

```

import numpy as np
import matplotlib.pyplot as plt

finiteVolumeDataEQ = np.loadtxt('../results/second_task/equivalent.csv',dtype=float,delimiter=','
finiteVolumeDataLI = np.loadtxt('../results/second_task/linear.csv',dtype=float,delimiter=', '
analyticalSolutionData = np.loadtxt('../results/second_task/analytic.csv',dtype=float,delimit

x = analyticalSolutionData.transpose()[0]
y = analyticalSolutionData.transpose()[1]

plt.plot(x,y,label = "Solução analítica")
x = finiteVolumeDataEQ.transpose()[0]
y1 = finiteVolumeDataEQ.transpose()[1]
plt.plot(x,y1,'ro', label = "Volumes finitos (resistência equivalente)")
y2=finiteVolumeDataLI.transpose()[1]

plt.plot(x,y2,'k+',markersize = 12, label = "Volumes finitos (interpolação linear)")
plt.xlabel('x [m]')
plt.ylabel('T(x) [°C]')
plt.grid(b=1)
#plt.ylim([300,370])
legend = plt.legend(loc='best')
legend.get_frame().set_facecolor('#D3D3D3')
plt.savefig('../results/second_task/curve.png')

file = open("../results/second_task/table.csv","w")
file.write("nodes, analytic, equivalent, interpolation \n")
for i in range(len(x)):
    strrr = str(x[i])+", "+str(y[i])+", " + str(y1[i]) + ", " + str(y2[i])
    file.write(strrr)
    file.write("\n")
file.close()

errorEQ = abs(y1 - y)/80
errorLI = abs(y2 - y)/80

plt.clf()
plt.plot(x,errorEQ,'bo',label = "Erro (resistência equivalente)")
plt.plot(x,errorLI,'go',label = "Erro (interpolação linear)")
plt.xlabel('x [m]')
plt.ylabel('Erro')
plt.grid(b=1)
#plt.ylim([300,370])
legend = plt.legend(loc='best')
plt.axis([0,0.1,-0.001,0.003])
legend.get_frame().set_facecolor('#D3D3D3')
plt.savefig('../results/second_task/erroCurve.png')

errorFile = open("../results/second_task/error_table.csv","w")
errorFile.write("nodes, equivalentError, interpolationError \n")
for i in range(len(x)):
    errorString = str(x[i])+", " + str(errorEQ[i]) + ", " + str(errorLI[i])
    errorFile.write(errorString)
    errorFile.write("\n")
errorFile.close()

```