

Universidade de São Paulo
Instituto de Matemática e Estatística: Dep. Mat. Aplicada
Escola Politécnica

Luciano Chaparin Luisi - 9016866
luciano.luisi@usp.br
Bruno Prasinos Bernal - 10355141
brunobernal@usp.br

**Modelagem de um sistema de resfriamento de
chips: EP3**

Trabalho apresentado como avaliação da disciplina
MAP 3121 - Métodos Numéricos e Aplicações

São Paulo
2022

RESUMO

O objetivo do projeto é desenvolver funções computacionais para resolução de equações a partir do modelo de elementos finitos, baseado no modelamento e análise da esquematização do comportamento de difusão térmica que ocorre em um processador ou chip de computador de tamanho $L \times L$ e altura a , com um resfriador (“cooler” ou placa fria) colado na parte superior do bloco do chip.

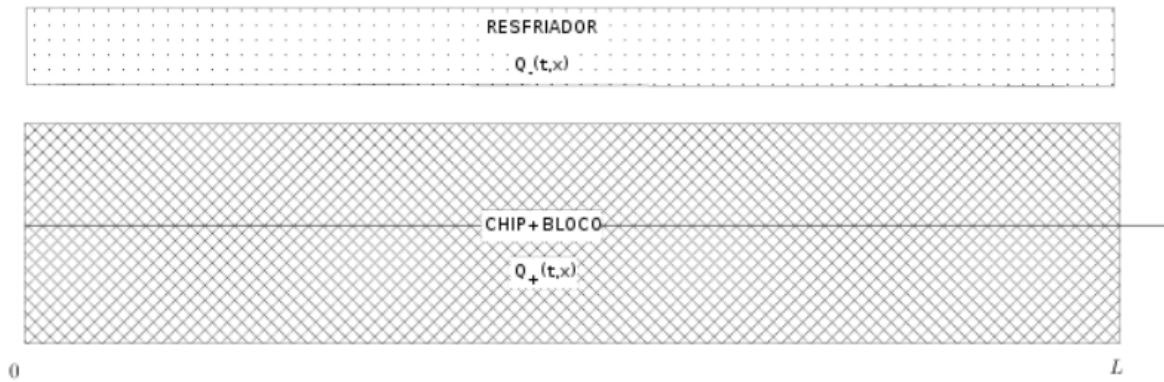
Implementou-se o método de elementos finitos para resolver a equação que é solução da equação de calor obtida da lei de Fourier e da propriedade de conservação de energia. Verificou-se a convergência do método calculando as aproximações com $n = 7, 15, 31$ e 63 , avaliando o erro associado às aproximações tal que a convergência é quadrática e proporcional a h^2 (com $h=1/(n+1)$).

Foram adicionadas complexidades ao projeto levando o algoritmo de solução da equação a trabalhar com parâmetros reais (alguns fornecidos e outros testados e estimados manualmente) tais como: produção e retirada de calor utilizando gaussianas, condutividade térmica do material, potência do chip, tamanho do chip, entre outras.

1 INTRODUÇÃO

1.1 CONCEITOS

É considerado o caso unidimensional analisando apenas a seção transversal do chip para cada x , de 0 a L , assumindo que a espessura do chip (h) é suficientemente fina para que a variação de temperatura na vertical fosse desprezível. A troca de calor no topo do chip com o resfriador é assumida perfeita e não há troca de calor na parte inferior do chip com o ambiente (a base é termicamente isolada), portanto a análise que foi feita leva em conta apenas as variações de temperatura na direção x .



Considera-se um processador que trabalhe em regime constante, supondo que trabalhe gerando sempre a mesma quantidade de calor e que o resfriador sempre consiga extrair a mesma quantidade de calor, obtendo assim um estado estacionário modelado por:

$$-\frac{\partial}{\partial x} \left(k(x) \frac{\partial T(x)}{\partial x} \right) = Q(x),$$

Assim foi obtida a seguinte equação a ser resolvida:

$$L(u(x)) := (-k(x)u'(x))' + q(x)u(x) = f(x) \quad x \in (0, 1), \quad u(0) = u(1) = 0$$

Usando o método de Ritz-Raleigh para a aproximação da solução do problema, obtém-se o sistema linear:

$$\begin{bmatrix} \langle \phi_1, \phi_1 \rangle_L & \langle \phi_2, \phi_1 \rangle_L & \dots & \langle \phi_n, \phi_1 \rangle_L \\ \dots & \dots & \dots & \dots \\ \langle \phi_1, \phi_n \rangle_L & \langle \phi_2, \phi_n \rangle_L & \dots & \langle \phi_n, \phi_n \rangle_L \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \dots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \langle f, \phi_1 \rangle \\ \dots \\ \langle f, \phi_n \rangle \end{bmatrix}$$

Cuja resolução é realizada através de funções definidas nos exercícios-programa 1 e 2, testadas e validadas como será apresentado nos testes a seguir.

2 IMPLEMENTAÇÃO E TESTES

2.1 ARQUIVOS DO PROJETO

2.1.1 requirements

Diretório contendo arquivos listando os módulos e versões instalados no projeto, testado e executado em ambiente Conda.

2.1.2 testes_exemplos_ep1

Diretório que possui arquivos .CSV utilizados para os testes apresentados no relatório do EP1 (outros testes foram realizados com diferentes matrizes e dimensões, mas não anexados ao projeto).

2.1.3 LEIAME.txt (um para cada: ep1, ep2 e ep3)

Documentação e instruções pertinentes ao uso de cada programa.

2.1.4 custom_functions.py

Módulo de funções usadas para recebimento, cálculo e devolução dos resultados, pertinentes ao exercício programa.

2.1.5 ep1.py, ep2.py e ep3.py

Executáveis que realizam o recebimento e retorno de dados do usuário num terminal através de linha de comando.

2.2 ALGORITMO

O programa foi escrito e testado em Python 3.7, utilizando a biblioteca NumPy para a realização dos cálculos e Matplotlib para criar figuras das curvas com os respectivos dados.

Observação: por padrão, a temperatura ambiente foi fixada em 20°C. Nesse sentido, os resultados de distribuição de temperatura se iniciam a partir desse valor, também observável nas extremidades.

2.2.1 solveElemFinitos(f,n,l,q,k,exata)

Essa função do módulo de funções customizadas obtém a solução aproximada $u(x)$ para uma equação do tipo:

$$L(u(x)) := (-k(x)u'(x))' + q(x)u(x) = f(x) \quad x \in (0, 1), \quad u(0) = u(1) = 0$$

Em que f é a função $f(x)$, n é o número de nós no intervalo, l é o tamanho do intervalo de x (de 0 a 1) argumento opcional (vale 1 por padrão), q é a função $q(x)$ argumento opcional (vale 0 por padrão - estado estacionário), k é a função $k(x)$ argumento opcional (vale 1 por padrão), $exata$ é a solução exata $\bar{u}(x)$ argumento opcional (vale *None* por padrão) e é utilizada para fins de comparação.

```
def solveElemFinitos(f, n: int, l: Optional[float] = 1, q: Optional['function'] = lambda x: 0, k: Optional['function'] = lambda
x: 1, exata: Optional['function'] = None) -> 'tuple(np.ndarray, np.ndarray, np.ndarray)':
    '''Resolve uma equação do tipo  $L(u(x)) := (-k(x)u'(x))' + q(x)u(x) = f(x)$ , para  $x$  em  $[0,1]$ , com  $u(0) = u(1) = 0$ , em
    'n' nós uniformemente espaçados....

    h = 1/(n+1) # Calcula valor de h
    xi = np.array([i*h for i in range(n+2)], float) # Calcula o valor das abscissas dos nós

    # Calcula os produtos internos de f com cada função chapéu, e entre as funções chapéu, utilizando função de integração
    numérica do EP2
    Q1, Q2, Q3, Q4, Q5, Q6 = (np.array([], float) for _ in range(6)) # Cria listas para receber os valores
    for i in range(1, n+1):
        if(i<n): Q1 = np.append(Q1,gaussIntegrate(lambda x: ((xi[i+1]-x)/h)*((x-xi[i])/h)*q(x),xi[i],xi[i+1],10))
        Q2 = np.append(Q2,gaussIntegrate(lambda x: (((x-xi[i-1])/h)**2)*q(x),xi[i-1],xi[i],10))
        Q3 = np.append(Q3,gaussIntegrate(lambda x: (((xi[i+1]-x)/h)**2)*q(x),xi[i],xi[i+1],10))
        Q4 = np.append(Q4,gaussIntegrate(lambda x: (1/h)*(1/h)*k(x),xi[i-1],xi[i],10))
        if(i==n): Q4 = np.append(Q4,gaussIntegrate(lambda x: (1/h)*(1/h)*k(x),xi[i],xi[i+1],10))
        Q5 = np.append(Q5,gaussIntegrate(lambda x: ((x-xi[i-1])/h)*(f(x)),xi[i-1],xi[i],10))
        Q6 = np.append(Q6,gaussIntegrate(lambda x: ((xi[i+1]-x)/h)*(f(x)),xi[i],xi[i+1],10))

    # Monta o sistema matricial referente à equação (8) do enunciado do exercício, da forma A*x = d
    subdiagonal = -Q4[1:-1] + Q1
    diagonal = Q4[:-1] + Q4[1:] + Q2 + Q3
    sobredagonal = -Q4[1:-1] + Q1
    independentes = Q5 + Q6

    # Resolve o sistema tridiagonal utilizando função do EP1
    u = solveTridi(subdiagonal,diagonal,sobredagonal,independentes)
    u = np.append(np.insert(u, 0, 0), 0)

    # Calcula a solução exata se foi definida
    if(exata is not None):
        u_bar = exata(xi[1:-1])
        u_bar = np.append(np.insert(u_bar, 0, 0), 0)
    else:
        u_bar = None

    return xi, u, u_bar
```

2.3 TESTES

A seguir são apresentados algumas capturas de tela do resultado de cada tipo de ação do programa, que apresenta ao usuário seus dados de entrada e resultados formatados. Os testes aqui apresentados não contemplam todos os testes, sendo apenas um demonstrativo do modelo de saída do programa.

2.3.1 Validação - Seção 4.2: $f(x)=12x(1-x)-2$, x no intervalo $[0,1]$, $q(x)=0$, $k(x)=1$, nós=[7,15,31,63], $\bar{u}(x)=x^2(1-x)^2$

Input

```
def f(x):
    return 12*x*(1-x)-2

def solucaoExata(x):
    return (x**2)*((1-x)**2)

L = 1

for n in nos:

    xi, u, u_bar = cf.solveElemFinitos(f,n,l=L,exata=solucaoExata)
```

Output

Para n=7 nós:

(7 nós) Abcissas: [0. 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1.]

(7 nós) Vetor de soluções u(x): [20. 20.01196289 20.03515625 20.05493164 20.0625 20.05493164 20.03515625 20.01196289 20.]

(7 nós) Vetor de soluções exatas u_bar(x): [20. 20.01196289 20.03515625 20.05493164 20.0625 20.05493164 20.03515625 20.01196289 20.]

(7 nós) Vetor diferença absoluta u-u_bar(x): [0. 0. 0. 0. 0. 0. 0. 0. 0.]

(7 nós) Erro máximo: 0.000000

Para n=15 nós:

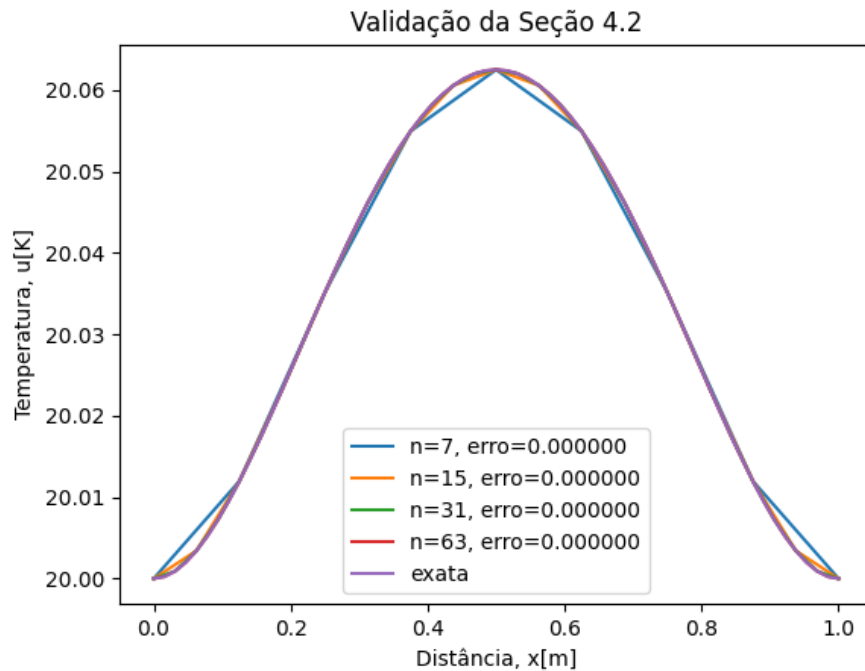
(15 nós) Abcissas: [0. 0.0625 0.125 0.1875 0.25 0.3125 0.375 0.4375 0.5 0.5625 0.625 0.6875 0.75 0.8125 0.875 0.9375 1.]

(15 nós) Vetor de soluções u(x): [20. 20.00343323 20.01196289 20.02320862 20.03515625 20.04615784 20.05493164 20.06056213 20.0625 20.06056213 20.05493164 20.04615784 20.03515625 20.02320862 20.01196289 20.00343323 20.]

(15 nós) Vetor de soluções exatas u_bar(x): [20. 20.00343323 20.01196289 20.02320862 20.03515625 20.04615784 20.05493164 20.06056213 20.0625 20.06056213 20.05493164 20.04615784 20.03515625 20.02320862 20.01196289 20.00343323 20.]

(15 nós) Vetor diferença absoluta u-u_bar(x): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

(63 nós) Erro máximo: 0.000000



2.3.2 Complemento da Seção 4.2: $f(x)=\exp(x)+1$, x no intervalo $[0,1]$, $q(x)=0$, $k(x)=\exp(x)$, nós=[7,15,31,63], $\bar{u}(x)=(x-1)(\exp(-x)-1)$

Input

```
def f(x):
    return np.exp(x)+1

def k(x):
    return np.exp(x)

def solucaoExata(x):
    return (x-1)*(np.exp(-x)-1)

L = 1

for n in nos:
    xi, u, u_bar = cf.solveElemFinitos(f,n,l=L,k=k,exata=solucaoExata)
```

Output

Para n=7 nós:

(7 nós) Abcissas: [0. 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1.]

(7 nós) Vetor de soluções u(x): [20. 20.10276066 20.16581358 20.19534565 20.19663803 20.1741936 20.13184695 20.07285922 20.]

(7 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [20. 20.10281521 20.16589941 20.1954442 20.19673467 20.17427696 20.13190836 20.07289225 20.]

(7 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0.00000000e+00 5.45511864e-05 8.58279933e-05 9.85468570e-05 9.66371710e-05 8.33611259e-05 6.14162536e-05 3.30230843e-05 0.00000000e+00]

(7 nós) Erro máximo: 0.000099

Para n=15 nós:

(15 nós) Abcissas: [0. 0.0625 0.125 0.1875 0.25 0.3125 0.375 0.4375 0.5 0.5625 0.625 0.6875 0.75 0.8125 0.875 0.9375 1.]

(15 nós) Vetor de soluções $u(x)$: [20. 20.05679262 20.10280156 20.13889563 20.16587794 20.18449068 20.19541955 20.19929789 20.1967105 20.18819719 20.17425611 20.15534679 20.131893 20.1042854 20.07288399 20.0380204 20.]

(15 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [20. 20.05680025 20.10281521 20.13891384 20.16589941 20.18451426 20.1954442 20.1993227 20.19673467 20.18822001 20.17427696 20.15536513 20.13190836 20.10429738 20.07289225 20.03802465 20.]

(15 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0.00000000e+00 7.63299162e-06 1.36457875e-05 1.82066942e-05 2.14695709e-05 2.35749629e-05 2.46511499e-05 2.48151162e-05 2.41734487e-05 2.28231672e-05 2.08524927e-05 1.83415571e-05 1.53630600e-05 1.19828746e-05 8.26060847e-06 4.25012076e-06 0.00000000e+00]

(15 nós) Erro máximo: 0.000025

Para n=31 nós:

(31 nós) Abcissas: [0. 0.03125 0.0625 0.09375 0.125 0.15625 0.1875 0.21875 0.25 0.28125 0.3125 0.34375 0.375 0.40625 0.4375 0.46875 0.5 0.53125 0.5625 0.59375 0.625 0.65625 0.6875 0.71875 0.75 0.78125 0.8125 0.84375 0.875 0.90625 0.9375 0.96875 1.]

(31 nós) Vetor de soluções $u(x)$: [20. 20.0298043 20.05679835 20.08109728 20.1028118 20.12204836 20.13890929 20.15349299 20.16589404 20.17620337 20.18450836 20.19089301 20.19543804 20.19822102 20.1993165 20.1987961 20.19672863 20.19318019 20.18821431 20.18189196 20.17427175 20.16540993 20.15536055 20.14417547 20.13190452 20.11859552 20.10429438 20.08904517 20.07289018 20.05587 20.03802359 20.0193883 20.]

(31 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [20. 20.0298053 20.05680025 20.08109998 20.10281521 20.12205238 20.13891384 20.15349799 20.16589941 20.17620904 20.18451426 20.19089907 20.1954442 20.19822723 20.1993227 20.19880224 20.19673467 20.19318609 20.18822001 20.18189744 20.17427696 20.16541485 20.15536513 20.1441797 20.13190836 20.11859895 20.10429738 20.08904771 20.07289225 20.05587158 20.03802465 20.01938884 20.]

(31 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0.00000000e+00 1.00768173e-06 1.90852743e-06 2.70815725e-06 3.41194661e-06 4.02503600e-06 4.55234029e-06 4.99855780e-06 5.36817896e-06 5.66549460e-06 5.89460407e-06 6.05942289e-06 6.16369023e-06 6.21097601e-06 6.20468781e-06 6.14807746e-06 6.04424742e-06 5.89615687e-06 5.70662760e-06 5.47834966e-06 5.21388680e-06 4.91568170e-06 4.58606097e-06 4.22724000e-06 3.84132761e-06 3.43033048e-06 2.99615747e-06 2.54062373e-06 2.06545463e-06 1.57228962e-06 1.06268584e-06 5.38121643e-07 0.00000000e+00]

(31 nós) Erro máximo: 0.000006

Para $n=63$ nós:

(63 nós) Abcissas: [0. 0.015625 0.03125 0.046875 0.0625 0.078125 0.09375 0.109375 0.125 0.140625 0.15625 0.171875 0.1875 0.203125 0.21875 0.234375 0.25 0.265625 0.28125 0.296875 0.3125 0.328125 0.34375 0.359375 0.375 0.390625 0.40625 0.421875 0.4375 0.453125 0.46875 0.484375 0.5 0.515625 0.53125 0.546875 0.5625 0.578125 0.59375 0.609375 0.625 0.640625 0.65625 0.671875 0.6875 0.703125 0.71875 0.734375 0.75 0.765625 0.78125 0.796875 0.8125 0.828125 0.84375 0.859375 0.875 0.890625 0.90625 0.921875 0.9375 0.953125 0.96875 0.984375 1.]

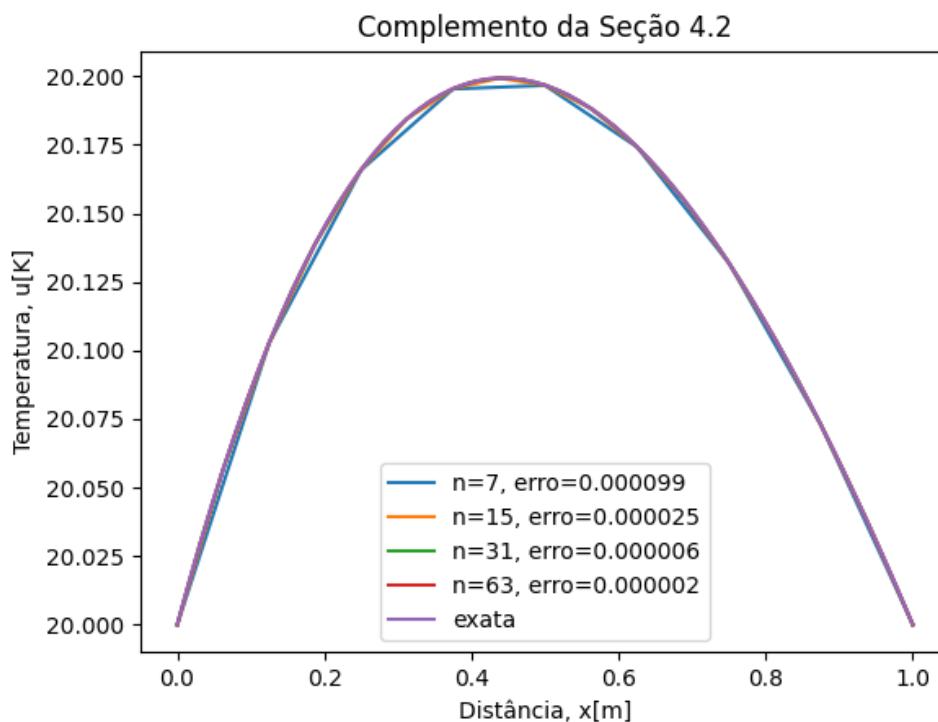
(63 nós) Vetor de soluções $u(x)$: [20. 20.01526119 20.02980505 20.0436464 20.05679978 20.06927942

20.08109931 20.09227314 20.10281436 20.11273613 20.12205137 20.13077276 20.1389127 20.14648338 20.15349674 20.15996448 20.16589807 20.17130877 20.17620762 20.18060542 20.18451278 20.1879401 20.19089755 20.19339514 20.19544266 20.1970497 20.19822568 20.19897981 20.19932115 20.19925855 20.19880071 20.19795613 20.19673316 20.19513998 20.19318462 20.19087492 20.18821859 20.18522317 20.18189607 20.17824453 20.17427566 20.16999642 20.16541362 20.16053396 20.15536399 20.14991012 20.14417864 20.13817572 20.1319074 20.12537959 20.11859809 20.11156859 20.10429663 20.09678768 20.08904708 20.08108005 20.07289173 20.06448714 20.05587118 20.04704869 20.03802438 20.02880288 20.0193887 20.0097863 20.]

(63 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [20. 20.01526132 20.0298053 20.04364677 20.05680025 20.06928 20.08109998 20.09227391 20.10281521 20.11273706 20.12205238 20.13077383 20.13891384 20.14648458 20.15349799 20.15996577 20.16589941 20.17131016 20.17620904 20.18060687 20.18451426 20.18794159 20.19089907 20.19339667 20.1954442 20.19705125 20.19822723 20.19898137 20.1993227 20.1992601 20.19880224 20.19795765 20.19673467 20.19514148 20.19318609 20.19087637 20.18822001 20.18522457 20.18189744 20.17824587 20.17427696 20.16999768 20.16541485 20.16053515 20.15536513 20.14991122 20.1441797 20.13817673 20.13190836 20.1253805 20.11859895 20.11156939 20.10429738 20.09678837 20.08904771 20.08108063 20.07289225 20.06448759 20.05587158 20.04704902 20.03802465 20.02880308 20.01938884 20.00978636 20.]

(63 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0.00000000e+00 1.29393854e-07 2.51929659e-07 3.67788960e-07 4.77149332e-07 5.80184430e-07 6.77064104e-07 7.67954450e-07 8.53017887e-07 9.32413254e-07 1.00629585e-06 1.07481752e-06 1.13812675e-06 1.19636868e-06 1.24968521e-06 1.29821508e-06 1.34209388e-06 1.38145418e-06 1.41642552e-06 1.44713453e-06 1.47370498e-06 1.49625781e-06 1.51491120e-06 1.52978064e-06 1.54097899e-06 1.54861649e-06 1.55280086e-06 1.55363735e-06 1.55122875e-06 1.54567550e-06 1.53707565e-06 1.52552503e-06 1.51111719e-06 1.49394350e-06 1.47409320e-06 1.45165339e-06 1.42670914e-06 1.39934351e-06 1.36963757e-06 1.33767046e-06 1.30351943e-06 1.26725989e-06 1.22896543e-06 1.18870785e-06 1.14655723e-06 1.10258194e-06 1.05684870e-06 1.00942258e-06 9.60367071e-07 9.09744085e-07 8.57614026e-07 8.04035782e-07 7.49066796e-07 6.92763066e-07 6.35179191e-07 5.76368397e-07 5.16382563e-07 4.55272271e-07 3.93086797e-07 3.29874169e-07 2.65681187e-07 2.00553441e-07 1.34535338e-07 6.76701468e-08 0.00000000e+00]

(63 nós) Erro máximo: 0.000002



2.3.3 Equilíbrio com forçantes de calor (distribuído: σ e $\theta \sim L/2$):

$f(x)=Q_+(x)-Q_-(x)$, x no intervalo $[0,0.02]$, $q(x)=0$, $k(x)=3.6$,

$nós=[7,15,31,63]$, $Q_+^0=15000$, $Q_-^0=5000$, $\sigma=0.01$, $\theta=0.01$

Input

```

Q_zero_mais = 15000    # Constante de geração de calor, em W/m³

Q_zero_menos = 5000    # Constante de resfriamento, em W/m³

L = 0.02              # Comprimento total do bloco do chip, em metros

sigma = 0.001         # Concentração de distribuição de aquecimento, em metros

theta = 0.001         # Concentração de distribuição de resfriamento, em metros

def Qmais(x, sigma, Q_zero_mais):
    return Q_zero_mais*np.exp(-((x-(L/2))**2)/(sigma**2))

def Qmenos(x, theta, Q_zero_menos):
    return Q_zero_menos*(np.exp(-(x**2)/(theta**2)) + np.exp(-(x-L)**2)/(theta**2)))

def f(x, sigma, theta, Q_zero_mais, Q_zero_menos):
    return Qmais(x, sigma, Q_zero_mais) - Qmenos(x, theta, Q_zero_menos)

def k(x):
    return 3.6

for n in nos:

```

```
xi, u, u_bar = cf.solveElemFinitos(lambda x: f(x, sigma, theta, Q_zero_mais,
Q_zero_menos),n,l=L,k=k)
```

Output

Para n=7 nós:

(7 nós) Abcissas: [0. 0.0025 0.005 0.0075 0.01 0.0125 0.015 0.0175 0.02]

(7 nós) Vetor de soluções u(x): [20. 20.04594235 20.08553769 20.11263058 20.12229148 20.11263058 20.08553769 20.04594235 20.]

Para n=15 nós:

(15 nós) Abcissas: [0. 0.00125 0.0025 0.00375 0.005 0.00625 0.0075 0.00875 0.01 0.01125 0.0125 0.01375 0.015 0.01625 0.0175 0.01875 0.02]

(15 nós) Vetor de soluções u(x): [20. 20.02338874 20.04594235 20.06692213 20.08553769 20.10100825 20.11263058 20.11984514 20.12229148 20.11984514 20.11263058 20.10100825 20.08553769 20.06692213 20.04594235 20.02338874 20.]

Para n=31 nós:

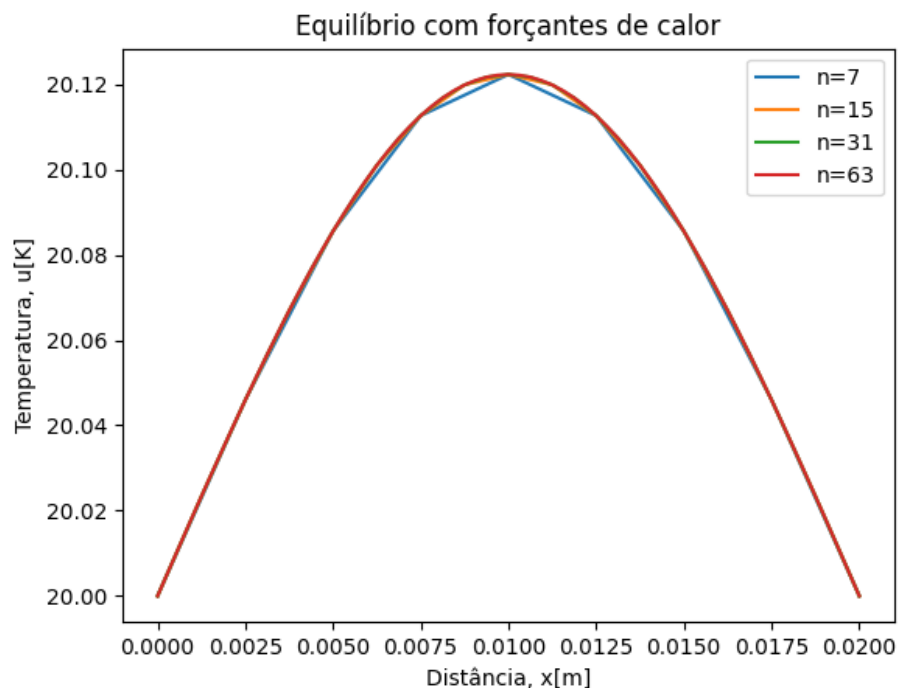
(31 nós) Abcissas: [0. 0.000625 0.00125 0.001875 0.0025 0.003125 0.00375 0.004375 0.005 0.005625 0.00625 0.006875 0.0075 0.008125 0.00875 0.009375 0.01 0.010625 0.01125 0.011875 0.0125 0.013125 0.01375 0.014375 0.015 0.015625 0.01625 0.016875 0.0175 0.018125 0.01875 0.019375 0.02]

(31 nós) Vetor de soluções u(x): [20. 20.01175604 20.02338874 20.03481425 20.04594235 20.05667783 20.06692213 20.07657525 20.08553769 20.09371267 20.10100825 20.1073395 20.11263058 20.11681657 20.11984514 20.12167793 20.12229148 20.12167793 20.11984514 20.11681657 20.11263058 20.1073395 20.10100825 20.09371267 20.08553769 20.07657525 20.06692213 20.05667783 20.04594235 20.03481425 20.02338874 20.01175604 20.]

Para n=63 nós:

(63 nós) Abcissas: [0. 0.0003125 0.000625 0.0009375 0.00125 0.0015625 0.001875 0.0021875 0.0025 0.0028125 0.003125 0.0034375 0.00375 0.0040625 0.004375 0.0046875 0.005 0.0053125 0.005625 0.0059375 0.00625 0.0065625 0.006875 0.0071875 0.0075 0.0078125 0.008125 0.0084375 0.00875 0.0090625 0.009375 0.0096875 0.01 0.0103125 0.010625 0.0109375 0.01125 0.0115625 0.011875 0.0121875 0.0125 0.0128125 0.013125 0.0134375 0.01375 0.0140625 0.014375 0.0146875 0.015 0.0153125 0.015625 0.0159375 0.01625 0.0165625 0.016875 0.0171875 0.0175 0.0178125 0.018125 0.0184375 0.01875 0.0190625 0.019375 0.0196875 0.02]

(63 nós) Vetor de soluções u(x): [20. 20.00588848 20.01175604 20.01759288 20.02338874 20.0291329 20.03481425 20.04042133 20.04594235 20.05136527 20.05667783 20.06186762 20.06692213 20.07182884 20.07657525 20.08114894 20.08553769 20.0897295 20.09371267 20.09747588 20.10100825 20.10429939 20.1073395 20.1101194 20.11263058 20.11486529 20.11681657 20.11847827 20.11984514 20.12091283 20.12167793 20.12213797 20.12229148 20.12213797 20.12167793 20.12091283 20.11984514 20.11847827 20.11681657 20.11486529 20.11263058 20.1101194 20.1073395 20.10429939 20.10100825 20.09747588 20.09371267 20.0897295 20.08553769 20.08114894 20.07657525 20.07182884 20.06692213 20.06186762 20.05667783 20.05136527 20.04594235 20.04042133 20.03481425 20.0291329 20.02338874 20.01759288 20.01175604 20.00588848 20.]



2.3.4 Equilíbrio com forçantes de calor (concentrado: σ e $\theta \sim L/20$):

$f(x)=Q_+(x)-Q_-(x)$, x no intervalo $[0,0.02]$, $q(x)=0$, $k(x)=3.6$,

$n\acute{o}s=[7,15,31,63]$, $Q_+^0=15000$, $Q_-^0=5000$, $\sigma=0.001$, $\theta=0.001$

Input

```

Q_zero_mais = 15000    # Constante de geraão de calor, em W/m³

Q_zero_menos = 5000    # Constante de resfriamento, em W/m³

L = 0.02               # Comprimento total do bloco do chip, em metros

sigma = 0.001          # Concentraão de distribuião de aquecimento, em metros

theta = 0.001          # Concentraão de distribuião de resfriamento, em metros

def Qmais(x, sigma, Q_zero_mais):
    return Q_zero_mais*np.exp(-((x-(L/2))**2)/(sigma**2))

def Qmenos(x, theta, Q_zero_menos):
    return Q_zero_menos*(np.exp(-(x**2)/(theta**2)) + np.exp(-(x-L)**2)/(theta**2)))

def f(x, sigma, theta, Q_zero_mais, Q_zero_menos):
    return Qmais(x, sigma, Q_zero_mais) - Qmenos(x, theta, Q_zero_menos)

def k(x):
    return 3.6

for n in nos:
    xi, u, u_bar = cf.solveElemFinitos(lambda x: f(x, sigma, theta, Q_zero_mais,
Q_zero_menos),n,l=L,k=k)

```

Output

Para n=7 nós:

(7 nós) Abcissas: [0. 0.0025 0.005 0.0075 0.01 0.0125 0.015 0.0175 0.02]

(7 nós) Vetor de soluções u(x): [20. 20.00853717 20.01776862 20.02699988 20.03414834 20.02699988 20.01776862 20.00853717 20.]

Para n=15 nós:

(15 nós) Abcissas: [0. 0.00125 0.0025 0.00375 0.005 0.00625 0.0075 0.00875 0.01

0.01125 0.0125 0.01375 0.015 0.01625 0.0175 0.01875 0.02]

(15 nós) Vetor de soluções u(x): [20. 20.00394826 20.00853717 20.01315285 20.01776862 20.02238438 20.02699988 20.0315351 20.03414834 20.0315351 20.02699988 20.02238438 20.01776862 20.01315285 20.00853717 20.00394826 20.]

Para n=31 nós:

(31 nós) Abcissas: [0. 0.000625 0.00125 0.001875 0.0025 0.003125 0.00375 0.004375 0.005 0.005625 0.00625 0.006875 0.0075 0.008125 0.00875 0.009375 0.01 0.010625 0.01125 0.011875 0.0125 0.013125 0.01375 0.014375 0.015 0.015625 0.01625 0.016875 0.0175 0.018125 0.01875 0.019375 0.02]

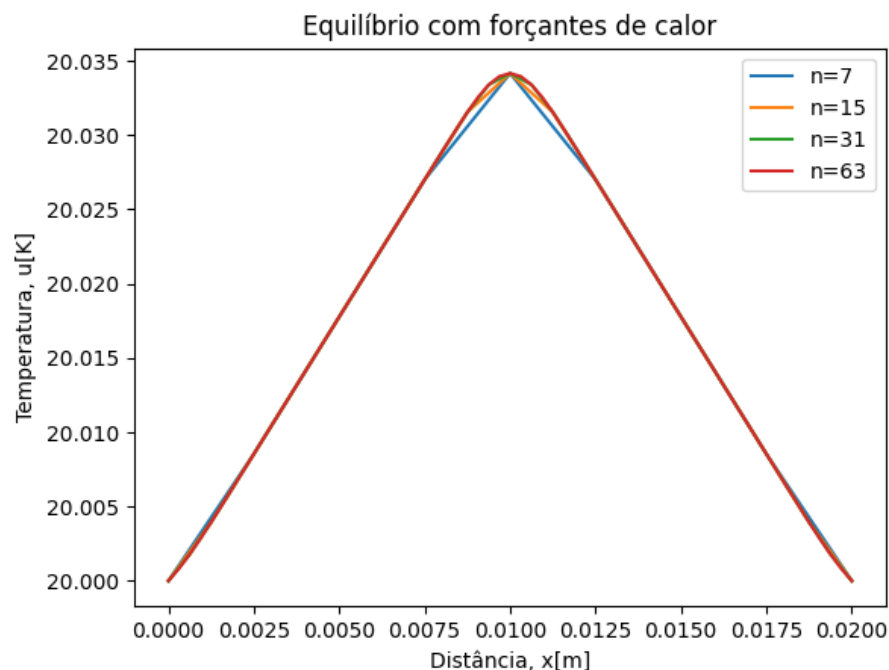
(31 nós) Vetor de soluções u(x): [20. 20.00179348 20.00394826 20.00623136 20.00853717 20.01084497 20.01315285 20.01546073 20.01776862 20.0200765 20.02238438 20.02469226 20.02699988 20.02930155 20.0315351 20.03338366 20.03414834 20.03338366 20.0315351 20.02930155 20.02699988 20.02469226 20.02238438 20.0200765 20.01776862 20.01546073 20.01315285 20.01084497 20.00853717 20.00623136 20.00394826 20.00179348 20.]

Para n=63 nós:

(63 nós) Abcissas: [0. 0.0003125 0.000625 0.0009375 0.00125 0.0015625 0.001875 0.0021875 0.0025 0.0028125 0.003125 0.0034375 0.00375 0.0040625 0.004375 0.0046875 0.005 0.0053125 0.005625 0.0059375 0.00625 0.0065625 0.006875 0.0071875 0.0075 0.0078125 0.008125 0.0084375 0.00875 0.0090625 0.009375 0.0096875 0.01 0.0103125 0.010625 0.0109375 0.01125 0.0115625 0.011875 0.0121875 0.0125 0.0128125 0.013125 0.0134375 0.01375 0.0140625 0.014375 0.0146875 0.015

0.0153125 0.015625 0.0159375 0.01625 0.0165625 0.016875 0.0171875 0.0175 0.0178125 0.018125 0.0184375 0.01875 0.0190625 0.019375 0.0196875 0.02]

(63 nós) Vetor de soluções u(x): [20. 20.00083603 20.00179348 20.00284238 20.00394826 20.00508354 20.00623136 20.00738362 20.00853717 20.00969104 20.01084497 20.01199891 20.01315285 20.01430679 20.01546073 20.01661468 20.01776862 20.01892256 20.0200765 20.02123044 20.02238438 20.02353832 20.02469226 20.02584616 20.02699988 20.02815266 20.02930155 20.03043715 20.0315351 20.03254486 20.03338366 20.03394814 20.03414834 20.03394814 20.03338366 20.03254486 20.0315351 20.03043715 20.02930155 20.02815266 20.02699988 20.02584616 20.02469226 20.02353832 20.02238438 20.02123044 20.0200765 20.01892256 20.01776862 20.01661468 20.01546073 20.01430679 20.01315285 20.01199891 20.01084497 20.00969104 20.00853717 20.00738362 20.00623136 20.00508354 20.00394826 20.00284238 20.00179348 20.00083603 20.]



2.3.5 Demonstração própria baseada na ilustração contida na referência BURDEN, R.; FAIRES, D.J.; BURDEN, A. Numerical Analysis, 10ed, cap.11, p.719-720, 2015.

Equação problema: $-y'' + \pi^2 y = 2\pi^2 \text{sen}(\pi x)$, $x \in (0, 1)$, $y(0) = y(1) = 0$

$q(x) = \pi^2$, $k(x) = 1$, $f(x) = 2\pi^2 \text{sen}(\pi x)$, $n = [3, 5, 7, 9]$ nós, $\bar{u}(x) = \text{sen}(\pi x)$

Input

```
def f(x):
    return 2*(np.pi**2)*np.sin(np.pi*x)

def q(x):
    return np.pi**2

def solucaoExata(x):
    return np.sin(np.pi*x)

for n in nos:
    xi, u, u_bar = cf.solveElemFinitos(f,n,q=q,exata=solucaoExata)
```

Output

Para n=3 nós:

(3 nós) Abcissas: [0. 0.25 0.5 0.75 1.]

(3 nós) Vetor de soluções u(x): [0. 0.72515557 1.02552485 0.72515557 0.]

(3 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [0. 0.70710678 1. 0.70710678 0.]

(3 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0. 0.01804879 0.02552485 0.01804879 0.]

(3 nós) Erro máximo: 0.025525

Para n=5 nós:

(5 nós) Abcissas: [0. 0.16666667 0.33333333 0.5 0.66666667 0.83333333 1.]

(5 nós) Vetor de soluções $u(x)$: [0. 0.50569659 0.8758922 1.01139319 0.8758922 0.50569659 0.]

(5 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [0. 0.5 0.8660254 1. 0.8660254 0.5 0.]

(5 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0. 0.00569659 0.00986679 0.01139319 0.00986679 0.00569659 0.]

(5 nós) Erro máximo: 0.011393

Para n=7 nós:

(7 nós) Abcissas: [0. 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1.]

(7 nós) Vetor de soluções $u(x)$: [0. 0.38513895 0.71164399 0.92980768 1.00641658 0.92980768 0.71164399 0.38513895 0.]

(7 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [0. 0.38268343 0.70710678 0.92387953 1. 0.92387953 0.70710678 0.38268343 0.]

(7 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0. 0.00245552 0.00453721 0.00592815 0.00641658 0.00592815 0.00453721 0.00245552 0.]

(7 nós) Erro máximo: 0.006417

Para n=9 nós:

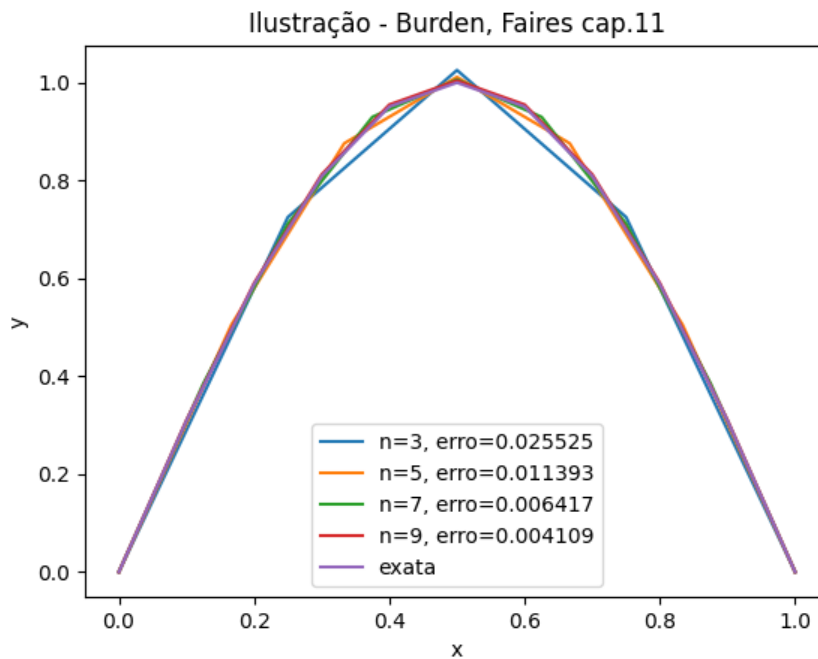
(9 nós) Abcissas: [0. 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.]

(9 nós) Vetor de soluções $u(x)$: [0. 0.31028668 0.59020033 0.81234106 0.95496419 1.00410877 0.95496419 0.81234106 0.59020033 0.31028668 0.]

(9 nós) Vetor de soluções exatas $u_{\text{bar}}(x)$: [0. 0.30901699 0.58778525 0.80901699 0.95105652 1. 0.95105652 0.80901699 0.58778525 0.30901699 0.]

(9 nós) Vetor diferença absoluta $u-u_{\text{bar}}(x)$: [0. 0.00126968 0.00241508 0.00332407 0.00390768 0.00410877 0.00390768 0.00332407 0.00241508 0.00126968 0.]

(9 nós) Erro máximo: 0.004109



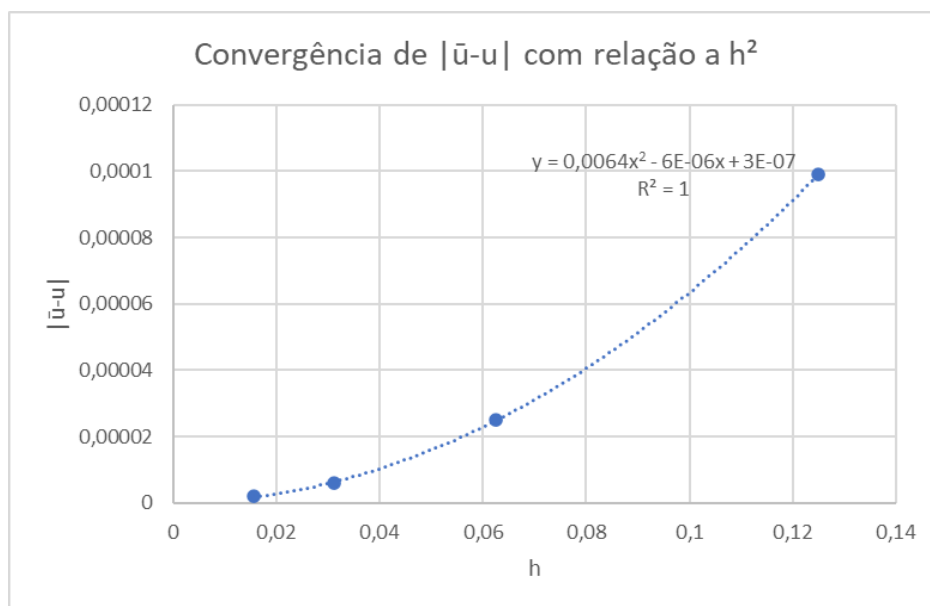
4 RESULTADOS

Sucessos e limitações observadas

Os testes apresentaram resultados consistentes, com melhor aproximação conforme o aumento do número n de nós. Como exemplo, a resolução do teste 2.3.2 deste relatório (referente ao complemento da seção 4.2 do EP3) apresentou a seguinte convergência para $n=[7, 15, 31, 63] \rightarrow ||\bar{u}_n - u|| = [0.000099, 0.000025, 0.000006, 0.000002]$ tal como:

$$||\bar{u}_n - u|| = \max_{i=1, \dots, n} |\bar{u}_n(x_i) - u(x_i)|$$

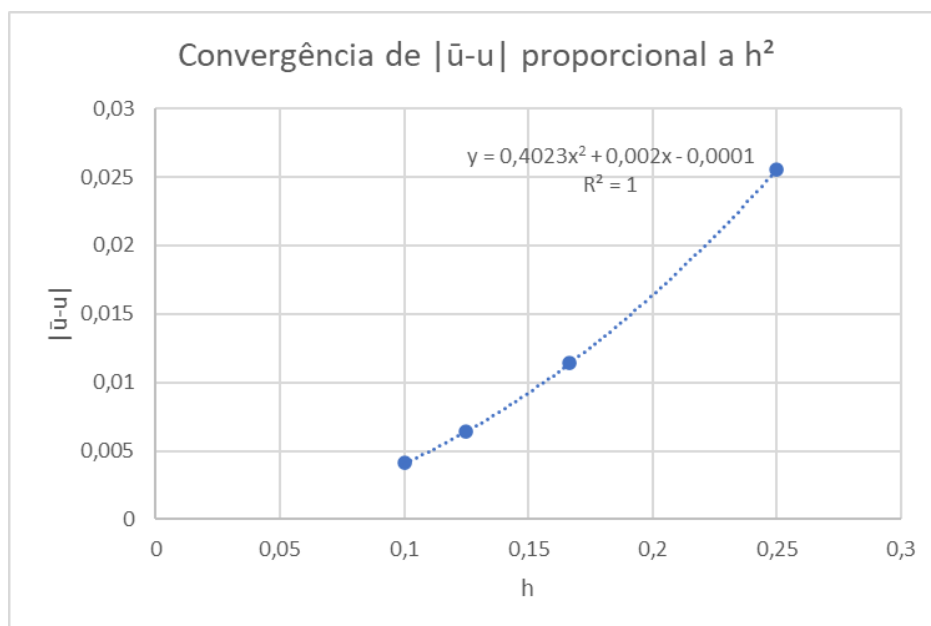
Nesse caso, para o intervalo x em $(0,1)$, obtém-se para $n=[7, 15, 31, 63]$ os valores de $h=[0.125, 0.0625, 0.03125, 0.015625]$, e o erro tende a zero proporcionalmente a h^2 , ilustrado pela seguinte curva resultante dos dados de 2.3.2:



A demonstração da solução da equação problema em 2.3.5 pela função contida no script do EP3, respectivo à demonstração contida na referência citada, apresentou resultados coerentes:

	Resultados da função do EP3	Resultados da literatura
i	$ \bar{u}_n(x_i)-u(x_i) $	$ \phi(x_i) - y(x_i) $
1	0.00126968	0.00127
2	0.00241508	0.00241
3	0.00332407	0.00332
4	0.00390768	0.00390
5	0.00410877	0.00411
6	0.00390768	0.00390
7	0.00332407	0.00332
8	0.00241508	0.00241
9	0.00126968	0.00127

Não somente com os 9 nós, mas também para um número crescente de nós ($n=[3,5,7,9]$) as soluções apresentaram a seguinte convergência para a diferença $||\bar{u}_n-u||=[0.025525, 0.011393, 0.006417, 0.004109]$, que também segue convergência proporcional a h^2 , como ilustra a seguinte curva resultante dos dados de 2.3.5:



A função contida no script também apresentou resultados consistentes para o intervalo de x entre 0 e L , como apresentado no teste 2.3.3 em que $L=0.02\text{m}$ (ou 20mm). Ainda sobre os testes 2.3.2 e 2.3.4, observa-se que os valores para σ e θ nas funções de forçantes de calor são limitadas superiormente por $L/2$ (acima deste valor, a função de distribuição de calor apresenta os mesmos resultados) em que a temperatura segue a distribuição homogênea (ou seja, constante e independente da posição x). Para σ e θ com valores mais próximos de zero, a distribuição de temperatura tende a ser linear, ainda com pico no centro. O aumento demasiado dos fatores Q_+^o ou Q_-^o pode deslocar a curva de temperatura de maneira que as extremidades fiquem mais quentes do que o centro (resfriamento maior e menos concentrado do que o aquecimento), por exemplo.