pressure drop behave smoothly, while the mass flows on the borders fluctuate. These fluctuations seemed to cause instabilities in the solution. Even though the mass flow differences do not seem alarmingly high in figure 6.4, the volumes of the elements in the investigated heat exchanger could be $\sim 10^{-6}\mathrm{m}^3$, causing high derivatives for the specific volume and energy in (4.1).

An alternative, consistent evaluation is to use the mean of the pressure drops at the inlet and outlet, but this also lead to similar, fluctuating behaviour.

The pressure drop evaluation was therefore changed to be at the inlet of the element, making it dependent only on the inlet mass flow. This choice made it possible to iterate through the circuits from inlet to outlet. It also made it possible to calculate outlet mass flow of the last element explicitly. By rearranging (4.5) we obtain

$$\dot{m}_{N,out} = \frac{-\frac{\partial p}{\partial v_N} v_N \dot{m}_{N,in} + \frac{\partial p}{\partial e_N}\left(-\dot{Q}_{N,conv} + (h_{N,in} - e_N)\dot{m}_{N,in}\right)}{-\frac{\partial p}{\partial v_N} v_N + \frac{\partial p}{\partial e_N}(h_{N,out} - e_N)}. \tag{6.1}$$

6.1.3 Special last element

When the fluid nodes are centred in the elements, as shown in figure 6.2, the last node will not be at the outlet of the heat exchanger. Applying (6.1) to the last element will fix the pressure at the node and not at the outlet. This could create large differences when solving the heat exchanger with different number of elements.

One way of trying to reduce this error is to solve the last element in two steps. The inlet mass flow is still calculated to fulfil the total pressure drop of the whole element. An additional mass flow, $\dot{m}_{add}$, close to the outlet is also solved for the pressure drop, and used in (6.1) along with $\dot{Q}_{add}$. This approach should fix the pressure at the center of this small control volume instead of the entire element.

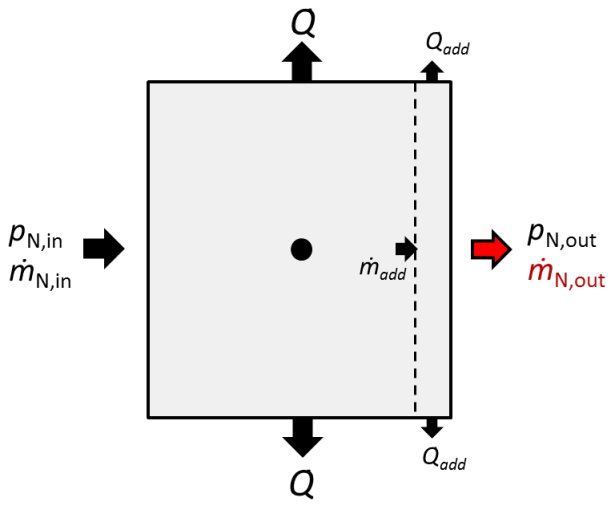

Figure 6.5: *Solving the last element in two steps by introducing an additional control volume.*

The approach was implemented and tested, but did not seem to have a large impact and was later discarded. An argument against this approach is the need of extrapolating the state variables which may be inaccurate, especially if it is a two-phase fluid.

6.2 Differential variables

6.2.1 uvflash

The thermodynamic constitutive relations are calculated by an external thermodynamic library, tplib. In this library, the temperature/enthalpy/entropy and pressure are seemingly the preferred state variables. In order to calculate the constitutive relations using the specific internal energy and volume, there is a function

named uvflash()[1] that solves the constitutive relations by calling to a temperature/enthalpy/entropy and pressure function iteratively.

Thus, using uvflash() will both be more computationally demanding and contain a risk of instability compared to using temperature/enthalpy/entropy and pressure. As few calls to uvflash() as possible is therefore desirable.

6.2.2 ph-profiles

One way of reducing the need of uvflash() is to use profiles for the pressure and enthalpy instead of the energy and specific volume. Whenever there is a need to obtain state variables at other positions than the nodes, these are found by interpolation. So, even though it seems natural to use specific internal energy and volume profiles, it is better to use pressure and enthalpy.

In addition, as the mass flows are determined from pressure drop correlations, one might only need the pressure for some calculations, hence avoiding a call to the thermodynamic library. Finer division than the element has been advised against in section 6.1.1, though. By using profiles for pressure and enthalpy, the need for uvflash() is kept at a minimum, only needed when a time step has been made and for calculating $\partial p/de$ and $\partial p/dv$ in (4.5) at the outlet. In these cases, fairly good initial approximations for p and h are also available.

6.2.3 ρ instead of v

A final remark on the differential variables is the use of ρ instead of v. As seen in section 3.2, ρ is the variable which appears naturally. v was implemented due to its existing position in FlexHX.

However, there is an advantage of using ρ as the differential variable instead of v. When using commercially available integrators with variable step-size, the error tolerance is given by absolute and relative errors. When solving two-phase fluids, the liquid phase will have a high density compared to the vapour phase. Relative error control will be desirable, as the liquid phase is less sensitive to density changes than the vapour phase.

The situation is opposite for v, and relative error integration can not be utilized. Thus, in order to be able to integrate with relative tolerances, ρ was reinstated as the integration variable and (3.9) was used instead of the conservation of specific volume in (4.1). In practise, there will be no complications as ρ and v are easily interchanged, i.e.

$$\frac{\partial p}{\partial \rho} = \frac{\partial p}{\partial (1/v)} = -v^2 \frac{\partial p}{\partial v}. \tag{6.2}$$

[1] In the thermodynamic library, u denotes the specific internal energy, e.

6.3 Boundary condition

The option two boundary condition has proven to be difficult to handle. Some of the difficulties observed will be presented below.

6.3.1 Solving set of equations

Before the decision of evaluating the pressure drop at the beginning of the element, as described in section 6.1.2, the mass flows in and out of the last element was solved as shown in figure 5.3. The non-linear dependency of the mass flow in the pressure drop correlation required the use of a non-linear equation solver. This was initially attempted solved using DNSQE by Hiebert (1993), which is a non-linear equation solver based on a modification of the Powell Hybrid method.

DNSQE turned out to have a very poor convergence rate of the problem. It appeared that the pressure drop was a slowly varying function of the mass flows while the algebraic boundary condition was rapidly changing. Hiebert (1993) describes that DNSQE *"can become trapped in a region where the minimum does not correspond to a zero of the system"*, which seemed to be the case as it constantly tried to fulfil the boundary condition first.

As the pressure drop correlation was using the mean mass flow, the solution became to use a secant method to solve this equation first, and then use DNSQE to do the small alterations in order to fulfil the boundary condition as well. The secant method is an approximation to Newton's method to solve $g(x) = 0$, when $g'(x)$ is unknown. Kreyszig (2006) gives it as

$$x_{n+1} = x_n - g(x_n)\frac{x_n - x_{n-1}}{g(x_n) - g(x_{n-1})}. \tag{6.3}$$

In fact, the secant method proved to converge so fast that it was used to solve all pressure drop correlations for the mass flows.

6.3.2 Two-phase at outlet

There are mainly two reasons for the rapidly changing boundary condition. The first is due the initialization of the problem. Having a more or less random initialization, the balancing equations might be very unbalanced, causing large fluctuations when making small perturbations on the mass flows. This is a transient behaviour which will fade away after some time steps.

The second is due to high values of $\partial p/\partial e$ and $\partial p/\partial v$. Especially when the fluid is in the middle of a phase transition, small changes in the specific internal energy and volume or mass density may have large effect on the phase compositions and the pressure. Avoiding that the outlet pressure drifts away from its original set value p_N during the time integration is therefore easier said than done.

Figure 6.6 show that the outlet pressure drifts rapidly at the early stages after the initialization. The deviation of the boundary condition was more severe for the natural gas than for the other two streams. Clearly, this follows from the fact

that the natural gas exits the heat exchanger in the middle of a phase change. The largely varying state derivatives make it difficult to maintain a constant pressure.

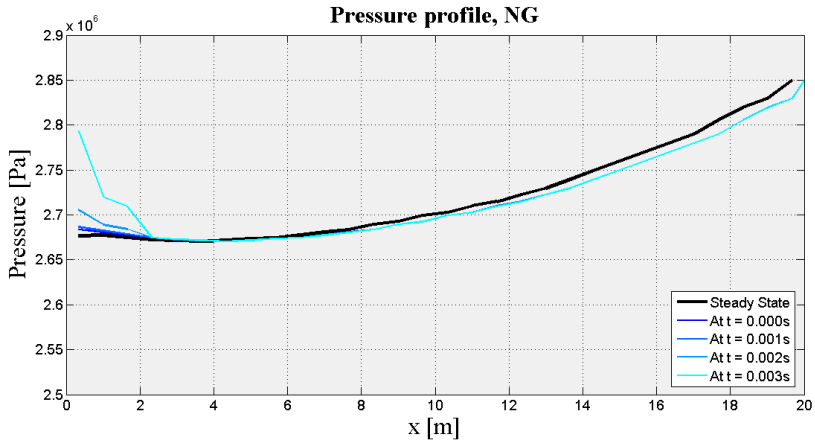

Figure 6.6: *Pressure profile for the stream containing natural gas. The direction of the stream is from right to left.*

A function trying to prevent this from happening was implemented. After each time step, the function calculates the outlet pressure and enthalpy from the new e and v or ρ. It then sets the outlet pressure manually to the originally set p_N, and recalculates the state variables using p_N and h.

This approach violates the conservation equations of mass and energy and was meant as a temporary solution in order to validate the rest of the solver. At small time steps, the correction made is quite small. The function is also set to print an error message when making a relative correction on p larger than 1%. It has been noticed that the mass flows near the outlet of the stream with natural gas have had small, but everlasting, fluctuations. These fluctuations appear as the solution approaches steady-state, and is probably caused by the extra constraint on the outlet pressure.

6.3.3 $\partial p/\partial e$ and $\partial p/\partial v$

Analytical evaluations of the state derivatives are possible to obtain from the thermodynamic library. However, as they may change rapidly, it might be desirable to use forward differencing in their calculation. Forward differencing originate from the definition of the derivative and is given, as done by Kreyszig (2006), as

$$\frac{\partial p}{\partial e} \approx \frac{p(v, e + \Delta e) - p(v, e)}{\Delta e}. \tag{6.4}$$

If the changes Δe and Δv that are going to be made on the current time step can be estimated well, this would probably make it easier to avoid drifting of the outlet pressure.

Figure 6.7 and 6.8 show the evaluations of the derivatives at the outlet of the circuits near steady-state. These figures show a number of interesting phenomena and difficulties worth discussing. First of all, at the outlet of the circuits, the cold stream is still in a liquid phase, while the hot streams are in the transition between being in a vapour phase or two-phase. As can be seen on the y-scales of the plots, the pressure gradients are smaller in a liquid phase.

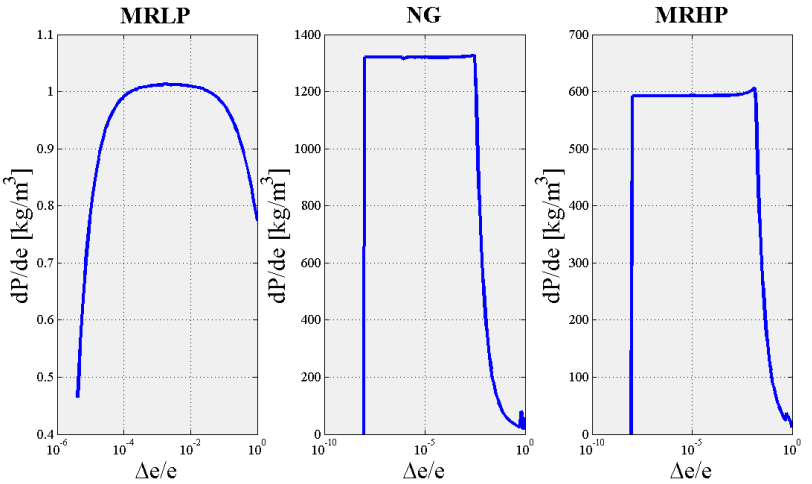

Figure 6.7: *dp/de estimated by forward differencing as a function of the relative energy change.*

Second of all, $\partial p/\partial v$ is $\sim 10^8$ larger than $\partial p/\partial e$. Thus, the pressure is more sensitive to mass density fluctuations than energy. Furthermore, the shape of the derivatives are seemingly non-physical as Δe and Δv become too small or too large. If they are too small, numerical errors will occur as the change in the pressure in (6.4) will be small and we divide by a very small number. In the opposite scenario, letting Δe and Δv becoming too large will provide a harsh approximation to the derivative. In figure 6.7 and 6.8 are the calculations continued until the relative change is one, or $\Delta e = e$. This will naturally be too much, and the integrator should not attempt steps this large.

A final remark on these plots are the opposite signs of $\partial p/\partial e$ and $\partial p/\partial v$. Without making a thorough analysis on them, their dependence can be related to the ideal gas law which is given as

$$p = \rho \frac{R}{M_r} T \tag{6.5}$$

by Kay and Nedderman (1985).

According to (6.5) the signs can be deduced. As the temperature increases with energy, the pressure will also increase and $\partial p/\partial e > 0$. The specific volume is the inverse of the mass density, such that an increase Δv would lead to a decrease $-\Delta\rho$. This would in return decrease the pressure, hence $\partial p/\partial v < 0$. The fluids

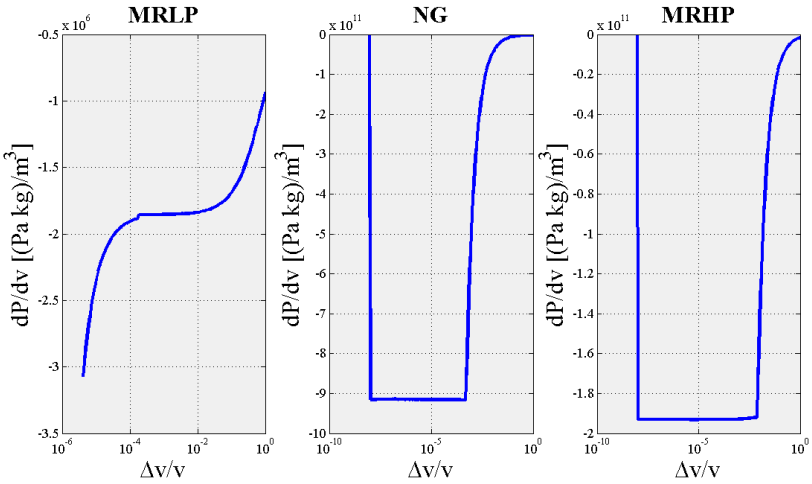

Figure 6.8: *dp/dv estimated by forward differencing as a function of the relative specific volume change.*

do not follow the ideal gas equation, so this nothing but a simplified, qualitative observation.

The results in figure 6.7 and figure 6.8 show that the state derivatives may change when taking non-infinitesimal steps. Thus, it could be more correct to use forward differencing. However, they should not be used uncritically. Small steps should be handled by analytically obtained derivatives and large steps should be avoided. Simulations have shown a tendency of taking small steps, such that the analytical derivatives probably would represent a safe approach. Due to difficulties in obtaining these from the thermodynamic library, they have not been tested.

Currently, the code uses only forward differencing.

6.3.4 Alternative index reduction

The index reduction algorithm that was used in section 4.2.2 lead to drifting of the outlet pressure as shown in figure 6.6. However, there are other possible reduction algorithms that will prevent this. A similar reduction algorithm presented by Moe (1995) is removal of differential equations.

When using removal of differential equations, both the original constraint

$$0 = p_N - p(v_N, e_N)$$

and the differentiated constraint

$$0 = -\frac{\partial p}{\partial v_N} v_N (\dot{m}_{N,in} - \dot{m}_{N,out}) + \frac{\partial p}{\partial e_N} \left(-\dot{Q}_{N,conv} + h_{N,in}\dot{m}_{N,in} - h_{N,out}\dot{m}_{N,out} + e_N(\dot{m}_{N,out} - \dot{m}_{N,in})\right)$$

are used. This can be done by omitting one of the differential equations for the last element. As shown in figure 6.9, the last element will have three algebraic variables which must be solved. In addition to the two boundary condition constraints, the pressure drop within the element is still present.

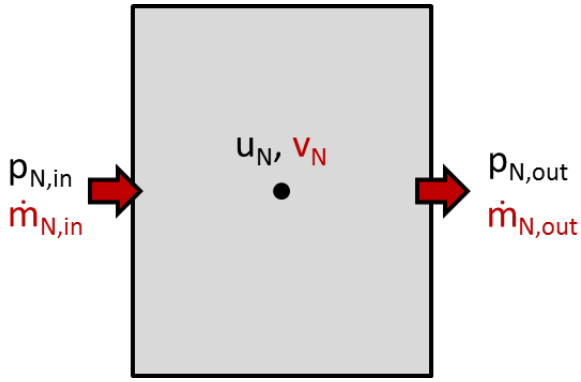

Figure 6.9: *Algebraic variables within the last element, when using removal of differential equations, are marked in red color.*

Here, the specific volume has been chosen as the extra algebraic variable. The specific volume can be determined by the first boundary condition constraint, as it is the only algebraic variable in the equation. By relating the pressure drop in the element to the inlet mass flow only, the inlet mass flow is found. This is again to avoid fluctuations in the mass flow profiles. Finally, the outlet mass flow is determined by the second boundary condition constraint.

This results in two implicit and one explicit equations. Both of the implicit equations have been solved by the secant method given in 6.3.

This approach might violate the conservation of mass within the last element. This is a concern not discussed by Moe (1995). Instead, another positive feature of this algorithm is emphasized; namely that the number of differential variables is the same as the number of dynamic freedoms. In practise, this means that there are no concerns regarding the initialization in order for the problems to be mathematically equivalent. The outlet pressure will always remain constant as the specific volume has become an algebraic variable. However, it is important that the problem is initialized such that it computationally is not too rigorous and still represent a physical system. Otherwise, non-physical fluctuations might occur and they may cause the solution to diverge.

As the two different derivations of the option 2 boundary condition result in different solutions, they will be denoted option 2.1 and option 2.2. Option 2.2 is the latter approach where one differential equation has been removed.

6.3.5 Option 1 boundary condition

As convergence problems were experienced, the option 2 boundary condition was thought to be part of the reason. Moe (1995) described that high index is compa-

rable to stiff ODEs, which may lead to instabilities. Before the second reduction algorithm was introduced, an option 1 boundary condition was also considered to prevent the drifting of the solutions.

The option 1 boundary conditions, as described in section 4.2.2, was implemented to see if there were noticeable changes. The inlet mass flow was fixed and the outlet mass flows of the elements were determined to fulfil the current pressure drops. These changes did not improve the solvability. They made it very dependent on the initialization, as the mass flows near the inlet would have to be very close. If the initial profiles on the pressure and temperature were chosen a little "arbitrarily", fluctuations in the mass flows would occur during the first time steps. Similar problems as described in section 6.1.2 were then observed.

As it is the option 2 boundary condition which is desirable, no extensive work was done to try and make this approach work.

6.4 DAE solver

After experiencing fluctuating mass flow profiles, as shown in figure 6.4, the alternative solution method described in section 5.2 was implemented. By solving the problem as a set of DAEs directly, the algebraic variables are preserved during the integration. A DAE solver should therefore be able to prevent mass flow fluctuations better.

However, the implementation of DASSL constantly failed after a few time steps. The error was that the absolute tolerances could not be met with the minimum step lengths. Increasing the tolerances only allowed the solution to diverge a little more before the same error occurred. These discoveries led to the belief that there was something wrong with the system of equations. Ultimately, the following investigations revealed an error in the theoretical derivation of the system equations.

Fixing this error did not cause an immediate effect on the DAE solver. Due to limited time the original solution method was prioritized, leaving further investigations of the direct DAE approach for later work.

6.5 Physical model

After the code was considered to be working properly and the theory to be correct, the physics of the test case were investigated. The multi-stream heat exchanger that has been tested is shown in figure 4.2. In the simulations, values for geometrical properties and working conditions were set by an existing base case in FlexHX. As the geometries were checked, the tubes stood out by having small diameters. Figure 6.10 shows the cross section of the heat exchanger.

In a steady-state solver, the size of the tubes are less critical than for a transient solver. In a transient solver the control volumes appear explicitly in the conservation equations given in (4.1). Small control volumes will lead to high time derivatives. From (4.1) it can also be seen that the explicit dependency on V disappear as the solution approaches steady-state ($\frac{d}{dt} \to 0$).

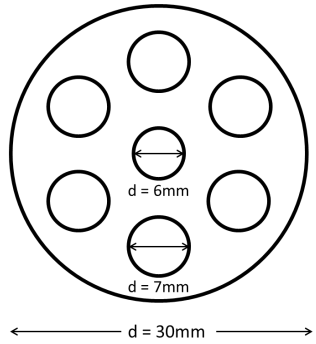

Figure 6.10: *Cross section and tube diameters of the base case.*

The physical reason why small volumes become a problem is due to the initialization. In order for the solver to be robust, it must be able to handle unsteady initializations. These will cause large energy and mass flows within the heat exchanger. To preserve the energy and mass, some control volumes may be required to have large uptakes of these quantities. When the control volumes are small, these uptakes may change the physics drastically and cause instabilities in the numerical solution.

Thus, larger tubes and control volumes should be less dependent on the initialization to maintain stable and need to be investigated.

Chapter 7

Simulation results

The troubleshooting in Chapter 6 gives rise to a number of possible simulation cases, including:

- Solution method: Semi-Explicit/DAE.
- Initialization: Steady-state/Linear temperature and pressure.
- Boundary condition: Option 1/Option 2.1/Option 2.2.
- Tube sizes.
- Number of elements.

In addition, the semi-explicit method could be solved by various ODE integrators. The ones that have been used are an own implementation of the forward Euler as described by Press et al. (2007) and two commercial integrators. These are based on the Adams-Bashworth-Moulton method (ABM) and Backward differentiation formula (BDF) and are described by Watts and Shampine (1993a) and Watts and Shampine (1993b), respectively. ABM is specialized for semi-stiff problems, while BDF is for stiff.

As it turned out, the transient problem formulation is quite stiff. In order for the forward Euler method to be stable, the time steps had to be $\sim 10^{-6}$-10^{-8}s. Since each time step uses around one second, a simulation time of one second would take 10-15 days. The forward Euler is an explicit integration method which is not suited to handle stiff problems, so the commercial integrators are much more efficient. However, due to time-consuming simulations, few good simulation results have been obtained.

Instead of having an inadequate comparison between all of the above-mentioned criteria, a selection of the most interesting results will be provided. Furthermore, the DAE solution method and option 1 boundary condition have not been updated and verified, and will be left out of the simulation comparison. The number of elements that has been investigated has been fixed to 30, as higher numbers have been found to require very small time steps. Lastly, the tube sizes that have been

investigated are given as amplifications of the base case. This means that ”2x base case” imply that all tube diameters shown in figure 6.10 has been multiplied by a factor two.

7.1 Test case 1: Initialization problem

As mentioned in section 6.5, small tube sizes in the heat exchanger cause problems. The test case in Table 7.1 exemplifies this.

Table 7.1: *Test case 1 parameters.*

Integrator:	Forward Euler
Initialization:	Steady-state
Boundary condition:	Option 2.1
Size:	1x base case

In figure 7.1 is the pressure in MRHP just after initialization shown. Even though the case is initialized very close to steady-state, the pressure profile drifts away from the solution. This is caused by small disturbances that result in a large deviation, due to the small tube sizes.

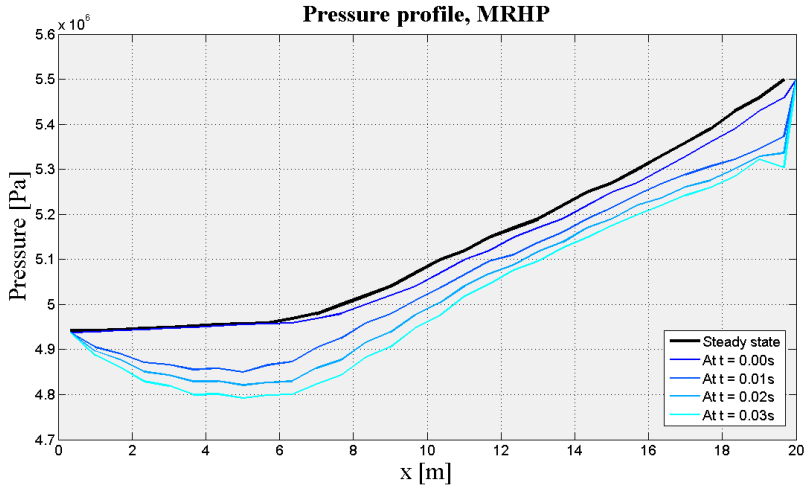

Figure 7.1: *Pressure profile for the stream containing natural gas. The direction of the stream is from right to left.*

As these disturbances have passed, the pressure return towards the steady-state profile, as shown in figure 7.2.

However, these initial fluctuations have caused some large errors in the temperature profile, as shown in figure 7.3. These fluctuations also occurred using ABM

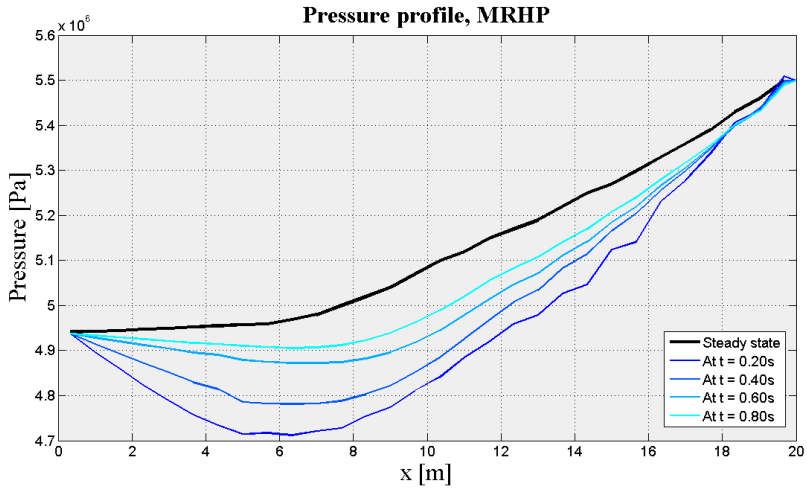

Figure 7.2: *Pressure profile for the stream containing natural gas. The direction of the stream is from right to left.*

as the integrator. They are probably related to the stiffness of the problem. The use of the BDF integrator was also attempted, but it failed due to too small time steps.

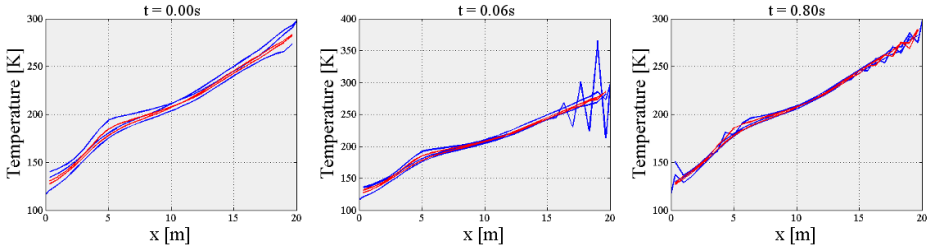

Figure 7.3: *The temperature profiles at initialization, after the initial disturbance and after the initial disturbance have faded.*

As the solution gets close to steady-state again, new fluctuations occur. These are thought to be a combination of the option 2.1 boundary condition and its extra constraint, and the stiffness of the problem.

Altogether, it seems that the heat exchanger sizes of the base case make the numerical problem too stiff to handle using the semi-explicit algorithm. The base case was attempted solved using option 2.2 boundary conditions as well, but this did not make the solution stable either.

7.2 Test case 2: Stability problems

In order to reduce the stiffness of the problem the tube diameters were increased, and the second test case exemplifies the stiffness problem. The parameters used are given in Table 7.2.

Table 7.2: *Test case 2 parameters.*

Integrator:	ABM/BDF
Initialization:	Linear
Boundary condition:	Option 2.2
Size:	12x base case

Increasing the diameters by a factor three was also attempted. However, this caused the BDF solver to fail and ABM to give non-physical solutions with stiffness warnings.

At first, when trying to solve test case two using the ABM solver, the integration tolerances were too low, and the solution diverged. After restricting the tolerances further stiffness warnings reoccurred. Furthermore, the solution that was obtained soon became non-physical, as shown in figure 7.4.

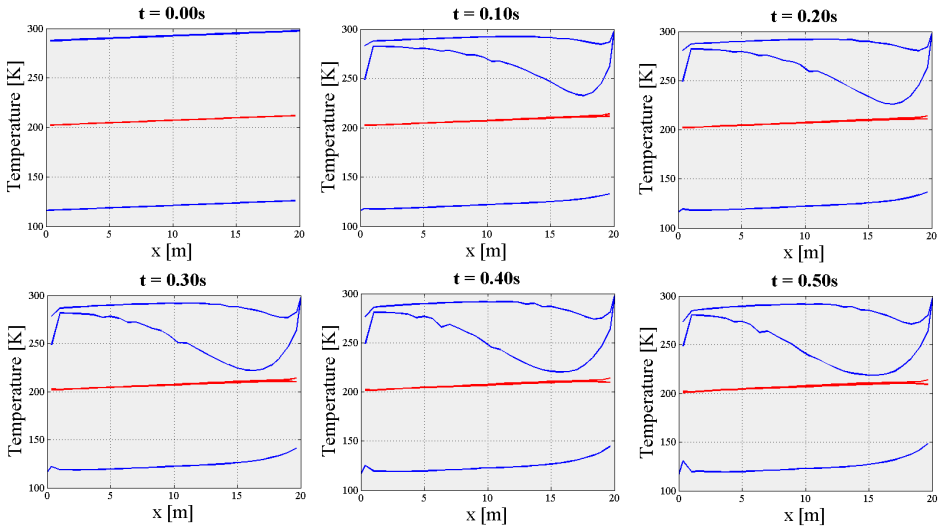

Figure 7.4: *The temperature profiles of the streams in blue and walls in red. The inlet of the hot streams are on the right hand side, while the cold stream enters from the left.*

First of all, the immediate temperature drops near the inlet of the hot streams are suspicious. The only way for these to appear physically is if there is an instant

back-flow from the center of the heat exchanger. This was not the case. Secondly, there are fluctuations appearing near the outlet of one of the hot streams and near the inlet of the cold stream which are clearly non-physical.

Apparently, the test case was too stiff for the ABM solver to handle, such that BDF was attempted instead. However, in order to keep the solution stable, the BDF solver spent too much time to get any proper results. In six days the solver managed to simulate 0.002s, which is shown in figure 7.5.

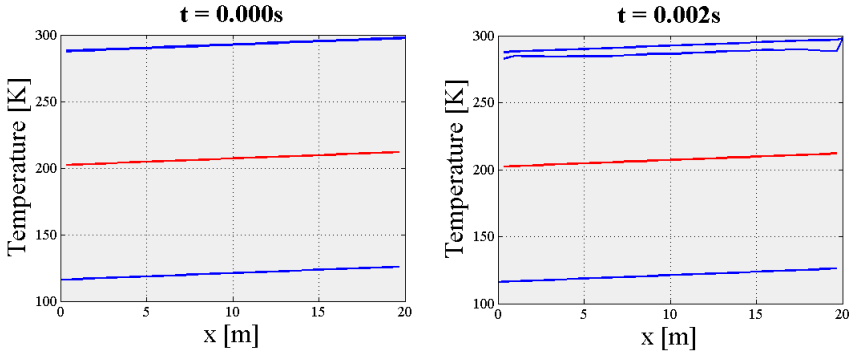

Figure 7.5: *The temperature profiles when using the BDF solver.*

7.3 Test case 3: Convergence

Evidently, the problem needed to be less stiff to get any proper results. The diameters were increased by a factor of 30, as shown in Table 7.3.

Table 7.3: *Test case 3 parameters.*

Integrator:	ABM
Initialization:	Linear
Boundary condition:	Option 2.2
Size:	30x base case

In this case, no stiffness warnings were given by the ABM integrator. Furthermore, the solution converged quickly as shown in figure 7.6.

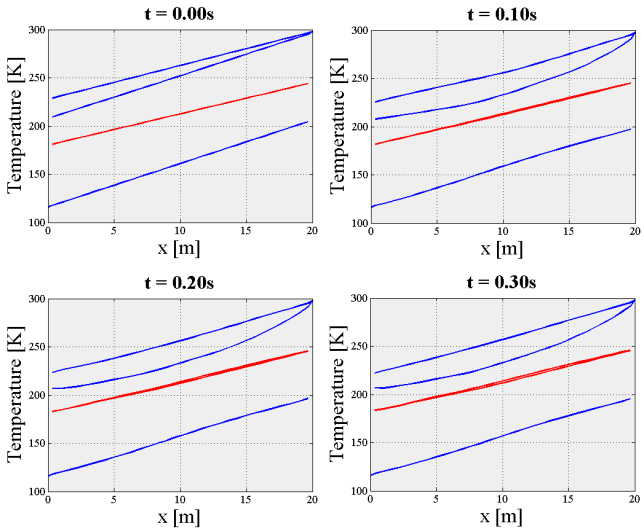

Figure 7.6: *The temperature profiles for the test case in Table 7.3 with the initialization close to steady-state.*

When initializing the heat exchanger far from steady-state, as shown in figure 7.7, the solution converges slower and is still far from steady-state in the last plot. In addition, figure 7.7 displays the different time scales in the transient behaviour of the fluid compared to the solid. The solid temperature changes are slower than the fluid. In fact, when taking a closer look, the cold stream is "overheated" at t

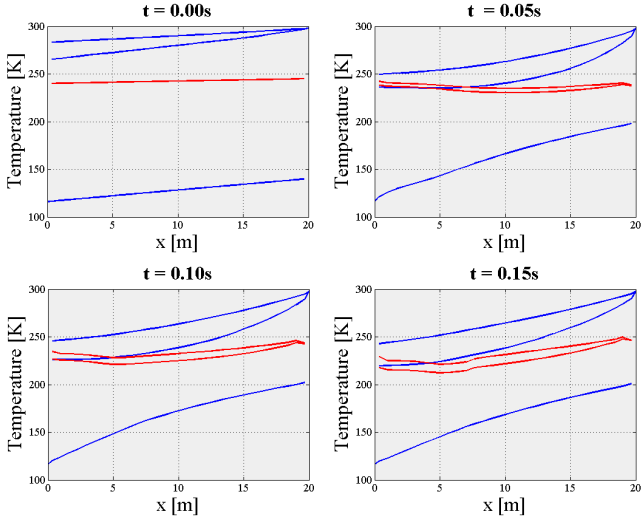

Figure 7.7: *The temperature profiles for the test case in Table 7.3 with the initialization far from steady-state.*

= 0.10s before it cools off again.

However, a serious issue with these results is that the steady-state solver provides a different result. The steady-state simulation shows that the temperature profiles with the same conditions should converge to the ones in figure 7.8.

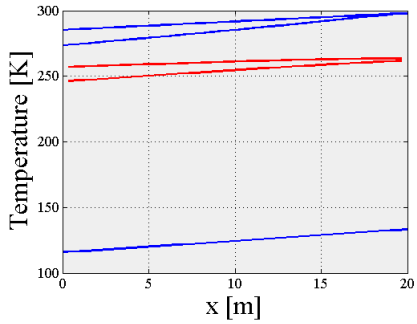

Figure 7.8: *The steady-state temperature profiles for the test case in Table 7.3.*

Obviously, figure 7.8 differs a lot from figure 7.6. Furthermore, the initialization in figure 7.7 is quite close to the steady-state solution. Nevertheless, it does not converge to the steady-state simulation.

This deviation is caused by one of two reasons. The first is that there is an error in the implementation or in the theory. Secondly, it could be that this test case represents the phenomena that was mentioned in Section 1.2 and illustrated in figure 1.2; namely that there is more than one possible solution. By comparing the pressure profiles from the two simulation results, it can be easier to conclude what the explanation is.

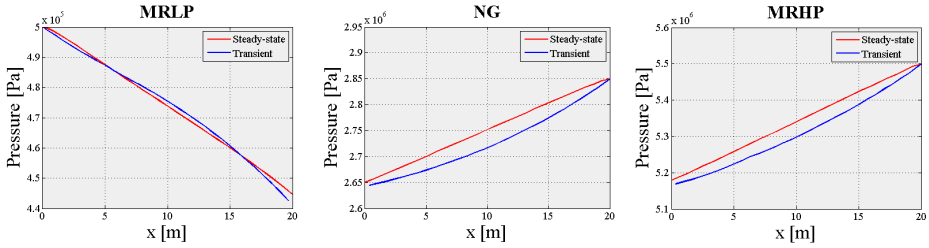

Figure 7.9: *Comparison of the pressure profiles between the steady-state and transient simulations.*

As can be seen from figure 7.9, the pressure profiles in the hot fluids differs. However, they do not represent totally different shapes, making it difficult to be entirely sure that they represent two different solutions. Furthermore, the mass flow profiles of the transient solution still have not reached an equilibrium. For this reason, steady-state simulations can not be run with the final mass flows of the transient solution. It could also be that the slowly-varying temperature profiles

in the solids change, such that the transient solution converges to a different state.

7.4 Test case 4: Steady-state solution

In order to verify that the implementation and theory were correct, an additional test case was initiated. Its specifications are given in Table 7.4.

Table 7.4: *Test case 4 parameters.*

Integrator:	Euler
Initialization:	Steady-state
Boundary condition:	Option 2.2
Size:	30x base case

The mass flows in this test case were lowered compared to test case 3 in order to avoid several possible steady-state solutions. In addition, this leads to an increase in the total heat transfer such that it is more similar to the performance of a desired heat exchanger.

The simulation results in figure 7.10 show only minor changes in the solution and does not indicate that it is going to drift. However, it has only simulated 0.03s and there is a possibility that it will drift in the long run.

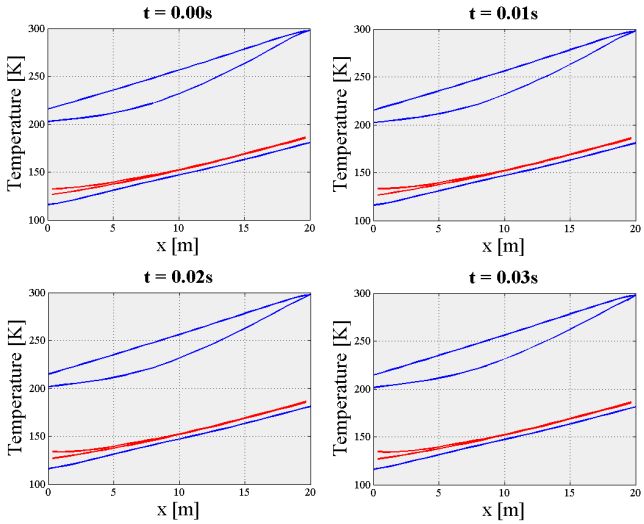

Figure 7.10: *The temperature profiles for the test case in Table 7.4, when initialized in steady-state.*

Chapter 8

Discussion

8.1 Transient solver for acquiring steady-state solutions

One of the interesting things to look at when implementing a transient solver is whether or not it can compete with a steady-state solver for obtaining the steady-state solution. An advantage of a good transient solver is that it is able to find stable steady-state solutions. Recalling from the introduction, steady-state simulations have shown that there might be more than one possible solution that result in the same total pressure drop. If nested simulations are performed with the steady-state solver to find the solution with a given pressure drop, it might end up in an unstable solution. In this case, unstable means that a small, physical disturbance in the heat exchanger will push it towards another steady-state solution. This is also exemplified by the traditional ball analogy which is unstable on the top of the hill and stable in the bottom, as shown in figure 8.1.

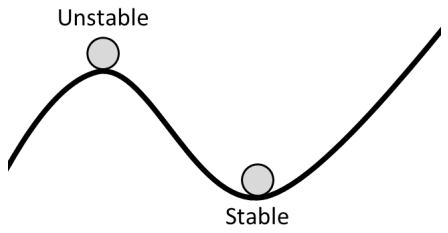

Figure 8.1: *Example of stable and unstable equilibrium conditions, in terms of potential energy.*

This is also what seems to cause the deviation between the steady-state and the transient simulation in Section 7.3, even though it is not certain.

The problem with a transient solver is of course related to efficiency. As the

simulation results have shown, the transient solver that has been implemented requires days in order to simulate a second. Depending on the initialization and the difficulty of the problem solved, the simulation could require several seconds before approaching steady-state. In the current state of the transient solver it is therefore easy to conclude that using it for steady-state simulations is out of the question.

Furthermore, when solving heat exchangers in optimization modes, the differences will accumulate. Thus, the transient solver is particularly not appropriate for optimizations. Another problem with the transient solver, is that it becomes too stiff for compact heat exchangers. The transient formulation is very dependent on the control volumes, such that the steady-state solvers are more robust and converge for more cases.

However, a transient formulation contains a lot more information with respect to the behaviour of the heat exchanger when approaching steady-state. The solution needs to be integrated accurately in order to maintain the function of the solver. Consequently, it requires more thermodynamic calls to obtain physical properties along the way. So even though the current implementation of the transient solver is not the most efficient of codes, it will probably never be an appropriate substitution to the steady-state solvers.

On the other hand, transient solvers can be an important supplement to steady-state solvers. After finding a steady-state solution of a problem, the transient solver can be used to investigate whether or not the solution is stable, and the effects of disturbances. How the behaviour of the solution is from one steady-state to another, i.e. due to changes in the inlet conditions, and during start-up and shut-down are of course other possible uses.

8.2 Conservation of momentum

The main reason for introducing the pressure drop correlation in section 3.2 was to avoid rapid changing pressure waves, which would require small time steps. However, as we have seen in the simulation results, the time steps needed for the solution to be stable are quite small.

Thus, it is interesting to investigate whether or not the full Euler equations in (3.7) could be solved. If this is the case, the problem with option 1 boundary conditions would become an explicit ODE. Using option 2 boundary conditions the problem would possibly still be a semi-explicit index 2, at least if the differential variables are unchanged. However, derivations would be required to be certain.

A condition for stability when solving partial differential equations by finite differences is the Courant-Friedrichs-Lewy (CFL) condition. Weisstein (2014) describes it as *"the time step must be kept small enough so that information has enough time to propagate through the space discretization"*. In other words it can be described as

$$\left(\frac{|u|\Delta t}{\Delta x}\right)_{max} < 1. \tag{8.1}$$

When applied to a heat exchanger problem, Smith (2005) formulates this condition as

$$\Delta t \leq \frac{0.5\Delta x}{|u| + c}, \tag{8.2}$$

where c is the velocity of sound as this is the speed of mechanical (pressure) disturbances. The factor of 0.5 is to ensure that the condition is fulfilled in case of back flows.

Doing a quick estimation of this criteria based on the base case with 30 elements[1] lead to

$$\Delta t \leq \frac{0.5 \cdot 0.20m}{30m/s + 2000m/s} \approx 5 \cdot 10^{-5}s. \tag{8.3}$$

Thus, solving the Euler equations directly could be an option.

8.3 Algebraic pressure drop

Another important factor when comparing the momentum equation and the algebraic pressure drop, is the validity of the latter approach. Even though the original Darcy-Weisbach equation was said to be a phenomenological equation, it has later been supported by theoretical derivations.

Kay and Nedderman (1985) investigate a fully developed, steady-state pipe flow. By using the skin-friction coefficient for a fully developed flow, which is defined as

$$c_f = \frac{\tau_0}{\frac{1}{2}\rho\bar{u}^2}, \tag{8.4}$$

the forces in the fluid can be balanced. $\bar{u}$ denotes the velocity averaged over the cross section. Steady-state condition assumes the cross sectional velocity profile to be independent of the longitudinal coordinate, x.

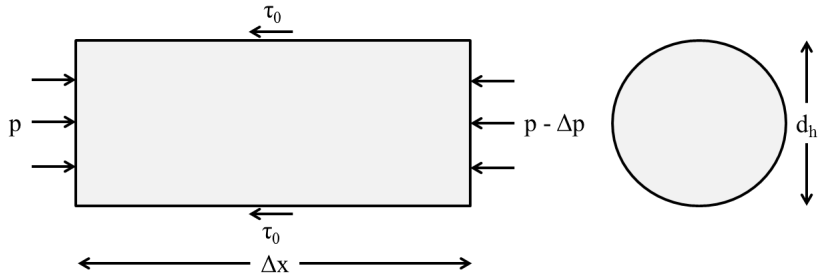

Figure 8.2: *Force balance for steady flow in a pipe. To the left is the pipe seen from the side, while the cross section is shown to the right.*

[1]The speed of sound has conservatively been set to 2000 m/s due to `http://www.engineeringtoolbox.com/sound-speed-liquids-d_715.html`

The only forces acting on this flow are the force due to pressure and the friction from the walls, as shown in figure 8.2. According to Kay and Nedderman (1985), the force balance is given by

$$\pi\left(\frac{d_h}{2}\right)^2 \Delta p = 2\pi\left(\frac{d_h}{2}\right)\Delta x \tau_0. \tag{8.5}$$

Using (8.4) and rearranging lead to

$$\frac{\Delta p}{\Delta x} = 2c_f \frac{1}{d_h}\rho \bar{u}^2, \tag{8.6}$$

which is the same as the Darcy-Weinbach equation if $4c_f = f_D$. (8.6) is also known as Fanning's equation.

In order to obtain (8.6) the flow was assumed to be fully developed and in steady-state. This also implies that any acceleration is out of the question. When initialized unsteady, these approximations cause discontinuous velocity profiles. In addition, the dependence on the velocity is non-linear so that the mean velocity will not suffice.

The extent of the error made by using these assumptions is no known. However, this concern supports the use of the momentum equation.

8.4 Dynamic grid

FlexHX divides the heat exchangers into elements and in this thesis the control volume applied to the system of equations has been the same size. It is however possible, as done by Bendapudi et al. (2004), to combine a finite volume method with a moving boundary approach by allowing the solver to divide the elements into several control volumes. Even though this has been found to improve the efficiency of the solver, there has been no attempt to modify the library into such a formulation.

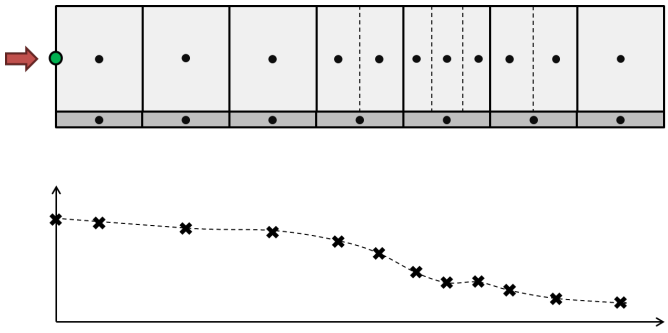

Figure 8.3: *Example of an approach with dynamic control volumes. Below we see a profile of a fluid property, i.e. the pressure, and how the profile may have large changes over small distances.*

The crucial point when dividing some elements into several control volumes, is that the distribution of the state variables must be known accurately. In other words, a larger number of nodes must be available in areas where rapid fluctuations occur, as shown in figure 8.3. As the positions of critical fluid states may change over time, it is important that these nodes are movable. In this formulation, it would be the fluid integration that would determine how many and the position of the nodes. The system of equations would then be dynamically sized.

Modifications of this could of course be made by fixing the nodes for a period of time, and then do a re-evaluation using the moving boundary approach.

Another feature of this approach is to create one small control volume at the end of the last element by default. Doing this will improve the boundary condition, even when the heat exchanger is solved by a small number of elements.

Chapter 9

Conclusion

Implementing a transient solver in the existing framework has been challenging. Even though the current state of the solver is believed to be working, it has not been validated properly due to time consuming simulations.

The long simulation times are related to the problem formulation which has proved to be very stiff. An alternative formulation by solving the momentum equation directly is thought to be less stiff. In particular, by using the option 1 boundary condition, the problem would become a set of explicit ODEs.

One of the main reasons that cause the problem to be stiff is the size of the heat exchanger. In a steady-state solver, the size is of little importance. However, as can be seen in the system of equations in (4.1), the inverse of the control volumes appear explicitly in the conservation equations. When the control volumes are small, the derivatives become large and the solution become very dependent on the time steps. Hence, the problem is stiff.

Increasing the sizes of the heat exchanger will reduce the stiffness of the problem and make the solution more stable. However, this also reduces heat transfer. Maximizing heat transfer and minimizing heat exchanger sizes are the most important features when attempting to utilize the framework for design purposes. Thus, the current solver is not suited for investigations of compact heat exchangers.

Altogether, the work described in this thesis seems rather inconclusive. There are some parts in the implementation that should have been investigated more thoroughly. Unfortunately, the theoretical error in the derivation was discovered late in project, and the large impact of heat exchanger sizes was not thought of either. This has caused an excessive amount of time spent on debugging the code which could have been used for testing, verification and alternative solution methods.

Chapter 10

Further work

As the conclusion showed, there is still a lot of work in order to verify the current implementation of the transient solver. There is a lot of testing which needs to be done, and further development of the code is probably needed. However, due to the stiffness of the problem formulation causing large simulation times, a re-evaluation is recommended if development of a transient solver is pursued. In particular, as small geometries have been identified to cause stiffness, it seems difficult to use this formulation when the aim is to solve compact heat exchangers.

The main alternative is to reinstate the momentum equation as discussed in section 8.2. The control volumes will still appear explicitly in the system of equations and the stiffness might not be reduced. However, it will bring the problem to a system of explicit ODEs if option 1 boundary conditions are used. ODEs are the most important class of DAEs, and more specialized solvers exist for stiff ODEs than DAEs.

Bibliography

[1] Bendapudi, S. and Braun, J.E. (2002) *A literature review of dynamic models of vapor compression equipment*, Purdue University.

[2] Bendapudi, S., Braun, J.E. and Groll, E.A. (2004) *Dynamic modelling of shell-and-tube heat-exchangers: moving boundary vs. finite volume*, Purdue University.

[3] Chi, J. and Didion, D.A. (1982) *A simulation model of the transient performance of a heat pump*, International Journal of Refrigeration, 5(1982/3), pp. 176-184.

[4] Hesselgreaves, J.E. (2001) *Compact Heat Exchangers: Selection, Design and Operation*. 1^{st} edition. Pergamon.

[5] Hiebert, K.L. (1993) *DNSQE description*. Available from: `http://www.netlib.org/slatec/src/dnsqe.f` (Collected: 20. May 2014).

[6] Incropera, F.P., Dewitt, D.P., Bergman, T.L. and Lavine, A.S. (2013) *Principles of heat and mass transfer*. 7^{th} edition. Singapore: Wiley.

[7] Kay, J.M. and Nedderman, R.M. (1985) *Fluid Mechanics and Transfer Processes*. Cambridge: Cambridge University Press.

[8] Kreyszig, E. (2006) *Advanced Engineering Mathematics*. 9^{th} edition. Singapore: Wiley.

[9] Moe, H.I. (1995) *Dynamic Process Simulation: Studies on Modeling and Index Reduction* (Doktoravhandling, Norges Tekniske Høyskole). Trondheim: Norges Tekniske Høyskole.

[10] Nyers, J. and Stoyan, G. (1994) *Dynamical model adequate for controlling the evaporator of a heat pump*, International Journal of Refrigeration 17(1994/2), pp. 101-108.

[11] Næss, E. (2010) *Experimental investigation of heat transfer and pressure drop in serrated-fin tube bundles with staggered tube layouts*, Applied Thermal Engineering 30(2010/13), pp. 1531-1537.

[12] Petzold, L.R. (1982) *DDASSL description.* Available from: `http://www.netlib.org/slatec/src/ddassl.f` (Collected: 20. May 2014).

[13] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (2007) *Numerical recipes: the art of scientific computing.* 3^{rd} edition. Cambridge: Cambridge University Press.

[14] Rossi, T.M and Braun, J.E. (1999) *A real-time transient model for air conditioners*, Proc. 20th International Congress of Refrigeration, Sydney, Paper No. 743.

[15] Skaugen, G., Gjøvåg, G.A., Nekså, P. and Wahl, P.E. (2010) *Use of sophisticated heat exchanger simulation models for investigation of possible design and operational pitfalls in LNG processes*, Journal of Natural Gas Science and Engineering, 2(2010/5), pp. 235-243.

[16] Skaugen, G., Kolsaker, K., Walnum, H.T. and Wilhelmsen, Ø. (2012) *A flexible and robust modelling framework for multi-stream heat exchangers*, Computers and Chemical Engineering, 49(2013), pp. 95-104.

[17] Skaugen, G., Walnum, H.T., Hammer, M., Wahl, P.E., Wilhelmsen, Ø. and Kolsaker, K. (2013) *Design and Optimization of Compact Heat Exchangers in Processes used for Liquefaction of Natural Gas.* Unpublished paper at International Conference of Applied Energy (ICAE), 2013, Pretoria, South Africa.

[18] Smith, E.M. (2005) *Advances in thermal design of heat exchangers : a numerical approach: direct-sizing, step-wise rating, and transients.* Chichester: Wiley.

[19] Tulapurkar, C. and Khandelwal, R. (2010) *Transient Lumped Parameter Modeling For Vapour Compression Cycle Based Refrigerator*, International Refrigeration and Air Conditioning Conference, Purdue University.

[20] Watts, H.A. and Shampine, L.F. (1993a) *DDEABM description.* Available from: `http://www.netlib.org/slatec/src/ddeabm.f` (Collected: 20. May 2014).

[21] Watts, H.A. and Shampine, L.F. (1993b) *DDEBDF description.* Available from: `http://www.netlib.org/slatec/src/ddebdf.f` (Collected: 20. May 2014).

[22] Wedekind, G.L., Bhatt, B.L. and Beck, B.T. (1978) *A system mean void fraction model for predicting various transient phenomena associated with two phase evaporating and condensing flows*, International Journal of Multiphase Flow, 4(1998), pp. 97-114.

[23] Weisstein, E.W. (2014) *Courant-Friedrichs-Lewy Condition.* Available from: `http://mathworld.wolfram.com/Courant-Friedrichs-LewyCondition.html` (Collected: 30. May 2014).

[24] Welty, J., Wicks, C.E., Rorrer, G.L. and Wilson, R.E. (2008) *Fundamentals of Momentum, Heat and Mass Transfer*. 5^{th} edition. .

[25] Williatzen M., Petit N.B.O.L. and Ploug-Sorensen L. (1998) *A general dynamic simulation model for evaporators and condenser in refrigeration: Part 1 – moving boundary formulation of two-phase flows with heat exchange*, International Journal of Refrigeration, 21(1998/5), pp. 398-403.

[26] Xian-Dong He, Sheng Liu and Asada, H.H. (1998) *Multivariable Control of Vapor Compression Systems*, International Journal of HVAC&R Research, 4(1998/3), pp. 205-230.

[27] Xuan, S., Aute, V. and Radermacher, R. (2006) *Generic Dynamic Model for Heat Exchangers*, Purdue University.