

# Heat Transfer and Fluid Flow Calculations of Industrial Shell Boilers and Evaluation of Operation Conditions – Draft

Saif-Aldain Aqel

2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Industrial Application of Shell Boilers</b>	<b>4</b>
2.1	Typical Industries . . . . .	4
2.2	Typical Steam Duties . . . . .	5
2.3	Advantages and Limitations . . . . .	5
2.3.1	Advantages . . . . .	5
2.3.2	Limitations . . . . .	6
2.4	Typical Multi-Pass Layout . . . . .	6
<b>3</b>	<b>Boiler Geometry and Configuration</b>	<b>8</b>
3.1	Overall layout . . . . .	8
3.2	Drum configuration . . . . .	9
3.3	Consolidated geometry and surface specification . . . . .	9
<b>4</b>	<b>Combustion Model</b>	<b>11</b>
4.1	Fuel composition . . . . .	11
4.2	Stoichiometric $O_2$ requirement . . . . .	12
4.3	Air–fuel ratio and excess air $\lambda$ . . . . .	13
4.3.1	Stoichiometric $O_2$ requirement (per mole of fuel mixture) . . . . .	14
4.3.2	Actual $O_2$ supplied . . . . .	14
4.3.3	Air required . . . . .	15
4.3.4	Air–fuel ratio . . . . .	15
4.4	Lower heating value (LHV) and heat release . . . . .	15
4.4.1	Method . . . . .	16
4.4.2	Numerical results for the present fuel . . . . .	17
4.4.3	Total heat input to the boiler $Q_{in}$ . . . . .	18
4.5	Adiabatic flame temperature . . . . .	19
4.5.1	Thermodynamic formulation . . . . .	19
4.5.2	Implementation . . . . .	20
4.5.3	Numerical result for the present case . . . . .	21
4.6	Flue-gas composition . . . . .	21
4.6.1	Definitions and distinction . . . . .	22
4.6.2	Equilibrium flue gas at adiabatic conditions . . . . .	22

4.6.3	Fully burnt boiler flue gas . . . . .	24
4.6.4	Output fields . . . . .	25
<b>5</b>	<b>Heat-Transfer Calculations</b>	<b>27</b>
5.1	Fundamental heat-balance equations . . . . .	27
5.2	Local energy balance . . . . .	27
5.3	Overall conductance and resistance network . . . . .	28
5.4	Stage- and boiler-level duties . . . . .	29
5.5	Gas-side . . . . .	30
5.5.1	Single-tube and reversal-chamber (internal) . . . . .	30
5.5.2	Tube-bank (internal) . . . . .	31
5.5.3	Economizer (external) . . . . .	32
5.5.4	Gas radiation model . . . . .	33
5.6	Water-side . . . . .	35
5.6.1	Economizer (internal) . . . . .	35
5.6.2	Tube-bank (external) . . . . .	37
5.6.3	Treatment of boiling . . . . .	38
5.7	Per-step resistance insertion . . . . .	39
5.8	Wall-temperature update and thermal convergence . . . . .	40
<b>6</b>	<b>Hydraulic Calculations</b>	<b>42</b>
6.1	Gas-Side $\Delta P$ per Stage . . . . .	43
6.2	Water-Side $\Delta P$ per Stage . . . . .	43
6.3	Total Boiler $\Delta P$ and Stack Pressure . . . . .	44
6.4	Consolidated $\Delta P$ Table (from solver output) . . . . .	44
<b>7</b>	<b>Boiler Performance Results</b>	<b>46</b>
7.1	Energy balance ( $Q_{in}$ , $Q_{useful}$ ) . . . . .	46
7.2	Efficiencies (direct and indirect) . . . . .	47
7.3	Steam generation rate and mass-flow convergence . . . . .	47
7.4	Stage level performance . . . . .	48
7.5	Overall boiler summary . . . . .	49
<b>8</b>	<b>Sensitivity Analysis</b>	<b>50</b>
8.1	Excess Air Ratio . . . . .	50
8.2	Drum Pressure . . . . .	50
<b>9</b>	<b>Conclusion</b>	<b>51</b>

# Chapter 1

## Introduction

placeholder

# Chapter 2

## Industrial Application of Shell Boilers

### 2.1 Typical Industries

Shell (fire-tube) boilers are widely used in small-to-medium steam and hot-water duties where compactness, robustness, and simple operation are prioritized over very high pressure or very large throughput. Typical sectors include:

- Food and beverage
  - Breweries, dairies, sugar refineries
  - Canneries, bakeries, confectionery plants
  - CIP (clean-in-place) systems and sterilization
- Chemical and pharmaceutical
  - Fine chemicals, specialty chemicals
  - Active pharmaceutical ingredient (API) and formulation plants
  - Steam for reactors, jacket heating, and clean steam generators
- Textiles and paper
  - Dyeing, washing, drying, and calendaring operations
  - Small paper mills and converting facilities
- Healthcare and institutional
  - Hospitals, clinics, and laboratories (space heating, humidification, sterilizers, autoclaves)
  - Universities, office complexes, district heating sub-plants
- Light manufacturing and general industry
  - Metal finishing, surface treatment, and cleaning
  - Rubber and plastics processing
  - Laundry services and commercial dry-cleaning

## 2.2 Typical Steam Duties

Shell boilers are normally applied in low-to-medium pressure ranges and moderate steam capacities:

- Typical operating pressure range:
  - Saturated steam: 6–25 bar, occasionally up to 30 bar
  - Hot-water service: 10–16 bar
- Steam-generation rates (order of magnitude):
  - Small units: 0.5–5 t/h
  - Medium units: 5–20 t/h
  - Large shell boilers (upper practical range): 20–40 t/h, beyond which water-tube designs are usually preferred

## 2.3 Advantages and Limitations

### 2.3.1 Advantages

- Compact and integrated construction
  - Furnace, passes, and steam/water space are combined in a single pressure body.
  - Relatively small footprint and simple installation.
- Operational simplicity
  - Straightforward start-up and shutdown procedures.
  - Typically tolerant of moderate load swings and cycling (within design limits).
  - Often delivered as packaged units with burner, controls, and safety devices pre-engineered.
- Low-to-moderate capital cost
  - Attractive for small and medium plants, boiler houses, and decentralized steam supply.
- Good part-load performance
  - Large water content provides thermal buffer, reducing short-cycling of the burner.
  - Reasonable efficiency across a wide load range, especially with economizers.
- Maintenance and inspection
  - Accessible gas passes and tube bundles (depending on design) for cleaning and inspection.
  - Long-established technology with wide service and parts availability.

### 2.3.2 Limitations

- Pressure and capacity limits
  - Practical upper bounds on shell diameter and plate thickness limit maximum pressure and steam rate.
  - For very high pressure (e.g., >40–60 bar) or very large capacities, water-tube boilers are more suitable.
- Response time
  - Large water inventory slows thermal response to rapid, large load changes compared with water-tube boilers.
- Efficiency ceiling
  - Radiative and convective heat-transfer surfaces are constrained by geometry.
  - Very high efficiencies often require additional heat-recovery equipment (economizers, condensing stages, air preheaters).
- Transport and installation constraints
  - Shell diameter and weight can be limited by route and lifting capacity.
  - Retrofitting within existing boiler houses may be constrained by overall envelope.

## 2.4 Typical Multi-Pass Layout

Industrial shell boilers typically adopt multi-pass fire-tube configurations to enhance convective heat transfer and maintain acceptable gas-side velocities:

- Two-pass layout
  - First pass: large diameter furnace tube running from burner front to rear tubeplate.
  - Second pass: return of flue gas through banks of small-diameter fire-tubes back to the front tubeplate and flue outlet.
  - Simpler construction but lower total heat-transfer surface compared with three-pass designs.
- Three-pass layout (most common for industrial shell boilers)
  - Pass 1: large diameter furnace tube running from burner front to rear tubeplate.
  - Pass 2: First bank of smoke-tubes (typically reversing at the rear turnaround chamber).
  - Pass 3: Second bank of smoke-tubes.
  - Provides higher overall heat-transfer surface, more uniform gas cooling, and lower exit-gas temperatures.
- Extended heat-recovery sections

- Economizer: additional convective heat exchanger in the flue-gas path downstream of the boiler to preheat feedwater.
- Air preheater / condensing sections (optional): for high-efficiency systems using suitable fuels and materials.
- Flow arrangement
  - Gas-side: burner → furnace (Pass 1) → turnaround chamber → tube bank(s) (Passes 2 and 3) → stack.
  - Water/steam side: natural circulation between heated tube surfaces and the upper steam space within the drum/shell; feedwater introduced at cooler regions (often via economizer), steam drawn from the top of the shell.

This multi-pass concept underpins the subsequent detailed modelling of each convective and radiative heat-transfer stage  $HX_1-HX_6$  in the simulation.



# Chapter 3

## Boiler Geometry and Configuration

The simulated unit is a three-pass fire-tube shell boiler with six distinct gas-side heat-transfer stages and a single common steam drum on the water/steam side. Hot flue gas from the burner traverses a radiative furnace, two reversal chambers, two convective tube banks, and a final economiser before leaving to the stack. The water/steam side is treated as a single circulating system coupled to all pressure parts.

### 3.1 Overall layout

The gas path is represented as:

$$\text{Burner} \rightarrow \text{HX}_1 \rightarrow \text{HX}_2 \rightarrow \text{HX}_3 \rightarrow \text{HX}_4 \rightarrow \text{HX}_5 \rightarrow \text{HX}_6 \rightarrow \text{stack}$$

with the following interpretation:

- $\text{HX}_1$  – Furnace (first pass, `single_tube`)  
Large, single furnace tube where combustion products enter directly from the burner and transfer heat mainly by radiation and high-temperature convection to the surrounding water/steam.
- $\text{HX}_2$  – First reversal chamber (`reversal_chamber`)  
Short cylindrical wet back chamber that turns the flow from the furnace outlet into the first convective tube bank (gas direction change =  $180^\circ$ ).
- $\text{HX}_3$  – First convective tube bank (second pass, `tube_bank`)  
Bank of small diameter fire tubes arranged in a staggered pattern inside the shell; flue gas flows inside of the tubes, water/steam outside.
- $\text{HX}_4$  – Second reversal chamber (`reversal_chamber`)  
Second turning chamber redirecting gas from the first to the second tube bank.

- $HX_5$  – Second convective tube bank (third pass, `tube_bank`)  
Second fire-tube bundle, again in cross-flow, representing the last in-boiler convective pass.
- $HX_6$  – Economiser (`economiser`)  
Separate, downstream tube bank used to preheat feedwater in single-phase operation before entering the drum/boiler circuit.

Pool boiling is enabled for  $HX_1$ – $HX_5$  (pressure parts);  $HX_6$  is explicitly single-phase on the water side.

## 3.2 Drum configuration

The boiler has a single horizontal steam drum described by the `Drum` object. Its inner diameter is

$$D_{i,\text{drum}} = 4.5 \text{ m}$$

and its length

$$L_{\text{drum}} = 5.0 \text{ m}$$

.

The drum is not modelled with internal separators or circulation hardware. It simply supplies the saturated water/steam state at boiler pressure, while all circulation effects are represented by the single 1-D water/steam stream used in the heat-transfer stages.

## 3.3 Consolidated geometry and surface specification

Table 3-1 summarises the principal geometric inputs used in the simulation for the drum and all six heat-transfer stages. Values are taken directly from the YAML configuration files (`drum.yaml` and `stages.yaml`).

Element	Kind	Di [m]	L [m]	N_tubes [-]	Wall t [mm]	Roughness [ $\mu\text{m}$ ]	Pool boiling [-]
DRUM	drum	4.50	5.00	–	–	0.5	–
$HX_1$	single_tube	1.40	5.276	1	2.9	0.5	true
$HX_2$	reversal_ch.	1.60	0.80	1	2.9	0.5	true
$HX_3$	tube_bank	0.076	4.975	118	2.9	0.5	true
$HX_4$	reversal_ch.	1.60	0.80	1	2.9	0.5	true
$HX_5$	tube_bank	0.076	5.620	100	2.9	0.5	true
$HX_6$	economiser	0.076	7.50	160	2.5	0.5	false

All pressure-part stages ( $HX_1$ – $HX_5$ ) share the same steel wall thermal conductivity of  $k_{\text{wall}} = 16 \text{ W/m/K}$ . The economiser ( $HX_6$ ) is modelled with a higher wall conductivity  $k_{\text{wall}} = 30 \text{ W/m/K}$  and a clean surface (zero fouling thickness) to represent a best-case heat-recovery configuration.

# Chapter 4

## Combustion Model

### 4.1 Fuel composition

The boiler is fired with a natural-gas-type fuel defined in the simulation input (`config/fuel.yaml`).

The fuel is supplied at  $300K$  and  $1.013 \times 10^5 Pa$  with a mass flow rate of  $0.5 kg/s$ . Its composition is specified on a mass-fraction basis and converted internally to mole fractions for all stoichiometric and thermodynamic calculations.

Table 4-1 summarises the fuel composition in both mass and mole fraction form.

Compound	Formula	Mass fraction $w_i$ [-]	Mole fraction $x_i$ [-]	Comment
Methane	$CH_4$	0.80	0.8895	Main combustible, dominant contributor to LHV
Ethane	$C_2H_6$	0.10	0.0593	Heavier hydrocarbon, increases LHV and required $O_2$
Propane	$C_3H_8$	0.04	0.0162	Heavier hydrocarbon, raises flame temperature
n-Butane	$C_4H_{10}$	0.01	0.00307	Minor heavy hydrocarbon fraction
Hydrogen sulfide	$H_2S$	0.01	0.00523	Sulfur-bearing contaminant $\rightarrow SO_2$ in flue gas
Nitrogen	$N_2$	0.02	0.0127	Inert ballast in the fuel stream
Carbon dioxide	$CO_2$	0.01	0.00405	Inert (already fully oxidised)
Water vapour	$H_2O$	0.01	0.00990	Moisture carried with the fuel

The mass fractions sum to 1.0 by definition. The mole fractions  $x_i$  are obtained from

$$x_i = \frac{\frac{w_i}{M_i}}{\sum_j \frac{w_j}{M_j}}$$

where  $M_i$  is the molar mass of species  $i$  from `molar_masses` in `common/constants.py`. The resulting fuel mixture is therefore predominantly methane with small amounts of heavier hydrocarbons and trace inert/contaminant species, representative of a typical processed natural gas for boiler firing.

## 4.2 Stoichiometric O<sub>2</sub> requirement

Evaluate the stoichiometric oxygen requirement via the function `stoich_O2_required_per_mol` in `combustion/flue.py`. The algorithm is:

1. Use per-mole-of-species stoichiometric O<sub>2</sub> factors  $\nu_{O_2,i}$  from `O2_per_mol` in `common/constants.py`:

Species	Global reaction (complete combustion)	$\nu_{O_2,i}$ [mol O <sub>2</sub> / mol species]
CH <sub>4</sub>	CH <sub>4</sub> + 2 O <sub>2</sub> → CO <sub>2</sub> + 2 H <sub>2</sub> O	2.0
C <sub>2</sub> H <sub>6</sub>	C <sub>2</sub> H <sub>6</sub> + 3.5 O <sub>2</sub> → 2 CO <sub>2</sub> + 3 H <sub>2</sub> O	3.5
C <sub>3</sub> H <sub>8</sub>	C <sub>3</sub> H <sub>8</sub> + 5 O <sub>2</sub> → 3 CO <sub>2</sub> + 4 H <sub>2</sub> O	5.0
C <sub>4</sub> H <sub>10</sub>	C <sub>4</sub> H <sub>10</sub> + 6.5 O <sub>2</sub> → 4 CO <sub>2</sub> + 5 H <sub>2</sub> O	6.5
H <sub>2</sub> S	H <sub>2</sub> S + 1 O <sub>2</sub> → SO <sub>2</sub> + H <sub>2</sub> O	1.0
N <sub>2</sub> , CO <sub>2</sub> , H <sub>2</sub> O	Inert/fully oxidised → no additional O <sub>2</sub>	0.0

2. Compute the stoichiometric O<sub>2</sub> requirement per mole of fuel mixture as

$$\nu_{O_2,stoich} = \sum_i x_i \nu_{O_2,i}$$

Using the mole fractions from Section 4.1 for the present fuel:

- $x_{CH_4} = 0.8895$
- $x_{C_2H_6} = 0.0593$
- $x_{C_3H_8} = 0.0162$
- $x_{C_4H_{10}} = 0.00307$
- $x_{H_2S} = 0.00523$

- remaining species:  $x_{N_2}$ ,  $x_{CO_2}$ ,  $x_{H_2O}$  are inert in the stoichiometric balance.

Hence

$$\begin{aligned}\nu_{O_2, \text{stoich}} &= 0.8895 \cdot 2.0 + 0.0593 \cdot 3.5 + 0.0162 \cdot 5.0 + 0.00307 \cdot 6.5 + 0.00523 \cdot 1.0 \\ &\approx 2.09 \text{ mol } O_2 \text{ per mol fuel mixture}\end{aligned}$$

This is exactly what `stoich_O2_required_per_mol_fuel` returns:

```
def stoich_O2_required_per_mol_fuel(fuel: GasStream) -> Q_:
    fuel_x = to_mole(fuel.comp)
    total = sum(fuel_x[k] * O2_per_mol.get(k, 0.0) for k in fuel_x)
    return Q_(total, "dimensionless")
```

For later hydraulic and performance interpretation, it is also useful to express this on a mass basis.

For 1 kg of fuel, the total fuel moles are

$$n_{\text{fuel}, \text{total}} = \sum_i \frac{w_i}{M_i} \approx 56.1 \text{ mol fuel/kg}$$

Thus the stoichiometric  $O_2$  requirement per unit fuel mass is

$$n_{O_2, \text{stoich}}^{(m)} = \nu_{O_2, \text{stoich}} n_{\text{fuel}, \text{total}} \approx 2.09 \times 56.1 \approx 1.17 \times 10^2 \text{ mol } O_2/\text{kg fuel}$$

Converting to mass of  $O_2$  per kg of fuel:

$$\dot{m}_{O_2, \text{stoich}} = n_{O_2, \text{stoich}}^{(m)} M_{O_2} \approx 117.3 \text{ mol/kg} \times 0.031998 \text{ kg/mol} \approx 3.75 \text{ kg } O_2/\text{kg fuel}$$

So, for this fuel:

- Stoichiometric oxygen requirement:

$$\nu_{O_2, \text{stoich}} \approx 2.09 \text{ mol } O_2 \text{ per mol fuel mixture}$$

- Equivalent mass requirement:

$$\dot{m}_{O_2, \text{stoich}} \approx 3.75 \text{ kg } O_2 \text{ per kg fuel}$$

## 4.3 Air–fuel ratio and excess air $\lambda$

The simulation specifies an excess air ratio

$$\lambda = 1.1$$

in `config/operation.yaml`. This value enters the calculation through `air_flow_rates(air, fuel, excess)` in `combustion/flue.py`.

---

### 4.3.1 Stoichiometric O<sub>2</sub> requirement (per mole of fuel mixture)

From Section 4.2:

$$\nu_{\text{O}_2, \text{stoich}} = 2.09 \text{ mol O}_2/\text{mol fuel}$$

### 4.3.2 Actual O<sub>2</sub> supplied

Using:

$$\dot{n}_{\text{O}_2, \text{actual}} = \lambda \dot{n}_{\text{O}_2, \text{stoich}}$$

Thus:

$$\dot{n}_{\text{O}_2, \text{actual}} = 1.1 \nu_{\text{O}_2, \text{stoich}} \dot{n}_{\text{fuel}}$$

The molar fuel flow is determined from the mass-flow rate:

- Fuel mass flow:

$$\dot{m}_f = 0.5 \text{ kg/s}$$

- Total moles per unit mass of fuel mixture (from the mixture molar mass calculation):

$$n_{\text{fuel, total}} \approx 56.1 \text{ mol/kg}$$

- Therefore the total molar fuel flow:

$$\dot{n}_f = 56.1 \times 0.5 \approx 28.05 \text{ mol/s}$$

Hence the stoichiometric and actual O<sub>2</sub> flows are:

$$\dot{n}_{\text{O}_2, \text{stoich}} = 2.09 \times 28.05 = 58.7 \text{ mol/s}$$

$$\dot{n}_{\text{O}_2, \text{actual}} = 1.1 \times 58.7 = 64.6 \text{ mol/s}$$

---

### 4.3.3 Air required

Air O<sub>2</sub> mole fraction (from `air.yaml`):

$$x_{\text{O}_2, \text{air}} = 0.2095$$

Thus:

$$\dot{n}_{\text{air}} = \frac{\dot{n}_{\text{O}_2, \text{actual}}}{x_{\text{O}_2, \text{air}}} = \frac{64.6}{0.2095} \approx 308 \text{ mol/s}$$

The air molar mass (mixture weighted) is:

$$M_{\text{air}} \approx 0.02897 \text{ kg/mol}$$

Therefore the mass-based air flow rate:

$$\dot{m}_{\text{air}} = \dot{n}_{\text{air}} M_{\text{air}} \approx 308 \times 0.02897 \approx 8.93 \text{ kg/s}$$

---

### 4.3.4 Air–fuel ratio

Mass-based air–fuel ratio:

$$\text{AFR} = \frac{\dot{m}_{\text{air}}}{\dot{m}_f} = \frac{8.93}{0.5} \approx 17.9$$

---

## 4.4 Lower heating value (LHV) and heat release

The fuel lower and higher heating values, and the corresponding firing rate, are evaluated in `combustion/heat.py` by the function `compute_LHV_HHV(fuel)` and then used by `total_input_heat(fuel, air)`.

---



## 4.4.1 Method

### 4.4.1.1 Latent heat of water

Obtain the latent heat of vaporisation of water at the reference pressure  $P_{\text{ref}} = 101,325 \text{ Pa}$  from the IAPWS-97 correlation:

```
latent_H2O = WaterProps.h_g(P_ref) - WaterProps.h_f(P_ref)
```

where:

- $h_g$  is the saturated vapour enthalpy,
- $h_f$  is the saturated liquid enthalpy.

### 4.4.1.2 Reference formation enthalpies

Standard formation enthalpies  $\Delta h_f^\circ$  (at 298.15 K, 1 bar) are taken from `common/constants.py` in kJ/mol:

Species	$\Delta h_f^\circ$ [kJ/mol]
$\text{CH}_4$	-74.8
$\text{C}_2\text{H}_6$	-84.7
$\text{C}_3\text{H}_8$	-103.8
$\text{C}_4\text{H}_{10}$	-126.1
$\text{SO}_2$	-296.8
$\text{CO}_2$	-393.5
$\text{H}_2\text{O}(l)$	-285.5

### 4.4.1.3 Products for HHV and LHV

For each fuel species, complete combustion is considered:

- $\text{CH}_4 + 2 \text{O}_2 \rightarrow \text{CO}_2 + 2 \text{H}_2\text{O}$
- $\text{C}_2\text{H}_6 + 3.5 \text{O}_2 \rightarrow 2 \text{CO}_2 + 3 \text{H}_2\text{O}$

Builds product formation enthalpies for:

- HHV assumption: water as liquid (condensed)
- LHV assumption: water as vapour (no condensation heat recovered)

```
H2O_liq = _dHf["H2O"] # kJ/mol
H2O_vap = _dHf["H2O"] + latent_H2O * M_H2O # (kJ/kg) * (kg/mol)
```

Then, looping over the *molar* fuel composition `mol_comp = to_mole(fuel.comp)`:

```
react = 0
HHV_p = 0
LHV_p = 0
```

```

for comp, x in mol_comp.items():
    dh = _dHf.get(comp, 0)
    react += x * dh

    C, H = parse_CH(comp)
    if C is not None:
        HHV_p += x * (C * _dHf["CO2"] + (H/2) * H2O_liq)
        LHV_p += x * (C * _dHf["CO2"] + (H/2) * H2O_vap)
    elif comp == "H2S":
        HHV_p += x * (_dHf["SO2"] + H2O_liq)
        LHV_p += x * (_dHf["SO2"] + H2O_vap)
    else:
        HHV_p += x * dh
        LHV_p += x * dh

```

Here:

- `react` represents the mixture-averaged formation enthalpy of the fuel (kJ/mol),
- `HHV_p`, `LHV_p` represent the mixture-averaged formation enthalpy of the ideal products for HHV and LHV definitions.

#### 4.4.1.4 Mixture HHV and LHV (molar, then mass-based)

The mixture molar higher and lower heating values are:

$$\text{HHV}_{\text{mol}} = h_{\text{react}} - h_{\text{prod,HHV}}, \quad \text{LHV}_{\text{mol}} = h_{\text{react}} - h_{\text{prod,LHV}}$$

$$\text{HHV}_{\text{mol}} = \text{react} - \text{HHV}_p \quad \# \text{ kJ/mol}$$

$$\text{LHV}_{\text{mol}} = \text{react} - \text{LHV}_p \quad \# \text{ kJ/mol}$$

These are converted to mass-based heating values using the mixture molar mass  $M_{\text{mix}}$  from `mix_molar_mass(mol_comp)`:

$$\text{HHV}_{\text{kg}} = \text{HHV}_{\text{mol}} / M_{\text{mix}} \quad \# \text{ kJ/kg}$$

$$\text{LHV}_{\text{kg}} = \text{LHV}_{\text{mol}} / M_{\text{mix}} \quad \# \text{ kJ/kg}$$

The function returns these, together with the corresponding firing powers:

$$P_{\text{HHV}} = (\text{HHV}_{\text{kg}} * \text{fuel.mass\_flow}).\text{to}(\text{"kW"})$$

$$P_{\text{LHV}} = (\text{LHV}_{\text{kg}} * \text{fuel.mass\_flow}).\text{to}(\text{"kW"})$$


---

#### 4.4.2 Numerical results for the present fuel

For the fuel specified above, the mixture heating values are:

- Higher heating value (HHV, mass-based):

$$\text{HHV}_{\text{mix}} \approx 52 \text{ MJ/kg}$$

- Lower heating value (LHV, mass-based):

$$\text{LHV}_{\text{mix}} \approx 47 \text{ MJ/kg}$$

For the specified fuel mass flow rate:

$$\dot{m}_f = 0.5 \text{ kg/s}$$

the resulting firing rates are:

- On an HHV basis:

$$P_{\text{HHV}} = \dot{m}_f \text{HHV}_{\text{mix}} \approx 0.5 \times 52 \text{ MJ/s} \approx 26 \text{ MW}$$

- On an LHV basis (used consistently in the simulation):

$$P_{\text{LHV}} = \dot{m}_f \text{LHV}_{\text{mix}} \approx 0.5 \times 47 \text{ MJ/s} \approx 23.6 \text{ MW}$$

These correspond directly to `P_HHV` and `P_LHV` returned by `compute_LHV_HHV`.

---

#### 4.4.3 Total heat input to the boiler $Q_{\text{in}}$

The function `total_input_heat(fuel, air)` combines chemical and sensible contributions:

```
def total_input_heat(fuel, air):
    _, _, _, power_LHV = compute_LHV_HHV(fuel)
    fuel_sens = sensible_heat(fuel)
    air_sens = sensible_heat(air)
    Q_in = (power_LHV + fuel_sens + air_sens).to("kW")
    return power_LHV, Q_in
```

where `sensible_heat(stream)` uses:

$$Q_{\text{sens}} = \dot{m} c_p (T - T_{\text{ref}})$$

Both fuel and air enter at 300 K, while the reference is 298.15 K; the resulting sensible contributions are small compared with the chemical term  $P_{\text{LHV}}$  (on the order of tens of kW versus tens of MW). Therefore, numerically:

- LHV-based chemical heat input:

$$P_{\text{LHV}} \approx 23.6 \text{ MW}$$

- Total heat input including sensible:

$$Q_{\text{in}} \approx P_{\text{LHV}} + Q_{\text{sens,fuel}} + Q_{\text{sens,air}} \approx 23.6 \text{ MW} \quad (\text{increase} < 0.1\%)$$

The quantity  $Q_{\text{in}}$  in the `CombustionResult` object is thus interpreted in the rest of the boiler model as the total LHV-based heat release available to be transferred to the water/steam side.

## 4.5 Adiabatic flame temperature

The adiabatic flame temperature  $T_{\text{ad}}$  is evaluated in the model by the function `adiabatic_flame_T(air, fuel)` in `combustion/adiabatic_flame_temperature`. This routine uses Cantera and an enthalpy–pressure equilibrium (HP) calculation to determine the final equilibrium temperature and composition of the flue gas, assuming:

- complete mixing of fuel and air,
- no heat losses to the surroundings (adiabatic),
- constant system pressure (equal to the air/fuel inlet pressure),
- chemical equilibrium among all gas species in `config/flue_cantera.yaml`.

### 4.5.1 Thermodynamic formulation

Let the fuel and air streams be characterised by:

- mass flows  $\dot{m}_{\text{fuel}}, \dot{m}_{\text{air}}$ ,
- inlet temperatures  $T_{\text{fuel}}, T_{\text{air}}$ ,
- pressure  $P$ ,
- compositions (mole fractions)  $X_{\text{fuel}}, X_{\text{air}}$ .

The total inlet enthalpy rate of the unmixed reactants is

$$\dot{H}_{\text{react}} = \dot{m}_{\text{air}} h_{\text{air}}(T_{\text{air}}, P, X_{\text{air}}) + \dot{m}_{\text{fuel}} h_{\text{fuel}}(T_{\text{fuel}}, P, X_{\text{fuel}})$$

The total mass flow is

$$\dot{m}_{\text{tot}} = \dot{m}_{\text{air}} + \dot{m}_{\text{fuel}}$$

so the mixture-averaged specific enthalpy of the reactants is

$$h_{\text{target}} = \frac{\dot{H}_{\text{react}}}{\dot{m}_{\text{tot}}}$$

The adiabatic, constant-pressure equilibrium state is then defined by the constraints:

$$\begin{aligned} h_{\text{products}}(T_{\text{ad}}, P, \mathbf{X}_{\text{eq}}) &= h_{\text{target}} \\ P_{\text{out}} &= P \\ \mathbf{X}_{\text{eq}} &\text{ satisfies chemical equilibrium at } (T_{\text{ad}}, P) \end{aligned}$$

Cantera is used to enforce this condition via its HP equilibrium mode.

## 4.5.2 Implementation

Key steps from `adiabatic_flame_T`:

1. Convert the mass-based composition of fuel and air to mole fractions using `to_mole(...)` (from `combustion/mass_mole.py`).
2. Create three Cantera Solution objects using the mechanism `config/flue_cantera.yaml`:

```
gas_air = ct.Solution("config/flue_cantera.yaml", "gas_mixture")
gas_fuel = ct.Solution("config/flue_cantera.yaml", "gas_mixture")
gas_mix = ct.Solution("config/flue_cantera.yaml", "gas_mixture")
```

3. Set the inlet states of the separate streams:

```
gas_air.TPX = T_air, P_Pa, X_air
gas_fuel.TPX = T_fuel, P_Pa, X_fuel
```

4. Compute reactant enthalpy rate and target specific enthalpy:

```
Hdot_react = m_air * gas_air.enthalpy_mass + m_fuel * gas_fuel.enthalpy_mass
h_target = Hdot_react / m_tot # J/kg of mixture
```

5. Build the overall reactant composition  $\mathbf{X}_{\text{react}}$  from the molar flow rates of each component in each stream:

```
n_air = molar_flow(air.comp, air.mass_flow)
n_fuel = molar_flow(fuel.comp, fuel.mass_flow)

# Accumulate species molar flow rates
n_dot_sp = {}
X_react = {k: v / n_sum for k, v in n_dot_sp.items() }
```

6. Initialise the mixture and perform HP equilibrium:

```

gas_mix.TPX = 300.0, P_Pa, X_react      # initial guess for T
gas_mix.HP   = h_target, P_Pa          # enforce (H,P)
gas_mix.equilibrate("HP")              # chemical equilibrium

```

7. Construct the resulting flue-gas stream:

```

Y_eq = gas_mix.Y      # equilibrium mass fractions
comp_eq = {sp: Q_(float(Y_eq[i]), "") for i, sp in enumerate(gas_mix.sp)}
          if Y_eq[i] > 1e-15}

flue = GasStream(
    mass_flow = Q_(m_tot, "kg/s"),
    T          = Q_(gas_mix.T, "K"),
    P          = air.P,
    comp       = comp_eq,
)

```

The adiabatic flame temperature is then available as `flue.T` and is also stored in `CombustionResult.T_ad`.

---

### 4.5.3 Numerical result for the present case

For the given conditions:

- Fuel: natural-gas-type mixture from Section 4.1,  
 $\dot{m}_{\text{fuel}} = 0.5 \text{ kg/s}$ ,  $T_{\text{fuel}} = 300 \text{ K}$ ,  $1.013 \times 10^5 \text{ Pa}$ .
- Air: dry air at 300 K and  $1.013 \times 10^5 \text{ Pa}$ , composition from `config/air.yaml`.
- Excess air:  $\lambda = 1.1$  (10 % excess air).

the HP-equilibrium calculation yields an adiabatic flame temperature on the order of:

$$T_{\text{ad}} \approx 2,050 \text{ K} \quad (\approx 1,780^\circ\text{C})$$

This value is consistent with typical adiabatic flame temperatures for natural gas with around 10 % excess air and confirms that the combustion zone (furnace) operates at very high gas temperatures, driving strong radiative and convective heat transfer to the shell-side water/steam.

The scalar `T_ad` is passed forward and written into the boiler summary CSV (`*_boiler_summary.csv`) for reference and later comparison with non-adiabatic stack temperatures obtained from the full boiler simulation.

## 4.6 Flue-gas composition

In the combustion model two different flue-gas streams are distinguished:

1. An **equilibrium flue gas at adiabatic flame conditions** (`flue_ad`), obtained from high-temperature HP equilibrium in Cantera.
2. A **fully burnt boiler flue gas** (`flue`), obtained from pure stoichiometry with excess air and no dissociation, used throughout the heat-exchanger network.

Both are represented as `GasStream` objects and stored in the `CombustionResult`, but they serve different purposes in the boiler calculation.

---

#### 4.6.1 Definitions and distinction

- **Equilibrium flue gas (`flue_ad`)**

- Thermodynamic state: high-temperature HP equilibrium at the adiabatic flame temperature.
- Contains all equilibrium species allowed by the mechanism (major products + dissociation products + radicals).
- Used only to:
  - \* determine the adiabatic flame temperature  $T_{ad}$ ,
  - \* report equilibrium composition in diagnostics/CSV.

- **Fully burnt flue gas (`flue`)**

- Thermodynamic state: chemically frozen, fully burnt mixture at the same temperature and pressure as the equilibrium gas at burner exit.
- Contains only “engineering” products ( $CO_2$ ,  $H_2O$ ,  $SO_2$ ,  $O_2$ ,  $N_2$ ,  $Ar$ ) with no  $CO$ ,  $H_2$ ,  $NO_x$  or radicals.
- Used as the hot-side gas in all boiler heat-transfer and pressure-drop calculations.

Hence, equilibrium chemistry is confined to the flame-temperature calculation, while the boiler itself is solved with a simplified, fully burnt flue gas consistent with complete combustion and 10 % excess air.

---

#### 4.6.2 Equilibrium flue gas at adiabatic conditions

The adiabatic flame calculation is performed in `combustion/adiabatic_flame_temperature` via the function `adiabatic_flame_T(air, fuel)`:

- The inlet **air** and **fuel** streams are:

- represented as `GasStream` objects (mass flow,  $T$ ,  $P$ , mass fractions),
- converted to mole fractions (`to_mole`) and set into separate Cantera Solution objects (`gas_air`, `gas_fuel`) based on `config/flue_cantera.yaml`.

- A mixed-reactant state is constructed at constant pressure:
  - Total enthalpy flow of reactants:

$$\dot{H}_{\text{react}} = \dot{m}_{\text{air}} h_{\text{air}} + \dot{m}_{\text{fuel}} h_{\text{fuel}}$$

- Target specific enthalpy:

$$h_{\text{target}} = \dot{H}_{\text{react}} / \dot{m}_{\text{tot}}$$

- Overall reactant mole fractions are built from molar flow rates of air and fuel.
- The mixture is then set in Cantera (gas\_mix) with:

- composition  $X_{\text{react}}$ ,
  - pressure  $P = P_{\text{air}}$ ,
  - specific enthalpy  $h = h_{\text{target}}$ ,
  - and equilibrated under HP constraints:

```
gas_mix.TPX = 300.0, P_Pa, X_react    # T placeholder
gas_mix.HP = h_target, P_Pa
gas_mix.equilibrate("HP")
```

- After equilibrium:
  - The **adiabatic flame temperature** is gas\_mix.T.
  - The **equilibrium mass fractions** are read from gas\_mix.Y:

```
Y_eq = gas_mix.Y
comp_eq = {
    sp: Q_(float(Y_eq[i]), "")
    for i, sp in enumerate(gas_mix.species_names)
    if Y_eq[i] > 1e-15
}
```

- These are stored in the equilibrium flue-gas stream:

```
flue_ad = GasStream(
    mass_flow = Q_(m_tot, "kg/s"),
    T          = Q_(gas_mix.T, "K"),
    P          = air.P,
    comp       = comp_eq,
)
```

Typical equilibrium composition ( $\lambda = 1.1$ , natural gas,  $T_{\text{ad}} \approx 2050$  K) is:

- Major species:
  - $\text{CO}_2 \approx 0.085\text{--}0.095$
  - $\text{H}_2\text{O} \approx 0.075\text{--}0.085$



- $O_2 \approx 0.020\text{--}0.030$  (excess air)
- $N_2 \approx 0.78\text{--}0.80$
- Dissociation / minor species:
  - $CO \approx 10^{-3}$
  - $H_2 \approx 10^{-4}$
  - $NO \approx 10^{-4}\text{--}10^{-5}$
  - $OH, O, \text{radicals} < 10^{-6}$
  - $SO_2 = 10^{-4}$  (from fuel  $H_2S$ )

This composition is physically consistent with high-temperature equilibrium at  $2000K$  and slight dissociation.

The object `flue_ad` is stored in `CombustionResult` and is only used to:

- provide  $T_{ad}$  and equilibrium composition to the boiler summary CSV,
- support diagnostic post-processing.

It is **not** used directly in the heat-exchanger network.

### 4.6.3 Fully burnt boiler flue gas

The boiler thermal model requires a chemically simple flue-gas mixture to compute heat transfer and pressure drop. For that purpose a **fully burnt** flue gas is constructed in `combustion/flue.py` and `combustion/combustor.py`:

1. In `Combustor.run()` the air mass flow is first set from stoichiometry plus excess air:

```
air.mass_flow = air_flow_rates(air, fuel, self.excess_air_
```

2. The fully burnt flue-gas composition is then computed from pure stoichiometry:

```
mass_comp_burnt, m_dot_flue = from_fuel_and_air(fuel, air)
```

- `from_fuel_and_air` assumes complete oxidation of:
  - C-containing species  $\rightarrow CO_2$ ,
  - $H \rightarrow H_2O$ ,
  - $S \rightarrow SO_2$ ,
- including  $CO_2$  and  $H_2O$  already present in the inlet fuel and air.
- The allowed product set is:
  - $CO_2, H_2O, SO_2, O_2, N_2, Ar$ .
- Residual  $O_2$  is determined by the imposed excess air ratio  $\lambda$ ; there is no  $CO$ ,  $H_2$ ,  $NO_x$ , or radicals in this stream.

Internally, `from_fuel_and_air` works with molar balances:

- determines stoichiometric  $O_2$  demand per mole of fuel (`stoich_o2_required_per_`

- combines fuel and air mole fractions to get:

$$\dot{n}_{\text{CO}_2}, \dot{n}_{\text{H}_2\text{O}}, \dot{n}_{\text{SO}_2}, \dot{n}_{\text{O}_2}, \dot{n}_{\text{N}_2}, \dot{n}_{\text{Ar}}$$

- normalises by total moles to obtain mole fractions, converts to mass fractions (to `mass`), and returns both:
  - `mass_comp` (mass fractions),
  - `m_dot` (total mass flow of flue gas).

3. The fully burnt flue-gas stream is then created as:

```
flue_boiler = GasStream(
    mass_flow = Q_(m_dot_flue, "kg/s"),
    T          = T_ad,          # assume recombination to near T
    P          = air.P,
    comp       = {sp: Q_(y, "") for sp, y in mass_comp_burnt})
```

4. `CombustionResult` is populated with both flue streams:

```
return CombustionResult(
    LHV          = power_LHV,
    Q_in         = Q_in,
    T_ad         = T_ad,
    flue         = flue_boiler, # fully burnt flue used in
    flue_ad      = flue_ad,     # equilibrium flue at T_ad (
    fuel_LHV_mass = LHV_mass,
    fuel_P_LHV    = P_LHV,
)
```

The **boiler solver** (`run_hx`) always receives `combustion.flue` (i.e. `flue_boiler`) as its gas inlet, and this fully burnt composition is used for:

- gas properties (`cp`, `p`, `μ`, `k`),
- heat-transfer coefficients,
- radiative heat transfer (emissivity based on  $\text{CO}_2/\text{H}_2\text{O}/\text{SO}_2$ ),
- pressure-drop estimates and stack temperature.

Thus, the equilibrium flue gas provides a physically consistent high-temperature reference, while the fully burnt flue gas represents the practical working fluid in the convective–radiative sections of the boiler.

---

#### 4.6.4 Output fields

The flue-gas information exposed to the rest of the model and to the post-processing is encapsulated in `CombustionResult`:

```

@dataclass(frozen=True)
class CombustionResult:
    LHV: Q_
    Q_in: Q_
    T_ad: Q_
    flue: GasStream          # fully-burnt flue used in all boiler HX calculations
    flue_ad: GasStream | None = None  # equilibrium flue at boiler inlet
    fuel_LHV_mass: Q_ | None = None
    fuel_P_LHV: Q_ | None = None

```

The relevant report/CSV entries are:

Field	Meaning
T_ad	Adiabatic flame temperature from HP equilibrium
flue_adGasStream	of equilibrium flue gas (adiabatic composition, diagnostics)
flue	GasStream of fully burnt flue gas used in all boiler HX calculations

This completes the description of how flue-gas composition is defined, distinguished, and used in the boiler model.

# Chapter 5

## Heat-Transfer Calculations

### 5.1 Fundamental heat-balance equations

The boiler is modelled as a one-dimensional counter-current heat exchanger composed of six stages (HX<sub>1</sub>–HX<sub>5</sub>). Heat transfer is resolved along the gas flow direction  $x$ , while water flows in the opposite direction. Each stage is discretized into segments of length  $dx$ ; all local quantities are defined per unit length.

- Notation (per segment)
  - $x$  – axial coordinate along the gas flow [m]
  - $dx$  – marching step in  $x$  [m]
  - $\dot{m}_g, \dot{m}_w$  – gas and water mass flow rates [kg/s]
  - $T_g(x), T_w(x)$  – bulk gas and water temperatures [K]
  - $T_{gw}(x), T_{ww}(x)$  – gas-side and water-side wall temperatures [K]
  - $h_g(x), h_w(x)$  – total gas-side and water-side heat-transfer coefficients [W/m<sup>2</sup>·K]
  - $P_g, P_w$  – gas-side and water-side wetted perimeters [m]
  - $q'(x)$  – linear heat flux (heat per unit length) [W/m]
  - $UA'(x)$  – overall conductance per unit length [W/K/m]
- 

### 5.2 Local energy balance

For each differential segment of length  $dx$ , the model enforces a one-dimensional steady-state energy balance between the gas, the water and the tube wall:

- Heat transferred across the wall:

$$q'(x) = U A'(x) [T_g(x) - T_w(x)]$$

- Relation to the segment duty:

$$dQ(x) = q'(x) dx$$

- Gas stream:

$$dQ(x) = -\dot{m}_g dh_g(x) \Rightarrow \frac{dh_g}{dx} = -\frac{q'(x)}{\dot{m}_g}$$

- Water stream:

$$dQ(x) = +\dot{m}_w dh_w(x) \Rightarrow \frac{dh_w}{dx} = +\frac{q'(x)}{\dot{m}_w}$$

In the numerical implementation these equations are applied in finite-difference form over each marching step:

$$Q_{\text{step}} = q'(x) \Delta x$$

$$\Delta h_g = -\frac{Q_{\text{step}}}{\dot{m}_g}, \quad \Delta h_w = +\frac{Q_{\text{step}}}{\dot{m}_w}$$


---

## 5.3 Overall conductance and resistance network

The overall conductance per unit length  $U A'(x)$  is obtained from a radial series of thermal resistances per unit length:

- Gas-side convection:

$$R'_g = \frac{1}{h_g(x) P_g}$$

- Gas-side fouling:

$$R'_{fg} = R'_{fi}(P_g) \quad (\text{from specified fouling thickness and conductivity})$$

- Tube wall:

$$R'_w = \frac{\ln(D_o/D_i)}{2\pi k_w}$$

- Water-side fouling:

$$R'_{fc} = R'_{fo}(P_w)$$

- Water-side convection:

$$R'_c = \frac{1}{h_w(x) P_w}$$

where  $D_i$  and  $D_o$  are the tube inner and outer diameters and  $k_w$  is the tube wall thermal conductivity. Combining these contributions:

$$\frac{1}{UA'(x)} = R'_g + R'_{fg} + R'_w + R'_{fc} + R'_c$$

or equivalently,

$$UA'(x) = \left[ \frac{1}{h_g P_g} + R'_{fg} + R'_w + R'_{fc} + \frac{1}{h_w P_w} \right]^{-1}$$

The linear heat flux then follows directly:

$$q'(x) = UA'(x) [T_g(x) - T_w(x)]$$


---

## 5.4 Stage- and boiler-level duties

For a stage of length  $L_j$ , the stage heat duty and stage-level conductance are obtained by integrating the local quantities along  $x$ :

$$Q_{\text{stage},j} = \int_0^{L_j} q'(x) dx \approx \sum_i q'_i \Delta x_i$$

$$(UA)_j = \int_0^{L_j} UA'(x) dx \approx \sum_i UA'_i \Delta x_i$$

The total useful boiler duty is the sum of all stage duties:

$$Q_{\text{useful}} = \sum_{j=1}^6 Q_{\text{stage},j}$$

These integrated quantities are later used in the performance and efficiency evaluation (Section 7) and for constructing stage-wise summary tables.

## 5.5 Gas-side

Gas-side heat transfer is computed with geometry-aware correlations based on local gas properties from Cantera (GasProps) and stage-specific geometry from the GeometryBuilder. For each marching step, the total gas-side HTC is split into a convective and a radiative contribution:

$$h_{g,\text{tot}} = h_{g,\text{conv}} + h_{g,\text{rad}}$$

The implementation uses the helper `gas_htc_parts(g, spec, T_{gw})`, which returns  $(h_{g,\text{conv}}, h_{g,\text{rad}})$  in  $\text{W/m}^2\cdot\text{K}$ , and then sums them in `gas_htc`.

---

### 5.5.1 Single-tube and reversal-chamber (internal)

Stages of kind "single\_tube" and "reversal\_chamber" are treated as internal forced convection in a circular duct. The characteristic quantities are:

- Diameter:  $D = D_i$  (tube inner diameter)
- Length:  $L$  (stage inner length)
- Flow area:  $A = A_{\text{hot,flow}}$  (from geometry builder)
- Velocity:

$$V = \frac{\dot{m}_g}{\rho_g A}$$

- Reynolds and Prandtl numbers:

$$\text{Re} = \frac{\rho_g V D}{\mu_g}, \quad \text{Pr} = \frac{c_{p,g} \mu_g}{k_g}$$

Local gas properties  $\rho_g, \mu_g, k_g, c_{p,g}$  are obtained from the Cantera mixture at the local gas temperature and pressure.

Laminar/developing flow (Graetz-type)

For  $Re < 2300$ , uses a Graetz correlation for thermally developing laminar flow:

$$Gz = Re Pr \frac{D}{L}$$

$$Nu = 3.66 + \frac{0.0668 Gz}{1 + 0.04 Gz^{2/3}}$$

(Incropera et al. 2011)

Turbulent flow (Gnielinski with Petukhov friction factor)

For  $Re \geq 2300$ , the Gnielinski correlation is applied with a Petukhov friction factor:

$$f = (0.79 \ln Re - 1.64)^{-2}$$

(Munson et al. 2013)

$$Nu = \frac{\frac{f}{8}(Re - 1000) Pr}{1 + 12.7 \sqrt{\frac{f}{8}} (Pr^{2/3} - 1)}$$

(Incropera et al. 2011) The local convective heat-transfer coefficient is then:

$$h_{g,conv} = \frac{Nu k_g}{D}$$

(Incropera et al. 2011)

This same internal correlation is used for "single\_tube", "reversal\_chamber" and "tube\_bank" gas-side flow (see below).

---

### 5.5.2 Tube-bank (internal)

Stages "tube\_bank" correspond to tube bundles inside the shell. In this model, the gas side is still treated as internal flow inside the tubes:

- Hot side (gas): inside tubes (inner diameter  $D_i$ ), using the same internal forced convection model as in Section 5.2.1.



Thus the gas-side convective HTC in tube-bank stages is:

$$h_{g,\text{conv}}^{(\text{HX3,5})} = \frac{\text{Nu}_{\text{internal}}(\text{Re}, \text{Pr}) k_g}{D_i}$$

with  $\text{Nu}_{\text{internal}}$  given by the Graetz/Gnielinski formulation above, and  $\text{Re}$ ,  $\text{Pr}$  computed from the local gas properties and tube hydraulic diameter.

---

### 5.5.3 Economizer (external)

The economizer "economiser" stage reverses the roles: gas flows outside the tubes in crossflow, while water flows inside. The gas-side convection is then modelled as external crossflow over a tube bank.

Key geometry quantities (from GeometryBuilder for the economizer):

- Tube outer diameter:  $D = D_o$
- Gas-side crossflow area:  $A_{\text{bulk}} = A_{\text{hot,flow}}$
- Optional maximum/mean velocity factor:

$$V_{\text{bulk}} = \frac{\dot{m}_g}{\rho_g A_{\text{bulk}}}, \quad V = u_{\text{max}} V_{\text{bulk}}$$

where  $u_{\text{max}}$  is calculated depending on the tube bank arrangement and spacing between tubes.

- Reynolds and Prandtl numbers:

$$\text{Re} = \frac{\rho_g V D}{\mu_g}, \quad \text{Pr} = \frac{c_{p,g} \mu_g}{k_g}$$

For "economiser" stages the primary correlation is a banded Zukauskas form for crossflow over tube banks:

$$\text{Nu} = C \text{Re}^m \text{Pr}^n$$

(Incropera et al. 2011)

where the coefficients  $C, m$  are selected from standard bands as a function of Reynolds number and tube arrangement (inline vs staggered), and the exponent  $n$  is:

$$n = \begin{cases} 0.36, & \text{Pr} \leq 10 \\ 0.25, & \text{Pr} > 10 \end{cases}$$

If Re falls outside the tabulated bands, the model falls back to the Churchill–Bernstein correlation for crossflow over a single cylinder:

$$\text{Nu} = 0.3 + \frac{0.62 \text{Re}^{1/2} \text{Pr}^{1/3}}{[1 + (0.4/\text{Pr})^{2/3}]^{1/4}} \left[ 1 + \left( \frac{\text{Re}}{282000} \right)^{5/8} \right]^{4/5}$$

(Incropera et al. 2011) The gas-side convective HTC in the economizer is then:

$$h_{g,\text{conv}}^{(\text{HX6})} = \frac{\text{Nu } k_g}{D_o}$$

(Incropera et al. 2011)

---

#### 5.5.4 Gas radiation model

Radiative heat transfer from the flue gas to the furnace surfaces is explicitly accounted for by a participating-medium model for the  $\text{H}_2\text{O}/\text{CO}_2$  mixture. The implementation follows a simplified Smith–Shen–Friedman style four-gray model.

For each step, the gas emissivity is computed as:

1. Partial pressures of participating species:

$$p_{\text{H}_2\text{O}} = y_{\text{H}_2\text{O}} P, \quad p_{\text{CO}_2} = y_{\text{CO}_2} P$$

(Modest 2013) where  $y_i$  are molar (or mass-fraction-equivalent) composition entries from the flue gas stream, and  $P$  is the local gas pressure.

2. Mean beam length:

$$L_b = \begin{cases} L_{\text{rad,override}}, & \text{if specified in the stage} \\ 0.9 D_{h,\text{gas}}, & \text{otherwise} \end{cases}$$

(Modest 2013) with  $D_{h,\text{gas}}$  the gas-side hydraulic diameter.

3. Effective optical thickness in each gray band:

$$p_{\text{ratio}} = \frac{p_{\text{H}_2\text{O}} + p_{\text{CO}_2}}{P_{\text{atm}}}$$

(Modest 2013)

$$\tau_j = K_j \left( \frac{T}{1000 \text{ K}} \right)^{T_{\text{exp}}} p_{\text{ratio}} L_b$$

(Modest 2013)

where  $K_j$  and weighting factors  $A_j$  are fixed band coefficients,  $T$  is the gas temperature, and  $T_{\text{exp}}$  is a temperature exponent (default 0.65, configurable per stage via `rad_Texp`).

4. Total gas emissivity:

$$\varepsilon_g = 1 - \sum_{j=1}^4 A_j \exp(-\tau_j)$$

(Modest 2013) with  $\varepsilon_g$  constrained to  $[0, 1]$ .

A mean-film temperature is used for the linearized radiative HTC:

$$T_{\text{film}} = \frac{T_g + T_{gw}}{2}$$

$$h_{g,\text{rad}} = 4 \sigma F \varepsilon_g T_{\text{film}}^3$$

(Modest 2013)

where:

- $\sigma$  is the Stefan–Boltzmann constant,
- $F$  is an effective view factor (default 1.0 or stage-specific `rad_F`).

The gas-side total HTC reported and used in the resistance network is then:

$$h_{g,\text{tot}} = h_{g,\text{conv}} + h_{g,\text{rad}}$$

and the corresponding convective/radiative contributions to the linear heat flux are tracked via:

$$q'_{\text{conv}} = q' \frac{h_{g,\text{conv}}}{h_{g,\text{tot}}}, \quad q'_{\text{rad}} = q' - q'_{\text{conv}}$$

These diagnostics are later integrated on a per-stage basis to quantify the share of convective vs radiative heat transfer in each section of the boiler.

## 5.6 Water-side

Water-side heat transfer is modelled with geometry-dependent correlations using local water properties from the `WaterProps` helper. The water side appears in two configurations:

1. Water inside tubes (single-tube, reversal chamber, economizer)
2. Water outside tubes in crossflow (tube-bank stages HX\_3 and HX\_5)

The total water-side HTC is computed at each marching step as:

$$h_w = h_{w,\text{conv}}$$

Water-side radiation is neglected.

In the present work, the water-side model is used in two distinct regimes:

- HX\*1–HX\_5 are treated as boiling surfaces in contact with a pool at saturation temperature. In these stages the bulk water temperature is forced to  $T^* \text{ sat}(p)$  and the heat-transfer coefficient is obtained from a pure pool-boiling correlation.
- HX\_6 (economizer) is treated as a single-phase / flow-boiling tube bundle with water flowing inside the tubes and heated by the flue-gas crossflow.

The underlying implementation is more general (it contains a full Chen-type flow-boiling formulation valid for internal forced convection), but for the final boiler calculations this capability is only used in the economizer; in HX\_1–HX\_5 the water side is deliberately simplified to a pool-boiling model.

---

### 5.6.1 Economizer (internal)

For the economiser stage (kind "economiser", HX<sub>6</sub>), where water flows inside the tubes, the model uses standard internal-flow correlations augmented with a viscosity-ratio correction and, when needed, a Chen-type flow-boiling enhancement. The tube inner diameter  $D_i$  is used as characteristic length.

### 5.6.1.1 Velocity and nondimensional groups

$$V_w = \frac{\dot{m}_w}{\rho_w A_{\text{cold,flow}}}$$
$$\text{Re}_w = \frac{\rho_w V_w D_i}{\mu_w}, \quad \text{Pr}_w = \frac{c_{p,w} \mu_w}{k_w}$$

Local water-side properties  $\rho_w, \mu_w, k_w, c_{p,w}$  are evaluated at the bulk water temperature.

### 5.6.1.2 Laminar regime ( $\text{Re} < 2300$ )

For fully developed laminar internal flow in a circular tube:

$$\text{Nu}_w = 3.66$$

(Incropera et al. 2011) For developing laminar flow, the same Graetz form used on the gas side is applied:

$$\text{Gz}_w = \text{Re}_w \text{Pr}_w \frac{D_i}{L}$$
$$\text{Nu}_w = 3.66 + \frac{0.0668 \text{Gz}_w}{1 + 0.04 \text{Gz}_w^{2/3}}$$

(Incropera et al. 2011)

### 5.6.1.3 Turbulent regime ( $\text{Re} \geq 2300$ )

The Gnielinski correlation is used:

$$f_w = (0.79 \ln \text{Re}_w - 1.64)^{-2}$$

(Munson et al. 2013)

$$\text{Nu}_w = \frac{\frac{f_w}{8} (\text{Re}_w - 1000) \text{Pr}_w}{1 + 12.7 \sqrt{\frac{f_w}{8}} (\text{Pr}_w^{2/3} - 1)}$$

(Incropera et al. 2011) In the implementation, the Nusselt number is multiplied by a viscosity-ratio correction  $(\mu_b/\mu_w)^{0.11}$  evaluated at bulk and wall temperatures, following the common Gnielinski extension for heated internal flow.

Finally:

$$h_{w,\text{conv}} = \frac{\text{Nu}_w k_w}{D_i}$$

(Incropera et al. 2011)

---

## 5.6.2 Tube-bank (external)

In the boiling sections (HX<sub>1</sub>–HX<sub>5</sub>) the water occupies the shell-side region around the heated tubes. When a crossflow description is needed (e.g. in HX<sub>3</sub> and HX<sub>5</sub>), a Zukauskas-type correlation is applied for flow over a tube bundle on the water side, using the outer tube diameter  $D_o$  and the cold-side flow area  $A_{\text{cold,flow}}$  supplied by the geometry builder.

### 5.6.2.1 Geometry inputs from GeometryBuilder

- Tube outer diameter:  $D_o$
- Cold-side flow area:  $A_{\text{cold,flow}}$
- Water velocity:

$$V_w = \frac{\dot{m}_w}{\rho_w A_{\text{cold,flow}}}$$

- Reynolds and Prandtl numbers:

$$\text{Re}_w = \frac{\rho_w V_w D_o}{\mu_w}, \quad \text{Pr}_w = \frac{c_{p,w} \mu_w}{k_w}$$

### 5.6.2.2 Zukauskas banded correlation

$$\text{Nu}_w = C \text{Re}_w^m \text{Pr}_w^n$$

Coefficient selection:

- $C, m$  chosen based on the Reynolds band and bundle arrangement (inline or staggered).
- Exponent  $n$ :

$$n = \begin{cases} 0.36, & \text{Pr}_w \leq 10 \\ 0.25, & \text{Pr}_w > 10 \end{cases}$$

If the Reynolds number lies outside the valid Zukauskas range, the model falls back to Churchill–Bernstein:

$$\text{Nu}_w = 0.3 + \frac{0.62 \text{Re}_w^{1/2} \text{Pr}_w^{1/3}}{[1 + (0.4/\text{Pr}_w)^{2/3}]^{1/4}} \left[ 1 + \left( \frac{\text{Re}_w}{282000} \right)^{5/8} \right]^{4/5}$$

(Incropera et al. 2011)

The external HTC is then:

$$h_{w,\text{conv}} = \frac{\text{Nu}_w k_w}{D_o}$$


---

### 5.6.3 Treatment of boiling

Boiling is treated differently in the pool-boiling stages (HX\_1–HX\_5) and in the economiser (HX\_6).

#### 5.6.3.1 Pool-boiling

For stages flagged as `pool_boiling = true` (HX\_1–HX\_5), the water side is deliberately simplified to a pure pool-boiling model:

- The bulk water temperature entering the wall-energy balance is fixed at the saturation temperature corresponding to the local pressure:

$$T_w = T_{\text{sat}}(p_w).$$

- The water-side heat-transfer coefficient is taken from a Cooper-type pool-boiling correlation:

$$h_{w,\text{nb}} = h_{\text{Cooper}}(p_w, q'')$$

(Incropera et al. 2011) where  $q''$  is the local heat flux on the water side and the roughness of the boiling surface enters through the correlation.

- This nucleate-boiling coefficient is used directly as the water-side HTC:

$$h_w = h_{w,\text{nb}},$$

and the region is always tagged as “boiling” in the post-processing.

In other words, HX\_1–HX\_5 are modelled as heated surfaces immersed in a saturated pool, with boiling controlled by the local heat flux and surface roughness rather than by a detailed prediction of the liquid velocity. This reflects the natural-circulation behavior of the boiler riser and furnace sections and follows the modelling simplification requested for the thesis.

### 5.6.3.2 Economizer

For the economizer stage HX\_6 (`pool_boiling = false`), the model uses a more general internal-flow formulation that can represent both single-phase convection and flow boiling:

1. Boiling detection.

A helper function checks whether the local state falls into the saturation enthalpy interval  $[h_f(p), h_g(p)]$  or, for slightly subcooled liquid, whether the wall superheat exceeds a threshold. If neither condition is met, the flow is treated as single-phase liquid.

2. Single-phase regime.

In single-phase operation, the water-side HTC is computed from an internal forced-convection correlation (Gnielinski with viscosity-ratio correction), as described in Section 5.3.1.

3. Flow-boiling regime (Chen-type model).

When boiling is detected, the HTC is assembled from a liquid-only contribution and a nucleate-boiling contribution:

$$h_{lo} = \text{single-phase liquid HTC at } T_{\text{sat}}(p),$$

$$h_{nb} = h_{\text{Cooper}}(p, q''),$$

$$h_w = F h_{lo} + S h_{nb}.$$

(Incropera et al. 2011) The factor  $F$  accounts for the effect of two-phase flow on the convective heat transfer (via a Martinelli-type parameter), while  $S$  modulates the nucleate-boiling contribution as a function of Reynolds number and mass flux. Both are bounded to remain within reasonable engineering limits.

In the present thesis, this full Chen-type flow-boiling capability is only exercised in the economizer stage. In the main boiling sections (HX\_1–HX\_5), where circulation is dominated by buoyancy and the flow pattern is closer to pool boiling, the simpler pool-boiling representation described above is preferred.

---

## 5.7 Per-step resistance insertion

The water-side resistance per unit length used in the overall  $UA'$  assembly is:



$$R'_c = \frac{1}{h_w P_w}$$

where the wetted perimeter is:

- $P_w = \pi D_i$  when water is inside the tubes.
- $P_w = N_{\text{tubes}} \pi D_o$  effective per bundle pitch when water is outside tubes, handled automatically by GeometryBuilder.

Fouling is added in series:

$$R'_{fc} = \frac{\delta_{f,\text{water}}}{k_{f,\text{water}} P_w}$$

Total water-side contribution:

$$R'_{w,\text{side}} = R'_{fc} + R'_c$$

This resistance is passed into the overall conductance formulation (Section 5.1.2).

---

## 5.8 Wall-temperature update and thermal convergence

The tube wall temperatures on the gas and water sides,  $T_{gw}$  and  $T_{ww}$ , are updated using a two-node wall model in each marching step.

Given  $q'(x)$ , the wall-side energy balances yield:

$$T_{gw} = T_g - \frac{q'}{h_{g,\text{tot}}}$$

$$T_{ww} = T_w + \frac{q'}{h_w}$$

The wall conduction temperature drop is:

$$\Delta T_{\text{wall}} = T_{gw} - T_{ww}$$

which is also equal to:

$$\Delta T_{\text{wall}} = q' [R'_{fg} + R'_w + R'_{fc}]$$

A consistency check is applied; if the implied wall temperature difference from conduction differs from the one implied by convection, the marching solver iterates the HTC evaluation once with relaxed updates (default under-relaxation factor 0.35). Full Picard iteration is omitted for performance reasons.

In the actual implementation this consistency check is performed by iterating on  $T_{gw}$ ,  $T_{ww}$ , and  $q'$  using the full resistance network (gas convection, gas fouling, wall, water fouling, water convection), with an under-relaxation factor applied to both wall temperatures and the linear heat flux.

If temperature overshoot (negative film coefficient, reversed driving force) is detected within a step, the step is automatically halved and recomputed.

# Chapter 6

## Hydraulic Calculations

Hydraulic behaviour is extracted directly from the solver through the per-step pressure-drop decomposition implemented in `heat/solver.py(_gas_dp_components, pressure_drop_gas)` and accumulated at the stage level in `heat/solver.py::solve` and in the boiler summary computed by `heat/postproc.py::summary_from_profile`

The model divides gas-side pressure losses into:

- Frictional losses:  
Computed by Colebrook–White (turbulent), laminar  $64/\text{Re}$ , and a linear transitional blend for  $2300 < \text{Re} < 4000$ .  
The per-step drop is

$$\Delta P_{\text{fric}} = -f \frac{\Delta x}{D_h} \left( \frac{\rho V^2}{2} \right)$$

where  $f$  is obtained from `_friction_factor()` and hydraulic diameter, velocity, and density come from the local gas state.

- Minor losses:  
Applied using per-stage catalogue  $K$ -values.  
For reversal chambers, inlet/outlet nozzle  $K$  plus bend-equivalent loss are included; tube-banks default to zero unless specified.  
In `solve_stage`, the total per-stage loss coefficient  $K_{\text{sum}}$  is uniformly distributed across  $N$  steps:

$$K_{\text{per step}} = \frac{K_{\text{sum}}}{N}$$

The per-step minor loss is

$$\Delta P_{\text{minor}} = -K_{\text{per step}} \left( \frac{\rho V^2}{2} \right)$$

- Total gas-side drop:

$$\Delta P_{\text{total}} = \Delta P_{\text{fric}} + \Delta P_{\text{minor}}$$

Water-side pressure losses are intentionally not included in this model (water at constant pressure).

---

## 6.1 Gas-Side $\Delta P$ per Stage

During each call to `solve_stage`, the solver marches through all steps and accumulates:

- `dP_stage_fric`
- `dP_stage_minor`
- `dP_stage_total`

These appear in each stage row of `summary_rows` returned by `run_hx()`. An example schema from `summary_from_profile()`:

```
"ΔP_stage_fric[Pa]": dP_fric,
"ΔP_stage_minor[Pa]": dP_minor,
"ΔP_stage_total[Pa]": dP_total,
```

Values are integrated over the entire stage length:

$$\Delta P_{\text{stage}} = \sum_{i=1}^N \Delta P(i)$$


---

## 6.2 Water-Side $\Delta P$ per Stage

The present solver does not compute water-side frictional or accelerational pressure losses.

From the code (`update_water_after_step`), pressure remains constant:

```
WaterStream(mass_flow=w.mass_flow, h=h_new, P=w.P)
```

Thus:

- Water-side  $\Delta P$  per stage = 0 Pa
- Total water-side  $\Delta P$  = 0 Pa

This assumption is consistent with pool-boiling and saturated-drum configurations where the water is not routed through high-velocity conduits.

---

## 6.3 Total Boiler $\Delta P$ and Stack Pressure

The boiler-level gas-side pressure drop is assembled in the `TOTAL_BOILER` row of `summary_from_profile()`:

```
"ΔP_stage_fric[Pa]": dP_total_fric,
"ΔP_stage_minor[Pa]": dP_total_minor,
"ΔP_stage_total[Pa]": dP_total_total,
```

This yields:

- Total frictional drop:

$$\Delta P_{\text{fric,tot}} = \sum_{k=1}^6 \Delta P_{\text{fric},k}$$

- Total minor-loss drop:

$$\Delta P_{\text{minor,tot}} = \sum_{k=1}^6 \Delta P_{\text{minor},k}$$

- Overall boiler gas-side drop:

$$\Delta P_{\text{boiler}} = \Delta P_{\text{fric,tot}} + \Delta P_{\text{minor,tot}}$$

Stack exit pressure is simply the outlet gas pressure after stage 6:

`gas_out.P`

reported separately in the boiler summary.

## 6.4 Consolidated $\Delta P$ Table (from solver output)

A typical extracted table structure (values populated after running `main.py`):

Stage	Kind	$\Delta P_{\text{fric}}$ [Pa]	$\Delta P_{\text{minor}}$ [Pa]	$\Delta P_{\text{total}}$ [Pa]
HX_1	single_tube	...	...	...
HX_2	reversal_chamber	...	...	...
HX_3	tube_bank	...	...	...
HX_4	reversal_chamber	...	...	...
HX_5	tube_bank	...	...	...
HX_6	economiser	0	0	0
TOTAL	—	$\Sigma$	$\Sigma$	$\Sigma$

$HX_6$  (economiser) contributes zero  $\Delta P$  by design (`_gas_dp_components` returns 0 for this stage).

The table is directly generated as part of `summary_rows` once `main.py` completes the mass-flow/efficiency iteration and writes final CSVs.

# Chapter 7

## Boiler Performance Results

This section summarizes the boiler level performance obtained from the coupled combustion–heat-transfer simulation. All numerical values are extracted from the stage summary and boiler summary data produced by the post-processing step (fields `Q_stage[MW]`, `UA_stage[MW/K]`, `η_direct[-]`, `η_indirect[-]`, `Q_total_useful[MW]`, `Q_in_total[MW]`, `P_LHV[MW]`, `stack_temperature[°C]` etc.).

### 7.1 Energy balance ( $Q_{in}$ , $Q_{useful}$ )

The total useful heat transferred from the flue gas to the water/steam side is obtained by integrating the local line heat flux  $q'(x)$  over all stages:

$$Q_{useful} = \sum_{k=1}^6 Q_{stage,k} = \sum_{k=1}^6 \int_{stage\ k} q'(x) dx$$

In the implementation this appears as the sum of `Q_stage[MW]` over all stages in `summary_rows`, with the boiler-level result reported in the `TOTAL_BOILER` row as `Q_total_useful[MW]`.

The total input heat from combustion  $Q_{in}$  is taken from the combustion module as the rate of heat release from complete fuel burnout (field `Q_in_total[MW]`):

$$Q_{in} = Q_{in,total}$$

For reference, the firing rate on an LHV basis is also reported as `P_LHV[MW]`, obtained from the fuel lower heating value and the fuel mass flow rate.

A concise numerical statement:

- $Q_{in} = Q_{in,total} =$
- $Q_{useful} = Q_{total,useful} =$

## 7.2 Efficiencies (direct and indirect)

Two boiler efficiencies are reported:

- Direct efficiency (LHV basis)  
Direct efficiency is defined as the ratio of useful heat transferred to the firing rate based on fuel LHV:

$$\eta_{\text{direct}} = \frac{Q_{\text{useful}}}{P_{\text{LHV}}}$$

where  $P_{\text{LHV}}$  is the firing capacity (field  $P_{\text{LHV}}$  [MW] ).

- Indirect efficiency (heat-balance basis)  
Indirect efficiency is defined as the ratio of useful heat to the total heat released by combustion:

$$\eta_{\text{indirect}} = \frac{Q_{\text{losses}}}{Q_{\text{in}}}$$

In the post-processing, these appear as the boiler-level fields:

- Direct (LHV) efficiency:  $\eta_{\text{direct}} =$
- Indirect efficiency:  $\eta_{\text{indirect}} =$

## 7.3 Steam generation rate and mass-flow convergence

The water/steam mass flow rate is not prescribed but obtained iteratively from an assumed overall boiler efficiency and the combustion heat input. At each iteration  $n$  the code:

1. Assumes an efficiency  $\eta^{(n)}$ .
2. Computes the target useful duty:

$$Q_{\text{target}}^{(n)} = \eta^{(n)} Q_{\text{in}}$$

3. Determines the required water mass flow  $\dot{m}_w^{(n)}$  from the enthalpy rise between feed-water and saturated steam at drum pressure:

$$\dot{m}_w^{(n)} = \frac{Q_{\text{target}}^{(n)}}{h_{\text{steam}}(P_{\text{drum}}) - h_{\text{fw}}}$$



4. Runs the full multi-stage heat-exchanger model with  $\dot{m}_w^{(n)}$  and reads back the resulting indirect efficiency  $\eta_{\text{indirect}}^{(n)}$ .
5. Sets the next efficiency guess  $\eta^{(n+1)} = \eta_{\text{indirect}}^{(n)}$  and repeats until the mass-flow change is below the specified tolerance:

$$|\dot{m}_w^{(n)} - \dot{m}_w^{(n-1)}| < 10^{-3} \text{ kg/s}$$

The final converged values to be reported are:

- Converged feedwater/steam mass flow:

$$\dot{m}_w = [m\_w, \text{ kg/s}]$$

- Number of outer iterations to achieve  $|\Delta \dot{m}_w| < 10^{-3} \text{ kg/s}$ :

$$N_{\text{iter}} = [N]$$

In the narrative, this subsection should state that the mass-flow/efficiency fixed point converged and that the final efficiency used in the performance summary is the converged  $\eta_{\text{indirect}}$ .

## 7.4 Stage level performance

Stage level performance is summarized from the per-stage rows in the summary table returned by the post-processor. For each stage  $k$  the following quantities are available:

- Heat duty:  $Q_{\text{stage}}$  [MW]
- Overall conductance:  $UA_{\text{stage}}$  [MW/K]
- Gas inlet/outlet temperatures:  $gas\_in\_T$  [°C],  $gas\_out\_T$  [°C]
- Water inlet/outlet temperatures:  $water\_in\_T$  [°C],  $water\_out\_T$  [°C]
- Gas side pressure drops:  $\Delta P_{\text{stage\_fric}}$  [Pa],  $\Delta P_{\text{stage\_minor}}$  [Pa],  $\Delta P_{\text{stage\_total}}$  [Pa]
- Decomposition of duty into convection and radiation:  $Q_{\text{conv\_stage}}$  [MW],  $Q_{\text{rad\_stage}}$  [MW]

Kind	$T_{g,in}$ [°C]	$T_{g,out}$ [°C]	$T_{w,in}$ [°C]	$T_{w,out}$ [°C]	$Q_{\text{stage}}$ [MW]	$UA_{\text{stage}}$ [MW/K]	$\Delta P_{\text{stage}}$ [Pa]
single tube	[·]	[·]	[·]	[·]	[·]	[·]	[·]
reversal ch.	[·]	[·]	[·]	[·]	[·]	[·]	[·]
tube bank	[·]	[·]	[·]	[·]	[·]	[·]	[·]

Kind	$T_{g,in}$ [°C]	$T_{g,out}$ [°C]	$T_{w,in}$ [°C]	$T_{w,out}$ [°C]	$Q_{stage}$ [MW]	$UA_{stage}$ [MW/K]	$\Delta P_{stage}$ [Pa]
reversal	[·]	[·]	[·]	[·]	[·]	[·]	[·]
ch.	[·]	[·]	[·]	[·]	[·]	[·]	[·]
tube	[·]	[·]	[·]	[·]	[·]	[·]	[·]
bank	[·]	[·]	[·]	[·]	[·]	[·]	[·]
economiser	[·]	[·]	[·]	[·]	[·]	[·]	[·]

## 7.5 Overall boiler summary

The overall boiler performance is finally summarized using the boiler summary table:

Quantity	Symbol	Value
Fuel firing (LHV basis)	$P_{LHV}$	
Total heat input (combustion)	$Q_{in}$	
Useful heat to water/steam	$Q_{useful}$	
Direct efficiency (LHV basis)	$\eta_{direct}$	
Indirect efficiency	$\eta_{indirect}$	
Stack gas temperature	$T_{stack}$	
Gas side friction loss	$\Delta P_{fric}$	
Gas side minor losses	$\Delta P_{minor}$	
Total gas side pressure drop	$\Delta P_{tot}$	
Total convective heat transfer	$Q_{conv}$	
Total radiative heat transfer	$Q_{rad}$	

These boiler-level results provide the basis for the sensitivity analysis in Section 8 and for comparing alternative design or operating scenarios.

# **Chapter 8**

## **Sensitivity Analysis**

### **8.1 Excess Air Ratio**

### **8.2 Drum Pressure**

# Chapter 9

## Conclusion

placeholder

Incropera, Frank P., David P. DeWitt, Theodore L. Bergman, and Adrienne Lavine. 2011. *Fundamentals of Heat and Mass Transfer*. 7th ed. Wiley.

Modest, Michael F. 2013. *Radiative Heat Transfer*. 3rd ed. Academic Press.

Munson, Bruce R., Donald F. Young, and Theodore H. Okiishi. 2013. *Fundamentals of Fluid Mechanics*. 7th ed. Wiley.