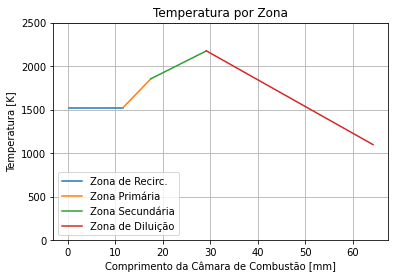
## **Lista de símbolos**

As seguintes equivalências são utilizadas neste relatório de engenharia, salvo indicação em contrário.

| °C | = Graus Celsius |
| --- | --- |
| K  m  mm | = Kelvin  = Metros  = Milímetros |

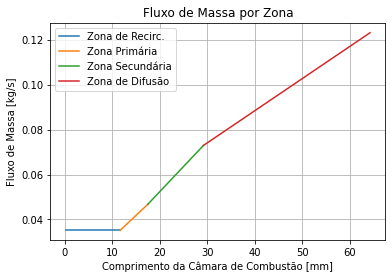
O trabalho foi iniciado com o levantamento de inputs necessários a partir do trabalho-base como, por exemplo, a temperatura e pressão na entrada da câmara de combustão. Em seguida, foram calculados os parâmetros necessários para a determinação da área de referência aerodinâmica e química, encontradas via aplicação de métodos numéricos de resolução de equações. Com isso, calculou-se as características do tubo de chama.

Em seguida, foram calculados os parâmetros de fluxo mássico na zona primária (ZP) e secundária (ZS). A partir deles, calculou-se o balanço de massa do sistema, que convergiu para zero. Com esses valores calculados, iniciou-se a implementação do cálculo da temperatura do gás ao longo da câmara de combustão, seguindo equacionamento presente na Tabela 7 do trabalho-base. Com isso, foi possível extrair o gráfico do perfil de temperatura ao longo da câmara de combustão (Figura 3).



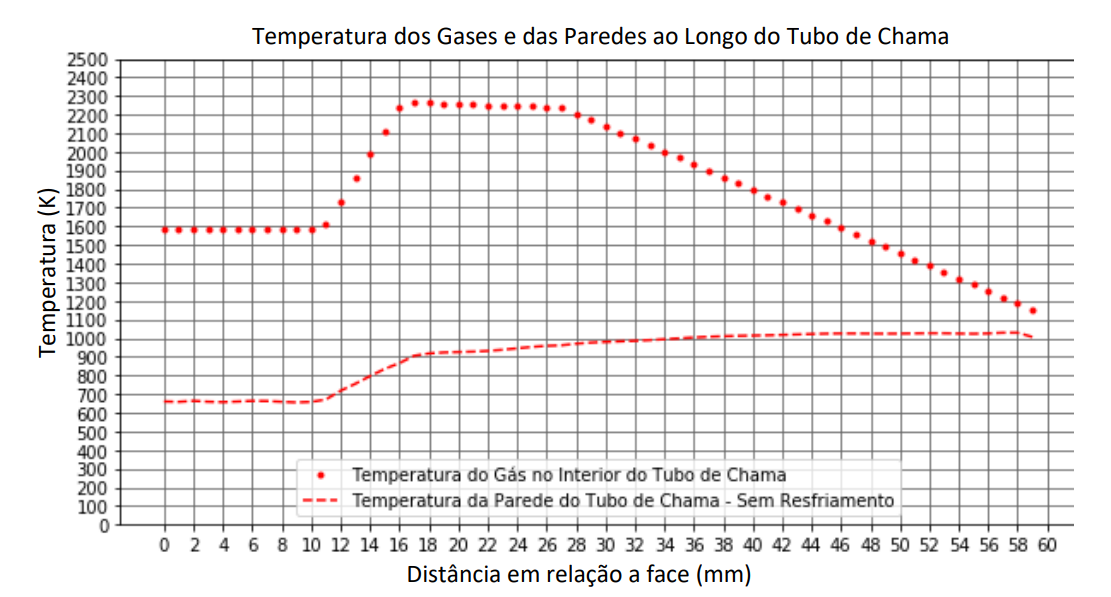
**Figura 3** – Temperatura do gás em função da zona.

De maneira análoga, foram efetuados os cálculos do fluxo mássico ao longo da câmara de combustão, de acordo com a Tabela 8 do trabalho-base, resultando no gráfico da Figura 4. Ao final da câmara de combustão, o fluxo mássico indicado no gráfico é exatamente metade do fluxo mássico de entrada (= 0.243 kg/s). Isso deve ao fato de que a análise feita leva em conta a parte externa e a parte interna da câmara de combustão.



**Figura 4** – Fluxo de massa em função da zona.

Com essas informações calculadas, foi possível iniciar o projeto de arrefecimento por filme de resfriamento. O resultado apresentado no trabalho-base para o perfil de temperatura ao longo da parede do tubo de chama indica a não necessidade de arrefecimento do sistema, pois a temperatura limite de 1100K estabelecida pelo autor não é atingida, como mostra a Figura 5. Devido a isso, alterou-se a temperatura alvo para 900K, afim de implementar a rotina de arrefecimento apresentada no trabalho-base. Em seguida, calculou-se o através dos dados fornecidos pelo GasTurb.



**Figura 5** – Temperatura dos Gases e das Paredes ao Longo do Tubo de Chama (GasTurb).

Em seguida foram implementados os cálculos das fendas em cada uma das zonas, adotando de forma arbitrária os parâmetros para a geometria das fendas, bem como a quantidade de fendas em cada uma das zonas, influenciando diretamente na vazão mássica das fendas. O cálculo da eficiência do arrefecimento () foi efetuado levando em conta a magnitude do parâmetro *m*.

Na primeira iteração, foi calculada a temperatura do gás próximo à parede, avaliando a eficiência de arrefecimento quando variados os parâmetros de abertura da fenda e quantidade de fendas.

Para a segunda iteração, manteve-se o número de fendas em cada zona, aumentando a abertura das fendas a fim de aumentar a vazão mássica e, consequentemente, aumentar o arrefecimento do sistema. Optou-se por alterar a abertura das fendas, pois observou-se que o processo de arrefecimento é mais sensível a esse parâmetro em relação a mudança no número de fendas. Foi possível verificar aumento na eficiência de arrefecimento.

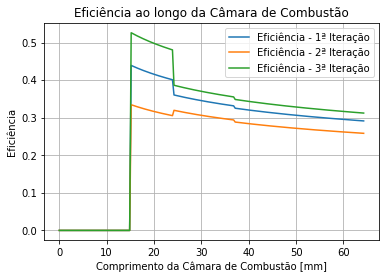
Por fim, na terceira iteração, aumentou-se o número de fendas com intuito de aproximar da quantidade real presente no microturbojato. Além disso, as aberturas das fendas também foram alteradas.

As propriedades geométricas e o número de fendas em cada uma das iterações podem ser visualizadas na Tabela 4.

| **Iteração** | **Zona** | **[mm]** | **[mm]** | **[mm]** | **Número de fendas** |
| --- | --- | --- | --- | --- | --- |
| **1** | **Primária** | 0,3 | 0,5 | 0,3 | 35 |
| **Secundária** | 0,3 | 0,5 | 0,3 | 20 |
| **Diluição** | 0,3 | 0,5 | 0,3 | 6 |
| **2** | **Primária** | 0,2 | 0,5 | 0,3 | 35 |
| **Secundária** | 0,2 | 0,5 | 0,3 | 20 |
| **Diluição** | 0,2 | 0,5 | 0,3 | 6 |
| **3** | **Primária** | 0,5 | 0,5 | 0,3 | 25 |
| **Secundária** | 0,8 | 0,5 | 0,3 | 29 |
| **Diluição** | 0,6 | 0,5 | 0,3 | 40 |

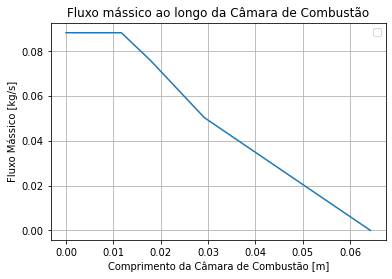
**Tabela 4** – Propriedades geométricas e número de fendas.

Os resultados obtidos para a eficiência nas três iterações estão apresentados na Figura 6.



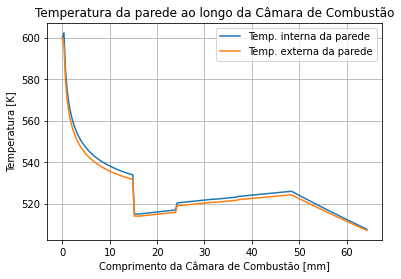
**Figura 6 –** Eficiência de cada iteração.

Para confirmar que o sistema foi corretamente dimensionado, é necessário fazer o balanceamento de massa do sistema. O gráfico da Figura 7 mostra que ao final da câmara de combustão, o fluxo de massa é zero, indicando que o sistema foi corretamente arrefecido.



**Figura 7 –** Convergência do fluxo mássico.

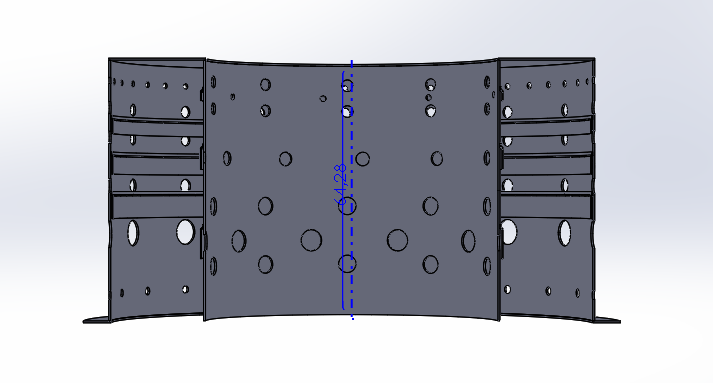
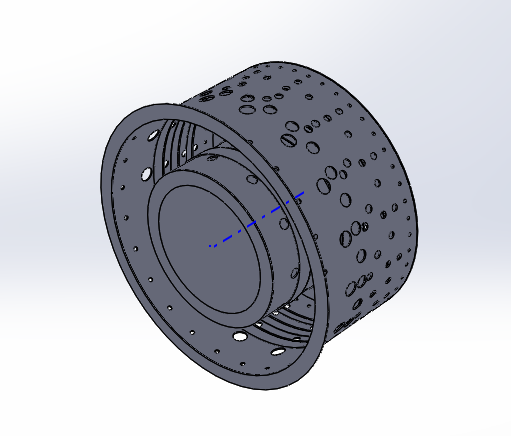
Em seguida, implementou-se a metodologia de arrefecimento proposta por Lefebvre e Ballal [3]. Ao fim dessa parte da implementação numérica, foi possível obter o valor da temperatura interna e externa da parede ao longo da câmara de combustão, visualizada no gráfico da Figura 8.



**Figura 8 –** Temperatura externa e interna da parede da câmara de combustão.

Com isso, conclui-se a implementação da metodologia de arrefecimento, à exceção do projeto dos orifícios de admissão de ar e tal etapa do projeto não foi executada devido às limitações de cronograma. Porém, a título de ilustração, as fendas foram esquematizadas no desenho CAD, seguindo os diâmetros originais apresentados no trabalho-base, como indicado na Figura 9.

Ao final, foi elaborado um modelo 3D indicando o posicionamento das fendas de arrefecimento, como mostra a Figura 9.



**Figura 9 –** Vistas das fendas de arrefecimento.

# **Conclusões**

Ao longo do presente trabalho, foi apresentada a implementação numérica da metodologia de arrefecimento da câmara de combustão de microturbojatos via filme de resfriamento. Conforme demonstrado, foi possível verificar que a metodologia se mostrou eficaz em aumentar a eficiência de arrefecimento e, consequentemente, reduzir a temperatura na câmara.

# **referências**

[1] VELLOSA, C. C. M. Projeto preliminar de câmara de combustão de microturbojato. UFABC, 2020.

[2] Y. A. Çengel e M. A. Boles, “Gas Power Cycles,” em Thermodynamics: An Engineering Approach, Boston, McGraw-Hill, 2006.

[3] A. H. LEFEBVRE, D. R. BALLAL, “Gas Turbine Combustion – Alternative Fuels and Emissions”, CRC Press, 2010.

# Bibliotecas usadas no decorrer do projeto

import numpy as np

import matplotlib.pyplot as plt

import math as m

import scipy.optimize as sp

#INPUTS

k = 143.5 #Constante para uso da área de referência, no SI

rho = 1.225 # [kg/m³] Densidade do ar

theta = 73\*(10\*\*6) # Parâmetro que relaciona eficiência da combustão (para máxima eficiência), no SI

deltaP3\_P3 = 0.06

deltaP3\_qref = 20

m\_dot\_3 = 0.243 # [kg/s] Vazão mássica de ar

m\_dot\_comb = 0.00398 # [kg/s] Vazão mássica do combustivél

m\_dot\_ZP = 0.243 # [kg/s]

P\_3 = 354.42\*(10\*\*3) # Pressão total estimada na entrada da câmara

P\_4 = 333.15\*(10\*\*3) # Pressão total estimada na saída da câmara

T\_3 = 498.42 # Temperatura estimada na entrada da câmara

T\_4 = 1100 # Temperatura na saída da câmara

q\_ref = 1 # Pressão dinâmica na máxima área de seção transversal da carcaça

phi\_ZP = 1.249

phi\_ZS = 0.8

phi\_ZD = 0.237

D\_int = 3\*(10\*\*(-2))

p\_3 = (1- 0.06)\*P\_3

P3\_deltaP = P\_3/(P\_3 - P\_4)

#INPUTS CALCULADOS

phi\_pobre = 0.70547 - 0.00046\*T\_3

phi\_rico = 1.46695 + 0.00172\*T\_3

phi\_real = m\_dot\_comb/m\_dot\_3

phi\_esteq = 0.06818

phi\_global = phi\_real/phi\_esteq

phi\_global\_rico = phi\_global/phi\_rico

phi\_global\_pobre = phi\_global/phi\_pobre

#CÁLCULO DO LIMITE DE FLAMABILIDADE

lim\_inf = phi\_global/1.07;

lim\_sup = phi\_global/1.5;

lim\_med = (lim\_inf+lim\_sup)/2;

#Determinação da massa da zona primária

m\_dot\_ZP = m\_dot\_3\*lim\_med;

#CÁLCULO DA ÁREA DE REFERÊNCIA E ÁREA DO TUBO DE CHAMA

A\_ref\_aero = np.sqrt(k\*(((m\_dot\_3\*np.sqrt(T\_3))/(P\_3))\*\*2)\*((deltaP3\_qref)/(deltaP3\_P3)))

D\_ref\_aero = (np.sqrt(((4\*A\_ref\_aero)/np.pi) + D\_int\*\*2) - D\_int)/2 #Tubo Anular

#Determinação do phi da Zona Primaria

phi\_ZP = phi\_global/(lim\_med)

#Cálculo do fator de correção

if phi\_ZP>=0.6 and phi\_ZP<1.0:

b = 245\*(1.39 + np.log(phi\_ZP))

elif phi\_ZP>=1.0 and phi\_ZP<1.4:

b = 170\*(2 - np.log(phi\_ZP))

A\_ref\_quim = (theta\*m\_dot\_3)/((P\_3\*\*1.75)\*(D\_ref\_aero\*\*0.75)\*m.e\*\*(T\_3/b))

D\_ref\_quim = (np.sqrt(((4\*A\_ref\_quim)/np.pi) + D\_int\*\*2) - D\_int)/2

A\_ref\_conv = A\_ref\_quim\*10\*\*6

D\_ref\_conv = D\_ref\_quim\*10\*\*3

def func\_1(A\_ref\_conv):

x = 0.5\*(np.sqrt((4\*A\_ref\_conv/np.pi + (D\_int\*\*2))) - D\_int) - D\_ref\_conv

return x

sol = sp.fsolve(func\_1, 1)

print(sol[0])

A\_ref = sol[0]/10\*\*6

D\_ref = (np.sqrt(((4\*A\_ref)/np.pi) + D\_int\*\*2) - D\_int)/2

#CARACTERÍSTICAS DO TUBO DE CHAMA

A\_ft = 0.65\*A\_ref

D\_ft = A\_ft/(np.pi\*(D\_int + D\_ref))

#CÁLCULO DO COMPRIMENTO DA CÂMARA DE COMBUSTÃO

L\_ZR = 0.5\*D\_ft

L\_ZP = 0.75\*D\_ft

L\_ZS = 0.5\*D\_ft

L\_ZD = 1.5\*D\_ft

L\_CC = L\_ZP + L\_ZS + L\_ZD

#CÁLCULO DO FLUXO DE MASSA NA ZONA SECUNDÁRIA

m\_dot\_ZS = (phi\_global/phi\_ZS)\*m\_dot\_3

#CÁLCULO DA PORCENTAGEM DE AR DESTINADA AO ARREFECIMENTO

m\_dot\_arref = m\_dot\_3\*((0.1\*T\_3)-30)

#CÁLCULO DA VAZÃO DE AR NA ZONA DE DILUIÇÃO

m\_dot\_ZD = m\_dot\_3\*(1 - ((m\_dot\_ZP + m\_dot\_ZS)/m\_dot\_3))

#BALANÇO DE MASSA

bal\_massa = m\_dot\_3 - (m\_dot\_ZS + m\_dot\_ZP + m\_dot\_ZD)

#CÁLCULO DAS TEMPERATURAS LOCAIS DA CÂMARA (TABELA 7)

#Cálculo para a temperatura da zona primária (recirculação)

deltaT\_ZR = 2350 - T\_3

n\_ZR\_a = 0.83 + 0.17\*np.tanh((1.5475\*10\*\*-3)\*(T\_3 + 108\*np.log(p\_3) - 1863))

T\_max\_ZR = T\_3 + n\_ZR\_a\*deltaT\_ZR

T\_med\_ZR = (1/3)\*T\_3 + (2/3)\*T\_max\_ZR

#Cálculo para a temperatura da zona primária (restante)

deltaT\_ZP = 3000 - T\_med\_ZR

n\_ZR\_b = 0.92 + 0.12\*np.tanh((1.5475\*10\*\*-3)\*(T\_3 + 108\*np.log(p\_3) - 1863))

T\_saida\_ZP = T\_3 + n\_ZR\_b\*deltaT\_ZP

#Cálculo da temperatura da zona secundária

T\_ent\_ZS = T\_saida\_ZP

V\_ZS = A\_ft\*L\_ZS

psi\_T3 = ((10\*\*(-3.054\*(phi\_ZS\*\*-1.205)))\*(T\_3\*\*(1.2327\*(phi\_ZS\*\*-1.205))))\*(m\_dot\_comb/(V\_ZS\*P\_3\*\*(2\*phi\_ZS)))

D\_estrela = 0.736 - 0.0173/P3\_deltaP

n\_ZS= 1/(10\*\*10\*\*(0.911\*np.log(psi\_T3) + 8.02\*phi\_ZS - 1.097 + D\_estrela))

deltaT\_ZS = 1.679421710365530\*10\*\*3

T\_saida\_ZS = T\_3 + n\_ZS\*deltaT\_ZS

#Cálculo da temperatura da zona de diluição

T\_ent\_ZD = T\_saida\_ZS

T\_saida\_ZD = T\_4 #T4 é um parâmetro padronizado conforme Fig. 22

comp1 = np.linspace(0, L\_ZR)

comp2 = np.linspace(L\_ZR, L\_ZP)

comp3 = np.linspace(L\_ZP, (L\_ZP + L\_ZS))

comp4 = np.linspace((L\_ZP+L\_ZS), L\_CC)

def temp\_ZR(x):

return T\_med\_ZR

def temp\_ZP(x):

return T\_med\_ZR + ((T\_saida\_ZP - T\_med\_ZR)/(L\_ZP - L\_ZR))\*(x-L\_ZR)

def temp\_ZS(x):

return T\_saida\_ZP + ((T\_saida\_ZS - T\_saida\_ZP)/L\_ZS)\*(x-L\_ZP)

def temp\_ZD(x):

return T\_saida\_ZS + ((T\_saida\_ZD - T\_saida\_ZS)/L\_ZD)\*(x - L\_ZP - L\_ZS)

temp\_1 = []

temp\_2 = []

temp\_3 = []

temp\_4 = []

for i in comp1:

temp\_1.append(temp\_ZR(i))

for i in comp2:

temp\_2.append(temp\_ZP(i))

for i in comp3:

temp\_3.append(temp\_ZS(i))

for i in comp4:

temp\_4.append(temp\_ZD(i))

T\_total = []

T\_total.extend(temp\_1)

T\_total.extend(temp\_2)

T\_total.extend(temp\_3)

T\_total.extend(temp\_4)

plt.plot(comp1\*1000, temp\_1, label='Zona de Recirc.')

plt.plot(comp2\*1000, temp\_2, label='Zona Primária')

plt.plot(comp3\*1000, temp\_3, label='Zona Secundária')

plt.plot(comp4\*1000, temp\_4, label='Zona de Diluição')

plt.grid(True, which='both')

plt.ylim([0, 2500])

plt.ylabel('Temperatura [K]')

plt.xlabel('Comprimento da Câmara de Combustão [mm]')

plt.title('Temperatura por Zona')

plt.legend()

plt.show()

#FLUXO DE MASSA EM CADA ZONA

comp\_m\_1 = np.linspace(0, L\_ZR)

comp\_m\_2 = np.linspace(L\_ZR, L\_ZP)

comp\_m\_3 = np.linspace(L\_ZP, (L\_ZP + L\_ZS))

comp\_m\_4 = np.linspace((L\_ZP+L\_ZS), L\_CC)

m\_g\_ZR = (3/4)\*m\_dot\_ZP

m\_g\_ZP = 1

m\_g\_ZS = 1

def vazao\_ZR(x):

return m\_g\_ZR

def vazao\_ZP(x):

global m\_g\_ZP

m\_g\_ZP = m\_g\_ZR + ((m\_dot\_ZP - m\_g\_ZR)\*(x - L\_ZR))/(L\_ZP - L\_ZR)

return m\_g\_ZP

def vazao\_ZS(x):

global m\_g\_ZS

m\_g\_ZS = m\_g\_ZP + ((m\_dot\_ZS - m\_g\_ZP)\*(x - L\_ZP))/(L\_ZS)

return m\_g\_ZS

def vazao\_ZD(x):

m\_g\_ZD = m\_g\_ZS + ((m\_dot\_ZD - m\_g\_ZS)\*(x - (L\_ZP + L\_ZS)))/(L\_ZD)

return m\_g\_ZD

mass\_1 = []

mass\_2 = []

mass\_3 = []

mass\_4 = []

for i in comp\_m\_1:

mass\_1.append(vazao\_ZR(i))

for i in comp\_m\_2:

mass\_2.append(vazao\_ZP(i))

for i in comp\_m\_3:

mass\_3.append(vazao\_ZS(i))

for i in comp\_m\_4:

mass\_4.append(vazao\_ZD(i))

plt.plot(comp\_m\_1\*1000, mass\_1, label = 'Zona de Recirc.')

plt.plot(comp\_m\_2\*1000, mass\_2, label = 'Zona Primária')

plt.plot(comp\_m\_3\*1000, mass\_3, label = 'Zona Secundária')

plt.plot(comp\_m\_4\*1000, mass\_4, label = 'Zona de Difusão')

plt.grid(True, which='both')

plt.ylabel('Fluxo de Massa [kg/s]')

plt.xlabel('Comprimento da Câmara de Combustão [mm]')

plt.title('Fluxo de Massa por Zona')

plt.legend()

plt.show()

m\_total = mass\_1 + mass\_2 + mass\_3 + mass\_4

comp\_total = []

comp\_total.extend(comp\_m\_1)

comp\_total.extend(comp\_m\_2)

comp\_total.extend(comp\_m\_3)

comp\_total.extend(comp\_m\_4)

## Arrefecimento da Parede do Tubo de Chama por Filme de Resfriamento

########## PRIMEIRA ITERAÇÃO DO NÚMERO DE FENDAS ##############

T\_alvo = 1100 #TEMPERATURA ACEITÁVEL PARA EVITAR O STRESS TÉRMICO

A\_an = 2\*np.pi\*(D\_ref-D\_ft)\*L\_CC

mu\_ar = (0.03863 + 0.00749\*T\_3 - 5.8564\*(10\*\*-6)\*(T\_3\*\*2) + 2.7769\*(10\*\*-9)\*(T\_3\*\*3) -

4.600774\*(10\*\*-13)\*(T\_3\*\*4))\*(10\*\*-5) #Viscosidade

#CÁLCULO DA FENDA DA ZONA PRIMÁRIA #modificado

t\_ZP = 0.3\*10\*\*-3 ###

t\_w\_ZP = 0.5\*10\*\*-3 ###

s\_ZP = 0.3\*10\*\*-3 ###

n\_fendas\_ZP = 0.0001

A\_total\_fendas\_ZP = n\_fendas\_ZP\*2\*np.pi\*s\_ZP\*(D\_ref - D\_ft)

m\_dot\_an\_ZP = m\_dot\_3

m\_dot\_g\_ZP = m\_dot\_3 - (3/4)\*m\_dot\_ZP

rho\_g\_u\_g\_ZP = m\_dot\_g\_ZP/A\_ft

m\_dot\_fenda\_ZP = m\_dot\_an\_ZP\*(A\_total\_fendas\_ZP/A\_an)

rho\_an\_u\_an\_ZP = m\_dot\_fenda\_ZP/A\_total\_fendas\_ZP

m\_ZP = (rho\_an\_u\_an\_ZP)/(rho\_g\_u\_g\_ZP)

#CÁLCULO DA FENDA DA ZONA SECUNDÁRIA

t\_ZS = 0.3\*10\*\*-3

s\_ZS = 0.3\*10\*\*-3

t\_w\_ZS = 0.5\*10\*\*-3

n\_fendas\_ZS = 20

A\_total\_fendas\_ZS = n\_fendas\_ZS\*2\*np.pi\*s\_ZS\*(D\_ref - D\_ft)

m\_dot\_an\_ZS = m\_dot\_3 - m\_dot\_ZS - m\_dot\_fenda\_ZP

m\_dot\_g\_ZS = m\_dot\_3 - m\_dot\_ZS

rho\_g\_u\_g\_ZS = m\_dot\_g\_ZS/A\_ft

m\_dot\_fenda\_ZS = m\_dot\_an\_ZS\*(A\_total\_fendas\_ZS/A\_an)

rho\_an\_u\_an\_ZS = m\_dot\_fenda\_ZS/A\_total\_fendas\_ZS

m\_ZS = (rho\_an\_u\_an\_ZS)/(rho\_g\_u\_g\_ZS)

#CÁLCULO DA FENDA DA ZONA DE DILUIÇÃO

t\_ZD = 0.3\*10\*\*-3

s\_ZD = 0.3\*10\*\*-3

t\_w\_ZD = 0.5\*10\*\*-3

n\_fendas\_ZD = 6

A\_total\_fendas\_ZD = n\_fendas\_ZD\*2\*np.pi\*s\_ZD\*(D\_ref - D\_ft)

m\_dot\_an\_ZD = m\_dot\_3 - m\_dot\_ZD - m\_dot\_fenda\_ZS

m\_dot\_g\_ZD = m\_dot\_3 - m\_dot\_ZD

rho\_g\_u\_g\_ZD = m\_dot\_g\_ZD/A\_ft

m\_dot\_fenda\_ZD = m\_dot\_an\_ZD\*(A\_total\_fendas\_ZD/A\_an)

rho\_an\_u\_an\_ZD = m\_dot\_fenda\_ZD/A\_total\_fendas\_ZD

m\_ZD = (rho\_an\_u\_an\_ZD)/(rho\_g\_u\_g\_ZD)

x = np.linspace(0, L\_CC, num=200)

a = range(len(x))

n\_r = []

for i in range(len(x)):

if x[i] < 0.015:

g\_1 = 0

n\_r.append(g\_1)

T\_g\_1 = T\_total[i]

elif x[i] >= 0.015 and x[i] < 0.024:

mu\_g = (0.03863 + 0.00749\*T\_g\_1 - 5.8564\*(10\*\*-6)\*(T\_g\_1\*\*2) + 2.7769\*(10\*\*-9)\*(T\_g\_1\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_1\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZS <= 1.3:

g\_3 = 1.1\*(m\_ZS\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZS)\*\*-0.2)\*((t\_ZS/s\_ZS)\*\*-0.2)

n\_r.append(g\_3)

elif 1.3 < m\_ZS < 4:

g\_4 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZS)\*\*-0.2)\*((t\_ZS/s\_ZS)\*\*-0.2)

n\_r.append(g\_4)

T\_g\_2 = T\_total[i]

elif x[i] >= 0.024 and x[i] < 0.037:

mu\_g = (0.03863 + 0.00749\*T\_g\_2 - 5.8564\*(10\*\*-6)\*(T\_g\_2\*\*2) +

2.7769\*(10\*\*-9)\*(T\_g\_2\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_2\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZD <= 1.3:

g\_5 = 1.1\*(m\_ZD\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r.append(g\_5)

elif 1.3 < m\_ZD < 4:

g\_6 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r.append(g\_6)

T\_g\_3 = T\_total[i]

elif x[i] >= 0.037 and x[i]<=L\_CC:

mu\_g = (0.03863 + 0.00749\*T\_g\_3 - 5.8564\*(10\*\*-6)\*(T\_g\_3\*\*2) +

2.7769\*(10\*\*-9)\*(T\_g\_3\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_3\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZD <= 1.3:

g\_7 = 1.1\*(m\_ZD\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r.append(g\_7)

elif 1.3 < m\_ZD < 4:

g\_8 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r.append(g\_8)

T\_g\_w = []

h = range(len(n\_r))

for i in h:

temp = T\_total[i] - n\_r[i]\*(T\_total[i] - T\_3)

T\_g\_w.append(temp)

## Arrefecimento da Parede do Tubo de Chama por Filme de Resfriamento

########## SEGUNDA ITERAÇÃO DO NÚMERO DE FENDAS ##############

T\_alvo = 900 #TEMPERATURA ACEITÁVEL PARA EVITAR O STRESS TÉRMICO

A\_an = 2\*np.pi\*(D\_ref-D\_ft)\*L\_CC

mu\_ar = (0.03863 + 0.00749\*T\_3 - 5.8564\*(10\*\*-6)\*(T\_3\*\*2) + 2.7769\*(10\*\*-9)\*(T\_3\*\*3) -

4.600774\*(10\*\*-13)\*(T\_3\*\*4))\*(10\*\*-5)

#CÁLCULO DA FENDA DA ZONA PRIMÁRIA

t\_ZP = 0.3\*10\*\*-3

t\_w\_ZP = 0.5\*10\*\*-3

s\_ZP = 0.2\*10\*\*-3

n\_fendas\_ZP = 35

A\_total\_fendas\_ZP = n\_fendas\_ZP\*2\*np.pi\*s\_ZP\*(D\_ref - D\_ft)

m\_dot\_an\_ZP = m\_dot\_3

m\_dot\_g\_ZP = m\_dot\_3 - (3/4)\*m\_dot\_ZP

rho\_g\_u\_g\_ZP = m\_dot\_g\_ZP/A\_ft

m\_dot\_fenda\_ZP = m\_dot\_an\_ZP\*(A\_total\_fendas\_ZP/A\_an)

rho\_an\_u\_an\_ZP = m\_dot\_fenda\_ZP/A\_total\_fendas\_ZP

m\_ZP = (rho\_an\_u\_an\_ZP)/(rho\_g\_u\_g\_ZP)

#CÁLCULO DA FENDA DA ZONA SECUNDÁRIA

t\_ZS = 0.3\*10\*\*-3

s\_ZS = 0.2\*10\*\*-3

t\_w\_ZS = 0.5\*10\*\*-3

n\_fendas\_ZS = 20

A\_total\_fendas\_ZS = n\_fendas\_ZS\*2\*np.pi\*s\_ZS\*(D\_ref - D\_ft)

m\_dot\_an\_ZS = m\_dot\_3 - m\_dot\_ZS - m\_dot\_fenda\_ZP

m\_dot\_g\_ZS = m\_dot\_3 - m\_dot\_ZS

rho\_g\_u\_g\_ZS = m\_dot\_g\_ZS/A\_ft

m\_dot\_fenda\_ZS = m\_dot\_an\_ZS\*(A\_total\_fendas\_ZS/A\_an)

rho\_an\_u\_an\_ZS = m\_dot\_fenda\_ZS/A\_total\_fendas\_ZS

m\_ZS = (rho\_an\_u\_an\_ZS)/(rho\_g\_u\_g\_ZS)

#CÁLCULO DA FENDA DA ZONA DE DILUIÇÃO

t\_ZD = 0.3\*10\*\*-3

s\_ZD = 0.2\*10\*\*-3

t\_w\_ZD = 0.5\*10\*\*-3

n\_fendas\_ZD = 6

A\_total\_fendas\_ZD = n\_fendas\_ZD\*2\*np.pi\*s\_ZD\*(D\_ref - D\_ft)

m\_dot\_an\_ZD = m\_dot\_3 - m\_dot\_ZD - m\_dot\_fenda\_ZS

m\_dot\_g\_ZD = m\_dot\_3 - m\_dot\_ZD

rho\_g\_u\_g\_ZD = m\_dot\_g\_ZD/A\_ft

m\_dot\_fenda\_ZD = m\_dot\_an\_ZD\*(A\_total\_fendas\_ZD/A\_an)

rho\_an\_u\_an\_ZD = m\_dot\_fenda\_ZD/A\_total\_fendas\_ZD

m\_ZD = (rho\_an\_u\_an\_ZD)/(rho\_g\_u\_g\_ZD)

x = np.linspace(0, L\_CC, num=200)

a = range(len(x))

n\_r\_2 = []

for i in range(len(x)):

if x[i] < 0.015:

g\_1 = 0

n\_r\_2.append(g\_1)

T\_g\_1 = T\_total[i]

elif x[i] >= 0.015 and x[i] < 0.024:

mu\_g = (0.03863 + 0.00749\*T\_g\_1 - 5.8564\*(10\*\*-6)\*(T\_g\_1\*\*2) + 2.7769\*(10\*\*-9)\*(T\_g\_1\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_1\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZS <= 1.3:

g\_3 = 1.1\*(m\_ZS\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZS)\*\*-0.2)\*((t\_ZS/s\_ZS)\*\*-0.2)

n\_r\_2.append(g\_3)

elif 1.3 < m\_ZS < 4:

g\_4 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZS)\*\*-0.2)\*((t\_ZS/s\_ZS)\*\*-0.2)

n\_r\_2.append(g\_4)

T\_g\_2 = T\_total[i]

elif x[i] >= 0.024 and x[i] < 0.037:

mu\_g = (0.03863 + 0.00749\*T\_g\_2 - 5.8564\*(10\*\*-6)\*(T\_g\_2\*\*2) + 2.7769\*(10\*\*-9)\*(T\_g\_2\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_2\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZD <= 1.3:

g\_5 = 1.1\*(m\_ZD\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_2.append(g\_5)

elif 1.3 < m\_ZD < 4:

g\_6 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_2.append(g\_6)

T\_g\_3 = T\_total[i]

elif x[i] >= 0.037 and x[i]<=L\_CC:

mu\_g = (0.03863 + 0.00749\*T\_g\_3 - 5.8564\*(10\*\*-6)\*(T\_g\_3\*\*2) + 2.7769\*(10\*\*-9)\*(T\_g\_3\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_3\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZD <= 1.3:

g\_7 = 1.1\*(m\_ZD\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_2.append(g\_7)

elif 1.3 < m\_ZD < 4:

g\_8 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_2.append(g\_8)

T\_g\_w\_2 = []

h = range(len(n\_r))

for i in h:

temp = T\_total[i] - n\_r\_2[i]\*(T\_total[i] - T\_3)

T\_g\_w\_2.append(temp)

########## TERCEIRA ITERAÇÃO DO NÚMERO DE FENDAS ##############

T\_alvo = 900 #TEMPERATURA ACEITÁVEL PARA EVITAR O STRESS TÉRMICO

A\_an = 2\*np.pi\*(D\_ref-D\_ft)\*L\_CC

mu\_ar = (0.03863 + 0.00749\*T\_3 - 5.8564\*(10\*\*-6)\*(T\_3\*\*2) + 2.7769\*(10\*\*-9)\*(T\_3\*\*3) -

4.600774\*(10\*\*-13)\*(T\_3\*\*4))\*(10\*\*-5)

#CÁLCULO DA FENDA DA ZONA PRIMÁRIA

t\_ZP = 0.3\*10\*\*-3

s\_ZP = 0.5\*10\*\*-3

t\_w\_ZP = 0.5\*10\*\*-3

n\_fendas\_ZP = 25

A\_total\_fendas\_ZP = n\_fendas\_ZP\*2\*np.pi\*s\_ZP\*(D\_ref - D\_ft)

m\_dot\_an\_ZP = m\_dot\_3

m\_dot\_g\_ZP = m\_dot\_3 - (3/4)\*m\_dot\_ZP

rho\_g\_u\_g\_ZP = m\_dot\_g\_ZP/A\_ft

m\_dot\_fenda\_ZP = m\_dot\_an\_ZP\*(A\_total\_fendas\_ZP/A\_an)

rho\_an\_u\_an\_ZP = m\_dot\_fenda\_ZP/A\_total\_fendas\_ZP

m\_ZP = (rho\_an\_u\_an\_ZP)/(rho\_g\_u\_g\_ZP)

#CÁLCULO DA FENDA DA ZONA SECUNDÁRIA

t\_ZS = 0.3\*10\*\*-3

s\_ZS = 0.8\*10\*\*-3

t\_w\_ZS = 0.5\*10\*\*-3

n\_fendas\_ZS = 29

A\_total\_fendas\_ZS = n\_fendas\_ZS\*2\*np.pi\*s\_ZS\*(D\_ref - D\_ft)

m\_dot\_an\_ZS = m\_dot\_3 - m\_dot\_ZS - m\_dot\_fenda\_ZP

m\_dot\_g\_ZS = m\_dot\_3 - m\_dot\_ZS

rho\_g\_u\_g\_ZS = m\_dot\_g\_ZS/A\_ft

m\_dot\_fenda\_ZS = m\_dot\_an\_ZS\*(A\_total\_fendas\_ZS/A\_an)

rho\_an\_u\_an\_ZS = m\_dot\_fenda\_ZS/A\_total\_fendas\_ZS

m\_ZS = (rho\_an\_u\_an\_ZS)/(rho\_g\_u\_g\_ZS)

#CÁLCULO DA FENDA DA ZONA DE DILUIÇÃO

t\_ZD = 0.3\*10\*\*-3

s\_ZD = 0.6\*10\*\*-3

t\_w\_ZD = 0.5\*10\*\*-3

n\_fendas\_ZD = 40

A\_total\_fendas\_ZD = n\_fendas\_ZD\*2\*np.pi\*s\_ZD\*(D\_ref - D\_ft)

m\_dot\_an\_ZD = m\_dot\_3 - m\_dot\_ZD - m\_dot\_fenda\_ZS

m\_dot\_g\_ZD = m\_dot\_3 - m\_dot\_ZD

rho\_g\_u\_g\_ZD = m\_dot\_g\_ZD/A\_ft

m\_dot\_fenda\_ZD = m\_dot\_an\_ZD\*(A\_total\_fendas\_ZD/A\_an)

rho\_an\_u\_an\_ZD = m\_dot\_fenda\_ZD/A\_total\_fendas\_ZD

m\_ZD = (rho\_an\_u\_an\_ZD)/(rho\_g\_u\_g\_ZD)

x = np.linspace(0, L\_CC, num=200)

a = range(len(x))

n\_r\_3 = []

for i in range(len(x)):

if x[i] < 0.015:

g\_1 = 0

n\_r\_3.append(g\_1)

T\_g\_1 = T\_total[i]

elif x[i] >= 0.015 and x[i] < 0.024:

mu\_g = (0.03863 + 0.00749\*T\_g\_1 - 5.8564\*(10\*\*-6)\*(T\_g\_1\*\*2) +

2.7769\*(10\*\*-9)\*(T\_g\_1\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_1\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZS <= 1.3:

g\_3 = 1.1\*(m\_ZS\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZS)\*\*-0.2)\*((t\_ZS/s\_ZS)\*\*-0.2)

n\_r\_3.append(g\_3)

elif 1.3 < m\_ZS < 4:

g\_4 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZS)\*\*-0.2)\*((t\_ZS/s\_ZS)\*\*-0.2)

n\_r\_3.append(g\_4)

T\_g\_2 = T\_total[i]

elif x[i] >= 0.024 and x[i] < 0.037:

mu\_g = (0.03863 + 0.00749\*T\_g\_2 - 5.8564\*(10\*\*-6)\*(T\_g\_2\*\*2) +

2.7769\*(10\*\*-9)\*(T\_g\_2\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_2\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZD <= 1.3:

g\_5 = 1.1\*(m\_ZD\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_3.append(g\_5)

elif 1.3 < m\_ZD < 4:

g\_6 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_3.append(g\_6)

T\_g\_3 = T\_total[i]

elif x[i] >= 0.037 and x[i]<=L\_CC:

mu\_g = (0.03863 + 0.00749\*T\_g\_3 - 5.8564\*(10\*\*-6)\*(T\_g\_3\*\*2) +

2.7769\*(10\*\*-9)\*(T\_g\_3\*\*3) - 4.600774\*(10\*\*-13)\*(T\_g\_3\*\*4))\*(10\*\*-5)

if 0.5 < m\_ZD <= 1.3:

g\_7 = 1.1\*(m\_ZD\*\*0.65)\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_3.append(g\_7)

elif 1.3 < m\_ZD < 4:

g\_8 = 1.28\*((mu\_ar/mu\_g)\*\*0.15)\*((x[i]/s\_ZD)\*\*-0.2)\*((t\_ZD/s\_ZD)\*\*-0.2)

n\_r\_3.append(g\_8)

T\_g\_w\_3 = []

h = range(len(n\_r))

for i in h:

temp = T\_total[i] - n\_r\_3[i]\*(T\_total[i] - T\_3)

T\_g\_w\_3.append(temp)

#plt.figure(figsize = (15,10))

plt.plot(x\*1000, n\_r, label = 'Eficiência - 1ª Iteração')

plt.plot(x\*1000, n\_r\_2, label = 'Eficiência - 2ª Iteração')

plt.plot(x\*1000, n\_r\_3, label = 'Eficiência - 3ª Iteração')

plt.grid(True, which='both')

plt.ylabel('Eficiência')

plt.xlabel('Comprimento da Câmara de Combustão [mm]')

plt.title('Eficiência ao longo da Câmara de Combustão')

plt.legend()

plt.show()

#plt.plot(x\*1000, T\_g\_w, label = 'Temperatura - 1ª Iteração')

#plt.plot(x\*1000, T\_g\_w\_2, label = 'Temperatura - 2ª Iteração')

#plt.plot(x\*1000, T\_g\_w\_3, label = 'Temperatura - 3ª Iteração')

#plt.grid(True, which='both')

#plt.ylabel('Temperatura [K]')

#plt.xlabel('Comprimento da Câmara de Combustão [mm]')

#plt.title('Temperatura ao longo da Câmara de Combustão')

#plt.legend()

#plt.show()

#BALANÇO DE MASSA

result\_2 = []

for i in range(len(m\_total)):

A = m\_dot\_3 - m\_total[i] - m\_dot\_fenda\_ZP - m\_dot\_fenda\_ZS - m\_dot\_fenda\_ZD

result\_2.append(A)

plt.plot(comp\_total, result\_2)

plt.grid(True, which='both')

plt.ylabel('Fluxo Mássico [kg/s]')

plt.xlabel('Comprimento da Câmara de Combustão [m]')

plt.title('Fluxo mássico ao longo da Câmara de Combustão')

plt.legend()

plt.show()

##### CÁLCULO DA TEMPERATURA EXTERNA E INTERNA DO TUBO DE CHAMA

### INTERNA #####

epsilon\_c = 0.7

epsilon\_w = 0.4

k\_w = 26

sigma = 5.67\*10\*\*-8

L = 1.7

Z = 0.6

q = 0.85\*phi\_ZP

l\_b\_int = D\_ft

k\_g = 5.92657\*10\*\*-4 + 9.80957\*10\*\*-5\*T\_3 - 4.89398\*10\*\*-8\*T\_3\*\*2 + 1.5011410\*10\*\*-11\*T\_3\*\*3

D\_an = D\_ref - D\_ft

epsilon\_g = 1 - np.e\*\*((-290\*P\_3\*L\*(q\*l\_b\_int)\*\*0.5)\*T\_3\*\*-1.5)

R\_1 = lambda T\_w\_1: 0.5\*sigma\*(1 + epsilon\_w)\*epsilon\_g\*T\_3\*\*1.5\*(T\_3\*\*2.5 - T\_w\_1\*\*2.5)

R\_2 = lambda T\_w\_2: Z\*sigma\*(T\_w\_2\*\*4 - T\_3\*\*4)

rey = lambda x: ((m\_dot\_g\_ZP)/(A\_ft))\*(x/mu\_g)

if 0.5<m\_ZP<=1.3:

C\_1 = lambda x, T\_w\_1: 0.069\*(k\_g/x)\*rey(x)\*\*0.7\*(T\_g\_w\_3[i] - T\_w\_1)

elif 1.3< m < 4:

C\_1 = lambda x, T\_w\_1: 0.01\*(k\_g/x[i])\*rey(x[i])\*\*0.8\*((x/s\_ZP)\*\*-0.36)\*(T\_g\_w\_3[i] - T\_w\_1)

C\_2 = lambda T\_w\_2: 0.02\*(k\_g/D\_an\*\*0.2)\*(m\_dot\_an\_ZP/(A\_an\*mu\_ar))\*(T\_w\_2 - T\_3)

K\_1\_2 = lambda T\_w\_1, T\_w\_2: (k\_w/t\_w\_ZP)\*(T\_w\_1 - T\_w\_2)

x = np.linspace(0, L\_CC,num=200)

a = range(len(x))

T\_w\_1\_res = []

T\_w\_2\_res = []

for i in a:

def func(args):

T\_w\_1, T\_w\_2 = args

eq1 = R\_1(T\_w\_1) + C\_1(x[i], T\_w\_1) - R\_2(T\_w\_2) - C\_2(T\_w\_2)

eq2 = R\_1(T\_w\_1) + C\_1(x[i], T\_w\_1) - K\_1\_2(T\_w\_1, T\_w\_2)

return eq1, eq2

T\_w\_1, T\_w\_2 = sp.fsolve(func, (600,600))

T\_w\_1\_res.append(T\_w\_1)

T\_w\_2\_res.append(T\_w\_2)

plt.plot(x\*1000, T\_w\_1\_res, label='Temp. interna da parede')

plt.plot(x\*1000, T\_w\_2\_res, label='Temp. externa da parede')

plt.grid(True, which='both')

plt.ylabel('Temperatura [K]')

plt.xlabel('Comprimento da Câmara de Combustão [mm]')

plt.title('Temperatura da parede ao longo da Câmara de Combustão')

plt.legend()

plt.show()