MODELING AND ANALYSIS OF CARNOT BATTERIES AND THERMODYNAMIC CYCLES IN JULIA

Sushrut Deshpande[1]*, Maarten Vergote[1] , Bart Janssen[1] , Arthur Adriaens[1] , Vincent Lemort[2] , Aitor Cendoya[2]

[1]Royal Military Academy, Brussels, Belgium
[2]University of Liège, Liège

*Corresponding Author: sushrut.deshpande@mil.be

ABSTRACT

The continuous growing demand of energy has led to the necessity of new technology for energy storage. Based on the current climate change dialogue the use of sustainable materials for energy storage is of paramount importance. One possible approach to solving this problem is the conception of Carnot Battery more specifically Pumped Thermal Energy Storage (PTES) system. The working cycle of a PTES system is divided in three phases - Charging Phase, Storage Phase, and Discharging Phase. In the Charging Phase excess generated electricity is stored in the thermocline in form of heat with the help of a heat pump. In the storage phase the heat is stored in the thermocline for future use. And finally in the Discharge Phase this heat is converted back to electricity by means of a heat engine.

The detailed study of these processes requires robust numerical and modelling tools. Such simulation codes need component based modelling and the softwares that deal with such problems are not generally open source. Hence the goal of this study is the development of a robust simulation software in Julia for the modelling of Thermodynamic Cycles. The simulation software is able to dynamic simulation specifically ODE solvers using ModelingToolKit.jl. It has modularity to include multiple models of components used in the cycles (eg: thermocline. compressor, expander, condenser, heat exchangers etc ...) as well as flexibility to implement additional models with minimal change to existing code. It also allows for the study of various power cycles as well as different Energy systems configurations.

The choice of Julia as a programming language is motivated by its high-level interface with performance of a low-level language. It has a package based framework and for this study a package is developed. This package will be registered in the Julia Library in the future. It is based on ModelingToolkit.jl. In later stages the simulation results of the software shall be compared with existing experimental results for specific configurations. This will allow for the fine tuning of parameters and models in the simulation code to capture realistic results. Finally, the described simulation software can be used by engineers for modeling Thermodynamic and Energy systems

1 INTRODUCTION

Modeling and simulation frameworks are necessary as they reduce the load of extensive experimentation for any given research problem at hand. Energy and thermodynamic systems are no exception to this idea. Hence there exist a few frameworks that allow the modelling of energy and thermodynamic systems. A few notable of them are Modelica, MATLAB - Simulink, CyclePad, Dynamola, etc.
This paper proposes a simulation framework in Julia based on ModelingToolKit (Y. Ma et al, 2021). It allows for the custom implementation of components and enables connection between components. This aspect is what makes it suitable for *acasual* modeling of any system. Here this principle is

extended to thermodynamic cycles and energy systems. Here a Julia package is proposed – "MTKThermoCycles.jl"
The objectives of the proposed package are:

1. Ability to simulate thermodynamic cycles.
2. Dynamic modeling of thermodynamic systems – specifically ODE systems.
3. Ease of execution and custom modeling at the user's end.

2 MODELING OF COMPNENTS

The modeling of thermodynamic cycles is based around ModelingToolkit.jl– a versatile Julia package for modeling various systems. The presented work aims to build a library around the framework provided by the Modeling Toolkit. It provides multiple components that are generally used in a thermodynamic cycle. Every component has at least one node with the variables being (h, p, ṁ). These nodes are part of every component and enable a connection between different components. Every component returns an ODE system. ModelingToolkit allows the connection of all the ODE systems of components into a large ODE system. Hence, the final stage is the connection of all the chosen components and solving the large-scale ODE system. The schematic of a generic component is given in Figure 1 and Figure 2.

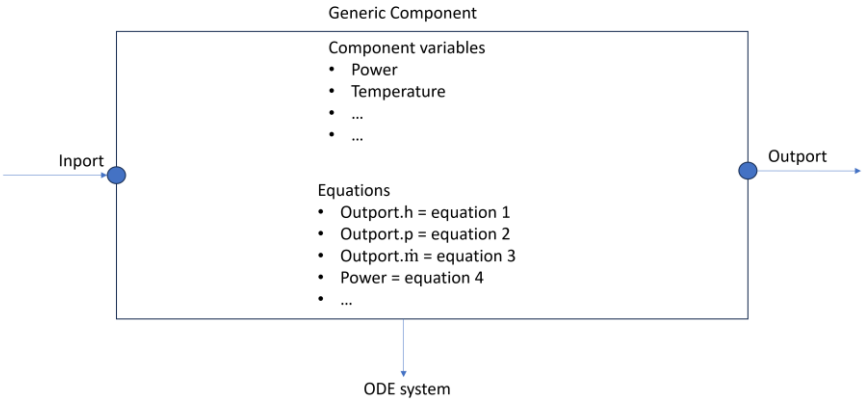


Figure 1:Generic Component Modelling

```
function SimpleCompressor(;name,πc,fluid)
    @named inport = CoolantPort()
    @named outport = CoolantPort()
    para = @parameters begin

    end
    vars = @variables begin
        P(t)
     end
   eqs = [  outport.mdot ~ abs(inport.mdot)
            outport.p ~ πc * inport.p
            outport.h ~ IsentropicCompression(πc, inport.h, inport.p,fluid)
            P ~ abs(inport.mdot)*(outport.h - inport.h)
   ]
   compose(ODESystem(eqs, t, vars, para;name), inport, outport)
end
```

Figure 2: Component Modelling Example Code - Isentropic Compressor

The large-scale ODE system can be solved using the ODE solvers provided by DifferentialEquations.jl (Rackauckas, C. and Nie, Q., 2017) a Julia library. It provides various methods to solve ODEs in Julia.

The thermodynamic properties are computed through the well-known library – CoolProp (I.H. Bell et al, 2014). The version used for this library is the extension of CoolProp to the Julia interface – CoolProp.jl.
The following sub-sections give a basic equations used to model individual components:

2.1 Compressor/Expander Models

There are two types of models implemented for compressors and expanders:

1. Isentropic Models
2. Isentropic Efficiency Models

The details of the implemented EoS are given in the following subsections.

2.1.1 Isentropic Models

This is achieved by changing the outport node's pressure variable to the desired pressure ratio and finding the appropriate enthalpy value for the outport node. It is assumed that there is no change in mass flow rate. The compressor and expander power is calculated as follows:

$$\mathrm{P} = \dot{m}(h_{\mathrm{in}} - h_{\mathrm{out}}) \tag{1}$$

2.1.2 Isentropic Efficiency Models

This model was proposed for sCO_2 cycles (Olumayegun, O., & Wang, M. 2019). It takes into account the isentropic efficiency of the components.

For compressor:

$$h_{out} = h_{in} + \frac{h_{out,isen} - h_{in}}{\eta_{\mathrm{c}}} \tag{2}$$

And for expanders

$$h_{out} = h_{in} - \eta_{\mathrm{e}}\left(h_{in} - h_{out,isen}\right) \tag{3}$$

2.2 Thermal Store Model

The modeling of the thermal store is through a differential equation that governs energy balance. Moreover, it attempts to model the location of the thermal front. The following assumptions are made:

1. The thermocline is long enough so that the output temperature of the working fluid is the initial temperature of the store.
2. The stored solid material equals the inlet temperature of the fluid while the fluid temperature drops to the initial temperature of the store medium.
3. The temperature gradient inside the thermal store has a discontinuity at the location of the thermal front.
4. There are no thermal losses during the heat transfer process. And there is no drop in pressure inside the thermocline.

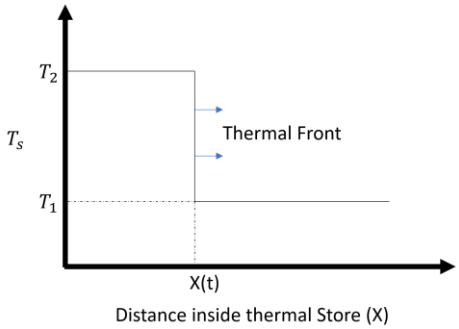


Figure 3:Thermal Store model schematic.

The energy balance equations for this are described below.

$$\dot{Q} = \dot{m}(h_{out} - h_{in}) \tag{4}$$

Equation (4) describes the rate of heat lost by the fluid. This is the heat gained by the thermal store. Hence,

$$\frac{\partial m_s C_{p,s} \Delta T}{\partial t} = -\dot{Q} \tag{5}$$

$$m_s = A\rho_s X(t) \tag{6}$$

ΔT is constant according to the second assumption. Hence the ODE is solved for $X(t)$ using (5) and (6)

2.3 Boiler/Heater – Condenser/Cooler Model

The model for all the Boiler/Heater – Condenser/Cooler is forward with desired temperatures as input parameters. The process is considered to be isobaric hence the new outport enthalpy is computed. The only difference between Boiler/Heater is the check for phase change of the fluid.

2.4 Electric Heater Model

This is an isobaric heater model. The parameters are power rating and efficiency of heating. With the outport enthalpy is calculated using equation (7).

$$h_{out} = h_{in} + \frac{\eta P}{\dot{m}} \tag{7}$$

3 TEST CASES

In this section two simple test cases will be presented with the following objectives:

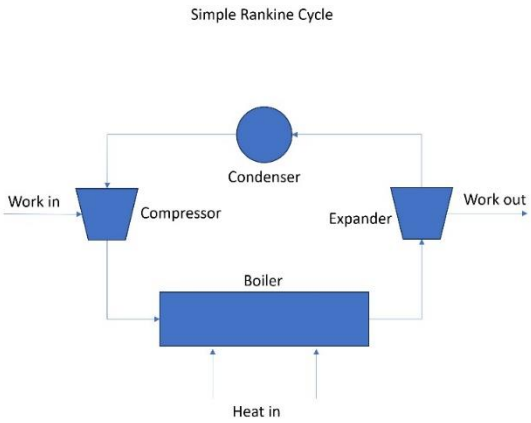
1. Method of Modeling of Thermodynamic Cycles
1. Ease of modeling for users
2. Basic analysis of the presented cycles

3.1 Organic Rankine Cycle

The test case simulated here is a generic organic Rankine cycle using the refrigerant R134a. The components are chosen as follows:

1. Compressor & Expander: Isentropic and has a fixed pressure ratio of 10 for both compressor and expander.
2. The boiler and condenser are assumed to be Isobaric processes where the outport temperature is passed as a parameter. For the boiler, the outport temperature needs to be chosen such that the fluid exists in the gaseous phase. The outport temperature of the condenser is chosen to be the source temperature.

Figure 3 shows the simple cycle schematic and the adjoining code to simulate it.




```
using MTKDecoupledPTES,ModelingToolkit, CoolProp, DifferentialEquations,Plots
@variables t
D = Differential(t)
fluid = "R134a"
πc = 10
source_temp = 170
source_pressure = 101325
source_enthalpy = PropsSI("H","T",source_temp,"P",source_pressure,fluid)
@named source = CycleSource(source_pressure = source_pressure, source_enthalpy = source_enthalpy)
@named compressor = SimpleCompressor(πc = πc, fluid =fluid)
@named boiler = IsobaricHeater(T_out = 350,fluid=fluid)
@named expander = SimpleExpander(πc = πc,fluid =fluid)
@named condensor = IsobaricCondensor(T_out = source_temp,fluid=fluid)

eqs = [
    connect(source.outport,compressor.inport)
    connect(compressor.outport,boiler.inport)
    connect(boiler.outport,expander.inport)
    connect(expander.outport,condensor.inport)
    connect(condensor.outport,source.inport)
]

@named SimpleRankineCycle = ODESystem(eqs, t, systems=[source,compressor,boiler,expander,condensor])
u0 = []
tspan = (0.0, 1.0)
sys = structural_simplify(SimpleRankineCycle)
prob = ODEProblem(sys,u0,tspan)
sol = solve(prob,saveat=1.0)
```

Figure 4: Cycle schematic and adjoining simulation code

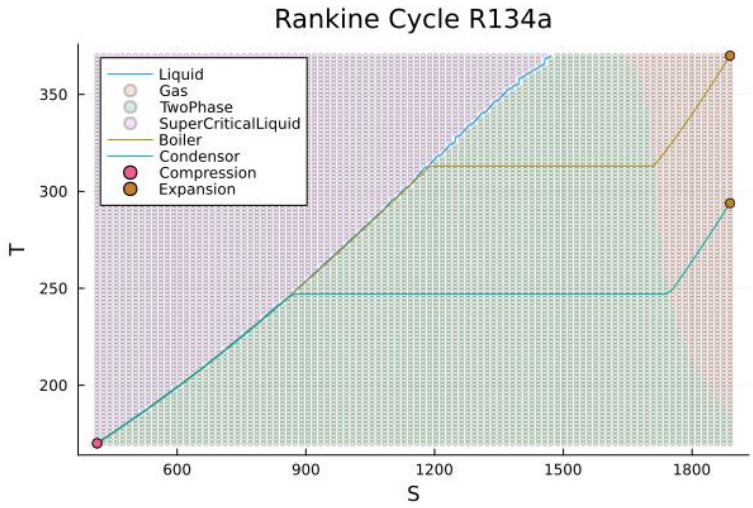


Figure 5: Phase Diagram for Rankine cycle

For computation of efficiency of the cycle generic thermodynamic equations is used.

$$\eta = \frac{\Delta h_{\text{expander}} - \Delta h_{\text{compressor}}}{Q_{\text{in,boiler}}} \quad (8)$$

In Figure 6: Efficiency vs Pressure ratio for Rankine cycle. Choice of Various boiler outlet temperatures the efficiency of the Rankine cycle is computed over pressure ratio's (πc) for different boiler temperatures.

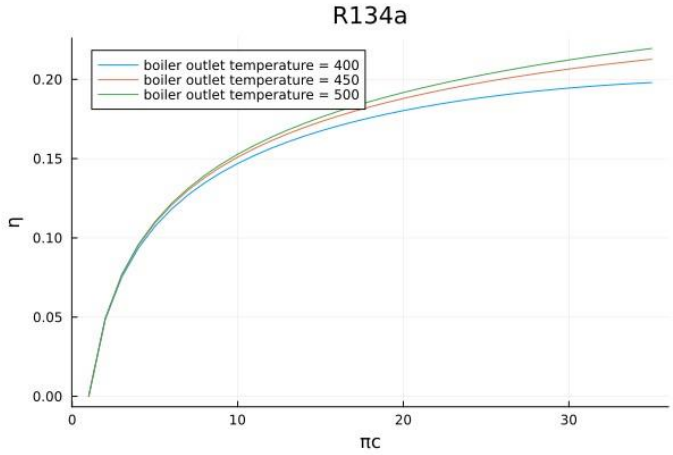


Figure 6: Efficiency vs Pressure ratio for Rankine cycle. Choice of Various boiler outlet temperatures

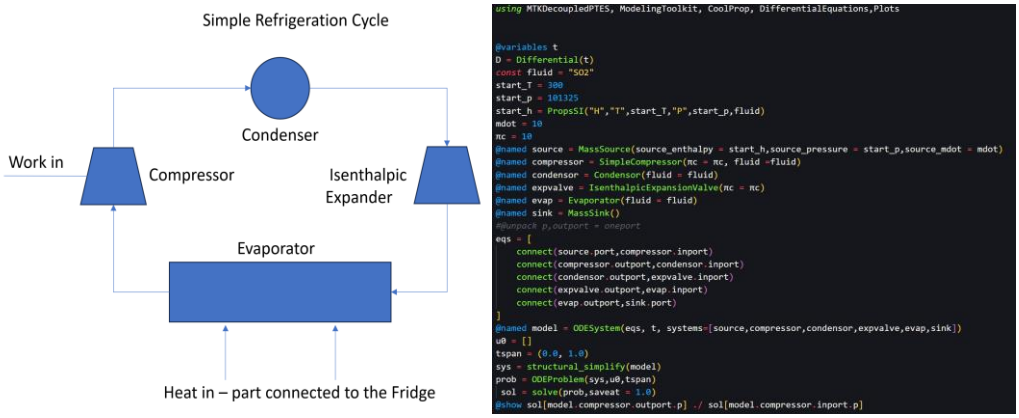
3.2 Refrigeration Cycle

The test case chosen here is a basic Refrigeration cycle using SO2 as the working fluid. The choice of components is as follows:

1. The Compressor chosen is isentropic with pressure ratio being a parameter.
2. The Expansion valve is assumed to be isenthalpic with the pressure ratio being equal to that of the compressor
3. The Evaporator and Condenser model is isobaric while fixing the vapor quality at the outport to equal 1 and 0 respectively

Figure 7: Refrigeration cycle schematic and simulation code shows the schematic which has been modeled along with the code snippet. The Figure 8: Phase Diagram for Refrigeration Cycle shows the

adequate phase diagram for the Refrigeration cycle. The fluid passes through the gaseous, two-phase, and liquid phases before coming back to the starting phase.




```
using MTKDecoupledPTES, ModelingToolkit, CoolProp, DifferentialEquations,Plots

@variables t
D = Differential(t)
const fluid = "SO2"
start_T = 300
start_p = 101325
start_h = PropsSI("H","T",start_T,"P",start_p,fluid)
mdot = 10
mc = 10
@named source = MassSource(source_enthalpy = start_h,source_pressure = start_p,source_mdot = mdot)
@named compressor = SimpleCompressor(mc = mc, fluid =fluid)
@named condensor = Condensor(fluid = fluid)
@named expvalve = IsenthalpicExpansionValve(mc = mc)
@named evap = Evaporator(fluid = fluid)
@named sink = MassSink()
#@unpack p,outport = oneport
eqs = [
    connect(source.port,compressor.inport)
    connect(compressor.outport,condensor.inport)
    connect(condensor.outport,expvalve.inport)
    connect(expvalve.outport,evap.inport)
    connect(evap.outport,sink.port)
]
@named model = ODESystem(eqs, t, systems=[source,compressor,condensor,expvalve,evap,sink])
u0 = []
tspan = (0.0, 1.0)
sys = structural_simplify(model)
prob = ODEProblem(sys,u0,tspan)
 sol = solve(prob,saveat = 1.0)
@show sol[model.compressor.outport.p] ./ sol[model.compressor.inport.p]
```

Figure 7: Refrigeration cycle schematic and simulation code

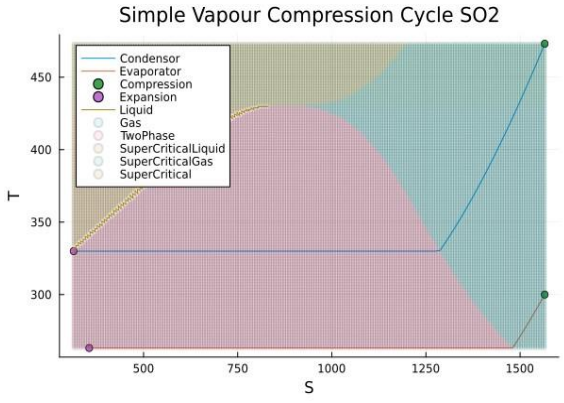


Figure 8: Phase Diagram for Refrigeration Cycle

4 CARNOT BATTERY ANALYSIS

4.1 Introduction

This section deals with the simulation of a chosen test case for a specific Carnot Battery simulation. Such a test case aims to show the basic dynamic capabilities for simulation. A generic Carnot Battery has compressors and expanders in the charging phase. However, this configuration is chosen to showcase different components for modeling.

4.2 Test Case

The test case chosen here is depicted in Figure 9. The goal of this setup is to use electric heater to heat the working fluid. This heat is transferred to the thermal store where the heat is stored. These two steps consist of the charging cycle. Then this thermal store is used as the boiler for an ORC cycle to generate electricity back from the expander/turbine.

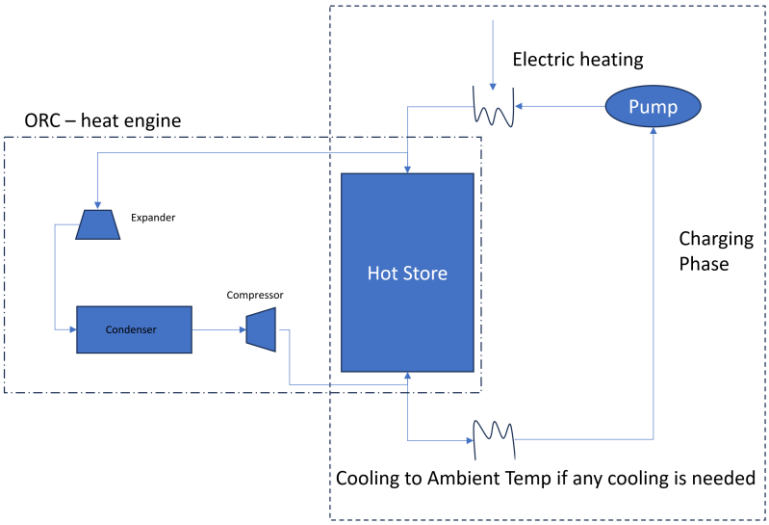


Figure 9: Carnot Battery Simulation Test Case using electric heating for charging

4.3 Simulation Results

5 THERMOCLINE ANALYSIS -SCHUMANN PACKED BEDS

This section deviates from the ODE system modeling approach used until now.
Apart from the cycle-based modeling of thermodynamic and energy systems ModelingToolkit also provides ability to solve PDE systems using an additional package – MethodOfLines.jl.
It is of particular interest for modeling thermoclines using the well know Schumann equations (Schumann, T. E). The solution to this set of equations has been solved through various approaches (Mctigue, J. 2016), (Zanganeh, G et al 2012). Most of the published results use customized code and in this section, the solver's capabilities are shown to solve the same. It allows for a unified framework to tackle such problems.

The equations considered here are:

$$\frac{\partial T_g}{\partial t} = \frac{G}{\epsilon \rho_g}\left(\frac{T_s - T_g}{l} - \frac{\partial T_g}{\partial x}\right) \tag{9}$$

$$\frac{\partial T_s}{\partial t} = \frac{T_g - T_s}{\tau} + \alpha\frac{\partial^2 T_s}{\partial x^2} - \beta(T_s - T_0) \tag{10}$$

It is assumed that there is no pressure drop over the thermocline. The boundary and initial conditions are :

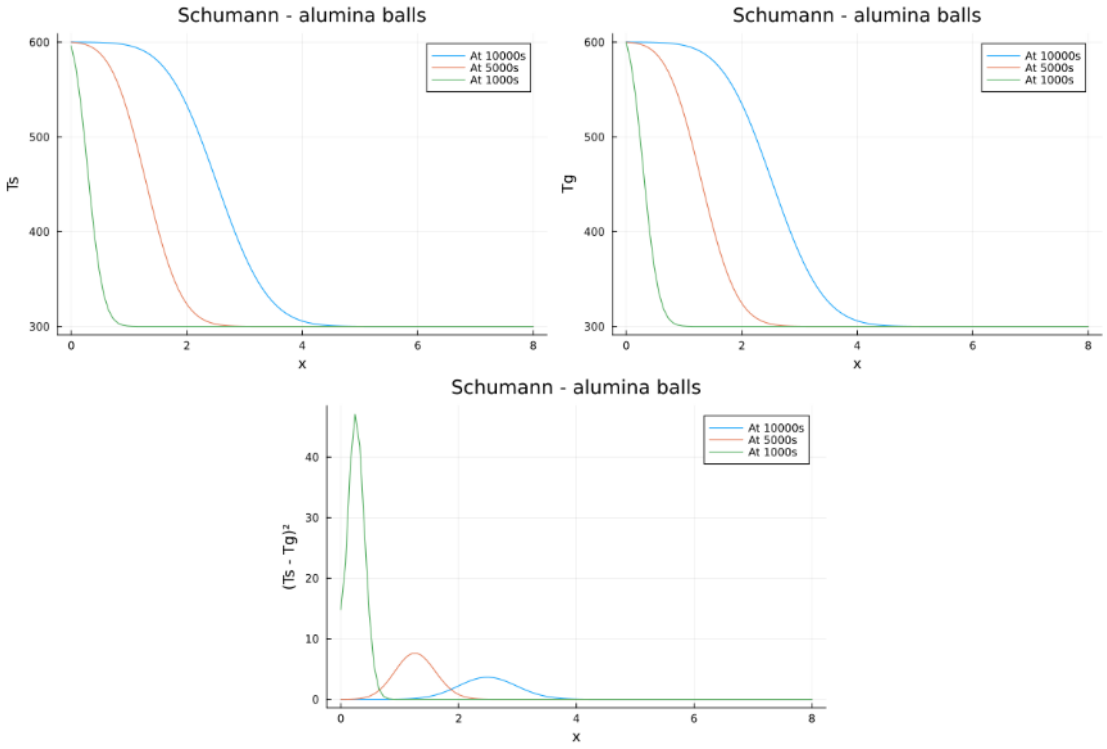
$$T_g(x,0) = T_s(x,0)$$
$$T_g(0,t) = 600$$
$$T_s(x,0) = 300$$

The test case chosen for this simulation was packed-bed aluminium balls. The gas chosen is Argon. The ρ_g and α change over temperature but for this case they are assumed constant over the duration of the simulation. The choice of parameter is given in [**Table *1*:** Choice of Parameters].

Table 1: Choice of Parameters

G	ϵ	ρ_g	l	τ	α	β
0.5	0.7	1.62	0.011	45.45	3.3×10^{-5}	0

The solution to the PDE system match the characteristic properties as published in (McTigue, J. D., & White, A. J. , 2018), (Mctigue, J. 2016), (Zanganeh, G et al 2012).



It is well noted that the thermal front travels further over time as well as the temperature difference between the two phases reduces. The solution provided here is in accordance to the solution to the dimensionless version (M. Riaz, 1977) of the problem as both cases do not consider pressure loss. Similar solutions have been found (Davenne, T. R., & Peters, B. M. , 2020), (Cascetta, M., Cau, G., Puddu, P., & Serra, F., 2014).

6 CONCLUSIONS

The simulation framework proposed here has showcased results for theoretical models of the Organic Rankine Cycle and a Refrigeration cycle. It has also shown a simple version of Carnot Battery Modelling using electric heating for the charging phase and ORC for the discharge phase. All the mentioned cycles have been analyzed in a generic thermodynamic manner.

The framework proposed here allows ease of implementation of models and integrates it with other components easily. The above-shown test cases also intend to showcase the ease of connecting components using Julia- ModelingToolkit. This indicates a desire to explore the modeling capabilities of Julia specifically ModelingToolkit.jl for thermodynamic and energy systems.

The current stage of the simulation framework is in an early phase but soon hopes to add more complex models that can assist engineering problems.

Current limitation:

1. The choice of fluids is restricted to CoolProp.jl fluids. CoolProp.jl as of now does not support mixing of fluids unlike CoolProp C++ or Python interfaces.
2. The PDE modeling is kept separate from the ODE – *acasual* modeling of cycles as the frame work for *acasual* modeling does not support PDE systems as of now.
3. The models implemented until now are 'Simple' i.e. they only concern with basic thermodynamic and component properties.

Future immediate additions:

1. Implementation of Semi-Empirical Model (Lemort 2008), (Winandy et al) which consider manufacturing data to approximate models
2. Improvement in models so as to get closer to realistic experiments.

As a whole the current framework proposed has successfully combined the thermodynamic library CoolProp and ModelingToolkit for making basic thermodynamic cycles while also show casing the dynamic simulation potential using Julia.

NOMENCLATURE

h	enthalpy	(J/kg)
p	pressure	(Pa)
$\dot{m}$	mass flow rate	(kg/s)
T	temperature	(K)
P	power	(J/s)
η	efficiency	(–)
γ	polytropic index	(–)
r_p	pressure ratio	(–)
πc	pressure ratio	(–)
C_p	Specific heat	(J/kg/K)
ODE	Ordinary Diff Eq.	(–)
PDE	Partial Diff Eq.	(–)
ϵ	packing fraction	(–)
ρ	Density	(kg/m^3)
α	Packed bed diffusivity	(m^2/s)
τ	Packed bed time scale	(s)
l	Packed bed length scale	(m)
G	Mass flow rate per unit area	(kg/m^2s)

Subscript

in	inlet
out	outlet
c	compressor
e	expander
s	solid
g	gas

REFERENCES

Cascetta, M., Cau, G., Puddu, P., & Serra, F. (2014). Numerical investigation of a packed bed thermal energy storage system with different heat transfer fluids. *Energy Procedia*, *45*, 598–607. https://doi.org/10.1016/j.egypro.2014.01.064

I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort, "Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop," Industrial & Engineering Chemistry Research, vol. 53, no. 6, pp. 2498–2508, 2014. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ie40339991

Lemort, V. (2008). CONTRIBUTION TO THE CHARACTERIZATION OF SCROLL MACHINES IN COMPRESSOR AND EXPANDER MODES.
M. Riaz, "Analytical solutions for single-and two-phase models of packed-bed thermal storage systems," J. Heat Transfer;(United States), vol. 99, no. 3, 1977
Mctigue, J. (2016). *ANALYSIS AND OPTIMISATION OF THERMAL ENERGY STORAGE*.
Rackauckas, C. and Nie, Q., 2017. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. Journal of Open Research Software, 5(1), p.15.DOI: https://doi.org/10.5334/jors.151
McTigue, J. D., & White, A. J. (2018). A comparison of radial-flow and axial-flow packed beds for thermal energy storage. *Applied Energy*, *227*, 533–541. https://doi.org/10.1016/j.apenergy.2017.08.179
Olumayegun, O., & Wang, M. (2019). Dynamic modelling and control of supercritical CO2 power cycle using waste heat from industrial processes. Fuel, 249, 89–102. https://doi.org/10.1016/j.fuel.2019.03.078
Schumann, T. E. W. (n.d.). *HEAT TRANSFER: A LIQUID FLOWING THROUGH A POROUS PRISM*.
Davenne, T. R., & Peters, B. M. (2020). An Analysis of Pumped Thermal Energy Storage With De-coupled Thermal Stores. *Frontiers in Energy Research*, *8*. https://doi.org/10.3389/fenrg.2020.00160
Winandy, E., Saavedra, C., & Lebrun, J. (n.d.). Experimental analysis and simpli®ed modelling of a hermetic scroll refrigeration compressor.www.elsevier.com/locate/apthermeng
Y. Ma, S. Gowda, R. Anantharaman, C. Laughman, V. Shah, and C. Rackauckas, "Modelingtoolkit: A composable graph transformation system for equation-based modeling," 2021
Zanganeh, G., Pedretti, A., Zavattoni, S., Barbato, M., & Steinfeld, A. (2012). Packed-bed thermal storage for concentrated solar power - Pilot-scale demonstration and industrial-scale design. *Solar Energy*, *86*(10), 3084–3098. https://doi.org/10.1016/j.solener.2012.07.019

ACKNOWLEDGEMENT

This work is supported by Belgian Defense under the project grant MSP-23-05 BCARN.