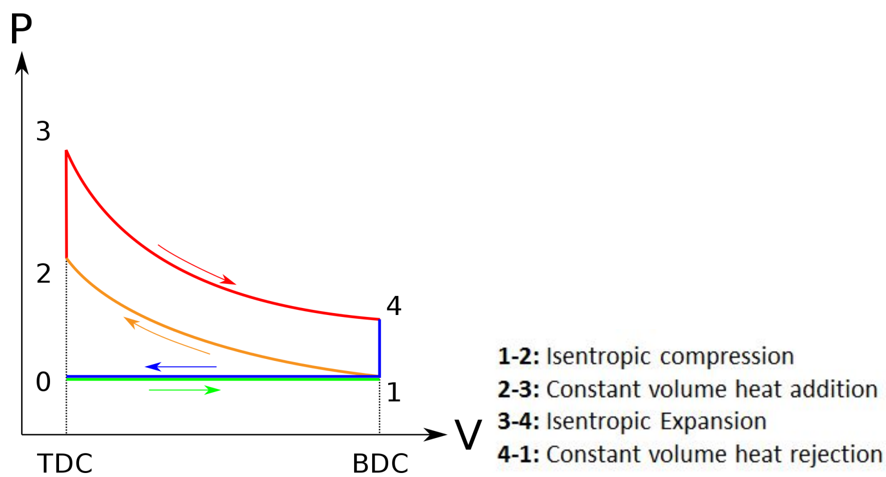
Otto Cycle using Python

Objective:

1.To write code that can solve an otto cycle and make plots for it
2. To plot a PV diagram
3. To find the thermal efficiency of the engine

Air Standard/Otto Cycle:

All the engines which are powered by petrol, follow this Air Standard cycle in order to execute the combustion process.
There are two different diagrams such as PV and TS, out of which we are interested only in PV diagram for this assignment project.



Here,TDC and BDC are Top dead centre and Bottom dead centre of the piston respectively.

Total Cylinder Volume:

It is the total volume (maximum volume) of the cylinder in which the Otto cycle takes place. In Otto cycle, Total cylinder volume = V1 = V4 = Vc + Vs

Where, Vc → Clearance Volume Vs → Stroke Volume

Clearance Volume (Vc):

At the end of the compression stroke, the piston approaches the Top Dead Center (TDC) position. The minimum volume of the space inside the cylinder, at the end of the compression stroke, is called clearance volume (Vc).

In Otto cycle, Clearance Volume, Vc = V2

Stroke Volume (Vs):

In Otto cycle, stroke volume is the difference between total cylinder volume and clearance volume.

Stroke Volume, Vs = Total Cylinder Volume – Clearance Volume = V1 – V2 = V4 – V3

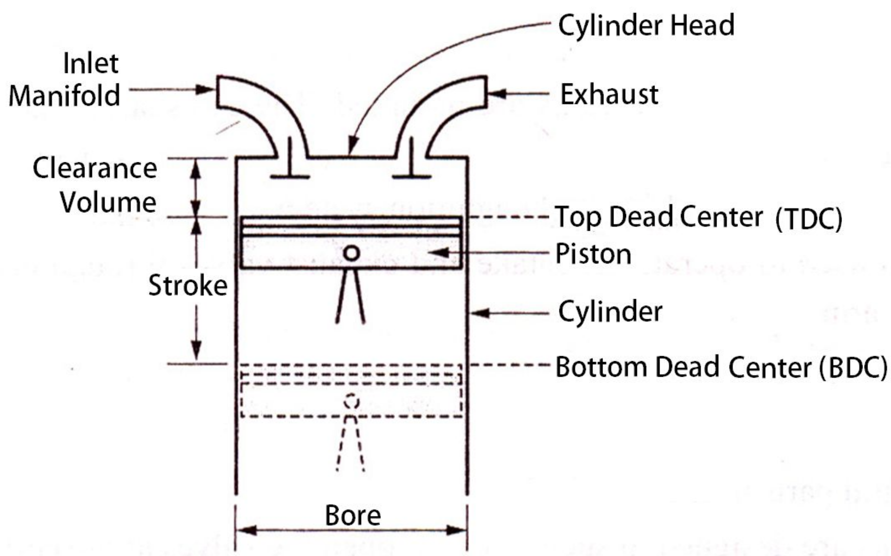
Compression Ratio:

Compression ratio (r) is the ratio of total cylinder volume to the clearance volume.

$$r = \frac{\text{Total Cylinder Volume}}{\text{Clearance Volume}}$$

$$r = \frac{V_1}{V_2} = \frac{V_4}{V_3}$$

Cylinder Nomenclature:



Piston Kinematics equation:

V/Vc=1+0.5(r-1)[R+1-cos(theta)-(R^2-(sin(theta)^2))^½

Here, 'theta' is the rotation of the crank.

'R' is the Effective radius of rotation of the crank.

Assumed Data:

Gamma=1.4 (polytropic index)

Pressure at 1 (before compression)=101325

Temperature at 1 (before compression)=500

Temperature at 3 (after constant volume heat addition)=2300

Bore of the cylinder=0.1

Stroke of the cylinder=0.1

Length of connecting rod=0.15

Compression Ratio=12

Process 1-2:

This process is an isentropic compression process.

p2=p1*(compression_ratio^gamma)

p1v1/t1=p2v2/t2

Volume varies as per Piston kinematics equation.

Process 2-3:

It is a Constant Volume Heat Addition

v3=v2

p3v3/t3=p2v2/t2

Process 3-4:

This process is an isentropic expansion process.

p4=p3*(v3/v4)^gamma

Volume varies as per Piston kinematics equation.

Process 4-1:

It is a Constant Volume Heat Rejection process.

v4=v1

Program:

```
'''Otto Cycle'''


#mathematical modules

import math

import matplotlib.pyplot as plt


#function to define the actual otto cycle engine kinematics

def engine_kinematics(bore,stroke,con_rod,cr,start_crank,end_crank):

    a=stroke/2

    R=con_rod/a


    #volume parameters: Stroke and clearance volume

    v_s= math.pi*(1/4)*pow(bore,2)*stroke

    v_c=v_s/(cr-1)


    #rotation angle of crank

    start_position=math.radians(start_crank)

    end_position=math.radians(end_crank)


    #number of divisions

    num=25

    d_theta=(end_position-start_position)/(num-1)
```

```
#Engine kinematics equation
    V=[]

    for i in range(0,num):
        theta=start_position+i*d_theta
        term1=0.5*(cr-1)
        term2=R+1-math.cos(theta)
        term3=pow(R,2)-pow(math.sin(theta),2)
        term4=pow(term3,0.5)
        V.append((1+term1*(term2-term4))*v_c)
    return V

#input parameters
p1=101325
t1=500
gamma=1.4
t3=2300
bore=0.1
stroke=0.1
con_rod=0.15
cr=12

#state points at 1
v1=v_s+v_c

#state point at 2
```

```
v2=v_c

p2=p1*pow(v1,gamma)/pow(v2,gamma)

rhs=p1*v1/t1
t2=p2*v2/rhs

v_compression= engine_kinematics(bore,stroke,con_rod,cr,0,180)
constant=p1*pow(v1,gamma)

p_compression=[]

for v in v_compression:
    p_compression.append(constant/pow(v,gamma))

#state point at 3
v3=v2

rhs=p2*v2/t2
p3=rhs*t3/v3

v_expansion= engine_kinematics(bore,stroke,con_rod,cr,0,180)
constant=p3*pow(v3,gamma)

p_expansion=[]
```

```
for v in v_expansion:

    p_expansion.append(constant/pow(v,gamma))


#state point at 4

v4=v1

p4=p3*pow(v3,gamma)/pow(v4,gamma)

t4=p4*v4/rhs


#plotting

plt.plot([v2,v3],[p2,p3])

plt.plot(v_compression,p_compression)

plt.plot(v_expansion,p_expansion)

plt.plot([v4,v1],[p4,p1])

plt.show()



'''Efficiency of Otto cycle'''

eta=1-1/pow(cr,gamma-1)

print(eta)
```

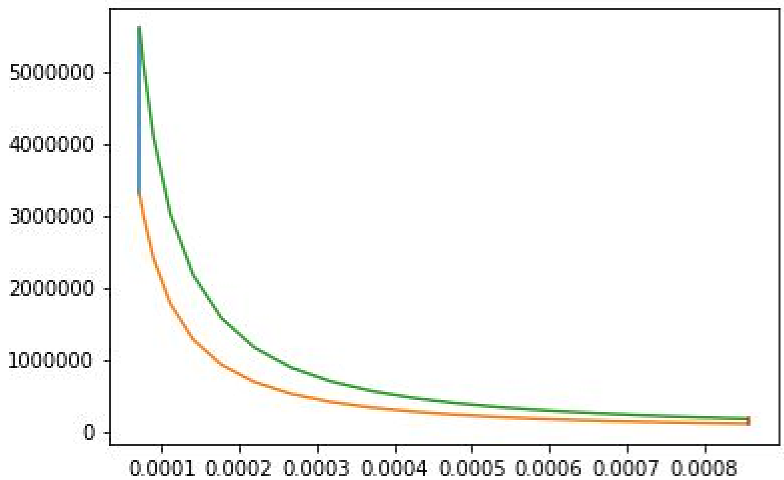
Thermal Efficiency:

$$\eta_{th} = 1 - \frac{1}{r^{(\gamma - 1)}}$$

Here, r is the compression ratio

Result:

PV Diagram:



Efficiency:

```
0.6298928275128466
```

The efficiency of the cycle is around 62.9%.