Escola Politécnica
Universidade de São Paulo

MAP3121 - Métodos Numéricos e Aplicações

EP - Parte 2
Um problema inverso para a equação do calor

Isadora Bisognin (10771347)
Luísa Heise (10705784)

Julho de 2020

Resumo

Os métodos inversos são aqueles que, possuindo a descrição de um problema, tentam chegar em suas condições de contorno. Isso em geral, não é uma tarefa fácil, já que, muitas vezes, um mesmo resultado admite várias condições de contorno possíveis. Neste trabalho buscou-se determinar a intensidade de uma fonte $f(t, x)$ de uma distribuição de calor, tendo essa distribuição num dado instante T e sabendo a intensidade de forçantes pontuais, além de outras condições de contorno. Para tal, utilizou-se métodos diretos para se encontrar várias soluções com cada um dos forçantes pontuais e depois foi feito um ajuste com método dos mínimos quadrados para encontrar os coeficientes dessa fonte. O método apresentou comportamento adequado e os valores encontrados de a foram compatíveis com expectativas teóricas.

Sumário

1 Introdução

1.1 Descrição do Problema

Este relatório visa apresentar e analisar os resultados obtidos a partir da resolução de um problema indireto da equação do calor por meio da aplicação de algoritmos em Python feitos pelo grupo.

Tal problema indireto da equação do calor trata-se de obter a intensidade das fontes de calor (causa) aplicadas em cada distância x de uma barra partindo da distribuição final de temperatura no instante T na respectiva barra (efeito).

2 Conceitos Teóricos

Primeiramente, é necessário definir como no enunciado as funções e a equação diferencial que aparecerão no problema, sendo u(T, x) a solução da equação de calor na barra em função do tempo t e da distância x dadas as condições:

$$u_t(t,x) = u_{xx}(t,x) + f(t,x), t \in [0,T], x \in [0,1] \tag{1}$$

$$u_0(x) = u(0,x), x \in [0,1] \tag{2}$$

$$g_1(t) = u(t,0), t \in [0,T] \tag{3}$$

$$g_2(t) = u(t,1), t \in [0,T] \tag{4}$$

Sendo que as seguintes condições de fronteira se aplicam a u(T, x):

$$u_0(x) = 0 \tag{5}$$

$$g_1(t) = 0 \tag{6}$$

$$g_2(t) = 0 \tag{7}$$

Além disso, define-se a fonte como:

$$f(t,x) = r(t) \sum_{k=1}^{nf} a_k b_h^k \tag{8}$$

Sendo que o forçante pontual é:

$$g_h^k(x) = \frac{1}{h}, x \in [p_k - \frac{h}{2}, p_k + \frac{h}{2}] \tag{9}$$

$$g_h^k(x) = 0, x \notin [p_k - \frac{h}{2}, p_k + \frac{h}{2}] \tag{10}$$

A solução u(T, x) verifica a seguinte igualdade:

$$u_T(x) = \sum_{k=1}^{nf} a_k u_k(T,x) \tag{11}$$

E as intensidades a_k são obtidas escolhendo o valor que minimiza a expressão, sendo este um problema de mínimos quadrados (MMQ):

$$E_2 = \sqrt{\Delta x \sum_{i=1}^{N-1} (u_T(x_i) - \sum_{k=1}^{nf} a_k u_k(T,x_i))^2} \tag{12}$$

2.1 Obtenção de $u_k(T, x)$

Para resolver as equação diferencial (1-4) apresentada anteriormente e com isso obter o conjunto de soluções, forma-se um sistema de equações na forma matricial e este deve ser resolvido, no caso desse trabalho, utilizando o método de Crank-Nicolson.

2.2 Método de Crank-Nicolson

O método de Crank-Nicolson estabelece a seguinte relação entre diversos valores em determinados valores k e i:

$$u_i^{k+1} = u_i^k + \frac{\lambda}{2}((u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}) + (u_{i-1}^k - 2u_i^k + u_{i+1}^k)) + \frac{\Delta t}{2}(f(x_i, t_k) + f(x_i, t_{k+1})) \tag{13}$$

Isolando u_i^{k+1} no lado esquerdo da equação, obtem-se o lado direito (b) do sistema de equações Ax = b.

2.2.0.1 O sistema de Crank-Nicolson

O lado **esquerdo** do sistema pode ser definido como:

$$A = \begin{bmatrix} 1+\lambda & -\frac{\lambda}{2} & 0 & ... & 0 \\ -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} & ... & ... \\ 0 & ... & ... & ... & 0 \\ ... & ... & -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} \\ 0 & ... & 0 & -\frac{\lambda}{2} & 1+\lambda \end{bmatrix} \tag{14}$$

2.2.0.2 O sistema de Crank-Nicolson

O lado **direito** do sistema pode ser definido como:

$$b^{k+1} = \begin{bmatrix} \mathrm{g}_1(t_{k+1})(\frac{\lambda}{2}) + u_1^k + \frac{\lambda}{2}(g_1(t_k) - 2u_1^k + u_2^k) + \frac{\Delta t}{2}(f(x_1, t_k) + f(x_1, t_{k+1})) \\ ... \\ \mathrm{u}_i^k + \frac{\lambda}{2}(u_{i-1}^k - 2u_i^k + u_{i+1}^k) + \frac{\Delta t}{2}(f(x_i, t_k) + f(x_i, t_{k+1})) \\ ... \\ \mathrm{g}_2(t_{k+1})(\frac{\lambda}{2}) + u_{N-1}^k + \frac{\lambda}{2}(u_{N-2}^k - 2u_{N-1}^k + g_2(t_k)) + \frac{\Delta t}{2}(f(x_{N-1}, t_k) + f(x_{N-1}, t_{k+1})) \end{bmatrix} \tag{15}$$

O erro de truncamento do método de Crank-Nicolson:

$$\tau_i^k = \frac{u_i^{k+1} - u_i^k}{\Delta t} - \frac{1}{2(\Delta x)^2}((u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}) + (u_{i-1}^k - 2u_i^k + u_{i+1}^k)) - \frac{1}{2}(f(x_i, t_k) + f(x_i, t_{k+1})) \tag{16}$$

2.2.0.3 A convergência do método implícito de Crank-Nicolson

No método implícito de Crank-Nicolson, a convergência é proporcional a Δt^2. Se o intervalo de tempo for divido por 2, portanto, o erro será cortado em 4.

2.3 Decomposição LDL^t

Resolver o sistema de equações na forma de matriz Ax = b não é computacionalmente eficiente, tendo um custo computacional alto, assim, resolvemos o sistema por meio de decomposição $LDLt$ da seguinte forma:

1. L e sua transposta são matrizes triangulares cujos elementos da diagonal valem 1 e D é uma matriz diagonal e a matriz A é tridiagonal simétrica. Todas elas são quadradas. Assim, temos, usando matrizes de dimensão 3x3 como exemplo:
   $A = LDL^t$

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{12} & a_{22} & a_{23} \\ 0 & a_{23} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{12} & a_{22} & a_{23} \\ 0 & a_{23} & a_{33} \end{bmatrix} = \begin{bmatrix} d_1 & d_1 l_{21} & d_1 l_{31} \\ d_1 l_{21} & d_2 + d_1 l_{21}^2 & d_2 l_{32} + d_1 l_{21} l_{31} \\ d_1 l_{31} & d_1 l_{21} l_{31} + d_2 l_{32} & d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \end{bmatrix}$$

2. Resolvendo a igualdade de elementos apresentada acima, obtem-se as 3 matrizes.

3. Com isso, é possível resolver 3 sistemas de matrizes muito mais simples, considerando Ax = b o sistema original:
   Ly = b $\rightarrow Dz = y \rightarrow L^t x = z$
   Com isso, consegue-se a solução final x.

2.4 O sistema final

Após a obtenção dos valores $u_k(T, x)$ por meio de Crank-Nicolson e dos valores a_k por meio da condição de minimização de E_2, é possível montar o sistema cujas soluções são as intensidades a_{nf} das fontes:

$$\begin{bmatrix} < u_1, u_1 > & < u_2, u_1 > & ... & < u_{nf}, u_1 > \\ < u_1, u_2 > & < u_2, u_2 > & ... & < u_{nf}, u_2 > \\ ... & ... & ... & ... \\ < u_1, u_{nf} > & < u_2, u_{nf} > & ... & < u_{nf}, u_{nf} > \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_{nf} \end{bmatrix} = \begin{bmatrix} < u_T, u_1 > \\ < u_T, u_2 > \\ ... \\ < u_T, u_{nf} > \end{bmatrix}$$

Tal que o produto escalar $< u, v >$ é definido como: $< u, v >= \sum_{i=1}^{N-1} u(x_i)v(x_i)$

3 Metodologia

Foram implementadas rotinas de código em `Python 3.6` para a realização das simulações numéricas.

A função que fornece o lado A, de Ax=b, de Crank-Nicolson e a que fornece a solução do sistema com a matriz diagonal estão contidas em `ex2.py`, que é o programa que foi utilizado na tarefa b do exercício-programa 1, que está dentro da pasta `EP1`, a qual contem também o `ex1.py`, que é o programa utilizado na tarefa a do exercício-programa 1 e que contem algumas funções importadas e utilizadas pelo programa da tarefa b.

Todas as outras funções e sua documentação mais o programa principal encontram-se em `EP2-functions.py`.

As instruções de execução estão no arquivo `readme.txt`.

4 Resultados e Discussão

4.1 Teste A

O resultado do teste A foi congruente com o valor esperado apontado no enunciado, com a_1 valendo 7, como pode-se ver a seguir:

```
Qual item deseja fazer a simulação (a, b, c ou d)? a
Digite o N: 128
N = 128
a(0) = 7.0
```

(a) Teste A

4.2 Teste B

O resultado do teste B foi congruente com o valor esperado, em que as intensidades a_k têm valores que podem ser aproximados a números com 2 algarismos significativos com um erro baixo, da ordem de 10^{-14}, como pode-se ver na tabela a seguir:

k	a_k
0	2.2999999999999954
1	3.700000000000003
2	0.29999999999998916
3	4.200000000000013

4.3 Teste C

4.3.1 Comparação de $u_T(x)$

Primeiramente, comparam-se os valores de $u_T(x)$ reais e os utilizados na simulação do teste C nos seguintes gráficos:

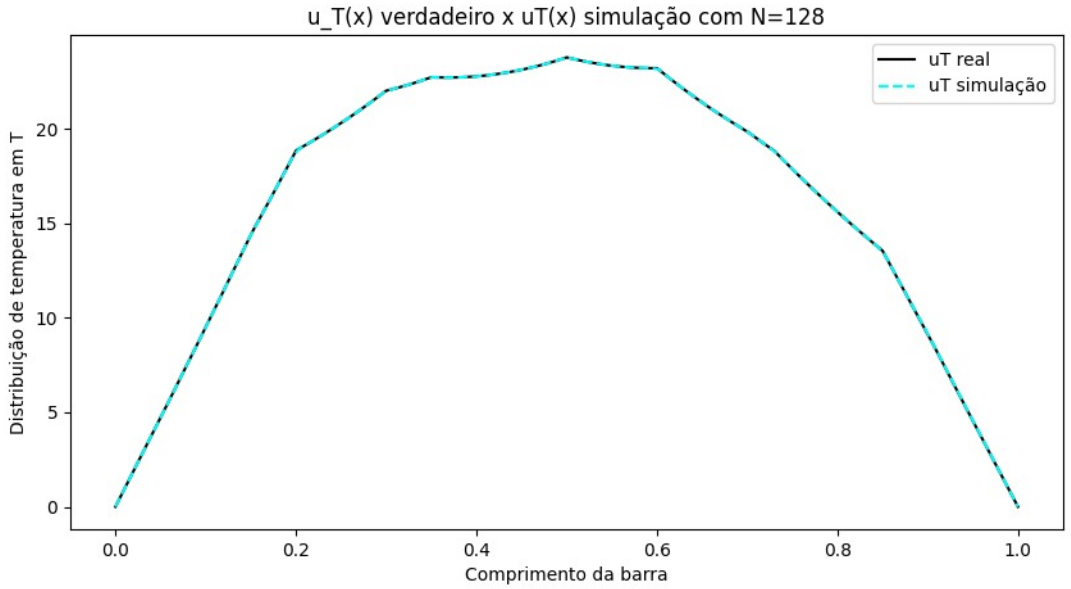

Figura 2: Teste C — N = 128

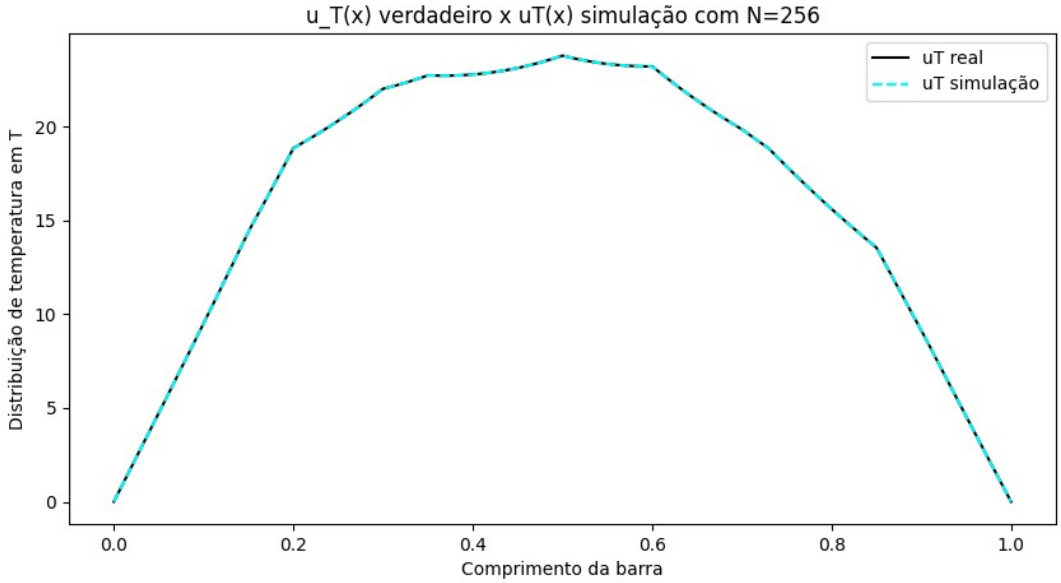

Figura 3: Teste C — N = 256

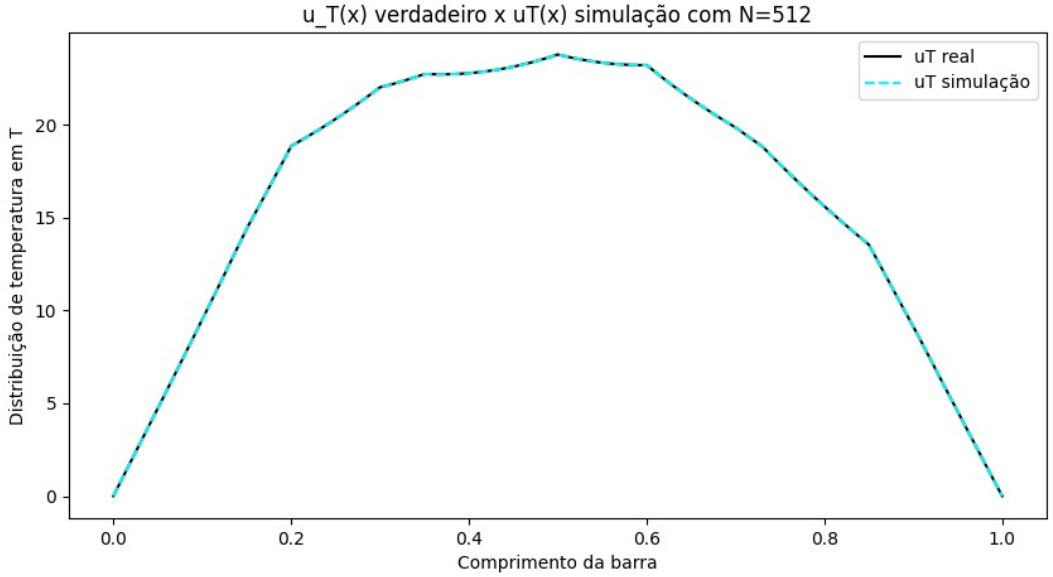

Figura 4: Teste C — N = 512

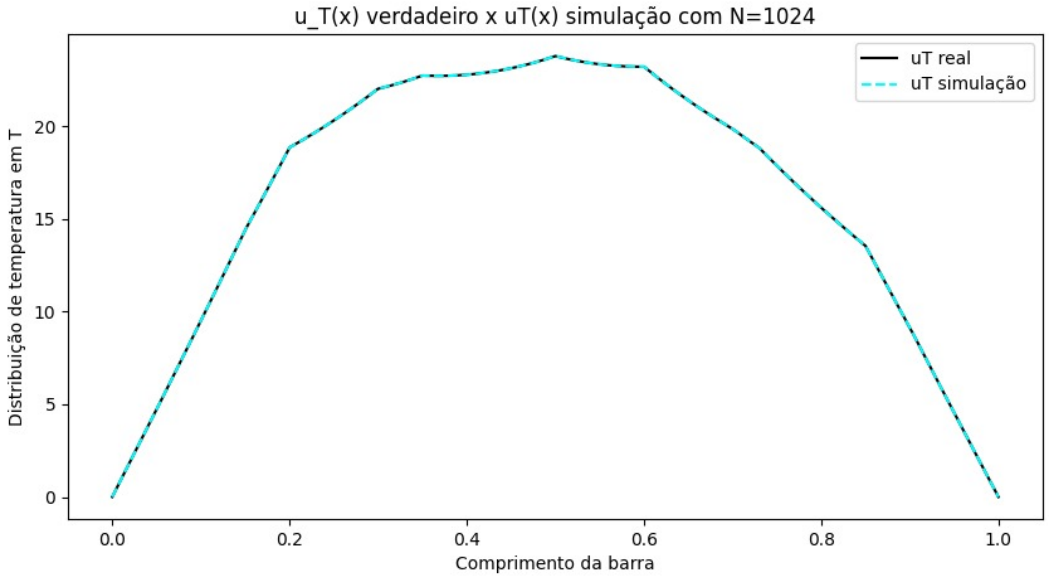

Figura 5: Teste C — N = 1024

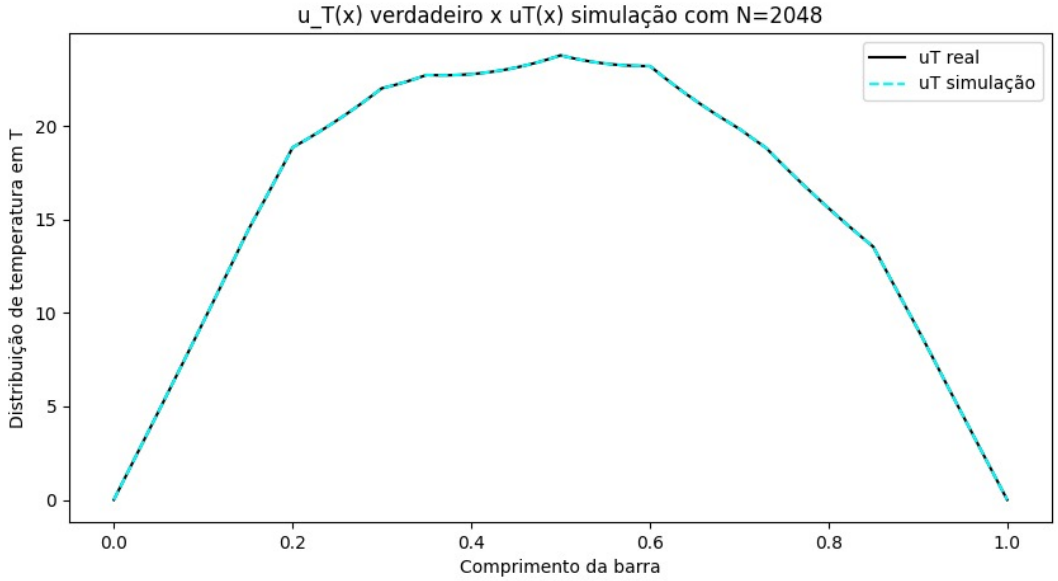

Figura 6: Teste C — N = 2048

Percebe-se uma extrema similaridade entre os valores e comportamento da função $u_T(x)$, sendo que é possível dizer que elas estão basicamente sobrepostas uma a outra no gráfico. Além disso, esse comportamento pode ser observado em todos os 5 valores de N (128, 256, 512, 1024, 2048).

4.3.2 Intensidades a_k e o erro quadrático E_2

Os valores de a_k e do erro quadrático E_2 do teste C estão listados nas figuras abaixo:

```
Qual item deseja fazer a simulação (a, b, c ou d)? c
Digite o N: 128
Por favor aguarde...
N = 128
a(0) = 1.2091231792047843
a(1) = 4.839258715746006
a(2) = 1.8872408557579199
a(3) = 1.583399931863081
a(4) = 2.214504046288363
a(5) = 3.1212947787772
a(6) = 0.37734028636697303
a(7) = 1.4923482881276282
a(8) = 3.9751388015978537
a(9) = 0.4041451536490676
O erro E2 = 0.024453403799692817
```

(a) $TesteC|N = 128$

```
Qual item deseja fazer a simulação (a, b, c ou d)? c
Digite o N: 256
Por favor aguarde...
N = 256
a(0) = 0.9045010343179207
a(1) = 5.077572635561182
a(2) = 2.1008535954785295
a(3) = 1.4141556850887298
a(4) = 2.2292450130538413
a(5) = 3.1046138569902695
a(6) = 0.5094525973942989
a(7) = 1.386508790454915
a(8) = 3.9498786461535587
a(9) = 0.4148931283298642 5
O erro E2 = 0.012363464048873776
```

(b) $TesteC|N = 256$

```
Qual item deseja fazer a simulação (a, b, c ou d)? c
Digite o N: 512
Por favor aguarde...
N = 512
a(0) = 0.9286883784933551
a(1) = 5.053707844480442
a(2) = 2.0437010489033334
a(3) = 1.4676706728643047
a(4) = 2.1967633319996036
a(5) = 3.0911311689006347
a(6) = 0.637587516381819
a(7) = 1.2716872153209682
a(8) = 3.8780948673287092
a(9) = 0.5305567786431724
O erro E2 = 0.008476628330815156
```

(c) $TesteC|N = 512$

```
Qual item deseja fazer a simulação (a, b, c ou d)? c
Digite o N: 1024
Por favor aguarde...
N = 1024
a(0) = 1.0072813220754462
a(1) = 4.992443012454515
a(2) = 1.985876727614631
a(3) = 1.5132584652248546
a(4) = 2.1926928376836017
a(5) = 3.095152875932305
a(6) = 0.6523266477852667
a(7) = 1.2537898890575239
a(8) = 3.8796670569421763
a(9) = 0.5297366253008918
O erro E2 = 0.003779310463289278
```

(d) $TesteC|N = 1024$

```
Qual item deseja fazer a simulação (a, b, c ou d)? c
Digite o N: 2048
Por favor aguarde...
N = 2048
a(0) = 0.9999999999950262
a(1) = 5.000000000013884
a(2) = 2.0000000000137668
a(3) = 1.5000000000500613
a(4) = 2.2000000000523414
a(5) = 3.100000000035384
a(6) = 0.6000000000186514
a(7) = 1.3000000000248093
a(8) = 3.900000000023448
a(9) = 0.5000000000065652
O erro E2 = 2.5508844460640285e-12
```

(e) $TesteC|N = 2048$

A partir da tabela, é possível notar que com um aumento do N, ou seja, com o maior refinamento da malha, todos os a_k tendem a convergir a um determinado valor: a_0 aproxima-se cada vez mais de 1; a_1 de 5; a_2 de 2; a_3 de 1,5; a_4 de 2,2; a_5 de 3,1; a_6 de 0,6; a_7 de 1,3; a_8 de 3,9 e a_9 de 0,5.

Nota-se também uma progressiva diminuição do erro quadrático: em N=256, o erro era aproximadamente metade do erro de N=128; em N=512, era aproximadamente 0,7 vezes o erro de N=256; em N=1024, vale aproximadamente 0,44 vezes o erro de N=512; e, por fim, nota-se uma redução drástica no valor do erro em N=2048, com ele valendo em torno de somente $6,75 * 10^{-10}$ vezes o erro de N=1024.

4.4 Teste D

4.4.1 Comparação de $u_T(x)$

Primeiramente, comparam-se os valores de $u_T(x)$ reais e os obtidos na simulação do teste D nos seguintes gráficos:

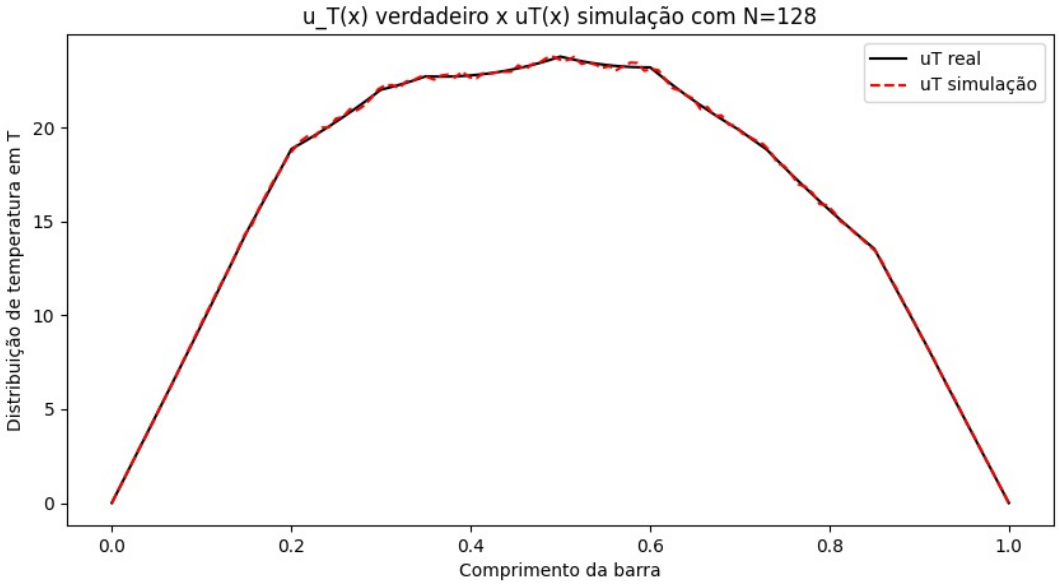

Figura 8: Teste D — N = 128

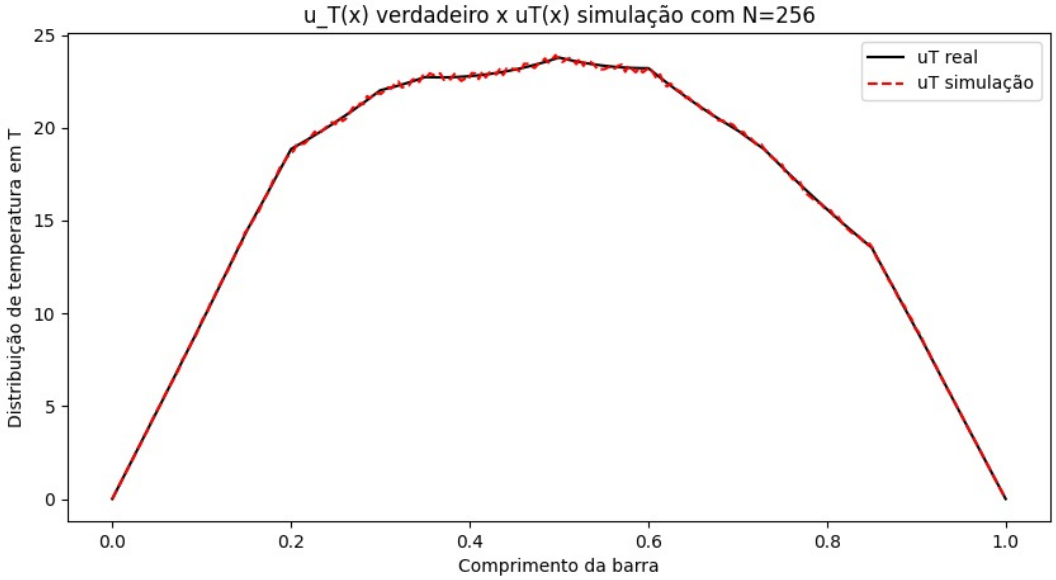

Figura 9: Teste D — N = 256

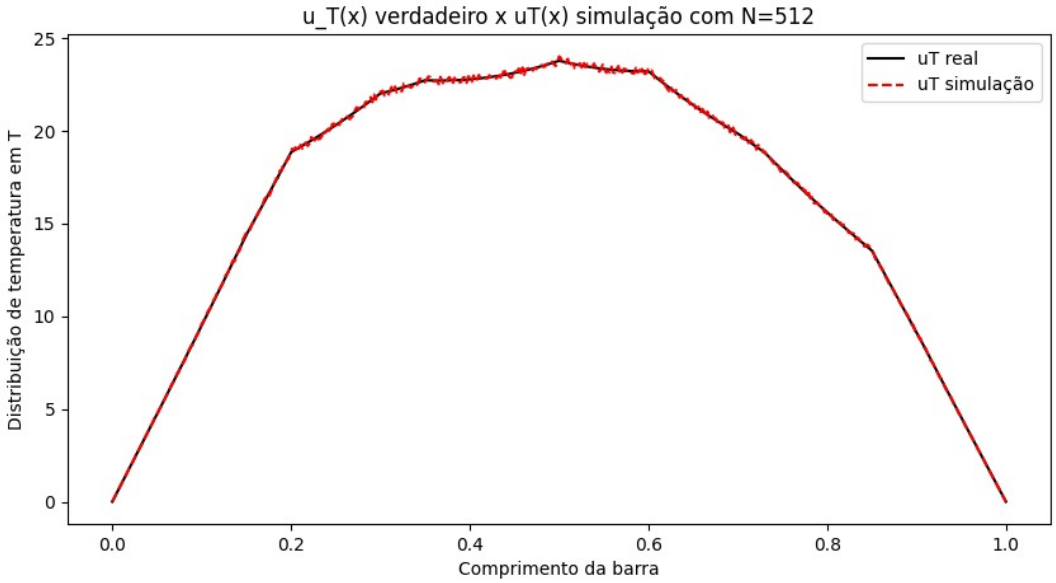

Figura 10: Teste D — N = 512

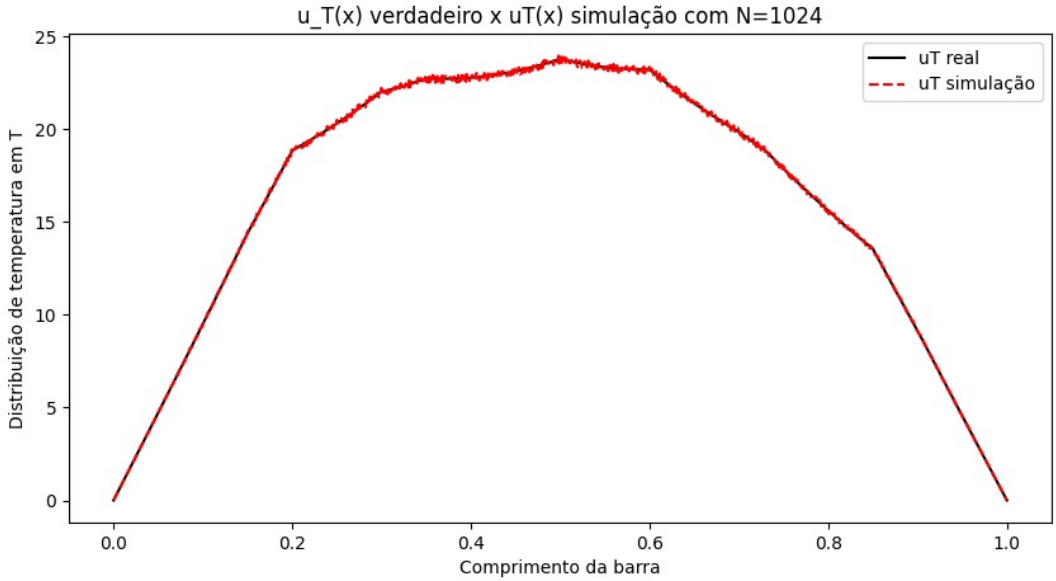

Figura 11: Teste D — N = 1024

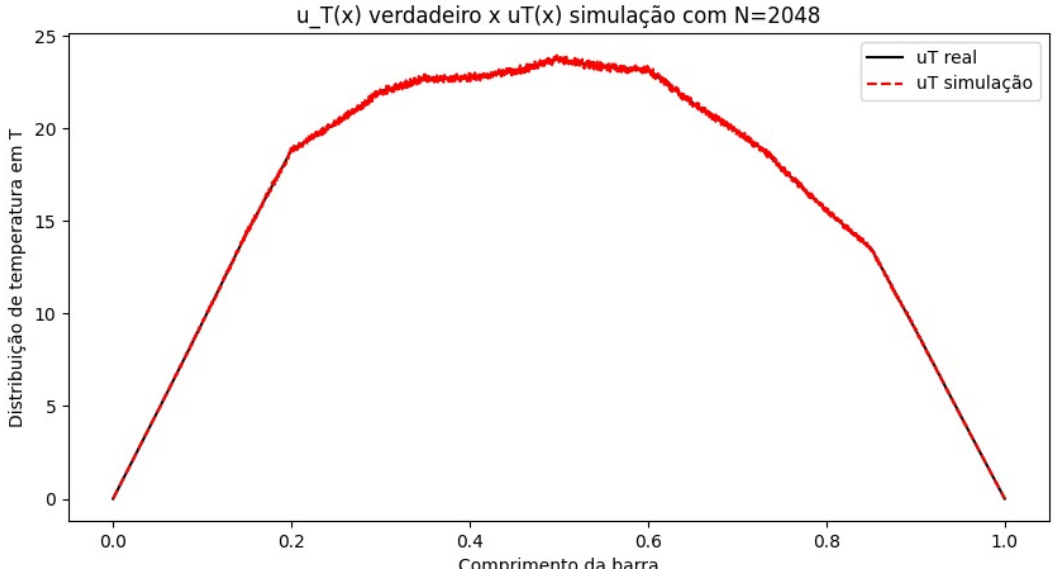

Figura 12: Teste D — N = 2048

No teste D, em todos os valores de N nota-se uma diferença entre o comportamento da função simulada em comparação com a do teste C, o que mostra a influência que o ruído tem sob $u_T(x)$ simulada: há uma diferença de valor entre a $u_T(x)$ simulada e a verdadeira, mostrando um maior erro de simulação.

4.4.2 Intensidades a_k e o erro quadrático E_2

Os valores de a_k e do erro quadrático E_2 do teste D estão listados nas figuras abaixo:

```
Qual item deseja fazer a simulação (a, b, c ou d)? d
Digite o N: 128
Por favor aguarde...
N = 128
a(0) = 1.140119101652445
a(1) = 4.876033225220013
a(2) = 2.032559704626472
a(3) = 1.4873295240334592
a(4) = 2.2098226843786453
a(5) = 3.00125142955371
a(6) = 0.6716302870557103
a(7) = 1.3139601312713118
a(8) = 3.862424780299662
a(9) = 0.5091095796145108
O erro E2 = 0.10110207897986831
```

(a) $TesteD|N = 128$

```
Qual item deseja fazer a simulação (a, b, c ou d)? d
Digite o N: 256
Por favor aguarde...
N = 256
a(0) = 0.9899311792608323
a(1) = 4.9697410283152905
a(2) = 2.2068232258621805
a(3) = 1.3060627446140565
a(4) = 2.2834081183400947
a(5) = 3.1694008765414594
a(6) = 0.3032029502651765
a(7) = 1.484432096881477
a(8) = 4.024545622559326
a(9) = 0.34726246372850045
O erro E2 = 0.10072664700831548
```

(b) $TesteD|N = 256$

```
Qual item deseja fazer a simulação (a, b, c ou d)? d
Digite o N: 512
Por favor aguarde...
N = 512
a(0) = 0.9902202981076655
a(1) = 5.004357304573361
a(2) = 1.9951043121746643
a(3) = 1.520887867635853
a(4) = 2.1856621734714565
a(5) = 3.0893456831452237
a(6) = 0.6431234860391664
a(7) = 1.27015093532499
a(8) = 3.8853468328997853
a(9) = 0.5180537231556496
O erro E2 = 0.10411818182197073
```

(c) $TesteD|N = 512$

```
Qual item deseja fazer a simulação (a, b, c ou d)? d
Digite o N: 1024
Por favor aguarde...
N = 1024
a(0) = 1.0721677356653991
a(1) = 4.895715668870508
a(2) = 2.0388835937987473
a(3) = 1.485810472205177
a(4) = 2.2254425433904625
a(5) = 3.0681620499276443
a(6) = 0.653872953089472
a(7) = 1.2582389067246318
a(8) = 3.88432772171344
a(9) = 0.5284306977986493
O erro E2 = 0.10340804378055327
```

(d) $TesteD|N = 1024$

```
Qual item deseja fazer a simulação (a, b, c ou d)? d
Digite o N: 2048
Por favor aguarde...
N = 2048
a(0) = 1.0302444180227752
a(1) = 4.9488998560470705
a(2) = 2.0223811995349834
a(3) = 1.5012342326243377
a(4) = 2.2081986345540745
a(5) = 3.0837982686634655
a(6) = 0.6293941806594301
a(7) = 1.2761052631490184
a(8) = 3.904357005404279
a(9) = 0.4978904912603951
O erro E2 = 0.10582015973063541
```

(e) $TesteD|N = 2048$

Similarmente ao que ocorreu no teste C, é possível notar uma tendência dos a_ks a convergir para os mesmos valores mencionados no teste C, entretanto, essa convergência mostra-se mais instável e não necessariamente um a_k vai ser mais próximo do valor de convergência com o aumento do N, embora, no maior aumento, de 1024 a 2048, em quase todos os a_k possa ser observada essa maior proximidade. Por exemplo, com o aumento de N de 512 para 1024, o valor de a_1 passa de aproximadamente 5,004 a 4,896 - um aumento da distância até 5 de 0,004 para 0,104 (em módulo, pois não convem estabelecer ordem de sentido/sinal nesse contexto).

Além disso, uma diferença ainda mais notável é que, ao contrário do que aconteceu no teste C, o aumento do refinamento da malha não significou uma diminuição do erro quadrático, o qual, com cada aumento de N, parecia variar aleatória ou ciclicamente: diminuiu de N=128 para N=256, aumentou de N=256 para N=512, diminuiu de N=512 para N=1024 e, ao fim, aumento de N=1024 para N=2048. E, comparando os valores inicial (N=128) e final (N=2048) de E_2 não só não há uma diferença significativa entre eles, como também há um aumento relativo pequeno de E_2: ele, em N=2048, é aproximadamente 1,0467 vezes o erro em N=128.

5 Conclusão

Em suma, a partir da implementação e análise dos algorítmos que resolveram o problema indireto proposto, é possível concluir que o método de resolução utilizado, partindo da obtenção de $u_k(T, x)$ por meio de Crank-Nicolson até a obtenção das intensidades a_k da fonte por meio da resolução de um problema de mínimos quadrados, é uma boa forma de resolver esse problema indireto da equação do calor, já que os valores obtidos verificaram as expectativas baseadas na teoria e o único caso em houve um maior erro quadrático associado a a_k foi quando foi inserido ruído, o que também era algo esperado.