HydDown Documentation

Release 0.40.0

Anders Andreasen

Mar 25, 2026

USER GUIDE

HydDown is an open source Python package for calculating hydrogen (or other pure gas phase species) pressure vessel filling and discharge incorporating heat transfer effects.

Download this documentation: PDF | EPUB

CHAPTER

ONE

INTRODUCTION

HydDown is an open source python3 tool for calculation of hydrogen (or other pure component) vessel/container depressurization and filling. The HydDown logo shown in Figure 1.1 visualizes the key parameters and transport phenomena during gas vessel filling or discharging. 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 either in or 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.

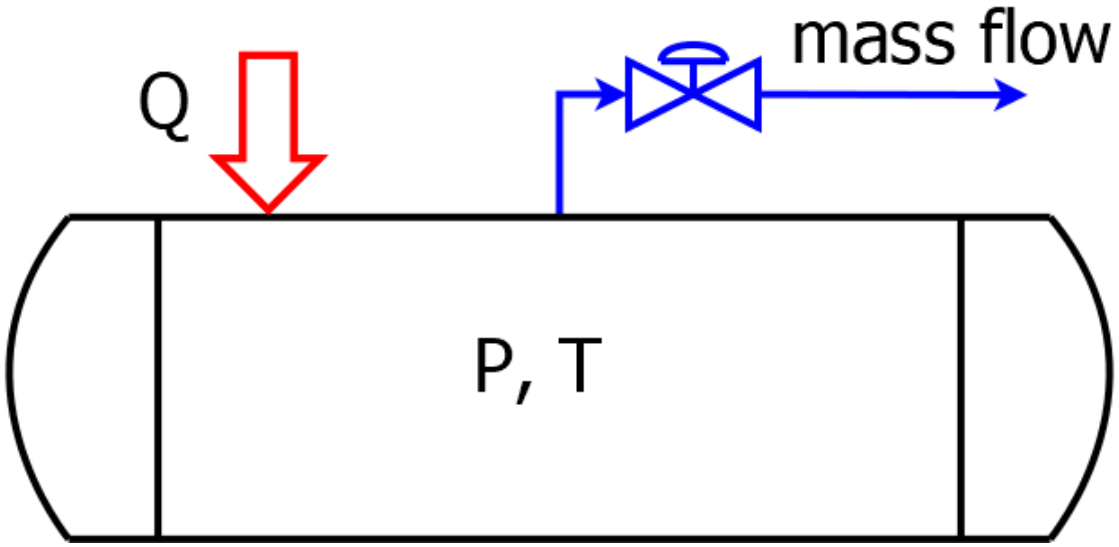


Figure 1.1: HydDown logo

Running the code is as simple as:

```
python main.py input.yml
```

where `main.py` is the main script and `input.yml` is the input file in Yaml syntax.

1.1 Citing HydDown

If you use HydDown please cite the following reference [And21]:

Andreasen, A., (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

```
@article{Andreasen2021,
  doi = {10.21105/joss.03695},
  url = {https://doi.org/10.21105/joss.03695},
  year = {2021},
  publisher = {The Open Journal},
  volume = {6},
  number = {66},
  pages = {3695},
  author = {Anders Andreasen},
  title = {HydDown: A Python package for calculation of hydrogen (or other gas) pressure␣
↪vessel filling and discharge},
  journal = {Journal of Open Source Software}
}
```

This manual can be cited as [And24]:

Anders Andreasen. HydDown - User guide and technical reference. 2024. ⟨hal-04858235⟩

```
@report{Andreasen2024,
  url = {https://hal.science/hal-04858235},
  year = {2024},
  publisher = {HAL open science},
  author = {Anders Andreasen},
  title = {HydDown -- user guide and technical reference},
}
```

1.2 Background

HydDown started as a small spare-time project for calculation of vessel filling and depressurization behaviour. This was mainly to demonstrate that although perceived as a very tedious and difficult task to write your own code for such an apparently complex problem, actually only a fairly limited amount of code is necessary if you have a good thermodynamic backend. The code has evolved to a point where additional features will increase the complexity to the point where it is no longer a simple tool.

A few choices has been made to keep things simple:

- Coolprop is used as thermodynamic backend
- Only pure substances are considered (limited multi-component capabilities are included)
- No temperature stratification in the gas phase
- A default of of no temperature gradient through vessel wall.
- Heat transfer is modelled as constant or simplified using empirical correlations

These choices make the problem much simpler to solve: First of all, the the pure substance Helmholtz energy based equation of state (HEOS) in `coolprop` offers a lot of convenience in terms of the property pairs/state variables that can be set independently. Using only a single gas phase species also means that component balances is redundant and 2 or 3-phase flash calculations are not required. That being said, the principle used for a single component is more or less the same, even for multicomponent mixtures with potentially more than one phase.

In the latest revision 1-D transient heat conduction through the vessel wall is now an option if required for low thermal conductivity materials and e.g. type III/IV vessels. The 1-D heat transfer model is fully integrated with fire heat load calculations (Stefan-Boltzmann method) and two-phase flow scenarios. Rigorous two-phase single component calculations are supported implementing an equilibrium assumption i.e.

gas and liquid phase are in thermal equilibrium, although the vessel wall in contact with gas and liquid, respectively, is allowed to differ in temperature. For two-phase systems, the 1-D model solves separate heat conduction problems for the wetted (liquid-contact) and unwetted (gas-contact) wall regions.

In case multi-component two-phase behaviour is required, the HydDown sibling ORS *openthermo* is recommended. This software package also implements a partial/non-equilibrium assumption for the gas and liquid phase [AS25].

```
@article{https://doi.org/10.1002/prs.70035,
  author = {Andreasen, Anders and Stegelmann, Carsten},
  title = {Open source pressure vessel blowdown modeling under partial phase equilibrium}
↪,
  journal = {Process Safety Progress},
  doi = {https://doi.org/10.1002/prs.70035},
}
```

A preprint of the paper is also available.

1.3 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/andr1976/HydDown

Clone the repo by:

```
git clone https://github.com/andr1976/HydDown.git
```

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

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

Installation of latest release via **pip** also installs dependencies automatically:

```
pip install hyddown
```

In case `pip` links to a v2.7 of python you will get an error. If so try the following:

```
python3 -m pip install hyddown
```

where python3 is the symlink or full path to the python3 executable installed on your system.

1.4 Requirements

- Python (Check CoolProp for latest supported python version)
- Numpy
- matplotlib
- Coolprop (6.4.1)
- cerberus
- PyYaml

- pandas
- Scipy
- tqdm

The script is running on Windows 10 x64, with stock python 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.

1.5 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. In writing 33+ test should pass.

For the package installed with pip navigate to the install directory `../site-packages/hyddown` and run:

```
python -m pytest
```

1.6 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 is 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 the table. 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.

1.7 Credit

In the making of this document a great deal of material has been sourced (and modified) from a good colleague's M.Sc. thesis [EB15], from co-published papers [ABZN+18, BEA+17]and from on-line material published under permissive licenses (with proper citation). Further, the making of this project would not have possible without the awesome CoolProp library [BWQL14]. I am also thankful for enlightening discussions with colleague Jacob Gram Iskov Eriksen (Ramboll Energy, Denmark) and former Ramboll Energy colleague Carsten Stegelmann (ORS Consulting) in relation to vessel depressurization, nozzle flow and heat transfer considerations.

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

1.8 License

CHAPTER

TWO

USAGE

2.1 Basic usage

From the source repository, running the code is as simple as:

```
python src/main.py input.yml
```

where main.py is the main script and input.yml is the input file in Yaml syntax.

The Yaml input file is edited to reflect the system of interest.

Further, a usable copy of a main script is installed in the python installation Scripts/ folder:

```
/python3x/Scripts/hyddown_main.py
```

or from GitHub/in the source folder:

```
HydDown/Scripts/hyddown_main.py
```

Various example files are included for inspiration under site-packages:

```
/site-packages/hyddown/examples/
```

or in the git/source directory tree

```
HydDown/src/hyddown/examples/
```

2.2 Demos

A few demonstrations of the codes capability are available as `streamlit` apps:

- Vessel gas pressurisation/depressurization for various gases at https://share.streamlit.io/andr1976/hyddown/main/scripts/streamlit_app.py
- Response to external fire of vessel equipped with a PSV using the Stefan-Boltzmann fire equation athttps://share.streamlit.io/andr1976/hyddown/main/scripts/streamlit_sbapp.py
- Vessel depressurization including slow opening of valve at https://share.streamlit.io/andr1976/hyddown/main/scripts/streamlit_cvapp.py

The various streamlit apps are also included in the scripts folder for running the scripts on a local machine.

2.3 Calculation methods

The following methods are implemented:

- Isothermal: constant temperature of the fluid during depressurization (for a very slow process with a large heat reservoir)
- Isenthalpic: constant enthalpy of the fluid, no heat transfer with surroundings, no work performed by the expanding fluid for depressurization, no work added from inlet stream
- Isentropic: constant entropy of the fluid, no heat transfer with surroundings, PV work performed by the expanding fluid
- Isenergetic: constant internal energy of the fluid
- Energy balance: this is the most general case and is based on the first law of thermodynamics applied to a flow process.
- Relief: Relief valve dimensioning for gas filled vessels subject to fire. This method provides a dynamic approach to relief valve dimensioning providing realistic orifice size, compared to the very conservative API521 approach.

For `isothermal/isenthalpic/isentropic/isenergetic` calculations the minimal input required are:

- Initial conditions (pressure, temperature)
- vessel dimensions (ID, length)
- valve parameters (Cd, diameter, back-pressure)
- Calculation setup (time step, end time)
- Type of gas

If heat transfer is to be considered the calculation type `energybalance` is required. A few options are possible:

- Fixed U: U-value, i.e. overall heat transfer coefficient, is required and ambient temperature required. Based on the ambient (external) temperature, the fluid temperature, and the U-value, the heat flux Q is calculated for each time step. In this calculation method the temperature of the vessel wall is not calculated:

$$Q = UA(T_{ambient} - T_{fluid})$$

- Fixed Q: The external heat flux, Q, applied to the fluid is specified and constant. The ambient temperature is not required. Further, the vessel wall temperature is redundant and not calculated.
- Specified h: The external heat transfer coefficient must be specified and either the internal heat transfer coefficient is provided or calculated from the assumption of natural convection from a vertical cylinder at high Gr number. Ambient temperature is required. Using this method the wall temperature is calculated from an energy balance over the vessel wall taking in and out flux to/from the external ambient plenum as well as heat flux to/from the fluid inventory into account.
- Fire: The Stefan-Boltzmann equation is applied for estimating the external heat duty. The fire heat flux depends on the vessel wall temperature, and the wall temperature is continuously updated as in the method with specified h.

More elaborate description of the required input for the different calculation types are provided in *Input fields and hierarchy*.

2.4 Script

HydDown comes with a script which can be used as a command-line tool to start calculations. If an input filename (path) is given as the first argument, this input file will be used. If no arguments are passed, the script will look for an input file with the name `input.yml`. The content of the main script is shown below:

```
import yaml
import sys
from hyddown import HydDown

if __name__ == "__main__":
    if len(sys.argv) > 1:
        input_filename = sys.argv[1]
    else:
        input_filename = "input.yml"

    with open(input_filename) as infile:
        input = yaml.load(infile, Loader=yaml.FullLoader)


    hdown=HydDown(input)
    hdown.run()
    hdown.verbose=1
    hdown.plot()
```

2.5 Module import

To use HydDown simple import the main calculation class `HydDown`.

```
from hyddown import HydDown
```

2.6 Input file examples

When using HydDown a dictionary holding all relevant input in order for HydDown to do vessel calculations shall be provided when the class is initialized. One way is to read an input file. For HydDown a Yaml format is chosen, but if JSON is a preference this should also work with the Yaml parser being substituted with a JSON parser.

An example of a minimal file for an isentropic vessel depressurization (no heat transfer) is shown below

```
vessel:
  length: 1.524
  diameter: 0.273
initial:
  temperature: 388.0
  pressure: 15000000.
  fluid: "N2"
calculation:
  type: "isentropic"
  time_step: 0.05
  end_time: 100.
valve:
```

(continues on next page)

(continued from previous page)

```
  flow: "discharge"
  type: "orifice"
  diameter: 0.00635
  discharge_coef: 0.8
  back_pressure: 101300.
```

A more elaborate example which includes heat transfer and with validation data (some data points dropped for simplicity) included:

```
vessel:
  length: 1.524
  diameter: 0.273
  thickness: 0.025
  heat_capacity: 500
  density: 7800.
  orientation: "vertical"
initial:
  temperature: 288.0
  pressure: 15000000.
  fluid: "N2"
calculation:
  type: "energybalance"
  time_step: 0.05
  end_time: 100.
valve:
  flow: "discharge"
  type: "orifice"
  diameter: 0.00635
  discharge_coef: 0.8
  back_pressure: 101300.
heat_transfer:
  type: "specified_h"
  temp_ambient: 288.
  h_outer: 5
  h_inner: 'calc'
validation:
  temperature:
    gas_high:
      time: [0.050285, ... , 99.994]
      temp: [288.93, ... ,241.29]
    gas_low:
      time: [0.32393, ..., 100.11]
      temp: [288.67, ... ,215.28]
    wall_low:
      time: [0.32276, ... , 100.08]
      temp: [288.93, ... ,281.72]
    wall_high:
      time: [0.049115, ... ,100.06]
      temp: [289.18, ... ,286.09]
  pressure:
    time: [0.28869, ... , 98.367]
    pres: [150.02, ... ,1.7204]
```

2.7 Input fields and hierarchy

In the following the full hierarchy of input for the different calculation types is summarised.

At the top level the following fields are accepted, with the last being optional and the second last dependant on calculation type:

```
initial: mandatory
vessel: mandatory
calculation: mandatory
valve: mandatory
heat_transfer: depends on calculation type
validation: optional
```

2.7.1 Calculation

The subfields under `calculation`, with value formats and options are:

```
calculation:
  type: "isothermal", "isentropic", "isenthalpic", "constantU", "energybalance"
  time_step: number
  end_time: number
```

The simulation end time is specified as well as the fixed time step used in the integration of the differential equations to be solved. The four main calculation types are shown as well.

2.7.2 Vessel

```
vessel:
  length: number, mandatory
  diameter: number, mandatory
  thickness: number, required when heat transfer is calculated
  heat_capacity: number, required when heat transfer is calculated
  density: number, required when heat transfer is calculated
  thermal_conductivity: number, required for 1-D transient heat transfer
  thermal_conductivity_biot: number, optional for Biot number calculation in lumped␣
↪models
  liner_thickness: number, required only for bi-material 1-D transient heat transfer
  liner_heat_capacity: number, required only for bi-material 1-D transient heat transfer
  liner_density: number, required only for bi-material 1-D transient heat transfer
  liner_thermal_conductivity: number, required only for bi-material 1-D transient heat␣
↪transfer
  orientation: string, required when heat transfer is calculated
  liquid_level: number, optional for two-phase
  type: string, optional for heads other than Flat-end, "DIN", "ASME F&D", or
↪"Hemispherical"
```

2.7.3 Initial

```
initial:
  temperature: number, mandatory
  pressure: number, mandatory
  fluid: string, mandatory , e.g. "N2", "H2"
```

2.7.4 Valve

The `valve` field determines the mode of the process is it for depressurization/discharge from the vessel using the value `discharge` or if mass flow is entering the vessel, filling it up/increasing pressure with the value `filling`.

Different types of mass flow devices can be specified:

- Restriction `orifice`
- Pressure safety valve `psv` (only for `discharge` not `filling`)
- Simplified relief valve `relief` (only for `discharge` not `filling`)
- Control valve `controlvalve`
- A specified mass flow `mdot`
- Homogeneous equilibrium model release `hem_release`

For the physical devices a `back_pressure` is required for the flow calculations. The value of the `back_pressure` **is also used to specify the reservoir pressure when the vessel is filled**. See also *Flow devices* for details about the calculation of flow rates.

```
valve:
  flow: string, mandatory "discharge" or "filling"
  type: string, mandatory "orifice", "controlvalve", "psv", "relief", "mdot", "hem_
↪release"
  back_pressure: number, required for type "orifice", "controlvalve", "psv", "relief"
  diameter: number, required for "orifice" and "psv"
  discharge_coef: number, required for "orifice" and "psv"
  Cv: number, required for "controlvalve"
  characteristic:  string, optional for "controlvalve"
  time_constant: number, optional for "controlvalve"
```

For the calculations using a control valve as mass transfer device a linear rate for the actuator can be specified using the `time_constant` field. An associated valve characteristic is associated in order to translate the linear actuator rate to an opening Cv (of max. Cv). Three caracteristics can be specified:

- Linear
- Equal percentage (exponential)
- Quick opening/fast (square root)

2.7.5 Heat transfer

For more information about the actual estimation of heat transfer see also *Heat transfer*. For the case of `fire` heat input predetermined parameters for the Stefan-Boltzmann fire equation are used to calculate different background heat loads cf. the table below

Table 2.1: Fire heat loads

Source	Fire type	Heat load (kW/m^2)
API521	Pool	60
API521	Jet	100
Scandpower	Pool	100
Scandpower	Jet	100

```
heat_transfer:
  type: string, mandatory, "specified_h", "specified_Q", "specified_U", "s-b"
  temp_ambient: number, required for type "specified_h", "specified_U"
  h_outer: number, required for type "specified_h"
  h_inner: number or 'calc', required for type "specified_h"
  U_fix: number, required for type "specified_U"
  Q_fix: number, required for type "specified_Q"
  fire: string, required for type "s-b", 'api_pool','api_jet','scandpower_pool',
↪'scandpower_jet'
  D_thoat: number, required for flow type "filling", set to vessel ID as a starting point
  scaling: number between 0 and 1, optional, scales total heat load
```

2.7.6 Validation

In order to plot measured data against simulated data the field `validation` is included.

The following arrays (one or more) are supported in the sub-field `temperature`:

- `gas_high`: highest measured values of the bulk gas
- `gas_low`: lowest measured values of the bulk gas
- `gas_mean`: average measured values of the bulk gas
- `wall_mean`: average measured values of the vessel (inner) wall
- `wall_high` i.e. highest measured values of the vessel (inner) wall
- `wall_low` i.e. lowest measured values of the vessel (inner) wall
- `wall_outer` i.e. external measured values of the vessel (outer) wall
- `wall_inner` i.e. internal measured values of the vessel (inner) wall

For each of the above fields arrays for `time`and `temp`shall be supplied with matching length.

A field for measured vessel pressure is also possible, where `time`and `pres` shall be identical length arrays. See also example below.

```
validation:
  temperature:
    gas_high:
      time: [0.050285, ... , 99.994]
      temp: [288.93, ... ,241.29]
    gas_low:
      time: [0.32393, ..., 100.11]
      temp: [288.67, ... ,215.28]
    wall_low:
      time: [0.32276, ... , 100.08]
      temp: [288.93, ... ,281.72]
    wall_high:
      time: [0.049115, ... ,100.06]
      temp: [289.18, ... ,286.09]
  pressure:
    time: [0.28869, ... , 98.367]
    pres: [150.02, ... ,1.7204]
```

CHAPTER

THREE

THEORY

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

- thermodynamics
- mass transfer
- heat transfer

3.1 Thermodynamics

3.1.1 Equation of state

The equation of state used by HydDown is the Helmholtz energy formulation as implemented in CoolProp [BWQL14]. Most of the text in the present section has been sourced from the CoolProp documentation to be as accurate and true to the source as possible. The Helmholtz energy formulation is a convenient construction of the equation of state because all the thermodynamic properties of interest can be obtained directly from partial derivatives of the Helmholtz energy.

It should be noted that the EOS are typically valid over the entire range of the fluid, from subcooled liquid to superheated vapor, to supercritical fluid. In general, the EOS are based on non-dimensional terms δ and τ, where these terms are defined by

$$\delta = \rho/\rho_c$$

$$\tau = T_c/T$$

where ρ_c and T_c are the critical density of the fluid if it is a pure fluid. For pseudo-pure mixtures, the critical point is typically not used as the reducing state point, and often the maximum condensing temperature on the saturation curve is used instead.

The non-dimensional Helmholtz energy of the fluid is given by

$$\alpha = \alpha^0 + \alpha^r$$

where α^0 is the ideal-gas contribution to the Helmholtz energy, and α^r is the residual Helmholtz energy contribution which accounts for non-ideal behavior. For a given set of δ and τ, each of the terms α^0 and α^r are known. The exact form of the Helmholtz energy terms is fluid dependent, but a relatively simple example is that of Nitrogen, which has the ideal-gas Helmholtz energy of

$$\alpha^0 = \ln\delta + a_1 \ln\tau + a_2 + a_3\tau + a_4\tau^{-1} + a_5\tau^{-2} + a_6\tau^{-3} + a_7 \ln[1 - \exp(-a_8\tau)]$$

and the non-dimensional residual Helmholtz energy of

$$\alpha^r = \sum_{k=1}^{6} N_k \delta^{i_k} \tau^{j_k} + \sum_{k=7}^{32} N_k \delta^{i_k} \tau^{j_k} \exp(-\delta^{l_k}) + \sum_{k=33}^{36} N_k \delta^{i_k} \tau^{j_k} \exp(-\phi_k(\delta-1)^2 - \beta_k(\tau-\gamma_k)^2)$$

and all the terms other than δ and τ are fluid-dependent correlation parameters.

The other thermodynamic parameters can then be obtained through analytic derivatives of the Helmholtz energy terms. For instance, the pressure is given by

$$p = \rho RT \left[1 + \delta \left(\frac{\partial \alpha^r}{\partial \delta}\right)_\tau\right]$$

and the specific internal energy by

$$\frac{u}{RT} = \tau \left[\left(\frac{\partial \alpha^0}{\partial \tau}\right)_\delta + \left(\frac{\partial \alpha^r}{\partial \tau}\right)_\delta\right]$$

and the specific enthalpy by

$$\frac{h}{RT} = \tau \left[\left(\frac{\partial \alpha^0}{\partial \tau}\right)_\delta + \left(\frac{\partial \alpha^r}{\partial \tau}\right)_\delta\right] + \delta \left(\frac{\partial \alpha^r}{\partial \delta}\right)_\tau + 1$$

which can also be written as

$$\frac{h}{RT} = \frac{u}{RT} + \frac{p}{\rho RT}$$

The specific entropy is given by

$$\frac{s}{R} = \tau \left[\left(\frac{\partial \alpha^0}{\partial \tau}\right)_\delta + \left(\frac{\partial \alpha^r}{\partial \tau}\right)_\delta\right] - \alpha^0 - \alpha^r$$

and the specific heats at constant volume and constant pressure respectively are given by

$$\frac{c_v}{R} = -\tau^2 \left[\left(\frac{\partial^2 \alpha^0}{\partial \tau^2}\right)_\delta + \left(\frac{\partial^2 \alpha^r}{\partial \tau^2}\right)_\delta\right]$$

$$\frac{c_p}{R} = \frac{c_v}{R} + \frac{\left[1 + \delta \left(\frac{\partial \alpha^r}{\partial \delta}\right)_\tau - \delta\tau \left(\frac{\partial^2 \alpha^r}{\partial \delta \partial \tau}\right)\right]^2}{\left[1 + 2\delta \left(\frac{\partial \alpha^r}{\partial \delta}\right)_\tau + \delta^2 \left(\frac{\partial^2 \alpha^r}{\partial \delta^2}\right)_\tau\right]}$$

The EOS is set up with temperature and density as the two independent properties, but often other inputs are known, most often temperature and pressure because they can be directly measured. As a result, if the density is desired for a known temperature and pressure, it can be obtained iteratively.

3.1.2 First law for flow process

The control volume sketched in Figure 3.1, 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 [SVNA96]

A general mass balance or continuity equation can be written:

$$\frac{m_{cv}}{dt} + \Delta \left(\dot{m}\right)_{fs} = 0 \tag{3.1}$$

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

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

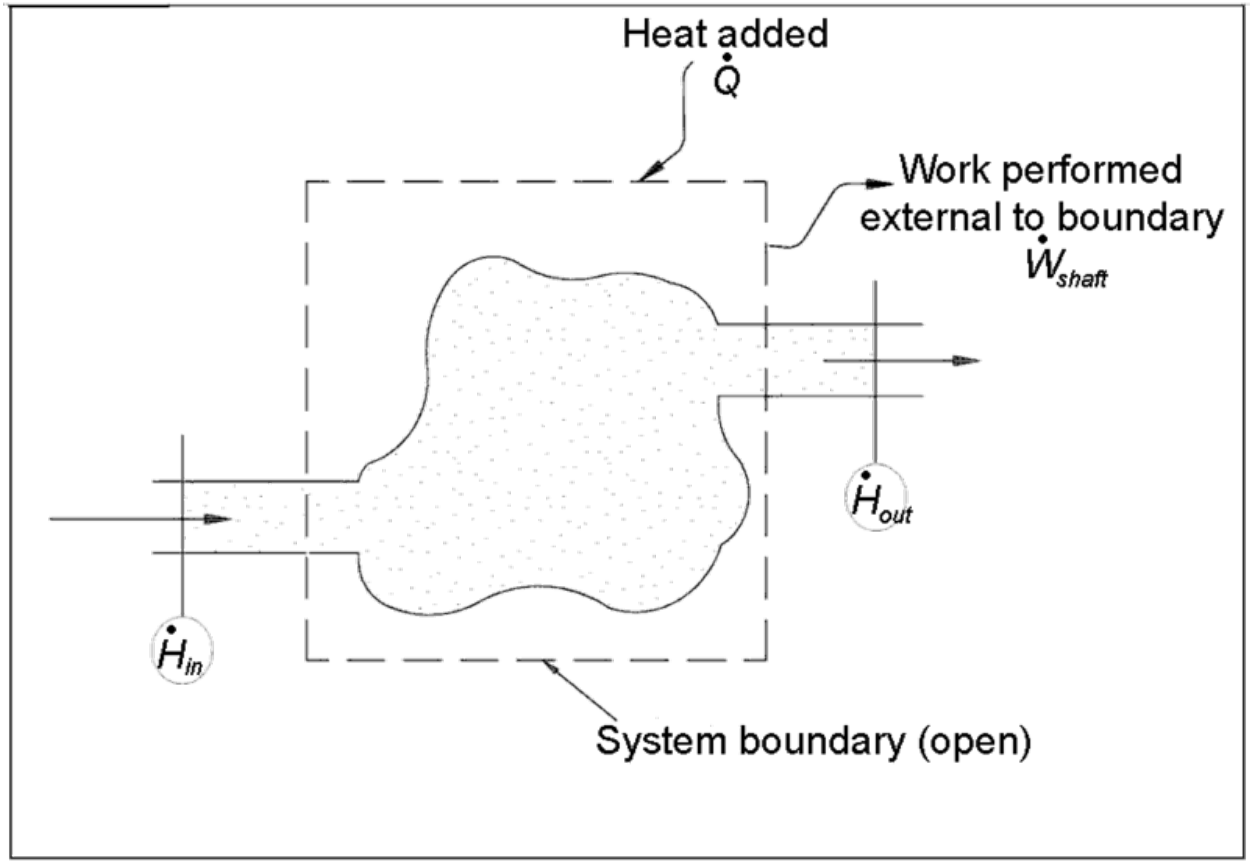


Figure 3.1: Control volume with one entrance and one exit. The image has been sourced from [Wik08].

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

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

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

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

and in total

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

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:

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

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:

$$\frac{d(mU)_{cv}}{dt} + \dot{m}H = \dot{Q} \tag{3.2}$$

where the sign of $\dot{m}$ determines if the control volume is either emptied or filled. The continuity equation (3.1) and the energy balance (3.2) 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.

3.2 Flow devices

3.2.1 Restriction Orifice

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 (3.3).

$$\frac{P_c}{P_u} = \left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} \tag{3.3}$$

- 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 [vdBW05].

To account for the difference in choked and non-choked flow a set limit pressure is introduced as in equation (3.4). 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 (3.5) 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} [vdBW05].

$$P_{limit} = P_{up} \cdot \left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} \tag{3.4}$$

$$\dot{m}_{flow} = C_d \cdot A \cdot \sqrt{\left(\frac{2k}{k-1}\right) \cdot P_{up} \cdot \rho \cdot \left(\frac{P_{used}}{P_{up}}\right)^{\frac{2}{k}} \left(1 - \left(\frac{P_{used}}{P_{up}}\right)^{\frac{k-1}{k}}\right)} \tag{3.5}$$

- 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. $[m^2]$

3.2.2 HEM orifice release

The HEM (homogeneous equilibrium model) assumption is based on the general steady-flow energy balance for isentropic nozzle flow. The fundamental assumption is that thermal and mechanical equilibrium exist between phases as the fluid passes through the release orifice.

Starting from the steady-flow energy balance (neglecting potential energy and friction):

$$h_0 = h_1 + \frac{v^2}{2}$$

where:

- h_0 = upstream stagnation enthalpy
- h_1 = fluid enthalpy at reduced pressure
- v = fluid velocity

Rearranging for velocity:

$$v = \sqrt{2(h_0 - h_1)}$$

The mass flux (mass flow per unit area) is:

$$G = \rho \cdot v = \rho\sqrt{2(h_0 - h_1)}$$

where ρ is the fluid density at the reduced pressure conditions.

For an isentropic expansion from upstream pressure P_0 to various downstream pressures, the maximum mass flux occurs either at:

1. The critical flow pressure (choked flow)
2. The backpressure (unchoked flow)

This can be expressed as:

$$G_{max} = \max\left[\rho(P)\sqrt{2\int_{P_0}^{P} v\, dP}\right]$$

where the integral is evaluated along an isentropic path, and v is the specific volume.

3.2.3 Pressure safety valve / Relief valve

A PSV / relief valve is a mechanical device actuated by the static pressure in the vessel and a conventional PSV is often used for gas/vapor systems. A conventional PSV is a spring-loaded device which will activate at a predetermined opening pressure, and relieve the vessel pressure until a given reseat pressure has been reached. Both the opening pressure and the reset pressure is above the vessel operating pressure and the PSV will remain closed until the pressure inside the vessel increases to the opening pressure. The operation of a conventional spring-loaded PSV is based on a force balance. A conventional PSV can be seen in Figure 3.2. A spring exerts a force on a disc blocking the inlet of the PSV. When the pressure inside the vessels reaches the opening pressure, the force exerted on the disc by the gas will be larger than the force exerted by the spring and the PSV will open and allow the gas to flow out of the vessel. The flow of gas out of the vessels will lower the pressure and thereby also the force exerted on the disc. When the pressure in the vessels is reduced to the reset pressure, the PSV will close and the disc will again hinder the gas flow.

The relief valve model implemented in HydDown is the API 520 equations [API14a] for gas relief for both sonic/critical as well as subcritical flow. No corrections factors are implemented in HydDown.

For sonic flow (critical flow), as indicated in equation, the mass flow through the PSV can be determined by equation (3.6).

$$W = \frac{AC \cdot K_d \cdot K_b \cdot K_c \cdot P_1}{\sqrt{\frac{T \cdot Z}{M}}} \tag{3.6}$$

- A is the effective discharge area. [mm^2]
- W is the mass flow through the device. [kg/h]
- C is a coefficient, as a function of k, as defined in equation (3.7).
- K_d, K_b, and K_c are correction factors.

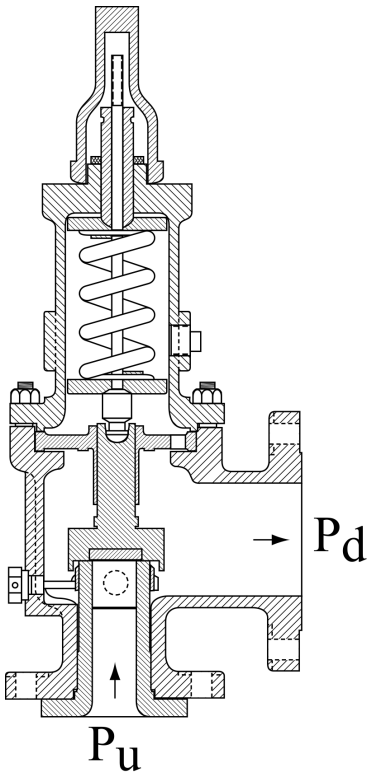


Figure 3.2: Conventional/pop action PSV adapted from [EB15] and [API14a]

- P_1 is the allowable upstream absolute pressure. [kPa]
- T is the temperature of the inlet gas at relieving conditions. [K]
- M is the molecular mass of the gas at relieving conditions. [kg/kmol]
- Z is the compressibility factor for the gas.

K_d is the effective coefficient of discharge, with a typical value of 0.975, for an installed PSV. K_b is a back-pressure correction factor between 0 and 1, assumed to be 1. K_c is a correction factor used when a rupture disk is installed upstream, otherwise it is 1. In the present implementation a value of 1 is assumed.

$$C = 0.03948\sqrt{k\left(\frac{2}{k+1}\right)^{\left(\frac{k+1}{k-1}\right)}} \tag{3.7}$$

For subsonic flow (subcritical flow), the effective discharge area of the PSV is determined by equation (3.8).

$$W = \frac{A \cdot F_2 \cdot K_d \cdot K_c}{17.9\sqrt{\frac{T \cdot Z}{M \cdot P_1 \cdot (P_1 - P_2)}}} \tag{3.8}$$

F_2 is the coefficient of subcritical flow which can be determined from (3.9).

$$F_2 = \sqrt{\left(\frac{k}{k-1}\right) r^{\left(\frac{2}{k}\right)} \left(\frac{1 - r^{\left(\frac{k-1}{k}\right)}}{1 - r}\right)} \tag{3.9}$$

where r is the ratio of backpressure to upstream relieving pressure, P_2/P_1.

When modelling a pop action PSV/relief valve under dynamic conditions, the valve will go from closed to fully open in a short period of time when the set pressure, P_{set}, is reached. The pop action is illustrated in

Figure 3.3 which shows the opening and closing hysteresis of the PSV as a function of pressure. In order to close the pressure shall be reduced below the reseat pressure.

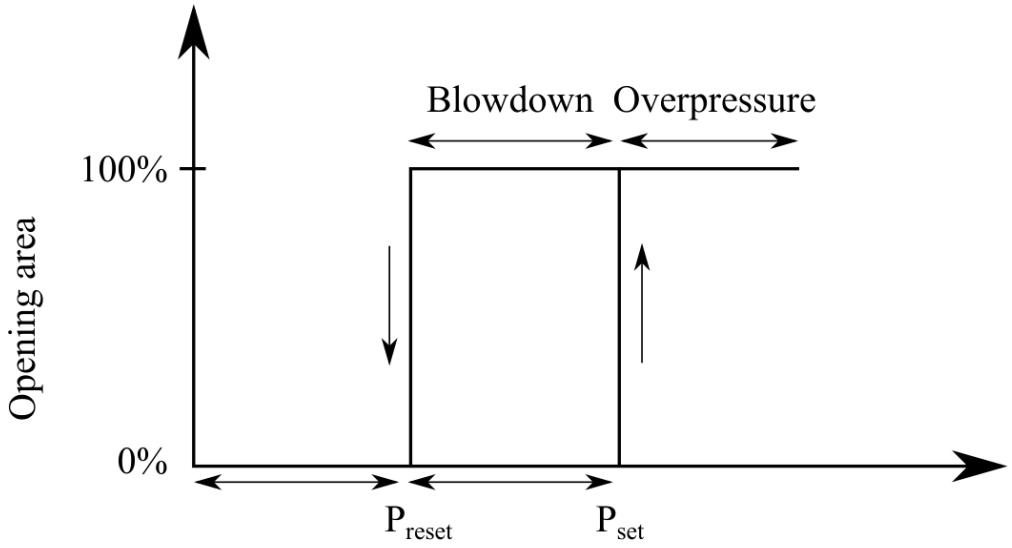


Figure 3.3: Relief valve hysteresis adapted from [EB15]

When specifying PSVs it common to use standard API sizes as shown in the table

Table 3.1: Standard PSV orifice sizes according to API

Size	Area [in^2]	Area [m^2]
D	0.110	$7.09676 \cdot 10^{-5}$
E	0.196	$1.26451 \cdot 10^{-4}$
F	0.307	$1.98064 \cdot 10^{-4}$
G	0.503	$3.24515 \cdot 10^{-4}$
H	0.785	$5.06450 \cdot 10^{-4}$
J	1.287	$8.30320 \cdot 10^{-4}$
K	1.838	$1.18580 \cdot 10^{-3}$
L	2.853	$1.84064 \cdot 10^{-3}$
M	3.600	$2.32257 \cdot 10^{-3}$
N	4.340	$2.79999 \cdot 10^{-3}$
P	6.380	$4.11612 \cdot 10^{-3}$
Q	11.050	$7.12901 \cdot 10^{-3}$
R	16.000	$1.03225 \cdot 10^{-2}$
T	26.000	$1.67741 \cdot 10^{-2}$

3.2.4 Control Valve

For calculating the mass flow through a control valve, the ANSI/ISA [Bor98, ISA95]methodology also described in IEC 60534 [IEC11] is applied.

The flow model for a compressible fluid in the turbulent regime is:

$$W = CN_6F_PY\sqrt{x_{sizing}p_1\rho_1}$$

or equivalently:

$$W = CN_8F_Pp_1Y\sqrt{\frac{x_{sizing}M}{T_1Z_1}}$$

- C is the flow coefficient (C_v or K_v)
- N_8 is a unit specific constant, 94.8 for C_v and bar as pressure unit
- F_P is a piping geometry factor [-]
- Y is the expansion factor [-]
- x_{sizing} is is the pressure drop used for sizing[-]
- p_1 is the upstream pressure [bar]
- ρ_1 is the upstream density [kg/m^3]
- M is the molecular weight [kg/kmol]
- T_1 is the upstream temperature [K]
- Z_1 is the upstream compressibility [-]

In HydDown, the piping geometry factor is not yet implemented and assumed to be 1. The pressure drop ratio x_{sizing} used for sizing is determined as the lesser of the actual pressure drop ratio, x, and the choked pressure drop ratio x_{choked}. The actual pressure drop ratio is given by:

$$x \frac{\Delta p}{p_1}$$

The pressure drop ratio at which flow no longer increases with increased value in pressure drop ratio is the choked pressure drop ratio, given by the following equation:

$$x_{choked} = F_\gamma x_{TP}$$

The factor x_T is based on air near atmospheric pressure as the flowing fluid with a specific heat ratio of 1.40. If the specific heat ratio for the flowing fluid is not 1.40, the factor F_γ is used to adjust x_T. Use the following equation to calculate the specific heat ratio factor:

$$F_\gamma = \frac{\gamma}{1.4}$$

where γ is the ideal gas C_p/C_v. It should be noted that the above equation has been derived from perfect gas behaviour and extension of an orifice model with γ in the range of 1.08 to 1.65. If used outside the assumptions flow calculations may become inaccurate.

The expansion factor Y accounts for the change in density as the fluid passes from the valve inlet to the vena contracta. It also accounts for the change in the vena contracta area as the pressure differential is varied.

$$Y = 1 - \frac{x_{sizing}}{3 x_{choked}}$$

3.3 Heat transfer

3.3.1 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 [WMM07].

To determine the heat transfer for the gas-wall interface, the following is applied cf. equation (3.10):

$$\frac{dQ}{dt} = \dot{Q} = hA(T_s - T_{gas}) \tag{3.10}$$

- 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 $\cdot$ 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 (3.11).

$$Nu = \frac{hL}{k} \tag{3.11}$$

- 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 (3.12):

$$Ra = Gr \cdot Pr \tag{3.12}$$

- 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 [(3.13):

$$Gr = \frac{\beta g \rho^2 L^3 \Delta T}{\mu^2} \tag{3.13}$$

The Prandtl number is a dimensionless number defined as the ratio of the momentum diffusivity to thermal diffusivity, as given in equation (3.14):

$$Pr = \frac{c_p \mu}{k} \tag{3.14}$$

- 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 [Gea93].

The mechanism of heat transfer on the outside of the vessel at ambient conditions is similar to the above. In HydDown the external heat transfer is not modelled currently. A heat transfer coefficient shall be provided.

3.3.2 Mixed convection

Experiments have indicated that the internal heat transfer mechanism for a vessel subject to filling can be well approximated by that of combined natural convection and forced convection as found from measured Nusselt numbers being well correlated with Rayleigh and Reynolds number [WMM07].

For mixed convection the effective Nusselt number, Nu, can be approximated by

$$Nu = (Nu_{forced}^n + Nu_{natural}^n)^{\frac{1}{n}}$$

During charging with different gases (H_2, N_2 and Argon), Woodfield *et al.* demonstrated that in order to provide a good fit to the experimentally determined Nusselt number a correlation based on both Reynolds and Rayleigh number was necessary. The following formula was found to provide a good fit (n=1):

$$Nu = Nu_{forced} + Nu_{natural} = 0.56Re_d^{0.67} + 0.104Ra_H^{0.352}$$

3.3.3 Nucleate boiling heat transfer

For heat transfer between vessel wall and liquid, this is treated as nucleate boiling and estimated via the Rohsenow correlation [Roh51] as implemented in the *ht* python library [BLVikramGovindarajanr24].

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

- 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 [JSRB04, Pio99].

3.3.4 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 conductivity 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.* [SBSK14] 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 [WMM07] the Biot number for a vessel with a wall thickness of 2 cm is 0.27. This is significantly higher than that approximated by [SBSK14]. 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.

Biot number calculation in lumped models

To help users assess the validity of the lumped capacitance assumption, HydDown now calculates the Biot number at each time step for lumped heat transfer models (0-D). When the optional parameter `thermal_conductivity_biot` is specified in the vessel properties, the Biot number is computed as:

$$Bi = \frac{h \cdot L}{k}$$

where h is the heat transfer coefficient, L is the wall thickness, and k is the thermal conductivity. The code issues warnings when $Bi > 0.1$, indicating that thermal gradients through the wall may be significant and the 1-D transient heat transfer model should be considered instead.

For two-phase systems, separate Biot numbers are calculated for the wetted (liquid-contact) and unwetted (gas-contact) regions, as these typically have very different heat transfer coefficients. The wetted region often exhibits much higher Biot numbers due to the enhanced heat transfer coefficient from boiling.

Note that `thermal_conductivity_biot` is distinct from `thermal_conductivity`: the former is used only for Biot number validation in lumped models, while the latter activates the full 1-D transient heat transfer solver.

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. HydDown incorporates the

thermesh code provided under an MIT license by Wouter Grouve [Gro]. The implemented model applies Cartesian coordinates as also applied in the h2fills program by NREL [KNP+21], i.e. the curved vessel wall is assumed a flat plate. This may to some extent be justied by the relatively large radius of a cylindrical storage container compared to the vessel wall thichness (see also justification references in [KNP+21]).

In HydDown, the 1-D transient heat conductivity problem is solved with *thermesh* assuming temperature independent vessel material density, heat capacity and thermal conductivity applying the Crank-Nicholson scheme using Neumann boundary conditions:

$$-k_z \frac{\partial T}{\partial z}\bigg|_{z=0} = q_{\text{left}}(t), \qquad -k_z \frac{\partial T}{\partial z}\bigg|_{z=L} = q_{\text{right}}(t)$$

The heat flux at the outer (left) and inner (right) is calculated/updated for each major time step for the explicit solver for mass and energy balances. For each major time step *thermesh* is used to update the outer and inner wall temperature by solving the 1-D transient heat conductivity using a *minor* time step equal to 1/10th of the major time step and 11 nodes. For each time *thermesh* is called during major time steps the mesh is initialised with the temperature profile at the end of the former major time step.

3.3.5 Fire heat loads

The heat transfer from the flame to the shell is modelled using the recommended approach from Scandpower [HS04] and API [API14b]. The heat transfer from the flame to the vessel shell is divided into radiation, convection, and reradiation as seen in equation (3.15):

$$q_f = \underbrace{\alpha_s \cdot \varepsilon_f \cdot \sigma \cdot T_f^4}_{\text{Radiation}} + \underbrace{h_f \cdot (T_f - T_s(t))}_{\text{Convection}} - \underbrace{\varepsilon_s \cdot \sigma \cdot T_s(t)^4}_{\text{Reradiation}} \tag{3.15}$$

- 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_f is the flame temperature. [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 [HS04]:

- $h_{jet\ fire}$ = 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 (3.16) for the incident heat flux in relation to the ambient conditions. The flame temperature is kept constant throughout the simulation:

$$q_{total} = \sigma \cdot T_f^4 + h_f \cdot (T_f - T_{amb}) \tag{3.16}$$

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

The heat flux used to calculate the flame temperature is given in table the table.

Table 3.2: Incident heat fluxes for various fire scenarios (Scandpower)

	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

Source: [HS04]

3.4 Vessel geometry

All vessels modelled are assumed of cylindrical shape. The following shapes are available in *openthermo* all provided by the Python *fluids* library [Bel25].

- 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 3.3: Vessel geometry details (torispherical tank heads)

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

For torispherical tank heads, the f *and* k *parameters are used in standards [Bel25].* f *is the dish-radius parameter,* k *is the knuckle-radius parameter.*

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*.

3.5 Rupture evaluation

It is assumed that when the von Mises stress, σ_e (MPa), exceeds the allowable tensile strength of the material, (ATS) (MPa), rupture will occur, i.e., when $\sigma_e >$ ATS.

The ATS is calculated as [HS04]:

$$ATS = UTSk_sk_y$$

Where

- UTS is the material Ultimate Tensike Strength
- k_s is a general safety factor for a specific material with known material data. If typical material specific data is applied a factor of 0.85 is recommended.
- k_y is an additional factor used for material with missing or uncertain data. Normally this factor is 1.0.

The von Mises stress is calculated by:

$$\sigma_e = \sqrt{3\left(\frac{pD^2}{D^2 - d^2}\right)^2 + \sigma_a^2}$$

Where

- p is the pressure (MPa)
- D is the vessel/pipe external diameter
- d is the vessel/pipe internal diameter
- σ_a is the longitudinal stress due to the external force. Assumed to be 30 MPa [HS04]

The evaluation of vessel rupture is performed as a post-calculation step following the actual depressurisation calculation. The depressurisation calculation is performed with the applicable back-ground heat load to generate the time dependent pressure profile of the vessel inventory. The background heat load is depending on the fire type as also summarised in the table below. During the depressurisation calculation the internal heat flux is calculated. In the post-calculation step an energy balance is made for the vessel material, with the external heat is generated by the applicable peak heat load Stefan-Boltzmann formulation as also provided in the table below, with the internal heat flux calculated using the back-gorund heat load. The heat balance for the vessel wall is used to solve for the vessel wall temperature as a function of time

$$\frac{dT}{dt} = \frac{q_{external} - q_{internal}}{C_p \rho dx}$$

Where:

- T is the vessel wall temperature (K)
- $q_{external}$ is time dependent peak fire heat flux (W/m^2 K)
- $q_{internal}$ is the internal convective heat flux (W/m^2 K)
- C_p is the material temperature dependent heat capacity (J/kg K)
- ρ is the materal density (assumed constant) (kg/m^3)
- dx is the vessel wall material thickness (m)

This oridinary differential equation is solved using a simple explicit Euler scheme.

The temperature dependent material properties has been sourced from the Scandpower guideline [HS04] and visualised in Figure 3.4 and Figure 3.5 for heat capacity and Ultimate Tensile Strength, respectively. The materials implemented in *hyddown* are summarised in the table below.

Table 3.4: Steel materials implemented in *openthermo*

Steel type	Type / alloy	ASME	DIN	ASTM
Carbon steel	235LT			A-333 / A-671
	360LT			
Duplex (SS)	2205	SA-770	1.4462	A-790
Austenitic (SS)	316	A-358 316	1.4401	A-320
Super austenitic (SS)	6Mo		1.4529	B-677

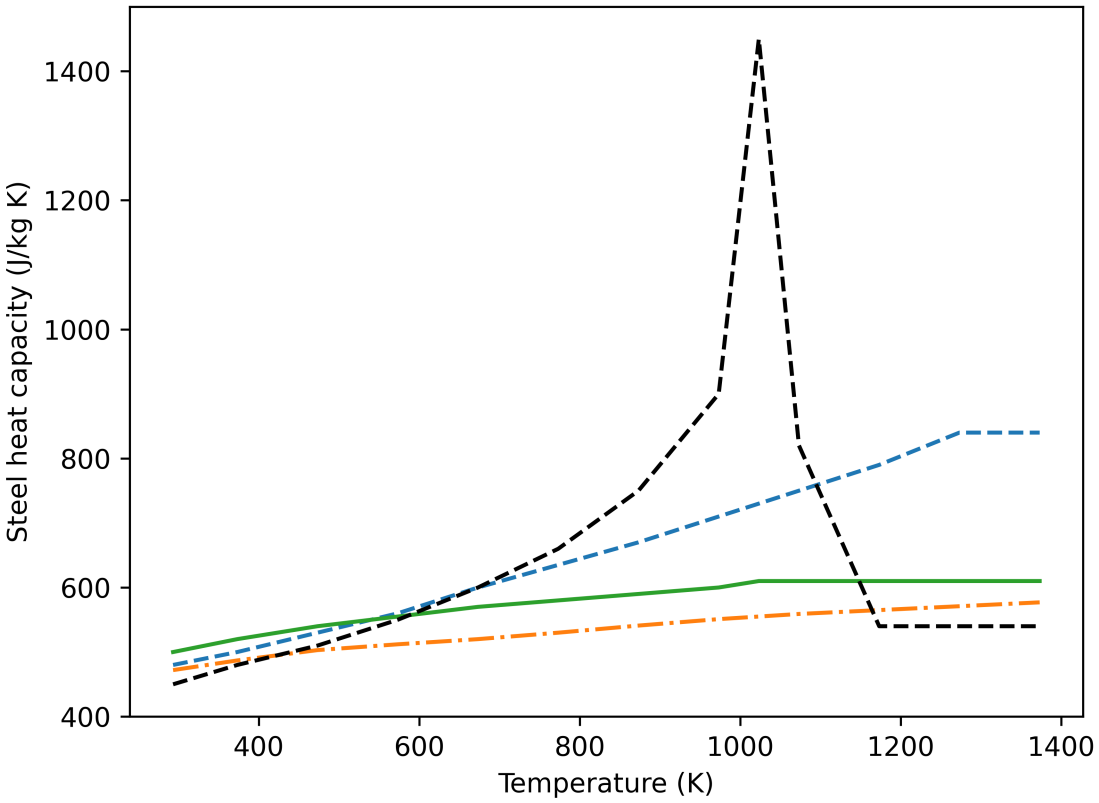


Figure 3.4: Steel heat capacities as a function of temperature for the materials implemented in *hyddown*. The values have been sourced from [HS04] and missing values extrapolated to cover the same temperature range.

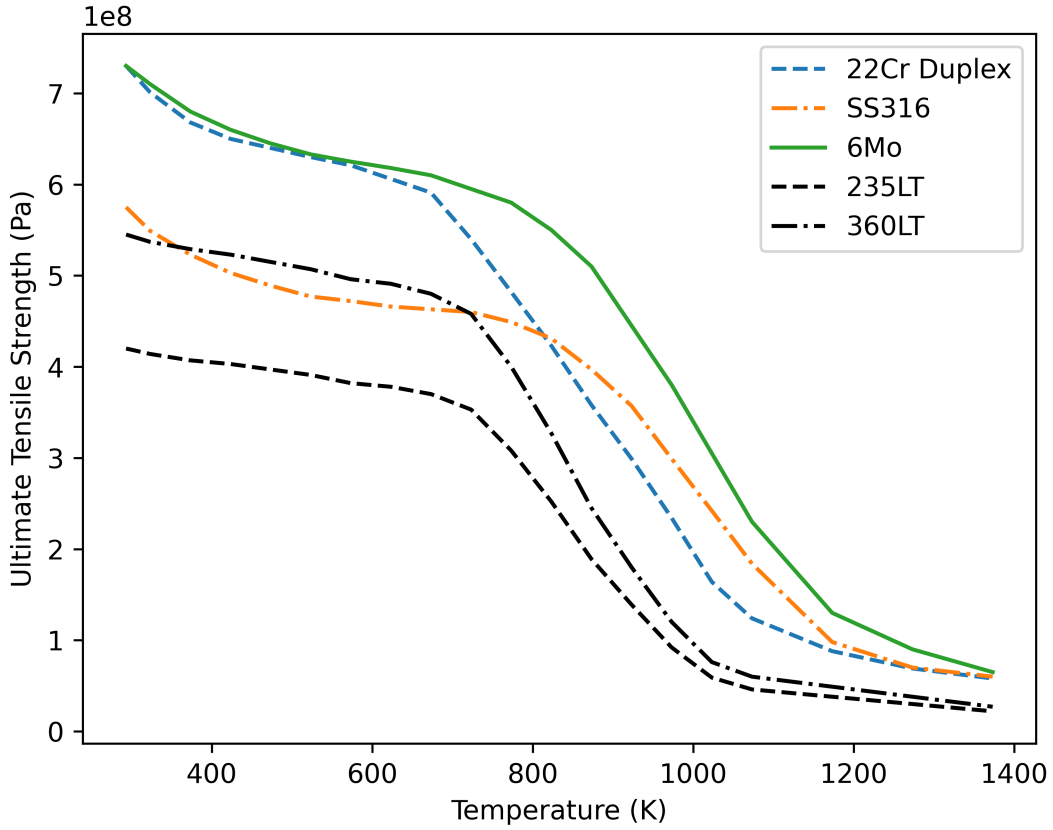


Figure 3.5: Steel Ultimate Tensile Strength as a function of temperature for the materials implemented in *hyddown*. The values have been sourced from [HS04] and missing values extrapolated to cover the same temperature range.

3.6 Model implementation

A simple (naive) explicit Euler scheme is implemented to integrate the mass balance over time, with the mass rate being calculated from an orifice/valve equation as described in *Flow devices*:

$$m_{cv}(i+1) = m_{cv}(i) + \dot{m}(i)\Delta t \tag{3.17}$$

$$\dot{m}(i) = f(P, T,)$$

For each step, the mass relief/ left in the vessel is known. Since the volume is fixed the mass density is directly given.

For the calculation methods (isentropic, isenthalpic, isenergetic, etc.), CoolProp allows specifying density and either H, S, or U directly - this is very handy and normally only TP, PH, or TS property pairs are implemented and you would need to code a second loop to make it into am UV, VH, or SV calculation.

In the following, the high-level steps in the solution procedure are outlined for the different calculation types. To illustrate calls to the equation of state in CoolProp, the notation $EOS(D, T)$ corresponds to calculation of the state at known density and temperature, for example.

3.6.1 Isothermal process

For an isothermal process, the solution procedure for each calculation step is the following with the mass in the control volume being calculated from (3.17):

$$T(i+1) = T(i)$$

$$D(i+1) = \frac{m_{cv}(i+1)}{V}$$

$$P(i+1) = EOS(D(i+1), T(i+1))$$

The notation $EOS(T, P)$ refers to a call to CoolProp either via PropsSI or the AbstractState solving the equation of state for specified temperature and pressure. Different state pairs are applied includeing combinations of density (D), specific enthalpy (H), specific entropy (S), specific internal energy (U).

3.6.2 Isenthalpic process

For an isenthalpic process:

$$H(i+1) = H(i)$$

$$D(i+1) = \frac{m_{cv}(i+1)}{V}$$

$$P(i+1) = EOS(D(i+1), H(i+1))$$

$$T(i+1) = EOS(D(i+1), H(i+1))$$

3.6.3 Isentropic process

For an isentropic process:

$$S(i+1) = S(i)$$

$$D(i+1) = \frac{m_{cv}(i+1)}{V}$$

$$P(i+1) = EOS(D(i+1), S(i+1))$$

$$T(i+1) = EOS(D(i+1), S(i+1))$$

3.6.4 Isenergetic process

For an isenergetic process:

$$U(i+1) = U(i)$$

$$D(i+1) = \frac{m_{cv}(i+1)}{V}$$

$$P(i+1) = EOS(D(i+1), U(i+1))$$

$$T(i+1) = EOS(D(i+1), U(i+1))$$

3.6.5 Energy balance

The general first law applied to a flow process as outlined in *First law for flow process* subject to an explicit Euler scheme is:

$$D(i+1) = \frac{m_{cv}(i+1)}{V}$$

$$U_{cv}(i+1) = \frac{m_{cv}(i)U_{cv}(i) - \left(\dot{m}(i)H(i) + \dot{Q}(i)\right)\Delta t}{m_{cv}(i+1)} \tag{3.18}$$

The above assumes that mass flow is positive when leaving the control volume and heat rate is positive when transferred to the control volume. $H(i)$ is the specific enthalpy of the fluid in the control volume for a discharging process and it is equal to the energy of the entering stream for a filling process. The heat rate is calculated as outlined in (see the relevant section above).

For the vessel wall a simple heat balance is also made:

$$T_{wall}(i+1) = T_{wall}(i) + \frac{\dot{Q}_{outer} - \dot{Q}_{inner}}{c_p m_{vessel}}\Delta t$$

where $\dot{Q}_{outer}$ is the convective heat transfer to or from the ambient surroundings from the outer surface of the vessel, with positive values indicating that heat is transferred from the surroundings to the vessel wall. This is either a fixed heat transfer coefficient with a specified ambient temperature (outer surface) or a calculated fire heat load. $\dot{Q}_{inner}$ is the internal heat transfer, either a fixed number or calculated as natural convection (for discharge) or mixed natural and forced convection (filling).

When the temperature profile of the vessel wall is calculated using the 1-D transient conductivity equation the vessel temperature as calculated above represent the mean vessel temperature and addition to this the outer and inner wall temperatures are calculated.

If a fixed $\dot{Q}$ is provided or a fixed overall heat transfer coefficient is provided the vessel wall temperature heat balance is not solved.

The remaining steps are update of temperature and pressure:

$$P(i+1) = EOS(D(i+1), U(i+1))$$

$$T(i+1) = EOS(D(i+1), U(i+1))$$

3.7 Multicomponent mixtures

Although not an initial requirement, the code can handle multi-component gas mixtures. When calculations are done for multicomponent mixtures, the code runs significantly slower. Please note, that in case that liquid condensate is formed during discharge calculations and even if the calculations does not stop, the results cannot be considered reliable. This is because component balances are not made and it is always assumed that the discharge composition is the same as the global composition inside the vessel. When liquid condensate is formed the composition of the vapour phase will differ from the global composition. Further, the only gas properties are estimated during multicomponent calculations, and no liquid heat transfer is considered, although a liquid phase may be present. This will make the estimation of wall temperature non-conservative if low temperature excursion is investigated.

There are a few examples of multicomponent mixtures included with HydDown. In order to specify multi-component mixtures the below example can be used as guidance:

```
"Methane[9.01290129e-01]&Ethane[6.35063506e-02]&N2[7.80078008e-03]&CO2[2.34023402e-02]&
↪Propane[3.50035004e-03]&Butane[5.00050005e-04]"
```

For component names please refer to the Coolprop manual.

An example calculation for the above mixture is shown in Figure 3.6.

The pressure and temperature trajectory is visualised along with the fluid mixture phase envelope in Figure 3.7. As seen from the figure, this case is borderline and the pressure/temperature trajectory just coincides with the dew line on the phase envelope. This plot is included in the example main script that comes with HydDown and serves as an important quality control.

For multi-component mixtures CoolProp does not provide solver for PH and UV-problems and these solvers have been included in HydDown by wrapping the CoolProp PT solver with a Nelder-Mead algorithm for solving for internal energy and density or pressure and enthalpy.

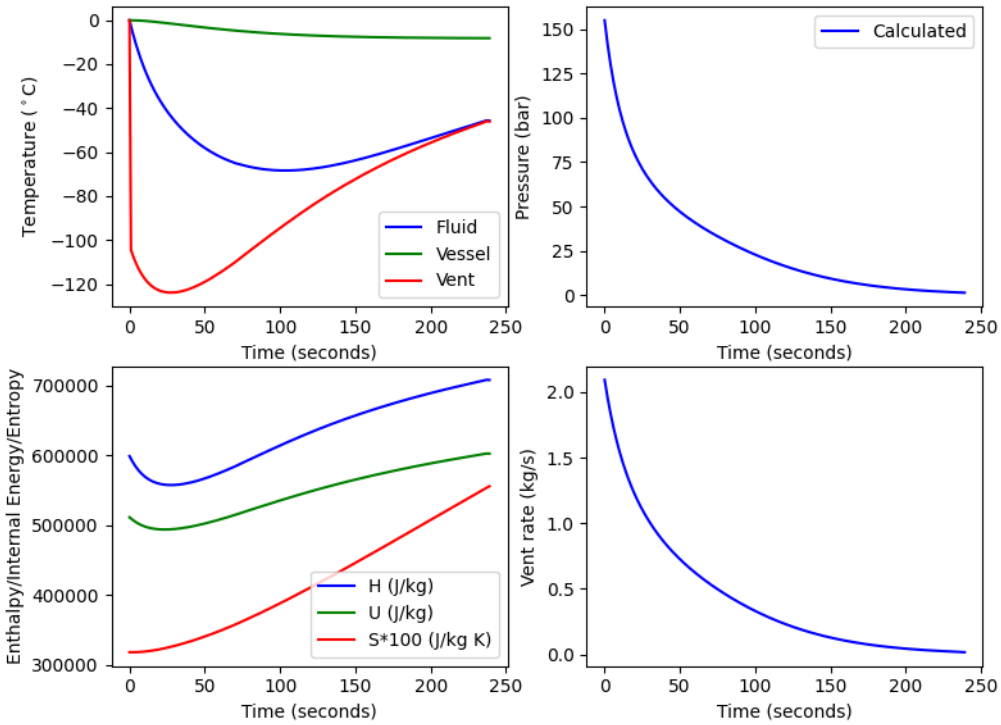


Figure 3.6: Gas discharge of multicomponent mixture

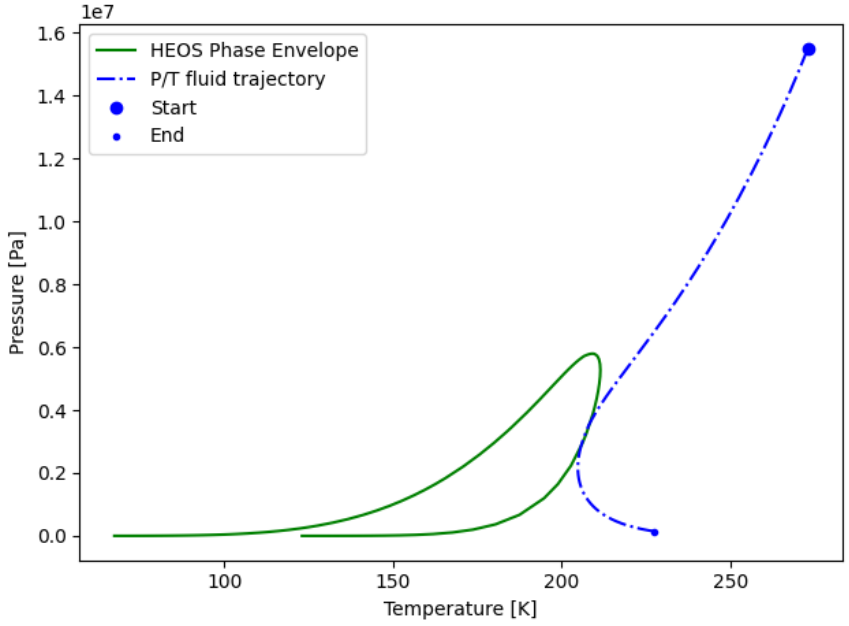


Figure 3.7: Multicomponent mixture phase envelope and pressure and temperature trajectory of the vessel inventory during discharge

CHAPTER

FOUR

VALIDATION

The code is provided as-is. However, comparisons have been made to some experiments from the literature.

The following gases and modes are considered:

- High pressure hydrogen discharge
- High pressure hydrogen filling
- High pressure nitrogen discharge
- High pressure multi-component gas discharge

These cases are all for Type I (steel) cylinders and the simple heat transfer / heat balance model has been applied, ignoring the thermal gradient in the cylinder wall. It has been checked by applying the 1D transient heat conduction model that the results are basically unaffected by this assumption. In addition to these, a few cases for Type IV cylinders have also been included in order to validate the 1D transient heat conduction model applied. This includes

- High hydrogen discharge - comparison with commercial software.
- High pressure helium discharge experiment (Karlshuhe Institute of Technology)
- High pressure hydrogen discharge (GasTeF experiment)

Details on Type III experiments are sparse but some general observations are made for the difference between type III and type IV cylinders and a single type III filling experiment is modelled.

Furthermore, HydDown has also been benchmarked against external code by Ruiz and Moscardelli [Ruiz-MaraggiM23], who compared their GeoH2 app against HydDown and found excellent agreement for both pressure, mass flow rate and gas temperature for three different cases of discharge and filling.

4.1 Hydrogen discharge (Type I)

Calculations with HydDown for high pressure hydrogen vessel discharge have been compared to three experiments from [BRR64]. Byrnes *et al.* [BRR64] have reported a number of rapid depressurization experiments using both nitrogen and hydrogen, with blowdown times being varied between 14 seconds to 33 minutes. The vessel used was a type "K" gas cylinder with a volume of 0.0499 m^3, an ID of 8.56 inches and shell length of 55 inches. The vessel is placed in a vertical position. A control valve is mounted on the neck of the cylinder with a fixed opening percentage during each experiment, set to provide the required depressurization rate. It is mentioned that the gas cylinder is placed inside a weather proof cabinet with perlite insulation in order to emulate adiabatic conditions. However the simulations run here are with a finite heat transfer coefficient on the outside of the cylinder. In their paper, detailed pressure and temperature traces are provided for three experimental runs: 7, 8, and 9, with depressurization from 2000 psia in 30 seconds, 480 seconds and 14 seconds, respectively. These experiments are simulated in Figure 4.1, Figure 4.2 and Figure 4.3.

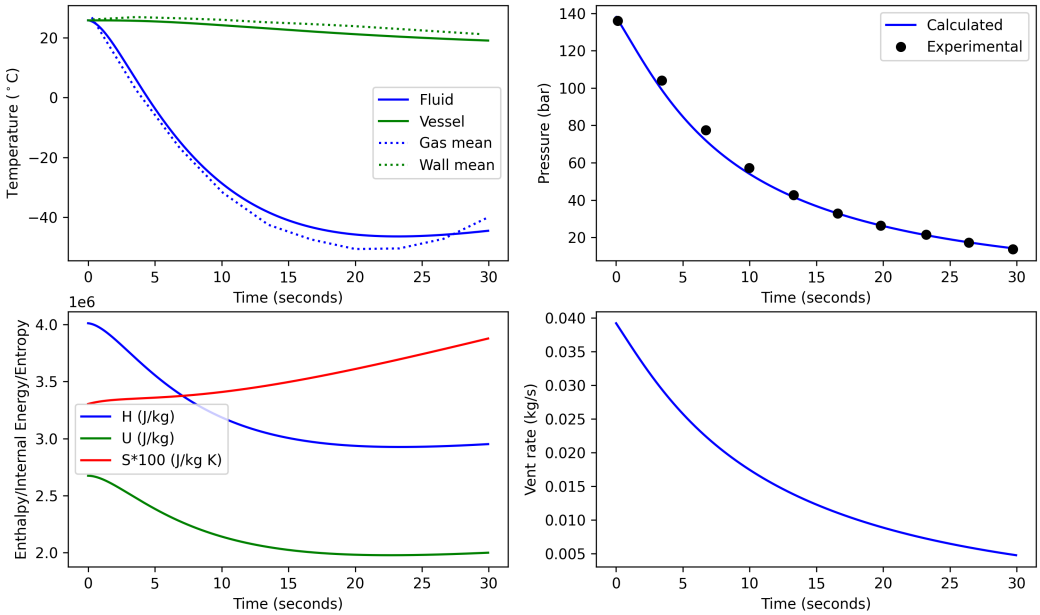


Figure 4.1: Calculations of hydrogen discharge experiment run 7 from [BRR64]. The figure shows calculated gas and wall temperature (full lines) compared to experiments (upper left), calculated and experimental pressure (upper right), specific thermodynamic state variables (lower left), and the calculated vent rate (lower right).

The figure layout is general for all the validation studies conducted and will be described in brief for eased readability. Each experiment figure is divided in 4 subplots: the upper left shows the simulated bulk gas temperature and vessel wall (average) temperature with measured values included with stipulated lines (if available), the upper right shows the simulated and measured pressure (if available, shown with points), the lower left shows the specific enthalpy, specific internal energy, and the specific entropy of the bulk gas content as a function of time, the lower right shows the simulated mass rate leaving or entering the vessel.

As seen from Figure 4.1, Figure 4.2, and Figure 4.3, the calculations generally compare well with the experimental results. It appears as though the faster depressurization times are predicted the best, as seen from run 7 and 9. The much slower run 8 does not have as good a fit for the bulk gas temperature. However, on an absolute scale the results are still within 5°C of the experimentally determined gas temperature. The trend in *goodness of fit* could indicate that the heat transfer from the outside of the vessel to the gas lacks some details, which become detectable at the very slow depressurization. For the fast depressurization the heat transfer from the outside is of less importance and the inside heat transfer coefficient may be predicted better.

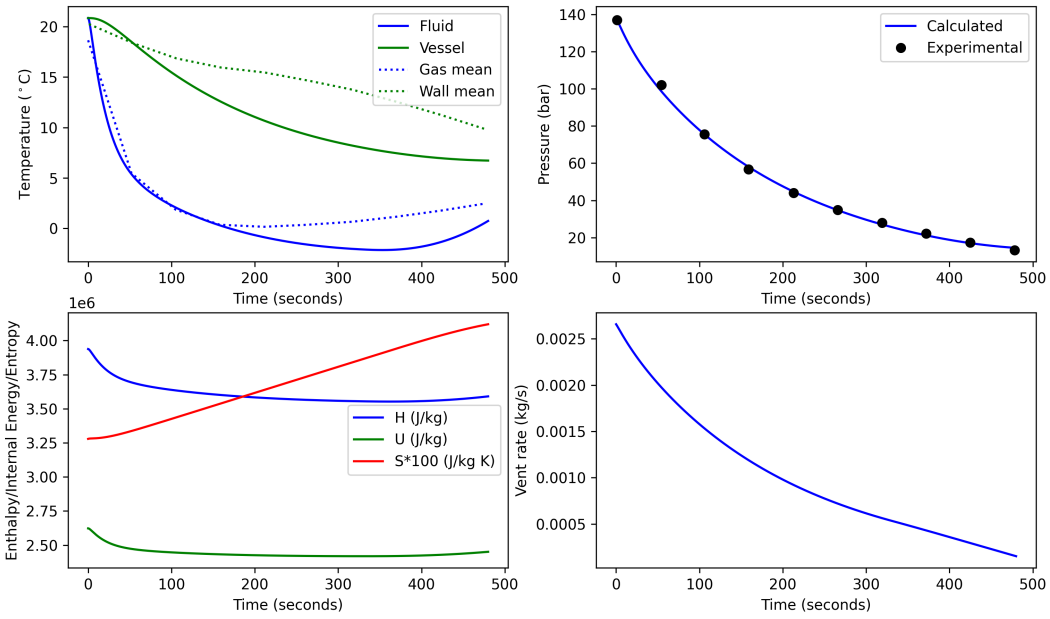


Figure 4.2: Simulation of run 8 from [BRR64].

4.2 Hydrogen filling (Type I)

In order to compare HydDown to experimental values of hydrogen filling operation the experiments reported by Striednig *et al.* [SBSK14] are used. Experimental results were obtained using a type I tank (steel) with a volume of 0.0235 m^3. The filling of hydrogen of gas into the vessel was carried out with an upstream reservoir kept at 350 bar and the flow was controlled with an electronically controlled dispenser. Vessel details are provided below:

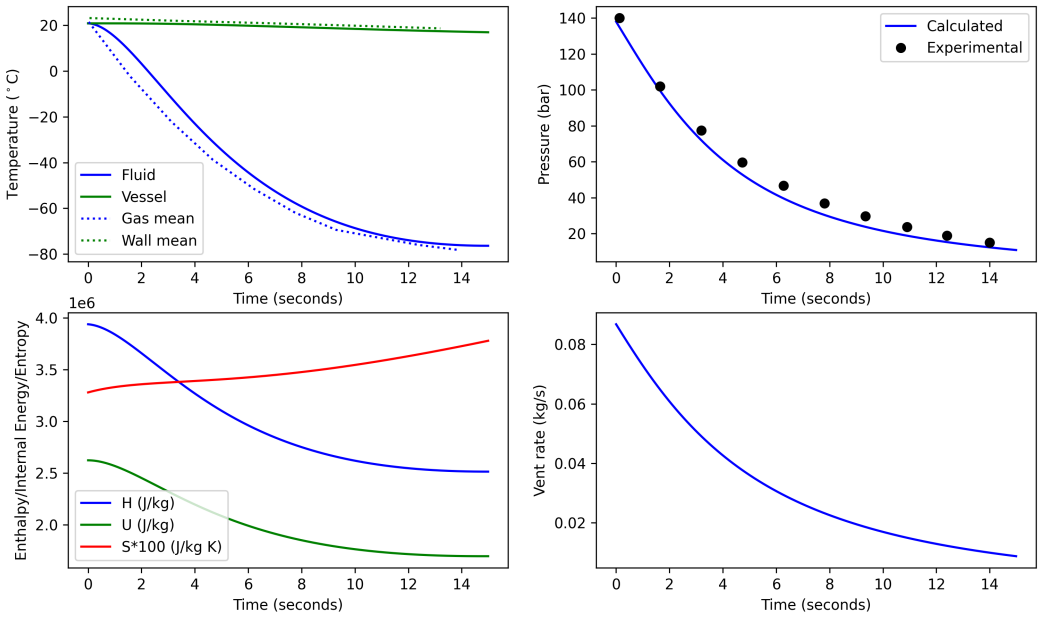


Figure 4.3: Simulation of run 9 from [BRR64].

Table 4.1: Key hydrogen vessel data

Parameter	Value
Mass	69.5 kg
Length	0.61 m
OD	0.280 m
Wall thickness	0.0129 m
Density	7740 kg/m^3
Heat capacity	470 J/(kg K)

Source: [SBSK14]

[SBSK14] reports many different experiments. In this comparison we will use experiments applying different pressurisation rates / durations using identical initial conditions. The experiments are reported in Fig. 6 and Fig. 7 in [SBSK14].

In its current stage HydDown does not directly support specifying a fixed pressurisation rate expressed as a constant rate of change in pressure / pressure ramp rate (PRR). In order to emulate the experiments, an orifice model is used with an artificially high reservoir pressure in order to ensure that the flow is critical during the entire pressurisation. This gives a constant mass rate and translates to a fairly constant pressure ramp rate, although with real gas effects being clearly observable.

The results of the simulations with HydDown and the experiments from [SBSK14] are shown in Figure 4.4, Figure 4.5, Figure 4.6. As seen from the figures the experimental gas temperature appears to be simulated very accurately. The simulation with the fastest pressurisation under-predicts the experimental temperature slightly. The main features of the pressurisation temperature profile is a rapid increase in temperature, at the beginning of the pressurisation, where the effects of filling dominate. At approximately 20-30 s, the temperature increase levels off to a plateau, where the length of the plateau depends on the pressurisation

rate. The slower the depressurization, the longer the plateau. During this temperature plateau, the heat transfer from the gas to the vessel and surroundings balance the temperature increase that would occur if performed at adiabatic (isolated) conditions. When the gas flow has ceased upon reaching the final pressure, the temperature starts to drop due to cooling of the gas by the colder vessel wall driven by convective heat transfer. The temperature stabilises at a temperature which is higher than the initial ambient temperature due to the heating of the steel vessel. The cooling of the vessel by ambient air is slower and would require a longer run time to be clearly visible.

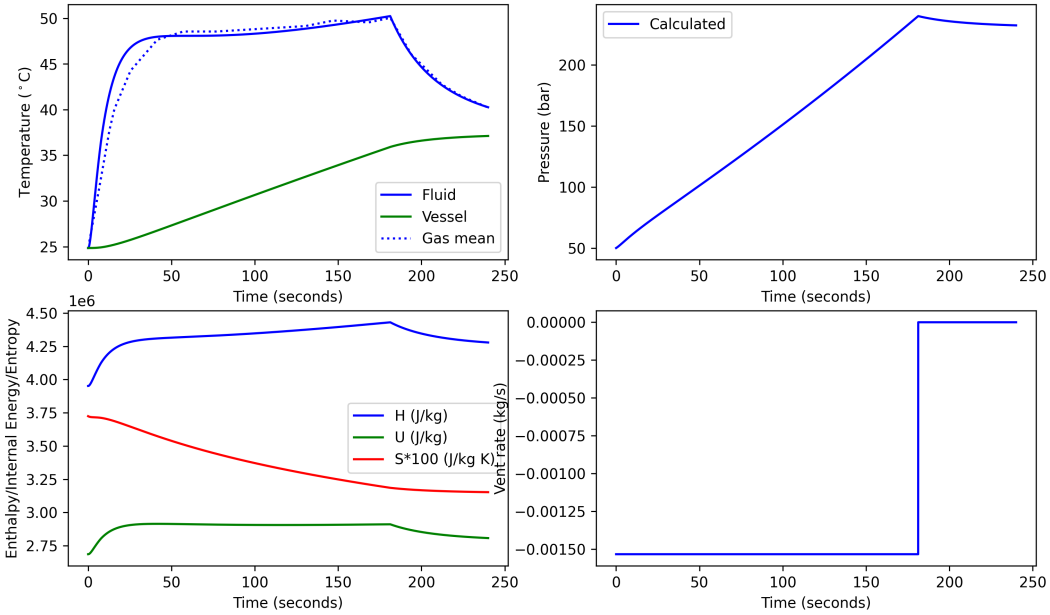


Figure 4.4: Simulation of H_2 pressurization using 5 MPa/min [SBSK14].

4.3 Nitrogen discharge (Type I)

Calculations with HydDown are compared to experiment I1 from ref. [HRS+92]. The experiment is a blowdown of a vertically oriented cylindrical vessel with flat ends. The vessel length is 1.524 m, the inside diameter is 0.273 m, and the wall thickness is 25 mm. The vessel is filled with N_2 at 150 bar and 15°C. Ambient temperature is 15°C. The blowdown orifice diameter is 6.35 mm. The results are shown in Figure 4.7. The discharge coefficient of the orifice has been set to 0.8 in order to match the vessel pressure profile. The back pressure is set to atmospheric conditions.

As seen from Figure 4.7, the calculations compare well with the experimental results. The calculated temperature of the bulk vapor is within the experimental range of measured temperature at all times during the simulation. It is also noted that the minimum temperature is reached at approximately the same time as in the experiments. The calculated vessel inner wall temperature does not decline as rapidly as the experiments, but from around a calculation time of 60 s, the temperature is within the experimentally observed inner wall temperature.

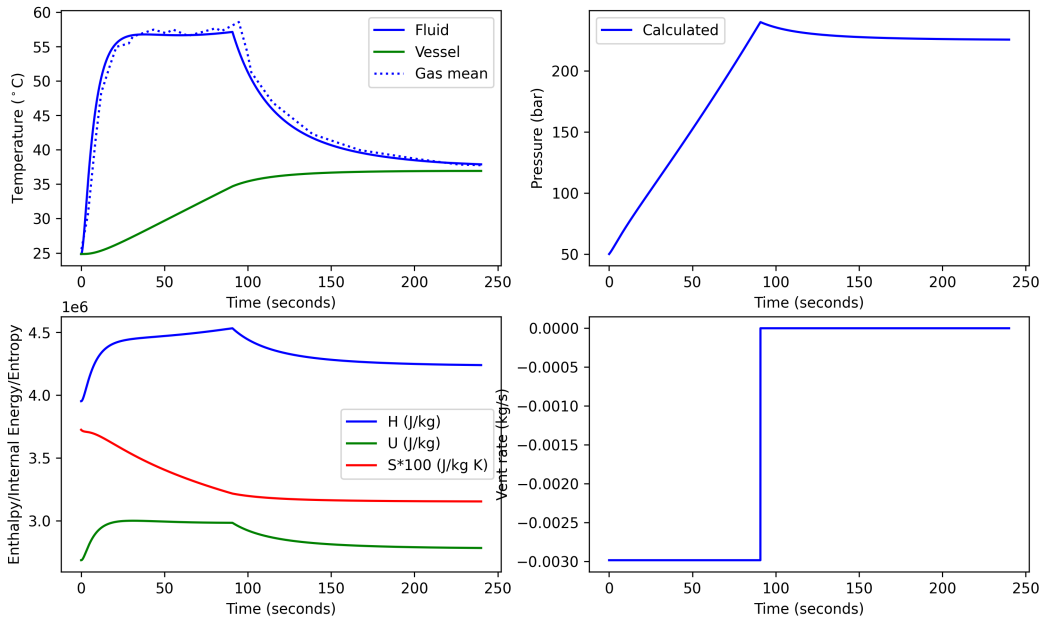


Figure 4.5: Simulation of H_2 pressurization using 10 MPa/min [SBSK14].

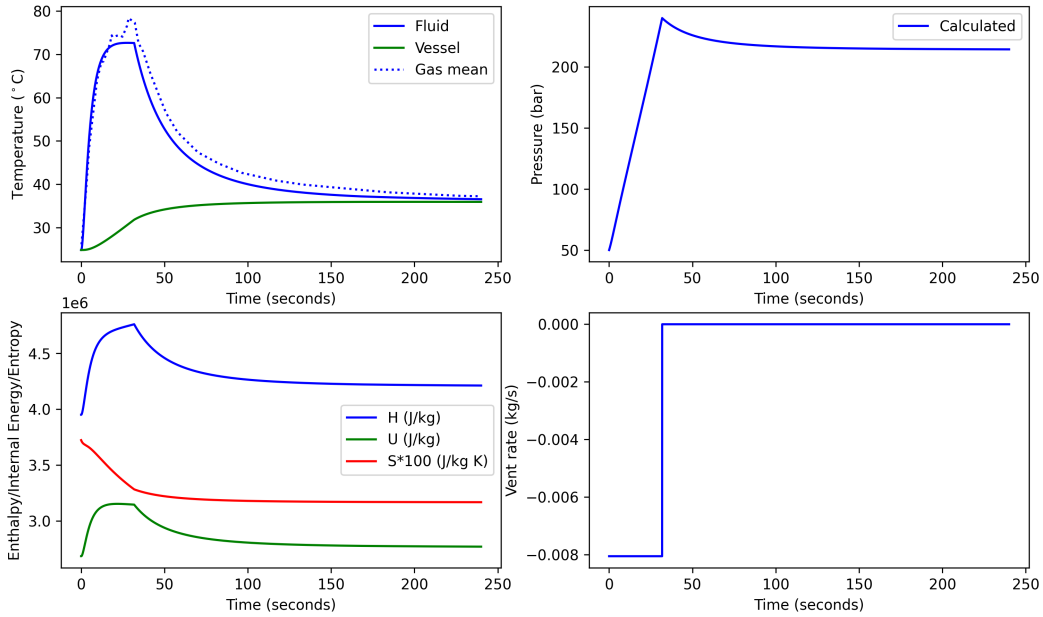


Figure 4.6: Simulation of H_2 pressurization using 30 MPa/min [SBSK14].

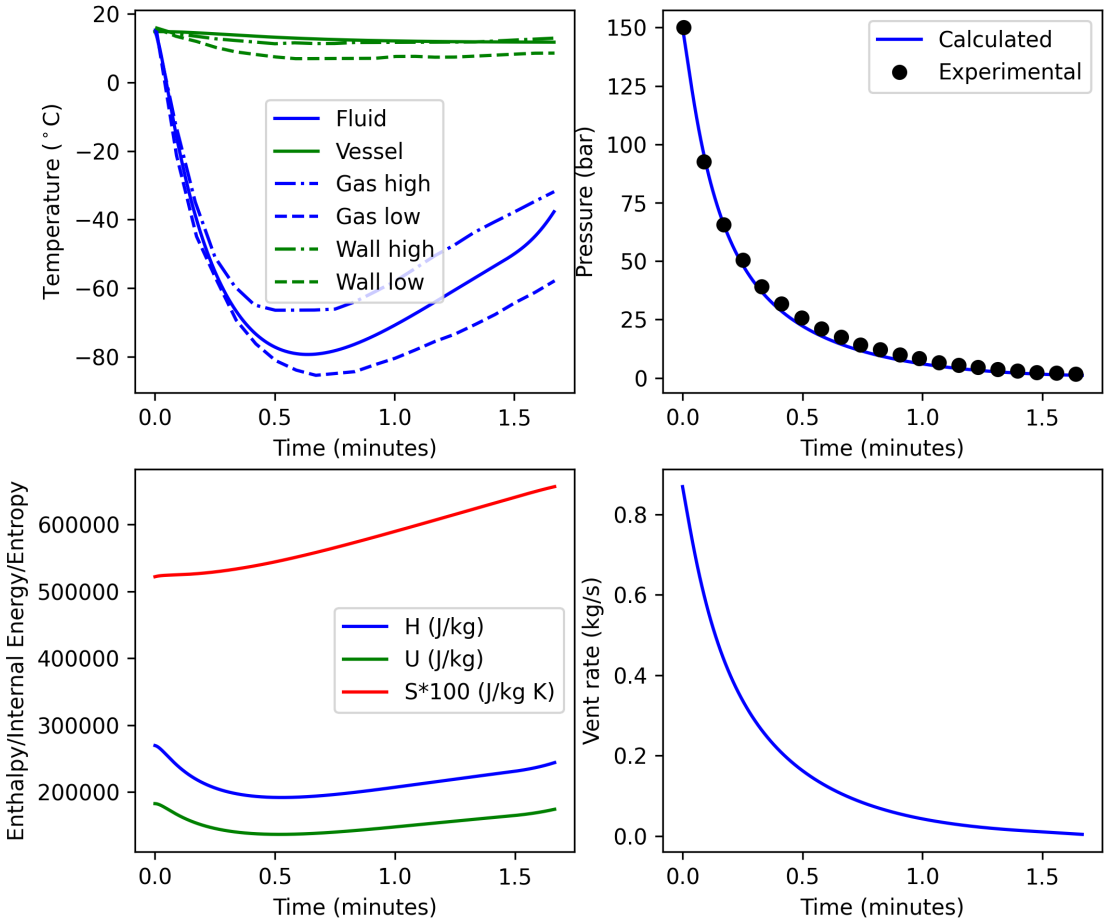
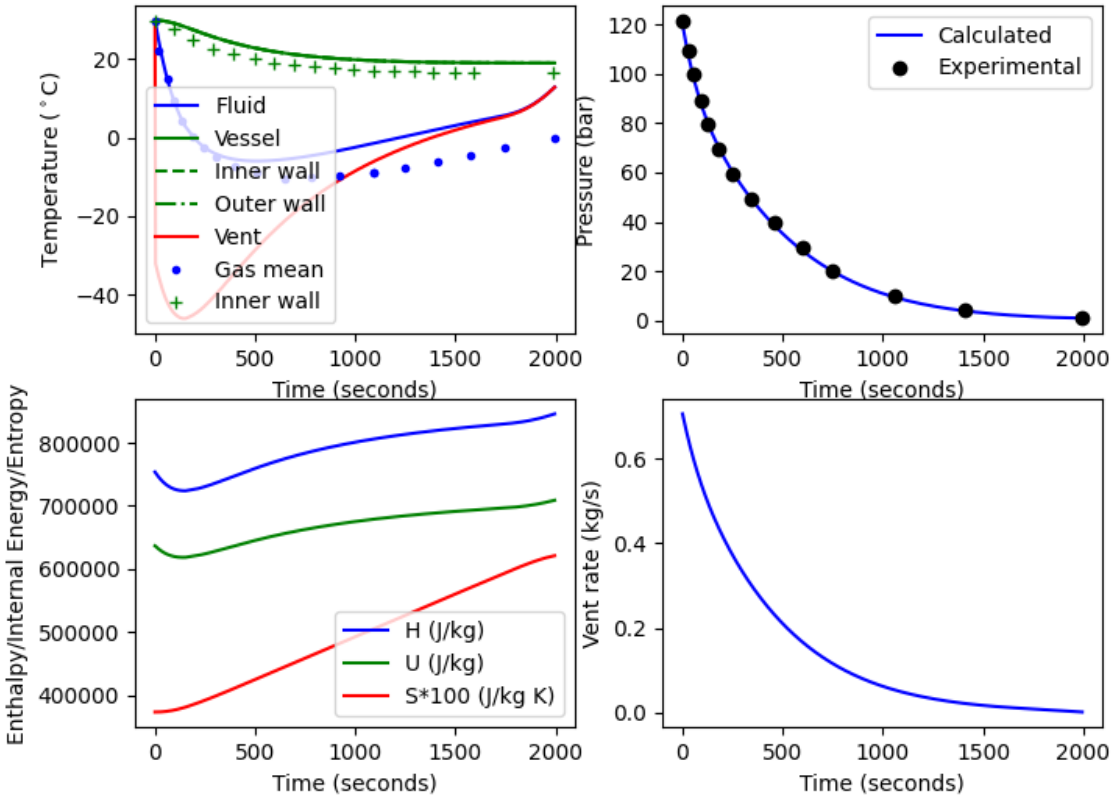


Figure 4.7: Calculations of nitrogen discharge emulating experiment I1 from [HRS+92]. The figure shows calculated gas and wall temperature (full lines) compared to experiments (upper left), calculated and experimental pressure (upper right), specific thermodynamic state variables (lower left), and the calculated vent rate (lower right).

4.4 Multi-component discharge (Type I)

The experiments were conducted by Haque *et al.* [HRS+92] and the experimental data has been extracted from the Ph.d. thesis of Wong [Won98].

The experimental vessel is a full-size suction scrubber for a gas compressor which has a total volume of 2.78 m3, with length and inside diameter of 3.240 m (2.75 m tan-to-tan) and 1.130 m respectively. The wall thickness is 59 mm. For modelling the length of a flat-ended cylinder has been adjusted to give a total volume of 2.78 m3. An orifice diameter of 6.35 mm was used for discharge and for HydDown modelling the discharge coefficient was adjusted to 0.97 to match the experimental pressure. The results of HydDown and comparison with the experimental results are shown in the figure.



The minimum gas temperature simulated by HydDown is -6°C, compared to the minimum measured average gas temperature of -10.3°C. The difference in measured vs calculated wall temperature is 2.3°C, with a lower measured wall temperature.

4.5 1-D transient heat transfer

A few notes about vessels with poor thermal conductivity and composite materials. When modelling systems with high Biot number and in particular composite materials the complexity increases significantly. Not just because of the more difficult numerical problem, but even more so because of uncertainty in key parameters such as thermal conductivity, density and heat capacity. Composite materials such as carbon fibre or glass fibre reinforced epoxy systems can be manufactured in many different ways (fibre orientation etc.) which effects the previously mentioned properties. These properties, in particular the thermal conductivity, will in-

fluence the results significantly. If these properties are not accurately informed for the system to be analysed, sourcing data from literature shall be done with caution.

4.5.1 Validation against commercial simulation tool

The first validation case is made against the commercial tool Honeywell Unisim Design in dynamics mode. The initial conditions and vessel details are summarised below cf. the table.

Table 4.2: Key vessel data and initial conditions

Parameter	Value
Length	6.4 m
ID	0.619 m
Wall thickness	0.024 m
Density	2000 kg/m^3
Heat capacity	962 J/(kg K)
Thermal conductivity	0.5 W/(m K)
Outside h	2 W/(m^2 K)
Initial pressure	182 bar
Initial temperature	6 °C
Discharge mass flow	0.02 kg/s
Gas	Hydrogen

The comparison between HydDown results and the corresponding simulations using Unisim Design Dynamics are shown in Figure Figure 4.8: As seen from the results the results obtained using HydDown closely resembles the Unisim Design results. The outer wall temperature is matched perfectly and the final gas temperature and inner vessel temperature deviates marginally by 1.5 and 1.9 °C, respectively. Unisim predicts a slightly lower gas temperature and HydDown predicts a slightly lower inner wall temperature.

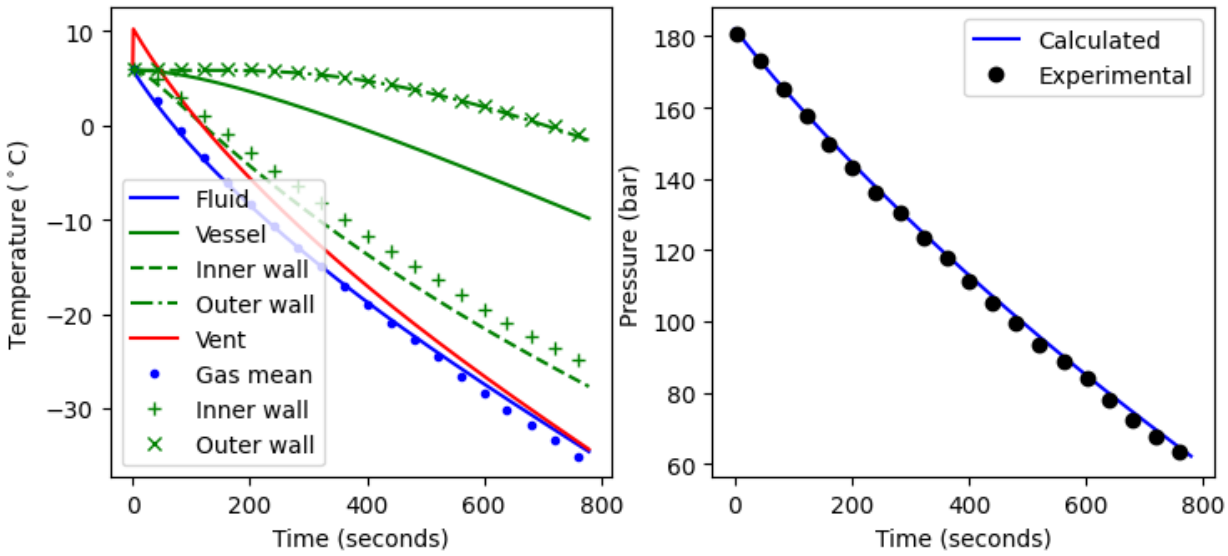


Figure 4.8: Calculations of vessel wall temperature (inner/outer) with 1D transient heat conduction during hydrogen discharge for comparison with simulation performed with Unisim Design Dynamics. The figure shows calculated gas and wall temperature (lines) compared to Unisim simulations (left) with Unisim results shown with points, calculated and Unisim simulation pressure (denoted "Experimental") (right).

4.5.2 Validation against KIT experiment (Type IV)

Using data from Molkov *et al.* [DMM17, MDKM21]experiments performed at HYKA-HyJet research facility at Karlsruhe Institute of Technology (KIT) for a 19 liter type IV pressure vessel are simulated. The storage vessel was initially charged to 700 bar with helium gas and then cooled down to a normal room temperature (293 K) before start of the depressurisation. Tank characteristics were not available and the required parameters were extracted from a similar tank by Molkov *et al.*, see references within refs. [DMM17, MDKM21]. Discharge was through 1 mm nozzle and a discharge coefficient of 0.9 was applied in HydDown as in the study by Molkov *et al.*. The initial conditions and vessel details are summarised below cf. the table.

Table 4.3: Key vessel data and initial conditions (Type IV)

Type IV tank	
External length	904 mm
ID	180 mm
OD	228 mm
HDPE liner	
Thickness	7 mm
Density	945 kg/m^3
Heat capacity	1584 J/(kg K)
Thermal conductivity	0.385 W/(m K)
CFRP shell	
Thickness	17 mm
Density	1360 kg/m^3
Heat capacity	1020 J/(kg K)
Thermal conductivity	0.5 W/(m K)
Initial conditions	
Outside h	8 W/(m^2 K)
Initial pressure	700 bar
Initial temperature	20 °C
Gas	Helium

In HydDown the type IV pressure vessel geometry as assumed that of a flat ended cylinder, with the length adjusted to give a total inventory volume of 19 liter. In order to simulate the system a lumped heat capacity and density for the type IV cylinder is made. The thermal conductivity is set to that of the liner material, and the outer heat transfer coefficient is set to 8 W/(m^{2}K) as applied in the H2fills software [KNP+21]. For a type IV vessel, the application of an average value for density, heat capacity and thermal conductivity can be an acceptable approximation since the liner and composite shell material are not too dissimilar.

HydDown simulations are compared with the KIT experiment in Figure Figure 4.9. As seen from the results the depressurisation pressure is matched very well. The experimental gas temperature is also matched fairly well. The lowest simulated gas temperature is 182.3 K and the lowest experimental temperature is 177.5 K a difference of 4.8 K. The minimum gas temperature occurs at an earlier time compared to the experimental results. The former at approx. 75 sec. and the latter at approx. 100 sec. This is also in agreement with the simulation model presented by Molkov *et al.* [DMM17]. The final gas temperature of the simulations is 237 K compared to the experimental value of 216 K.

Molkov *et al.* [DMM17] also made a thermal analysis of the thermocouple arrangement used in KIT experiment in order to estimate thermal lag in the temperature measurement. Incorporating the thermal model of the thermocouple arrangement displayed an improved prediction of the time of the minimum measured gas temperature as well as the final measured temperature.

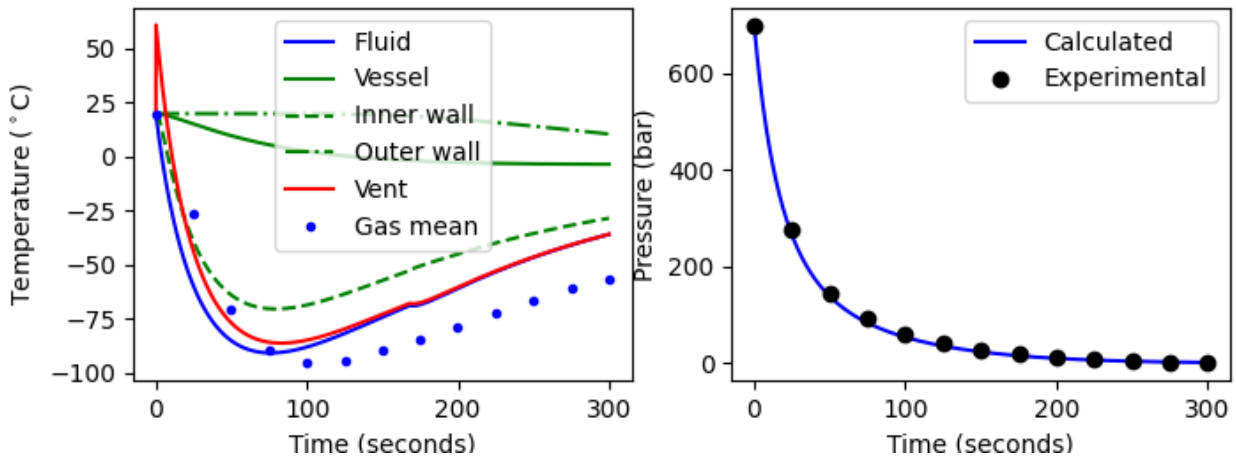


Figure 4.9: Calculations of vessel wall temperature (inner/outer) with 1D transient heat conduction during helium discharge for comparison with KIT experiments. The figure shows calculated gas and wall temperature (lines) compared to experimental gas temperature (left) and calculated and measured pressure (right).

4.6 Validation against GasTeF experiments (Type IV)

The experimental setup is described in more detail in refs. [AMdeMiguel+14, deMiguelAMC15]. Different tank types where tested both for filling and discharge including a 29 liter type IV tank and a 40 liter type III tank. In the following the type IV tank is modelled. Key details such as the initial conditions and vessel details are summarised below cf. the table. The geometry and thermal properties of the liner and composite shell material was not informed in the cited references and it is assumed to be similar to the KIT vessel described in the previous section, with the liner and composite shell thickness scaled to match the total thickness of the GasTeF type IV vessel.

Table 4.4: Key vessel data and initial conditions (GasTeF experiment)

Type IV tank	
External length	827 mm
ID	230 mm
OD	279 mm
Liner	HDPE
Composite shell	G&CRPE
Initial conditions	
Initial pressure	700 bar
Initial temperature	25±2°C
Gas	Hydrogen
Discharge rate	1.8 g/s

Note: The mass flow during discharge is constant at 1.8 g/s until the pressure drops below 5 MPa after which it drops.

The simulations with HydDown and comparison against the GasTeF experiment is shown in Figure Figure 4.10. The experimental setup included several thermocouples mounted in different positions inside the test vessel, both at the center line and nearer the top and bottom. The experimental setup did not include a direct measurement of the internal liner temperature interface towards the gas nor the composite shell. The experimental points for the gas temperature as shown in the figure includes the lowest measured temperatures

(near the bottom), the gas temperature measured in the middle and towards the top (highest temperatures). The experiments display significant temperature stratification during discharge experiments.

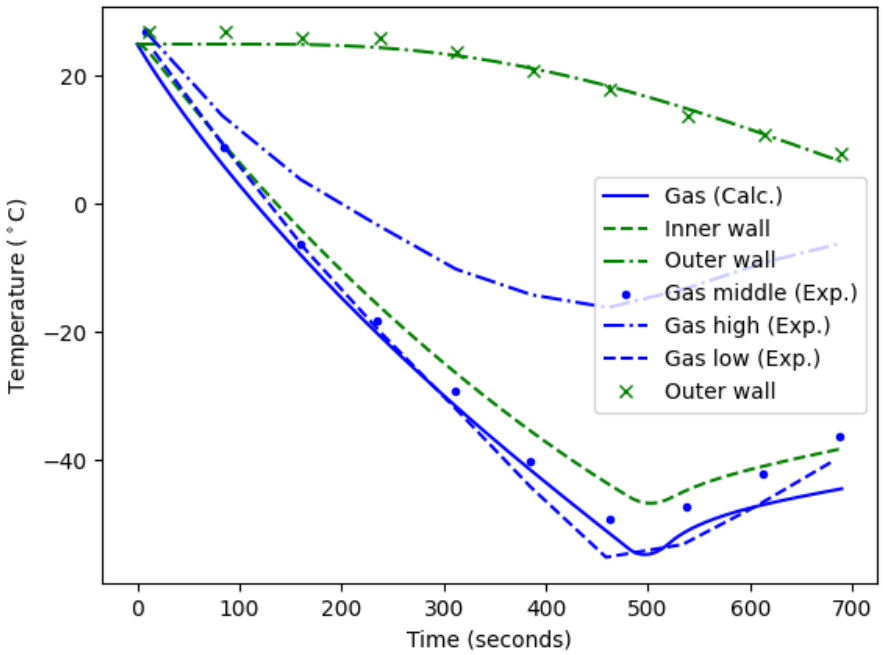


Figure 4.10: Calculations of vessel wall temperature (inner/outer) with 1D transient heat conduction during hydrogen discharge for comparison with GasTeF experiments. The figure shows calculated gas and wall temperature (lines) compared to experimental gas temperature.

As seen the prediction of the external surface temperature of the composite shell is matched very well. The average gas temperature as calculated with HydDown matches the temperatures recorded in the lower half of the test vessel. Further, the minimum in gas temperature between 450 and 500 seconds are also matched well. The measured pressure was not reported in ref. [deMiguelAMC15]

4.7 Validation against Type III cylinder discharge experiments

As shown previously type IV cylinder thermal behaviour is matched fairly well when the liner/composite thermal material properties are lumped. For a Type III vessel which has very dissimilar thermal properties of the liner and the composite shell, this approach is not advisable. Luckily, the implemented 1D transient heat conduction code *thermesh* allows modelling bi-materials and this is also implemented in HydDown.

Unfortunately, it has not been possible to find reported discharge experiments which include detailed information about the exact liner and shell composite thickness. In lack of a good and complete validation cases, an example is made using the KIT cylinder in the previous section and replacing the HPDE liner with an aluminum liner.

Table 4.5: Key vessel data and initial conditions for Type III cylinder simulations

Type III tank	
External length	904 mm
ID	180 mm
OD	228 mm
Aluminum liner	
Thickness	7 mm
Density	2700 kg/m^3
Heat capacity	900 J/(kg K)
Thermal conductivity	237 W/(m K)
CFRP shell	
Thickness	17 mm
Density	1360 kg/m^3
Heat capacity	1020 J/(kg K)
Thermal conductivity	0.5 W/(m K)
Initial conditions	
Outside h	8 W/(m^2 K)
Initial pressure	700 bar
Initial temperature	20 °C
Gas	Helium

Note: The mass flow during discharge is constant at 1.8 g/s until the pressure drops below 5 MPa after which it drops.

The results of HydDown simulation with the *artificial* type III cylinder is shown in Figure Figure 4.11. As seen the immediate effect of replacing the HDPE liner with aluminum (compared to Figure Figure 4.9) is a lower temperature drop in the gas phase and a liner temperature which approached the gas temperature very closely. This is in general agreement with the observations in ref. [deMiguelAMC15].

In Figures the figure below and the figure below the simulated temperature across the vessel liner/composite wall is shown for the real type IV cylinder from the KIT experiment and for the *artificial* type III cylinder, respectively. As seen there is a significant difference between the two bi-materials. The type III cylinder has almost zero temperature gradient across the aluminum liner, which can be rationalised by the very high thermal conductivity. Thus, the entire thermal gradient is over the composite shell material. These observations are in-line with simulation results from ref. [MBAI+17].

Calculations of type IV vessel wall temperature profile with 1D transient heat conduction. z=0 is the bonding interfance between liner and composite, z<0 is the liner and z>0 is the composite shell

Calculations of type III vessel wall temperature profile with 1D transient heat conduction. z=0 is the bonding interfance between liner and composite, z<0 is the liner and z>0 is the composite shell

4.8 Validation against Type III cylinder filling experiments

For HydDown type III cylinder filling experimental validation, the work of Dicken and Mérida [DM07] is used. They conducted experiments of filling a 74 liter type III cylinder and also compared measurements with CFD calculations. Details of type III cylinder tested is shown in Table the table. For HydDown simulation the length of a flat-ended cylinder has been adjusted to give a total cylinder volume of 74 liter. The time-dependent filling mass flow has been sourced from ref. [DM07] and scaled in order to match the final experimental pressure.

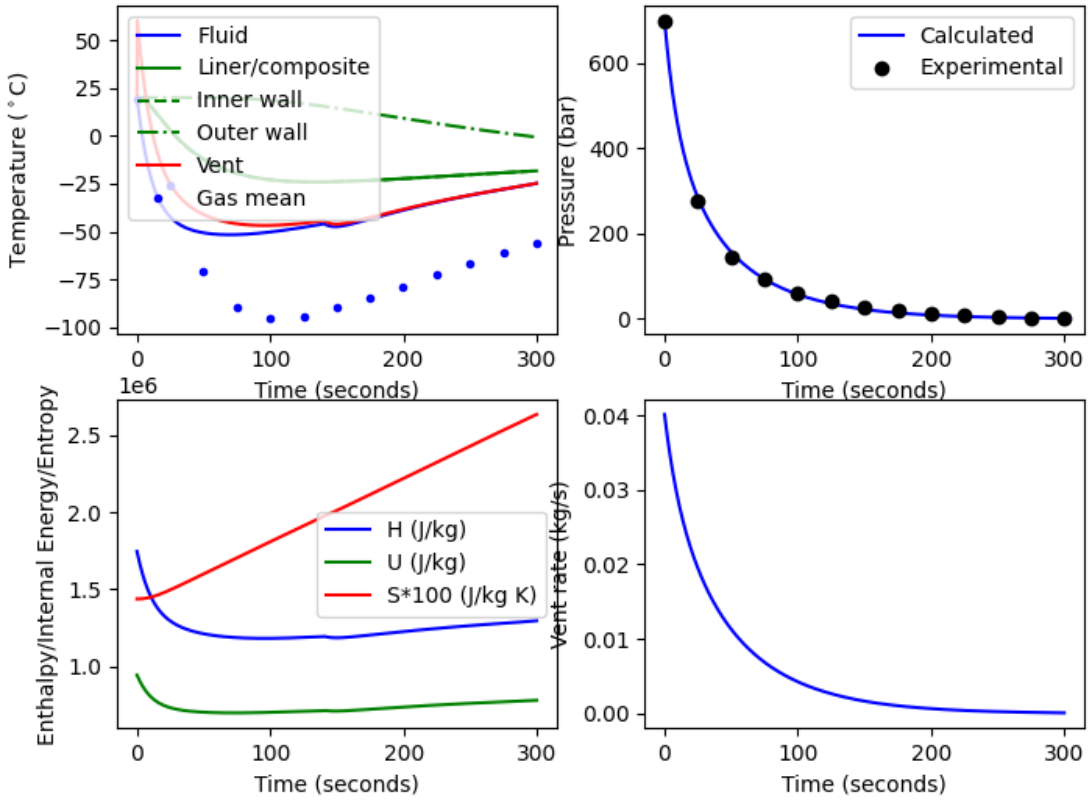


Figure 4.11: Calculations of vessel wall temperature (inner/outer) with 1D transient heat conduction during helium discharge for a hypothetical type III cylinder with dimensions similar to the KIT type IV cylinder.

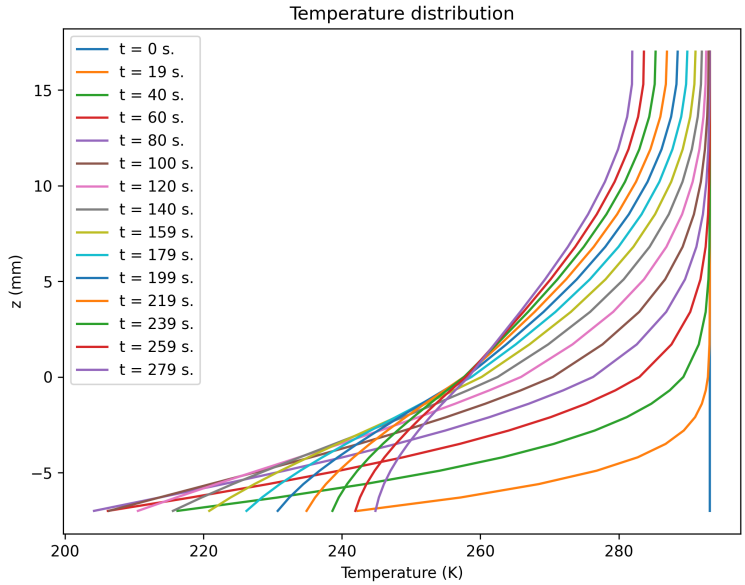


Figure 4.12: Temperature through wall at different times

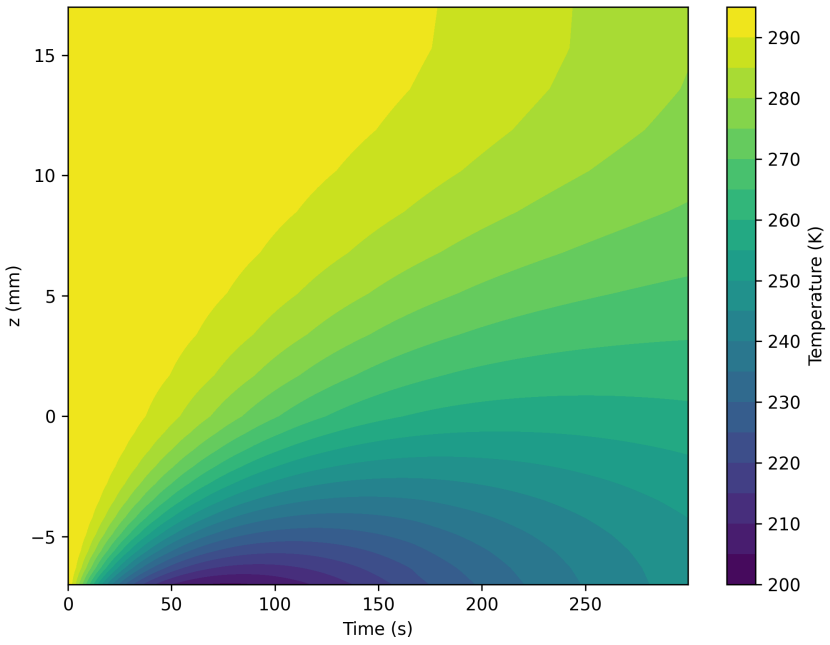


Figure 4.13: Spatial and temporal temperature contour

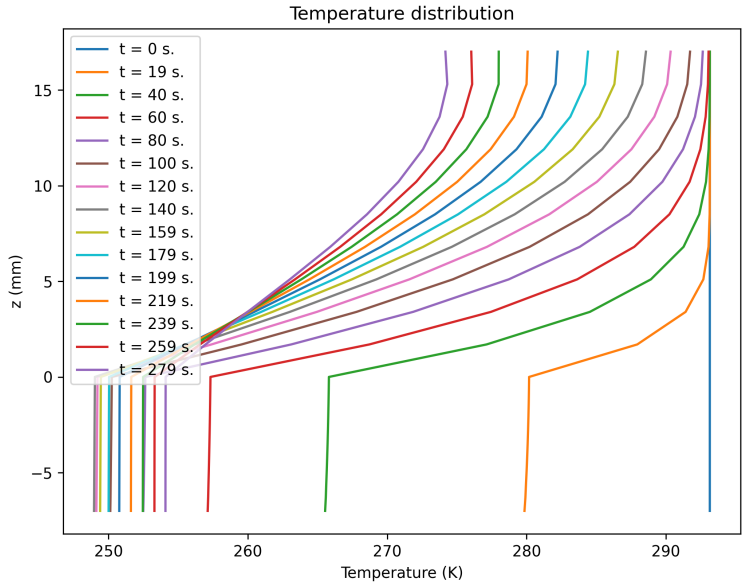


Figure 4.14: Temperature through wall at different times

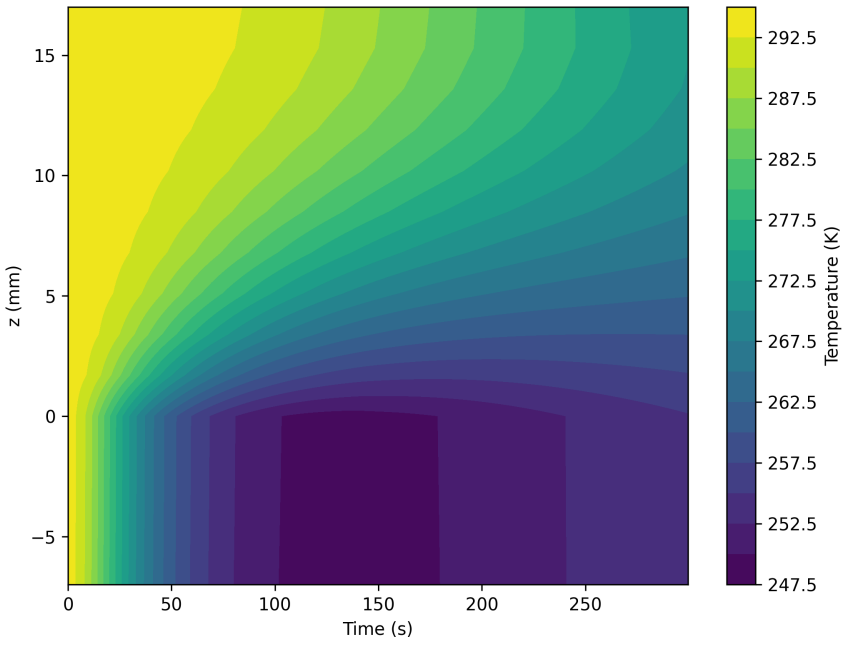


Figure 4.15: Spatial and temporal temperature contour

Table 4.6: Key vessel data and initial conditions for Type III cylinder simulations

Type III tank	
External length	893 mm
ID	358 mm
OD	396 mm
Aluminum liner	
Thickness	4 mm
Density	2700 kg/m^3
Heat capacity	900 J/(kg K)
Thermal conductivity	167 W/(m K)
CFRP shell	
Thickness	15 mm
Density	938 kg/m^3
Heat capacity	1494 J/(kg K)
Thermal conductivity	1.0 W/(m K)
Initial conditions	
Outside h	8 W/(m^2 K)
Initial pressure	93 bar
Initial temperature	20.25 °C
Gas	H2

Simulation results from HydDown is compared to the experimental data of Dicken and Mérida in Figure Figure 4.16. As seen from the results the gas temperature calculated with HydDown is generally higher than the experimental results. The final calculated temperature is 74.2°C, compared to the final measured temperature of 68.9°C. The CFD calculations performed by Dicken and Mérida [DM07] was also higher than the experimental value (71.3°C). Another CFD analysis of the investigated system by Hall and Ramasamy [HR24] revealed a final mean gas temperature of 71.4°C, and a zero-dimensional model by the same authors gave an end temperature of 72.4°C.

In lack of measured vessel inner wall temperature (liner) the values calculated at the end of filling by Dicken and Mérida using their CFD model is used for comparison. The CFD simulations revealed very large variations in the heat transfer coefficient and resulting wall temperatures. For comparison with HydDown the average CFD calculated temperature is used. This is taken as the arithmetric average of all seven positions from front to back of the cylinder. As seen from Figure Figure 4.16 the average value from HydDown (52.5°C)compares very well with the average CFD results at the end of filling (52.1°C). However, estimated inner wall temperature spanned temperatures from 26 to 67 °C, with the highest temperatures recorded opposite to the entry of hydrogen.

4.9 Validation of fire heat load

A simulation with a steel vessel subject to fire heat load is compared against calculations performed with Honeywell Unisim Design using the built-in dynamic depressuring utility and applying a Stefan-Boltzmann heat load. In Unisim the convective part of the Stefan-Boltzmann fire heat load is set by manually specifying the external heat transfer coefficient to 100 W/m^2 K and setting the ambient temperature equal to the flame temperature. The simulation in put is summarised in the table.

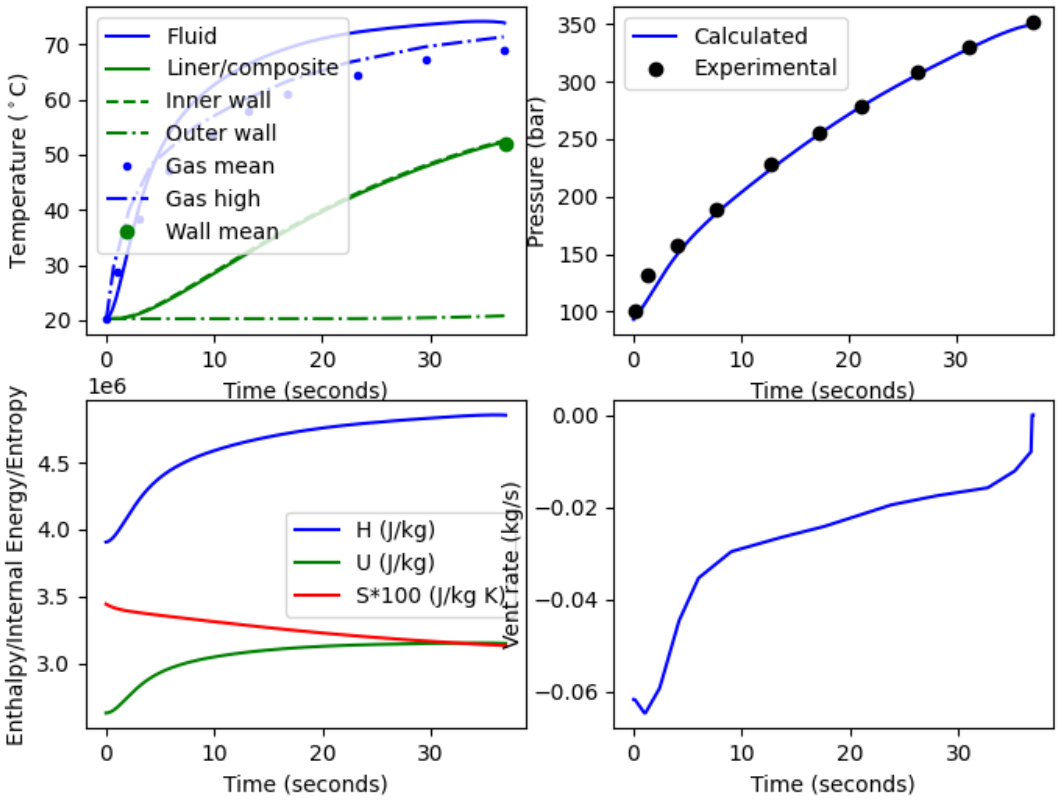


Figure 4.16: Calculations of vessel pressure, gas temperature and wall temperature (inner/outer) with 1D transient heat conduction model during hydrogen filling. Comparison is made against experimental results from Dicken and Mérida [DM07] for pressure and gas temperature. Inner wall (liner) temperature is compared to CFD simulations [DM07]

Table 4.7: Data for validation of fire heat load calculations

Parameter	
External length	9 m
ID	3 m
Shell	
Thickness	136 mm
Density	7700 kg/m^3
Heat capacity	500 J/(kg K)
Initial conditions	
Initial pressure	115 bar
Initial temperature	25 °C
Gas	Methane
Heat load	
Fire type	Jet fire
Incident heat flux	100 kW/m^2

The results are displayed in Figure 4.17 and Figure 4.18. As seen the agreement between the two simulation codes is indeed adequate. The main difference is the intitial temperature of the steel vessel wall which is higher in Unisim. In Hyddown the vessel wall temperature is initialised at the fluid temperature, whereas in Unisim it is likely based on a steady-heat balance using the applied ambient temperature (set to equal the flame temperature for calculation purpose).

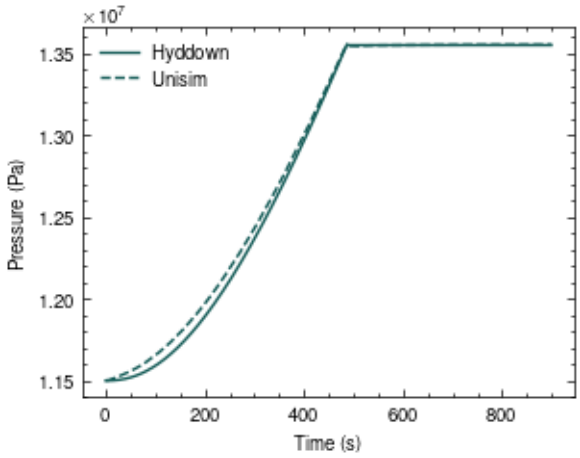


Figure 4.17: Calculation of vessel pressure as a function of time for a steel vessel subject to a jet fire backgorund heat load with an incident heat flux of 100 W/m^2.

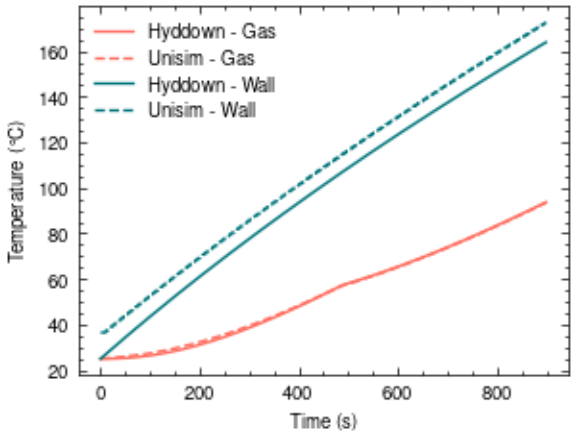


Figure 4.18: Calculation of vessl temperature as a function of time for a steel vessel subject to a jet fire back-gorund heat load with an incident heat flux of 100 W/m^2.

CHAPTER

FIVE

EXAMPLE USE CASES

5.1 LPG vessel subject to fire heat load

A vessel containing LPG (propane liquified by pressure) is subject to a back-gorund pool fire heat load. The input data are shown below and the results are shown in Figure 5.1. Vessel data and initial conditions have been sourced from Moodie et al. [MCD+88], where a 5 tonne horizontal cylindrical Liquefied Petroleum Gas (LPG) tank was exposed to an engulfing kerosene pool fire.

```
vessel:
  length: 4.64
  diameter: 1.7
  orientation: "horizontal"
  heat_capacity: 500
  density: 7700
  thickness: 0.01185
  liquid_level: 0.4668
initial:
  temperature: 279
  pressure: 550000
  fluid: "propane"
calculation:
  type: "energybalance"
  time_step: 1
  end_time: 660.
valve:
  flow: "discharge"
  type: "psv"
  diameter: 0.040
  discharge_coef: 0.975
  set_pressure: 1430000
  blowdown: 0.20
  back_pressure: 101300.
heat_transfer:
  type: "s-b"
  fire: "scandpower_pool"
```

5.1.1 LPG vessel with 1-D heat transfer

The LPG fire scenario can also be simulated using the 1-D transient heat conduction model by adding the `thermal_conductivity` parameter to the vessel properties. This enables separate tracking of inner and outer wall temperatures for both wetted and unwetted regions:

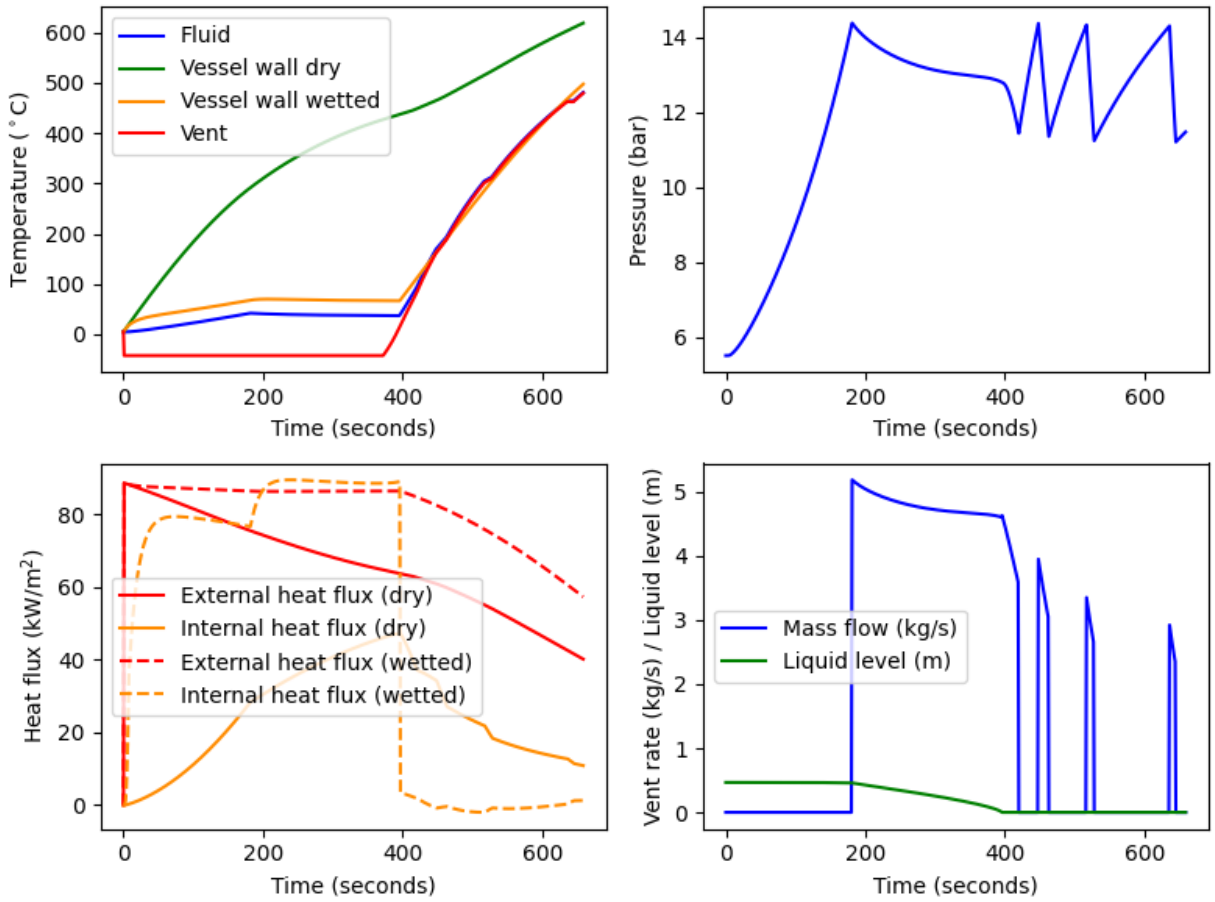


Figure 5.1: LPG vessel subject to pool fire back-gorund heat load with an incident heat flux of 100 W/m^2.

```
vessel:
  length: 4.64
  diameter: 1.7
  orientation: "horizontal"
  heat_capacity: 500
  density: 7700
  thickness: 0.01185
  thermal_conductivity: 45  # Steel thermal conductivity W/m-K
  liquid_level: 0.4668
```

With the 1-D model, the code solves separate heat conduction equations for the wetted (liquid-contact) and unwetted (gas-contact) wall regions. This is particularly important for two-phase fire scenarios where:

- The wetted region experiences higher internal heat transfer coefficients due to boiling
- The unwetted region has lower heat transfer coefficients (gas convection only)
- Wall temperature gradients through the thickness become significant with thin walls and high heat fluxes

For this steel vessel case with 11.85 mm wall thickness, the lumped capacitance model ($Bi \approx 0.03$ for unwetted, $Bi \approx 0.79$ for wetted region) gives results very close to the 1-D model. However, for thicker walls, lower thermal conductivity materials (Type III/IV vessels), or when accurate wall temperature profiles are needed, the 1-D model should be used.

5.2 Rupture estimation

The below example is a calculation where a methane filled cylindrical flat-end vessel is subject to a background heat load from a jet fire. The background heat load is given by the general Stefan-Boltzmann fire equation and heat load according to the Scandpower guideline [HS04]. In addition to this for the post calculation of rupture according to the von Mises stress criterion as outlined in (see the relevant section). The discharge model is specified as **relief**, in this case the pressure is allowed to increase, until the relief valve set pressure is reached. From this point the pressure is kept constant and the required relief rate in order to keep the pressure constant is calculated. This method only works in case a heat load is applied.

```
vessel:
  length: 9
  diameter: 3
  orientation: "vertical"
  type: "Flat-end"
  heat_capacity: 500
  density: 7700
  thickness: 0.136
initial:
  temperature: 298.15
  pressure: 11500000
  fluid: "CH4"
calculation:
  type: "energybalance"
  time_step: 1
  end_time: 900.
valve:
  flow: "discharge"
  type: "relief"
```

(continues on next page)

(continued from previous page)

```
  set_pressure: 13550000
  back_pressure: 101300.
heat_transfer:
  type: "s-b"
  fire: "scandpower_pool"
rupture:
  material: "CS_360LT"
  fire: "scandpower_jet_peak_large"
```

The rupture evaluation is run as a post-processing step by running the class method **analyze_rupture()**.

```
import yaml
import sys
from hyddown import HydDown

if __name__ == "__main__":
    if len(sys.argv) > 1:
        input_filename = sys.argv[1]
    else:
        input_filename = "input.yml"

    with open(input_filename) as infile:
        input = yaml.load(infile, Loader=yaml.FullLoader)


    hdown=HydDown(input)
    hdown.run()
    hdown.verbose=1
    hdown.plot()
    hddown.analyze_rupture()
```

5.3 Composite cylinder subject to blowdown

```
vessel:
  length: 0.7466
  diameter: 0.18
  thickness: 0.017
  heat_capacity: 1020
  density: 1360.
  thermal_conductivity: 0.5
  liner_thickness: 0.007
  liner_heat_capacity: 1584
  liner_density: 945.
  liner_thermal_conductivity: 0.385
  orientation: "horizontal"
initial:
  temperature: 293.
  pressure: 70000000.
  fluid: "He"
calculation:
  type: "energybalance"
```

(continues on next page)

(continued from previous page)

```
  time_step: .2
  end_time: 300.
valve:
  flow: "discharge"
  type: "orifice"
  diameter : 0.001
  discharge_coef: 0.9
  back_pressure: 101300.
heat_transfer:
  type: "specified_h"
  temp_ambient: 293.15
  h_outer: 8.
  h_inner: "calc"
```

CHAPTER

SIX

INSTALLATION

6.1 Requirements

HydDown requires Python 3.10, 3.11, or 3.12.

6.2 Installation via pip

The simplest way to install HydDown is via pip:

```
pip install hyddown
```

6.3 Installation from Source

To install HydDown from source for development:

1. Clone the repository:

```
git clone https://github.com/andr1976/HydDown.git
cd HydDown
```

2. Install in development mode:

```
pip install -e .
```

3. Install dependencies:

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

6.4 Dependencies

HydDown depends on the following packages:

- numpy >= 1.19.5
- scipy >= 1.6.0
- matplotlib >= 3.3.3
- pandas >= 1.1.4
- CoolProp (thermodynamic backend)
- PyYAML >= 5.4.1

- Cerberus >= 1.3.3 (input validation)
- fluids (vessel geometry calculations)
- ht (heat transfer correlations)
- tqdm (progress bars)

6.5 Optional Dependencies

For running the Streamlit web application:

```
pip install streamlit
```

For testing:

```
pip install pytest pytest-cov
```

CHAPTER

SEVEN

QUICK START

7.1 Basic Usage

Running HydDown is as simple as:

```
python scripts/hyddown_main.py input.yml
```

where `input.yml` is your input file in YAML syntax.

7.2 Simple Example

Here's a minimal example for a vessel depressurization calculation:

```yaml
vessel:
  length: 2.0
  diameter: 0.5
  orientation: "vertical"

initial:
  pressure: 15000000
  temperature: 293.15
  fluid: "Hydrogen"

calculation:
  type: "isentropic"
  time_step: 0.1
  end_time: 100

valve:
  flow: "discharge"
  type: "orifice"
  diameter: 0.01
  discharge_coef: 0.84
  back_pressure: 101325
```

Save this as `input.yml` and run:

```
python scripts/hyddown_main.py input.yml
```

7.3 Streamlit Application

HydDown includes an interactive web interface built with Streamlit:

```
streamlit run scripts/streamlit_app.py
```

This provides a user-friendly interface for:

- Setting up calculations without writing YAML
- Visualizing results in real-time
- Comparing with validation data

7.3.1 Available Streamlit Apps

- `streamlit_app.py` - Main application
- `streamlit_genapp.py` - General purpose calculator
- `streamlit_h2app.py` - Hydrogen-specific calculator
- `streamlit_sbapp.py` - Stefan-Boltzmann fire scenario
- `streamlit_bdv_sbapp.py` - Blowdown valve with fire

7.4 Example Files

Example input files for various calculation types are located in:

```
src/hyddown/examples/
```

These include:

- Isothermal filling/discharge
- Isentropic expansion
- Isenthalpic throttling
- Energy balance with heat transfer
- Fire scenarios
- Two-phase calculations
- Relief valve sizing

CHAPTER

EIGHT

EXAMPLES

This page provides examples of common HydDown calculations.

8.1 Vessel Depressurization

8.1.1 Isentropic Depressurization

Calculate adiabatic depressurization of a hydrogen vessel through an orifice:

```yaml
vessel:
  length: 2.0
  diameter: 0.5
  orientation: "vertical"
  type: "Flat-end"

initial:
  pressure: 15000000
  temperature: 293.15
  fluid: "Hydrogen"

calculation:
  type: "isentropic"
  time_step: 0.1
  end_time: 100

valve:
  flow: "discharge"
  type: "orifice"
  diameter: 0.01
  discharge_coef: 0.84
  back_pressure: 101325
```

8.2 Vessel Filling

8.2.1 Filling with Heat Transfer

Model vessel filling with heat transfer to surroundings:

```yaml
vessel:
  length: 2.0
```

(continues on next page)

(continued from previous page)

```
  diameter: 0.5
  thickness: 0.01
  heat_capacity: 500
  density: 7800
  orientation: "vertical"

initial:
  pressure: 100000
  temperature: 293.15
  fluid: "Hydrogen"

calculation:
  type: "energybalance"
  time_step: 0.1
  end_time: 300

valve:
  flow: "filling"
  type: "controlvalve"
  Cv: 0.1
  back_pressure: 20000000

heat_transfer:
  type: "specified_h"
  h_inner: 100
  h_outer: 10
  temp_ambient: 293.15
```

8.3 Fire Scenario

8.3.1 Pool Fire Heat Load

Simulate vessel response to a pool fire:

```
vessel:
  length: 5.0
  diameter: 1.0
  thickness: 0.02
  heat_capacity: 500
  density: 7800
  orientation: "horizontal"
  type: "ASME F&D"

initial:
  pressure: 10000000
  temperature: 293.15
  fluid: "Hydrogen"

calculation:
  type: "energybalance"
  time_step: 0.1
```

(continues on next page)

(continued from previous page)

```
  end_time: 600

valve:
  flow: "discharge"
  type: "relief"
  set_pressure: 12000000
  back_pressure: 101325

heat_transfer:
  type: "s-b"
  fire: "api_pool"
```

8.4 Relief Valve Sizing

8.4.1 API 521 Relief Valve

Size a relief valve for fire scenario per API 521:

```
vessel:
  length: 10.0
  diameter: 2.0
  thickness: 0.03
  heat_capacity: 500
  density: 7800
  orientation: "horizontal"
  type: "Hemispherical"

initial:
  pressure: 8000000
  temperature: 293.15
  fluid: "Methane"

calculation:
  type: "energybalance"
  time_step: 0.5
  end_time: 1800

valve:
  flow: "discharge"
  type: "relief"
  set_pressure: 8500000
  back_pressure: 100000

heat_transfer:
  type: "s-b"
  fire: "api_jet"
```

8.5 Two-Phase Calculations

8.5.1 Liquid Hydrogen Storage

Model two-phase hydrogen storage with boiling:

```yaml
vessel:
  length: 3.0
  diameter: 1.0
  thickness: 0.015
  heat_capacity: 900
  density: 2700
  thermal_conductivity: 200
  orientation: "vertical"
  type: "Flat-end"
  liquid_level: 0.5

initial:
  pressure: 200000
  temperature: 25.0
  fluid: "Hydrogen"

calculation:
  type: "energybalance"
  time_step: 1.0
  end_time: 3600

valve:
  flow: "discharge"
  type: "mdot"
  mdot: 0.01
  back_pressure: 101325

heat_transfer:
  type: "specified_h"
  h_inner: 50
  h_outer: 5
  temp_ambient: 293.15
```

CHAPTER

NINE

HDCLASS - MAIN CALCULATION ENGINE

The `hdclass` module contains the core `HydDown` class that manages the calculation engine, time-stepping integration, and thermodynamic state calculations. Main calculation engine for pressure vessel filling and discharge simulations.

This module contains the HydDown class, which is the core of the HydDown package. It integrates mass and energy balances over time to simulate pressure vessel depressurization (discharge) and pressurization (filling) with heat transfer effects.

The HydDown class: - Reads and validates YAML input defining vessel geometry, initial conditions,

calculation type, valve parameters, and heat transfer settings

- Initializes thermodynamic state using CoolProp for fluid properties
- Integrates mass and energy balances using explicit Euler time stepping
- Calculates heat transfer between fluid and vessel wall
- Handles various thermodynamic paths: isothermal, isenthalpic, isentropic, constant internal energy, and full energy balance
- Supports multiple valve types: orifice, control valve, relief valve, constant mass flow
- Models fire heat loads using Stefan-Boltzmann approach
- Tracks two-phase systems with separate gas/liquid temperatures
- Can model 1-D transient heat conduction through vessel walls (composite materials)
- Stores time-series results and provides plotting capabilities

Calculation types: - isothermal: Constant temperature (very slow process with large heat reservoir) - isenthalpic: Constant enthalpy (adiabatic, no work) - isentropic: Constant entropy (adiabatic with PV work) - specified_U: Constant internal energy - energybalance: Full energy balance with heat transfer and work

Heat transfer modes (for energybalance): - fixed_U: Fixed overall heat transfer coefficient - fixed_Q: Fixed heat input rate - specified_h: Specified internal/external heat transfer coefficients - detailed: 1-D transient conduction through vessel wall - fire: External fire heat load using Stefan-Boltzmann equation

Valve types: - orifice: Compressible flow through orifice (Yellow Book equation) - control_valve: Control valve with Cv characteristic - relief_valve: API 520/521 relief valve sizing - mdot: Constant mass flow rate

The integration scheme uses explicit Euler method with user-specified time step. Results are stored in numpy arrays for time, pressure, temperature, mass flow, etc.

Typical usage:

import yaml from hyddown import HydDown

with open('input.yml') as f:

input_data = yaml.load(f, Loader=yaml.FullLoader)

hdown = HydDown(input_data) hdown.run() hdown.plot()

class hyddown.hdclass.**HydDown**(*input*)

Bases: object

Main class to to hold problem definition, running problem, storing results etc.

__init__(*input*)

Parameters
input (*dict*) – Dict holding problem definition

validate_input()

Validating the provided problem definition dict

Raises
ValueError – If missing input is detected.

read_input()

Reading in input/ problem definition dict and assigning to classs attributes.

initialize()

Preparing for running problem by creating the fluid objects required instantiating arrays for storing time-dependent results, setting additional required class attributes.

calc_liquid_level()

Calculate liquid level height based on current two-phase fluid state.

For two-phase systems (0 quality 1), calculates the height of liquid phase in the vessel based on vapor quality, phase densities, and vessel geometry. Uses vessel geometry from fluids.TANK to convert liquid volume to height.

Parameters
fluid (*CoolProp AbstractState*) – Current fluid state

Returns
Liquid level height from vessel bottom [m]. Returns 0.0 for single-phase gas (quality > 1 or subcooled liquid).

Return type
float

PHres(*T*, *P*, *H*)

Residual enthalpy function to be minimized during PH-problem.

Used by numerical optimizer (scipy.optimize.minimize) to find temperature that satisfies constant pressure-enthalpy constraints for multicomponent fluids. The optimizer adjusts T until residual approaches zero.

Parameters

- **H** (*float*) – Enthalpy at initial/final conditions [J/kg]
- **P** (*float*) – Pressure at final conditions [Pa]
- **T** (*float*) – Updated estimate for the final temperature at (P,H) [K]

Returns
Squared normalized enthalpy residual (dimensionless). Zero when correct temperature is found.

Return type
float

PHres_relief(*T*, *P*, *H*)

Residual enthalpy function for PH-problem during relief valve calculations.

Used by numerical optimizer (scipy.optimize.root_scalar) to find temperature for relief valve discharge calculations. Similar to PHres() but uses the main fluid state instead of vent_fluid state.

Parameters

- **H** (*float*) – Enthalpy at initial/final conditions [J/kg]
- **P** (*float*) – Pressure at final conditions [Pa]
- **T** (*float*) – Updated estimate for the final temperature at (P,H) [K]

Returns

Normalized enthalpy residual (dimensionless). Zero when correct temperature is found.

Return type

float

PHproblem(*H*, *P*, *Tguess*, *relief=False*)

Defining a constant pressure, constant enthalpy problem i.e. typical adiabatic problem like e.g. valve flow for the vented flow (during discharge). For multicomponent mixture the final temperature is changed/optimised until the residual enthalpy is near zero in an optimisation step. For single component fluid the coolprop built in methods are used for speed.

Parameters

- **H** (*float*) – Enthalpy at initial/final conditions
- **P** (*float*) – Pressure at final conditions.
- **Tguess** (*float*) – Initial guess for the final temperature at P,H

UDres(*x*, *U*, *rho*)

Residual U-rho to be minimised during a U-rho/UV-problem

Parameters

- **U** (*float*) – Internal energy at final conditions
- **rho** (*float*) – Density at final conditions

UDproblem(*U*, *rho*, *Pguess*, *Tguess*)

Defining a constant UV problem i.e. constant internal energy and density/volume problem relevant for the 1. law of thermodynamics. For multicomponent mixture the final temperature/pressure is changed/optimised until the residual U/rho is near zero. For single component fluid the coolprop built in methods are used for speed.

Parameters

- **U** (*float*) – Internal energy at final conditions
- **rho** (*float*) – Density at final conditions.
- **Pguess** (*float*) – Initial guess for the final pressure at U, rho
- **Tguess** (*float*) – Initial guess for the final temperature at U, rho

run(*disable_pbar=True*)

Routine for running the actual problem defined i.e. integrating the mass and energy balances

get_dataframe()

Export simulation results to pandas DataFrame for analysis and archiving.

Collects all time-series results from the simulation and organizes them into a pandas DataFrame with labeled columns. Useful for exporting to CSV/Excel, post-processing, or custom plotting.

Returns

DataFrame with simulation results. Columns include: - Time (s) - Pressure (bar) - Fluid temperature (°C) - Wall temperature (°C) - Vent temperature (°C) - Fluid enthalpy (J/kg) - Fluid entropy (J/(kg·K)) - Fluid internal energy (J/kg) - Discharge mass rate (kg/s) - Fluid mass (kg) - Fluid density (kg/m^3) - Inner heat transfer coefficient (W/(m^2·K)) - Internal heat flux (W/m^2) - External heat flux (W/m^2) - Inner wall temperature (°C) - Outer wall temperature (°C)

Return type

pd.DataFrame

Notes

Only returns data if simulation has been run (self.isrun == True). Temperature values are converted from Kelvin to Celsius. Pressure values are converted from Pa to bar.

plot(*filename=None*, *verbose=True*)

Creating standard plots for the solved problem

Parameters

- **filename** (*str*) – Saving plots to filename if provideed (optional)
- **verbose** (*bool*) – Plotting on screen if True (optional)

plot_envelope(*filename=None*, *verbose=True*)

Creating standard plots for the solved problem

Parameters

- **filename** (*str*) – Saving plots to filename if provideed (optional)
- **verbose** (*bool*) – Plotting on screen if True (optional)

plot_tprofile(*filename=None*, *verbose=True*)

Creating standard plots for the solved problem

Parameters

- **filename** (*str*) – Saving plots to filename if provideed (optional)
- **verbose** (*bool*) – Plotting on screen if True (optional)

analyze_rupture(*filename=None*)

Analyze vessel rupture potential under fire exposure conditions.

Performs a simplified rupture analysis by calculating vessel wall temperatures under external fire heat load and comparing von Mises stress (from internal pressure) against temperature-dependent allowable tensile stress (ATS). Determines if and when vessel rupture may occur.

The analysis: 1. Calculates wall temperature evolution under fire exposure 2. Computes von Mises equivalent stress from internal pressure 3. Evaluates temperature-dependent material strength (ATS) 4. Identifies rupture time when stress exceeds strength 5. Generates diagnostic plots

Parameters
filename (*str, optional*) – Base filename for saving plots. If None, plots are displayed on screen. Generates two plot files: - {filename}_peak_wall_temp.png - {filename}_ATS_vonmises.png

Returns

Results are stored in instance variables: - self.rupture_time : float or None

Time when rupture occurs [s], or None if no rupture predicted

- **self.peak_times**
  [ndarray] Time array for rupture analysis [s]
- **self.von_mises**
  [ndarray] von Mises equivalent stress history [Pa]
- **self.ATS_wetted**
  [ndarray] Allowable tensile stress for wetted region [Pa]
- **self.ATS_unwetted**
  [ndarray] Allowable tensile stress for unwetted region [Pa]
- **self.peak_T_wetted**
  [ndarray] Wall temperature history for wetted region [K]
- **self.peak_T_unwetted**
  [ndarray] Wall temperature history for unwetted region [K]

Return type
None

Notes

Requires rupture analysis parameters in input: - self.rupture_fire: Fire type (e.g., 'api_pool', 'scandpower_jet') - self.rupture_material: Material type for ATS calculation

The method uses a simplified lumped capacitance model for wall heating. Time step for rupture analysis is fixed at 10 seconds.

generate_report()

Generating a report summarising key features for the problem solved. Can be used for e.g. case studies, problem optimisation (external) etc.

9.1 Key Classes

9.1.1 HydDown

class hyddown.hdclass.**HydDown**(*input*)

Bases: object

Main class to to hold problem definition, running problem, storing results etc.

__init__(*input*)

Parameters
input (*dict*) – Dict holding problem definition

validate_input()

Validating the provided problem definition dict

Raises
ValueError – If missing input is detected.

read_input()

Reading in input/ problem definition dict and assigning to classs attributes.

initialize()

Preparing for running problem by creating the fluid objects required instantiating arrays for storing time-dependent results, setting additional required class attributes.

calc_liquid_level()

Calculate liquid level height based on current two-phase fluid state.

For two-phase systems (0 quality 1), calculates the height of liquid phase in the vessel based on vapor quality, phase densities, and vessel geometry. Uses vessel geometry from fluids.TANK to convert liquid volume to height.

Parameters
fluid (*CoolProp AbstractState*) – Current fluid state

Returns
Liquid level height from vessel bottom [m]. Returns 0.0 for single-phase gas (quality > 1 or subcooled liquid).

Return type
float

PHres(*T*, *P*, *H*)

Residual enthalpy function to be minimized during PH-problem.

Used by numerical optimizer (scipy.optimize.minimize) to find temperature that satisfies constant pressure-enthalpy constraints for multicomponent fluids. The optimizer adjusts T until residual approaches zero.

Parameters

- **H** (*float*) – Enthalpy at initial/final conditions [J/kg]
- **P** (*float*) – Pressure at final conditions [Pa]
- **T** (*float*) – Updated estimate for the final temperature at (P,H) [K]

Returns
Squared normalized enthalpy residual (dimensionless). Zero when correct temperature is found.

Return type
float

PHres_relief(*T*, *P*, *H*)

Residual enthalpy function for PH-problem during relief valve calculations.

Used by numerical optimizer (scipy.optimize.root_scalar) to find temperature for relief valve discharge calculations. Similar to PHres() but uses the main fluid state instead of vent_fluid state.

Parameters

- **H** (*float*) – Enthalpy at initial/final conditions [J/kg]
- **P** (*float*) – Pressure at final conditions [Pa]

- **T** (*float*) – Updated estimate for the final temperature at (P,H) [K]

Returns
Normalized enthalpy residual (dimensionless). Zero when correct temperature is found.

Return type
float

PHproblem(*H*, *P*, *Tguess*, *relief=False*)
Defining a constant pressure, constant enthalpy problem i.e. typical adiabatic problem like e.g. valve flow for the vented flow (during discharge). For multicomponent mixture the final temperature is changed/optimised until the residual enthalpy is near zero in an optimisation step. For single component fluid the coolprop built in methods are used for speed.

Parameters

- **H** (*float*) – Enthalpy at initial/final conditions
- **P** (*float*) – Pressure at final conditions.
- **Tguess** (*float*) – Initial guess for the final temperature at P,H

UDres(*x*, *U*, *rho*)
Residual U-rho to be minimised during a U-rho/UV-problem

Parameters

- **U** (*float*) – Internal energy at final conditions
- **rho** (*float*) – Density at final conditions

UDproblem(*U*, *rho*, *Pguess*, *Tguess*)
Defining a constant UV problem i.e. constant internal energy and density/volume problem relevant for the 1. law of thermodynamics. For multicomponent mixture the final temperature/pressure is changed/optimised until the residual U/rho is near zero. For single component fluid the coolprop built in methods are used for speed.

Parameters

- **U** (*float*) – Internal energy at final conditions
- **rho** (*float*) – Density at final conditions.
- **Pguess** (*float*) – Initial guess for the final pressure at U, rho
- **Tguess** (*float*) – Initial guess for the final temperature at U, rho

run(*disable_pbar=True*)
Routine for running the actual problem defined i.e. integrating the mass and energy balances

get_dataframe()
Export simulation results to pandas DataFrame for analysis and archiving.

Collects all time-series results from the simulation and organizes them into a pandas DataFrame with labeled columns. Useful for exporting to CSV/Excel, post-processing, or custom plotting.

Returns
DataFrame with simulation results. Columns include: - Time (s) - Pressure (bar) - Fluid temperature (°C) - Wall temperature (°C) - Vent temperature (°C) - Fluid enthalpy (J/kg) - Fluid entropy (J/(kg·K)) - Fluid internal energy (J/kg) - Discharge mass rate (kg/s) - Fluid mass (kg) - Fluid density (kg/m^3) - Inner heat transfer coefficient (W/(m^2·K)) - Internal heat flux (W/m^2) - External heat flux (W/m^2) - Inner wall temperature (°C) - Outer wall temperature (°C)

Return type
pd.DataFrame

Notes

Only returns data if simulation has been run (self.isrun == True). Temperature values are converted from Kelvin to Celsius. Pressure values are converted from Pa to bar.

plot(*filename=None*, *verbose=True*)

Creating standard plots for the solved problem

Parameters

- **filename** (*str*) – Saving plots to filename if provideed (optional)
- **verbose** (*bool*) – Plotting on screen if True (optional)

plot_envelope(*filename=None*, *verbose=True*)

Creating standard plots for the solved problem

Parameters

- **filename** (*str*) – Saving plots to filename if provideed (optional)
- **verbose** (*bool*) – Plotting on screen if True (optional)

plot_tprofile(*filename=None*, *verbose=True*)

Creating standard plots for the solved problem

Parameters

- **filename** (*str*) – Saving plots to filename if provideed (optional)
- **verbose** (*bool*) – Plotting on screen if True (optional)

analyze_rupture(*filename=None*)

Analyze vessel rupture potential under fire exposure conditions.

Performs a simplified rupture analysis by calculating vessel wall temperatures under external fire heat load and comparing von Mises stress (from internal pressure) against temperature-dependent allowable tensile stress (ATS). Determines if and when vessel rupture may occur.

The analysis: 1. Calculates wall temperature evolution under fire exposure 2. Computes von Mises equivalent stress from internal pressure 3. Evaluates temperature-dependent material strength (ATS) 4. Identifies rupture time when stress exceeds strength 5. Generates diagnostic plots

Parameters
filename (*str, optional*) – Base filename for saving plots. If None, plots are displayed on screen. Generates two plot files: - {filename}_peak_wall_temp.png - {filename}_ATS_vonmises.png

Returns

Results are stored in instance variables: - self.rupture_time : float or None

Time when rupture occurs [s], or None if no rupture predicted

- **self.peak_times**
  [ndarray] Time array for rupture analysis [s]
- **self.von_mises**
  [ndarray] von Mises equivalent stress history [Pa]

- **self.ATS_wetted**
  [ndarray] Allowable tensile stress for wetted region [Pa]
- **self.ATS_unwetted**
  [ndarray] Allowable tensile stress for unwetted region [Pa]
- **self.peak_T_wetted**
  [ndarray] Wall temperature history for wetted region [K]
- **self.peak_T_unwetted**
  [ndarray] Wall temperature history for unwetted region [K]

Return type
None

Notes

Requires rupture analysis parameters in input: - self.rupture_fire: Fire type (e.g., 'api_pool', 'scandpower_jet') - self.rupture_material: Material type for ATS calculation

The method uses a simplified lumped capacitance model for wall heating. Time step for rupture analysis is fixed at 10 seconds.

generate_report()

Generating a report summarising key features for the problem solved. Can be used for e.g. case studies, problem optimisation (external) etc.

9.1.2 Key Methods

The `HydDown` class implements the following key methods:

- `run()`: Main integration loop for mass/energy balances
- `step()`: Single time step calculation
- `PHproblem()`: Solve thermodynamic state from pressure and enthalpy
- `UDproblem()`: Solve thermodynamic state from internal energy and density
- `plot()`: Generate results visualization

9.2 Implementation Notes

9.2.1 Time Integration

The code uses an explicit Euler method for mass balance integration. The time step `dt` must be small enough for numerical stability (typically 0.01-1 second).

9.2.2 Calculation Types

The `calculation.type` parameter determines the thermodynamic path:

- `isothermal`: Constant temperature
- `isenthalpic`: Constant enthalpy, adiabatic expansion without work
- `isentropic`: Constant entropy, adiabatic expansion with PV work
- `specified_U`: Constant internal energy

- `energybalance`: Most general case with heat transfer

CHAPTER

TEN

TRANSPORT - HEAT AND MASS TRANSFER

The `transport` module provides calculations for heat and mass transfer, including dimensionless numbers, heat transfer coefficients, and mass flow rate calculations for different valve types. Heat and mass transfer correlations for vessel depressurization/pressurization.

This module provides functions for calculating: - Dimensionless numbers (Grashof, Prandtl, Nusselt, Rayleigh) - Heat transfer coefficients for natural and forced convection - Mass flow rates through orifices, control valves, and relief valves - Boiling heat transfer (pool boiling, film boiling)

The correlations are based on established literature including: - Geankoplis, Transport Processes and Unit Operations - API Standard 520/521 for relief valve sizing - Rohsenow pool boiling correlation

All functions use SI units unless otherwise specified. CoolProp is used as the thermodynamic backend for fluid property calculations.

hyddown.transport.**Gr**(*L*, *Tfluid*, *Tvessel*, *P*, *species*)

Calculation of Grasshof number. See eq. 4.7-4 in C. J. Geankoplis Transport Processes and Unit Operations, International Edition, Prentice-Hall, 1993

Parameters

- **L** (*float*) – Vessel length
- **Tfluid** (*float*) – Temperature of the bulk fluid inventory
- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **P** (*float*) – Pressure of fluid inventory

Returns
Gr – Grasshof number

Return type
float

hyddown.transport.**Pr**(*T*, *P*, *species*)

Calculation of Prandtl number, eq. 4.5-6 in C. J. Geankoplis Transport Processes and Unit Operations, International Edition, Prentice-Hall, 1993

Parameters

- **T** (*float*) – Temperature of the fluid film interface
- **P** (*float*) – Pressure of fluid inventory

Returns
Pr – Prantdl number

Return type
float

hyddown.transport.**Nu**(*Ra*, *Pr*)

Calculation of Nusselt number for natural convection. See eq. 4.7-4 and Table 4.7-1 in C. J. Geankoplis Transport Processes and Unit Operations, International Edition, Prentice-Hall, 1993

Parameters

- **Ra** (*float*) – Raleigh number
- **Pr** (*float*) – Prandtl number

Returns
Nu – Nusselt numebr

Return type
float

hyddown.transport.**h_inside**(*L*, *Tvessel*, *Tfluid*, *fluid*)

Calculation of internal natural convective heat transfer coefficient from Nusselt number and using the coolprop low level interface.

Parameters

- **L** (*float*) – Vessel length
- **Tfluid** (*float*) – Temperature of the bulk fluid inventory
- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **fluid** (*obj*) – Coolprop fluid object

Returns
h_inner – Heat transfer coefficient

Return type
float

hyddown.transport.**h_inner**(*L*, *Tfluid*, *Tvessel*, *P*, *species*)

Calculation of internal natural convective heat transfer coefficient from Nusselt number and using the coolprop high level interface. Not currently in use.

Parameters

- **L** (*float*) – Vessel length
- **Tfluid** (*float*) – Temperature of the bulk fluid inventory
- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **species** (*str*) – Fluid definition string

Returns
h_inner – Heat transfer coefficient

Return type
float

hyddown.transport.**h_inside_mixed**(*L*, *Tvessel*, *Tfluid*, *fluid*, *mdot*, *D*)

Calculation of internal mixed natural/forced convective heat transfer coefficient from Nusselt number and using the coolprop low level interface.

Parameters

- **L** (*float*) – Vessel length
- **Tfluid** (*float*) – Temperature of the bulk fluid inventory

- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **fluid** (*obj*) – Coolprop fluid object
- **mdot** (*float*) – Mass flow
- **D** (*float*) – Characteristic diameter for Reynolds number estimation

Returns
h_inner – Heat transfer coefficient

Return type
float

hyddown.transport.**h_inner_mixed**(*L*, *Tfluid*, *Tvessel*, *P*, *species*, *mdot*, *D*)

Calculation of internal mixed (nutural/forced convective) heat transfer coefficient from Nusselt number and using the coolprop high level interface. Not currently in use.

Parameters

- **L** (*float*) – Vessel length
- **Tfluid** (*float*) – Temperature of the bulk fluid inventory
- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **P** (*float*) – Fluid pressure
- **species** (*str*) – Fluid definition string
- **mdot** (*float*) – Mass flow
- **D** (*float*) – Characteristic diameter for Reynolds number estimation

Returns
h_inner – Heat transfer coefficient

Return type
float

hyddown.transport.**h_inside_liquid**(*L*, *Tvessel*, *Tfluid*, *fluid*)

Calculation of internal natural convective heat transfer coefficient from Nusselt number

Parameters

- **L** (*float*) – Vessel length
- **Tfluid** (*float*) – Temperature of the bulk fluid inventory
- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **fluid** (*obj*) – Liquid Fluid object equilibrated at film temperature
- **master_fluid** (*obj*) – Master Fluid object (used for surface tension etc.)

Returns
h_inner – Heat transfer coefficient (W/m2 K)

Return type
float

hyddown.transport.**h_inside_wetted**(*L*, *Tvessel*, *Tfluid*, *fluid*, *master_fluid*)

Calculation of internal heat transfer coefficient for boiling liquid

Parameters

- **L** (*float*) – Vessel length

- **Tfluid** (*float*) – Temperature of the bulk fluid inventory
- **Tvessel** (*float*) – Temperature of the vessel wall (bulk)
- **fluid** (*obj*) – Gas object equilibrated at film temperature

Returns
h_inner – Heat transfer coefficient (W/m2 K)

Return type
float

hyddown.transport.**hem_release_rate**(*P1*, *Pback*, *Cd*, *area*, *fluid*)

Fluid mass flow (kg/s) trough a hole at critical (sonic) or subcritical flow conditions calculated applying the HEM (Homogenous Equilibrium Model) assumption.

Parameters

- **P1** (*float*) – Upstream pressure
- **Pback** (*float*) – Back/downstream pressure
- **Cd** (*float*) – Coefficient of discharge
- **area** (*float*) – Orifice area
- **fluid** (*obj*) – Fluid object

Returns
: float Gas release rate / mass flow of discharge

hyddown.transport.**gas_release_rate**(*P1*, *P2*, *rho*, *k*, *CD*, *area*)

Gas mass flow (kg/s) trough a hole at critical (sonic) or subcritical flow conditions. The formula is based on Yellow Book equation 2.22.

Methods for the calculation of physical effects, CPR 14E, van den Bosch and Weterings (Eds.), 1996

Parameters

- **P1** (*float*) – Upstream pressure
- **P2** (*float*) – Downstream pressure
- **rho** (*float*) – Fluid density
- **k** (*float*) – Ideal gas k (Cp/Cv)
- **CD** (*float*) – Coefficient of discharge
- **area** (*float*) – Orifice area

Returns
: float Gas release rate / mass flow of discharge

hyddown.transport.**relief_valve**(*P1*, *Pback*, *Pset*, *blowdown*, *k*, *CD*, *T1*, *Z*, *MW*, *area*)

Pop action relief valve model including hysteresis. The pressure shall rise above P_set to open and decrease below P_reseat (P_set*(1-blowdown)) to close

Parameters

- **P1** (*float*) – Upstream pressure
- **Pback** (*float*) – Downstream / backpressure
- **Pset** (*float*) – Set pressure of the PSV / relief valve
- **blowdown** (*float*) – The percentage of the set pressure at which the valve reseats

- **k** (*float*) – Ideal gas k (Cp/Cv)
- **CD** (*float*) – Coefficient of discharge
- **T1** (*float*) – Upstream temperature
- **Z** (*float*) – Compressibility
- **MW** (*float*) – Molecular weight of the gas relieved
- **area** (*float*) – PSV orifice area

Returns
: float Relief rate / mass flow

hyddown.transport.**api_psv_release_rate**(*P1*, *Pback*, *k*, *CD*, *T1*, *Z*, *MW*, *area*)

PSV vapour relief rate calculated according to API 520 Part I 2014 Eq. 5, 9, 15, 18

Parameters

- **P1** (*float*) – Upstream pressure
- **Pback** (*float*) – Downstream / backpressure
- **k** (*float*) – Ideal gas k (Cp/Cv)
- **CD** (*float*) – Coefficient of discharge
- **T1** (*float*) – Upstream temperature
- **Z** (*float*) – Compressibility
- **MW** (*float*) – Molecular weight of the gas relieved
- **area** (*float*) – PSV orifice area

Returns
: float Relief rate / mass flow

hyddown.transport.**cv_vs_time**(*Cv_max*, *t*, *time_constant=0*, *characteristic='linear'*)

Control valve flow coefficient vs time / actuator postion assuming a linear rate of actuator for the three archetypes of characteristics: linear, equal percentage and fast/quick opening.

Parameters

- **Cv_max** (*float*) – Valve flow coefficient at full open position
- **t** (*float*) – Time
- **time_constant** (*float (optional)*) – The time required for the actuator to fully open. Default to instant open
- **characteristic** (*string (optional)*) – Valve characteristic Default to linear.

hyddown.transport.**control_valve**(*P1*, *P2*, *T*, *Z*, *MW*, *gamma*, *Cv*, *xT=0.75*, *FP=1*)

Flow calculated from ANSI/ISA control valve equations for single phase gas flow. Equation 19 pp. 132 in Control Valves / Guy Borden, editor; Paul Friedmann, style editor

Parameters

- **P1** (*float*) – Upstream pressure
- **P2** (*float*) – Downstream / backpressure
- **T** (*float*) – Upstream temperature
- **Z** (*float*) – Upstream compressibility

- **MW** (*float*) – Molecular weight of the gas relieved
- **gamma** (*float*) – Upstream Ideal gas k (Cp/Cv)
- **Cv** (*float*) – Valve coefficient
- **xT** (*float*) – Value of xT for valve fitting assembly, default value
- **FP** (*float*) – Piping geometry factor

Returns
: float Mass flow

10.1 Dimensionless Numbers

The module calculates key dimensionless numbers for heat transfer correlations:

- Grashof number (Gr)
- Prandtl number (Pr)
- Nusselt number (Nu)
- Rayleigh number (Ra)

10.2 Heat Transfer Coefficients

Functions for calculating heat transfer coefficients for:

- Natural convection
- Forced convection
- Pool boiling
- Film boiling

10.3 Mass Flow Rate Calculations

The module provides mass flow rate functions for different valve types. These functions are called internally by the main HydDown class based on the valve type specified in the input file.

10.4 Valve Types

The `valve.type` parameter determines the mass flow calculation:

- `orifice`: Compressible flow through orifice (requires `diameter`, `discharge_coef`)
- `control_valve`: Control valve sizing equation (requires `Cv`, `N9`)
- `relief_valve`: API 520/521 relief valve (requires `diameter`, `set_pressure`)
- `mdot`: Constant mass flow rate (requires `mass_flow`)

Flow direction is set by `valve.flow`: `"discharge"` or `"filling"`.

CHAPTER

ELEVEN

FIRE - FIRE HEAT LOAD MODELING

The `fire` module provides fire scenario modeling using the Stefan-Boltzmann radiation approach combined with convection heat transfer. Fire heat load calculations using the Stefan-Boltzmann approach.

This module provides functions to calculate heat flux from fire scenarios (pool fire, jet fire) to pressure vessel surfaces using combined radiative and convective heat transfer models. The models implement industry-standard approaches from API 521 and Scandpower guidelines.

The Stefan-Boltzmann equation accounts for: - Radiative heat transfer from flame to vessel - Convective heat transfer from hot gases - Radiative emission from vessel surface

Available fire scenarios: - API 521 pool fire: 60 kW/m^2 incident heat flux - API 521 jet fire: 100 kW/m^2 incident heat flux - Scandpower pool fire: 100 kW/m^2 incident heat flux - Scandpower jet fire: 100 kW/m^2 incident heat flux - Scandpower peak fires: 150-350 kW/m^2 incident heat flux

References: - API Standard 521, Pressure-relieving and Depressuring Systems - Scandpower Risk Management AS guidelines for fire scenarios

hyddown.fire.**stefan_boltzmann**(*alpha*, *e_flame*, *e_surface*, *h*, *Tflame*, *Tradiative*, *Tvessel*)

Calculate heat flux from flame to vessel surface using Stefan-Boltzmann equation.

Combines radiative and convective heat transfer to determine the net heat flux to a vessel surface exposed to fire conditions.

Parameters

- **alpha** (*float*) – Absorptivity of the vessel surface (dimensionless, 0-1)
- **e_flame** (*float*) – Emissivity of the flame (dimensionless, 0-1)
- **e_surface** (*float*) – Emissivity of the vessel surface (dimensionless, 0-1)
- **h** (*float*) – Convective heat transfer coefficient [W/(m^2·K)]
- **Tflame** (*float*) – Temperature of the flame/hot gases [K]
- **Tradiative** (*float*) – Effective radiative temperature of the flame [K]
- **Tvessel** (*float*) – Temperature of the vessel surface [K]

Returns

Net heat flux to vessel surface [W/m^2]

Return type

float

Notes

The heat flux is calculated as: Q = ·_flame··T_rad^4 + h·(T_flame - T_vessel) - _surface··T_vessel^4

where = 5.67×10^{8} W/(m^{2}·K^{4}) is the Stefan-Boltzmann constant.

See also API 521 and Scandpower guidelines for typical parameter values.

hyddown.fire.**pool_fire_api521**(*Tvessel*)

Calculate heat flux for API 521 pool fire scenario.

Implements API 521 standard pool fire with typical incident heat flux of 60 kW/m^{2}. Uses conservative parameter values for absorptivity, emissivity, and heat transfer coefficient as specified in API Standard 521.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^{2}]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.75 - Flame emissivity (_flame): 0.75 - Surface emissivity (_surface): 0.75 - Convective heat transfer coefficient (h): 20 W/(m^{2}·K) - Flame temperature: 600°C (873.15 K) - Radiative temperature: 750°C (1023.15 K)

References

API Standard 521, Pressure-relieving and Depressuring Systems

hyddown.fire.**pool_fire_scandpower**(*Tvessel*)

Calculate heat flux for Scandpower pool fire scenario.

Implements Scandpower pool fire model with typical incident heat flux of 100 kW/m^{2}. Uses higher absorptivity and emissivity values compared to API 521 for a more severe fire scenario.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^{2}]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.85 - Flame emissivity (_flame): 1.0 - Surface emissivity (_surface): 0.85 - Convective heat transfer coefficient (h): 30 W/(m^2·K) - Flame temperature: 804°C (1077.15 K) - Radiative temperature: 804°C (1077.15 K)

References

Scandpower Risk Management AS guidelines for fire scenarios

hyddown.fire.**jet_fire_api521**(*Tvessel*)

Calculate heat flux for API 521 jet fire scenario.

Implements API 521 standard jet fire with typical incident heat flux of 100 kW/m^2. Jet fires have higher flame temperatures and convective heat transfer compared to pool fires due to forced convection from jet momentum.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^2]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.75 - Flame emissivity (_flame): 0.33 - Surface emissivity (_surface): 0.75 - Convective heat transfer coefficient (h): 40 W/(m^2·K) - Flame temperature: 900°C (1173.15 K) - Radiative temperature: 1100°C (1373.15 K)

References

API Standard 521, Pressure-relieving and Depressuring Systems

hyddown.fire.**jet_fire_scandpower**(*Tvessel*)

Calculate heat flux for Scandpower jet fire scenario.

Implements Scandpower jet fire model with typical incident heat flux of 100 kW/m^2. Features high convective heat transfer coefficient due to forced convection from the jet impingement.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^2]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.85 - Flame emissivity (_flame): 1.0 - Surface emissivity (_surface): 0.85 - Convective heat transfer coefficient (h): 100 W/(m^2·K) - Flame temperature: 635°C (908.15 K) - Radiative temperature: 635°C (908.15 K)

References

Scandpower Risk Management AS guidelines for fire scenarios

hyddown.fire.**jet_fire_peak_large_scandpower**(*Tvessel*)

Calculate heat flux for Scandpower large peak jet fire scenario.

Implements severe jet fire scenario with very high incident heat flux of 350 kW/m^2. This represents peak impingement conditions from large high-pressure gas releases in close proximity to vessel surfaces.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^2]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.85 - Flame emissivity (_flame): 1.0 - Surface emissivity (_surface): 0.85 - Convective heat transfer coefficient (h): 100 W/(m^2·K) - Flame temperature: 1429.61 K - Radiative temperature: 1429.61 K

References

Scandpower Risk Management AS guidelines for fire scenarios

hyddown.fire.**jet_fire_peak_small_scandpower**(*Tvessel*)

Calculate heat flux for Scandpower small peak jet fire scenario.

Implements severe jet fire scenario with high incident heat flux of 250 kW/m^2. This represents peak impingement conditions from moderate high-pressure gas releases or peripheral zones of large jet fires.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^2]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.85 - Flame emissivity (_flame): 1.0 - Surface emissivity (_surface): 0.85 - Convective heat transfer coefficient (h): 100 W/(m^2·K) - Flame temperature: 1279.29 K - Radiative temperature: 1279.29 K

References

Scandpower Risk Management AS guidelines for fire scenarios

hyddown.fire.**pool_fire_peak_scandpower**(*Tvessel*)

Calculate heat flux for Scandpower peak pool fire scenario.

Implements severe pool fire scenario with high incident heat flux of 150 kW/m^2. This represents very large pool fires or locations close to the fire source with maximum radiative and convective heat transfer.

Parameters
Tvessel (*float*) – Temperature of the vessel surface [K]

Returns
Net heat flux to vessel surface [W/m^2]

Return type
float

Notes

Fire parameters: - Absorptivity (): 0.85 - Flame emissivity (_flame): 1.0 - Surface emissivity (_surface): 0.85 - Convective heat transfer coefficient (h): 30 W/(m^2·K) - Flame temperature: 1212.54 K - Radiative temperature: 1212.54 K

References

Scandpower Risk Management AS guidelines for fire scenarios

hyddown.fire.**sb_fire**(*T_vessel*, *fire_type*)

Dispatcher function for Stefan-Boltzmann fire heat load calculations.

Selects and calls the appropriate fire scenario function based on the specified fire type string. This is the main entry point for fire heat load calculations used by the HydDown energy balance solver.

Parameters

- **T_vessel** (*float*) – Temperature of the vessel surface [K]
- **fire_type** (*str*) – Fire scenario type. Valid options: - “api_jet”: API 521 jet fire (100 kW/m^2) - “api_pool”: API 521 pool fire (60 kW/m^2) - “scandpower_pool”: Scandpower pool fire (100 kW/m^2) - “scandpower_jet”: Scandpower jet fire (100 kW/m^2) - “scandpower_jet_peak_large”: Scandpower large peak jet fire (350 kW/m^2) - “scandpower_jet_peak_small”: Scandpower small peak jet fire (250 kW/m^2) - “scandpower_pool_peak”: Scandpower peak pool fire (150 kW/m^2)

Returns
Net heat flux to vessel surface [W/m^2]

Return type
float

Raises
ValueError – If fire_type is not one of the recognized fire scenario strings

Examples

```
>>> T_vessel = 400  # K
>>> Q = sb_fire(T_vessel, "api_pool")
>>> print(f"Heat flux: {Q:.0f} W/m²")
```

11.1 Predefined Fire Scenarios

The module provides predefined fire scenarios with different heat flux intensities:

- **api_pool**: ~60 kW/m^2 incident heat flux per API 521 (pool fire)
- **api_jet**: ~100 kW/m^2 incident heat flux per API 521 (jet fire)
- **scandpower**: Higher intensity fire per Scandpower guidelines

11.2 Stefan-Boltzmann Calculation

Fire heat load is calculated using:

$$Q = varepsilon sigma A(T_{fire}^4 - T_{wall}^4) + h_{conv} A(T_{fire} - T_{wall})$$

where:

- $varepsilon$ is surface emissivity (built-in value)
- $sigma$ is the Stefan-Boltzmann constant (5.67×10^8 W/m^{2}K^4)
- A is the exposed surface area
- T_{fire} is the fire temperature
- T_{wall} is the vessel wall temperature
- h_{conv} is the convective heat transfer coefficient

11.3 Usage Example

To use a fire scenario in your input file:

```
heat_transfer:
  type: "s-b"        # Stefan-Boltzmann radiation model
  fire: "api_pool"   # API 521 pool fire scenario
```

CHAPTER

TWELVE

THERMESH - 1-D TRANSIENT HEAT CONDUCTION

The thermesh module provides 1-D transient heat conduction modeling for vessel walls using the finite element method. This is particularly important for Type III/IV vessels with low thermal conductivity composite materials. 1-D transient heat conduction solver using finite element method.

This module provides a finite element solver for transient heat conduction problems in 1-D domains. It is particularly useful for calculating temperature distributions through vessel walls with composite materials or low thermal conductivity.

The code is adapted from the thermesh library (https://github.com/wjbg/thermesh) and implements: - Linear and quadratic finite elements - Theta-method time integration (explicit, implicit, Crank-Nicolson) - Temperature-dependent material properties - Dirichlet and Neumann boundary conditions - Multi-layer/composite material domains

Key classes: - Domain: Represents the computational domain with mesh, material model, and BCs - Mesh: 1-D finite element mesh with elements and nodes - Element: Abstract base class for finite elements - LinearElement: 2-node linear finite element - QuadraticElement: 3-node quadratic finite element

Typical usage: 1. Create Mesh with nodes and elements 2. Define material model (isothermal_model or piecewise_linear_model) 3. Create Domain with mesh, material, initial temperature, and BCs 4. Solve using solve_ht() with time step and end time

The solver uses the theta-method for time integration: - theta = 0: Forward Euler (explicit, conditionally stable) - theta = 0.5: Crank-Nicolson (implicit, unconditionally stable, 2nd order) - theta = 1: Backward Euler (implicit, unconditionally stable, 1st order)

hyddown.thermesh.**solve_ht**(*domain*, *solver*)

Solves heat transfer problem.

Parameters

- **domain** (Domain) – Domain with a mesh, material model and boundary conditions.
- **solver** (*dict*) – Solver information, dictionary with the following keys: dt, t_end, theta

Return type

tuple[ndarray[tuple[Any, ...], dtype[float64]], ndarray[tuple[Any, ...], dtype[float64]]]

Returns

- **t** (*nd.array(dtype=float, dim=1, len=int(t_end/dt))*) – Times.
- **T** (*nd.array(dtype=float, dim=2, shape=(int(t_end/dt), mesh.nn))*) – Temperature data.

class hyddown.thermesh.**Domain**(*mesh*, *constitutive_model*, *bc={}*)

Bases: object

Class to represent a domain.

mesh

Object with mesh information.

Type
Mesh

bc

The boundary conditions are provided in a two-item list of dictionaries. The first dictionary (or zeroth item in the list) applies to the start or left side of the domain, while the second item applies to the end or right side of the domain. The dictionaries can have the keys: "T" OR (("h" and "T_inf") AND/OR "q"), with "T" an applied temperature, "h" and "T_inf" the convective heat transfer coefficient and far field temperature, respectively, while "q" represents a direct flux on the surface which is directed inwards.

Type
two-item list of dicts

constitutive_model

Functions that takes temperature as an input and provides a dictionary with the following keys: k, cp, rho. The list has a length equal to the number of subdomains in the mesh. The i-th function in the list belongs to the i-th subdomain.

Type
list[function]

t

Time.

Type
float

T

Temperature at time t.

Type
nd.array(dim=1, len=mesh.nn)

q

Heat flux at time t.

Type
nd.array(dim=1, len=mesh.nn)

timestep(*dt*, *theta=0.5*)

Apply a timestep dt and update data.

system_matrices()

Returns domain stiffness and dampling matrix.

check_bc()
Checks if bc's are valid.

set_T(*T) and set_q(q*)
Set temperature and heat flux.

clear()
Clears data.

__init__(*mesh*, *constitutive_model*, *bc={}*)

timestep(*dt*, *theta=0.5*)
Apply timestep and update data.

Parameters

- **dt** (*float*) – Timestep size.
- **theta** (*float (0 < theta <= 1)*) – Timestepping type.
- **NOTE** (*The internal heat source Q is not implemented yet.*) – repeatedly constructed.
- **NOTE** – problem becomes nonlinear, e.g. due to a radiative bc or due to temperature-dependent material properties.
- **NOTE**

system_matrices()
Returns domain stiffness and damping matrix.

Return type
tuple[ndarray[tuple[Any, ...], dtype[float64]], ndarray[tuple[Any, ...], dtype[float64]]]

check_bc()
Check if boundary conditions are valid.

Return type
bool

set_T(*T*)
Set temperature.

12.1 Parameter

T
[float OR np.ndarray(dim=1, dtype=float, len=nn)] Temperature at nodes.

set_q(*q*)
Set heat flux.

12.2 Parameter

q
[float OR np.ndarray(dim=1, dtype=float, len=nn)] Heat flux at nodes.

clear()
Clears time and temperature data.

hyddown.thermesh.**isothermal_model**(*k*, *rho*, *cp*)
Returns a function that represents an isothermal material model.

12.3 Parameter

k
[float] Thermal conductivity.

rho
[float] Density

cp
[float] Specific heat.

returns
model – Function that returns a dictionary with the provided constitutive properties.

rtype
Callable

hyddown.thermesh.**piecewise_linear_model**(*k*, *rho*, *cp*)
Returns a function that represents an isothermal material model.

12.4 Parameter

k
[np.ndarray(dim=2, dtype=float)] Temperature vs. thermal conductivity.

rho
[np.ndarray(dim=2, dtype=float)] Temperature vs. density

cp
[np.ndarray(dim=2, dtype=float)] Temperature vs. specific heat.

returns
model – Function that returns a dictionary with the provided constitutive properties.

rtype
Callable

class hyddown.thermesh.**Mesh**(*z*, *element*)

Bases: object

Class to represent a mesh.

nodes

Node locations.

Type
nd.array()

elem

List of elements.

Type
list[*Element*]

nn

Number of nodes.

Type
int

nel

Number of elements.

Type
int

subdomain

Number that indicates the subdomain in the mesh. Correlates with the constitutive model that will be used for the analysis.

Type
list[int] (defaults to a list with zeros)

__init__(*z*, *element*)

Initializes Mesh instance.

Parameters

- **z** (*np.ndarray(dim=1, dtype=float)*) – Node locations.
- **element** (Element) – Element type.

class hyddown.thermesh.**Element**(*nodes*)

Bases: object

Class to represent an element.

order

Order of the element.

Type
int

dim

Dimension of the element.

Type
int

dim = 1

order = 0

__init__(*nodes*)

Initializes Element instance.

Parameters
nodes (*np.ndarray(dim=1, dtype=float)*) – Node locations.

length()

Return type
float

class hyddown.thermesh.**LinearElement**(*nodes*)

Bases: *Element*

Class to represent a linear element (order = 1).

K()

Returns element stiffness matrix.

C()

Returns element damping matrix.

order = 1

K(*mat*)

Returns element stiffness matrix.

Return type
ndarray[tuple[Any, ...], dtype[float64]]

C(*mat*)

Returns element damping matrix.

class hyddown.thermesh.**QuadraticElement**(*nodes*)

Bases: *Element*

Class to represent a linear element (order = 2).

K()

Returns element stiffness matrix.

C()

Returns element damping matrix.

order = 2

K(*mat*)

Returns element stiffness matrix.

Return type
ndarray[tuple[Any, ...], dtype[float64]]

C(*mat*)

Returns element damping matrix.

Return type
ndarray[tuple[Any, ...], dtype[float64]]

12.5 Overview

The module is adapted from https://github.com/wjbg/thermesh and implements finite element analysis for transient heat conduction through vessel walls.

12.6 Key Features

- Finite element method for accurate temperature distribution
- Support for composite materials (multi-layer walls)
- Time-dependent boundary conditions
- Integration with fire heat load calculations
- Separate models for wetted/unwetted regions in two-phase flow

12.7 Applications

The detailed heat conduction model is essential for:

- Type III/IV composite pressure vessels
- Materials with low thermal conductivity
- Accurate wall temperature prediction during fire scenarios
- Two-phase systems with different heat transfer in gas/liquid contact regions

12.8 Wall Heat Conduction Modes

HydDown supports two modes:

1. **Simple**: Uniform wall temperature (lumped capacitance) - fast but approximate
2. **Detailed**: 1-D transient conduction via `thermesh` - slower but accurate

The detailed mode is activated by setting appropriate parameters in the `heat_transfer` section of the input file.

12.9 Composite Materials

For multi-layer walls (e.g., Type III vessels with liner, composite overwrap, and outer layer):

1. Define material properties for each layer
2. Specify layer thicknesses
3. The solver handles interfaces automatically

12.10 Two-Phase Modeling

For two-phase systems, the module solves separate heat conduction problems for:

- Wetted region (liquid-contact) - typically higher heat transfer
- Unwetted region (gas-contact) - typically lower heat transfer

The boundary between regions is determined by the liquid level in the vessel.

CHAPTER

THIRTEEN

VALIDATOR - INPUT VALIDATION

The `validator` module provides input validation using the Cerberus library. It ensures input files are correctly structured with required parameters. Input validation for HydDown YAML configuration files.

This module uses the Cerberus validation library to enforce schema-based validation of user-provided input dictionaries parsed from YAML files. It ensures that all required parameters are present, values are of correct types, and fall within acceptable ranges before calculations begin.

The validation is hierarchical and mode-dependent: - Mandatory ruleset: vessel geometry, initial conditions, calculation setup - Heat transfer validation: depends on calculation type (energybalance, etc.) - Valve validation: depends on valve type (orifice, control valve, relief valve)

Validation errors are reported with descriptive messages to help users correct their input files. The module prevents invalid calculations that would lead to runtime errors or physically meaningless results.

Main functions: - validation(): Top-level validation function called by HydDown class - validate_mandatory_ruleset(): Validates core required parameters - heat_transfer_validation(): Validates heat transfer settings - valve_validation(): Validates valve parameters based on valve type

hyddown.validator.**validate_mandatory_ruleset**(*input*)

Validate input

Parameters
input (*dict*) – Structure holding input

Returns
retval – True for success, False for failure

Return type
bool

hyddown.validator.**heat_transfer_validation**(*input*)

Validate input['heat_transfer'] deeper than cerberus

Parameters
input (*dict*) – Structure holding input

Returns
: bool True for success, False for failure

hyddown.validator.**valve_validation**(*input*)

Validate input['valve'] deeper than cerberus

Parameters
input (*dict*) – Structure holding input

Returns
: bool True for success, False for failure

hyddown.validator.**validation**(*input*)

Aggregate validation using cerberus

Parameters
input (*dict*) – Structure holding input

Returns
: bool True for success, False for failure

CHAPTER

FOURTEEN

MATERIALS - MATERIAL PROPERTIES

The `materials` module provides temperature-dependent material properties for pressure vessel calculations, primarily focused on structural steels used in fire scenarios. Material property database for pressure vessel calculations.

This module provides temperature-dependent material properties for common pressure vessel materials including: - Stainless steels (SS316, Duplex, Super Duplex) - Carbon steels (low temperature grade) - Vessel wall materials (steel, aluminum, composites)

Properties available: - Heat capacity (Cp) as function of temperature [J/(kg·K)] - Ultimate tensile strength (UTS) as function of temperature [MPa] - Allowable tensile stress (ATS) as function of temperature [MPa] - von Mises equivalent stress calculations

Data sources: - Scandpower Risk Management AS guidelines for fire scenarios - EN standards for structural steel properties - Literature values for composite materials

All temperature data are stored in Kelvin [K]. Property values are interpolated using numpy.interp() for intermediate temperatures.

hyddown.materials.**von_mises**(*p*, *d*, *wt*, *sigma_a=30000000.0*)

von Mises stress calculated according to: Hekkelstrand, B.; Skulstad, P. Guidelines for the Protection of Pressurised Systems Exposed to Fire; Scandpower Risk Management AS: Kjeller, Norway, 2004.

As also applied in: Andreasen, A.; Borroni, F.; Zan Nieto, M.; Stegelmann, C.; P. Nielsen, R. On the Adequacy of API 521 Relief-Valve Sizing Method for Gas-Filled Pressure Vessels Exposed to Fire. Safety 2018, 4, 11. https://doi.org/10.3390/safety4010011

Parameters

- **p** (*float*) – Pressure (Pa)
- **d** (*float*) – Inner diameter (m)
- **D** (*float*) – Outer diameter (m)
- **wt** (*float*) – Wall thickness (m)
- **sigma_a** (*float*) – Default

Returns

sigma_e – von Mises stress (Pa)

Return type

float

hyddown.materials.**ATS**(*temperature*, *material*, *k_s=0.85*, *k_y=1*)

Calculation of Allowable Tensile Strength according to: Hekkelstrand, B.; Skulstad, P. Guidelines for the Protection of Pressurised Systems Exposed to Fire; Scandpower Risk Management AS: Kjeller, Norway, 2004.

Parameters

- **temperature** (*float*) – Temperature (K)
- **material** (*string*) – Material type: 235LT, 360LT (ASTM A-333/A-671), 2205 (SA-790/ASTM A-790), 316 (ASTM A-320, ASME A-358), 6Mo (ASTM B-677)
- **k_s** (*float*) – General safety factor. For typical materials 0.85 is used. If "guaranteed" minimum values a factor 1.0 can be used.
- **k_y** (*float*) – Additional factor used for materials with missing or uncertain material data. Normally 1.0.

Returns
ATS – Allowable Tensile Strength (Pa)

Return type
float

hyddown.materials.**UTS**(*temperature*, *material*)

Tabulation look-up / interpolation to retrieve the Ultimate Tensile Strength as a function of temperature for various typical materials according to:

Hekkelstrand, B.; Skulstad, P. Guidelines for the Protection of Pressurised Systems Exposed to Fire; Scandpower Risk Management AS: Kjeller, Norway, 2004.

Parameters

- **temperature** (*float*) – Temperature (K)
- **material** (*string*) – Material type: 235LT, 360LT (ASTM A-333/A-671), Duplex (2205, SA-790/ASTM A-790), 316 (ASTM A-320, ASME A-358), 6Mo (ASTM B-677)

Returns
UTS – Ultimate Tensile Strength (Pa)

Return type
float

hyddown.materials.**steel_Cp**(*temperature*, *material*)

Tabulation look-up / interpolation to retrieve the heat capacity as a function of temperature for various typical materials according to:

Hekkelstrand, B.; Skulstad, P. Guidelines for the Protection of Pressurised Systems Exposed to Fire; Scandpower Risk Management AS: Kjeller, Norway, 2004.

Parameters

- **temperature** (*float*) – Temperature (K)
- **material** (*string*) – Material type: 235LT, 360LT (ASTM A-333/A-671), Duplex (2205, SA-790/ASTM A-790), 316 (ASTM A-320, ASME A-358), 6Mo (ASTM B-677)

Returns
Cp – Ultimate Tensile Strength (Pa)

Return type
float

CHAPTER

FIFTEEN

CITING HYDDOWN

If you use HydDown in your research, please cite the following references:

15.1 Primary Citation

Andreasen, A., (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

BibTeX:

```
@article{Andreasen2021,
  doi = {10.21105/joss.03695},
  url = {https://doi.org/10.21105/joss.03695},
  year = {2021},
  publisher = {The Open Journal},
  volume = {6},
  number = {66},
  pages = {3695},
  author = {Anders Andreasen},
  title = {HydDown: A Python package for calculation of hydrogen (or other gas)
           pressure vessel filling and discharge},
  journal = {Journal of Open Source Software}
}
```

15.2 Manual Citation

Anders Andreasen. HydDown - User guide and technical reference. 2024. ⟨hal-04858235⟩

BibTeX:

```
@report{Andreasen2024,
  url = {https://hal.science/hal-04858235},
  year = {2024},
  publisher = {HAL open science},
  author = {Anders Andreasen},
  title = {HydDown -- user guide and technical reference},
}
```

15.3 Related Publications

For multi-component two-phase behavior with partial/non-equilibrium assumptions, see the HydDown sibling project openthermo:

Andreasen, A., & Stegelmann, C. (2025). Open source pressure vessel blowdown modeling under partial phase equilibrium. Process Safety Progress. https://doi.org/10.1002/prs.70035

BibTeX:

```
@article{Andreasen2025,
  author = {Andreasen, Anders and Stegelmann, Carsten},
  title = {Open source pressure vessel blowdown modeling under partial phase equilibrium}
,
  journal = {Process Safety Progress},
  doi = {10.1002/prs.70035},
  year = {2025}
}
```

A preprint is also available.

CHAPTER

SIXTEEN

CHANGELOG

This page tracks major changes and improvements to HydDown.

16.1 Latest Version

New Features:

- 1-D transient heat conduction for vessel walls (for Type III/IV vessels)
- Stefan-Boltzmann fire heat load modeling
- Two-phase single component modeling with separate gas/liquid heat transfer
- Detailed heat transfer with wall conduction model
- Support for composite material vessels

Improvements:

- Enhanced validation framework for comparing with experimental data
- Improved documentation and examples
- Better error handling and input validation

16.2 Previous Versions

See the GitHub releases page for detailed version history.

CHAPTER

SEVENTEEN

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[AMdeMiguel+14] B. Acosta, P. Moretto, N. de Miguel, R. Ortiz, F. Harskamp, and C. Bonato. Jrc reference data from experiments of on-board hydrogen tanks fast filling. *International Journal of Hydrogen Energy*, 39(35):20531–20537, 2014. doi:10.1016/j.ijhydene.2014.03.227.

[And21] Anders Andreasen. Hyddown: a python package for calculation of hydrogen (or other gas) pressure vessel filling and discharge. *Journal of Open Source Software*, 6(66):3695, 2021. URL: https://doi.org/10.21105/joss.03695, doi:10.21105/joss.03695.

[And24] Anders Andreasen. Hyddown – user guide and technical reference. 2024. URL: https://hal.science/hal-04858235v1.

[ABZN+18] Anders Andreasen, Filippo Borroni, Marcos Zan Nieto, Carsten Stegelmann, and Rudi P. Nielsen. On the adequacy of api 521 relief-valve sizing method for gas-filled pressure vessels exposed to fire. *Safety*, 2018. doi:10.3390/safety4010011.

[AS25] Anders Andreasen and Carsten Stegelmann. Open source pressure vessel blowdown modeling under partial phase equilibrium. *Process Safety Progress*, 2025. doi:10.1002/prs.70035.

[API14a] API. *API 520 Sizing, Selection, and Installation of Pressure-relieving Devices*. American Petroleum Institute, 2014.

[Bel25] Caleb Bell. Fluids: fluid dynamics component of chemical engineering design library (chedl). https://github.com/CalebBell/fluids, 2025.

[BLVikramGovindarajanr24] Caleb Bell, Pierre Lesouhaitier, VikramGovindarajan, and jeremy rutman. CalebBell/ht: 1.0.8. December 2024. URL: https://doi.org/10.5281/zenodo.14321704, doi:10.5281/zenodo.14321704.

[BWQL14] Ian H. Bell, Jorrit Wronski, Sylvain Quoilin, and Vincent Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53(6):2498–2508, 2014. URL: http://pubs.acs.org/doi/abs/10.1021/ie4033999, arXiv:http://pubs.acs.org/doi/pdf/10.1021/ie4033999, doi:10.1021/ie4033999.

[BEA+17] Michael Bjerre, Jacob G. I. Eriksen, Anders Andreasen, Carsten Stegelmann, and Matthias Mandø. Analysis of pressure safety valves for fire protection on offshore oil and gas installations. *Process Safety and Environmental Protection*, 105:60–68, 2017. doi:10.1016/j.psep.2016.10.008.

[Bor98] Guy Borden, editor. *Control Valves – Practical Guides for Measurement and Control*. Instrument Society of America, 1998.

[BRR64] W. R. Byrnes, R. C. Reid, and F. E. Ruccia. Rapid depressurization of gas storage cylinder. *Industrial & Engineering Chemistry Process Design and Development*, 3(3):206–209, 1964. doi:10.1021/i260011a004.

[DMM17] M. Dadashzadeh, D. Makarov, and V. Molkov. Non-adiabatic blowdown model: a complimentary tool for the safety design of tank-TPRD system. In *Proceedings of the Internationsl Conference On Hydrogen Safety*, 1–18. Hamburg, Germany, September 2017. International Association for Hydrogen Safety. URL: https://hysafe.info/uploads/papers/2017/186.pdf.

[DM07] C. J. B. Dicken and W. Mérida. Modeling the transient temperature distribution within a hydrogen cylinder during refueling. *Numerical Heat Transfer, Part A: Applications*, 53(7):685–708, 2007. doi:10.1080/10407780701634383.

[EB15] Jacob Gram Iskov Eriksen and Michael Skov Bjerre. Analysis of the application and sizing of pressure safety valves for fire protection on offshore oil and gas installations. Master's thesis, Aalborg University, 2015. URL: 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.

[Gea93] C.J. Geankoplis. *Transport Processes and Unit Operations*. Chemical Engineering. PTR Prentice Hall, 1993.

[Gro] Walter Grouve. Thermesh – finite element code for transient one-dimensional heat conduction in python. https://github.com/wjbg/thermesh. Accessed: 2024-12-25.

[HR24] C. Hall and V. Ramasamy. Modelling the conjugate heat transfer during the fast-filling of high-pressure hydrogen vessels for vehicular transport. *International Journal of Thermofluids*, 21:100527, 2024. doi:10.1016/j.ijft.2023.100527.

[HRS+92] M. A. Haque, M. Richardson, G. Saville, G. Chamberlain, and L. Shirvill. Blowdown of pressure vessels. Pt. II: Experimental validation of computer model and case studie. *Trans. IChemE B*, 70:10–17, 1992.

[HS04] B. Hekkelstrand and P. Skulstad. *Guidelines for the Protection of Pressurised Systems Exposed to Fire*. Scandpower Risk Management AS, 2004.

[IEC11] IEC. Iec 60534-2-1 industrial-process control valves – part 2-1: flow capacity – sizing equations for fluid flow under installed conditions. Technical Report, IEC, 2011.

[ISA95] ISA. *Flow Equations for Sizing Control Valves, ANSI/ISA S75.01 Standard, ISA-S75.01-1985 (R 1995)*. Instrument Society of America, 1995.

[JSRB04] J. M. Saiz Jabardo, E. Fockink da Silva, G. Ribatski, and S. F. de Barros. 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*, 26:218–230, 2004. doi:10.1590/S1678-58782004000200015.

[KNP+21] Taichi Kuroki, Kazunori Nagasawa, Michael Peters, Daniel Leighton, Jennifer Kurtz, Naoya Sakoda, Masanori Monde, and Yasuyuki Takata. Thermodynamic modeling of hydrogen fueling process from high-pressure storage tank to vehicle tank. *International Journal of Hydrogen Energy*, 46(42):22004–22017, 2021. doi:10.1016/j.ijhydene.2021.04.037.

[MBAI+17] D. Melideo, D. Baraldi, B. Acosta-Iborra, R. Ortiz Cebolla, and P. Moretto. Cfd simulations of filling and emptying of hydrogen tanks. *International Journal of Hydrogen Energy*, 42(11):7304–7313, 2017. doi:10.1016/j.ijhydene.2016.05.262.

[MDKM21] V. Molkov, M. Dadashzadeh, S. Kashkarov, and D. Makarov. Performance of hydrogen storage tank with tprd in an engulfing fire. *International Journal of Hydrogen Energy*, 46(73):36581–36597, 2021. doi:10.1016/j.ijhydene.2021.08.128.

[MCD+88] K. Moodie, L.T. Cowley, R.B. Denny, L.M. Small, and I. Williams. Fire engulfment tests on a 5 tonne lpg tank. *Journal of Hazardous Materials*, 20:55–71, 1988. doi:10.1016/0304-3894(88)87006-7.

[Pio99] I. l Pioro. Experimental evaluation of constants for the Rohsenow pool boiling correlation. *International Journal of Heat and Mass Transfer*, 42(11):2003–2013, 1999. doi:10.1016/S0017-9310(98)00294-4.

[Roh51] Warren M. Rohsenow. A method of correlating heat transfer data for surface boiling of liquids. Technical Report, Cambridge, Mass. : M.I.T. Division of Industrial Cooporation, [1951], 1951. URL: https://dspace.mit.edu/handle/1721.1/61431 (visited on 2025-03-27).

[SVNA96] J. M. Smith, H. C. Van Ness, and M. M. Abbott. *Introduction to Chemical Engineering Thermodynamcis*. McGraw-Hill, fifth edition, 1996.

[SBSK14] Michael Striednig, Stefan Brandstätter, Markus Sartory, and Manfred Klell. Thermodynamic real gas analysis of a tank filling process. *International Journal of Hydrogen Energy*, 39(16):8495–8509, 2014. doi:https://doi.org/10.1016/j.ijhydene.2014.03.028.

[vdBW05] C.J.H. van den Bosch and R.A.P.M. Weterings, editors. *Methods for the calculation of physical effects (Yellow Book) – CPR 14E*. Committee for the Prevention of Disasters, 3 edition, 2005.

[Wik08] WikiMedia. First law open system diagram. 2008. Adapted from DOE-HDBK-1012/1-92, Thermodynamics, Heat Transfer, and Fluid Flow, Volume 1 of 3, 2014. U.S. Department of Energy. URL: https://commons.wikimedia.org/wiki/File:First_law_open_system.svg.

[Won98] Shan Meng Angela Wong. *DEVELOPMENT OF A MATHEMATICAL MODEL FOR BLOWDOWN OF VESSELS CONTAINING MULTICOMPONENT HYDROCARBON MIXTURES*. PhD thesis, University College London, 1998. URL: https://discovery.ucl.ac.uk/id/eprint/1317912/1/299471.pdf.

[WMM07] Peter Lloyd Woodfield, Masanori Monde, and Yuichi Mitsutake. 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):180–191, 2007. doi:10.1299/jtst.2.180.

[API14b] API. Pressure-relieving and Depressuring Systems, API Standard 521. Technical Report API Standard 521, Sixth Edition, January, American Petroleum Institute, 2014.

[deMiguelAMC15] N. de Miguel, B. Acosta, P. Moretto, and R. Ortiz Cebolla. The effect of defueling rate on the temperature evolution of on-board hydrogen tanks. *International Journal of Hydrogen Energy*, 40(42):14768–14774, 2015. doi:10.1016/j.ijhydene.2015.06.038.

[RuizMaraggiM23] Leopoldo M. Ruiz Maraggi and Lorena G. Moscardelli. Modeling hydrogen storage capacities, injection and withdrawal cycles in salt caverns: introducing the geoh2 salt storage and cycling app. *International Journal of Hydrogen Energy*, 48(69):26921–26936, 2023. doi:10.1016/j.ijhydene.2023.03.293.

PYTHON MODULE INDEX

h