O principal tema abordado no EP3 foi a aplicação do método dos elementos finitos para modelar o comportamento de difusão térmica e, nesse sentido, criar um algoritmo em C/C++ ou python capaz de resolver essa equação diferenciais. A implementação dos algoritmos foi feita em Python 3.10.4 e com o uso das bibliotecas numpy e math e com o auxílio da IDE VScode.

O problema foi dividido em três partes: O algoritmo de resolução de sistemas lineares tridiagonais, o algoritmo de resolução de integrais e o método dos elementos finitos propriamente dito. A primeira parte e a segunda parte foram retirados dos EP1 e EP2 logo não serão abordados no relatório.

Primeiramente, para implementar o algoritmo dos elementos finitos para a função de forma $(-k(x)u'(x))' + q(x)u(x) = f(x)$, foram utilizadas as equações do livro texto (*Numerical Analysis* de Burden e Faires.) Nesse sentido, para cria a matriz tridiagonal com os produtos internos das φ foram definidas as seguintes integrais:

$(\phi_i, \phi_i) =$

$$\int_0^1 \left\{p(x)[\phi_i'(x)]^2 + q(x)[\phi_i(x)]^2\right\} dx$$

para i de 1 a n

$(\phi_i, \phi_{i-1}) =$

$$\int_0^1 \{p(x)\phi_i'(x)\phi_{i-1}'(x) + q(x)\phi_i(x)\phi_{i-1}(x)\} dx$$

para i de 2 a n

$(\phi_i, \phi_{i+1}) =$

$$\int_0^1 \{p(x)\phi_i'(x)\phi_{i+1}'(x) + q(x)\phi_i(x)\phi_{i+1}(x)\} dx$$

para i de 1 a n - 1

$$b_i = \int_0^1 f(x)\phi_i(x)\, dx = \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} (x - x_{i-1}) f(x)\, dx + \frac{1}{h_i} \int_{x_i}^{x_{i+1}} (x_{i+1} - x) f(x)\, dx,$$

para i de 2 a n

Seguindo o método do livro e desenvolvendo as equações, chegou-se em 6 integrais a serem resolvidas

$$Q_{1,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i) q(x)\, dx, \quad \text{for each } i = 1, 2, \ldots, n-1,$$

$$Q_{2,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2 q(x)\, dx, \quad \text{for each } i = 1, 2, \ldots, n,$$

$$Q_{3,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 q(x)\, dx, \quad \text{for each } i = 1, 2, \ldots, n,$$

$$Q_{4,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x)\, dx, \quad \text{for each } i = 1, 2, \ldots, n+1,$$

$$Q_{5,i} = \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} (x - x_{i-1}) f(x)\, dx, \quad \text{for each } i = 1, 2, \ldots, n,$$

and

$$Q_{6,i} = \frac{1}{h_i} \int_{x_i}^{x_{i+1}} (x_{i+1} - x) f(x)\, dx, \quad \text{for each } i = 1, 2, \ldots, n.$$

H foi definido no enunciado do EP como L/(n + 1).

Para implementar essa equações criou-se a função “Vetor_DiagonalttLU” que recebe os seguintes parâmetros: n (número de pontos), k (função k(x)), q (função q(x)), l (valor de L), f (função f(x)), e mod (modo da função). O modo da função foi definido da seguinte forma: 0 para devolver o vetor da diagonal inferior, 1 para devolver o vetor da diagonal principal, 2 para devolver o vetor da diagonal superior e 3 para devolver o vetor das igualdades do sistema linear proposto (bi). Para isso calculou-se os Q1i, Q2i, Q3i, Q4i ,Q5i, Q6i com o algoritmo da quadratura gaussiana do EP2 de forma separada e guardou seus valores em diferentes vetores. Segue o algoritmo utilizado:

```
for i in range (0,n):
    xi = (i)*h
    xim1 = (i-1)*h
    xiM1 = (i+1)*h
    Q1[i] = (1/h)**2 *  quadratura_gaussiana(10,xi , xiM1,str("({} - x)* (x - {})* {}".format(xiM1,xi,q)))
for i in range (0,n+1):
    xi = (i)*h
    xim1 = (i-1)*h
    Q2[i] = (1/h)**2 * quadratura_gaussiana(10,xim1 , xi,str("(x - {})**2*({})".format(xim1, q)))
for i in range (0,n+1):
    xi = (i)*h
    xiM1 = (i+1)*h
    Q3[i] = (1/h)**2 * quadratura_gaussiana(10,xi , xiM1,str("({} - x)**2*({})".format(xiM1, q)))
for i in range (0,n+2):
    xi = (i)*h
    xim1 = (i-1)*h
    Q4[i] = (1/h )**2 * quadratura_gaussiana(10,xim1 , xi,str(k))
for i in range (0,n+1):
    xi = (i)*h
    xim1 = (i-1)*h
    Q5[i] = (1/h )* quadratura_gaussiana(10,xim1 , xi,str(" (x - {})* ({})".format(xim1,f)))
for i in range (0,n+1):
    xi = (i)*h
    xiM1 = (i+1)*h
    Q6[i] =( 1/h )* quadratura_gaussiana(10,xi , xiM1,str("({} - x)* ({})".format(xiM1,f)))
```

Feito os cálculos dos Qs e seguindo o roteiro do livro texto criou-se o vetores das diagonais:

$$a_{i,i} = Q_{4,i} + Q_{4,i+1} + Q_{2,i} + Q_{3,i}, \quad \text{for each } i = 1, 2, \ldots, n,$$

$$a_{i,i+1} = -Q_{4,i+1} + Q_{1,i}, \quad \text{for each } i = 1, 2, \ldots, n-1,$$

$$a_{i,i-1} = -Q_{4,i} + Q_{1,i-1}, \quad \text{for each } i = 2, 3, \ldots, n,$$

Para isso utilizou-se o seguinte algoritmo:

```
if mod == 0:
    a = np.zeros(n)
    a[0] = 0
    for i in range(1,n):
        a[i] = -Q4[i+1] + Q1[i]
    return a
if mod == 1:
    b = np.zeros(n)
    for i in range(0,n):
        b[i] = Q4[i+1] + Q4[i+2] + Q2[i+1] + Q3[i+1]
    return b
if mod == 2:
    c = np.zeros(n)
    c[c.size-1] = 0
    for i in range(0,n-1):
        c[i] = -Q4[i+2] + Q1[i+1]
    return c
if mod == 3:
    d = np.zeros(n)
    for i in range(0,n):
        d[i] = Q5[i+1] + Q6[i+1]
    return d
```

Montada a matriz dos produtos internos, usa-se o algoritmos de solução de sistemas lineares tridimensionais. observa-se que no código o primeiro valor do vetor de “a” e o último valor do vetor de “c” são zeros para que, pela forma com que foi implementado no o método de resolução de sistemas lineares no EP1, esse sistema seja não cíclico.

Para o caso não homogêneo tem-se a expressão que reduz o caso para um homogêneo dada por:

$(-k(x)\,v'(x))' + q(x)\,v(x) = \left(f(x) + (b - a)k'(x) - q(x)(a + (b - a)x) \right)$

Nesse sentido cria-se f_ por meio do seguinte algoritmo

```
def n_homogenio(k,q,f,a,b):
    f_= str("{} + ({} - {})*{} - ({} * ({} + ({} - {})*x))".format(f,b,a,k,q,a,b,a))
    return f_
```

onde u(0)=a, u(1) = b, f = f(x), q = q(x), k = k(x)

dessa forma “n_homogenio” se torna parâmetro de “Vetor_DiagonalttLU” e resolvendo assim um problema homogenio.

Validação:

Para o primeiro exemplo com $q(x) \equiv 0$, no intervalo [0, L] e condições de contorno $u(0) = a$, $u(L) = b$. No intervalo [0, 1] onde $k(x) = 1$, $q(x) = 0$, $f(x) = 12x(1 - x) - 2$, com condições de contorno homogêneas. Neste caso, a solução é exata é u(x) = x^2 (1 − x)^2.

solução como o método para n = 7 , 15 , 31, 63 respectivamente:

7-

```
Solucao pelo método
[0.01196289 0.03515625 0.05493164 0.0625     0.05493164 0.03515625
 0.01196289]
Solucao exata:
[0.01196289 0.03515625 0.05493164 0.0625     0.05493164 0.03515625
 0.01196289]
```

15-

```
Solucao pelo método
[0.00343323 0.01196289 0.02320862 0.03515625 0.04615784 0.05493164
 0.06056213 0.0625     0.06056213 0.05493164 0.04615784 0.03515625
 0.02320862 0.01196289 0.00343323]
Solucao exata:
[0.00343323 0.01196289 0.02320862 0.03515625 0.04615784 0.05493164
 0.06056213 0.0625     0.06056213 0.05493164 0.04615784 0.03515625
 0.02320862 0.01196289 0.00343323]
```

31-

```
Solucao pelo método
[0.00091648 0.00343323 0.00721836 0.01196289 0.01738071 0.02320862
 0.02920628 0.03515625 0.04086399 0.04615784 0.05088902 0.05493164
 0.05818272 0.06056213 0.06201267 0.0625     0.06201267 0.06056213
 0.05818272 0.05493164 0.05088902 0.04615784 0.04086399 0.03515625
 0.02920628 0.02320862 0.01738071 0.01196289 0.00721836 0.00343323
 0.00091648]
Solucao exata:
[0.00091648 0.00343323 0.00721836 0.01196289 0.01738071 0.02320862
 0.02920628 0.03515625 0.04086399 0.04615784 0.05088902 0.05493164
 0.05818272 0.06056213 0.06201267 0.0625     0.06201267 0.06056213
 0.05818272 0.05493164 0.05088902 0.04615784 0.04086399 0.03515625
 0.02920628 0.02320862 0.01738071 0.01196289 0.00721836 0.00343323
 0.00091648]
```

63-

```
Solucao pelo método
[0.00023657 0.00091648 0.0019961  0.00343323 0.00518709 0.00721836
 0.00948912 0.01196289 0.01460463 0.01738071 0.02025896 0.02320862
 0.02620035 0.02920628 0.03219992 0.03515625 0.03805166 0.04086399
 0.04357249 0.04615784 0.04860216 0.05088902 0.05300337 0.05493164
 0.05666167 0.05818272 0.0594855  0.06056213 0.0614062  0.06201267
 0.06237799 0.0625     0.06237799 0.06201267 0.0614062  0.06056213
 0.0594855  0.05818272 0.05666167 0.05493164 0.05300337 0.05088902
 0.04860216 0.04615784 0.04357249 0.04086399 0.03805166 0.03515625
 0.03219992 0.02920628 0.02620035 0.02320862 0.02025896 0.01738071
 0.01460463 0.01196289 0.00948912 0.00721836 0.00518709 0.00343323
 0.0019961  0.00091648 0.00023657]
Solucao exata:
[0.00023657 0.00091648 0.0019961  0.00343323 0.00518709 0.00721836
 0.00948912 0.01196289 0.01460463 0.01738071 0.02025896 0.02320862
 0.02620035 0.02920628 0.03219992 0.03515625 0.03805166 0.04086399
 0.04357249 0.04615784 0.04860216 0.05088902 0.05300337 0.05493164
 0.05666167 0.05818272 0.0594855  0.06056213 0.0614062  0.06201267
 0.06237799 0.0625     0.06237799 0.06201267 0.0614062  0.06056213
 0.0594855  0.05818272 0.05666167 0.05493164 0.05300337 0.05088902
 0.04860216 0.04615784 0.04357249 0.04086399 0.03805166 0.03515625
 0.03219992 0.02920628 0.02620035 0.02320862 0.02025896 0.01738071
 0.01460463 0.01196289 0.00948912 0.00721836 0.00518709 0.00343323
 0.0019961  0.00091648 0.00023657]
```

Gráfico dos pontos pelo método

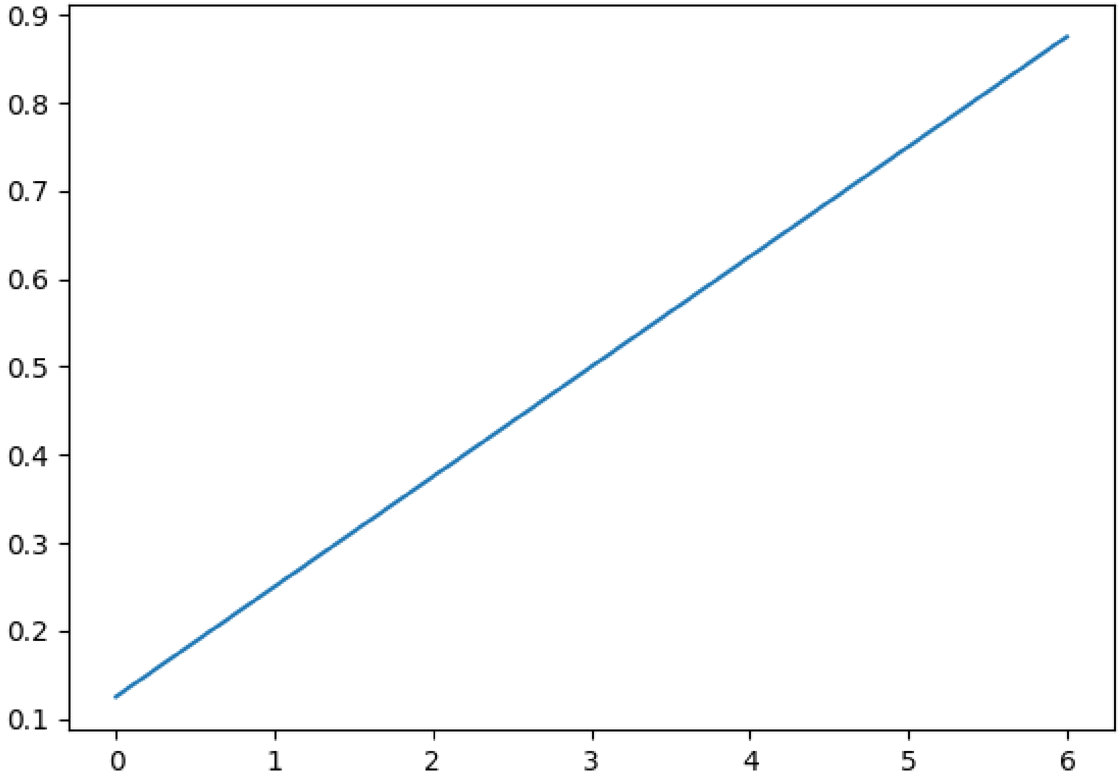
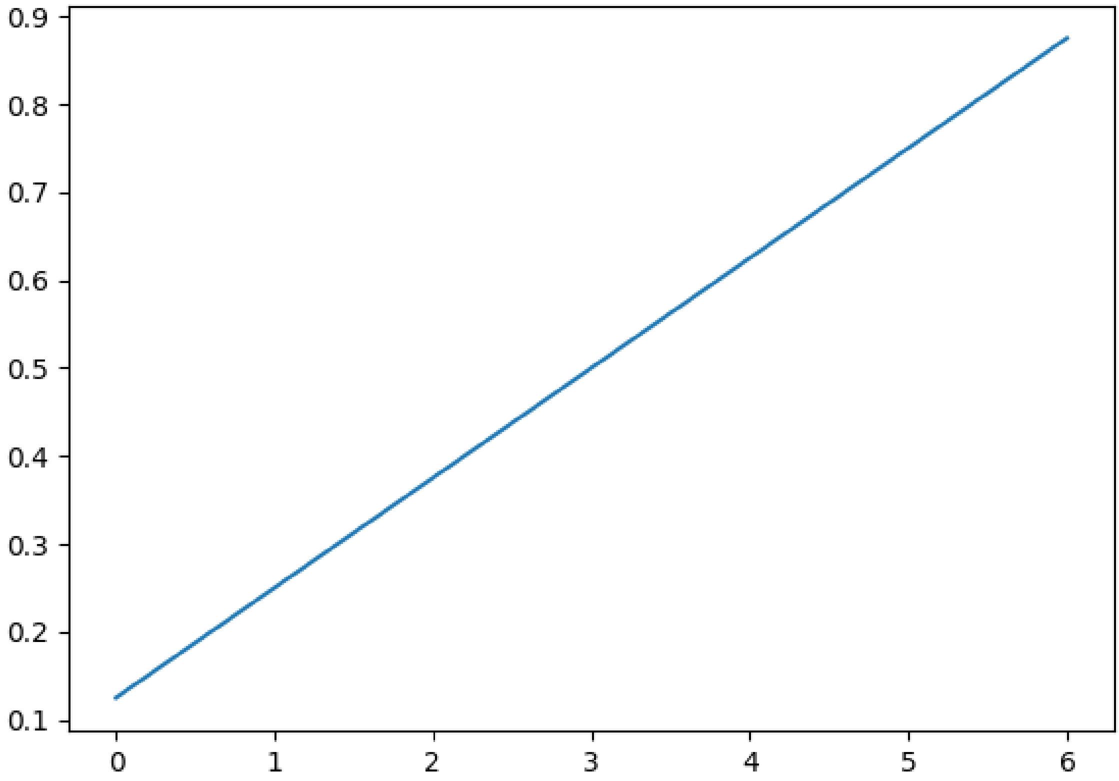


Gráfico dos pontos pela solução exata



Validação 2:

Teste do programa com o exemplo no intervalo [0, 1] onde k(x) = e x , q(x) = 0, f(x) = e x + 1, com condições de contorno homogeias. Neste caso, a solução exata é u(x) = (x−1)(e^ −x −1). com n = 7, 15, 31 e 63

n = 7

```
Solucao pelo método
[0.10276066 0.16581358 0.19534565 0.19663803 0.1741936  0.13184695
 0.07285922]
Solucao exata:
[0.10281521 0.16589941 0.1954442  0.19673467 0.17427696 0.13190836
 0.07289225]
```

n = 15

```
Solucao pelo método
[0.05679262 0.10280156 0.13889563 0.16587794 0.18449068 0.19541955
 0.19929789 0.1967105  0.18819719 0.17425611 0.15534679 0.131893
 0.1042854  0.07288399 0.0380204 ]
Solucao exata:
[0.05680025 0.10281521 0.13891384 0.16589941 0.18451426 0.1954442
 0.1993227  0.19673467 0.18822001 0.17427696 0.15536513 0.13190836
 0.10429738 0.07289225 0.03802465]
```

n = 31

```
Solucao pelo método
[0.0298043  0.05679835 0.08109728 0.1028118  0.12204836 0.13890929
 0.15349299 0.16589404 0.17620337 0.18450836 0.19089301 0.19543804
 0.19822102 0.1993165  0.1987961  0.19672863 0.19318019 0.18821431
 0.18189196 0.17427175 0.16540993 0.15536055 0.14417547 0.13190452
 0.11859552 0.10429438 0.08904517 0.07289018 0.05587    0.03802359
 0.0193883 ]
Solucao exata:
[0.0298053  0.05680025 0.08109998 0.10281521 0.12205238 0.13891384
 0.15349799 0.16589941 0.17620904 0.18451426 0.19089907 0.1954442
 0.19822723 0.1993227  0.19880224 0.19673467 0.19318609 0.18822001
 0.18189744 0.17427696 0.16541485 0.15536513 0.1441797  0.13190836
 0.11859895 0.10429738 0.08904771 0.07289225 0.05587158 0.03802465
 0.01938884]
```

n= 63

```
Solucao pelo método
[0.01526119 0.02980505 0.0436464  0.05679978 0.06927942 0.08109931
 0.09227314 0.10281436 0.11273613 0.12205137 0.13077276 0.1389127
 0.14648338 0.15349674 0.15996448 0.16589807 0.17130877 0.17620762
 0.18060542 0.18451278 0.1879401  0.19089755 0.19339514 0.19544266
 0.1970497  0.19822568 0.19897981 0.19932115 0.19925855 0.19880071
 0.19795613 0.19673316 0.19513998 0.19318462 0.19087492 0.18821859
 0.18522317 0.18189607 0.17824453 0.17427566 0.16999642 0.16541362
 0.16053396 0.15536399 0.14991012 0.14417864 0.13817572 0.1319074
 0.12537959 0.11859809 0.11156859 0.10429663 0.09678768 0.08904708
 0.08108005 0.07289173 0.06448714 0.05587118 0.04704869 0.03802438
 0.02880288 0.0193887  0.0097863 ]
Solucao exata:
[0.01526132 0.0298053  0.04364677 0.05680025 0.06928    0.08109998
 0.09227391 0.10281521 0.11273706 0.12205238 0.13077383 0.13891384
 0.14648458 0.15349799 0.15996577 0.16589941 0.17131016 0.17620904
 0.18060687 0.18451426 0.18794159 0.19089907 0.19339667 0.1954442
 0.19705125 0.19822723 0.19898137 0.1993227  0.1992601  0.19880224
 0.19795765 0.19673467 0.19514148 0.19318609 0.19087637 0.18822001
 0.18522457 0.18189744 0.17824587 0.17427696 0.16999768 0.16541485
 0.16053515 0.15536513 0.14991122 0.1441797  0.13817673 0.13190836
 0.1253805  0.11859895 0.11156939 0.10429738 0.09678837 0.08904771
 0.08108063 0.07289225 0.06448759 0.05587158 0.04704902 0.03802465
 0.02880308 0.01938884 0.00978636]
```

gráfico dos ponto gerados pelo método para n = 63

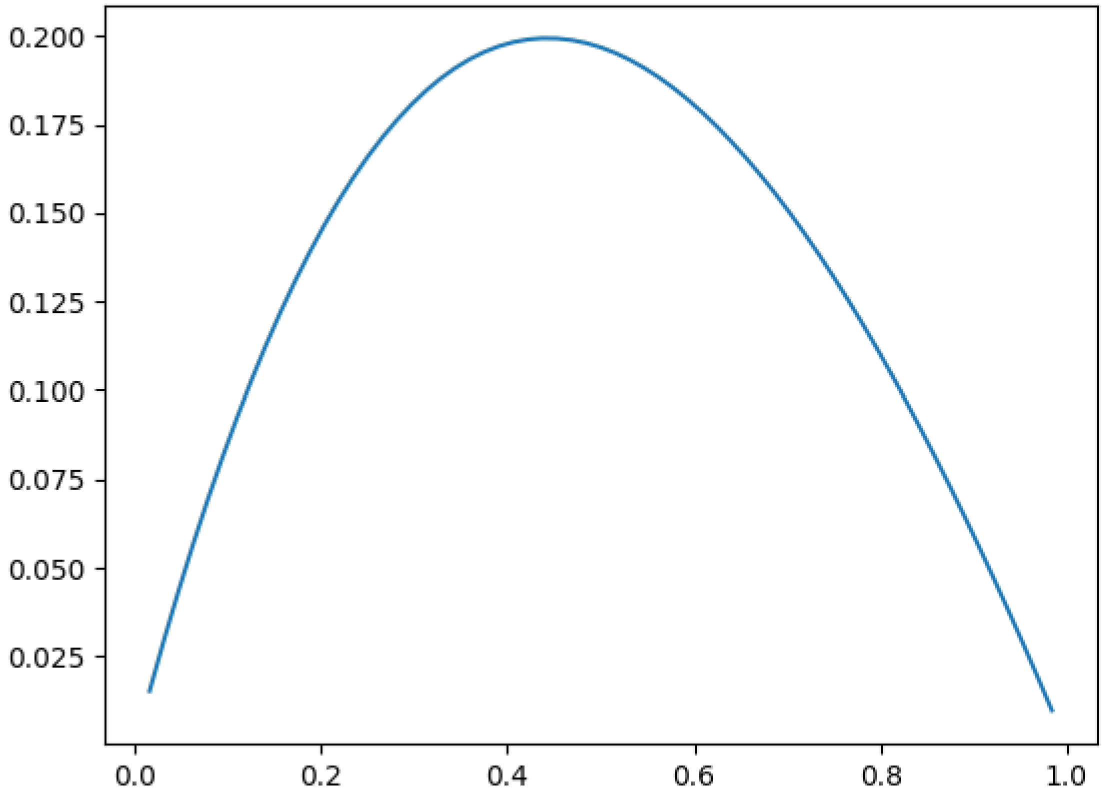
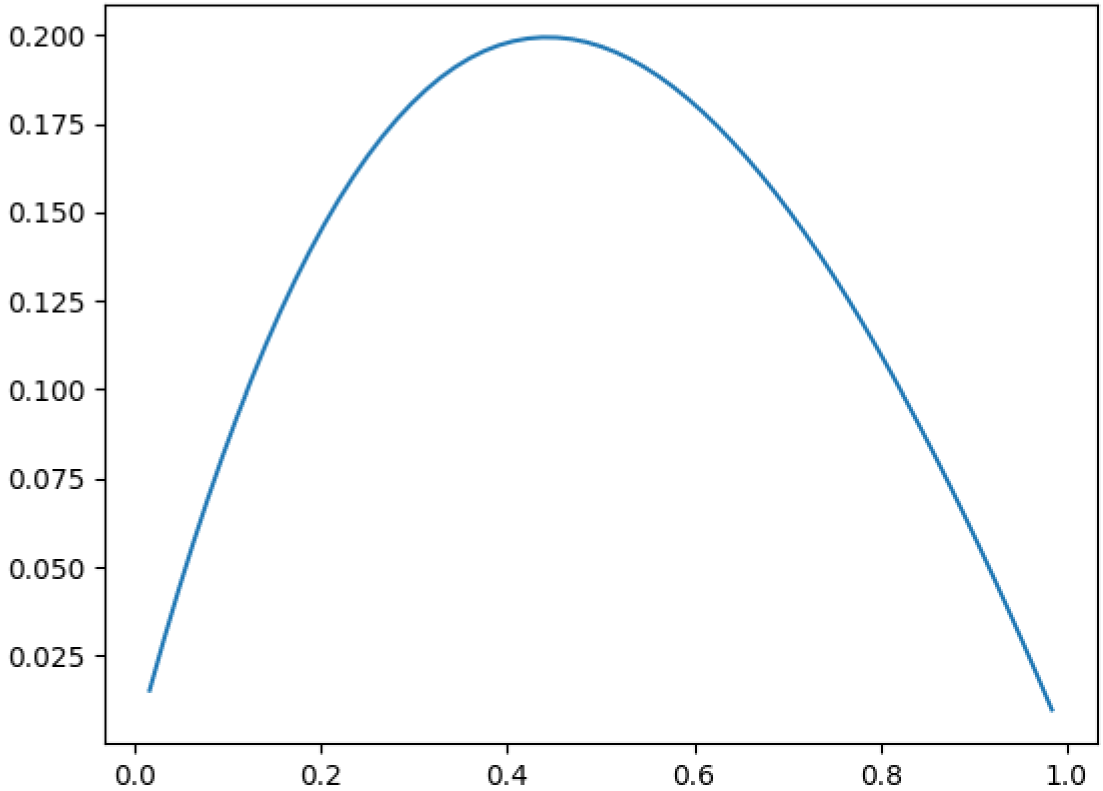


gráfico dos pontos para solução exata para n = 63



Para o equilíbrio com forças de calor, opção 4, modelamos as situações de equilíbrio. No programa é possível alterar os parâmetros pela interface no terminal fixamos a constante k em 30. Divide-se em três situações: resfriamento constante, resfriamento análogo ao aquecimento e por fim resfriamento mais intenso nas bordas. Segue alguns exemplos para

Q+ = 100

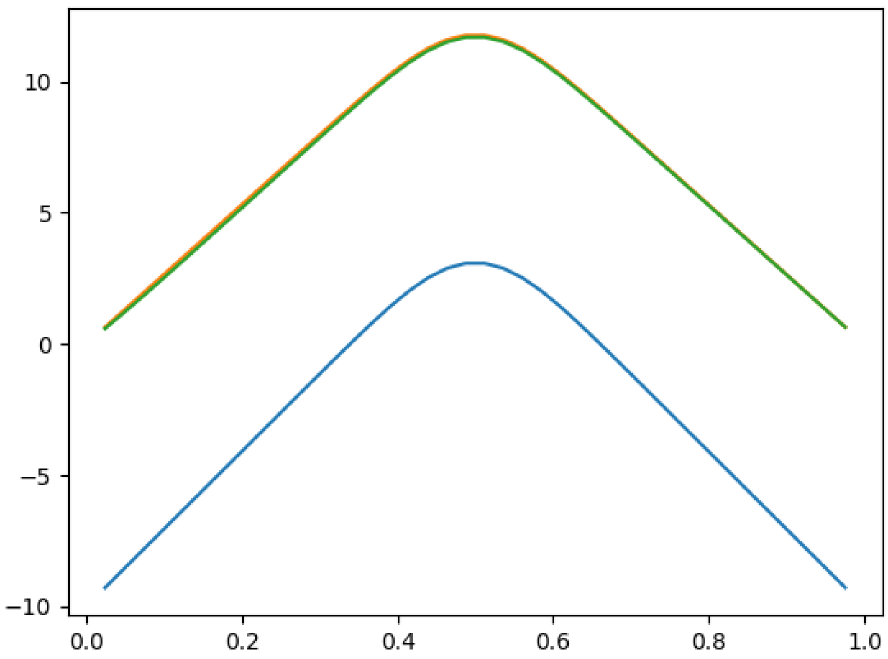
sigma = 1

teta = 1

L = 10

Q_ = 10

```
--------------------------------------------------
1 - Resfriamento constate
 [-9.51572299 -9.03144598 -8.54716897 -8.06289197 -7.57861496 -7.09433795
 -6.61006096 -6.125784   -5.64150717 -5.15723073 -4.67295553 -4.18868388
 -3.70442202 -3.22018573 -2.73601265 -2.25198783 -1.7682927  -1.28529261
 -0.80368181 -0.32470388  0.14954331  0.61573136  1.06890237  1.50213923
  1.90643287  2.27088282  2.58332885  2.83142065  3.00400623  3.09259691
  3.09259691  3.00400623  2.83142065  2.58332885  2.27088282  1.90643287
  1.50213923  1.06890237  0.61573136  0.14954331 -0.32470388 -0.80368181
 -1.28529261 -1.7682927  -2.25198783 -2.73601265 -3.22018573 -3.70442202
 -4.18868388 -4.67295553 -5.15723073 -5.64150717 -6.125784   -6.61006096
 -7.09433795 -7.57861496 -8.06289197 -8.54716897 -9.03144598 -9.51572299]
2 - Resfriamento analogo ao aquecimento
 [ 0.43584931  0.87169862  1.30754792  1.74339723  2.17924654  2.61509584
  3.05094514  3.4867944   3.92264355  4.35849234  4.79434003  5.23018451
  5.66602018  6.10183285  6.53758862  6.97321095  7.40853657  7.84323665
  8.27668637  8.7077665   9.13458898  9.55415822  9.96201213 10.35192531
 10.71578958 11.04379454 11.32499597 11.54827858 11.7036056  11.78333722
 11.78333722 11.7036056  11.54827858 11.32499597 11.04379454 10.71578958
 10.35192531  9.96201213  9.55415822  9.13458898  8.7077665   8.27668637
  7.84323665  7.40853657  6.97321095  6.53758862  6.10183285  5.66602018
  5.23018451  4.79434003  4.35849234  3.92264355  3.4867944   3.05094514
  2.61509584  2.17924654  1.74339723  1.30754792  0.87169862  0.43584931]
3 - Resfriamento mais intenso nos extremos
 [ 0.39461298  0.7979098   1.20922359  1.6275547   2.05170951  2.48044668
  2.91260242  3.34717617  3.78337142  4.2205974   4.65844385  5.09664238
  5.53502463  5.9734828   6.41193226  6.85027054  7.28832186  7.72575168
  8.16193273  8.59574479  9.02529941  9.44760086  9.858187   10.25083241
 10.61742893 10.94816613 11.23209979 11.45811465 11.61617391 11.69863776
 11.70137    11.62437063 11.47177585 11.25122548 10.97275629 10.64748357
 10.28635154  9.8991706   9.49404893  9.07721193  8.6531217   8.2247738
  7.79405632  7.36208848  6.9294951   6.49660501  6.06358148  5.63050105
  5.19739763  4.76428538  4.33116994  3.89805339  3.46493648  3.03181945
  2.5987024   2.16558533  1.73246827  1.2993512   0.86623413  0.43311707]
```



curva azul: resfriamento constante
curva laranja: resfriamento análogo
curva verde: resfriamento maior nas bordas