A biomass-fueled combined steam and sCO2 heat and power cycle for Southern African conditions

Prepared by:

Rashid Ahmed Haffejee

HFFRAS002

Thesis Presented for the Degree of

DOCTOR OF PHILOSOPHY

In the Department of Mechanical Engineering

University of Cape Town

Supervisor: Prof Pieter Rousseau

Co-supervisor: Prof Ryno Laubscher

April 2024

Declaration

I, Rashid Ahmed Haffejee, hereby declare the work contained in this thesis to be my own. All information which has been gained from various journal articles, textbooks or other sources has been referenced accordingly. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor

_____	22/04/2024
Name	Date

Declaration on the Inclusion of Publications in a PhD Thesis

I, Rashid Ahmed Haffejee, confirm that I have been granted permission by the University of Cape Town's Doctoral Degrees Board to include the following publication(s) in my PhD thesis, and where co-authorships are involved, my co-authors have agreed that I may include the publication(s):

Conference publications

1. R. A. Haffejee, P. G. Rousseau, and R. Laubscher, "Impact of integrated a supercritical-CO2 heat exchanger within an existing biomass-fired boiler of a heat and power steam cycle," *in Proceeding of International Heat Transfer Conference 17*, 2023, p. 10. doi: 10.1615/IHTC17.190-50.

Journal publications

2. R. A. Haffejee, P. Rousseau, and R. Laubscher, "Integrated Performance of a Modular Biomass Boiler With a Combined Heat and Power Industrial Rankine Cycle and Supplementary sCO2 Brayton Cycle," J. Therm. Sci. Eng. Appl., vol. 16, no. 9, pp. 1–26, May 2024, doi: 10.1115/1.4065491.
3. R. A. Haffejee, P. G. Rousseau, and R. Laubscher, "Part-load performance analysis of a modular biomass boiler with a combined heat and power industrial Rankine cycle and supplementary sCO2 Brayton cycle," 2024 (Submitted, under review) *

Signature: Signed by candidate Date: 31/03/2024

Student Name: Rashid Ahmed Haffejee Student Number: HFFRAS002

Abstract

Biomass is a renewable, cost-efficient, carbon-neutral fuel obtained from agricultural waste streams or energy crops that can be combusted in a furnace to co-generate electricity and heat. Integrating a supplementary high efficiency cycle, such as the supercritical-CO2 (sCO2) Brayton cycle, with an existing industrial Rankine cycle and a biomass fired boiler may be an economical option to increase overall thermal efficiency and net generation. However, the integration of sCO2 heaters within the biomass boiler presents challenges related to operating philosophies and component specifications.

The focus of this research was to investigate the integration of a sCO2 Brayton cycle with a combined heat and power steam cycle with a modular biomass boiler firing typical Southern African bagasse fuel. A quasi-steady state 1D thermofluid network-based process model of the sCO2, steam and flue gas cycles was developed for nominal and partial load analysis. It accounts for the detailed component characteristics for the Rankine and Brayton cycles, as well as the biomass boiler, together with the complex interactions between all of the components in the different cycles. To facilitate the analysis of these intricate systems, a sophisticated simulation code was developed to allow for necessary customization and enforcement of required boundary conditions and control parameters. The network model solves the mass, energy, momentum, and species balance equations for the various fluid streams, accounting for radiative and convective heat transfer phenomena in the boiler. Due to the novelty of the proposed integrated cycle, high-fidelity 3D CFD modelling was then also used to validate the heat uptakes for the sCO2 heaters in the biomass boiler. Two configurations with the sCO2 heater/s situated within the flue gas flow path were investigated, namely a single convective-dominant heater, and a dual heater configuration with a radiative and a convective heater.

At nominal load, the network model results show the required rate of overfiring for the sCO2 configurations, with a 15.3% increase in fuel flow rate resulting in an additional 21.2% in net power output. The impact of the sCO2 heaters situated in the gas flow path was quantified, with reduced heat uptakes for downstream steam heat exchangers offset by increased furnace waterwall heat transfer. At partial loads, between 100%-60%, inventory control proves to be the better performing control strategy for load following, maintaining high thermal efficiency across partial loads. Notably, at 60% load, the sCO2 compressor inlet conditions are near the pseudo-critical point, which requires careful management of inventory control. The boiler CFD modelling highlighted lower heat uptakes for sCO2 heaters compared to the 1D model, exacerbated at lower loads, particularly for the dual heater configuration. The 1D model was consequently calibrated based on these results.

The single sCO2 heater configuration is recommended as the preferred configuration to minimise adverse impacts on the Rankine cycle superheaters. Further iterations between the 1D process model and CFD model are recommended.

Key Words: Biomass; supercritical-CO2, thermofluid process modelling; computational fluid dynamics;

Acknowledgements

In the name of Allah SWT, the Most Gracious, the Most Merciful.

All praise is due to Allah SWT.

I would like to thank my supervisors, Prof Pieter Rousseau and Prof Ryno Laubscher, for their continuous support, mentorship, patience, and feedback over the last three years and beyond, as well as for the various opportunities to publish my research and expand my horizons. I would also like to thank you for guiding me to follow a path in research, and for helping me stick through this journey.

I would like to thank the Harry Crossley Foundation, the University of Cape Town, the Applied Thermofluid Process Modelling (ATProM) research group, and the Reino Stegen Scholarship donors for funding my PhD research over the last three years. Moreover, I would also like to thank the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources used in this study.

Additionally, I would like to thank all my friends, as well as academics and colleagues at the ATProM office, UCT, and beyond for their support during this journey. A special thanks to Priyesh Gosai for his insight and support throughout my postgrad studies, and for encouraging me to take the leap forward. I would also like to thank the Department of Mechanical Engineering for their assistance throughout the degree.

Importantly, I would like to thank my parents and siblings, for their unwavering and invaluable support and sacrifices, their love, and their continuous encouragement to complete this long journey, which I will always treasure and value.

Table of Contents

List of Figures

List of Tables

List of Nomenclature for Integrated 1D Process Model

General symbols

A	Area	[m^2]
a_0/a_1	Linearised component characteristics	
b	Heat exchanger depth	[m]
Bo	Boltzmann number	
b_l	Tube bank heat exchanger depth	
$CM_{C/T}$	Non-dimensional mass flow rate	
d_{OD}	Tube outer diameter	[m]
f	Friction factor	
$f_{radloss}$	Furnace radiation loss factor	
g	Gravitational acceleration	[m/s^2]
H_{avg}	Duct height	[m]
HHV	Higher heating value	[kJ/kg]
h	Enthalpy	[J/kg]
	Heat transfer coefficient	[W/m^2K]
K_{loss}	Secondary loss factor	
k	Conductivity	[W/mK]
L_{duct}	Boiler heat exchanger duct length	[m]
LHV	Lower heating value	[kJ/kg]
M	Flame modification factor	
$\dot{m}$	Mass flow rate	[kg/s]
N	Number of mols	
N_l	Number of longitudinal tubes	
N_t	Number of transverse tubes	
Nu	Nusselt number	
p	Pressure	[Pa]
Pr	Prandtl number	
Δp_{0L}	Pipe pressure loss	[Pa]
Δp_{0M}	Machine pressure change	[Pa]

q'''	Volumetric heat transfer rate	[W/m^3]
$\dot{Q}$	Rate of heat transfer	[W]
Q_c	Compressor volume flow rate	[m^3/s]
Q_{abs}/Q_{bsc}	Non-dimensional particle absorption and scattering efficiency	
Re	Reynolds number	
R_{fi}	Tube inner fouling thermal resistance	[m^{2}K/W]
R_{fo}	Tube outer fouling thermal resistance	[m^{2}K/W]
S	Mean beam length	[m]
S_l	Tube bank longitudinal pitch	[m]
S_t	Tube bank transverse pitch	[m]
t_{wall}	Tube wall thickness	[m]
UA	Overall conductance	[W/K]
V	Volume	[m^3]
w'''	Volumetric work rate	[W/m^3]
$\dot{W}$	Work done	[W]
W_{avg}	Duct width	[m]
x	Coordinate in flow direction	[m]
x_{ang}	Angular coefficient	

Greek symbols

α	Excess air ratio	
β	Re-radiation coefficient	
	Heat transfer coefficient correction factor	
$\varepsilon_{eff,fur}$	Effective gas and particulate emissivity	
ε	Effectiveness	
	Emissivity	
η	Efficiency	
	Coefficient of heat flux non-uniformity	
	Fin efficiency	
ω	Absolute humidity	[kg/kg]
Φ_{gp}	Optical thickness	

ϕ_{fur}	Heat preservation coefficient	
ϕ_{ang}	Angular coefficient of incident direct radiation	
ψ	Furnace efficiency, (area-weighted with overbar)	
	Gas void fraction	
ψ_{HX}	Heat exchanger utilisation factor	
ρ	Density	[kg/m^3]
σ_0	Stefan-Boltzmann constant	[W/m^{2}K^4]
ξ	Furnace wall fouling factor	

Acronyms

AFT	Adiabatic flame temperature
AH	Air heater
AR	As-received
ATT	Attemperator
BFP	Boiler feed pump
CA	Combustion air
CAV	Cavity
CFP	Condensate feed pump
DA	De-aerator, distribution air
DAF	Dry ash-free
EC	Economiser
EV	Evaporator
FA	Fly ash
FE	Furnace exit
FG	Flue gas
FGR	Flue gas ratio
Fur	Furnace
HAR	Humid air ratio
HPT	High pressure turbine
LPT	Low pressure turbine
PA	Primary air
PC	Pre-cooler
RF	Refractory walls

RX	Recuperator
RXHP/RXLP	Recuperator – High/low pressure side
SA	Secondary air
SD	Steam drum
SCR	Screen
SH	Superheater
sCO2	Supercritical carbon dioxide
sCO2H1	Radiative-platen sCO2 heater for dual configuration
sCO2H2	Convective sCO2 heater for dual configuration
sCO2HX	Convective sCO2 heater for single configuration
TIT	Turbine inlet temperature
TTD	Terminal temperature difference
UC	Unburnt carbon

Subscripts and Superscripts

abs	Absorbed radiation
conv	Convective
e	Outlet
ex/ext	External
fa	Fly-ash
fe	Furnace exit
fg	Fluegas
fo	Outer fouling layer
i	Inlet
int	Internal
p	Products
r	Reactants
rad	Radiative
rf	Refractory
ri	Incoming incident radiation
re	Outgoing radiation
sec	Secondary fluid stream
WW	Water walls
wi	Inner tube wall
wo	Outer tube wall

List of Nomenclature for CFD modelling

General symbols

A_{vol}	Volatile pre-exponential factor	[1/s]
C_D	Drag coefficient	
$\vec{D}_{k,m}$	Binary diffusion coefficient	[m^2/s]
D_o	Oxygen diffusion rate coefficient	
$D_{k,m}$	Laminar diffusion coefficient	[m^2/s]
$E_{a,vol}$	Activation energy	[J/kmol]
E_p	Particle emissive power	
E_{source}	Volumetric energy source term	[W/m^3]
F_D	Particle drag variable	[1/s]
$F_{y,drag}$	Particle drag force	[N]
f_p	Particle scattering factor	
$f_{v,0}$	Fuel volatile dry-fraction	
G	Incident radiation flux	[W/m^2]
I	Radiation intensity	[W/m^2sr]
k	Turbulence kinetic energy	[m^2/s^2]
	Tube thermal conductivity	[W/mK]
k_{vol}	Devolatilisation Arrhenius rate	[1/s]
$\vec{J}_k$	Mass diffusion flux	[kg/m^2s]
M_w	Molecular weight	[kg/kmol]
N_φ	Azimuthal angle discretisations	
N_θ	Polar angle discretisations	
N_p	Number of particles	
$R_{k,r}$	Reaction rate	[kg/m^3s]
R_c	Char combustion kinetic rate coefficient	
S	Mass source term	[kg/m^3s]
	Momentum source term	[N/m^3]

	Mean beam length	[m]
Sc_t	Turbulent Schmidt number	
t_{grate}	Grate thickness	[m]
u / v	Velocity	[m/s]
x_{mesh}	Representative mesh cell size	[m]
Y	Mass fraction	

Greek symbols

α	Absorption coefficient	[1/m]
	Absolute permeability	
ε	Turbulence dissipation rate	[m^2/s^3]
γ	Porosity	
λ	Thermal conductivity	[W/mK]
μ	Viscosity	[kg/ms]
$\dot{\omega}$	Volumetric generation rate	[m^3/s]
ω_j	Quadrature weights	
ρ	Density	[kg/m^3]
σ_p	Scattering coefficient	[1/m]
θ_R	Radiation temperature	[K]
$\vartheta_{k,r}$	Stoichiometric coefficient	

Acronyms

AR	As-received
DAF	Dry ash-free
DO	Discrete Ordinates
EDC	Eddy-dissipation concept
FC	Fixed carbon
RANS	Reynolds Averaged Navier-Stokes
RTE	Radiative Transport Equation
TTD	Terminal temperature difference

VM	Volatile matter
WSGGM	Weighted-sum-of-gray-gases model

Subscripts

c	Char
eff	Effective
g	Gas-phase
hi	Internal heat transfer
int	Internal
k	k^{th} porous media zone
	Species
m	Mixture
p	Particle
p,0	Initial particle property
pm	Porous medium
rad	Radiative
vol	Volatile

1. Introduction

1.1 Background and Motivation

Fossil fuels are currently the world's largest source of energy [1]. However, there are growing concerns about global warming, unsustainability, and increasing greenhouse gas emissions from the use of fossil fuels. The International Energy Agency (IEA) has outlined a roadmap for net-zero emissions by 2050, where 90% of electricity production is predicted to come from renewable sources [2]. Biomass can be classified as renewable organic matter from plants or animals, derived from waste streams or commercial crops grown specifically for energy generation, known as energy crops [3]. Typical biomass fuels include wood, by-products from crops (such as bagasse from sugarcane processing), animal waste, and biogenic material from municipal solid waste. Biomass is therefore classed as a renewable, cost-efficient, carbon-neutral fuel. South Africa is the largest producer of sugarcane in Africa [4], producing approximately 18 million tons of sugarcane in 2022/2023 [5]. Consequently, bagasse, a by-product from sugarcane processing, also accounted for approximately 93% of electricity produced by biomass in South Africa in 2023/2024, estimated at 242 MWe [6]. Sugarcane is also grown all year round in South Africa, making bagasse a promising candidate for biomass fuel.

Common methods to extract energy from solid biomass are through thermochemical processes such as direct combustion, pyrolysis, or gasification [7]. Direct combustion is generally the method of choice for heat and power production, due to its flexibility and adaptability [3], [8]. Direct combustion of biomass has inherent issues due to fuel preparation and constituent variability of the fuel [9]. The high moisture content in biomass fuel, around 50%, can lead to combustion instability and discrepancies in boiler efficiency, necessitating simulation models to study the effects of fuel variability on boiler design and operation [10]. Biomass combustion also has benefits for distributed or embedded generation due to the availability and proximity to fuel sources [11].

Small- to medium-scale, modular, co-generating biomass boilers are well-suited to rural agro-industrial settings which require electricity and process heat, allowing for self-sustaining generation systems [12]. Excess electricity can also potentially be supplied to surrounding rural communities at cheaper costs compared to national grid access, providing an additional revenue stream [13], [14]. In a South African context, there have been several changes in energy policy for generation projects over the last few years to address the critical shortfall of generation capacity in the country [15], [16]. These policy adjustments reduce barriers for implementation of medium-scale generation systems, including those using biomass fuels, to help relieve a severely constrained power grid. The MicroGen® boiler is a modular grate-fired boiler, capable of producing up to 35 MW_{th} for combined

heat and power generation. The boiler is developed and manufactured by John Thompson, a South African boiler manufacturer, and can accommodate a wide range of renewable biomass fuels, as well as coal [17].

Biomass fuel is typically used to generate electricity through various Rankine-type steam cycles, with an industrial steam Rankine heat and power cycle shown in Figure 1.1. However, supercritical carbon dioxide (sCO2) Brayton power cycles have been identified as a potentially attractive technology with a small physical footprint and improved thermal efficiency compared to Rankine cycles, reducing initial capital and operating expenditure [18].

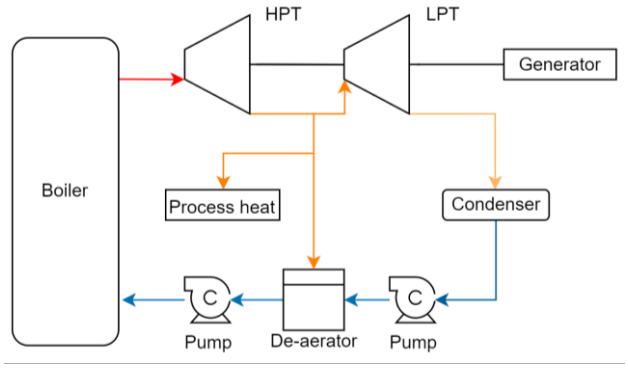


Figure 1.1: Typical industrial Rankine cycle

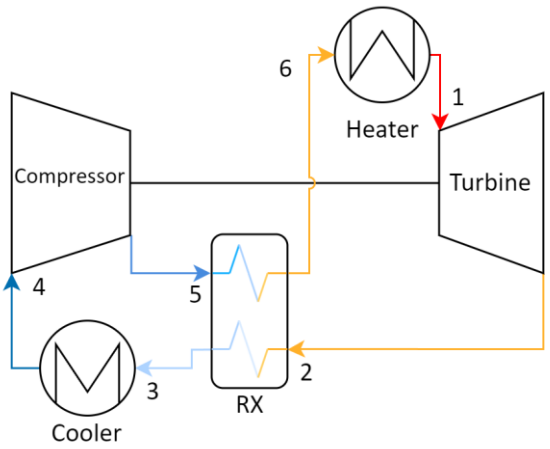


Figure 1.2: Simple recuperated sCO2 Brayton cycle layout

The sCO2 Brayton cycle was initially proposed by Angelino [19] as well as Feher [20] in 1968, and has garnered significant research attention particularly over the past decade. Since the cycle operates in the supercritical region (>7.4 MPa and 31°C) at high pressures and densities, component sizes can be reduced. Additionally, the real gas compressibility and density of CO2 near the critical point can be leveraged to decrease the compression work required. Several cycle layouts have been investigated, along with operating and process parameters [21]. The simple recuperated

cycle has been studied extensively as a stand-alone cycle for various applications, including nuclear, solar, and waste heat recovery [21]–[25], and additionally for combined cycles [26]–[28] with topping and bottoming configurations. Figure 1.2 shows the simple recuperated cycle layout, consisting of a compressor and turbine, cooler (or pre-cooler), heater, and recuperator (RX). Currently, there are no utility-scale sCO2 plants in operation, however, there are pilot plants in development [29]. By leveraging the high efficiency of the sCO2 Brayton cycle, increases in net generation may provide avenues for additional revenue streams. Moreover, when integrated with a biomass boiler, the additional power generation may be used to drive a CO2 sequestration plant, which captures CO2 released from biomass combustion, helping reduce emissions. These potential benefits highlight the utility of integrating an sCO2 Brayton cycle with existing low efficiency generation systems.

1.2 Research statement

Energy efficiency plays a key role in net-zero emissions by 2050 [2], and the integration of high efficiency supplementary cycles with existing generation systems can potentially increase energy efficiency and net generation. For existing industrial Rankine cycles, increasing generation capacity poses challenges, particularly with large capital expenditure requirements and potential boiler limitations. This is further exacerbated by the relatively low thermal efficiency of industrial Rankine cycles. Retrofitting an existing modular boiler with a high efficiency supplementary sCO2 power cycle may prove to be more economical regarding capital expenditure as well as overall thermal efficiency.

The proposed approach involves situating the sCO2 heater directly in the flue gas flow path of the boiler. Due to the compact and modular nature of the MicroGen® boiler, integration of a sCO2 power cycle with a Rankine steam cycle fuelled by biomass in the MicroGen® boiler has potential. The reduced physical footprint of sCO2 power cycles is particularly beneficial when incorporating a supplementary sCO2 power cycle in existing generation systems. The integration of these cycles together with the biomass boiler requires adjustments to system configuration and operating philosophy. Further research is required to identify and analyse the integrated system layout, how the integrated cycle operates, the impacts on the existing cycle, and the overall performance and efficiency for various operating conditions. There is also a need to focus on the applicability of the proposed cycle for Southern African conditions using Southern African biomass fuel. Moreover, the integration of a sCO2 Brayton cycle into a biomass boiler, such as the MicroGen®, together with an existing Rankine cycle has not been previously explored in literature.

With the novelty of the proposed cycle and integration, an experimental approach is not practical. Numerical modelling is a robust tool that can be used to gain insight into feasibility and overall

performance for such a cycle. 1D thermofluid process modelling is a powerful simulation tool and is used to simulate power cycles at varying levels of complexity. These models are also computationally efficient and able to accurately capture various flow phenomena and component characteristics, including empirically-derived heat transfer correlations. Thus, 1D process modelling is well-suited to simulate the proposed integrated cycle. High-fidelity computational fluid dynamics (CFD) simulation is typically the most accurate approach, particularly for boiler modelling, however this approach has high computational expense. Empirical models commonly used for 1D modelling of combustion and radiative heat transfer lack accuracy compared to detailed CFD modelling, mainly due to the spatial characteristics of these phenomena. 3D CFD modelling can therefore be used to capture detailed combustion modelling, as well as radiative and convective heat transfer for boilers.

In summary, there is a need to investigate a steam Rankine and sCO2 Brayton combined heat and power cycle for typical Southern African biomass fuels by applying 1D and 3D thermofluid modelling to gain insight into the overall performance and operation of such an integrated cycle. Particular attention is paid to the integration of these cycles in the biomass boiler and the practical implications that arise from integrating these cycles.

1.3 Hypothesis

The hypothesis for this research is as follows: A Rankine and sCO2 Brayton combined heat and power cycle with a modular biomass boiler firing typical Southern African bagasse fuel can be analysed by applying 1D thermofluid network-based process modelling and 3D CFD to study the integrated system performance, the interaction between the components in the different cycles, and the impact on operating philosophies.

1.4 Research approach and objectives

The focus of the project is to investigate the integration and configuration of a steam Rankine and sCO2 Brayton combined heat and power cycle with the MicroGen® boiler using typical Southern African biomass fuel, by applying 1D and 3D thermofluid modelling. This includes a quasi-steady-state analysis of the integrated cycle for nominal and partial loads. The research will be completed through the following research objectives:

- Develop a 1D thermofluid process model of a steam Rankine and sCO2 Brayton combined heat and power cycle incorporating the MicroGen® boiler for typical Southern African biomass fuel.
    - Specify detailed component characteristics, including sCO2 turbomachinery.

  - Specify the size and layout of sCO2 heaters situated in the boiler, including two different sCO2 heater configurations.
  - Develop a solution methodology for quasi-steady-state analysis at various loads.
- Analyse the integrated system performance at nominal and partial loads for the two different configurations.
- Develop a 3D CFD model of the MicroGen® boiler to explore the practical implications and effects of integrating the sCO2 heaters into the biomass boiler, and to validate heat uptakes at the sCO2 heaters.

Transient analysis of the integrated cycle lies outside the scope of this study. Part-load case studies are explored through quasi-steady-state analysis and control. Additionally, a techno-economic analysis of the integrated cycle is not within the scope of the study.

1.5 Thesis layout

The thesis is laid out in two parts. The first part addresses the development of the integrated 1D thermofluid process model and analysis of the integrated cycle at nominal and part loads. The second part focuses on the development of the 3D CFD model and its results. Figure 1.3 shows the thesis layout, broken down into chapters.

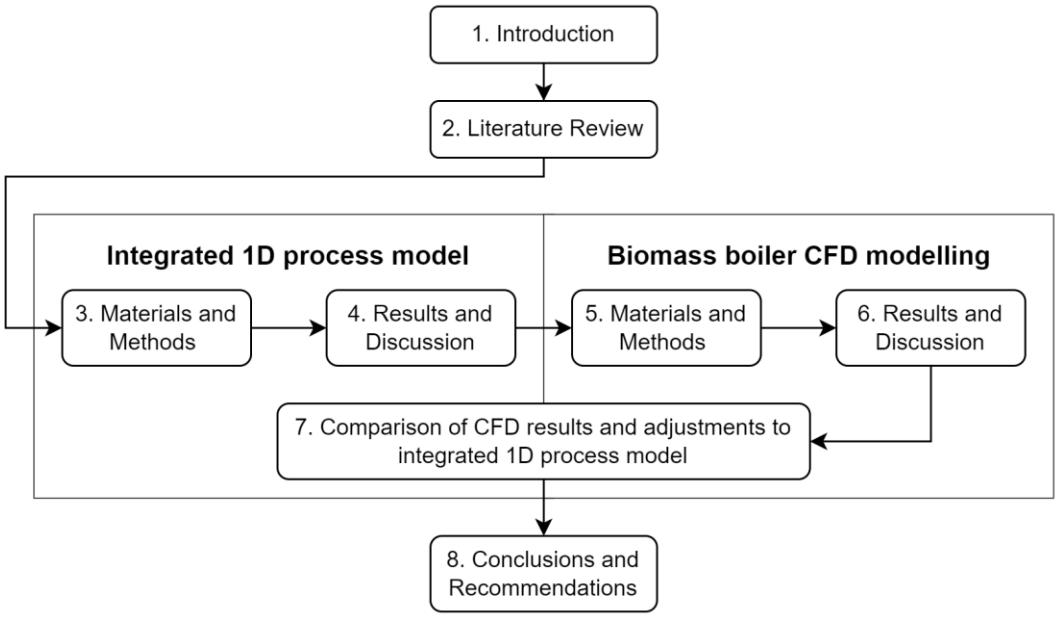


Figure 1.3: Thesis layout

Chapter 2 provides a review of existing literature relating to sCO2 power cycles, modelling of biomass-fuelled boilers, including both 1D process models and 3D CFD models, and lastly sCO2 power cycles using biomass fuels.

Chapter 3 and Chapter 4 form the first part of the thesis. Chapter 3 details the case study cycle layout and operation, as well as the integrated 1D process model theory and development. Chapter 4 discusses simulation results for the integrated cycle, including two sCO2 heater configurations. Part-load analysis is also conducted for the integrated cycle, with various control methodologies investigated for the sCO2 Brayton cycle. The first two publications, as mentioned in the Declaration on the inclusion of publications, form a part of Chapters 3 and 4, particularly the nominal load case study discussions. The third publication forms a part of Chapter 3 and 4, particularly the part-load case study. These publications contribute directly to the research forming this PhD thesis.

Chapter 5, Chapter 6, and Chapter 7 form the second part of the thesis. Chapter 5 discusses the theory and development of the 3D CFD model, while Chapter 6 explores simulation results from the CFD models. Chapter 7 compares the results from the CFD modelling and integrated 1D process model and explores tuning the 1D process model accordingly. Lastly, Chapter 8 provides conclusions and recommendations for future work.

2. Literature Review

The literature review is split into three sections. Firstly, sCO2 Brayton power cycles were investigated, including cycle-level modelling, turbomachinery models, and using a sCO2 cycle as a topping/bottoming cycle. Next, biomass boiler models are investigated, specifically 1D process models and detailed 3D CFD studies. Lastly, a review of sCO2 power cycles integrated with generation systems fuelled by biomass is presented.

2.1 sCO2 power cycles

sCO2 power cycles have garnered significant interest over the past decade due to technological breakthroughs, high cycle efficiency, and compactness. sCO2 cycles have been researched for a wide range of applications, including nuclear [30], solar [31], and waste heat recovery (WHR) [32]. 1D modelling is used extensively to characterize sCO2 Brayton cycles due to its computational efficiency when analysing and optimising various cycle layouts [21], [33]–[35]. Typically, simple thermodynamic analysis is used to model the cycle, with the majority of studies solving mass and energy balance equations. While several cycle layouts are explored in literature, the simple recuperated cycle is selected as the sCO2 Brayton cycle layout for this study, due to its simplicity while maintaining a relatively high thermal efficiency, as high as 40% [21].

Sarkar investigated steady-state energy and exergy analyses and the optimisation of maximum cycle efficiency for a sCO2 cycle at a system-level for nuclear applications [36]. The model solved the mass and energy balance equations for cycle components. Inputs provided to the model included isentropic efficiencies for the turbine and compressors, compressor pressure ratio, high temperature recuperator (HTR) and low temperature recuperator (LTR) effectiveness, compressor inlet temperature (CIT), turbine inlet temperature (TIT), and the maximum cycle pressure. These parameters were varied in a sensitivity analysis. For increasing CITs, the thermal efficiency decreased, as the main compressor work increased as the CIT shifted further from the critical point.

Turchi et al. performed a system-level study of sCO2 power cycles for concentrated solar power (CSP) systems using cycle component models [34]. The authors investigated various cycle layouts, including the recompression cycle with reheat, and the partial cooling cycle with recompression and reheat. The selected cycles were optimized for maximum thermal efficiency for a range of operating conditions, including cycle reheat, TITs, and CITs. The turbomachinery was modelled using fixed isentropic efficiencies. Recuperators were modelled as counterflow heat exchangers and were discretized to account for the variation in thermophysical properties of CO2 near the critical point. A fixed effectiveness value was assumed for the HTR and the high-temperature flow streams for

both recuperators. Heat exchangers with external sources or sinks, such as the pre-cooler or heater, were modelled to meet specified temperatures at the heat exchanger outlet, such as the CIT or TIT respectively. The sizes of these heat exchangers and external conditions were not considered. The recompression cycle with reheat was the best performing cycle below a 700 °C TIT, with a thermal efficiency of approximately 52.3%.

Dostal et al. compared several sCO2 cycles in detail for nuclear applications, using a code developed in-house called CYCLES [35]. The authors recommended the standard recompression cycle due to its relative simplicity, compactness, and high efficiency. A steady-state 1D detailed model of the recompression cycle and simple recuperated cycle was developed. Printed circuit heat exchangers (PCHE) were used for all the heat exchangers including recuperators, and pressure drops in the PCHEs were also modelled. Multi-stage axial turbomachinery was modelled using real gas properties with the ability to account for off-design conditions using characteristic curves developed using the AXIAL code library. The authors conducted a performance analysis and optimization on the recompression cycle using parameters such as pressure ratios, CITs, TITs, sizes of recuperators, and the size of the pre-cooler. The recompression fraction was highlighted as an important cycle parameter, particularly for off-design modelling, as it directly affected the flow rate through the compressors. The optimal size of total heat exchanger volume was also found based on cycle efficiency and cost. Component sizes were specified, with the dimensions of the 600 MW_{th} power cycle at 18 m tall and 7.6 m wide.

The sCO2 power cycle has also been researched as a viable supplementary cycle for several applications, including biomass fuelled systems and WHR [32], [37]–[39]. Akbari et al. performed a system-level thermo-economic analysis and optimization of a topping sCO2 recompression cycle and a bottoming organic Rankine cycle (ORC) for nuclear applications [40]. A parametric study was carried out to investigate the performance of both cycles as well as a total unit cost optimisation, achieving a 43% thermal efficiency. Moroz et al. evaluated gas turbine exhaust heat recovery using a sCO2 bottoming cycle [37]. Approximately 16.30 MW (32.01% increase) of power could theoretically be recovered using the sCO2 power cycles. Wright et al. performed a thermo-economic analysis of four sCO2 cycles for WHR from a gas turbine [32]. The authors found that sCO2 cycles designed for WHR increased net cycle efficiency to approximately 46.5-49%, from 35.5% for only the gas turbine.

Part load modelling and analysis requires sufficiently detailed component models to accurately capture how a cycle performs for a wide operating envelope, particularly for turbomachinery. While the current work focuses on quasi-steady-state analysis, it is important to review current control methodologies used for sCO2 Brayton power cycles. This is typically explored through dynamic modelling, however, can also be explored through steady state control. For example, Dostal et al.

[35], discussed above, also explored several control methodologies for steady-state control typically used for gas power cycles with a focus on the sCO2 recompression cycle. These included valve control, as well as inventory control. Inventory control involves controlling the mass flow rate of the cycle by charging or discharging working fluid from the cycle to pressurised storage tanks. By controlling the inventory of the cycle, the cycle power output can be varied for different loads. Dostal et al. recommended a mixed control methodology, highlighting valve control for its simplicity, and the high thermal efficiency typically maintained across part loads using inventory control. Using dynamic modelling, Wang et al. [41] performed a review of control strategies for sCO2 Brayton cycles. Several control strategies for load-following are explored, including inventory control, valve-based control, as well as controlling the heat source inputs. The authors noted that the main advantage with inventory control was its ability to maintain high thermal efficiency. However, the drawbacks are its slow transient response as well as narrow load range. In contrast, valve control allows quick transient response but with a low cycle efficiency. Heifetz et al. [42] explored both inventory control and a turbine bypass valve for a sCO2-recompression Brayton cycle, as well as a mixed-mode control methodology utilising both inventory control and the turbine bypass valve. The mixed-mode controller utilised the quick response of the turbine valve and thereafter gradually switched to inventory control to maintain high thermal efficiency. Wang et al. [43] compared different load-following control strategies for a sCO2 recompression Brayton cycle through dynamic modelling. The authors concluded that a coupled control strategy using inventory control and compressor anti-surge had a large advantage over valve control.

With regards to part-load performance of specifically supplementary sCO2 Brayton cycles, Fan et al. [44] explored part-load performance of a recompression sCO2 Brayton cycle combined with a transcritical CO2 cycle. Various control strategies were explored, including adjusting compressor rotational speed, a turbine bypass valve, and inventory control. Inventory control was found to be the most efficient method for 100-35% load operating range. A composite control strategy combining both inventory control and a bypass valve was suggested for the sCO2 cycle. Fan et al. [45] also explored part-load analysis for a combined sCO2 and Kalina cycle. A bypass valve and variable-speed compressor were used to control the sCO2 cycle, while a sliding pressure control was used for the Kalina cycle which was sufficient for a wide operating load range.

Colombo [46] performed part load analysis of sCO2 power cycles for gas turbine bottoming applications. For various cooling methods, the TIT was varied to maximise cycle net power output. Notably, the influence of inventory control was also explored, and through a technoeconomic analysis it was found that inventory control was beneficial for part-load operation for maintaining high thermal efficiencies.

The studies above highlight the benefits of using inventory control as a control strategy for the sCO2 cycle. Additionally, the coupled use of valve control and inventory control was found to be the better overall performing control strategy, providing the benefits of responsive control through valves, and maintaining high thermal efficiency through inventory control.

2.2 Biomass-fuelled boilers

1D modelling has been used extensively for modelling power cycles due to its computational efficiency and relative accuracy [47], and has been used to capture steady-state and dynamic behaviour of coal-fired power plants [48]–[50]. Additionally, to capture highly non-linear behaviour and complex heat transfer phenomena in furnaces and boiler heat exchangers, lumped parameter models are often used. Several commercial simulation codes have been developed with varying levels of complexity, including Engineering Equation Solver (EES), Flownex SE, and Advanced Process Simulation Software (APROS), with the latter commonly used for coal-fired power plants. 1D process modelling is well-suited for modelling large integrated power cycles and has been used to investigate various operating conditions and for sensitivity analyses.

In a biomass context, several studies have focused on cycle-level applications using biomass combusted in boilers as an energy source. Singh investigated the integration of a Kalina cycle with an existing bagasse-fired 16 MW_e cogeneration plant [51]. The Kalina cycle, using NH_3-H_2O as the working fluid, was integrated as a WHR system. A 1D steady-state mass and energy balance was used to model components within the co-generation system and boiler. Detailed gas-side heat transfer and balance equations were not modelled for the boiler, with the focus on the steam/NH_3-H_2O side. Cavalcanti et al. performed energy and exergy analyses of a sugarcane bagasse power cogeneration system in Brazil [52]. A 1D mass and energy balance was used to model the air, flue gas, and water/steam fluid streams at a cycle level. The 200 ton/hr boiler was modelled, including the furnace, air heaters, economizers, evaporator, and superheater. The energy balance in the furnace was carried out by calculating the stoichiometric coefficients of the gaseous combustion products through a chemical species balance.

The two studies discussed above focused on high-level cycle modelling without modelling detailed heat transfer in the boiler or considering sizes of integrated heat exchangers. Chantasiriwan explored the optimum installation of the economizer, air heater, and a flue gas dryer for a biomass boiler [53]. The 1D model solved mass and energy balance equations and considered the effects of component sizing on boiler performance. An energy balance of the furnace including the fuel, air, and flue gas streams was used to calculate the furnace exit gas temperature.

Laubscher and De Villiers developed an integrated 1D model of a 105 ton/hr bagasse-fired industrial water tube boiler in Southern Africa with varying fuel moisture content [10]. The model accounted for detailed heat transfer in the boiler heat exchangers and solved the mass, momentum, and energy balance equations for the water/steam, flue gas, and air streams. A moving-grate stoker boiler was modelled with two superheaters, a spray-type desuperheater, evaporator tube bank, air heater, and economizer. A higher heating value (HHV) correlation for Southern African bagasse was used, accounting for varying moisture content. The model was simulated for three load cases and validated using plant measurements and a CFD model, with good agreement for higher boiler loads. For a sensitivity analysis, fuel moisture content was varied between 30-60%, and the impacts on boiler efficiency, furnace exit gas temperature and superheater outlet temperature were quantified.

Rousseau et al. performed heat transfer analysis using thermofluid network models for industrial biomass and utility-scale coal-fired boilers [54]. The analysis focuses on the integrated whole-boiler performance, incorporating 0D furnace models combined with the solution of mass, momentum, and energy balance equations. An instantaneous combustion model is used, assuming complete combustion with a fraction of unburned carbon. Two 0D furnace models are investigated, including the projected method and direct method. A generic radiative-convective heat exchanger model is formulated to capture component characteristics of various boiler heat exchangers, including superheaters, reheaters, economisers, and air-heaters. The heat exchanger model accounts for both convective and gas radiative heat transfer, and direct radiation. The developed methodology was used to investigate various loads for two different boilers, including a sugarcane bagasse fired biomass boiler located in Southern Africa. Results compared well to site measurements for full load and part load operation. The current research builds upon this methodology by Rousseau et al. [54] and is applied for 1D modelling of the case study biomass boiler, the MicroGen®.

Due to the complexities associated with the direct combustion of biomass, CFD has been used to simulate combustion and heat transfer in biomass boilers. Grate fired boilers are widely used in the direct combustion of biomass and are the focus of this research. Bermúdez et al. developed a transient 3D CFD model of a 4 MW_{th} grate fired biomass furnace for wet woody biomass fuel [3]. The simulation was segregated into a bed and freeboard, where the bed was modelled as a porous zone with user-defined scalers, and the freeboard was characterised by the gas phase. ANSYS-Fluent was used to simulate the gas phase, and a coupled Eulerian fixed bed sub-model was used for solving the biomass fuel bed and interaction with the gas phase. Bed temperature profiles and fuel mass fractions were calculated across the bed height and distance along the grate.

Zhou et al. investigated the effect of varying air supply and oxygen enrichment on biomass combustion in a 40 t/h grate fired boiler [55]. A 1D dynamic model was used to simulate the fuel

conversion process of biomass on the grate bed, accounting for varying primary air conditions and furnace radiation. A 3D CFD model was used to simulate the combustion process in the freeboard, taking the gas production results from the dynamic 1D model as inlet boundary conditions. An optimum ratio of primary air to secondary air was found for thermal efficiency and minimization of ash slagging.

Centeno-González et al. modelled the combustion of sugarcane bagasse in an industrial boiler using CFD [56]. User-defined functions were developed for devolatilization and grate combustion, and thermal field and combustion behaviour were predicted. The model assumed a continuous phase of reacting gases and a discrete phase of fuel particles immersed in the continuous phase. The conversion process of biomass fuel particles from initial heating to char combustion and ash was tracked. Devolatilization was found to be the dominant phenomenon of the particle trajectory and an important factor in predicting thermal fields.

Laubscher and Van Der Merwe investigated heat transfer modelling of a 100 ton/hr semi-suspension bagasse-fired industrial boiler at full- and part-load using CFD [57]. In contrast to the previous studies mentioned above, Laubscher and Van Der Merwe focused on resolving the heat transfer to the furnace walls, superheater tubes, and evaporator tube bank. The combustion process of biomass fuel was simplified into four steps: heating and evaporation, devolatilization, char combustion, and volatile combustion. A 1D mass and energy balance model was coupled to the gas-side in the CFD model to predict steam temperatures in superheaters. The model was validated using experimental data, as well as verified using a boiler process model for heat load predictions.

2.3 sCO2 power cycles using biomass fuels

The integration of sCO2 power cycles into generation systems using biomass as an energy source has been explored before at a cycle level. Several studies investigate the utilisation of biomass through gasification or digester systems [38], [39]. There are also a few studies investigating biomass-fuelled systems through direct combustion. Manente et al. proposed a biomass to power conversion system based on cascaded sCO2 Brayton cycles [58]. A 1D cycle level model was developed to explore the proposed system. A combination of cascaded cycle configurations using the simple cycle and recompression cycle were explored to maximize heat recovery from flue gas. Two biomass boiler configurations burning woody biomass were considered, including a radiative-convective configuration and a convective configuration. These configurations were also considered with co-current and counter-current flows. Optimization of the configurations with respect to net power output was performed. The configuration with a recompression topping cycle and simple bottoming cycle with a counter-current radiative-convective boiler achieved a maximum heat

recovery efficiency of 36%. For this configuration, the thermal efficiencies of the topping and bottoming cycles were 44.1%, and 36.9% respectively.

Mutlu et al. developed a hybrid system incorporating a single-flash geothermal power plant with a topping sCO2 and bottoming Rankine combined cycle fuelled by olive residue biomass [59]. The authors used the radiative-convective counter-current boiler configuration outlined by Manente et al [58]. The bottoming cycle Rankine cycle utilized waste heat rejected from the sCO2 topping cycle, as well as the existing turbine from the geothermal system. A 1D steady-state model of the combined cycle was developed for a nominal operating point. Additional net powers of $3.4\ MW_e$ and $5.3\ MW_e$ were generated by the sCO2 topping cycle and Rankine bottoming cycle, with thermal efficiencies of 40.1% and 16.9% respectively.

Zignani et al. [60] performed a preliminary feasibility study on the application of sCO2 power cycles to biomass-fired power plants. An existing $16\ MW_e$ power plant using a Rankine cycle fuelled by woody biomass burnt in a moving-grate stoker boiler was selected as a case study. The proposed sCO2 cycle was intended to replace the Rankine cycle. Design optimization was completed for three sCO2 cycle configurations, including the recuperated cycle, recompression cycle, and partial cooling cycle. A 1D steady-state model was developed for both the Rankine cycle and sCO2 cycle configurations. Turbine and compressor models assumed a fixed isentropic efficiency. PCHEs recuperators were used, modelling the pressure drop as well as a calculated surface area based on the overall heat transfer coefficient. A water-cooled shell and tube heat exchanger was used for the heat rejection unit. A thermodynamic and geometric sizing of the sCO2 boiler was also investigated for counter-current flow. The recompression cycle had the highest thermal efficiency compared to other configurations over a range of TITs from $450\,°C - 570°C$. Partial load analysis for the recuperated cycle was also investigated, using MATLAB and the 'fsolve' solver tool. Heat exchanger areas obtained from design conditions were used to calculate pressure drops and heat transfer coefficients for off-design conditions. Isentropic efficiencies for turbomachinery remained the same for off-design operation. However, compressor rotational speeds were unfixed, and the maximum operating pressure was determined by the turbine swallowing capacity. Plant load was varied from 100%-30%, and the effects on the cycle, boiler, and plant efficiency were quantified. Inventory control was also explored as a control strategy by varying the minimum cycle pressure and was found to have less impact on plant efficiency.

These studies provide valuable insights into the integration of sCO2 power cycles with biomass boilers. However, the analyses include high level lumped parameter component models, with limited focus on detailed boiler modelling. Very few studies on supplementary sCO2 cycles fuelled by biomass focus on part-load analysis. Additionally, to the author's knowledge, there have not been

any studies focusing on sCO2 cycles integrated with an existing cycle in the same biomass boiler. Further investigation is therefore warranted, specifically looking at the integration of an sCO2 Brayton cycle with a combined heat and power Rankine cycle, and the effects on boiler operation.

3. Integrated 1D process model: Materials and Methods

Using a high detail 1D model capable of capturing complex boiler heat transfer as well as detailed component characteristics for the integrated cycle is key to understanding the proposed integrated cycle behaviour. The novelty of the research lies with the proposed integrated cycle configuration, as well as the complexity of the developed 1D model. While works of similar nature focus on simple thermodynamic analyses through mass and energy balances, the present work solves the momentum balance equations in addition to mass and energy balances. This requires detailed component characteristics, including performance curves for turbomachinery, and allows for quasi-steady-state analysis for nominal and part loads. Moreover, Southern African bagasse is selected as the fuel source for this study, giving relevance to the prospective application of the integrated cycle.

A conventional combined heat and power industrial Rankine cycle without sCO2 integration, as well as with integration of two sCO2 Brayton cycle layouts are analysed and compared. The two Brayton cycle layouts differ only in terms of its heater configurations, namely a single convective-dominant heater, and a dual heater configuration with a radiative and a convective heater. The simple recuperated Brayton cycle configuration is selected for both sCO2 Brayton cycle layouts. The heaters for the supplementary sCO2 Brayton cycle are situated in the flue gas flow path inside the boiler. A 1D thermofluid process model is developed and applied to investigate the three cycle layouts with a biomass-fired boiler. These cycle layouts are simulated at nominal and part loads by applying a steady-state control methodology.

This chapter describes the development of the 1D process model. The integrated cycle including the MicroGen® boiler is described, along with two proposed sCO2 heater configurations. Additionally, governing equations, the integrated cycle layout, component characteristics, steady-state control, and solver methodology is discussed.

3.1 Cycle layout and operation

The case study existing biomass boiler is a compact modular grate-fired boiler, accommodating a wide range of fuels, such as bagasse which is the fuel used for this study. The boiler is rated at approximately 35 MW_{th} nominal heat generation, heating steam to 485°C at 69 bar. Schematics of the boiler geometry with the proposed single and dual sCO2 heater configurations included are shown at the top and bottom of Figure 3.1. The green arrows indicate the combustion air inlets and the black arrows the fuel inlet, while the red arrows indicate the flue gas flow path. For the existing

cycle without the sCO2 Brayton cycle integration, the schematic remains the same, except for the presence of the sCO2 heaters (shown in grey).

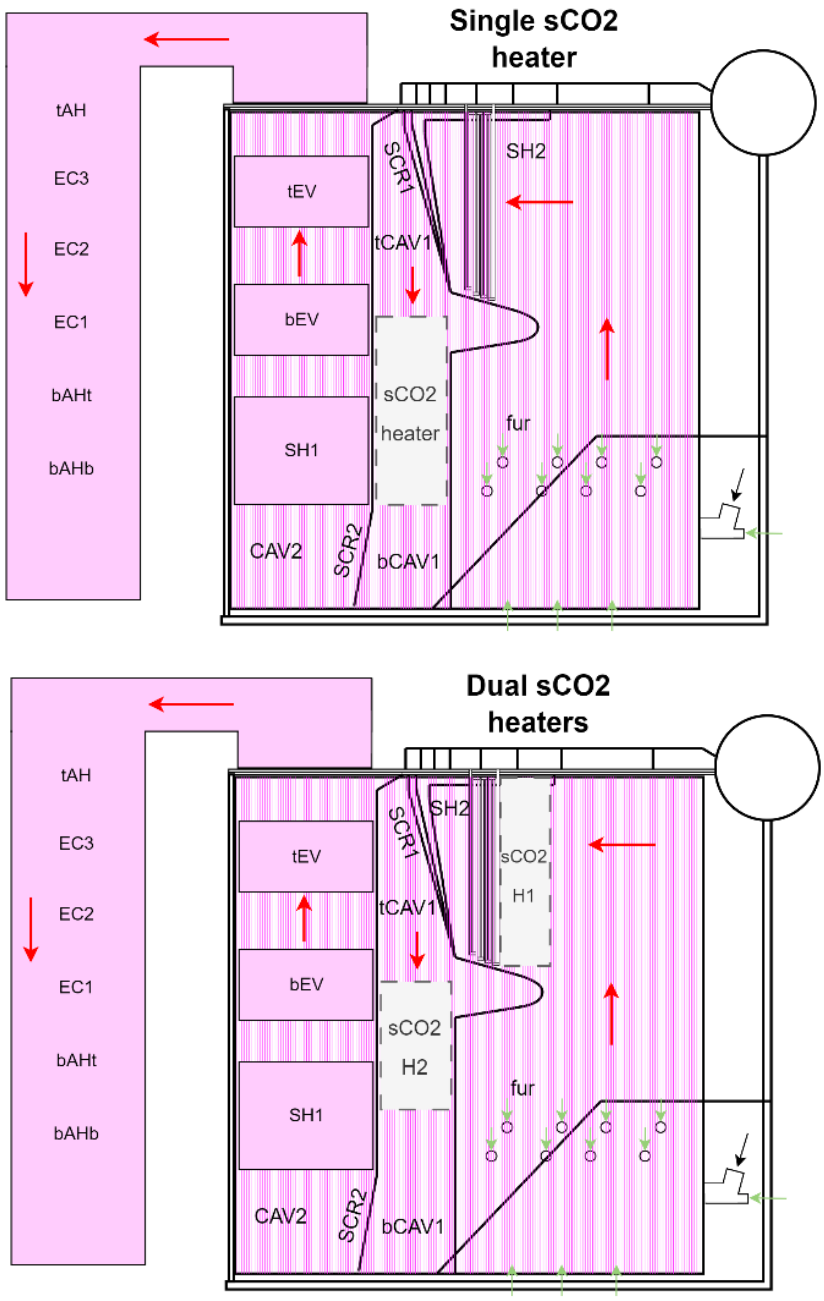


Figure 3.1: Boiler schematic with single sCO2 heater (top) and dual sCO2 heaters (bottom)

From Figure 3.1, evaporative water-walls fed by the steam drum via the downcomer pipe surrounds the furnace and other heat exchangers. There are two in-line steam superheaters (SH1 and SH2), an indirect attemperator, top and bottom evaporator tube banks (tEV, bEV), two heat exchanger

screens (SCR1, SCR2), two cavities (CAV1 - split into tCAV1, bCAV1, and CAV2), several tubular air-heaters (tAH, bAHt, bAHb), and three economizer heat exchangers (EC1, EC2, EC3).

The sCO2 heaters are a crucial component in the proposed integrated cycle, directly influencing the sCO2 Brayton cycle performance. The location of the sCO2 heaters in the boiler is also a key consideration which impacts existing downstream boiler heat exchangers. Two sCO2 heater configurations are proposed, namely, the single sCO2 heater configuration which has a convective sCO2 heater, and the dual sCO2 heater configuration which has a radiative and a convective sCO2 heater. The driving concept behind the dual configuration is to utilise incoming direct radiation from the furnace in sCO2H1 in addition to the convective heater, sCO2H2. In order to maximise the capacity of the sCO2 Brayton cycle, adequate sizing of the sCO2 heaters is required. The largest available space for the sCO2 convective heater is at CAV1. Thus, the convective sCO2 heaters, sCO2HX for the single configuration and sCO2H2 for the dual configuration, are positioned in CAV1, splitting the cavity into tCAV1 and bCAV1. For the dual configuration, the radiative sCO2 heater, sCO2H1, is positioned before SH2 allowing for absorption of direct radiation from the furnace.

Fuel, in this case wet Southern African bagasse in its raw form, is distributed via distribution air onto a moving grate which transports the fuel to the furnace rear. For industrial applications at sugarcane mills, boilers are typically designed to fire bagasse with high moisture, i.e. without any drying treatment [61]. Heated primary and secondary combustion air streams respectively are injected from below through the grate and directly into the lower furnace volume. The bagasse has a higher heating value (HHV) and lower heating value (LHV) of 8838 kJ/kg and 7165 kJ/kg respectively, with as-received ultimate analysis shown in Table 3.1. An excess air percentage of 27% is assumed, with bagasse boiler furnaces typically using an excess air percentage of 25-35% [62].

Table 3.1: Bagasse composition (as-received)

Fuel constituent	Carbon	Hydrogen	Oxygen	Nitrogen	Sulfur	Moisture	Ash	Unburnt carbon
Mass fraction (%)	21.09	2.68	20.77	0.16	0.02	50.0	4.66	0.62

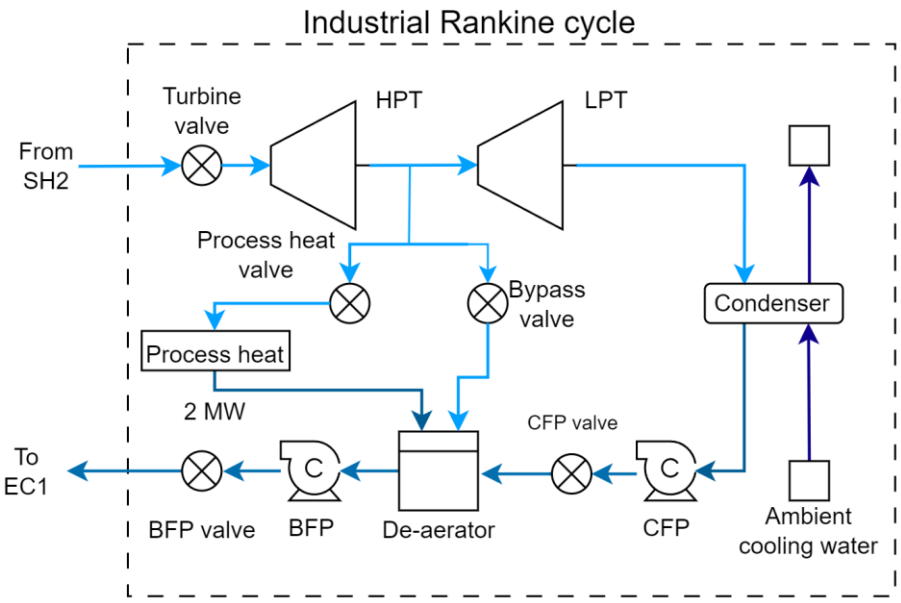


Figure 3.2: Industrial Rankine cycle process flow diagram

Figure 3.2 shows the industrial Rankine cycle schematic (without the boiler circuit). The CFP and BFP are used to pump the condensate and feedwater streams through the boiler evaporating circuit. Additionally, there are valves to allow for quasi-steady-state control of the Rankine cycle for various operating conditions. These include the CFP valve, BFP valve, turbine valve, process heat valve, and bypass valve. Power is generated by the HPT and LPT. The main flow stream splits after the HPT into the process heat line and the LPT. Process heat is provided according to the required demand. For this study, it is assumed that the process heat demand is fixed at 2 MW and water returning from the external process is saturated liquid. Moreover, a bypass line exists in parallel with the process heat line to allow for additional steam flow directly to the de-aerator. The steam flow through the LPT passes through the condenser where it is cooled by 1000 kg/s of cooling water provided at 25 °C. The CFP then pumps the condensate into the de-aerator, where the process heat and bypass lines also connect.

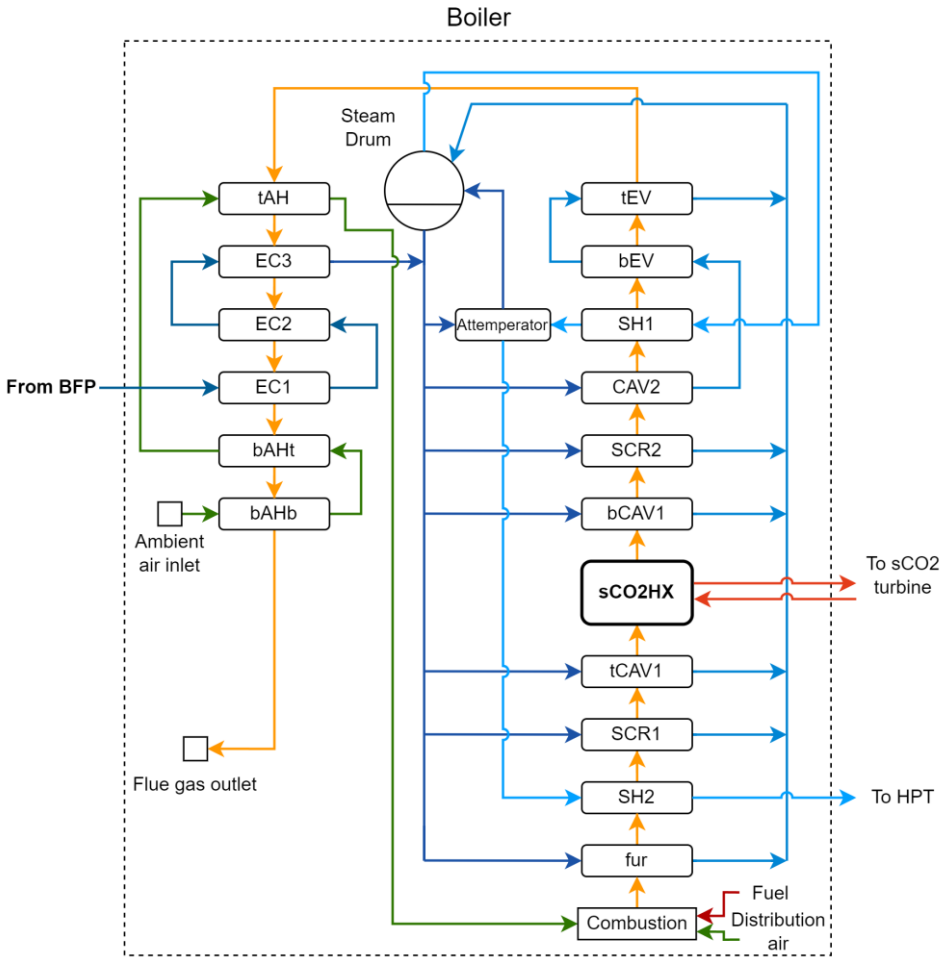


Figure 3.3: Boiler process flow diagram for single sCO2 heater configuration

Figure 3.3 shows the boiler schematic for the single sCO2 heater configuration. Boiler heat exchangers are modelled as tube bank heat exchangers. As mentioned previously, the convective sCO2 heater (sCO2HX) is slotted in CAV1. Note for the existing configuration, there is no sCO2 heater, and a single cavity is modelled in place (CAV1). For the air stream (green), primary and secondary air inlet is preheated through the air heaters and fed into the combustion chamber, alongside distribution air and fuel. Flue gas created from combustion goes through the boiler (orange), starting from the furnace and then downstream to various boiler heat exchangers as shown in Figure 3.1. The BFP pumps the main feedwater through the boiler (blue), including the economizers, water walls, evaporators, attemperator, and superheaters. Steam from the waterwalls and evaporating circuit collects in the steam drum, which feeds the superheaters, and in turn, the HPT. The indirect attemperator receives water from the drum, attemperates steam coming from SH1, and returns the heated water back to the drum.

Process flow diagrams of the complete integrated cycle for the single and dual sCO2 heater configurations are shown in Figure 3.4 and Figure 3.5 respectively. The process flow diagram for the

existing cycle without the integration does not include the sCO2 Brayton cycles nor sCO2 heaters. For this study a simple recuperated sCO2 Brayton cycle layout is selected. It includes a single compressor and turbine, recuperator, pre-cooler, and direct attemperator as shown in the process flow diagrams (Figure 3.4 or Figure 3.5). High-pressure sCO2 from the compressor outlet splits into two flow streams, with the majority flowing to the high-pressure side of the recuperator (RXHP) and the remaining going to the direct attemperator. From the turbine outlet, high temperature sCO2 on the low-pressure side of the recuperator (RXLP) heats up the sCO2 at the RXHP. sCO2 then enters the pre-cooler, where it is cooled by 500 kg/s of cooling water at 25°C. The sCO2 is cooled to 33°C at 10 MPa, which are the compressor inlet conditions.

It is important to include attemperation in the sCO2 cycle, allowing for temperature control at the heater outlet, i.e. the turbine inlet. For the single heater configuration, a portion of the flow bypasses the RXHP and is then mixed with the sCO2 heated in the RXHP, from where it flows into the sCO2 heater (sCO2HX). The dual heater configuration, shown at the bottom of Figure 3.1 and in Figure 3.5, utilises two sCO2 heaters. The first heater, sCO2H1, is a platen radiative heat exchanger, positioned in front of SH2. The second heater, sCO2H2, is like the heater in the single heater configuration (sCO2HX) and is positioned between tCAV1 and bCAV1. Attemperation for the dual configuration takes place between sCO2H1 and sCO2H2, with a portion of the flow bypassing sCO2H1 which is then mixed with the CO2 heated in sCO2H1.

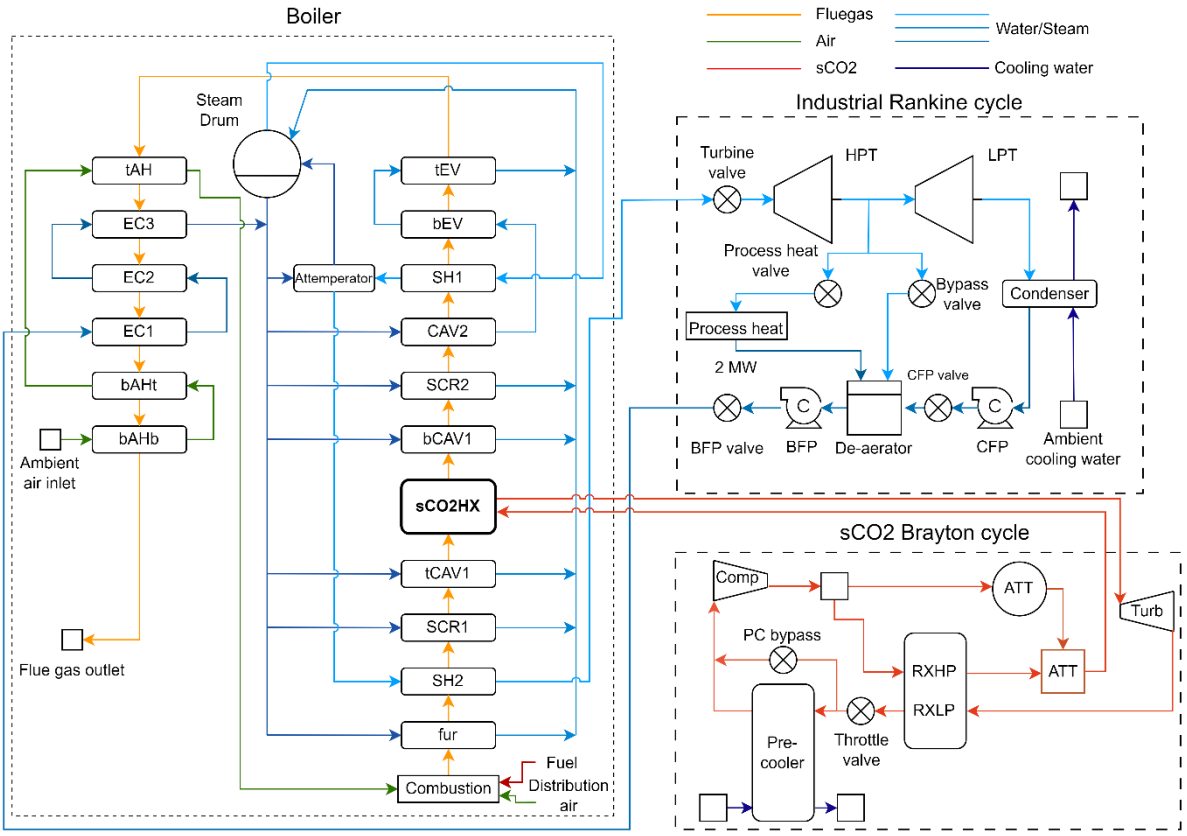


Figure 3.4: Process flow diagram for integrated cycle with single sCO2 heater

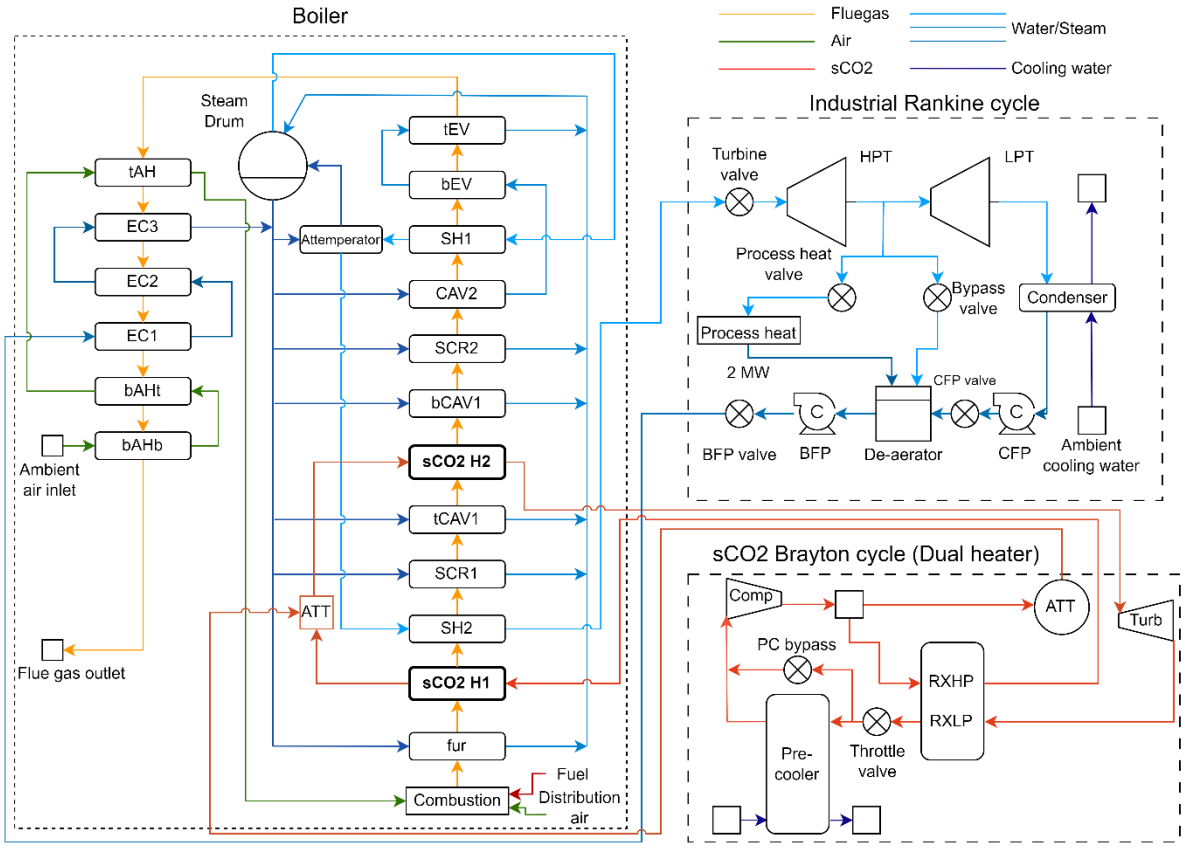


Figure 3.5: Process flow diagram for integrated cycle with dual sCO2 heaters

The high flue gas temperatures (>700 °C) allow for high maximum sCO2 Brayton cycle temperatures. Consequently, three turbine inlet temperatures (TIT) were initially explored, namely 550 °C, 650 °C and 720 °C. The maximum TIT was limited by incoming gas temperatures. At TITs greater than 720 °C, the terminal temperature difference (TTD) across the tube bank reduced significantly. A parametric study was conducted for the three TITs using initial sCO2 heater specifications. The sCO2 heater specifications were then tuned for the selected 550 °C TIT, shown in Table 3.2.

An iterative design process was used to select heater specifications to ensure that the Rankine cycle performance is not adversely impacted. TP347HFG was selected for the sCO2 tube bank material, typically used for superheaters in supercritical boilers [63]. The tube outer diameter and thickness are selected from standard ASTM sizes based on an allowable material stress (30 MPa fluid pressure) at 700 °C [64]. Figure 3.6 shows the layout and geometry of a horizontal boiler tube bank heat exchanger, used for the convective sCO2 heater (sCO2HX/H2) as well as other in-line tube bank heat exchangers such as SH1, evaporators, and economisers. Similarly, Figure 3.7 shows a pendant heat exchanger, used for the radiative sCO2 heater (sCO2H1) and SH2.

Table 3.2: sCO2 heater specifications

	W_{avg} (m)	H_{avg} (m)	L_{duct} (m)	N_t	N_l	S_t (m)	S_l (m)	d_{OD} (m)	t_{wall} (m)	A_{ho} (m^2)
Single heater										
sCO2HX	1.535	4.41	4.833	7	81	0.2	0.06	0.0381	0.00711	298.85
Dual heaters										
sCO2H1	4.41	4.272	0.12	22	3	0.2	0.042	0.0381	0.00711	20.83
sCO2H2	1.535	4.41	3.40	7	43	0.2	0.08	0.0381	0.00711	251.27

The transverse pitch, S_t, is fixed at 0.2 m due to boiler spacing constraints. For the dual heater configuration, the width and height of sCO2H1 is based on the geometry of SH2. The geometry for sCO2H2 is limited by the size of CAV1, particularly the width and height of the heat exchanger tube bundle. The duct length, L_{duct}, and longitudinal pitch, S_l, were adjusted iteratively for both sCO2H1 and sCO2H2. sCO2H1 is modelled as a platen heat exchanger due to the small longitudinal pitch compared to the tube outer diameter. For the single heater configuration, similar to sCO2H2, the geometry is limited to the space available in CAV1, and the same geometric parameters are iteratively determined (L_{duct}, S_l). By adjusting L_{duct} for sCO2HX and sCO2H2, the lengths of tCAV1

and bCAV1 are also adjusted. For the conventional cycle without the sCO2 integration, CAV1 is equally split to form tCAV1 and bCAV1.

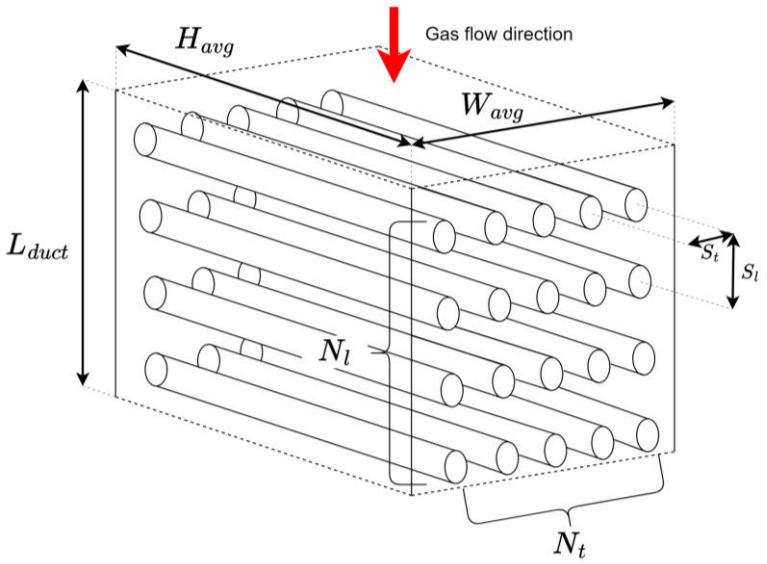


Figure 3.6: Horizontal boiler heat exchanger layout and geometry

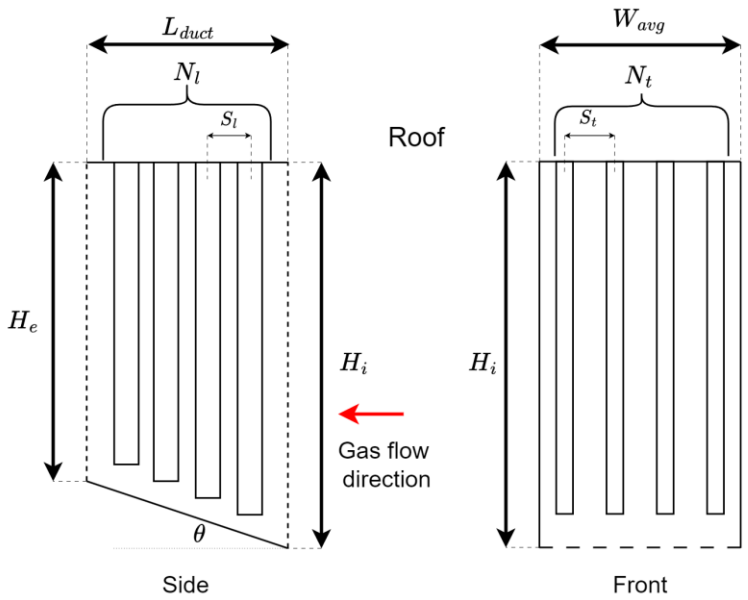


Figure 3.7: Pendant heat exchanger layout and geometry

For comparison purposes, the heater geometries are determined to ensure that both heater configurations are simulated with similar fuel flow rates. The objective of the iterative design process for the sCO2 heaters was to maximise the heat uptakes at the sCO2 heaters while minimising impacts on Rankine cycle boiler heat exchangers. To ensure Rankine cycle performance

was unaffected, the nominal steam temperature at the outlet of SH2 (485 °C) had to be reached for a given sCO2 heater configuration. This included a sufficient margin for attemperation between SH1 and SH2 of at least 300 kW (or ≈ 10 °C at low load). Heater configurations were then finalised by iterating the tube bank geometric parameters (L_{duct}, S_l) to maximise sCO2 heat uptakes with these constraints in place. Detailed heat exchanger geometries are specified for all boiler heat exchangers, including the proposed sCO2 heaters, as shown in Table 3.2. Heat exchanger surface areas are computed from the provided boiler heat exchanger geometry. External areas (A_{ho}) for all boiler heat exchangers are provided in Appendix B for reference.

3.2 1D Thermofluid process model

The analysis is based on a thermofluid network modelling approach as described by Laubscher and De Villiers [10] and Rousseau et al. [54]. To analyse the integrated performance of a large system, each of the components, such as pipes, valves, heat exchangers, pumps, and turbines, are approximated as one-dimensional (1D) elements having one inlet and one outlet. The different elements are connected at nodes to form a thermofluid network as shown in Figure 3.8. The analysis consists of the simultaneous solution of (i) the balance equations of mass, energy, and momentum written between the inlet and outlet of each element; (ii) the component characteristics of each element such as the rate of heat transfer, the power input or output, and the pressure drop or pressure rise; (iii) the specific fluid property relations, and (iv) the boundary values at the boundaries of the network. The balance equations applied here are obtained from the differential form by integrating over the length of each element in the flow direction between the inlet and outlet, while assuming uniform properties over the cross-section. An element may also represent a single increment within a physical component that is discretised into a number of control volumes, if more detail is required. Heat transfer elements are used to facilitate heat transfer between two elements, as shown in Figure 3.8. Boundary conditions are enforced at boundary nodes by specifying pressure, temperature/enthalpy, or mass source/sinks. The thermofluid network modelling approach has been validated with site data for industrial-scale biomass-fired and coal-fired boilers [10], [54].

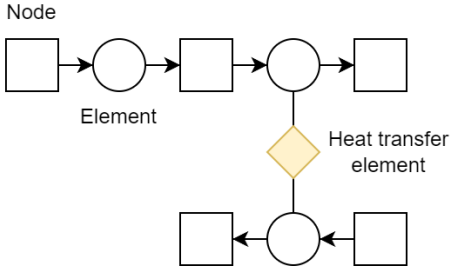


Figure 3.8: 1D network comprising of nodes, elements, and heat transfer elements

The differential forms of the 1D steady-state mass, energy, and momentum balance equations are given respectively as:

$$\frac{\partial(\rho v)}{\partial x} = 0 \tag{3.1}$$

$$\frac{\partial(\rho v(h + \frac{1}{2}v^2 + gz))}{\partial x} = q''' - w''' \tag{3.2}$$

$$\rho v \frac{\partial v}{\partial x} + \frac{\partial p}{\partial x} + \rho g \frac{\partial z}{\partial x} = \frac{\partial p_{0,M}}{\partial x} - \frac{\partial p_{0,L}}{\partial x} \tag{3.3}$$

where x is aligned with the flow direction between the inlet and outlet. The energy balance equation can be written in terms of total enthalpy $h_0 = h + \frac{1}{2}v^2 + gz$, and the momentum balance equation in terms of total pressure $p_0 = p + \frac{1}{2}\rho v^2 + \rho gz$. By integrating over the length of each element in the flow direction between the inlet and outlet Eqs. (3.4) to (3.6) are derived.

$$\sum \dot{m}_e - \sum \dot{m}_i = 0 \tag{3.4}$$

$$\sum \dot{m}_e h_{0e} - \sum \dot{m}_i h_{0i} = \dot{Q} - \dot{W} \tag{3.5}$$

$$p_{0e} - p_{0i} = \Delta p_0 \tag{3.6}$$

with $\Delta p_0 = \Delta p_{0,M} - \Delta p_{0,L}$.

For gas mixtures, Eq. (3.7) is used to ensure species balance for each species constituent, where subscript j is a species constituent.

$$\sum \dot{m}_e Y_{e_j} - \sum \dot{m}_i Y_{i_j} = 0 \tag{3.7}$$

The thermofluid network model therefore solves the spatially integrated 1D incompressible steady state mass, energy, momentum, and species balance equations (Eqs (3.4)-(3.7) respectively) for the networks shown in the process flow diagrams (Figure 3.4 and Figure 3.5). Component characteristics are solved at elements, including $\dot{Q}$ and $\dot{W}$ for the energy balance, and Δp_0 for momentum balance. Additionally, mass flow rates are updated at elements. The custom-built 1D thermofluid network solver was developed in Python 3.7.2, leveraging NumPy, SciPy and Pandas libraries. Fluid properties are computed at nodes using the real-fluid CoolProp library [65].

The energy balance Eq. (3.5) results in a set of simultaneous equations that can be solved for the total enthalpy at each node. For momentum balance, the component characteristics are written in a generic linearised form of

$$\Delta p_0 = a_0 + a_1 \dot{m} \tag{3.8}$$

This is substituted into the momentum balance Eq. (3.6) to get

$$\dot{m} = \frac{p_{0e} - p_{0i} - a_0}{a_1} \tag{3.9}$$

This in turn is substituted into the mass balance Eq. (3.4) to get a set of simultaneous equations that can be solved for the total pressure at each node, which also satisfies the mass balance. The generic linearised form used to capture component characteristics, a_0 and a_1 are computed as follows:

$$a_1 = \frac{\partial}{\partial \dot{m}}(\Delta p_0) \tag{3.10}$$

$$a_0 = \Delta p_0 - a_1 \dot{m} \tag{3.11}$$

For turbomachinery components, polynomial equations are used to compute the pressure ratio for a given non-dimensional flow rate, and in turn, used to compute Δp_0. For heat exchangers and pipes, the total pressure change is calculated using:

$$\Delta p_0 = -\frac{K_{loss}}{2\rho_{avg}A^2}|\dot{m}|\dot{m} \tag{3.12}$$

with K_{loss} a non-dimensional pressure loss factor determined by the specific characteristics of each component. The equations for Δp_0 are then used to compute the generic linearised component characteristics in Eq. (3.10) and Eq. (3.11).

An iterative solution procedure is used to enforce the balance equations for each flow stream. Figure 3.9 shows the solver process flow diagram. Using initial guesses and specified boundary values, component characteristics are updated for each element, followed by heat transfer elements. Momentum balance and energy balance matrix equations are constructed based on a network connectivity table, which identifies which nodes each element is connected to. A similar connectivity table is setup for heat transfer elements These matrix equations are of the form $A\mathrm{x} = \mathbf{b}$, where A is the balance equation matrix, $\mathbf{b}$ the source term vector, and x a vector of pressures or enthalpies at nodes. The balance equation matrix is inverted, and the matrix equation solved using the Scipy *linalg* module. Relative residuals are then computed for pressures, enthalpies, and mass flows. Relaxation is applied in the solution of node enthalpies and pressures to ensure numerical stability, and node pressures and enthalpies are updated for non-boundary nodes. Fluid properties are then updated using the newly calculated enthalpies and pressures through CoolProp, and the solver repeats till convergence. A relative convergence of 1E-6 is used for pressures, and 1E-4 for enthalpies.

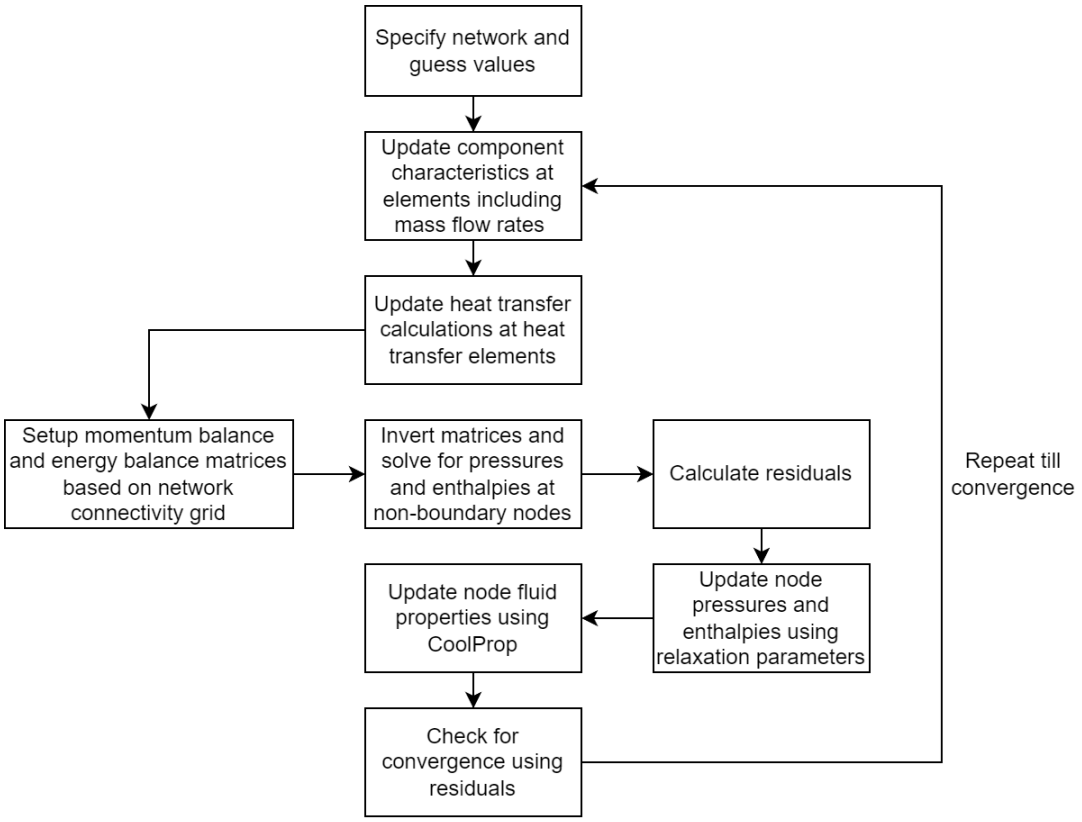


Figure 3.9: Solver process flow diagram

The thermofluid solver has been thoroughly verified. Component characteristic sub-component models have been verified using similar models developed in Mathcad and Python, with agreement within 2%. Additionally, the implementation of the balance equations has also been verified using Flownex SE® 2022, which is a commercial 1D thermofluid solver. Unfortunately, the current model could not be validated via comparison with plant measurements for this study since the plant has not yet been constructed. However, a detailed CFD analysis is used to verify the heat uptakes of boiler heat exchangers and to further calibrate the 1D integrated model, as described in Chapters 5 and 6.

The following sub-sections describe the component characteristics and modelling methodology for the cycle components shown in the process flow diagrams in Figure 3.4 and Figure 3.5

3.2.1 Boiler

3.2.1.1 *Combustion*

The combustion model is based on the assumption of instantaneous combustion (i.e. not accounting for the kinetics) with an assumed fraction of unburned carbon. The inputs to the combustion model

include the HHV, fuel composition, fuel temperature, fuel mass flow rate, air temperature and absolute humidity, fly-ash fraction, excess air ratio, and the fractional distribution of air across primary, secondary, and distribution streams. The combustion model computes the mass fractions of flue gas, the mass flow rates of combustion air, flue gas, and fly ash, as well as the adiabatic flame temperature [54].

$$\begin{aligned} &C_{n_{fC}}H_{n_{fH}}O_{n_{fO}}N_{n_{fN}}S_{n_{fS}}M_{n_{fM}} + n_{rH2O}H_2O + n_{rO2}O_2 + n_{rN2}N_2 \\ &\rightarrow n_{pCO2}CO_2 + n_{pN2}N_2 + n_{pSO2}SO_2 + n_{pH2O}H_2O + n_{pO2}O_2 \end{aligned} \tag{3.13}$$

$$\begin{aligned} &n_{rH2O} = 4.7619\alpha\omega\frac{M_{air}}{M_{H2O}}n_{stO2}; \qquad n_{rO2} = \alpha n_{stO2}; \qquad n_{rN2}=3.7619\alpha n_{stO2} \\ &n_{pCO2} = n_{fC}; \qquad n_{pN2} = \frac{1}{2}n_{fN} + 3.7619\alpha n_{StO2}; \qquad n_{pSO2} = n_{fS} \\ &n_{pH2O} = n_{fM} + \frac{1}{2}n_{fH2} + 4.7619\alpha\omega\frac{M_{air}}{M_{H2O}}n_{StO2}; \qquad n_{pO2} = (\alpha - 1)(n_{StO2}) \\ &n_{StO2} = n_{fC} + \tfrac{1}{4}n_{fH} - \tfrac{1}{2}n_{fO} + n_{fS} \end{aligned} \tag{3.14}$$

Mass fractions of the bagasse from Table 3.1 and excess air ratio are used to calculate the resulting flue gas mixture composition. The flue gas mixture produced during combustion consists of H_2O, CO_2, N_2, SO_2, and O_2. A mass balance between the reactants and products of the combustion reaction is performed by balancing the molar amounts of reactants and products per kilogram of fuel shown in Eq. (3.13). The reactants include fuel (ash-free) and combustion air, accounting for moisture in the fuel and air. Molar amounts of combustion air are computed for stoichiometric combustion and adjusted for excess air and moisture. The mass fraction constituents of the flue gas are then computed based on the molar fractions shown in Eq. (3.14). Additionally, the humid air ratio (HAR) and flue gas ratio (FGR) are calculated per kilogram fuel in Eq. (3.15), and the mass flow rate of combustion air, flue gas, and fly ash in Eq. (3.16). Fly-ash is not explicitly included in the gas-mixture, however, is accounted for during radiation heat transfer as well as energy balances in the furnace and downstream heat exchangers.

$$\begin{aligned} &HAR = \sum_j N_{CA_j}M_j;\ [kg_{CA}\ /\ kg_{fuel}] \\ &FGR = \sum_j N_{FG_j}M_j;\ [kg_{FG}\ /\ kg_{fuel}] \end{aligned} \tag{3.15}$$

$$\begin{aligned} &\dot{m}_{ca} = HAR\ \dot{m}_{fuel},\ \dot{m}_{FG} = FGR\ \dot{m}_{fuel} \\ &\dot{m}_{fa} = Y_{fa}(Y_{fuel_{ash}} + Y_{fuel_{UC}})\dot{m}_{fuel} \end{aligned} \tag{3.16}$$

An energy balance is conducted over the combustion zone control volume to determine the adiabatic flame temperature, assuming all the combustion heat is released in the furnace control volume. Eq. (3.17) shows the energy balance equation.

$$\dot{m}_{FG} h_{FG_{AFT}} + \dot{m}_{FA} c_{P_{ash}} (T_{AFT} - T_{ref}) = \dot{m}_{fuel} (c_{P\,fuel} (T_{fuel} - T_{ref}) + HHV_{fuel} - Y_{fuel_{UC}} HHV_C) + \dot{m}_{PA} h_{PA} + \dot{m}_{SA} h_{SA} + \dot{m}_{DA} h_{DA} \quad (3.17)$$

As the adiabatic flame temperature (T_{AFT}) is initially unknown, an iterative procedure is used to satisfy Eq. (3.17), accounting for the flue gas and fly ash exiting the combustion control volume, as well as fuel, unburned char, and various air supply streams entering the control volume. The constants in Eq. (3.17) are shown in Table 3.3. The enthalpy of flue gas exiting the boiler ($h_{FG_{AFT}}$) is derived from T_{AFT} based on the appropriate real gas mixture properties. Primary and secondary air streams are assumed to come from the air heaters and enter the combustion control volume together. Ambient distribution air also feeds into the combustion control volume. The fractions of these air streams are shown in Table 3.3, as specified by the manufacturer for operation of the MicroGen® boiler and applied for all cycle configurations. The mass flow rates of the air streams are then computed using the respective fraction split and the total combustion air mass flow rate, $\dot{m}_{ca}$.

Table 3.3: Constants used in Eq (3.17) and air stream fractions

$c_{P,ash}$	$c_{P,fuel}$	T_{ref}	HHV_C	f_{PA}	f_{SA}	f_{DA}
710 J/kgK	1200 J/kgK	298.15K	32763 kJ/kg	0.508	0.402	0.090

3.2.1.2 *Furnace*

The furnace is dominated by radiative heat transfer to the furnace water walls. The Gurvich method [66] is used to calculate the projected radiative heat transfer from hot gases and particles to the furnace water walls, as well as the furnace exit temperature. Eq. (3.18) shows the empirical formula for calculating the furnace exit gas temperature.

$$T_{FG,FE} = \frac{T_{AFT}}{M_{flame}\left[\frac{\varepsilon_{eff,fur}}{Bo}\right]^{0.6} + 1} \quad (3.18)$$

The flame modification factor M is calculated as:

$$M = A' - B'(X_r + \Delta X) \quad (3.19)$$

In Eq. (3.19), for the present study with a grate-fired bagasse boiler, $A' = 0.59$, $B' = 0.59$, and $(X_r + \Delta X) = 0.14$ [67]. Typically, the flame modification factor may be tuned further based on experimental data or CFD simulations for a particular boiler. Bo, the Boltzmann number, in Eq. (3.18) is calculated as:

$$Bo = \frac{\phi_{fur}\dot{m}_{fuel}\overline{VC}}{\sigma_0 \psi A_{rad,fur} T_{AFT}^3} \tag{3.20}$$

In Eq. 10, $\overline{VC}$ is the mean overall heat capacity of flue gas per kg of fuel, $\phi_{fur} = 1 - f_{radloss}$ where $f_{radloss} = 0.0101$, $A_{rad,fur}$ is the projected radiative furnace area accounting for the angular coefficient at the tube walls, refractory walls, furnace exit, and grate. ψ is the area weighted furnace efficiency factor, accounting for the angular coefficient, geometric arrangement, and fouling factors of furnace surfaces. The furnace efficiency factor is calculated using Eq. (3.21).

$$\psi = x_{ang}\xi \tag{3.21}$$

Where x_{ang} are the angular coefficient and ξ furnace wall fouling factor. For membrane water walls, x_{ang} is calculated below based on empirically fitted data:

$$x_{ang} = 1.15484 \left(\frac{b_{tubewalls}}{d_{tubewalls}}\right)^{-0.850475} \tag{3.22}$$

The furnace wall fouling factor accounts for the ash deposits, which inhibit heat transfer from the gasses to the working fluid. Zhang [24] recommends a value of 0.6 for grate fuels. For refractory, $\xi = 0.1$, and for the furnace exit plane and grate, $\xi = 1$. The area-weighted efficiency factor is then found using:

$$\overline{\psi} = \frac{\sum \psi_i A_{rad,i}}{A_{rad}} \tag{3.23}$$

From Eq. (3.18), $\epsilon_{eff,fur}$ is the effective gas emissivity in the furnace, calculated using the approach by Brummel [68], considering high particulate loading and radiation scattering as recommended by Rousseau et al. [54]. The calculation is detailed below. The furnace gas emissivity is influenced by tri-atomic gas radiation, coke particle radiation, ash particulate absorption and radiation scattering. To characterise the geometry of the furnace heat transfer surface, the mean beam length is used.

$$S_{fur} = \frac{3.6V_{fur}}{A_{fur}} \tag{3.24}$$

Flue gas tri-atomic emissivity is computed using the weighted-sum-of-gray-gas model at a gas temperature T_g and pressure p_{fur} in bar, shown in Eq. (3.25), with coefficients presented in Table 3.4 [54].

$$\varepsilon_g = \sum_{i=1}^{3}[(b_{1,i} + b_{2,i}T_g)(1 - \exp(-k_i(X_{fg,H2O} + X_{fg,CO2})p_{fur}S_{fur})] \tag{3.25}$$

Table 3.4: Coefficients for Eq. (3.25)

i	$b_{0,i}$ [-]	$b_{1,i}$ [1/K]	k_i [m bar]$^{-1}$
1	0.130	$2.65\text{x}10^{-4}$	0.0
2	0.595	$-1.50\text{x}10^{-4}$	0.824
3	0.275	$-1.15\text{x}10^{-4}$	25.91

Consequently, the emission coefficient of the flue gas mixture can be calculated:

$$k_{emi,T_g} = -\frac{\ln(1-\varepsilon_g(T_g))}{S_{fur}} \tag{3.26}$$

Non-dimensional particle absorption and scattering efficiency, Q_{abs} and Q_{bsc} respectively, are empirically determined based on ash particle diameters [68]. A particle diameter of 20 µm is used, typical for grate-fired boilers. Laubscher and Rousseau [69] presented correlations derived from graphs of [68] for Q_{abs} and Q_{bsc}:

$$\begin{aligned} Q_{abs} &= 0.275 d_p^{0.298} - 0.305 \\ Q_{bsc} &= \frac{6.2188\text{x}10^{-3} - 1.0492\text{x}10^{-2} d_p + 7.287\text{x}10^{-3} d_p^2 - 2.1925\text{x}10^{-5} d_p^3}{1.851\text{x}10^{-1} - 2.0405\text{x}10^{-3} d_p^2 + 6.254\text{x}10^{-4} d_p^3} \end{aligned} \tag{3.27}$$

The optical thickness of the gas-particulate mixture is then calculated:

$$\Phi_{gp} = (Q_{abs} A_p L_p + k_{emi,T_g} + k_{coke} x_1 x_2) S_{fur} \gamma \tag{3.28}$$

where the particulate area $A_p = \frac{3}{2\rho_{ash} d_{fa}}$, particulate loading $L_p = \frac{\dot{m}_{ash}}{\dot{Q}_{fg}}$, scattering factor $\gamma = \left(1 + \frac{2Q_{scat}}{Q_{abs}}\right)^{0.5}$, and $k_{coke} = 10.2\left(\text{m MPa}\right)^{-1}$. x_1 and x_2 are empirical coefficients, set to 0.5 high volatile matter and 0.03 for grate firing respectively. The effective emissivity of the flue gas and particle mixture is calculated as:

$$\varepsilon_{gp} = (1-\beta)\frac{1-\exp(-\Phi_{gp}(T_g))}{1+\beta\exp(-\Phi_{gp}(T_g))} \tag{3.29}$$

where $\beta = \frac{\gamma - 1}{\gamma + 1}$. Finally, the furnace effective emissivity, $\epsilon_{eff,fur}$ from Eq. (3.18) can be calculated using area-weighted furnace efficiency factor from Eq. (3.23), concluding the furnace effective emissivity calculation:

$$\varepsilon_{eff,fur} = \left[1 + \overline{\psi} \left(\frac{1}{\varepsilon_{gp}} - 1 \right) \right]^{-1} \tag{3.30}$$

Returning to the furnace heat transfer calculation, once $T_{FG,FE}$ from Eq. (3.18) has been computed, the heat transfer in the furnace control volume is computed using:

$$\dot{Q}_{FG,fur} = \dot{Q}_{FE} + \dot{Q}_{fur,loss} + \dot{Q}_{WW} + \dot{Q}_{RF} \tag{3.31}$$

$\dot{Q}_{FG,fur}$ is evaluated using an energy balance over the furnace control volume, shown in Eq. (3.32). The energy balance accounts for flue gas and fly-ash particles, using the calculated furnace exit gas temperature and the adiabatic flame temperature calculated at the combustion control volume.

$$\dot{Q}_{FG,fur} = \dot{m}_{FG}(h_{FG,AFT} - h_{FG,FE}) + \dot{m}_{FA}(h_{FA,AFT} - h_{FA,FE}) \tag{3.32}$$

$$\dot{Q}_{FE} = \beta_{FE} \eta_{FE} A_{FE} \frac{\dot{Q}_{fur} \phi_{fur}}{A_{eff,fur}} \tag{3.33}$$

$$\dot{Q}_{RF} = \eta_{RF} A_{RF} \frac{\dot{Q}_{fur} \phi_{fur}}{A_{eff,fur}} \tag{3.34}$$

$$\dot{Q}_{fur,loss} = (1 - \phi_{fur}) \dot{Q}_{fur} \tag{3.35}$$

$\dot{Q}_{FE}$ is the direct radiation leaving at the furnace exit plane and is evaluated using Eq. (3.33), where β_{FE} is an empirical re-radiation coefficient dependent on the furnace exit temperature, η_{FE} is the coefficient of heat flux non-uniformity at the furnace exit determined by empirical curves along the furnace height, A_{FE} is the projected furnace exit area, and $A_{eff,fur}$ the total effective projected furnace area. $\dot{Q}_{RF}$ is the heat transfer at the refractory covered walls in the furnace evaluated using Eq. (3.34). $\dot{Q}_{fur,loss}$ is calculated in Eq. (3.35), and consequently the water wall heat transfer $\dot{Q}_{WW}$ is calculated using Eq. (3.31) with all other variables known.

3.2.1.3 *Boiler heat exchangers*

Boiler heat exchangers include radiative-convective heat exchangers, evaporators, screens, and cavities. Radiative and convective heat transfer is modelled for each heat exchanger, as well as to water walls surrounding the heat exchanger. Flue gas control volumes are explicitly modelled for each heat exchanger, as shown in the process flow diagrams in Figure 3.4 and Figure 3.5. A lumped control volume for the water side, EV, is used to capture heat transfer to the water walls, cavities, evaporators, and screens. Geometric parameters, such as those described for the sCO2 heater in Table 3.2, are specified for each boiler heat exchanger.

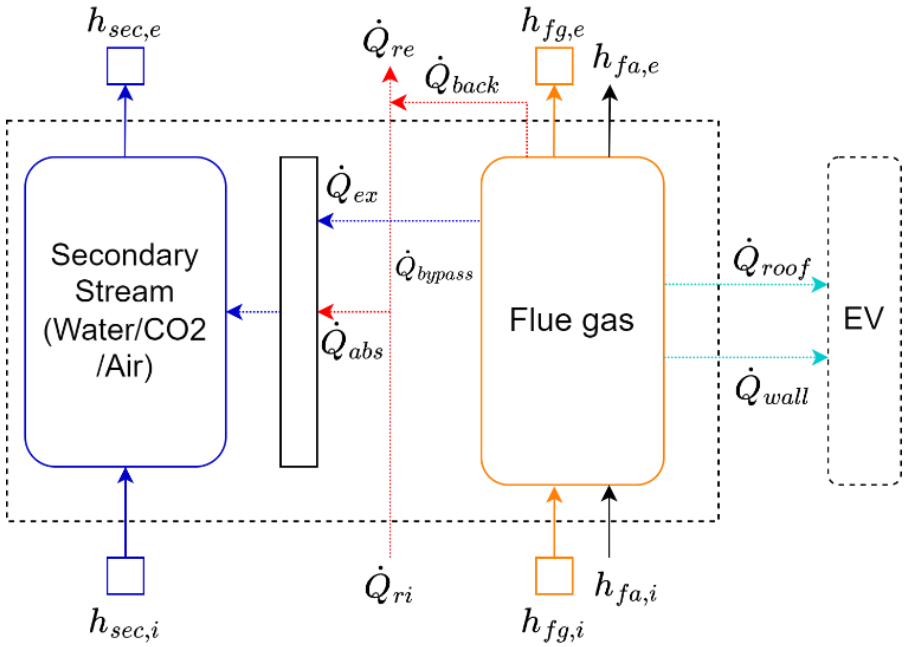


Figure 3.10: Radiative-convective heat exchanger modelling

Radiative-convective tube bank heat exchangers include superheaters (SH1, SH2), sCO2 heaters (sCO2HX, sCO2H1, sCO2H2), economizers (EC1, EC2, EC3), and air heaters (tAH, bAHt, bAHb). For radiative-convective heat exchangers, the secondary fluid stream (water/CO2/air) is explicitly modelled and coupled to the corresponding flue gas control volume for heat transfer modelling. Figure 3.10 shows the generic heat transfer flow diagram for radiative-convective heat exchangers.

Fly-ash in the gas stream is accounted for in the energy balance for the flue gas volume using the inlet and outlet flue gas temperatures, $c_{P,ash} = 710$ J/kgK, and $\dot{m}_{fa}$ calculated at the combustion control volume. For flue gas, the gas and particulate mixture emissivity is again calculated using the approach by Brummel [68] described in section 3.2.1.2 for the furnace. However, the mean beam length for boiler heat exchangers and a platen heat exchanger is calculated as:

$$S_{HX} = 0.9 d_{OD} \left(\frac{4 S_t S_l}{\pi d_{OD}^{\ 2}} - 1 \right)$$
$$S_{platen.HX} = \frac{1.8}{\frac{1}{H_{ave}} + \frac{1}{b_l} + \frac{1}{S_t}} \tag{3.36}$$

For radiative cavities, the mean beam length is calculated similarly to the furnace using Eq. (3.24). $\dot{Q}_{ri}$ is the heat transfer rate of incoming direct radiation from up-stream gas flows. For example, at SH2 which is downstream the furnace control volume, $\dot{Q}_{ri,SH2} = \dot{Q}_{FE} = \dot{Q}_{re,fur}$. Absorbed radiation, $\dot{Q}_{abs}$, is the difference between $\dot{Q}_{ri}$ and bypassing radiation, $\dot{Q}_{bypass}$ calculated using Eq. (3.37) [70]. The angular coefficient of incident direct radiation absorbed by the heat exchanger, ϕ_{ang}, is calculated using Eq. (3.38), dependent on heat exchanger depth, b, and S_t. For all heat exchangers downstream of the furnace, $\beta = 1$. For radiative cavities, with no tube bank heat exchangers in the flue gas flow path, $\phi_{ang} = 1$

$$\dot{Q}_{bypass} = \frac{\dot{Q}_{ri}(1 - \varepsilon_{fg})\phi_{ang}}{\beta} \tag{3.37}$$

$$\phi_{ang} = \left[\left(\frac{b}{S_t} \right)^2 + 1 \right]^{1/2} - \frac{b}{S_t} \tag{3.38}$$

The outgoing direct radiation from the heat exchanger, $\dot{Q}_{re}$, is made up of $\dot{Q}_{bypass}$ and additional gas radiation leaving the outlet plane of the heat exchanger, $\dot{Q}_{back}$, calculated using Eq. (3.39).

$$\dot{Q}_{back} = 0.5 \epsilon_{fg} \sigma_0 A_{HXout} T^4_{\ fg,avg} \tag{3.39}$$

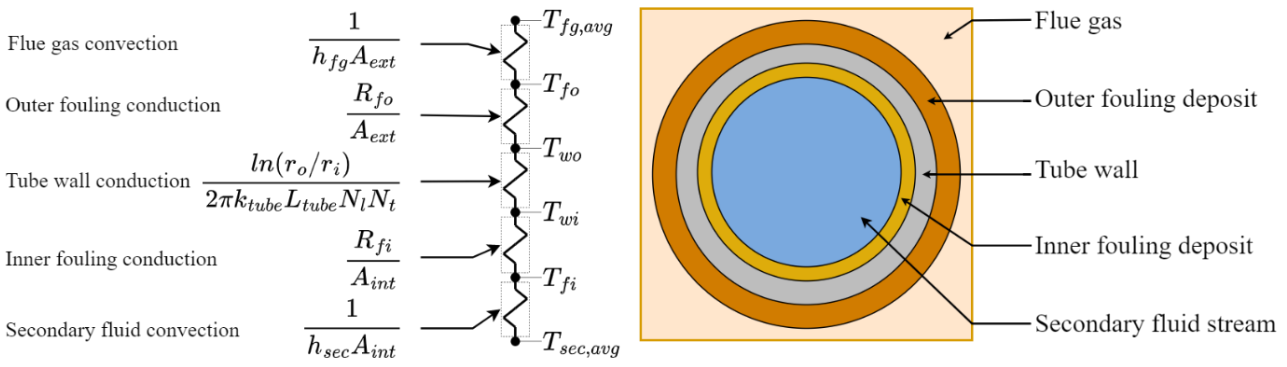


Figure 3.11: Thermal resistance network for radiative-convective heat exchanger tube bank

Heat transfer for radiative-convective heat exchangers is dominated by two mechanisms, namely gas and particulate radiation, and forced convection from the flue gas surrounding the tube bank. Figure 3.11 shows the thermal resistance network that characterise the heat exchanger tubes.

The combined flue gas external heat transfer coefficient, h_{fg}, is the sum of the convective heat transfer coefficient, $h_{fg,conv}$, and gas radiative heat transfer coefficient, $h_{fg,rad}$. $h_{fg,rad}$ is calculated using the radiative heat flux from the flue gas to the outer fouling layer of the heat exchanger tubes.

$$\begin{aligned} h_{fg,rad} &= \frac{q_{fg,rad}}{T_{fg,avg} - T_{fo}} \\ q_{fg,rad} &= \frac{\epsilon_{fo}+1}{2}\sigma_0 \epsilon_{fg,i}(T^4{}_{fg,avg} - T_{fo}^4) \end{aligned} \tag{3.40}$$

ϵ_{fo} is the emissivity of the outer ash layer on the tubes, set to 0.8, typically used for solid-fuel fired boilers [67]. The external forced convection heat transfer coefficient, $h_{fg,conv}$, is calculated using correlations by Gnielinski [71]. Eq. (3.41) computes the gas void fractions through the tube bank. Eq. (3.42) computes the Reynolds number using the open duct velocity adjusted for the gas void fraction. Additionally, the effect of oblique flow over the tube bank is accounted for through the $\cos(\theta)$ term, particularly for SH2 and the radiative sCO2 heater sCO2H1.

$$\begin{aligned} a &= S_t \big/ d_{OD}; \quad b = S_l \big/ d_{OD} \\ \psi &= \begin{cases} 1 - \dfrac{\pi}{4a} & b \geq 1 \\ 1 - \dfrac{\pi}{4ab} & b < 1 \end{cases} \end{aligned} \tag{3.41}$$

$$\begin{aligned} v_{fg,open} &= \frac{\dot{m}_{fg}}{\rho_{fg,avg} A_{open}} \\ L_{Gni} &= \frac{\pi}{2} d_{OD} \\ \mathrm{Re}_{\psi} &= \frac{\rho_{fg,avg} v_{fg,open} L_{Gni}}{\psi \mu_{fg,avg}} \cos(\theta) \end{aligned} \tag{3.42}$$

Eq. (3.43) calculates the laminar-turbulent smoothed Nusselt number for a single heat exchanger row. Eq. (3.44) corrects the single row Nusselt number, accounting for an in-line tube bank arrangement and for the depth of the tube bank.

$$
\begin{aligned}
Nu_{l,lam} &= 0.664\sqrt{\mathrm{Re}_{\psi}}\,\mathrm{Pr}_{fg,avg}^{\frac{1}{3}} \\
Nu_{l,turb} &= \frac{0.037\mathrm{Re}_{\psi}^{\,0.8}\,\mathrm{Pr}_{fg,avg}}{1+2.443\mathrm{Re}_{\psi}^{\,-0.1}(\mathrm{Pr}^{2/3}-1)} \\
Nu_{l,0} &= 0.3+\sqrt{Nu_{l,lam}^{\;2}+Nu_{l,turb}^{\;2}}
\end{aligned}
\tag{3.43}
$$

$$
\begin{aligned}
f_A &= 1+\frac{0.7(\frac{b}{a}-0.3)}{\psi^{1.5}(\frac{b}{a}+0.7)^2} \\
Nu_{0,bundle} &= Nu_{l,0}\,\frac{1+(N_l-1)f_A}{N_l}
\end{aligned}
\tag{3.44}
$$

Lastly, the effect of the difference in the bulk flue gas temperature and outer fouling surface temperature on fluid properties is accounted for in Eq (3.45).

$$
\begin{aligned}
Nu_{bundle} &= Nu_{0,bundle}\left(\frac{T_{fg,avg}}{T_{fo}}\right)^{0.12} \\
h_{fg,conv} &= \frac{Nu_{bundle}k_{fg}}{L_{Gni}}
\end{aligned}
\tag{3.45}
$$

For radiative cavities, with no tube bank in the gas flow path, radiation is assumed to be the dominant heat transfer mechanism to the water walls surrounding the cavity, with convection assumed negligible. This assumption is made as majority of the gas flow path is unobstructed, and the water walls form a relatively small part of the overall control volume. Consequently, $h_{fg,conv,CAV}=0$. The effective convective/radiative fluegas heat transfer coefficient is then:

$$h_{fg}=h_{fg,rad}+h_{fg,conv} \tag{3.46}$$

For the platen-type sCO2H1, where tubes are stacked closely in the flow direction, the projected flat surface area formed by the tubes are used for the external heat transfer area. When calculating h_{fg} for the platen heat exchanger, an area correction factor is applied accounting for the total wetted external surface area as opposed to the projected flat surface area, shown in Eq. (3.47).

$$h_{fg,platen}=h_{fg,rad}+h_{fg,conv}\left(\frac{\pi d_{OD}}{2S_l}\right) \tag{3.47}$$

For internal forced convection of the secondary fluid, h_{sec}, the Gnielinski turbulent convection correlation is used [72]. The correlation is valid for $3\times10^3<\mathrm{Re}<5\times10^6$, and applicable for water, air, and CO2 flow streams.

$$
\begin{aligned}
f &= (0.79\ln(\mathrm{Re}) - 1.64)^{-2} \\
Nu_{conv,\mathrm{int}} &= \frac{\frac{f}{8}(\mathrm{Re}-1000)\mathrm{Pr}}{1+12.7(\frac{f}{8})^{0.5}(\mathrm{Pr}^{2/3}-1)} \\
h_{\mathrm{sec}} &= \frac{Nu_{conv,\mathrm{int}} k_{\mathrm{sec},avg}}{d_{ID}}
\end{aligned}
\tag{3.48}
$$

For evaporator heat exchangers, the convective thermal resistance of the bulk water volume, EV, is assumed negligible as the water is evaporating. For cavities, where there is no secondary flow stream explicitly modelled, only the direct radiation and wall heat transfer is accounted for.

An outer and inner fouling layer around the tube walls is accounted for in the model. R_{fo} is the thermal resistance due to ash deposits on the tube bank, set to $0.0043\ \frac{\mathrm{m^2K}}{\mathrm{W}}$ [67]. The inner fouling thermal resistance is assumed to be negligible. For air heaters, a utilization factor, $\psi_{HX,AH} = 0.9$, is applied to impose fouling thermal resistances, while the utilization factor is set to 1 for other boiler heat exchangers. A balance between the internal heat transfer rate to the secondary fluid, $\dot{Q}_{\mathrm{sec}}$, and the external heat transfer rate from the flue gas and absorbed direct radiation, $\dot{Q}_{ex}$ and $\dot{Q}_{abs}$ is obtained via iteration while varying the outer surface temperature of the fouling T_{fo}. The external overall heat transfer coefficient and effectiveness-NTU method is used to calculate $\dot{Q}_{ex}$ as shown in Eq. (3.49).

$$
\begin{aligned}
UA_{fg} &= \eta_o h_{fg} A_{ext} \\
NTU_{fg} &= \frac{UA_{fg}}{\dot{m}_{fg} c_{p.fg,avg}} \\
\varepsilon_{HXfg} &= 1 - e^{-NTU_{fg}} \\
\dot{Q}_{ex} &= \epsilon_{HXfg} \dot{m}_{fg} c_{p.fg,avg} (T_{fg,i} - T_{fo})
\end{aligned}
\tag{3.49}
$$

Similarly, $\dot{Q}_{\mathrm{sec}}$ is calculated using the thermal resistances from the outer surface of the fouling to the secondary fluid as shown in Figure 3.11, including outer and inner fouling layers, and tube wall conduction.

$$UA_{\text{sec}} = \frac{\psi_{HX}}{\frac{1}{\eta_{\text{int}} h_{\text{sec}} A_{\text{int}}} + \frac{R_{fi}}{\eta_{\text{int}} A_{\text{int}}} + \frac{\ln(r_o / r_i)}{2\pi k_{tube} L_{tube} N_l N_t} + \frac{R_{fo}}{\eta_{ext} A_{ext}}}$$

$$NTU_{\text{sec}} = \frac{UA_{\text{sec}}}{\dot{m}_{sec} c_{p.\text{sec},avg}} \tag{3.50}$$

$$\varepsilon_{HXsec} = 1 - e^{-NTU_{sec}}$$

$$\dot{Q}_{sec} = \varepsilon_{HXsec} \dot{m}_{sec} c_{p.\text{sec},avg} (T_{fo} - T_{\text{sec},i})$$

where A_{int} is the tube internal surface area, A_{ext} is the tube external surface area, and ψ_{HX} is the heat exchanger utilisation factor. Note for the platen-type sCO2H1, A_{ext} is adjusted to account for the tubes being modelled as flat plates as previously discussed.

Lastly, the outer and inner tube wall metal temperatures, T_{wo} and T_{wi} are computed using the respective thermal resistances and $\dot{Q}_{sec}$.

$$T_{wo} = T_{fo} - \frac{R_{fo}}{\eta_{ext} A_{ext}} \dot{Q}_{sec}$$
$$T_{wi} = T_{wo} - \frac{\ln(r_o / r_i)}{2\pi k_{tube} L_{tube} N_l N_t} \dot{Q}_{sec} \tag{3.51}$$

The heat transfer rate to the water walls, $\dot{Q}_{wall}$, and roof, $\dot{Q}_{roof}$ is calculated using the radiative and convective heat transfer coefficients calculated for flue gas across the tube bank. However, to account for the reduced gas flow velocities surrounding the water walls, the heat transfer coefficient is corrected as shown in Eq. (3.52) using tuning coefficients derived from preliminary CFD analyses. The preliminary CFD analyses modelled the convective pass in the boiler, including the convective sCO2 heater. The calculated heat transfer coefficients at the waterwalls were then used to derive the tuning coefficients. The tuning coefficients used are $\beta_{conv} = -0.678$ and $\beta_{rad} = 0.375$. Consequently, Eq. (3.53) is used to calculate $\dot{Q}_{wall}$ and $\dot{Q}_{roof}$. A_{wall} is the water wall/roof area surrounding the tube bank, and T_{wall} is the outside surface temperature of the water walls of the evaporating circuit, which is assumed to be equal to the saturation temperature of the steam, i.e. 285°C.

$$h_{fg,wall} = (1 + \beta_{conv}) h_{fg,conv} + (1 + \beta_{rad}) h_{fg,rad} \tag{3.52}$$

$$\dot{Q}_{wall} = h_{fg,wall} A_{wall} (T_{fg,avg} - T_{wall}) \tag{3.53}$$

For tubular air heaters, the internal Gnielinski correlation is used for flue gas, while the external flow correlations are used for the air stream. Economisers are modelled as radiative-convective heat exchangers as described above. The air heaters and economisers are not surrounded by water walls.

The steam drum separates the incoming flow from the economiser into vapour and liquid streams, modelled as a simple mixture node with two-phase separation. The level control in the drum is simulated by specifying a fixed quality within the drum. An external control loop is used to ensure energy balance across the steam drum while maintaining the drum pressure at 6.914 MPa, explained further in Section 3.2.4. The attemperator is used to ensure a fixed outlet temperature at SH2 of 485 °C by extracting heat as required between SH1 and SH2. The required $\dot{Q}_{ATT}$ is found iteratively via an external control loop.

3.2.2 Rankine cycle

The component characteristics for the turbomachinery, including the CFP, BFP, HPT, and LPT, are specified via generic performance curves for the pressure ratio and isentropic efficiency versus normalized mass flow rate. Second-order polynomial curves of the form $a_0 + a_1 CM_T + a_2 CM_T^2$ are used for Rankine cycle turbines, and $a_0 + a_1 \dot{m} + a_2 \dot{m}^2$ for the pumps. These performance curves determine the Rankine cycle feedwater flowrate, provided by the BFP, as well as the performance of the turbomachinery through the isentropic efficiency performance curves. For turbines, the isentropic efficiency at the nominal operating point is designed to be 0.7 and 0.6 for the HPT and LPT respectively and 0.65 for pumps, observed in literature for similar steam conditions and cycle size [52], [73], [74]. Valve components are modelled as simple 1D pipe components with a secondary loss factor, K_{valve}. For various valves, K_{valve} is iteratively tuned to satisfy a selected target function which in reality would be achieved via a controller.

In this study, an assumed fixed amount of process heat, 2 MW, is extracted at the process heat line, with the flow returning from the process as saturated liquid. The external control loop ensures an energy balance for the process heat line. The de-aerator is modelled like the steam drum, with only a saturated liquid outflow. The pressure at the de-aerator is fixed at 835 kPa, and a mass balance across the de-aerator is enforced via an external control loop, discussed further in Section 3.2.4. The steam drum pressure is 6.9 MPa, and respective saturation temperature is 285 °C. The condenser is modelled as a heat exchanger with a fixed UA value. The ε-NTU method is used to model heat transfer between the incoming steam and ambient cooling water in a counter-flow arrangement.

3.2.3 Brayton cycle

The turbine for the Brayton cycle is also modelled using generic performance characteristic curves for pressure ratio and isentropic efficiency, captured using the normalized mass flow rate shown in Eq. (3.54), through 4th order polynomials, $a_0 + a_1 CM_T + a_2 CM_T^{\,2} + a_3 CM_T^{\,3} + a_4 CM_T^{\,4}$. Figure 3.12 and Figure 3.13 show the performance curves for the sCO2 turbine and compressor respectively. The development of these curves were informed by sCO2 turbomachinery curves observed in literature for a wide range of operating conditions [75]. Detail design of the turbomachines is outside the scope of this study. For the proposed integrated cycle, sCO2 turbomachinery will be grid-synchronised, hence constant speed curves are used.

$$CM_T = \frac{\dot{m}\sqrt{T_{0i}}}{p_{0i}} \tag{3.54}$$

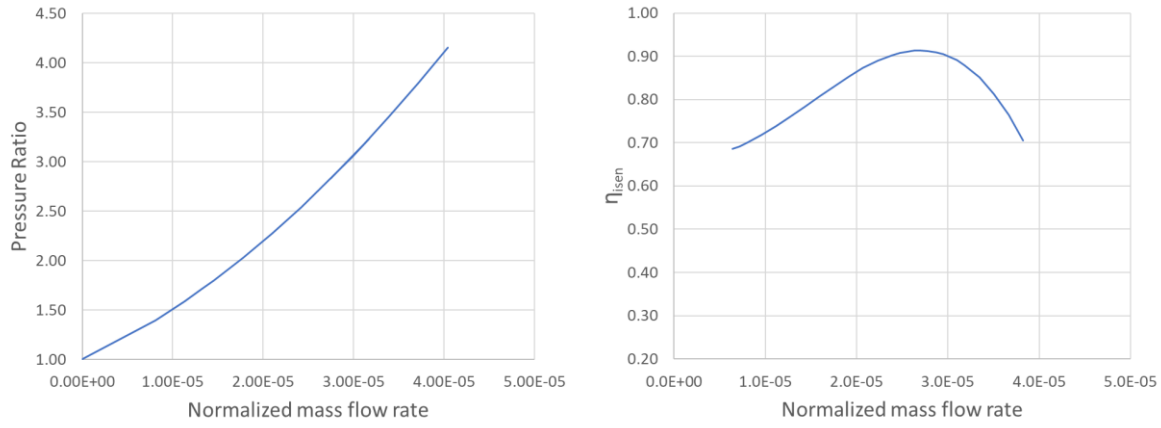


Figure 3.12: sCO2 Turbine performance curves

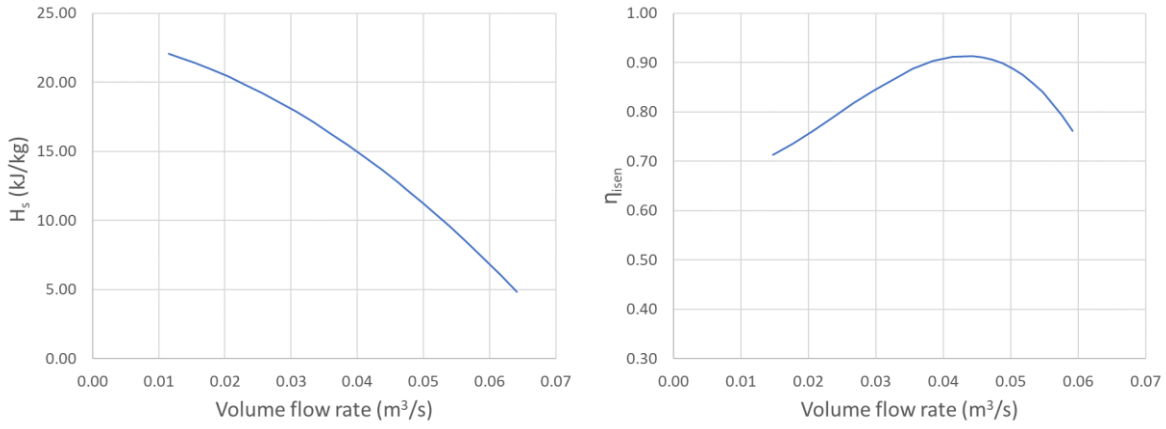


Figure 3.13: sCO2 Compressor performance curves

For the compressor, generic performance characteristic curves are used for the isentropic head and isentropic efficiency against volume flow rate in the form $a_0 + a_1 Q_C + a_2 Q_C^{\,2} + a_3 Q_C^{\,3} + a_4 Q_C^{\,4}$, where Q_C is the compressor volume flow rate. This formulation takes the inlet density to the compressor into account, which is important to consider for highly non-ideal gas behaviour of sCO2 near the critical point. The approach has been used for comparisons to experimental data for various inlet conditions with reasonable success [76].

The compressor pressure ratio performance curve is specified to ensure a pressure ratio of 2.5 at nominal load. The nominal mass flow rate for the Brayton cycle is selected to ensure the chosen turbine inlet temperature (550°C/650°C/720°C). Similarly, isentropic efficiency curves are specified with nominal full load efficiency of 0.9 for both compressor and turbine. A compressor inlet pressure of 10 MPa is selected at nominal load operating conditions. This pressure needs to be high enough to ensure that at lower loads the compressor inlet pressure will not be below the sCO2 psuedo-critical point (≈7.4 MPa) resulting in liquid entering the compressor. The same set of performance curves are used for both heater configurations.

The recuperator and pre-cooler are both modelled using a pre-defined overall conductance, *UA* value, and detailed geometry is not considered. Due to the large variation in CO2 fluid properties near the critical point, it is important to discretize the pre-cooler to capture the considerable change in fluid properties. Similarly, the recuperator is also discretised. A grid independence study was conducted, and a discretization resolution of six elements was selected for both recuperator and pre-cooler. The overall conductance is divided equally for each discretized element. The recuperator is sized using UA_{RX}, which is selected by ensuring an overall nominal full-load recuperator effectiveness, $\varepsilon_{RX} = 0.9$ in Eq.(3.55) [35]. The pre-cooler is sized using UA_{PC}, which is selected to ensure an outlet temperature of $T_{PC,e} = 33°C = T_{comp,i}$.

$$\varepsilon_{RX} = \frac{h_{c,e} - h_{c,i}}{h_{h,i} - h|_{p_{h,e}, T_{c,i}}} = \frac{h_{h,i} - h_{h,e}}{h_{h,i} - h|_{p_{h,e}, T_{c,i}}} \tag{3.55}$$

A direct attemperator is used for the sCO2 Brayton cycle. The fraction of the flow stream for attemperation is fixed at 10% for both heater configurations, enforced through a secondary loss factor specified at the attemperator control valve, K_{ATT}. The attemperation stream and primary flow stream are then mixed at the attemperation node, modelled as a simple mixing chamber. The sCO2 heaters are modelled as radiative-convective heat exchangers, discussed in section 3.2.1.3. A 1% pressure drop is assumed for both sCO2 heat exchangers, implemented using specified secondary loss factors.

3.2.4 Boundary conditions, control philosophy, and solution procedure

Geometric parameters and properties described in the previous section are used to compute component characteristics for various components in the integrated cycle model. Figure 3.14 shows various inputs and boundary conditions for the integrated cycle. Variables shown in black text are boundary conditions. Component characteristic inputs are as described in the previous section, such as turbomachinery performance curves, and not explicitly shown in the figure. As the model solves the momentum balance equations, set points for various cycle parameters are not enforced directly, and are instead converged to via steady state control. For example, to enforce a mass flow rate for a valve, the secondary loss factor for that valve, K_{loss}, is specified and iteratively tuned via steady state control to obtain the required mass flow rate. While this approach requires steady state control of the integrated cycle, it also captures component characteristics which in turn determine cycle operating points. This is beneficial when simulating various operating conditions and is required for part load analysis. In contrast, without solving the momentum balance equations, operating points such as mass flow rates, must be directly imposed, meaning component characteristics may not necessarily be captured, which is the approach typically used for similar studies observed in literature. For pipes, the loss factor remains fixed, while the loss factor for valves is tuned via steady state control.

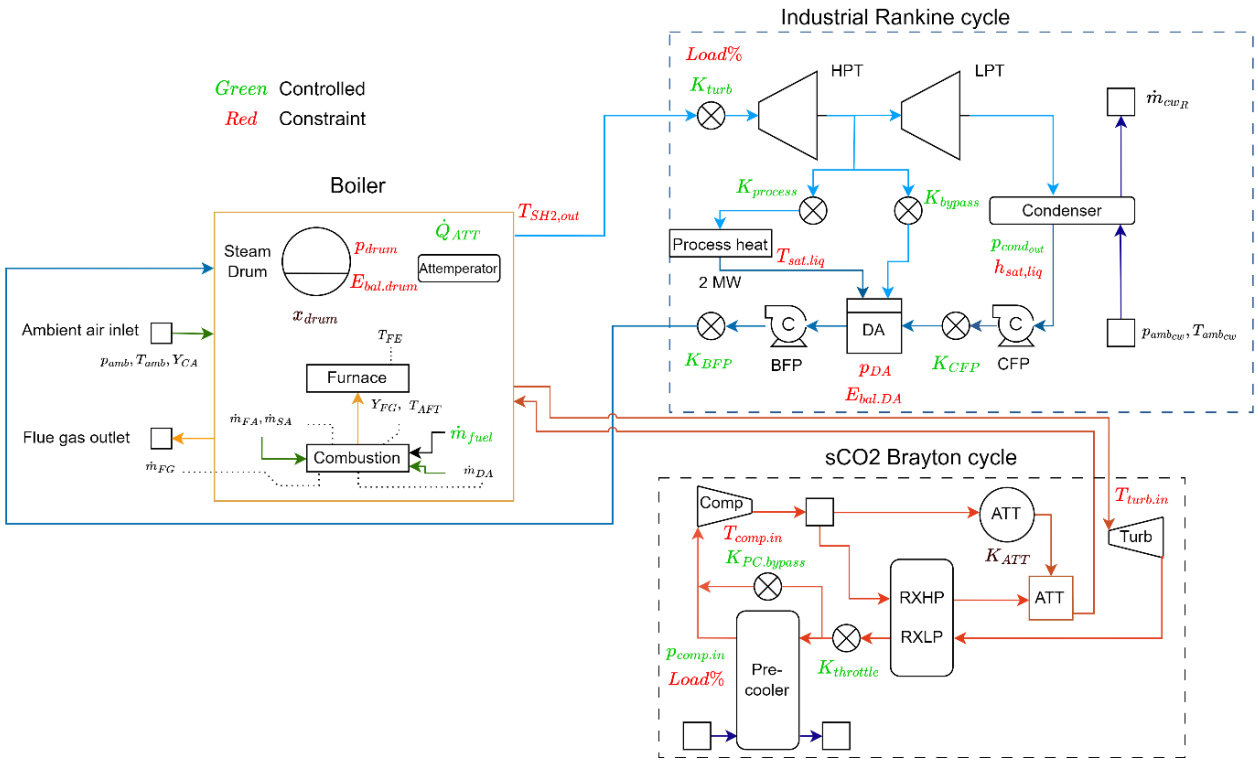


Figure 3.14: Inputs and boundary conditions for integrated cycle

For the boiler, inputs for the combustion reaction are provided, including HHV, excess air ratio, fuel constituents, and more as described in section 3.2.1.1. Initially, for a given $\dot{m}_{fuel}$, the combustion model provides the required mass flow rate for air supply and flue gas streams as well as flue gas composition, which is used as a boundary condition. Iteratively, T_{AFT} and T_{FE} are calculated at the combustion and furnace control volumes and applied each iteration as temperature boundary conditions.

For the Rankine cycle, $p_{cond,out}$, the K_{loss} for various valves, and $\dot{Q}_{ATT}$ for attemperation are specified. The condenser ambient cooling water inlet pressure, temperature, and mass flow rate are also specified. K_{turb} is used to vary the load of the Rankine cycle. At nominal load, the turbine valve is fully open. For partial load operation, the turbine valve is adjusted to meet the required load percentage, defined as the percentage of feedwater flow rate at partial load compared to full load. For industrial boilers of this size and specific application (sugarcane mill), at lower loads, $\approx 40\%$, the boiler becomes highly inefficient and unstable. Therefore, the plant typically operates between 60-100% of the maximum continuous rating (MCR). Thus, in this study Rankine cycle load is varied from 100% (nominal load) to 60% load.

For both Brayton cycle configurations, the pre-cooler outlet pressure/compressor inlet pressure is specified, as well as the ambient cooling water inlet pressure and temperature, and mass flow rate. The pre-cooler bypass valve shown in Figure 3.14 is used to maintain the compressor inlet temperature/pre-cooler outlet temperature above 33°C to avoid liquid entering the compressor.

The sCO2 Brayton cycle is proposed as a supplementary cycle, consequently, its performance also varies based on the Rankine cycle load, using a load-following strategy. Throughout this work, the specified load percentage refers to the Rankine cycle load. Initially, to investigate various control methodologies for the sCO2 Brayton cycle, the single heater configuration is used. A control methodology is then selected and used to simulate both single and dual configurations. Two control methodologies are explored, namely inventory control only, as well as inventory control combined with a throttling valve. Additionally, the sCO2 Brayton cycle is also simulated without any control methodology for comparison purposes. Inventory control requires a pressure differential between the fluid in the cycle and the fluid in the storage tanks to allow for fluid in the cycle to be charged or discharged. Reducing the overall fluid pressure in the cycle maintains the same operating points on the T-s diagram (given other operating conditions are unchanged). This allows for the cycle efficiency to be maintained at lower loads while reducing the mass flow rate and consequently net power output. For ideal gases, turbomachinery operating points also remain unchanged at lower loads. For this work, the minimum sCO2 pressure is constrained by operation above the sCO2 critical pressure (7.4 MPa). For the maximum pressure, moving too far away from the critical sCO2 pressure

reduces the cycle efficiency, as well as increases the maximum cycle operating pressure which can impact tube material operating lifetime. As such, for inventory control, $p_{comp.in}$ is varied linearly from 10 MPa at nominal load to 7.5 MPa at low load (60%), while there is no control on the maximum Brayton cycle temperature. For inventory control combined with a throttling valve, $p_{comp.in}$ is again varied linearly from 10 MPa at nominal load to 7.5 MPa at low load, however, the throttling valve in the Brayton cycle, seen in Figure 3.14, is adjusted to maintain the turbine inlet temperature, $T_{turb.in}$, at 550°C. Lastly, for the sCO2 Brayton cycle without any control methodology, the throttling valve remains fully open and $p_{comp.in}$ remains fixed at 10 MPa for all loads.

Table 3.5: External control loop parameters, target functions, and constraints

Controlled parameter	$p_{cond,out}$	K_{CFP}	$K_{process}$	K_{bypass}	K_{BFP}
Target	Saturated liquid at condenser outlet	Mass balance over de-aerator	Energy balance for 2 MW process heat extraction	Energy balance at de-aerator	Fixed pressure at steam drum
Constraint	$h_{cond} = h_{sat,liq}@p_{cond}$	$p_{DA} =$ 835 kPa	$T_{process,e} = T_{sat,liq}@p_{process}$	$\dot{m}_{DA}h_{DA} = \dot{m}_{CFP}h_{CFP} + \dot{m}_{HPT.bypass}h_{HPT.bypass} + \dot{m}_{process.out}h_{process.out}$	$p_{SD} =$ 6.914 MPa
Controlled parameter	$\dot{m}_{fuel}$	$\dot{Q}_{ATT}$	$K_{PC.bypass}$	$K_{throttle}$	
Target	Energy balance at steam drum	Fixed temperature at SH2 outlet	sCO2 compressor inlet temperature	sCO2 turbine inlet temperature	
Constraint	$\dot{m}_{SD}h_{SD} = \dot{m}_{EC3.out}h_{EC3.out} + \dot{m}_{EV.out}h_{EV.out}$	$T_{SH2,e} =$ 485 °C	$T_{comp.in} \geq 33°C$	$T_{turb.in} = 550°C$	

Several constraints are applied to the model, shown in red text in Figure 3.14. The various controlled parameters are shown in green text in Figure 3.14. Table 3.5 shows the control parameters, target functions, and constraints satisfied by the external loop. For each iteration of the external search loop, the balance equations are solved to convergence. K_{turb} is a primary input to the Rankine cycle model and specified for the required load percentage. Similarly for the Brayton cycle, $p_{comp.in}$ is also a primary input, specified for the required load percentage for inventory control. The controlled parameters in Table 3.5 are adjusted for full convergence with the provided component characteristics, sCO2 heater configurations, K_{turb} , and $p_{comp.in}$. Effectively, this means that Rankine

cycle performance for a given load percentage is the same for various sCO2 heater configurations, however, boiler operation and sCO2 Brayton cycle performance varies.

To arrive at a fully converged solution where the specified constraints are satisfied, the iterative solution process is wrapped within an external search loop, as shown in Figure 3.15. With all inputs, boundary conditions and component characteristics specified, the model is solved till full convergence of balance equations. The target functions for the respective controlled parameters are then setup and provided to the external loop. For each iteration of the external loop, the balance equations are solved to convergence. The target functions are then re-calculated, and the controlled parameters updated. Due to the large non-linearities associated with the different control variables, a multi-variate search algorithm resulted in solution instability. As such, a single-variate secant algorithm from the SciPy library [77] is used to iterate the various control parameters to satisfy the respective target functions and provided constraints. The secant method is derived from Newton's method [78], shown in Eq. (3.56) where x_n is the updated value of the control parameter, and x_{n-1} the value at the previous iteration. As there is no closed-form analytical expression for the target function derivative, a finite-difference method is used to approximate the derivative shown in Eq. (3.57).

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \tag{3.56}$$

$$f'(x_{n-1}) \approx \frac{f(x_{n-1}) - f(x_{n-2})}{x_{n-1} - x_{n-2}} \tag{3.57}$$

The controlled parameters are updated every iteration of the external loop. Once convergence of the controlled parameters is reached, the calculation concludes. A relative convergence of 1% is assumed for all controlled parameters to meet the respective constraints.

The complete integrated model consists of 82 elements and solves within 1-15 minutes on a 12-core AMD Ryzen 5900X CPU depending on the initial guesses. Although the thermofluid network modelling approach applied here has been validated with site data for industrial-scale biomass-fired and coal-fired boilers [10], [54], it has not been validated via comparison with plant measurements for this specific layout since the plant has not yet been constructed. A detailed CFD analysis is conducted in Chapters 5, 6, and 7 to verify the heat uptakes of boiler heat exchangers and to further calibrate the 1D integrated model.

Appendix A contains a brief description and code listing for the integrated 1D process model. The structure of the model is explored, as well as a summary of the codebase. Code for a driver program for the single heater configuration is presented as an example, as well as several boiler classes and their implementation.

This concludes the chapter on the integrated cycle layout and 1D thermofluid process modelling development.

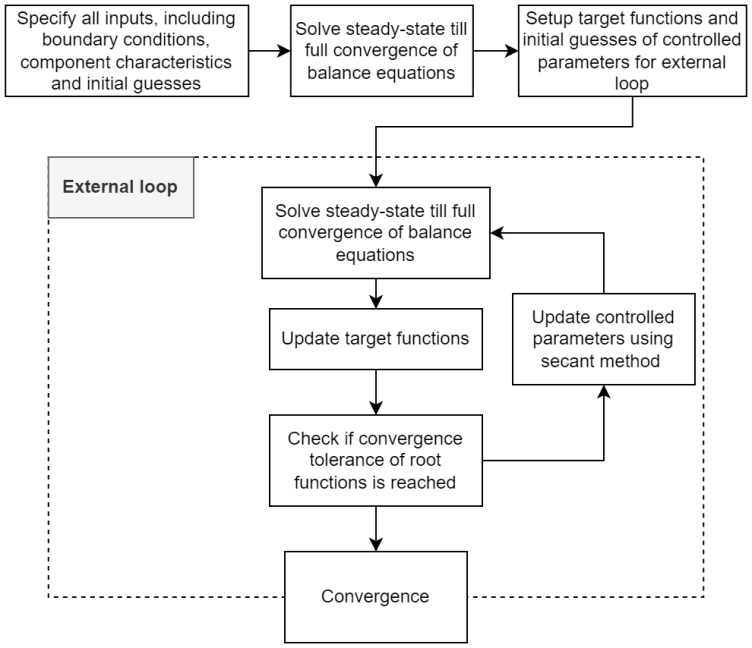


Figure 3.15: External solver loop process flow diagram

4. Integrated 1D process model: Results and discussion

Chapter 4 discusses the results from the integrated 1D process model. The chapter first details the integrated cycle performance at nominal load, including overall cycle performance, as well as a detailed analysis of boiler heat exchangers. Initially, a preliminary study to select the sCO2 Brayton cycle maximum temperature, or TIT, is discussed. The selected TIT is thereafter used for further investigation. The 1D model is used to simulate three configurations, namely, the existing Rankine cycle without the sCO2 Brayton cycle integration, the integrated cycle with the single sCO2 heater configuration, and the integrated cycle with the dual sCO2 heater configuration. For reference, these are referred to as existing cycle, single configuration, and dual configuration respectively.

The second part of this chapter discusses the part-load analysis of the integrated cycle. Various sCO2 Brayton cycle control methodologies are explored across the operating load range. The selected control strategy is then used to investigate the integrated cycle for both single and dual sCO2 heater configurations.

4.1 Nominal load case study

A preliminary parametric study was conducted to select an appropriate TIT from three options namely 550°C, 650°C, and 720°C. The integrated model was simulated for these TITs at nominal load. Detailed results are presented thereafter for only one selected TIT.

Table 4.1 shows high level cycle results for the existing cycle without the sCO2 Brayton cycle integration, single sCO2 heater configuration, and dual sCO2 heater configuration. Since all cases are simulated at nominal load for the Rankine cycle, the feedwater flow rate, condenser pressure, and net electrical power output for the Rankine cycle are the same for all cycle configurations, shown in Table 4.2.

The Rankine cycle has a net electrical power output of 8.8 MW (excluding the 2 MW of process heat), and thermal efficiency of 25.30%. Comparing the fuel flow rates for the existing cycle and for the sCO2-integrated cycles, an increase of approximately 15% is observed for the 550°C case, 9% for the 650°C, and 5% for the 720°C. This difference represents the overfiring required to maintain the Rankine cycle at full load while providing sufficient additional heat to the sCO2 Brayton cycle. The increase in fuel flow rate results in an increase in the flue gas mass flow rate due to the increase in the mass of the reactants in the combustion process. The fuel flow rate decreases for higher TITs because the required sCO2 mass flow rate decreases with higher TIT.

Table 4.1: High level simulation results for the existing cycle, dual sCO2 heaters, and single sCO2 heater for TITs of 550°C, 650°C and 720°C

		Fuel flow rate [kg/s]	**Flue gas flow rate [kg/s]**	$\dot{Q}_{ATT_{Rankine}}$ **[kW]**	**sCO2 flow rate [kg/s]**
	Existing cycle (no sCO2)	5.647	23.099	1580	-
550°C	Single sCO2 heater	6.512	26.636	1271	19.358
	Dual sCO2 heaters	6.501	26.591	922	19.076
650°C	Single sCO2 heater	6.151	25.158	1331	10.577
	Dual sCO2 heaters	6.149	25.153	1033	10.350
720°C	Single sCO2 heater	5.888	24.083	1373	4.970
	Dual sCO2 heaters	5.979	24.454	1045	
		sCO2 Brayton cycle net electrical power output [kW]	**Total net electrical power output [kW]**	$\eta_{th.Brayton}$ **[%]**	$\eta_{th.Overall}$ **[%]**
	Existing cycle (no sCO2)	-	8760	-	25.32
550°C	Single sCO2 heater	1861	10621	35.68	26.66
	Dual sCO2 heaters	1811	10621	35.41	26.60
650°C	Single sCO2 heater	1221	9981	40.37	26.50
	Dual sCO2 heaters	1197	9957	39.80	26.62
720°C	Single sCO2 heater	627.9	9388	42.34	26.00
	Dual sCO2 heaters	833.2	9593	41.66	26.19

Table 4.2: Rankine cycle results for all configurations

	Feedwater flow rate [kg/s]	**Condenser pressure [kPa]**	**Rankine cycle net electrical power output [kW]**	$\eta_{th.Rankine}$ **[%]**
All cycle configurations	13.127	18.3	8760	25.30%

Comparing the three TIT cases, the 650°C and 720°C cases have lower sCO2 mass flow rates (-45% and -73% compared to 550°C) to obtain the higher TITs. Consequently, the sCO2 heaters for the 650°C and 720°C cases have lower heat uptakes in the boiler, resulting in lower fuel flow rates and flue gas flow rates. Additionally, the sCO2 Brayton cycle net electrical power output for the 650°C and 720°C cases are 1.2 MW and 0.7 MW, compared to 1.8 MW for the 550°C cases (-34% and -61%). However, as expected, with the higher TIT at 650°C and 720°C, the sCO2 Brayton cycle thermal efficiency is 4.4% and 6.4% higher compared to the 550°C cases.

Comparing the overall integrated cycle thermal efficiency with that of the existing Rankine cycle alone, the 550°C cases have the largest increase (+1.31%), followed by the 650°C cases (+1.24%), and lastly the 720°C cases (+0.78%). While the Brayton cycle thermal efficiency does increase slightly when increasing the TIT, the reduction in net sCO2 Brayton cycle generation results in a slight decrease in the overall integrated cycle efficiency for the 650°C and 720°C cases compared to the 550°C cases. As the sCO2 Brayton cycle efficiency at 550°C is already 10% higher than the Rankine cycle efficiency, the further thermal efficiency gain at higher TITs is negated by the drop in net electrical power output for the sCO2 Brayton cycle. Therefore, the 550°C TIT is selected for further investigation due to it having the highest efficiency and net electrical power output. The 550°C cases require a 15.3% increase in fuel flow rate for a 21.2% increase in net electrical power output.

The Rankine cycle attemperation rate, $\dot{Q}_{ATT_{Rankine}}$, allows for control of the SH2 outlet steam temperatures, with a set point of $T_{SH2,out} = 485\,°C$. The existing cycle requires 1580 kW of attemperation, while for the selected 550 °C sCO2 cases, the single configuration requires 1271 kW (-19.6%) and the dual configuration 922 kW (-41.6%). The reduction in required attemperation is due to the heat uptake in the radiative sCO2 heater, which results in less heat uptake to the steam in SH2. Notably, the single configuration requires more attemperation compared to the dual configuration (+46%), even though both configurations have nearly the same sCO2 flow rates. The significant difference in attemperation is attributed to the different heater configurations and its positioning in the boiler. The sCO2 heater positioning influences downstream flue gas temperatures as well as heat uptakes in downstream heat exchangers, which is discussed in detail in Section 4.1.3.

4.1.1 Rankine cycle

Figure 4.1 shows the Rankine cycle T-s diagram for the existing cycle. Note that this diagram would be relatively the same for the configurations with the sCO2 Brayton cycle, with only a difference in attemperation capacity and boiler heat exchanger performance. The process heat line is also shown as a dashed line, with 2 MW of heat extraction. Various valve pressure drops, necessary for Rankine

cycle control, are accounted for, such as CFP to DA (CFP valve), and HPT to DA (process heat valve). The HPT and LPT generate approximately 3.9 MW and 5.0 MW respectively.

The total feedwater flow rate at nominal load is 13.13 kg/s. Of this, 78% flows through the LPT and 22% towards the process heat line. The process heat line then splits again with 29% providing the required process heat and 71% bypassing the process heat extraction and flowing to the DA. For the existing cycle, the single sCO2 heater configuration, and the dual sCO2 heater configuration respectively, temperature drops of 41°C, 23°C, and 31°C are calculated due to attemperation between SH1 and SH2.

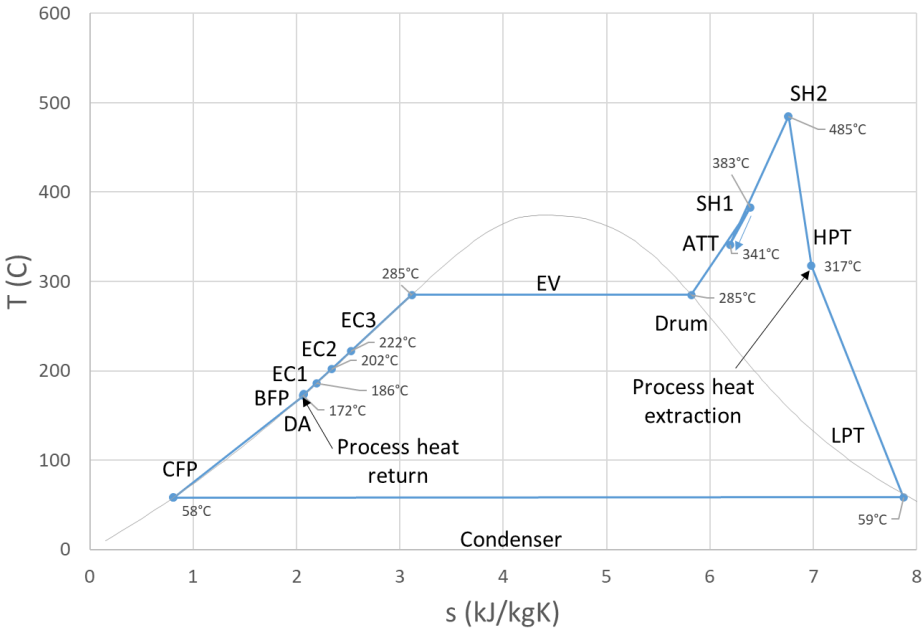


Figure 4.1: Rankine cycle T-s diagram

4.1.2 sCO2 Brayton cycle

Figure 4.2 shows the T-s diagrams for both configurations of the sCO2 Brayton cycle with a TIT of 550°C. The recuperator, which is split into RXHP and RXLP, and pre-cooler are each discretised into 6 increments. The lowest temperature in the cycle of 32°C is just above the critical point, as shown by the saturation curve in grey. At the lowest pressure in the cycle of 10 MPa, the corresponding density is 741 kg/m^3, leveraging the high density of CO2 near the critical point. This pressure is selected above the critical pressure of 7.377 MPa to allow for inventory control at lower loads by lowering the overall pressure level in the cycle.

The two sCO2 Brayton cycle configurations have similar performance. For the turbomachinery, both configurations have a TIT of ~555°C, as well as a PR_C of 2.5, and PR_T of 2.4, as specified through the performance curves. As the sCO2 mass flow rates are approximately the same between the two configurations, net electrical power output and thermal efficiency are also similar. The Brayton cycles have a thermal efficiency of $\eta_{th_{Brayton}} = 35.5\%$. This is lower than many of the simple recuperated sCO2 Brayton cycles reported in literature of up to 40% [35]. This is due to the selected higher minimum cycle pressure of 10 MPa, which is essential for inventory control purposes, demonstrated in the part-load case study. The pre-cooler and recuperator sizing is the same for both configurations, with $UA_{PC} = 170$ kW/K and $UA_{RX} = 210$ kW/K.

The main differences between the two Brayton cycle configurations are the heater arrangement and attemperation point. For the dual configuration, attemperation occurs between sCO2H1 and sCO2H2. For the single configuration, attemperation occurs before the flow stream enters sCO2HX. The attemperation extraction point is the same for both configurations, with a flow stream split before the recuperator, as shown in Figure 4.2.

Table 4.3: Outlet temperatures at RXHP, sCO2H1, and ATT for dual and single heater configuration

	$T_{RXHP_{out}}$ [°C]	$T_{sCO2H1_{out}}$ [°C]	$T_{ATT_{out}}$ [°C]
Single config	381	-	340
Dual config	384	447	397

Table 4.3 shows the outlet temperatures at RXHP, sCO2H1 (dual configuration) and ATT. For the dual configuration, where attemperation occurs after the flow stream is heated at sCO2H1, the inlet temperature to the convective sCO2 heater, sCO2H2, is 397°C, compared to 340°C for the single heater (sCO2HX) case.

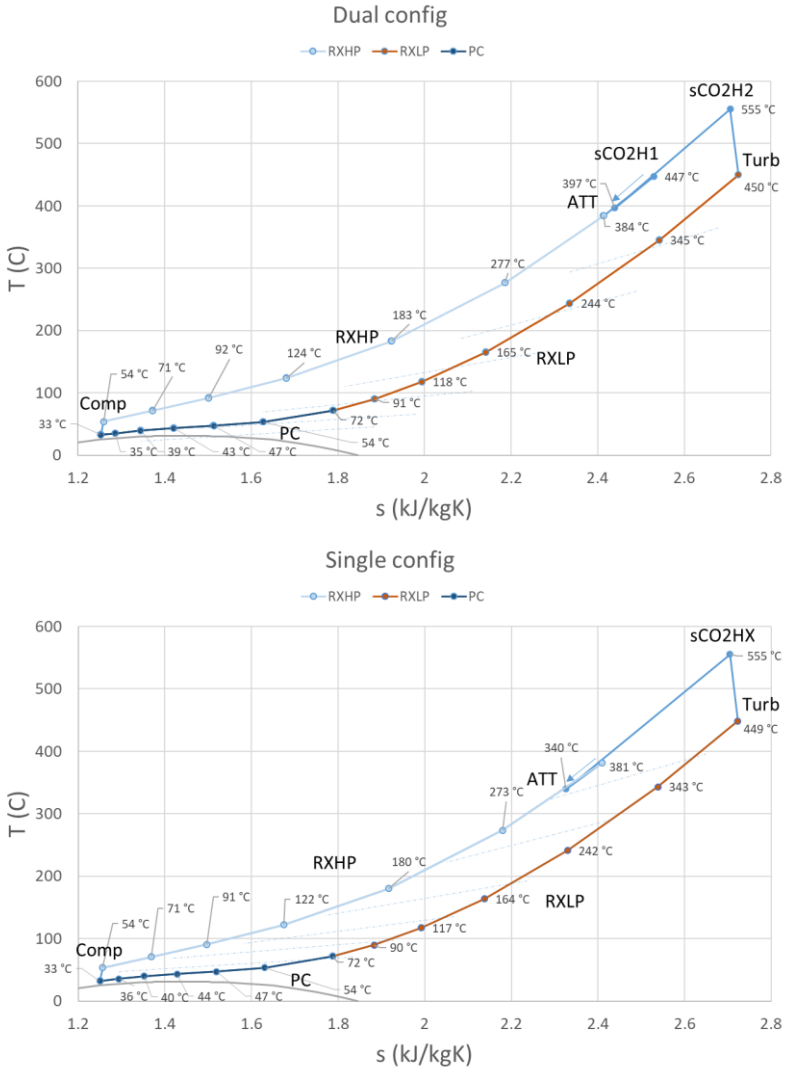


Figure 4.2: T-s diagram for $TIT_{Brayton}$= 550°C; Dual sCO2 heater configuration (top) and single sCO2 heater configuration (bottom)

4.1.3 Boiler heat exchangers

4.1.3.1 *Flue gas heat transfer rates and temperatures*

Figure 4.3 shows the flue gas heat transfer rates at the different boiler heat exchangers for the existing, single, and dual configurations. $\dot{Q}_{fg}$ accounts for all heat loss from a flue gas volume, excluding $\dot{Q}_{abs}$ from upstream incident radiation. For the existing cycle, the largest flue gas heat transfer rates are at the furnace, SH1, and SH2. The higher flue gas heat transfer rate at SH1 over SH2 is due to the larger heat exchanger surface area at SH1 (+109% of SH2), which counteracts the

lower inlet flue gas temperature (-33% of SH2). Note that this does not account for $\dot{Q}_{abs}$ which contributes significantly more to overall heat transfer at SH2 compared to SH1.

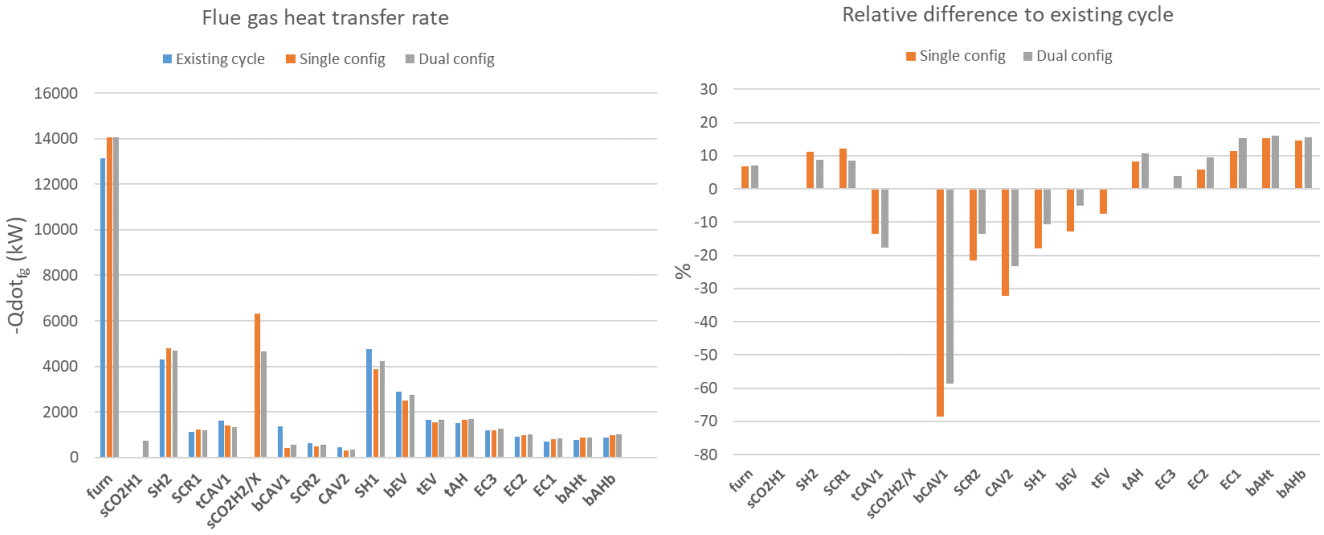


Figure 4.3: Flue gas heat transfer rate (loss) for existing, single, and dual configs (left). Relative difference to existing cycle (right).

At the combustion process outlet, T_{AFT} is the same for both configurations at 1342°C. The furnace exit temperature, T_{FE}, is higher for configurations with the sCO2 heater integration compared to the existing cycle (991°C vs 965°C). The higher T_{FE} for sCO2 heater configurations is due to the increased fuel firing rate, consequently increasing Bo in Eq. (3.18) and T_{FE}. This also results in increased furnace radiant heat at the furnace exit plane, $\dot{Q}_{FE}$, and to furnace waterwalls and refractory walls, shown in Table 4.4. The majority of the heat uptake in the furnace is via the waterwalls (82.5%) followed by furnace radiation through the exit plane (14.8%), which is absorbed by downstream heat exchangers.

Table 4.4: Furnace heat uptake breakdown

	$\dot{Q}_{FE}$ [kW]	$\dot{Q}_{WW}$ [kW]	$\dot{Q}_{RF}$ [kW]	$\dot{Q}_{loss.fur}$ [kW]	$\dot{Q}_{fg.fur}$ (Tot) [kW]
Existing	1940	10865	210	133	-13148
Single	2091	11588	226	142	-14047
Dual	2094	11603	226	142	-14065

For the dual configuration, $\dot{Q}_{fg,sCO2H1}$ is relatively small, due to the small physical size of the heat exchanger to minimise adversely impacting heat uptake at SH2. Consequently, $\Delta T_{fg,sCO2H1} = -19°C$, which counteracts the increased T_{FE} from the increase in fuel firing rate. This brings the flue gas inlet temperature at SH2 for the dual configuration closer to the existing cycle, (972°C vs 965°C respectively), while the inlet temperature to SH2 for the single configuration is $T_{FE} = 991°C$. The increased flue gas inlet temperature, as well as increased flue gas mass flow rate from overfiring, results in increased flue gas heat transfer rates at SH2 for the sCO2-integrated configurations compared to the existing cycle. Additionally, the increased inlet flue gas temperature for the single configuration over the dual configuration results in slightly higher $\dot{Q}_{fg,SH2}$ (+2.2%).

For the single and dual configurations, there is a significant flue gas temperature decrease due to heat transfer to the sCO2 heater (sCO2H2/sCO2HX) slotted in CAV1. For the single configuration, $\dot{Q}_{fg,sCO2HX}$ is the largest flue gas heat transfer rate at -6.3 MW, excluding the furnace. Comparing the single and dual configurations, $\dot{Q}_{fg,sCO2HX}$ is higher than $\dot{Q}_{fg,sCO2H2}$ primarily due to the larger surface area of sCO2HX, as well as a lower inlet sCO2 temperature as shown in Table 4.3.

Because of the position of sCO2H2/sCO2HX in the flue gas flow path, flue gas temperatures and heat transfer rates downstream are reduced. The effect of reduced flue gas temperatures can be seen in Figure 4.3 with reduced heat uptakes at bCAV1, SCR2, CAV2, SH1, bEV, and tEV, which then approaches and exceeds the existing cycle heat uptakes thereafter. sCO2HX has a larger impact on the downstream heat exchangers compared to sCO2H2 due to the larger gas heat loss. Notably, at CAV2 where radiative heat transfer dominates, the reduced flue gas temperatures have a larger impact on radiative heat flux. Consequently, there is a 32% and 23% reduction in $\dot{Q}_{fg.CAV2}$ for the single and dual configuration respectively. Additionally, at SH1, there is a 18% and 10.8% reduction in flue gas heat transfer rates for the single and dual configuration respectively.

4.1.3.2 *Water/sCO2/Air temperatures and heat transfer*

Figure 4.4 shows the heat transfer rates to various flow streams, $\dot{Q}_{sec}$, for the steam Rankine cycle, sCO2 Brayton cycle, and combustion air streams for the three cycle configurations. $\dot{Q}_{sec}$ is broken down into $\dot{Q}_{abs}$, absorbed radiation from upstream gas volumes, and $\dot{Q}_{ex}$, external convective and radiative heat transfer from the coupled flue gas control volume.

For the water stream, the main areas of interest are the outlet of SH1/inlet of ATT, and outlet of ATT/inlet of SH2. At the inlet to SH1, all configurations have the same inlet temperature from the steam drum. For sCO2-integrated configurations, the placement of sCO2H2/X reduces flue gas

temperatures at SH1 and consequently $\dot{Q}_{fg,SH1}$, as discussed previously. This also results in a decrease in $\dot{Q}_{\text{sec},SH1}$ compared to the existing cycle (-18% single config, -11% dual config). The single configuration is impacted more in this regard due to the larger area of sCO2HX compared to sCO2H2 for the dual config.

Figure 4.4: Heat transfer rates to water, sCO2, and air streams (left). Relative difference of Qdot$_{sec}$ to existing cycle (right).

As a general trend, $\dot{Q}_{abs}$ decreases significantly along the flue gas flow path, with the majority of furnace radiant heat absorbed by SH2. Additionally, heat radiated by the flue gas at the boiler heat exchangers, $\dot{Q}_{back}$, also reduces due to lower flue gas temperatures, with 180 kW at SH2 compared to 30 kW at SH1. At SH2, all configurations have the same outlet temperature as this is controlled via attemperation, $T_{w,SH2,out} = 485°C$. The single configuration has the largest $\dot{Q}_{sec,SH2}$ (+10% vs existing), followed by the existing cycle, and lastly the dual configuration (-3% vs existing).

For the single config, the increase in $\dot{Q}_{sec,SH2}$ is attributed to the increased $\dot{m}_{fg}$ from overfiring, as well as increased furnace exit temperature. The increase in $\dot{Q}_{sec,SH2}$ for the single configuration balances out with a decrease in $\dot{Q}_{sec,SH1}$ and slight reduction in $\dot{Q}_{ATT,Rankine}$.

For the dual configuration, the radiative sCO2H1 has direct impact on the heat uptake at SH2. From Figure 4.4, $\dot{Q}_{abs,sCO2H1}$ absorbed from furnace incident radiation forms majority of $\dot{Q}_{sec,sCO2H1}$ compared to $\dot{Q}_{ex,sCO2H1}$ (63% vs 37%). The absorption of furnace radiant heat at sCO2H1 results in a decrease in $\dot{Q}_{abs,SH2}$, 1301 kW for the dual configuration, compared to 1788 kW for the existing cycle,

and 1927 kW for the single configuration (-27% and -32% respectively). On the other hand, $\dot{Q}_{ex,SH2}$ for the dual configuration is higher than the existing cycle, due to the higher flue gas mass flow rate from overfiring. The overall result is that $\dot{Q}_{sec,SH2}$ for the dual configuration is only 3% lower than the existing cycle.

4.1.3.3 *Tube metal temperatures and degradation potential*

A detailed thermal resistance network is captured by the model, allowing for the calculation of tube metal temperatures, as shown in Figure 3.11, including the outer fouling layer and outer and inner tube metal temperatures. This highlights an additional feature of the developed 1D model, which would not be captured by a lumped parameter model. As overfiring is required with sCO2-integrated configurations, it is important to check tube metal temperatures are within material specifications. Figure 4.5 shows the maximum tube metal temperatures for the radiative-convective boiler heat exchangers.

Due to the outer insulating fouling layer between the flue gas and the tubes, internal fluid temperatures have a larger influence on tube metal temperatures due to the lower internal thermal resistances. The highest tube metal temperature is at sCO2H2/X due to the higher sCO2 operating temperature, at 497 °C and 506 °C for the single and dual configuration respectively.

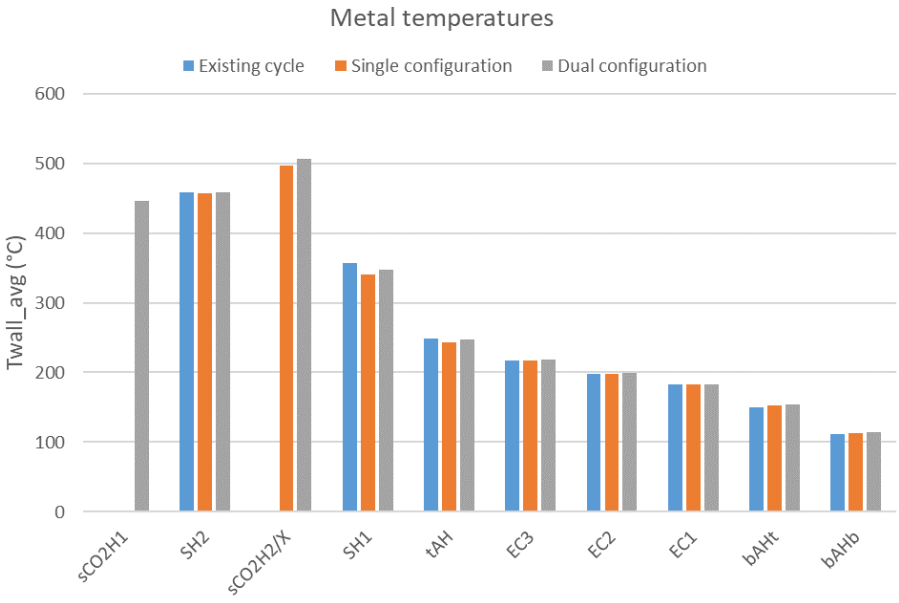


Figure 4.5: Maximum tube metal temperatures

At SH2, tube metal temperatures are similar for the existing, single, and dual configurations, at 458°C, 457°C and 459°C respectively. An increase in tube metal temperatures at SH2 would be expected for the sCO2-integrated configurations, due to the increased flue gas temperatures resulting from overfiring. However, the placement of the sCO2H2/X and its impact on SH1 also results in a change in steam temperatures at SH2. The average steam temperature at SH2 is 6°C lower for the single configuration, and 2°C lower for the dual configuration compared to the existing cycle. In comparison, average flue gas temperatures are 27°C higher for the single configuration, and 11°C higher for the dual configuration. The reduced steam temperatures at SH2 counteract the higher flue gas temperatures for sCO2-integrated cycles, resulting in similar tube metal temperatures between all cycle configurations. This effect is less pronounced for the dual configuration due to the flue gas temperature drop across sCO2H1.

At SH1, sCO2-integrated cycles have lower average flue gas temperatures, as well as lower average steam temperatures due to the presence of sCO2H2/X upstream of SH1. This results in decreased tube metal temperatures compared to the existing cycle (357°C) for the single (341°C), and dual (347°C) configurations.

Cold end corrosion can occur in boilers when the exhausting flue gas temperature approaches dew point temperature. For coal applications, typically the sulfuric acid dew point temperature (between 100 °C – 150°C) is used to check for cold end corrosion. However, for biomass fuels, sulfur content is significantly lower, and thus presents low risk of corrosion from sulfuric acid. Instead, for biomass boilers, cold end corrosion can be caused by alkaline hygroscopic salts which absorb moisture from the flue gas. Deliquescence, which is when the hygroscopic salts fully dissolve in the moisture and form a solution, occurs at temperatures below 120 °C [79]. For all configurations, the lowest observed flue gas temperature was 165 °C, indicating low risk of cold end corrosion. This flue gas temperature also lies above the typical sulfuric acid dew point temperature.

For the superheaters and radiative sCO2H1, the predicted metal temperatures and operating gas temperatures lie within the stable zone to reduce the risk of fuel ash corrosion on the heat exchanger tubes as described by [62]. The maximum gas velocity through each tube bank is limited to the erosion potential of fly ash and the amount of fly ash in the fuel [62]. The maximum allowable gas velocity is 19.8 m/s based on a ash to fluegas ratio of 0.52% observed in this study [62]. The highest gas velocities were at sCO2H2/X at 14.3 m/s, followed by EC3 at 12.3 m/s. The highest observed gas velocity is therefore below the maximum allowable gas velocity to prevent high levels of erosion.

4.1.3.4 *Boiler heat load breakdown*

The majority of the boiler heat load contributes towards heating the evaporating circuit, through the water walls around the furnace and boiler heat exchangers, evaporators, and screens. Table 4.5 shows a summary of heat loads for all the Rankine cycle heat exchangers as well as the sCO2 heaters. Importantly, radiant heat absorbed by waterwalls surrounding the furnace makes up a larger percentage of the EV heat uptake, at 3.4% for the single configuration and 3.7% for the dual configuration. There is also an increase in heat uptakes at the EC heat exchangers (+4.8% and 8.6% respectively. The increase in these heat uptakes counteracts the reduced heat uptake at other evaporator heat exchangers due to the reduction in flue gas temperatures caused by the sCO2 heater(s).

As previously discussed, for the SHs, there is a shift in heat uptake between SH1 and SH2. For the single configuration, SH1 has reduced heat uptake compared to SH2 due to sCO2HX positioned upstream of SH1. Consequently, SH2 makes up a higher percentage of the required heat load, also due to the increased flue gas temperatures and mass flow rate from overfiring. For the dual configuration where SH2 is also adversely impacted by sCO2H1, attemperation is reduced to ensure steam is heated to the required exit temperature.

Table 4.5: Boiler heat load breakdown for existing cycle, single configuration, dual configuration

		Existing (kW)	Single (kW)	Dual (kW)	Existing %	Single %	Dual %
EC	EC1	715	797	824	25.2	26.8	26.8
	EC2	922	975	1010	32.5	32.8	32.8
	EC3	1199	1201	1246	42.3	40.4	40.5
		2836	2973	3080		(+4.8%)	(+8.6%)
EV	furn	11075	11814	11830	46.4	49.8	50.1
	bEV	2897	2522	2746	12.1	10.6	11.6
	tEV	1665	1540	1667	6.97	6.49	7.06
	Other	8251	7853	7382	34.5	33.1	31.2
		23889	23729	23626			
SH	SH1	4462	3634	3950	56.4	46.0	50.0
	ATT	-1580	-1271	-922	-20.0	-16.1	-11.7
	SH2	5030	5543	4879	63.6	70.1	61.7

		7912	7906	7907			
Total		34637	34608	34612			
sCO2	sCO2H1	-	-	1336	-	-	26.1
	sCO2H2/X	-	5215	3778	-	100	73.9
			5215	5114			
$\dot{Q}_{Tot}$		34637	39823	39726			
η_{boiler} (%)		64.36	64.32	64.18			

Lastly, overall boiler efficiency, η_{boiler} is calculated using Eq. (4.1), where $\dot{Q}_{water}$ is defined as the total heat transfer rate to the water from the BFP outlet to SH2 outlet.

$$\eta_{boiler} = \frac{\dot{Q}_{water} + \dot{Q}_{sCO2}}{\dot{m}_{fuel}(h_{fuel} + HHV_{fuel}) + \dot{m}_{PA+SA}h_{air,tAH,out} + \dot{m}_{DA}h_{amb,air}} \tag{4.1}$$

Boiler efficiency remains similar for all the configurations, around 64.29% on average, which correlates well with the value of 64.48% reported in Laubscher & De Villiers [10]. However, the sCO2-integrated configurations have slightly higher flue gas exit temperatures (169 °C, 170 °C, and 165 °C for the single, dual, and existing configuration respectively), which indicates increased dry gas losses via the flue gas. The overall integrated cycle thermal efficiency increased for the sCO2-integrated cycles at 26.5% compared to 25.3% for the existing cycle. The incremental efficiency from adding the sCO2 cycle is calculated as the increase in power output from the sCO2-integration ($\dot{W}_{net.sCO2}$) divided by the thermal power associated with the increase in fuel flow rate ($\Delta\dot{m}_{fuel}HHV_{fuel}$). The average incremental efficiency for sCO2-integrated configurations is 24.7%.

4.1.4 Summary of results

For the sCO2-integrated cycle configurations a 15.3% increase in fuel firing rate is required to provide sufficient heat uptake for both the Rankine and Brayton cycles. A 21.2% increase in net electrical power output is observed for sCO2-integrated cycles, with an additional 1.9 MW generated by the sCO2 Brayton cycle over the 10.8 MW of combined heat and power generated by the Rankine cycle. This highlights the potential for increased net electrical power output at higher thermal efficiencies, as well as the feasibility of the proposed integration. Due to overfiring, furnace exit temperatures are elevated for sCO2-integrated configurations. There is also an increase in

radiative heat uptake at water walls surrounding the furnace, offsetting the reduced heat uptake at boiler heat exchangers downstream the sCO2 heaters.

Comparing the sCO2 heater configurations, the single heater configuration appears to perform similar to the dual heater configuration in terms of fuel firing rate and net electrical power output. However, the single heater configuration has less impact on SH2, and more Rankine cycle attemperation capacity which improves controllability. Furthermore, the single heater configuration is inherently less complex than the dual heater configuration. With overfiring required to run both configurations, the practical impacts on boiler longevity, with increased fouling and furnace exit temperatures, must also be considered. A detailed sensitivity analysis exploring the impacts on boiler cleanliness and varying fouling factors may be warranted.

The integrated 1D model is computationally efficient and numerically robust, allowing for the impacts of the sCO2 integration to be quantified through detailed analysis on boiler heat exchanger and overall cycle performance. In the next section, the model is also used to study the integrated cycle at varying loads, which highlights further impacts of the sCO2 Brayton cycle integration on the overall system operation.

4.2 Part-load case study

For the part-load case study, the 1D model is used to simulate the existing Rankine cycle for 100% load down to 60% load in 10% increments. Next, the two sCO2 Brayton cycle control methodologies are explored using the single sCO2 heater cycle configuration. The Brayton cycle control methodologies include inventory control only, and inventory control together with a throttling valve. For comparison purposes no control of the sCO2 cycle is also simulated, i.e. where the Rankine cycle load is reduced while allowing the sCO2 cycle to find its own resultant operating point. Lastly, the selected Brayton cycle control methodology is implemented for both single and dual sCO2 heater configurations and overall integrated cycle performance is explored.

4.2.1 Rankine cycle

Figure 4.6 shows the Rankine cycle T-s and p-h diagrams at 100% down to 60% load for the existing cycle. As the sCO2 Brayton cycle is proposed to load-follow the Rankine cycle, Rankine cycle performance only differs in attemperation and boiler heat exchanger performance for sCO2-integrated configurations. The effect of various control valves on the Rankine cycle is highlighted in the p-h diagram. The turbine valve is closed more for lower loads, while the CFP valve and process valves are opened more.

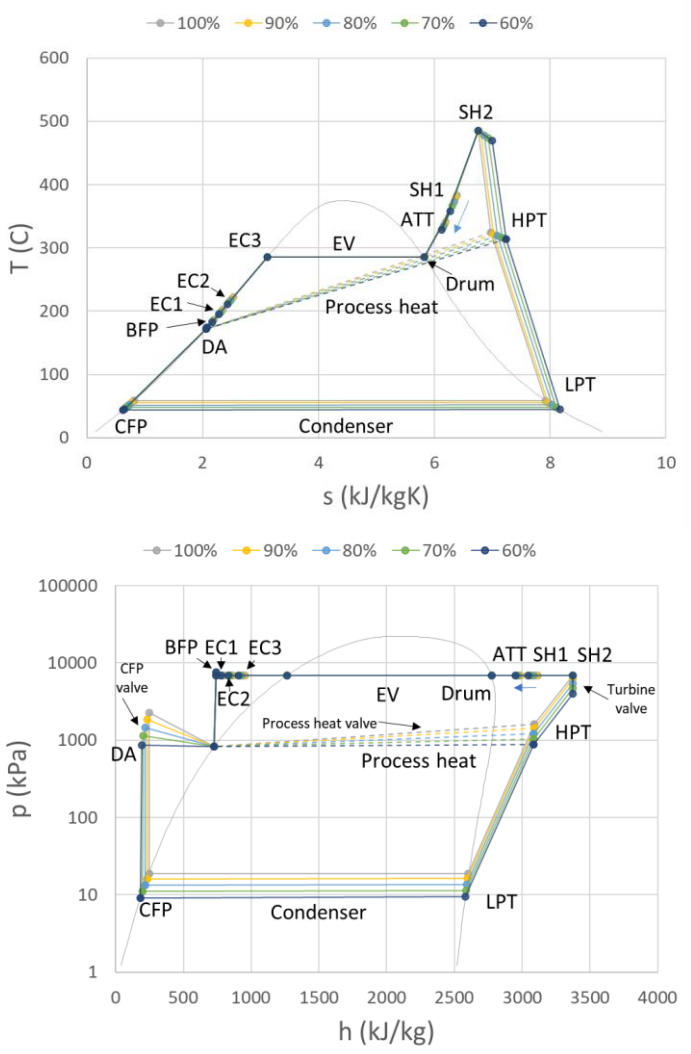


Figure 4.6: Rankine cycle T-s and p-h diagram for 100%-60% load

Table 4.6: Rankine cycle results for 100%-60% loads for all cycle configurations

	$\dot{m}_{fuel}$ [kg/s] (Existing cycle)	Feedwater flow rate [kg/s]	Condenser pressure [kPa]	Rankine cycle net electrical power output [kW]	$\eta_{th.Rankine}$ [%]
100% load	5.664	13.17	18.66	8575	24.69
90% load	5.141	12.00	15.95	7834	24.76
80% load	4.552	10.67	13.31	6956	24.72
70% load	3.959	9.32	11.06	6035	24.57
60% load	3.407	8.05	9.20	5292	24.55

Table 4.6 shows Rankine cycle results for 100%-60% loads. As the load percentage decreases, feedwater flow rate and consequently net Rankine cycle generation decreases. Process heat is fixed at 2 MW for all loads. The required fuel flow rate also decreases with the decrease in Rankine cycle load. Additionally, the condenser backpressure decreases due to the reduced cooling load through the condenser at lower cycle loads.

4.2.2 sCO2 Brayton cycle control methodologies

The two control methodologies for the sCO2 Brayton cycle were explored using the single sCO2 heater configuration. Figure 4.7 and Figure 4.8 shows the T-s and p-h diagrams for the single heater sCO2 Brayton cycle simulated at 100% down to 60% loads with two control methodologies namely, (i) inventory control only, (ii) inventory together with a throttling valve, and additionally with no control. Figure 4.9 shows various results, including sCO2 Brayton cycle efficiency, overall integrated cycle efficiency, sCO2 Brayton cycle net electrical power output, required boiler fuel flow rate, and sCO2 Brayton cycle mass flow rate for varying loads. Initially, 100%-70% load is discussed, with the 60% load cases discussed thereafter.

Starting with inventory control, for 100%-70% load, the compressor inlet pressure is decreased for lower loads, as shown in Figure 4.8. This results in a lower maximum cycle pressure. The use of inventory control results in a decrease in sCO2 mass flow rate, from 19.29 kg/s at 100% load down to 16.21 kg/s at 70% load (-16%), as seen at the bottom of Figure 4.9. A pressure ratio of 2.5 is maintained for the various load percentages due to lower densities at lower loads, negating the decrease in mass flow rate. With the reduction in fuel flow rate for lower Rankine cycle loads, the reduction in sCO2 mass flow rate from inventory control alone is not sufficient to maintain the nominal maximum Brayton cycle temperature of 550°C for lower loads. Consequently, the

maximum Brayton cycle temperature decreases, from 555°C at 100% load to 478°C at 70% load, seen in Figure 4.7.

The second control methodology uses inventory control with a throttling valve to maintain the maximum sCO2 Brayton cycle temperature at 550°C. Inventory control alone is not sufficient to maintain the 550°C temperature at lower loads as shown previously, which warrants the use of a throttling valve. The same inventory control methodology is used as the first method, with the same compressor inlet pressures for lower loads. However, to overcome the additional pressure drop across the throttling valve, the compressor pressure ratio increases, from 2.5 at 100% load to 2.8 at 70% load. This results in a higher maximum Brayton cycle pressure compared to using only inventory control. The pressure drop across the throttling valve increases from 595 kPa at 90% load to 4000 kPa at 70% load. The throttling valve is used to maintain the turbine inlet temperature at 550°C, as shown in the T-s diagram in Figure 4.7. This results in a decrease in sCO2 mass flow rate from 19.29 kg/s at 100% load to 12.09 kg/s at 70% load (-37.3%). The throttling valve allows for a further reduction in sCO2 mass flow rate compared to only using inventory control, while maintaining the 550°C maximum cycle temperature.

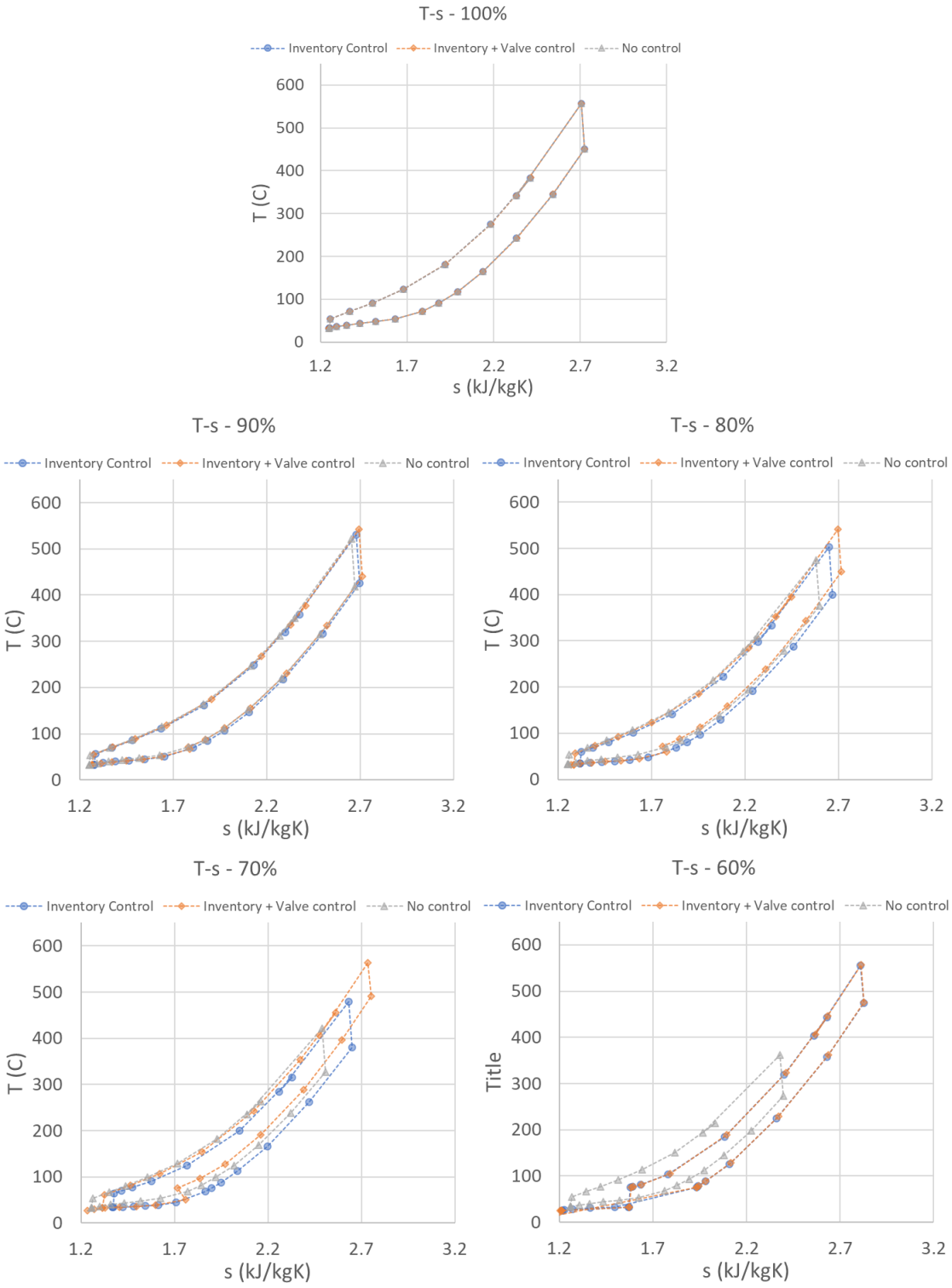


Figure 4.7: T-s diagrams for the sCO2 Brayton cycle (single configuration) at 100%-60% load for various control methodologies

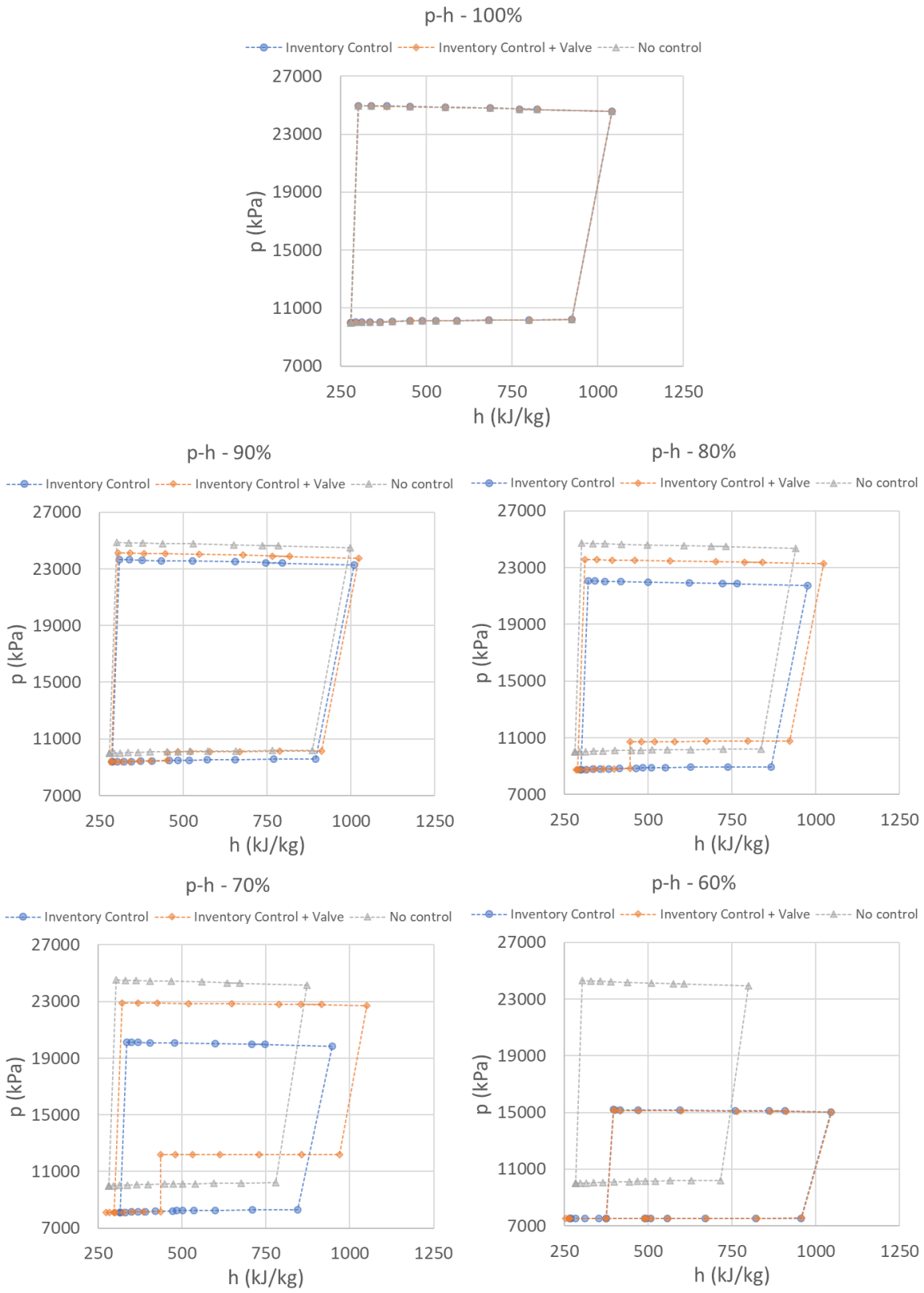


Figure 4.8: p-h diagrams for the sCO2 Brayton cycle (single configuration) at 100%-60% load for various control methodologies

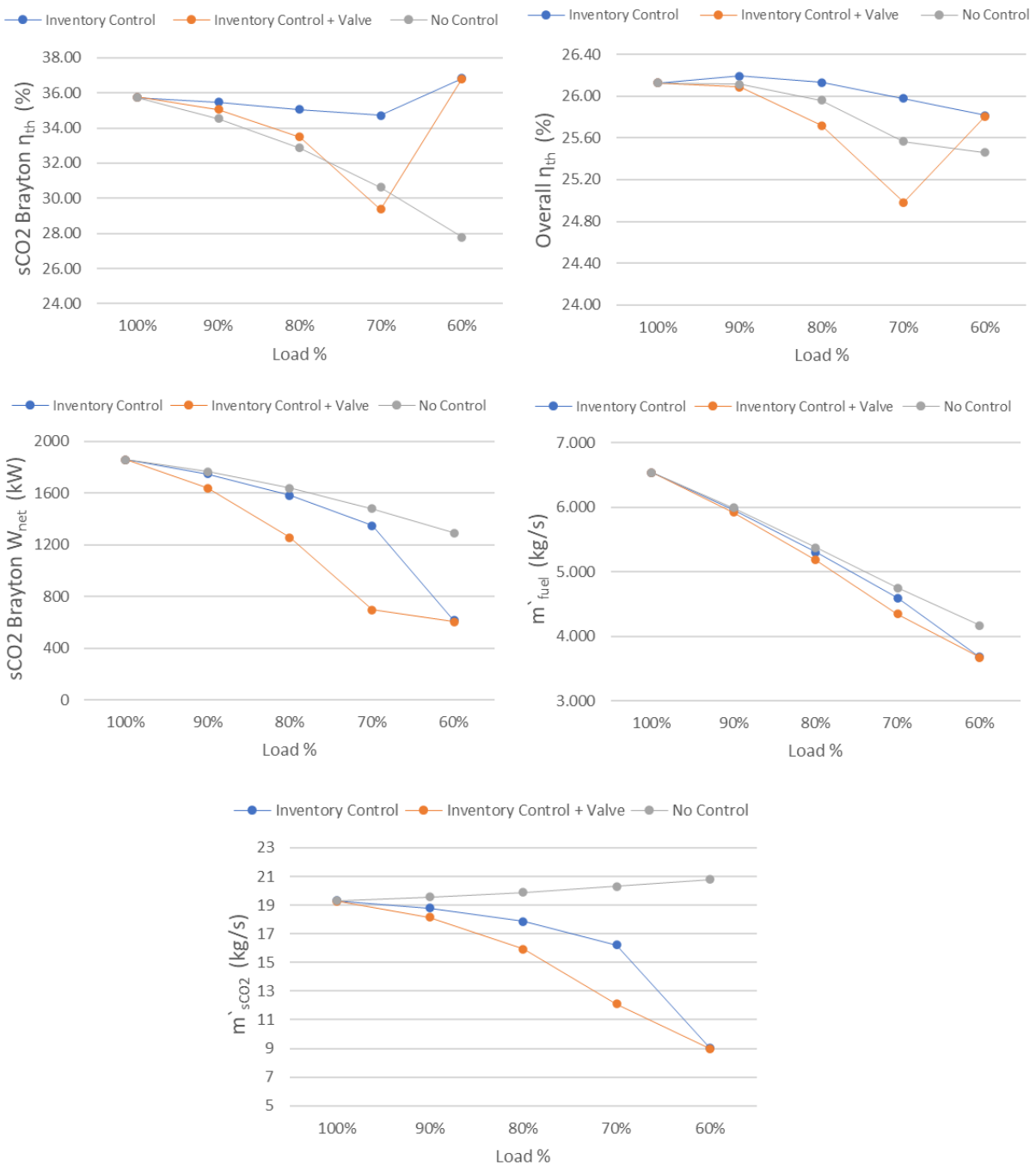


Figure 4.9: Results for various sCO2 Brayton cycle control methodologies at 100%-60% load.

Top left: sCO2 Brayton cycle efficiency, Top right: Overall integrated cycle efficiency, Middle left: sCO2 Brayton cycle net generation, Middle right: Fuel flow rate, Bottom: sCO2 Brayton cycle mass flow rate

With no control of the sCO2 Brayton cycle, there is no control on the Brayton cycle max temperature, and the compressor inlet pressure remains at 10 MPa for all loads as seen in the p-h diagram in Figure 4.8. With lower fuel flow rates required at lower Rankine cycle loads, the maximum temperature of the sCO2 Brayton cycle decreases due to less heat available from lower flue gas flow rates, from 550°C at 100% load to 362°C at 60% load, shown in the T-s diagram in Figure 4.7. The sCO2 mass flow rate in this case increases from 19.29 kg/s at 100% load to 20.30 kg/s at 70% load (+5.2%). Figure 4.10 shows the operating points for the sCO2 turbomachinery for each control methodology. For no control, the reduction in turbine inlet temperature, and consequent increase in density, results in a decreased pressure ratio over the turbine. Thus, the volume flow rate through the compressor increases for a reduced isentropic head and pressure ratio. This results in the slight increase in sCO2 mass flow rate for lower loads with no control in place.

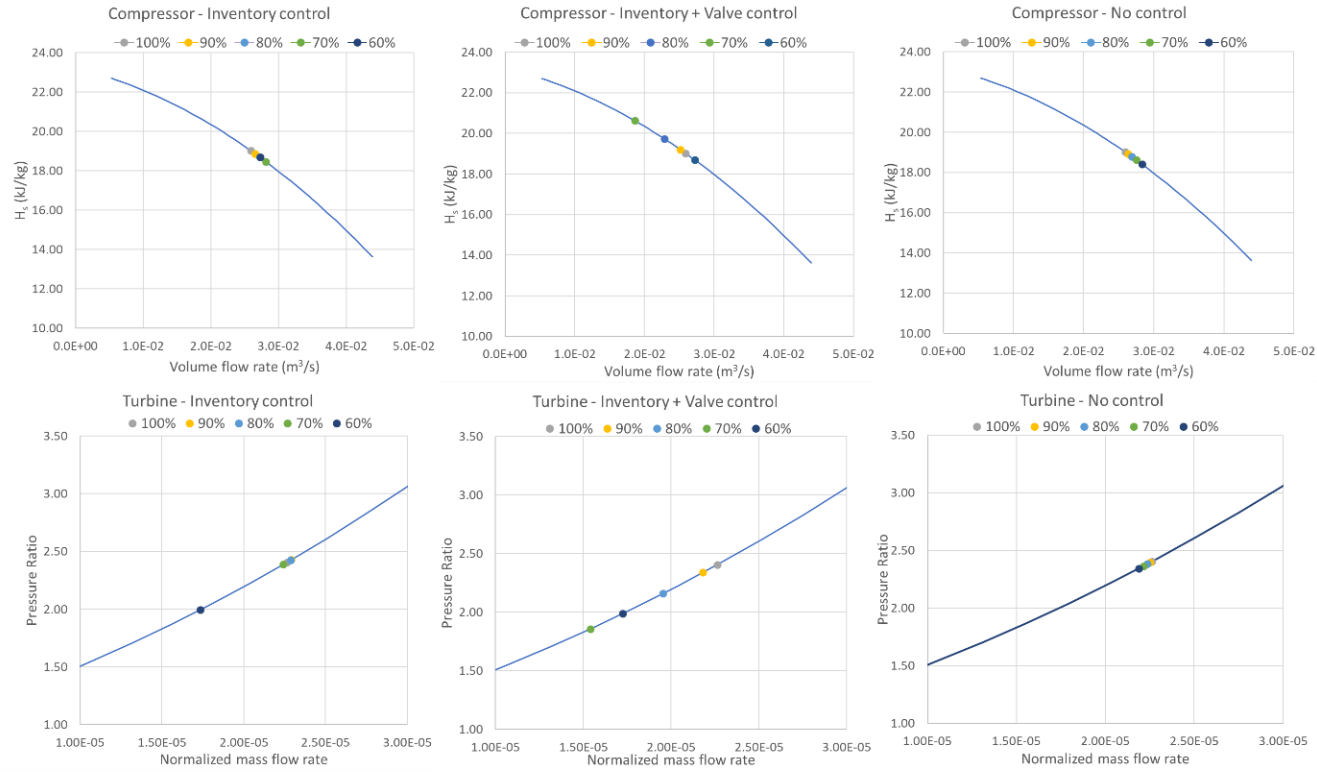


Figure 4.10: Operating point on turbomachinery curves at 100%-60% loads for each control methodology. Top: Compressor, Bottom: Turbine

At 100% load, $\eta_{th.Brayton}$ is 35.75%. At 70% load, $\eta_{th.Brayton}$ for inventory control is 34.71% (-1.04%), inventory control with throttling is 29.39% (-6.36%), and for no control 30.64% (-5.11%). This can also be seen at the top left of Figure 4.9. Inventory control alone maintains thermal efficiency better from nominal load down to 70% load compared to inventory control with the throttling valve. The

larger decrease in thermal efficiency is due to the pressure reduction and entropy generation across the throttling valve, which is required to maintain the cycle temperature at 550°C. Even though the Brayton cycle temperature is higher for the control strategy with the throttling valve, the increase in compressor work required to overcome the flow resistance of the throttling valve, as well as the reduced work done by the turbine, results in a substantial nett decrease in thermal efficiency. Lastly, for the case with no control, the reduced turbine inlet temperature results in a decrease in thermal efficiency at lower loads, demonstrating the need for a control methodology to maintain thermal efficiency at lower loads.

Notably, the sCO2 Brayton cycle with no control has a higher thermal efficiency at 70% load compared to using inventory control with a throttling valve. Comparing sCO2 Brayton net electrical power output at 70% load, the case with no control has the largest net electrical power output at 1480 kW but the lowest thermal efficiency (30.64%), the inventory control case has the 2nd largest net electrical power output at 1349 kW and the highest thermal efficiency (34.71%), and the throttling case has the lowest net electrical power output at 698 kW and the lowest thermal efficiency (29.39%). These results therefore show that inventory control alone maintains the highest thermal efficiency across the low load operating range while still providing nearly the same net electrical power output as the case with no control.

Observing the results at 60% load for inventory control and for the throttling valve cases, there is a large shift in performance for the sCO2 Brayton cycle. At 60% load for inventory control, the cycle low pressure is 7.5 MPa. At a compressor inlet temperature of 33°C, the compressor inlet operating conditions are close to the pseudo-critical point, which can be seen in Figure 4.11. Comparatively, for 100% - 70% load operation, the compressor inlet conditions are away from the pseudo-critical point for the respective compressor inlet pressures. However, for the 60% load case, there is a large reduction in compressor inlet density, from around 576 kg/m^3 at 70% load down to 332 kg/m^3 at 60% load (-42%).

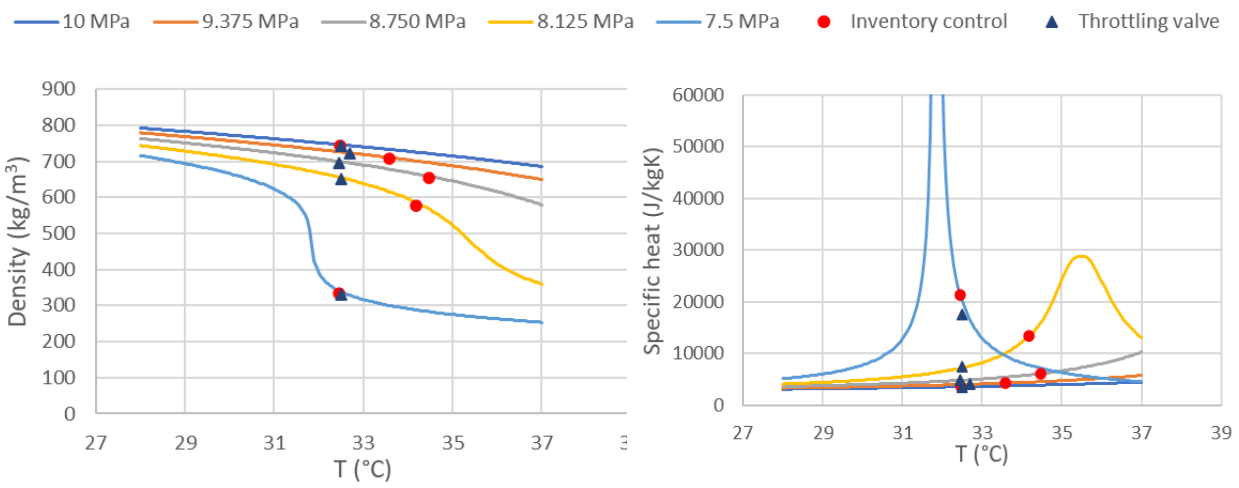


Figure 4.11: sCO2 density and specific heat at compressor inlet for various inlet pressures with inventory control and throttling control.

From Figure 4.10, the compressor volume flow rate for inventory control remains relatively the same between 70% and 60% loads. However, with the large reduction in density at the compressor inlet at 60% load, the mass flow rate through the compressor also reduces significantly, with $\dot{m}_{sCO2}$ decreasing by 44% compared to 70% load, as seen at the bottom of Figure 4.9. This also results in a substantial decrease in net electrical power output of the sCO2 Brayton cycle. Additionally, the compressor pressure ratio drops from 2.5 to 2.0 at 60% load. However, with the decrease in sCO2 mass flow rate, the maximum sCO2 cycle temperature increases, from 478°C at 70% load to 555°C at 60% load. Notably, 555°C is close to the nominal maximum temperature of the sCO2 Brayton cycle. This also results in a thermal efficiency increase, from 34.7% at 70% load to 36.8% at 60% load.

For the inventory control with throttling similar results are observed. However, as throttling is used to reduce the sCO2 cycle flow rate, there is a smaller reduction in $\dot{m}_{sCO2}$ from 70% to 60% load (-25.7%). Due to the significant reduction in sCO2 mass flow rate at 60% load, the sCO2 maximum cycle temperature automatically reaches the nominal temperature of 555°C. As a result, throttling at the 60% load case is not required, and the throttling valve is fully open. However, this implies that a quick change in operation is required down from 70% load, where the throttling valve is partially closed.

Effectively, cycle operation near the pseudo-critical point results in a large reduction in cycle capacity and generation, however, with an increase in thermal efficiency. Importantly, these results demonstrate that reducing the minimum cycle pressure through inventory control must be carefully

managed. Approaching the pseudo-critical point through inventory control at low loads results in large fluctuations in cycle performance and operation. Compressor surge may also become problematic with the sharp decrease in mass flow rate through the compressor. It may be beneficial to increase the cycle minimum pressure for 60% load to avoid approaching the pseudo-critical point at 32.5°C, or similarly, raise the cycle low temperature. This may however impact negatively on the efficiency at nominal load.

While dynamic modelling is outside the scope of this study, the transient response of different control strategies should also be considered when selecting a control methodology. Typically, inventory control has a slower response time compared to valve control as seen in literature. For effective control of the sCO2 Brayton cycle, a coupled control strategy will be required. The throttling valve and attemperation valve can be used for fast control, and inventory control for slower response times like load-following while maintaining high thermal efficiency. This is the typical approach used in several studies [35], [41]–[43]. For the purposes of this study, inventory control is selected as the control strategy for further investigation.

4.2.3 Single and dual sCO2 heater configurations

The single heater and dual heater sCO2 configurations were compared with inventory control selected as the control methodology for the sCO2 Brayton cycle. Both sCO2 heater configurations have been previously compared 100% load, with the dual heater configuration having a bigger impact on Rankine cycle attemperation due to the placement of both radiative and convective heaters before SH2 and SH1 respectively.

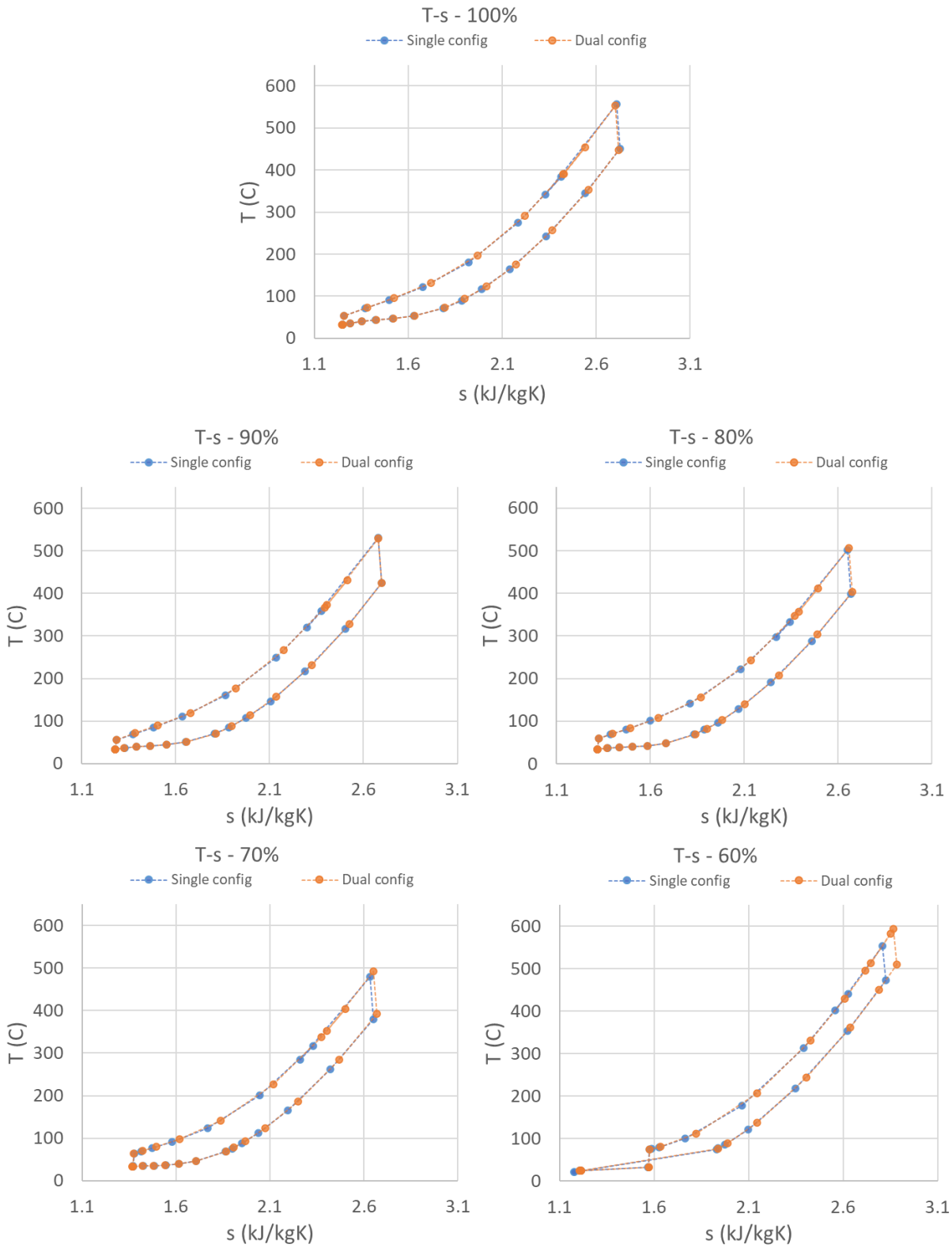


Figure 4.12: T-s diagrams for the sCO2 Brayton cycle – single heater and dual heater configurations - at 100%-60% load

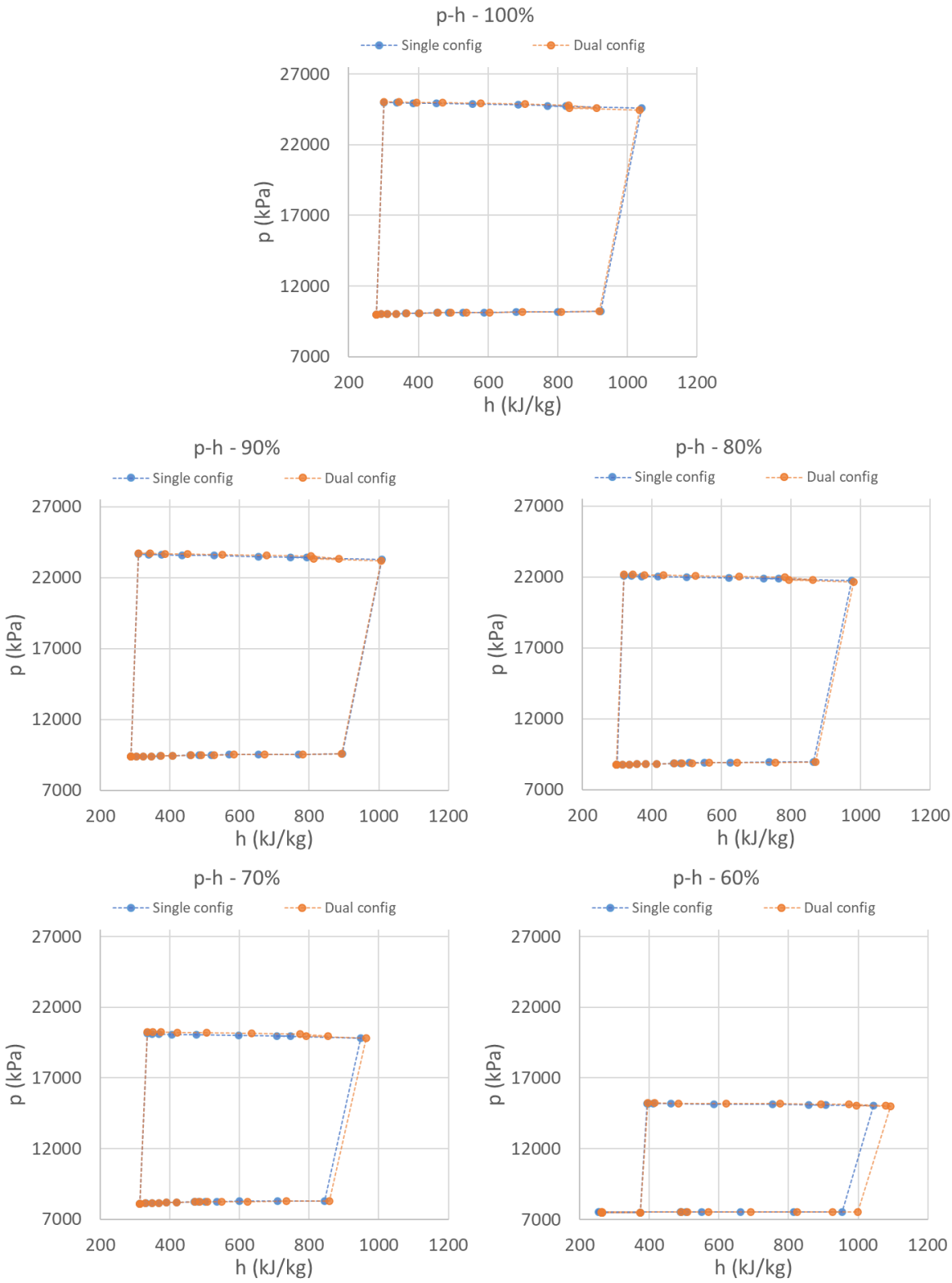


Figure 4.13: p-h diagrams for the sCO2 Brayton cycle – single heater and dual heater configurations - at 100%-60% load

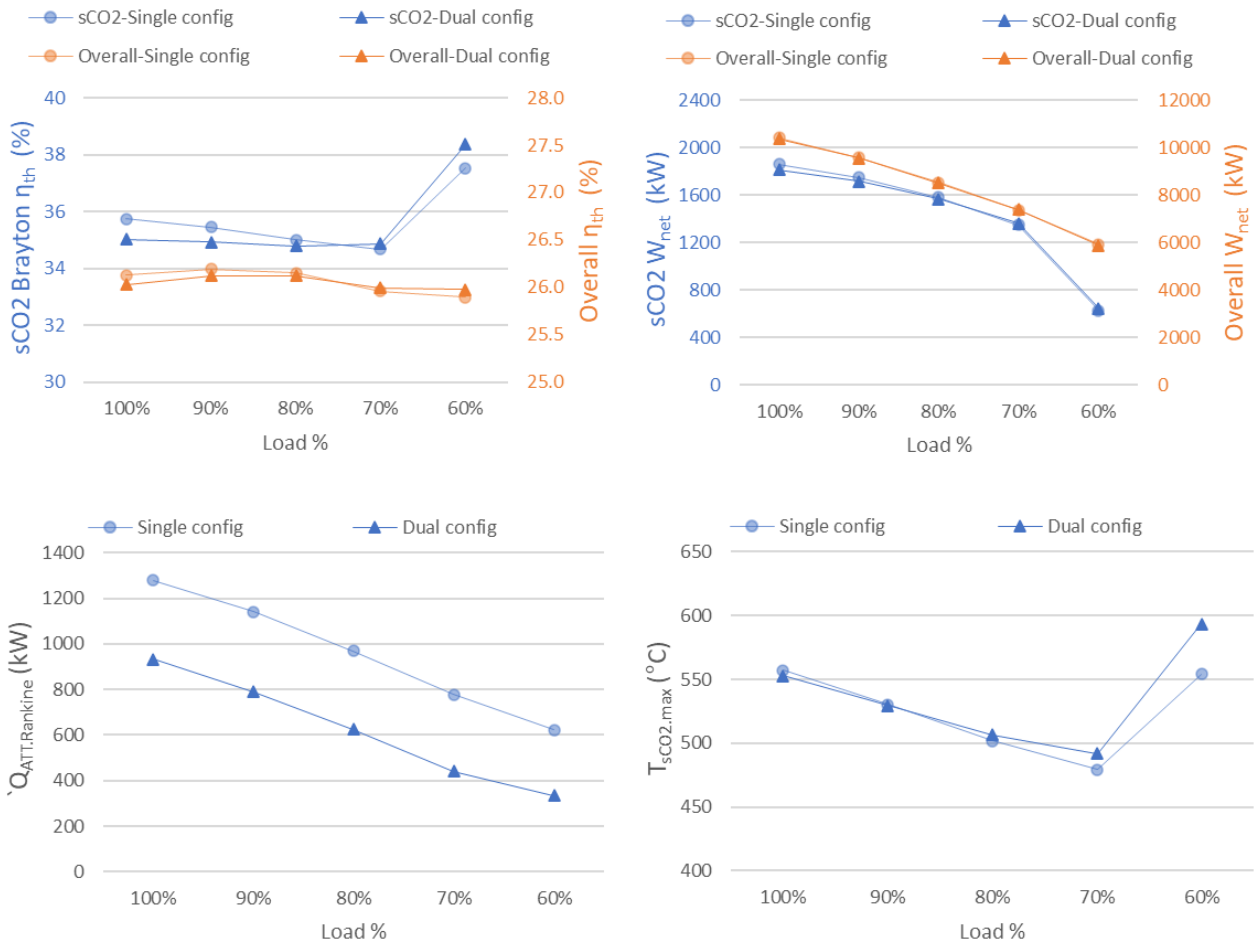


Figure 4.14: Results for single heater and dual heater sCO2 Brayton cycle configurations at 100%-60% load.

Top left: Thermal efficiency, Top right: Net generation, Bottom left: Rankine cycle attemperation, Bottom right: sCO2 Brayton cycle maximum temperature

Figure 4.12 and Figure 4.13 show the T-s and p-h diagrams respectively for the single and dual heater configurations at 100% - 60% load. Additionally, Figure 4.14 shows various results for the single and dual heater configurations at varying loads, including thermal efficiency, net electrical power output, Rankine cycle attemperation, and sCO2 Brayton cycle maximum temperature. For further insight, Appendix C shows tabulated pressures and temperatures for both the single and dual heater configurations at 100%-60% load.

As the sCO2 heaters have been similarly sized for comparison purposes, there are only slight differences in high level results which become more apparent at lower loads. This can be seen in the T-s and p-h diagrams in Figure 4.12 and Figure 4.13, where the single and dual heater

configurations exhibit similar cycle performance at 100%-80% loads. Additionally, net electrical power output and thermal efficiency are also relatively the same. At 70% and 60% loads, the dual heater configuration has a higher maximum sCO2 Brayton cycle temperature compared to the single heater configuration, +13°C and +40°C for 70% and 60% load respectively, shown in the bottom right of Figure 4.14. The fuel flow rates are relatively the same between the two heater configurations for all load percentages.

For the dual configuration, the radiative-dominant heater absorbs direct radiation from the furnace. For low loads, heat uptakes at the convective-dominant sCO2 heaters (sCO2H2/X) are adversely impacted more than the radiative dominant sCO2H1. From 100% - 60% load, there is a 38% decrease in heat uptake at the radiative-dominant sCO2H1, while there is a 77% and 67% decrease in heat uptake for the convective-dominant sCO2H2 and sCO2HX respectively. Therefore, at low loads direct radiation absorbed at sCO2H1 for the dual configuration makes up a higher percentage of sCO2 heat uptake, and results in a higher maximum sCO2 cycle temperature at 70% and 60% loads. Additionally, this leads to an increase in thermal efficiency of the sCO2 Brayton cycle for the dual heater configuration at 70% and 60% loads (34.9% and 38.4%) compared to the single heater configuration (34.7% and 37.5%).

With the dual configuration, the direct radiation absorbed by sCO2H1 results in reduced heat uptakes at SH2. A reduction of Rankine cycle attemperation also follows, shown on the bottom left of Figure 4.14. At 60% load, the single heater configuration has 662 kW of attemperation, compared to 335 kW for the dual configuration. A reduction in attemperation may impede effective control of SH2 outlet temperatures, particularly at lower loads.

These results show that the dual configuration performs slightly better at lower loads compared to the single heater configuration. However, the slight increase in thermal efficiency of the sCO2 Brayton cycle may be negated by the reduced ability to control the Rankine cycle maximum temperature via attemperation. Importantly, fuel variability as well as boiler cleanliness can potentially further impact heat uptakes for the Rankine cycle superheaters. As the dual configuration already has less attemperation, a further reduction may result in reduced SH2 outlet temperatures, which would adversely affect Rankine cycle performance.

Conservatively, the single heater configuration may be more feasible to increase energy efficiency and net electrical power output over the full load range via the integration of the sCO2 supplementary cycle.

4.2.4 Summary of results

The results show that inventory control is the most effective control methodology for slow load following, maintaining high thermal efficiency for a wide load range. In comparison, the use of a throttling valve to maintain the maximum cycle temperature results in a decrease in thermal efficiency and net electrical power output due to the additional pressure loss and reduced turbine work.

Using inventory control at 60%, compressor inlet conditions are near the pseudo-critical point. This results in large fluctuations in cycle performance, with a sharp decrease in sCO2 Brayton cycle capacity due to the density of sCO2 nearly halving compared to higher load cases. Comparing the two sCO2 heater configurations controlled via inventory control, it was found that the dual heater configuration performed marginally better at lower loads due to the smaller reduction in radiative heat transfer at low loads. However, with the dual configuration, both SH1 and SH2 for the Rankine cycle are adversely affected, demonstrated by the reduced attemperation requirement.

5. Biomass boiler CFD modelling: Materials and Methods

Empirical models commonly used for 1D modelling of combustion and radiative heat transfer lack accuracy compared to detailed CFD modelling, mainly due to the spatial characteristics of these phenomena inside a boiler furnace. CFD modelling can be used to capture detailed combustion modelling phenomena, as well as radiative and convective heat transfer for boilers. The aim of the CFD modelling in this research was to provide validation of the heat uptakes calculated by the 1D model for the sCO2 heaters. With overfiring required for sCO2-integrated configurations, CFD modelling may also provide further understanding of overfiring implications. As CFD modelling can capture spatial characteristics of the relevant phenomena in the boiler, and with the lack of experimental data, CFD modelling may be used to provide validation of the results for the 1D process model, particularly heat uptakes at the sCO2 heaters. Additionally, CFD model results can be used to further tune empirical parameters and heat exchanger performance of the 1D process model. An iterative tuning methodology between the CFD models and 1D process models may be beneficial, however this lies outside the scope of the current work and is recommended for future improvements to the models. A single tuning iteration is explored in this work.

This chapter details the materials and methods applied in the development of the CFD model for the selected biomass boiler, including the relevant governing equations and modelling procedure. In the next chapter, chapter 6, the CFD modelling results are presented, including results at nominal and part loads. In chapter 7 comparisons between CFD and integrated 1D process model are made. The 1D model is then tuned accordingly and updated results presented.

Similar to the 1D process models, CFD models were developed for three cases, namely, the existing Rankine cycle without the sCO2 Brayton cycle integration, the integrated cycle with the single sCO2 heater configuration, and the integrated cycle with the dual sCO2 heater configuration. Each of these cases were simulated at 100%, 80%, and 60% loads, resulting in a total of nine models. The inputs/boundary conditions for the CFD models were derived from the 1D process model.

The case study boiler was presented in section 3.1. However, additional geometry considerations are required for the CFD model, as presented in the following section. The fuel composition for the typical Southern African bagasse was previously presented in Table 3.1. For the CFD models, the ultimate analysis (DAF) and proximate analysis (AR) are presented in Table 5.1. The proximate analysis was derived from a study by Laubscher and van der Merwe [57], which also used Southern African bagasse, and determined proximate analysis from fuel sample tests. Moisture and ash content is known for the bagasse used in this study. The percentage weighting of volatile matter

and fixed carbon from Laubscher and van der Merwe's [57] analysis was used to determine the proximate analysis for this study. The HHV of 8838 kJ/kg was used [80], the same as for the 1D process model.

Table 5.1: Bagasse ultimate and proximate analysis

	Constituent	C	H	O	N	S
Ultimate analysis (DAF)	Mass fraction (%)	47.88	5.91	45.82	0.35	0.04
	Constituent	Volatile matter	Fixed carbon	Moisture	Ash	
Proximate analysis (AR)	Mass fraction (%)	39.5	5.8	50.0	4.7	

The CFD modelling in this research was conducted using ANSYS® Fluent 2023 R1. Two user-defined sub-models developed in C were used. The Eulerian-Lagrangian approach is used for the model. The Eulerian description of the gas phase solves the governing conservation equations using a finite volume approach with a stationary frame of reference. On the other hand, the solid fuel particles are modelled in a Lagrangian (discrete) reference frame, which tracks the particle trajectories moving through the computational domain. The Eulerian phase is modelled using the species transport approach, where a species conservation equation is solved for each species present in the mixture excluding N_2. The gas mixture defined for the gas phase uses the following species: O_2, CO_2, H_2O, CO, SO_2, N_2, and $C_XH_YO_ZN_QS_P$ for the pseudo-volatile species. Source term coupling was used to account for the mass, momentum, species, and energy transfer between gas and discrete phases.

5.1 Domain, numerical mesh, and inputs

Figure 5.1 shows the boiler geometry for the existing configuration without the sCO2-integration, as well as various boundary conditions. The focus of the CFD modelling is capturing the effects of the sCO2 heater integration into the boiler, using the existing configuration as a baseline. Therefore, to reduce computational expense, the boiler is modelled up to CAV2. Only SH2 and the sCO2 heaters are modelled with detailed geometry, while other boiler heat exchangers, particularly SCR1 and SCR2, are modelled as porous media with accompanying heat and momentum sources. Boiler and furnace membrane water walls as well as refractory are modelled as planar surfaces. For the grate, in practice, zoning dampers are used to force $\approx$60% of primary air through $\approx$40% of the grate area towards the rear wall, with the remainder of primary air through the front $\approx$60% of the grate. The

grate is modelled using five porous media volumes, with the two volumes closest to the rear wall accounting for 40% of the grate.

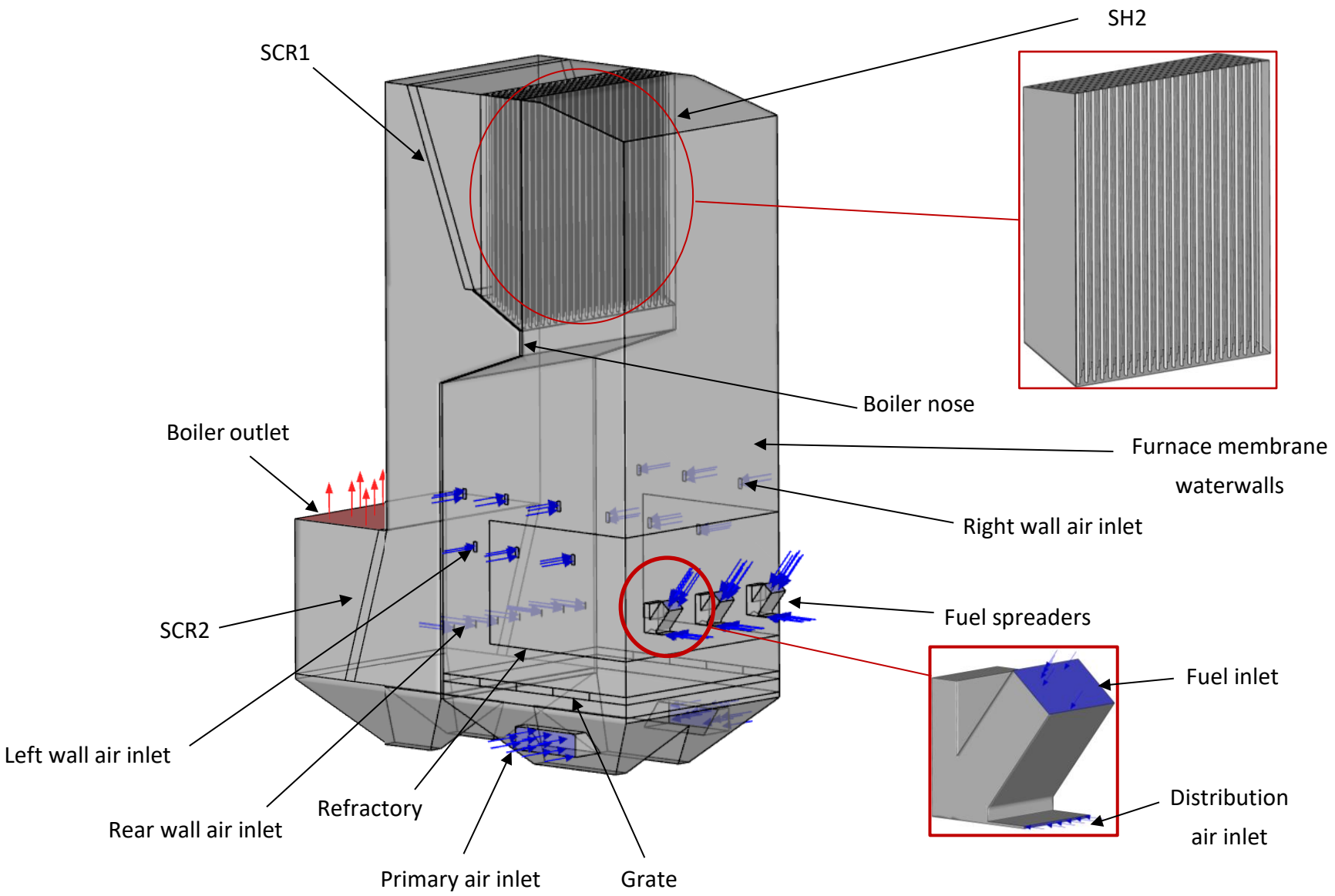


Figure 5.1: Boiler geometry for existing configuration

Bagasse is fed into the furnace through three fuel spreaders. Each spreader has two inlets, including a fuel inlet for bagasse particles fed by gravity, and a distribution air inlet to spread the bagasse particles towards the furnace rear. Primary air is injected through the bottom of the furnace. Secondary air is injected through six nozzles on the right and left walls, as shown in Figure 5.1. Distribution air is injected through seven nozzles at the rear wall, in addition to the fuel spreaders. SH2 is modelled using explicit vertical tube walls stemming from the boiler roof towards the nose. In practice, SH2 tubes serpentine, however, to simplify the meshing procedure, vertical tubes were assumed, allowing for the bends at the bottom of the tubes to be ignored which assisted in ensuring a pure hexahedral mesh can be achieved. For the existing configuration, a cavity, CAV1 is modelled in absence of a convective sCO2 heater.

Figure 5.2 shows the boiler geometry for the single heater configuration. The sCO2 convective heater for the single heater configuration, referred to as sCO2HX, is modelled using explicit horizontal tubes between the side walls. Similar to SH2, bends were ignored to simplify the mesh. sCO2HX is sized such that the tube bank does not extend into the cavity below the heater, bCAV1. This is done to minimise dead spots in the heater, while maximising available space in the cavity sCO2HX is positioned in. The sCO2 heaters are proposed with a counter-current design.

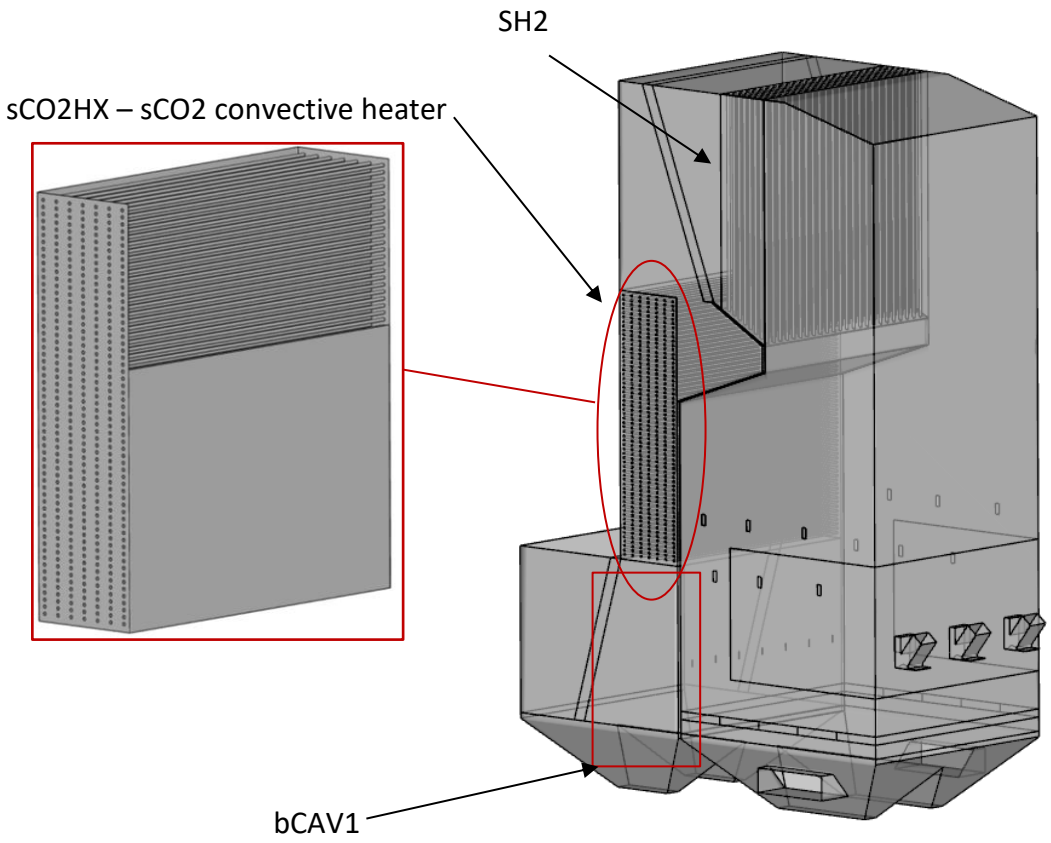


Figure 5.2: Boiler geometry for the single heater configuration

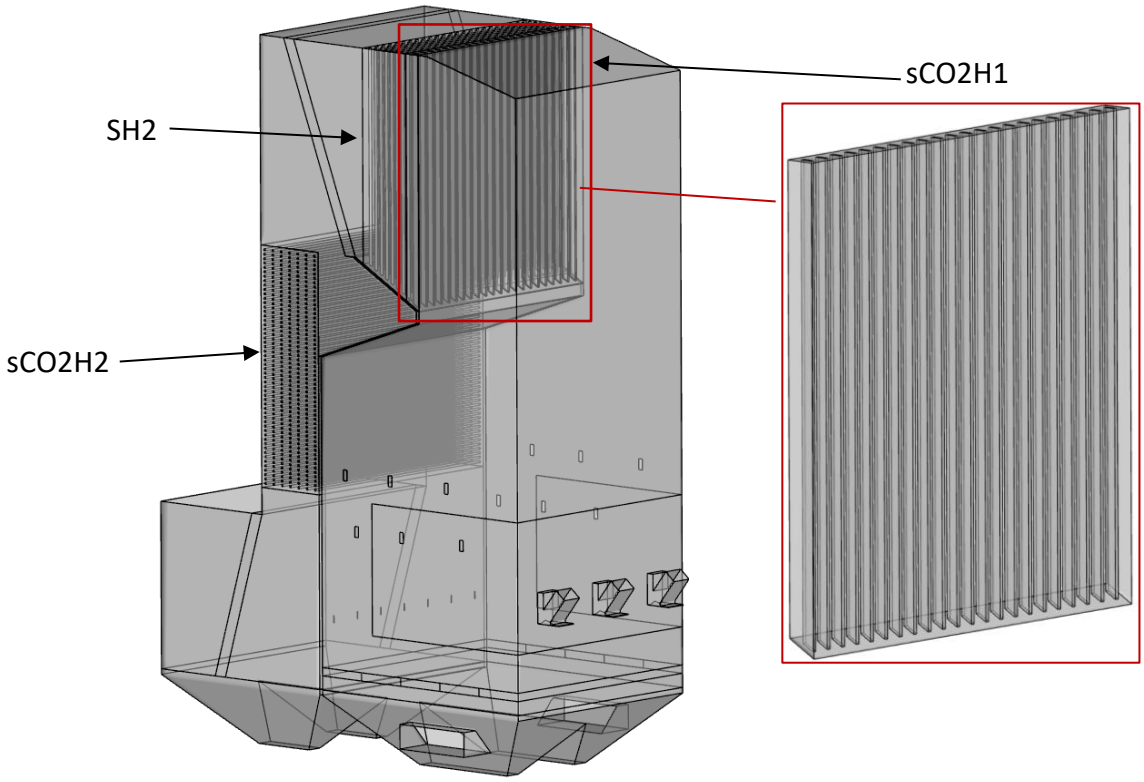


Figure 5.3: Boiler geometry for the dual heater configuration

Figure 5.3 shows the boiler geometry for the dual sCO2 heater configuration. For the radiative platen sCO2H1, tubes are modelled as flat plates in the longitudinal (horizontal) direction. A similar approach is used as SH2, where vertical tubes are modelled from the roof. For the dual heater configuration, a slight extension of the boiler nose is required to accommodate sCO2H1 to avoid dead spots at the upper half of the heater, and to shield the heat exchanger from high radiative heat fluxes from the furnace. This includes extrusion of the nose to form the required space, shown in Figure 5.4. Practically, this results in an increase in the furnace exit plane area, as well as slightly increased furnace water wall area. The convective heater for the dual configuration is referred to as sCO2H2. The dual heater configuration has a smaller convective heater (sCO2H2) compared to the single configuration (sCO2HX), as established from the 1D process model results to reduce adverse impacts on the Rankine cycle, particularly the superheaters.

The existing configuration mesh had 11.3 million cells. For the single heater configuration, the final mesh consisted of 14.6 million cells, while the dual heater configuration mesh had 15.9 million cells due to the additional refinement of sCO2H1 and SH2. For the single and dual heater configurations, grid independence tests were conducted using three mesh sizes, shown in Table 5.2.

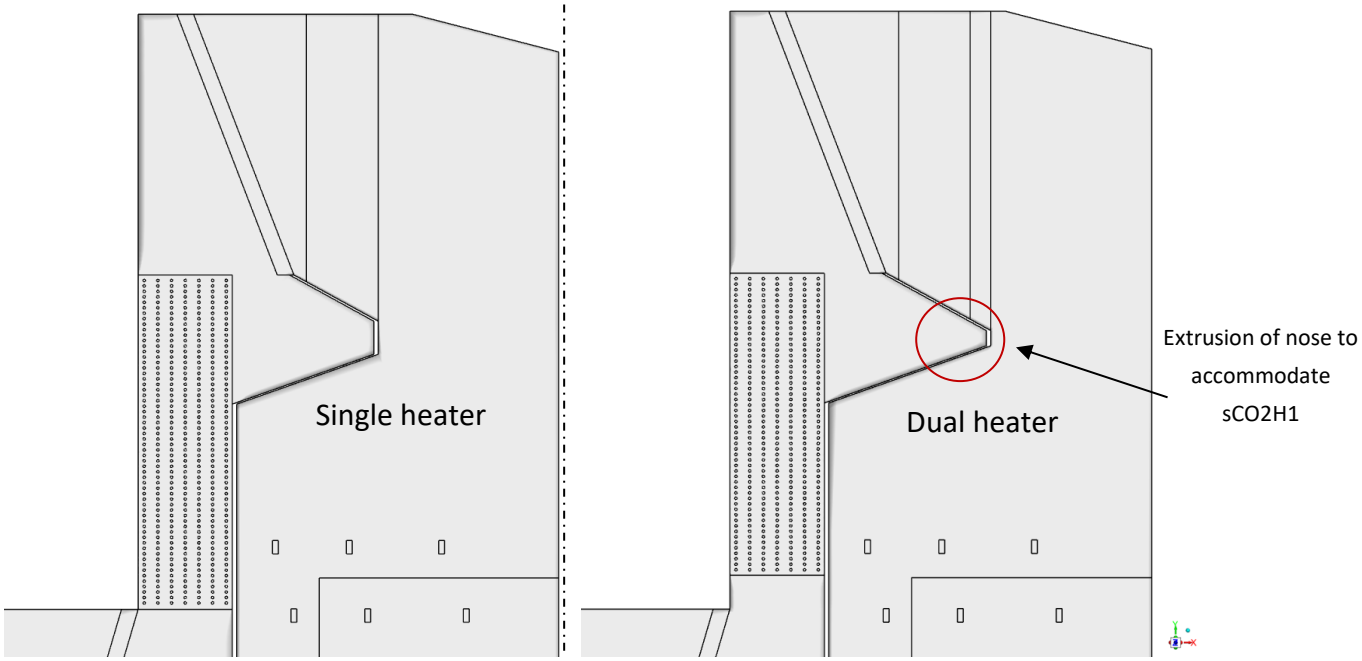


Figure 5.4: Adjustments to nose for dual heater configuration (right) compared to original nose for single heater configuration (left)

Table 5.2: Mesh cell counts for grid independency tests

		Cell count		
		Mesh 1 (M1)	Mesh 2 (M2)	Mesh 3 (M3)
Single heater configuration mesh	Cell count	11.8m	13.3m	14.6m
	x_{mesh} (mm)	0.0104	0.0092	0.0084
Dual heater configuration mesh	Cell count	12.0m	13.6m	15.9m
	x_{mesh} (mm)	0.0103	0.0095	0.0078

For grid independence, a representative cell size, x_{mesh} (m), was calculated using Eq. (5.1) [81], with results shown in Table 6.1:

$$x_{mesh} = \left(\frac{1}{N} \sum_{i=1}^{N} V_i \right)^{1/3} \tag{5.1}$$

where N is the number of cells in the domain, and V_i the cell volume for the ith cell.

The wall fluxes, particularly for the sCO2 heaters, were compared, and the fine mesh, M3, was deemed suitable for both configurations. Consequently, the existing cycle configuration adopts a similarly fine mesh. To ensure numerical stability, mesh orthogonal quality was kept above 0.15, and skewness below 0.9. To ensure convective heat transfer at the walls of the sCO2 heaters and SH2 was correctly captured, a 3-layer and 5-layer cell inflation was adopted respectively, with a first layer thickness of 5mm. This resulted in an area weighted average $Y^+ \approx 40$, which is suitable for the standard wall function used in this study, valid for $30 \leq Y^+ < 300$ [82]. The alternative of accurately resolving the viscous sublayer through mesh refinement would require a significantly finer mesh which would introduce substantial computational expense at this scale.

Table 5.3 shows the various inputs and boundary conditions used for the CFD models. These inputs were extracted from the 1D process model results. Mass flow rate boundary conditions were used for the various air inlets. These include spreader/distribution air inlets, primary air inlets, side wall air nozzles, and rear wall air inlets. Air temperatures were specified for each inlet, with distribution air temperature constant for all models. The fuel flow rate was specified for the particle injection stream. The convective thermal boundary condition was used for tube walls for the explicitly modelled tube bank heat exchangers, as well as for boiler water walls. This requires conductivities of the tube and ash layers, as well as the average internal fluid temperature and heat transfer coefficient. This is specified in Table 5.3 for SH2, sCO2H1, and sCO2HX/2. As screens were modelled as porous media, energy source terms for the screen volumes were specified to capture heat transfer. These are found from the 1D process model and divided by the specified duct volume to determine the volumetric energy source term. For the evaporator water walls, the steam saturation temperature of $T_w = 285\,°C$ was used for the internal fluid temperature, applied using the convective thermal boundary condition.

Table 5.3: Inputs and boundary conditions for CFD models

	Single heater configuration			Dual heater configuration			Existing configuration		
Load %	100%	80%	60%	100%	80%	60%	100%	80%	60%
Fuel flow rate (kg/s)	6.510	5.276	3.675	6.533	5.302	3.698	5.664	4.552	3.407
Total primary air inlet (kg/s)	10.394	8.250	5.467	10.432	8.291	5.504	9.043	7.111	5.121
Side wall air nozzle inlet (kg/s)	4.113	3.245	2.121	4.127	3.262	2.136	3.578	2.797	1.993
Total secondary air inlet (kg/s)	8.225	6.492	4.243	8.255	6.525	4.272	7.156	5.594	3.986
Spreader air inlet (kg/s)	1.154	1.154	1.154	1.158	1.158	1.158	1.004	1.004	1.004
Rear wall air nozzle inlet (kg/s)	0.687	0.687	0.687	0.690	0.690	0.690	0.598	0.598	0.598
Total distribution air inlet (kg/s)	1.841	1.841	1.841	1.848	1.848	1.848	1.602	1.602	1.602
$T_{PA/SA}$ (°C)	208.2	200.6	196.8	210.4	202.8	198.7	214.3	206.5	199.8
$h_{SH2}\left(\frac{\mathrm{W}}{\mathrm{m^2K}}\right)$	1579.7	1338.5	1060.8	1565.5	1321.9	1050.7	1568.7	1323.6	1054.1
$T_{steam,avg,SH2}$ (°C)	401.8	394.6	395.4	411.0	406.2	406.8	408.7	404.9	401.1

	Single heater configuration			Dual heater configuration			Existing configuration		
Load %	100%	80%	60%	100%	80%	60%	100%	80%	60%
$h_{sCO2H1}\left(\frac{W}{m^2K}\right)$	-	-	-	3230.3	2977.4	1729.1	-	-	-
$T_{avg,sCO2H1}$ (°C)	-	-	-	423.1	380.8	543.3	-	-	-
$h_{sCO2HX/2}\left(\frac{W}{m^2K}\right)$	9385.6	8613.9	4879.1	9328.4	8569.1	4859.6	-	-	-
$T_{avg,sCO2HX/2}$ (°C)	405.9	358.9	464.0	473.3	433.3	556.5	-	-	-
$E_{source,SCR1}$ (kW/m^3)	-304.0	-254.2	-182.4	-294.0	-244.3	-176.2	-272.2	-224.1	-169.6
$E_{source,SCR2}$ (kW/m^3)	-151.6	-111.6	-87.1	-163.1	-123.5	-99.0	-193.8	-152.3	-107.5

The sections below discuss the relevant governing equations and modelling procedure used for the CFD model.

5.2 Gas phase modelling

As previously discussed, the gas mixture transport is modelled using a Eulerian approach. Equations for discrete phase modelling are presented thereafter.

Since the flow being modelled is highly turbulent, the Reynolds averaged form of the Navier-Stokes (RANS) equations are used. The steady-state form of the differential continuity equation is written as:

$$\frac{\partial}{\partial x_i}(\rho_g \bar{u}_i) = S_p \tag{5.2}$$

where S_p is the continuity source term, accounting for mass transferred between the discrete particle phase to the gas phase, and $\bar{u}_i$ the mean velocity vector in Cartesian coordinates. S_p is calculated as:

$$S_p = \sum_{i=1}^{N_p} \frac{dm_p}{dt}\frac{1}{V} \tag{5.3}$$

where N_p is the number of particle parcels for a given cell, $\frac{dm_p}{dt}$ is the rate of mass release due to combustion related processes, such as moisture evaporation, devolatilization, or char combustion, and V is the volume of the cell. In Eq. (5.2), ρ_g is the density of the gas mixture calculated using the ideal gas law, which depends on the mixture pressure, temperature, and species concentration.

The RANS form of the momentum equation is given as:

$$\frac{\partial}{\partial x_i}\rho_g u_i u_j + \frac{\partial \bar{P}}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3}\delta_{ij}\frac{\partial u_i}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_i}(-\rho_g \overline{u_i' u_j'}) + S_{fp} + S_{pm,k} \tag{5.4}$$

In this work, the Reynolds stress term, $(-\rho_g \overline{u_i' u_j'})$, is related to the mean velocity gradient using the Boussinesq model. The Boussinesq equation together with the RANS governing equations were closed, as mentioned, with a two-equation turbulence model. This method has been successfully implemented for various boiler analyses [3], [57]. In Eq. (5.4), S_{fp} is the drag force source term due to the relative velocity difference between gas and particle phases. $S_{pm,k}$ is the momentum source term for a given porous media zone, k. These source terms are expanded upon in subsequent sections below.

The RANS form of the energy equation is given as:

$$\frac{\partial}{\partial x_i}(u_i[\rho_g E + P]) = \frac{\partial}{\partial x_j}\left[\lambda_l \frac{\partial T_g}{\partial x_j} + \rho_g \overline{u_j' E'} + u_i \tau_{eff}\right] + S_h + S_{h,pm,k} \quad (5.5)$$

where E is the total energy of the fluid, and λ_l the laminar thermal conductivity. S_h is the energy source term accounting for various convective and radiative heat transfer mechanisms between particles, energy released by gas phase reactions, and additional energy source/sink terms. $S_{h,pm,k}$ is the energy source term for a given porous media zone, k . Both SCR1 and SCR2 are modelled as porous media. $\rho_g \overline{u_j' E'}$ in Eq. (5.5) is the turbulent energy flux, and τ_{eff} the viscous dissipation term, discussed later on. The total energy term is defined as:

$$E = h - \frac{P}{\rho_g} + \frac{u^2}{2} \quad (5.6)$$

where $h = \sum_{k=1}^{N} Y_k h_k$, with k referring to an individual species, and Y_k the corresponding mass fraction.

The species transport approach accounts for the transport, creation, and destruction of various gas mixture constituents throughout the boiler gas domain. For a species mass fraction, Y_k , the species transport equation is defined as:

$$\frac{\partial}{\partial x_i}(\rho_g u_j Y_k) = -\frac{\partial}{\partial x_j}(\vec{J}_k) + \dot{\omega}_k + S_{rp} \quad (5.7)$$

where $\dot{\omega}_k$ is the net rate of creation (or destruction) of the respective species due to chemical reactions calculated using a turbulence-chemistry interaction model, and $S_{rp} = \sum_{i=1}^{N_p} \frac{dm_{p,k}}{dt}\frac{1}{V}$ is the rate of species creation from the release of gaseous components from the particle phase. $\frac{dm_{p,k}}{dt}$ represents the rate of mass release of species k , which is calculated with the aid of evaporation, devolatilization, or char oxidation rate equations. $\vec{J}_k$ is the mass diffusion flux. For turbulent flows, it is calculated as:

$$\vec{J}_k = -\left(\rho_g \vec{D}_{k,m} + \frac{\mu_t}{Sc_t}\right)\frac{\partial Y_k}{\partial x_j} \quad (5.8)$$

The turbulent Schmidt number, Sc_t , is set to a default of 0.7 in Fluent v23.1 [82]. To accurately capture the diffusion coefficient, $\vec{D}_{k,m}$, for species k into mixture m requires a computationally expensive calculation. Therefore, for this work, the constant dilute phase approximation value of

$D_{k,m} = 2.88\text{x}10^{-5}$ m^2/s is used. This assumption can be made due to the dominance of turbulent diffusion, represented by $\frac{\mu_t}{Sc_t}$, in the furnace and around boiler heat exchangers.

5.2.1 Turbulence

In this work, turbulence is modelled with the RANS approach, using a two-equation realizable $k-\varepsilon$ model [82]. The realizable $k-\varepsilon$ is well-suited for modelling reactive flows, and has been used successfully for boiler analyses [57], [83], as well as biomass particle combustion [84]. Compared to the standard $k-\varepsilon$, the realizable model uses an enhanced model dissipation rate equation, which better captures separation and swirling flows, typically experienced close to fuel spreaders or air nozzles. The steady-state realizable $k-\varepsilon$ transport equations for kinetic energy and dissipation rate are given as:

$$\frac{\partial}{\partial x_j}(\rho_g k u_j) = \frac{\partial}{\partial x_i}\left[\left(\mu_l + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k G_b - \rho_g \varepsilon - Y_M \tag{5.9}$$

$$\begin{aligned}&\frac{\partial}{\partial x_j}(\rho_g \varepsilon u_j) = \frac{\partial}{\partial x_i}\left[\left(\mu_l + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \rho_g S C_1 \varepsilon - \rho_g C_2 \frac{\varepsilon^2}{k+\sqrt{\vartheta\varepsilon}} + C_{1\varepsilon}\frac{\varepsilon}{k}C_{3\varepsilon}G_b \\ &C_1 = \max(0.43, \frac{\eta}{\eta+5});\quad \eta = S\frac{k}{\varepsilon};\quad S = \sqrt{2S_{ij}S_{ij}}\end{aligned} \tag{5.10}$$

where G_k represents the generation of turbulence kinetic energy due to mean velocity gradients, G_b is the turbulence generation term due to buoyancy forces, and Y_M represents the contribution of fluctuating dilatation of compressible turbulent flows. The constants, $C_{1\varepsilon}$, C_2, σ_k, and σ_ε are set to 1.44, 1.9, 1.0, and 1.2 respectively [82]. The turbulent viscosity is expressed as a function of k and ε:

$$\mu_t = \rho_g C_\mu \frac{k^2}{\varepsilon} \tag{5.11}$$

The Boussinesq approximation of Reynolds stresses in the momentum transport equation (Eq. (5.4)) is typically used alongside two-equation turbulence models:

$$-\rho_g \overline{u_i' u_j'} = \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\left(\rho_g k + \mu_t \frac{\partial u_k}{\partial x_k}\right) \tag{5.12}$$

Consequently, by using the above approximation and realizable $k-\varepsilon$, the turbulent flux terms shown in the conservation equations above (Eqs. (5.4), (5.5), and (5.8)), are closed off. The turbulent energy flux from Eq. (5.5) is also modelled using a gradient assumption [85]:

$$\rho_g \overline{u_j' E'} = \frac{c_p \mu_t}{Pr_t} \frac{\partial T_g}{\partial x_j} \tag{5.13}$$

where the Prandtl number, Pr_t, is set to 0.85 by default [82]. The turbulent thermal conductivity is then defined as:

$$\lambda_t = \frac{c_p \mu_t}{Pr_t} \tag{5.14}$$

The effective thermal conductivity which is utilised in the energy balance equation, Eq. (5.5) is then determined as in Eq. (5.15).

$$\lambda_{eff} = \lambda_l + \lambda_t \tag{5.15}$$

5.3 Discrete phase modelling

Discrete phase modelling (DPM) is used to model particle dispersion as well as solid phase combustion. Individual fuel particles are tracked via a Lagrangian reference frame, while the continuous gas-phase is resolved using a Eulerian reference frame. A two-way coupling accounting for particle interaction with the gas (continuous) phase is implemented using Fluent® v23.1, with source terms for the particles incorporated into the gas phase conservation equations. Correctly capturing the motion pattern of the fuel particles is important, as it influences particle residence time in the furnace, and consequently combustion efficiency. In this work, bagasse fuel particles are modelled as small cylinders by modifying the shape factor for drag coefficient calculations.

A force balance using Newton's second law resolves the particle motion, given as:

$$\frac{d\vec{u}_p}{dt} = F_D(\vec{u}_g - \vec{u}_p) + \frac{\vec{g}(\rho_p - \rho_g)}{\rho_p} \tag{5.16}$$

Particle density, ρ_p, is set to 900 kg/m^3 for bagasse, as taken from [56]. The discrete random walk model in Fluent® v23.1 is used to account for the effect of gas phase turbulence fields on particle momentum. The instantaneous gas phase velocity vector, $\vec{u}_g$, is calculated as:

$$\vec{u}_g = \left\langle u = \overline{u} + u', v = \overline{v} + v', w = \overline{w} + w' \right\rangle \tag{5.17}$$

With kinetic energy of turbulence known, the fluctuating velocity component in the x-direction u', is calculated as $u' = \zeta \sqrt{\overline{u'^2}} = \sqrt{\frac{2k}{3}}$ where ζ is a normally distributed random number. This also applies for y and z components. For particle size distribution, a Rosin-Rammler [82] size distribution for typical Southern-African sugarcane bagasse was used [57], [86].

Table 5.4: Bagasse particle size distributions

Size distribution parameter	Value
Mean particle diameter (mm)	1.43
Maximum particle diameter (mm)	9.146
Minimum particle diameter (mm)	0.189
Spread parameter	0.8379

The particle drag force, F_D, in Eq. (5.16), is calculated using [87]:

$$F_D = \frac{18\mu}{\rho_p d_p^2}\frac{C_D Re_d}{24} \tag{5.18}$$

where d_p is the particle diameter, C_D the drag force coefficient, Re_d the relative Reynolds number based on the particle diameter, defined as:

$$\mathrm{Re}_d = \frac{\rho d_p \left|\vec{u}_p - \vec{u}_g\right|}{\mu} \tag{5.19}$$

As previously mentioned, biomass particles are typically larger and have a more non-uniform shape compared to fuels such as pulverised coal. Consequently, the bagasse fuel particles are modelled as non-spherical particles. The drag force coefficient, C_D, is modelled using the correlation of Haider and Levenspiel [88]:

$$C_D = \frac{24}{Re_{sphere}}(1 + b_1 Re_{sphere}^{b_2}) + \frac{b_3 Re_{sphere}}{b_4 + Re_{sphere}} \tag{5.20}$$
$$b_1 = \exp(2.3288 - 6.4581\phi_{sf} + 2.4486\phi_{sf}^{\,2})$$
$$b_2 = 0.0964 + 0.5565\phi_{sf}$$
$$b_3 = \exp(4.905 - 13.8944\phi_{sf} + 18.422\phi_{sf}^{\,2} - 10.2599\phi_{sf}^{\,3})$$
$$b_4 = \exp(1.4681 + 12.2584\phi_{sf} - 20.7322\phi_{sf}^{\,2} + 15.8855\phi_{sf}^{\,3}$$

where ϕ_{sf} is the shape factor, defined as $\phi_{sf} = s / S$, with s the surface area of the bagasse particle, and S the surface area of an equivalent volume sphere [57].

Particles interact with the surrounding gas phase through convective and radiative heat transfer as they move through the boiler domain. The energy balance equation for each particle parcel is given as:

$$m_p c_{p_p}\frac{dT_p}{dt} = hA_p(T_g - T_p) + \varepsilon_p A_p \sigma(\theta_R^4 - T_p^4) - \frac{dm_c}{dt}H_{CO} - \frac{dm_{evap}}{dt}h_{fg,water} \tag{5.21}$$

Where $\theta_R = \left(G/4\sigma\right)^{0.25}$, and G the incident radiation, h the convective heat transfer coefficient between the discrete and gas phases, and H_{CO} the heat of reaction for CO. The specific heat for the biomass particle was set to $c_{p_p} = 1200$ J/kg [89], and the emissivity of the particles set to $\varepsilon_p = 0.9$ [56]. Laubscher and van der Merwe [57] emphasised the importance of using the correct drag coefficient formulation, with bagasse particles experiencing significantly higher drag forces compared to spherical particles. Bagasse particles spread deeper into the depth of the furnace, with a higher percentage of particles in suspension.

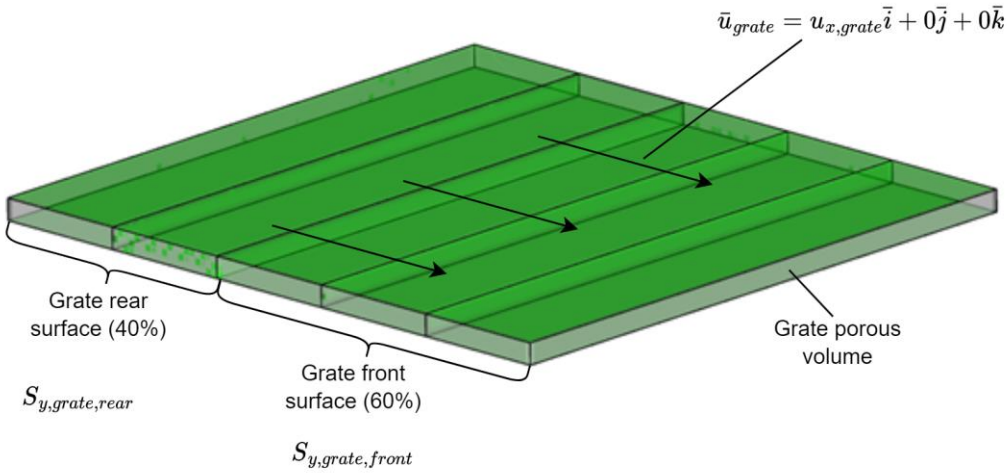


Figure 5.5: Grate modelling, including grate rear and front zones

To model the interaction between the grate and particles, a custom C program is used [57], which considers the grate motion and effects of drag and gravitational forces on particles lying on the grate surface. Figure 5.5 shows the grate zones, split into rear and front zones. Five volumes are used to model the grate. In reality, zoning dampers are used to force $\approx 60\%$ of the inlet PA through the rear grate zone, which in this case constitutes $\approx 40\%$ of the grate area. In the model, the rear grate and front grate zones are shown in Figure 5.5, with two of the five grate volumes making up the rear grate zone (40%). The PA pressure drop through the grate is accounted for by modelling the grate volume as a porous media, which allows for the zoning damper settings to reflect in the model. A discrete phase boundary condition at the grate top plane is used to incorporate the grate model.

For the biomass boiler investigated by Laubscher and van der Merwe [57], a PA pressure drop of 100-200 Pa is estimated based on practical experience with all the dampers completely open. For this work, a PA pressure drop of 150 Pa is initially assumed. The grate porous volume was modelled with a thickness of 150 mm. The y-direction source term is then calculated as:

$$S_{y,grate} = \frac{dp_{grate}}{dt_{grate}} = -C_0 \left| v \right|^{C_1} \tag{5.22}$$

where t_{grate} is the grate thickness. To account for the rear and front grate zones, the power law formulation was used, shown in Eq. (5.22) as the term on the far right. For simplification, $C_1 = 1$, which results in a negative source term for a positive velocity. To determine C_0, the velocity is calculated using $\dot{m}_{PA}$, the grate area, and the density determined at ambient pressure and $T_{PA,inlet}$. With the initially assumed pressure drop and the grate thickness, the constant C_0 is calculated for both the rear grate zone and front grate zone as $C_{0,rear} \approx 1500 \ \frac{\text{Pa}\cdot\text{s}}{\text{m}^2}$ and $C_{0,front} \approx 3425 \frac{\text{Pa}\cdot\text{s}}{\text{m}^2}$, which is used to enforce the correct damper configuration.

Particles which end up on the grate surface are assumed to have a velocity vector equal to that of the grate surface, however, only if the gravitational force acting on the particle is greater than the induced y-directional drag force. If the drag force is greater than the gravitational force, the particle is allowed to move in the freeboard. The gravitational and drag forces are calculated as [57]:

$$\begin{aligned} F_{y,drag} &= F_D (u_y - u_{p,y}) m_p \\ F_{y,grav} &= (\rho_p - \rho) \bar{g} \frac{m_p}{\rho_p} \end{aligned} \tag{5.23}$$

5.4 Combustion modelling

Combustion processes are driven by three main criteria, temperature, turbulence, and time. High enough temperatures are required to activate chemical reactions, turbulence to ensure sufficient mixing of reactants and oxygen, and time to ensure complete combustion takes place. Solid fuel combustion includes modelling of discrete phase heterogeneous and gas-phase homogeneous reaction rates. Heterogeneous reactions produce the oxidising reactants, which then allow for gas-phase homogeneous reactions to take place.

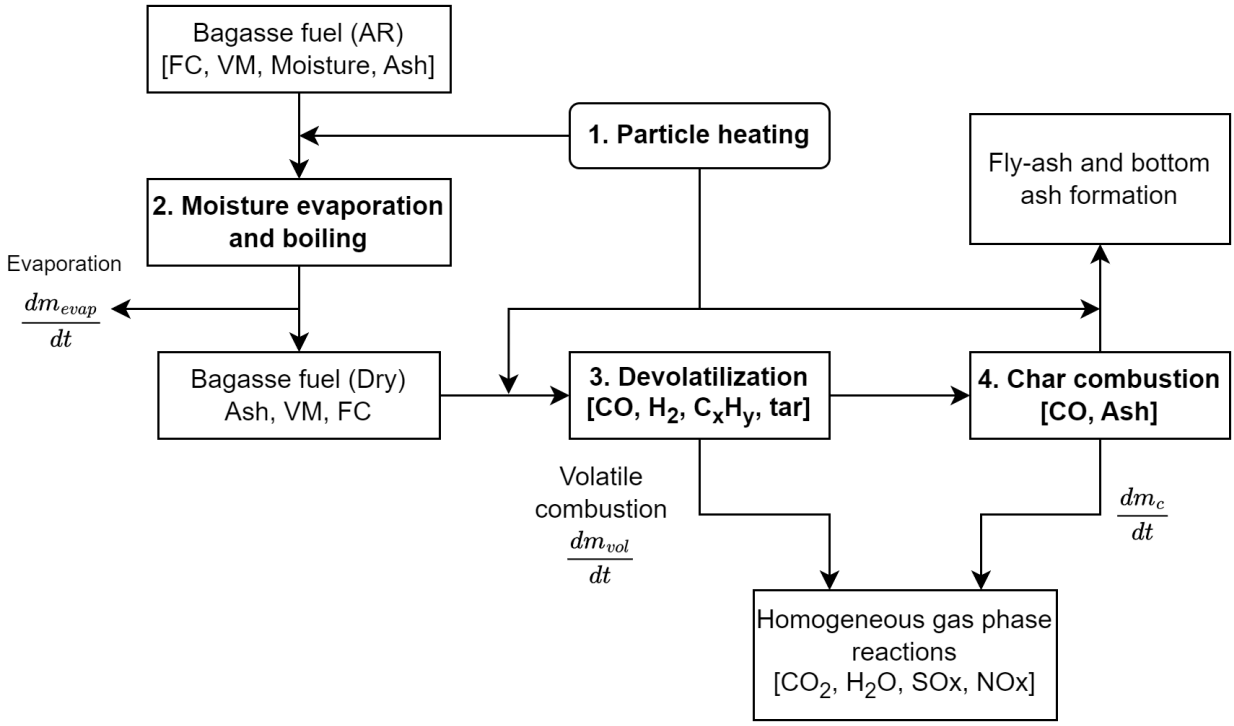


Figure 5.6: Combustion sequential process

Figure 5.6 shows the combustion processes, consisting of four sequential steps, including heating, moisture evaporation/boiling, devolatilization, and char oxidation. The sequential approach for combustion modelling has been shown to be a reasonable numerical simplification for biomass fuels [57], [84].

Fuel particles enter the furnace domain at ambient temperature (27 °C). In the furnace domain, particles are then heated via convection and radiation from the surrounding hot gases, resulting in particle moisture loss through evaporation. This continues up to the water boiling temperature of $T_{boil,water} = 100$ °C. Correlations by Miller [90] and Sazhin [91] are used in Fluent® [82] to calculate the rate of evaporation. At $T_{boil,water}$ the moisture in the particles boils off, which greatly increases

the rate of moisture loss from the particle. The rate of mass loss while boiling is calculated by dividing the net heat transfer rate to the particle by the latent heat of vaporisation, $h_{fg,water} = 2263$ kJ/kg [82]. When all moisture is boiled off from the fuel particles, the particle is then heated up to the devolatilisation temperature $T_{devol,p} = 127$ °C [57], [92]. To capture the rate of volatile release, a simplified single kinetic rate formulation is used. Single-rate Arrhenius based models are computationally inexpensive and are adequate to capture system-level heat transfer characteristics [93]. Several other advanced models exist, which generally use complex turbulence-chemistry interaction models at higher computational expense. These models are typically used for prediction of pollutants [57]. The focus of this study lies with determining performance of various boiler heat exchangers, therefore the single rate Arrhenius based model is sufficient. The single kinetic rate formulation calculates the rate of mass loss using [94]:

$$-\frac{dm_{vol}}{dt} = k_{vol}\left[m_p - (1 - f_{v,0})m_{p,0}\right] \tag{5.24}$$

where $f_{v,0}$ is the initial mass fraction of volatiles, and $m_{p,0}$ the initial particle mass. The devolatilisation process kinetic rate is calculated as:

$$k_{vol} = A_{vol}\exp\left(\frac{-E_{a,vol}}{RT_p}\right) \tag{5.25}$$

The empirical rate constants of Gera et al. [89] are used, with $A_{vol} = 3.12\text{x}10^6$ 1/s and the activation energy $E_{a,vol} = 7.4\text{x}10^7$ J/kmol. These constants are based on burnout data from Sandia National Laboratories Multifuel Combustor for switch grass, and were successfully implemented for modelling of Southern African sugarcane bagasse by du Toit [92], [95]. To account for shrinkage of the bagasse particles during the devolatilisation process, a swelling coefficient of 0.56 is used [89].

The devolatilisation process is followed by char combustion, consuming the remaining fixed carbon. The rate of particle mass loss during char combustion is modelled using a diffusion-kinetics limited approach of Baum and Street [96], successfully implemented by Laubscher and van der Merwe [57]. The model considers chemical reaction kinetics alongside the diffusion of oxygen through the porous structure of bagasse particles. The char combustion rate is calculated as [82]:

$$\frac{dm_c}{dt} = -A_p\frac{\rho RT_g Y_{O2}}{M_{w,O2}}\frac{D_0 R_c}{D_0 + R_c} \tag{5.26}$$

where A_p is the surface area of the particle, $\frac{\rho RT_g Y_{O_2}}{M_{w,O_2}}$ the partial pressure of O_2, D_0 the diffusion rate coefficient, and R_c the kinetic rate coefficient. Note that Eq. (5.26) is used in the particle energy balance equation, Eq. (5.21). D_0 and R_c are calculated as:

$$D_0 = C_1 \frac{\left[0.5\left(T_p + T_g\right)\right]^{0.75}}{d_p}$$
$$R_c = C_2 \exp\left(\frac{-E_{a,c}}{RT_p}\right) \tag{5.27}$$

Empirical coefficients are taken from [56], and are given as $C_1 = 1\text{x}10^{-11}$ kg/(m^{2}sPa), $C_2 = 6.7$ kg/(m^{2}sPa) and $E_{a,c} = 1.138\text{x}10^8$ J/kmol. The surface reaction of the char combustion produces CO, and is given as:

$$C_{(s)} + 0.5O_{2(g)} \rightarrow CO_{(g)} \tag{5.28}$$

For the product species CO, it is assumed the particles absorb 100% of the char combustion heat [97]. The heat of reaction for char burnout is set to $1.01\text{x}10^7$ J/kg [98].

The gaseous species released from the devolatilisation and char burnout processes above react with supplied oxygen from the provided combustion air to form the respective flue gas product species. The modelled gas phase reactions are given as:

$$CH_2O_{0.97}N_{0.0085}S_{0.0004} + 0.51O_2 \rightarrow CO + H_2O + 0.0042N_2 + 0.0004SO_2$$
$$CO + 0.5O_2 \rightarrow CO_2 \tag{5.29}$$

where $CH_2O_{0.97}N_{0.0085}S_{0.0004}$ is the pseudo-volatile species released into the domain. The composition of the pseudo-volatile species is determined through an elemental mass balance using the dry ash-free ultimate analysis and proximate analysis provided in Table 5.1. A stoichiometric balance of reactants and products results in the first reaction in Eq. (5.29).

For gas phase reactions, the rate of reaction for volatiles and CO is influenced by gas temperatures, reactant concentrations, and turbulence. Turbulence-chemistry interaction models are used to capture the gas phase reactions. There are two main approaches, including fast reaction models and detailed chemical kinetic models such as the Eddy-dissipation concept (EDC) model [99]. The EDC model solves chemical kinetics differential equations for each cell, making it very computationally expensive. The EDC model is typically recommended when the fast chemistry assumption is invalid. For modelling gas-phase reactions in boilers, the time scale of chemical reactions is significantly smaller than the that of turbulent mixing [57], [100]. Consequently, the fast reaction chemistry is assumed in this work to reduce computation time. The fast reaction-based eddy dissipation model by Magnussen and Hjertager [101], provided in Fluent® v23.1 [82], is applied in this study. This model has been used successfully for CFD modelling of bagasse boilers [56], [57]. The reaction rate for each chemical reaction can be calculated using:

$$R_{k,r} = \min \begin{pmatrix} \vartheta_{k,r} M_{W,k} A \rho \frac{\varepsilon}{k} \min\left(\frac{Y_{reactant}}{\vartheta_{j,r} M_{w,j}} \right) \\ \vartheta_{k,r} M_{W,k} A B \rho \frac{\varepsilon}{k} \frac{\sum_{products} Y_{products}}{\sum_{j} \vartheta_{j,r} M_{W,j}} \end{pmatrix} \tag{5.30}$$

where $\vartheta_{k,r}$ is the stoichiometric coefficient of reactant k for reaction r, Y the mass fraction of reactant or product species, and $M_{w,j}$ the molecular weight of the j^{th} species. The model constants, A and B are set to 4 and 0.5 respectively in Fluent® [82].

5.5 Radiative heat transfer modelling

Radiation heat transfer is the dominant heat transfer mechanism in the furnace, making up the majority of heat absorbed by furnace water walls and thermal energy transport through the gas and particulate volume. Modelling gaseous and particle heat transfer interactions due to radiative effects is therefore crucial to obtain realistic results. The radiative transport equation (RTE) is used to solve for radiation transport between the gas, particles, and various heat exchanger surfaces in the boiler. For a gray-participating gas and particle medium [102]:

$$\underbrace{\frac{dI(r,\hat{s})}{ds}}_{\text{Radiation intensity}} = \overbrace{\alpha_g I_b}^{\text{Gas emission}} + \underbrace{\alpha_p I_{b,p}}_{\text{Particle emission}} - \overbrace{(\alpha_g + \alpha_p + \sigma_p) I(r,\hat{s})}^{\text{Absorption and scattering losses}} + \underbrace{\frac{\sigma_p}{4\pi} \int_{4\pi} I(r,\hat{s}) \Phi d\omega}_{\text{Internal scattering}} \tag{5.31}$$

where $dI(r,\omega)$ is the radiation intensity at point r in direction $\hat{s}$, I_b is the intensity of radiation emitted by a black body, α_g and α_p are the overall gas and particle absorption coefficients respectively, and Φ the phase function, which tends to 0 when assuming isotropic scattering [83].

Numerical modelling of radiative heat transfer through Eq. (5.31) include several methods, such as statistical, zonal, or flux methods. Flux methods assume uniform radiation intensity for a finite number of discrete solid angles. In this work, the Discrete Ordinates (DO) method is used for radiative heat transfer modelling. By discretizing the solid angle into discrete number of directions, the integral on the RHS of Eq. (5.31) can then be replaced by a summation of terms over the discrete number of directions, M :

$$\int_{4\pi} I(r,\hat{s})\Phi d\omega \approx \sum_{j=1}^{M} I_j(r,\hat{s}_j)\Phi_{m,j}\omega_j \qquad m=1,2,\ldots M \tag{5.32}$$

where ω_j are the quadrature weights associated with directions $\hat{s}_j$. The DO method therefore solves the radiative transport equation, including particulate effects, for the specified finite number of discrete solid angles, M:

$$\frac{dI(r,\hat{s}_m)}{ds} = \alpha_g I_b + \alpha_p I_{b,p} - (\alpha_g + \alpha_p + \sigma_p) I(r,\hat{s}_m) + E_p + \frac{\sigma_p}{4\pi}\sum_{j=1}^{M} I_j(r,\hat{s}_j)\Phi_{m,j}\omega_j \tag{5.33}$$

where E_p is the equivalent emissive power of particles. Scattering due to the gas phase is assumed negligible compared to particle scattering [68]. In Fluent® v23.1, for the implementation of the DO model, each octant of the angular space 4π at any spatial location is discretized into $N_\theta N_\varphi$ solid angles, with extent ω_j. θ is the polar angle, and φ the azimuthal angle, both measured with respect to the global Cartesian system. Therefore, a total of $8 N_\theta N_\varphi$ RTE equations are solved. In the present work, N_θ and N_φ are both set to four divisions [57] to sufficiently capture large spatial variations in temperature and small geometric features.

With the gray gas assumption, the gaseous mixture absorption coefficient, α_g from Eq. (5.31), is calculated through the domain-based weighted-sum-of-gray-gases model (WSGGM). The WSGGM has been previously described in section 3.2.1.2, Eqs. (3.25) and (3.26), where α_g is estimated using Eq. (3.26).

Particle radiative effects are defined by E_p, the equivalent emissive power, α_p, absorption coefficient, and σ_p, scattering coefficient. These parameters are defined as [82], [102]:

$$\begin{aligned} E_p &= \lim_{V\to 0}\sum_{n=1}^{N}\frac{\varepsilon_{p,n}A_{proj,n}\sigma_{SB}T_p^4}{\pi V} \\ \alpha_p &= \lim_{V\to 0}\sum_{n=1}^{N}\frac{\varepsilon_{p,n}A_{proj,n}}{V} \\ \sigma_p &= \lim_{V\to 0}\sum_{n=1}^{N}(1-\varepsilon_{p,n})(1-f_{p,n})\frac{A_{proj,n}}{V} \end{aligned} \tag{5.34}$$

where parameters are defined for particle n with N particles in volume V. $\varepsilon_{p,n}$ is the particle emissivity, $A_{proj,n}$ the particle projected area, and $f_{p,n}$ the particle scattering factor. $\varepsilon_{p,n}$ and $f_{p,n}$ are set to constant values of 0.9 and 0.6 respectively [57], [103].

5.6 General heat transfer

To model heat transfer to boiler water walls and heat exchanger tubes, an effective heat transfer coefficient approach is used. The internal fluid temperature, $T_{fluid,int}$, and internal heat transfer coefficient, h_{int}, are specified. Moreover, the tube wall thickness, $t_{tube,wall}$, and ash layer thickness, $t_{ash,fouling}$ are specified. An effective thermal conductivity is calculated, k_{eff}, which accounts for the thickness and thermal conductivities of both the tube wall and ash fouling layer, calculated as:

$$k_{eff} = \frac{t_{tube,wall}}{\frac{t_{tube,wall}}{k_{tube}} + \frac{t_{ash,fouling}}{k_{ash,fouling}}} \tag{5.35}$$

where $k_{tube,steel} = 47$ W/mK, $\frac{t_{ash,fouling}}{k_{ash,fouling}} = 0.0043$ m^{2}K/W. For refractory covered furnace water walls:

$$k_{eff,refr} = \frac{t_{tube,wall}}{\frac{t_{tube,wall}}{k_{tube}} + \frac{t_{refr}}{k_{refr}}} \tag{5.36}$$

where a refractory thickness of 150 mm is used and $k_{refr} = 1.4$ W/mK. k_{eff} is then used with $t_{tube,wall}$ to calculate the effective heat transfer coefficient. To implement this in Fluent®, k_{eff} is the specified thermal conductivity for the combined tube metal and ash fouling material/refractory material. The effective heat transfer coefficient is then calculated as:

$$h_{eff} = \frac{1}{\frac{1}{h_{int}A_{hi}} + \frac{t_{tube,wall}}{k_{eff}A_{hi}}} \tag{5.37}$$

For water wall/tube wall boundary conditions, the convective thermal boundary condition in Fluent® v23.1 is used. The model balances the heat flux to the walls with the heat transfer rate from the boundary walls to the internal fluid, shown in Eq. (5.38)

$$h_g(T_g - T_{wall}) + q_{rad} = h_{int}(T_{wall} - T_{int}) \tag{5.38}$$

where subscript g refers to flue gas, and $wall$ the external tube surface. q_{rad} is the radiative heat flux absorbed by the walls. The emissivity of all wall surfaces is set to 0.8 [67]. For SH2 and the sCO2 heaters, h_{int} and the average T_{int} are extracted from the 1D process model, shown in Table 5.3. For boiler water walls, $T_{int} = 285$ °C which is the saturation temperature of water at the drum pressure, and $h_{int} = 20000$ W/m^{2}K for internal flow boiling in the water wall tubes to account for the fact that the internal thermal resistance is negligible compared to the external resistance.

5.7 Porous media modelling

Both SCR1 and SCR2 are modelled as porous media. The physical velocity formulation is used in Fluent®, with porosity set to 1. By modelling SCR1 and SCR2 as porous media, the relevant pressure drops and heat transfer from the flue gas can be captured at minimal computational expense. The momentum and energy sources/sinks are included as source terms for the gas phase momentum and energy balance equations (Eq. (5.4) and Eq. (5.5)) as described above. The momentum source term is calculated as:

$$S_{pm,k,j} = -\left(\mu \frac{1}{\alpha} v_j + C_2 \frac{1}{2} \rho |v| v_j \right) \tag{5.39}$$

where $S_{pm,k,j}$ is the momentum source term for a porous medium k in the j^{th} direction, $\frac{1}{\alpha}$ is the inverse absolute permeability, and C_2 the inertial resistance factor [82]. For SCR1 and SCR2, inertial resistance is assumed negligible compared to the viscous resistance, and $C_2 = 0$. Therefore, only the viscous loss term, or Darcy term, is accounted for. For both SCR1 and SCR2, the inverse absolute permeability is set to $\left(\frac{1}{\alpha}\right)_x = 5\text{x}10^5\ \text{m}^{-2}$, $\left(\frac{1}{\alpha}\right)_y = 1\text{x}10^5\ \text{m}^{-2}$, and $\left(\frac{1}{\alpha}\right)_z = 5\text{x}10^5\ \text{m}^{-2}$. For the energy source term, $S_{h,pm,k}$, the provided inputs from the 1D process model in Table 5.3 are specified for both SCR1 and SCR2.

5.8 Numerical setup

The model is developed in ANSYS Fluent® v23.1. The pressure-based solver is used, alongside the SIMPLE pressure-velocity coupling. The solver process diagram is shown in Figure 5.7 [82]. The momentum, energy, and species equations were discretized using the second-order upwind method. The pressure equation was discretized using PRESTO!. The remainder of the field variables, including turbulent kinetic energy and dissipation, and DO, were discretised using the first order upwind scheme. DO radiation source terms were updated every 10 iterations. Discrete phase source terms were updated every 100 iterations.

All simulations were run until convergence, as well as when various monitors, including SH2 and sCO2 heater temperatures, and species mass fractions at the outlet plane, stabilised. The maximum mass imbalance recorded for all simulation cases was 0.0003 kg/s for a total flue gas flow rate of 26.63 kg/s , and heat imbalance of 0.0090 MW for a total heat transfer rate of 43.63 MW at 100% load. Simulation residuals were below 5E-3 for continuity and 1E-6 for energy. All simulations were run for approximately 10,000 iterations with an approximate wall-time between 50-60 hours

depending on the model. The simulations were run using 144 Intel(R) Xeon(R) E5-2690 2.90 GHz cores, made available by the Centre of High-Performance Computing in South Africa.

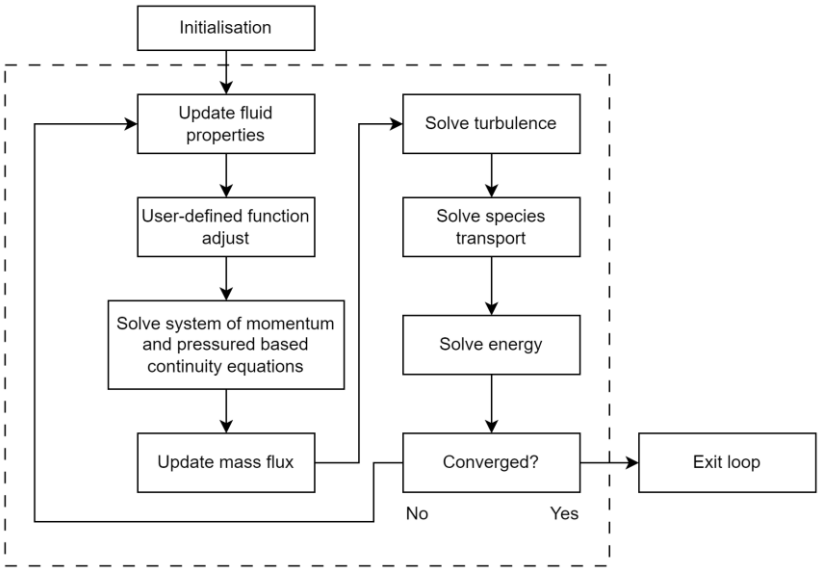


Figure 5.7: Pressure-based coupled solver process diagram

6. Biomass boiler CFD modelling: Results and discussion

This section compares the results of the CFD modelling, broken down into nominal (100%) load comparisons and low (80% and 60%) load comparisons. For each load percentage, the existing cycle configuration without the sCO2-integration, single sCO2 heater configuration, and dual sCO2 heater configuration are compared. Lastly, the 1D process model is compared with the CFD results. The CFD results are then used to inform the 1D process model, and the updated 1D process model results presented.

The main aim of the CFD study was to validate the heat uptakes at sCO2 heaters predicted by the 1D process model. 3D CFD modelling is also able to capture spatial characteristics of heat and mass transfer which 1D models lack. Therefore, the CFD model is also used to study some of the practical implications and to determine feasibility of the sCO2-integrated configurations such as heat exchanger metal temperatures and species concentration variations due to overfiring. This applies not only to heat and mass flux distribution over the heat exchangers, but also the combustion processes.

For the grid independency study, to compare mesh refinement, the mass-weighted temperature at the outlet of SH2, and sCO2HX/H2 is compared, along with the total heat uptakes at those heat exchangers in Table 6.1.

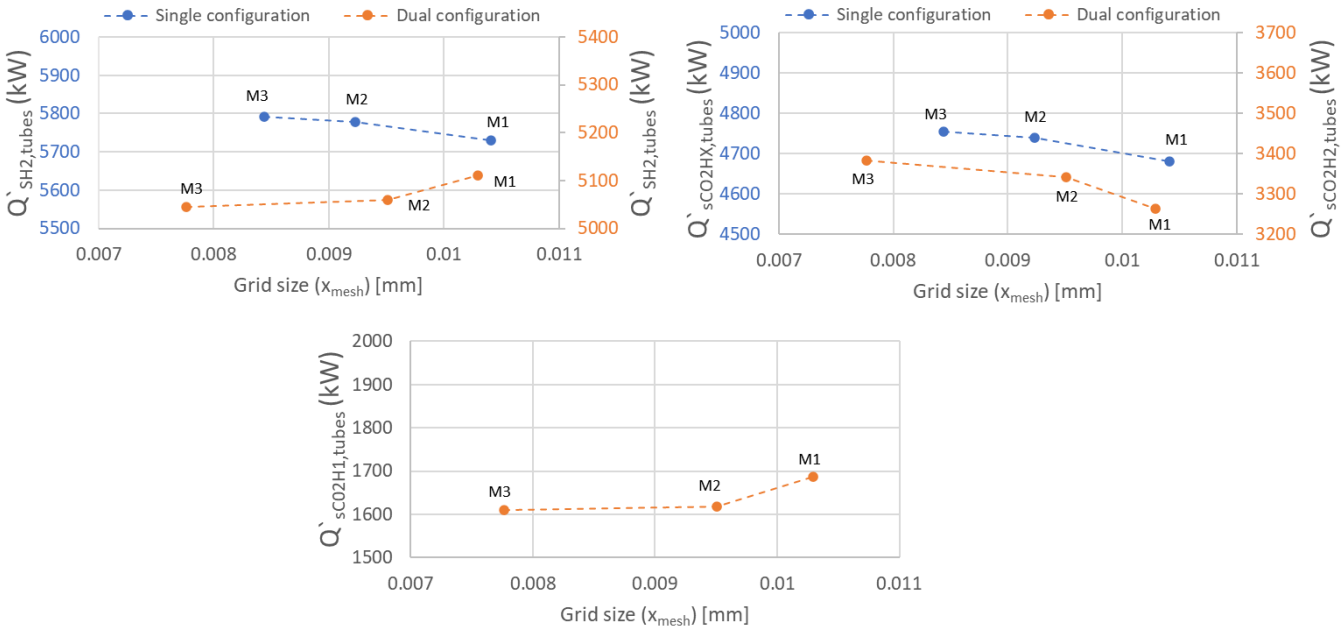


Figure 6.1: Heat uptakes at SH2 and sCO2 heat exchangers for meshes M1, M2, and M3

Table 6.1: Mesh refinement error for single and dual heater configurations

	Single configuration model error (%)		Dual configuration model error (%)	
	M1 to M2	M2 to M3	M1 to M2	M2 to M3
$T_{SH2,out}$	0.802	0.259	-0.756	-0.109
$T_{sCO2H1,out}$	-	-	-2.250	-0.689
$T_{sCO2HX/2,out}$	0.302	0.025	1.870	1.003
$\dot{Q}_{SH2}$	0.841	0.239	-1.023	-0.291
$\dot{Q}_{sCO2H1}$	-	-	-4.120	-0.544
$\dot{Q}_{sCO2HX/H2}$	1.217	0.346	2.410	1.232

From Table 6.1, gas temperatures had relatively low errors between the three meshes, with slight reductions observed between M2 to M3 compared to M1 to M2. The largest error decreases were observed for heat uptakes at sCO2 heaters from M2 to M3, particularly for sCO2H1.

Figure 6.1 shows the heat uptakes at SH2 and sCO2 heat exchanger tube banks for both the single and dual configurations plotted against the representative grid size for each mesh. For each plot, as x_{mesh} becomes smaller, the heat uptakes tend to a fixed value. M3 was selected as the final mesh for both single and dual configurations. While only slight errors were observed between M2 and M3, the finer mesh, M3, was still selected to minimise errors, particularly within the sCO2 heat exchangers.

6.1 Nominal load

For biomass boiler analysis the flue gas temperature distribution in the furnace and convective passes are important considerations. High gas temperatures can potentially lead to thermal damage of heat exchanger tube banks or excessive slagging if the gas temperature is higher than the ash deformation temperature. For bagasse, this temperature is between 1210°C-1350°C [57], [104]. This becomes even more important when considering elevated gas temperatures from overfiring, which is required for the sCO2-integrated configurations. Figure 6.2 shows the gas temperature distribution along the centreline of the boiler for all three configurations. Table 6.2 shows gas temperatures at the furnace exit, as well as furnace metal temperatures.

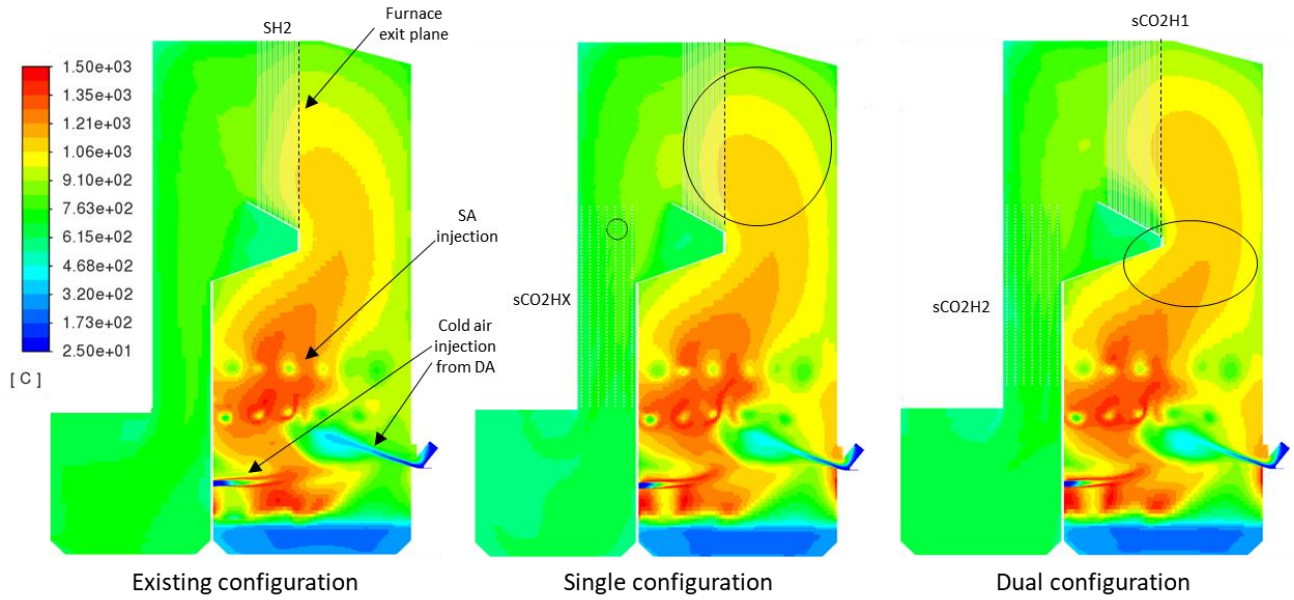


Figure 6.2: Gas temperatures at nominal load for the three configurations (left: existing configuration, middle: single sCO2 heater configuration, right: dual heater configuration)

Table 6.2: Furnace gas temperatures and metal temperatures at nominal load

	Existing configuration	Single configuration	Dual configuration
Furnace maximum exit temperature (°C)	1136	1156	1146
Furnace mass-weighted exit temperature (°C)	958	989	978
Furnace maximum metal temperature (°C)	298	298	305

The maximum furnace exit temperatures for all configurations were below the ash deformation temperatures for bagasse. The mass-weighted furnace exit temperature was highest for the single configuration, followed by the dual configuration, and lastly the existing configuration at 958°C (-3.1% from single configuration). This can also be seen in Figure 6.3, which shows gas temperatures and velocity profiles at the furnace exit plane. The higher gas temperatures at the furnace exit are expected for sCO2-integrated configurations, due to overfiring and higher gas flow rates. For the existing configuration, $\dot{m}_{fg,outlet} = 23.19$ kg/s, while $\dot{m}_{fg,outlet} = 26.65$ kg/s for the single configuration, and $\dot{m}_{fg,outlet} = 26.74$ kg/s for the dual configuration. Observing the gas temperature distribution for the single configuration compared to the existing configuration, hotter gases extend further towards SH2 as well as the boiler roof as a result of overfiring, shown with the black circle in

Figure 6.2. This can also be observed for the dual configuration; however, the effect is even more pronounced due to the extension of the boiler nose required to house sCO2H1. This results in an increase in heat flux towards the boiler roof. This is also observed for gas mass fluxes and heat fluxes, from Figure 6.3. Hotter gas is pushed towards the roof of the boiler, as seen by the increase in temperatures at the upper half of the furnace exit plane. Similarly, gas velocity in that region is also higher relative to the single and existing configurations

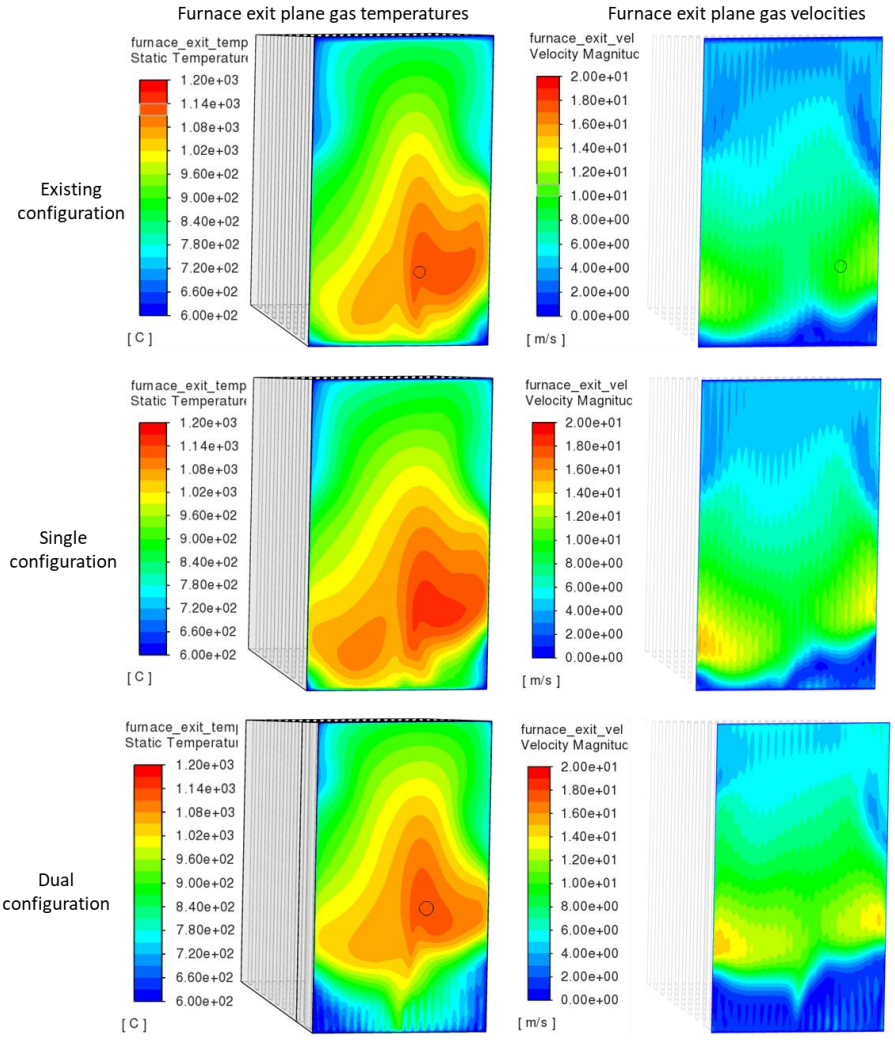


Figure 6.3: Furnace exit plane temperatures and velocity profiles at nominal load for all configurations

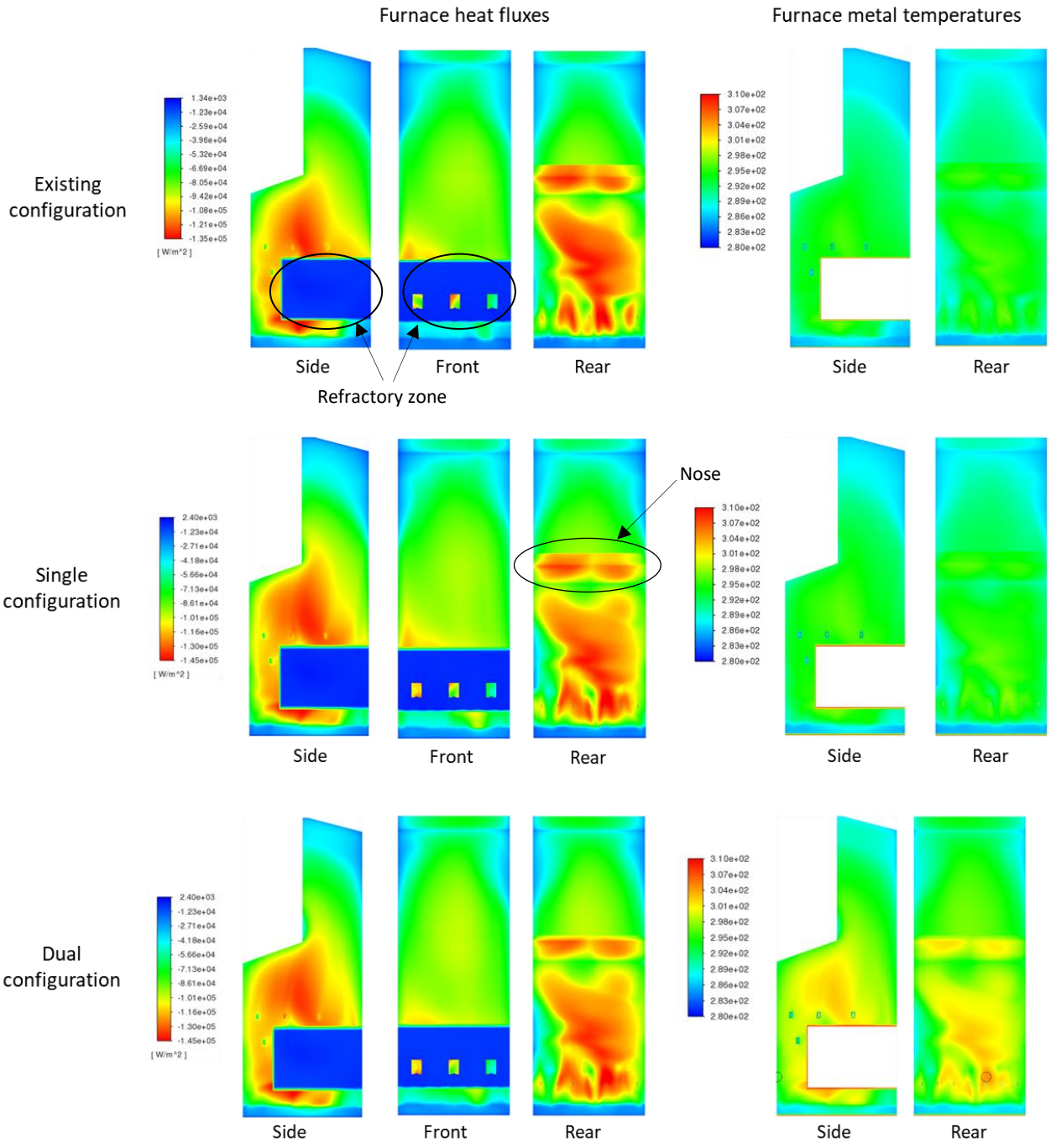


Figure 6.4: Furnace heat fluxes and metal temperatures at nominal load for all configurations

Figure 6.4 shows the heat flux distribution in the furnace as well as furnace metal temperatures for all configurations. The highest heat fluxes were observed at the side walls and rear walls just above the refractory zone, as well as towards the boiler nose. Notably, portions of the refractory walls radiate heat back into the furnace domain ($\approx$100 kW/m^2). This counteracts the colder direct air injection at the fuel spreaders and maintains a high temperature at that zone, where moisture

evaporation occurs. Comparing the single configuration to the existing configuration, higher heat fluxes were observed to the furnace water walls as a result of overfiring. Additionally, as previously mentioned, higher heat fluxes were also observed towards the boiler roof. For the dual configuration, the adjustment of the boiler nose closer towards the front wall forces the heat flux higher up towards the roof of the boiler. This can be seen in Figure 6.4 in the side and rear heat flux contours. Moreover, with combustion taking place lower in the furnace, and closer to the rear wall, higher rear wall metal temperatures are observed.

Table 6.3: Furnace exit plane and water wall heat transfer rates at nominal load for all configurations

	Existing configuration	Single configuration	Dual configuration
$\dot{Q}_{fe}$ (kW)	2137	2190	2815
$\dot{Q}_{ww,furn}$ (kW)	11612	12659	12600

Table 6.3 shows the heat transfer rates at the furnace exit plane and to the furnace water walls for all configurations. sCO2-integrated configurations had higher furnace water wall heat transfer rates +9.0%), driven primarily by overfiring and increases in flue gas flow rates from combustion. The direct radiation heat transfer rate through the furnace exit plane, $\dot{Q}_{fe}$, was calculated indirectly from the CFD results through a mass and energy balance over the furnace control volume, as shown in Appendix A.

The dual configuration had a higher $\dot{Q}_{fe}$ compared to the single configuration (+28.5%), which correlates with the higher flame position mentioned before. Inputs such as fuel flow rates and air flow rates were much the same for the single and dual configurations. Additionally, furnace water wall heat uptakes were also very similar. The difference between the two configurations lies with geometric changes, particularly the extension of the boiler nose to house sCO2H1. This extension shifts the furnace exit plane slightly further towards the front wall. The notable difference in $\dot{Q}_{fe}$ is therefore attributed to the spatial characteristics of radiative heat transfer throughout the furnace volume. The increases in gas mass flux towards the boiler roof result in an increase in direct radiation through the furnace exit plane.

Tube metal temperatures were calculated from the CFD model using Eq. (6.1), with $q_{CFD,calc}$ the total heat flux to the external wall.

$$T_{tube,metal} = T_{wall,ext} - q_{CFD,calc} R_{fo} \tag{6.1}$$

From Table 6.2 and Figure 6.4, the maximum furnace metal temperature was determined for the dual configuration at 305 °C and 298 °C for the single and existing configurations, approximately 20°C higher than the water saturation temperature in the tubes. From Figure 6.4, the metal temperatures for the dual configuration were slightly higher, particularly in the zone below the boiler nose, at +7 °C. While still within operational limits, prolonged exposure to elevated temperatures may reduce the lifetime of the furnace waterwalls [105].

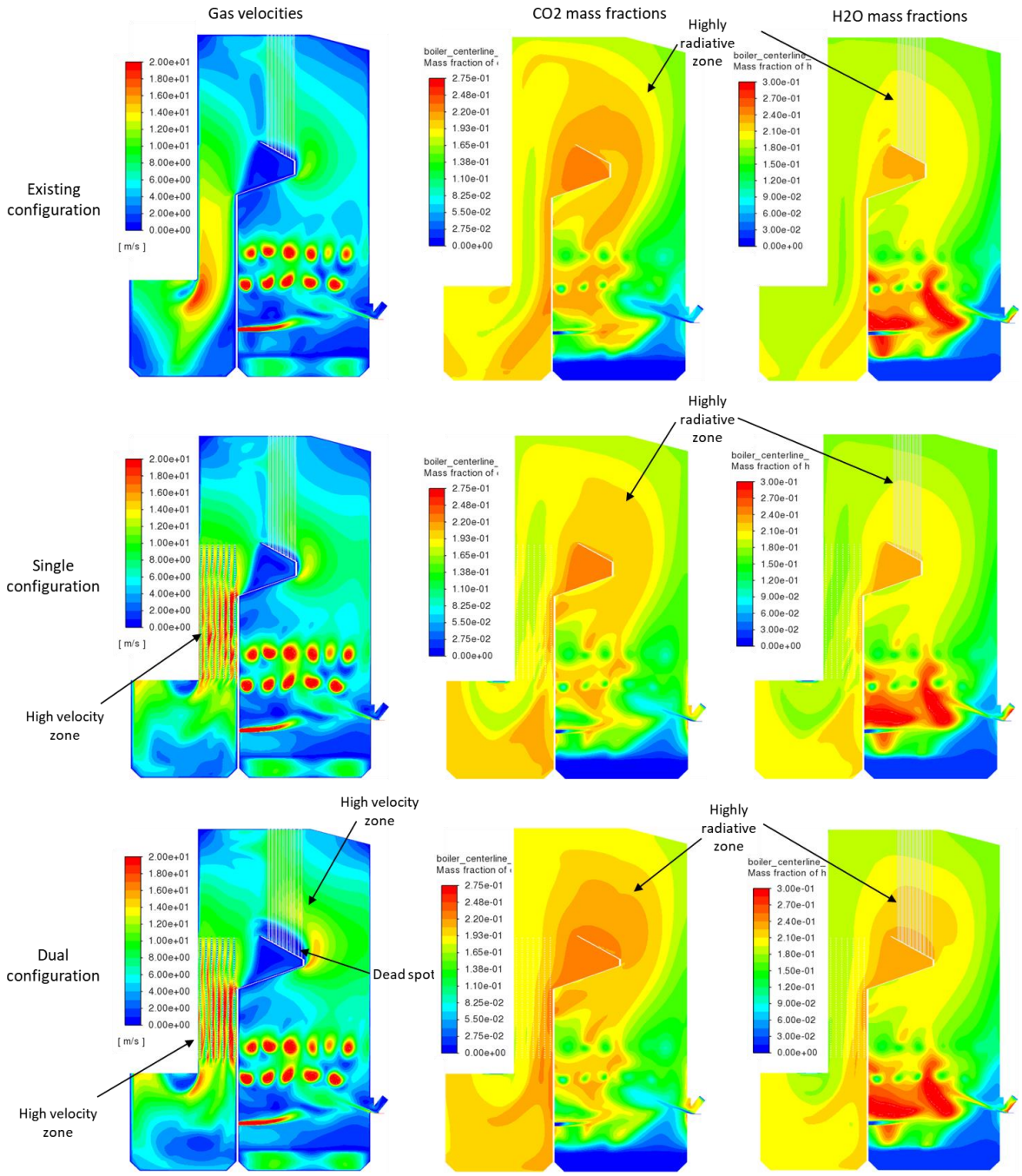


Figure 6.5: Gas velocities and water and carbon dioxide mass fractions at nominal load for all configurations

Figure 6.5 shows the gas velocities and CO2 and H2O mass fractions for all configurations. From the gas velocity contour, several high velocity zones were identified. For the single and dual configurations, high velocities were observed through the convective sCO2 heater. Notably, highly turbulent wakes were observed within the tube banks, which would assist with convective heat transfer. In order to more accurately characterize heat transfer within the highly turbulent regions, a more refined mesh would be required with a significantly lower Y^+ value to resolve the boundary layer near the tube walls. As the current study focuses on the overall boiler, such a mesh would incur significant computational expense. A study focusing on the convective sCO2 heater, using the velocity profiles from the current work, may prove valuable to gain further insights into heat transfer for the convective sCO2 heaters. For the dual configuration, a high velocity zone is also observed at sCO2H1, created by the extension of the boiler nose. Importantly, a dead spot (zones with low heat/mass flux) is also observed towards the bottom of sCO2H1 and SH2, which is not present for the existing and single configurations. The presence of a dead spot results in reduced heat uptakes for the given zone of the heat exchanger.

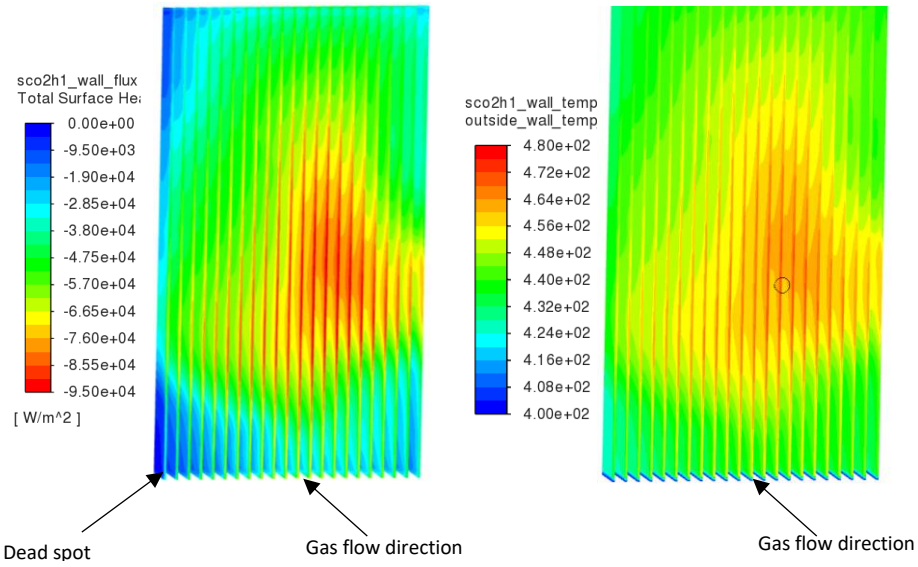


Figure 6.6: sCO2H1 heat fluxes and tube metal temperatures at nominal load

Table 6.4 shows the maximum and mass-weighted average gas velocities at SH2 and the sCO2 heaters. The peak gas velocities observed at SH2 were 14.7 m/s for the existing configuration, 19.5 m/s for the single configuration, and 18.0 m/s for the dual configuration. At the radiative platen heater for the dual configuration, sCO2H1, the peak gas velocity observed was 17.0 m/s. sCO2-integrated configurations had higher peak velocities due to overfiring. For the convective

sCO2 heaters, sCO2HX and sCO2H2 for the single and dual configurations, the peak velocities observed were 38.4 m/s and 36.5 m/s respectively. In contrast, for the existing configuration, the peak velocity observed in CAV1 was 20.1 m/s. These peak velocities are observed in localised limited regions of the heat exchangers. For sCO2HX/2, the peak velocities are observed at the intersection of the tube bank and boiler nose cavity. A better representation of the gas velocity in the tube bank is the mass-weighted average gas velocity, shown in Table 6.4. The mass-weighted average gas velocities are all within specifications to prevent erosion, as previously discussed in section 4.1.3.3. An erosion screen situated before the convective sCO2 heaters may however be beneficial to reduce erosion potential and high localised velocities.

Table 6.4: Maximum and mass-weighted gas velocities at SH2 and sCO2 heat exchangers for all configurations

	Existing configuration	Single configuration	Dual configuration
SH2 maximum gas velocity (m/s)	14.7	19.5	18.0
SH2 mass-weighted average gas velocity (m/s)	5.86	6.61	6.73
sCO2H1 maximum gas velocity (m/s)	-	-	17.0
sCO2H1 mass-weighted average gas velocity (m/s)	-	-	7.90
sCO2HX/2 maximum gas velocity (m/s)	-	38.4	36.5
sCO2HX/2 mass-weighted average gas velocity (m/s)	-	11.1	10.9

Concentrations of CO2 and H2O directly influence tri-atomic gas emissivity and can be used to determine highly radiative zones. Highly radiative zones were therefore identified at SH2 as well as the radiative platen sCO2H1. For the existing configuration, there is no tube bank situated in the first convective pass. As a result, the gas remains at a high temperature relative to the sCO2-integrated configurations. The high temperature gas radiates heat to the water walls surrounding the cavity. In contrast, for sCO2-integrated configurations, mass fractions of CO2 and H2O drop significantly in the convective pass relative to the existing configuration due to the presence of the convective sCO2 heater which would result in decreased CO2 and H2O partial pressures and thus reduce radiative heat transfer. For the dual configuration at sCO2H1, a high velocity region coincides with a highly radiative region, which identifies an area for localised peak heat fluxes and metal temperatures.

Figure 6.6 shows the heat fluxes and tube metal temperatures at sCO2H1 for the dual configuration. As previously identified, the maximum heat fluxes were located towards the centre of the heat exchanger, in line with the temperatures observed at the furnace exit plane in Figure 6.3. Additionally, a dead spot is identified at the lower left zone of the heat exchanger, as observed in the velocity contour in Figure 6.5. The highest tube metal temperatures were also observed at the confluence of the high velocity and radiative zones, which had the highest heat flux. The maximum observed tube metal temperature was 470 °C, within material specification.

Figure 6.7 shows the heat fluxes and tube metal temperatures at SH2 for all configurations. For the single and dual heater configurations, the localised maximum heat flux was located towards the lower half of the heat exchanger. In contrast, for the dual configuration the maximum localised heat flux was towards the centre of the heat exchanger, caused by the adjustment to nose geometry. Notably, dead spots with low heat fluxes were identified for all configurations, as shown in Figure 6.7. The existing and single configurations exhibited similar dead spots towards the upper left half of the heat exchanger. For the dual configuration, an additional dead spot towards the lower left half of the heat exchanger was identified, again due to geometry adjustments. With the spatial characterisation of heat fluxes through CFD modelling, heat exchanger utilisation factors may need to be used in the 1D process model to account for the dead spots.

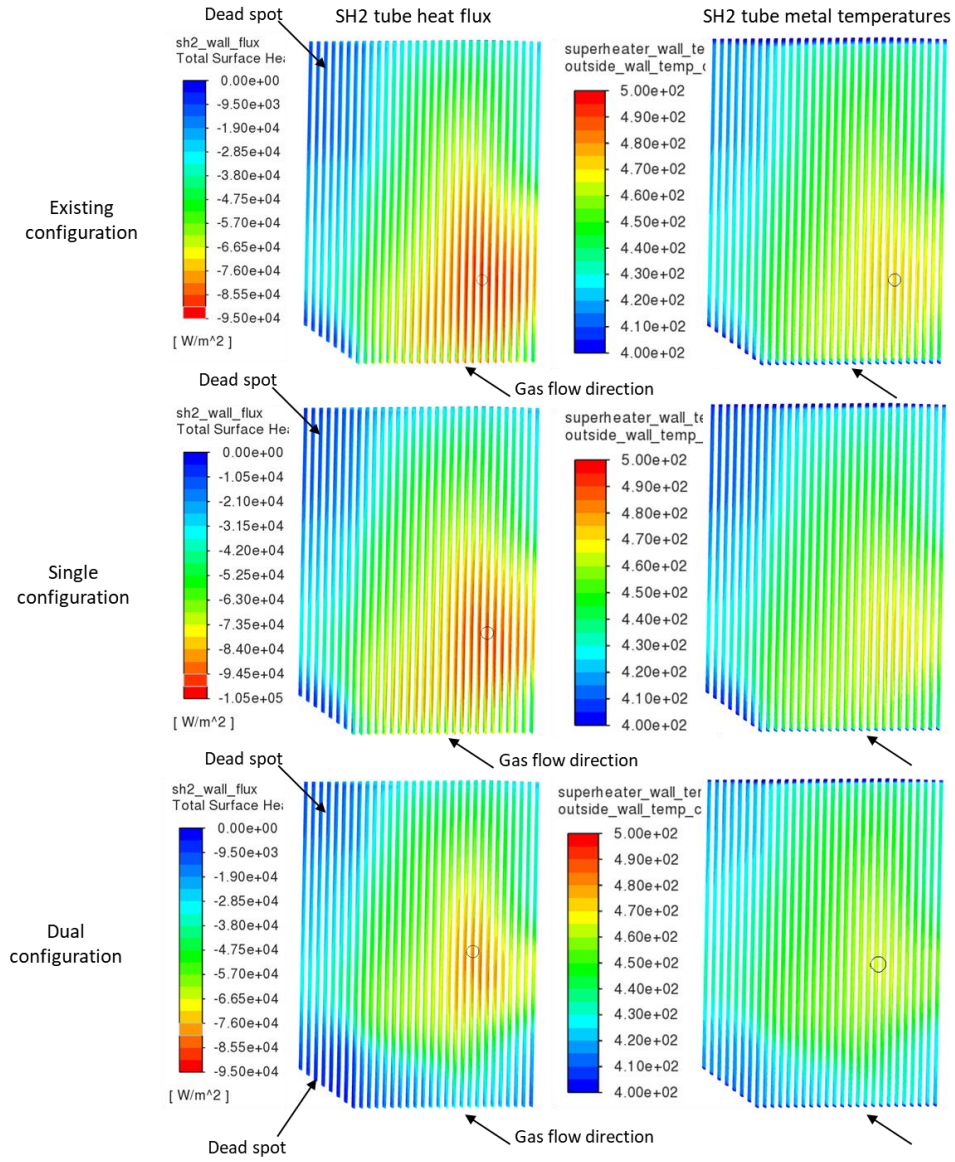


Figure 6.7: SH2 heat fluxes and tube metal temperatures at nominal load

Table 6.5: Heat uptakes at SH2 and sCO2 heat exchangers and water walls surrounding the heat exchangers

	Existing configuration	Single configuration	Dual configuration
$\dot{Q}_{SH2,tubes}$ (kW)	5352	5791	5044
$\dot{Q}_{sCO2H1,tubes}$ (kW)	-	-	1609
$\dot{Q}_{sCO2HX/2,tubes}$ (kW)	-	4755	3383
$\dot{Q}_{ww,SH2}$ (kW)	690	746	598
$\dot{Q}_{ww,sCO2H1}$ (kW)	-	-	239
$\dot{Q}_{ww,sCO2HX/2}$ (kW)	-	1071	937

Table 6.5 shows the heat uptakes at SH2 and the sCO2 heaters, as well as to the water walls surrounding those heat exchangers. Comparing heat uptakes at SH2, the single configuration had the highest heat uptakes, due to overfiring and increased gas flow rates. The dual configuration had the lowest SH2 heat uptake, caused by the heat absorption at sCO2H1 situated in front of SH2. This can be seen in the heat flux contour in Figure 6.7, with reduced heat fluxes in comparison to the other configurations. This also results in reduced heat uptakes at water walls surrounding SH2. Mass-weighted average gas temperatures at the inlet and outlet planes for SH2 and the sCO2 heat exchangers are presented in Table 6.6. The dual configuration also has a lower $T_{SH2,in}$, due to absorbed heat flux by sCO2H1. The maximum tube metal temperatures observed at SH2 is provided in Table 6.7. The dual configuration had a slightly lower SH2 maximum tube metal temperature, influenced by the reduced heat uptakes in comparison to the other configurations. SH2 tube metals were within operational specifications.

Table 6.6: Mass-weighted average gas temperatures for SH2 and sCO2 heat exchangers at nominal load

	Existing configuration	Single configuration	Dual configuration
$T_{sCO2H1,in}$ (°C)	-	-	978
$T_{sCO2H1,out}$ (°C)	-	-	945
$T_{SH2,in}$ (°C)	958	989	945
$T_{SH2,out}$ (°C)	820	863	847
$T_{sCO2HX/2,in}$ (°C)	-	755	744
$T_{sCO2HX/2,out}$ (°C)	-	628	644

Table 6.7: Tube metal temperatures at SH2 and sCO2 heat exchangers at nominal load

	Existing configuration	Single configuration	Dual configuration
SH2 maximum tube metal temperature (°C)	477	474	468
sCO2H1 maximum tube metal temperature (°C)	-	-	470
sCO2HX/2 maximum tube metal temperature (°C)	-	421	485

Figure 6.8 and Figure 6.9 show the tube wall heat fluxes and metal temperatures for the convective sCO2 heater (sCO2HX and sCO2H2 for the single and dual configuration respectively). Note the different scales between the figures for the metal temperatures. Localised maximum heat fluxes are observed at the top right half of the heat exchanger, influenced by the gas velocity profile incident to the heat exchanger. The opening from the boiler nose allows gas to enter the tube bank from the side, as shown in the middle contour in Figure 6.8. Additionally, peak metal temperatures were observed at the same region, peaking at 421 °C for the single heater configuration, and 485 °C for

the dual heater configuration. The higher metal temperatures for sCO2H2 were due to the higher internal sCO2 temperature, 473 °C compared to 406 °C for the single configuration. Metal temperatures were therefore 12 °C and 15 °C higher than internal sCO2 temperatures respectively. From Table 6.5, the total heat uptake for sCO2HX was 4755 kW compared to 3783 kW for the smaller sized sCO2H2. Similarly, water wall heat uptakes were also slightly lower for sCO2H2, due to reduced water wall surface area.

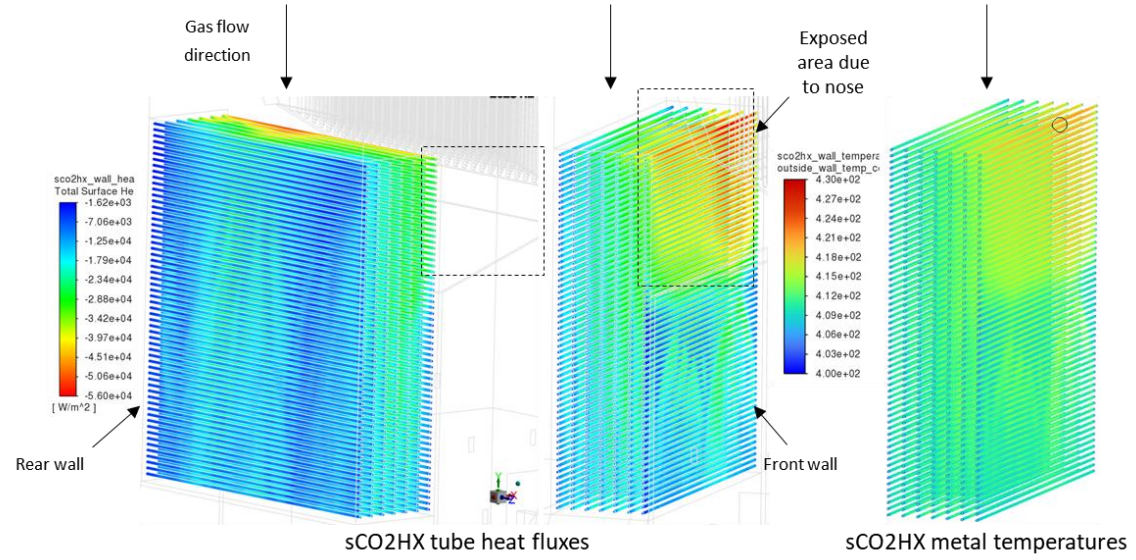


Figure 6.8: Single configuration sCO2HX tube wall heat fluxes and metal temperatures at nominal load

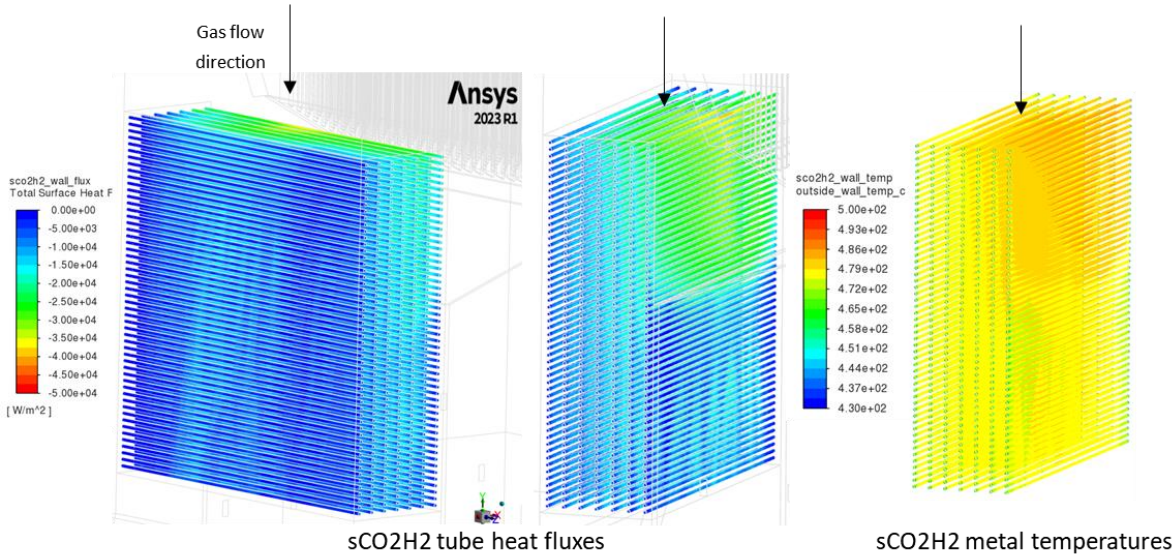


Figure 6.9: Dual configuration sCO2H2 tube wall heat fluxes and metal temperatures at nominal load

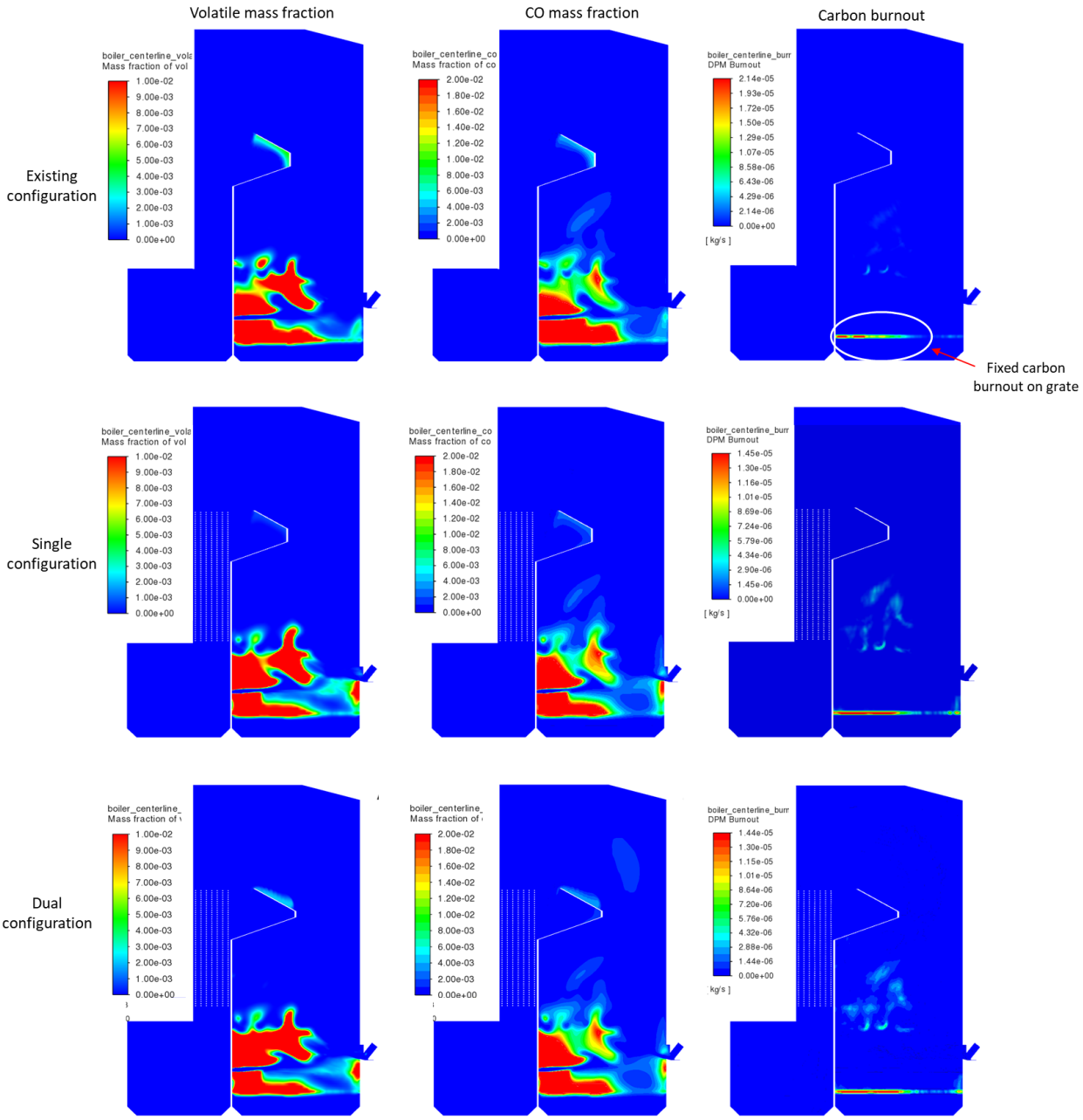


Figure 6.10: Volatile matter, CO mass fractions, and carbon burnout at nominal load for all configurations

Figure 6.10 shows the volatile matter (VM) and CO mass fractions, and carbon burnout for all configurations. The carbon burnout contours highlight that majority of the fixed carbon in the particles were consumed on the grate. For sCO2-integrated configurations, additional carbon burnout was observed in the freeboard as a result of overfiring. The VM contour shows that most of the release of VM takes place towards the rear wall of the furnace, above the grate in the freeboard. For sCO2-integrated configurations, a small amount of VM was also observed at the front wall below the fuel spreader. The CO mass fraction contours highlight high concentrations in the furnace where combustion takes place. Importantly, mass-weighted CO mass fractions were

negligible at the furnace exit, at 890 ppm, 517 ppm, and 570 ppm for the existing, single, and dual configurations respectively. The low CO concentrations at the furnace exit indicates complete combustion within the furnace volume.

6.2 Part load

Figure 6.11 shows the gas temperature distribution for all three configurations at 80% and 60% load. Compared to the nominal load case, the high temperature zone is shifted lower in the furnace.

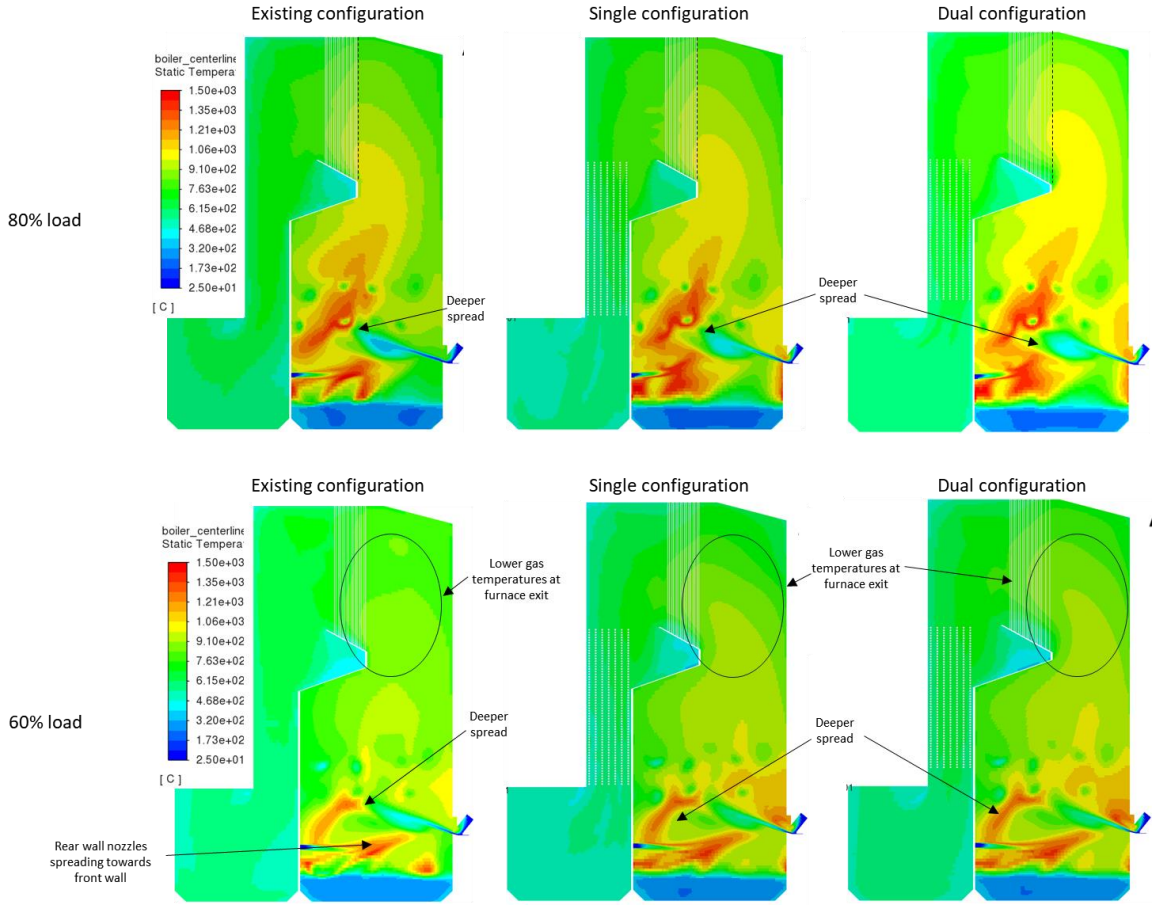


Figure 6.11: Gas temperatures at 80% and 60% load for all configurations

Additionally, from the inputs provided in Table 5.3, the direct air mass flow rate at the fuel spreader and rear nozzles remains fixed for all loads to ensure adequate spreading of the bagasse into the furnace domain. This results in a higher ratio of direct air mass flow rate to fuel flow rate, and consequently a deeper spread as shown in Figure 6.11, with combustion closer to the lower rear wall. This effect is further exacerbated by the lower PA and SA mass flow rates at lower loads. Table 6.8 shows the furnace exit gas temperatures and furnace metal temperatures for 80% and 60% loads. Similar to nominal load, at part loads, the single configuration had the highest furnace exit temperatures, followed by the dual configuration and the existing configuration. As expected, gas

temperatures were lower for low loads, and below bagasse ash deformation temperatures. Compared to nominal load, the mass weighted gas temperatures at the furnace exit were 3.2% lower on average at 80% load, and 13.5% lower on average at 60% load. Furnace metal temperatures were mainly influenced by the water saturation temperature and were relatively unchanged at 80% load and slightly lower at 60% load.

Table 6.8: Furnace gas temperatures and metal temperatures at part loads

Load %	Parameter	Existing configuration	Single configuration	Dual configuration
80%	Furnace maximum exit temperature (°C)	1107	1148	1111
	Furnace mass-weighted exit temperature (°C)	927	955	946
	Furnace maximum metal temperature (°C)	298	299	305
60%	Furnace maximum exit temperature (°C)	1026	1048	1037
	Furnace mass-weighted exit temperature (°C)	834	847	848
	Furnace maximum metal temperature (°C)	296	297	302

Figure 6.12 shows the heat flux distribution and furnace metal temperatures for all configurations at part loads. For part load cases, the heat flux distribution at the front wall is higher compared to the nominal load case. Additionally, the heat flux distribution at the rear wall is lower and is concentrated towards the lower half of the furnace. Compared to nominal load, these differences arise due to the higher ratio of DA mass flow rate at the rear wall nozzles for part load cases. Moreover, the heat flux at the nose is lower for the part load cases, particularly the 60% load case, due to the shift in heat flux distribution towards the front wall. Furnace metal temperatures were also reduced for the same reason, with the maximum furnace metal temperatures still located towards the lower rear wall of the furnace.

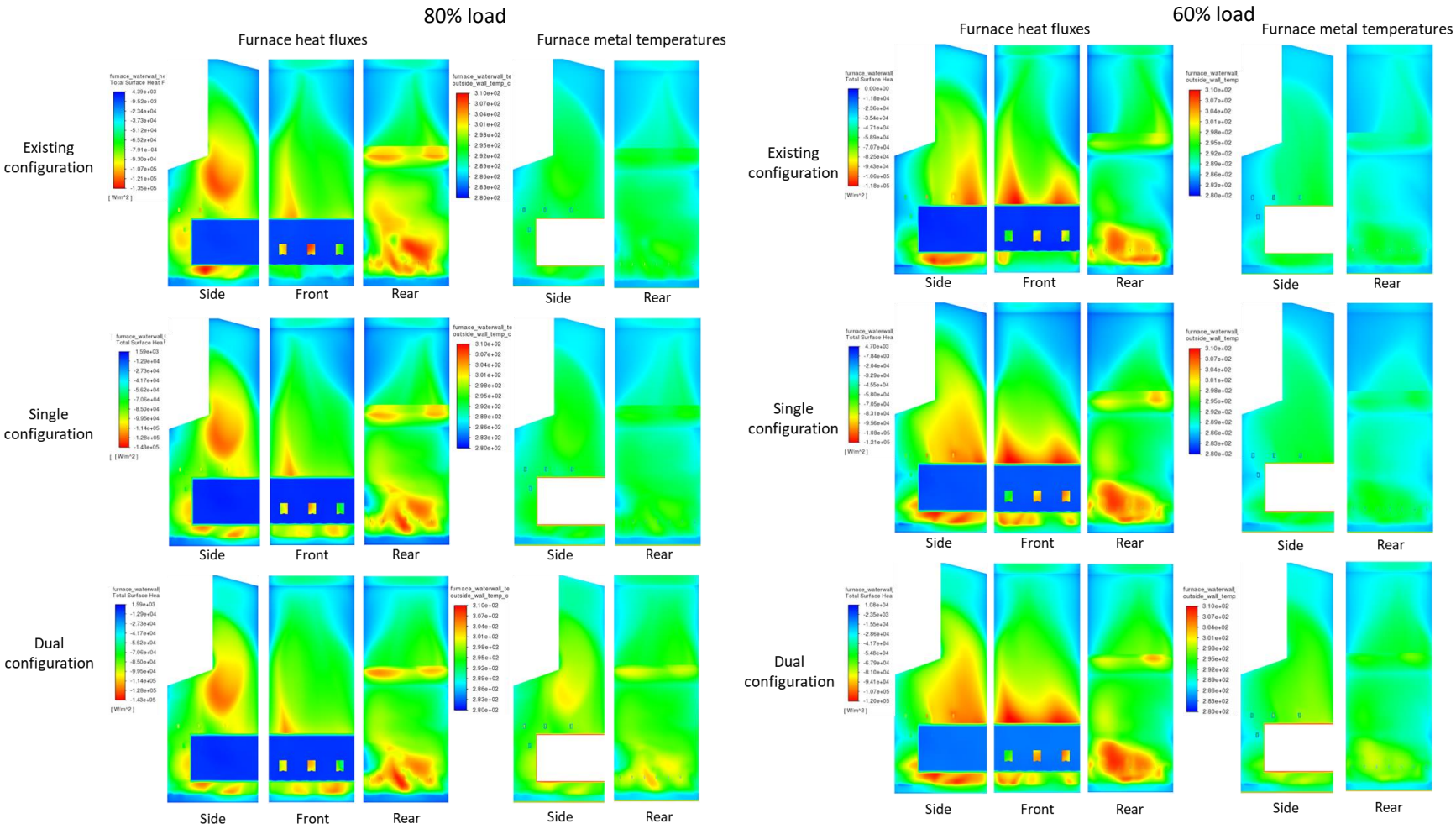


Figure 6.12: Furnace heat fluxes and metal temperatures at 80% load (left) and 60% load (right) for all configurations

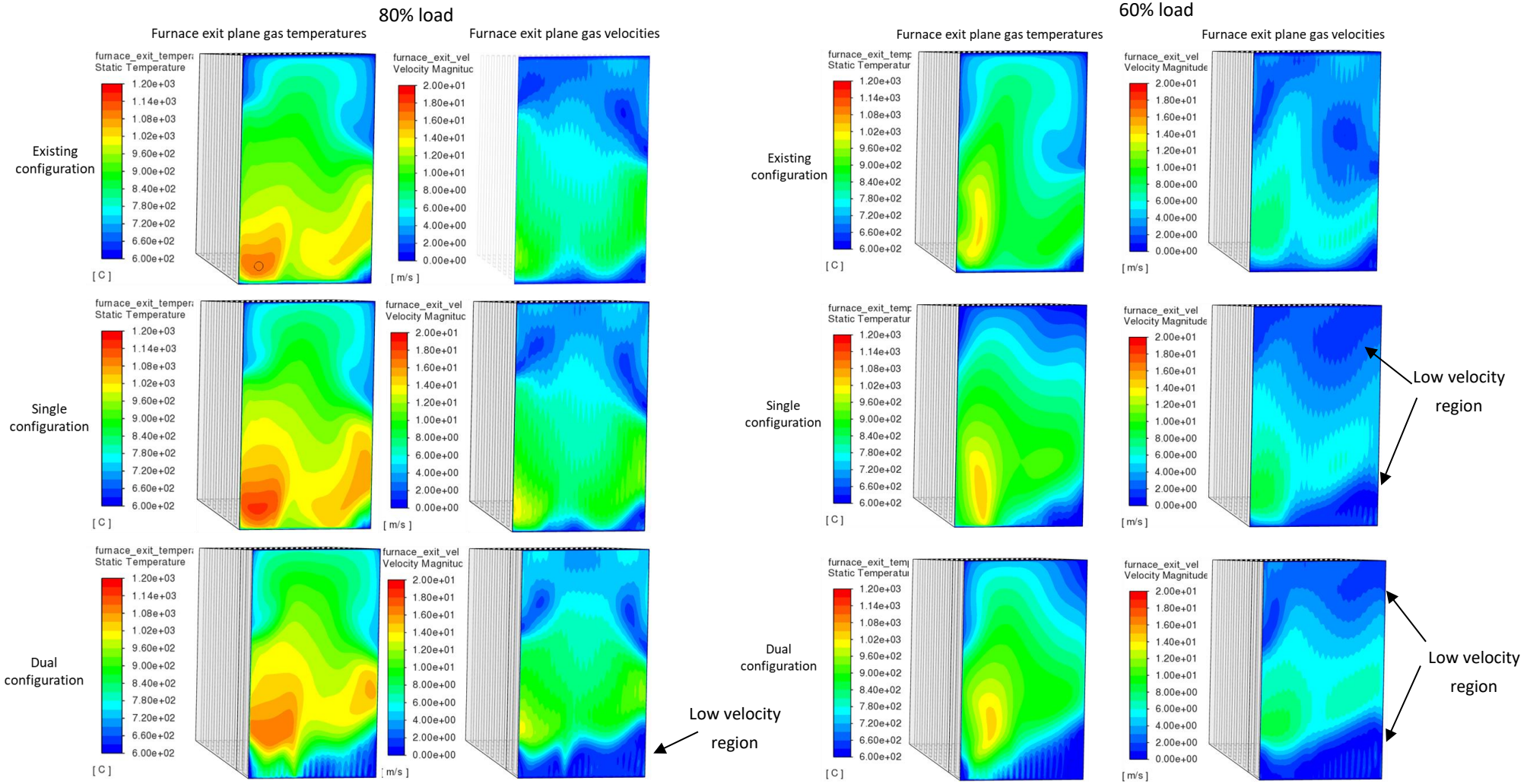


Figure 6.13: Furnace exit plane temperatures and velocity profiles at 80% load (left) and 60% load (right) for all configurations

Figure 6.13 shows the gas temperatures and velocity at the furnace exit plane for part load cases. Compared to the same parameters presented for the nominal load case in Figure 6.3, observed furnace exit gas temperatures are lower, with a change in the higher temperature regions. For the 80% load case, high temperature zones were observed at the lower half of the exit plane, towards the left and right sides. For the 60% case, high temperature zones were observed at the lower left half of the furnace exit plane. This contrasts with the nominal load cases, which had a relatively even temperature distribution across the lower half of the furnace exit plane. These differences were attributed to the variation in air mass flow rates, as well as the increase in DA mass flow rate ratios. Low gas velocities were observed towards the top of the furnace exit plane, i.e. the boiler roof. At 60% load, for sCO2 integrated configurations, low velocities were observed at both the top and bottom of the furnace exit plane, indicating regions of low mass flux and potential dead spots for SH2 and sCO2H1 for the dual configuration.

Table 6.9 shows the heat transfer rates at the furnace exit plane and to water walls surrounding the furnace at part load. As expected, heat transfer rates were lower for lower load percentages, due to the reduction in fuel, air, and flue gas mass flow rates produced from combustion. sCO2-integrated configurations had higher heat transfer rates due to overfiring. The dual configuration had the highest $\dot{Q}_{fe}$, as observed at nominal load. On average, $\dot{Q}_{ww,furn}$ was 1617 kW lower (-13.2%) at 80% load, and 3633 kW lower (-29.6%) at 60% load compared to nominal load.

Table 6.9: Furnace exit plane and water wall heat transfer rates at part load

		Existing configuration	Single configuration	Dual configuration
80% load	$\dot{Q}_{fe}$ (kW)	1804	1864	2319
	$\dot{Q}_{ww,furn}$ (kW)	10195	10878	10947
60% load	$\dot{Q}_{fe}$ (kW)	1456	1506	1732
	$\dot{Q}_{ww,furn}$ (kW)	8343	8820	8810

Figure 6.14 shows gas velocities at 80% and 60% loads respectively. Due to reduced flue gas flow rates, lower velocities were observed for lower loads. Notably, at the convective sCO2 heaters, fewer turbulent wakes were observed compared to the nominal load case, which would adversely impact convective heat transfer. Additionally, localised high velocity regions were observed towards SH2 around the boiler nose. At 60% load, regions of low velocity/dead spots were identified for

sCO2-integrated configurations at SH2, as previously discussed. High velocities were also observed at the rear wall nozzles and spreader distribution air.

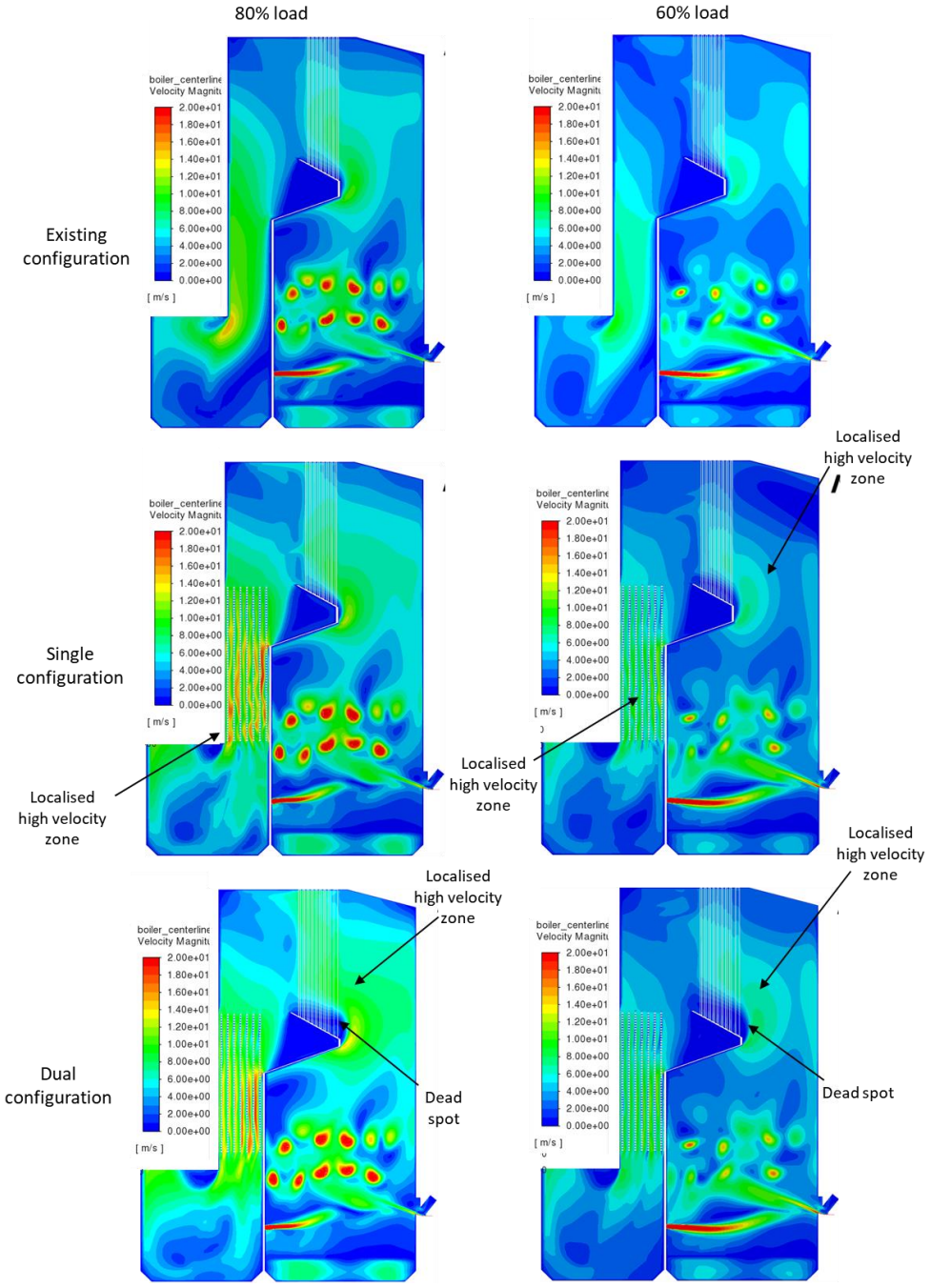


Figure 6.14: Gas velocities at 80% load (left) and 60% load (right) for all configurations

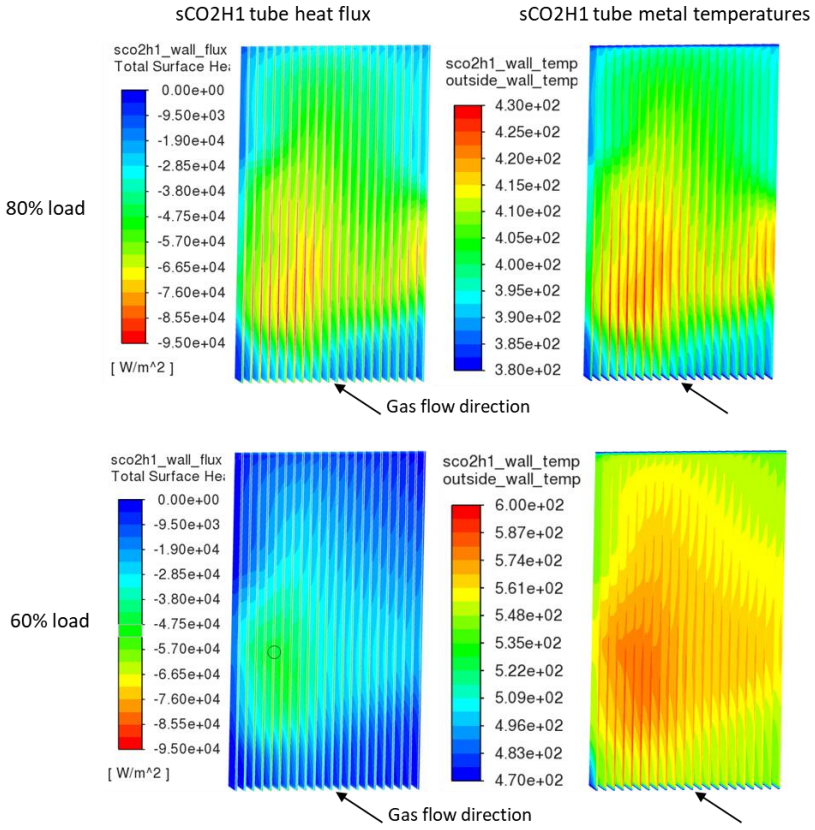


Figure 6.15: sCO2H1 heat fluxes and tube metal temperatures at part load

Figure 6.15 shows the heat fluxes and tube metal temperatures at sCO2H1 (dual configuration) at 80% and 60% loads. Localised maximum heat fluxes were identified as observed from the furnace gas exit temperature distributions in Figure 6.13. At 60% load, dead spots were observed by the low heat flux distributions around the heat exchanger extremities, particularly at the top and bottom. This is attributed to the lower velocities at 60% load and lower mass flux towards the boiler roof. The maximum tube metal temperatures were 427 °C at 80% load, and 586 °C at 60% load. The high tube metal temperature at 60% load was largely influenced by the high internal sCO2 temperature, 543 °C, compared to 381 °C at 80% load. The metal temperature is within operational limits of the sCO2 heat exchanger material (TP347HFG). The high internal fluid temperature was a result of operation near the sCO2 critical point, as highlighted in the part load study of the integrated 1D process modelling section. Reduction in this internal fluid temperature, which would then reduce metal temperatures, would be possible by adjusting operation at 60% load away from the critical point.

Figure 6.16 shows the heat fluxes and tube metal temperatures at SH2 for 80% and 60% loads. At 80% load, maximum heat fluxes were observed towards the lower left and right half of SH2, compared to only the lower right half at nominal load. At 60% load, the maximum heat flux was observed towards the lower left half of SH2. The heat flux distributions were influenced by the gas

mass flux/velocity, as shown in Figure 6.14. As expected, the highest metal temperatures were also observed at the regions with the highest heat flux. Dead spots can be observed in Figure 6.16 located towards the rear of SH2 at the top and bottom of the heat exchanger. Noticeably, at 60% load, the dual configuration had significantly larger dead spots, influenced by lower gas velocities and the additional heat uptake at sCO2H1. In comparison to the nominal load case, dead spots were observed in similar regions at low load, however particularly exacerbated at 60% loads for the dual configuration.

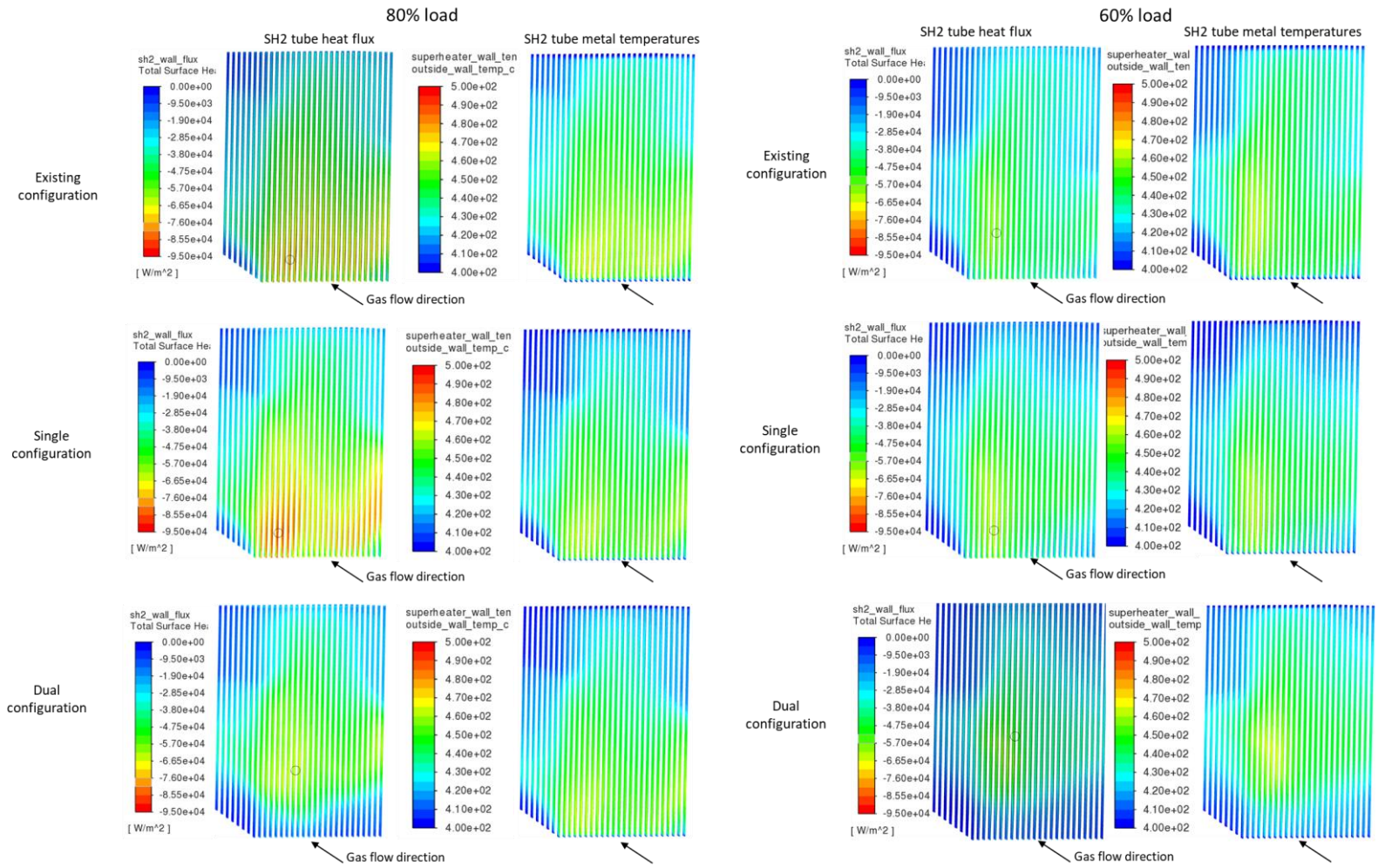


Figure 6.16: SH2 heat fluxes and tube metal temperatures at 80% load (left) and 60% load (right)

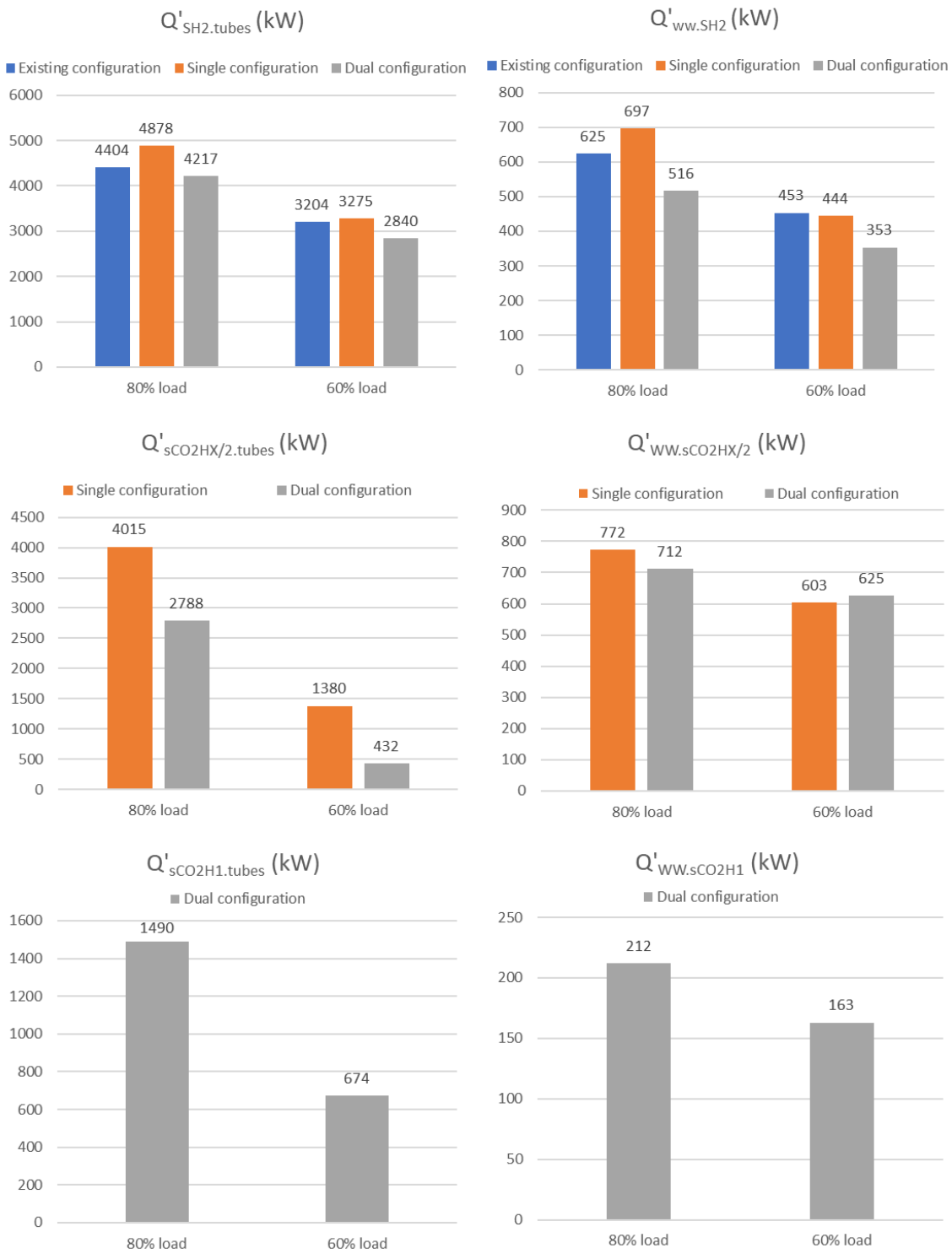


Figure 6.17: Heat transfer rates at heat exchangers and water walls at part loads

Figure 6.17 shows the heat uptakes at SH2 and sCO2 heaters, and the water walls surrounding those heat exchangers, at 80% and 60% load. Moreover, Figure 6.18 shows the inlet and outlet gas temperatures at SH2 and the sCO2 heaters at part load. Comparing heat uptakes at SH2, like the nominal load case, the single configuration had the highest heat uptakes, followed by the existing

configuration and lastly the dual configuration. The dual configuration had lower heat uptakes due to the heat uptakes and absorbed radiation at sCO2H1, as well as reduced incoming gas temperatures into SH2. Similarly, water wall heat uptake was also adversely affected for the dual configuration at 80% and 60% loads. On average, heat uptakes to SH2 tubes decreased by 896 kW (-16.6%) at 80% load, and 2289 kW (-42.4%) at 60% load compared to the nominal load cases.

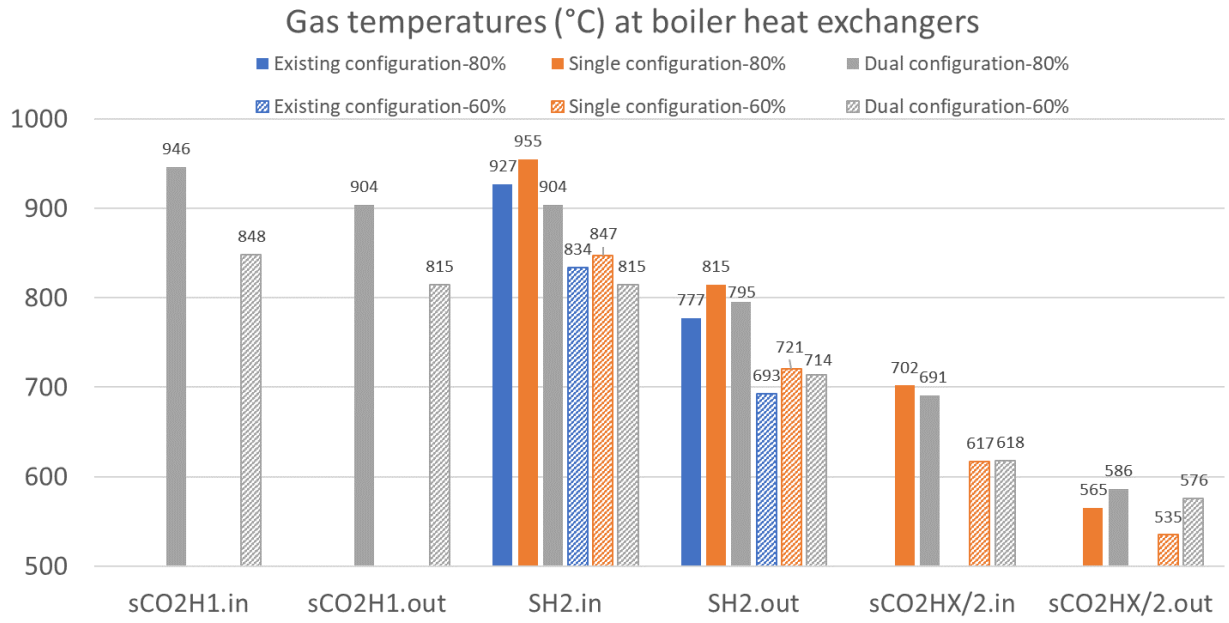


Figure 6.18: Mass-weighted gas temperatures at boiler heat exchangers for part loads

Comparing heat uptakes at sCO2H1, a significant decrease was observed at 60% load. Between 100% and 80% load, $\dot{Q}_{sCO2H1,tubes}$ decreased by 119 kW (-7.4%), and between 80% and 60% loads by 816 kW (-54.8%). The substantial decrease for the 60% load case can be attributed to the reduced direction radiation from the furnace exit plane, as well as reduced gas velocities, mass flow rates and inlet temperatures incident to sCO2H1, with a decrease of 98 °C observed at 60% load. Additionally, the internal sCO2 temperature for the 60% load case is the highest out of all loads (as shown in Table 5.3), due to operation near the critical point. These factors adversely impact the heat uptakes at the sCO2 heaters. For sCO2HX/2, these adverse effects were even more pronounced. Between 100% and 80% loads, a decrease of 740 kW (-15.6%) was observed for the single configuration, and 595 kW (-17.6%) for the dual configuration. In contrast, between 80% and 60% loads, a respective decrease of 2635 kW (-65.6%) and 2356 kW (-84.5%) was observed. The significant decrease, particularly for the dual configuration, was also due to the lower incoming gas temperatures and higher internal sCO2 temperatures. From Figure 6.18, gas temperatures at sCO2HX/2 inlet decreased by approximately 79 °C. The average temperature difference between

the internal sCO2 in the tube bank and the incoming gas for the single configuration was 280 °C and 112 °C at 80% and 60% load. Similarly, for the dual configuration, the temperature differences were 205 °C and 41 °C respectively. This can be visualised using heat exchanger duty (T-$\dot{Q}$) diagrams for the sCO2 heaters shown at the end of Appendix E, using Appendix Figure 11, Appendix Figure 12, and Appendix Figure 13. The low TTD, particularly for the dual configuration at 60% load, results in a sharp decrease in heat uptakes at the convective sCO2 heater. In contrast, water wall heat uptakes were affected less, due to the lower internal steam temperature at 285 °C, as shown in Figure 6.17. As the internal sCO2 temperatures were derived from the integrated 1D process model, the substantial decrease in heat uptakes at 60% load indicates adjustments would potentially be required for cycle operation at 60% load. This is explored further in the next section.

Table 6.10: Tube metal temperatures at SH2 and sCO2 heat exchangers at part loads

		Existing configuration	Single configuration	Dual configuration
80% load	SH2 maximum tube metal temperature (°C)	473	469	462
	sCO2H1 maximum tube metal temperature (°C)	-	-	427
	sCO2HX/2 maximum tube metal temperature (°C)	-	374	445
60% load	SH2 maximum tube metal temperature (°C)	471	469	469
	sCO2H1 maximum tube metal temperature (°C)	-	-	586
	sCO2HX/2 maximum tube metal temperature (°C)	-	474	563

Table 6.10 shows the tube metal temperatures at SH2 and the sCO2 heaters. As previously mentioned, the highest tube metal temperatures were observed at sCO2H1 at 60% load, followed by sCO2H2. While the temperatures are still below material specifications at approximately ≈ 600 °C , they are high relative to other operating cases. These high tube metal temperatures were primarily

influenced by the high internal sCO2 temperatures in the sCO2 heaters. The remainder of the metal temperatures were all below 475 °C.

Appendix E highlights a few additional results of the part load case study, including CO2 and H2O mass fractions, and tube heat flux distributions at the sCO2 heaters, as well as metal temperatures which were discussed above. VM, CO mass fractions, and carbon burnout are also presented. Slight differences were observed compared to the nominal load case, particularly with VM release towards the centre of the furnace, as opposed to near the rear wall for the nominal load case. The higher ratio of DA influences the trajectory of the bagasse particles from the spreader, specifically the rear wall nozzles. This results in a larger amount of VM towards the centre of the furnace at the grate.

7. Comparison of CFD results and calibration of integrated 1D process model

The main purpose behind the CFD modelling of the boiler was to validate heat uptakes at sCO2 heaters predicted by the 1D process model. In addition, practical implications were also explored with regards to heat exchanger geometry and spatial characteristics that cannot be captured by the 1D process model. This chapter compares the CFD modelling results with the 1D process model and addresses the variances by tuning various parameters within the 1D model.

7.1 Comparison of CFD and integrated 1D process modelling results

Table 7.1 compares various parameters between the 1D process model and the CFD model, including flue gas mass flow rate, furnace exit temperature, direct radiation at the furnace exit plane, and heat transfer to furnace water walls. Flue gas mass flow rates, $\dot{m}_{fg}$, and furnace exit temperatures, T_{fe}, were relatively the same between the 1D process model and CFD models. Particular attention is paid to $\dot{Q}_{ww,furn}$ and $\dot{Q}_{fe}$. At 100% load, the CFD models predicted a higher heat transfer rate to the furnace water walls, $\dot{Q}_{ww,furn}$, by 943 kW (+8.3%) on average. This discrepancy reduces at lower loads, at +5.6% and +3.2% for 80% and 60% loads respectively. Notably, the furnace exit temperatures were relatively the same for the existing and single configurations even with the difference in $\dot{Q}_{ww,furn}$. For the existing and single configurations, the direct radiation at the furnace exit, $\dot{Q}_{fe}$ is within 200 kW between the 1D process models and CFD models. However, for the dual configuration, the CFD model predicts a higher $\dot{Q}_{fe}$ at all load percentages, up to 34% higher at nominal load, dropping to 12% higher at 60% load. The higher $\dot{Q}_{fe}$ determined by the CFD model may be attributed to the adjustment of the furnace exit plane to accommodate sCO2H1, as well as geometric adjustments to the boiler nose.

Table 7.1: 1D process model compared to CFD results for various parameters

		Existing configuration				Single configuration				Dual configuration			
		1D Process	CFD	Rel diff %	Abs diff	1D Process	CFD	Rel diff %	Abs diff	1D Process	CFD	Rel diff %	Abs diff
100% load	$\dot{m}_{fg}$ (kg/s)	23.17	23.19	0.09	0.02	26.63	26.65	0.09	0.02	26.72	26.74	0.07	0.02
	T_{fe} (°C)	967	958	-0.96	-9	991	989	-0.21	-2	992	978	-1.37	-14
	$\dot{Q}_{fe}$ (kW)	1954	2137	9.34	183	2091	2190	4.75	99	2099	2815	34.13	716
	$\dot{Q}_{ww,furn}$ (kW)	11587	12659	7.22	782	11625	12600	9.25	1072	10830	11612	8.39	975
80% load	$\dot{m}_{fg}$ (kg/s)	18.62	18.63	0.05	0.01	21.58	21.60	0.09	0.02	21.69	21.64	-0.23	-0.05
	T_{fe} (°C)	923	927	0.41	4	951	955	0.42	4	952	946	-0.65	-6
	$\dot{Q}_{fe}$ (kW)	1732	1804	4.16	72	1865	1864	-0.08	-1	1873	2319	23.82	446
	$\dot{Q}_{ww,furn}$ (kW)	9598	10195	6.23	598	10337	10878	5.23	541	10380	10947	5.47	568
60% load	$\dot{m}_{fg}$ (kg/s)	13.94	13.93	-0.02	-0.003	15.03	15.03	-0.01	-0.002	15.13	15.13	-0.01	-0.002
	T_{fe} (°C)	862	834	-3.24	-28	878	847	-3.51	-31	879	848	-3.55	-31
	$\dot{Q}_{fe}$ (kW)	1471	1456	-1.01	-15	1531	1506	-1.63	-25	1538	1732	12.58	193
	$\dot{Q}_{ww,furn}$ (kW)	8151	8343	2.36	193	8485	8820	3.95	335	8525	8810	3.35	285

Due to the boiler nose adjustment, the variation of heat flux along the height of the furnace is impacted compared to the single and existing configurations, as shown in the heat flux contours in Figure 6.4 and Figure 6.12. The 1D process model can be tuned by adjusting the non-uniformity factor η_{FE}, which in turn adjusts $\dot{Q}_{fe}$. Moreover, the species mass fractions determined by the CFD model and process model at the boiler outlet were within 2%, with only O_2 species fractions within 5%.

Figure 7.1 shows the heat uptakes at SH2 tubes and water walls for the 1D process model and CFD models. For all configurations, the CFD model predicted a slightly higher heat uptake for the tube bank at 100% and 80% load by 5.1% and 1.7% on average. At 60% load, the CFD model predicted lower heat uptakes at the tube bank, -211 kW (-6.2%), -364 kW (-10.0%), and -274 kW (-8.8%) for the existing, single, and dual configurations respectively. Interestingly, the CFD model already underpredicts the heat uptakes for the existing configuration without any sCO2-integration, indicating sCO2-integrated configurations would be even more impacted.

The direct radiation at the furnace exit, $\dot{Q}_{fe}$, was relatively the same between the 1D process model and the CFD model (for the existing and single configurations). At 60% load, the CFD model calculated lower furnace exit temperatures, by approximately 30 °C (-3.4%) on average, indicating lower gas temperatures inlet to SH2, which would reduce heat uptakes, including direct gas radiation. Observing water wall heat transfer, the CFD model calculates lower heat uptakes for water walls for all cases. This holds true for the sCO2 heaters as well. To calculate water wall heat transfer in the 1D process model, the heat transfer coefficients calculated for the tube bank were also used to calculate water wall heat transfer, with correction factors. The lower heat uptake predicted by the CFD model indicates the correction factors used for heat transfer to water walls in the 1D process model, β_{conv} and β_{rad} from Eq. (3.52), require further adjustment.

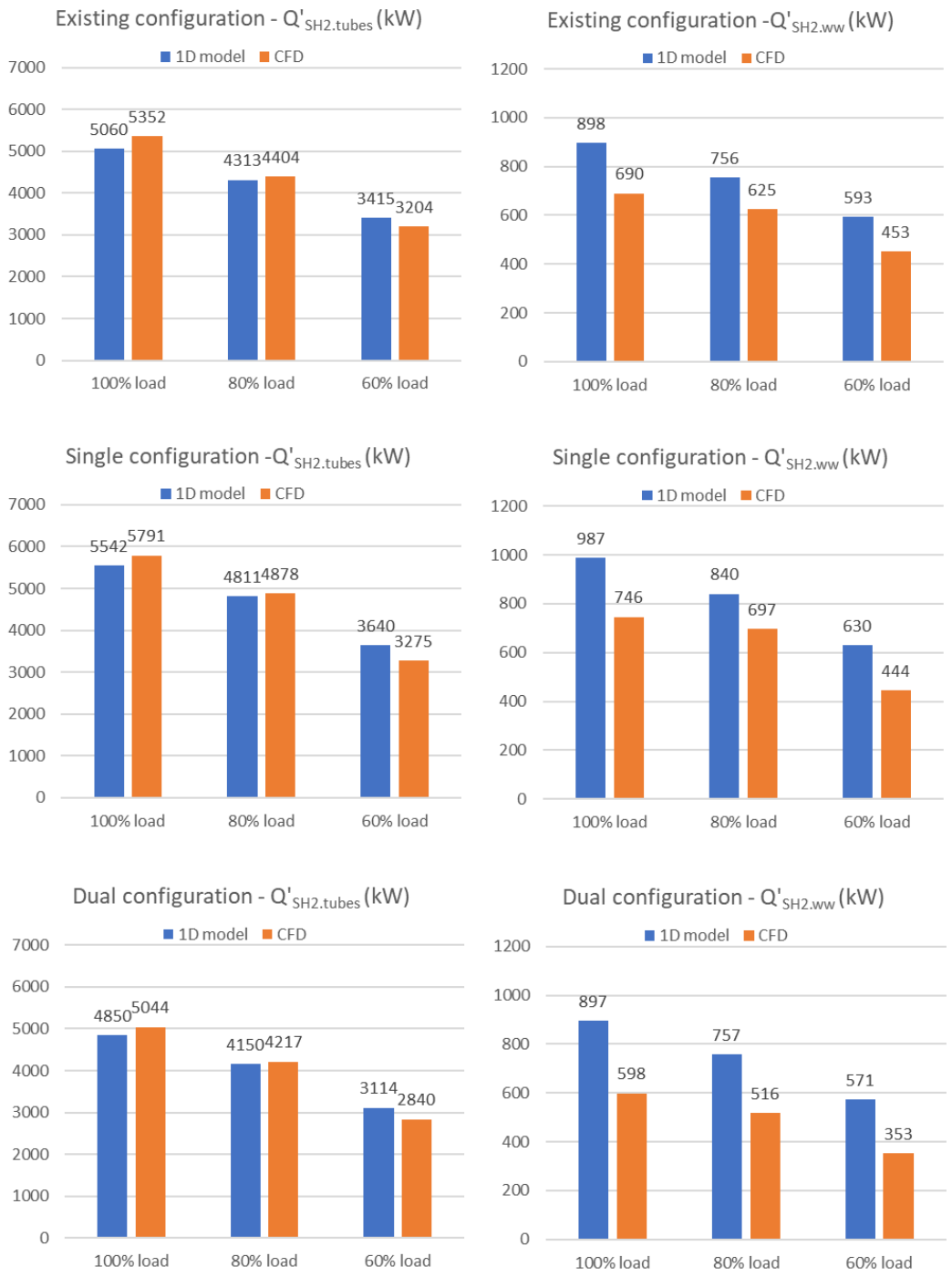


Figure 7.1: SH2 tubes (left) and water wall (right) heat uptakes from 1D process model and CFD model at all loads

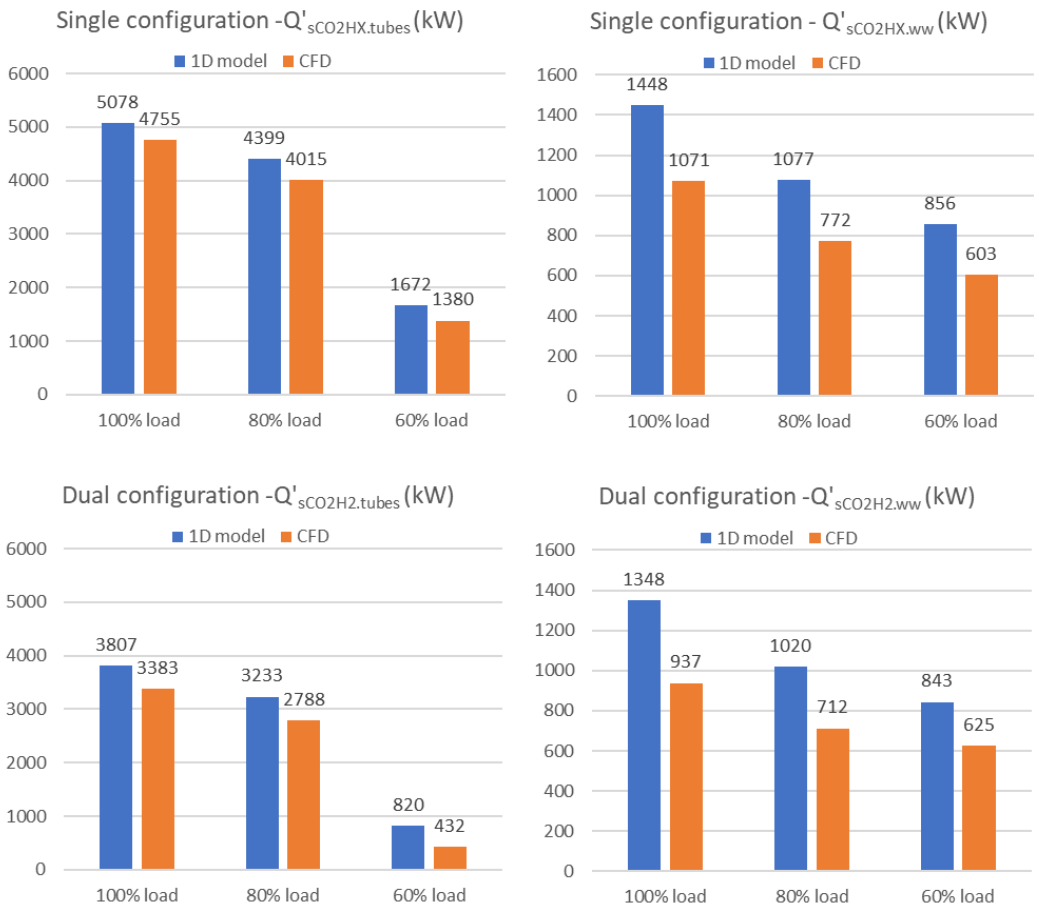


Figure 7.2: sCO2HX/2 tubes (left) and water wall (right) heat uptakes from 1D process model and CFD model at all loads

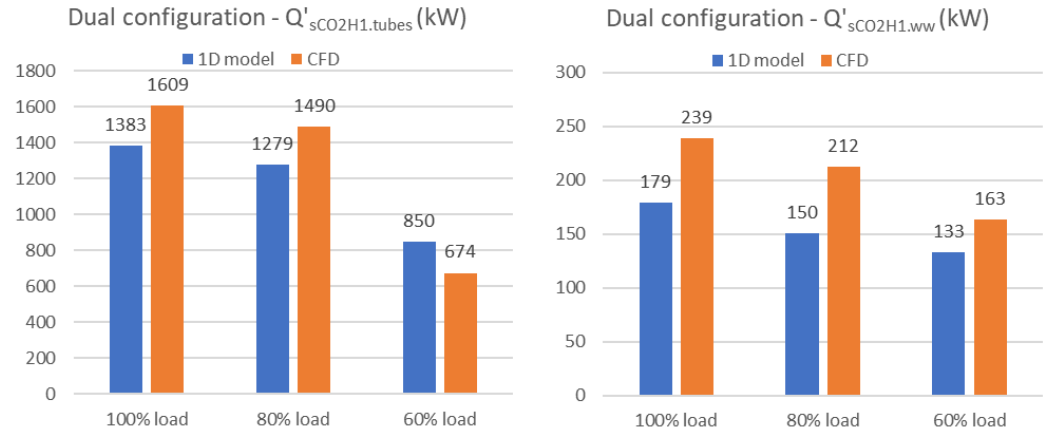


Figure 7.3: sCO2H1 tube (left) and water wall (right) heat uptakes from 1D process model and CFD model at all loads

Figure 7.2 shows the heat uptakes at the convective sCO2 heaters, sCO2HX/2 for the single and dual configuration respectively. The CFD model underpredicts heat uptakes at the tube bank for both the single and dual configurations. For the single configuration, the CFD model underpredicts by 322 kW (-6.4%), 384 kW (-8.7%) and 292 kW (-10.0%) at 100%, 80%, and 60% loads respectively. For the dual configuration, a bigger discrepancy was observed, with an underprediction of 425 kW (-11.2%), 445 kW (-13.8%) and 388 kW (-47.3%) respectively. On average, gas inlet temperatures to the convective sCO2 heaters were 6% lower for the single configuration, and 5% lower for the dual configuration compared to the 1D model. The significant discrepancy for the dual configuration at 60% load was attributed to the low TTD between the gas and internal sCO2 temperature as previously discussed. Inlet gas temperatures for that case were approximately 40 °C higher for the 1D process model, resulting in higher predicted heat uptakes compared to the CFD model.

It is important to note that the CFD model also captures the accurate inlet gas velocity profile, while the 1D process model assumes a uniform profile. As shown in the heat flux contours in the previous section, the gas inlet to the convective sCO2 heat exchanger enters at an angle due to the opening created by the boiler nose. This results in non-uniform distribution of the gas over the tube bank, which reduces heat uptakes in certain parts of the heat exchanger. The large discrepancies at 60% load were driven by the lower TTD across the tube bank. The high internal sCO2 temperatures at 60% load were caused by operation close to the critical point, which results in a rapid decrease in sCO2 mass flow rate across the compressor, thereby resulting in a high sCO2 temperature in the heat exchanger.

As the CFD model highlights large decreases in heat uptakes at 60% load, cycle operation at 60% load may need to be adjusted. This can be done by adjusting the compressor operating pressure/temperature at this load percentage or adjusting turbomachinery performance curves to shift operation away from the pseudo-critical point. Additionally, to address the slight decreases in heat uptakes at 100% and 80% loads, the sCO2 cycle capacity may need to be reduced slightly to reach the required cycle maximum temperature of 550 °C. This would again require adjustments to component characteristics of the cycle, particularly turbomachinery and valves. For the single configuration, the variances observed between the CFD model and 1D process model were within 10% maximum, which is within the uncertainty range of the heat transfer correlations used in the 1D process model. Nevertheless, a heat exchanger utilisation factor could also be used to tune the convective heat exchanger performance for the 1D model.

Figure 7.3 shows the heat uptakes at the radiative platen sCO2H1 for the dual configuration. At 100% and 80% loads, the CFD model predicts a higher heat uptake compared to the 1D process model at +226 kW (16.4%) and +211 kW (16.5%). This is driven largely by the higher predicted direct radiation from the furnace exit plane, $\dot{Q}_{fe}$, which increases absorbed radiation at sCO2H1. At 60%

load, the CFD model predicts only a slightly higher $\dot{Q}_{fe}$ compared to the higher load cases, as shown in Table 7.1. Additionally, the incoming gas temperature from the furnace exit is slightly lower compared to the 1D process model (-31 °C). This results in a slightly lower heat uptake for sCO2H1 at 60% load, 176 kW lower than the 1D process model. For water wall heat uptakes surrounding sCO2H1, the CFD model calculates slightly higher heat uptakes for all load cases, at +51 kW on average.

7.2 Adjustments to integrated 1D process models

To inform the 1D process model by the CFD model results, several adjustments were made to the 1D process model. For sCO2 heaters, a heat exchanger utilisation factor, ψ_{sCO2}, was used to factor in discrepancies between the heat uptakes predicted by the CFD model and the 1D process model. Practically, the heat exchanger utilisation factor is used to account for dead spots on the heat exchanger. UA_{fg} from Eq. (3.49) is multiplied by $(1+\psi_{sCO2})$. For the dual configuration, to account for the adjustments to the boiler nose and resulting effects, the non-uniformity factor, η_{FE}, from Eq. (3.33) is multiplied by a correction factor, $(1+\alpha_{NUF})$ based on the calculated $\dot{Q}_{fe}$ from the CFD models. Table 7.2 shows the tuned parameters for the 1D process model.

It is important to note that tuning the 1D model using the CFD model is an iterative process. By adjusting the parameters in Table 7.2, integrated cycle performance is impacted, including sCO2 and gas temperatures, as well as other inputs and boundary conditions used for the CFD model. These changes would update the boundary conditions used for the CFD models, which would then need to be simulated again. This lies outside the scope of the current study. The integrated 1D model is updated based on the CFD results discussed in the previous sections, demonstrating how CFD modelling can be used to inform the 1D process model. Further iterations are recommended for future work.

Table 7.2: Tuned parameters for the 1D process model

		100% load	80% load	60% load
Single configuration	ψ_{sCO2HX}	-0.42	-0.45	-0.8
Dual configuration	ψ_{sCO2H1}	0.00	0.00	-0.70
	ψ_{sCO2H2}	-0.20	-0.45	-0.90
	α_{NUF}	0.18	0.18	0.18

Table 7.3: Initial and tuned integrated 1D model high level results for the single configuration

Single configuration	100% load		80% load		60% load	
	Initial 1D model	Tuned 1D model	Initial 1D model	Tuned 1D model	Initial 1D model	Tuned 1D model
$\dot{W}_{net,tot}$ (kW)	10436	10262	8535	8313	5924	5798
$\dot{W}_{net.Brayton}$ (kW)	1860	1687	1580	1357	632	506
η_{tot} (%)	26.03	25.87	26.15	25.64	25.92	25.43
$\eta_{Brayton}$ (%)	35.75	34.51	35.00	31.54	37.60	31.82
$T_{max,Brayton}$ (°C)	557	510	502	421	554	429
$\dot{m}_{sCO2}$ (kg/s)	19.30	19.76	17.86	18.35	9.07	9.53

Table 7.4: Initial and tuned integrated 1D model high level results for the dual configuration

Dual configuration	100% load		80% load		60% load	
	Initial 1D model	Tuned 1D model	Initial 1D model	Tuned 1D model	Initial 1D model	Tuned 1D model
$\dot{W}_{net,tot}$ (kW)	10390	10341	8525	8323	5887	5637
$\dot{W}_{net.Brayton}$ (kW)	1815	1766	1569	1368	644	394
η_{tot} (%)	26.03	26.02	26.11	25.68	25.61	24.67
$\eta_{Brayton}$ (%)	35.03	35.08	34.79	31.94	38.37	26.13
$T_{max,Brayton}$ (°C)	553	530	507	429	594	333
$\dot{m}_{sCO2}$ (kg/s)	19.11	19.34	17.67	18.19	8.77	9.88

Table 7.3 and Table 7.4 shows the tuned as well as the initial integrated 1D model results for comparison for the single and dual configurations respectively. The tuned results were simulated by implementing the tuned parameters in Table 7.2 and re-simulating all load cases to convergence using the prescribed solving methodology.

From Table 7.3, for the single configuration, the maximum Brayton cycle temperature, $T_{max,Brayton}$, reduces at all loads due to the decrease in heat uptakes enforced by ψ_{sCO2HX}. This leads to a decrease in Brayton cycle thermal efficiency compared to the initial 1D model results, up to a 5.8% decrease at 60% load. Consequently, the overall thermal efficiency also decreases. For the worst case, at 60% load, the overall thermal efficiency decreases from 25.92% to 25.43%, while net Brayton cycle generation decreases from 632 kW to 506 kW. Due to the decrease in turbine inlet temperature/ $T_{max,Brayton}$, the operating point on the turbine shifts. As a result, the tuned 1D models exhibit slightly higher sCO2 mass flow rates compared to the initial model results.

From Table 7.4 for the dual configuration, similar results were observed as the single configuration. However, sharper decreases in heat uptakes for sCO2 heaters were predicted by the CFD model at 60% load compared to the single configuration. Consequently, the heat exchanger utilisation factors, ψ_{sCO2H1} and ψ_{sCO2H2}, were lower at 60% load for the dual configuration. This resulted in a significantly lower $T_{max,Brayton}$ at 60% load for the dual configuration compared to the single configuration, and consequently a lower $\eta_{Brayton}$. As previously discussed, the low heat uptakes at 60% load were driven primarily by the low TTD across sCO2H2. With the tuned 1D model results, the average internal sCO2 temperature across the convective heater is $T_{sCO2H2,avg} = 348\,°C$, compared to 553 °C from the initial results. The updated internal sCO2 temperature would result in a significantly higher TTD compared to that currently used in the CFD model, which highlights the importance of an iterative procedure between the 1D model and CFD models.

Another improvement that can be made to the integrated cycle is operation of the Brayton cycle at 60% load. Currently, compressor inlet conditions lie close to the pseudo-critical point, which results in a sharp decrease in sCO2 mass flow rate, as can be seen in Table 7.3 and Table 7.4 above. For the initial 1D model, this resulted in significant increases in average sCO2 temperatures across the sCO2 heaters. The compressor inlet pressure/temperature can be increased to move away from the pseudo-critical point, which should increase sCO2 Brayton cycle capacity at 60% load and reduce the fluctuation in fluid temperatures. This would improve net generation; however, thermal efficiency may be adversely affected due to lower $T_{max,Brayton}$ achievable with higher sCO2 mass flow rates. Nonetheless, further iterations between the CFD model and 1D model will be required particularly for the 60% load cases. Slight degradations in overall cycle performance were observed

with the updated 1D model results for the 100% and 80% load cases in terms of Brayton cycle efficiency and net electrical power output. To improve Brayton cycle efficiency, the Brayton cycle capacity may need to be reduced slightly to achieve nominal maximum Brayton cycle temperatures at 550 °C. This would require adjustments to turbomachinery performance characteristics to reduce sCO2 mass flow rates at nominal operating conditions.

8. Summary, conclusions, and recommendations

The focus of this research was to investigate the integration of a sCO2 Brayton cycle with a combined heat and power steam cycle with a modular biomass boiler firing typical Southern African bagasse fuel. Two numerical modelling methods were used, namely integrated 1D thermofluid process modelling, and high-fidelity 3D CFD modelling. The low computational expense and robustness of the 1D thermofluid process model was leveraged to simulate the Rankine and Brayton cycles along with the biomass boiler in an integrated manner, gaining insight into overall cycle and component level performance. Due to the novelty of the proposed cycle, high-fidelity CFD modelling was used to validate the heat uptakes predicted by the 1D model at the sCO2 heaters situated in the boiler. Additionally, by capturing spatial characteristics in the CFD model, which the 1D process model cannot do, various practical implications of the sCO2 Brayton cycle integration were also explored.

8.1 Integrated 1D process model

To facilitate the analysis of these intricate systems, a sophisticated simulation code was developed to allow for necessary customization and enforcement of required boundary conditions and control parameters. The quasi-steady state 1D thermofluid network-based process model of the sCO2, steam and flue gas cycles was developed for nominal and partial load analysis. It accounted for the detailed component characteristics for the Rankine and Brayton cycles, as well as the biomass boiler, together with the complex interactions between all the components in the different cycles. Particular attention was paid to the specifications and layout of the sCO2 heaters situated in the boiler. The 1D thermofluid process model was used to analyse the integrated cycle performance at nominal and part loads for the two sCO2 Brayton cycle heater configurations.

For the sCO2-integrated cycle configurations a 15.3% increase (+7.67 MW_{th}) in fuel firing rate was required to provide sufficient heat uptake for both the Rankine and Brayton cycles. A 21.2% increase in net generation was observed for sCO2-integrated cycles, with an additional 1.9 MW generated by the sCO2 Brayton cycle in addition to the 10.8 MW of combined heat and power generated by the Rankine cycle. The average incremental efficiency arising from the sCO2-integration was 24.7%. This highlighted the potential for increased net generation at higher thermal efficiencies, as well as the feasibility of the proposed integration.

For the sCO2-integrated configurations, furnace exit temperatures were elevated due to overfiring. This resulted in increased radiative heat uptake at water walls surrounding the furnace, contributing a higher percentage to the total evaporative circuit heat load compared to the existing cycle. For the dual configuration, the sCO2H1 absorbed a portion of the incoming radiative heat from the

furnace, resulting in a decrease of absorbed radiant heat at SH2 compared to the existing case. Conversely, for the single configuration, there was an increase in heat uptake at SH2 due to the larger flue gas mass flow rate from overfiring. The convective sCO2 heaters had considerable impact on downstream heat exchangers, including reduced flue gas temperatures and heat uptake at SH1.

The total heat uptake at the superheaters required to heat steam to the desired temperature remained the same for all configurations. However, the heat uptake ratio between SH1 and SH2 was affected. Consequently, attemperation for the dual configuration decreased more compared to the single configuration. The reduced heat uptakes at evaporative heat exchangers downstream of the convective sCO2 heaters was counteracted by the increased heat uptake at furnace waterwalls and economisers.

The results of the part load analysis show that inventory control was the most effective control methodology for slow load following, maintaining high thermal efficiency for a wide load range. In comparison, the use of a throttling valve to maintain the maximum cycle temperature resulted in a decrease in thermal efficiency and net electrical power output due to the additional pressure loss and reduced turbine work. With the use of inventory control at 60% load, the compressor inlet pressure and temperature were close to the pseudo-critical point. This resulted in a sharp decrease in sCO2 Brayton cycle capacity, as the density of sCO2 nearly halves compared to the higher load cases, causing a substantial decrease in sCO2 mass flow rate through the compressor.

With inventory control selected as the preferred control methodology for the sCO2 Brayton cycle, the single and dual heater configurations were simulated and compared. It was found that the dual heater configuration performed slightly better at lower loads. For lower loads, the heat uptake at the radiative sCO2 platen heater decreased less compared to the convective dominant sCO2 heaters. This was due to the direct radiation absorbed by the radiative sCO2 heater, while convective-dominant heaters were more negatively affected at lower loads due to decreased flow rates.

For the dual heater configuration, the placement of the radiative and convective sCO2 heaters impacted both SH1 and SH2. For the single heater configuration, only SH1 was adversely impacted. The heat uptakes at the Rankine cycle superheaters were therefore impacted more for the dual heater configuration, which was demonstrated by the reduced Rankine cycle attemperation requirement.

The outcomes from the integrated 1D process modelling addressed the first two research objectives proposed in section 1.4, namely the development of a 1D process model to analyse the proposed integrated cycle, and the analysis of the cycle at nominal and part loads. The developed simulation code allowed for the necessary customization and enforcement of required boundary conditions

and control parameters for the various operating conditions in a computationally efficient and robust manner. Moreover, overall cycle- and component-level performance were characterised. The integration of the sCO2 Brayton cycle highlighted the potential for increased net electrical output as well as higher overall thermal efficiencies. Impacts on Rankine cycle performance was quantified by an adjustment in the boiler heat load profile and reduced attemperation requirements between the superheaters, indicating adjustments in operation philosophy required for the integrated cycle. At low loads, particularly 60% load where the sCO2 compressor operates near the pseudo-critical point, large fluctuations in cycle performance were observed. This signified the importance of carefully managing inventory control at lower loads, particularly near the pseudo-critical point, in order to avoid large reductions in sCO2 compressor throughput. It may be beneficial to increase the compressor inlet pressure or inlet temperature at 60% load to avoid the large fluctuations in sCO2 fluid properties near the pseudo-critical point. Based on the 1D process model results alone, the single heater configuration is recommended as it had less impact on the Rankine cycle and only performed marginally worse at lower loads compared to the dual heater configuration. The integration of the sCO2 Brayton cycle demonstrated tangible benefits at nominal and part loads and could potentially be a more feasible option to retrofit existing boilers for increased net electrical power output and to improve thermal efficiencies. The sCO2 Brayton cycle integration has been investigated in this work as load-following the Rankine cycle. Prioritising the Brayton cycle for power generation, with its higher thermal efficiencies, while using the Rankine cycle for process heat generation may also be a configuration worth exploring.

8.2 Biomass boiler CFD modelling

High-fidelity CFD modelling of the biomass boiler was used to validate the heat uptakes calculated by the 1D process model for the sCO2 heaters. 3D CFD modelling captures various spatial characteristics and detailed radiative heat transfer that the 1D process model cannot. Therefore, practical implications of the sCO2 heater integrations were explored using the CFD model, in addition to detailed combustion characteristics and any effects caused by overfiring.

The CFD models characterised gas temperatures and velocities, tube and water wall heat fluxes, tube metal temperatures and more. For all load cases and configurations, including sCO2-integrated configurations with overfiring, furnace exit temperatures were below the ash deformation temperature of bagasse. The maximum metal temperature observed in the furnace was 305 °C, well within operating limits, with the dual configuration exhibiting slightly higher metal temperatures compared to other configurations. Compared to the existing configuration, sCO2-integrated configurations exhibited higher furnace exit temperatures and velocities due to overfiring, as expected. At 100% load, heat fluxes in the furnace were concentrated towards the side and rear

walls. In contrast, at 80% and 60% loads, heat fluxes were slightly higher towards the front wall, caused by the increase in the ratio of DA at lower loads and higher air flow rates at rear wall nozzles.

For the convective sCO2 heaters (sCO2HX and sCO2H2 for the single and dual configurations respectively), a non-uniform gas velocity inlet profile was observed. The opening created by the boiler nose resulted in gas entering the tube bank towards the top right corner, resulting in localised maximum heat fluxes as well as tube metal temperatures in that region. For the dual configuration, the adjustment of the boiler nose to accommodate the radiative platen type heater, sCO2H1, shifted the flame position closer to the boiler roof and influenced the gas flow path. At 100% load, this resulted in higher heat fluxes observed at the boiler roof. Moreover, because of the boiler nose adjustment, a dead spot was observed towards the bottom of both sCO2H1 and SH2. This was exacerbated further at lower loads due to lower gas mass flux, with dead spots identified at the top and bottom of sCO2H1. The direct radiation at the furnace exit was also impacted, with a 28% increase observed for the dual configuration compared to the single configuration at 100% load.

Heat uptakes were characterised at SH2 and the sCO2 heaters. For the dual configuration, the reduction in heat uptakes at SH2 due to the heat absorbed by sCO2H1 was quantified. For both sCO2-integrated configurations, heat uptakes at sCO2 heaters decreased substantially from 80% to 60% load. At 60% load, as simulated by the 1D process model, the sCO2 cycle compressor inlet conditions were close to the pseudo-critical point, resulting in a large decrease in sCO2 mass flow rate, and consequently, a higher maximum cycle temperature. This increased the average sCO2 temperature in the convective sCO2 heaters. Moreover, at 60% load, incoming gas temperatures were also reduced compared to higher loads. This resulted in substantially lower TTDs in the CFD model. These factors resulted in significantly lower heat uptakes at 60% load. The highest tube metal temperatures were observed for the dual configuration at 60% load, with 586 °C observed at sCO2H1 and 563 °C observed at sCO2H2, largely influenced by the high internal sCO2 temperatures at 60% load. While these metal temperatures were within operating limits, they were highly elevated compared to other load cases.

By capturing detailed combustion processes and spatial characteristics within the boiler, the CFD models provided additional insights into the impacts of the sCO2 heater integrations into the biomass boiler. Quantifying impacts from practical implications arising from the integration of the sCO2 heaters, such as modifications to existing boiler geometry, can only be captured by CFD models in the absence of plant measurements.

Heat flux distribution within the furnace were concentrated at the rear wall for higher loads, compared to the front wall for lower loads. Moreover, for the dual configuration which required adjustments to the boiler nose, the shifted flame position also resulted in higher heat fluxes at the boiler roof. The shifts in heat flux distribution results in different hotspots for tube metal

temperatures, which is an important consideration for potential failure zones and maintenance considerations. For the dual configuration, higher maximum furnace tube metal temperatures were observed. Additionally, at 60% load, the dual configuration had the highest tube metal temperatures at the sCO2 heaters. While within specification, prolonged exposure to these elevated metal temperatures can decrease the material lifetime, leading to additional maintenance requirements and costs. Other than increased metal temperatures, no considerable issues with overfiring were observed, indicating feasibility of overfiring required for the sCO2-integrated configurations. For the convective sCO2 heaters, highly turbulent wakes were observed, particularly at higher load. To better characterise heat transfer for the convective sCO2 heaters a more refined mesh would be required to resolve the viscous sublayer, which would introduce significant computational expense at the current scale of the model. A standalone model of the convective heaters may therefore be beneficial, using boundary conditions obtained from the present research.

8.3 Comparison of CFD results and adjustments to 1D process model

The results from the CFD modelling, particularly the heat uptakes at the sCO2 heaters, were compared to those from the 1D model. Various parameters in the 1D model were then tuned to ensure parity with the heat uptakes observed with the CFD models, and the 1D models simulated to convergence.

Furnace exit temperatures were very similar between CFD and 1D model results. The CFD models predicted a higher $\dot{Q}_{ww,furn}$, up to 8.3% higher at 100% load. For the existing and single configurations, the incoming direct radiation through the furnace exit plane, $\dot{Q}_{fe}$, was within 200 kW between CFD and 1D model results. For the dual configuration, the CFD models predicted a higher $\dot{Q}_{fe}$ for all load percentages, up to 34% higher at 100% load. The higher $\dot{Q}_{fe}$ for the dual configuration was attributed to the boiler nose adjustments, which affected the flame position within the furnace, and the variation of heat flux along the furnace height. For the 1D model, the non-uniformity factor, η_{FE}, was tuned to account for this.

Comparing heat uptakes at the convective sCO2 heaters, the CFD model predicted lower heat uptakes for both the single and dual configurations, with a larger discrepancy observed for the dual configuration, particularly at 60% load. This finding is opposite to that from the 1D model, where the dual configuration slightly outperformed the single configuration at 60% load, underscoring the importance of the CFD modelling. The CFD model also predicted lower gas temperatures at 60% load, which resulted in lower heat uptakes with a smaller TTD across the convective heaters. For the radiative platen heater, sCO2H1, the CFD model predicted higher heat uptakes at 100% and 80%

loads, particularly due to a higher $\dot{Q}_{fe}$ compared to the 1D model. These findings highlight the necessity of the CFD models in this research.

To inform the 1D models based on the CFD results, a heat exchanger utilisation factor for the sCO2 heaters was introduced, accounting for dead spots within the sCO2 heaters. For the dual configuration, η_{FE} was also tuned to account for the higher $\dot{Q}_{fe}$ arising from boiler nose geometry adjustments. With the tuned 1D model, a decrease in maximum Brayton cycle temperatures was observed for both single and dual configurations, and consequently, a decrease in thermal efficiency. Slight degradations in cycle performance were observed for the 100% and 80% load cases with regards to Brayton cycle efficiency and net electrical power output for the single configuration. The biggest decreases were at 60% loads, which saw overall cycle thermal efficiency decrease from 25.92% to 25.43% for the single configuration, and 25.61% to 24.67% for the dual configuration.

The dual configuration was impacted significantly more after tuning the 1D model, particularly with the significantly lower heat uptakes observed in the CFD model. At 60% load, the low TTD across the convective heaters resulted in significantly lower heat uptakes. With the tuned 1D model, the lower heat uptakes resulted in lower internal sCO2 temperatures compared to those observed in the initial 1D model, nearly 205 °C lower for the dual configuration. This significant decrease in internal sCO2 temperature would indicate a much higher TTD compared to that currently observed for the CFD model. This highlights the importance of an iterative procedure for tuning the 1D model. As the CFD model inputs are derived from the 1D process model, major changes in inputs and boundary conditions, such as the internal sCO2 fluid temperature, would directly impact the CFD results. While this iterative procedure lies outside the scope of this study, the work highlights how the CFD models can be used to tune the 1D model and the importance of doing so.

Based on the single iteration performed in this work, the single heater configuration is recommended, particularly for its higher overall performance at lower loads. Moreover, the single heater configuration does not require adjustments to the boiler nose and only uses a single heat exchanger, reducing barriers to implementation. However, further iterations are strongly recommended between the CFD and 1D models to gain better insights into cycle performance with the adjusted tuning parameters. Even with the reductions in net generation and thermal efficiency observed after tuning the 1D model, the sCO2 Brayton cycle integration still highlights tangible improvements to overall thermal efficiency and net electrical power output. To increase Brayton cycle thermal efficiency, turbomachinery performance curves may need to be adjusted to decrease Brayton cycle capacity, which would allow for the nominal maximum cycle temperature of 550 °C to be reached. Another consideration for the 60% load case is to adjust the compressor inlet pressure to move away from the pseudo-critical point. This would reduce the impact on Brayton

cycle capacity at 60% load, however, at the cost of a lower maximum cycle temperature. It is important to note that increases in tube metal temperatures associated with overfiring of the boiler may adversely impact boiler lifetime. Further iterations between the 1D and CFD models would also provide a more accurate prediction of tube metal temperatures, particularly at lower loads. The effect of overfiring on boiler longevity in this study has been limited to tube metal temperatures and areas with high heat fluxes. Further investigations into the effect of overfiring on boiler longevity are recommended, taking into account boiler thermal capacity and throughput restrictions.

The outcomes from the CFD modelling and informing the 1D process models addressed the last objective of the study as proposed in section 1.4, namely the development of a 3D CFD model of the MicroGen® boiler to explore practical implications of the sCO2 heater integration and validating heat uptakes at the sCO2 heaters. Informing the 1D process model based on the results of the CFD model allows for better predictions of cycle performance, particularly for lower loads where a bigger discrepancy was observed. This sets up the 1D process model to be used for further sensitivity analyses for the single and dual heater configurations, as well as the implementation of different sCO2 Brayton cycle configurations.

By considering the above summaries and conclusions, all the research objectives outlined in section 1.4 are met, with recommendations for future work presented below.

8.4 Recommendations for future work

Based on the conclusions derived from this research, the following recommendations are made for further research and consideration:

- Adjustments to 60% load operation for the sCO2 Brayton cycle could be considered, such as higher compressor inlet pressures, to move away from the pseudo-critical point which would increase sCO2 compressor throughput and net electrical power output.
- For operation at 100%, the turbomachinery performance curves should be adjusted to ensure the nominal maximum Brayton cycle temperature of 550 °C is reached. This would ensure high thermal efficiencies at nominal loads.
- Further iterations between the CFD and 1D models are recommended to gain insight into more accurate heat uptakes at the sCO2 heaters, and the resulting effects on the integrated cycle performance, particularly at 60% load.
- Additional sensitivity analyses could be conducted using the tuned 1D model, considering boiler cleanliness, variations in fuel constituents and biomass fuel types, adjustments to excess air and PA/SA ratios, as well as process heat requirements.

- More complex sCO2 Brayton cycle configurations with higher thermal efficiencies could be explored, such as the recompression cycle, to investigate potential improvements to overall thermal efficiency.
- Lastly, an economic analysis on the proposed sCO2 Brayton cycle integration is beneficial to investigate the financial viability of the proposed cycle compared to traditional capacity uprate methods.

9. List of References

[1] World Bioenergy Association, “Global Bioenergy Statistics Report,” 2023.

[2] IEA, “Net Zero by 2050 - A Roadmap for the Global Energy Sector,” 2021.

[3] C. A. Bermúdez, J. Porteiro, L. G. Varela, S. Chapela, and D. Patiño, “Three-dimensional CFD simulation of a large-scale grate-fired biomass furnace,” *Fuel Process. Technol.*, vol. 198, no. October 2019, p. 106219, 2020, doi: 10.1016/j.fuproc.2019.106219.

[4] L. M. Mohlala, M. O. Bodunrin, A. A. Awosusi, M. O. Daramola, N. P. Cele, and P. A. Olubambi, “Beneficiation of corncob and sugarcane bagasse for energy generation and materials development in Nigeria and South Africa: A short overview,” *Alexandria Eng. J.*, vol. 55, no. 3, pp. 3025–3036, 2016, doi: 10.1016/j.aej.2016.05.014.

[5] The South African Sugar Association, “Sugar statistical information,” *SASA*, 2023. [Online]. Available: https://sasa.org.za/facts-and-figures/. [Accessed: 22-Feb-2024].

[6] IRENA, *Renewable Capacity Statistics 2024*. 2024.

[7] J. Islas, F. Manzini, O. Masera, and V. Vargas, “Solid Biomass to Heat and Power,” in *The Role of Bioenergy in the Bioeconomy*, C. Lago, N. Caldés, and Y. Lechón, Eds. Elsevier, 2019, pp. 145–177.

[8] R. L. Bain, R. P. Overend, and K. R. Craig, “Biomass-fired power generation,” *Fuel Process. Technol.*, vol. 54, no. 1–3, pp. 1–16, 1998, doi: 10.1016/S0378-3820(97)00058-1.

[9] A. Demirbas, “Potential applications of renewable energy sources, biomass combustion problems in boiler power systems and combustion related environmental issues,” *Prog. Energy Combust. Sci.*, vol. 31, no. 2, pp. 171–192, Jan. 2005, doi: 10.1016/j.pecs.2005.02.002.

[10] R. Laubscher and E. De Villiers, “Integrated mathematical modelling of a 105 t/h biomass fired industrial watertube boiler system with varying fuel moisture content,” *Energy*, vol. 228, p. 120537, Aug. 2021, doi: 10.1016/j.energy.2021.120537.

[11] L. Gustavsson, P. Börjesson, B. Johansson, and P. Svenningsson, “Reducing CO2 emissions by substituting biomass for fossil fuels,” *Energy*, vol. 20, no. 11, pp. 1097–1113, 1995, doi: 10.1016/0360-5442(95)00065-O.

[12] Y. S. Mohammed, A. S. Mokhtar, N. Bashir, and R. Saidur, “An overview of agricultural biomass for decentralized rural energy in Ghana,” *Renew. Sustain. Energy Rev.*, vol. 20, pp. 15–25, Apr. 2013, doi: 10.1016/J.RSER.2012.11.047.

[13] J. Coady and J. Duquette, “Quantifying the impacts of biomass driven combined heat and power grids in northern rural and remote communities,” *Renew. Sustain. Energy Rev.*, vol. 148, p. 111296, Sep. 2021, doi: 10.1016/J.RSER.2021.111296.

[14] IRENA, *Sugarcane bioenergy in southern Africa: Economic potential for sustainable scale-up*, no. May 2019. Abu Dhabi, 2019.

[15] S. Liedtke, “Nersa welcomes 100 MW embedded-generation licensing-exemption decision,” *Engineering News*, 11-Jun-2021.

[16] I. Venter, “City of Cape Town issues independent power tender,” *Engineering News*, 26-Jan-

2024. [Online]. Available: https://www.engineeringnews.co.za/article/city-of-cape-town-issues-independent-power-tender-2024-01-26. [Accessed: 22-Feb-2024].

[17] John Thompson, “MicroGen ® Biomass-/ Fossil Fuel-fired Boiler.”

[18] K. Brun, P. Friedman, and R. Dennis, *Fundamentals and Applications of Supercritical Carbon Dioxide (sCO2) Based Power Cycles*, 1st Ed. Woodhead Publishing, Elsevier, 2017.

[19] G. Angelino, “Carbon Dioxide Condensation Cycles For Power Production,” *J. Eng. Power*, vol. 90, no. 3, pp. 287–295, Jul. 1968, doi: 10.1115/1.3609190.

[20] E. G. Feher, “The supercritical thermodynamic power cycle,” *Energy Convers.*, vol. 8, no. 2, pp. 85–90, Sep. 1968, doi: 10.1016/0013-7480(68)90105-8.

[21] F. Crespi, G. Gavagnin, D. Sánchez, and G. S. Martínez, “Supercritical carbon dioxide cycles for power generation: A review,” *Appl. Energy*, vol. 195, pp. 152–183, Jun. 2017, doi: 10.1016/j.apenergy.2017.02.048.

[22] S. A. Wright, T. M. Conboy, R. F. Radel, and G. E. Rochau, “Modeling and experimental results for condensing supercritical CO2 power cycles.,” Albuquerque, NM, and Livermore, CA (United States), Jan. 2011.

[23] M. Kulhánek and V. Dostál, “Supercritical Carbon Dioxide Cycles Thermodynamic Analysis and Comparison,” *Proc. SCCO2 Power Cycle Symp.*, pp. 1–12, 2011.

[24] V. Dostal, P. Hejzlar, and M. J. Driscoll, “The supercritical carbon dioxide power cycle: Comparison to other advanced power cycles,” *Nucl. Technol.*, vol. 154, no. 3, pp. 283–301, 2006, doi: 10.13182/NT06-A3734.

[25] T. Neises and C. Turchi, “A comparison of supercritical carbon dioxide power cycle configurations with an emphasis on CSP applications,” *Energy Procedia*, vol. 49, pp. 1187–1196, 2014, doi: 10.1016/j.egypro.2014.03.128.

[26] R. Chacartegui, D. Sanchez, F. Jiménez-Espadafor, A. Muñoz, and T. Sanchez, “Analysis of Intermediate Temperature Combined Cycles With a Carbon Dioxide Topping Cycle,” *Proc. ASME Turbo Expo*, vol. 2, pp. 673–680, Aug. 2009, doi: 10.1115/GT2008-51053.

[27] S. M. Besarati and D. Y. Goswami, “Analysis of Advanced Supercritical Carbon Dioxide Power Cycles With a Bottoming Cycle for Concentrating Solar Power Applications,” *J. Sol. Energy Eng. Trans. ASME*, vol. 136, no. 1, Feb. 2014, doi: 10.1115/1.4025700/378746.

[28] X. Wang, J. Wang, P. Zhao, and Y. Dai, “Thermodynamic Comparison and Optimization of Supercritical CO2 Brayton Cycles with a Bottoming Transcritical CO2 Cycle,” *J. Energy Eng.*, vol. 142, no. 3, p. 04015028, Jul. 2015, doi: 10.1061/(ASCE)EY.1943-7897.0000292.

[29] J. Marion, M. Kutin, A. McClung, J. Mortzheim, and R. Ames, “The STEP 10 MWe sCO2 Pilot Plant Demonstration,” in *Volume 9: Oil and Gas Applications; Supercritical CO2 Power Cycles; Wind Energy*, 2019, vol. 9, doi: 10.1115/GT2019-91917.

[30] J. I. Linares, A. Cantizano, B. Y. Moratilla, V. Martín-Palacios, and L. Batet, “Supercritical CO2 Brayton power cycles for DEMO (demonstration power plant) fusion reactor based on dual coolant lithium lead blanket,” *Energy*, vol. 98, pp. 271–283, Mar. 2016, doi: 10.1016/J.ENERGY.2016.01.020.

[31] M. L. Bauer, R. Vijaykumar, M. Lausten, and J. Stekli, “Pathways to Cost Competitive

Concentrated Solar Power Incorporating Supercritical Carbon Dioxide Power Cycles," *5th Int. Supercrit. CO2 Power Cycles Symp.*, pp. 1–22, Mar. 2016.

[32] S. A. Wright, C. S. Davidson, and W. O. Scammell, "Thermo-Economic Analysis of Four sCO2 Waste Heat Recovery Power Systems," in *The 5th Supercritical CO2 Power Cycles Symposium*, 2016.

[33] J. Sarkar and S. Bhattacharyya, "Optimization of recompression S-CO2 power cycle with reheating," *Energy Convers. Manag.*, vol. 50, no. 8, pp. 1939–1945, Aug. 2009, doi: 10.1016/j.enconman.2009.04.015.

[34] C. S. Turchi, Z. Ma, T. W. Neises, and M. J. Wagner, "Thermodynamic Study of Advanced Supercritical Carbon Dioxide Power Cycles for Concentrating Solar Power Systems," *J. Sol. Energy Eng.*, vol. 135, no. 4, pp. 1–7, Nov. 2013, doi: 10.1115/1.4024030.

[35] V. Dostal, M. J. Driscoll, and P. Hejzlar, "A Supercritical Carbon Dioxide Cycle for Next Generation Nuclear Reactors," Massachusetts Institute of Technology, 2004.

[36] J. Sarkar, "Second law analysis of supercritical CO2 recompression Brayton cycle," *Energy*, vol. 34, no. 9, pp. 1172–1178, Sep. 2009, doi: 10.1016/j.energy.2009.04.030.

[37] L. Moroz, M. Burlaka, O. Rudenko, and C. Joly, "Evaluation of Gas Turbine Exhaust Heat Recovery Utilizing Composite Supercritical CO2 Cycle," in *Proceedings of International Gas Turbine Congress 2015*, 2015, no. November, p. 7.

[38] Y. Ji-chao and B. Sobhani, "Integration of biomass gasification with a supercritical CO2 and Kalina cycles in a combined heating and power system: A thermodynamic and exergoeconomic analysis," *Energy*, vol. 222, p. 119980, May 2021, doi: 10.1016/j.energy.2021.119980.

[39] S. Balafkandeh, V. Zare, and E. Gholamian, "Multi-objective optimization of a tri-generation system based on biomass gasification/digestion combined with S-CO2 cycle and absorption chiller," *Energy Convers. Manag.*, vol. 200, p. 112057, Nov. 2019, doi: 10.1016/j.enconman.2019.112057.

[40] A. D. Akbari and S. M. S. Mahmoudi, "Thermoeconomic analysis & optimization of the combined supercritical CO2 (carbon dioxide) recompression Brayton/organic Rankine cycle," *Energy*, vol. 78, pp. 501–512, 2014, doi: 10.1016/j.energy.2014.10.037.

[41] X. Wang *et al.*, "Review of dynamic performance and control strategy of supercritical CO2 Brayton cycle," *Energy AI*, vol. 5, no. January, p. 100078, 2021, doi: 10.1016/j.egyai.2021.100078.

[42] A. Heifetz and R. Vilim, "Turbine bypass, mass inventory, and mixed-mode generator power control of S-CO2 recompression cycle," *Nucl. Technol.*, vol. 189, no. 3, pp. 268–277, 2015, doi: 10.13182/NT13-113.

[43] R. Wang *et al.*, "Comparison of different load-following control strategies of a sCO2 Brayton cycle under full load range," *Energy*, vol. 246, p. 123378, 2022, doi: 10.1016/j.energy.2022.123378.

[44] G. Fan, H. Li, Y. Du, K. Chen, S. Zheng, and Y. Dai, "Preliminary design and part-load performance analysis of a recompression supercritical carbon dioxide cycle combined with a transcritical carbon dioxide cycle," *Energy Convers. Manag.*, vol. 212, no. 28, p. 112758, 2020,

doi: 10.1016/j.enconman.2020.112758.

[45] G. Fan and Y. Dai, “Thermo-economic optimization and part-load analysis of the combined supercritical CO2 and Kalina cycle,” *Energy Convers. Manag.*, vol. 245, no. 28, p. 114572, 2021, doi: 10.1016/j.enconman.2021.114572.

[46] F. Colombo, “Preliminary design and part load analysis of supercritical carbon dioxide power cycles for gas turbine bottoming applications,” Politecnico Milano, 2022.

[47] F. Alobaid, N. Mertens, R. Starkloff, T. Lanz, C. Heinze, and B. Epple, “Progress in dynamic simulation of thermal power plants,” *Prog. Energy Combust. Sci.*, vol. 59, pp. 79–162, Mar. 2017, doi: 10.1016/j.pecs.2016.11.001.

[48] J. Kuronen, M. Hotti, and S. Tuuri, “Modelling and Dynamic Simulation of Cyclically Operated Pulverized Coal-Fired Power Plant,” *Proc. 9th EUROSIM Congr. Model. Simulation, EUROSIM 2016, 57th SIMS Conf. Simul. Model. SIMS 2016*, vol. 142, pp. 122–128, 2018, doi: 10.3384/ecp17142122.

[49] W. Zima *et al.*, “Thermal calculations of a natural circulation power boiler operating under a wide range of loads,” *Energy*, vol. 261, no. August, 2022, doi: 10.1016/j.energy.2022.125357.

[50] R. Starkloff, F. Alobaid, K. Karner, B. Epple, M. Schmitz, and F. Boehm, “Development and validation of a dynamic simulation model for a large coal-fired power plant,” *Appl. Therm. Eng.*, vol. 91, pp. 496–506, 2015, doi: 10.1016/j.applthermaleng.2015.08.015.

[51] O. K. Singh, “Application of Kalina cycle for augmenting performance of bagasse-fired cogeneration plant of sugar industry,” *Fuel*, vol. 267, no. December 2019, p. 117176, 2020, doi: 10.1016/j.fuel.2020.117176.

[52] E. J. C. Cavalcanti, M. Carvalho, and D. R. S. da Silva, “Energy, exergy and exergoenvironmental analyses of a sugarcane bagasse power cogeneration system,” *Energy Convers. Manag.*, vol. 222, no. May, p. 113232, 2020, doi: 10.1016/j.enconman.2020.113232.

[53] S. Chantasiriwan, “Optimum installation of economizer, air heater, and flue gas dryer in biomass boiler,” *Comput. Chem. Eng.*, vol. 150, p. 107328, Jul. 2021, doi: 10.1016/j.compchemeng.2021.107328.

[54] P. Rousseau, R. Laubscher, and B. T. Rawlins, “Heat Transfer Analysis Using Thermofluid Network Models for Industrial Biomass and Utility Scale Coal-Fired Boilers,” *Energies*, vol. 16, no. 4, p. 1741, Feb. 2023, doi: 10.3390/en16041741.

[55] A. Zhou, H. Xu, Y. Tu, F. Zhao, Z. Zheng, and W. Yang, “Numerical investigation of the effect of air supply and oxygen enrichment on the biomass combustion in the grate boiler,” *Appl. Therm. Eng.*, vol. 156, pp. 550–561, Jun. 2019, doi: 10.1016/j.applthermaleng.2019.04.053.

[56] F. O. Centeno-González *et al.*, “CFD modeling of combustion of sugarcane bagasse in an industrial boiler,” *Fuel*, vol. 193, pp. 31–38, Apr. 2017, doi: 10.1016/j.fuel.2016.11.105.

[57] R. Laubscher and S. van der Merwe, “Heat transfer modelling of semi-suspension biomass fired industrial watertube boiler at full- and part-load using CFD,” *Therm. Sci. Eng. Prog.*, vol. 25, p. 100969, Oct. 2021, doi: 10.1016/j.tsep.2021.100969.

[58] G. Manente and A. Lazzaretto, “Innovative biomass to power conversion systems based on cascaded supercritical CO2 Brayton cycles,” *Biomass and Bioenergy*, vol. 69, pp. 155–168, Oct. 2014, doi: 10.1016/j.biombioe.2014.07.016.

[59] B. Mutlu, D. Baker, and F. Kazanç, "Development and Analysis of the Novel Hybridization of a Single-Flash Geothermal Power Plant with Biomass Driven sCO2-Steam Rankine Combined Cycle," *Entropy*, vol. 23, no. 6, p. 766, Jun. 2021, doi: 10.3390/e23060766.

[60] T. Zignani, M. Binotti, M. Astolfi, and D. Alfani, "Feasibility study on the application of Supercritical Carbon Dioxide cycles to biomass-fired power plants," MSc. Dissertation, Politecnico di Milano, Milan, Italy, 2020.

[61] C. Kotze, "The Effects of Bagasse Moisture Variations on the Performance of a 105 Tph Boiler," *Proc. South-African Sugar Technol. Assoc.*, pp. 309–319, 2019.

[62] S. C. Stultz and J. B. Kitto, *Steam: its generation and use*, 41st ed. Babcock & Wilcox, 2005.

[63] G. Golański, A. Zieliński, and M. Sroka, "Microstructure and mechanical properties of TP347HFG austenitic stainless steel after long-term service," *Int. J. Press. Vessel. Pip.*, vol. 188, no. July 2018, 2020, doi: 10.1016/j.ijpvp.2020.104160.

[64] ASTM, *ASTM A192/A192M-02 Standard Specification for Seamless Carbon Steel Boiler Tubes for High-Pressure Service 1*. 2017.

[65] 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," *Ind. Eng. Chem. Res.*, vol. 53, no. 6, pp. 2498–2508, Feb. 2014, doi: 10.1021/ie4033999.

[66] A. G. Blokh, "Heat transfer in steam boiler furnaces."

[67] Y. Zhang, Q. Li, and H. Zhou, *Theory and Calculation of Heat Transfer in Furnaces*. Elsevier, 2016.

[68] H.-G. Brummel, "K4 Thermal Radiation of Gas–Solids–Dispersions," 2010, pp. 989–1000.

[69] R. Laubscher and P. Rousseau, "An enhanced model of particle radiation properties in high ash gas-particle dispersion flow through industrial gas-to-steam heat exchangers," *Fuel*, vol. 285, no. April 2020, p. 119153, 2021, doi: 10.1016/j.fuel.2020.119153.

[70] Z. Lin, "Thermohydraulic Design of Fossil-Fuel-Fired Boiler Components," in *Boilers, evaporators, and condensers*, S. Kakac, Ed. Nashville, TN: John Wiley \& Sons, 1991.

[71] V. Gnielinski, "Heat transfer in cross-flow around single rows of tubes and through tube bundles," in *VDI Heat Atlas*, Berlin: Springer-Verlag, 2016, pp. 756–760.

[72] Y. A. Cengel, *Heat transfer: A practical approach*. New York, NY: McGraw-Hill, 1997.

[73] R. Deshmukh, A. Jacobson, C. Chamberlin, and D. Kammen, "Thermal gasification or direct combustion? Comparison of advanced cogeneration systems in the sugarcane industry," *Biomass and Bioenergy*, vol. 55, pp. 163–174, Aug. 2013, doi: 10.1016/j.biombioe.2013.01.033.

[74] J. Servert, G. San Miguel, and D. López, "Hybrid solar - Biomass plants for power generation; technical and economic assessment," *Glob. Nest J.*, vol. 13, no. 3, pp. 266–276, 2011, doi: 10.30955/gnj.000696.

[75] J. Lee, J. I. Lee, H. J. Yoon, and J. E. Cha, "Supercritical Carbon Dioxide turbomachinery design for water-cooled Small Modular Reactor application," *Nucl. Eng. Des.*, vol. 270, pp. 76–89, Apr. 2014, doi: 10.1016/j.nucengdes.2013.12.039.

[76] Y. Zhu *et al.*, "Experimental Study on a Supercritical CO2 Centrifugal Compressor Used in a MWe Scale Power Cycle," *Appl. Sci.*, vol. 13, no. 1, 2023, doi: 10.3390/app13010385.

[77] SciPy, "scipy.optimize.newton — SciPy v1.12.0 Manual," *SciPy v1.12.0 Manual*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.newton.html. [Accessed: 28-Feb-2024].

[78] M. B. Allen and E. L. Isaacson, *Numerical Analysis for Applied Science*. Wiley, 2019.

[79] A. Ruozzi, E. Vainio, H. Kinnunen, and L. Hupa, "Cold-end corrosion in biomass combustion – Role of calcium chloride in the deposit," *Fuel*, vol. 349, no. December 2022, p. 128344, 2023, doi: 10.1016/j.fuel.2023.128344.

[80] SASTA, *Laboratory manual for South African Sugar Factories*. SASTA, 2005.

[81] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, H. Coleman, and P. E. Raad, "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications," *J. Fluids Eng.*, vol. 130, no. 7, p. 078001, 2008, doi: 10.1115/1.2960953.

[82] "Ansys Fluent Theory Guide." Ansys Inc, 2023.

[83] B. T. Rawlins, "An integrated furnace co-simulation methodology based on a reduced order CFD approach," University of Cape Town, South Africa, 2022.

[84] C. Yin, L. Rosendahl, and S. K. Kær, "Towards a better understanding of biomass suspension co-firing impacts via investigating a coal flame and a biomass flame in a swirl-stabilized burner flow reactor under same conditions," *Fuel Process. Technol.*, vol. 98, pp. 65–73, Jun. 2012, doi: 10.1016/J.FUPROC.2012.01.024.

[85] T. Poinsot and D. Veynante, "Theoretical and Numerical Combustion," *Prog. Energy Combust. Sci.*, vol. 28, 2005.

[86] B. H. W. Bernhardt, "Shape factors of bagasse particles," *Milling*, no. June, 1993.

[87] A. D. Gosman and E. Loannides, "Aspects of Computer Simulation of Liquid-Fueled Combustors," *J. Energy*, vol. 7, no. 6, pp. 482–490, Nov. 1983, doi: 10.2514/3.62687.

[88] A. Haider and O. Levenspiel, "Drag coefficient and terminal velocity of spherical and nonspherical particles," *Powder Technol.*, vol. 58, no. 1, pp. 63–70, May 1989, doi: 10.1016/0032-5910(89)80008-7.

[89] D. Gera, M. P. Mathur, M. C. Freeman, and A. L. Robinson, "A numerical study of the effects of biomass coal cofiring on unburned carbon and NOx emissions," in *American ame reserach committee (AFRC) fall symposium*, 1999.

[90] R. S. Miller, K. Harstad, and J. Bellan, "Evaluation of equilibrium and non-equilibrium evaporation models for many-droplet gas-liquid flow simulations," *Int. J. Multiph. Flow*, vol. 24, no. 6, pp. 1025–1055, Sep. 1998, doi: 10.1016/S0301-9322(98)00028-7.

[91] S. S. Sazhin, "Advanced models of fuel droplet heating and evaporation," *Prog. Energy Combust. Sci.*, vol. 32, no. 2, pp. 162–214, Jan. 2006, doi: 10.1016/J.PECS.2005.11.001.

[92] P. C. Du Toit, "An artificial intelligence approach for biomass devolatilisation in an industrial CFD model with advanced turbulence-chemistry interaction," Stellenbosch University, 2018.

[93] M. Farokhi and M. Birouk, "Application of Eddy Dissipation Concept for Modeling Biomass Combustion, Part 2: Gas-Phase Combustion Modeling of a Small-Scale Fixed Bed Furnace," *Energy and Fuels*, vol. 30, no. 12, pp. 10800–10808, Dec. 2016, doi: 10.1021/ACS.ENERGYFUELS.6B01948/ASSET/IMAGES/LARGE/EF-2016-01948W_0008.JPEG.

[94] C. Sheng and J. L. T. Azevedo, "Modeling biomass devolatilization using the chemical percolation devolatilization model for the main components," *Proc. Combust. Inst.*, vol. 29, no. 1, pp. 407–414, Jan. 2002, doi: 10.1016/S1540-7489(02)80054-2.

[95] P. C. du Toit, "The development and validation of a computational fluid dynamics model of an industrial watertube boiler burning bagasse," no. March, 2015.

[96] M. M. Baum and P. J. Street, "Predicting the Combustion Behaviour of Coal Particles," *Combust. Sci. Technol.*, vol. 3, no. 5, pp. 231–243, 1971, doi: 10.1080/00102207108952290.

[97] R. K. Boyd and J. H. Kent, "Three-dimensional furnace computer modelling," *Symp. Combust.*, vol. 21, no. 1, pp. 265–274, Jan. 1988, doi: 10.1016/S0082-0784(88)80254-6.

[98] M. A. Boles and D. Yunus A. Cengel, *Thermodynamics: An Engineering Approach*. McGraw-Hill Education, 2014.

[99] G. Sankar, D. S. Kumar, and K. R. Balasubramanian, "Computational modeling of pulverized coal fired boilers – A review on the current position," *Fuel*, vol. 236, pp. 643–665, Jan. 2019, doi: 10.1016/J.FUEL.2018.08.154.

[100] R. Laubscher and P. Rousseau, "Numerical investigation into the effect of burner swirl direction on furnace and superheater heat absorption for a 620 MWe opposing wall-fired pulverized coal boiler," *Int. J. Heat Mass Transf.*, vol. 137, pp. 506–522, Jul. 2019, doi: 10.1016/J.IJHEATMASSTRANSFER.2019.03.150.

[101] B. F. Magnussen and B. H. Hjertager, "On mathematical modeling of turbulent combustion with special emphasis on soot formation and combustion," *Symp. Combust.*, vol. 16, no. 1, pp. 719–729, Jan. 1977, doi: 10.1016/S0082-0784(77)80366-4.

[102] M. F. Modest, *Radiative Heat Transfer, Third Edition*. 2013.

[103] R. I. Backreedy, L. M. Fletcher, L. Ma, M. Pourkashanian, and A. Williams, "Modelling pulverised coal combustion using a detailed coal combustion model," *Combust. Sci. Technol.*, vol. 178, no. 4, pp. 763–787, Apr. 2006, doi: 10.1080/00102200500248532.

[104] P. Honig, *Principles of sugar technology*. Elsevier, 2013.

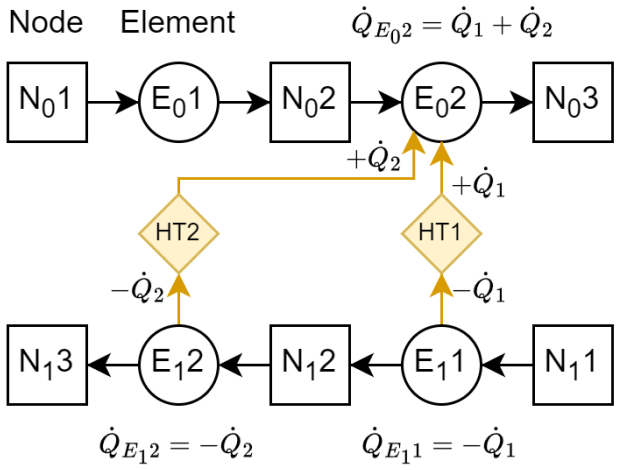
[105] AISI, *High-temperature characteristics of stainless steels*, vol. 9004. Nickel Development Institute, 2015.

Appendix A. Integrated 1D model – Python code listing

This appendix details the Python code developed for the integrated 1D process model. A brief structure of the codebase is presented, as well as a code listing in subsequent sections. Due to the size of the codebase, only snippets of the codebase are presented to provide the reader with an understanding of how the model works.

The 1D thermofluid network solver used in this study comprises of four fundamental modules/classes. This includes the node class, element class, heat transfer element class, and lastly the solver code module, '*solver.py*'. These classes form the fundamental building blocks to create thermofluid networks used by this program. The node and element methodology presented in section 3.2 is implemented using these classes. Heat transfer elements allow for heat transfer between elements, including those in two different thermofluid networks.

At each node, thermofluid properties are updated every iteration. For specified boundary nodes, boundary conditions remain fixed. For momentum balance, pressures must be specified at beginning and end nodes. Alternatively, a combination of mass source/sink and pressure can also be provided as a momentum boundary condition. For energy balance, enthalpies must be provided at boundary nodes. For closed cycles, a floating enthalpy boundary is assumed for the first node, which is updated in the following iteration using the last node's enthalpy value. For non-boundary nodes, guess values must be specified to allow for the solver to proceed.



Appendix Figure 1: Heat transfer element example

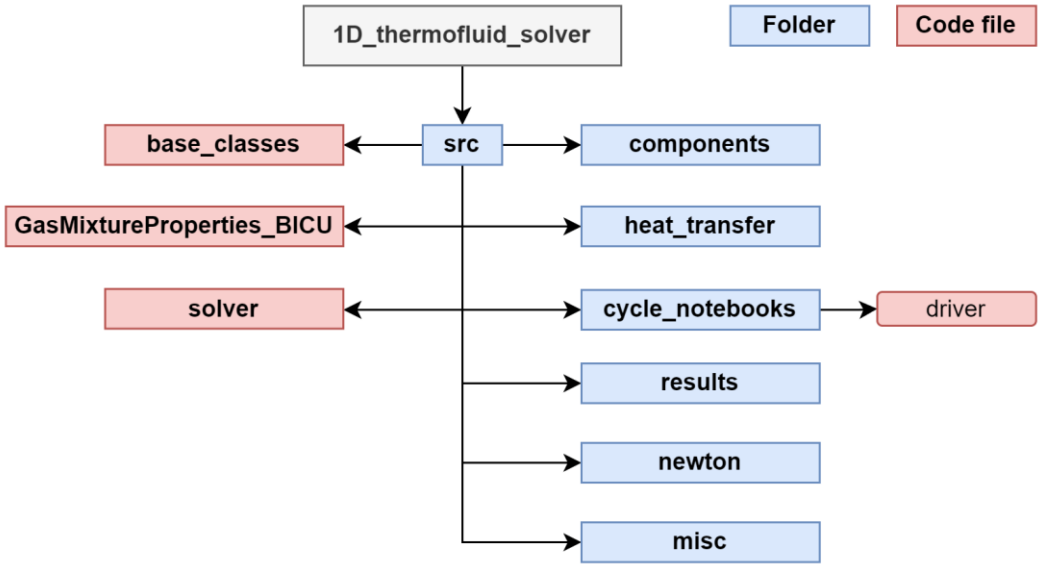
At elements, linearised component characteristics used in the momentum balance matrix equations are calculated each iteration. Additionally, average thermofluid properties are also computed using the averaged value of the two nodes the element is connected between. Each element also has a $\dot{W}$ and $\dot{Q}$ term. $\dot{W}$, such as with turbomachinery, is calculated in the element object itself. On the other hand, $\dot{Q}$ is calculated using heat transfer elements. Heat transfer elements can be connected between two elements. For each steady state iteration, the heat transfer element is used to solve the required heat transfer phenomena and calculates the $\dot{Q}$ term that the respective heat transfer element represents. Each element can be connected to multiple heat transfer elements, representing different heat transfer mediums. Consequently, heat transfer elements add or subtract the calculated $\dot{Q}$ from the respective element as defined by the user. Thereafter, the total $\dot{Q}$ for that respective element is used in the energy balance matrix equations. An example is provided in Appendix Figure 1. Two heat transfer elements are connected to $E_0 2$, one from $E_1 1$ and one from $E_1 2$, where the subscript refers to a specific thermofluid network. The total $\dot{Q}$ term at $E_0 2$ is therefore made up of the summed contributions from HT1 and HT2, $\dot{Q}_1$ and $\dot{Q}_2$.

The element class is a fundamental class in the thermofluid network solver program. Each component, such as a pipe, turbine, compressor, etc. inherits the element class. Pipe components are also a core element. Heat exchangers are represented by pipe components. Heat transfer elements are then used to connect the respective pipes to facilitate heat transfer. For fluegas streams, a component called *'fluegas pipe'* is developed, inheriting the '*pipe'* component but also accounting for fly-ash streams and potential ingress air. Compound components are also used, such as with discretised heat exchangers, or to represent a large component made up of smaller components, such as a boiler. Compound components are also setup as classes, allowing for flexibility and the ability to create multiple instances of a compound component if required. A compound component must be initialised, which creates all the required nodes and elements in the compound component, or sub-compound components.

Lastly, the solver code and function are contained in '*solver.py*'. The solver function takes the connectivity matrices provided by the user, in addition to the list of nodes, elements, and heat transfer elements. User-defined solver parameters are also available, such as convergence criteria, maximum iteration count, verbosity, and more. The solver function, when called, then goes through the solver process as shown in Figure 3.9, involving matrix inversion and the solution of the momentum and energy matrix balance equations.

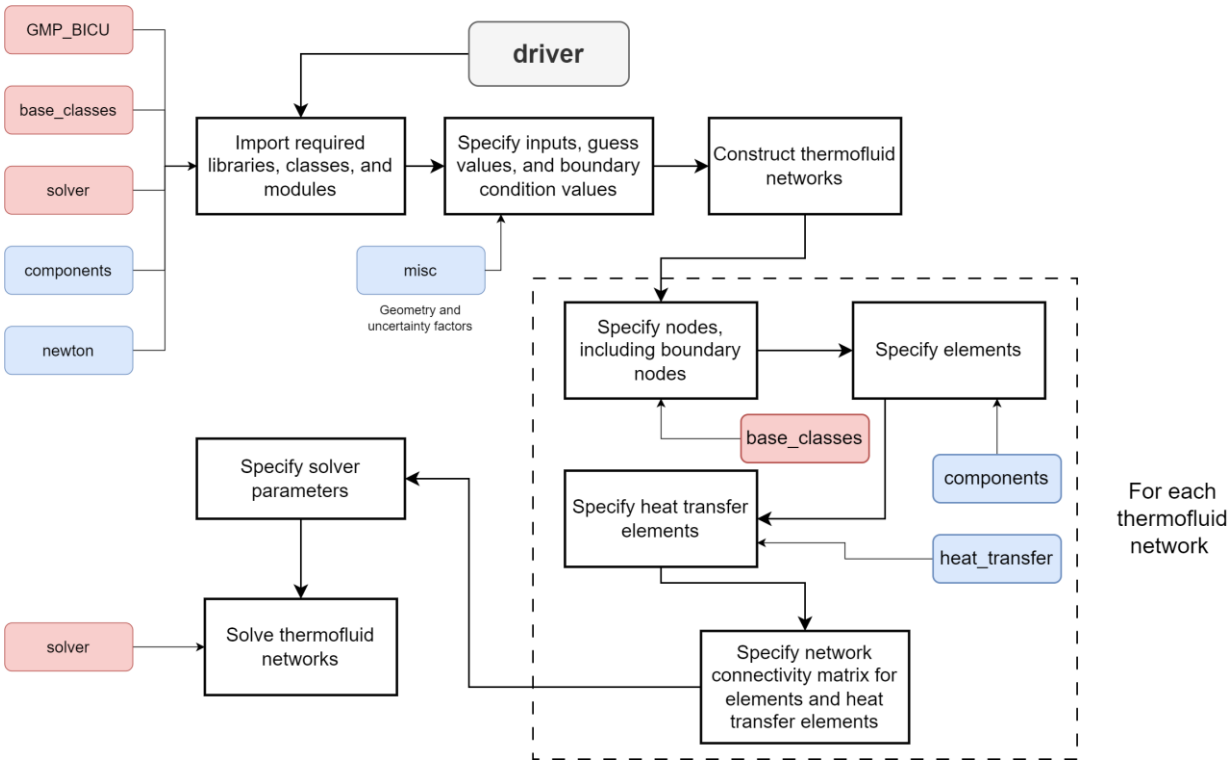
Appendix Figure 2: Codebase structure shows the structure for the codebase. Blue blocks represent folders, while red blocks represent code and other files. The driver files are written as Jupyter notebooks using VS Code. A brief description of the various folders and files are provided. The

'components' folder contains various .py code for the different components/elements used in the study, such as turbines, pumps, compressors, etc. The classes for each component are contained in this folder and are imported as required in the driver folder. As the boiler has a lot of components, a separate boiler class is used, also found in the '*components*' folder. The boiler class is then called in the driver file to initialise all the boiler sub-compound components, nodes, and elements. Similarly, the '*heat transfer*' folder contains various classes for the heat transfer elements, including detailed heat transfer calculations used for heat exchangers. '*Cycle_notebooks*' contains several driver files (as Jupyter .ipynb notebooks) for the different configurations. The '*results*' folder contains .xlsx files as results from the driver files. In '*newton*', a modified version of the Newton Raphson solver from the SciPy library is present, modified to account for additional relaxation parameters between iterations. Lastly, the '*misc*' folder contains several .xlsx input files for the various configurations, including geometry, uncertainty factors, and tuning coefficients.

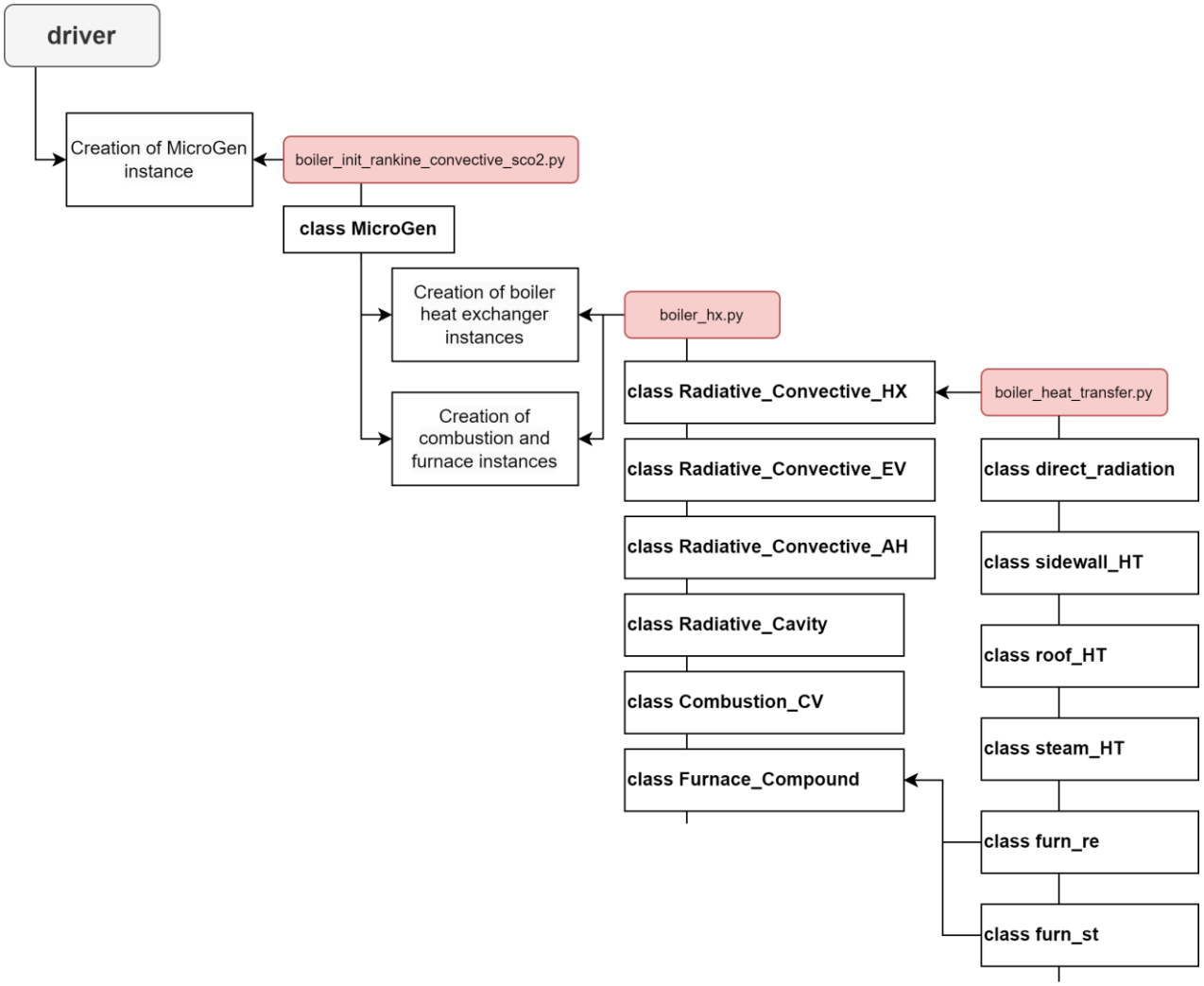


Appendix Figure 2: Codebase structure

The '*base_classes*' .py file contains the core classes for the codebase, including node, element, and heat transfer classes. For elements, components in the 'component' folder inherit the element class, and similarly for the heat transfer classes. '*GasMixtureProperties_BICU*'.py, referred to as GMP, is used to calculate fluid properties for gas mixtures and has been further modified to leverage the bicubic interpolation method by CoolProp for significant speed up. Lastly, the most important code in the codebase, '*solver*'.py contains the code which solves the conservation equations based on the created thermofluid networks from the driver file.



Appendix Figure 3: Generic driver file code process flow shows the process flow diagram for a generic driver file. The driver file begins by importing required external libraries, as well as internal modules. The internal modules include base classes for nodes, GMP if required, the solver, as well as components and elements as required. Next, inputs are specified, which can be imported through .xlsx files using Pandas. The required thermofluid networks are then constructed using the node class, component/element classes, and heat transfer element classes. The connectivity matrix contains which elements are connected to which nodes, and similarly which heat transfer elements are connected to which elements, as defined by the user through a 'connect_IO' (input/output) function. The connectivity matrix is automatically created by the solver. This process is repeated for each thermofluid network required. Lastly, solver parameters are specified, including relaxation and convergence values, as well as iteration maximums and more. The solver function is then called in the driver file, and the steady-state solver commences till convergence or the maximum iteration count is reached.



Appendix Figure 4: Class instance creation for boiler class in driver program

For the current work, a boiler class called '*MicroGen*' is created for each configuration analysed. As the boiler contains a significant number of components, a class for the boiler is useful to localise code for boiler compound component creation and keep the driver file clutter free. Appendix Figure 4: Class instance creation for boiler class in driver program shows how the boiler class is implemented along with other sub- compound component classes. In the '*MicroGen*' class, instances of several other compound component classes are created, including the combustion, furnace, and various boiler heat exchanger classes. The code for these compound component classes is presented below. Appendix A.2 shows the code listing for the '*MicroGen*' class, Appendix A.3 shows the code listing for the various boiler heat exchanger classes, and lastly Appendix A.4 shows the code listing for the boiler heat transfer element classes. There are four boiler heat exchanger compound component classes, including a radiative convective heat exchanger, radiative

convective evaporator, radiative cavity, and radiative convective air heater. Additionally, classes are used for the combustion control volume, as well as the furnace. For each of these boiler heat exchanger compound components, pipe elements for each fluid stream, including the flue gas and secondary streams, are created (except for radiative cavities). Moreover, instances of heat transfer element classes are also created in each compound component, accounting for various phenomena, such as direct radiation, external heat transfer to the secondary fluid through convection and gas radiation, as well as heat transfer to water walls in the roof and side walls. The heat transfer element classes in '*boiler_heat_transfer*'.py is used to calculate the required $\dot{Q}$ for each heat transfer element. This includes various heat transfer correlations, implemented as functions, and described in the theory section 3.2. Consequently, for each boiler heat exchanger instance (other than radiative cavities), two pipes and four heat transfer elements are created. Each of these heat transfer elements is then connected to the relevant input and output pipes to facilitate the required heat transfer phenomena.

It is important to note that only the boiler class is described here-in. Elements such as turbines, compressors, pipes, as well as other compound heat exchangers used in this work are not presented. The reasoning is to provide the reader with a brief understanding of the inner workings of the integrated network model, rather than provide a complete code listing and in-depth review of the solver. The driver code listing is presented below in Appendix A.1, with several miscellaneous functions omitted for clarity.

A.1 Driver file code - single sCO2 heater configuration

```
'''Driver file for single configuration integrated cycle'''

#import required libraries and modules
import numpy as np
import sympy
import CoolProp.CoolProp as CP
import pandas as pd
from newton.zeros import * #adjusted newton to include relaxation

#import required modules for solver, elements, and heat transfer elements
from solver import *
from base_classes import Node
from heat_transfer.heat_transfer import *
from components.compressor import Compressor_isentropic_head_v2
from components.turbine import Turbine
from components.pipe import Pipe
from components.compound_hx import *
from components.pump import Pump
from components.two_phase_tank import Two_Phase_Tank
from components.compound_hx import Compound_HX_Counterflow

#import required boiler element modules
from components.boiler.combustion_cv import Combustion
import components.boiler.furnace_projected as fp
from components.boiler.boiler_init_rankine_convective_sco2 import MicroGen

import GasMixtureProperties_BICU as gmp

##############INPUTS##############
inputs = pd.read_excel(('results//brayton//conv_invcontrol_100%_load_10%ATT_CFD_compare.xlsx')\
                       , nrows = 20, usecols = 'A:B')
load = 100
p_cond_out = inputs.loc[0,0]*1E3
K_CFP_pipe = inputs.loc[1,0]
K_HPT_bypass = inputs.loc[2,0]
Qdot_ATT = inputs.loc[3,0]*1E3
K_BFP_pipe = inputs.loc[4,0]
#Turb valve value based on fitted data for various loads
K_turb_valve = 9.87437E+04 + -2.31872E+05 * load/100 + 1.83143E+05\
        * (load/100)**2 + -5.00000E+04 * (load/100)**3
K_process =  inputs.loc[6,0]
K_process_2 = 4.0
K_cond_pipe = 1.5
T_ext_cond = 25 + 273.15
Qdot_process = 2000E3
bool_process_heat = True
mdot_f = inputs.loc[11,0]
K_ATT_co2 = inputs.loc[12,0]
K_RXLP_valve = inputs.loc[13,0]
K_PC_bypass = inputs.loc[14,0]
p_comp_in = (load - 60)/(100 - 60) * (10E6 - 7.5E6) + 7.5E6
UA_RX =  inputs.loc[16,0]
UA_cooler = inputs.loc[17,0]
UA_cond = inputs.loc[18,0]
##################################

#Ambient inputs
H = 250 #[m]
p_amb = barometric_pressure(H)
T_amb = 25+kel
RH_amb = 0.8
w_amb = CP.HAPropsSI('W','T',T_amb,'RH',RH_amb,'P',p_amb)
p_fur = p_amb
T_f = 27+kel
T_ca = T_amb
w_ca = w_amb
#Mass fractions
Y_amb = {'O2': 0.22923, 'N2': 0.75450, 'H2O': 0.01626}
Y_fg = {'H2O': 0.15, 'CO2': 0.15, 'N2': 0.5, 'SO2': 9.0e-05, 'O2': 0.030}
```

```
70   T_ia = T_ca
71   f_ia = 0
72   alpha_air = 1.27
73   HHV = 8.838E6
74   cp_f = 1200
75   Y_fa = 0.40
76   T_ba = 800+kel
77     Y_f  =  {'C':0.2171-0.0062,  'H':0.0268,  'O':0.2077,  'N':0.0016,  'S':0.0002,  'H2O':0.50,  'ASH':0.0466,  'UC':0.0062}
#[kg/kg_f]
78
79   f_pa = 0.508
80   f_sa = 0.402
81   f_da = 0.090
82   T_pa = 240+kel
83   T_sa = T_pa
84   T_da = 32+kel
85
86   #Uncertainty factors - for uncertainty analysis
87   alpha_angular_coef = UFx['fur']['alpha_angular_coef']
88   alpha_flame_modification = UFx['fur']['alpha_angular_coef']
89
90   #Guess
91   T_aft = 1360
92
93   h_pa = gmp.enthalpy(Y_amb, T=T_pa)
94   h_aft = gmp.enthalpy(Y_fg, T=T_aft)
95
96   #Furnace Inputs
97   f_radloss = 0.0101
98   # Combustion chamber geometry
99   d_tubewalls = 0.0762    #[m]
100  b_tubewalls = 0.1       #[m]
101  A_fur = {}
102  A_fur['tubewalls'] = 150.3832 #152.082       #[m^2]
103  A_fur['refractory'] = 26.704      #[m^2]
104  A_fur['furnaceexit'] = 19.814     #[m^2]
105  A_fur['grate'] = 20.758           #[m^2]
106  Vol_fur =  184.09                #[m^3]
107
108  # Wall fouling factor - xi
109  xi_fur = {}
110  xi_fur['tubewalls'] = fp.wall_fouling_factor['grate_fuels']
111  xi_fur['refractory'] = fp.wall_fouling_factor['refractory']
112  xi_fur['furnaceexit'] = 1.0
113  xi_fur['grate'] = 1.0
114
115  # Wall angular coefficient - x
116  x_fur = fp.angular_coefficient(b_tubewalls, d_tubewalls, alpha_angular_coef)
117
118  #Effective furnace emissivity
119  d_fa = fp.d_fa['grate_firing']
120  x1_coke = fp.x_1_coke['other_coals']
121  x2_coke = fp.x_2_coke['grate_firing']
122
123  #Flame modification factor
124  X_fl = 0.14 #nominal
125  M_flame = (fp.A_prime['grate_stoker'] - fp.B_prime['grate_stoker']*X_fl) * (1 + alpha_flame_modification)
126
127  #CoolProp using BICU interpolation for speed up
128  BICU_water = CP.AbstractState('BICUBIC&HEOS', 'Water')
129
130  #Guess values for Rankine cycle
131  h_cond_out = CP.PropsSI('H', 'Q', 0, 'P', p_cond_out, 'Water')
132  p_CFP_out = 880E3
133  h_CFP_out = CP.PropsSI('H', 'Q', 0, 'P', p_CFP_out, 'Water')
134  p_BFP_out = 6800E3
135  h_BFP_out = CP.PropsSI('H', 'Q', 0, 'P', p_BFP_out, 'Water')
136  p_boiler_out = 6790E3
137  T_boiler_out = 485 + 273.15
138  h_boiler_out = CP.PropsSI('H', 'T', T_boiler_out, 'P', p_boiler_out, 'Water')
139  T_HPT_out = 320 + 273.15
140  p_HPT_out = 860E3
```

```
h_HPT_out = CP.PropsSI('H', 'T', T_HPT_out, 'P', p_HPT_out, 'Water')
p_IPT_out = 1.05*p_cond_out
h_IPT_out = CP.PropsSI('H', 'P', p_IPT_out, 'Q', 0.9, 'Water')

p_cond_in_ext = 101.3E3
p_cond_out_ext = 101.2E3

h_cond_in_ext = CP.PropsSI('H', 'T', T_ext_cond, 'P', p_cond_in_ext, 'Water')
h_cond_out_ext = CP.PropsSI('H', 'T', T_ext_cond + 5, 'P', p_cond_out_ext, 'Water')
K_cond_ext = 1
A_cond_ext = 1

p_DA = 835E3
x_DA = 0.5
h_DA = CP.PropsSI('H', 'P', p_DA, 'Q', x_DA, 'Water')

#Rankine cycle turbomachinery curves
a_CFP = np.array([2.0155E+02, -1.85E+03, -606840])
a_BFP = np.array([1.00E1, -2.20E3, -8.50E6])
b_CFP = np.array([4.00E-1, 4.80E1, -2.10E3])
b_BFP = np.array([4.00E-1, 2.21E3, -4.61E6])

a_HPT = np.array([0.00E+00, 0.00E+00, 1.54E+09])
a_IPT = np.array([5.68E-13, 4.76E+05, 5.22E+08])
b_HPT = np.array([-3.00E-01, 4.70E+04, -5.31E+08])
b_IPT = np.array([-1.00E-01, 7.40E+03, -1.82E+07])

#Pipe areas - dummy variables used for now
A_CFP_pipe = 0.1
A_BFP_pipe = 0.05
A_cond_pipe = 1
A_turb_valve = 0.1
A_HPT_bypass = 0.1
A_process_heat = 0.1
A_process_cond_split = 0.1
A_IPT_valve = 0.1

######################################## sCO2 cycle ########################################
#Simple recuperated Brayton cycle network
#region
#CoolProp BICU interpolation for speed up
BICU_CO2 = CP.AbstractState('HEOS', 'CO2')
BICU_Water_sCO2 = CP.AbstractState('BICUBIC&HEOS', 'Water')

PR_comp = 2.5
PR_turb = 2.25
#Boundary Conditions and guess values
T_comp_in = 32.6491 + kel
h_comp_in = CP.PropsSI('H', 'P', p_comp_in, 'T', T_comp_in, 'CO2')
p_RXHP_in = PR_comp * p_comp_in
h_RXHP_in = CP.PropsSI('H', 'P', p_RXHP_in, 'T', T_comp_in+30, 'CO2')
p_heater_in = 0.995*p_RXHP_in
h_heater_in = 700E3
p_turb_in = 0.991*p_RXHP_in
h_turb_in = 1080E3
p_RXLP_in = p_turb_in/PR_turb
h_RXLP_in = 800E3
p_cooler_in = 0.995*p_RXLP_in
h_cooler_in = 620E3

p_cooler_ext_in = 101.3E3
T_cooler_ext_in = 25 + kel
h_cooler_ext_in = CP.PropsSI('H', 'P', p_cooler_ext_in, 'T', T_cooler_ext_in, 'Water')
p_cooler_ext_out = 100.3E3
T_cooler_ext_out = 38 + kel
h_cooler_ext_out = CP.PropsSI('H', 'P', p_cooler_ext_out, 'T', T_cooler_ext_out, 'Water')

K_RXHP = 1.81E4
K_H = 0.5E6
K_RXLP = 0.2E4
K_PC = 0.75E4
K_PC_mid = 1E2
```

```
213 K_PC_ext = 1
214 K_RX_pipe = 50
215
216 #Turbomachinery curves
217 a_C = np.array([2.32200E+01, -8.13240E+01, -3.10990E+03, -3.63450E+02])
218 b_C = np.array([5.7017E-01, 3.0758E+01, -8.0928E+02, 4.5139E+03])
219 a_T = np.array([1.0000E+00, 4.1900E+04, 8.9500E+08])
220 b_T = np.array([3.7606E-01, 5.0077E+04, -1.3483E+09, 7.0178E+12])
221
222 RX_n_disc = 6
223 PC_n_disc = 6
224
225 T_heater_ext = 600+273.15
226 T_PC_ext = 35 + 273.15
227
228 #Setup nodes for sCO2 Brayton cycle
229 n_comp_in = Node(BICU_CO2, 'n_comp_in', p = p_comp_in, h = h_comp_in, p_bc = True, h_bc = True)
230 n_RXHP_in = Node(BICU_CO2, 'n_RXHP_in', p = p_RXHP_in, h = h_RXHP_in)
231 n_RXHP_out = Node(BICU_CO2, 'n_RXHP_out', p = 0.99*p_RXHP_in, h = h_RXHP_in)
232 n_RXLP_in = Node(BICU_CO2, 'n_RXLP_in', p = p_RXLP_in, h = h_RXLP_in)
233 # n_RXLP_valve_in = Node(BICU_CO2, 'n_RXLP_valve_in', p = 0.99*p_RXLP_in, h = h_RXLP_in)
234 n_cooler_in = Node(BICU_CO2, 'n_cooler_in', p = p_cooler_in, h = h_cooler_in)
235 n_cooler_out = Node(BICU_CO2, 'n_cooler_out', p = 0.99*p_cooler_in, h = h_cooler_in)
236 n_end_sCO2 = Node(BICU_CO2, 'n_comp_in', p = p_comp_in, h = h_comp_in, p_bc = True)
237
238 n_cooler_ext_in = Node(BICU_Water_sCO2, 'n_cooler_ext_in', p = p_cooler_ext_in, h = h_cooler_ext_in, \
239       p_bc = True, h_bc = True)
240 n_cooler_ext_out = Node(BICU_Water_sCO2, 'n_cooler_ext_out', p = p_cooler_ext_out, h = h_cooler_ext_out,\
241        p_bc = False, h_bc = False, m_source = -500)
242
243 #Setup elements for sCO2 Brayton cycle
244 comp = Compressor_isentropic_head_v2(a_C, b_C, 'comp', mdot_guess= 20)
245 turb = Turbine(a_T, b_T, 'turb', mdot_guess = 20)
246 RX = Compound_HX_Counterflow(n_disc = RX_n_disc, K_1 = K_RXHP, K_2 = K_RXLP, UA_HX = UA_RX, id = 'RX')
247 # RXLP_valve = Pipe(A = 1, fins = False, n_disc = 1, K = K_RXLP_valve, id = 'RXLP_valve')
248 RXHP_mid_pipe =  Pipe(1, 0, 1, K_RX_pipe, 'RX_internal_pipe')
249 pre_cooler = Compound_HX_Counterflow(n_disc = PC_n_disc, K_1 = K_PC, K_2 = K_PC_ext, UA_HX = UA_cooler, id = 'PC')
250 ATT_pipe_co2 = Pipe(1, 0, 1, K_ATT_co2, 'ATT_co2_pipe')
251 PC_bypass_pipe = Pipe(A = 1, fins = False, n_disc = 1, K = K_PC_bypass, id = 'PC_bypass')
252 PC_mid_pipe = Pipe(A = 1, fins = False, n_disc = 1, K = K_PC_mid, id = 'PC_midpipe')
253 #endregion
254 ##############################################################################################
255
256 ########Solver parameters########
257 #Global max and min pressures, temperatures, and enthalpy
258 p_global_max = 1E9
259 p_global_min = 1E3
260 T_global_max = 2000 + 273.15
261 T_global_min = 0 + 273.15
262 p_global_min = [1E3, 1E3, 1E3, 1E6, 1E3]
263 p_global_max = [1E9, 1E9, 1E9, 1E9, 1E9]
264 h_global_min = [-5E6, 20E3, 1E3, 1E5, 1E5]
265 h_global_max = [5E6, 5E6, 5E6, 1.5E6, 3E6]
266 vals_global = [p_global_min, p_global_max, h_global_min, h_global_max]
267
268 #Relax params
269 p_relax = 0.8
270 h_relax = 0.10
271 p_relax_co2 = 0.5
272 h_relax_co2 = 0.05
273 ##########################################################
274
275 #Initialisation function for all boiler components
276 MicroGen_1 = MicroGen(Y_f, mdot_f, T_f, T_da, Y_fa, f_pa, f_sa, f_da, HHV, alpha_air,\
277                       w_ca, cp_f, p_amb, p_fur, f_radloss, A_fur, Vol_fur, \
278                        xi_fur, x_fur, d_fa, x1_coke, x2_coke, M_flame, Qdot_ATT,\
279                         HX_params = Gx, uncertainty_factors=UFx, high_ash= 1)
280 MicroGen_1.initialise()
281
282 #Setup nodes for Rankine cycle
283 n_cond_out = Node(BICU_water, 'n_cond_out', p_cond_out, h_cond_out, p_bc = True, h_bc = True)
284 n_CFP_out = Node(BICU_water, 'n_CFP_out', p_CFP_out, h_CFP_out, p_bc = False, h_bc = False)
```

```
n_DA = Two_Phase_Tank(BICU_water, p = p_DA, h = h_DA, x_mix = x_DA, p_bc = True, liq = True,\
                      vap = False, id = 'DA', K_liq= 1)
n_BFP_out = Node(BICU_water, 'n_BFP_out', p_BFP_out, h_BFP_out, p_bc = False, h_bc = False)
n_turbvalve_out = Node(BICU_water, 'n_turbvalve_out', 0.98*MicroGen_1.node_SH2_out_w.p, \
                      h_HPT_out, p_bc = False, h_bc = False, m_source= 0)
n_HPT_out = Node(BICU_water, 'n_HPT_out', p_HPT_out, h_HPT_out, p_bc = False, h_bc = False, m_source= 0)
n_IPT_out = Node(BICU_water, 'n_IPT_out', p_IPT_out, h_IPT_out, p_bc = False, h_bc = False)
n_IPT_valve_out = Node(BICU_water, 'n_IPT_valve_out', 0.99*p_IPT_out, h_IPT_out, p_bc = False, h_bc = False)
n_end = Node(BICU_water, 'n_end', p_cond_out, h_cond_out, p_bc = True, h_bc = False)
n_process = Node(BICU_water, 'n_process', 0.95*p_HPT_out, h_IPT_out, p_bc = False, h_bc = False,\
                 T_bc = bool_process_heat, T = 50+kel)

n_cond_in_ext = Node(BICU_water, 'n_cond_in_ext', p_cond_in_ext, h_cond_in_ext, p_bc = True,\
                      h_bc = True)
n_cond_out_ext = Node(BICU_water, 'n_cond_out_ext', p_cond_out_ext, h_cond_out_ext, p_bc = False, \
                      h_bc = False, m_source = -10000)

#Setup elements for Rankine cycle
CFP = Pump(a_CFP, b_CFP, 'CFP', mdot_guess = 5.5)
CFP_pipe = Pipe(A_CFP_pipe, 0, 0, K_CFP_pipe, 'CFPpipe')
BFP = Pump(a_BFP, b_BFP, 'BFP', mdot_guess = 12)
BFP_pipe = Pipe(A_BFP_pipe, 0, 0, K_BFP_pipe, 'BFPpipe')
turb_valve = Pipe(A_turb_valve, 0, 0, K_turb_valve, 'turbValve')
HPT = Turbine(a_HPT, b_HPT, 'HPT', mdot_guess = 12)
IPT = Turbine(a_IPT, b_IPT, 'IPT', mdot_guess = 6)
cond = Compound_HX_Counterflow(n_disc=1, K_1=K_cond_pipe, K_2= K_cond_ext, \
                               UA_HX=UA_cond, id='cond', A_1 = A_cond_pipe, A_2 = A_cond_ext, cond_1=True)
HPT_bypass = Pipe(A_HPT_bypass, 0, 0, K_HPT_bypass, 'HPTbypass')
process_heat_pipe = Pipe(A_process_heat, 0, 0, K_process, 'Process_heat_pipe')
process_heat_2_pipe = Pipe(A_process_heat, 0, 0, K_process_2, 'Process_heat_pipe_2')

#Connect nodes to elements
comp.connect_IO(n_comp_in, n_RXHP_in)
RX.connect_IO_dual(n_RXHP_in, n_RXHP_out, n_RXLP_in, n_cooler_in)
RXHP_mid_pipe.connect_IO(n_RXHP_out, MicroGen_1.node_inlet_co2)
ATT_pipe_co2.connect_IO(n_RXHP_in, MicroGen_1.node_inlet_co2)
turb.connect_IO(MicroGen_1.node_outlet_co2, n_RXLP_in)
# RXLP_valve.connect_IO(n_RXLP_valve_in, n_cooler_in)
pre_cooler.connect_IO_dual(n_cooler_in, n_cooler_out,\
                           n_cooler_ext_in, n_cooler_ext_out)
PC_mid_pipe.connect_IO(n_cooler_out, n_end_sCO2)
PC_bypass_pipe.connect_IO(n_cooler_in, n_end_sCO2)

#Connect nodes to elements for Rankine cycle
CFP.connect_IO(n_cond_out, n_CFP_out)
CFP_pipe.connect_IO(n_CFP_out, n_DA.node_mix)
BFP.connect_IO(n_DA.node_liq, n_BFP_out)
BFP_pipe.connect_IO(n_BFP_out, MicroGen_1.node_inlet_w)
# boiler_pipe.connect_IO(n_BFP_pipe_out, n_boiler_out)
turb_valve.connect_IO(MicroGen_1.node_SH2_out_w, n_turbvalve_out)
HPT.connect_IO(n_turbvalve_out, n_HPT_out)
# process_cond_split_pipe.connect_IO(n_HPT_out, n_process_cond_split)
HPT_bypass.connect_IO(n_HPT_out, n_DA.node_mix)
IPT.connect_IO(n_HPT_out, n_IPT_out)
# cond_pipe.connect_IO(n_IPT_out, n_end)
process_heat_pipe.connect_IO(n_HPT_out, n_process)
process_heat_2_pipe.connect_IO(n_process, n_DA.node_mix)
cond.connect_IO_dual(n_IPT_out, n_end, n_cond_in_ext, n_cond_out_ext)

#Initialise compound heat exchangers
RX.initialise()
pre_cooler.initialise()

cond.initialise()

#Specify whether closed or open cycle
closed_cycle_fg_air = [False]
closed_cycle_rankine = [True, n_cond_out, n_end]
closed_cycle_cond_ext = [False]
closed_cycle_sCO2 = [False, n_comp_in, n_end_sCO2]
closed_cycle_sCO2_cooler_ext = [False]

```

```
#Process heat extraction
HT_process = Fixed_Q(Qdot_process, id = 'process_heat')

HT_process.connect_IO(process_heat_pipe, 'null')

solve_connectivity_fluid_iter = [1, 1, 1, 1, 1]

#Specify lists of nodes and elements for each network
nodes_fg_air = MicroGen_1.fg_nodes + MicroGen_1.air_nodes
elements_fg_air = MicroGen_1.fg_elements + MicroGen_1.air_elements

nodes_w = [n_cond_out, n_CFP_out, n_DA.node_mix, n_DA.node_liq,\
            n_BFP_out] + MicroGen_1.w_nodes + [n_turbvalve_out,\
            n_HPT_out, n_process, n_IPT_out, n_end] + cond.nodes_1
elements_w = [CFP, CFP_pipe, n_DA.pipe_liq, BFP, BFP_pipe] +\
             MicroGen_1.w_elements + [turb_valve, HPT, IPT,\
            HPT_bypass, process_heat_pipe, process_heat_2_pipe] + cond.elements_1

nodes_ext = [n_cond_in_ext, n_cond_out_ext] + cond.nodes_2
elements_ext = [] + cond.elements_2

nodes_CO2 = [n_comp_in, n_RXHP_in] + RX.nodes_1 + [n_RXHP_out] + \
      MicroGen_1.co2_nodes + [n_RXLP_in] + RX.nodes_2 + [n_cooler_in]\
              + pre_cooler.nodes_1 + [n_cooler_out, n_end_sCO2]
elements_CO2 = [comp, ATT_pipe_co2] + RX.elements_1 + [RXHP_mid_pipe] \
      + MicroGen_1.co2_elements + [turb] + RX.elements_2 + pre_cooler.elements_1 \
            + [PC_mid_pipe, PC_bypass_pipe]

nodes_sCO2_cooler_ext = [n_cooler_ext_in] + pre_cooler.nodes_2 + [n_cooler_ext_out]
elements_sCO2_cooler_ext = pre_cooler.elements_2

nodes_all_ls = [nodes_fg_air, nodes_w, nodes_ext, nodes_CO2, nodes_sCO2_cooler_ext]
elements_all_ls = [elements_fg_air, elements_w, elements_ext, elements_CO2,\
                    elements_sCO2_cooler_ext]

elements_all = MicroGen_1.air_elements + MicroGen_1.fg_elements + \
            [CFP, CFP_pipe, n_DA.pipe_liq, BFP, BFP_pipe, turb_valve, HPT,\
            IPT, HPT_bypass] + MicroGen_1.w_elements + cond.elements_1 + \
                  cond.elements_2 + elements_CO2 + elements_sCO2_cooler_ext
#List of heat transfer elements
elements_HT = [HT_process] + MicroGen_1.HT_elements + cond.HT_elements + RX.HT_elements \
      + pre_cooler.HT_elements

#Connectivity matrix for each network
conn_matrix_1 = conn_matrix_fn(nodes_fg_air, elements_fg_air)
conn_matrix_2 = conn_matrix_fn(nodes_w, elements_w)
conn_matrix_3 = conn_matrix_fn(nodes_ext, elements_ext)
conn_matrix_CO2 = conn_matrix_fn(nodes_CO2, elements_CO2)
connectivity_matrix_sCO2_cooler_ext = conn_matrix_fn(nodes_sCO2_cooler_ext, elements_sCO2_cooler_ext)

#Connectivity matrix for each HT element
connectivity_matrix_HT = conn_matrix_fn_HT(elements_all, elements_HT)
connectivity_fluid = [conn_matrix_1, conn_matrix_2, conn_matrix_3, conn_matrix_CO2, \
                      connectivity_matrix_sCO2_cooler_ext]
closed_cycle = [closed_cycle_fg_air, closed_cycle_rankine, closed_cycle_cond_ext, \
                closed_cycle_sCO2, closed_cycle_sCO2_cooler_ext]

#Solver function
mom_conv, enth_conv = network_solver(connectivity_fluid, connectivity_matrix_HT, closed_cycle,\
                        relax_params = [[p_relax, h_relax], [p_relax, h_relax], [p_relax, h_relax], \
                                        [p_relax, h_relax_co2], [p_relax, h_relax]], \
                        enthalpy_conv_crit=1E-3, mom_conv_crit=1e-4,global_limits = vals_global, \
                        timer = False, mom_bal = True, energy_bal = True, debug = '', max_iter= 1500, \
                        solve_connectivity_fluid_iter=solve_connectivity_fluid_iter)

#Print inputs and results
res_ins, res_comp = res_inputs_process()
print(res_ins)
print(res_comp)
```

	0
Cond outlet pressure (kPa)	1.865657e+01
CFP Valve K	2.596213e+04
HPT Bypass K	2.798208e+04
ATT Qdot (kW)	1.256906e+03
BFP Pipe K	7.962829e+02
Turb Valve K	1.470000e+01
Process heat K	2.465100e+05
Process heat pipe 2 K	4.000000e+00
Cond Pipe K	1.500000e+00
T_ext_cond (C)	2.500000e+01
Process heat (kW)	2.000000e+03
Fuel Flowrate (kg/s)	6.509767e+00
K_ATT_co2	4.800000e+07
sCO2 Main valve	0.000000e+00
K_PC_bypass	2.000000e+11
sCO2 Compressor inlet pressure (MPa)	1.000000e+01
UA RX (W/K)	2.100000e+05
UA cooler (W/K)	1.700000e+05
UA_cond (W/K)	7.181772e+05
sCO2 eta_th	3.412503e-01

	Set	Actual	Diff	Rel Diff
hsat_liq (kJ/kg)	245.164096	245.164418	0.000322	0.000000
mdot_cond_bal (kg/s)	10.243412	10.243123	-0.000289	-0.002823
esource_DA (kW)	0	-0.610210	-0.610210	NaN
esource_SH2 (kW)	0	0.168526	0.168526	NaN
Boiler mdot_bal (kg/s)	13.174206	13.174213	0.000007	0.000054
Boiler pres bal (kPa)	6914.663114	6914.663106	-0.000008	-0.000000
esource_SD (kW)	0	0.341185	0.341185	NaN
process_esource (kW)	0	-0.021277	-0.021277	NaN
T_sco2_out (C)	550	509.631140	-40.368860	-7.339793
T_sco2_comp_in (C)	32.500000	32.649108	0.149108	0.458793
sCO2 RX efficiency	0.850000	0.891174	0.041174	4.844032

```
426  #Print inputs and results for each node and element in each network
427  res_nodes, res_elems = results_gasmix()
428  display(res_nodes[0])
429  display(res_elems[0])
430  display(res_nodes[1])
431  display(res_elems[1])
432  display(res_nodes[2])
433  display(res_elems[2])
434  display(res_nodes[3])
435  display(res_elems[3])
436  display(res_nodes[4])
437  display(res_elems[4])
```

	p (kPa)	h (kJ/kg)	T (°C)	Density (kg/m3)	Energy Source (kW)	Mass Source (kg/s)	Gas Mixture Y
furn_node_inlet	98.344441	2271.284937	1340.518184	0.198607	56184.717008	8.007506	{'H2O': 0.1932903799772433, 'CO2': 0.188926813...
node_furn_out_fg	98.342849	1744.671976	991.094101	0.253496	34.111443	0.000000	{'H2O': 0.1932903799772433, 'CO2': 0.188926813...
n_SH2_out_fg	98.341519	1565.499203	867.737219	0.280902	12.083314	0.000000	{'H2O': 0.19329037997724333, 'CO2': 0.18892681...
n_SCR1_out_fg	98.340275	1518.858145	835.152120	0.289157	3.176130	0.000000	{'H2O': 0.19329037997724333, 'CO2': 0.18892681...
n_tCAV1_out_fg	98.339070	1466.142269	798.063977	0.299164	3.602088	0.000000	{'H2O': 0.19329037997724333, 'CO2': 0.18892681...
n_sCO2HX_out_fg	98.337973	1244.678146	638.892987	0.351371	15.997516	0.000000	{'H2O': 0.19329037997724333, 'CO2': 0.18892681...
n_bCAV1_out_fg	98.336971	1227.677612	626.426416	0.356237	1.176164	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_SCR2_out_fg	98.335983	1208.461210	612.288553	0.361921	1.345400	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_CAV2_out_fg	98.335009	1196.098653	603.166747	0.365685	0.861180	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_SH1_out_fg	98.334102	1044.775390	489.730597	0.420056	10.834005	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_bEV_out_fg	98.333300	947.340028	414.802553	0.465803	7.178864	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_tEV_out_fg	98.332564	887.902377	368.305887	0.499563	4.479702	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_tAH_out_fg	98.331882	825.827893	319.078045	0.541085	4.795596	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_EC3_out_fg	98.331247	779.952275	282.247077	0.576963	3.595195	0.000000	{'H2O': 0.19329037997724335, 'CO2': 0.18892681...
n_EC2_out_fg	98.330649	742.909945	252.226376	0.609928	2.935955	0.000000	{'H2O': 0.19329037997724338, 'CO2': 0.18892681...
n_EC1_out_fg	98.330082	712.689998	227.549213	0.639984	2.416579	0.000000	{'H2O': 0.19329037997724338, 'CO2': 0.18892681...
n_bAHt_out_fg	98.329543	679.786321	200.493816	0.676538	2.647244	0.000000	{'H2O': 0.19329037997724338, 'CO2': 0.18892681...
n_bAHb_out_fg	98.329036	642.359288	169.490441	0.723920	-17100.917970	-26.626779	{'H2O': 0.19329037997724338, 'CO2': 0.18892681...
n_inlet_air	98.355940	39.798565	25.000000	1.133591	741.020287	18.619272	{'O2': 0.2291296237414236, 'N2': 0.75460878973...
n_bAHb_out_air	98.354277	93.484201	77.225716	0.964606	-0.005157	0.000000	{'O2': 0.22912962374142362, 'N2': 0.7546087897...
n_bAHt_out_air	98.352362	140.816671	123.093175	0.852931	-0.005896	0.000000	{'O2': 0.22912962374142362, 'N2': 0.7546087897...
n_tAH_out_air	98.350109	230.529148	209.284783	0.700530	0.013812	0.000000	{'O2': 0.22912962374142362, 'N2': 0.7546087897...

	mdot (kg/s)	Qdot (kW)	Wdot (kW)
Combustion_CV	18.619273	0.000000	0.0
furn_CV_furn	26.626779	-14056.116127	0.0
SH2_pipe_1	26.626780	-4782.870955	0.0
SCR1_pipe_1	26.626782	-1245.078889	0.0
tCAV1_pipe_1	26.626781	-1407.221565	0.0
sCO2HX_pipe_1	26.626806	-5912.798791	0.0
bCAV1_pipe_1	26.626834	-453.854592	0.0
SCR2_pipe_1	26.626831	-513.024240	0.0
CAV2_pipe_1	26.626828	-330.053987	0.0
SH1_pipe_1	26.626816	-4040.136465	0.0
bEV_pipe_1	26.626795	-2601.598190	0.0
tEV_pipe_1	26.626780	-1587.125163	0.0
tAH_pipe_1	26.626773	-1657.641666	0.0
EC3_pipe_1	26.626771	-1225.115523	0.0
EC2_pipe_1	26.626770	-989.252952	0.0
EC1_pipe_1	26.626770	-807.074962	0.0
bAHt_pipe_1	26.626771	-878.764856	0.0
bAHb_pipe_1	26.626771	-999.591923	0.0
bAHb_pipe_2	18.619271	999.591923	0.0
bAHt_pipe_2	18.619270	881.301645	0.0
tAH_pipe_2	18.619272	1670.368818	0.0

	p (kPa)	h (kJ/kg)	T (°C)	Density (kg/m3)	Energy Source (kW)	Mass Source (kg/s)	Gas Mixture Y
n_cond_out	18.656570	245.160839	58.561556	983.899089	2511.216931	0	{}
n_CFP_out	2283.040221	248.596500	58.927273	984.695820	-0.000046	0	{}
n_mix_DA	835.000000	1749.361158	172.189780	8.625156	-0.608217	0	{}
n_liq_DA	834.994921	728.678691	172.189525	895.201792	-13446.669152	0	{}
n_BFP_out	6951.495819	739.596229	173.934686	897.262872	0.000038	0	{}
n_BFP_pipe_out	6920.690092	739.596215	173.938514	897.239617	-0.000258	0	{}
n_EC1_out_w	6918.739741	800.898728	187.924739	882.343241	-0.006471	0	{}
n_EC2_out_w	6916.751269	876.039760	204.814941	863.013534	-0.003338	0	{}
n_EC3_out_w	6914.710548	968.748063	225.192898	837.461181	-0.008932	0	{}
n_mix_steam_drum	6914.663890	2018.490604	285.000008	68.715913	-140.807015	0	{}
n_liq_steam_drum	6914.663890	1263.249760	285.000963	741.252972	-9949.689556	0	{}
n_gas_steam_drum	6914.662055	2773.731471	284.998147	36.027888	9949.731940	0	{}
n_SH1_out_w	6914.100182	3059.680550	363.568497	26.928660	0.225475	0	{}
n_ATT_out_w	6913.483309	2964.277689	332.614665	29.451556	0.000870	0	{}
n_SH2_out_w	6912.774654	3375.964676	485.000000	21.000159	-103.412754	0	{}
n_turbvalve_out	6906.697431	3375.964676	484.968678	20.981574	-0.000056	0	{}
n_HPT_out	1625.942708	3089.249795	324.460449	6.105960	0.000085	0	{}
n_process	835.008396	728.243492	172.090201	895.305768	-0.018776	0	{}
n_IPT_out	18.974740	2603.357294	58.924866	0.124642	0.000005	0	{}
n_end	18.656570	245.160843	58.561557	983.899089	-2511.279306	0	{}

	mdot (kg/s)	Qdot (kW)	Wdot (kW)
CFP	10.243141	0.000000	-35.192006
CFPpipe	10.243141	0.000000	0.000000
pipe_liq_DA	13.174194	0.000000	0.000000
BFP	13.174195	0.000000	-143.829648
BFPpipe	13.174194	0.000000	0.000000
EC1_pipe_2	13.174194	807.617002	0.000000
EC2_pipe_2	13.174194	989.925499	0.000000
EC3_pipe_2	13.174194	1221.364147	0.000000
EC3_Pipe	13.174193	0.000000	0.000000
pipe_liq_steam_drum	13.174194	0.000000	0.000000
pipe_vap_steam_drum	13.174194	0.000000	0.000000
EV_w	13.174194	23920.012419	0.000000
SH1_pipe_2	13.174210	3766.988107	0.000000
ATT	13.174222	-1256.905847	0.000000
SH2_pipe_2	13.174206	5526.999538	0.000000
turbValve	13.174194	0.000000	0.000000
HPT	13.174194	0.000000	3777.237567
IPT	10.243395	0.000000	4977.188627
HPTbypass	2.083695	0.000000	0.000000
Process_heat_pipe	0.847104	-2000.000000	0.000000
Process_heat_pipe_2	0.847104	0.000000	0.000000
pipe_1_0cond	10.243395	-24155.936592	0.000000

	p (kPa)	h (kJ/kg)	T (°C)	Density (kg/m3)	Energy Source (kW)	Mass Source (kg/s)	Gas Mixture Y
n_cond_in_ext	101.300000	104.920097	24.999998	997.047644	1.049201e+06	0	{}
n_cond_out_ext	51.147558	107.335585	25.588785	996.872397	-1.073357e+06	-10000	{}

	mdot (kg/s)	Qdot (kW)	Wdot (kW)
pipe_2_0cond	10000.0	24155.936592	0.0

	p (kPa)	h (kJ/kg)	T (°C)	Density (kg/m3)	Energy Source (kW)	Mass Source (kg/s)	Gas Mixture Y
n_comp_in	10000.000000	280.651714	32.649108	742.497568	5819.551163	0	{}
n_RXHP_in	24528.163089	301.310304	53.383146	814.075678	0.008161	0	{}
n_10RX	24504.096716	328.857871	65.908746	751.711736	-0.217929	0	{}
n_11RX	24477.856701	358.135432	78.817717	684.712135	-0.025316	0	{}
n_12RX	24448.519455	395.603544	95.236819	601.165650	0.442962	0	{}
n_13RX	24413.890147	450.321841	120.647347	492.278514	1.070401	0	{}
n_14RX	24369.642534	531.488570	165.399573	372.144704	0.541120	0	{}
n_RXHP_out	24310.353974	637.513607	237.555410	277.270394	-0.235570	0	{}
n_sCO2HX_in	24310.323954	603.368230	212.920765	301.890530	-0.006391	0	{}
n_turb_in	24171.404759	835.694079	392.147091	190.404908	-0.735332	0	{}
n_RXLP_in	10218.987876	747.181277	299.472508	97.226826	-0.689036	0	{}
n_20RX	10194.789370	651.924316	217.212769	118.528491	0.185462	0	{}
n_21RX	10174.958964	579.007834	156.340794	144.744170	-0.519421	0	{}
n_22RX	10158.527867	529.850358	118.568219	171.919608	-1.043578	0	{}
n_23RX	10144.513087	496.187480	95.663871	198.243276	-0.434544	0	{}
n_24RX	10132.286788	469.884009	80.268106	225.524699	0.046244	0	{}
n_cooler_in	10121.586541	445.135912	68.342458	258.770555	0.251214	0	{}
n_10PC	10089.278115	398.486387	53.325948	349.288460	0.239458	0	{}
n_11PC	10064.712627	365.063293	47.457698	443.088533	0.183520	0	{}
n_12PC	10044.962474	337.696968	43.848284	537.637789	0.142396	0	{}
n_13PC	10028.317455	314.872202	40.396443	622.376940	0.163947	0	{}
n_14PC	10013.596103	296.697469	36.741406	688.101000	0.207214	0	{}
n_cooler_out	10000.029228	283.170225	33.342685	734.189265	0.238838	0	{}
n_comp_in	10000.000000	283.351929	33.391897	733.585620	-5875.496443	0	{}

	mdot (kg/s)	Qdot (kW)	Wdot (kW)
comp	20.735848	0.000000	-428.357173
ATT_co2_pipe	2.106235	0.000000	0.000000
pipe_1_0RX	18.629585	513.404875	0.000000
pipe_1_1RX	18.629579	545.478936	0.000000
pipe_1_2RX	18.629652	697.681702	0.000000
pipe_1_3RX	18.629866	1018.462913	0.000000
pipe_1_4RX	18.630055	1511.599576	0.000000
pipe_1_5RX	18.630005	1975.425203	0.000000
RX_internal_pipe	18.629933	0.000000	0.000000
sCO2HX_pipe_2	20.736027	4818.090821	0.000000
turb	20.735880	0.000000	1834.704322
pipe_2_0RX	20.735923	-1975.425203	0.000000
pipe_2_1RX	20.735911	-1511.599576	0.000000
pipe_2_2RX	20.735732	-1018.462913	0.000000
pipe_2_3RX	20.735472	-697.681702	0.000000
pipe_2_4RX	20.735313	-545.478936	0.000000
pipe_2_5RX	20.735261	-513.404875	0.000000
pipe_1_0PC	20.711966	-966.422176	0.000000
pipe_1_1PC	20.711988	-692.405804	0.000000
pipe_1_2PC	20.712063	-566.912028	0.000000
pipe_1_3PC	20.712170	-472.869248	0.000000
pipe_1_4PC	20.712288	-376.606777	0.000000
pipe_1_5PC	20.712390	-280.396180	0.000000
PC_midpipe	20.712433	0.000000	0.000000
PC_bypass	0.023277	0.000000	0.000000

	p (kPa)	h (kJ/kg)	T (°C)	Density (kg/m3)	Energy Source (kW)	Mass Source (kg/s)	Gas Mixture Y
n_cooler_ext_in	101.300000	104.920097	24.999998	997.047644	52460.048265	0	{}
n_20PC	100.547766	105.480618	25.134219	997.012793	-0.150338	0	{}
n_21PC	99.795501	106.233576	25.314465	996.965841	-0.142026	0	{}
n_22PC	99.043196	107.179094	25.540769	996.906543	-0.122940	0	{}
n_23PC	98.290842	108.312709	25.812065	996.834890	-0.116358	0	{}
n_24PC	97.538428	109.697233	26.143379	996.746526	-0.160023	0	{}
n_cooler_ext_out	96.785933	111.629624	26.605748	996.621635	-55814.968793	-500	{}

	mdot (kg/s)	Qdot (kW)	Wdot (kW)
pipe_2_0PC	499.999999	280.396180	0.0
pipe_2_1PC	499.999998	376.606777	0.0
pipe_2_2PC	499.999997	472.869248	0.0
pipe_2_3PC	499.999996	566.912028	0.0
pipe_2_4PC	499.999995	692.405804	0.0
pipe_2_5PC	499.999993	966.422176	0.0

```
438  #Fuel flowrate control - Outer loops
439  def rankine_process_solver_inner_bp(x):
440        #Set inputs from x input vector
441        #Backpressure, CFP valve, Process heat valve, Bypass valve
442        n_cond_out.p = x[0]
443        n_end.p = x[0]
444        CFP_pipe.K = np.abs(x[1])
445        process_heat_pipe.K = np.abs(x[2])
446        HPT_bypass.K = np.abs(x[3])
447
```

```
448         #Update connectivity matrices if required
449         elements_all = [CFP, CFP_pipe, n_DA.pipe_liq, BFP, BFP_pipe, turb_valve, HPT,\
450               IPT, HPT_bypass] + MicroGen_1.w_elements + cond.elements_1 + cond.elements_2
451         elements_HT = [HT_process]  + cond.HT_elements
452 
453         conn_matrix_2 = conn_matrix_fn(nodes_w, elements_w)
454         conn_matrix_3 = conn_matrix_fn(nodes_ext, elements_ext)
455 
456         connectivity_matrix_HT = conn_matrix_fn_HT(elements_all, elements_HT)
457         connectivity_fluid = [conn_matrix_2, conn_matrix_3]
458         closed_cycle = [closed_cycle_rankine, closed_cycle_cond_ext]
459 
460         #Global max and min pressures, temperatures, and enthalpy
461         p_global_min = [1E3, 1E3, 1E3, 1E6, 1E3]
462         p_global_max = [1E9, 1E9, 1E9, 1E9, 1E9]
463         T_global_max = 2000 + 273.15
464         T_global_min = 0 + 273.15
465         h_global_min = [1e3, 1e3]
466         h_global_max = [5E6, 5e6]
467         vals_global = [p_global_min, p_global_max, h_global_min, h_global_max]
468 
469         #Enforce required boundary conditions for quick boiler update
470         MicroGen_1.node_EC1_out_w.alpha_h = True
471         MicroGen_1.node_EC2_out_w.alpha_h = True
472         MicroGen_1.node_EC3_out_w.alpha_h = True
473         MicroGen_1.node_SH1_out_w.alpha_h = True
474         MicroGen_1.node_ATT_out_w.alpha_h = True
475         # MicroGen_1.node_SH2_out_w.alpha_h = True
476 
477         # print("Inputs: cond pressure: {0:.4f} (kPa), CFP_pipe_K: {1:4f}, ATT_Qdot: {2:4f} (kW),\
478         # BFP_pipe_K: {3:4f}, Fuel flow: {7:4f}, Turb_valve: {4:4f}, process_K: {5:4f},\
479         # bypass_K: {6:4f}".format(n_cond_out.p/1E3, CFP_pipe.K, MicroGen_1.ATT_HT.Qdot_fixed/1E3,\
480         # BFP_pipe.K, turb_valve.K, process_heat_pipe.K, HPT_bypass.K, MicroGen_1.mdot_f))
481 
482         #Solve network
483            mom_conv, enth_conv = network_solver(connectivity_fluid, connectivity_matrix_HT, closed_cycle, relax_params =
[[p_relax, h_relax]], \
484        enthalpy_conv_crit=1E-4, mom_conv_crit= 1E-6, global_limits = vals_global, timer = False, mom_bal = True, energy_bal
= True, debug = '', verbose=False, iter_print = 100, max_iter=1500)
485 
486         #Calculate target functions
487         h_sat_liq_CFP_in = CP.PropsSI('H', 'P', n_cond_out.p, 'Q', 0, 'Water')
488         eq_p = (n_cond_out.h - h_sat_liq_CFP_in)
489         eq_CFP = CFP.mdot - cond.elements_1[0].mdot
490         eq_DA = n_DA.node_mix.energy_source
491         eq_boiler_valve = MicroGen_1.SH1.pipe_2.mdot - MicroGen_1.EC3.pipe_2.mdot
492         eq_ATT = MicroGen_1.node_SH2_out_w.energy_source
493         eq_SD = MicroGen_1.steam_drum.node_mix.energy_source
494         eq_process = n_process.energy_source
495 
496         #List of target functions corresponding to inputs
497         x_eq = [eq_p, eq_CFP, eq_process, eq_DA]
498 
499         #Dataframe to show results each iteration of secant algorithm
500         indep_var = [n_cond_out.p/1E3, CFP_pipe.K, MicroGen_1.ATT_HT.Qdot_fixed/1E3, BFP_pipe.K,\
501                      MicroGen_1.mdot_f, process_heat_pipe.K, HPT_bypass.K]
502         dep_var = [eq_p/1E3, eq_CFP, eq_ATT/1E3, eq_boiler_valve, eq_SD/1E3, eq_process/1E3, eq_DA/1E3]
503 
504         df_res = pd.DataFrame(index = ['Inputs', 'Outputs'], columns = ['Backpressure (h_satliq)', 'CFP Valve (cond mdot)',
\
505                                   'ATT Qdot (SH2_T_out)', 'BFP pipe valve (boiler pres)', 'Fuel flow rate (SD esource)', \
506                                   'Process heat K (e_process)', 'Bypass K (e_DA)'])
507         df_res.iloc[0:,] = indep_var
508         df_res.iloc[1:,] = dep_var
509 
510         # display(df_res)
511 
512         #Return target functions to secant algorithm
513         return x_eq
514 
515     def rankine_process_solver_outer(x):
516         #Set inputs from x vector
```

```
517         #Qatt, BFP valve, Fuel flow rate, pre-cooler bypass valve
518         MicroGen_1.ATT_HT.Qdot_fixed = x[0]
519         BFP_pipe.K = np.abs(x[1])
520         MicroGen_1.mdot_f = x[2]
521         PC_bypass_pipe.K = np.abs(x[3])
522         K_PC_bypass = x[3]
523         print('Inner backpressure, CFP valve, bypass valve iteration')
524            print("Inputs: cond pressure: {0:.4f} (kPa), CFP_pipe_K: {1:4f}, ATT_Qdot: {2:4f} (kW), BFP_pipe_K: {3:4f},
Turb_valve:     {4:4f}     (kg/s),     process_K:     {5:4f},     bypass_K:     {6:4f}".format(n_cond_out.p/1E3,     CFP_pipe.K,
MicroGen_1.ATT_HT.Qdot_fixed/1E3, BFP_pipe.K, turb_valve.K, process_heat_pipe.K, HPT_bypass.K))
525                         h_boiler   =   [MicroGen_1.node_EC1_out_w,   MicroGen_1.node_EC2_out_w,   MicroGen_1.node_EC3_out_w,
MicroGen_1.node_SH1_out_w, MicroGen_1.node_ATT_out_w, MicroGen_1.node_SH2_out_w]
526         # res = newton(rankine_process_solver_inner_bp, [n_cond_out.p, CFP_pipe.K, process_heat_pipe.K, HPT_bypass.K], \
527         #maxiter = 100, tol = [1e-1, 1e-1, 1e-3, 1e-3], relax = 0.5)
528
529         #Adjust BC for quick boiler update
530         MicroGen_1.node_EC1_out_w.alpha_h = False
531         MicroGen_1.node_EC2_out_w.alpha_h = False
532         MicroGen_1.node_EC3_out_w.alpha_h = False
533         MicroGen_1.node_SH1_out_w.alpha_h = False
534         MicroGen_1.node_ATT_out_w.alpha_h = False
535
536         #Target functions for inner iteration loop
537         h_sat_liq_CFP_in = CP.PropsSI('H', 'P', n_cond_out.p, 'Q', 0, 'Water')
538         eq_p = (n_cond_out.h - h_sat_liq_CFP_in)
539         eq_CFP = CFP.mdot - cond.elements_1[0].mdot
540         eq_DA = n_DA.node_mix.energy_source
541         eq_boiler_valve = MicroGen_1.SH1.pipe_2.mdot - MicroGen_1.EC3.pipe_2.mdot
542         eq_ATT = MicroGen_1.node_SH2_out_w.energy_source
543         eq_SD = MicroGen_1.steam_drum.node_mix.energy_source
544         eq_process = n_process.energy_source
545
546         indep_var = [n_cond_out.p/1E3, CFP_pipe.K, MicroGen_1.ATT_HT.Qdot_fixed/1E3, BFP_pipe.K, \
547                      MicroGen_1.mdot_f, process_heat_pipe.K, HPT_bypass.K]
548         dep_var = [eq_p/1E3, eq_CFP, eq_ATT/1E3, eq_boiler_valve, eq_SD/1E3, eq_process/1E3, eq_DA/1E3]
549        df_res = pd.DataFrame(index = ['Inputs', 'Outputs'], columns = ['Backpressure (h_satliq)', 'CFP Valve (cond mdot)',\
550                                  'ATT Qdot (SH2_T_out)', 'BFP pipe valve (boiler pres)', 'Fuel flow rate (SD esource)', \
551                                  'Process heat K (e_process)', 'Bypass K (e_DA)'])
552         df_res.iloc[0:,] = indep_var
553         df_res.iloc[1:,] = dep_var
554
555         #Quick boiler update iteration
556         boiler_update_short()
557         print('Outer fuel flow, process, and cond iteration')
558
559         #Re-establish connectivity matrices if required after boiler update iteration
560          elements_all = MicroGen_1.air_elements + MicroGen_1.fg_elements + [CFP, CFP_pipe, n_DA.pipe_liq, BFP, BFP_pipe,
turb_valve, HPT,\
561                         IPT,  HPT_bypass] + MicroGen_1.w_elements + cond.elements_1 + cond.elements_2 + elements_CO2 +
elements_sCO2_cooler_ext
562         elements_HT = [HT_process] + MicroGen_1.HT_elements + cond.HT_elements + RX.HT_elements + pre_cooler.HT_elements
563
564         conn_matrix_1 = conn_matrix_fn(nodes_fg_air, elements_fg_air)
565         conn_matrix_2 = conn_matrix_fn(nodes_w, elements_w)
566         conn_matrix_3 = conn_matrix_fn(nodes_ext, elements_ext)
567         conn_matrix_CO2 = conn_matrix_fn(nodes_CO2, elements_CO2)
568         connectivity_matrix_sCO2_cooler_ext = conn_matrix_fn(nodes_sCO2_cooler_ext, elements_sCO2_cooler_ext)
569
570         connectivity_matrix_HT = conn_matrix_fn_HT(elements_all, elements_HT)
571                             connectivity_fluid    =    [conn_matrix_1,    conn_matrix_2,    conn_matrix_3,    conn_matrix_CO2,
connectivity_matrix_sCO2_cooler_ext]
572                  closed_cycle   =   [closed_cycle_fg_air,   closed_cycle_rankine,   closed_cycle_cond_ext,   closed_cycle_sCO2,
closed_cycle_sCO2_cooler_ext]
573
574         #Global max and min pressures, temperatures, and enthalpy
575         p_global_min = [1E3, 1E3, 1E3, 1E6, 1E3]
576         p_global_max = [1E9, 1E9, 1E9, 1E9, 1E9]
577         T_global_max = 2000 + 273.15
578         T_global_min = 0 + 273.15
579         #fg, steam (Rankine), cond_rankine_ext, sCO2, water_cond_sCO2
580         h_global_min = [-5E6, 20E3, 1E3, 1E5, 1E2]
581         h_global_max = [5E6, 5E6, 5E6, 3E6, 1e8]
```

```
582         vals_global = [p_global_min, p_global_max, h_global_min, h_global_max]
583 
584         #Relax params
585         p_relax = 0.8
586         h_relax = 0.1
587         p_relax_co2 = 0.7
588         h_relax_co2 = 0.05
589         relax_fg = [p_relax, h_relax]
590         relax_rankine = [p_relax, h_relax]
591         relax_rankine_cond = [p_relax, h_relax]
592         relax_sCO2 = [p_relax_co2, 0.08]
593         relax_sCO2_cond = [p_relax, 0.1]
594 
595         mom_conv, enth_conv = network_solver(connectivity_fluid, connectivity_matrix_HT, closed_cycle, relax_params = \
596                                 [relax_fg, relax_rankine, relax_rankine_cond, relax_sCO2, relax_sCO2_cond], \
597                                 enthalpy_conv_crit=1E-3, mom_conv_crit=1E-6, global_limits = vals_global, timer = False, \
598                               mom_bal = True, energy_bal = True, debug = '', verbose=False, iter_print = 100, max_iter=500)
599 
600         #Update target functions for outer loop
601         h_sat_liq_CFP_in = CP.PropsSI('H', 'P', n_cond_out.p, 'Q', 0, 'Water')
602         eq_p = (n_cond_out.h - h_sat_liq_CFP_in)
603         eq_CFP = CFP.mdot - cond.elements_1[0].mdot
604         eq_DA = n_DA.node_mix.energy_source
605         eq_BFP_valve = 6914.663114 - MicroGen_1.steam_drum.node_mix.p/1E3
606         eq_ATT = MicroGen_1.node_SH2_out_w.energy_source
607         eq_SD = MicroGen_1.steam_drum.node_mix.energy_source
608         eq_process = n_process.energy_source
609         eq_throttle_co2 = MicroGen_1.node_outlet_co2.T - 273.15 - 550
610         eq_PC_bypass = n_end_sCO2.T - 273.15 - 32.5
611         x_eq = [eq_ATT/1E3, eq_BFP_valve, eq_SD/1E3, eq_PC_bypass]
612 
613         indep_var = [n_cond_out.p/1E3, CFP_pipe.K, MicroGen_1.ATT_HT.Qdot_fixed/1E3,BFP_pipe.K, \
614                      MicroGen_1.mdot_f, turb_valve.K, process_heat_pipe.K, HPT_bypass.K, K_RXLP_valve, PC_bypass_pipe.K]
615         dep_var = [eq_p/1E3, eq_CFP, eq_ATT/1E3, eq_BFP_valve, eq_SD/1E3, eq_SD/1E3, eq_process/1E3,\
616                    eq_DA/1E3, eq_throttle_co2, eq_PC_bypass]
617         df_res = pd.DataFrame(index = ['Inputs', 'Outputs'], columns = ['Backpressure (h_satliq)', \
618               'CFP Valve (cond mdot)', 'ATT Qdot (SH2_T_out)', 'BFP pipe valve (SD esource)', 'Fuel flowrate (SDEsource)',\
619               'Turb Valve K (SD Pres)','Process heat K (e_process)', 'Bypass K (e_DA)', 'CO2 Throttle K (T_co2heater_out)',
\
620                'PC Bypass K (T_co2_comp_in)'])
621         df_res.iloc[0:,] = indep_var
622         df_res.iloc[1:,] = dep_var
623 
624         display(df_res)
625         return x_eq

626  #Newton raphson/Secant function call for outer loop solve
627  newton(rankine_process_solver_outer, [MicroGen_1.ATT_HT.Qdot_fixed, BFP_pipe.K, MicroGen_1.mdot_f, PC_bypass_pipe.K], tol
= [1e1, 1e0, 1e-4, 1e2], relax = 0.8)
```

A.2 Boiler class code – single sCO2 heater configuration

boiler_init_rankine_convective_sco2.py

```
#import dependencies
import numpy as np
import CoolProp.CoolProp as CP
from scipy.optimize import fsolve

from base_classes import Node
from heat_transfer.heat_transfer import *
from heat_transfer.boiler_heat_transfer import *
from components.pipe import Energy_Pipe, Pipe
from components.two_phase_tank import Steam_Drum
from components.boiler.combustion_cv import Combustion

from components.boiler.boiler_hx import *
import GasMixtureProperties as gmp
kel = 273.15

#MicroGen boiler class
class MicroGen():
    #Constructor
    def __init__(self, Y_f, mdot_f, T_f, T_da, Y_fa, f_pa, f_sa, f_da, HHV, alpha_air, w_air, cp_f, p_amb,\
        p_fur, f_radloss, A_fur, Vol_fur, xi_fur, x_fur, d_fa, x1_coke, x2_coke, M_flame, Qdot_ATT,\
              HX_params, uncertainty_factors, high_ash = 1):
        self.Y_f = Y_f
        self.mdot_f = mdot_f
        self.T_f = T_f
        self.T_da = T_da
        self.Y_fa = Y_fa
        self.f_pa = f_pa
        self.f_sa = f_sa
        self.f_da = f_da
        self.HHV = HHV
        self.alpha_air = alpha_air
        self.w_air = w_air
        self.cp_f = cp_f

        self.p_amb = p_amb
        self.p_fur = p_fur
        self.f_radloss = f_radloss
        self.A_fur = A_fur
        self.Vol_fur = Vol_fur
        self.xi_fur = xi_fur
        self.x_fur = x_fur
        self.d_fa = d_fa
        self.x1_coke = x1_coke
        self.x2_coke = x2_coke
        self.M_flame = M_flame
        self.Qdot_ATT = Qdot_ATT
        self.high_ash = high_ash
        self.HX_params = HX_params
        self.uncertainty_factors = uncertainty_factors

    def initialise(self):
        cp_fa = 710
        T_amb = 25 + 273.15
        T_aft_guess = 1100 + kel
        p_w_in = 6914.914588E3
        T_fw_i = 130+kel
        h_w_in = CP.PropsSI('H','T', T_fw_i,'P', p_w_in,'Water')
        x_drum = 0.5
        p_drum = CP.PropsSI('P','T', 285+kel,'Q', x_drum, 'Water')
        h_drum = CP.PropsSI('H','T', 285+kel,'Q', x_drum, 'Water')

        #Combustion model and initialisation
        self.combustion_CV = Combustion(self.Y_f, self.mdot_f, self.T_f, self.T_da, 1.005*self.p_fur,\
                                        self.Y_fa, self.f_pa, self.f_sa, self.f_da, \
            self.HHV, id = 'Combustion_CV', alpha = self.alpha_air, w_air = self.w_air, cp_f = self.cp_f)
        self.combustion_CV.initialise()
```

```
68
69          Y_ca = self.combustion_CV.Y_ca
70          mdot_fa = self.combustion_CV.mdot_fa
71          h_amb_air = gmp.enthalpy(Y_ca, T_amb, self.p_amb)
72          self.h_aft = gmp.enthalpy(self.combustion_CV.Y_fg, T = T_aft_guess, p = self.p_fur)
73
74          BICU_water = CP.AbstractState('BICUBIC&HEOS', 'Water')
75          BICU_CO2 = CP.AbstractState('BICUBIC&HEOS', 'CO2')
76
77          #Set up gas mixture for CoolProp BICU updates
78          fg_gasmix = self.combustion_CV.Y_fg.copy()
79          air_gasmix = Y_ca.copy()
80          fg_gasmix.update(air_gasmix)
81          self.BICU_fg_dict = fg_gasmix.copy()
82
83          for key in self.BICU_fg_dict:
84              self.BICU_fg_dict[key] = CP.AbstractState('HEOS&BICUBIC', key)
85
86          #Enthalpy guess and temperature values
87          n_fg_nodes = 16
88          n_w_nodes = 8
89          T_fg_out = 100 + kel
90          T_fw_out = 485 + kel
91          T_fg_guess = np.zeros(n_fg_nodes)
92          T_fg_guess[0] = T_aft_guess
93          for i in range(1, n_fg_nodes):
94              T_fg_guess[i] = T_fg_guess[i-1] - (T_aft_guess - T_fg_out)/n_fg_nodes
95          h_fg_guess = np.zeros(n_fg_nodes) + gmp.enthalpy(self.combustion_CV.Y_fg, T_fg_guess, self.p_fur)
96
97          T_w_guess = np.zeros(n_w_nodes)
98          T_w_guess[0] = T_fw_i + kel
99          for i in range(1, n_w_nodes):
100             T_w_guess[i] = T_w_guess[i-1] + (T_fw_out - T_fw_i)/n_w_nodes
101         h_w_guess = np.zeros(n_w_nodes) + CP.PropsSI('H', 'T', T_w_guess, 'P', p_w_in, 'Water')
102
103         #Assign guess values
104         h_sh2_out = h_fg_guess[1]
105         h_scr1 = h_fg_guess[2]
106         h_cav1 = h_fg_guess[3]
107         h_scr2 = h_fg_guess[4]
108         h_cav2 = h_fg_guess[5]
109         h_sh1 = h_fg_guess[6]
110         h_bev = h_fg_guess[7]
111         h_tev = h_fg_guess[8]
112         h_tah = h_fg_guess[9]
113         h_ec3 = h_fg_guess[10]
114         h_ec2 = h_fg_guess[11]
115         h_ec1 = h_fg_guess[12]
116         h_baht = h_fg_guess[13]
117         h_bahb = h_fg_guess[14]
118
119         h_ec1_w = h_w_guess[1]
120         h_ec2_w = h_w_guess[2]
121         h_ec3_w = h_w_guess[3]
122         h_sh1_w = h_w_guess[5]
123         h_att_w = h_w_guess[6]
124         h_sh2_w = h_w_guess[7]
125
126         #fg nodes
127         self.furn_node_inlet = Node(self.BICU_fg_dict, 'furn_node_inlet', 0.992*self.p_fur, self.h_aft, T =
T_aft_guess,\
128                                     p_bc = False, h_bc = True, Y_bc = True, gasmixture=self.combustion_CV.Y_fg, m_source=
self.combustion_CV.mdot_fg\
129                                         - (self.combustion_CV.mdot_pa + self.combustion_CV.mdot_sa))
130         h_fur_guess = self.furn_node_inlet.h
131         Y_fg_fur = self.furn_node_inlet.gasmixture
132         self.node_furn_out_fg = Node(self.BICU_fg_dict, 'node_furn_out_fg', 0.991*self.p_fur, h_fur_guess, T_aft_guess,
\
133             p_bc = False, h_bc = True, gasmixture= Y_fg_fur)
134         self.node_SH2_out_fg = Node(self.BICU_fg_dict, 'n_SH2_out_fg', 0.990*self.p_fur, h_sh2_out, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
```

```
135          self.node_SCR1_out_fg = Node(self.BICU_fg_dict, 'n_SCR1_out_fg', 0.989*self.p_fur, h_scr1, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
136          self.node_tCAV1_out_fg = Node(self.BICU_fg_dict, 'n_tCAV1_out_fg', 0.988*self.p_fur, h_cav1, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
137          self.node_sCO2HX_out_fg = Node(self.BICU_fg_dict, 'n_sCO2HX_out_fg', 0.985*self.p_fur, h_cav1, T_aft_guess, h_bc
= False, gasmixture= Y_fg_fur)
138          self.node_bCAV1_out_fg = Node(self.BICU_fg_dict, 'n_bCAV1_out_fg', 0.977*self.p_fur, h_cav1, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
139          self.node_SCR2_out_fg = Node(self.BICU_fg_dict, 'n_SCR2_out_fg', 0.976*self.p_fur, h_scr2, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
140          self.node_CAV2_out_fg = Node(self.BICU_fg_dict, 'n_CAV2_out_fg', 0.975*self.p_fur, h_cav2, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
141          self.node_SH1_out_fg = Node(self.BICU_fg_dict, 'n_SH1_out_fg', 0.974*self.p_fur, h_sh1, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
142          self.node_bEV_out_fg = Node(self.BICU_fg_dict, 'n_bEV_out_fg', 0.94*self.p_fur, h_bev, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
143          self.node_tEV_out_fg = Node(self.BICU_fg_dict, 'n_tEV_out_fg', 0.935*self.p_fur, h_tev, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
144          self.node_tAH_out_fg = Node(self.BICU_fg_dict, 'n_tAH_out_fg', 0.93*self.p_fur, h_tah, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
145          self.node_EC3_out_fg = Node(self.BICU_fg_dict, 'n_EC3_out_fg', 0.925*self.p_fur, h_ec3, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
146          self.node_EC2_out_fg = Node(self.BICU_fg_dict, 'n_EC2_out_fg', 0.92*self.p_fur, h_ec2, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
147          self.node_EC1_out_fg = Node(self.BICU_fg_dict, 'n_EC1_out_fg', 0.915*self.p_fur, h_ec1, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
148          self.node_bAHt_out_fg = Node(self.BICU_fg_dict, 'n_bAHt_out_fg', 0.91*self.p_fur, h_baht, T_aft_guess, h_bc =
False, gasmixture= Y_fg_fur)
149          self.node_bAHb_out_fg = Node(self.BICU_fg_dict, 'n_bAHb_out_fg', 0.905*self.p_fur, h_bahb, T_aft_guess, \
150               m_source= -self.combustion_CV.mdot_fg, p_bc = False, h_bc = False, gasmixture= Y_fg_fur)
151
152          #air nodes
153          self.node_inlet_air = Node(self.BICU_fg_dict, 'n_inlet_air', self.p_amb, h_amb_air, T_amb,  \
154              m_source = self.combustion_CV.mdot_pa + self.combustion_CV.mdot_sa, p_bc = True, h_bc = True, Y_bc = True,
gasmixture= Y_ca)
155          self.node_bAHb_out_air = Node(self.BICU_fg_dict, 'n_bAHb_out_air', 0.999*self.p_amb, h_amb_air, T_amb, h_bc =
False, gasmixture= Y_ca)
156          self.node_bAHt_out_air = Node(self.BICU_fg_dict, 'n_bAHt_out_air', 0.995*self.p_amb, h_amb_air, T_amb, h_bc =
False, gasmixture= Y_ca)
157          self.node_tAH_out_air = Node(self.BICU_fg_dict, 'n_tAH_out_air', 0.993*self.p_amb, h_amb_air, T_amb, p_bc =
False, h_bc = False, \
158              gasmixture= Y_ca)
159
160          # water/steam nodes
161          self.node_inlet_w = Node(BICU_water, 'n_BFP_pipe_out', p_w_in, h_w_in, p_bc = False, h_bc = False)
162          self.node_EC1_out_w = Node(BICU_water, 'n_EC1_out_w', 0.9999*p_w_in, h_ec1_w, h_bc = False)
163          self.node_EC2_out_w = Node(BICU_water, 'n_EC2_out_w', 0.9998*p_w_in, h_ec2_w, h_bc = False)
164          self.node_EC3_out_w = Node(BICU_water, 'n_EC3_out_w', 0.9997*p_w_in, h_ec3_w, h_bc = False)
165          self.steam_drum = Steam_Drum(BICU_water, p_drum, h_drum, x_drum, liq = True, vap = True, id = 'steam_drum', \
166              p_bc = False, K_liq= 1,K_vap=1)
167          self.node_SH1_out_w = Node(BICU_water, 'n_SH1_out_w', 0.999*p_drum, h_sh1_w, h_bc = False)
168          self.node_ATT_out_w = Node(BICU_water, 'n_ATT_out_w', 0.998*p_drum, h_att_w, h_bc = False)
169          self.node_SH2_out_w = Node(BICU_water, 'n_SH2_out_w', 0.997*p_drum, h_sh2_w, T = T_fw_out, p_bc = False, h_bc =
False, T_bc = True, m_source= 0)
170
171          # CO2 nodes
172          #Initial guesses
173          p_heater_in = 20.9E6
174          h_heater_in = 700E3
175          p_turb_in = 20.8E6
176          h_turb_in = 1080E3
177          self.node_inlet_co2 = Node(BICU_CO2, 'n_sCO2HX_in', p_heater_in, h_heater_in, p_bc = False, h_bc = False)
178          self.node_outlet_co2 = Node(BICU_CO2, 'n_turb_in', p_turb_in, h_turb_in, p_bc = False, h_bc = False)
179
180          #Secondary loss factors
181          #fg
182          K_furn = 1E-3
183          #assume same for all HX to obtain required % dp
184          K_fg = 1E-3
185          K_SH2_fg = K_fg
186          K_SCR1_fg = K_fg
187          K_tCAV1_fg = K_fg
```

```
188          K_sco2_fg =K_fg
189          K_bCAV1_fg = K_fg
190          K_SCR2_fg = K_fg
191          K_CAV2_fg = K_fg
192          K_SH1_fg = K_fg
193          K_bEV_fg = K_fg
194          K_tEV_fg = K_fg
195          K_tAH_fg = K_fg
196          K_EC3_fg = K_fg
197          K_EC2_fg = K_fg
198          K_EC1_fg = K_fg
199          K_bAHt_fg = K_fg
200          K_bAHb_fg = K_fg
201
202          #air
203          K_a = 1E-2
204          K_bAHb_a = K_a
205          K_bAHt_a = K_a
206          K_tAH_a = K_a
207
208          #water
209          K_w = 20000
210          K_st = 200
211          K_EC1_w = K_w
212          K_EC2_w = K_w
213          K_EC3_w = K_w
214          K_SH1_w = K_st
215          K_SH2_w = K_st
216
217          #co2
218          K_sCO2HX = 1.5E5
219
220          #elements
221          #EV element to represent evaporating circuit heat uptakes
222          self.EV_w = Energy_Pipe(1,0,0,K_w,'EV_w')
223
224          self.ATT = Pipe(1,0,1,K_st, 'ATT')
225          #Intermediate pipe for EC3
226          self.EC3_pipe = Pipe(1,0,1,100, 'EC3_Pipe')
227
228          #HT elem for ATT
229          self.ATT_HT = attemperator_HT(Qdot_fixed= self.Qdot_ATT)
230
231          #Setup compound elements for boiler heat exchangers
232          #Furnace, Radiative Convective HX, Radiative Cavity, Radiative Convective Evaporator, Radiative Convective Air
heater
233          self.furn = Furnace_Compound(self.combustion_CV.mdot_f, self.combustion_CV.mdot_fa, self.combustion_CV.mdot_fg,
\
234              self.f_radloss, self.A_fur, self.Vol_fur, self.xi_fur, self.x_fur, self.d_fa, self.x1_coke, self.x2_coke, \
235                   self.M_flame, 'furn_CV', high_ash= self.high_ash, K = K_furn,\
236                  **self.uncertainty_factors['fur'])
237          self.SH2 = Radiative_Convective_HX(mdot_fa, cp_fa, K_SH2_fg, K_SH2_w, 'SH2', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
238               **{**self.HX_params['SH2'], **self.uncertainty_factors['SH2']})
239          self.SCR1 = Radiative_Convective_EV(mdot_fa, cp_fa, K_SCR1_fg, 'SCR1', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
240              **{**self.HX_params['SCR1'], **self.uncertainty_factors['SCR1']})
241          self.tCAV1 = Radiative_Cavity(mdot_fa, cp_fa, K_tCAV1_fg, 'tCAV1', mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash =
1, \
242              **{**self.HX_params['tCAV1'], **self.uncertainty_factors['tCAV1']})
243          self.sCO2HX = Radiative_Convective_HX(mdot_fa, cp_fa, K_sco2_fg, K_sCO2HX, 'sCO2HX', mdot_ia = 0, Y_ia = {},
T_ia = 0, high_ash = 1, \
244               **{**self.HX_params['sCO2HX'], **self.uncertainty_factors['sCO2HX']})
245          self.bCAV1 = Radiative_Cavity(mdot_fa, cp_fa, K_bCAV1_fg, 'bCAV1', mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash =
1, \
246              **{**self.HX_params['bCAV1'], **self.uncertainty_factors['bCAV1']})
247          self.SCR2 = Radiative_Convective_EV(mdot_fa, cp_fa, K_SCR2_fg, 'SCR2', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
248              **{**self.HX_params['SCR2'], **self.uncertainty_factors['SCR2']})
249          self.CAV2 = Radiative_Cavity(mdot_fa, cp_fa, K_CAV2_fg, 'CAV2', mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash = 1,
\
250              **{**self.HX_params['CAV2'], **self.uncertainty_factors['CAV2']})
```

```
251         self.SH1 = Radiative_Convective_HX(mdot_fa, cp_fa, K_SH1_fg, K_SH1_w, 'SH1', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
252             **{**self.HX_params['SH1'], **self.uncertainty_factors['SH1']})
253         self.bEV = Radiative_Convective_EV(mdot_fa, cp_fa, K_bEV_fg, 'bEV', mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash =
1, \
254             **{**self.HX_params['bEV'], **self.uncertainty_factors['bEV']})
255         self.tEV = Radiative_Convective_EV(mdot_fa, cp_fa, K_tEV_fg, 'tEV', mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash =
1, \
256             **{**self.HX_params['tEV'], **self.uncertainty_factors['tEV']})
257         self.tAH = Radiative_Convective_AH(mdot_fa, cp_fa, K_tAH_fg, K_tAH_a, 'tAH', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
258             **{**self.HX_params['tAH'], **self.uncertainty_factors['tAH']})
259         self.EC3 = Radiative_Convective_HX(mdot_fa, cp_fa, K_EC3_fg, K_EC3_w, 'EC3', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
260             **{**self.HX_params['EC3'], **self.uncertainty_factors['EC3']})
261         self.EC2 = Radiative_Convective_HX(mdot_fa, cp_fa, K_EC2_fg, K_EC2_w, 'EC2', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
262             **{**self.HX_params['EC2'], **self.uncertainty_factors['EC2']})
263         self.EC1 = Radiative_Convective_HX(mdot_fa, cp_fa, K_EC1_fg, K_EC1_w, 'EC1', mdot_ia = 0, Y_ia = {}, T_ia = 0,
high_ash = 1, \
264             **{**self.HX_params['EC1'], **self.uncertainty_factors['EC1']})
265         self.bAHt = Radiative_Convective_AH(mdot_fa, cp_fa, K_bAHt_fg, K_bAHt_a, 'bAHt', mdot_ia = 0, Y_ia = {}, T_ia =
0, high_ash = 1, \
266             **{**self.HX_params['bAHt'], **self.uncertainty_factors['bAHt']})
267         self.bAHb = Radiative_Convective_AH(mdot_fa, cp_fa, K_bAHb_fg, K_bAHb_a, 'bAHb', mdot_ia = 0, Y_ia = {}, T_ia =
0, high_ash = 1, \
268             **{**self.HX_params['bAHb'], **self.uncertainty_factors['bAHb']})
269
270
271         #Connect elements with nodes
272         self.combustion_CV.connect_IO(self.node_tAH_out_air, self.furn_node_inlet)
273
274         self.furn.connect_IO(self.furn_node_inlet, self.node_furn_out_fg)
275         self.SH2.connect_IO_dual(self.node_furn_out_fg, self.node_SH2_out_fg, self.node_ATT_out_w, self.node_SH2_out_w)
276         self.SCR1.connect_IO(self.node_SH2_out_fg, self.node_SCR1_out_fg)
277         self.tCAV1.connect_IO(self.node_SCR1_out_fg, self.node_tCAV1_out_fg)
278         self.sCO2HX.connect_IO_dual(self.node_tCAV1_out_fg, self.node_sCO2HX_out_fg, self.node_inlet_co2,
self.node_outlet_co2)
279         self.bCAV1.connect_IO(self.node_sCO2HX_out_fg, self.node_bCAV1_out_fg)
280         self.SCR2.connect_IO(self.node_bCAV1_out_fg, self.node_SCR2_out_fg)
281         self.CAV2.connect_IO(self.node_SCR2_out_fg, self.node_CAV2_out_fg)
282         self.SH1.connect_IO_dual(self.node_CAV2_out_fg, self.node_SH1_out_fg, self.steam_drum.node_vap,
self.node_SH1_out_w)
283         self.bEV.connect_IO(self.node_SH1_out_fg, self.node_bEV_out_fg)
284         self.tEV.connect_IO(self.node_bEV_out_fg, self.node_tEV_out_fg)
285         self.tAH.connect_IO_dual(self.node_tEV_out_fg, self.node_tAH_out_fg, self.node_bAHt_out_air,
self.node_tAH_out_air)
286         self.EC3.connect_IO_dual(self.node_tAH_out_fg, self.node_EC3_out_fg, self.node_EC2_out_w, self.node_EC3_out_w)
287         self.EC2.connect_IO_dual(self.node_EC3_out_fg, self.node_EC2_out_fg, self.node_EC1_out_w, self.node_EC2_out_w)
288         self.EC1.connect_IO_dual(self.node_EC2_out_fg, self.node_EC1_out_fg, self.node_inlet_w, self.node_EC1_out_w)
289         self.bAHt.connect_IO_dual(self.node_EC1_out_fg, self.node_bAHt_out_fg, self.node_bAHb_out_air,
self.node_bAHt_out_air)
290         self.bAHb.connect_IO_dual(self.node_bAHt_out_fg, self.node_bAHb_out_fg, self.node_inlet_air,
self.node_bAHb_out_air)
291
292         self.EV_w.connect_IO(self.steam_drum.node_liq, self.steam_drum.node_mix)
293         self.ATT.connect_IO(self.node_SH1_out_w, self.node_ATT_out_w)
294         self.EC3_pipe.connect_IO(self.node_EC3_out_w, self.steam_drum.node_mix)
295         self.ATT_HT.connect_IO(self.ATT, self.EV_w)
296
297         #Initialise compound components
298         self.furn.initialise()
299         self.SH2.initialise()
300         self.SCR1.initialise()
301         self.tCAV1.initialise()
302         self.sCO2HX.initialise()
303         self.bCAV1.initialise()
304         self.SCR2.initialise()
305         self.CAV2.initialise()
306         self.SH1.initialise()
307         self.bEV.initialise()
308         self.tEV.initialise()
```

```
309         self.tAH.initialise()
310         self.EC3.initialise()
311         self.EC2.initialise()
312         self.EC1.initialise()
313         self.bAHt.initialise()
314         self.bAHb.initialise()
315 
316         #Assign relevant pipes from steam drum
317         self.EV_w.fixed_pipe = self.EC3.pipe_2
318         self.steam_drum.pipe_liq.fixed_pipe = self.EC3.pipe_2
319 
320         #Connect HT elements for each HX
321         self.furn.steam_HT_elem.connect_IO(self.furn.furnace, self.EV_w)
322         self.furn.re_HT_elem.connect_IO(self.furn.furnace, self.SH2.pipe_1)
323 
324         hx_list = [self.SH2, self.SCR1, self.tCAV1, self.sCO2HX, self.bCAV1, self.SCR2, self.CAV2, self.SH1, self.bEV,
self.tEV, self.tAH, self.EC3, self.EC2, \
325             self.EC1, self.bAHt, self.bAHb]
326         hx_ev_list = [self.SCR1, self.tCAV1, self.bCAV1, self.SCR2, self.CAV2, self.bEV, self.tEV]
327         hx_ah_list = [self.bAHb, self.bAHt, self.tAH]
328         hx_cavity_list = [self.tCAV1, self.bCAV1, self.CAV2]
329 
330         #Connect HT elems from various HXs
331         for i in range(len(hx_list)):
332             hx = hx_list[i]
333             if i != (len(hx_list) - 1):
334                 hx_ds = hx_list[i+1]
335                 hx.re_HT_elem.connect_IO(hx.pipe_1, hx_ds.pipe_1)
336             if hx not in hx_ah_list and hx not in hx_cavity_list:
337                 hx.wall_HT_elem.connect_IO(hx.pipe_1, self.EV_w)
338                 hx.roof_HT_elem.connect_IO(hx.pipe_1, self.EV_w)
339 
340             if hx in hx_ev_list:
341                 hx.steam_HT_elem.connect_IO(hx.pipe_1, self.EV_w)
342 
343         #Collect nodes + elems for each stream
344         self.air_nodes = []
345         self.air_elements = []
346 
347         self.fg_nodes = [self.furn_node_inlet]
348         self.fg_elements = [self.combustion_CV , self.furn.furnace]
349         self.HT_elements = self.furn.HT_elements
350 
351         self.w_nodes = [self.node_inlet_w, self.node_EC1_out_w, self.node_EC2_out_w, self.node_EC3_out_w,
self.steam_drum.node_mix, \
352             self.steam_drum.node_liq, self.steam_drum.node_vap, self.node_SH1_out_w, \
353                 self.node_ATT_out_w, self.node_SH2_out_w]
354         self.w_elements = [self.EC1.pipe_2, self.EC2.pipe_2, self.EC3.pipe_2, self.EC3_pipe, self.steam_drum.pipe_liq,
self.steam_drum.pipe_vap, \
355         self.EV_w, self.SH1.pipe_2,  self.ATT, self.SH2.pipe_2]
356 
357         self.co2_nodes =[self.node_inlet_co2, self.node_outlet_co2]
358         self.co2_elements = [self.sCO2HX.pipe_2]
359 
360         for i in range(len(hx_ah_list)):
361             hx = hx_ah_list[i]
362             self.air_elements.extend(hx.elements_2)
363             self.air_nodes.extend(hx.nodes_2)
364 
365             self.air_nodes.append(hx.inlet_2)
366             if i == (len(hx_ah_list) - 1):
367                 self.air_nodes.append(hx.outlet_2)
368 
369         for i in range(len(hx_list)):
370             hx = hx_list[i]
371             self.fg_elements.extend(hx.elements_1)
372             self.HT_elements.extend(hx.HT_elements)
373             self.fg_nodes.extend(hx.nodes_1)
374             if hx in hx_ev_list:
375                 self.fg_nodes.append(hx.inlet)
376                 if i == (len(hx_list) - 1):
377                     self.fg_nodes.append(hx.outlet)
```

```
378             else:
379                 self.fg_nodes.append(hx.inlet_1)
380                 if i == (len(hx_list) - 1):
381                     self.fg_nodes.append(hx.outlet_1)
382 
383         self.HT_elements.remove(self.bAHb.re_HT_elem)
384         self.HT_elements.append(self.ATT_HT)
385 
```

A.3 Boiler heat exchanger classes

boiler_hx.py

```
#import dependencies
import sys
import numpy as np
import CoolProp.CoolProp as CP
from scipy.optimize import fsolve

from base_classes import Node
from base_classes import Element
from heat_transfer.heat_transfer import *
from heat_transfer.boiler_heat_transfer import *
from components.pipe import Fluegas_Pipe, Pipe, Pipe_Energy
from components.boiler.furnace_projected import Furnace
import GasMixtureProperties as gmp

#Radiative Convective HX
class Radiative_Convective_HX(Element):
    #Constructor
    #K_1, K_2 = secondary loss factor for each fluid stream in series
    #1 - fg, 2 - sec
    def __init__(self, mdot_fa, cp_fa, K_1, K_2, id, mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash = 1, **kwargs):
        Element.__init__(self)
        self.mdot_fa = mdot_fa
        self.mdot_ia = mdot_ia
        self.Y_ia = Y_ia
        self.T_ia = T_ia
        self.cp_fa = cp_fa
        self.K_1 = K_1
        self.K_2 = K_2
        self.id = id
        self.high_ash = high_ash

        self.alpha_conv_i = 0
        self.alpha_conv_o = 0
        self.alpha_rad_epsilon_g = 0

        self.__dict__.update(kwargs)
        self.geo_inputs = kwargs

    #Initialise function once input and output streams have been connected to external nodes
    def initialise(self):

        #Initialise pipes for both streams
        #Stream 1 = fg
        #Stream 2 = other fluid - sec
        # self.pipe_1 = Pipe(0,0,1,self.K_1, self.id+'_pipe_1', fixed_mdot= 46.575893)
       self.pipe_1 = Fluegas_Pipe(self.mdot_fa, self.mdot_ia, self.Y_ia, self.T_ia, self.cp_fa, self.K_1, self.id+'_pipe_1',
\
            high_ash = self.high_ash, fixed_mdot= 0, **self.geo_inputs)
        self.pipe_2 = Pipe(1,0,1,self.K_2, self.id+'_pipe_2', fixed_mdot=0, **self.geo_inputs)

        #Specify other properties

        #Flue gas side
        #A_open for open duct
        self.pipe_1.W_ave = (self.pipe_1.W_i + self.pipe_1.W_e)/2
        self.pipe_1.H_ave = (self.pipe_1.H_i + self.pipe_1.H_e)/2
        self.pipe_1.A_open = self.pipe_1.W_ave * self.pipe_1.H_ave
        self.pipe_2.A_open = self.pipe_1.W_ave * self.pipe_1.H_ave

        #A_ho for external flow area
        if self.pipe_1.platen == 1:
            self.pipe_1.A_ho = 2 * self.pipe_1.L_avg * self.pipe_1.d_OD * self.pipe_1.N_l * self.pipe_1.N_t
            self.pipe_2.A_ho = 2 * self.pipe_1.L_avg * self.pipe_1.d_OD * self.pipe_1.N_l * self.pipe_1.N_t
            self.pipe_1.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
            self.pipe_2.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
```

```
65              self.pipe_1.S_mbl = 1.8 / (1/self.pipe_1.H_ave + 1/self.pipe_1.b_l + self.pipe_1.S_t)
66              self.pipe_2.S_mbl = 1.8 / (1/self.pipe_1.H_ave + 1/self.pipe_1.b_l + self.pipe_1.S_t)
67          else:
68              self.pipe_1.A_ho = np.pi * self.pipe_1.d_OD * (self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t)
69              self.pipe_2.A_ho = np.pi * self.pipe_1.d_OD * (self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t)
70              self.pipe_1.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
71              self.pipe_2.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
72            self.pipe_1.S_mbl = 0.9 * self.pipe_1.d_OD * ((4 * self.pipe_1.S_t * self.pipe_1.S_l)/(np.pi * self.pipe_1.d_OD**2)
- 1)
73            self.pipe_2.S_mbl = 0.9 * self.pipe_1.d_OD * ((4 * self.pipe_1.S_t * self.pipe_1.S_l)/(np.pi * self.pipe_1.d_OD**2)
- 1)
74
75          #Steam side
76          #Internal diameter
77          self.pipe_1.d_ID = self.pipe_1.d_OD - 2 * self.pipe_1.t_wall
78          self.pipe_2.d_ID = self.pipe_1.d_OD - 2 * self.pipe_1.t_wall
79
80          #Free flow internal area
81          self.pipe_1.A_ff = self.pipe_1.N_t * self.pipe_1.pp * np.pi/4 * self.pipe_1.d_ID**2
82          self.pipe_2.A_ff = self.pipe_1.N_t * self.pipe_1.pp * np.pi/4 * self.pipe_1.d_ID**2
83
84          #Internal heat exchange area
85          self.pipe_1.A_hi = np.pi * self.pipe_1.d_ID * self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t
86          self.pipe_2.A_hi = np.pi * self.pipe_1.d_ID * self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t
87
88          #Ingress air adjustment
89          if self.mdot_ia != 0:
90              self.h_ia = gmp.enthalpy(self.Y_ia, self.T_ia, self.inlet_1.p)
91              self.node_ia = Node('gasmix', self.id+'_node_ia', self.inlet_1.p , self.h_ia, self.T_ia, \
92                                  p_bc = True, h_bc = True, Y_bc = True, gasmixture= self.Y_ia)
93              self.pipe_ia = Pipe(0,0,1,1E-6,self.id+'_pipe_ia', fixed_mdot = self.mdot_ia)
94              self.pipe_pre_fg = Pipe(0,0,1,1E-10,self.id+'_pipe_pre_fg')
95              self.node_pre_fg = Node('gasmix', self.id+'_node_pre_fg', self.inlet_1.p , \
96                                      self.inlet_1.h, self.inlet_1.T, gasmixture= self.inlet_1.gasmixture)
97
98          #Heat transfer elements
99          #Radiation
100         self.re_HT_elem = direct_radiation(id = self.id+'_re_HT')
101         #Wall and roof
102         self.wall_HT_elem = sidewall_HT(id = self.id+'_wall_HT')
103         self.roof_HT_elem = roof_HT(id = self.id+'_roof_HT')
104         #Sec - steam/sCO2/air
105         self.steam_HT_elem = steam_HT(id = self.id+'_steam_HT', HX_type = 'radiative_convective')
106
107         #Account for ingress air if required when connecting nodes and elements
108         if self.mdot_ia != 0:
109             self.pipe_ia.connect_IO(self.node_ia, self.node_pre_fg)
110             self.pipe_pre_fg.connect_IO(self.inlet_1, self.node_pre_fg)
111             self.pipe_1.connect_IO(self.node_pre_fg, self.outlet_1)
112             self.pipe_2.connect_IO(self.inlet_2, self.outlet_2)
113         else:
114             self.pipe_1.connect_IO(self.inlet_1, self.outlet_1)
115             self.pipe_2.connect_IO(self.inlet_2, self.outlet_2)
116
117         #Connect HT element between fluegas and sec fluid
118         self.steam_HT_elem.connect_IO(self.pipe_1, self.pipe_2)
119
120         if self.mdot_ia !=0:
121             self.nodes_1 = [self.node_ia, self.node_pre_fg]
122             self.elements_1 = [self.pipe_ia, self.pipe_pre_fg, self.pipe_1]
123             self.nodes_2 = []
124             self.elements_2 = [self.pipe_2]
125         else:
126             self.nodes_1 = []
127             self.elements_1 = [self.pipe_1]
128             self.nodes_2 = []
129             self.elements_2 = [self.pipe_2]
130         #Collect HT elements
131         self.HT_elements = [self.re_HT_elem, self.wall_HT_elem, self.roof_HT_elem, self.steam_HT_elem]
132
133 class Radiative_Convective_EV(Element):
134     #Constructor
```

```
135     #K_1, K_2 = secondary loss factor for each fluid stream in series
136     def __init__(self, mdot_fa, cp_fa, K_1, id, mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash = 1, **kwargs):
137         Element.__init__(self)
138         self.mdot_fa = mdot_fa
139         self.mdot_ia = mdot_ia
140         self.Y_ia = Y_ia
141         self.T_ia = T_ia
142         self.cp_fa = cp_fa
143         self.K_1 = K_1
144         self.id = id
145         self.high_ash = high_ash
146 
147         self.alpha_conv_i = 0
148         self.alpha_conv_o = 0
149         self.alpha_rad_epsilon_g = 0
150 
151         self.__dict__.update(kwargs)
152         self.geo_inputs = kwargs
153 
154     #Initialise function once input and output streams have been connected to external nodes
155     def initialise(self):
156 
157         #Initialise pipes for both streams
158         #Stream 1 = fg
159         #Stream 2 = other fluid
160        self.pipe_1 = Fluegas_Pipe(self.mdot_fa, self.mdot_ia, self.Y_ia, self.T_ia, self.cp_fa, self.K_1, self.id+'_pipe_1',
\
161             high_ash = self.high_ash, fixed_mdot= 0, **self.geo_inputs)
162 
163         #Specify other properties
164         W_ave = (self.pipe_1.W_i + self.pipe_1.W_e)/2
165         H_ave = (self.pipe_1.H_i + self.pipe_1.H_e)/2
166         self.pipe_1.A_open = W_ave * H_ave
167 
168         #A_ho for external flow area
169         if self.pipe_1.platen == 1:
170             self.pipe_1.A_ho = 2 * self.pipe_1.L_avg * self.pipe_1.d_OD * self.pipe_1.N_l * self.pipe_1.N_t
171             self.pipe_1.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
172             self.pipe_1.S_mbl = 1.8 / (1/self.pipe_1.H_ave + 1/self.pipe_1.b_l + self.pipe_1.S_t)
173         else:
174             self.pipe_1.A_ho = np.pi * self.pipe_1.d_OD * (self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t)
175             self.pipe_1.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
176           self.pipe_1.S_mbl = 0.9 * self.pipe_1.d_OD * ((4 * self.pipe_1.S_t * self.pipe_1.S_l)/(np.pi * self.pipe_1.d_OD**2)
- 1)
177 
178         #Steam side
179         #Internal diameter
180         self.pipe_1.d_ID = self.pipe_1.d_OD - 2 * self.pipe_1.t_wall
181 
182         #Free flow internal area
183         self.pipe_1.A_ff = self.pipe_1.N_t * self.pipe_1.pp * np.pi/4 * self.pipe_1.d_ID**2
184 
185         #Internal heat exchange area
186         self.pipe_1.A_hi = np.pi * self.pipe_1.d_ID * self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t
187 
188         #Ingress air adjustment
189         if self.mdot_ia != 0:
190             self.h_ia = gmp.enthalpy(self.Y_ia, self.T_ia, self.inlet.p)
191            self.node_ia = Node('gasmix', self.id+'_node_ia', self.inlet.p , self.h_ia, self.T_ia, p_bc = True, h_bc = True,
\
192                  Y_bc = True, gasmixture= self.Y_ia)
193             self.pipe_ia = Pipe(0,0,1,1E-6,self.id+'_pipe_ia', fixed_mdot = self.mdot_ia)
194             self.pipe_pre_fg = Pipe(0,0,1,1E-10,self.id+'_pipe_pre_fg')
195             self.node_pre_fg = Node('gasmix', self.id+'_node_pre_fg', self.inlet.p , self.inlet.h, self.inlet.T, \
196                 gasmixture= self.inlet.gasmixture)
197 
198         #Heat transfer elements
199         #Radiation
200         self.re_HT_elem = direct_radiation(id = self.id+'_re_HT')
201         #Wall and roof
202         self.wall_HT_elem = sidewall_HT(id = self.id+'_wall_HT')
203         self.roof_HT_elem = roof_HT(id = self.id+'_roof_HT')
```

```
204         #Sec - water/steam
205         self.steam_HT_elem = steam_HT(id = self.id+'_steam_HT', HX_type = 'radiative_convective_EV')
206 
207         #Account for ingress air if required when connecting nodes and elements
208         if self.mdot_ia != 0:
209             self.pipe_ia.connect_IO(self.node_ia, self.node_pre_fg)
210             self.pipe_pre_fg.connect_IO(self.inlet, self.node_pre_fg)
211             self.pipe_1.connect_IO(self.node_pre_fg, self.outlet)
212         else:
213             self.pipe_1.connect_IO(self.inlet, self.outlet)
214 
215         if self.mdot_ia !=0:
216             self.nodes_1 = [self.node_ia, self.node_pre_fg]
217             self.elements_1 = [self.pipe_ia, self.pipe_pre_fg, self.pipe_1]
218         else:
219             self.nodes_1 = []
220             self.elements_1 = [self.pipe_1]
221 
222         self.HT_elements = [self.re_HT_elem, self.wall_HT_elem, self.roof_HT_elem, self.steam_HT_elem]
223 
224     #For radiative cavity, the linked steam side is the EV element, therefore all HT is to walls and roofs for the steam
side
225class Radiative_Cavity(Element):
226     #Constructor
227     #K_1, secondary loss factor for fg in series
228     def __init__(self, mdot_fa, cp_fa, K_1, id, mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash = 1, **kwargs):
229         Element.__init__(self)
230         self.mdot_fa = mdot_fa
231         self.mdot_ia = mdot_ia
232         self.Y_ia = Y_ia
233         self.T_ia = T_ia
234         self.cp_fa = cp_fa
235         self.K_1 = K_1
236         self.id = id
237         self.high_ash = high_ash
238 
239         self.alpha_conv_i = 0
240         self.alpha_conv_o = 0
241         self.alpha_rad_epsilon_g = 0
242 
243         self.__dict__.update(kwargs)
244         self.geo_inputs = kwargs
245 
246     #Initialise function once input and output streams have been connected to external nodes
247     def initialise(self):
248 
249         #Initialise pipes for both streams
250         #Stream 1 = fg
251       self.pipe_1 = Fluegas_Pipe(self.mdot_fa, self.mdot_ia, self.Y_ia, self.T_ia, self.cp_fa, self.K_1, self.id+'_pipe_1',
\
252             high_ash = self.high_ash, fixed_mdot= 0, **self.geo_inputs)
253 
254         #Specify other properties
255 
256         #Flue gas side
257         #A_open for open duct
258         W_ave = (self.pipe_1.W_i + self.pipe_1.W_e)/2
259         H_ave = (self.pipe_1.H_i + self.pipe_1.H_e)/2
260         self.pipe_1.A_open = W_ave * H_ave
261 
262         #Cavity volume and area for mean beam length - Assume full planar area
263         self.pipe_1.V_cav = self.pipe_1.A_open * self.pipe_1.L_duct
264         self.pipe_1.A_cav = 2 * (W_ave + H_ave) * self.pipe_1.L_duct + 2 * self.pipe_1.A_open
265 
266         #A_ho for external flow area
267         self.pipe_1.A_ho = self.pipe_1.A_wall
268         self.pipe_1.S_mbl = 3.6 * self.pipe_1.V_cav / self.pipe_1.A_cav
269         self.pipe_1.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
270 
271         #Steam side
272         #Internal heat exchanger area = External area
273         self.pipe_1.A_hi = self.pipe_1.A_ho
```

```
274
275         if self.mdot_ia != 0:
276             self.h_ia = gmp.enthalpy(self.Y_ia, self.T_ia, self.inlet.p)
277             self.node_ia = Node('gasmix', self.id+'_node_ia', self.inlet.p , self.h_ia, self.T_ia, p_bc = True, h_bc = True,
Y_bc = True, gasmixture= self.Y_ia)
278             self.pipe_ia = Pipe(0,0,1,1E-6,self.id+'_pipe_ia', fixed_mdot = self.mdot_ia)
279             self.pipe_pre_fg = Pipe(0,0,1,1E-10,self.id+'_pipe_pre_fg')
280             self.node_pre_fg = Node('gasmix', self.id+'_node_pre_fg', self.inlet.p , self.inlet.h, self.inlet.T, gasmixture=
self.inlet.gasmixture)
281
282         #Radiation
283         self.re_HT_elem = direct_radiation(id = self.id+'_re_HT', HX_type = 'radiative_cavity')
284         #Steam side - HT to evaporator
285         self.steam_HT_elem = steam_HT(id = self.id+'_steam_HT', HX_type = 'radiative_cavity')
286
287         #Note steam HT elem is connected to evaporator control volume in the boiler initialisation file
288
289         #Account for ingress air if required when connecting nodes and elements
290         if self.mdot_ia != 0:
291             self.pipe_ia.connect_IO(self.node_ia, self.node_pre_fg)
292             self.pipe_pre_fg.connect_IO(self.inlet, self.node_pre_fg)
293             self.pipe_1.connect_IO(self.node_pre_fg, self.outlet)
294         else:
295             self.pipe_1.connect_IO(self.inlet, self.outlet)
296
297         if self.mdot_ia !=0:
298             self.nodes_1 = [self.node_ia, self.node_pre_fg]
299             self.elements_1 = [self.pipe_ia, self.pipe_pre_fg, self.pipe_1]
300         else:
301             self.nodes_1 = []
302             self.elements_1 = [self.pipe_1]
303         self.HT_elements = [self.re_HT_elem, self.steam_HT_elem]
304
305 class Radiative_Convective_AH(Element):
306     #Constructor
307     #K_1, K_2 = secondary loss factor for each fluid stream in series
308     def __init__(self, mdot_fa, cp_fa, K_1, K_2, id, mdot_ia = 0, Y_ia = {}, T_ia = 0, high_ash = 1, **kwargs):
309         Element.__init__(self)
310         self.mdot_fa = mdot_fa
311         self.mdot_ia = mdot_ia
312         self.Y_ia = Y_ia
313         self.T_ia = T_ia
314         self.cp_fa = cp_fa
315         self.K_1 = K_1
316         self.K_2 = K_2
317         self.id = id
318         self.high_ash = high_ash
319
320         self.alpha_conv_i = 0
321         self.alpha_conv_o = 0
322         self.alpha_rad_epsilon_g = 0
323
324         self.__dict__.update(kwargs)
325         self.geo_inputs = kwargs
326
327     #Initialise function once input and output streams have been connected to external nodes
328     def initialise(self):
329
330         #Initialise pipes for both streams
331         #Stream 1 = fg
332         #Stream 2 = air
333        self.pipe_1 = Fluegas_Pipe(self.mdot_fa, self.mdot_ia, self.Y_ia, self.T_ia, self.cp_fa, self.K_1, self.id+'_pipe_1',
\
334             high_ash = self.high_ash, fixed_mdot= 0, **self.geo_inputs)
335         self.pipe_2 = Pipe(1,0,1,self.K_2, self.id+'_pipe_2', fixed_mdot=0, **self.geo_inputs)
336
337         #Specify other properties
338
339         #Air side
340         #A_open for open duct
341         W_ave = (self.pipe_1.W_i + self.pipe_1.W_e)/2
342         H_ave = (self.pipe_1.H_i + self.pipe_1.H_e)/2
```

```
343         self.pipe_1.A_open = W_ave * H_ave
344         self.pipe_2.A_open = W_ave * H_ave
345 
346         #A_ho for external flow area
347         self.pipe_1.A_ho = np.pi * self.pipe_1.d_OD * (self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t)
348         self.pipe_2.A_ho = np.pi * self.pipe_1.d_OD * (self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t)
349         self.pipe_1.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
350         self.pipe_2.b_l = (self.pipe_1.N_l - 1) * self.pipe_1.S_l
351 
352         #Flue gas side
353         #Internal diameter
354         self.pipe_1.d_ID = self.pipe_1.d_OD - 2 * self.pipe_1.t_wall
355         self.pipe_2.d_ID = self.pipe_1.d_OD - 2 * self.pipe_1.t_wall
356 
357         #Mean beam length
358         self.pipe_1.S_mbl = 0.9 * self.pipe_1.d_ID
359         self.pipe_2.S_mbl = 0.9 * self.pipe_1.d_ID
360 
361         #Free flow internal area
362         self.pipe_1.A_ff = self.pipe_1.N_t * self.pipe_1.pp * np.pi/4 * self.pipe_1.d_ID**2
363         self.pipe_2.A_ff = self.pipe_1.N_t * self.pipe_1.pp * np.pi/4 * self.pipe_1.d_ID**2
364 
365         #Internal heat exchange area
366         self.pipe_1.A_hi = np.pi * self.pipe_1.d_ID * self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t
367         self.pipe_2.A_hi = np.pi * self.pipe_1.d_ID * self.pipe_1.L_avg * self.pipe_1.N_l * self.pipe_1.N_t
368 
369         #Ingress air adjustment
370         if self.mdot_ia != 0:
371             self.h_ia = gmp.enthalpy(self.Y_ia, self.T_ia, self.inlet_1.p)
372              self.node_ia = Node('gasmix', self.id+'_node_ia', self.inlet_1.p , self.h_ia, self.T_ia, p_bc = True, h_bc =
True, Y_bc = True, gasmixture= self.Y_ia)
373             self.pipe_ia = Pipe(0,0,1,1E-6,self.id+'_pipe_ia', fixed_mdot = self.mdot_ia)
374             self.pipe_pre_fg = Pipe(0,0,1,1E-10,self.id+'_pipe_pre_fg')
375               self.node_pre_fg = Node('gasmix', self.id+'_node_pre_fg', self.inlet_1.p , self.inlet_1.h, self.inlet_1.T,
gasmixture= self.inlet_1.gasmixture)
376 
377         self.re_HT_elem = direct_radiation(id = self.id+'_re_HT', HX_type = 'radiative_convective_ah')
378         self.air_HT_elem = air_HT(id = self.id+'_steam_HT')
379 
380         #Account for ingress air if required when connecting nodes and elements
381         if self.mdot_ia != 0:
382             self.pipe_ia.connect_IO(self.node_ia, self.node_pre_fg)
383             self.pipe_pre_fg.connect_IO(self.inlet_1, self.node_pre_fg)
384             self.pipe_1.connect_IO(self.node_pre_fg, self.outlet_1)
385             self.pipe_2.connect_IO(self.inlet_2, self.outlet_2)
386         else:
387             self.pipe_1.connect_IO(self.inlet_1, self.outlet_1)
388             self.pipe_2.connect_IO(self.inlet_2, self.outlet_2)
389 
390         self.air_HT_elem.connect_IO(self.pipe_1, self.pipe_2)
391 
392         if self.mdot_ia !=0:
393             self.nodes_1 = [self.node_ia, self.node_pre_fg]
394             self.elements_1 = [self.pipe_ia, self.pipe_pre_fg, self.pipe_1]
395             self.nodes_2 = []
396             self.elements_2 = [self.pipe_2]
397         else:
398             self.nodes_1 = []
399             self.elements_1 = [self.pipe_1]
400             self.nodes_2 = []
401             self.elements_2 = [self.pipe_2]
402         #Collect HT elements
403         self.HT_elements = [self.re_HT_elem, self.air_HT_elem]
404 
405 #Compound furnace element
406 class Furnace_Compound(Element):
407     def __init__(self, mdot_f, mdot_fa, mdot_fg, f_radloss, A_fur, Vol_fur, xi_fur, x_fur, d_fa, x1_coke, \
408         x2_coke, M_flame, id, high_ash = 1, K = 1E-5, **kwargs):
409         Element.__init__(self)
410         self.mdot_f = mdot_f
411         self.mdot_fa = mdot_fa
412         self.mdot_fg = mdot_fg
```

```
413         self.f_radloss = f_radloss
414         self.A_fur = A_fur
415         self.Vol_fur = Vol_fur
416         self.xi_fur = xi_fur
417         self.x_fur = x_fur
418         self.d_fa = d_fa
419         self.x1_coke = x1_coke
420         self.x2_coke = x2_coke
421         self.M_flame = M_flame
422         self.id = id
423         self.high_ash = high_ash
424         self.K = K
425 
426         self.__dict__.update(kwargs)
427         self.geo_inputs = kwargs
428 
429     def initialise(self):
430         #Object of furnace element
431       self.furnace = Furnace(self.mdot_f, self.mdot_fa, self.mdot_fg, self.f_radloss, self.A_fur, self.Vol_fur, self.xi_fur,
self.x_fur, \
432             self.d_fa, self.x1_coke, self.x2_coke, self.M_flame, self.id+'_furn', high_ash= self.high_ash, \
433                 K = self.K, fixed_mdot=0,  **self.geo_inputs)
434 
435         #Direct radiation and HT to steam
436         self.re_HT_elem = furn_re(self.id)
437         self.steam_HT_elem = furn_st(self.id)
438 
439         self.furnace.connect_IO(self.inlet, self.outlet)
440 
441         self.elements = [self.furnace]
442         self.HT_elements = [self.re_HT_elem, self.steam_HT_elem]
443 
```

A.4 Boiler heat exchanger heat transfer element classes

boiler_heat_transfer.py

```
#import dependencies
import numpy as np
from base_classes import HeatTransfer
import GasMixtureProperties_BICU as gmp
from scipy.optimize import fsolve
import scipy as sp

#Heat transfer elements

#References from BoilerProcessModels.py by Prof Pieter Rousseau
kel = 273.15
rho_ash = 2200
sigma_sb = 5.6704e-8    #[W/m2·K^4]
cp_ash = 710

#Relevant equations to use in heat transfer classes
#high ash fluegas (gas + particles) emissivity
def highash_fluegas_emissivity(fg_pipe, T):
    """
    Calculates high ash fluegas (gas + particles) emissivity.

    Parameters
    ----------
    fg_pipe: Fluegas pipe object

    Returns
    -------
    epsilon_g_p: Fluegas (gas + particle) emissivity

    """
    #WSGGM model coefficients
    fg_pipe.b_0 = np.array([0.13, 0.595, 0.275])
    fg_pipe.b_1 = np.array([0.000265, -0.00015, -0.000115])
    fg_pipe.k_gi = np.array([0.0, 0.824, 25.907])
    fg_pipe.a_i = fg_pipe.b_0 + fg_pipe.b_1*T

    #Particulate area and loading
    fg_pipe.A_p = 3/(2*rho_ash*fg_pipe.d_fa)
    rho_avg = gmp.density(fg_pipe.inlet.gasmixture, T = T, p = fg_pipe.p_avg)
    fg_pipe.L_p = fg_pipe.mdot_fa/(fg_pipe.mdot/rho_avg)

    #Gas emissivity
    X_fg_H2O = fg_pipe.outlet.X['H2O']
    X_fg_CO2 = fg_pipe.outlet.X['CO2']
    fg_pipe.epsilon_gi = fg_pipe.a_i * (1 - np.exp(-fg_pipe.k_gi*((X_fg_H2O+X_fg_CO2) *
fg_pipe.p_avg/1E5)*fg_pipe.S_mbl))

    #Epsilon_g with uncertainty factor
    fg_pipe.epsilon_g = np.sum(fg_pipe.epsilon_gi) * (1 + fg_pipe.alpha_rad_epsilon_g)

    fg_pipe.K_emi_g = -np.log(1-fg_pipe.epsilon_g)/fg_pipe.S_mbl
    fg_pipe.K_coke = 10.2*fg_pipe.x1_coke*fg_pipe.x2_coke * fg_pipe.p_avg/1E6 #MPa

    #Account for particle absorptivity and back scatter - uncertainty factors for each
    Q_abs = 0.275*(fg_pipe.d_fa*1E6)**0.298 - 0.305
    fg_pipe.Q_abs = Q_abs * (1 + fg_pipe.alpha_Q_abs)

    Q_scat = (6.2188E-3 - 1.0492E-2*(fg_pipe.d_fa*1E6) + 7.287E-3*(fg_pipe.d_fa*1E6)**2 - 2.1925E-
5*(fg_pipe.d_fa*1E6)**3) \
             / (1.851E-1 - 2.0405E-3*(fg_pipe.d_fa*1E6)**2 + 6.254E-4*(fg_pipe.d_fa*1E6)**3)
    fg_pipe.Q_scat = Q_scat * (1 + fg_pipe.alpha_Q_scat)
    fg_pipe.gamma = (1 + 2*Q_scat/fg_pipe.Q_abs)**0.5

    #Optical thickness of gas-particulate dispersion
```

```
63        fg_pipe.phi_g_p = (fg_pipe.Q_abs*fg_pipe.A_p*fg_pipe.L_p + fg_pipe.K_emi_g + fg_pipe.K_coke) * fg_pipe.S_mbl *
fg_pipe.gamma
64        beta = (fg_pipe.gamma - 1)/(fg_pipe.gamma + 1)
65        epsilon_g_p = (1 - beta) * (1- np.exp(-fg_pipe.phi_g_p))/(1 + beta * np.exp(-fg_pipe.phi_g_p))
66
67        return epsilon_g_p
68
69    #Calculate bypass direct radiation through tube bank
70    def bypass_direct_radiation(fg_pipe, epsilon_g):
71        #Angular coefficient of incident direct radiation
72        phi = ((fg_pipe.b_l/fg_pipe.S_t)**2 + 1)**0.5 - fg_pipe.b_l/fg_pipe.S_t
73
74        #Bypass direct rad
75        #Supplied beta value ---- need to query
76        Qdot_bp = (fg_pipe.Qdot_ri*(1 - epsilon_g)*phi)/fg_pipe.beta
77
78        return Qdot_bp
79
80    def back_direct_radiation(fg_pipe, epsilon_g):
81        A_e_fg = fg_pipe.W_e * fg_pipe.H_e
82        Qdot_back = 0.5*epsilon_g*sigma_sb*A_e_fg*fg_pipe.T_avg**4
83        return Qdot_back
84
85    #Gnielinski external heat transfer correlations
86    def Gnielinski_external_tube_bank(pipe, T_w):
87        #Void fraction for gas velocity in bank
88        pipe.a_Gni = pipe.S_t/pipe.d_OD
89        pipe.b_Gni = pipe.S_l/pipe.d_OD
90
91        if pipe.b_Gni < 1:
92            psi = 1 - np.pi/(4*pipe.a_Gni*pipe.b_Gni)
93        else:
94            psi = 1 - np.pi/(4*pipe.a_Gni)
95
96        # #Open duct flow area
97        # W_ave = (pipe.W_i + pipe.W_e)/2
98        # H_ave = (pipe.H_i + pipe.H_e)/2
99        # A_open = W_ave * H_ave
100       #Open duct velocity
101       pipe.v_open = pipe.mdot/(pipe.rho_avg * pipe.A_open)
102       pipe.L_Gni = np.pi/2 * pipe.d_OD
103
104       #Nusselt number - Laminar & turbulent
105       pipe.Re_o = (pipe.rho_avg * pipe.v_open * pipe.L_Gni)/(psi * pipe.mu_avg) *np.sin(pipe.theta*np.pi/180)
106       pipe.Pr_o = (pipe.cp_avg * pipe.mu_avg)/pipe.k_avg
107       pipe.Nu_l_turb_o = (0.037 * pipe.Re_o**0.8 * pipe.Pr_o)/(1 + 2.443 * pipe.Re_o**-0.1 * (pipe.Pr_o**(2/3) - 1))
108       pipe.Nu_l_lam_o = 0.664 * np.sqrt(pipe.Re_o) * pipe.Pr_o**(1/3)
109       pipe.Nu_l0_o = 0.3 + np.sqrt(pipe.Nu_l_lam_o**2 + pipe.Nu_l_turb_o**2)
110
111       #Bundle correction
112       if pipe.inline == 1:
113           pipe.f_A = 1 + (0.7 * (pipe.b_Gni/pipe.a_Gni - 0.3))/(psi**1.5 * (pipe.b_Gni/pipe.a_Gni + 0.7)**2)
114       else:
115           pipe.f_A = 1 + 2/(3*pipe.b_Gni)
116       pipe.Nu_0_bundle_o = pipe.Nu_l0_o * (1+(pipe.N_l - 1)*pipe.f_A)/pipe.N_l
117       # pipe.Nu_0_bundle_o = pipe.f_A * pipe.Nu_l0_o
118       #Boundary layer correction
119       pipe.Nu_bundle_o = pipe.Nu_0_bundle_o * (pipe.T_avg/T_w)**0.12
120
121       #Uncertainty factor correction
122       pipe.Nu_bundle_o = pipe.Nu_bundle_o * (1 + pipe.alpha_conv_o)
123       pipe.htc_conv_o = pipe.Nu_bundle_o*pipe.k_avg/pipe.L_Gni
124
125       return pipe.htc_conv_o
126
127   #Gnielinski internal heat transfer correlations
128   def Gnielinski_internal_tube(pipe):
129       #Velocity of fluid in tube (using free flow area of all tubes)
130       pipe.v_i = pipe.mdot/(pipe.rho_avg*pipe.A_ff)
131
132       pipe.Re_i = (pipe.rho_avg * pipe.v_i * pipe.d_ID)/pipe.mu_avg
133       pipe.Pr_i = (pipe.cp_avg * pipe.mu_avg)/pipe.k_avg
```

```
134
135     #Friction factor
136     pipe.f = (0.79 * np.log(pipe.Re_i) - 1.64) **-2
137     #Nusselt number and uncertainty factor
138     pipe.Nu_i = ((pipe.f/8) * (pipe.Re_i - 1000) * pipe.Pr_i)/(1 + 12.7 * (pipe.f/8)**0.5 * (pipe.Pr_i**(2/3) - 1))
139     pipe.Nu_i = pipe.Nu_i * (1 + pipe.alpha_conv_i)
140
141     pipe.htc_conv_i = pipe.Nu_i*pipe.k_avg/pipe.d_ID
142
143     return pipe.htc_conv_i
144
145 #Calcuate radiative heat transfer coefficient from heat flux
146 def effective_radiation_htc(fg_pipe, epsilon_g, T_w):
147     q_fg_rad = (fg_pipe.epsilon_w + 1)/2 * sigma_sb * epsilon_g * (fg_pipe.T_avg**4 - T_w**4)
148     fg_pipe.htc_rad = q_fg_rad/(fg_pipe.T_avg - T_w)
149     return fg_pipe.htc_rad
150
151 #Gurvich projected method - solved using fsolve
152 def fet_gurvich(T_fe, furnace):
153     T_fe = T_fe[0]
154
155     furnace.epsilon_g = highash_fluegas_emissivity(furnace, T_fe)
156     furnace.epsilon_eff_fur = 1/(1+furnace.psi_eff_fur*(1/furnace.epsilon_g -1))
157
158     #Boltzmann number calculation
159     furnace.h_fg_fe = gmp.enthalpy(furnace.inlet.gasmixture, T = T_fe, p = furnace.inlet.p)
160     furnace.h_fa_aft = cp_ash*(furnace.inlet.T - (25+kel))
161     furnace.h_fa_fet = cp_ash*(T_fe - (25+kel))
162     furnace.VC = ((furnace.inlet.h - furnace.h_fg_fe)*furnace.mdot + furnace.mdot_fa*(furnace.h_fa_aft -\
163                     furnace.h_fa_fet))/(furnace.mdot_f*(furnace.inlet.T - T_fe))
164     furnace.phi_fur = 1 - furnace.f_radloss
165     furnace.Bo = furnace.phi_fur*furnace.mdot_f*furnace.VC /
(sigma_sb*furnace.psi_eff_fur*furnace.A_rad_fur*furnace.inlet.T**3)
166
167     eq = T_fe - furnace.inlet.T/(furnace.M_flame*(furnace.epsilon_eff_fur/furnace.Bo)**0.6 + 1)
168     return eq
169
170 #Direct radiation heat transfer
171 class direct_radiation(HeatTransfer):
172     def __init__(self, Qdot_bp = 0, Qdot_back = 0, id = 'dir_rad', HX_type = 'radiative_convective'):
173         HeatTransfer.__init__(self)
174         self.Qdot_bp = Qdot_bp
175         self.Qdot_back = Qdot_back
176         self.id = id
177         self.HX_type = HX_type
178
179     #Solved each steady state iteration
180     def comp_char(self):
181         #Flue gas + particulate emissivity
182         self.epsilon_g = highash_fluegas_emissivity(self.inlet, self.inlet.T_avg)
183
184         #Bypass radiation
185         #Bypass for cavity
186         if self.HX_type == 'radiative_cavity':
187             try:
188                 self.Qdot_bp = self.inlet.Qdot_ri * (1 - self.epsilon_g)
189             except:
190                 self.inlet.Qdot_ri = 0
191                 self.Qdot_bp = self.inlet.Qdot_ri * (1 - self.epsilon_g)
192         #No bypass for airheater
193         elif self.HX_type == 'radiative_convective_ah':
194             self.Qdot_bp = 0
195         else:
196             try:
197                 self.Qdot_bp = bypass_direct_radiation(self.inlet, self.epsilon_g)
198                 self.Qdot_bp = self.Qdot_bp * (1 + self.inlet.beta_fg_bp)
199             except:
200                 self.inlet.Qdot_ri = 0
201                 self.Qdot_bp = bypass_direct_radiation(self.inlet, self.epsilon_g)
202         #Back radiation
203         #No back radiation for airheater
204         if self.HX_type == 'radiative_convective_ah':
```

```
205             self.Qdot_back = 0
206         else:
207             self.Qdot_back = back_direct_radiation(self.inlet, self.epsilon_g)
208 
209         #Total outlet (outgoing) direct radiation
210         self.Qdot_re = self.Qdot_bp + self.Qdot_back
211 
212     def Qdot_fn_inlet(self):
213 
214         self.inlet.Qdot -= self.Qdot_re
215 
216         #Set outlet radiation, bypass, and back radiation for inlet HX
217         self.inlet.Qdot_re = self.Qdot_re
218         self.inlet.Qdot_bp_out = self.Qdot_bp
219         self.inlet.Qdot_back_out = self.Qdot_back
220 
221     def Qdot_fn_outlet(self):
222 
223         self.outlet.Qdot += self.Qdot_re
224 
225         #Set incoming direct radiation, bypass, and back radiation for outlet HX
226         self.outlet.Qdot_ri = self.Qdot_re
227         self.outlet.Qdot_bp_in = self.Qdot_bp
228         self.outlet.Qdot_back_in = self.Qdot_back
229 
230 #Water wall heat transfer
231 class sidewall_HT(HeatTransfer):
232     def __init__(self, Qdot = 0, id = 'sidewall'):
233         HeatTransfer.__init__(self)
234         self.Qdot = Qdot
235         self.id = id
236     #Solve each steady state iteration
237         #htc wall informed from steam_HT heat transfer class
238     def comp_char(self):
239         try:
240             self.htc_wall_conv = self.inlet.htc_conv_fg * (1 + self.inlet.beta_wall_conv)
241             self.htc_wall_rad = self.inlet.htc_rad_fg * (1 + self.inlet.beta_wall_rad)
242             self.htc_wall = self.htc_wall_conv + self.htc_wall_rad
243             self.Qdot_wall = self.htc_wall * self.inlet.A_wall * (self.inlet.T_avg - self.inlet.T_wall)
244         except:
245             #If htc_fg hasn't been initialised yet, set to default value
246             self.htc_wall = 100
247             self.Qdot_wall = self.htc_wall * self.inlet.A_wall * (self.inlet.T_avg - self.inlet.T_wall)
248 
249     def Qdot_fn_inlet(self):
250         self.inlet.Qdot -= self.Qdot_wall
251         self.inlet.Qdot_wall_out = self.Qdot_wall
252 
253     def Qdot_fn_outlet(self):
254 
255         self.outlet.Qdot += self.Qdot_wall
256 
257 #Water wall - roof - heat transfer
258 class roof_HT(HeatTransfer):
259     def __init__(self, Qdot = 0, id = 'roof'):
260         HeatTransfer.__init__(self)
261         self.Qdot = Qdot
262         self.id = id
263 
264     #Solve each steady state iteration
265     def comp_char(self):
266         try:
267             self.htc_roof_conv = self.inlet.htc_conv_fg * (1 + self.inlet.beta_wall_conv)
268             self.htc_roof_rad = self.inlet.htc_rad_fg * (1 + self.inlet.beta_wall_rad)
269             self.htc_roof = self.htc_roof_conv + self.htc_roof_rad
270             self.Qdot_roof = self.htc_roof * self.inlet.A_roof * (self.inlet.T_avg - self.inlet.T_roof)
271         except:
272             #If htc_fg hasn't been initialised yet, set to default value
273             self.htc_roof = 100
274             self.Qdot_roof = self.htc_roof * self.inlet.A_roof * (self.inlet.T_avg - self.inlet.T_roof)
275 
276     def Qdot_fn_inlet(self):
```

```
277
278         self.inlet.Qdot -= self.Qdot_roof
279         self.inlet.Qdot_roof_out = self.Qdot_roof
280
281     def Qdot_fn_outlet(self):
282
283         self.outlet.Qdot += self.Qdot_roof
284
285 #Heat transfer to secondary fluid (sec)
286 class steam_HT(HeatTransfer):
287     def __init__(self, Qdot = 0, id = 'steam_HT', HX_type = 'radiative_convective'):
288         HeatTransfer.__init__(self)
289         self.Qdot = Qdot
290         self.id = id
291         self.HX_type = HX_type
292
293     #Solve each steady state iteration
294     def comp_char(self):
295         #If not initialised yet, initialise Qdot abs
296         try:
297             self.Qdot_abs = self.inlet.Qdot_ri - self.inlet.Qdot_bp_out
298         except:
299             self.inlet.Qdot_ri = 0
300             self.inlet.Qdot_bp_out= 0
301             self.Qdot_abs = self.inlet.Qdot_ri - self.inlet.Qdot_bp_out
302
303         #Guess temperature for T_fo - outer wall fouling temperature
304         T_fo = (self.inlet.T_avg + self.outlet.T_avg)/2
305
306         #Balance function to solve for wall temperatures
307         def balance_eqns(x):
308             self.T_fo = x[0]
309
310             #Flue gas + particulate emissivity
311             self.epsilon_g = highash_fluegas_emissivity(self.inlet, self.inlet.T_avg)
312
313             #Convective and radiative flue gas heat transfer coefficient (htc)
314             #Assume convective heat transfer negligible for radiative cavities
315             if self.HX_type == 'radiative_cavity':
316                 self.htc_conv_fg = 0
317             else:
318                 self.htc_conv_fg = Gnielinski_external_tube_bank(self.inlet, self.T_fo)
319             #Radiative heat transfer
320             self.htc_rad_fg = effective_radiation_htc(self.inlet, self.epsilon_g, self.T_fo)
321
322             #Convective steam htc
323             if self.HX_type == 'radiative_convective':
324                 self.htc_st = Gnielinski_internal_tube(self.outlet)
325             #For boiling, assume negligible resistance for water boiling
326             elif self.HX_type == 'radiative_convective_EV' or self.HX_type == 'radiative_cavity':
327                 self.htc_st = 1E30
328
329             #Adjust heat transfer coefficients using tuning coefficients as required
330             self.htc_conv_fg_adj = self.htc_conv_fg * (1 + self.inlet.beta_tube_conv)
331             self.htc_rad_fg_adj = self.htc_rad_fg * (1 + self.inlet.beta_tube_rad)
332
333             #Platen external area correction for combined flue gas htc
334             if self.inlet.platen == 1:
335                 self.htc_fg = ((np.pi*self.inlet.d_OD)/(2*self.inlet.S_l) * self.htc_conv_fg_adj + self.htc_rad_fg_adj)
\
336                 * (1 + self.inlet.beta_fg_tot)
337
338             else:
339                 self.htc_fg = (self.htc_conv_fg_adj + self.htc_rad_fg_adj) * (1 + self.inlet.beta_fg_tot)
340
341             #Overall external (fg) heat transfer coeff - accounting for surface efficiency
342             self.psi = self.inlet.psi * (1 + self.inlet.alpha_psi)
343             self.eta_o = 1 - self.inlet.AfoverA_o*(1-self.inlet.eta_fin_o)
344             self.UA_fg = self.psi/(1/(self.eta_o*self.htc_fg*self.inlet.A_ho))
345             self.UA_fg = self.UA_fg * (1 + self.inlet.beta_UA_fg)
346             #eff-NTU method - use fg as limiting side
347             self.NTU_fg = self.UA_fg/(self.inlet.mdot*self.inlet.cp_avg)
```

```
348             self.epsilon_hxfg = 1 - np.exp(-self.NTU_fg)
349 
350             #Overall internal (st) heat transfer coefficient - accounting for surface efficiency
351             self.eta_i = 1 - self.inlet.AfoverA_i*(1 - self.inlet.eta_fin_i)
352             self.R_fi = self.inlet.R_dep_i * (1 + self.inlet.alpha_R_fi)
353             self.R_fo = self.inlet.R_dep_o * (1 + self.inlet.alpha_R_fo)
354 
355             #Wall thermal resistance
356             #Cavity resistance is just tube thickness
357             if self.HX_type == 'radiative_cavity':
358                 self.R_w = self.inlet.t_wall / (self.inlet.k_tube * self.inlet.A_ho)
359             #Tube wall thickness
360             else:
361                 self.R_w = (np.log(self.inlet.d_OD/self.inlet.d_ID))/(2 * np.pi * self.inlet.k_tube * self.inlet.L_avg *
\
362                     self.inlet.N_l * self.inlet.N_t)
363 
364             self.UA_st = self.psi/(1/(self.eta_i*self.htc_st*self.inlet.A_hi) + self.R_fi/(self.eta_i*self.inlet.A_hi) +
self.R_w + \
365                 self.R_fo/(self.eta_o*self.inlet.A_ho))
366 
367             if self.HX_type == 'radiative_convective':
368                 #eff-NTU method - use st as Cmin
369                 self.NTU_st = self.UA_st/(self.outlet.mdot*self.outlet.cp_avg)
370                 self.epsilon_hxst = 1 - np.exp(-self.NTU_st)
371                 self.Qdot_st = self.epsilon_hxst * self.outlet.mdot * self.outlet.cp_avg * (self.T_fo -
self.outlet.inlet.T)
372             elif self.HX_type == 'radiative_convective_EV':
373                 self.Qdot_st = self.UA_st * (self.T_fo - self.outlet.T_avg)
374 
375             elif self.HX_type == 'radiative_cavity':
376                 self.Qdot_wall = self.UA_st * (self.T_fo - self.outlet.T_avg)
377                 self.Qdot_st = 0
378 
379             #Heat transfer rates
380             self.Qdot_ex = self.epsilon_hxfg * self.inlet.mdot * self.inlet.cp_avg * (self.inlet.inlet.T - self.T_fo)
381 
382             if self.HX_type == 'radiative_cavity':
383                 eq = self.Qdot_wall - (self.Qdot_ex + self.Qdot_abs)
384             else:
385                 eq = self.Qdot_st - (self.Qdot_ex + self.Qdot_abs)
386 
387             return eq
388 
389         #fsolve to iteratively heat transfer balance equations
390         self.T_fo = fsolve(balance_eqns, T_fo)[0]
391 
392         #Calculate required wall temperatures
393         if self.HX_type == 'radiative_cavity':
394             self.T_wo = self.T_fo - self.R_fo/(self.eta_o * self.inlet.A_ho) * self.Qdot_wall
395             self.T_wi = self.T_wo - self.R_w*self.Qdot_wall
396             self.T_fi = self.T_wi - self.R_fi/(self.eta_i * self.inlet.A_hi) * self.Qdot_wall
397 
398         else:
399             self.T_wo = self.T_fo - self.R_fo/(self.eta_o * self.inlet.A_ho) * self.Qdot_st
400             self.T_wi = self.T_wo - self.R_w*self.Qdot_st
401             self.T_fi = self.T_wi - self.R_fi/(self.eta_i * self.inlet.A_hi) * self.Qdot_st
402 
403         self.inlet.T_fo = self.T_fo
404         self.inlet.T_wo = self.T_wo
405         self.inlet.T_wi = self.T_wi
406         self.inlet.T_fi = self.T_fi
407 
408         self.outlet.T_fo = self.T_fo
409         self.outlet.T_wo = self.T_wo
410         self.outlet.T_wi = self.T_wi
411         self.outlet.T_fi = self.T_fi
412 
413         #Assign heat transfer coefficients
414         self.inlet.htc_fg = self.htc_fg
415         self.inlet.htc_conv_fg = self.htc_conv_fg
416         self.inlet.htc_rad_fg = self.htc_rad_fg
```

```
417         self.inlet.htc_conv_fg_adj = self.htc_conv_fg_adj
418         self.inlet.htc_rad_fg_adj = self.htc_rad_fg_adj
419 
420     #Qdot update functions for heat transfer elements
421     def Qdot_fn_inlet(self):
422         self.inlet.Qdot -= self.Qdot_ex + self.Qdot_abs
423         self.inlet.Qdot_st_out = self.Qdot_st
424         self.inlet.Qdot_abs_out = self.Qdot_abs
425         self.inlet.Qdot_ex_out = self.Qdot_ex
426         self.inlet.Qdot_a_out = 0
427         if self.HX_type == 'radiative_cavity':
428             self.inlet.Qdot_wall_out = self.Qdot_wall
429 
430     def Qdot_fn_outlet(self):
431         self.outlet.Qdot += self.Qdot_st
432         if self.HX_type == 'radiative_cavity':
433             self.outlet.Qdot += self.Qdot_wall
434         self.outlet.Qdot_st_in = self.Qdot_st
435         self.outlet.Qdot_abs_in = self.Qdot_abs
436         self.outlet.Qdot_ex_in = self.Qdot_ex
437 
438 #Furnace direct radiation heat transfer
439 class furn_re(HeatTransfer):
440     def __init__(self, Qdot = 0, id = 'furn_re'):
441         HeatTransfer.__init__(self)
442         self.id = id
443 
444     #Update each steady state iteration
445     def comp_char(self):
446         #Solve Gurvich method (projected) using fsolve for iteration
447         T_fe_guess = 0.7*self.inlet.inlet.T
448         self.T_fe = fsolve(fet_gurvich, T_fe_guess, args = (self.inlet))[0]
449         self.Qdot_fg = self.inlet.mdot * (self.inlet.inlet.h - self.inlet.h_fg_fe) +  \
450             self.inlet.mdot_fa * (self.inlet.h_fa_aft - self.inlet.h_fa_fet)
451         self.Qdot_fur = self.inlet.phi_fur * self.Qdot_fg
452         self.Qdot_loss_fur = (1 - self.inlet.phi_fur) * self.Qdot_fg
453         self.q_fur = self.Qdot_fur/(self.inlet.A_rad_fur * self.inlet.psi_eff_fur)
454 
455         self.eta_fe = non_uniformity_factor(self.inlet.NUF_fuel_type, 1.0) * (1 + self.inlet.alpha_nuf_coef)
456         self.beta_fe = reradiation_coeffecient(self.inlet.RC_fuel_type, self.T_fe, self.inlet.alpha_reradiation_coef)
457         self.Qdot_fe = self.beta_fe * self.eta_fe * (self.inlet.psi_fur['furnaceexit'] *
self.inlet.A_fur['furnaceexit']) * self.q_fur
458 
459         self.Qdot_re = self.Qdot_fe
460         self.inlet.outlet.h = self.inlet.h_fg_fe
461 
462     #Qdot update functions for heat transfer elements
463     def Qdot_fn_inlet(self):
464         self.inlet.Qdot -= self.Qdot_re + self.Qdot_loss_fur
465         self.inlet.Qdot_loss_fur = self.Qdot_loss_fur
466         self.inlet.Qdot_re = self.Qdot_re
467 
468     def Qdot_fn_outlet(self):
469         self.outlet.Qdot += self.Qdot_re
470         self.outlet.Qdot_ri = self.Qdot_re
471 
472 #Furnace water wall heat transfer
473 class furn_st(HeatTransfer):
474     def __init__(self, Qdot = 0, id = 'furn_st'):
475         HeatTransfer.__init__(self)
476         self.id = id
477 
478     def comp_char(self):
479         T_fe_guess = 0.7*self.inlet.inlet.T
480         self.T_fe = fsolve(fet_gurvich, T_fe_guess, args = (self.inlet))[0]
481 
482         self.Qdot_fg = self.inlet.mdot * (self.inlet.inlet.h - self.inlet.h_fg_fe) +  \
483             self.inlet.mdot_fa * (self.inlet.h_fa_aft - self.inlet.h_fa_fet)
484         self.Qdot_fur = self.inlet.phi_fur * self.Qdot_fg
485         self.q_fur = self.Qdot_fur/(self.inlet.A_rad_fur * self.inlet.psi_eff_fur)
486 
487         self.eta_fe = non_uniformity_factor(self.inlet.NUF_fuel_type, 1.0) * (1 + self.inlet.alpha_nuf_coef)
```

```
488          self.beta_fe = reradiation_coeffecient(self.inlet.RC_fuel_type, self.T_fe, self.inlet.alpha_reradiation_coef)
489          self.Qdot_fe = self.beta_fe * self.eta_fe * (self.inlet.psi_fur['furnaceexit'] *
self.inlet.A_fur['furnaceexit']) * self.q_fur
490
491          self.eta_rf = non_uniformity_factor(self.inlet.NUF_fuel_type, 0.0)
492          self.Qdot_rf = self.eta_rf*(self.inlet.psi_fur['refractory']*self.inlet.A_fur['refractory'])*self.q_fur
493
494          self.Qdot_ww = self.Qdot_fur - (self.Qdot_fe + self.Qdot_rf)
495
496          self.inlet.outlet.h = self.inlet.h_fg_fe
497
498          self.Qdot_st = self.Qdot_ww + self.Qdot_rf
499
500      def Qdot_fn_inlet(self):
501
502          self.inlet.Qdot -= self.Qdot_st
503          self.inlet.Qdot_st_out = self.Qdot_st
504
505      def Qdot_fn_outlet(self):
506
507          self.outlet.Qdot += self.Qdot_st
508
```

A.5 Combustion and furnace classes

Combustion class:

```
#import dependencies
import numpy as np
import CoolProp.CoolProp as CP
from base_classes import Element
import GasMixtureProperties as gmp
from scipy.optimize import fsolve

#Combustion element class
#References from BoilerProcessModels.py by Prof Pieter Rousseau

class Combustion(Element):
    #Constructor
    def __init__(self, Y_f, mdot_f, T_f, T_da, p_fur, Y_fa, f_pa, f_sa, f_da, HHV, id, alpha = 1.0, w_air = 0.0, f_ia = 0.0, \
                 cp_f = 710, cp_ash = 710):
        Element.__init__(self)
        self.Y_f = Y_f
        self.mdot_f = mdot_f
        self.id = id
        self.alpha = alpha
        self.w_air = w_air
        self.Y_fa = Y_fa
        self.f_ia = f_ia
        self.T_f = T_f
        self.T_da = T_da
        self.p_fur = p_fur
        self.f_pa = f_pa
        self.f_sa = f_sa
        self.f_da = f_da
        self.HHV = HHV

        self.cp_f = cp_f
        self.cp_ash = cp_ash
        self.K = 1E-2
        self.A = 1
        self.id = id

    #Initialise function is called on creation of combustion element
    def initialise(self):
        #Assign molar mass dict
        self.M = {'C':12.011e-3, 'H':1.00795e-3, 'O':15.9995e-3, 'N':14.0065e-3, 'S':32.065e-3, 'H2O':CP.PropsSI('M','H2O'), \
        'CO2':CP.PropsSI('M','CO2'), 'N2':CP.PropsSI('M','N2'), 'SO2':CP.PropsSI('M','SO2'), 'O2':CP.PropsSI('M','O2')} #[kg/mol]

        #Combustion element is only for fuels with the below composition
        self.M_fuel_comp = {'C':12.011e-3, 'H':1.00795e-3, 'O':15.9995e-3, 'N':14.0065e-3, 'S':32.065e-3, 'H2O':CP.PropsSI('M','H2O'), \
        'UC':12.011e-3}   #[kg/mol]

        #Gas composition molar masses
        self.M_gas_comp = {'H2O':CP.PropsSI('M','H2O'), 'CO2':CP.PropsSI('M','CO2'), 'N2':CP.PropsSI('M','N2'), \
             'SO2':CP.PropsSI('M','SO2'), 'O2':CP.PropsSI('M','O2')}  #[kg/mol]

        #Dry air molar mass
        M_air_dry = 0.2101*self.M['O2']+(1-0.2101)*self.M['N2'] #[kg/mol]

        #Molar amounts of reactants in fuel mix [mol/kg_f] (no ash)
        self.N_f = {}
        N_f_tot = 0.0
        for key in self.M_fuel_comp:
            self.N_f[key] = self.Y_f[key]/self.M_fuel_comp[key]
            N_f_tot += self.N_f[key]

        #Molar amounts of reactants in combustion air [mol/kg_f]
        self.N_ca = {}
```

```
63            N_st_O2 = self.N_f['C'] + 0.25*self.N_f['H'] - 0.5*self.N_f['O'] + self.N_f['S']
64            N_st_N2 = 3.7619 * N_st_O2
65            self.N_ca['O2'] = self.alpha*N_st_O2
66            self.N_ca['N2'] = self.alpha*N_st_N2
67            self.N_ca['H2O'] = 4.7619*self.alpha*self.w_air*(M_air_dry/self.M_fuel_comp['H2O'])*N_st_O2
68            N_ca_tot = self.N_ca['O2']+self.N_ca['N2']+ self.N_ca['H2O']
69
70            #Molar amounts of products in flue gas [mol/kg_f]
71            self.N_fg = {}
72            self.N_fg['CO2'] = self.N_f['C']
73            self.N_fg['N2'] = 0.5*self.N_f['N'] + self.alpha*N_st_N2
74            self.N_fg['SO2'] = self.N_f['S']
75            self.N_fg['H2O'] = self.N_f['H2O'] + 0.5*self.N_f['H'] + 4.7619*self.alpha*self.w_air \
76                *(M_air_dry/self.M_fuel_comp['H2O'])*N_st_O2
77            self.N_fg['O2'] = (self.alpha - 1)*N_st_O2
78            N_fg_tot = self.N_fg['CO2']+self.N_fg['N2']+self.N_fg['SO2']+self.N_fg['H2O']+self.N_fg['O2']
79
80            #Mol fractions for combustion air [mol/mol_ca]
81            self.X_ca = {}
82            self.X_ca['O2'] = self.N_ca['O2']/N_ca_tot
83            self.X_ca['N2'] = self.N_ca['N2']/N_ca_tot
84            self.X_ca['H2O'] = self.N_ca['H2O']/N_ca_tot
85            X_ca_tot = self.X_ca['O2']+self.X_ca['N2']+self.X_ca['H2O']
86            if abs(X_ca_tot - 1) > 1E-3:
87                err_msg = 'Sum of combustion air mol fractions {0:.4e}'.format(X_ca_tot)
88                raise Exception(err_msg)
89
90            #Molecular weight of combustion air [kg/mol_ca]
91            M_ca = self.X_ca['O2']*self.M_gas_comp['O2'] + self.X_ca['N2']*self.M_gas_comp['N2'] +
self.X_ca['H2O']*self.M_gas_comp['H2O']
92
93            #Mol fractions for flue gas [mol/mol_fg]
94            self.X_fg = {}
95            X_fg_tot = 0.0
96            for key in self.M_gas_comp:
97                self.X_fg[key] = self.N_fg[key]/N_fg_tot
98                X_fg_tot += self.X_fg[key]
99            if abs(X_fg_tot - 1) > 1E-3:
100               err_msg = 'Sum of flue gas mol fractions {0:.4e}'.format(X_fg_tot)
101               raise Exception(err_msg)
102
103           #Molecular weight of flue gas [kg/mol_fg]
104           M_fg = 0.0
105           for key in self.M_gas_comp:
106               M_fg += self.X_fg[key]*self.M_gas_comp[key]
107
108           #Mass fractions of combustion air [kg/kg_ca]
109           self.Y_ca = {}
110           Y_ca_tot = 0.0
111           for key in self.X_ca:
112               self.Y_ca[key] = self.X_ca[key] * self.M_gas_comp[key]/M_ca
113               Y_ca_tot += self.Y_ca[key]
114           if abs(Y_ca_tot - 1) > 1E-3:
115               err_msg = 'Sum of combustion air mass fractions {0:.4e}'.format(Y_ca_tot)
116               raise Exception(err_msg)
117
118           #Mass fractions of flue gas [kg/kg_fg]
119           self.Y_fg = {}
120           Y_fg_tot = 0.0
121           for key in self.X_fg:
122               self.Y_fg[key] = self.X_fg[key] * self.M_gas_comp[key]/M_fg
123               Y_fg_tot += self.Y_fg[key]
124               if abs(Y_ca_tot - 1) > 1E-3:
125                   err_msg = 'Sum of flue gas mass fractions {0:.4e}'.format(Y_fg_tot)
126                   raise Exception(err_msg)
127
128           #Mass flow rate calcs
129           TAR = N_st_O2*self.M_gas_comp['O2'] + N_st_N2*self.M_gas_comp['N2']
130           DAR = self.N_ca['O2']*self.M_gas_comp['O2']+self.N_ca['N2']*self.M_gas_comp['N2']
131           HAR = self.N_ca['O2']*self.M_gas_comp['O2']+self.N_ca['N2']*self.M_gas_comp['N2'] \
132               +self.N_ca['H2O']*self.M_gas_comp['H2O']
133           FGR = self.N_fg['O2']*self.M_gas_comp['O2']+self.N_fg['N2']*self.M_gas_comp['N2'] \
```

```
134             +self.N_fg['H2O']*self.M_gas_comp['H2O']+self.N_fg['CO2']*self.M_gas_comp['CO2'] \
135             +self.N_fg['SO2']*self.M_gas_comp['SO2']
136 
137         self.mdot_ca = HAR*self.mdot_f
138         self.mdot_pa = self.f_pa*self.mdot_ca
139         self.mdot_sa = self.f_sa*self.mdot_ca
140         self.mdot_da = self.f_da*self.mdot_ca
141         self.mdot_fg = FGR*self.mdot_f
142         self.mdot_fa = self.Y_fa*(self.Y_f['ASH']+self.Y_f['UC'])*self.mdot_f
143         self.mdot_ba = (1 - self.Y_fa)*(self.Y_f['ASH']+self.Y_f['UC'])*self.mdot_f
144         self.mdot_ia = self.f_ia * HAR * self.mdot_f
145 
146     #Component characteristics - updated every steady state iteration
147     def comp_char(self):
148         # Set mdot to pa+sa since upstream air side is dependent on this mdot value
149         self.mdot = self.mdot_pa+self.mdot_sa
150         #Average fluid properties from upstream and downstream nodes
151         self.p_avg = (self.inlet.p + self.outlet.p)/2
152         self.T_avg = (self.inlet.T + self.outlet.T)/2
153 
154         #If CoolProp BICU interpolation is used, update fluid properties as such
155         if type(self.outlet.fluid) == CP.AbstractState:
156             self.h_avg = gmp.enthalpy(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X, self.outlet.BICU)
157             self.rho_avg = gmp.density(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.M_M,
self.outlet.BICU)
158             self.cp_avg = gmp.specific_heat_pressure(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X,
self.outlet.BICU)
159             self.mu_avg = gmp.viscosity(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X, self.outlet.BICU)
160             self.k_avg = gmp.conductivity(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X,
self.outlet.BICU)
161         #For regular CoolProp interface (slower)
162         else:
163             self.h_avg = gmp.enthalpy(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X)
164             self.rho_avg = gmp.density(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.M_M)
165             self.cp_avg = gmp.specific_heat_pressure(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X)
166             self.mu_avg = gmp.viscosity(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X)
167             self.k_avg = gmp.conductivity(self.outlet.gasmixture, self.T_avg, self.p_avg, self.outlet.X)
168 
169         #Pressure drop, mass flow rate, pressure component characteristics
170         self.dp = self.outlet.p - self.inlet.p
171         self.C_l = self.K/(2 * self.rho_avg * self.A**2)
172         self.mdot = -self.dp/np.abs(-self.dp) * np.sqrt(np.abs(-self.dp)/np.abs(self.C_l))
173 
174         self.A_1 = -self.C_l * np.abs(self.mdot)
175         self.A_0 = self.dp - self.A_1*self.mdot
176 
177         kel = 273.15
178         HHV_C = 32.763E6 #[J/kg]
179         self.h_f =  self.cp_f*(self.T_f-(25+kel))
180         self.h_da = gmp.enthalpy(self.Y_ca, T=self.T_da, p = self.p_fur)
181         self.outlet.gasmixture = self.Y_fg
182 
183         #Adiabatic flame temperature calculation
184         def energy_balance(T_aft):
185             h_fg_aft = gmp.enthalpy(self.Y_fg, T = T_aft, p = self.p_fur)
186             h_fa_aft = self.cp_ash*(T_aft - (25+kel))
187 
188             #AFT Energy balance
189             #Products
190             RHS = self.mdot_fg*h_fg_aft+self.mdot_fa*h_fa_aft
191             #Reactants
192             LHS = self.mdot_f*(self.h_f + self.HHV - self.Y_f['UC']*HHV_C) + self.mdot_pa*self.inlet.h +
self.mdot_sa*self.inlet.h \
193                 + self.mdot_da*self.h_da
194             eq = RHS - LHS
195             return eq
196 
197         self.T_aft = fsolve(energy_balance, 1500 + kel)[0]
198 
199         self.h_fg_aft = gmp.enthalpy(self.Y_fg, T = self.T_aft, p = self.p_fur)
200         self.h_fa_aft = self.cp_ash*(self.T_aft - (25 + kel))
201 
```

```
202         #Set outlet node enthalpy to enthalpy at adiabatic flame temperature
203         self.outlet.h = self.h_fg_aft
204 
205         self.energy_sources = 0
206 
```

Furnace class:

```
1   #import dependencies
2   import numpy as np
3   import CoolProp.CoolProp as CP
4   from base_classes import Element
5   import GasMixtureProperties as gmp
6   from scipy.optimize import fsolve
7   import scipy as sp
8   
9   #Furnace model - projected (Gurvich)
10  #References from BoilerProcessModels.py by Prof Pieter Rousseau
11  kel = 273.15
12  rho_ash = 2200
13  sigma_sb = 5.6704e-8    #[W/m2·K^4]
14  cp_ash = 710
15  
16  #Furnace factors for various fuels and conditions
17  wall_fouling_factor = {'gas_oil':0.65, 'heavy_oil':0.55, 'coal':0.4, 'grate_fuels':0.6, \
18                         'other_fuels':0.2, 'open_pitch':0.2, 'refractory':0.1}
19  A_prime = {'gas_fuel_oil':0.54, 'high_volatile_coal':0.59, 'low_reactive_coal':0.56, 'grate_stoker':0.59}
20  B_prime = {'gas_fuel_oil':0.2, 'high_volatile_coal':0.5, 'low_reactive_coal':0.5, 'grate_stoker':0.5}
21  dX = {'horizontal':0.0, 'tilting_horizontal_pos':0.1, 'tilting_horizontal_neg':-0.1, 'front_opposing_walls':0.075}
22  d_fa = {'tubular_ball_mill':13e-6, 'hammer_mill':16e-6, 'grate_firing':20e-6}
23  x_1_coke = {'anthracite':1.0, 'other_coals':0.5}
24  x_2_coke = {'suspension_firing':0.1, 'grate_firing':0.03}
25  
26  #Functions to extract inputs for required fuels
27  def angular_coefficient(b_tubewalls, d_tubewalls, alpha_ang_coef):
28      return {'tubewalls': (1.15484*(b_tubewalls/d_tubewalls)**(-8.50475E-1)) * (1 + alpha_ang_coef), \
29              'refractory': 1.0, 'furnaceexit':1.0, 'grate': 0.0}
30  
31  def non_uniformity_factor(fuel, h_hfur):
32      c_1 = {'lignite': 1.4728, 'coal': 1.9745, 'oil_gas': 1.5214}
33      c_2 = {'lignite': 1.3324, 'coal': 1.808, 'oil_gas': 2.9035}
34      lam = {'lignite': 1.7131, 'coal': 0.9821, 'oil_gas': 1.575}
35      mu = {'lignite': 0.2275, 'coal': 0.2275, 'oil_gas': 0.2675}
36      sigma = {'lignite': 0.1907, 'coal': 0.2371, 'oil_gas': 0.1968}
37  
38      return c_1[fuel] * lam[fuel]/2 * np.exp(lam[fuel]/2 *(2*mu[fuel]**c_2[fuel] + lam[fuel]*sigma[fuel]**2 - 2*h_hfur))
\
39          * sp.special.erfc((mu[fuel]**(c_2[fuel])+lam[fuel]*sigma[fuel]**2 - h_hfur)/(np.sqrt(2)*sigma[fuel]))
40  
41  def reradiation_coeffecient(fuel, T_fe):
42      if fuel == 'solid_fuel':
43          if T_fe < 1060+kel:
44              beta = 1.0
45          else:
46              beta = (-2.18793E-6)*(T_fe-kel)**2 + 4.36338E-3*(T_fe-kel) - 1.16738
47      elif fuel == 'heavy_oil':
48          if T_fe<950+kel:
49              beta = 0.9
50          else:
51              beta = -1.43393E-6*(T_fe-kel)**2 + 2.53702E-3*(T_fe-kel) - 0.217727
52      return beta
53  
54  #Furnace element
55  class Furnace(Element):
56      #Constructor
57      def __init__(self, mdot_f, mdot_fa, mdot_fg, f_radloss, A_fur, Vol_fur, xi_fur, x_fur, d_fa, x1_coke, x2_coke,
M_flame, id, \
58          K = 1E-5, high_ash = 1, NUF_fuel_type = 'coal', RC_fuel_type = 'solid_fuel', fixed_mdot = 0, A = 1,  **kwargs):
59          Element.__init__(self)
60          self.mdot_f = mdot_f
61          self.mdot_fa = mdot_fa
```

```
62          self.mdot_fg = mdot_fg
63          self.f_radloss = f_radloss
64          self.A_fur = A_fur
65          self.Vol_fur = Vol_fur
66          self.xi_fur = xi_fur
67          self.x_fur = x_fur
68          self.d_fa = d_fa
69          self.x1_coke = x1_coke
70          self.x2_coke = x2_coke
71          self.M_flame = M_flame
72          self.id = id
73          self.K = K
74          self.high_ash = high_ash
75          self.NUF_fuel_type = NUF_fuel_type
76          self.RC_fuel_type = RC_fuel_type
77          self.fixed_mdot = fixed_mdot
78          self.A = A
79
80          self.__dict__.update(kwargs)
81
82  #Component characteristic function for furnace
83      def comp_char(self):
84          self.p_avg = (self.inlet.p + self.outlet.p)/2
85          self.T_avg = (self.inlet.T + self.outlet.T)/2
86
87          self.h_avg = gmp.enthalpy(self.inlet.gasmixture, self.T_avg, self.p_avg)
88          self.rho_avg = gmp.density(self.inlet.gasmixture, self.T_avg, self.p_avg)
89          self.cp_avg = gmp.specific_heat_pressure(self.inlet.gasmixture, self.T_avg, self.p_avg)
90          self.mu_avg = gmp.viscosity(self.inlet.gasmixture, self.T_avg, self.p_avg)
91          self.k_avg = gmp.conductivity(self.inlet.gasmixture, self.T_avg, self.p_avg)
92
93          #Furnace projected radiative area
94          self.A_rad_fur = 0.0
95          for key in self.x_fur:
96              self.A_rad_fur += self.x_fur[key]*self.A_fur[key]
97
98          #Furnace wall efficiency - psi
99          self.psi_fur = {}
100         self.psi_eff_fur = 0.0
101         for key in self.x_fur:
102             self.psi_fur[key] = self.x_fur[key]*self.xi_fur[key]
103             self.psi_eff_fur += (self.psi_fur[key]*self.A_fur[key])/self.A_rad_fur
104
105         # Furnace mean beam length S
106         self.A_tot_fur = 0.0
107         for key in self.A_fur:
108             self.A_tot_fur += self.A_fur[key]
109         self.S_mbl = 3.6 * self.Vol_fur / self.A_tot_fur
110
111         #Pressure drop, mass flow rate, pressure component characteristics
112         self.dp = self.outlet.p - self.inlet.p
113         self.C_l = self.K/(2 * self.rho_avg * self.A**2)
114         if self.fixed_mdot == 0:
115             self.mdot = -self.dp/np.abs(-self.dp) * np.sqrt(np.abs(-self.dp)/np.abs(self.C_l))
116         else:
117             self.mdot = self.fixed_mdot
118
119         self.A_1 = -self.C_l * np.abs(self.mdot)
120         self.A_0 = self.dp - self.A_1*self.mdot
121
122         #No work
123         self.Wdot = 0
124         #Qdot updated through heat transfer element
125         self.Qdot = 0
126
127         self.energy_sources = 0
128
```

Appendix B. Boiler heat exchanger areas

Appendix Table 1: External heat transfer surface areas for boiler heat exchangers

Heat exchanger	A_{ho} [m^2]
sCO2H1	20.83
SH2	120.39
SCR1	32.56
tCAV1	46.63
sCO2HX	298.85
sCO2H2	251.27
bCAV1	35.44
SCR2	23.94
CAV2	24.00
SH1	252.58
bEV	219.59
tEV	219.59
tAH	425.71
EC3	200.58
EC2	200.58
EC1	200.58
bAHt	340.57
bAHb	340.57

Appendix C. sCO2 Brayton cycle pressures and temperatures for varying loads

Appendix Table 2: Pressures and temperatures for the single heater configuration at various loads using inventory control

Load	100%		90%		80%		70%		60%	
	p (kPa)	**T (°C)**	**p (kPa)**	**T (°C)**	**p (kPa)**	**T (°C)**	**p (kPa)**	**T (°C)**	**p (kPa)**	**T (°C)**
comp_in	10000	32.5	9375	33.6	8750	34.5	8125	34.2	7500	32.5
RXHP_in	24974	53.5	23655	56.5	22079	60.2	20140	64.5	15193	74.6
RXHP_out	24730	383.4	23427	357.9	21874	332.5	19966	316.1	15102	441.1
sCO2HX_in	24730	341.5	23427	319.5	21874	297.9	19966	284.6	15102	401.5
turb_in	24572	557.2	23275	530.5	21733	501.7	19844	479.2	15043	554.3
RXLP_in	10211	450.6	9586	425.2	8954	399.0	8309	380.4	7535	473.3
PC_in	10106	71.9	9485	70.1	8861	69.0	8229	68.9	7504	75.0
PC_out	10000	32.4	9375	33.5	8750	34.4	8125	34.2	7500	25.0

Appendix Table 3: Pressures and temperatures for the dual heater configuration at various loads using inventory control

Load	100%		90%		80%		70%		60%	
	p (kPa)	**T (°C)**	**p (kPa)**	**T (°C)**	**p (kPa)**	**T (°C)**	**p (kPa)**	**T (°C)**	**p (kPa)**	**T (°C)**
comp_in	10000	32.5	9375	33.5	8750	34.4	8125	34.1	7500	32.5
RXHP_in	25022	53.6	23726	56.4	22174	60.2	20271	64.4	15229	75.0
RXHP_out	24788	389.9	23507	367.2	21975	346.9	20101	337.4	15123	496.4
sCO2H1_in	24788	389.9	23507	367.2	21975	346.9	20101	337.4	15123	496.4
sCO2H1_out	24600	453.8	23326	431.0	21807	411.0	19952	403.4	15046	583.6
sCO2_ATT	24600	391.2	23326	372.4	21807	356.4	19952	351.5	15046	513.9
turb_in	24438	552.6	23169	529.4	21661	506.6	19823	491.9	14985	593.6
RXLP_in	10210	447.0	9585	424.7	8953	403.8	8308	392.2	7539	510.8
PC_in	10104	73.2	9483	71.2	8859	70.0	8227	69.6	7504	77.2
PC_out	10000	32.4	9375	33.5	8750	34.3	8125	34.1	7500	25.6

Appendix D. Calculation of direct radiation through furnace exit plane for CFD models

This appendix details the calculation of the direct radiation through the furnace exit plane, $\dot{Q}_{fe}$, for the CFD models. A Mathcad Prime 9.0 model was setup for the calculation, taking inputs in from the CFD model. Special thanks to Tinashe Kadakure from the Applied Thermofluid Process Modelling research unit for providing the Mathcad model for this calculation.

An energy balance is conducted around the furnace control volume to determine $\dot{Q}_{fe}$. Various mass flow rates, including total air, fuel, and fluegas, are extracted from the CFD model. Additionally, the fraction of fly-ash is also extracted from the mass flow report from particle tracking. The ash, fly-ash, and bottom ash mass flow rates are then calculated as:

$$\begin{aligned} \dot{m}_{ash} &= \dot{m}_{air} + \dot{m}_{fuel} - \dot{m}_{fg} \\ \dot{m}_{fa} &= Y_{fa}\dot{m}_{ash} \\ \dot{m}_{ba} &= \dot{m}_{ash} - \dot{m}_{fa} \end{aligned} \tag{D.1}$$

Next the sensible enthalpies for the fuel, air, fluegas, fly-ash and bottom-ash are determined. This requires the inlet fuel temperature, $T_f = 27\ °\text{C}$, T_{air} taken as the mass weighted average at all air inlets, T_{fg} is taken as the mass weighted average at the furnace exit plane, and T_{grate} is taken as the area weighted average at the grate top plane. The sensible enthalpy of air, h_{air} , is taken as the mass weighted average at all air inlets, and for fluegas, h_{fg} is taken as the mass weighted average at the furnace exit plane. For fuel and ash, the sensible enthalpies are calculated as:

$$\begin{aligned} h_{fuel} &= c_{p_{fuel}}(T_{fuel} - 25\ °\text{C}) \\ h_{fa} &= c_{p_{ash}}(T_{fg} - 25\ °\text{C}) \\ h_{ba} &= c_{p_{ash}}(T_{gr} - 25\ °\text{C}) \end{aligned} \tag{D.2}$$

Consequently, the sensible energy flows can be computed, by the respective flow stream mass flow rate and sensible enthalpy, resulting in the calculation of $\dot{Q}_{air}$, $\dot{Q}_{fuel}$, $\dot{Q}_{fg}$, $\dot{Q}_{fa}$, and $\dot{Q}_{ba}$. Energy sources are then computed, including from heat of reaction released from the gas, $\dot{Q}_{gas}$, and from char combustion $\dot{Q}_{char}$. $\dot{Q}_{gas}$ is determined from Fluent®, through the total heat of reaction source. To calculate $\dot{Q}_{char}$, the amount of char burnout, $DPM_{burnout,char}$ is determined in the furnace volume from discrete phase sources. The heat of reaction for char is given as $LCV_{char} = 10100\ \text{kJ/kg}$. Consequently, $\dot{Q}_{char}$ can be calculated:

$$\dot{Q}_{char} = DPM_{burnout,char} LCV_{char} \tag{D.3}$$

The total heat of reaction, $\dot{Q}_{react}$ is then computed by summing $\dot{Q}_{gas}$ and $\dot{Q}_{char}$.

The latent heat required for moisture evaporation is calculated. The latent heat of vaporisation is given as $h_{fg,water} = 2263$ kJ/kg. The total latent heat required for moisture evaporation is then calculated as:

$$\dot{Q}_{latent} = Y_{AR,M}\dot{m}_{fuel}h_{fg,water} \tag{D.4}$$

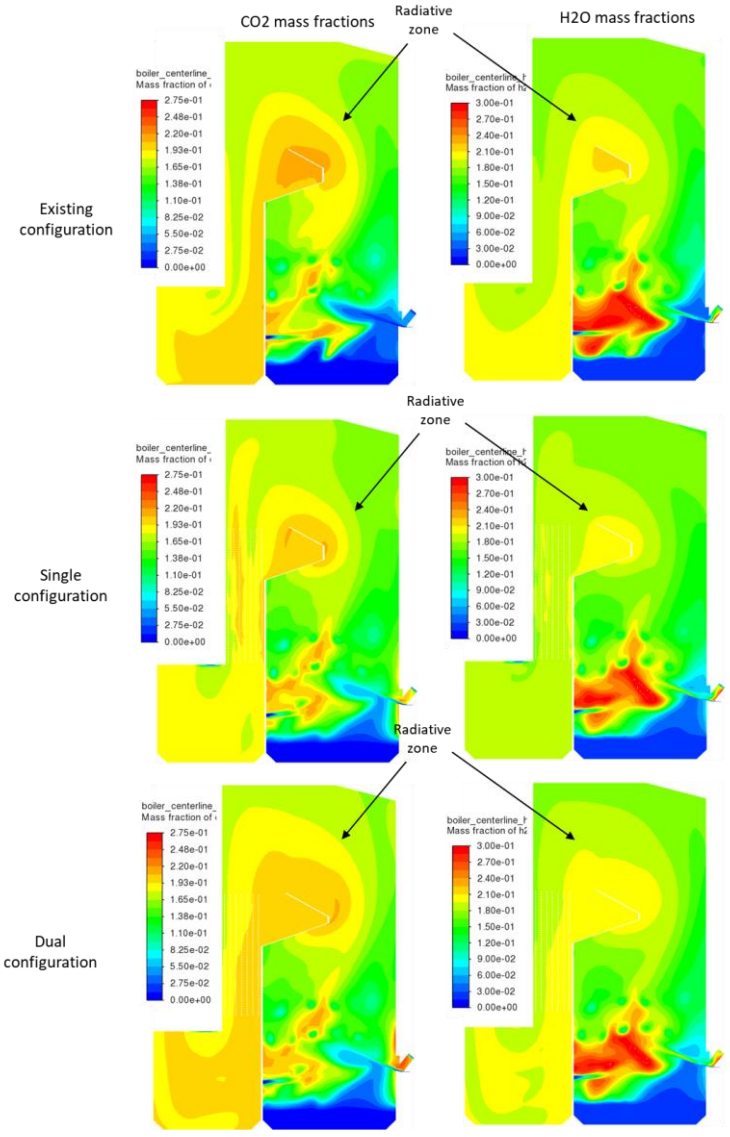
The heat transfer rates to the water walls surrounding the furnace volume, $\dot{Q}_{ww}$, is extracted. The final energy balance is written in Eq. (D.5), with the calculation of $\dot{Q}_{fe}$:

$$\begin{aligned}
\dot{Q}_{in} &= \dot{Q}_{air} + \dot{Q}_{fuel} + \dot{Q}_{react} \\
\dot{Q}_{out} &= \dot{Q}_{fg} + \dot{Q}_{fa} + \dot{Q}_{ba} + \dot{Q}_{ww} + \dot{Q}_{latent} \\
\Delta\dot{Q} &= \dot{Q}_{fe} = \dot{Q}_{in} - \dot{Q}_{out}
\end{aligned} \tag{D.5}$$

Appendix E. Additional CFD modelling results: Part load

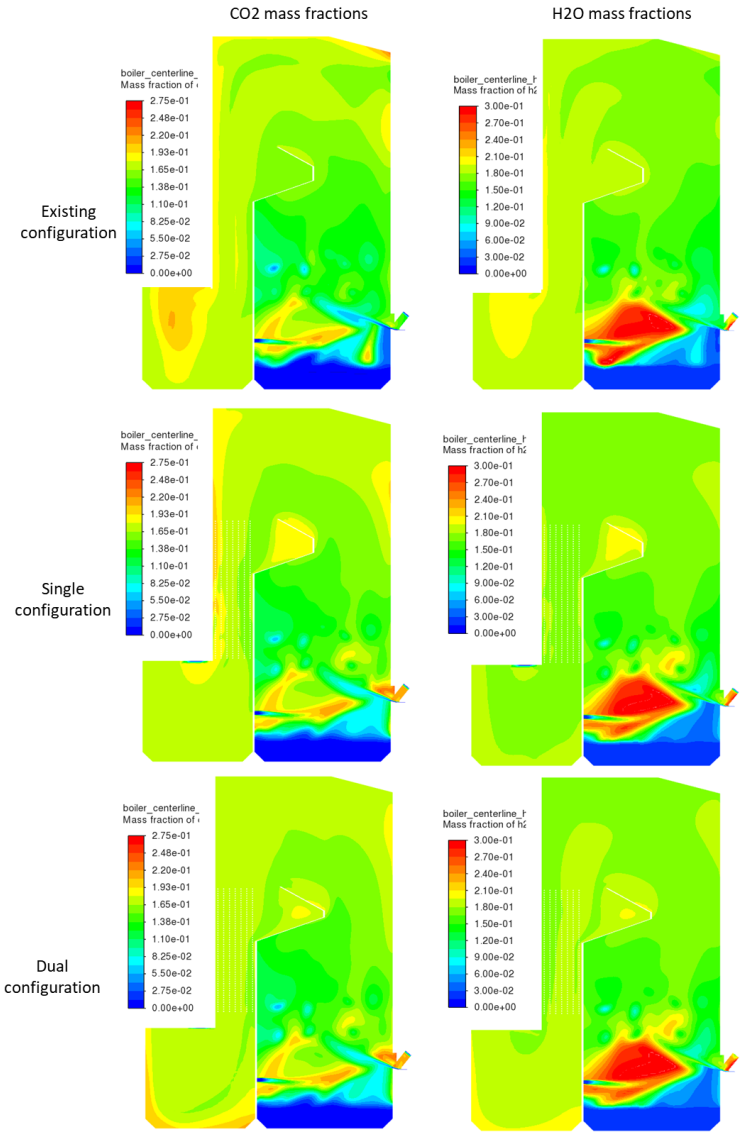
This appendix discusses additional results from the part load case study of the CFD modelling.

Observing concentrations of CO2 and H2O in Appendix Figure 5, radiative zones were identified at SH2 above the boiler nose. For the dual configuration, this zone was slightly larger compared to the single configuration and holds true for all load percentages. Compared to the nominal load case, the radiative zones were smaller and concentrated around the boiler nose, rather than extending further up to the boiler roof. For the dual and existing configurations, higher mass fractions were observed in the convective pass compared to the single configuration.



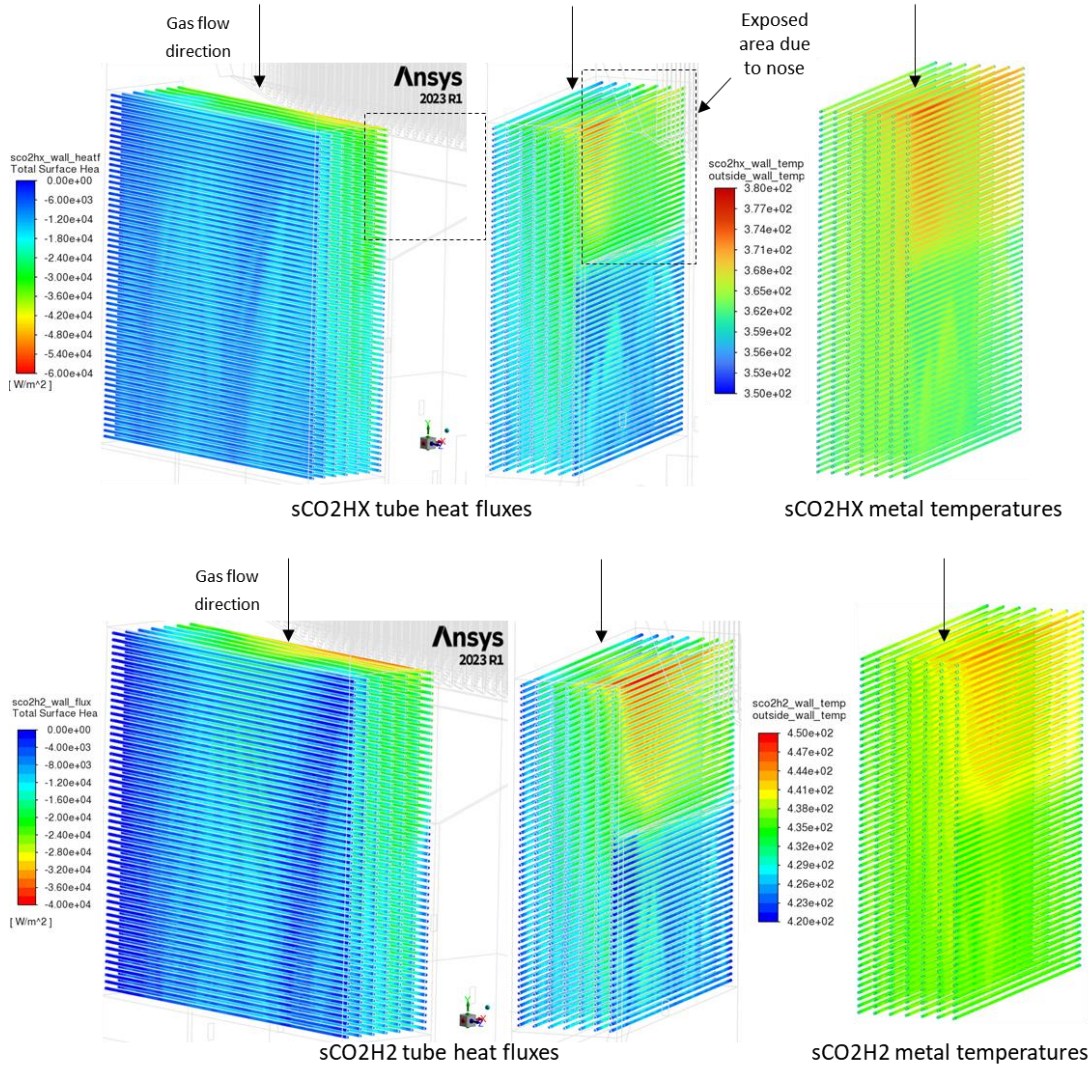
Appendix Figure 5: Carbon dioxide and water mass fractions at 80% load for all configurations

At 60% load, shown in Appendix Figure 6, concentrations of CO2 and H2O were lower around the boiler nose compared to 80% load, and were more spread out throughout the furnace domain. The spread of radiative species throughout the furnace domain would result in decreased partial pressures at previously observed radiative zones, near SH2 and the boiler nose, and would reduce local radiative heat transfer rates.



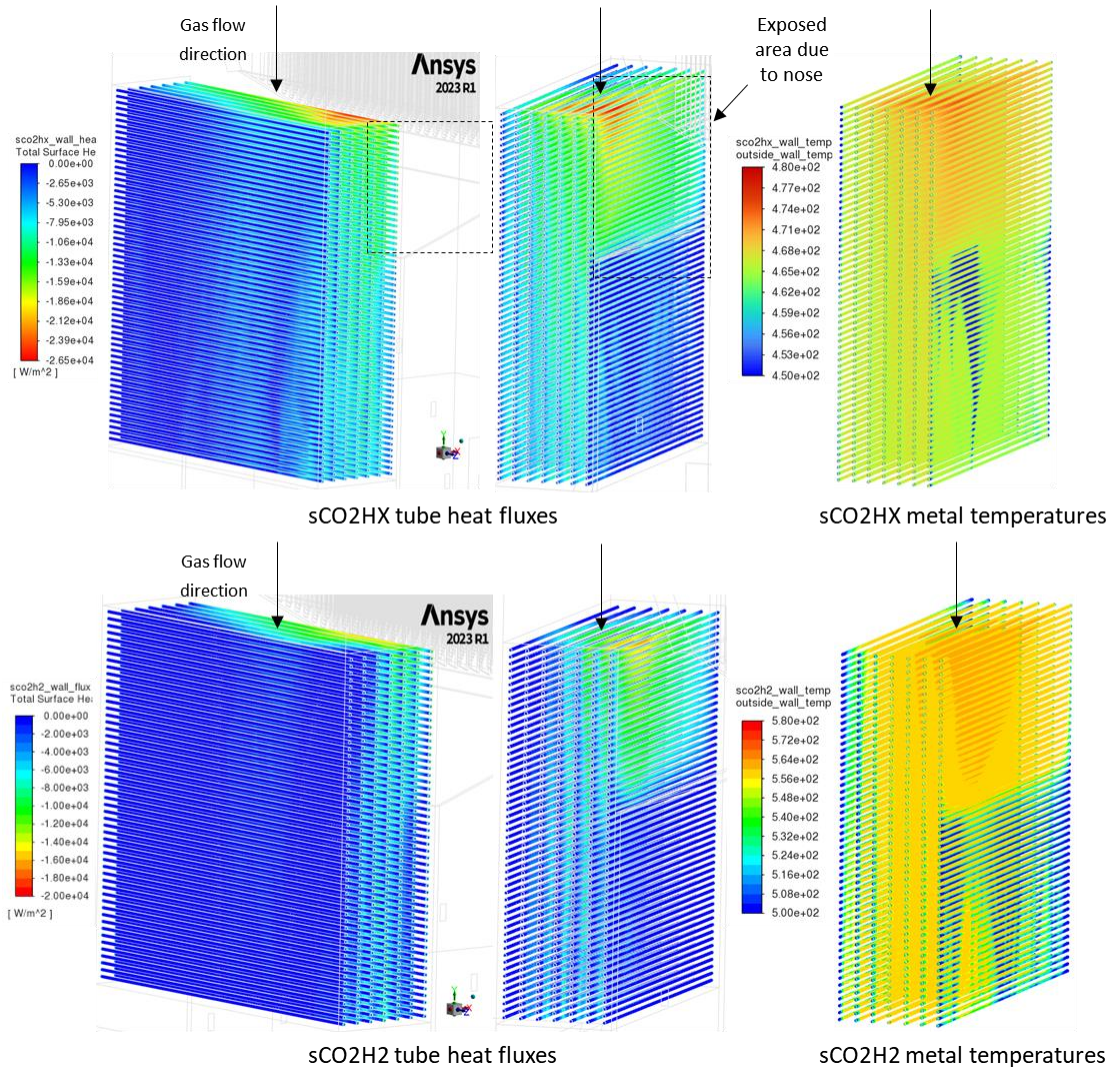
Appendix Figure 6: Carbon dioxide and water mass fractions at 60% load for all configurations

Appendix Figure 7 and Appendix Figure 8 shows the tube wall heat fluxes and metal temperatures for the convective sCO2 heater (sCO2HX and sCO2H2 for the single and dual configurations respectively). Localised maximum heat fluxes can be observed at the top right half of the heat exchanger, towards the side wall. Compared to the nominal load case, the zone with maximum heat flux is smaller, particularly for the 60% load case with the dual configuration. Notably, the heat flux distribution through the entire tube bank is significantly reduced for the dual configuration at 60% load, attributed to the low TTD between the gas and tube wall temperature.



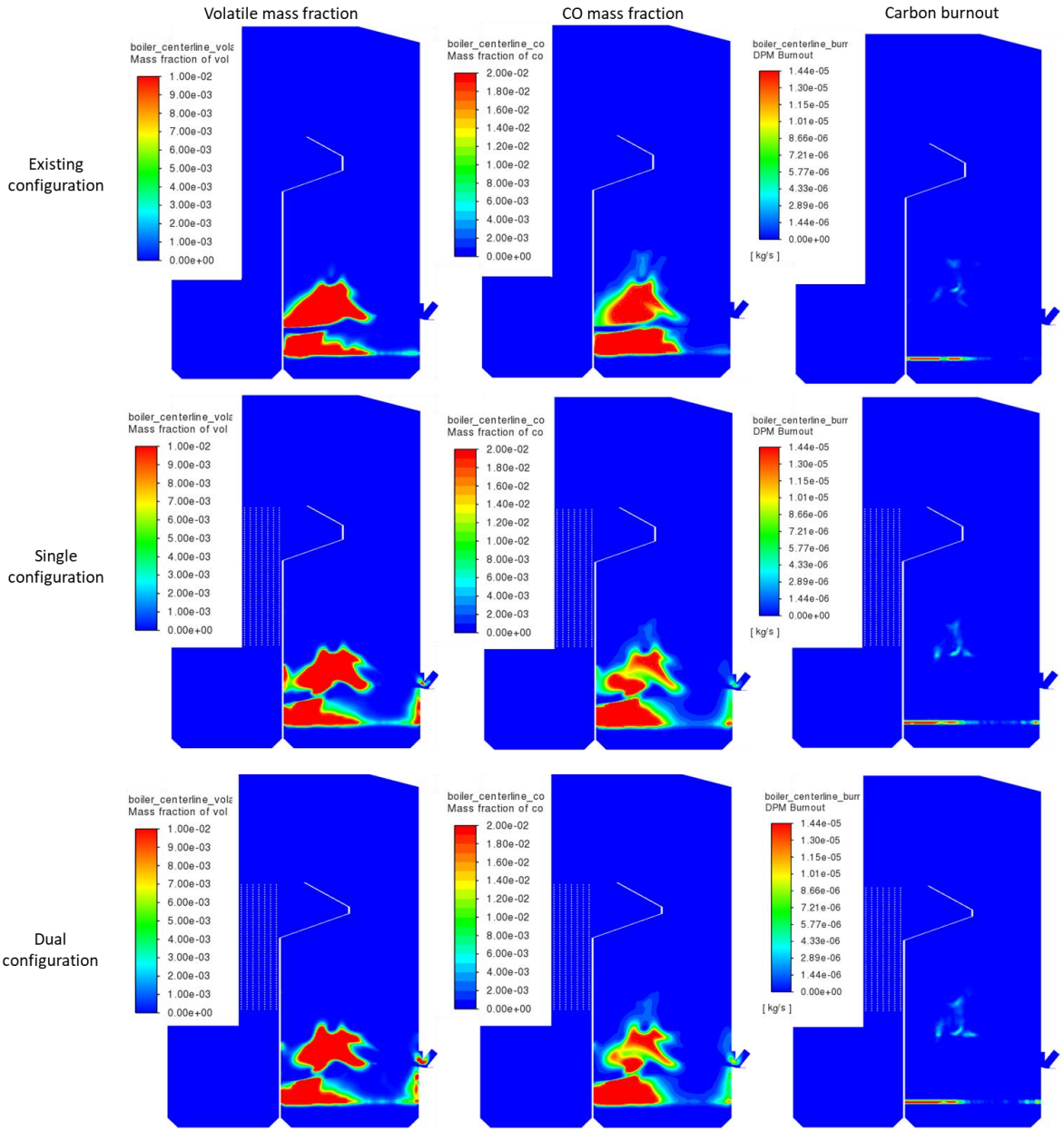
Appendix Figure 7: sCO2HX and sCO2H2 tube wall heat fluxes and metal temperatures at 80% load

The highest tube metal temperatures were observed at the exposed area of the tubes due to the boiler nose, where the localised maximum heat flux zone was located. The remainder of the tubes were largely influenced by the internal sCO2 temperature. This is evident especially for the 60% load cases.



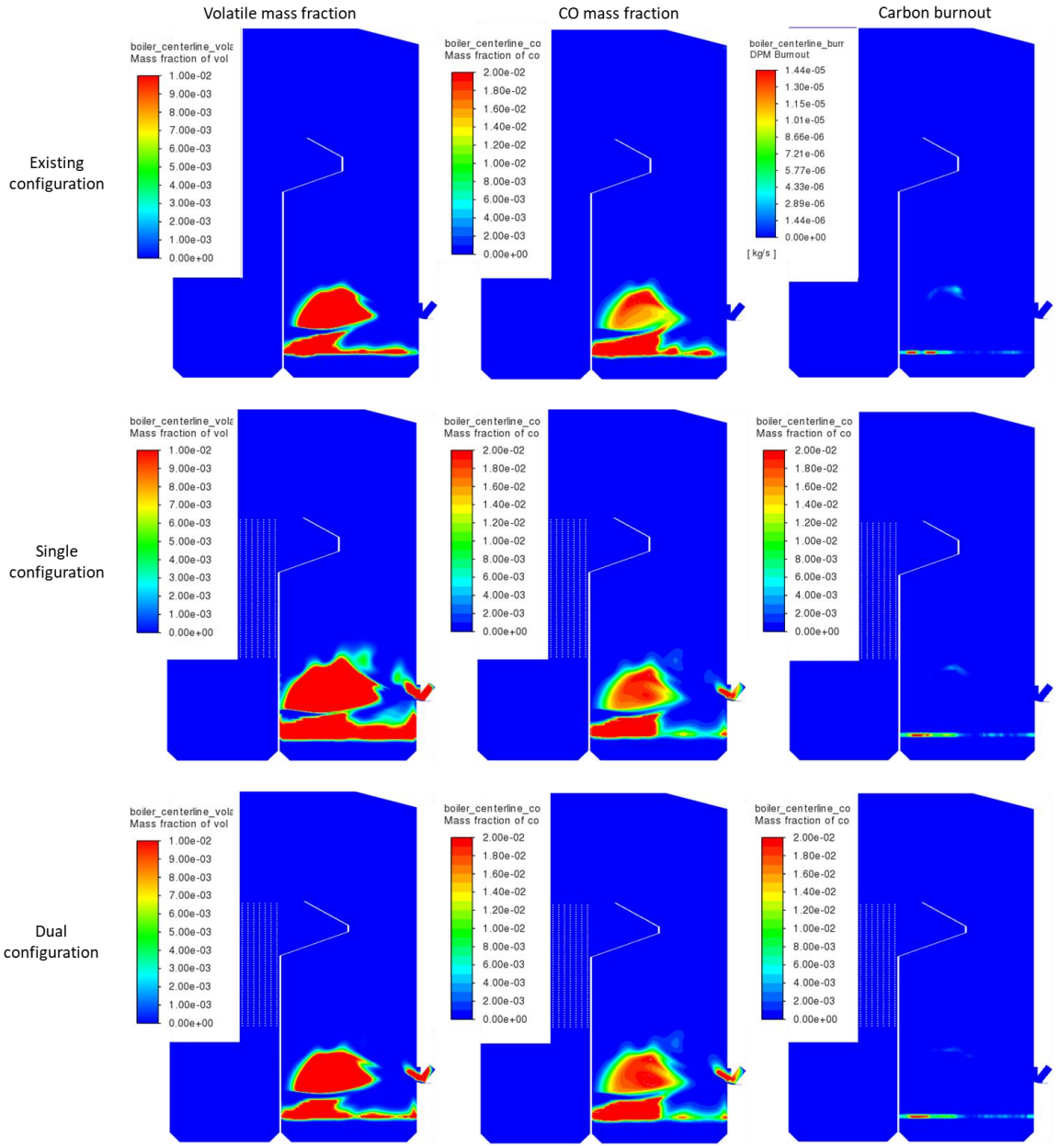
Appendix Figure 8: sCO2HX and sCO2H2 tube wall heat fluxes and metal temperatures at 60% load

Appendix Figure 9 and Appendix Figure 10 show the volatile matter, CO mass fractions, and carbon burnout at part loads. Similar to the nominal load case, carbon burnout contours highlight majority of fixed carbon is consumed on the grate. On the other hand, the VM contours highlight that most of the release of VM occurs in the freeboard, closer to the centre of the furnace. For the nominal load case, majority of the VM released occurred towards the rear of the furnace. The higher ratio of DA mass flow rate for part load cases compared to the nominal load case shifts the particles away from the rear of the furnace. For 60% load cases, this effect is more pronounced, with a significant fraction of VM released on the grate at the centre of the furnace. The reduction in PA and SA mass flow rates relative to DA mass flow rates at lower loads may influence the spread of particles, especially along the grate, and consequently the release of VM during combustion.



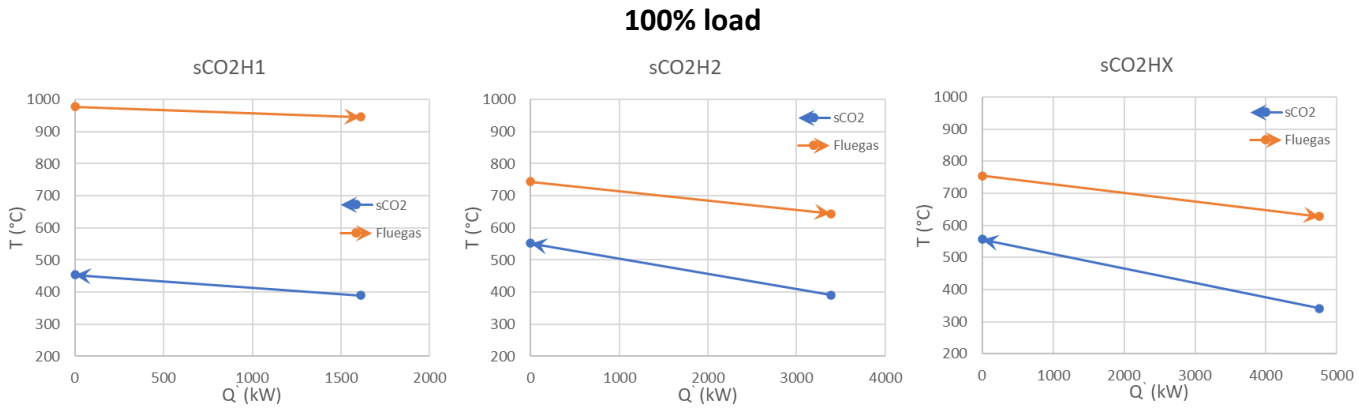
Appendix Figure 9: Volatile matter, CO mass fractions, and carbon burnout at 80% load

Lastly, the CO contours highlight majority concentration in the furnace where combustion takes place, similar to the nominal load case. Negligible CO was observed outside of the furnace domain, indicating complete combustion within the furnace.

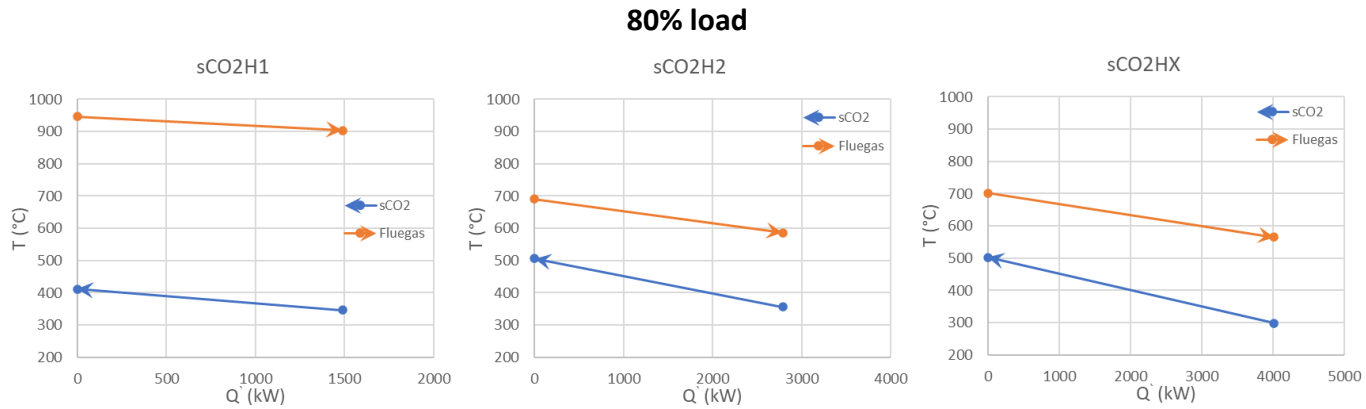


Appendix Figure 10: Volatile matter, CO mass fractions, and carbon burnout at 60% load

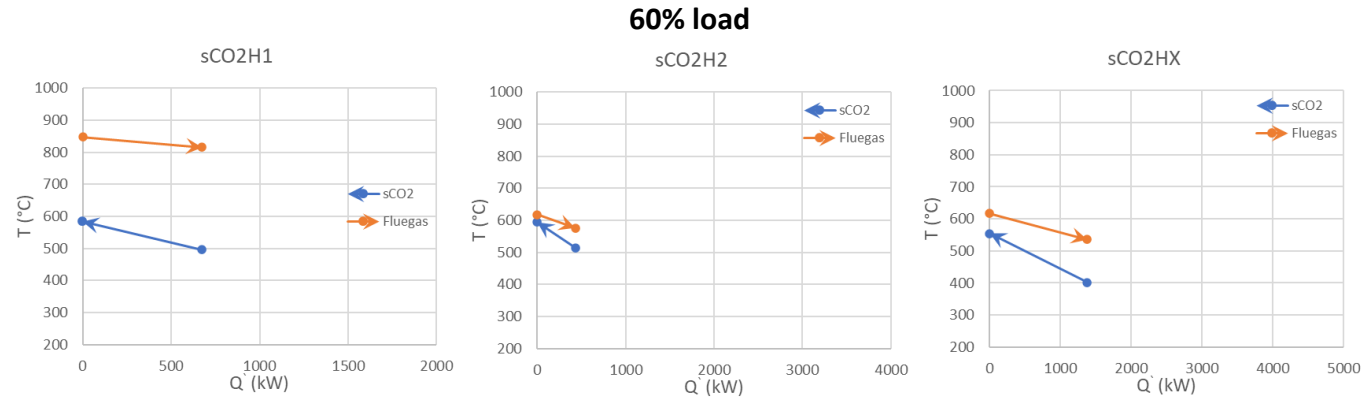
Appendix Figure 11, Appendix Figure 12, and Appendix Figure 13 show the sCO2 heat exchanger duty (T-Q`) diagrams for sCO2H1, sCO2H2 and sCO2HX at 100%, 80%, and 60% loads respectively based on the CFD results. All sCO2 heaters are configured for a counter-current configuration. The decrease in heat uptakes is visualised in these figures, particularly at 60% load. A pinch point is observed for the convective heaters (sCO2HX/2) at 60% load. This reduction in TTD results in a significant decrease in heat uptakes at the sCO2 heaters compared to higher loads, as discussed in Chapter 7.1.



Appendix Figure 11: sCO2 heat exchanger duty diagrams (T-Q`) at 100% load



Appendix Figure 12: sCO2 heat exchanger duty diagrams (T-Q`) at 80% load



Appendix Figure 13: sCO2 heat exchanger duty diagrams (T-Q`) at 60% load