Carnot to Rankine supercritical cycles efficiency calculation in Python.

Richard J Smith
richardsmith@asia.com

22 Mar 2021

Disclaimer

This document is provided by Richard J Smith 'as is' and 'with all faults'. The provider makes no representations or warranties of any kind concerning the safety, suitability, inaccuracies, typographical errors, or other harmful components of this document and software.

The sole purpose of this document is to illustrate and show calculations for the progression from the theoretical Carnot cycle proposed in 'Reflections on the Motive Power of Fire 1824' by Sadi Carnot to the advanced Rankine Supercritical cycle employed today in many power generation plants.

Each advancement in steam cycles will be shown individually and calculations given to find the cycle thermal efficiency.

Use the information and code contained at your own risk.

Contents

1 Steam cycle progression

The laws of thermodynamics dictate that all heat engines must have heat added to the working fluid in some part of the cycle and must reject heat in some other part. The heat rejected represents a loss and hence no heat engine cycle, even an ideal one, can have an efficiency of 100%. Practical cycles work between two temperature extremes, an upper temperature at which heat is added and a lower temperature at which heat is rejected.

1.1 Carnot cycle

Figure 1 shows the Ts (Temperature - entropy[1]) diagram for a hypothetical cycle operating between two absolute temperatures T_1 and T_2. A cycle which follows this rectangular shape on the Ts diagram has the greatest efficiency possible for any cycle operating between T_1 and T_2. Various cycles have been proposed which would, in theory, achieve this. One such cycle is known as the Carnot cycle[2]. It consists of an isothermal addition of heat (in the boiler) from point 2 - 3, isentropic expansion (in a turbine) from 3 - 4, isothermal rejection of heat from 4 to 1 (in the condenser) and isentropic compression from 1 - 2, to complete the cycle. The heat added is represented by the area under 2 - 3, i.e. the useful heat plus the rejected heat. The heat rejected is represented by the area under 4 - 1, i.e. the area beneath the condensation line. The heat available to do useful work is the area bounded by 1 - 2 - 3 - 4, i.e. between T_1 and T_2. Hence, the maximum efficiency of the cycle is heat output (useful work) divided by the heat input (heat added).

$$\text{Thermal efficiency} = \frac{\text{added heat} - \text{rejected heat}}{\text{added heat}}$$

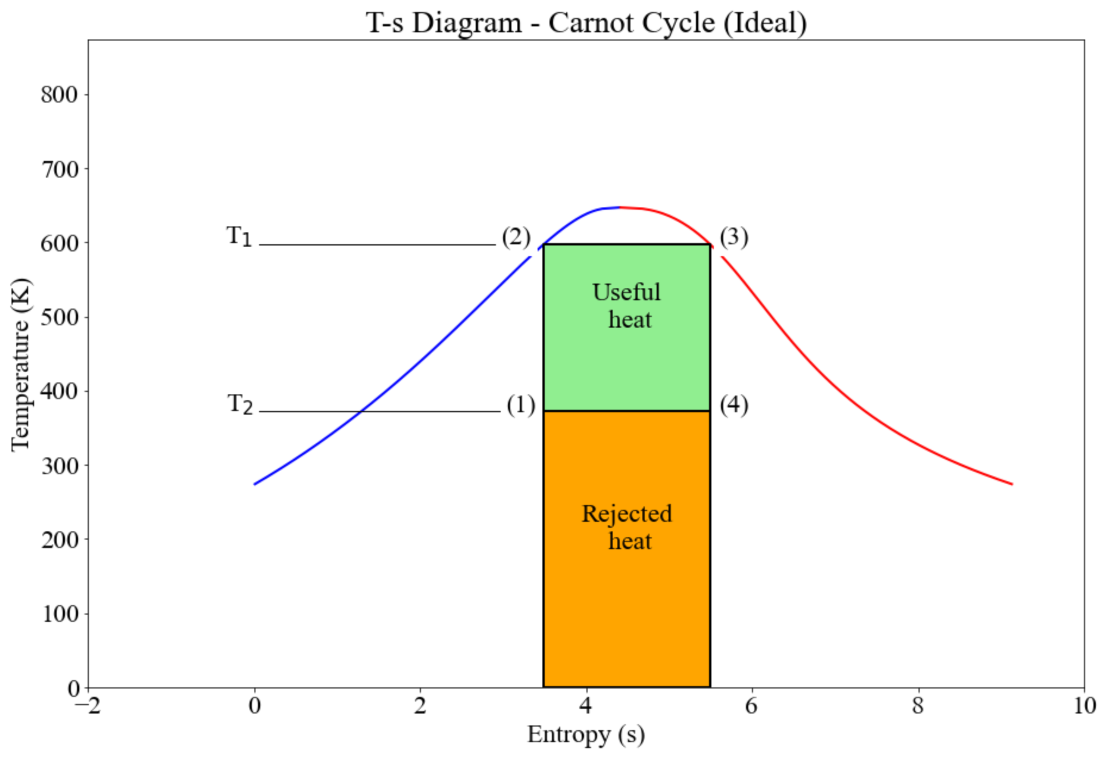

Figure 1: Ts plot of Carnot cycle showing useful and rejected heat

The Carnot efficiency for any two temperatures T_1 and T_2 represents the maximum possible efficiency that can be achieved within those two temperatures. The closer a practical cycle comes to filling the whole rectangle, 1

[1] A temperature–entropy diagram, or Ts diagram, is a thermodynamic diagram used in thermodynamics to visualize changes to temperature and specific entropy during a thermodynamic process or cycle as the graph of a curve. It is a useful and common tool, particularly because it helps to visualize the heat transfer during a process. For reversible (ideal) processes, the area under the Ts curve of a process is the heat transferred to the system during that process - according to wikipedia.org

[2] Nicolas Léonard Sadi Carnot (1 June 1796 – 24 August 1832) was a French mechanical engineer in the French Army, military scientist and physicist, and often described as the 'father of thermodynamics.' - according to wikipedia.org

- 2 - 3 - 4, the closer it will come to achieving the maximum efficiency for given values of T_1 and T_2.

In order to maximise efficiency it is necessary that T_1 be as large as possible and that T_2 be as small as possible. In a practical cycle T_2 is approximately the temperature of the available cooling medium, e.g. river water or sea water. T_1 is set by a metallurgical limit - the highest temperature to which the boiler components may be subjected.

Example 1

The Carnot cycle is not practical to construct, even for an engine operating with a perfect gas as the working medium. To construct such an engine using steam/water as the working medium would be virtually impossible. Figure 1 shows the Carnot cycle operating on steam/water. Point 2 consists of saturated water and 2 - 3 represents converting this to dry saturated steam. From 3 - 4 represents expansion to condenser conditions. At point 4 the fluid is wet steam. From 4 - 1 represents condensation of this steam towards the saturated water condition. However, at point 1 the condensation process must be halted and the wet steam compressed to saturated water at point 2.

This last step would be fraught with difficulty requiring a compressor of similar size to the turbine (It would be much simpler to condense the vapour completely and then compress the liquid in a relatively small feed pump. The resulting cycle is known as the Rankine cycle and is discussed below).

Although as stated the Carnot cycle is virtually impossible to construct, let us nevertheless considering a theoretical Carnot cycle based power plant that operates between a condenser pressure of 1 bar (atmospheric pressure) and a maximum boiler pressure of 120 bar.

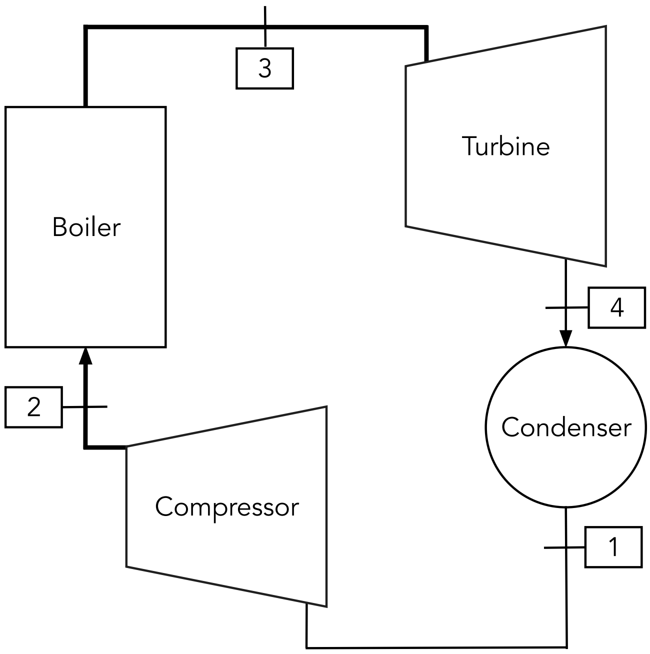

Figure 2: Components for a potential Carnot cycle power generation system

The saturation temperature of water at 120 bar is only 597.85 K (or 324.7°C) so in well within the metallurgical limits of ordinary steel construction (SA-210).

This theoretical cycle between a heat source of 324.7°C and a sink of 99.6°C will result in a cycle thermal efficiency of 37.6%. This is the maximum efficiency that can be achieved between these two temperatures, no matter the type of steam cycle used.

Plot output

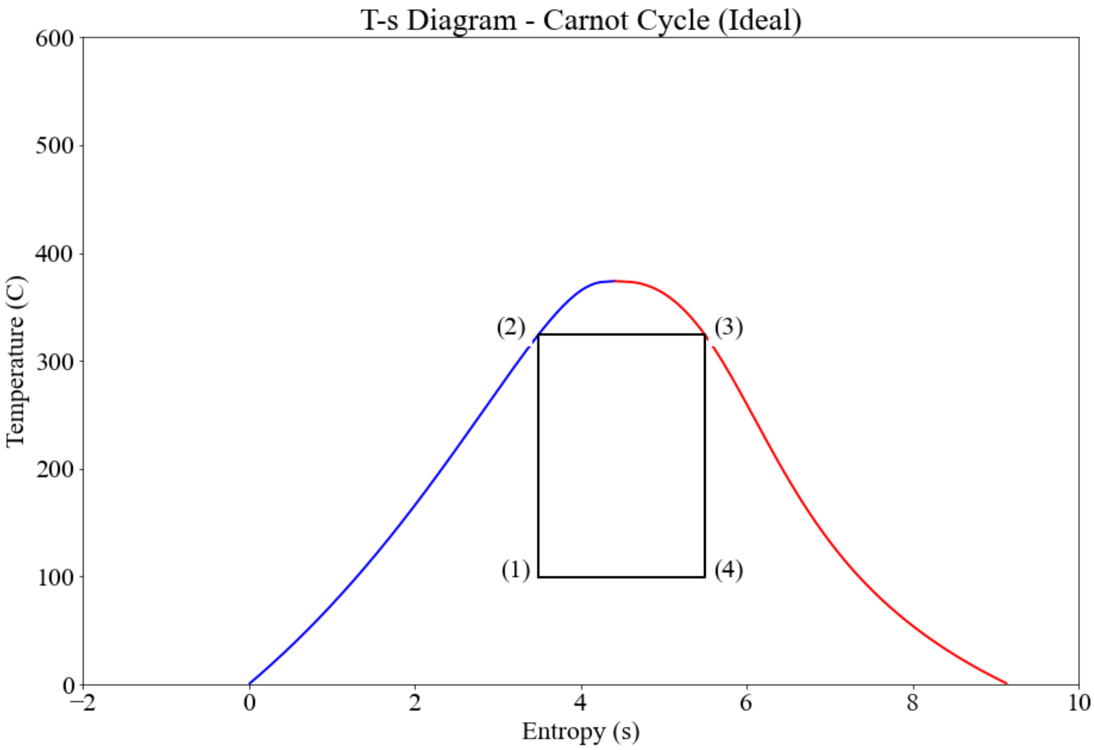

Figure 3: Ts plot of Carnot cycle - NOTE: vertical axis is in degrees C.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.1.

Code output

```
Carnot cycle analysis

Point 1
P1: 1.0 bar
T1: 99.6 degC
S1: 3.496 kJ/kg K
H1: 1235.2 kJ/kg

Point 2
P2: 120.0 bar
T2: 324.7 degC
H2: 1491.3 kJ/kg
S2: 3.496 kJ/kg K

Point 3
P3: 120.0 bar
T3: 324.7 degC
H3: 2685.6 kJ/kg
S3: 5.494 kJ/kg K

Point 4
P4: 1.0 bar
T4: 99.6 degC
H4: 1979.9 kJ/kg
S4: 5.494 kJ/kg K
x4: 69.2 % dry

Summary
Work required by pump: 256.1 kJ/kg
Work generated by turbine: 705.7 kJ/kg
Heat input by boiler: 1194.3 kJ/kg
```

```
Heat rejected by the condenser: 744.6 kJ/kg
Thermal efficiency is: 37.6%
HR rankine cycle: 9562.3 kJ/kWh
```

1.2 Rankine cycle

As mentioned in the above section, if would be much simpler to condense the vapour completely and then compress the liquid using a relatively small feed pump.

Figure 4 shows the Ts diagram for a Rankine cycle[3] operating between the same two absolute temperatures T_1 and T_2. Whilst the area of useful heat has increased slightly, the area of rejected heat has approximately doubled, thus meaning the the cycle thermal efficiency will be lower than the 37.6% calculated for the Carnot cycle.

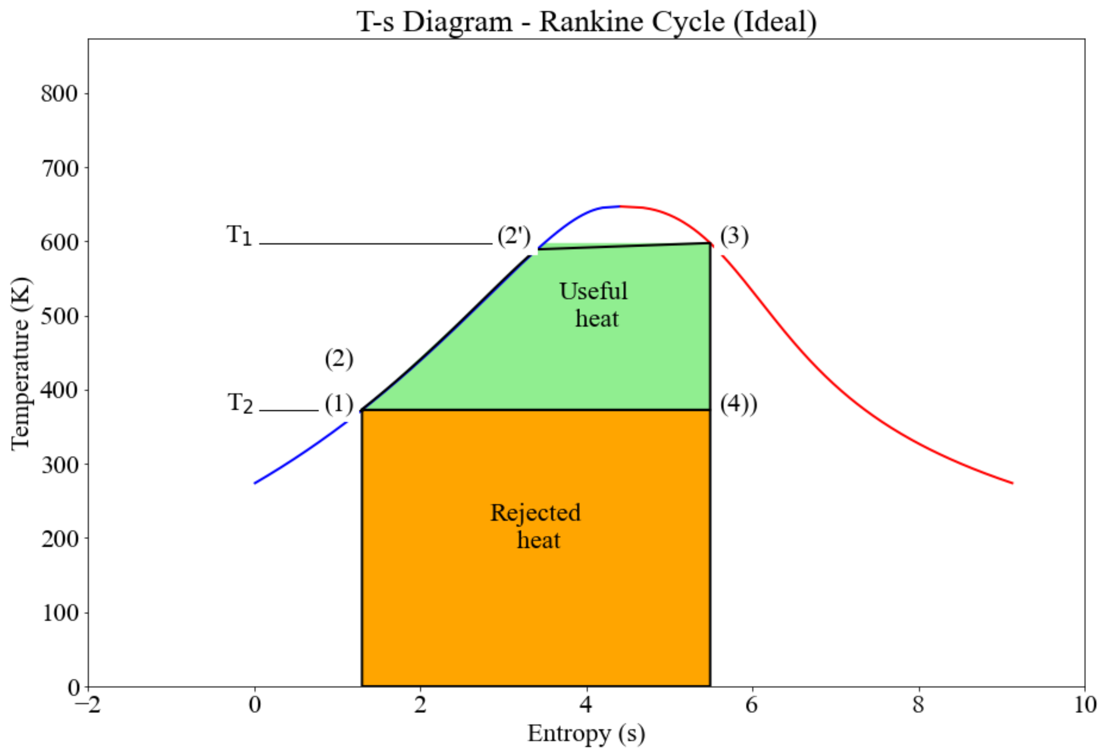

Figure 4: Ts plot of Rankine cycle showing useful and rejected heat

It commences with work done by the feed pump 1 - 2, then the water is heated to its saturation temperature 2 - 2'. Followed by conversion to steam 2' - 3. Expansion in the turbine is from 3 - 4, and condensation to saturated water from 4 - 1.

Example 2

Let us now considering a theoretical Rankine cycle based power plant that operates between a condenser pressure of 1 bar (atmospheric pressure) and a maximum boiler pressure of 120 bar.

[3]William John Macquorn Rankine FRSE FRS (5 July 1820 – 24 December 1872) was a Scottish mechanical engineer who also contributed to civil engineering, physics and mathematics. He was a founding contributor, with Rudolf Clausius and William Thomson (Lord Kelvin), to the science of thermodynamics. - according to wikipedia.org

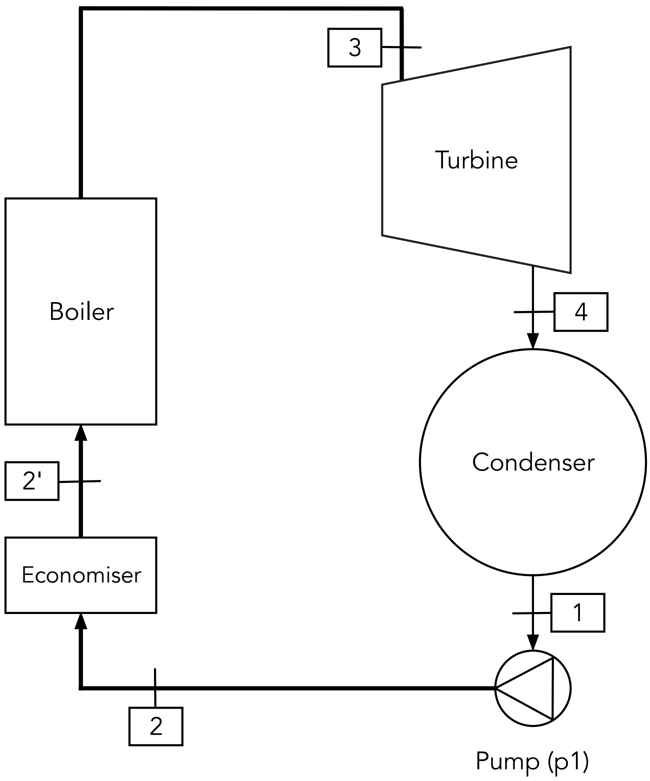

Figure 5: Components for a Rankine cycle power generation system

Plot output

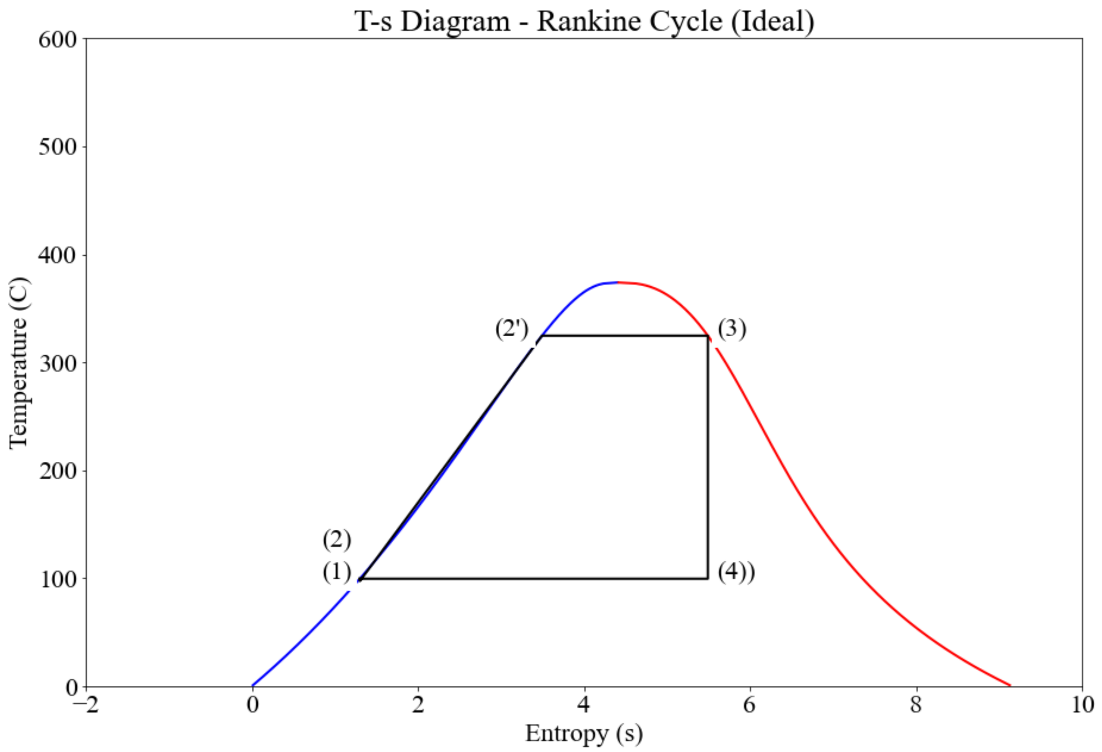

Figure 6: Ts plot of Rankine cycle - NOTE: vertical axis is in degrees C.

Again the saturation temperature of water at 120 bar is 597.85 K (or 324.7°C).

This cycle between a heat source of 324.7°C and a sink of 99.6°C will result in a cycle thermal efficiency of only 31.1%.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.2.

Code output

```
Rankine cycle analysis

Point 1
p1: 1.0 bar
T1: 99.6 degC
H1: 417.4 kJ/kg
S1: 1.303 kJ/kg K

Point 2
H2: 417.6 kJ/kg
T2: 97.5 degC

Point 2dash
H2dash: 1491.3 kJ/kg
S2dash: 3.496 kJ/kg K
T2dash: 324.7 degC

Point 3
P3: 120.0 bar
T3: 324.7 degC
H3: 2685.6 kJ/kg
S3: 5.494 kJ/kg K

Point 4
P4: 1.0 bar
T4: 99.6 degC
H4: 1979.9 kJ/kg
S4: 5.494 kJ/kg K
x4: 69.2 % dry

Summary
Work required by pump: 0.1 kJ/kg
Work generated by turbine: 705.7 kJ/kg
Heat input by boiler: 2268.0 kJ/kg
Heat rejected by the condenser: 1562.4 kJ/kg
Thermal efficiency is: 31.1%
HR rankine cycle: 11571.8 kJ/kWh
```

1.3 Rankine cycle (reduced condenser pressure)

The 31.1% thermal efficiency achieved by the Rankine cycle example above is considerable less than the 37.6% efficiency of the Carnot cycle operating between the same temperature extremes and whilst the change has resulted in a cycle that would be possible to build, it would be nice to reclaim some of that lost efficiency.

The first method of increasing Rankine cycle is to lower the temperature at which heat is rejected. If instead of expanding the steam in the turbine down to a pressure of 1 bar absolute (atmospheric pressure) we continue the expansion and reach a pressure of 0.1 bar absolute by forming a vacuum in the condenser, we see that the saturation temperature is now only $45.8°C$.

Example 3

Let us now considering a modified Rankine cycle still with a maximum boiler pressure of 120 bar, but now a condenser pressure of 0.1 bar.

This cycle between a heat source of $324.7°C$ and a sink of $45.8°C$ will result in a cycle thermal efficiency of 38.0%[4]. A rather large increase over the 31.1% achieved by the original Rankine cycle example.

Plot output

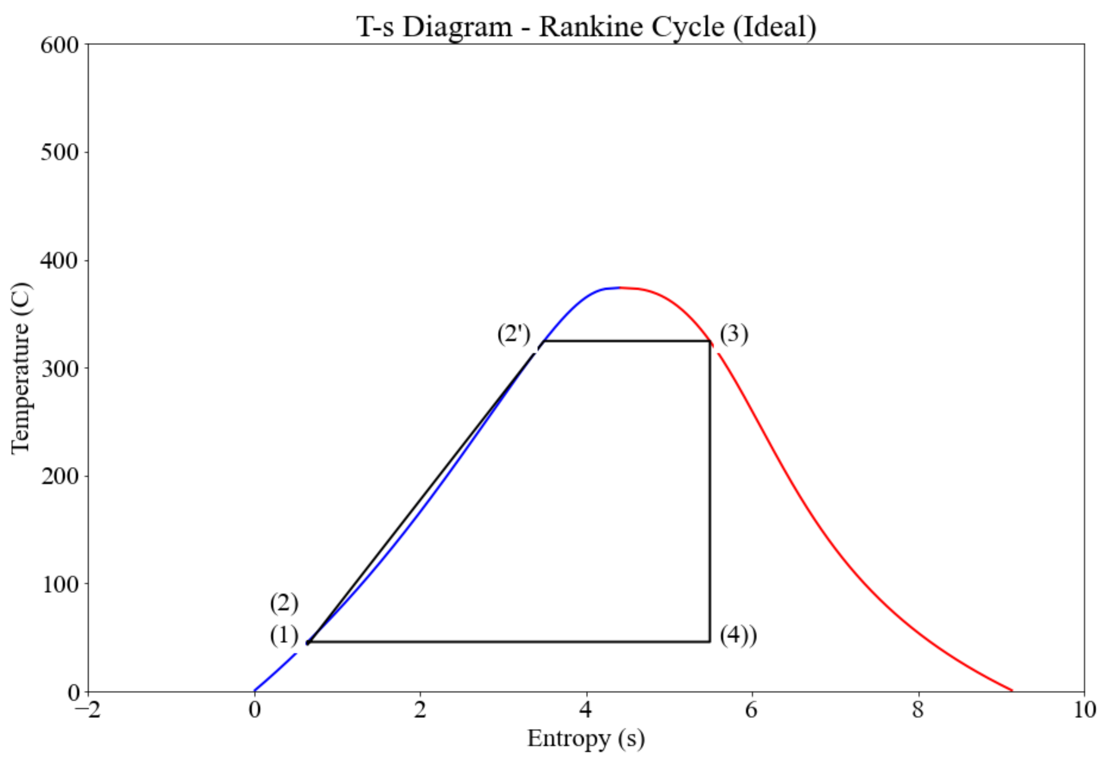

Figure 7: Ts plot of Rankine cycle with condenser under vacuum.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.3.

Code output

```
Rankine cycle analysis

Point 1
p1: 0.1 bar
```

[4]This value is higher than the 37.6% for Carnot cycle shown above, but remember heat is rejected at a lower temperature $45.8°C$ vs $99.6°C$. If we were to calculate the Carnot cycle thermal efficiency over the same temperatures - heat added at $324.7°C$ and rejected at $45.8°C$ we would get 46.6%

```
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 191.9 kJ/kg
T2: 43.3 degC

Point 2dash
H2dash: 1491.3 kJ/kg
S2dash: 3.496 kJ/kg K
T2dash: 324.7 degC

Point 3
P3: 120.0 bar
T3: 324.7 degC
H3: 2685.6 kJ/kg
S3: 5.494 kJ/kg K

Point 4
P4: 0.1 bar
T4: 45.8 degC
H4: 1737.1 kJ/kg
S4: 5.494 kJ/kg K
x4: 64.6 % dry

Summary
Work required by pump: 0.1 kJ/kg
Work generated by turbine: 948.5 kJ/kg
Heat input by boiler: 2493.6 kJ/kg
Heat rejected by the condenser: 1545.3 kJ/kg
Thermal efficiency is: 38.0%
HR rankine cycle: 9466.2 kJ/kWh
```

1.4 Rankine cycle (increased boiler pressure)

The second method of increasing Rankine cycle efficiency is to increase the boiler pressure and hence the temperature at which heat is added. By increasing from 120 bar up to 180 bar the saturation temperature has increased from 324.7°C to 357.0°C.

Example 4

Let us now considering a further modified Rankine cycle still with a maximum boiler pressure of 180 bar and a condenser pressure of 0.1 bar.

This cycle between a heat source of 357.0°C and a sink of 45.8°C will result in a cycle thermal efficiency of 38.7%. A rather small increase over the 38.0% achieved in the previous example, but still worthwhile.

Plot output

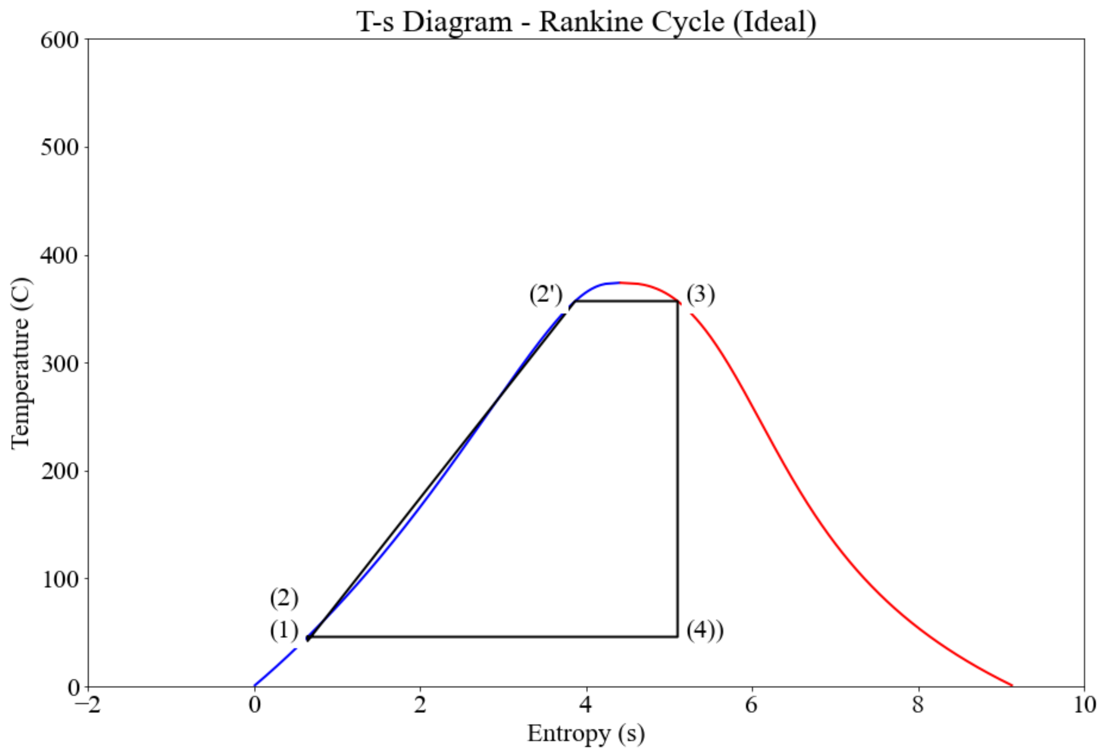

Figure 8: Ts plot of Rankine cycle with increased boiler pressure.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.4.

Code output

```
Rankine cycle analysis

Point 1
p1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 192.0 kJ/kg
T2: 42.1 degC

Point 2dash
H2dash: 1732.0 kJ/kg
```

```
S2dash: 3.872 kJ/kg K
T2dash: 357.0 degC

Point 3
P3: 180.0 bar
T3: 357.0 degC
H3: 2509.5 kJ/kg
S3: 5.106 kJ/kg K

Point 4
P4: 0.1 bar
T4: 45.8 degC
H4: 1613.2 kJ/kg
S4: 5.106 kJ/kg K
x4: 59.4 % dry

Summary
Work required by pump: 0.2 kJ/kg
Work generated by turbine: 896.3 kJ/kg
Heat input by boiler: 2317.5 kJ/kg
Heat rejected by the condenser: 1421.4 kJ/kg
Thermal efficiency is: 38.7%
HR rankine cycle: 9309.9 kJ/kWh
```

1.5 Rankine Superheat cycle

It was stated that the upper temperature of the cycle must be as high as possible to maximise efficiency. The limit to this is the metallurgical limit of the materials of the boiler. Clearly, this limit is not reached using saturated steam since, even at the critical point, the temperature is only $374°C$. In Figure 9 the basic Rankine cycle has been modified by adding a superheater to the boiler.

It commences with work done by the feed pump 1 - 2, the the water is heated to its saturation temperature 2 - 2'. Followed by conversion to steam 2' - 3'. At this point however, the saturated steam passes through superheater tubes where it is heated to point 3. Expansion in the turbine takes place from 3 - 4, and condensation to saturated water from 4 - 1.

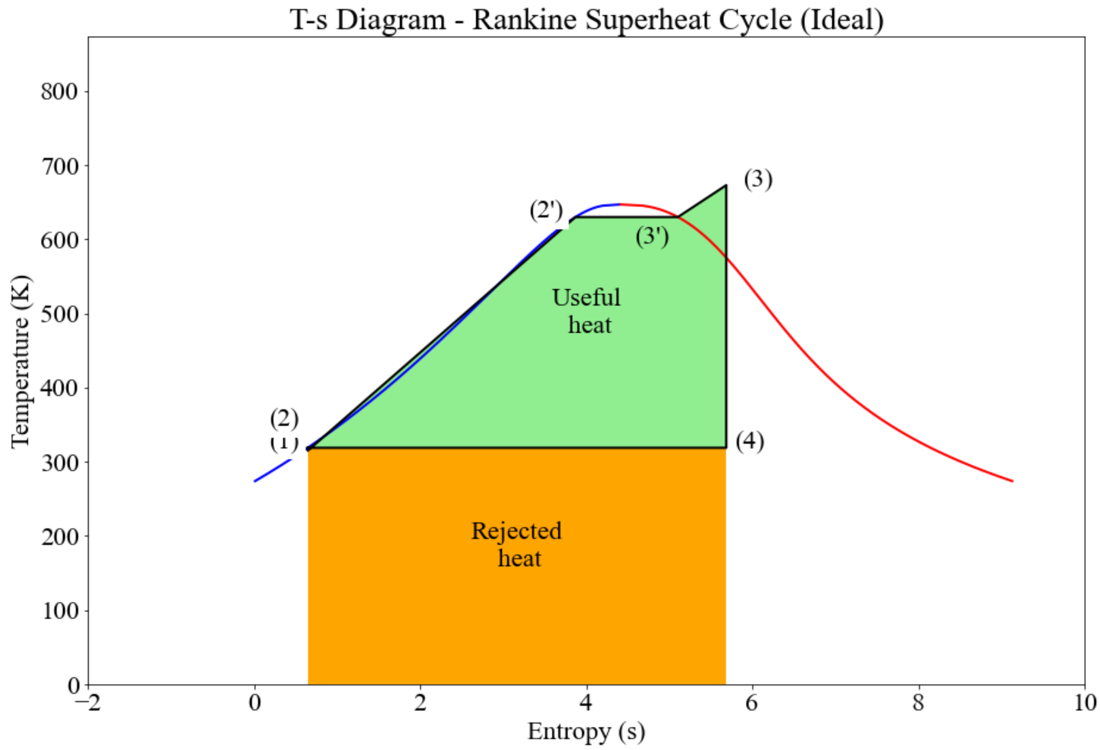

Figure 9: Ts plot of Rankine Superheat cycle showing useful and rejected heat

It can be seen from Figure 9 that the area of useful heat has been increased and, although the heat rejected has also increased, there is an overall gain in efficiency through superheating the steam. Another important feature of Figure 9 is the distance of point 4 from the saturated steam line i.e. 4 has moved further towards the right hand side, thus increasing the dryness fraction of the steam leaving the LP turbine (This is important to limit water droplet erosion of the last stage blades).

Example 5

Modelling this further modified Rankine cycle still with a maximum boiler pressure of 180 bar and a condenser pressure of 0.1 bar, but now with a boiler superheater section added that increases the steam temperature at boiler output to $400°C$.

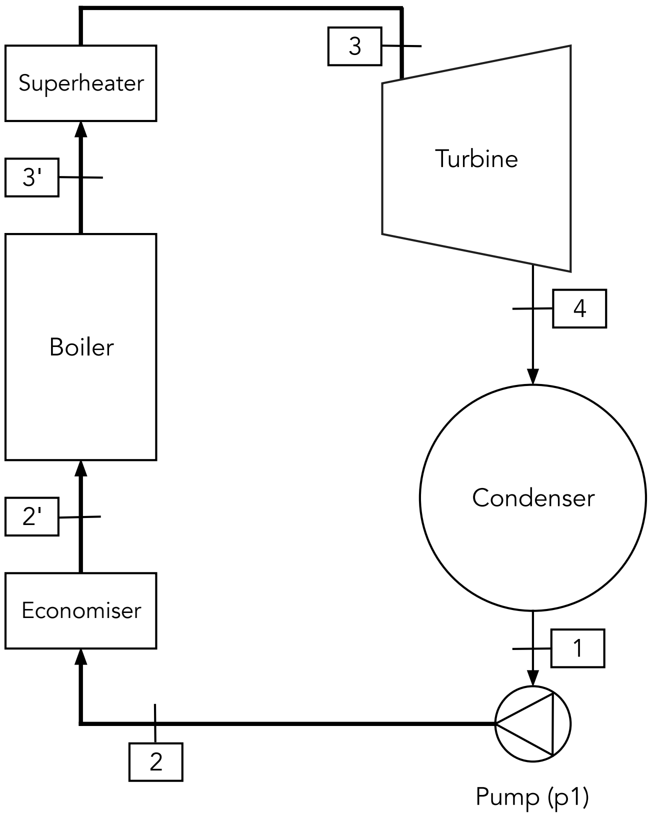

Figure 10: Components for a Rankine Superheat cycle power generation system

Plot output

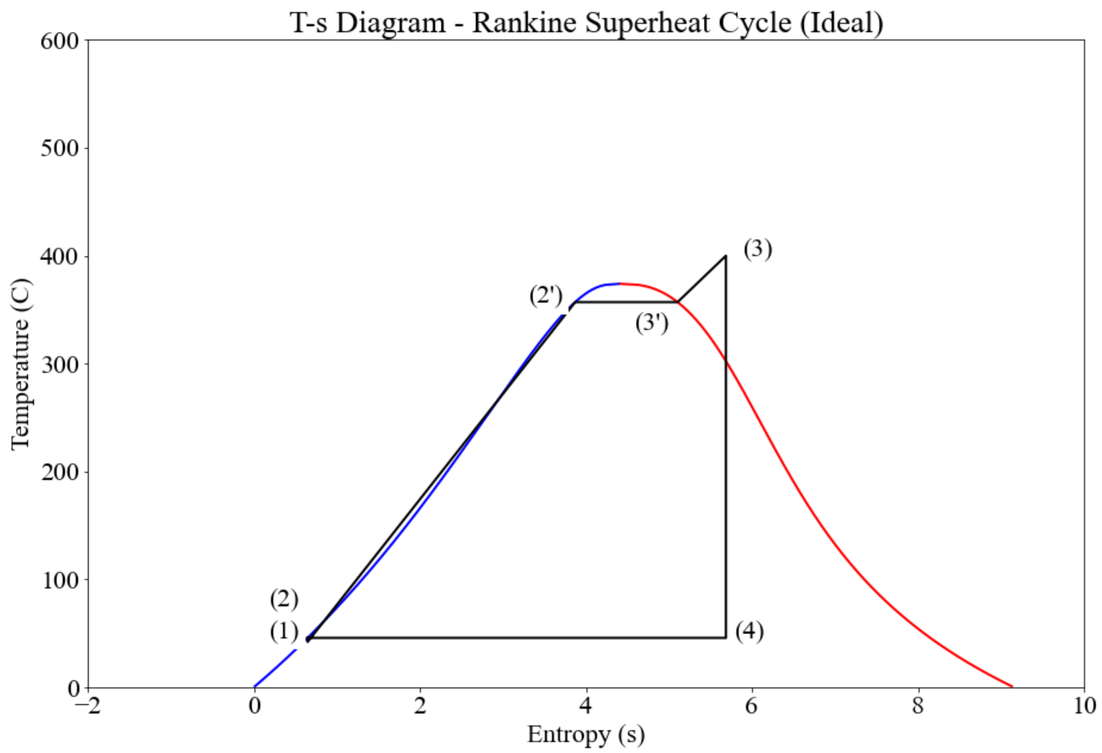

Figure 11: Ts plot of Rankine Superheat cycle.

The efficiency of this cycle compared to the previous example is 40.3% as against 38.7%. Again a small increase but also still worthwhile. Of almost equal importance, the useful work obtained from a given quantity of steam is increased i.e. there is more heat energy contained in each kg of steam when it is superheated.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.5.

Code output

```
Rankine superheat cycle analysis

Point 1
p1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 192.0 kJ/kg
T2: 42.1 degC

Point 2 dash
T2dash: 357.0 degC
p2dash: 180.0 bar
H2dash: 1732.0 kJ/kg
S2dash: 3.872 kJ/kg K

Point 3dash
T3dash: 357.0 degC
H3dash: 2509.5 kJ/kg
S3dash: 5.106 kJ/kg K

Point 3
T3: 400.0 degC
p3: 180.0 bar
H3: 2886.3 kJ/kg
S3: 5.688 kJ/kg K

Point 4
T4: 45.8 degC
p4: 0.1 bar
H4: 1799.0 kJ/kg
S4: 5.688 kJ/kg K
x4: 67.2 % dry

Summary
Work required by pump: 0.2 kJ/kg
Work generated by turbine: 1087.3 kJ/kg
Heat input by boiler: 2694.3 kJ/kg
Heat rejected by the condenser: 1607.2 kJ/kg
Thermal efficiency is: 40.3%
HR rankine cycle: 8922.1 kJ/kWh
```

1.6 Rankine Superheat cycle (Increased superheat temperature)

The next method of increasing Rankine cycle efficiency is to increase the boiler superheat temperature to the metallurgical limit of the material used for construction (SA-213). Given considerations of material cost, availability, etc, this limit since the 1970's has been approximately 540°C. (NOTE: this has become the norm for sub-critical boiler around the world, although the metallurgical limit for modern materials is much higher).

Example 6

Modelling this modified Rankine cycle, still with a maximum boiler pressure of 180 bar and a condenser pressure of 0.1 bar, but now with a superheater outlet steam temperature of 540°C.

Plot output

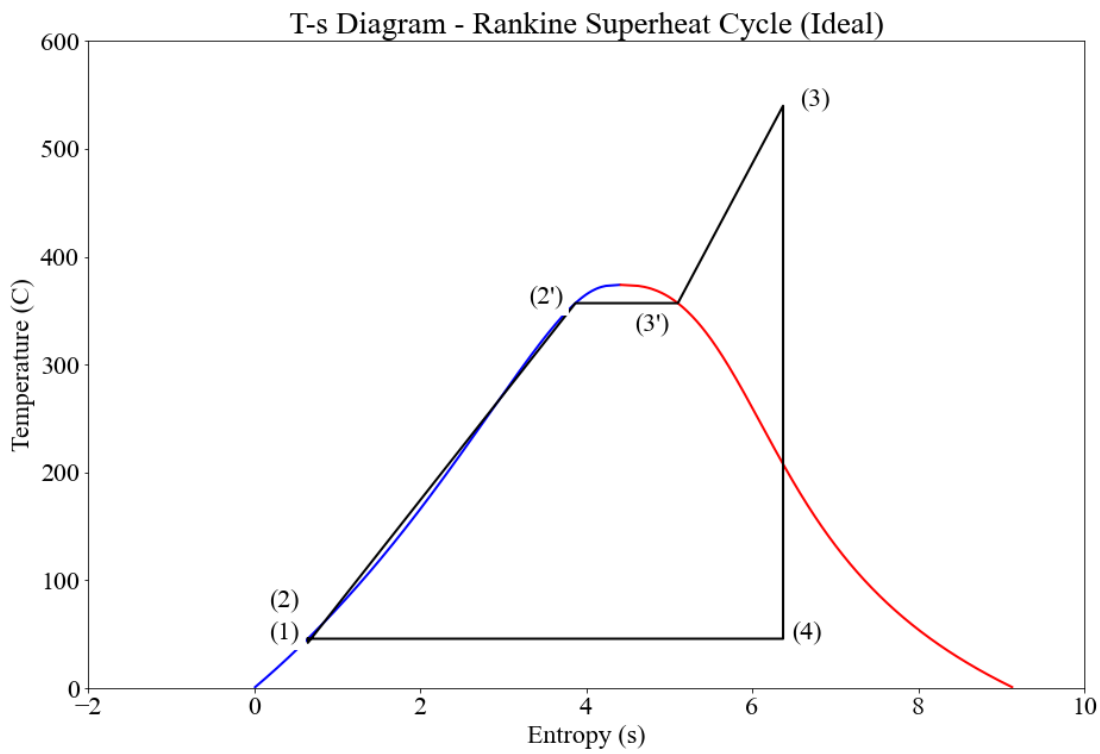

Figure 12: Ts plot of Rankine Superheat cycle.

The efficiency of this cycle compared to the previous example is 42.9% as against 40.3%.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.6

Code output

```
Rankine superheat cycle analysis

Point 1
p1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 192.0 kJ/kg
T2: 42.1 degC

Point 2 dash
T2dash: 357.0 degC
```

```
p2dash: 180.0 bar
H2dash: 1732.0 kJ/kg
S2dash: 3.872 kJ/kg K

Point 3dash
T3dash: 357.0 degC
H3dash: 2509.5 kJ/kg
S3dash: 5.106 kJ/kg K

Point 3
T3: 540.0 degC
p3: 180.0 bar
H3: 3389.5 kJ/kg
S3: 6.373 kJ/kg K

Point 4
T4: 45.8 degC
p4: 0.1 bar
H4: 2017.6 kJ/kg
S4: 6.373 kJ/kg K
x4: 76.3 % dry

Summary
Work required by pump: 0.2 kJ/kg
Work generated by turbine: 1372.0 kJ/kg
Heat input by boiler: 3197.5 kJ/kg
Heat rejected by the condenser: 1825.8 kJ/kg
Thermal efficiency is: 42.9%
HR rankine cycle: 8391.4 kJ/kWh
```

1.7 Rankine Reheat cycle

The Rankine cycle without superheat, as shown in section 1.2, had the entire turbine operating on wet steam. This presents a number of problems for the turbine, the main ones being the difficulties of designing blading to cope with steam containing water droplets and the erosion of the blading due to impact with these droplets. The superheat cycle as shown in section 1.6 partially overcomes this but the latter stages of expansion are still in the wet steam region of the Ts diagram.

A further refinement involves halting the expansion process before it enters the wet region and then again reheating the steam to the metallurgical. Wet steam in the turbine could be reduced and the work output per unit steam flow increased even further.

Plot output

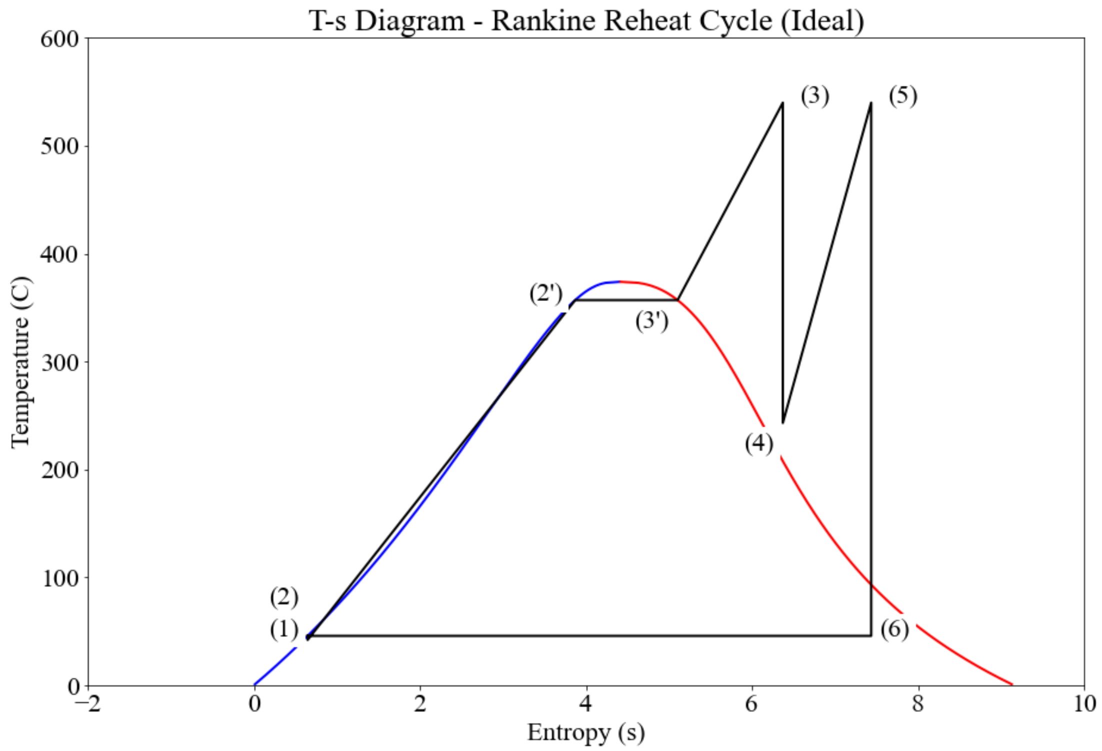

Figure 13: Ts plot of Rankine Reheat cycle.

Figure 13 shows the addition of a reheater to the previous superheat cycle. Boiling and superheating occurs as before and expansion in the turbine takes place from 3. However, at point 4 the steam passes out of the turbine and is returned to the boiler where further heat is added in a reheater, raising the temperature back to $540°C$. This steam passes back to the turbine where it is further expanded from 5 - 6. Condensing, feed pump work and heating to saturation then occur, as before.

Example 7

With a reheat section added to the boiler and other condition similar to the previous example (boiler pressure 180 bar, condenser pressure 0.1 bar, boiler superheat temperature $540°C$) the cycle thermal efficiency has increased to 44.3% from 42.9%.

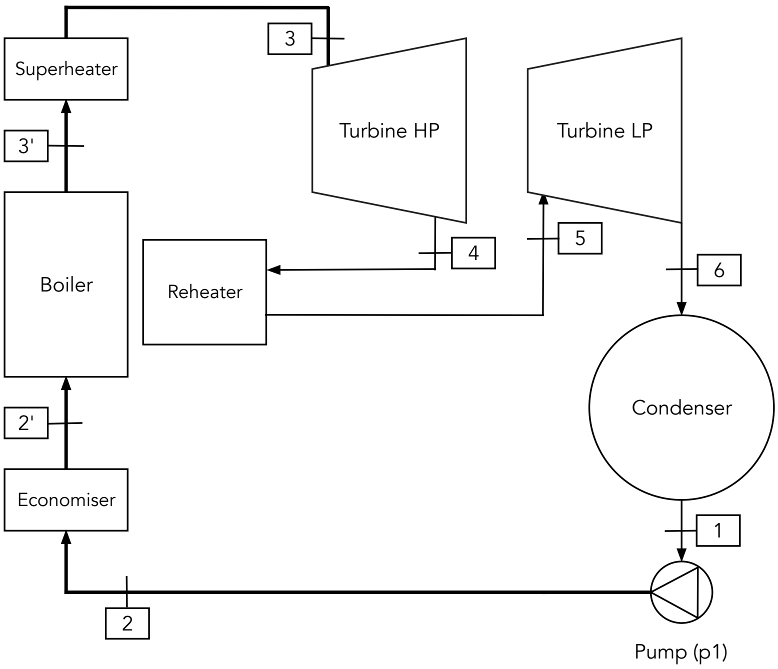

Figure 14: Components for a Rankine Reheat cycle power generation system

The increase in the complexity (and therefore the cost) of the plant in adding reheat is large. Both the boiler and the turbine become more complex and there is additional high temperature steam pipework and extra control equipment required. This increase in complexity cannot be justified by such a small improvement in efficiency alone. However, the work available per unit steam flow is increased. The size of the plant is therefore reduced. This reduction in the cost of the plant almost offsets the increase due to the complexity. However, the main improvement which justifies the use of reheat is the reduction of wetness in the last rows of LP turbine blades.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.7

Code output

```
Rankine reheat cycle analysis

Point 1
P1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 192.0 kJ/kg
T2: 42.1 degC

Point 2dash
T2dash: 357.0 degC
P2dash: 180.0 bar
H2dash: 1732.0 kJ/kg
S2dash: 3.872 kJ/kg K

Point 3dash
T3dash: 357.0 degC
H3dash: 2509.5 kJ/kg
S3dash: 5.106 kJ/kg K

Point 3
T3: 540.0 degC
P3: 180.0 bar
H3: 3389.5 kJ/kg
S3: 6.373 kJ/kg K
Reheat Pressure: 25.0 bar

Point 4
```

```
T4: 243.2 degC
P4: 25.0 bar
H4: 2861.5 kJ/kg
S4: 6.373 kJ/kg K

Point 5
T5: 540.0 degC
p5: 25.0 bar
H5: 3551.9 kJ/kg
S5: 7.438 kJ/kg K

Point 6
T6: 45.8 degC
p6: 0.1 bar
H6: 2357.0 kJ/kg
S6: 7.438 kJ/kg K
x6: 90.5 % dry

Summary
Work required by pump: 0.2 kJ/kg
Work generated by HP turbine: 528.0 kJ/kg
Work generated by LP turbine: 1194.8 kJ/kg
Total work output by turbine: 1722.8 kJ/kg
Heat input by boiler: 3887.9 kJ/kg
Heat rejected by the condenser: 2165.2 kJ/kg
Thermal efficiency is: 44.3%
HR rankine cycle: 8124.9 kJ/kWh
```

1.8 Rankine Regenerative cycle

An important method of further improving the cycle efficiency is by the installation of regenerative feedwater heating. Steam, having given up some of its energy in the turbine, is bled from various points of the turbine and used to heat the feedwater before it enters the boiler. If the steam was not bled from the turbine, its enthalpy of condensation would be lost to the circulating water in the condenser. With feedwater heating it is retained within the cycle.

Example 8

With one open feedwater heater installed in the cycle the thermal efficiency has increased to 46.8% from the previous 44.3%

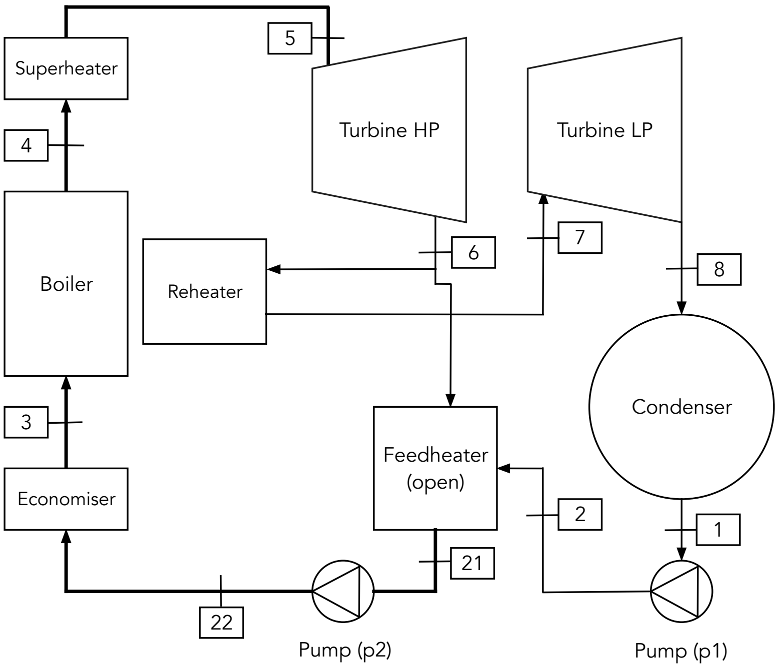

Figure 15: Components for a Rankine Reheat cycle with 1 open feedwater heater.

Plot output

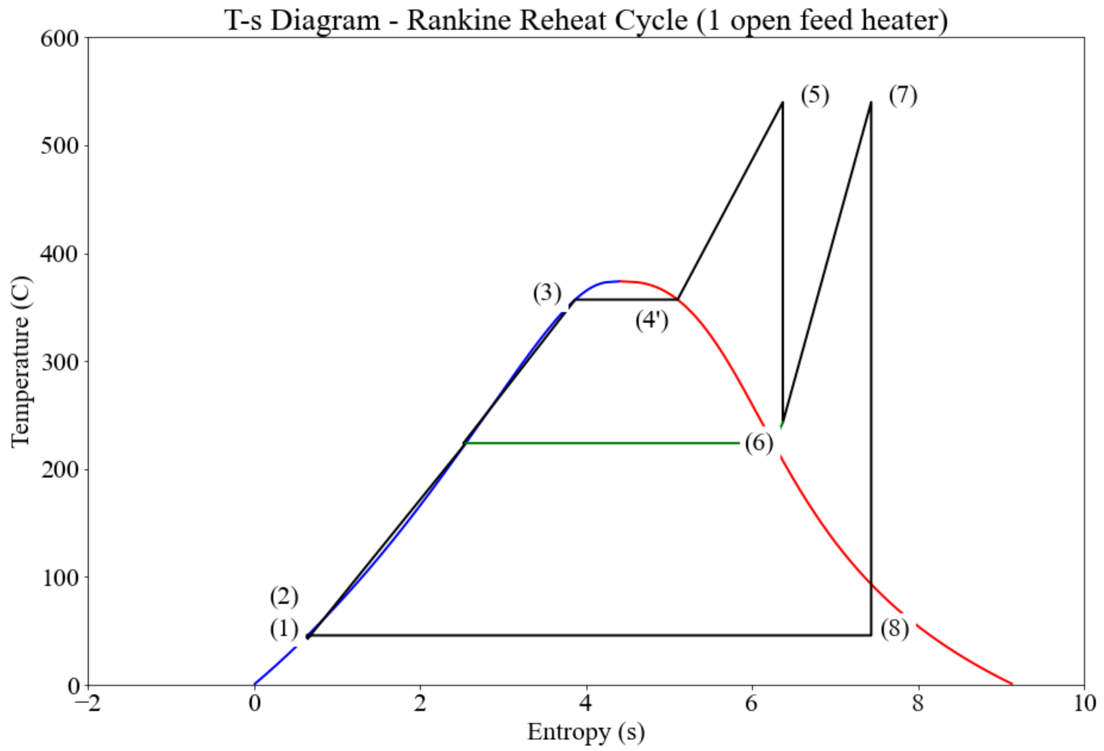

Figure 16: Ts plot of Rankine Reheat cycle with 1 open feedwater heater.

This example uses one large open feedwater heater only, but in practice this would be very large, costly and inefficient. A better approach is to use more, but smaller heaters with steam extractions from various points of the turbine and reheat steam lines.

A modern sub-critical Rankine reheat cycle plant would normally have 3 LP (low pressure closed type), 1 deaerator (open type feedwater heater) and 3 HP (high pressure closed type) heaters.

Additional feedwater heaters can be added (at additional cost), although the percentage performance increase will get less and less for each one added until economically it does not make sense to add more.

This point represents the limit of plant performance in a Rankine cycle at sub-critical pressures (less than 220.6 bar) and to further increase cycle thermal efficiency we must increase the boiler pressure above this limit at greatly increased complexity and material costs.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.8.

Code output

```
Rankine reheat cycle analysis with 1 open feedheater

Point 1
P1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K
Work required by pump 1: 2.52 kJ/kg

Point 2
P2: 25.0 bar
H2: 194.3 kJ/kg
T2: 42.6 degC
```

```

Point 21 - feedheater outlet before pump 2
P21: 25.0 bar
T21: 223.9 degC
H21: 962.0 kJ/kg
S21: 2.554 kJ/kg K
Work required by pump 2: 18.56 kJ/kg

Point 22 - pump 2 outlet
P22: 180.0 bar
T22: 223.9 degC
H22: 980.5 kJ/kg
S22: 2.526 kJ/kg K

Point 3
P3: 180.0 bar
T3: 357.0 degC
H3: 980.5 kJ/kg
S3: 3.872 kJ/kg K

Point 4
P4: 180.0 bar
T4: 357.0 degC
H4: 2509.5 kJ/kg
S4: 5.106 kJ/kg K

Point 5 - main steam conditions
P5: 180.0 bar
T5: 540.0 degC
H5: 3389.5 kJ/kg
S5: 6.373 kJ/kg K
Reheat Pressure: 25.0 bar

Point 6
P6: 25.0 bar
T6: 243.2 degC
H6: 2861.5 kJ/kg
S6: 6.373 kJ/kg K

Point 7 - IP/LP steam conditions
P7: 25.0 bar
T7: 540.0 degC
H7: 3551.9 kJ/kg
S7: 7.438 kJ/kg K

Point 8 - turbine exhaust conditions
P8: 0.1 bar
T8: 45.8 degC
H8: 2357.0 kJ/kg
S8: 7.438 kJ/kg K
x8: 90.5 % dry

feedwater heater mass flow ratio: 0.7122

Summary
Heat input by boiler: 3099.3 kJ/kg
Heat rejected to condenser: -2165.2 kJ/kg
Work generated by HP turbine: 528.0 kJ/kg
Work generated by LP turbine: 1194.8 kJ/kg
Total work output by turbine: 1722.8 kJ/kg
Thermal efficiency is: 1358.6 kJ/kg
Thermal efficiency is: 46.8 %
HR rankine cycle: 7686.1 kJ/kWh
Required steam flow: 73.6 kg/s
Steam flow to condenser: 52.4 kg/s
```

1.9 Rankine Supercritical cycle

To further increase the cycle thermal efficiency it is necessary to increase the boiler pressure above the critical pressure (220.6 bar) and at the same time it is possible with modern advances in metallurgy to increase the main steam and reheat steam temperatures.

Example 9

With the boiler pressure increased to 350 bar main steam and 85 bar reheat pressure plus steam temperature in both main and reheat increased to 600°C the cycle thermal efficiency has increased to 47.4%, a handy increase over the 44.3% attained by the sub-critical reheat cycle in Section 1.7.

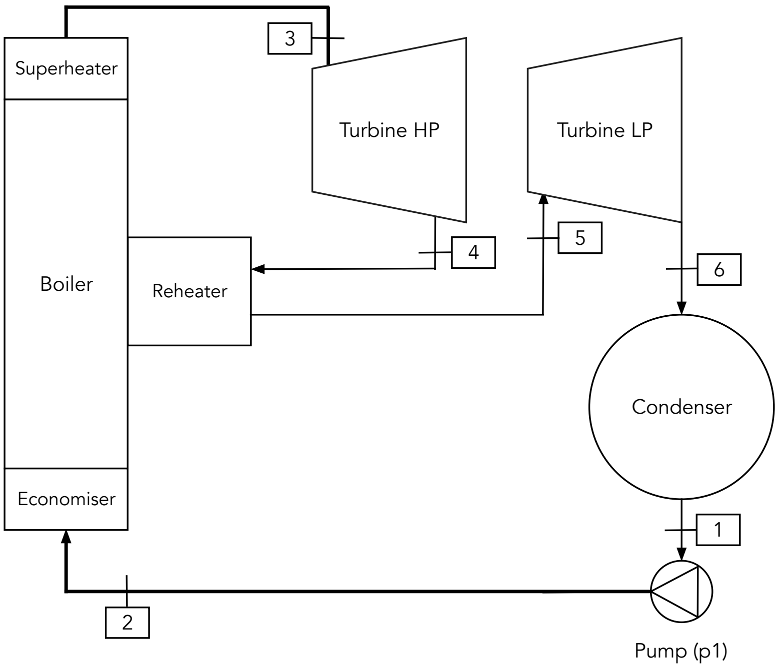

Figure 17: Components for a Rankine Supercritical cycle.

Plot output

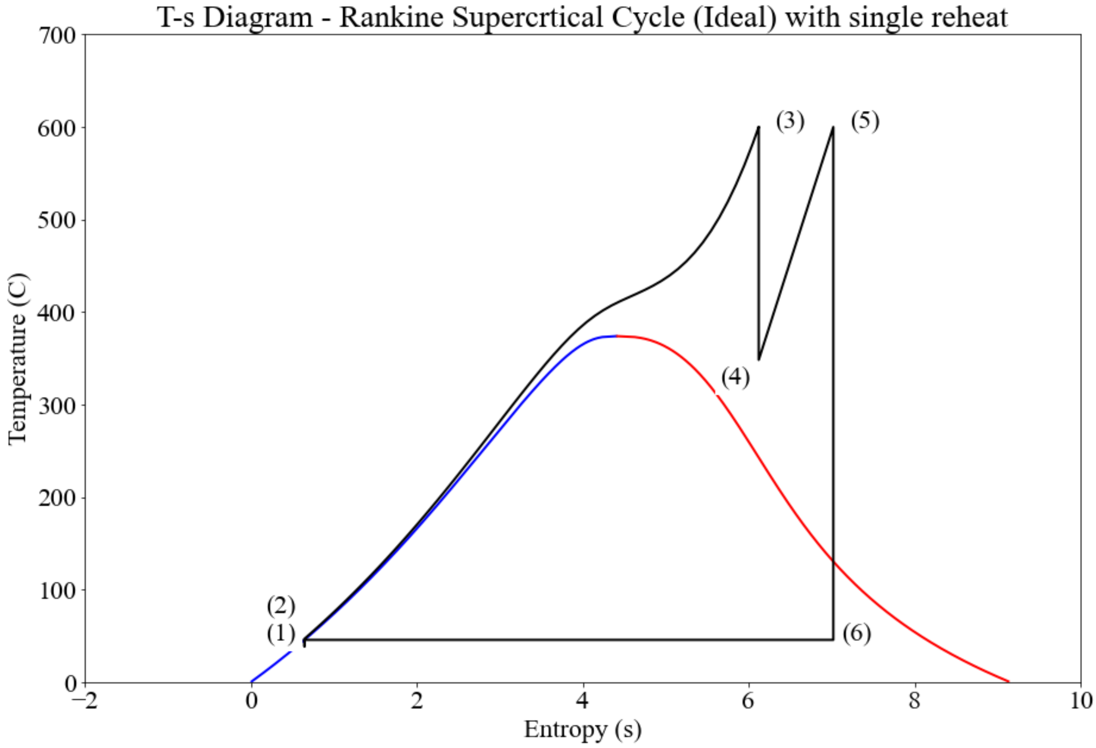

Figure 18: Ts plot of Rankine Supercritical cycle with single reheat.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.9

Code output

```
Rankine supercritical cycle (single reheat) analysis

Point 1
P1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 192.2 kJ/kg
T2: 38.5 degC

Point 3
T3: 600.0 degC
P3: 350.0 bar
H3: 3399.0 kJ/kg
S3: 6.123 kJ/kg K
Reheat Pressure: 80.0 bar

Point 4
T4: 348.3 degC
P4: 80.0 bar
H4: 2982.4 kJ/kg
S4: 6.123 kJ/kg K

Point 5
T5: 600.0 degC
p5: 80.0 bar
H5: 3642.4 kJ/kg
S5: 7.022 kJ/kg K

```

```
Point 6
T6: 45.8 degC
p6: 0.1 bar
H6: 2224.5 kJ/kg
S6: 7.022 kJ/kg K
x6: 85.0 % dry

Summary
Work required by pump: 0.4 kJ/kg
Work generated by HP turbine: 416.6 kJ/kg
Work generated by LP turbine: 1417.9 kJ/kg
Total work output by turbine: 1834.5 kJ/kg
Heat input by boiler: 3866.9 kJ/kg
Heat rejected by the condenser: 2032.7 kJ/kg
Thermal efficiency is: 47.4%
HR rankine cycle: 7589.6 kJ/kWh
```

1.10 Rankine Supercritical cycle with double reheat

One issue with the increase to supercritical boiler pressures is that with a single reheat system the turbine outlet dryness has reduced down to 85.0% in section 1.9. To minimise droplet erosion on the last stage of LP turbine blades it is recommended to maintain dryness above 88.0%

Example 10

Using the same main steam and 1st reheat condition as the previous example, we add a 2nd reheat stage at 25 bar and 600°C outlet temperature. This has only a small effect on cycle thermal efficiency, increasing to 48.6% from 47.4%, however LP turbine exhaust dryness has increased from a rather low 85.0% up to a much better 92.7%.

To further increase cycle thermal efficiency we can further raise boiler pressure and steam temperatures until the current metallurgical limits, however in the interests of reliability and keeping construction costs under control it is advantageous to not always be on the bleeding edge of technology.

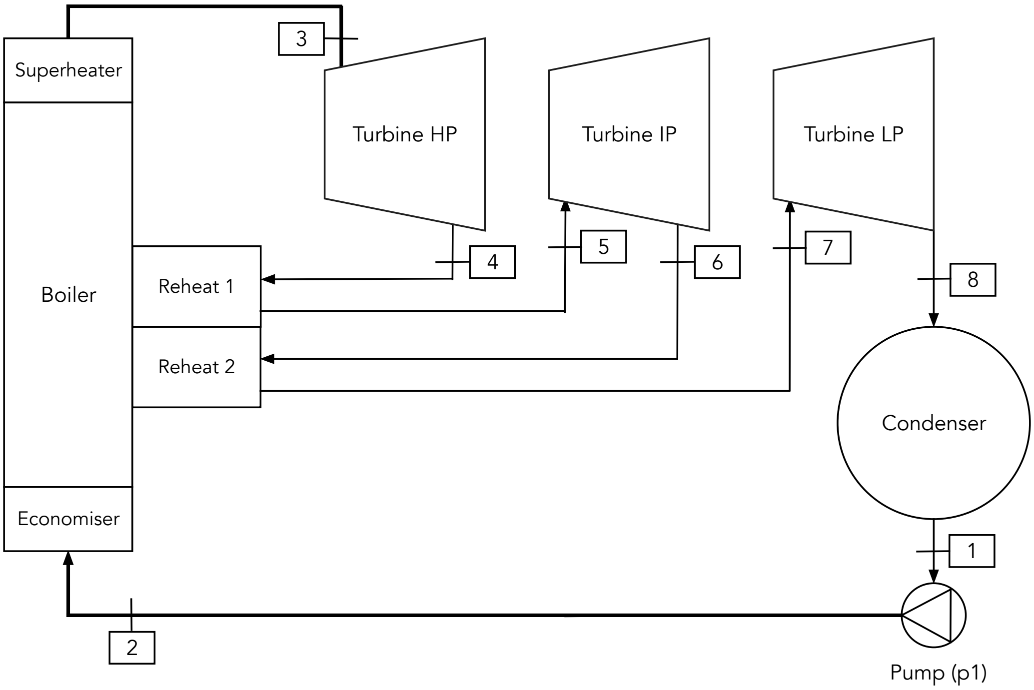

Figure 19: Components for a Rankine Supercritical cycle with double reheat.

Plot output

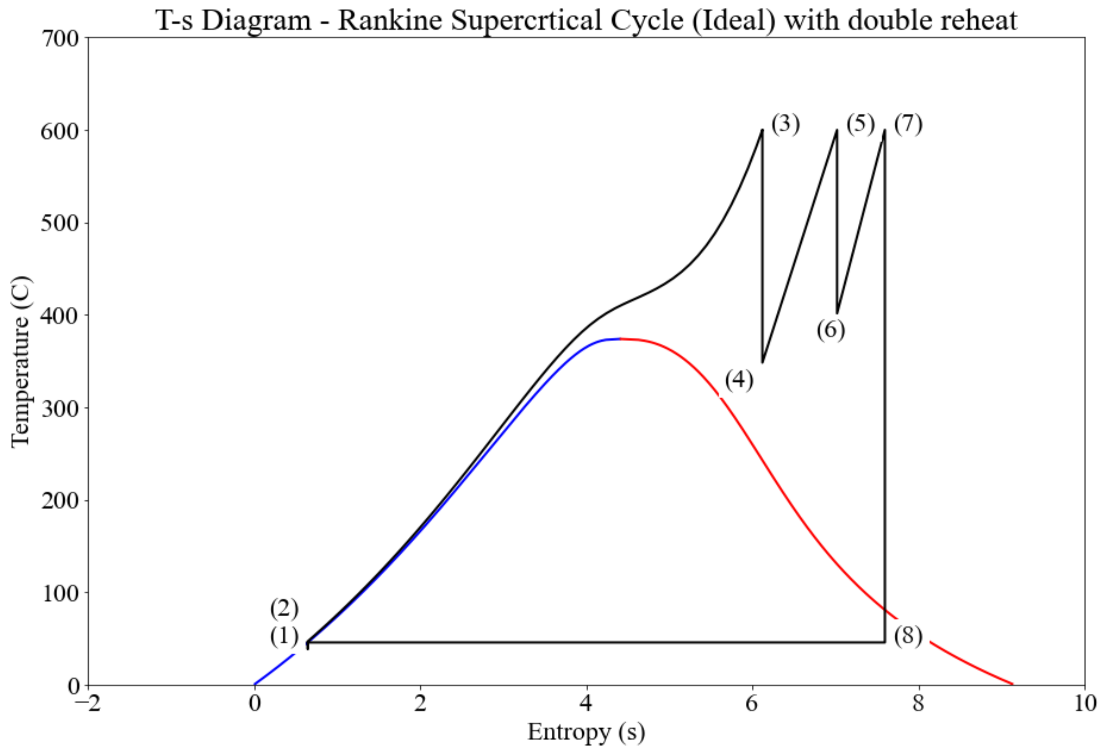

Figure 20: Ts plot of Rankine Supercritical cycle with single reheat.

The Python code to generate the plot above and efficiency calculations result below can be found in appendix E.10

Code output

```
Rankine supercritical cycle (double reheat) analysis

Point 1
P1: 0.1 bar
T1: 45.8 degC
H1: 191.8 kJ/kg
S1: 0.649 kJ/kg K

Point 2
H2: 192.2 kJ/kg
T2: 38.5 degC

Point 3
T3: 600.0 degC
P3: 350.0 bar
H3: 3399.0 kJ/kg
S3: 6.123 kJ/kg K
Reheat 1 Pressure: 80.0 bar

Point 4
T4: 348.3 degC
P4: 80.0 bar
H4: 2982.4 kJ/kg
S4: 6.123 kJ/kg K

Point 5
T5: 600.0 degC
p5: 80.0 bar
H5: 3642.4 kJ/kg
S5: 7.022 kJ/kg K
Reheat 2 Pressure: 25.0 bar
```

```

Point 6
T6: 401.6 degC
P6: 25.0 bar
H6: 3243.6 kJ/kg
S6: 7.022 kJ/kg K

Point 7
T7: 600.0 degC
p7: 25.0 bar
H7: 3686.8 kJ/kg
S7: 7.598 kJ/kg K

Point 8
T8: 45.8 degC
p8: 0.1 bar
H8: 2408.1 kJ/kg
S8: 7.598 kJ/kg K
x8: 92.7 % dry

Summary
Work required by pump: 0.4 kJ/kg
Work generated by HP turbine: 416.6 kJ/kg
Work generated by IP turbine: 398.9 kJ/kg
Work generated by LP turbine: 1278.7 kJ/kg
Total work output by turbine: 2094.1 kJ/kg
Heat input by boiler: 4310.1 kJ/kg
Heat rejected by the condenser: 2216.3 kJ/kg
Thermal efficiency is: 48.6%
HR rankine cycle: 7410.7 kJ/kWh
```

A Results tabulated

As each increase in boiler pressure and temperature is allowed by the development of new construction materials the resulted has been an increase in cycle theoretical thermal efficiency.

The results from Sections 1.1 to 1.10 are shown in the table below.

Table 1: Summary of cycle theoretical thermal efficiencies

Cycle type	**boiler P** (bar)	**boiler T** (deg C)	**reheat P** (bar)	**reheat T** (deg C)	**2nd reheat P** (bar)	**2nd reheat T** (deg C)	**condenser P** (bar)	**efficiency** (%)	**exhaust dryness** (%)
Carnot	120	324.7	na	na	na	na	1.0	37.6	69.2
Rankine	120	324.7	na	na	na	na	1.0	31.1	69.2
Rankine reduced condenser pressure	120	324.7	na	na	na	na	0.1	38.0	64.6
Rankine increased boiler pressure	180	357.0	na	na	na	na	0.1	38.7	59.4
Rankine superheat	180	400	na	na	na	na	0.1	40.3	67.2
Rankine superheat increased temperature	180	540	na	na	na	na	0.1	42.9	76.3
Rankine reheat	180	540	25	540	na	na	0.1	44.3	90.5
Rankine regenerative	180	540	25	540	na	na	0.1	46.8	90.5
Supercritical single reheat	350	600	85	600	na	na	0.1	47.4	85.0
Supercritical double reheat	350	600	85	600	25	600	0.1	48.6	92.7

P = pressure, T = temperature, na = not applicable

B Additional code

B.1 Cycle efficiency as fuel consumed

If you insert the following code at the end of examples E.1 to E.10 we can visualise the the efficiency in terms of fuel (coal in this case) required to generate a set generator output for a certain time (100 MW for a 24 hour period = 2400MWh).

```
#Additional code to calculate coal usage over time for a set power output
coalPA_GCV_HtVI = 24680.41 #Coal analysis higher heating value kJ/kg
PowerOutput = 100 #Power output from turbine in MW
TestSpan = 24 #24 hours
Wfe = HRcycle/coalPA_GCV_HtVI*1000*PowerOutput*TestSpan/1000 #coal flow required to meet power output over timespan
print(f"To generate {round(float(PowerOutput),0)} MW for {round(float(TestSpan),0)} hour requires {round(float(Wfe),1)} tons of coal")
```

Results for each cycle are shown in table 2 on page 34.

B.2 Cycle efficiency, site location and cooling water required

Another important aspect of designing and especially choosing a location for a power station is availability of a cooling medium. Generally cooling is provided by water from either a river or the ocean but it is usual for some limitation to be placed on the amount of water or the maximum temperature rise allowed by the local government.

If you insert the following code at the end of each example we can get an idea of the expected cooling water temperature rise of each cycle.

NOTE: the code is slightly different for some examples as shown below.

For examples E.1 ,E.2, E.3, E.4, E.5, E.6 please use the following code.

```
#Additional code to calculate CW temperature rise
CWflow = 5000 #kg/s
MassFlow = PowerOutput*1000/(w_HPt-w_p)
print(f"Required steam flow: {round(float(MassFlow),1)} kg/s")
MassFlowCond = PowerOutput*1000/(w_HPt-w_p)
print(f"Steam flow to condenser: {round(float(MassFlowCond),1)} kg/s")
QL = MassFlow * q_L
print(f"Heat rejected to condenser (total): {round(float(QL/1000),1)} MJ/s or MW")
DeltaTcw = QL/(CWflow * 4.18)
print(f"Temperature increase of cooling water: {round(float(DeltaTcw),1)} Deg C")
```

For examples E.7 and E.9 please use the following code.

```
#Additional code to calculate CW temperature rise
CWflow = 5000 #kg/s
MassFlow = PowerOutput*1000/(w_HPt+w_LPt-w_p)
print(f"Required steam flow: {round(float(MassFlow),1)} kg/s")
MassFlowCond = PowerOutput*1000/(w_HPt+w_LPt-w_p)
print(f"Steam flow to condenser: {round(float(MassFlowCond),1)} kg/s")
QL = MassFlow * q_L
print(f"Heat rejected to condenser (total): {round(float(QL/1000),1)} MJ/s or MW")
DeltaTcw = QL/(CWflow * 4.18)
print(f"Temperature increase of cooling water: {round(float(DeltaTcw),1)} Deg C")
```

For example E.8 please use the following code.

```
#Additional code to calculate CW temperature rise
CWflow = 5000 #kg/s
Wnett = (h5-h6)+(m2DIVm21*(h7-h8))-w_p2-(m2DIVm21*w_p1)
MassFlow = PowerOutput*1000/Wnett
print(f"Required steam flow: {round(float(MassFlow),1)} kg/s")
MassFlowCond = PowerOutput*1000/Wnett*m2DIVm21
print(f"Steam flow to condenser: {round(float(MassFlowCond),1)} kg/s")
QL = (MassFlow* m2DIVm21) * q_L*-1
print(f"Heat rejected to condenser (total): {round(float(QL/1000),1)} MJ/s or MW")
DeltaTcw = QL/(CWflow * 4.18)
print(f"Temperature increase of cooling water: {round(float(DeltaTcw),1)} Deg C")
```

For example E.10 please use the following code.

```
#Additional code to calculate CW temperature rise
CWflow = 5000 #kg/s
MassFlow = PowerOutput*1000/(w_HPt+w_IPt+w_LPt-w_p)
print(f"Required steam flow: {round(float(MassFlow),1)} kg/s")
MassFlowCond = PowerOutput*1000/(w_HPt+w_IPt+w_LPt-w_p)
print(f"Steam flow to condenser: {round(float(MassFlowCond),1)} kg/s")
QL = MassFlow * q_L
print(f"Heat rejected to condenser (total): {round(float(QL/1000),1)} MJ/s or MW")
DeltaTcw = QL/(CWflow * 4.18)
print(f"Temperature increase of cooling water: {round(float(DeltaTcw),1)} Deg C")
```

Results for each cycle are shown in table 2 on page 34.

B.3 Results interpreted

Table 2: Summary of cycle fuel usage and CW temperature rise for a given power output

Cycle type	**coal flow** (tons)	**cooling water T rise** (deg C)
Carnot	na	na
Rankine	1125.3	10.6
Rankine reduced condenser pressure	920.5	7.8
Rankine increased boiler pressure	905.3	7.6
Rankine superheat	867.6	7.1
Rankine superheat increased temperature	816.0	6.4
Rankine reheat	790.1	6.0
Rankine regenerative	747.4	5.4
Supercritical single reheat	738.0	5.3
Supercritical double reheat	720.6	5.1

For the same electrical power output (100MW) over the same period (24 hours) we find that a basic rankine cycle will use 1125.2 tons whilst an ultra modern supercritical plant with double reheat will use only 720.6 tons of coal (when a coal of CV 24680.41 kJ/kg is used).

This reduction in coal usage has many advantages, including;

- Reduced fuel costs.
- Reduced fuel transportation costs.
- Reduced stack emission quantity.
- Reduced fly ash and bottom ash for disposal.
- Coal storage area can be reduced.
- Coal handling plant can be smaller.
- Reduced plant size as the increase in pressure and temperature of a supercritical double reheat unit means that the energy contained in each kg of steam is much higher than in the basic rankine cycle plant.

Some disadvantages however do exist, including;

- Material costs to survive the supercritical pressures and temperatures are greatly increased.
- Plant complexity with double reheat is increased due to extra steam piping and turbine inlets, again increasing construction costs.

For a plant with small output (upto 200MW) it still makes financial sense to use a Rankine superheat cycle with no reheat. As we get into bigger size units (200-800MW) then a Rankine reheat cycle plant makes more sense. Above 800MW or so, then the additional cost of a supercritical plant is worthwhile.

Again for the same electrical power output (100MW) over the same period (24 hours) and using a set cooling water flow rate (5000kg/s) we find that a basic rankine cycle will have a condenser temperature rise of 10.6$°C$ whilst an ultra modern supercritical plant with double reheat will have a rise of only 5.1$°C$.

It is normal when preparing the license to build a new power plant to have some agreement on the amount of cooling water used and the allowed maximum temperature rise between inlet and outlet of the condenser to prevent damage to the local ecosystem. The agreed limits can have some impact on the size and type of plant to be built and in the extreme may force a change to the use of an air-cooled condenser. This change reduces water usage considerably but at the expense of lowering turbine efficiency (higher condenser pressure compared to a water cooled condenser).

Assuming we have an agreement to limit cooling water temperature rise to $6.0°C$, then those cycles in the above table 2 whose rise is above this value will need to increase the cooling water flow to bring it down to $6.0°C$. Any increase in cooling water flow will require a larger condenser, cooling water pipework and larger pumps with increased electrical consumption (and cost). In the case of the basic Rankine cycle to reduce the condenser temperature rise from $10.6°C$ down to the limit of $6.0°C$, the cooling water flow must be increased from 5000kg/s up to nearly 9000 kg/s meaning pumps, pipework and condenser water spaces needs to be nearly twice as large.

Notice that the Rankine reheat cycle and the Rankine regenerative cycle are basically the same cycle with the addition of a feedwater heater. This improves efficiency, hence the reduction in coal usage from 790.1 to 747.4 tons per day, but also reduces the condenser temperature rise from $6.0°C$ to $5.4°C$. This reduction in temperature rise is a result of a proportion of the steam, having done some work in the turbine, being bled off to heat the feedwater in a heater. This bled steam does not continue to the condenser so the total steam flow to the condenser is reduced and therefore the 5000kg/s of cooling water flow will result in a lower condenser temperature rise.

In reality the supercritical cycles would also be fitted with feedwater heater and there efficiency would also be increase further, plus reduction in coal flow and CW temperature rise.

C Ideal theoretical cycle meets the real world

No process that converts energy from one form to another can ever be 100% efficient and this is definitely the case in every power station where chemical energy (in my examples I have used coal) is converted to heat in the form of steam to drive a turbine which in turn is connected to a generator to produce electrical energy. Each of these steps have their own losses, which are detailed below.

For a typical coal fired boiler we have the following losses;

- Carbon in ash losses - unburnt carbon in fly and bottom ash.
- Moisture in fuel losses - heat added to moisture in the fuel is lost as steam at the stack.
- Hydrogen losses - burning hydrogen with oxygen produces water vapour which is also lost as steam at the stack.
- Moisture in combustion air loss - moisture in air inlet is heated and lost at stack. This is a very small loss as the moisture is already in the form of a water vapour at the air inlet.
- Dry flue gas loss - difference in temperature of fuel entering the boiler compared to exit temperature of combustion products is a loss.
- Radiation loss - heat lost from boiler surface due.
- Auxiliary power required to drive feedwater pumps, boiler fans, coal pulverisers, etc is also a loss.

Depending on the boiler size and the quality of fuel burnt these losses generally add up to 10-12%, therefore giving a boiler efficiency of 88-90%. In the calculations below for boiler efficiency a value of 89% has been used.

For a typical steam turbine we have the following major losses;

- Nozzle losses - friction between the steam and the nozzle wall.
- Blade losses - friction of steam passing over blade and formation of eddy current due to turbulence.
- Residual velocity losses - steam leaving the last row of blades has a velocity that is not used (also called leaving losses).
- Wetness losses - moisture entrained in the steam impacting the last row blades.

Plus some other minor losses;

- Disc friction loss.
- Diaphragm gland and blade tip leakage
- Shaft gland leakage
- Journal and thrust bearing friction losses
- Radiation losses
- Auxiliary power required to drive lube oil, control oil pumps, etc is also a loss.

And some losses associated with the turbine condenser;

- Air ingress to condenser steam space.
- Dirty or fouled tubes.

As the precision in manufacturing has increased, some of these losses have reduced, however they still add up to about 10%. The condenser losses however can have a large effect if left to deteriorate, so care should be taken to ensure tubes are keep clean and all sources of air ingress are found and eliminated. In the calculations below for turbine efficiency a value of 90% has been used.

When it comes to generator efficiency it is already approaching 98% and this is the value used below. To improve this further a jump to super-conductive windings is probably necessary.

Table 3: Expected efficiency of each cycle component

Component	Expected efficiency range	How to find the efficiency
Boiler	85 - 92%	ASME PTC 4 - Fire Steam Generators
Turbine	84 - 92%	ASME PTC 6 - Steam Turbines
Generator	96 - 98%	Generator OEM manual

Example 11

Taking as an example the results obtained in Section 1.7 where we looked at a Rankine reheat cycle which is representative of the cycle used in many actual power plant around the world. With main steam conditions of 180 bar, 540°C; reheat conditions of 25 bar, 540°C; and condenser pressure of 0.1 bar the theoretical thermal cycle efficiency attained is 44.3%.

Assuming the following efficiencies for each cycle component;

- Boiler = 89%
- Turbine = 90%
- Generator = 98%

$$\text{Real cycle efficiency} = \text{theoretical efficiency} \times \text{Boiler eff} \times \text{Turbine eff} \times \text{Generator eff}$$

$$\text{Real cycle efficiency} = 44.3\% \times 89\% \times 90\% \times 98\%$$

$$\text{Real cycle efficiency} = 34.8\%$$

So from a theoretical cycle efficiency of 44.3% for this Rankine reheat cycle we achieve in the real world only an actual efficiency of 34.8% in converting the fuel fed to the boiler into electrical output at the generator terminals.

A list of each cycle and the more realistic actual efficiency that can be achieved given the same component efficiencies above can be found in table 4.

Table 4: Summary of cycle theoretical thermal efficiencies and actual realistic efficiencies possible

Cycle type	efficiency (theoretical) (%)	efficiency (Realistic) (%)
Carnot	37.6	na
Rankine	31.1	24.4
Rankine reduced condenser pressure	38.0	29.8
Rankine increased boiler pressure	38.7	30.4
Rankine superheat	40.3	31.6
Rankine superheat increased temperature	42.9	33.7
Rankine reheat	44.3	34.8
Rankine regenerative	46.8	36.7
Supercritical single reheat	47.4	37.2
Supercritical double reheat	48.6	38.2

D Software

All code example in this document have been tested using Python v3.7.6 under macOS and Windows 8. In addition the following packages within python have been used;

- Matplotlib v3.2.2 ⟶ https://matplotlib.org
- Numpy v1.18.5 ⟶ https://numpy.org
- pyXSteam v0.4.4 ⟶ https://pypi.org/project/pyXSteam

E Code printouts

E.1 Carnot cycle

```
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

p1 = 1
p2 = 120
s1 = steamTable.sL_p(p2)

print('Carnot cycle analysis')

T1 = steamTable.t_ps(p1, s1)
print('\nPoint 1')
print(f"P1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"S1: {round(float(s1),3)} kJ/kg K")

s2 = s1

h1 = steamTable.h_ps(p1, s1)
print(f"H1: {round(float(h1),1)} kJ/kg")

h2 = steamTable.hL_p(p2)
s2 = steamTable.sL_p(p2)
T2 = steamTable.t_ph(p2, h2)
print('\nPoint 2')
print(f"P2: {round(float(p2),1)} bar")
print(f"T2: {round(float(T2),1)} degC")
print(f"H2: {round(float(h2),1)} kJ/kg")
print(f"S2: {round(float(s2),3)} kJ/kg K")

h3 = steamTable.hV_p(p2)
s3 = steamTable.sV_p(p2)
T3 = T2
p3 = p2
print('\nPoint 3')
print(f"P3: {round(float(p3),1)} bar")
print(f"T3: {round(float(T3),1)} degC")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = p1
s4 = s3
T4 = steamTable.t_ps(p4, s4)
x4 = steamTable.x_ps(p4, s4)
h4 = steamTable.h_ps(p4, s4)
print('\nPoint 4')
print(f"P4: {round(float(p4),1)} bar")
print(f"T4: {round(float(T4),1)} degC")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")
print(f"x4: {round(float(x4*100),1)} % dry")

print('\nSummary')
w_p = (h2 - h1)
print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by turbine: {round(float(w_HPt),1)} kJ/kg")

q_H = (h3-h2)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h4-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

eta_th = ((w_HPt-w_p)/q_H)*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")
```

```
font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Carnot Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s3, s4, s1],[T1, T2, T3, T4, T1], 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s2-.15,T2,f"(2)",
    ha='right',backgroundcolor='white')
plt.text(s3+.1,T3,f"(3)",
    ha='left',backgroundcolor='white')
plt.text(s4+.1,T4,f'(4)',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-01.png')
```

E.2 Rankine cycle

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine cycle analysis')

p1 = 1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"p1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 120
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

h2dash = steamTable.hL_p(p2)
s2dash = steamTable.sL_p(p2)
T2dash = steamTable.t_ph(p2, h2dash)
print('\nPoint 2dash')
print(f"H2dash: {round(float(h2dash),1)} kJ/kg")
print(f"S2dash: {round(float(s2dash),3)} kJ/kg K")
print(f"T2dash: {round(float(T2dash),1)} degC")

p3 = p2
h3 = steamTable.hV_p(p2)
s3 = steamTable.sV_p(p2)
T3 = T2dash
print('\nPoint 3')
print(f"P3: {round(float(p3),1)} bar")
print(f"T3: {round(float(T3),1)} degC")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = p1
s4 = s3
T4 = steamTable.t_ps(p4, s4)
x4 = steamTable.x_ps(p4, s4)
h4 = steamTable.h_px(p4, x4)
print('\nPoint 4')
print(f"P4: {round(float(p4),1)} bar")
print(f"T4: {round(float(T4),1)} degC")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")
print(f"x4: {round(float(x4*100),1)} % dry")

print('\nSummary')

print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by turbine: {round(float(w_HPt),1)} kJ/kg")

q_H = (h3-h2)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h4-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

```

```
eta_th = (w_HPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s2dash, s3, s4, s1],[T1, T2, T2dash, T3, T4, T1], 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s2dash-.15,T2dash,f"(2')",
    ha='right',backgroundcolor='white')
plt.text(s3+.1,T3,f"(3)",
    ha='left',backgroundcolor='white')
plt.text(s4+.1,T4,f'(4))',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-02.png')
```

E.3 Rankine cycle (reduced condenser pressure)

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine cycle analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"p1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 120
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

h2dash = steamTable.hL_p(p2)
s2dash = steamTable.sL_p(p2)
T2dash = steamTable.t_ph(p2, h2dash)
print('\nPoint 2dash')
print(f"H2dash: {round(float(h2dash),1)} kJ/kg")
print(f"S2dash: {round(float(s2dash),3)} kJ/kg K")
print(f"T2dash: {round(float(T2dash),1)} degC")

p3 = p2
h3 = steamTable.hV_p(p2)
s3 = steamTable.sV_p(p2)
T3 = T2dash
print('\nPoint 3')
print(f"P3: {round(float(p3),1)} bar")
print(f"T3: {round(float(T3),1)} degC")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = p1
s4 = s3
T4 = steamTable.t_ps(p4, s4)
x4 = steamTable.x_ps(p4, s4)
h4 = steamTable.h_px(p4, x4)
print('\nPoint 4')
print(f"P4: {round(float(p4),1)} bar")
print(f"T4: {round(float(T4),1)} degC")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")
print(f"x4: {round(float(x4*100),1)} % dry")

print('\nSummary')

print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by turbine: {round(float(w_HPt),1)} kJ/kg")

q_H = (h3-h2)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h4-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

```

```python
eta_th = (w_HPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s2dash, s3, s4, s1],[T1, T2, T2dash, T3, T4, T1], 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s2dash-.15,T2dash,f"(2')",
    ha='right',backgroundcolor='white')
plt.text(s3+.1,T3,f"(3)",
    ha='left',backgroundcolor='white')
plt.text(s4+.1,T4,f'(4))',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-03.png')
```

E.4 Rankine cycle (increased boiler pressure)

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine cycle analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"p1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 180
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

h2dash = steamTable.hL_p(p2)
s2dash = steamTable.sL_p(p2)
T2dash = steamTable.t_ph(p2, h2dash)
print('\nPoint 2dash')
print(f"H2dash: {round(float(h2dash),1)} kJ/kg")
print(f"S2dash: {round(float(s2dash),3)} kJ/kg K")
print(f"T2dash: {round(float(T2dash),1)} degC")

p3 = p2
h3 = steamTable.hV_p(p2)
s3 = steamTable.sV_p(p2)
T3 = T2dash
print('\nPoint 3')
print(f"P3: {round(float(p3),1)} bar")
print(f"T3: {round(float(T3),1)} degC")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = p1
s4 = s3
T4 = steamTable.t_ps(p4, s4)
x4 = steamTable.x_ps(p4, s4)
h4 = steamTable.h_px(p4, x4)
print('\nPoint 4')
print(f"P4: {round(float(p4),1)} bar")
print(f"T4: {round(float(T4),1)} degC")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")
print(f"x4: {round(float(x4*100),1)} % dry")

print('\nSummary')

print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by turbine: {round(float(w_HPt),1)} kJ/kg")

q_H = (h3-h2)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h4-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

```

```
eta_th = (w_HPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s2dash, s3, s4, s1],[T1, T2, T2dash, T3, T4, T1], 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s2dash-.15,T2dash,f"(2')",
    ha='right',backgroundcolor='white')
plt.text(s3+.1,T3,f"(3)",
    ha='left',backgroundcolor='white')
plt.text(s4+.1,T4,f'(4))',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-04.png')
```

E.5 Rankine superheat cycle

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine superheat cycle analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"p1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 180
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

h2dash = steamTable.hL_p(p2)
s2dash = steamTable.sL_p(p2)
T2dash = steamTable.t_ph(p2, h2dash)
print('\nPoint 2 dash')
print(f"T2dash: {round(float(T2dash),1)} degC")
print(f"p2dash: {round(float(p2),1)} bar")
print(f"H2dash: {round(float(h2dash),1)} kJ/kg")
print(f"S2dash: {round(float(s2dash),3)} kJ/kg K")

h3dash = steamTable.hV_p(p2)
s3dash = steamTable.sV_p(p2)
T3dash = T2dash
print('\nPoint 3dash')
print(f"T3dash: {round(float(T3dash),1)} degC")
print(f"H3dash: {round(float(h3dash),1)} kJ/kg")
print(f"S3dash: {round(float(s3dash),3)} kJ/kg K")

p3 = p2
T3 = 400
h3 = steamTable.h_pt(p3, T3)
s3 = steamTable.s_pt(p3, T3)
print('\nPoint 3')
print(f"T3: {round(float(T3),1)} degC")
print(f"p3: {round(float(p3),1)} bar")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = p1
s4 = s3
T4 = steamTable.t_ps(p4, s4)
x4 = steamTable.x_ps(p4, s4)
h4 = steamTable.h_px(p4, x4)
print('\nPoint 4')
print(f"T4: {round(float(T4),1)} degC")
print(f"p4: {round(float(p4),1)} bar")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")
print(f"x4: {round(float(x4*100),1)} % dry")

print('\nSummary')
print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
```

```
print(f"Work generated by turbine: {round(float(w_HPt),1)} kJ/kg")

q_H = (h3-h2)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h4-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

eta_th = (w_HPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Superheat Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s2dash, s3dash, s3, s4, s1],[T1, T2, T2dash, T3dash, T3, T4, T1], 'black',
    linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s2dash-.15,T2dash,f"(2')",
    ha='right',backgroundcolor='white')
plt.text(s3dash-.1,T3dash-25,f"(3')",
    ha='right',backgroundcolor='white')
plt.text(s3+.2,T3,f'(3)',
    ha='left',backgroundcolor='white')
plt.text(s4+.1,T4,f'(4)',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-05.png')
```

E.6 Rankine superheat cycle (increased superheat temperature)

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine superheat cycle analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"p1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 180
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

h2dash = steamTable.hL_p(p2)
s2dash = steamTable.sL_p(p2)
T2dash = steamTable.t_ph(p2, h2dash)
print('\nPoint 2 dash')
print(f"T2dash: {round(float(T2dash),1)} degC")
print(f"p2dash: {round(float(p2),1)} bar")
print(f"H2dash: {round(float(h2dash),1)} kJ/kg")
print(f"S2dash: {round(float(s2dash),3)} kJ/kg K")

h3dash = steamTable.hV_p(p2)
s3dash = steamTable.sV_p(p2)
T3dash = T2dash
print('\nPoint 3dash')
print(f"T3dash: {round(float(T3dash),1)} degC")
print(f"H3dash: {round(float(h3dash),1)} kJ/kg")
print(f"S3dash: {round(float(s3dash),3)} kJ/kg K")

p3 = p2
T3 = 540
h3 = steamTable.h_pt(p3, T3)
s3 = steamTable.s_pt(p3, T3)
print('\nPoint 3')
print(f"T3: {round(float(T3),1)} degC")
print(f"p3: {round(float(p3),1)} bar")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = p1
s4 = s3
T4 = steamTable.t_ps(p4, s4)
x4 = steamTable.x_ps(p4, s4)
h4 = steamTable.h_px(p4, x4)
print('\nPoint 4')
print(f"T4: {round(float(T4),1)} degC")
print(f"p4: {round(float(p4),1)} bar")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")
print(f"x4: {round(float(x4*100),1)} % dry")

print('\nSummary')
print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
```

```
print(f"Work generated by turbine: {round(float(w_HPt),1)} kJ/kg")

q_H = (h3-h2)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h4-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

eta_th = (w_HPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Superheat Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s2dash, s3dash, s3, s4, s1],[T1, T2, T2dash, T3dash, T3, T4, T1], 'black',
    linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s2dash-.15,T2dash,f"(2')",
    ha='right',backgroundcolor='white')
plt.text(s3dash-.1,T3dash-25,f"(3')",
    ha='right',backgroundcolor='white')
plt.text(s3+.2,T3,f'(3)',
    ha='left',backgroundcolor='white')
plt.text(s4+.1,T4,f'(4)',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-06.png')
```

E.7 Rankine reheat cycle

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine reheat cycle analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"P1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 180
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

h2dash = steamTable.hL_p(p2)
s2dash = steamTable.sL_p(p2)
T2dash = steamTable.t_ph(p2, h2dash)
print('\nPoint 2dash')
print(f"T2dash: {round(float(T2dash),1)} degC")
print(f"P2dash: {round(float(p2),1)} bar")
print(f"H2dash: {round(float(h2dash),1)} kJ/kg")
print(f"S2dash: {round(float(s2dash),3)} kJ/kg K")

h3dash = steamTable.hV_p(p2)
s3dash = steamTable.sV_p(p2)
T3dash = T2dash
print('\nPoint 3dash')
print(f"T3dash: {round(float(T3dash),1)} degC")
print(f"H3dash: {round(float(h3dash),1)} kJ/kg")
print(f"S3dash: {round(float(s3dash),3)} kJ/kg K")

p3 = p2
T3 = 540
h3 = steamTable.h_pt(p3, T3)
s3 = steamTable.s_pt(p3, T3)
print('\nPoint 3')
print(f"T3: {round(float(T3),1)} degC")
print(f"P3: {round(float(p3),1)} bar")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = 25
print(f"Reheat Pressure: {round(float(p4),1)} bar")
s4 = s3
T4 = steamTable.t_ps(p4, s4)
h4 = steamTable.h_pt(p4, T4)
print('\nPoint 4')
print(f"T4: {round(float(T4),1)} degC")
print(f"P4: {round(float(p4),1)} bar")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")

p5 = p4
T5 = T3
h5 = steamTable.h_pt(p5, T5)
s5 = steamTable.s_pt(p5, T5)
print('\nPoint 5')
```

```
print(f"T5: {round(float(T5),1)} degC")
print(f"p5: {round(float(p5),1)} bar")
print(f"H5: {round(float(h5),1)} kJ/kg")
print(f"S5: {round(float(s5),3)} kJ/kg K")

p6 = p1
s6 = s5
T6 = steamTable.t_ps(p6, s6)
x6 = steamTable.x_ps(p6, s6)
h6 = steamTable.h_px(p6, x6)
print('\nPoint 6')
print(f"T6: {round(float(T6),1)} degC")
print(f"p6: {round(float(p6),1)} bar")
print(f"H6: {round(float(h6),1)} kJ/kg")
print(f"S6: {round(float(s6),3)} kJ/kg K")
print(f"x6: {round(float(x6*100),1)} % dry")

print('\nSummary')
print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by HP turbine: {round(float(w_HPt),1)} kJ/kg")

w_LPt = h5-h6
print(f"Work generated by LP turbine: {round(float(w_LPt),1)} kJ/kg")
print(f"Total work output by turbine: {round(float(w_HPt+w_LPt),1)} kJ/kg")

q_H = (h3-h2)+(h5-h4)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h6-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

eta_th = (w_HPt+w_LPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Reheat Cycle (Ideal)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s2dash, s3dash, s3, s4, s5, s6, s1],[T1, T2, T2dash, T3dash, T3, T4, T5, T6,
     T1], 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s2dash-.15,T2dash,f"(2')",
    ha='right',backgroundcolor='white')
plt.text(s3dash-.1,T3dash-25,f"(3')",
    ha='right',backgroundcolor='white')
plt.text(s3+.2,T3,f'(3)',
    ha='left',backgroundcolor='white')
plt.text(s4-.1,T4-25,f'(4)',
    ha='right',backgroundcolor='white')
plt.text(s5+.2,T5,f'(5)',
    ha='left',backgroundcolor='white')
```

```
plt.text(s6+.1,T6,f'(6)',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-07.png')
```

E.8 Rankine reheat cycle with heaters

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine reheat cycle analysis with 1 open feedheater')

PowerOutput = 100 #MW electrical generation
p1 = 0.1 #condenser pressure (bar)
p2 = 25 #turbine bled steam pressure (bar) and pump 1 discharge pressure (bar)
p3 = 180 #feedwater pressure (bar)
p4 = 180 #main steam pressure (bar)
T5 = 540 #superheat temperature
T7 = 540 #reheat temperature

s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"P1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

s2 = s1

v = 1/steamTable.rhoL_p(p1)*100
w_p1 = v*(p2-p1)
print(f"Work required by pump 1: {round(float(w_p1),2)} kJ/kg")

print('\nPoint 2')
h2 = h1+w_p1
T2 = steamTable.t_ph(p3, h2)
print(f"P2: {round(float(p2),1)} bar")
print(f"H2: {round(float(h2),1)} kJ/kg")
print(f"T2: {round(float(T2),1)} degC")

print('\nPoint 21 - feedheater outlet before pump 2')
p21 = p2
h21 = steamTable.hL_p(p2)
s21 = steamTable.sL_p(p2)
T21 = steamTable.t_hs(h21, s21)
print(f"P21: {round(float(p21),1)} bar")
print(f"T21: {round(float(T21),1)} degC")
print(f"H21: {round(float(h21),1)} kJ/kg")
print(f"S21: {round(float(s21),3)} kJ/kg K")

p22 = p3
v2 = 1/steamTable.rhoL_p(p21)*100
w_p2 = v2*(p22-p21)
print(f"Work required by pump 2: {round(float(w_p2),2)} kJ/kg")

T22 = T21
h22 = h21 + w_p2
s22 = steamTable.s_pt(p22,T22)
print('\nPoint 22 - pump 2 outlet')
print(f"P22: {round(float(p22),1)} bar")
print(f"T22: {round(float(T22),1)} degC")
print(f"H22: {round(float(h22),1)} kJ/kg")
print(f"S22: {round(float(s22),3)} kJ/kg K")

h3 = h22
s3 = steamTable.sL_p(p3)
T3 = steamTable.t_ph(p3, h3)
T3 = steamTable.tsat_p(p3)
print('\nPoint 3')
print(f"P3: {round(float(p3),1)} bar")
print(f"T3: {round(float(T3),1)} degC")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

h4 = steamTable.hV_p(p4)
s4 = steamTable.sV_p(p4)
```

```
T4 = T3
print('\nPoint 4')
print(f"P4: {round(float(p4),1)} bar")
print(f"T4: {round(float(T4),1)} degC")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")

#HP turbine inlet conditions
p5 = p4
h5 = steamTable.h_pt(p5, T5)
s5 = steamTable.s_pt(p5, T5)
print('\nPoint 5 - main steam conditions')
print(f"P5: {round(float(p5),1)} bar")
print(f"T5: {round(float(T5),1)} degC")
print(f"H5: {round(float(h5),1)} kJ/kg")
print(f"S5: {round(float(s5),3)} kJ/kg K")

p6 = p2
print(f"Reheat Pressure: {round(float(p6),1)} bar")
s6 = s5
T6 = steamTable.t_ps(p6, s6)
h6 = steamTable.h_pt(p6, T6)
print('\nPoint 6')
print(f"P6: {round(float(p6),1)} bar")
print(f"T6: {round(float(T6),1)} degC")
print(f"H6: {round(float(h6),1)} kJ/kg")
print(f"S6: {round(float(s6),3)} kJ/kg K")

#LP turbine inlet conditions
p7 = p2
h7 = steamTable.h_pt(p7, T7)
s7 = steamTable.s_pt(p7, T7)
print('\nPoint 7 - IP/LP steam conditions')
print(f"P7: {round(float(p7),1)} bar")
print(f"T7: {round(float(T7),1)} degC")
print(f"H7: {round(float(h7),1)} kJ/kg")
print(f"S7: {round(float(s7),3)} kJ/kg K")

#turbine outlet conditions
p8 = p1
s8 = s7 #assume isentropic expansion in turbine
T8 = steamTable.t_ps(p8, s8)
x8 = steamTable.x_ps(p8, s8)
h8 = steamTable.h_px(p8, x8)
print('\nPoint 8 - turbine exhaust conditions')
print(f"P8: {round(float(p8),1)} bar")
print(f"T8: {round(float(T8),1)} degC")
print(f"H8: {round(float(h8),1)} kJ/kg")
print(f"S8: {round(float(s8),3)} kJ/kg K")
print(f"x8: {round(float(x8*100),1)} % dry")

#calculate heater dry saturation point for plot
p61 = p6
h61 = h6
T61 = steamTable.tsat_p(p61)
s61 = steamTable.sV_p(p61)

# differing mass flow rates at various points
m2DIVm21 = ((h21-h6)/(h2-h6))
print(f"\nfeedwater heater mass flow ratio: {round(float(m2DIVm21),4)} ")

print('\nSummary')
q_H = (h5-h22)+(h7-h6)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = (h1-h8)
print(f"Heat rejected to condenser: {round(float(q_L),1)} kJ/kg")

w_HPt = h5-h6
print(f"Work generated by HP turbine: {round(float(w_HPt),1)} kJ/kg")

w_LPt = h7-h8
print(f"Work generated by LP turbine: {round(float(w_LPt),1)} kJ/kg")
print(f"Total work output by turbine: {round(float(w_HPt+w_LPt),1)} kJ/kg")

Wnett = (h5-h6)+(m2DIVm21*(h7-h8))-w_p2-(m2DIVm21*w_p1)
```

```
print(f"Thermal efficiency is: {round(float(Wnett),1)} kJ/kg")

eta_th = Wnett/((h5-h22)+(m2DIVm21*(h7-h6)))*100
print(f"Thermal efficiency is: {round(float(eta_th),1)} %")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

MassFlow = PowerOutput*1000/Wnett
print(f"Required steam flow: {round(float(MassFlow),1)} kg/s")

MassFlowCond = PowerOutput*1000/Wnett*m2DIVm21
print(f"Steam flow to condenser: {round(float(MassFlowCond),1)} kg/s")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Reheat Cycle (1 open feed heater)')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,600)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

plt.plot([s1, s2, s21, s22, s3, s4, s5, s6, s7, s8, s1],[T1, T2, T21, T22, T3, T4, T5, T6, T7, T8, T1], 'black', linewidth=2.0)
plt.plot([s6, s61, s21],[T6, T61, T21], 'green', linewidth=2.0) #feed heater line

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s3-.15,T3,f"(3)",
    ha='right',backgroundcolor='white')
plt.text(s4-.1,T4-25,f"(4')",
    ha='right',backgroundcolor='white')
plt.text(s5+.2,T5,f'(5)',
    ha='left',backgroundcolor='white')
plt.text(s6-.1,T6-25,f'(6)',
    ha='right',backgroundcolor='white')
plt.text(s7+.2,T7,f'(7)',
    ha='left',backgroundcolor='white')
plt.text(s8+.1,T8,f'(8)',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-08.png')
```

E.9 Rankine supercritical cycle (single reheat)

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine supercritical cycle (single reheat) analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"P1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 350
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

p3 = p2
T3 = 600
h3 = steamTable.h_pt(p3, T3)
s3 = steamTable.s_pt(p3, T3)
print('\nPoint 3')
print(f"T3: {round(float(T3),1)} degC")
print(f"P3: {round(float(p3),1)} bar")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = 80
print(f"Reheat Pressure: {round(float(p4),1)} bar")
s4 = s3
T4 = steamTable.t_ps(p4, s4)
h4 = steamTable.h_pt(p4, T4)
print('\nPoint 4')
print(f"T4: {round(float(T4),1)} degC")
print(f"P4: {round(float(p4),1)} bar")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")

p5 = p4
T5 = T3
h5 = steamTable.h_pt(p5, T5)
s5 = steamTable.s_pt(p5, T5)
print('\nPoint 5')
print(f"T5: {round(float(T5),1)} degC")
print(f"p5: {round(float(p5),1)} bar")
print(f"H5: {round(float(h5),1)} kJ/kg")
print(f"S5: {round(float(s5),3)} kJ/kg K")

p6 = p1
s6 = s5
T6 = steamTable.t_ps(p6, s6)
x6 = steamTable.x_ps(p6, s6)
h6 = steamTable.h_px(p6, x6)
print('\nPoint 6')
print(f"T6: {round(float(T6),1)} degC")
print(f"p6: {round(float(p6),1)} bar")
print(f"H6: {round(float(h6),1)} kJ/kg")
print(f"S6: {round(float(s6),3)} kJ/kg K")
print(f"x6: {round(float(x6*100),1)} % dry")

```

```
print('\nSummary')
print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by HP turbine: {round(float(w_HPt),1)} kJ/kg")

w_LPt = h5-h6
print(f"Work generated by LP turbine: {round(float(w_LPt),1)} kJ/kg")
print(f"Total work output by turbine: {round(float(w_HPt+w_LPt),1)} kJ/kg")

q_H = (h3-h2)+(h5-h4)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h6-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

eta_th = (w_HPt+w_LPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Supercrtical Cycle (Ideal) with single reheat')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,700)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

superlistx = [s1, s2]
superlisty = [T1, T2]
for x in np.arange(s1, s3, 0.1):
    Tx = steamTable.t_ps(p2, x)
    hxdash = steamTable.h_pt(p2, Tx)
    sxdash = steamTable.s_pt(p2, Tx)
    Txdash = steamTable.t_ps(p2, sxdash)
    superlistx.append(sxdash)
    superlisty.append(Txdash)

hxdash = steamTable.h_pt(p2, T3)
sxdash = steamTable.s_pt(p2, T3)
Txdash = steamTable.t_ps(p2, sxdash)
superlistx.append(sxdash)
superlisty.append(Txdash)

superlistx.extend([s3, s4, s5, s6, s1])
superlisty.extend([T3, T4, T5, T6, T1])

plt.plot(superlistx, superlisty, 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s3+.2,T3,f'(3)',
    ha='left',backgroundcolor='white')
plt.text(s4-.1,T4-25,f'(4)',
    ha='right',backgroundcolor='white')
plt.text(s5+.2,T5,f'(5)',
    ha='left',backgroundcolor='white')
plt.text(s6+.1,T6,f'(6)',
    ha='left',backgroundcolor='white')

```

```
plt.savefig('Plot-09.png')
```

E.10 Rankine supercritical cycle (double reheat)

```python
import matplotlib.pyplot as plt
import numpy as np
from pyXSteam.XSteam import XSteam

steamTable = XSteam(XSteam.UNIT_SYSTEM_MKS)

print('Rankine supercritical cycle (double reheat) analysis')

p1 = 0.1
s1 = steamTable.sL_p(p1)
T1 = steamTable.t_ps(p1, s1)
h1 = steamTable.hL_p(p1)
print('\nPoint 1')
print(f"P1: {round(float(p1),1)} bar")
print(f"T1: {round(float(T1),1)} degC")
print(f"H1: {round(float(h1),1)} kJ/kg")
print(f"S1: {round(float(s1),3)} kJ/kg K")

p2 = 350
s2 = s1

v = 1/steamTable.rhoL_p(p1)
w_p = v*(p2-p1)

print('\nPoint 2')
h2 = h1+w_p
print(f"H2: {round(float(h2),1)} kJ/kg")

T2 = steamTable.t_ph(p2, h2)
print(f"T2: {round(float(T2),1)} degC")

p3 = p2
T3 = 600
h3 = steamTable.h_pt(p3, T3)
s3 = steamTable.s_pt(p3, T3)
print('\nPoint 3')
print(f"T3: {round(float(T3),1)} degC")
print(f"P3: {round(float(p3),1)} bar")
print(f"H3: {round(float(h3),1)} kJ/kg")
print(f"S3: {round(float(s3),3)} kJ/kg K")

p4 = 80
print(f"Reheat 1 Pressure: {round(float(p4),1)} bar")
s4 = s3
T4 = steamTable.t_ps(p4, s4)
h4 = steamTable.h_pt(p4, T4)
print('\nPoint 4')
print(f"T4: {round(float(T4),1)} degC")
print(f"P4: {round(float(p4),1)} bar")
print(f"H4: {round(float(h4),1)} kJ/kg")
print(f"S4: {round(float(s4),3)} kJ/kg K")

p5 = p4
T5 = T3
h5 = steamTable.h_pt(p5, T5)
s5 = steamTable.s_pt(p5, T5)
print('\nPoint 5')
print(f"T5: {round(float(T5),1)} degC")
print(f"p5: {round(float(p5),1)} bar")
print(f"H5: {round(float(h5),1)} kJ/kg")
print(f"S5: {round(float(s5),3)} kJ/kg K")

#second reheat
p6 = 25
print(f"Reheat 2 Pressure: {round(float(p6),1)} bar")
s6 = s5
T6 = steamTable.t_ps(p6, s6)
h6 = steamTable.h_pt(p6, T6)
print('\nPoint 6')
print(f"T6: {round(float(T6),1)} degC")
print(f"P6: {round(float(p6),1)} bar")
print(f"H6: {round(float(h6),1)} kJ/kg")
print(f"S6: {round(float(s6),3)} kJ/kg K")

```

```
p7 = p6
T7 = T3
h7 = steamTable.h_pt(p7, T7)
s7 = steamTable.s_pt(p7, T7)
print('\nPoint 7')
print(f"T7: {round(float(T7),1)} degC")
print(f"p7: {round(float(p7),1)} bar")
print(f"H7: {round(float(h7),1)} kJ/kg")
print(f"S7: {round(float(s7),3)} kJ/kg K")

p8 = p1
s8 = s7
T8 = steamTable.t_ps(p8, s8)
x8 = steamTable.x_ps(p8, s8)
h8 = steamTable.h_px(p8, x8)
print('\nPoint 8')
print(f"T8: {round(float(T8),1)} degC")
print(f"p8: {round(float(p8),1)} bar")
print(f"H8: {round(float(h8),1)} kJ/kg")
print(f"S8: {round(float(s8),3)} kJ/kg K")
print(f"x8: {round(float(x8*100),1)} % dry")

print('\nSummary')
print(f"Work required by pump: {round(float(w_p),1)} kJ/kg")

w_HPt = h3-h4
print(f"Work generated by HP turbine: {round(float(w_HPt),1)} kJ/kg")

w_IPt = h5-h6
print(f"Work generated by IP turbine: {round(float(w_IPt),1)} kJ/kg")

w_LPt = h7-h8
print(f"Work generated by LP turbine: {round(float(w_LPt),1)} kJ/kg")
print(f"Total work output by turbine: {round(float(w_HPt+w_IPt+w_LPt),1)} kJ/kg")

q_H = (h3-h2)+(h5-h4)+(h7-h6)
print(f"Heat input by boiler: {round(float(q_H),1)} kJ/kg")

q_L = h8-h1
print(f"Heat rejected by the condenser: {round(float(q_L),1)} kJ/kg")

eta_th = (w_HPt+w_IPt+w_LPt-w_p)/q_H*100
print(f"Thermal efficiency is: {round(float(eta_th),1)}%")

HRcycle = 3600*100/eta_th
print(f"HR rankine cycle: {round(float(HRcycle),1)} kJ/kWh")

font = {'family' : 'Times New Roman',
        'size'   : 22}

plt.figure(figsize=(15,10))
plt.title('T-s Diagram - Rankine Supercrtical Cycle (Ideal) with double reheat')
plt.rc('font', **font)

plt.ylabel('Temperature (C)')
plt.xlabel('Entropy (s)')
plt.xlim(-2,10)
plt.ylim(0,700)

T = np.linspace(0, 373.945, 400) # range of temperatures
# saturated vapor and liquid entropy lines
svap = [s for s in [steamTable.sL_t(t) for t in T]]
sliq = [s for s in [steamTable.sV_t(t) for t in T]]

plt.plot(svap, T, 'b-', linewidth=2.0)
plt.plot(sliq, T, 'r-', linewidth=2.0)

superlistx = [s1, s2]
superlisty = [T1, T2]
for x in np.arange(s1, s3, 0.1):
    Tx = steamTable.t_ps(p2, x)
    hxdash = steamTable.h_pt(p2, Tx)
    sxdash = steamTable.s_pt(p2, Tx)
    Txdash = steamTable.t_ps(p2, sxdash)
    superlistx.append(sxdash)
    superlisty.append(Txdash)
```

```python

hxdash = steamTable.h_pt(p2, T3)
sxdash = steamTable.s_pt(p2, T3)
Txdash = steamTable.t_ps(p2, sxdash)
superlistx.append(sxdash)
superlisty.append(Txdash)

superlistx.extend([s3, s4, s5, s6, s7, s8, s1])
superlisty.extend([T3, T4, T5, T6, T7, T8, T1])

plt.plot(superlistx, superlisty, 'black', linewidth=2.0)

plt.text(s1-.1,T1,f'(1)',
    ha='right',backgroundcolor='white')
plt.text(s1-.1,T1+30,f'(2)',
    ha='right',backgroundcolor='white')
plt.text(s3+.1,T3,f'(3)',
    ha='left',backgroundcolor='white')
plt.text(s4-.1,T4-25,f'(4)',
    ha='right',backgroundcolor='white')
plt.text(s5+.1,T5,f'(5)',
    ha='left',backgroundcolor='white')
plt.text(s6+.1,T6-25,f'(6)',
    ha='right',backgroundcolor='white')
plt.text(s7+.1,T7,f'(7)',
    ha='left',backgroundcolor='white')
plt.text(s8+.1,T8,f'(8)',
    ha='left',backgroundcolor='white')

plt.savefig('Plot-10.png')
```

F Figures and Tables

List of Figures

List of Tables