openthermo

User guide and technical reference

Anders Andreasen

Contents

1 Introduction

openthermo is an open source Python3 tool for calculation of vessel depressurization / blowdown. The main phenomena modelled are visualized in Fig. 1.1. The thermodynamic state inside the vessel changes over time as seen from immediately observable variables temperature (T) and pressure (P). This is caused by change in fluid inventory (density) due to flow of gas and/or liquid out of the vessel. Further, heat is transferred from or to the surroundings via convective heat transfer on the in- and outside of the vessel with heat being conducted through the vessel wall. Due to differences in thermal resistance the vessel wall will obtain a temperature different from the fluid. Depending on the assumptions regarding the description of the fluid inside the vessel, the gas and liquid may have the same temperature (equilibrium assumption) or the two-phases may have different temperature (partial equilibrium assumption aka PPE).

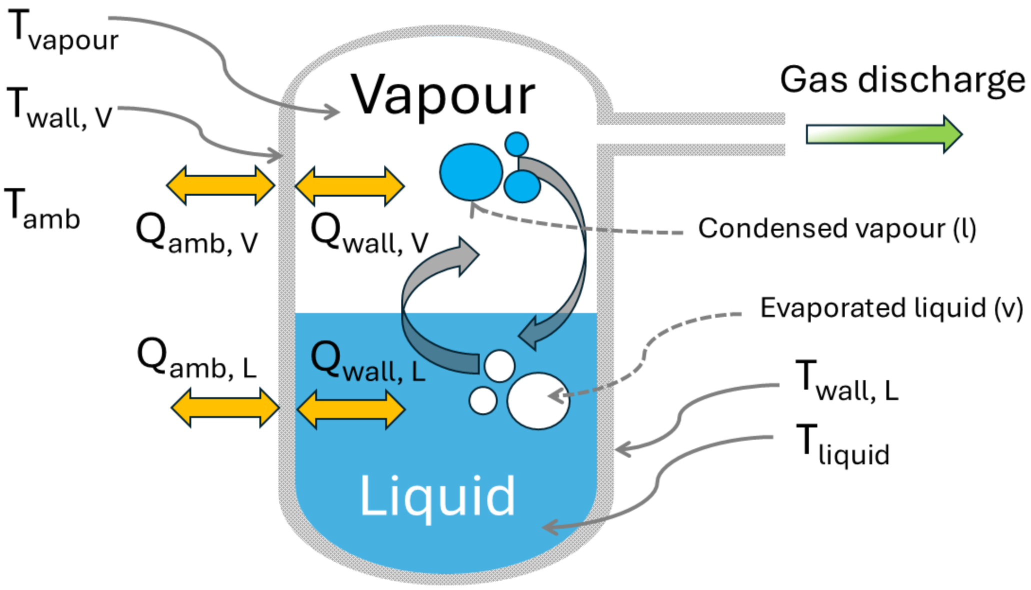

Figure 1.1: openthermo main sketch

1.1 Citing *openthermo*

If you use *openthermo* please cite the following reference:

Andreasen, A., Stegelmann, C. (2025). Open source pressure vessel blowdown modelling under partial phase equilibrium. Process Safety Progress (Accepted)

```
@article{AndreasenStegelmann,
  year = {2025},
  author = {Anders Andreasen and Carsten Stegelmann},
  title = {Open source pressure vessel blowdown modelling under partial phase equilibrium (Under review)},
  journal = {Process Safety Progress},
  doi     = {10.1002/prs.70035},
  publisher = {Wiley},
}
```

A pre-print of the paper is available on ChemRxiv: https://doi.org/10.26434/chemrxiv-2025-00xzc-v2. This is also recommended reading for a more elaborate detailing of the equations solved.

1.2 Getting the software

The source code can be obtained either from GitHub (via `git` or via the latest tar-ball release) or via **pip** . No packaged releases have currently been planned for **conda**.

The main branch is located here:

`https://github.com/ORS-Consulting/ORS-openthermo`

Clone the repo by:

```
git clone https://github.com/ORS-Consulting/ORS-openthermo.git
```

Running from source via `git` the dependencies must be installed manually from the repo root dir:

```
pip install -r requirements.txt
pip install -e .
```

Alternatively `pip` can install directly from github:

```
pip install git+https://github.com/ORS-Consulting/ORS-openthermo.git
```

Installation of latest release via **pip** also installs dependencies automatically (still pending):

```
pip install openthermo
```

1.3 Requirements

- Python
- thermopack
- thermno
- chemicals
- ht
- fluids
- Scipy
- Numpy
- matplotlib
- SciencePlots
- cerberus
- PyYaml
- pandas
- tqdm
- openpyxl
- tqdm

The code is running on Windows 10/11 x64, with stock Python 3.11 installation from Python.org and packages installed using pip. It should also run on Linux (it does on an Ubuntu image on GitHub) or in any conda environment as well, but this hasn't been checked.

1.4 Testing

Although testing is mainly intended for automated testing (CI) during development using github actions, testing of the installed package can be done for source install by:

```
python -m pytest
```

run from the root folder.

1.5 Units of measure

The SI unit system is adapted for this project. The following common units are used in the present project and this also applies to the units used in the input files:

Table 1.1: Unit system

Property	Unit	Comment
Temperature	K	° C may be used in plots
Pressure	Pa	bar is used in plots
Mass	kg	
Volume	m^3	
Time	s	
Energy	J	
Duty/power	W	
Length	m	
Area	m^2	
Heat flux	W/m^2	
Heat transfer coefficient	W/(m^2 K)	
Thermal conductivity	W/(m K)	
Density	kg/m^3	
Heat capacity	J/(kg K)	

As will be noted when presenting the equations implemented in the code, some of the equations utilise different units than the ones listed in tbl. 1.1. However, it is important to note that unit conversions are built in to the methods implemented, so the user shall not worry about unit conversion.

1.6 Credit

In the making of this document a great deal of material has been sourced (and modified) from a former colleague's M.Sc. thesis (Eriksen and Bjerre 2015), from co-published papers (Bjerre et al. 2017; Andreasen et al. 2018) and from on-line material published under permissive licenses (with proper citation). In particular the HydDown manual (Andreasen 2024) has been heavily scavenged for usable material,

Further, the making of this project would not have possible without the magnificent *thermopack* library (Wilhelmsen et al. 2017). I am also thankful for enlightening discussions with former colleague

Jacob Gram Iskov Eriksen (Ramboll Energy, Denmark) and former and now present colleague Carsten Stegelmann (ORS Consulting).

The present document is typeset using Markdown + pandoc with the Eisvogel template.

1.7 License

2 Background

2.1 Early works

The foundation of *openthermo* was laid by Carsten Stegelmann, more than a decade ago, who developed code for running blowdown calculations in a spreadsheet relying heavily on VBA and a legacy flash calculation routine (DLL) coded in FORTRAN by late Prof. Michael Michelsen (Michelsen 1982a, 1982b). This worked surprisingly well and executed very efficiently. The short-comings where lacking heat transfer modelling as well as an *equilibrium* only approach i.e. two-phase fluids were in full equilibrium at all times.

2.2 Challenges

The VBA code was tricky to maintain in a version control system, and in the meantime the availability of high-quality thermodynamic packages for Python increased significantly. However, reimplementing an entire codebase is a time-consuming task, and this proved difficult to manage working full time as engineering consultants, and years went by without being able to fully do the long haul required.

2.3 Proof of concept

Having worked together in the same company, Carsten and I split ways 5 years ago. Then the Covid-19 hit and that freed up some spare time for me, staying at home without a lot of activities being possible. That lead to the development of HydDown (Andreasen 2021). It started as a small spare-time project for calculation of vessel filling and depressurization behaviour. At that time the expectation was, that a lot of engineering work was expected in high-pressure storage and filling stations. The work on HydDown served as a proof of concept for an efficient implementation in Python mainly provided by the Coolprop back-end (I. H. Bell et al. 2014). Eventually HydDown matured and is now in a state where it can model heat transfer in both steel and dual-layer low thermal conductivity composites during depressurisation/pressurisation. However, it cannot manage two-phase (gas/liquid) behaviour due to limitations in the flash calculation in CoolProp. Thus, a change in thermodynamic back-end was inevitable.

2.4 *openthermo* development

Recently, I joined ORS Consulting with Carsten and we revived our plans for a rigorous blowdown simulation tool. The remaining challenge was the more complex two-phase (or three-phase) flash problem and the non-equilibrium / partial equilibrium assumption. At all times the big inspiration has been the work done at Empirical College London, University College London on the codes BLOWDOWN (A. Haque et al. 1990; M. A. Haque, Richardson, and Saville 1992) and BLOWSIM (Wong 1998; Mahgerefteh and Wong 1999), respectively, and later VBsim (D'Alessandro et al. 2015). BLOWDOWN was acquired by AspenTech and made available in HYSYS. Further the motivation has also been to have a tool easily accessible as a supplement to commerical (and expensive) tools. Both to reduce load on license pools, but also to provide more efficient workflows. We wanted to make the tool *open source* and available to the public. However, license limitations on the legacy flash calculation by Prof. Michelsen required an alternative flash calculation with a more permissive license. Several tools are now available such as Python *thermo* (C. Bell 2025b), NeqSim (Solbraa 2025) and *thermopack*. Handling vessel depressurisation, which is effectively a UV-flash problem (Internal Energy - Volume) requires an excessive number of flash calculations to be performed. Thus, a fast and stable flash calculation is required. In order to provide speed and stability the preliminary choice has been *thermopack* from SINTEF, although it may change in the future in order to provide a three-phase (VLLE) flash.

3 Limitations and implementation details

A few choices has been made to keep things simple:

- *thermopack* (Wilhelmsen et al. 2017) is used as thermodynamic backend
- Thermodynamic and transport properties (enthalpy, entropy, internal energy, liquid density, thermal conductivity, viscosity) is provided via Python *thermo* / *chemicals*.
- No temperature stratification inside bulk phases
- No temperature gradient through vessel wall.
- Heat transfer is modelled using empirical correlations
- Only single and two-phase (VLE) is handled. Three-phase (VLLE) cannot currently be modelled in the open source version.
- Currently only the Peng-Robinson (PR) (Peng and Robinson 1976) and Soave-Redlich-Kwong (Soave 1972) cubic equations of state are made available.

- *openthermo* and it's legacy code has been built by engineers NOT software developers.

Ignoring temperature gradients in the vessel wall is an acceptable assumption for (not too thick) steel vessel walls. However, low thermal conductivity materials cannot accurately be modelled. In order to do so, a 1-D heat transfer model shall be implemented.

One limitation of *thermopack* is that including pseudo components is not easy if the internal property calculations shall be used. Further, *thermopack* does not provide transport properties. In order to allow an implementation for pseudo components and transport properties the *thermopack* library is only used for flash calculations and used as a plugin, the rest (and less computationally demanding tasks) is provided by Python *thermo*.

While *openthermo* has been extensively validated and fidelity has been built, we have not had the workforce or man-months/years available as the legacy university projects or the commerical software providers. This means that there will likely be situations where the code fails to provide results, cases that are not covered or simply the code will not work. This is the cost and risk of using open source software.

4 Usage

When using *openthermo* a dictionary holding all relevant input in order for the program to do vessel calculations shall be provided when the main Blowdown class is initialized. The depressurisation is run by calling the class method **depressurize()** or **depressurize_euler()**. The former utilises an adaptive step-size Runge-Kutta family integration of relevant mass and energy balances. The latter employs a simple explicit Euler integration with fixed user defined step size.

4.1 Basic usage

A minimal example for running *openthermo* is provided below. The example is for an isentropic blowdown i.e. with a rigorous energy and mass balance, taking into account work done by the discharged fluid, but without any heat transfer to or from vessel wall/surroundings. The model assumes full equilibrium between gas and liquid (if two-phases exists at any time during the simulation).

```python
from openthermo.vessel.blowdown import Blowdown

input = {}
P = 12e5
T = 298.15

input["mode"] = "isentropic"
input["eos_model"] = "PR"
input["liquid_density"] = "eos"
input["max_time"] = 900
input["length"] = 10
input["diameter"] = 3
input["vessel_type"] = "Flat-end"
input["orientation"] = "horizontal"
input["liquid_level"] = 1.5
input["operating_temperature"] = T
input["operating_pressure"] = P
input["ambient_temperature"] = 273
input["back_pressure"] = 1.01e5
input["bdv_orifice_size"] = 0.03  # m
input["bdv_orifice_cd"] = 0.84
input['component_names'] =  ["methane", "propane", "n-butane", "i-butane", "n-decane"]
```

```python
input["molefracs"] =  [0.8, 0.05, 0.01, 0.01, 0.10]

segment = Blowdown(input)
segment.depressurize()
segment.plot()
```

4.2 More advanced usage

Below is a more advanced example also taking heat transfer into account between the vessel wall and the surroundings and from/to the vessel fluid. The model applies the partial / non-equilibrium approach between gas and liquid (if two-phase occur at any time during the blowdown). The below example is for a condensing fluid initially at super-critical conditions where significant temperature difference between the gas and condensed liquid occurs during the experiment/simulation. The example is from an experiment reported by Szczepanski (Szczepanski 1994) and widely applied for demonstration and validation of blowdown simulation programs e.g. (M. A. Haque et al. 1992; Wong 1998; Park et al. 2018).

```python
from openthermo.vessel.blowdown import Blowdown

input = {}
P = 116*1.013e5
T = 293.0

input["mode"] = "isentropic"
input["heat_transfer"] = "rigorous"
input["wall_thickness"] = 0.059  # m
input["eos_model"] = "PR"
input["liquid_density"] = "eos"
input["max_time"] = 1500
input["delay"] = 0
input["time_step"] = 10
input["length"] = 2.25
input["diameter"] = 1.13
input["vessel_type"] = "ASME F&D"
input["orientation"] = "vertical"
input["liquid_level"] = 0
input["water_level"] = 0.0
input["operating_temperature"] = T
input["operating_pressure"] = P
input["ambient_temperature"] = 293.0
input["back_pressure"] = 1.01e5
input["bdv_orifice_size"] = 0.01  # m
input["bdv_orifice_cd"] = 0.8

names = ["methane", "ethane", "propane", "n-butane"]
molefracs = [0.64, 0.06, 0.28, 0.02]
```

```

input["molefracs"] = molefracs
input["component_names"] = names

segment = Blowdown(input)
segment.depressurize_euler()
```

4.3 Calculation methods

The following methods are implemented:

- Isothermal: constant temperature of the fluid during depressurization
- Adiabatic: the energy balance does not account for work done by the fluid during discharge. No heat transfer is included.

- Isentropic: the energy balance accounts for work done by the fluid during discharge. This method can be run without or with heat transfer modelled. A variation of this method includes external backgorund heat load from a fire (pool or jet) using the Stefan-Boltzmann equation.
- Fire: This method is a variant of the isentropic method where an API 521 pool-fire heat load is applied to the wetted surface inside the vessel. No other heat transfer is included.

For calculations the minimal input required are:

- Initial conditions (pressure, temperature, liquid level)
- vessel dimensions (ID, length)
- orifice parameters (Cd, diameter, back-pressure)
- Calculation setup (end time)
- Fluid components and composition

For rigorous heat transfer additional input are required:

- Vessel thickness
- Vessel material heat capacity
- Vessel material density
- Ambient temperature or a Stefan-Boltzmann external fire heat load type.

More elaborate description of the required input for the different calculation types are provided in Sec. 4.4.

4.4 Input fields and hierarchy

In the following the full listing of input for the different calculation types is summarised cf. Tbl. 4.1.

Table 4.1: Input overview

Input field	Unit	Description	Mandatory? / Depends on	Options
`operating_temperature`	K	Initial temperature	Yes	N/A
`operating_pressure`	Pa	Initial pressure	Yes	N/A
`mode`	N/A	Calculation mode	Yes	`isothermal`
^^				`adiabatic`
^^				`fire`
^^				`rigorous`
`eos_model`	N/A	Equation of state	Yes	`PR`
^^				`SRK`
`liquid_density`	N/A	Liquid density model	Yes	`eos`
`max_time`	s	Simulation end time	Yes	N/A
`delay`	s	Delay before blowdown	No	N/A
`time_step`	s	Required for PPE	No/Yes	N/A
`length`	m	Vessel length/height	Yes	N/A
`diameter`	m	Vessel diameter	Yes	N/A
`vessel_type`	N/A	Vessel end type	Yes	`Flat-end`
^^				`ASME F&D`
^^				`DIN`

Input field	Unit	Description	Mandatory? / Depends on	Options
orientation	N/A	Vessel orientation	Yes	horizontal
^^				vertical
liquid_level	m	Initial liquid level	Yes	N/A
back_pressure	Pa	Blowdown back-pressure	Yes	N/A
bdv_orifice_size	m	BDV orifice diameter	Yes	N/A
bdv_orifice_cd	N/A	Orifice discharge coefficient	Yes	N/A
heat_transfer	N/A	Heat transfer option	No	rigorous
^^				rigorous_sb_fire
wall_thickness	m	Material thickness	heat_transfer	N/A
ambient_temperature	K	Ambient temperature	heat_transfer	N/A
sb_fire_type	N/A	S-B fire type	heat_transfer =rigorous_sb_fire	api_pool
^^				api_jet
^^				scandpower_pool
^^				scandpower_jet
molefracs	N/A	List of mole fractions	Yes	N/A

Input field	Unit	Description	Mandatory? / Depends on	Options
component_names	N/A	Pure component names	Yes	N/A
fire_type	N/A	API 521 fire type	Yes, mode=fire	API521 (default)
^^				API521_CONFINED
drain_fire_fighting	N/A	API521 pool fire input	No, mode=fire and fire_type=API521	Inadequate (default)
^^				Adequate
exposed_area		API 521 exposed area	No, mode=fire	Wetted (default)
^^				Total
^^				Manual
pseudo_names	N/A	Pseudo component names	No	N/A
pseudo_molefracs	N/A	Pseudo component mole fractions	pseudo_names	N/A
pseudo_Tbs	K	True boiling point of pseudos	pseudo_names	N/A
pseudo_SG	N/A	Specific gravity of pseudos	pseudo_names	N/A
leak_active	N/A	Additional outflow via leak at t=0	No	0
^^				1
leak_size	m	Equivalent orifice size of leak	leak_active=1	N/A

Input field	Unit	Description	Mandatory? / Depends on	Options
leak_cd	N/A	Leak discharge coefiicient	leak_active=1	N/A
leak_type	N/A	Fluid released from leak	leak_active=1	liquid
^^				gas
^^				two-phase

5 Theory and methods

In this chapter the basic theory and governing equations for the model implementation in *openthermo* is presented. The following main topics are covered:

- Thermodynamics
- Mass transfer
- Heat transfer
- Equilibrium modelling
- Vessel geometry
- Handling pseudo components

5.1 Thermodynamics and property estimation

5.1.1 Equation of state

Currently only the Peng-Robinson (Peng and Robinson 1976) and Soave-Redlich-Kwong (Soave 1972) cubic equations of stare are available despite many being available in both *thermopack* and *thermo*. In the future more cubic equations of state may be made available.

For more background information and theory regarding the equations of state (especially for mixtures), please refer to e.g. *thermo* documentation, the DWSIM user guide, a *thermopack* memo or textbooks such as e.g. (Smith, Van Ness, and Abbott 1996; Reid, Prausnitz, and Poling 1987).

5.1.2 Property estimation

Property estimation is provided by the Python *thermo* package (C. Bell 2025b), this includes thermodynamic properties such as enthalpy, entropy and internal energy as well as tranport properties such as thermal conductivity, viscosity and surafec tension. In general all components that are available with *thermo* can be used. Although *thermopack* comes with a smaller set of predefined components this is not limiting since *thermopack* is called with minimal input for the EOS/flash TP calculation i.e. mole fractions, critical pressure, critical temperature and accentric factor (and binary interaction parameters). Any property estimation is managed by *thermo*.

5.1.3 First law for flow process

The control volume sketched in Fig. 5.1, separated from the surrounding by a control surface, is used as a basis for the analysis of an open thermodynamic system with flowing streams (fs) in and out, according to (Smith, Van Ness, and Abbott 1996)

A general mass balance or continuity equation can be written:

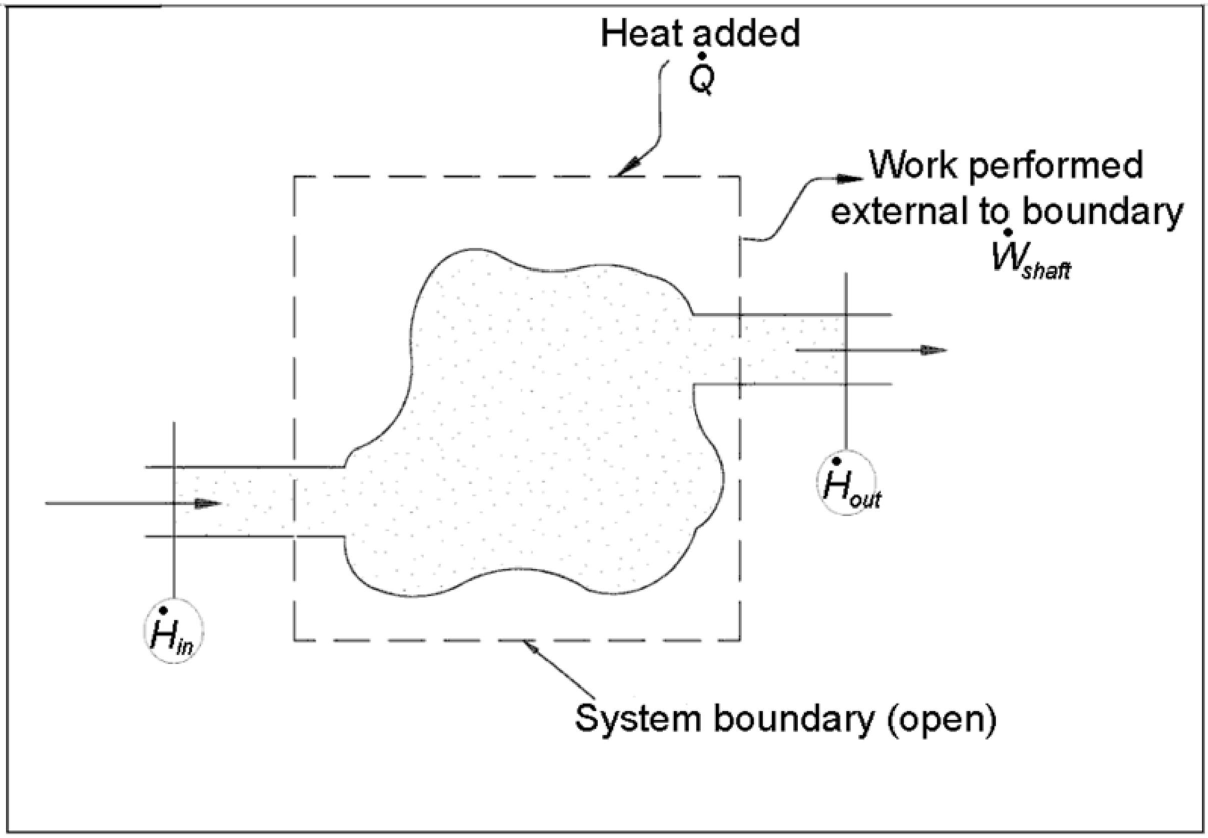

Figure 5.1: Control volume with one entrance and one exit. The image has been sourced from (WikiMedia 2008,).

$$\frac{m_{cv}}{dt} + \Delta\left(\dot{m}\right)_{fs} = 0 \tag{5.1}$$

The first term is the accumulation term i.e. the rate of change of the mass inside the control volume, m_cv, and the Δ in the second term represents the difference between the outflow and the inflow

$$\Delta\left(\dot{m}\right)_{fs} = \dot{m}_2 - \dot{m}_1$$

An energy balance for the control volume, with the first law of thermodynamics applied, needs to account for all the energy modes with can cross the control surface. Each stream has a total energy

$$u + \frac{1}{2}w^2 + zg$$

where the first term is the specific internal energy, the second term is the kinetic energy and the last term is the potential energy. The rate at which each stream transports energy in or out of the control volume is given by

$$\dot{m}(u + \frac{1}{2}W^2 + zg)$$

and in total

$$\Delta \left[\dot{m}(U + \frac{1}{2}w^2 + zg) \right]_{fs}$$

Furthermore, work (not to be confused with shaft work) is also associated with each stream in order to move the stream into or out from the control volume (one can think of a hypothetical piston pushing the fluid at constant pressure), and the work is equal to PV on the basis of the specific fluid volume. The work rate for each stream is

$$\dot{m}(pV)$$

and in total

$$\Delta \left[\dot{m}(pV) \right]_{fs}$$

Further, heat may be transferred to (or from) the control volume at a rate $\dot{Q}$ and shaft work may be applied, $\dot{W}_{shaft}$. Combining all this with the accumulation term given by the change in total internal energy the following general energy balance can be written:

$$\frac{d(mu)_{cv}}{dt} + \Delta \left[\dot{m}(u + \frac{1}{2}w^2 + zg) \right]_{fs} + \Delta \left[\dot{m}(pV) \right]_{fs} = \dot{Q} + \dot{W}_{shaft}$$

Applying the relation $h = u + pV$, setting $\dot{W}_{shaft} = 0$ since no shaft work is applied to or extracted from the vessel, and assuming that kinetic and potential energy changes can be omitted the energy balance simplifies to:

$$\frac{d(mu)_{cv}}{dt} + \Delta \left[\dot{m}h\right]_{fs} = \dot{Q}$$

The equation can be further simplified if only a single port acting as either inlet or outlet is present:

$$\frac{d(mu)_{cv}}{dt} + \dot{m}h = \dot{Q} \tag{5.2}$$

where the sign of $\dot{m}$ determines if the control volume is either emptied or filled. In this case only dischartge is considered in opposition to HydDown (Andreasen 2021). The continuity equation Eq. 5.1 and the energy balance Eq. 5.2 combined with the equation of state are the key equations that shall be solved/integrated in order to calculate the change in temperature and pressure as a function of time.

If the fluid inventory is a multi-phase mixture and the discharge is only sourced from one of the phases, a mole balance is also required in addition to the energy balance and mass balance equations above, in order to account for the dynamical change in composition during the blowdown process. For all components, i, the following balance is made where the change in total moles of component i in the fluid inventory is given by the rate of discharge of the component, which is related to the mole fraction in the vapour phase

$$\frac{dN_i}{dt} = \dot{N}_i = \frac{\dot{m}}{M} y_i$$

In order to update the phase equilibrium i.e. account for flashing liquid (typically) or condensing gas, for each calculation step a UV-flash calculation is called with the current global composition. This is under the assumption of full thermal and compositional equilibrium between gas and liquid. In the current state of the code the PT-flash calculation provided by *thermopack* is wrapped inside a Flash object inspired by *thermo* solving for prescribed specific internal energy and molar volume.

5.2 Flow devices

5.2.1 Restriction Orifice (gasseous discharge)

When a fluid flows through a restriction or opening such as an orifice, the velocity will be affected by conditions upstream and downstream. If the upstream pressure is high enough, relative to the downstream pressure, the velocity will reach the speed of sound (Ma = 1) and the flow rate obtained will be the critical flow rate. This condition is referred to as choked flow. The maximum downstream pressure for the flow to still be sonic (Ma = 1), is when $P_d = P_c$. The ratio of the critical and upstream pressure is defined by equation Eq. 5.3.

$$\frac{P_c}{P_u} = \left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} \tag{5.3}$$

- P_c is the critical pressure. [kPa]
- P_d is the downstream pressure. [kPa]
- P_u is the upstream pressure. [kPa]
- k is the isentropic coefficient, approximated by the ideal gas heat capacity ratio C_p/C_v. [-]

In order to calculate the mass flow rate through an orifice equations are used based on literature from the Committee for the Prevention of Disasters (Bosch and Weterings 2005).

To account for the difference in choked and non-choked flow a set limit pressure is introduced as in equation Eq. 5.4. If the downstream pressure, P_{down}, is below the pressure limit, $ P_{limit}$, then the flow is choked, and the pressure used, P_{used}, in equation Eq. 5.5 should be the pressure limit, P_{limit}. Otherwise if the downstream pressure, P_{down}, is greater than or equal to the pressure limit, P_{limit}, the flow is no longer choked and the pressure used should be the downstream pressure, P_{down} (Bosch and Weterings 2005).

$$P_{limit} = P_{up} \cdot \left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} \tag{5.4}$$

$$\dot{m}_{flow} = C_d \cdot A \cdot \sqrt{\left(\frac{2k}{k-1}\right) \cdot P_{up} \cdot \rho \cdot \left(\frac{P_{used}}{P_{up}}\right)^{\frac{2}{k}} \left(1 - \left(\frac{P_{used}}{P_{up}}\right)^{\frac{k-1}{k}}\right)} \tag{5.5}$$

- ρ is the density of the gas upstream. $[kg/m^3]$
- P_{limit} is the pressure limit of the upstream absolute pressure. $[bara]$
- P_{up} is the absolute pressure upstream of the orifice. $[bara]$
- k is the ratio of the heat capacities at constant pressure, C_p, and at constant volume, C_v.
- P_{down} is the absolute pressure downstream of the orifice. $[bara]$
- P_{used} is the pressure used in the mass flow equation based on choked or non-choked conditions. $[bara]$
- $\dot{m}_{flow}$ is the mass flow through the orifice. $[kg/s]$
- C_d is the discharge coefficient of the orifice opening. $[-]$
- A is the cross sectional area of the orifice. $[m^2]$

5.2.2 Leaks

In addition to the vapour outflow through a Blowdown valve/orifice it is also possible to include leaks. Three leak types are possible:

- Gas/vapour
- Liquid
- Two-phase

For gasseous leaks the methodology presented in the previous section is applied. For liquid leaks a simple Bernouilli equation is used (Bosch and Weterings 2005) and for two-phase a simple Fauske expression is used (Bosch and Weterings 2005; Technica 1990). For liquid release the assumption is that static head due to liquid level is added to the internal vessel pressure upstream the leak.

5.3 Heat transfer

5.3.1 Natural convection

Experiments have indicated that the internal heat transfer mechanism for a vessel subject to depressurization can be well approximated by that of natural convection as found from measured Nusselt numbers being well correlated with Rayleigh number, with no apparent improvement in model performance by inclusion of the Reynolds number in the model (Woodfield, Monde, and Mitsutake 2007).

To determine the heat transfer for the gas-wall interface, the following is applied cf. equation Eq. 5.6:

$$\frac{dQ}{dt} = \dot{Q} = hA(T_s - T_{gas}) \tag{5.6}$$

- dQ is the change in thermal energy due to convective heat transfer. [J]
- dt is the change in time during the heat transfer. [s]
- h is the convective heat transfer. [W/m

  2

  K]
- A is the area normal to the direction of the heat transfer. [m^2]
- T_s is the inner surface temperature of the geometry. [K]
- T_{gas} is the temperature of the bulk gas inside the vessel. [K]

The convective heat transfer will need to be estimated for the the gas-wall interface, by the use of empirical relations for the Nusselt number. The Nusselt number describes the ratio of convective heat transfer to conductive heat transfer, normal to a surface area, as given in equation eq. 5.7.

$$Nu = \frac{hL}{k} \tag{5.7}$$

- Nu is the Nusselt number. [-]

- h is the convective heat transfer. [W/m^{2}K]
- L is a characteristic length of the geometry. [m]
- k is the thermal conductivity of the gas. [W/m·K]

The characteristic length L used is the height of the gas volume i.e. the length of a vertical vessel or the diameter of a horizontal vessel.

The empirical correlations used to calculate the Nusselt number of the gas-wall interface is a function of the Rayleigh number, which can be defined by the Grashof number and Prandtl number, as in equation Eq. 5.8:

$$Ra = Gr \cdot Pr \tag{5.8}$$

- Ra is the Rayleigh number. [-]
- Gr is the Grashof number. [-]
- Pr is the Prandtl number. [-]

The Grashof number is a dimensionless number which approximates the ratio of the buoyancy forces to viscous forces, as given in equation [Eq. 5.9}:

$$Gr = \frac{\beta g \rho^2 L^3 \Delta T}{\mu^2} \tag{5.9}$$

The Prandtl number is a dimensionless number defined as the ratio of the momentum diffusivity to thermal diffusivity, as given in equation Eq. 5.10:

$$Pr = \frac{c_p \mu}{k} \tag{5.10}$$

- β is the coefficient of volume expansion. [1/K]
- g is the standard acceleration of gravity. [m/s^2]
- ρ is the gas density. [kg/m^3]
- L is the characteristic length. [m]
- ΔT is the temperature difference of the surface and gas. [K]
- μ is the dynamic viscosity. [kg/m·s]
- c_p is the heat capacity of gas. [J/kg·K]
- k is the thermal conductivity of gas. [J/m·K]

It is important to note that the properties in the above equations shall be evaluated at the fluid film temperature which can be approximated by the average of the the fluid bulk temperature and the vessel wall temperature (Geankoplis 1993).

The mechanism of heat transfer on the outside of the vessel at ambient conditions is similar to the above, but in the current version in case of ambient heat transfer a fixed coefficient of 8 W/m^2 K is applied.

5.3.2 Nucleate boiling heat transfer

For heat transfer between vessel wall and liquid, this is treated as nucleate boiling and estimated via the Rohsenow correlation (Rohsenow 1951) as implemented in the *ht* python library (C. Bell et al. 2024).

$$h = \mu_L \Delta H_{vap} \left[\frac{g(\rho_L - \rho_v)}{\sigma}\right]^{0.5} \left[\frac{C_{p,L}\Delta T_e^{2/3}}{C_{sf}\Delta H_{vap} Pr_L^n}\right]^3$$

- ρ_L is the density of the liquid [kg/m^3]
- ρ_v is the density of the produced gas [kg/m^3]
- μ_l is the viscosity of liquid [Pa s]
- k_l is the thermal conductivity of liquid [W/m K]
- $C_{p,l}$ is the heat capacity of liquid [J/kg K]
- H_{vap} is the heat of vaporization of the fluid [J/kg]
- σ is the surface tension of liquid [N/m]
- T_e is the excess wall temperature, [K]
- C_{sf} is Rohsenow coefficient specific to fluid and metal [-]
- n constant, 1 for water, 1.7 (default) for other fluids usually [-]

In the present study a value of the Rohsenow coefficient of 0.013 is applied and the exponent n is set to 1.7. See also (Pioro 1999; Jabardo et al. 2004).

5.3.3 Conduction

For accurate prediction of the outer and especially the inner wall temperature for correct estimation of internal convective heat transfer and the average material temperature, the general equation of 1-D unsteady heat transfer shall be solved:

$$\frac{\delta T}{\delta t} = \frac{k}{\rho c_p}\frac{\delta^2 T}{\delta x^2}$$

- T is temperature
- x is the spatial (1-D) coordinate
- k is the thermal conductivity

- ρ is the material density
- C_p is the specific heat capacity

To be solved, the initial values and boundary values must be specified. In its default state (if thermal cobductivity is not applied for the vessel), HydDown does not include the unsteady heat transfer model, i.e. the assumption is that the temperature from outer to inner surface is uniform and equal to the average temperature. This is obviously a crude approximation, but might be justified depending in the Biot number:

$$Bi = \frac{hL}{k}$$

The Biot number is a simple measure of the ratio of the heat transfer resistances at the surface of a body to the inside of a body. The ratio gives an indication to which extent the temperature will vary in space (gradient) when the body is subject to a displacement in temperature at the surface boundary layer. Striednig *et al.* (Striednig et al. 2014) concluded that for a type I (steel) cylinder the Biot number was approx. 0.03 and hence the error in assuming a uniform temperature in the vessel wall was low.

With a typical thermal conductivity of 45 W/mK for steel and a heat transfer coefficient up to 600 W/m^2K (Woodfield, Monde, and Mitsutake 2007) the Biot number for a vessel with a wall thickness of 2 cm is 0.27. This is significantly higher than that approximated by (Striednig et al. 2014). However, the Biot number is significantly lower than 1, and the assumption of a uniform temperature is reasonable. However, for increased wall thickness, and/or for different materials with lower thermal conductivity, the error may grow to an unacceptable level.

Especially for vessels with low conductivity materials (or very thick walls) accurate estimation of the vessel wall temperatures requires the 1-D transient heat transfer problem to be solved.

5.3.4 General fire heat loads

The heat transfer from the flame to the shell is modelled using the recommended approach from Scandpower (Hekkelstrand and Skulstad 2004) and API (API 2014). The heat transfer from the flame to the vessel shell is divided into radiation, convection, and reradiation as seen in equation Eq. 5.11:

$$q_f = \underbrace{\alpha_s \cdot \varepsilon_f \cdot \sigma \cdot T_{rf}^4}_{\text{Radiation}} + \underbrace{h_f \cdot (T_f - T_s(t))}_{\text{Convection}} - \underbrace{\varepsilon_s \cdot \sigma \cdot T_s(t)^4}_{\text{Reradiation}} \tag{5.11}$$

- q_f is the flame heat flux. [W/m^2]
- α_s is the vessel surface absorptivity. [-]
- ε_f is the flame emissivity. [-]

- σ is the Stefan-Boltzmann constant, $\sigma = 5.67 \cdot 10^{-8}$ [W/m^2· K^4]
- T_{rf} is the radiative flame temperature. Used for radiative heat transfer. [K]
- T_f is the flame temperature engulfing the vessel. Used for convective heat transfer. [K]
- h_f is the convection heat transfer coefficient between the flame and the surface. [W/m^2· K]
- $T_s(t)$ is the time dependent surface temperature. [K]
- ε_s is the surface emissivity. [-]

This model assumes that the pressure vessel is fully engulfed by the flame. This means that the view factor for the radiation is unity and is therefore not taken into consideration. The convective heat transfer coefficients for a jet fire and a pool fire, and recommended values for the emissivity and absorptivity, are given by Scandpower as (Hekkelstrand and Skulstad 2004):

- $h_{jet\ fire}$ = 100 [W/m^2·K]
- $h_{pool\ fire}$ = 30 [W/m^2·K]
- α_s = 0.85
- ε_s = 0.85
- ε_f = 1.0 (optical thick flames, thickness > 1 m)

The flame temperature is found by solving equation Eq. 5.12 for the incident heat flux in relation to the ambient conditions. The flame temperature is kept constant throughout the simulation:

$$q_{total} = \sigma \cdot T_{rf}^4 + h_f \cdot (T_f - T_{amb}) \tag{5.12}$$

- q_{total} is the incident flame heat flux as given in table Tbl. 5.1. [W/m^2]
- T_{amb} is the ambient temperature $\approx$ 293 K (20° C)

The heat flux used to calculate the flame temperature is given in table tbl. 5.1. Different coefficient are proposed by API 521 (API 2014) and this source is referenced for more information.

Table 5.1: Incident heat fluxes for various fire scenarios given by Scandpower (Hekkelstrand and Skulstad 2004)

	Small jet fire [kW/m^2]	Large jet fire [kW/m^2]	Pool fire [kW/m^2]
Peak heat load	250	350	150
Background heat load	0	100	100

5.3.5 API 521 pool-fire heat load

The amount of heat absorbed by a vessel exposed to an open pool fire is markedly affected by the type of fuel feeding the fire, the degree to which the vessel is enveloped by the flames (a function of vessel size and shape), the environment factor, firefighting, and drainage. Eq. 5.13 is used to evaluate these conditions if there are prompt firefighting efforts and drainage of flammable materials away from the vessels.

$$Q = C_1 F A_{ws}^{0.82} \tag{5.13}$$

Where:

- Q is the total heat absorption (input) to the wetted surface, expressed in W
- C_1 is a constant = 43,200 in SI units
- F is an environment factor (see Tbl. 5.2)
- A_{ws} is the total wetted surface, expressed in m²

> **Note 1:** See API 521 for guidance.
> **Note 2:** The expression $A_{ws}^{0.82}$ is the area exposure factor or ratio. This ratio recognises that large vessels are less likely than small ones to be completely exposed to the flame of an open pool fire.

Table 5.2: Environmental factors suggested by API521 (API 2014). a Insulation thermal conductivity divided by thickness for fire exposure conditions in W/m²·K The environment factor, F, in Eq. 5.13 and eq. 5.14 does not apply to uninsulated vessels. The environment factor should be replaced by 1.0 when calculating heat input to uninsulated vessels. b See API521 for additional notes.

Type of Equipment	Environment Factor F
Bare vessel	1.0
Insulated vessel with insulation conductance valuesa	
22.71	0.3
11.36	0.15
5.68	0.075
3.80	0.05
2.84	0.0376
2.27	0.03

Type of Equipment	Environment Factor F
1.87	0.026
Water application facilities, on bare vesselb	1.0
Depressurizing and emptying facilitiesb	1.0
Earth-covered storage	0.03
Below-grade storage	0.00

Where adequate drainage and firefighting equipment do not exist, Eq. 5.14 should be used:

$$Q = C_2 \cdot F \cdot A_{ws}^{0.82} \tag{5.14}$$

Where:

- C_2 is a constant = 70,900 in SI units

If the ratio between fire volume and confined volume becomes large, then the use of the open pool fire equations could underestimate the heat input to exposed equipment. In these cases, the Stefan-Boltzman methodology descibed in the previous section should be used with an increased fire temperature to account for effects of preheating and reradiation. Partial confinement can also result in higher heat fluxes and enhanced exposure of the wetted surfaces to the pool fire. An example is where a vessel is partially confined by adjacent embankments or walls with a height comparable to the vessel's height. For confined areas the conservative approach would be to apply Eq. 5.14 but with the wetted area term (AWS) raised to the 1.0 power instead of the 0.82 power and $C = 108,900 W/m^2$.

$$Q = C_3 \cdot F \cdot A_{ws} \tag{5.15}$$

Where:

- C_3 is a constant = 108,900 in SI units

5.4 Vessel geometry

All vessels modelled are assumed of cylindrical shape. The following shapes are available in *openthermo* all provided by the Python *fluids* library (C. Bell 2025a).

- Flat-end vessel

- ASME F&D
- DIN (28011)
- 2:1 Semi-elliptical
- Hemisperical

Both ASME F&D, DIN and 2:1 semielliptical are variants of a torispherical vessel. See also the document Calculating Tank Volume. The hemisperical ends are half-speres extending one radius out.

Table 5.3: Vessel geometry details. For torispherical tank heads, the following *f* and *k* parameters are used in standards (C. Bell 2025a). *f* is the dish-radius parameter for tanks with torispherical heads or bottoms, *k* is the knuckle-radius parameter for tanks with torispherical heads or bottoms

Vessel geometry	f	k
2:1 semi-elliptical	0.9	0.17
ASME F&D	1	0.06
DIN 28011	1	0.1

Using the *fluids* library partial volumes, surface area (full and partial) and liquid level (from partial volume) can be calculated and used internally in *openthermo*.

5.5 Model implementation

Two main implementations have been made, which are referred to as *homogeneous equilibrium* or *full equilibrium* and *partial phase equilibrium* (PPE), which will be further elaborated in the following.

5.5.1 Full equilibrium

In the full/homogeneous equilibrium (HE) concept it is assumed that any phase will be in full equilibrium with the other phases and that there is a uniform pressure and temperature throughout all phases. However, in the case of both vapour and liquid being present, different vessel wall temperatures will be solved for. The ordinary differential equations (mass, energy, mole and wall temperatures) are solved using the *dopri5* method (Dormand and Prince 1980; Hairer, Wanner, and Nørsett 1993), which is an explicit Runge-Kutta method of order (4)5 as provided in *scipy* (Virtanen et al. 2020). In each time step, using the input for component moles and total internal energy an UV-flash is solved in order to estimate the the pressure and temperature as well as phase split and properties for each phase. The UV-flash is obtained by formulating a minimisation problem around the standard PT-flash, iteratively solving for pressure and temperature while minimizing the error in internal energy and molar volume using a Nelder-Mead algorithm (Gao and Han 2012) as provided by *scipy* (Virtanen et al. 2020). Using the estimated liquid content (if any) the wetted surface area can be estimated and the heat transfer to both liquid and vapour can be estimated. With the vapour properties the mass discharge rate can be calculated. Thus, all derivatives can be quantified and passed to the Runge-Kutta solver. In the homogenous equilibrium approach the density of the liquid phase can be estimated both by the equation of state or using the COSTALD method (Hankinson and Thomson 1979) for saturated liquids for a more realistic liquid density (still experimental).

5.5.2 Partial phase equilibrium

The concept of partial phase equilibrium (PPE) was first introduced by Speranza and Terenzi (Speranza and Terenzi 2005) in order to describe the non-equilibrium between the phases during the blowdown process and further refined by Ricci *et al.* (Ricci et al. 2015) and Terenzi *et al.* (D'Alessandro et al. 2015).

In the PPE approach the fluid in the vessel is divided in two different zones during the blowdown : zone **V** (vapour phase) and zone **L** (liquid phase). See also Fig. 1.1. In order to account for the mass and energy exchanged between the liquid and vapour phases, it is necessary to introduce two new phases **v** and **l**. In the following **v** and **l** phases are denoted as **child** phases while **V** and **L** will be named **parent phases**. The child phase **v** is formed (as bubbles) from the liquid parent (**L**) vaporization while the child phase **l** is formed from the parent vapour (**V**) condensation (as droplets). It is assumed that the new phase **v** is in phase equilibrium with the liquid **L** and likewise that the new phase **l** is in phase equilibrium with the gas (bulk) **V**. While the child phases may physically appear as droplets in the gas phase and bubbles in the liquid phase and have a significant lifetime during droplet settling/buoyant rising of bubbles, for the calculation purpose they are assumed to form and immediately drop and homogenize in their corresponding bulk phase (actually within each major time step). Thus, the droplets will drop immediately into the liquid (bulk), while the vapour (bubbles) will immediately rise into the gas (bulk) phase.

Consequently, it is possible to consider the whole system as split into two partial sub-system **V+l** and **L+v**, constituted of one **parent** (the bulk phase) and one **child** phase. The whole idea of the child phases is to enable mass and energy exchange between the parent phases and account for thermal non-equilibrium in a relatively simple fashion.

Following the conceptualisation of the partial phase equilibrium, the mass and energy conservation can be described (Speranza and Terenzi 2005; Ricci et al. 2015; D'Alessandro et al. 2015; Park et al. 2018) with separate mass conservation equations for parent vapour and liquid phase. See also

(Andreasen and Stegelmann 2025) for a write-up of the differential energy and continuity equations to be integrated.

The solution strategy for the partial phase equilibrium differs somewhat from the homogeneous equilibrium. Instead of the adaptive step size Runge-Kutta method a very simple explicit Euler method with fixed time step is applied. This is mainly due to the solving of the source terms of vapour condensation and liquid evaporation within each time step. As previously described, it is assumed that the equilibrium between parent and child phase is instantaneous. Thus, the moles transported between each main phase is quantifiable, but in order to estimate the rate of the mass transfer, the time step must be exactly known. For numerical stability a relaxation factor of 0.9 is applied to the condensation / evaporation rates.

Further, a combined UV-flash is solved for each phase at each time step. While the total volume is known from the vessel volume, the exact volume of each phase is not known before-hand. Thus, the two UV-flashes are linked by the total volume, given by the volume of the liquid and the volume of gas which must equal the vessel volume. Further, they are linked by a common pressure, while the temperature in each phase is allowed to differ.

5.6 Handling pseudo components

Oil fractions above C7+ are typically lumped into a limited number of pseudo components in order to reduce complexity. This is typically done based on boiling point ranges. In *openthermo* pseudo components can be defined based on *boiling point* and *Specific gravity*. With these two properties the following properties are estimated internally:

- Critical pressure
- Critical temperature
- Critical compressibility
- Critical molar volume
- Acentric factor
- Molecular weight
- HC-ratio

5.6.1 Critical pressure

The critical pressure P_c is estimated with the Kesler-Lee correlation (Kesler and Lee 1976; Ahmed 2007).

$$\ln(P_c) = 8.3634 - \frac{0.0566}{SG} - \left[0.24244 + \frac{2.2898}{SG} + \frac{0.11857}{SG^2}\right] 10^{-3} T_b \tag{5.16}$$

$$+ \left[1.4685 + \frac{3.648}{SG} + \frac{0.47227}{SG^2}\right] 10^{-7} T_b^2 - \left[0.42019 + \frac{1.6977}{SG^2}\right] 10^{-10} T_b^3 \tag{5.17}$$

- SG is the Specific gravity of the fluid at 60 degrees Farenheight [-]
- Tb is the Boiling point the fluid [K]
- Pc is the critical pressure [Pa]

5.6.2 Critical temperature

The critical temperature is estimated using the Kesler-Lee correlation (Kesler and Lee 1976; Ahmed 2007).

$$T_c = 341.7 + 811.1SG + [0.4244 + 0.1174SG]T_b + \frac{[0.4669 - 3.26238SG]10^5}{T_b}$$

- SG is the Specific gravity of the fluid at 60 degrees Farenheight [-]
- Tb is the Boiling point the fluid [K]
- Tc is the crtical temperature [K]

5.6.3 Acentric factor

The accentric factor is estimated the from the Kesler-Lee correlation (Kesler and Lee 1976; Ahmed 2007).

For Tbr > 0.8:

$$\omega = -7.904 + 0.1352K - 0.007465K^2 + 8.359T_{br} + ([1.408 - 0.01063K]/T_{br})$$

Otherwise:

$$\omega = \frac{-\ln \frac{P_c}{14.7} - 5.92714 + \frac{6.09648}{T_{br}} + 1.28862 \ln T_{br} - 0.169347 T_{br}^6}{15.2518 - \frac{15.6875}{T_{br}} - 13.4721 \ln T_{br} + 0.43577 T_{br}^6}$$

$$K = \frac{T_b^{1/3}}{SG}$$

$$T_{br} = \frac{T_b}{T_c}$$

- SG is the Specific gravity of the fluid at 60 degrees Farenheight [-]
- Tb is the Boiling point the fluid [K]
- Tc is the Estimated critical temperature [K]
- Pc is the Estimated critical pressure [Pa]

5.6.4 Critical compressibility

The function calculating critical compresibility [-] for pseudo components is based on (Lee and Kesler 1975).

$$Z_c = 0.2905 - 0.085 * \omega$$

- ω is accentric factor
- Z_c is the critical compressibility

5.6.5 Critical molar volume

The critical volume for pseudo components is based on definition of compresibility.

$$Vc = \frac{Z_c T_c * 8.314}{P_c}$$

- Tc is the Estimated critical temperature [K]
- Pc is the Estimated critical pressure [Pa]
- Zc is the Estimated critical compressibility
- Vc is the Estimated critical molar volume [m3/mol]

5.6.6 Molecular weight

The molecular weight is estimated using the Kesler-Lee correlation (Kesler and Lee 1976; Ahmed 2007).

$$MW = -12272.6 + 9486.4SG + [4.6523 - 3.3287SG]T_b + [1 - 0.77084SG - 0.02058SG^2] \tag{5.18}$$

$$\left[1.3437 - \frac{720.79}{T_b}\right]\frac{10^7}{T_b} + [1 - 0.80882SG + 0.02226SG^2][1.8828 - \frac{181.98}{T_b}]\frac{10^{12}}{T_b^3} \tag{5.19}$$

- SG is the Specific gravity of the fluid at 60 degrees Farenheight [-]
- Tb is the Boiling point the fluid [K]

5.6.7 HC-ratio

The HC atomic ratio is estimated according to [Riazi and Daubert (1986);riazi2005characterization].

The CH weight ratio (Carbon-to-hydrogen ratio) is calculated from:

$$CH = 8.7743 \cdot 10^{-10} \left[\exp 7.176 \cdot 10^{-3} T_b + 30.06242 SG - 7.35 \cdot 10^{-3} TbSG\right] Tb^{-0.98445} SG^{-18.2753}$$

The Hydrogen-to-Carbon ratio is calculated from:

$$HC_{atomic_ratio} = 11.9147/CH$$

- SG is the Specific gravity of the fluid at 60 degrees Farenheight [-]
- Tb is the Boiling point the fluid [K]

6 Supplementing examples

In general for validation against experiments, the paper (Andreasen and Stegelmann 2025) which can also be accessed as preprint is referred. The code for running the simulations matching the experiments are all included in the *test* folder in the GitHUb repo in the file *test-blowdown.py*. A few additional examples and validation cases supplementing these are presented in the following.

6.1 API 521 pool fire

A fictive example is made for an API 521 pool fire scenario for a horizontal vessel (ID 3 m/TT 10 m) half full of liquid. The input details are provided below. The simulation results are compared to simulations performed with the legacy depressuring utility in AspenTech HYSYS ®. The results are shown in Fig. 6.1, Fig. 6.2 and Fig. 6.3. As seen the match between the two programs is excellent. When applying the API 521 pool fire the heat load is added directly to the vessel inventory and vessel wall material temperature is not solved for. In the below example the heat load is applied to the wetted surface area (default setting).

```python
from openthermo.vessel.blowdown import Blowdown

input = {}
P = 12e5
T = 298.15

input["mode"] = "fire"
input["drain_fire_fighting"] = "Inadequate"
input["eos_model"] = "PR"
input["liquid_density"] = "costald"
input["max_time"] = 900
input["length"] = 10
input["diameter"] = 3
input["vessel_type"] = "Flat-end"
input["orientation"] = "horizontal"
input["liquid_level"] = 1.5
input["operating_temperature"] = T
input["operating_pressure"] = P
input["ambient_temperature"] = 273
input["back_pressure"] = 1.01e5
```

```
input["bdv_orifice_size"] = 0.03  # m
input["bdv_orifice_cd"] = 0.84
input['component_names'] =  ["methane", "propane", "n-butane", "i-butane", "n-decane"]
input["molefracs"] =  [0.8, 0.05, 0.01, 0.01, 0.10]

segment = Blowdown(input)
segment.depressurize()
```

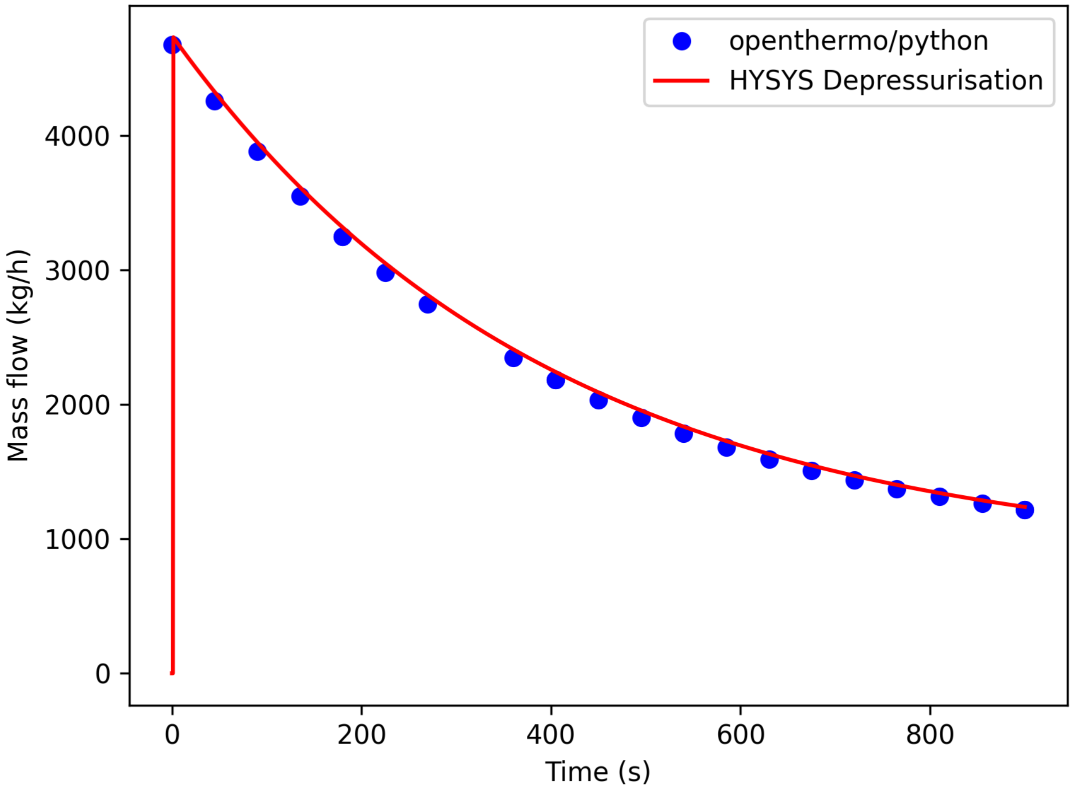

Figure 6.1: Simulation of mass flow as a function of time for vessel subject to API 521 pool fire heat load. Comparison with HYSYS

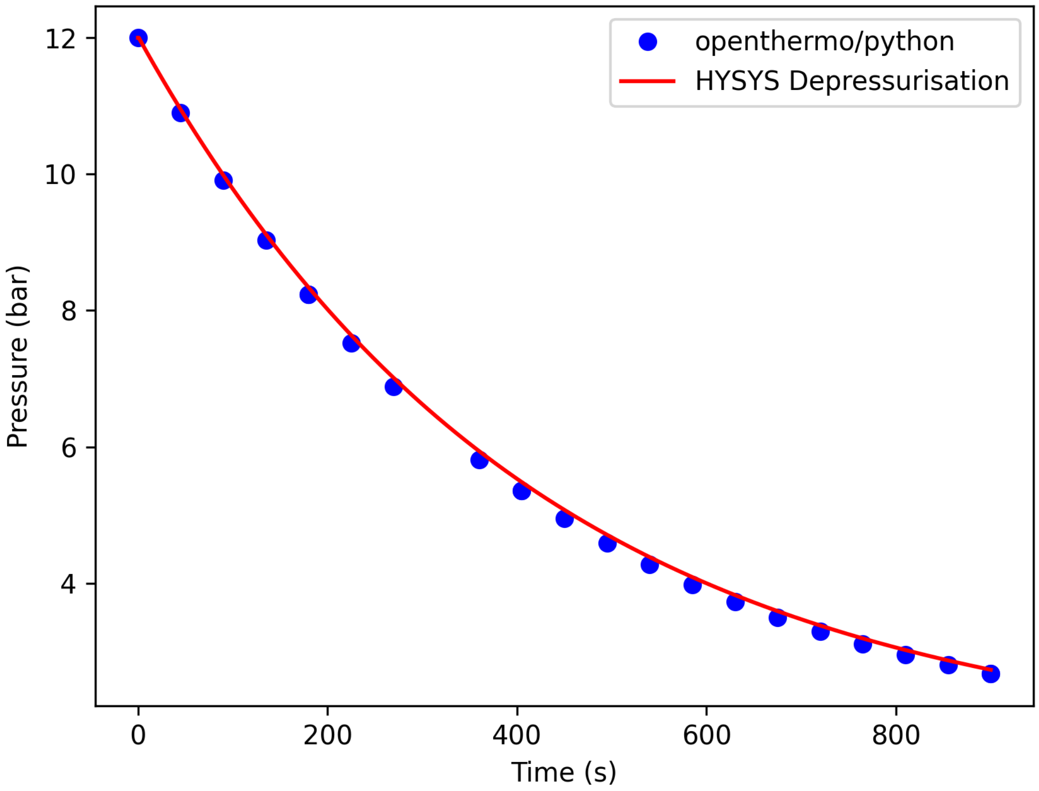

Figure 6.2: Simulation of pressure as a function of time for vessel subject to API 521 pool fire heat load. Comparison with HYSYS

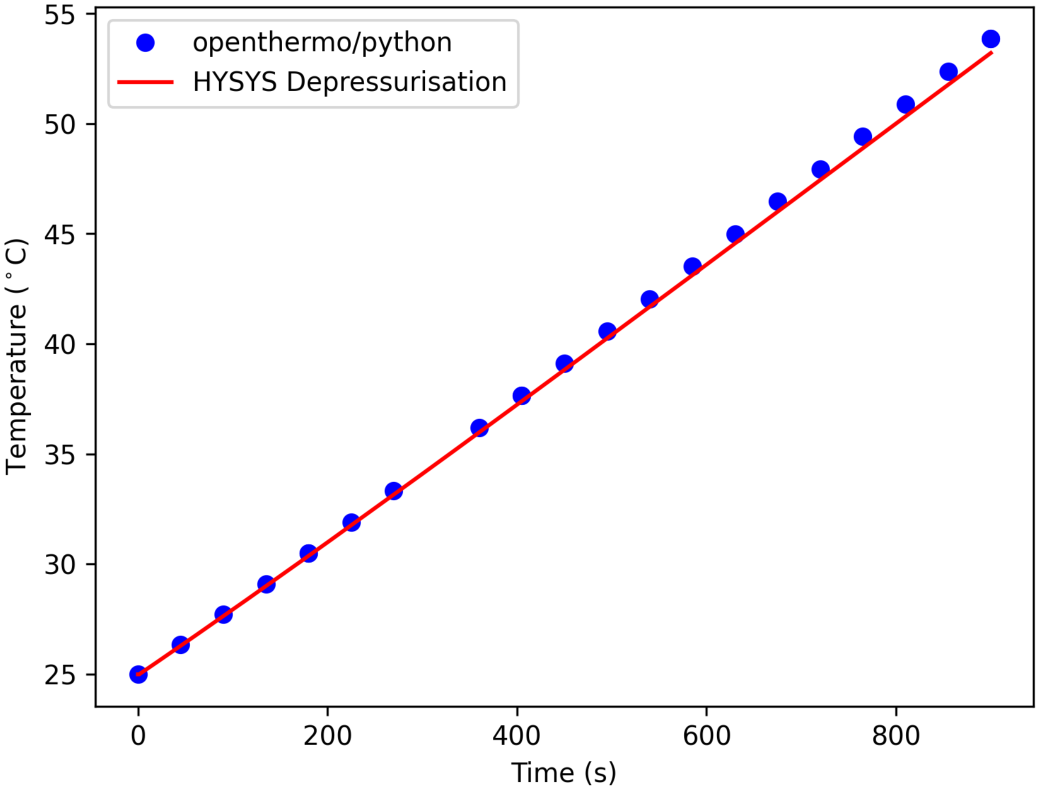

Figure 6.3: Simulation of fluid temperature as a function of time for vessel subject to API 521 pool fire heat load. Comparison with HYSYS

6.2 Stefan-Boltzmann fire heat load

The example investigated in the previous section is modified and subject to a Stefan-Boltzmann fire heat load. In this case a jet-fire backgorund heat load according to the Scandpower guideline (Hekkelstrand and Skulstad 2004). The full input is listed below. As seen the composition of the fluid is different from above.

```python
from openthermo.vessel.blowdown import Blowdown

input["mode"] = "isentropic"
input["heat_transfer"] = "rigorous_sb_fire"
input["sb_fire_type"] = "scandpower_jet"
input["wall_thickness"] = 0.019  # m
input["eos_model"] = "PR"
```

```
input["liquid_density"] = "eos"
input["max_time"] = 500
input["delay"] = 0
input["length"] = 10
input["diameter"] = 3
input["vessel_type"] = "Flat-end"
input["orientation"] = "horizontal"
input["liquid_level"] = 0.27 * 3
input["water_level"] = 0.0
input["operating_temperature"] = T
input["operating_pressure"] = P
input["ambient_temperature"] = 298
input["back_pressure"] = 1.01e5
input["bdv_orifice_size"] = 0.04  # m
input["bdv_orifice_cd"] = 0.84

names = ["methane", "propane", "n-butane", "i-butane", "n-decane"]
molefracs = [
    0.291970802919708,
    1.82481751824818e-2,
    3.64963503649635e-3,
    3.64963503649635e-3,
    0.682481751824818,
]

input["molefracs"] = molefracs
input["component_names"] = names
```

The simulation results are compared to simulations performed with the EO Blowodwn utility in Honeywell Unisim Design ®. The results are shown in Fig. 6.4, Fig. 6.5 and Fig. 6.6. As seen the agreement is generally very good. The difference between the two codes is mainly in terms of the wall heat transfer modelling and the phase equilibrium. In the EO blowdown tool heat conduction through the wall is modelled and there is a differece between the innner and outer wall temperature. This illustrates, when the assumption of a uniform wall temperature works well, and when it works less well. For the wall in contact with vapour, the assumption applied in *openthermo* works very well, since the gradient thorugh the wall is small. For the wall in contact with liquid some discrepancy is observed, especially for the outside temperature. Although in this case, it is of less importance compared to the much higher wall temperature (and more pronounced thermal weakening) for the part in contact with vapour. The EO Blowdown tool also applies a Non-equilibrium/partial equilibrium approch as *openthermo*. However, in *openthermo* the partial equilibirum approach is not yet combinable with the Stefan-Boltzmann fire heat load method. Despite this difference in equilibrium modelling the results are indeed comparable.

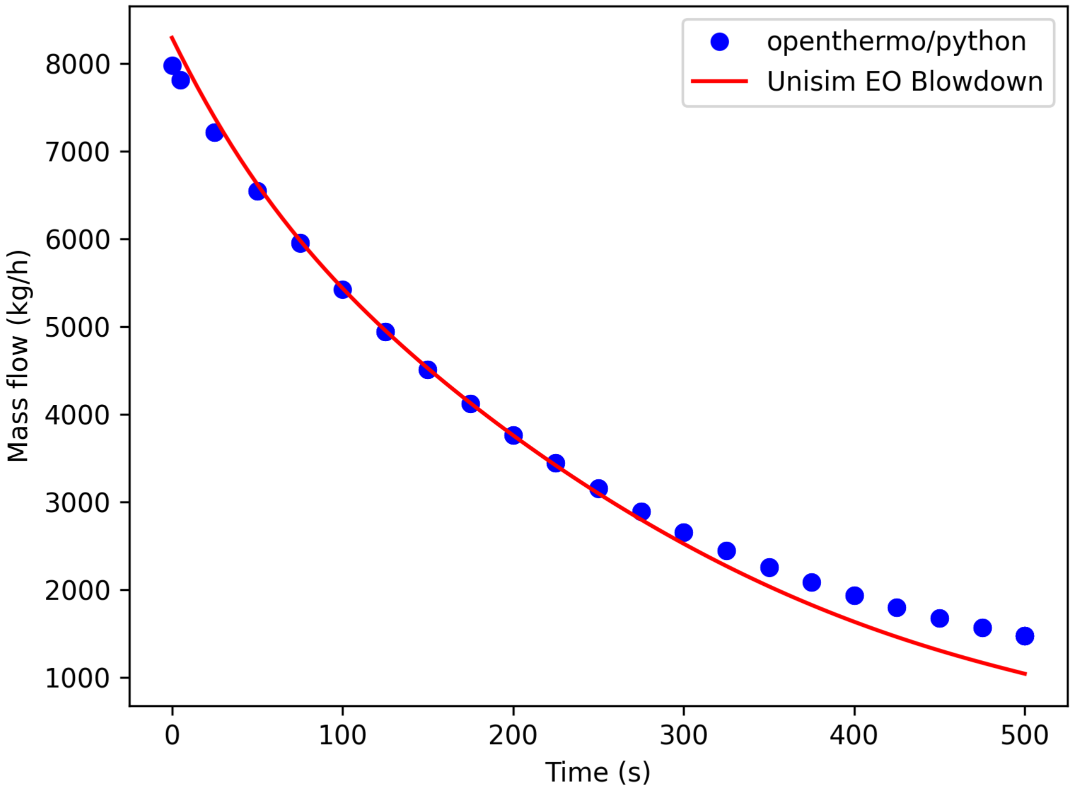

Figure 6.4: Simulation of mass flow as a function of time for vessel subject to Scandpower jet fire heat load. Comparison with Unisim.

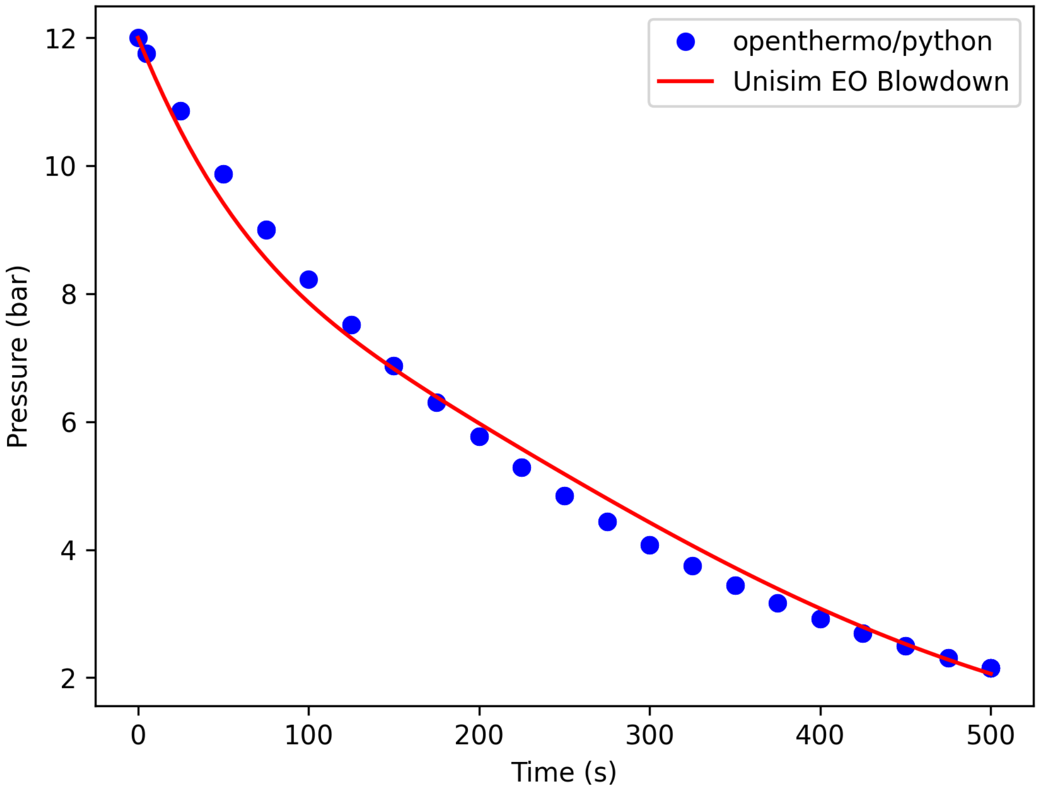

Figure 6.5: Simulation of pressure as a function of time for vessel subject to Scandpower jet fire heat load. Comparison with Unisim.

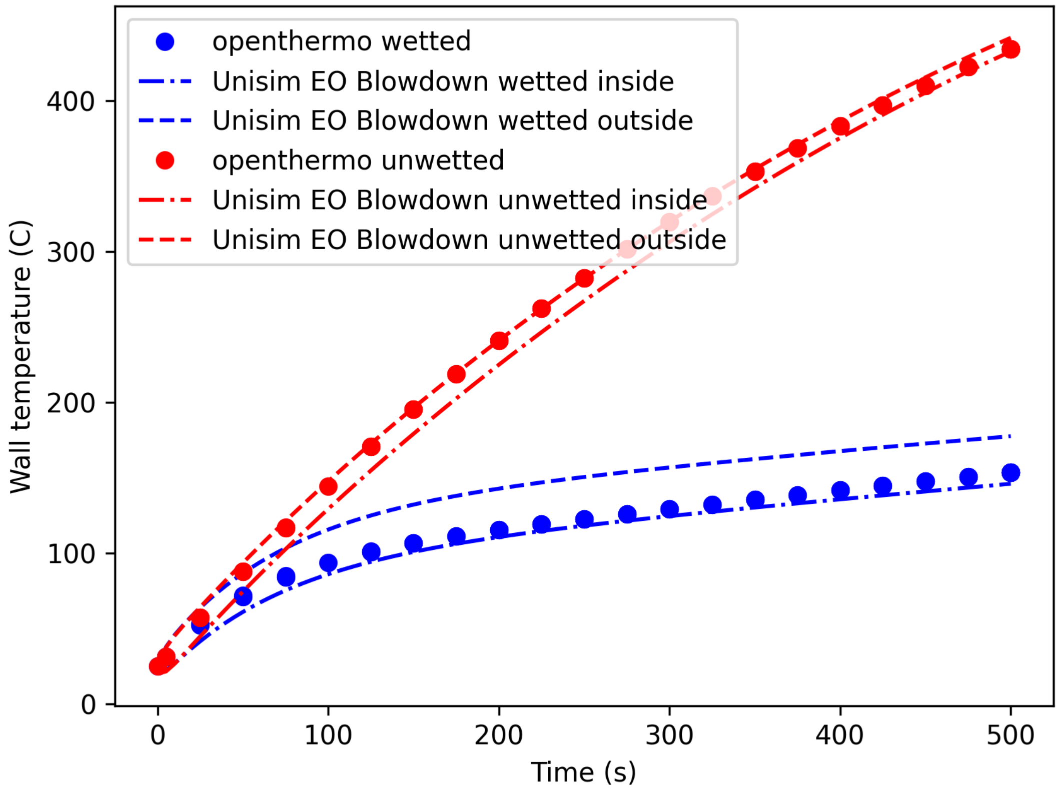

Figure 6.6: Simulation of fluid temperature as a function of time for vessel subject to Scandpower jet fire heat load. Comparison with Unisim.

The heat flux for the Stefan-Boltzmann case for both wetted and unwetted wall is displayed in Fig. 6.7. As seen the heat flux decreases as the wall temperature increses (lower convective heat transfer and more back-radiation). This is more pronounced for the unwetted wall, since it increases more in temperature due to a lower heat tranfer rate internally. It is also noted that as the wetted wall increases in temperature the heat transfer rate also increases significantly, which is due to the nucleate boiling heat transfer type. This signigficant increase in heat transfer coefficeint is responsible for the more moderate temperature increase of the wetted vessel wall. Eventually the nucleate boiling heat transfer becomes comparable to the external heat transfer rate.

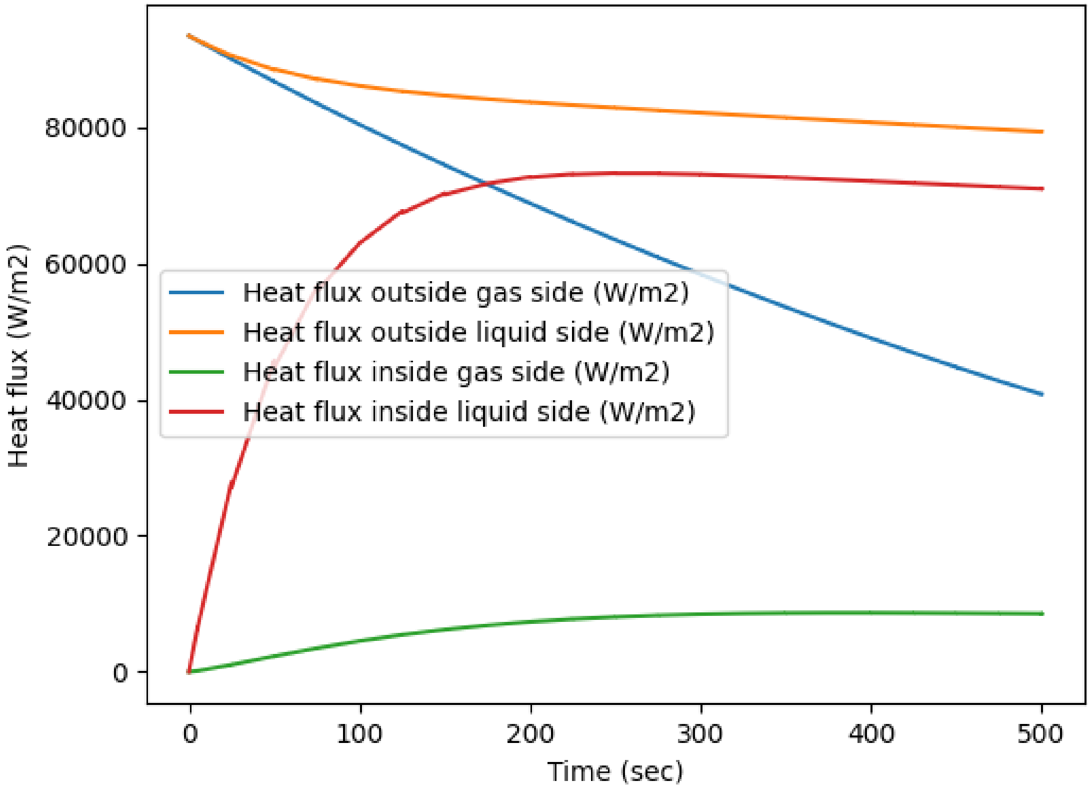

Figure 6.7: Simulation of external and internal heat flux as a function of time for vessel subject to Scandpower jet fire heat load for both the wetted and unwetted part of the vessel.

References

Ahmed, Tarek H. 2007. *Equations of State and PVT Analysis: Applications for Improved Reservoir Modeling*. Gulf Publishing.

Andreasen, Anders. 2021. "HydDown: A Python Package for Calculation of Hydrogen (or Other Gas) Pressure Vessel Filling and Discharge." *Journal of Open Source Software* 6 (66): 3695. https://doi.org/10.21105/joss.03695.

———. 2024. "HydDown – User Guide and Technical Reference." HAL open science. https://hal.science/hal-04858235v2.

Andreasen, Anders, Filippo Borroni, Marcos Zan Nieto, Carsten Stegelmann, and Rudi P. Nielsen. 2018. "On the Adequacy of API 521 Relief-Valve Sizing Method for Gas-Filled Pressure Vessels Exposed to Fire." *Safety* 4 (1). https://doi.org/10.3390/safety4010011.

Andreasen, Anders, and Carsten Stegelmann. 2025. "Open Source Pressure Vessel Blowdown Modelling Under Partial Phase Equilibrium (Under Review)." *Process Safety Progress*. https://doi.org/10.1002/prs.70035.

API. 2014. "Pressure-relieving and Depressuring Systems, API Standard 521." API Standard 521, Sixth Edition, January. American Petroleum Institute.

Bell, Caleb. 2025a. "Fluids: Fluid Dynamics Component of Chemical Engineering Design Library (ChEDL)." https://github.com/CalebBell/fluids.

———. 2025b. "Thermo: Chemical Properties Component of Chemical Engineering Design Library (ChEDL)." https://github.com/CalebBell/thermo.

Bell, Caleb, Pierre Lesouhaitier, VikramGovindarajan, and jeremy rutman. 2024. "CalebBell/Ht: 1.0.8." Zenodo. https://doi.org/10.5281/zenodo.14321704.

Bell, Ian H., Jorrit Wronski, Sylvain Quoilin, and Vincent Lemort. 2014. "Pure and Pseudo-Pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp." *Industrial & Engineering Chemistry Research* 53 (6): 2498–508. https://doi.org/10.1021/ie4033999.

Bjerre, Michael, Jacob G. I. Eriksen, Anders Andreasen, Carsten Stegelmann, and Matthias Mandø. 2017. "Analysis of pressure safety valves for fire protection on offshore oil and gas installations." *Process Safety and Environmental Protection* 105: 60–68. https://doi.org/10.1016/j.psep.2016.10.008.

Bosch, C. J. H. van den, and R. A. P. M. Weterings, eds. 2005. *Methods for the Calculation of Physical Effects (Yellow Book) – CPR 14E*. 3rd ed. Committee for the Prevention of Disasters.

D'Alessandro, Valerio, Giancarlo Giacchetta, Mariella Leporini, Barbara Marchetti, and Alessandro

Terenzi. 2015. “Modelling Blowdown of Pressure Vessels Containing Two-Phase Hydrocarbons Mixtures with the Partial Phase Equilibrium Approach.” *Chemical Engineering Science* 126: 719–29. https://doi.org/10.1016/j.ces.2015.01.019.

Dormand, J. R., and P. J. Prince. 1980. “A Family of Embedded Runge-Kutta Formulae.” *Journal of Computational and Applied Mathematics* 6 (1): 19–26. https://doi.org/10.1016/0771-050X(80)90013-3.

Eriksen, Jacob Gram Iskov, and Michael Skov Bjerre. 2015. “Analysis of the Application and Sizing of Pressure Safety Valves for Fire Protection on Offshore Oil and Gas Installations.” Aalborg University. https://projekter.aau.dk/projekter/files/213880237/Analysis_of_the_application_and_sizing_of_pressure_safety_valves_for_fire_protection_on_offshore_oil_and_gas_installations._By_PECT10_1_F15.pdf.

Gao, Fuchang, and Lixing Han. 2012. “Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters.” *Computational Optimization and Applications* 51 (1): 259–77. https://doi.org/10.1007/s10589-010-9329-3.

Geankoplis, C. J. 1993. *Transport Processes and Unit Operations*. Chemical Engineering. PTR Prentice Hall.

Hairer, Ernst, Gerhard Wanner, and Syvert P. Nørsett. 1993. *Solving Ordinary Differential Equations I*. Vol. 8. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-78862-1.

Hankinson, Risdon W., and George H. Thomson. 1979. “A New Correlation for Saturated Densities of Liquids and Their Mixtures.” *AIChE Journal* 25 (4): 653–63. https://doi.org/10.1002/aic.690250412.

Haque, Afzal, Stephen Richardson, Graham Saville, and Geoffrey Chamberlain. 1990. “Rapid Depressurization of Pressure Vessels.” *Journal of Loss Prevention in the Process Industries* 3 (1): 4–7. https://doi.org/10.1016/0950-4230(90)85015-2.

Haque, M. A., M. Richardson, G. Saville, G. Chamberlain, and L. Shirvill. 1992. “Blowdown of pressure vessels. Pt. II: Experimental validation of computer model and case studie.” *Trans. IChemE B* 70: 10–17.

Haque, M. A., S. M. Richardson, and G. Saville. 1992. “Blowdown of Pressure Vessels, I. Computer Model.” *Process Safety & Environmental Protection: Transactions of the Institution of Chemical Engineers, Part B* 70: 3–10.

Hekkelstrand, B., and P. Skulstad. 2004. *Guidelines for the Protection of Pressurised Systems Exposed to Fire*. Scandpower Risk Management AS.

Jabardo, J. M. Saiz, E. Fockink da Silva, G. Ribatski, and S. F. de Barros. 2004. “Evaluation of the Rohsenow Correlation Through Experimental Pool Boiling of Halocarbon Refrigerants on Cylindrical Surfaces.” *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 26: 218–30. https://doi.org/10.1590/S1678-58782004000200015.

Kesler, M. G., and B. I. Lee. 1976. “Improve Prediction of Enthalpy of Fractions.” *Hydrocarbon Processing*, March, 153–58.

Lee, B. I., and M. G. Kesler. 1975. "A Generalized Thermodynamic Correlation Based on Three-Parameter Corresponding States." *AIChE Journal*. https://doi.org/10.1002/aic.690210313.

Mahgerefteh, Haroun, and Shan M. A Wong. 1999. "A Numerical Blowdown Simulation Incorporating Cubic Equations of State." *Computers & Chemical Engineering* 23 (9): 1309–17. https://doi.org/10.1016/S0098-1354(99)00296-3.

Michelsen, Michael L. 1982a. "The Isothermal Flash Problem. Part I. Stability." *Fluid Phase Equilibria* 9 (1): 1–19. https://doi.org/10.1016/0378-3812(82)85001-2.

———. 1982b. "The Isothermal Flash Problem. Part II. Phase-Split Calculation." *Fluid Phase Equilibria* 9 (1): 21–40. https://doi.org/10.1016/0378-3812(82)85002-4.

Park, Ahmin, Yoonae Ko, Sijin Ryu, and Youngsub Lim. 2018. "Numerical Modeling of Rapid Depressurization of a Pressure Vessel Containing Two-Phase Hydrocarbon Mixture." *Process Safety and Environmental Protection* 113: 343–56. https://doi.org/10.1016/j.psep.2017.10.017.

Peng, Ding-Yu, and Donald B. Robinson. 1976. "A New Two-Constant Equation of State." *Industrial & Engineering Chemistry Fundamentals* 15 (1): 59–64. https://doi.org/10.1021/i160057a011.

Pioro, I. l. 1999. "Experimental Evaluation of Constants for the Rohsenow Pool Boiling Correlation." *International Journal of Heat and Mass Transfer* 42 (11): 2003–13. https://doi.org/10.1016/S0017-9310(98)00294-4.

Reid, C Robert, M John Prausnitz, and E Bruce Poling. 1987. *The Properties of Gases & Liquids*. 4th ed. New York: McGraw-Hill.

Riazi, M. R., and T. E. Daubert. 1986. "Prediction of Molecular Type Analysis of Petroleum Fractions and Coal Liquids." *Industrial and Engineering Chemistry, Process Design and Development* 25 (4): 1009–15.

Ricci, R, V D'Alessandro, S Montelpare, L Binci, and A Zoppi. 2015. "An Unsteady Model for the Simulation of the Rapid Depressurization of Vessels Containing Two-Phase Mixtures in Non-Equilibrium Conditions." *Journal of Physics: Conference Series* 655 (1): 012031. https://doi.org/10.1088/1742-6596/655/1/012031.

Rohsenow, Warren M. 1951. "A Method of Correlating Heat Transfer Data for Surface Boiling of Liquids." Technical {Report}. Cambridge, Mass. : M.I.T. Division of Industrial Cooporation, [1951]. https://dspace.mit.edu/handle/1721.1/61431.

Smith, J. M., H. C. Van Ness, and M. M. Abbott. 1996. *Introduction to Chemical Engineering Thermodynamcis*. Fifth. McGraw-Hill.

Soave, Giorgio. 1972. "Equilibrium Constants from a Modified Redlich-Kwong Equation of State." *Chemical Engineering Science* 27 (6): 1197–1203. https://doi.org/https://doi.org/10.1016/0009-2509(72)80096-4.

Solbraa, Even. 2025. "NeqSim - an Open Source Process Simulation Software." https://equinor.github.io/neqsimhome/.

Speranza, A., and A. Terenzi. 2005. "Blowdown of Hydrocarbons Pressure Vessel with Partial Phase Separation." In *Applied and Industrial Mathematics in Italy*, Volume 69:508–19. Series on Advances in Mathematics for Applied Sciences, Volume 69. WORLD SCIENTIFIC. https://doi.org/10.1142/9789

812701817_0046.

Striednig, Michael, Stefan Brandstätter, Markus Sartory, and Manfred Klell. 2014. “Thermodynamic Real Gas Analysis of a Tank Filling Process.” *International Journal of Hydrogen Energy* 39 (16): 8495–8509. https://doi.org/https://doi.org/10.1016/j.ijhydene.2014.03.028.

Szczepanski, R. 1994. “Simulation programs for blowdown of pressure vessels.” *Proceedings of the IChemE SONG Meeting 12*.

Technica. 1990. “Techniques for Assessing Industrial Hazards: World Bank Technical Paper Number 55.” The International Bank for Reconstruction; Development; https://documents1.worldbank.org/curated/en/55748146

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” *Nature Methods* 17: 261–72. https://doi.org/10.1038/s41592-019-0686-2.

WikiMedia. 2008,. https://commons.wikimedia.org/wiki/File:First_law_open_system.svg.

Wilhelmsen, Øivind, Ailo Aasen, Geir Skaugen, Peder Aursand, Anders Austegard, Eskil Aursand, Magnus Aa. Gjennestad, Halvor Lund, Gaute Linga, and Morten Hammer. 2017. “Thermodynamic Modeling with Equations of State: Present Challenges with Established Methods.” *Industrial & Engineering Chemistry Research* 56 (13): 3503–15. https://doi.org/10.1021/acs.iecr.7b00317.

Wong, Shan Meng Angela. 1998. “DEVELOPMENT OF a MATHEMATICAL MODEL FOR BLOWDOWN OF VESSELS CONTAINING MULTICOMPONENT HYDROCARBON MIXTURES.” Department of Chemical Engineering, University College London. https://discovery.ucl.ac.uk/id/eprint/1317912/1/299471.pdf.

Woodfield, Peter Lloyd, Masanori Monde, and Yuichi Mitsutake. 2007. “Measurement of Averaged Heat Transfer Coefficients in High-Pressure Vessel During Charging with Hydrogen, Nitrogen or Argon Gas.” *Journal of Thermal Science and Technology* 2 (2): 180–91. https://doi.org/10.1299/jtst.2.180.