Consider an adiabatic turbine with an isentropic efficiency of $\eta_s = 0.8$ and an exit pressure of 50 kPa (or 0.5 bar). For a given inlet temperature $T_1$, calculate the inlet pressure $P_1$, which will yield a saturated vapor exit state. Make a plot of $P_1$ vs. $T_1$.

This problem has some relevance. Often the inlet temperature of a turbine is fixed due to material limitations. The exit pressure will also be fixed by the condenser conditions, and the exit should also be a saturated vapor to prevent liquid damage of the turbine blades. The unknown variable is the inlet pressure – which can be controlled by throttling the steam from the boiler.

This problem is trivial to solve if the turbine is isentropic: $s_1$ will be equal to $s_2$, and $s_1$, $T_1$ will fix $P_1$.

The problem is a good deal more complicated if the turbine is non–ideal, as is the case here. The isentropic efficiency of the turbine is defined by

$$\eta_s = \frac{w_a}{w_s} = \frac{h_1 - h_{2a}}{h_1 - h_{2s}} \tag{1}$$

in which state $2s$ represents the state with $s_{2a} = s_1$ and $P_{2s} = P_2 = 50$ kPa. The actual exit enthalpy is fixed (sat. vapor at 50 kPa), so $h_{2a} = 2646$ kJ/kg. If Eq. (1) is rearranged, and the independent variables noted, we get

$$h_1(T_1, P_1) = \frac{1}{1 - \eta_s} (h_{2a} - \eta_s h_{2a}(P_2, s_{2s} = s_1)) \tag{2}$$

This is one equation for two unknowns: $P_1$ and $s_1$. A second equation is simply

$$s_1 = s_1(T_1, P_1) \tag{3}$$

i.e., $s_1$ is a function of the two properties $T_1$ and $P_1$. Combining the equations together, we get a single nonlinear equation for $P_1$; that is, we have something like

$$F(P_1) = 0$$

What makes this problem especially difficult is that we do not have a simple formula for the function $F$: it is represented either in the water tables or via a black–box code. Solving this problem will require some sort of iteration strategy.

The basic approach to an iteration method is to first make a guess of the variable ($P_1$, in this case) and to work out the problem to see if the guess satisfies the equation. Most likely the guess will not, and a new guess is made. A key element in the procedure is how to refine the guess so that you get closer to the sought solution; this is especially relevant if you are using tables to work out the problem (i.e., calculation is "expensive").

To illustrate such a process, refer to the $T–s$ diagram in Fig. (2). All possible inlet conditions are on the horizontal line corresponding to $T_1 = 500^\circ$C, and the actual exit condition ($2a$) is a saturated vapor at $P_2 = 50$ kPa. Make a guess of the inlet pressure: say $P_1 = 5$ MPa (we don’t know if this is too high or too low). Using this guess, calculate the actual exit state:

$$h_{1'} = h(500^\circ$C, 5 MPa) = 3434 kJ/kg

$s_{1'} = s(500^\circ$C, 5 MPa) = 6.976 kJ/kg/kg

$h_{2a'} = h(50$ kPa, $s_{1'}) = 2427$ kJ/kg

$w_{a'} = h_{1'} - h_{2a'} = 1007$ kJ/kg

$w_{a'} = \eta_s w_{a'} = 805.6$ kJ/kg

$h_{2a'} = h_{1'} - w_{a'} = 2628$ kJ/kg

$s_{2a'} = s(50$ kPa, $h_{2a'}) = 7.543$ kJ/kg/kg

The actual exit enthalpy must be $s_{2a} = s(50$ kPa) = 7.594 kJ/kg/K, and (referring to Fig. (2)) the difference between actual and guess is $\Delta s = 0.051$ kJ/kg/K. We can **assume** that this difference is about the same for state 1, so

$$s_1 = s_{1'} + \Delta s = 6.976 + 0.051 = 7.027 \text{ kJ/kg}$

$$P_1 = P(500^\circ$C, $s = s_1) = 4.526 \text{ MPa}$$

---

**Non–ideal turbine with an unknown inlet state**

![Fig. 1: Non ideal turbine problem.](image1)

![FIG. 1: Non ideal turbine problem.](image1)

![FIG. 2: Iteration strategy.](image2)

![FIG. 2: Iteration strategy.](image2)
The process can then be repeated until $\Delta s$ gets adequately small (or, alternatively, until we calculate an exit quality that is adequately close to 1).

If we have an inexpensive method of calculating property values (such as the Matlab code), we can apply a simple, brute-force approach. Consider, for example, the following algorithm: Start with $P_1 = 100$ kPa (which is likely to be way too low), and do the following sequence of calculations:

1. Calculate the isentropic exit state for the given $P_1$:
   
   \begin{align*}
   h_1 &= h_1(T_1, P_1), \quad s_1 = s_1(T_1, P_1) \\
   s_{2s} &= s_1, \quad P_{2s} = P_2 = 50 \text{ kPa} \\
   h_{2s} &= h_{2s}(s_{2s}, P_2) \\
   w_s &= h_1 - h_{2s}
   \end{align*}

2. Now calculate the actual exit state for the guessed $P_1$:
   
   \begin{align*}
   w_{a'} &= \eta_T w_s \quad (4) \\
   h_{2a'} &= h_1 - w_{a'} \quad (5)
   \end{align*}

3. Calculate the difference between the estimated and the actual exit enthalpy:
   \[ \Delta h = h_{2a'} - h_{2a} = h_{2a'} - 2646 \text{ kJ/kg} \]

4. Increment $P_1$ by a small value (i.e., $P_1 = P_1 + 1 \text{ kPa}$), and repeat the calculation.

5. Stop when the enthalpy difference $\Delta h$ changes sign from the previous calculation. The current pressure is the sought pressure.

In the calculation sequence, we will get to the point where the enthalpy difference goes from being slightly positive to slightly negative. At this point we have bracketed the root, i.e., the previous $P_1$ was slightly lower than the correct value, whereas the current $P_1$ is slightly higher. If our increment in $P_1$ is sufficiently small, then we can simply take one of the bracket points as our solution. If we wanted to get more accuracy, we could linearly interpolate between the two bracket points using the $\Delta h$ values, with $\Delta h = 0$ as the sought condition.

**MatLab Code**

A code is listed below which implements the accelerated iteration strategy. For a given $T_1$ and a guessed value of $P_1 = P_1'$, the code calculates the actual exit state and the difference in entropy $\Delta s = s_2 - s_2'$. The guess is then revised using $P_1'' = P(T_1, s_{new})$, where $s_{new} = s_{1', old} + \Delta s$.

The code is written in MatLab and uses the XSteam.m package which is available on the course website. Make sure you place this file in your MatLab working directory. Note also that the units for pressure in the functions are bars, with 1 bar = 100 kPa.

A problem in implementing the iteration method is that XSteam does not have a function which calculates pressure as a function of temperature and entropy (it should, but that is beyond the point). It does, however, have functions which calculate pressure as a function of enthalpy and entropy, and enthalpy as a function of temperature and pressure. These two can be iterated together to get $P$ as a function of $T$ and $s$: given $T_1$ and $s_{new}$, repeat the following calculation pair until the pressure converges on a constant:

\[ h_1 = h(T_1, P_1') \quad P_1'' = P(h_1, s_{new}) \]

The code listing is given below. There are two (nested) iteration loops: the outer loop calculates $P_1$ using the $\Delta s$ approach, and the inner solves for $P(T, s)$ per the procedure discussed above. I calculate convergence in both loops by use of the while test block: this command repeats a series of commands (up to an end statement) as long as the test is true. Note that I include a loop counter in each test to prevent a runaway loop.

This is the function I define to perform the calculation for a given $T_1$ and $P_2$:

```matlab
function[p1,wa]= turbine(t1,p2)
eta=0.8;
x2a=1.;
h2a=XSteam('hV_p',p2);
s2a=XSteam('sV_p',p2);
t2a=XSteam('Tsat_p',p2);
p1g=10;
```

**FIG. 3: Turbine inlet pressure vs. inlet temperature.**
iter=0;
err=1.;
while (err>.001 & iter<100);
    iter=iter+1;
    h1g=XSteam('h_pT',p1g,t1);
    s1g=XSteam('s_pT',p1g,t1);
    h2s=XSteam('h_ps',p2,s1g);
    ws=h1g-h2s;
    wa=eta*ws;
    h2ag=h1g-wa;
    s2ag=XSteam('s_ph',p2,h2ag);
    dels=s2a-s2ag;
    err=abs(dels);
    s1g=s1g+dels;
    i=0;
    perr=1.;
    while (i<100 & perr>.001);
        i=i+1;
        pold=p1g;
        h1g=XSteam('h_pT',p1g,t1);
        p1g=XSteam('p_hs',h1g,s1g);
        perr=abs(p1g-pold);
    end
end
p1=p1g;

Note that it returns $P_1$ and $w$.

From the main MATLAB window, I calculated plotting points via,

>> t1=400:700;
>> npts=size(t1)
>> for i=1:npts;
    >> [p1(i),wa(i)]=turbeff(t1(i),.5);
end
>> plot(t1,p1)