# Homework 6-1

## Imports

[1]:
```python
from thermostate import Q_, State, units
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

## Definitions

[2]:
```python
substance = 'air'

p_1 = Q_(1.0, 'bar')
T_1 = Q_(300.0, 'K')

# pressure ratio: p_2 / p_1
p2_p1 = Q_(8.0, 'dimensionless')

T_3 = Q_(1700.0, 'K')

p_lo = 2.0   # (bar)
p_hi = 50.0  # (bar)
```

## Problem Statement

An ideal air-standard Brayton cycle operates at steady state with compressor inlet conditions of 300 K and 1 bar, a fixed turbine inlet temperature of 1700 K, and a compressor pressure ratio of 8. For the cycle,

1. determine the net work developed per unit mass flowing, in kJ/kg,
2. determine the thermal efficiency,
3. plot the net work developed per unit mass flowing, in kJ/kg, as a function of the compressor pressure ratio from 2.0 to 50.0,
4. on the same graph, plot the thermal efficiency as a function of the compressor pressure ratio from 2.0 to 50.0,
5. discuss any trends you find in Parts 3 and 4.

## Solution

Fixing States:

| State | 1 | 2 |
|---|---|---|
| 1 | $p_1$ | $T_1$ |
| 2 | $p_2$ | $s_2 = s_1$ |
| 3 | $p_3 = p_2$ | $T_3$ |

| State | 1 | 2 |
|---|---|---|
| 4 | $p_4 = p_1$ | $s_4 = s_3$ |

[3]:
```python
st_1 = State(substance, p=p_1, T=T_1)
st_2 = State(substance, p=p2_p1*p_1, s=st_1.s)
st_3 = State(substance, p=st_2.p, T=T_3)
st_4 = State(substance, p=p_1, s=st_3.s)
```

### Part 1: Net Work Developed per Unit Mass Flow Rate, in kJ/kg

Important Equations:

1. $\frac{\dot{W}_{net}}{\dot{m}} = \frac{\dot{W}_t}{\dot{m}} + \frac{\dot{W}_c}{\dot{m}}$
2. $\frac{\dot{W}_t}{\dot{m}} = h_3 - h_4$
3. $\frac{\dot{W}_c}{\dot{m}} = h_1 - h_2$

[4]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
wdot_t = st_3.h-st_4.h
wdot_c = st_1.h-st_2.h
w_net = wdot_t + wdot_c
print(w_net.to("kJ/kg"))
```

```
556.5732042751549 kilojoule / kilogram
```

**Answer:** 556.57 kJ/kg

### Part 2: Thermal Efficiency

Important Equations:

1. $\eta = \frac{\dot{W}_{net}/\dot{m}}{\dot{Q}_{in}/\dot{m}}$
2. $\frac{\dot{Q}_{in}}{\dot{m}} = h_3 - h_2$

[5]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
qdot_in = st_3.h - st_2.h
eta = w_net/qdot_in
print(eta)
```

```
0.4164598992529169 dimensionless
```

**Answer:**

**Parts 3 & 4: Plot Net Work and Thermal Efficiency vs. Compressor Pressure Ratio**

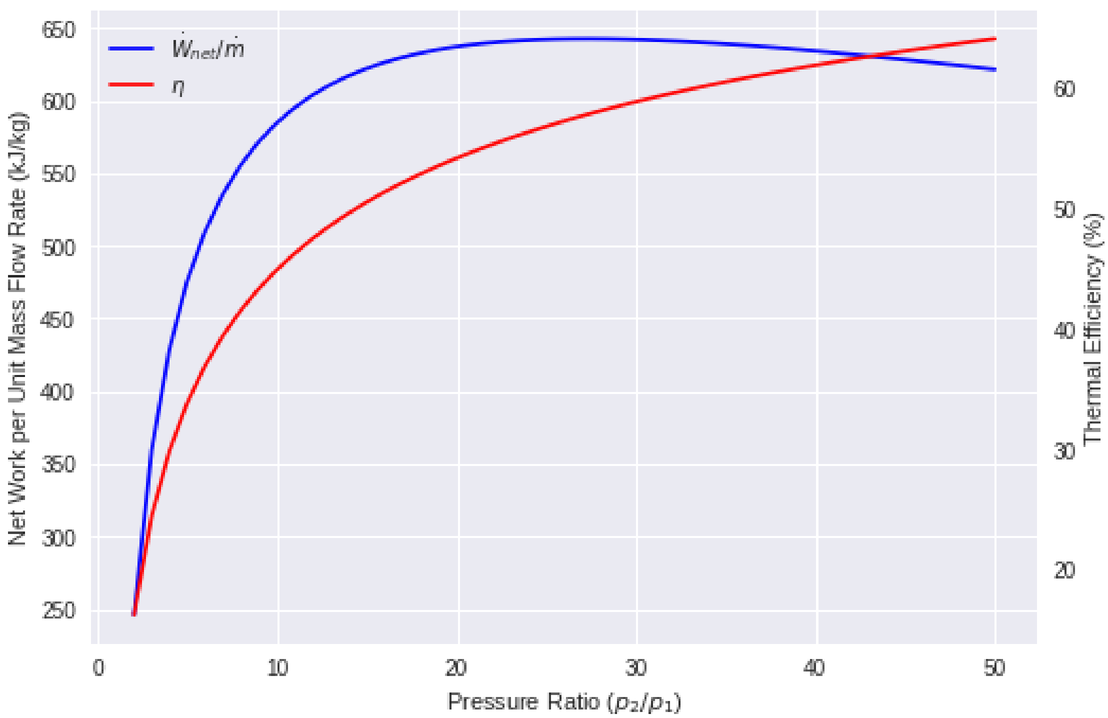
Copy the skeleton code below and fill in the function with the code you wrote in the parts above.

```
def calc_net_work_and_eta(p2_p1):
    '''Calculate the net work of the ideal Brayton cycle
    per unit mass flow rate, as well as the thermal efficiency,
    given the compressor pressure ratio.'''

    # TODO: fill this in, and calculate w_net and eta
    # Be sure to use the same variable names as in the
    # return line below!

    w_net = w_net.to('kJ/kg')
    eta = eta.to('percent')
    return w_net, eta



p2_p1_vals = np.linspace(p_lo, p_hi)
w_net_vals = np.zeros_like(p2_p1_vals)*units('kJ/kg')
eta_vals = np.zeros_like(p2_p1_vals)*units('percent')

for i, p2_p1 in enumerate(p2_p1_vals):
    w_net_vals[i], eta_vals[i] = calc_net_work_and_eta(p2_p1)

plt.style.use('seaborn')
fig, work_ax = plt.subplots()
eta_ax = work_ax.twinx()
work_ax.plot(p2_p1_vals, w_net_vals, label='$\dot{W}_{net}/\dot{m}$', color='blue')
eta_ax.plot(p2_p1_vals, eta_vals, label='$\eta$', color='red')
work_ax.set_xlabel('Pressure Ratio ($p_2/p_1$)')
work_ax.set_ylabel('Net Work per Unit Mass Flow Rate (kJ/kg)')
eta_ax.set_ylabel('Thermal Efficiency (%)')
lines, labels = work_ax.get_legend_handles_labels()
lines2, labels2 = eta_ax.get_legend_handles_labels()
work_ax.legend(lines + lines2, labels + labels2, loc='best');
plt.grid()
```

**Write your engineering model, equations, and/or explanation of your process here.**

[6]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
def calc_net_work_and_eta(p2_p1):
    '''Calculate the net work of the ideal Brayton cycle
    per unit mass flow rate, as well as the thermal efficiency,
    given the compressor pressure ratio.'''

    st_1 = State(substance, p=p_1, T=T_1)
```

```
    st_2 = State(substance, p=p2_p1*p_1, s=st_1.s)
    st_3 = State(substance, p=st_2.p, T=T_3)
    st_4 = State(substance, p=p_1, s=st_3.s)

    wdot_t = st_3.h-st_4.h
    wdot_c = st_1.h-st_2.h
    w_net = wdot_t + wdot_c

    qdot_in = st_3.h - st_2.h
    eta = w_net/qdot_in

    w_net = w_net.to('kJ/kg')
    eta = eta.to('percent')
    return w_net, eta



p2_p1_vals = np.linspace(p_lo, p_hi)
w_net_vals = np.zeros_like(p2_p1_vals)*units('kJ/kg')
eta_vals = np.zeros_like(p2_p1_vals)*units('percent')

for i, p2_p1 in enumerate(p2_p1_vals):
    w_net_vals[i], eta_vals[i] = calc_net_work_and_eta(p2_p1)

plt.style.use('seaborn')
fig, work_ax = plt.subplots()
eta_ax = work_ax.twinx()
work_ax.plot(p2_p1_vals, w_net_vals, label='$\dot{W}_{net}/\dot{m}$', color='blue')
eta_ax.plot(p2_p1_vals, eta_vals, label='$\eta$', color='red')
work_ax.set_xlabel('Pressure Ratio ($p_2/p_1$)')
work_ax.set_ylabel('Net Work per Unit Mass Flow Rate (kJ/kg)')
eta_ax.set_ylabel('Thermal Efficiency (%)')
lines, labels = work_ax.get_legend_handles_labels()
lines2, labels2 = eta_ax.get_legend_handles_labels()
work_ax.legend(lines + lines2, labels + labels2, loc='best');
plt.grid()
```




**Answer:**

**Part 5: Discuss Trends**

**Write your engineering model, equations, and/or explanation of your process here.**

[7]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
```

**Answer:** We see here that there are two "best" backwork scenario. One where net work is maximized and one where thermal efficency is maximized. And increase in the pressure reation will always increase thermal efficency but after a certain point, increasing the backwork ratio would actually decrease the net power output.

# Homework 6-2

## Imports

[1]:
```
from thermostate import Q_, State, units
import numpy as np
import matplotlib.pyplot as plt
import warnings

# tidies up plots
%matplotlib inline
```

## Definitions

[2]:
```
substance = 'air'
mdot = Q_(6.0, 'kg/s')

p_1 = Q_(1.0, 'bar')
T_1 = Q_(300.0, 'K')

p2_p1 = Q_(10.0, 'dimensionless')

T_3 = Q_(1400.0, 'K')

T_3_lo = 1000.0  # (K)
T_3_hi = 1800.0  # (K)
```

## Problem Statement

An ideal **cold** air-standard Brayton cycle operates at steady state with compressor inlet conditions of 300 K and 1 bar, a fixed turbine inlet temperature of 1400 K, and a compressor pressure ratio of 10. The mass flow rate of the air is 6 kg/s. For the cycle,

1. determine the back work ratio (BWR),
2. determine the net power output, in kW,
3. determine the thermal efficiency,
4. plot the net power output, in kW, and the thermal efficiency as a function of the turbine inlet temperature from 1000 K to 1800 K,
5. discuss any trends you find in Part 4.

**Hint**: In a cold air-standard analysis, the specific heats are constant and evaluated at the ambient temperature of 300 K. You can retrieve the specific heats $c_p$ and $c_v$ from a `State` with the `cp` and `cv` attributes. Since air is an ideal gas, properties like enthalpy, internal energy, and the specific heats depend only on the temperature. This means that you can use *any pressure you want* to fix the ambient state, and find $c_p$ and $c_v$.

```python
st_amb = State(substance, T=Q_(300.0, "K"), p=Q_(1.0, "atm"))
c_v = st_amb.cv
c_p = st_amb.cp
```

This is the **only** `State` function you should use in the whole problem.

## Solution

### Part 1: Back Work Ratio (BWR)

Important Equations:

1. $\Delta h = c_p \Delta(T)$
2. $k = \frac{c_p}{c_v}$
3. $(h_2 - h_1) = T_1 c_p((\frac{p_2}{p_1})^{\frac{k-1}{k}} - 1)$
4. $\dot{W}_t = \dot{m}(h_3 - h_4)$
5. $\dot{W}_p = \dot{m}(h_1 - h_2)$
6. $BWR = \frac{|\dot{W}_p|}{|\dot{W}_t|}$

[3]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
p4_p3 = 1/p2_p1

st_amb = State(substance, T=Q_(300.0, "K"), p=Q_(1.0, "atm"))
c_v = st_amb.cv
c_p = st_amb.cp
k = c_p/c_v
k_p = (k-1)/k

h2_h1 = T_1*c_p*(p2_p1**k_p - 1)
h4_h3 = T_3*c_p*(p4_p3**k_p - 1)

Wdot_t = mdot*(-h4_h3)
Wdot_p = mdot*(-h2_h1)
BWR = -Wdot_p/Wdot_t
print(BWR)
```

```
0.4145424023759423 dimensionless
```

**Answer:** 41.45%

### Part 2: Net Power Output

Important Equations:

1. $\dot{W}_{net} = \dot{W}_t + \dot{W}_p$

[9]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
Wdot_net = Wdot_t + Wdot_p
print(Wdot_net.to("kW"))
```

```
2390.8491214985074 kilowatt
```

**Answer:** 2390.85 kW

#### Part 3: Thermal Efficiency

Important Equations:

1. $\eta = 1 - \frac{1}{(p_2/p_1)^{(\frac{k-1}{k})}}$

**Write your engineering model, equations, and/or explanation of your process here.**

[5]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
eta = 1 - 1/(p2_p1**k_p)
print(eta)
```

```
0.48307890083730987 dimensionless
```

**Answer:** 48.31%

#### Part 4: Plot Net Power Output and Thermal Efficiency vs. Compressor Pressure Ratio

Copy the skeleton code below and fill in the function with the code you wrote in the parts above.

```python
def calc_net_power_and_eta(T_3):

    # TODO: fill this in, and calculate w_net and eta
    # Be sure to use the same variable names as in the
    # return line below!

    Wdot_net = Wdot_net.to('kW')
    eta = eta.to('percent')
    return Wdot_net, eta


T_3_vals = np.linspace(T_3_lo, T_3_hi)*units.K
Wdot_net_vals = np.zeros_like(T_3_vals)*units('kW')
eta_vals = np.zeros_like(T_3_vals)*units('percent')

for i, T_3 in enumerate(T_3_vals):
    Wdot_net_vals[i], eta_vals[i] = calc_net_power_and_eta(T_3)
```

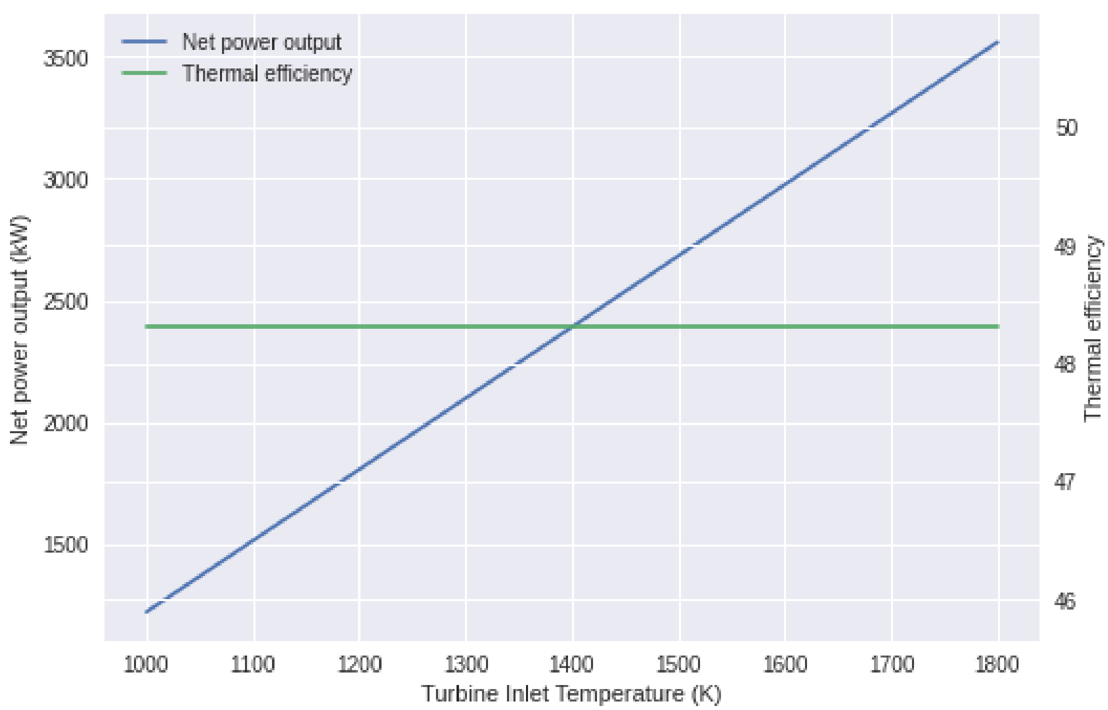
```
plt.style.use('seaborn')
fig, power_ax = plt.subplots()
power_ax.plot(T_3_vals, Wdot_net_vals, label='Net power output', color='C0')
eta_ax = power_ax.twinx()
eta_ax.plot(T_3_vals, eta_vals, label='Thermal efficiency', color='C1')
power_ax.set_xlabel('Turbine Inlet Temperature (K)')
power_ax.set_ylabel('Net power output (kW)')
eta_ax.set_ylabel('Thermal efficiency')
lines, labels = power_ax.get_legend_handles_labels()
lines2, labels2 = eta_ax.get_legend_handles_labels()
power_ax.legend(lines + lines2, labels + labels2, loc='best');
```

**Write your engineering model, equations, and/or explanation of your process here.**

[6]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
st_amb = State(substance, T=Q_(300.0, "K"), p=Q_(1.0, "atm"))
c_v = st_amb.cv
c_p = st_amb.cp
k = c_p/c_v
k_p = (k-1)/k

def calc_net_power_and_eta(T_3):



    h2_h1 = T_1*c_p*(p2_p1**k_p - 1)
    h4_h3 = T_3*c_p*(p4_p3**k_p - 1)

    Wdot_t = mdot*(-h4_h3)
    Wdot_p = mdot*(-h2_h1)



    Wdot_net = Wdot_t + Wdot_p


    eta = 1 - 1/(p2_p1**k_p)


    Wdot_net = Wdot_net.to('kW')
    eta = eta.to('percent')
    return Wdot_net, eta


T_3_vals = np.linspace(T_3_lo, T_3_hi)*units.K
Wdot_net_vals = np.zeros_like(T_3_vals)*units('kW')
eta_vals = np.zeros_like(T_3_vals)*units('percent')
```

```
for i, T_3 in enumerate(T_3_vals):
    Wdot_net_vals[i], eta_vals[i] = calc_net_power_and_eta(T_3)

plt.style.use('seaborn')
fig, power_ax = plt.subplots()
power_ax.plot(T_3_vals, Wdot_net_vals, label='Net power output', color='C0')
eta_ax = power_ax.twinx()
eta_ax.plot(T_3_vals, eta_vals, label='Thermal efficiency', color='C1')
power_ax.set_xlabel('Turbine Inlet Temperature (K)')
power_ax.set_ylabel('Net power output (kW)')
eta_ax.set_ylabel('Thermal efficiency')
lines, labels = power_ax.get_legend_handles_labels()
lines2, labels2 = eta_ax.get_legend_handles_labels()
power_ax.legend(lines + lines2, labels + labels2, loc='best');
```

```
/opt/conda/lib/python3.8/site-packages/pint/quantity.py:1377:
UnitStrippedWarning: The unit of the quantity is stripped.
  warnings.warn("The unit of the quantity is stripped.", UnitStrippedWarning)
```




**Answer:**

**Part 5: Discuss Trends**

**Write your engineering model, equations, and/or explanation of your process here.**

[7]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
```

**Answer:** This graph proves that thermal efficiency is only a function of the compresser pressure ratio and thermal efficency does not change for an increasing turbine temperature. What does increase is the net power. As mechanical engineers our goal should be just to increase our turbine inlet temperature as much as possible, then let a material science engineer solve the materials issue. Increasing the temperature will have no down sides on the systems efficiency.

# Homework 6-3

## Imports

[1]:
```
from thermostate import Q_, State, units
```

## Definitions

[2]:
```
substance = 'air'

Vdot_1 = Q_(60.0, 'm**3/s')

p_1 = Q_(0.8, 'bar')
T_1 = Q_(280.0, 'K')

p2_p1 = Q_(20.0, 'dimensionless')

T_3 = Q_(2100.0, 'K')

eta_t = Q_(0.92, 'dimensionless')
eta_c = Q_(0.95, 'dimensionless')
```

## Problem Statement

An air-standard Brayton cycle operates at steady state with compressor inlet conditions of 280 K and 0.8 bar, a fixed turbine inlet temperature of 2100 K, and a compressor pressure ratio of 20. The volumetric flow rate of the air at the inlet to the compressor is 60 $m^3/s$, and the isentropic efficiencies of the turbine and compressor are 92% and 95%, respectively.

1. Determine the net power output, in MW.
2. Determine the rate of heat addition in the combustor, in MW.
3. Determine the thermal efficiency.

## Solution

Defining States:

| | State | 1 | 2 |
|---|---|---|---|
| 1 | $p_1$ | | $T_1$ |
| 2s | $p_2$ | | $s_2 = s_1$ |
| 2 | $p_2 = p_{2s}$ | | $h_2$ |
| 3 | $p_3 = p_2$ | | $T_3$ |
| 4s | $p_4 = p_1$ | | $s_4 = s_3$ |
| 4 | $p_4 = p_{4s}$ | | $h_4$ |

[13]:
```python
st_1 = State(substance, p=p_1, T=T_1)
st_2s = State(substance, p=p2_p1*p_1, s=st_1.s)
h_2 = (st_2s.h-st_1.h)/eta_c + st_1.h
st_2 = State(substance, p=st_2s.p, h=h_2)
st_3 = State(substance, p=st_2.p, T=T_3)
st_4s = State(substance, p=p_1, s=st_3.s)
h_4 = eta_t*(st_4s.h-st_3.h)+st_3.h
st_4 = State(substance, p=st_4s.p, h=h_4)
```

Calculating $\dot{m}$:

$\dot{m} = \frac{\dot{V}_1}{v_1}$

[4]:
```python
mdot = Vdot_1/st_1.v
print(1/st_1.v)
```

```
0.9957568993686244 kilogram / meter ** 3
```

### Part 1: Determine the Net Power Output, in MW

Important Equations:

1. $\dot{W}_{net} = \dot{W}_t + \dot{W}_c$
2. $\dot{W}_t = \dot{m}(h_3 - h_4)$
3. $\dot{W}_c = \dot{m}(h_1 - h_2)$

[10]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
Wdot_t = mdot*(st_3.h-st_4.h)
Wdot_p = mdot*(st_1.h-st_2.h)
Wdot_net = Wdot_t + Wdot_p
print(Wdot_net.to("MW"))
```

```
47.53421520708249 megawatt
```

**Answer:** 47.53 MW

### Part 2: Rate of Heat Addition, in MW

Important Equations:

1. $\dot{Q}_{in} = \dot{m}(h_3 - h_2)$

[11]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
Qdot_in = mdot*(st_3.h-st_2.h)
print(Qdot_in.to("MW"))
```

```
101.51374576751284 megawatt
```

**Answer:** 101.51 MW

**Part 3: Thermal Efficiency**

Important Equations:

1. $\eta = \frac{\dot{W}_{net}}{\dot{Q}_{in}}$

[12]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
eta = Wdot_net/Qdot_in
print(eta)
```

```
0.46825397730811286 dimensionless
```

**Answer:** 46.83%

# Homework 6-4

## Imports

[1]:
```python
from thermostate import Q_, State, units
import numpy as np
```

## Definitions

[2]:
```python
substance = 'air'

Vdot_1 = Q_(10000.0, 'ft**3/min')

p_1 = Q_(14.0, 'psi')
T_1 = Q_(520.0, 'degR')

p2_p1 = Q_(14.0, 'dimensionless')

T_3 = Q_(2500.0, 'degR')

eta_c = Q_(0.83, 'dimensionless')
eta_t = Q_(0.87, 'dimensionless')
```

## Problem Statement

Air enters the compressor of a simple gas turbine at 14 psi, 520 °R, and a volumetric flow rate of 10,000 $\text{ft}^3$/min. The isentropic efficiencies of the compressor and turbine are 83% and 87%, respectively. The compressor pressure ratio is 14, and the temperature at the turbine inlet is 2500 °R. On the basis of an air-standard analysis, calculate

1. the thermal efficiency of the cycle,
2. the net power developed, in hp,
3. the rates at which entropy is produced within the compressor and turbine, each in hp/°R.

## Solution

Defining States:

| State | 1 | 2 |
|---|---|---|
| 1 | $p_1$ | $T_1$ |
| 2s | $p_2$ | $s_2 = s_1$ |
| 2 | $p_2 = p_{2s}$ | $h_2$ |
| 3 | $p_3 = p_2$ | $T_3$ |
| 4s | $p_4 = p_1$ | $s_4 = s_3$ |
| 4 | $p_4 = p_{4s}$ | $h_4$ |

[3]:
```
st_1 = State(substance, p=p_1, T=T_1)
st_2s = State(substance, p=p2_p1*p_1, s=st_1.s)
h_2 = (st_2s.h-st_1.h)/eta_c + st_1.h
st_2 = State(substance, p=st_2s.p, h=h_2)
st_3 = State(substance, p=st_2.p, T=T_3)
st_4s = State(substance, p=p_1, s=st_3.s)
h_4 = eta_t*(st_4s.h-st_3.h)+st_3.h
st_4 = State(substance, p=st_4s.p, h=h_4)
```

Calculating $\dot{m}$:

$\dot{m} = \frac{\dot{V}_1}{v_1}$

[4]:
```
mdot = Vdot_1/st_1.v
print(1/st_1.v)
```

```
1.1644837461580195 kilogram / meter ** 3
```

### Part 1: Thermal Efficiency

Important Equations:

1. $\eta = \frac{(h_3 - h_4) + (h_1 - h_2)}{(h_3 - h_2)}$

[5]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
eta = (st_3.h-st_4.h+(st_1.h-st_2.h))/(st_3.h-st_2.h)
print(eta)
```

```
0.3416505066518584 dimensionless
```

**Answer:** 34.17%

### Part 2: Net Power Developed, in hp

Important Equations:

1. $\dot{W}_{net} = \dot{W}_t + \dot{W}_c$
2. $\dot{W}_t = \dot{m}(h_3 - h_4)$
3. $\dot{W}_c = \dot{m}(h_1 - h_2)$

[6]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
Wdot_t = mdot*(st_3.h-st_4.h)
Wdot_p = mdot*(st_1.h-st_2.h)
Wdot_net = Wdot_t + Wdot_p
```

```python
print(Wdot_net.to("hp"))
```

```
2068.1336378802785 horsepower
```

**Answer:** 2068.13 hp

### Part 3: Rate of Entropy Production, in hp/°R

Important equations:

1. $\dot{\sigma}_{compressor} = \dot{m}(s_2 - s_1)$
2. $\dot{\sigma}_{turbine} = \dot{m}(s_4 - s_3)$

[8]:
```python
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
sigma_c = mdot*(st_2.s-st_1.s)
sigma_t = mdot*(st_4.s-st_3.s)
print(sigma_c.to("hp/degR"))
print(sigma_t.to("hp/degR"))
```

```
0.4283668649753249 horsepower / degR
0.5424026970153698 horsepower / degR
```

**Answer:** 0.428 hp/$^o$R produced in the compressor
0.542 hp/$^o$R produced in the turbine

# Homework 6-5

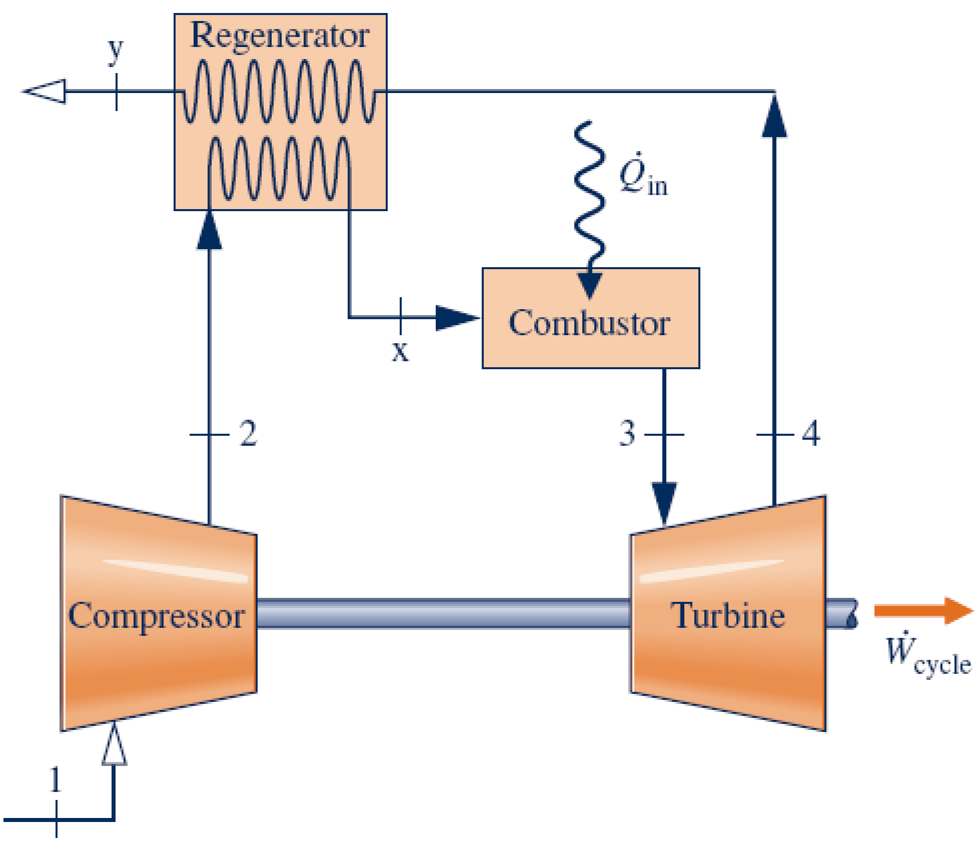
## Problem Statement

On the basis of a cold air-standard analysis, show that the thermal efficiency of an ideal regenerative gas turbine can be expressed alternatively as

1. $\eta = 1 - \left(\frac{T_1}{T_3}\right)(r)^{(k-1)/k}$ where $r$ is the compressor pressure ratio, $T_1$ and $T_3$ denote the temperatures at the compressor and turbine inlets, respectively.
2. $\eta = 1 - \frac{T_2}{T_3}$ where $T_2$ is the temperature at the compressor exit.

Part 1 Hints:

- Start by defining the thermal efficiency of the cycle in terms of specific enthalpy values.
- Make substitutions using the cold-air relation $h_i - h_f = c_p(T_i - T_f)$.
- Which temperature values are the same if $\eta_{reg} = 1.0$? Make a substitution.
- Algebraically modify your equation and Eqs. (9.23) and (9.24) from the textbook, and make two substitutions.
- Simplify the resulting equation.



## Solution

### Part 1: Derive the first given alternate form of thermal efficiency.

Starting off with the general idea of thermal efficiency, we know $\eta$ is what we want / what we put in. Therefore:

$$\eta = \frac{\dot{W}_{net}}{\dot{Q}_{in}}$$

Rearranging this with the energy balences of each individual systems yeilds:

$$\eta = \frac{(h_1 - h_2) + (h_3 - h_4)}{h_3 - h_x}$$

There is a pesky $h_x$ here but luckily we can get rid of it because we know the process is ideal therefore:

$$\eta_{reg} = \frac{h_x - h_2}{h_4 - h_2} = 1$$

For this to be true then $h_x$ must be $h_4$ therefore:

$$\eta = \frac{(h_1 - h_2) + (h_3 - h_4)}{h_3 - h_4}$$

We also know that for cold air, all of these states have the same $c_p$ so we can substitute the following equation in:

$$h_b - h_a = c_p(T_b - T_a)$$

And subsituting that in to our $\eta$ equation we get:

$$\eta = \frac{-c_p(T_2 - T_1) + c_p(T_3 - T_4)}{c_p(T_3 - T_4)} = 1 - \frac{T_2 - T_1}{T_3 - T_4}$$

Next we will multiply both the top by the fraction by $\frac{1}{T_1}$ and bottom of the fraction by $\frac{1}{T_3}$ to yield:

$$\eta = 1 - \frac{T_1(\frac{T_2}{T_1} - 1)}{T_3(1 - \frac{T_4}{T_3})}$$

Next we can use the following equations from the textbook:

$$\frac{T_4}{T_3} = (\frac{p_1}{p_2})^{\frac{k-1}{k}} \quad and \quad \frac{T_2}{T_1} = (\frac{p_2}{p_1})^{\frac{k-1}{k}}$$

Calling $\frac{p_2}{p_1}$ as $r$ and substituting yields:

$$\eta = 1 - \frac{T_1(r^{\frac{k-1}{k}} - 1)}{T_3(1 - (\frac{1}{r})^{\frac{k-1}{k}})} = 1 - \frac{T_1}{T_3}\frac{r^{\frac{k-1}{k}} - 1}{1 - (\frac{1}{r})^{\frac{k-1}{k}}} = 1 - \frac{T_1}{T_3}\frac{r^{\frac{k-1}{k}} - 1}{(\frac{r^{\frac{k-1}{k}}-1}{r^{\frac{k-1}{k}}})} = 1 - \frac{T_1}{T_3}\frac{r^{\frac{k-1}{k}} - 1}{r^{\frac{k-1}{k}} - 1}r^{\frac{k-1}{k}}$$

Final simplification verifies:

$$\eta = 1 - \frac{T_1}{T_3}r^{\frac{k-1}{k}}$$

**Part 2: Derive the second given alternate form of thermal efficiency.**

For this derivation, we just need to use the following equation:

$$\frac{T_2}{T_1} = (r)^{\frac{k-1}{k}}$$

And multiplying $T_1$ by both sides allows us to substitute in $T_2$ yielding:

$$\eta = 1 - \frac{T_2}{T_3}$$

# Homework 6-6

### Imports

[1]:
```
from thermostate import Q_
import matplotlib.pyplot as plt
import math
%matplotlib inline
```

### Definitions

[2]:
```
substance = 'air'   # *wink*
c_p = Q_(1.005, 'kJ/kg/K')

m_Pu238 = Q_(5, 'kg')
q_Pu238 = Q_(2.8, 'W/kg')

p_1 = Q_(100, 'kPa')
T_1 = Q_(300, 'K')

p2_p1 = Q_(8, 'dimensionless')

T_3 = Q_(1700, 'K')

k = Q_(1.4, 'dimensionless')
```

### Problem Statement

On Feb. 18, 2021, NASA's Jet Propulsion Laboratory successfully landed the Perseverance Rover in the Jezero Crater on Mars.

The Perseverance Rover is powered by a Multi-Mission Radioisotope Thermoelectric Generator (MMRTG or RTG for short). The RTG produces about 2.8 W per kg of plutonium-238 of heat due to the nuclear decay of the Pu-238 at the beginning of its life. This value decreases as the Pu-238 decays.

Charge-4-Mars is a private company founded by Melon Husk, with the goal of creating rover charging stations on the surface of Mars. These charging stations are designed to be receive heat transfer input from Pu-238 capsules, containing 5 kg of nuclear material each, which Melon infamously ~~stole~~ borrowed from Libyan terrorists in 1985.

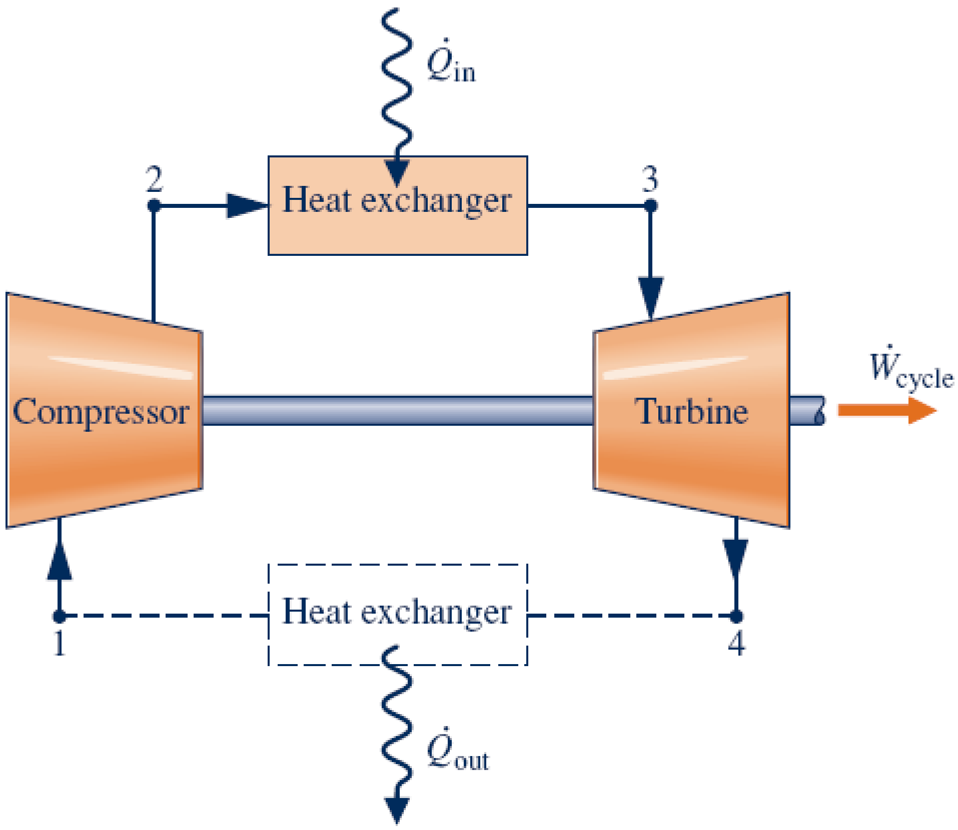
Melon has designed the charging stations to operate on an ideal cold **air**-standard Brayton cycle. The cycle operates at steady state, with compressor inlet conditions of 300 K, 100 kPa, a fixed turbine inlet temperature of 1700 K (with the initial amount of plutonium), and $k = 1.4$. The compressor pressure ratio is 8, and the compressor and turbine can be modeled as perfectly efficient.

1. Find the net work developed per unit mass flowing, in kJ/kg.
2. Find the mass flow rate of **air** through the system, in g/s (this value is *very* small due to the thin **atmosphere** on Mars).

3. Find the net power output of the cycle with the initial amount of plutonium, in W.

The following parts are for **bonus credit** on HW 6.

4. NASA wants rovers to charge at a rate of at least 5 W. After how many years of operation (rounded down) will the charging stations require the Pu-238 to be replaced?
5. Assuming cold **air**-standard analysis is perfectly valid, why might the system not operate as Melon expects on Mars?




## Solution

### Part 1: Net Work per Unit Mass Flow Rate, in kJ/kg

Important Equations:

1. $\frac{\dot{W}_{net}}{\dot{m}} = \frac{\dot{W}_t}{\dot{m}} + \frac{\dot{W}_c}{\dot{m}}$
2. $\frac{\dot{W}_t}{\dot{m}} = (h_3 - h_4)$
3. $\frac{\dot{W}_c}{\dot{m}} = (h_1 - h_2)$
4. $\Delta h = c_p \Delta(T)$
5. $k = \frac{c_p}{c_v}$

6. $(h_2 - h_1) = T_1 c_p((\frac{p_2}{p_1})^{\frac{k-1}{k}} - 1)$

[3]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
p4_p3 = 1/p2_p1
k_p = (k-1)/k

wdot_t = -T_1*c_p*(p2_p1**k_p - 1)
wdot_c = -T_3*c_p*(p4_p3**k_p - 1)
wdot_net = wdot_t + wdot_c
print(wdot_net)
```

```
520.68016339301 kilojoule / kilogram
```

**Answer:** 520.68 kJ/kg

### Part 2: Find the Mass Flow Rate of Air, in g/s

Important Equations:

1. $\eta = 1 - \frac{1}{(p_2/p_1)^{(\frac{k-1}{k})}}$

[4]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
eta = 1 - 1/(p2_p1**k_p)
Qdot_in = m_Pu238 * q_Pu238
Wdot_net = eta * Qdot_in
mdot = Wdot_net/wdot_net
print(mdot.to("g/s"))
```

```
0.012044579081745568 gram / second
```

**Answer:** 0.012 g/s

### Part 3: Initial Net Power Developed

**Write your engineering model, equations, and/or explanation of your process here.**

[5]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
print(Wdot_net)
```

```
6.271373404283313 watt
```

**Answer:** 6.27 W

### (Bonus) Part 4: How long until replacement Pu-238 is needed?

Important Equations:

1. $A = A_o(\frac{1}{2})^{\frac{t}{t_{1/2}}}$ *Note $A_o$ is 5 kg
2. Half life of Pu-238 is 87.74 years

[11]:
```
# Write your code here to solve the problem
# Make sure to write your final answer in the cell below.
A = Q_(5,"W")/q_Pu238
A_o = Q_(5,"kg")
t = math.log(A/A_o,0.5)*Q_(87.7, "years")
print(t.to("years"))
```

```
130.2719327428302 year
```

**Answer:** 130 years

### (Bonus) Part 5: Why might this not work as Melon intends on Mars?

**Answer:** 


**Answer:** Also the system will not work on Mars because the composion of Mars' atmosphere is different than the air model for Earth.