

# Equação do Calor e Diferenças Finitas

Disciplina: Métodos Numéricos em Termofluidos

Professor: Adriano Possebon Rosa

## 1 Introdução

A equação do calor, que governa o comportamento da temperatura em um corpo, é de grande importância na matemática, na física e na engenharia. Ainda, diversos outros processos físicos são modelados por equações similares, como a difusão química, o movimento browniano e o efeito da viscosidade na dinâmica de um fluido.

Nessa aula, primeiramente, vamos **deduzir** e **entender** a equação do calor. Depois vamos resolver, numericamente, com o **método das diferenças finitas**, o problema de condução de calor transiente, tanto o unidimensional quanto o bidimensional.

A figura (1) mostra a **barra** que será estudada no problema unidimensional. A barra possui massa específica  $\rho$ , calor específico  $c$ , condutividade térmica  $k$  e comprimento  $L$ . No lado esquerdo, em  $x = 0$ , a temperatura é  $T_c$ , e no lado direito ( $x = L$ ) a temperatura é  $T_h$ . Não há calor passando pelas superfícies inferior e superior da barra, apenas pelas laterais. A figura (2) apresenta a **placa plana** quadrada que será estudada no problema bidimensional. A placa também é caracterizada fisicamente por  $\rho$ ,  $c$  e  $k$ , e possui lado  $L$ . As temperaturas são especificadas em cada parede (condições de contorno):  $T_c$  na esquerda ( $x = 0$ ), na inferior ( $y = 0$ ) e na direita ( $x = L$ ), e  $T_h$  na superior ( $y = L$ ). Além disso, também são dados os valores da temperatura no tempo  $t = 0$ , que estamos definindo como sendo o início do experimento (condição inicial). Vamos assumir que todos os pontos do interior da barra e da placa estão a uma temperatura  $T_c$  no tempo inicial.

O **objetivo** é encontrar o valor da temperatura em cada ponto e em cada instante de tempo. Ou seja, dadas as condições de contorno e iniciais, queremos encontrar  $T(x, t)$  no caso unidimensional e  $T(x, y, t)$  no caso bidimensional.

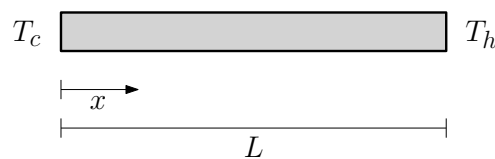


Figura 1: Barra com comprimento  $L$ .

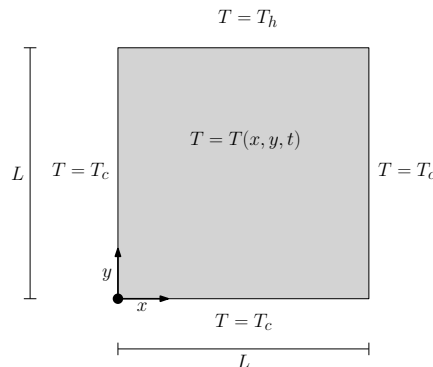


Figura 2: Placa quadrada com lado  $L$ .

## 2 Dedução da Equação do Calor

A **Primeira Lei da Termodinâmica** nos diz que a troca líquida de energia pela fronteira ao longo de um processo é igual à variação líquida da energia total do sistema durante o processo. Ou seja:

$$\Delta E = E_{entra} - E_{sai} . \quad (1)$$

A energia cruza a fronteira do sistema na forma de calor  $Q$  e de trabalho  $W$ . Assim, considerando como positivos o calor que chega ao sistema e o trabalho realizado pelo sistema, temos

$$\Delta E = Q - W . \quad (2)$$

Nosso foco aqui é na variação de energia causada pela **transferência de calor**. Assim, nos casos estudados não há realização de trabalho. Assim:

$$\Delta E = Q . \quad (3)$$

Além disso, estamos interessados apenas na variação da **Energia Interna**  $U$  do sistema, pois o sistema está em repouso e não há variação significativa das outras formas de energia que compõem a energia total (cinética, potencial, nuclear, química, ...). Resulta:

$$\Delta U = Q . \quad (4)$$

Se o processo tem uma duração  $\Delta t$ , então

$$U(t + \Delta t) - U(t) = Q . \quad (5)$$

Note que  $U(t)$  é a energia interna do sistema no início do processo e  $U(t + \Delta t)$  é a energia interna no final. A **energia interna** é toda a energia de um sistema que está associada com os componentes microscópicos (átomos e moléculas) quando vistos a partir de um referencial em repouso com o sistema. A **variação** da energia interna está diretamente relacionada com a **variação** da temperatura por

$$\Delta U = mc\Delta T , \quad (6)$$

ou

$$U(t + \Delta t) - U(t) = mc(T(t + \Delta t) - T(t)) . \quad (7)$$

Aqui,  $m$  é a **massa do sistema** e  $c$  é o **calor específico** (a volume constante, pra ser mais preciso, mas como o calor específico a volume constante e a pressão constante são praticamente iguais em líquidos e sólidos, essa distinção não terá grande relevância em nosso desenvolvimento).

Vamos considerar o nosso sistema como sendo **um pequeno bloco** da barra unidimensional apresentada na seção anterior. A figura (3) mostra uma representação desse sistema. A energia que entra no sistema é dada por  $Q(x)$  e a energia que sai é  $Q(x + \Delta x)$ . Ou seja,

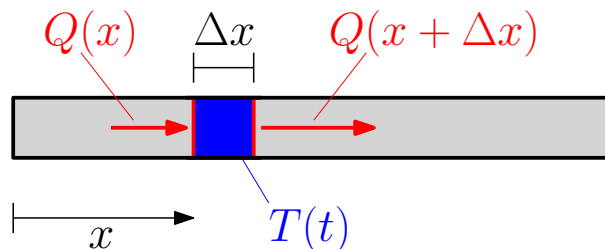


Figura 3: Bloco de largura  $\Delta x$  utilizado para a dedução da equação do calor.

$$Q = Q_{entra} - Q_{sai} = Q(x) - Q(x + \Delta x) . \quad (8)$$

Vamos utilizar a **taxa de transferência de calor**, que é a quantidade de calor transferida por unidade de tempo. Considerando que  $\Delta t$  é um intervalo pequeno, podemos aproximar

$$Q(x) = \Delta t \dot{Q}(x) . \quad (9)$$

Assim:

$$Q = \Delta t \dot{Q}(x) - \Delta t \dot{Q}(x + \Delta x) . \quad (10)$$

Substituindo os resultados (7) e (10) de volta em (3),

$$mc [T(t + \Delta t) - T(t)] = \Delta t [\dot{Q}(x) - \dot{Q}(x + \Delta x)] \quad (11)$$

A massa do bloco pode ser escrita como sendo a massa específica vezes o volume,

$$m = \rho V = \rho \Delta x A , \quad (12)$$

em que  $A$  é a área perpendicular à direção de propagação do calor ( $A = \Delta y \Delta z$ ). Assim:

$$\rho c \Delta x A [T(t + \Delta t) - T(t)] = \Delta t [\dot{Q}(x) - \dot{Q}(x + \Delta x)] . \quad (13)$$

Rearranjando:

$$\rho c \left( \frac{T(t + \Delta t) - T(t)}{\Delta t} \right) = - \frac{1}{A} \left( \frac{\dot{Q}(x + \Delta x) - \dot{Q}(x)}{\Delta x} \right) \quad (14)$$

No limite em que  $\Delta t \rightarrow 0$  e  $\Delta x \rightarrow 0$ , temos

$$\rho c \frac{\partial T}{\partial t} = - \frac{1}{A} \frac{\partial \dot{Q}}{\partial x} . \quad (15)$$

Temos uma equação para a temperatura, mas ainda não é possível resolvê-la, pois temos uma outra incógnita, o  $\dot{Q}$ . A transferência de calor está relacionada à diferença de temperatura no espaço: **se não há variação de temperatura não há transferência de calor**. A lei que relaciona  $\dot{Q}$  e  $T$  é a Lei de Fourier,

$$\dot{Q} = -kA \frac{\partial T}{\partial x} , \quad (16)$$

em que  $k$  é a **condutividade térmica** do material. Substituindo essa relação em (15) resulta em

$$\rho c \frac{\partial T}{\partial t} = \frac{1}{A} \frac{\partial}{\partial x} \left( kA \frac{\partial T}{\partial x} \right) . \quad (17)$$

Considerando o  $k$  constante,

$$\rho c \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} , \quad (18)$$

ou

$$\frac{\partial T}{\partial t} = \frac{k}{\rho c} \frac{\partial^2 T}{\partial x^2} , \quad (19)$$

Definindo a **difusividade térmica** como

$$\alpha = \frac{k}{\rho c_p} , \quad (20)$$

temos, finalmente,

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} . \quad (\text{unidimensional transiente}) . \quad (21)$$

Essa é a **Equação da Condução de Calor** unidimensional transiente. Essa equação governa o comportamento da temperatura e nos mostra que para ter uma variação local de  $T$  (no tempo) é necessário que haja uma variação espacial da derivada espacial de  $T$  (ou seja, uma derivada segunda não nula).

No **regime permanente**, ou seja, quando não há mais variação com relação ao tempo, a equação do calor unidimensional se reduz a

$$\frac{d^2 T}{dx^2} = 0 \quad (\text{unidimensional permanente}) . \quad (22)$$

Essa é a equação do calor para o caso unidimensional permanente.

O desenvolvimento da equação do calor para duas dimensões é análogo ao apresentado aqui nessa seção para o caso unidimensional, com a inclusão da possibilidade de troca de calor na direção  $y$ . A equação resultante é dada por

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + \alpha \frac{\partial^2 T}{\partial y^2} \quad (\text{bidimensional transiente}) . \quad (23)$$

Juntamente com as condições de contorno para a temperatura e uma condição inicial, a equação (23) permite encontrar a temperatura em cada ponto de uma placa bidimensional com  $k$  constante. A equação (23) representa o caso **bidimensional transiente**. No caso de **regime permanente**, a equação (23) se torna:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (\text{bidimensional permanente}) . \quad (24)$$

Vamos resolver aqui nessa aula apenas os casos transientes. Estudaremos os problemas permanentes mais adiante em nosso curso.

### 3 Adimensionalização das Equações Governantes

Para facilitar e generalizar a solução das equações apresentadas na seção anterior, é proposta uma adimensionalização utilizando grandezas características do problema físico. As variáveis dimensionais são substituídas por variáveis adimensionais, com o asterisco. Assim, para a equação (23), temos:

$$x^* = \frac{x}{L} \quad y^* = \frac{y}{L} \quad (25)$$

$$t^* = \frac{\alpha}{L^2} t \quad T^* = \frac{T - T_c}{T_h - T_c} \quad (26)$$

Substituindo as variáveis adimensionais de volta em (23), temos:

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} . \quad (27)$$

E as condições de contorno passam a ser:

$$T^* = T_c^* = 0 \quad \text{em} \quad x^* = 0, x^* = 1 \text{ e } y^* = 0 \quad (28)$$

$$T^* = T_h^* = 1 \quad \text{em} \quad y^* = 1 \quad (29)$$

Para o caso unidimensional temos

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} , \quad (30)$$

com

$$T^* = T_c^* = 0 \quad \text{em} \quad x^* = 0 \quad (31)$$

$$T^* = T_h^* = 1 \quad \text{em} \quad x^* = 1 \quad (32)$$

Usaremos, no restante desta aula, **apenas a forma adimensional da equação do calor**, ou seja, equações (27) e (30) e suas respectivas condições. Assim, para evitar uma notação carregada, não será utilizado o símbolo  $*$  para indicar que a variável é adimensional. Ou seja, **todas as variáveis a partir deste ponto são adimensionais**, mesmo sem o  $*$ .

### 3.1 Equações Finais

**Conclusão:** vamos resolver a equação da condução de calor na barra (ver figura 4),

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < 1, \quad t > 0, \quad (33)$$

com

$$T(x, 0) = 0, \quad 0 < x < 1 \quad (34)$$

$$T(0, t) = 0, \quad t \geq 0 \quad (35)$$

$$T(1, t) = 1, \quad t \geq 0, \quad (36)$$

e depois a equação da condução de calor em uma placa (ver figura 5),

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}, \quad 0 < x < 1, \quad 0 < y < 1, \quad t > 0, \quad (37)$$

com

$$T(x = 0, y, t \geq 0) = 0, \quad T(x = 1, y, t \geq 0) = 0, \quad 0 \leq y \leq 1 \quad (38)$$

$$T(x, y = 0, t) = 0, \quad T(x, y = 1, t) = 1, \quad 0 < x < 1 \quad (39)$$

$$T(x, y, t = 0) = 0, \quad 0 < x, y < 1 \quad (40)$$

O primeiro problema será chamado de **unidimensional transiente** e o segundo de **bidimensional transiente**.

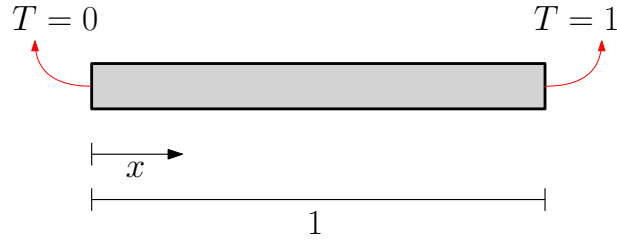


Figura 4: Barra com comprimento 1.

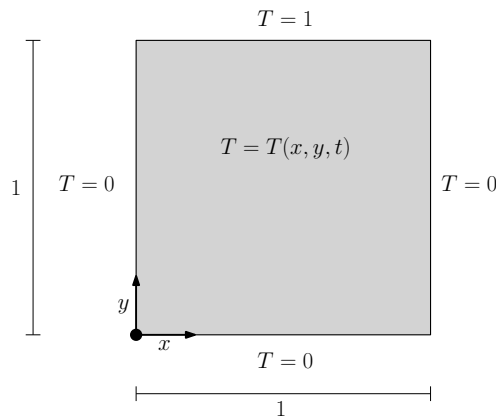


Figura 5: Placa quadrada de lado 1.

**Observação:** o desenvolvimento está sendo feito com as condições de contorno apresentadas nas figuras (4) e (5), mas a metodologia pode ser facilmente generalizada para outros tipos de condições de contorno impostas.

### 3.2 Comportamento da Solução

**Como é o comportamento da temperatura no problema unidimensional transiente?** Bom, isso é o que nós queremos descobrir. Mas antes de resolvermos o problema, vamos dar uma olhada em como será a solução (*spoiler alert*).

A figura (6) mostra a temperatura em  $t = 0$ , que é a condição inicial. Note que em todos os pontos a temperatura é igual a zero, exceto em  $x = 1$ , para o qual  $T = 1$ . A temperatura em  $x = 0$  e em  $x = 1$  não vai mudar ao longo do tempo, devido às condições de contorno.

A figura (7) mostra a temperatura em todos os pontos da barra para  $t = 0.01$ . Note que os pontos próximos da extremidade direita tiveram um aumento na temperatura, causado pela condução de calor. **Os pontos próximos à extremidade esquerda ainda não sentiram esse aumento.**

A figura (8) mostra a temperatura para  $t = 0.05$ . Aqui todos os pontos já sentiram a presença da extremidade direita aquecida. **Na figura (9), para  $t = 0.5$ , o perfil de temperatura é uma reta e não vai mais mudar com o tempo: chegamos no regime permanente. E por que não muda mais com o tempo?** Porque, lembrando da equação governante

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2},$$

para que exista uma mudança no tempo é necessário que a derivada segunda da temperatura com relação a  $x$  seja diferente de zero. Mas se o gráfico é uma reta, então a derivada segunda é igual a zero, o que significa que a temperatura não muda mais com o tempo. (Observação: para um bloco no meio da barra, note que não há variação de energia no tempo, mesmo com o bloco vizinho da direita estando a uma temperatura maior que o bloco vizinho da esquerda; isso acontece porque a energia recebida do vizinho mais quente é exatamente igual à energia transmitida ao vizinho mais frio; para haver variação da temperatura no tempo, a energia recebida de um lado tem que ser diferente da energia transmitida do outro.)

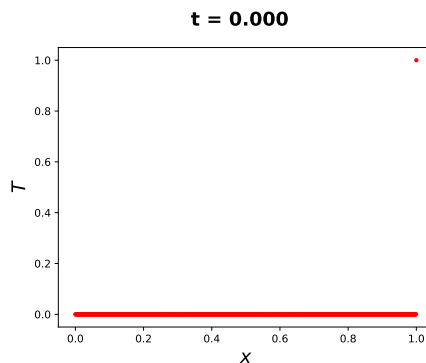


Figura 6: Temperatura em  $t = 0$ .

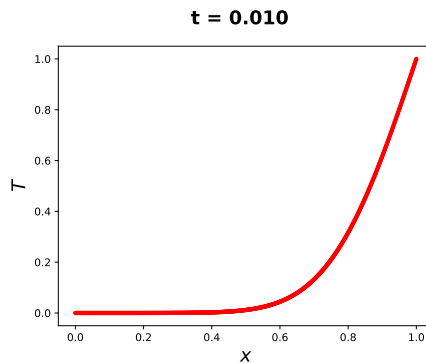


Figura 7: Temperatura em  $t = 0.01$ .

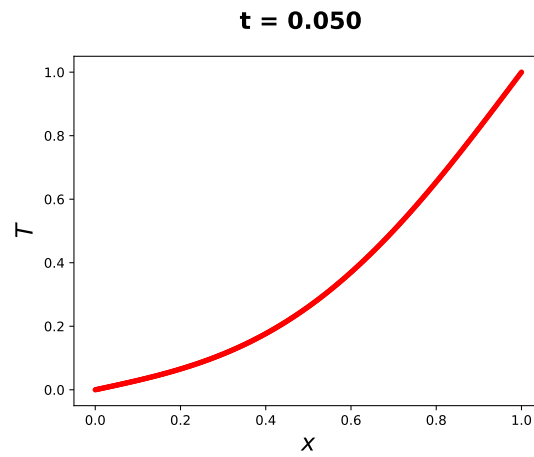


Figura 8: Temperatura em  $t = 0.05$ .

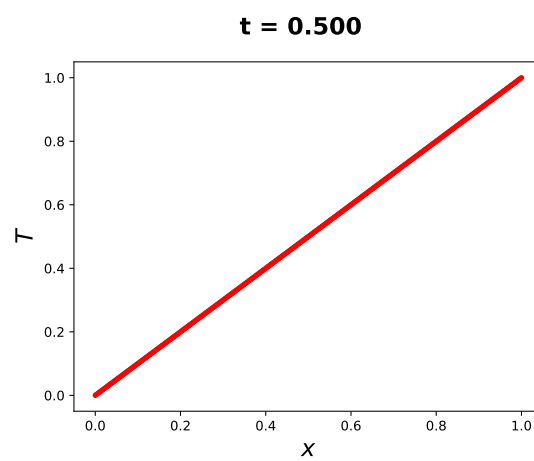


Figura 9: Temperatura em  $t = 0.5$ .

## 4 Metodologia Numérica

Resolver um problema numericamente significa dividir o domínio em um número finito de pontos (ou regiões) e transformar as equações diferenciais governantes em equações algébricas. As milhares (ou milhões) de equações algébricas, por sua vez, são resolvidas pelo computador. Neste trabalho, utilizaremos o **método das diferenças finitas**. Outros métodos também muito utilizados são os métodos dos volumes finitos e dos elementos finitos.

## 5 Método das Diferenças Finitas: Problema Unidimensional Transiente

A resolução de um problema usando o Método das Diferenças Finitas consiste de 4 etapas (HOFFMAN, 1992):

1. Dividir o domínio
2. Aproximar as derivadas usando séries de Taylor
3. Substituir na equação original
4. Resolver

Vamos ver essas etapas detalhadamente.

### 5.1 Dividindo o Domínio

No método das diferenças finitas, a temperatura é determinada apenas em pontos específicos do domínio e em certos instantes de tempo. Para o nosso problema da barra unidimensional, isso significa que a temperatura num ponto  $x_i$  e num tempo  $t_k$  é aproximada pela temperatura numérica (ou discreta):

$$T(x_i, t_k) \approx T_i^k. \quad (41)$$

Temos  $x_i = i\Delta x$  e no instante de tempo  $t_k = k\Delta t$ . Aqui  $\Delta x$  é o incremento na variável espacial e  $\Delta t$  é o passo de tempo. Aqui,  $i$  vai de 0 a  $N$ , onde  $N$  é o número de divisões do domínio. Já  $k$  vai de 0 até um valor qualquer definido.  $k$  e  $i$  são inteiros. Não temos mais a temperatura em qualquer ponto do domínio e em qualquer instante de tempo, mas apenas em posições e em tempos específicos (não temos mais infinitos pontos reais, mas apenas um número finito de pontos, que serão armazenados na memória do computador).

Para ilustrar essa discretização do domínio, vamos considerar aqui que a nossa barra está dividida em 5 partes ( $N = 5$ ) iguais, ou seja, temos 6 pontos, como mostra a figura (10). (Note que a escolha  $N = 5$  foi feita por mim, arbitrariamente. Veremos, mais a frente, que quanto maior  $N$  mais correta é a solução, porém maior é o custo computacional).

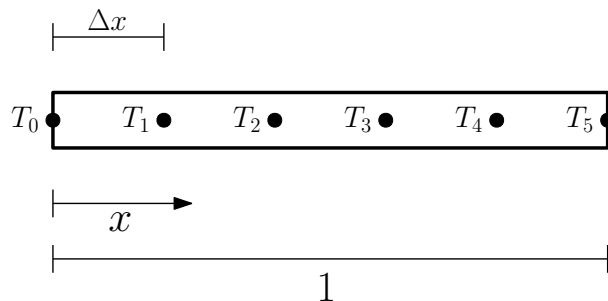


Figura 10: Barra unidimensional discretizada, com  $N = 5$ .

Temos os pontos 0, 1, 2, 3, 4 e 5. Essas são as identificações dos pontos, e não suas posições ( $x$  vai de zero a 1 somente). Esses pontos estão afastados por uma distância  $\Delta x = 0.2$  entre si (a barra tem comprimento 1, então  $\Delta x = 1/N = 1/5 = 0.2$ ). É importante notar que  $\Delta x$  e  $N$  não são independentes: com o tamanho da barra fixo, a escolha de um automaticamente define o outro. A temperatura é calculada



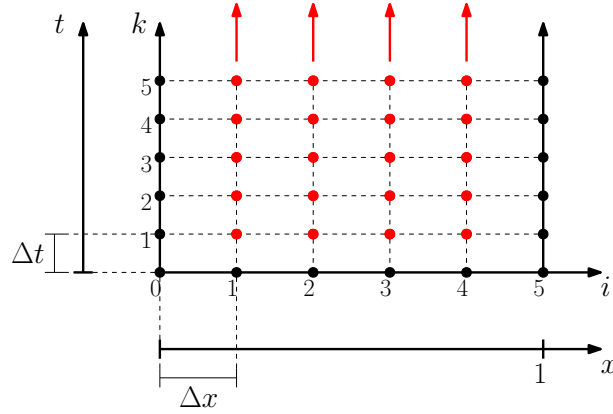


Figura 11: Dimensões temporal e espacial, para o problema unidimensional. Os pontos pretos são conhecidos, enquanto os vermelhos são desconhecidos.

apenas nesses pontos. Assim, a temperatura no ponto 0 é  $T_0$ , no ponto 1 é  $T_1$ , no ponto 2 é  $T_2$  e assim por diante. De maneira geral, a temperatura no ponto  $i$  é  $T_i$ . O ponto  $i + 1$  é o vizinho da direita do ponto  $i$  e o ponto  $i - 1$  é o vizinho da esquerda do ponto  $i$ . Assim,  $T_{i+1}$  é a temperatura do ponto que está à direita do ponto  $i$  e  $T_{i-1}$  é a temperatura do ponto que está à esquerda do ponto  $i$ .

**Mas além de mudar no espaço, a temperatura também muda no tempo.** A figura (11) mostra os dois domínios, do espaço (fechado) e do tempo (aberto). O intervalo entre duas medições no tempo será  $\Delta t$ . Assim, vamos começar com  $t = 0$ , que é  $0 \times \Delta t$ . Depois vamos para o tempo  $t_1 = 1 \times \Delta t$ . Em seguida para  $t_2 = 2 \times \Delta t$ , e assim por diante. De maneira geral, temos  $t_k = k \times \Delta t$ . Se o instante atual é representado por  $k$ , o instante futuro próximo é dado por  $k + 1$ .

**A temperatura em um ponto  $i$  em um dado passo de tempo  $k$  é representada por  $T_i^k$ .** Vamos considerar, por exemplo,  $\Delta t = 0.01$ . Então a temperatura em  $x = 0.4$  e em  $t = 0.02$  é dada por  $T_2^2$ . A temperatura em  $x = 0.8$  e em  $t = 0.13$  é  $T_4^{13}$ . Acho que deu pra entender a ideia. (Note, novamente, que eu escolhi  $\Delta t = 0.01$  arbitrariamente. Na verdade, nem tanto. Para dar certo, temos que ter  $\Delta t < 0.5\Delta x^2$ , devido à condição de estabilidade. Além disso, quanto menor  $\Delta t$  mais precisa é a solução.)

Para o nosso problema, que é dado pelas equações (33), (34), (34) e (35), temos que **a condição inicial nos informa que todos os pontos do interior da barra têm temperatura igual a zero em  $t = 0$** . Para a nossa solução numérica, isso significa que

$$T_1^0 = 0 \quad T_2^0 = 0 \quad T_3^0 = 0 \quad T_4^0 = 0 .$$

Por outro lado, **as condições de contorno nos informam que a temperatura no ponto da extremidade esquerda é igual a zero sempre e a temperatura na extremidade direita é igual a 1 sempre**. Isso significa que a temperatura do ponto 0 será sempre 0 e a temperatura do ponto 5 será sempre 1. Traduzindo para o nosso caso, temos:

$$\begin{aligned} T_0^0 &= 0 & T_5^0 &= 1 \\ T_0^1 &= 0 & T_5^1 &= 1 \\ T_0^2 &= 0 & T_5^2 &= 1 \\ &\dots\dots\dots \end{aligned}$$

Portanto, os pontos 0 e 5 já estão resolvidos. Só precisamos resolver os pontos 1, 2, 3 e 4. A figura (11) mostra os pontos conhecidos (preto) e os pontos desconhecidos (vermelho).

Para o tempo  $t = 0$  já está tudo resolvido. **Pergunta: como calcular os valores de  $T$  no próximo passo de tempo?** Ou seja, como calcular  $T_1^1$ ,  $T_2^1$ ,  $T_3^1$  e  $T_4^1$ ? **Afinal é esse o nosso interesse, determinar o comportamento da temperatura ao longo do tempo.** Nós conhecemos apenas valores em  $k = 0$ . Para determinar a temperatura no futuro vamos usar a equação governante.

## 5.2 Aproximando as Derivadas

Aí vem a segunda parte do método de diferenças finitas: vamos transformar a equação diferencial governante em várias equações algébricas. Para isso, vamos aproximar as derivadas da equação original utilizando séries de Taylor. Na equação original temos uma derivada primeira no tempo

$$\frac{\partial T}{\partial t}$$

e uma derivada segunda no espaço

$$\frac{\partial^2 T}{\partial x^2}.$$

Considere a expansão em série de Taylor para a temperatura em  $x_i + \Delta x$  em torno de  $x_i$  (em um tempo  $t_k$  qualquer fixo):

$$T(x_i + \Delta x, t_k) = T(x_i, t_k) + \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, t_k} + \frac{\Delta x^3}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, t_k} + \dots \quad (42)$$

Mas  $T(x_i + \Delta x, t_k) = T_{i+1}^k$  e  $T(x_i, t_k) = T_i^k$ . Assim, reescrevendo essa equação temos:

$$T_{i+1}^k = T_i^k + \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, t_k} + \frac{\Delta x^3}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, t_k} + \dots \quad (43)$$

Isolando a primeira derivada de  $T$  com relação a  $x$  temos:

$$\left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} = \frac{T_{i+1}^k - T_i^k}{\Delta x} - \frac{\Delta x}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, t_k} - \frac{\Delta x^2}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, t_k} + \dots \quad (44)$$

Podemos escrever:

$$\left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} = \frac{T_{i+1}^k - T_i^k}{\Delta x} + R(\Delta x) \quad (45)$$

Aqui,  $R(\Delta x)$  é uma função  $\mathcal{O}(\Delta x)$ , ou seja, ordem  $\Delta x$ . Se o último termo na equação (45) for truncado ou desprezado, temos então uma aproximação para a primeira derivada em  $x_i$ :

$$\left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} \approx \frac{T_{i+1}^k - T_i^k}{\Delta x}. \quad (46)$$

Essa é a aproximação da primeira derivada de uma função  $T(x, t)$  no ponto  $x_i$  e no tempo  $t_k$  usando diferenças finitas. **Transformamos uma derivada espacial em uma equação algébrica, que usa pontos vizinhos do domínio discreto numérico.** Essa aproximação é para frente (pois usa um ponto que está na frente de  $x_i$ ) e de primeira ordem (o erro é proporcional a  $\Delta x$ ). Existem outras aproximações.

Considere novamente a série de Taylor em torno de  $x_i$ , mas agora para a função  $T$  em  $x_{i-1}$ :

$$T_{i-1}^k = T_i^k - \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, t_k} - \frac{\Delta x^3}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, t_k} + \dots \quad (47)$$

Da equação (47), podemos aproximar:

$$\left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} \approx \frac{T_i^k - T_{i-1}^k}{\Delta x}. \quad (48)$$

Essa também é uma aproximação da primeira derivada espacial de  $T(x, t)$  em  $x_i$ , mas agora usando um ponto para trás de  $i$ . Essa aproximação também é de ordem  $\Delta x$ .

Esses são exemplos de aproximações para a primeira derivada espacial. **Mas nós precisamos da segunda derivada espacial.** Para a segunda derivada, considere a soma das equações (43) e (47):

$$\begin{aligned} T_{i+1}^k + T_{i-1}^k &= 2T_i^k + \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} - \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, t_k} + \\ &+ \frac{\Delta x^3}{6} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, t_k} - \frac{\Delta x^3}{6} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, t_k} + \frac{\Delta x^4}{24} \left. \frac{\partial^4 T}{\partial x^4} \right|_{x_i, t_k} + \frac{\Delta x^4}{24} \left. \frac{\partial^4 T}{\partial x^4} \right|_{x_i, t_k} + \dots \end{aligned} \quad (49)$$

Os termos com derivadas ímpares se cancelam (isso é fundamental, pois a primeira derivada desaparece da equação):

$$\begin{aligned} T_{i+1}^k + T_{i-1}^k = 2T_i^k + \cancel{\Delta x \frac{\partial T}{\partial x} \Big|_{x_i, t_k}} - \cancel{\Delta x \frac{\partial T}{\partial x} \Big|_{x_i, t_k}} + \frac{\Delta x^2}{2} \frac{\partial^2 T}{\partial x^2} \Big|_{x_i, t_k} + \frac{\Delta x^2}{2} \frac{\partial^2 T}{\partial x^2} \Big|_{x_i, t_k} + \\ + \cancel{\frac{\Delta x^3}{6} \frac{\partial^3 T}{\partial x^3} \Big|_{x_i, t_k}} - \cancel{\frac{\Delta x^3}{6} \frac{\partial^3 T}{\partial x^3} \Big|_{x_i, t_k}} + \frac{\Delta x^4}{24} \frac{\partial^4 T}{\partial x^4} \Big|_{x_i, t_k} + \frac{\Delta x^4}{24} \frac{\partial^4 T}{\partial x^4} \Big|_{x_i, t_k} + \dots \end{aligned} \quad (50)$$

Isolando a segunda derivada:

$$\frac{\partial^2 T}{\partial x^2} \Big|_{x_i, t_k} = \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2} + R(\Delta x^2). \quad (51)$$

Truncando a função  $R$ , resulta na aproximação usando diferenças finitas centradas da segunda derivada da função  $T(x, t)$  no ponto  $x_i$  e em  $t_k$ :

$$\frac{\partial^2 T}{\partial x^2} \Big|_{x_i, t_k} \approx \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2}. \quad (52)$$

Note que essa aproximação é de segunda ordem, ou seja, estamos cometendo um erro que é proporcional a  $\Delta x^2$ . Nós não sabemos o valor do erro, mas sabemos que esse erro diminui 4 vezes quando o  $\Delta x$  é dividido por 2.

**A equação (52) é uma parte do que a gente queria: uma aproximação para a derivada segunda no espaço. Falta uma aproximação para a derivada primeira no tempo.**

Considere a expansão em série de Taylor para a temperatura em  $t_k + \Delta t$  em torno de  $t_k$  (em um ponto  $x_i$  qualquer fixo):

$$T(x_i, t_k + \Delta t) = T(x_i, t_k) + \Delta t \frac{\partial T}{\partial t} \Big|_{x_i, t_k} + \frac{\Delta t^2}{2} \frac{\partial^2 T}{\partial t^2} \Big|_{x_i, t_k} + \frac{\Delta t^3}{3!} \frac{\partial^3 T}{\partial t^3} \Big|_{x_i, t_k} + \dots \quad (53)$$

Mas  $T(x_i, t_k + \Delta t) = T_i^{k+1}$  e  $T(x_i, t_k) = T_i^k$ . Assim, reescrevendo:

$$T_i^{k+1} = T_i^k + \Delta t \frac{\partial T}{\partial t} \Big|_{x_i, t_k} + \frac{\Delta t^2}{2} \frac{\partial^2 T}{\partial t^2} \Big|_{x_i, t_k} + \frac{\Delta t^3}{3!} \frac{\partial^3 T}{\partial t^3} \Big|_{x_i, t_k} + \dots \quad (54)$$

Isolando a primeira derivada de  $T$  com relação a  $t$  temos:

$$\frac{\partial T}{\partial t} \Big|_{x_i, t_k} = \frac{T_i^{k+1} - T_i^k}{\Delta t} - \frac{\Delta t}{2} \frac{\partial^2 T}{\partial t^2} \Big|_{x_i, t_k} - \frac{\Delta t^2}{3!} \frac{\partial^3 T}{\partial t^3} \Big|_{x_i, t_k} + \dots \quad (55)$$

Podemos escrever:

$$\frac{\partial T}{\partial t} \Big|_{x_i, t_k} = \frac{T_i^{k+1} - T_i^k}{\Delta t} + R(\Delta t) \quad (56)$$

Aqui,  $R(\Delta t)$  é uma função  $\mathcal{O}(\Delta t)$ , ou seja, ordem  $\Delta t$ . Se o último termo na equação é truncado ou desprezado, temos então uma aproximação para a primeira derivada no ponto  $x_i$  no tempo  $t_k$ :

$$\frac{\partial T}{\partial t} \Big|_{x_i, t_k} \approx \frac{T_i^{k+1} - T_i^k}{\Delta t}. \quad (57)$$

Essa é a aproximação da primeira derivada temporal de uma função  $T(x, t)$  no ponto  $x_i$  e no tempo  $t_k$  usando diferenças finitas. **Transformamos uma derivada temporal em uma equação algébrica, que usa um ponto no futuro e um ponto no presente.**

As equações (52) e (57) são as aproximações das derivadas que queremos. Vamos substituí-las na equação diferencial parcial original.

### 5.3 Substituindo as Aproximações na Equação Original

A equação diferencial parcial original é dada por

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} . \quad (58)$$

Vamos aproximar essa equação em um dado ponto e em um dado tempo usando as relações dadas nas equações (52) e (57). Disso resulta:

$$\frac{T_i^{k+1} - T_i^k}{\Delta t} = \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2} . \quad (59)$$

**Essa é a nossa Equação de Diferenças Finitas.** Para cada ponto e em cada instante de tempo temos uma equação algébrica. Isso significa que transformamos uma única equação diferencial parcial em várias equações algébricas.

Com as condições de contorno e a condição inicial dadas é possível calcular a nova temperatura  $T_i^{k+1}$  em todos os pontos  $i$  e, com isso, fazer a evolução temporal da temperatura, com um passo de cada vez. Para o nosso problema exemplo, vamos ver como se faz isso.

### 5.4 Resolvendo

A nossa equação de diferenças é a equação (59). Note que os valores no tempo  $k$  são conhecidos, já que é o tempo presente. Não conhecemos os valores em  $k+1$ , que é o tempo futuro. Podemos isolar  $T_i^{k+1}$  na equação, resultando em:

$$T_i^{k+1} = T_i^k + \frac{\Delta t}{\Delta x^2} (T_{i+1}^k - 2T_i^k + T_{i-1}^k) . \quad (60)$$

Vamos voltar agora ao nosso problema particular, o exemplo que estamos resolvendo neste roteiro. Temos  $\Delta x = 0.2$  e  $\Delta t = 0.01$ , que resulta em  $\Delta t / \Delta x^2 = 0.25$ . Para  $k = 0$  (ver figura 12),

$$T_0^0 = 0 \quad T_1^0 = 0 \quad T_2^0 = 0 \quad T_3^0 = 0 \quad T_4^0 = 0 \quad T_5^0 = 1 .$$

Com esses valores, vamos usar a equação (60) com  $k = 0$  e com  $i$  variando de 1 a 4:

$$T_1^1 = T_1^0 + 0.25 (T_2^0 - 2T_1^0 + T_0^0) = 0 + 0.25(0 - 2 \times 0 + 0) = 0 . \quad (61)$$

$$T_2^1 = T_2^0 + 0.25 (T_3^0 - 2T_2^0 + T_1^0) = 0 + 0.25(0 - 2 \times 0 + 0) = 0 . \quad (62)$$

$$T_3^1 = T_3^0 + 0.25 (T_4^0 - 2T_3^0 + T_2^0) = 0 + 0.25(0 - 2 \times 0 + 0) = 0 . \quad (63)$$

$$T_4^1 = T_4^0 + 0.25 (T_5^0 - 2T_4^0 + T_3^0) = 0 + 0.25(1 - 2 \times 0 + 0) = 0.25 . \quad (64)$$

Assim, no tempo 0.01, que corresponde a  $k = 1$ , temos (ver figura 13)

$$T_0^1 = 0 \quad T_1^1 = 0 \quad T_2^1 = 0 \quad T_3^1 = 0 \quad T_4^1 = 0,25 \quad T_5^1 = 1 .$$

Com isso avançamos 1 passo de tempo. Agora é só continuar avançando no tempo. Duas observações:

- para avançar um passo no futuro, precisamos apenas da solução no tempo presente;
- para um dado tempo, primeiro calculamos a temperatura em todos os pontos da barra, e somente depois é possível avançar no tempo. **Não é possível avançar no tempo apenas em um ponto e depois voltar no tempo e calcular para um outro ponto (fisicamente também não faz sentido).**

Vamos usar agora a equação (60) com  $k = 1$  e com  $i$  indo de 1 a 4:

$$T_1^2 = T_1^1 + 0.25 (T_2^1 - 2T_1^1 + T_0^1) = 0 + 0.25(0 - 2 \times 0 + 0) = 0 . \quad (65)$$

$$T_2^2 = T_2^1 + 0.25 (T_3^1 - 2T_2^1 + T_1^1) = 0 + 0.25(0 - 2 \times 0 + 0) = 0 . \quad (66)$$

$$T_3^2 = T_3^1 + 0.25 (T_4^1 - 2T_3^1 + T_2^1) = 0 + 0.25(0.25 - 2 \times 0 + 0) = 0.0625 . \quad (67)$$

$$T_4^2 = T_4^1 + 0.25 (T_5^1 - 2T_4^1 + T_3^1) = 0.25 + 0.25(1 - 2 \times 0.25 + 0) = 0.375 . \quad (68)$$

Assim, no tempo 0.02, que corresponde a  $k = 2$ , temos (ver figura 14)

$$T_0^2 = 0 \quad T_1^2 = 0 \quad T_2^2 = 0 \quad T_3^2 = 0.0625 \quad T_4^2 = 0.375 \quad T_5^2 = 1 .$$

E podemos continuar essas contas no tempo, para encontrar a temperatura em tempos futuros. **Note que  $x$  é limitado, entre 0 e 1. Mas o tempo vai de zero até o valor que você quiser.**

Se pensarmos na temperatura como um vetor, temos, para cada tempo:

$$T^0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$$

$$T^1 = [0 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 1]$$

$$T^2 = [0 \quad 0 \quad 0 \quad 0.0625 \quad 0.375 \quad 1]$$

.....

Essa ideia de vetor será útil na programação.

Observação: os valores de temperatura são calculados apenas nos 6 pontos escolhidos, como mostram as figuras (12), (13) e (14). Você pode traçar uma curva unindo esses pontos, pra ficar mais 'informativo', como mostra a figura (15). Mas é importante ficar claro que os valores calculados são apenas os 6 pontos, e os valores entre os pontos são apenas suposições.

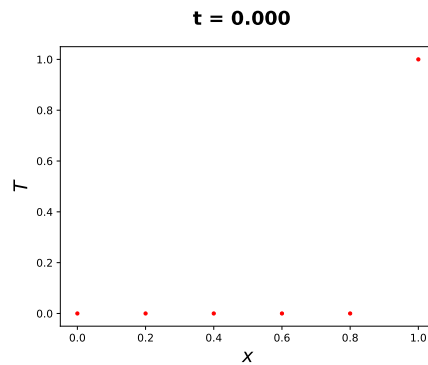


Figura 12: Temperatura para  $k = 0$ .

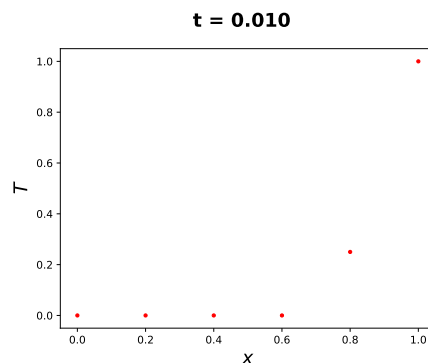
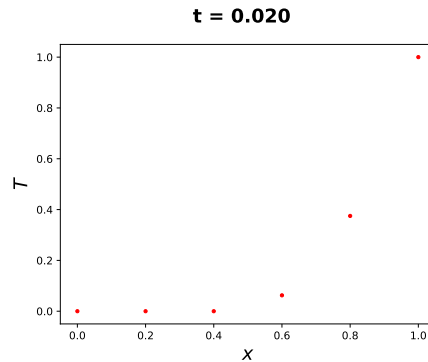
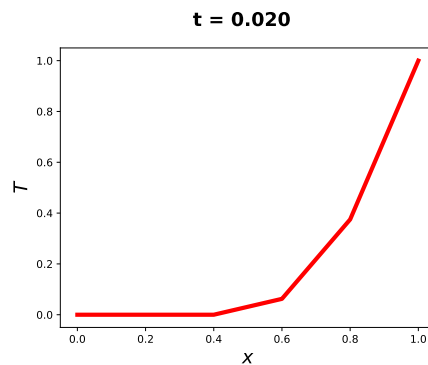


Figura 13: Temperatura para  $k = 1$ .

Figura 14: Temperatura para  $k = 2$ .Figura 15: Temperatura para  $k = 2$ .

Ok, tudo certo. Mas se é possível calcular na mão, como fizemos até aqui, **por que usar um computador?**

Bom, como vimos na parte de série de Taylor, a nossa aproximação tem um erro que é proporcional a  $\Delta x^2$  e a  $\Delta t$ . Então quanto menores  $\Delta x$  e  $\Delta t$ , menor é o nosso erro numérico, e mais a nossa solução se aproxima da solução verdadeira. Por isso precisamos colocar muitos pontos no domínio. Nesse exemplo, nós utilizamos  $N = 5$ . Mas e se  $N = 1000$ ? E se eu quiser a temperatura em  $t = 10$ ? Vou ter que fazer essas contas 1000 vezes. Aí se torna muito complicado realizar esses cálculos à mão. **Melhor deixar o computador fazer essas contas pra gente.**

## 6 Implementação do Problema Unidimensional Transiente no Python

Vamos implementar, no Python, a solução numérica desenvolvida na seção anterior. Os passos podem ser encontrados no anexo.

## 7 Problema Bidimensional Transiente

Vamos agora para o problema bidimensional transiente. A equação que estamos querendo resolver é

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}, \quad 0 < x < 1, \quad 0 < y < 1, \quad t > 0, \quad (69)$$

com

$$T(x=0, y, t \geq 0) = 0, \quad T(x=1, y, t \geq 0) = 0, \quad 0 \leq y \leq 1 \quad (70)$$

$$T(x, y=0, t) = 0, \quad T(x, y=1, t) = 1, \quad 0 < x < 1 \quad (71)$$

$$T(x, y, t=0) = 0, \quad 0 < x, y < 1 \quad (72)$$

Vamos utilizar o método das diferenças finitas para resolver esse problema. Para isso, vamos seguir novamente cada uma das 4 etapas.

### 7.1 Discretizando o Domínio

Nosso domínio agora é bidimensional, como mostra a figura (16). A temperatura num ponto  $(x_i, y_j)$ , num tempo  $t_k$ , é aproximada pela temperatura numérica (ou discreta):

$$T(x_i, y_j, t_k) \approx T_{i,j}^k. \quad (73)$$

Temos  $x_i = i\Delta x$ ,  $y_j = j\Delta y$  e  $t_k = k\Delta t$ . Aqui  $\Delta x$  é o incremento na direção  $x$ ,  $\Delta y$  é o incremento na direção  $y$  e  $\Delta t$  é o passo de tempo. Assim,

$$T_{4,2}^7$$

nos dá uma aproximação numérica da temperatura no ponto  $(4\Delta x, 2\Delta y)$  e no tempo  $7\Delta t$ . Ou seja,

$$T_{4,2}^7 \approx T(x=4\Delta x, y=2\Delta y, t=7\Delta t). \quad (74)$$

O domínio é dividido em  $N_x$  partes iguais na direção  $x$  e em  $N_y$  partes iguais na direção  $y$ . Temos, portanto,  $(N_x+1) \times (N_y+1)$  pontos no domínio discretizado. A figura (17) mostra o domínio discretizado para o caso em que  $N_x = N_y = 5$ . Temos 36 pontos no total, como mostra a figura. Os pontos pretos na figura pertencem às paredes. Nesse problema, a temperatura nesses pontos já está determinada pelas condições de contorno (já estão resolvidos). Os pontos vermelhos, no interior do domínio, serão resolvidos com uma aproximação da equação diferencial parcial governante.

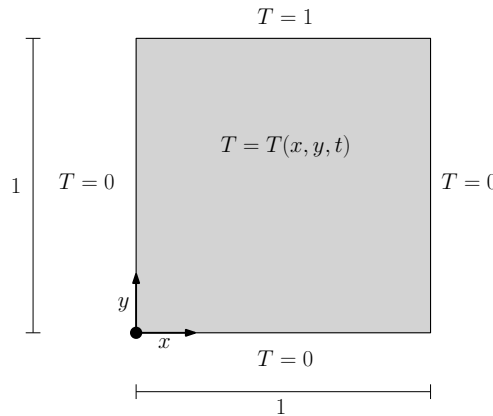


Figura 16: Placa quadrada de lado 1.

**Em cada tempo  $k$  temos uma matriz de temperaturas:**

$$T^k = \begin{bmatrix} T_{0,0}^k & T_{0,1}^k & T_{0,2}^k & T_{0,3}^k & T_{0,4}^k & T_{0,5}^k \\ T_{1,0}^k & T_{1,1}^k & T_{1,2}^k & T_{1,3}^k & T_{1,4}^k & T_{1,5}^k \\ T_{2,0}^k & T_{2,1}^k & T_{2,2}^k & T_{2,3}^k & T_{2,4}^k & T_{2,5}^k \\ T_{3,0}^k & T_{3,1}^k & T_{3,2}^k & T_{3,3}^k & T_{3,4}^k & T_{3,5}^k \\ T_{4,0}^k & T_{4,1}^k & T_{4,2}^k & T_{4,3}^k & T_{4,4}^k & T_{4,5}^k \\ T_{5,0}^k & T_{5,1}^k & T_{5,2}^k & T_{5,3}^k & T_{5,4}^k & T_{5,5}^k \end{bmatrix} \quad (75)$$

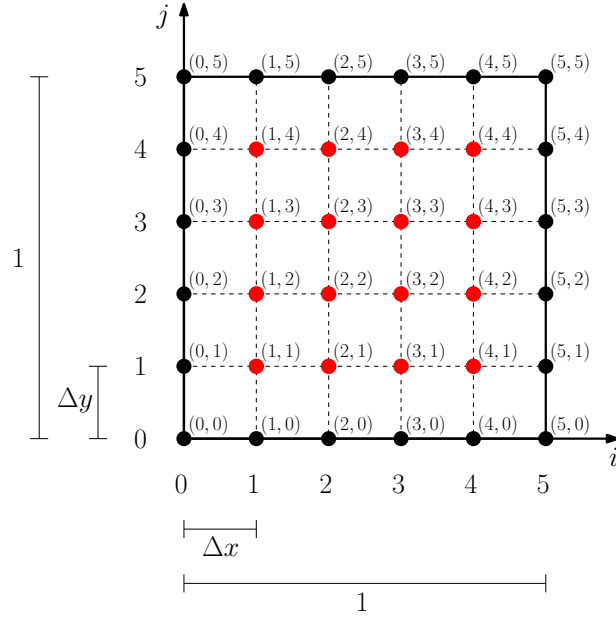


Figura 17: Domínio da placa discretizado. Os pontos pretos pertencem às paredes, enquanto os pontos vermelhos fazem parte do interior do domínio. Nesse caso:  $N_x = N_y = 5$ .

Note que uma linha na matriz corresponde a um conjunto de pontos na vertical da placa, e uma coluna na matriz corresponde a uma estrutura de pontos horizontais na placa. A figura (18) ilustra isso. É importante levar esse fato em consideração na hora de plotar os gráficos dos resultados.

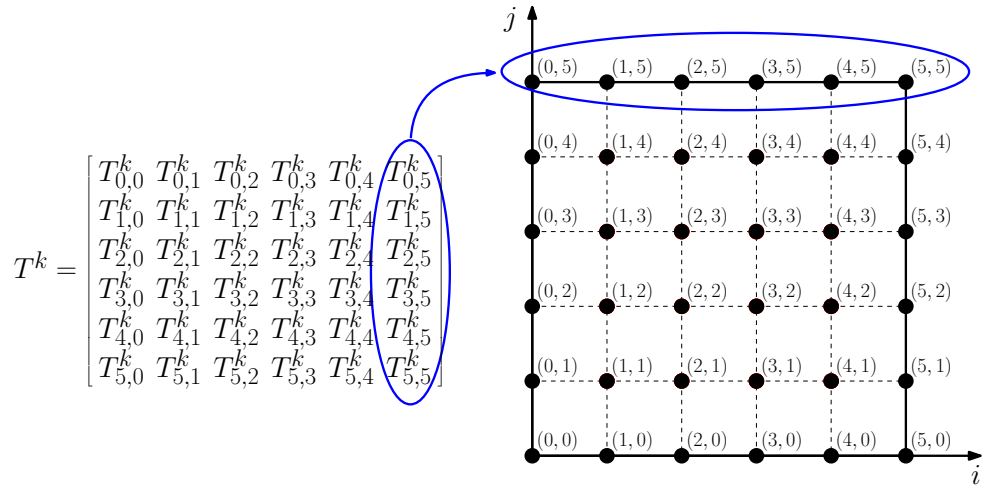


Figura 18: Correspondência entre pontos da matriz e pontos na placa.

Pelas condições de contorno, as temperaturas na parede já estão definidas. Portanto, temos para todos os tempos  $k$ :

$$T^k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & T^k_{1,1} & T^k_{1,2} & T^k_{1,3} & T^k_{1,4} & 1 \\ 0 & T^k_{2,1} & T^k_{2,2} & T^k_{2,3} & T^k_{2,4} & 1 \\ 0 & T^k_{3,1} & T^k_{3,2} & T^k_{3,3} & T^k_{3,4} & 1 \\ 0 & T^k_{4,1} & T^k_{4,2} & T^k_{4,3} & T^k_{4,4} & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (76)$$

Em  $k = 0$ , a temperatura é dada pela condição inicial:



$$T^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (77)$$

Pronto, já temos o domínio bidimensional discreto e a matriz de temperaturas em cada tempo (com as condições de contorno e com a condição inicial). Agora vamos aproximar as derivadas usando séries de Taylor.

## 7.2 Aproximando as Derivadas

Vamos aproximar as seguintes derivadas da equação original:

$$\frac{\partial T}{\partial t}, \quad \frac{\partial^2 T}{\partial x^2} \quad \text{e} \quad \frac{\partial^2 T}{\partial y^2}. \quad (78)$$

Considere a expansão em série de Taylor para a temperatura em  $t_k + \Delta t$  em torno de  $t_k$  (em um ponto  $(x_i, y_j)$  qualquer fixo):

$$T(x_i, y_j, t_k + \Delta t) = T(x_i, y_j, t_k) + \Delta t \left. \frac{\partial T}{\partial t} \right|_{x_i, y_j, t_k} + \frac{\Delta t^2}{2} \left. \frac{\partial^2 T}{\partial t^2} \right|_{x_i, y_j, t_k} + \frac{\Delta t^3}{3!} \left. \frac{\partial^3 T}{\partial t^3} \right|_{x_i, y_j, t_k} + \dots \quad (79)$$

Mas  $T(x_i, y_j, t_k + \Delta t) = T_{i,j}^{k+1}$  e  $T(x_i, y_j, t_k) = T_{i,j}^k$ . Assim, reescrevendo:

$$T_{i,j}^{k+1} = T_{i,j}^k + \Delta t \left. \frac{\partial T}{\partial t} \right|_{x_i, y_j, t_k} + \frac{\Delta t^2}{2} \left. \frac{\partial^2 T}{\partial t^2} \right|_{x_i, y_j, t_k} + \frac{\Delta t^3}{3!} \left. \frac{\partial^3 T}{\partial t^3} \right|_{x_i, y_j, t_k} + \dots \quad (80)$$

Isolando a primeira derivada de  $T$  com relação a  $t$  temos:

$$\left. \frac{\partial T}{\partial t} \right|_{x_i, y_j, t_k} = \frac{T_{i,j}^{k+1} - T_{i,j}^k}{\Delta t} - \frac{\Delta t}{2} \left. \frac{\partial^2 T}{\partial t^2} \right|_{x_i, y_j, t_k} - \frac{\Delta t^2}{3!} \left. \frac{\partial^3 T}{\partial t^3} \right|_{x_i, y_j, t_k} + \dots \quad (81)$$

Podemos escrever:

$$\left. \frac{\partial T}{\partial t} \right|_{x_i, y_j, t_k} = \frac{T_{i,j}^{k+1} - T_{i,j}^k}{\Delta t} + R(\Delta t) \quad (82)$$

Ou

$$\left. \frac{\partial T}{\partial t} \right|_{x_i, y_j, t_k} \approx \frac{T_{i,j}^{k+1} - T_{i,j}^k}{\Delta t}. \quad (83)$$

**Essa é a aproximação da primeira derivada temporal de uma função  $T(x, y, t)$  no ponto  $(x_i, y_j)$  e no tempo  $t_k$  usando diferenças finitas.**

Considere agora a expansão em série de Taylor para a temperatura em  $(x_i + \Delta x, y_j)$  em torno de  $(x_i, y_j)$  em um tempo  $t_k$  qualquer fixo:

$$T(x_i + \Delta x, y_j, t_k) = T(x_i, y_j, t_k) + \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, y_j, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} + \frac{\Delta x^3}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, y_j, t_k} + \dots \quad (84)$$

Mas  $T(x_i + \Delta x, y_j, t_k) = T_{i+1,j}^k$  e  $T(x_i, y_j, t_k) = T_{i,j}^k$ . Assim, reescrevendo essa equação temos:

$$T_{i+1,j}^k = T_{i,j}^k + \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, y_j, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} + \frac{\Delta x^3}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, y_j, t_k} + \dots \quad (85)$$

Considere novamente a série de Taylor em torno de  $(x_i, y_j)$ , mas agora para a função  $T$  em  $(x_{i-1}, y_j)$ :

$$T_{i-1,j}^k = T_{i,j}^k - \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, y_j, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} - \frac{\Delta x^3}{3!} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, y_j, t_k} + \dots \quad (86)$$

Somando as equações (85) e (86) temos:

$$\begin{aligned} T_{i+1,j}^k + T_{i-1,j}^k = & 2T_{i,j}^k + \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, y_j, t_k} - \Delta x \left. \frac{\partial T}{\partial x} \right|_{x_i, y_j, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} + \frac{\Delta x^2}{2} \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} + \\ & + \frac{\Delta x^3}{6} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, y_j, t_k} - \frac{\Delta x^3}{6} \left. \frac{\partial^3 T}{\partial x^3} \right|_{x_i, y_j, t_k} + \frac{\Delta x^4}{24} \left. \frac{\partial^4 T}{\partial x^4} \right|_{x_i, y_j, t_k} + \frac{\Delta x^4}{24} \left. \frac{\partial^4 T}{\partial x^4} \right|_{x_i, y_j, t_k} + \dots \end{aligned} \quad (87)$$

Isolando a segunda derivada:

$$\left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} = \frac{T_{i+1,j}^k - 2T_{i,j}^k + T_{i-1,j}^k}{\Delta x^2} + R(\Delta x^2). \quad (88)$$

Truncando a função  $R$ , resulta na aproximação usando diferenças finitas centradas da segunda derivada da função  $T(x, y, t)$  no ponto  $(x_i, y_j)$  e em  $t_k$ :

$$\left. \frac{\partial^2 T}{\partial x^2} \right|_{x_i, y_j, t_k} \approx \frac{T_{i+1,j}^k - 2T_{i,j}^k + T_{i-1,j}^k}{\Delta x^2}. \quad (89)$$

De maneira análoga, para a derivada segunda com relação a  $y$  temos:

$$\left. \frac{\partial^2 T}{\partial y^2} \right|_{x_i, y_j, t_k} \approx \frac{T_{i,j+1}^k - 2T_{i,j}^k + T_{i,j-1}^k}{\Delta y^2}. \quad (90)$$

### 7.3 Substituindo as na Equação Original

Substituindo as aproximações obtidas, equações (83), (89) e (90), na equação original, (69), temos

$$\frac{T_{i,j}^{k+1} - T_{i,j}^k}{\Delta t} = \frac{T_{i+1,j}^k - 2T_{i,j}^k + T_{i-1,j}^k}{\Delta x^2} + \frac{T_{i,j+1}^k - 2T_{i,j}^k + T_{i,j-1}^k}{\Delta y^2}. \quad (91)$$

Essa é a equação de diferenças finitas para esse problema.

### 7.4 Resolvendo as Equações

Na equação (91) já conhecemos todas as temperaturas no tempo atual. Para determinar a temperatura no tempo futuro, basta isolar o termo que está em  $k+1$  no lado esquerdo. Disso resulta:

$$T_{i,j}^{k+1} = T_{i,j}^k + \left( \frac{\Delta t}{\Delta x^2} \right) (T_{i+1,j}^k - 2T_{i,j}^k + T_{i-1,j}^k) + \left( \frac{\Delta t}{\Delta y^2} \right) (T_{i,j+1}^k - 2T_{i,j}^k + T_{i,j-1}^k). \quad (92)$$

Essa é a equação principal.

Como exemplo, vamos calcular a temperatura no primeiro passo de tempo,  $k = 1$ , considerando  $\Delta t = 0.01$ . Nesse caso,  $i$  e  $j$  vão variar de 1 a 4, pois os pontos dos cantos já estão resolvidos. Começando com  $i = 1$  e  $j = 1$ , temos

$$T_{1,1}^1 = T_{1,1}^0 + \left( \frac{0.01}{0.2^2} \right) (T_{2,1}^0 - 2T_{1,1}^0 + T_{0,1}^0) + \left( \frac{0.01}{0.2^2} \right) (T_{1,2}^0 - 2T_{1,1}^0 + T_{1,0}^0) \quad (93)$$

Substituindo os valores das temperaturas:

$$T_{1,1}^1 = 0 + \left( \frac{0.01}{0.2^2} \right) (0 - 0 + 0) + \left( \frac{0.01}{0.2^2} \right) (0 - 0 + 0) = 0 \quad (94)$$

Esse ponto vai continuar com temperatura igual a zero no tempo com  $k = 1$ . Isso acontece porque ele está próximo à parede de baixo, que tem temperatura zero. A informação sobre a existência de uma parede quente ainda não chegou nesse ponto.

Vamos calcular o valor da temperatura no ponto  $i = 3$  e  $j = 4$  (ponto próximo à parede de cima):

$$T_{3,4}^1 = T_{3,4}^0 + \left(\frac{0.01}{0.2^2}\right) (T_{4,4}^0 - 2T_{3,4}^0 + T_{2,4}^0) + \left(\frac{0.01}{0.2^2}\right) (T_{3,5}^0 - 2T_{3,4}^0 + T_{3,3}^0) . \quad (95)$$

Substituindo os valores das temperaturas, resulta

$$T_{3,4}^1 = 0 + \left(\frac{0.01}{0.2^2}\right) (0 - 0 + 0) + \left(\frac{0.01}{0.2^2}\right) (1 - 0 + 0) = 0.25 . \quad (96)$$

O valor da temperatura mudou nesse ponto. Como ele está próximo à parede quente, a informação já chegou nele. Resolvendo para todos os outros pontos, temos a matriz de temperaturas no tempo  $k = 1$ :

$$T^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 1 \\ 0 & 0 & 0 & 0 & 0.25 & 1 \\ 0 & 0 & 0 & 0 & 0.25 & 1 \\ 0 & 0 & 0 & 0 & 0.25 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (97)$$

Para encontrar a temperatura no tempo  $k = 2$ , devemos resolver a equação (92) com  $k = 1$ . E para encontrar as temperaturas nos tempos seguintes, é só fazer  $k = 2, 3, \dots$ .

A equação (92) é a que nós vamos implementar no nosso código.

## 7.5 Comentário 1

Note que para calcular o valor de  $T_{3,4}$  nós utilizamos  $T_{4,4}$ ,  $T_{2,4}$ ,  $T_{3,5}$  e  $T_{3,3}$ , além do próprio  $T_{3,4}$ . De uma maneira geral, cada ponto precisa do ponto de baixo, de cima, da esquerda e da direita, além dele mesmo. A representação esquemática dessa operação é feita por meio do **estêncil**, como mostra a figura (19).

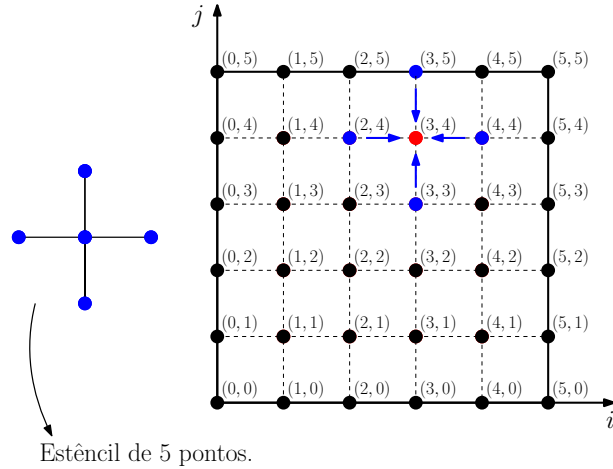


Figura 19: Estêncil e domínio computacional.

## 7.6 Comentário 2

Mesmo nesse caso bidimensional, poderíamos ter optado por trabalhar com um **vetor de temperaturas** no lugar de uma matriz. A figura (20) mostra como seria o arranjo dos pontos nesse caso. A temperatura, em um dado tempo  $k$ , seria dada por

$$T^k = \begin{bmatrix} T_0^k & T_1^k & T_2^k & T_3^k & T_4^k & T_5^k & T_6^k & T_7^k & T_8^k & T_9^k & T_{10}^k & T_{11}^k & T_{12}^k & T_{13}^k & T_{14}^k & T_{15}^k & T_{16}^k & T_{17}^k & \dots \\ \dots & T_{18}^k & T_{19}^k & T_{20}^k & T_{21}^k & T_{22}^k & T_{23}^k & T_{24}^k & T_{25}^k & T_{26}^k & T_{27}^k & T_{28}^k & T_{29}^k & T_{30}^k & T_{31}^k & T_{32}^k & T_{33}^k & T_{34}^k & T_{35}^k \end{bmatrix}$$

E a condição inicial é

$$T^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

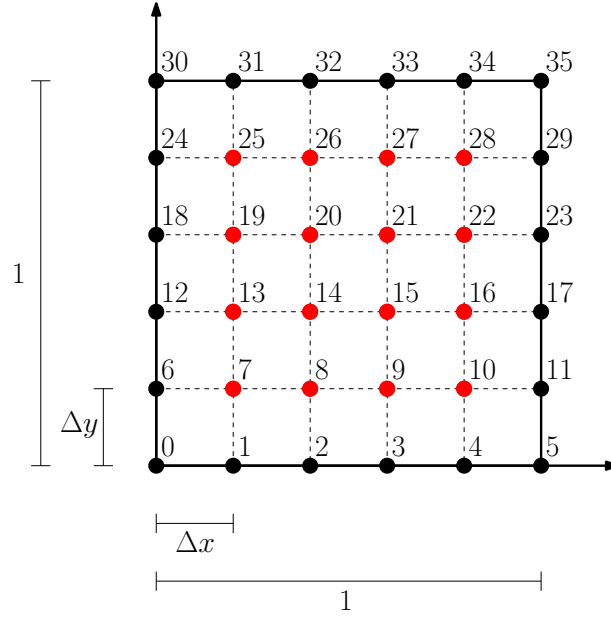


Figura 20: Opção com um vetor de temperaturas.

Essa forma de organização (com vetor) é um pouco mais trabalhosa na hora de programar, mas possui algumas vantagens computacionais. **Vamos usar a forma de matrizes por ser mais didática e mais fácil de implementar.**

## 8 Implementação do Problema Bidimensional Transiente no Python

Vamos implementar a solução numérica desenvolvida na seção anterior no Python. Os passos podem ser encontrados no anexo.

## 9 Problema Unidimensional Transiente: Condições de Contorno de Neumann

Considere agora o seguinte problema unidimensional transiente:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < 1, \quad t > 0, \quad (98)$$

com

$$T(x, 0) = x, \quad 0 < x < 1 \quad (99)$$

$$T(0, t) = 0, \quad t \geq 0 \quad (100)$$

$$\frac{\partial T}{\partial x}(1, t) = 0, \quad t > 0. \quad (101)$$

Temos agora, no lado direito da barra, uma condição de contorno para a **derivada da temperatura**, e não para a temperatura. Condições de contorno desse tipo são chamadas de **Condições de Contorno de Neumann**.

Fisicamente, a condição de contorno de derivada está relacionada com o fluxo de calor que entra na barra (veja a lei de Fourier). Assim, nesse problema, o lado direito da barra não permite que calor entre ou saia.

Existem várias formas de aplicar essa condição de contorno. Veremos 3 delas. Vamos considerar novamente o caso com  $N = 5$ , mostrado na figura (21).

**É importante notar que  $T_5$  não é mais constante! Vai variar no tempo.**

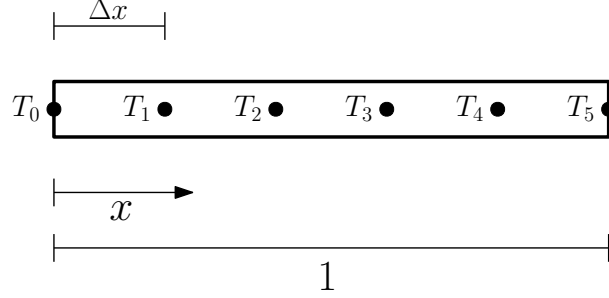


Figura 21: Barra unidimensional discretizada, com  $N = 5$ .

### 9.1 Opção 1: Aproximação de Primeira Ordem

Vamos aproximar a derivada no ponto  $i = 5$  como

$$\left. \frac{\partial T}{\partial x} \right|_{i=5} \approx \frac{T_5 - T_4}{\Delta x} = 0 \quad (102)$$

Daí resulta que

$$T_5 = T_4 \quad (103)$$

Nesse caso, resolve normalmente até  $i = 4$  e depois iguala  $T_5$  a  $T_4$ .

### 9.2 Opção 2: Aproximação de Segunda Ordem

Vamos aproximar a derivada no ponto  $i = 5$  como

$$\left. \frac{\partial T}{\partial x} \right|_{i=5} \approx \frac{T_3 - 4T_4 + 3T_5}{2\Delta x} = 0 \quad (104)$$

Daí resulta que

$$T_5 = \frac{4}{3}T_4 - \frac{1}{3}T_3. \quad (105)$$

Aqui, vamos calcular normalmente até  $i = 4$  e depois vamos usar  $T_3$  e  $T_4$  para atualizar o valor de  $T_5$ . (Observação: a aproximação da derivada primeira usando dois pontos pra trás foi obtida a partir de combinações de séries de Taylor. Vamos ver isso com mais calma em outro momento.)

### 9.3 Opção 3: Aproximação de Segunda Ordem com Ponto Fantasma

Imagine agora que existe um ponto 6 na barra, no qual a temperatura é  $T_6$ . A figura (22) mostra esse ponto extra. Esse ponto é chamado de Ponto Fantasma ou *Ghost Point* e é utilizado como um artifício para calcular a derivada em  $i = 5$ .

A equação geral de diferenças finitas em  $i = 5$  é

$$T_5^{k+1} = T_5^k + \frac{\Delta t}{\Delta x^2} (T_6^k - 2T_5^k + T_4^k). \quad (106)$$

Uma aproximação da derivada primeira em  $i = 5$ , usando os pontos vizinhos, é dada por

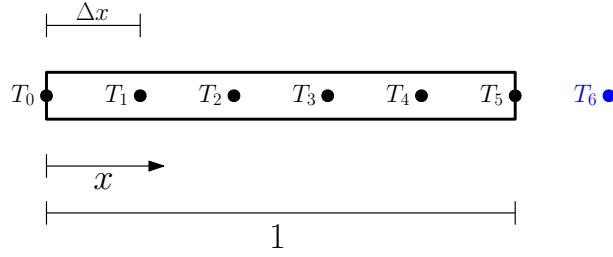


Figura 22: Barra unidimensional discretizada, com  $N = 5$  e *Ghost Point*.

$$\left. \frac{\partial T}{\partial x} \right|_{i=5} \approx \frac{T_6 - T_4}{2\Delta x} = 0, \quad (107)$$

o que resulta em

$$T_6 = T_4. \quad (108)$$

Substituindo essa relação em (106), resulta

$$T_5^{k+1} = T_5^k + \frac{\Delta t}{\Delta x^2} (2T_4^k - 2T_5^k). \quad (109)$$

Essa relação será utilizada para atualizar a temperatura  $T_5$ . Note que o *ghost point* não aparece explicitamente na equação.

## 9.4 Implementação no Python

Vamos implementar a solução numérica desenvolvida nessa seção no Python. Os passos podem ser encontrados no anexo.

## Bibliografia

CENGEL, Y. A.; GHAJAR, A. J. *Transferência de Calor e Massa: Uma Abordagem Prática*. 4. ed. Nova Iorque: McGraw-Hill, 2012.  
 HOFFMAN, J. D. *Numerical Methods for Engineers and Scientists*, 1. ed. Nova Iorque: McGraw-Hill, 1992.

## 10 ANEXOS

# Equação do Calor - Implementação

March 10, 2021

## 1 Problema Unidimensional Transiente

Vamos resolver o seguinte problema usando diferenças finitas:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < 1, \quad t > 0, \quad (1)$$

com

$$T(x, 0) = 0, \quad 0 < x < 1 \quad (2)$$

$$T(0, t) = 0, \quad t \geq 0 \quad (3)$$

$$T(1, t) = 1, \quad t \geq 0, \quad (4)$$

A equação de diferenças é dada por:

$$T_i^{k+1} = T_i^k + \frac{\Delta t}{\Delta x^2} \left( T_{i+1}^k - 2T_i^k + T_{i-1}^k \right). \quad (5)$$

Abaixo temos uma implementação em Python.

```
[1]: # Importando os módulos que serão
    # utilizados nos códigos.

    # Para criar os vetores e matrizes.
    import numpy as np

    # Para plotar os gráficos.
    import matplotlib.pyplot as plt
    from matplotlib.ticker import AutoMinorLocator
    from matplotlib import cm

    # Para plotar os gráficos 3D.
    from mpl_toolkits import mplot3d
```

```

# Módulo timeit: vai ser usado apenas para
# dar a duração da simulação no computador,
# em segundos.
import timeit

```

```

[2]: # Número de pontos no domínio.
N = 5

# Valor de Delta x.
dx = 1.0/N

# Tempo inicial.
t = 0.0

# Tempo final.
t_final = 0.1

# Passo de tempo.
dt = 0.01

# Criando o vetor de temperaturas.
Temp = np.zeros(N+1,float)

# Aplicando a condição de contorno
# do lado direito da barra.
Temp[N] = 1.0

# Esse é um vetor auxiliar que será
# utilizado para avançar a temperatura
# no tempo.
Temp_new = np.copy(Temp)

# Loop no tempo. Essa é a parte principal
# principal do código.
while t < t_final-0.01*dt:

    # Loop no espaço.
    for i in range(1,N):
        # Calculando a nova temperatura.
        Temp_new[i] = Temp[i] + (dt/(dx**2.0))*(Temp[i+1]-2.0*Temp[i]+Temp[i-1])

    # Essa é uma outra maneira de fazer o
    # loop acima. O resultado é o mesmo,
    # mas dessa maneira aqui é mais rápido,
    # pois utiliza as otimizações do numpy.
    Temp_new[1:N] = Temp[1:N] + (dt/(dx**2.0))*(Temp[2:N+1]-2.0*Temp[1:
↪N]+Temp[0:N-1])

```



```

# Atualizando a temperatura.
Temp = np.copy(Temp_new)

# Atualizando o tempo.
t += dt

# Printando o resultado na tela, apenas
# para acompanhar mesmo.
print(f'Tempo : {t:.3f}')

# Configurações para printar o valor da
# Temperatura.
np.set_printoptions(precision=3)

# Printando o resultado final
print(f'\nA Temperatura no tempo {t:.3f} é {Temp}')

# Para plotar vamos precisar do vetor x.
x = np.linspace(0.0,1.0,N+1)

# Agora vamos fazer o gráfico da solução.

# Deixando a fonte parecida com a do Latex.
plt.rc('text', usetex=True)
plt.rc('font', family='serif')

#Criando a figura.
fig = plt.figure()

# Dando nome aos eixos.
ax = fig.add_subplot(111)
ax.set_xlabel('$x$', fontsize = 24)
ax.set_ylabel('$T$', fontsize = 24)

# Colocando as marcas.
plt.xticks(size = 22)
plt.yticks(size = 22)
minor_locator = AutoMinorLocator(5)
ax.xaxis.set_minor_locator(minor_locator)
minor_locator = AutoMinorLocator(5)
ax.yaxis.set_minor_locator(minor_locator)
plt.tick_params(which='both', width=0.8)
plt.tick_params(which='major', length=5)
plt.tick_params(which='minor', length=2.5)

```

```

# Título do gráfico.
plt.title(f'Temperatura para $t =\$ {t:.2f}, com $\Delta t =\$ {dt:.3f}\n',
         ↳ fontsize = 24)

# Limites dos eixos na hora de plotar.
plt.axis((-0.1, 1.1, -0.1, 1.1))

# Plotando.
plt.plot(x, Temp, 'ro')

# Exportando o gráfico como pdf.
plt.savefig('figura1.pdf', format='pdf', dpi=1200, bbox_inches='tight')

# Mostrando o gráfico.
plt.show()

```

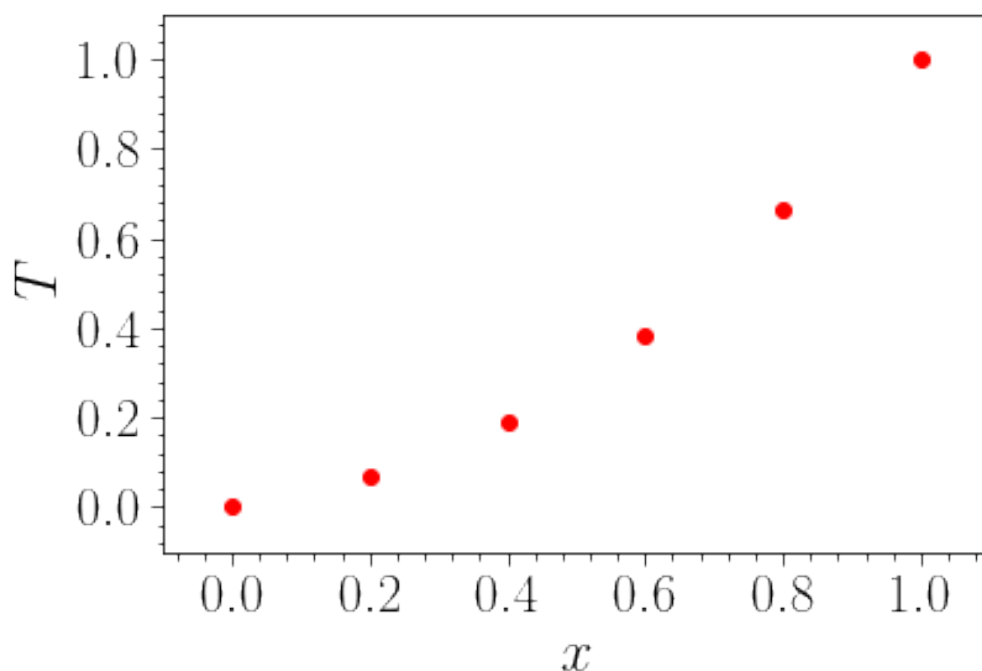
```

Tempo : 0.010
Tempo : 0.020
Tempo : 0.030
Tempo : 0.040
Tempo : 0.050
Tempo : 0.060
Tempo : 0.070
Tempo : 0.080
Tempo : 0.090
Tempo : 0.100

```

A Temperatura no tempo 0.100 é [0. 0.071 0.188 0.383 0.664 1. ]

Temperatura para  $t = 0.10$ , com  $\Delta t = 0.010$



## 2 Problema Bidimensional Transiente

Vamos resolver o seguinte problema usando diferenças finitas:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}, \quad 0 < x < 1, \quad 0 < y < 1, \quad t > 0, \quad (6)$$

com

$$T(x=0, y, t \geq 0) = 0, \quad T(x=1, y, t \geq 0) = 0, \quad 0 \leq y \leq 1 \quad (7)$$

$$T(x, y=0, t) = 0, \quad T(x, y=1, t) = 1, \quad 0 < x < 1 \quad (8)$$

$$T(x, y, t=0) = 0, \quad 0 < x, y < 1 \quad (9)$$

A equação de diferenças é dada por:

$$T_{i,j}^{k+1} = T_{i,j}^k + \left( \frac{\Delta t}{\Delta x^2} \right) (T_{i+1,j}^k - 2T_{i,j}^k + T_{i-1,j}^k) + \left( \frac{\Delta t}{\Delta y^2} \right) (T_{i,j+1}^k - 2T_{i,j}^k + T_{i,j-1}^k) . \quad (10)$$

Abaixo temos uma implementação em Python.

```
[3]: # Essa é o momento do início da simulação.
# Vai ser usado para calcular a duração da simulação no
# computador. Não tem nada a ver com o tempo usado
# nas diferenças finitas ou com o tempo da equação do calor.
# É apenas para termos uma noção de quanto tempo demora a
# simulação no computador. Esse é o Custo Computacional.
start = timeit.default_timer()

# Número de intervalos na direção x.
N_x = 5

# Número de intervalos na direção y.
N_y = 5

# Valor de Delta x.
dx = 1.0/N

# Valor de Delta y.
dy = 1.0/N

# Tempo inicial.
t = 0.0

# Tempo final.
t_final = 0.1

# Passo de tempo.
dt = 0.01

# Criando a matriz de temperaturas.
Temp = np.zeros((N_x+1,N_y+1),float)

# Aplicando a condição de contorno
# do lado de cima da placa.
for i in range(1,N_x):
    Temp[i,N_y] = 1.0

# A condição de contorno pode ser
# aplicada dessa maneira.
#Temp[1:N_x,N_y] = 1.0

# Essa é uma matriz auxiliar que será
```

```

# utilizado para avançar a temperatura
# no tempo.
Temp_new = np.copy(Temp)

# Loop no tempo. Essa é a parte principal
# principal do código.
while t < t_final-0.01*dt:

    # Loop no espaço.
    for i in range(1,N_x):
        for j in range(1,N_y):
            # Calculando a nova temperatura.
            Temp_new[i,j] = Temp[i,j] + (dt/(dx**2.0))*(Temp[i+1,j]
                -2.0*Temp[i,j]+Temp[i-1,j]) \
            + (dt/(dy**2.0))*(Temp[i,j+1]
                -2.0*Temp[i,j]+Temp[i,j-1]) \

    # Atualizando a temperatura.
    Temp = np.copy(Temp_new)

    # Atualizando o tempo.
    t += dt

    # Printando o resultado na tela, apenas
    # para acompanhar mesmo.
    #print(f'Tempo : {t:.3f}')

# Configurações para printar o valor da
# Temperatura.
np.set_printoptions(precision=3)

# Printando o resultado final
print(f'\nA Temperatura no tempo {t:.3f} é: \n\n{Temp}')

# Aqui é o momento do final da simulação.
end = timeit.default_timer()

# O tempo de simulação no computador então é
# end - start, que é mostrado na tela. Esse é
# o "custo computacional".
print(f'\nO custo computacional foi de {end-start:6.4f} s.')

```

A Temperatura no tempo 0.100 é:

```

[[0.    0.    0.    0.    0.    0.   ]
 [0.    0.03  0.085 0.198 0.438 1.   ]

```

```
[0.    0.047 0.13  0.288 0.569 1.    ]
[0.    0.047 0.13  0.288 0.569 1.    ]
[0.    0.03  0.085 0.198 0.438 1.    ]
[0.    0.    0.    0.    0.    0.    ]]
```

O custo computacional foi de 0.0041 s.

Temos diversas opções para plotar o resultado. Podemos fazer gráficos de contorno ou gráfico tridimensionais. Abaixo, vamos ver algumas dessas opções.

Primeiro os gráficos de contorno.

```
[4]: # Vamos precisar dos vetores x e y.
x = np.linspace(0.0, 1.0, N_x+1)
y = np.linspace(0.0, 1.0, N_y+1)

# Plotando os resultados.

#Gráfico de contorno.

# Esses são os intervalos para plotar os contornos.
# Temos 10 intervalos de 0.0 a 1.0 .
intervals = np.linspace(0.0,1.0,5)

#Comando para plotar a figura.
fig = plt.figure()

# Dando nome aos eixos.
ax = fig.add_subplot(111)

# Plotando os contornos da solução numérica com cor preta.
# Note que estou plotando a transposta de Temp, porque não
# definimos os pontos (i,j) no domínio físico
# da mesma forma como definimos (i,j) na matriz Temp.

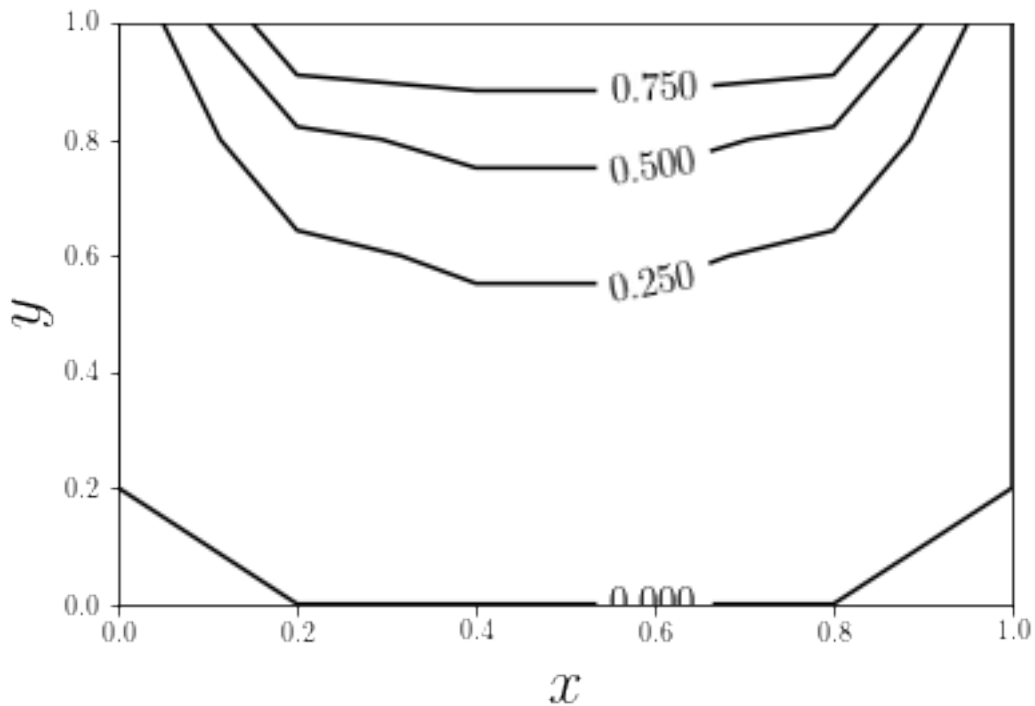
# Algumas opções.
contours = plt.contour(x, y, np.transpose(Temp), intervals, colors = 'black')
#contours = plt.contour(x, y, np.transpose(Temp), colors = 'black')
#contours = plt.contour(x, y, np.transpose(Temp))
#contours = plt.contourf(x, y, np.transpose(Temp))
#contours = plt.contourf(x, y, np.transpose(Temp), cmap='RdGy')

# Colocando os valores dos contornos no gráfico.
plt.clabel(contours, inline=True, fontsize=15, colors = 'k')

# Dando nomes aos eixos.
ax.set_xlabel('$x$', fontsize = 24)
ax.set_ylabel('$y$', fontsize = 24)
```

```
# Exportando o gráfico como pdf.
plt.savefig('figura2.pdf', format='pdf', dpi=1200, bbox_inches='tight')

# Mostrando o gráfico.
plt.show()
```



Agora o gráfico 3D.

```
[5]: # Agora o gráfico 3D.
fig = plt.figure()

# Projeção 3D.
ax = plt.axes(projection='3d')

# Vamos precisar gerar uma malha.
X,Y = np.meshgrid(x,y)

# Nomes dos eixos.
ax.set_xlabel('$x$', fontsize = 24)
ax.set_ylabel('$y$', fontsize = 24)
ax.set_zlabel('$T$', fontsize = 24)

# Existem diversas maneiras de plotar 3D.
```

```

# Aqui estão alguns exemplos.

# Exemplo 1.
#ax.contourf3D(X, Y, np.transpose(Temp), 100)

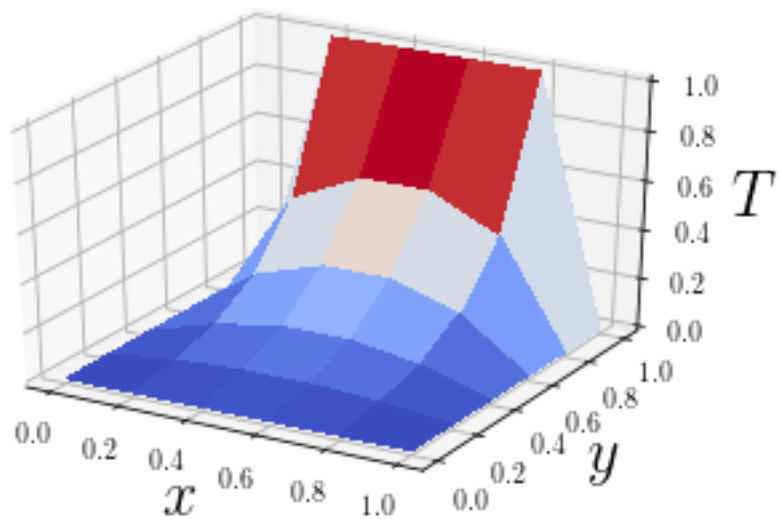
#Exemplo 2.
#ax.plot_wireframe(X,Y,np.transpose(Temp))

#Exemplo 3.
ax.dist = 11
ax.plot_surface(X, Y, np.transpose(Temp), cmap=cm.coolwarm,
               linewidth=0, antialiased=False)
#ax.plot_surface(X, Y, np.transpose(Temp), cmap=cm.inferno,
#               linewidth=0, antialiased=False)

# Exportando o gráfico como pdf.
plt.savefig('figura3.pdf', format='pdf', dpi=1200, bbox_inches='tight')

# Mostrando os gráficos.
plt.show()

```





### 3 Problema Unidimensional com Condição de Contorno de Neumann

Vamos resolver o seguinte problema

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} , \quad 0 < x < 1, \quad t > 0 , \quad (11)$$

com

$$T(x, 0) = x , \quad 0 < x < 1 \quad (12)$$

$$T(0, t) = 0 , \quad t \geq 0 \quad (13)$$

$$\frac{\partial T}{\partial x}(1, t) = 0 , \quad t > 0 . \quad (14)$$

A equação de diferenças é dada por

$$T_i^{k+1} = T_i^k + \frac{\Delta t}{\Delta x^2} \left( T_{i+1}^k - 2T_i^k + T_{i-1}^k \right) . \quad (15)$$

Vamos implementar a condição de contorno de derivada de 3 formas diferentes.

#### 3.1 Opção 1: Aproximação de Primeira Ordem

$$T_5^{k+1} = T_4^{k+1} \quad (16)$$

```
[6]: N = 5

dx = 1.0/N

t = 0.0

t_final = 0.2

dt = 0.01

# Criando o vetor de temperaturas.
# Temp_1 se refere à opção 1.
Temp_1 = np.zeros(N+1, float)

# Condição inicial.
Temp_1 = np.linspace(0.0, 1.0, N+1)
```

```

Temp_1_new = np.copy(Temp_1)

while t < t_final-0.01*dt:

    # Loop no espaço.
    for i in range(1,N):
        # Calculando a nova temperatura.
        Temp_1_new[i] = Temp_1[i] + (dt/(dx**2.0))*(Temp_1[i+1]-2.
↪0*Temp_1[i]+Temp_1[i-1])

    # Atualizando o ponto N.
    Temp_1_new[N] = Temp_1_new[N-1]

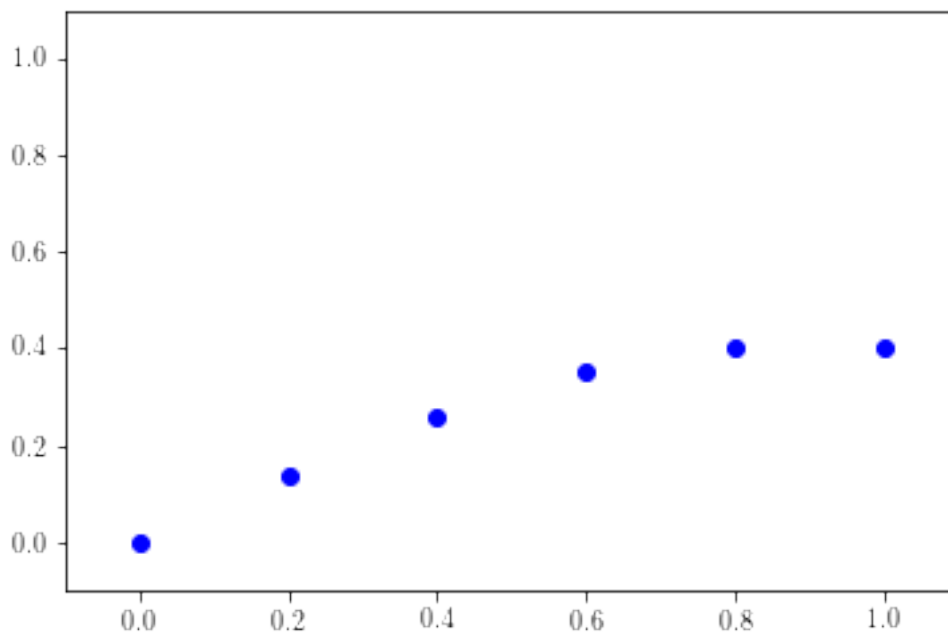
    # Atualizando o vetor Temp_1.
    Temp_1 = np.copy(Temp_1_new)

    # Atualizando o tempo.
    t += dt

x = np.linspace(0.0,1.0,N+1)

plt.axis((-0.1, 1.1, -0.1, 1.1))
plt.plot(x,Temp_1,'bo')
plt.show()

```



### 3.2 Opção 2: Aproximação de Segunda Ordem

$$T_5 = \frac{4}{3}T_4 - \frac{1}{3}T_3 . \quad (17)$$

```
[7]: t = 0.0

# Criando o vetor de temperaturas.
# Temp_2 se refere à opção 2.
Temp_2 = np.zeros(N+1,float)

# Condição inicial.
Temp_2 = np.linspace(0.0,1.0,N+1)

Temp_2_new = np.copy(Temp_2)

while t < t_final-0.01*dt:

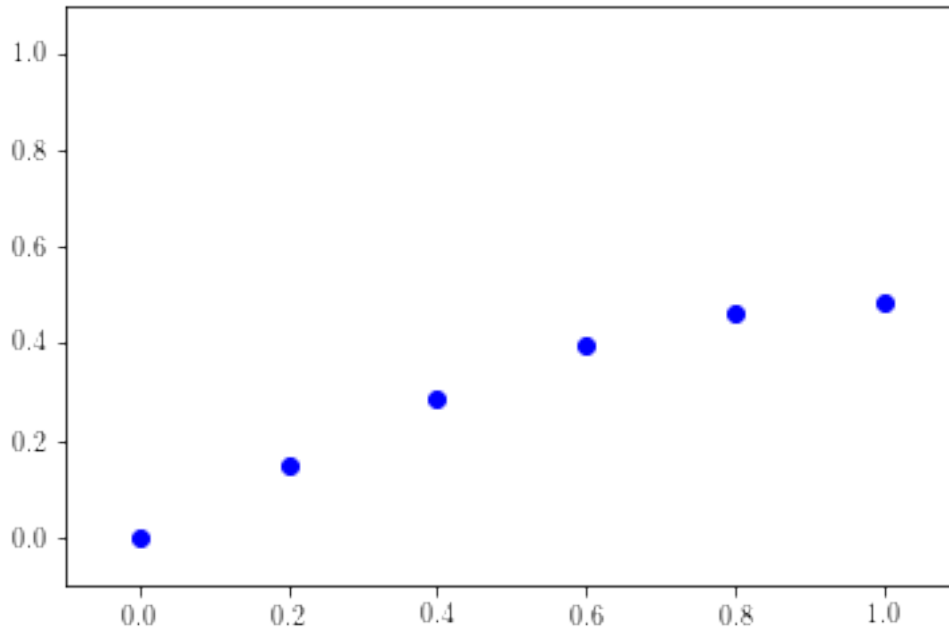
    # Loop no espaço.
    for i in range(1,N):
        # Calculando a nova temperatura.
        Temp_2_new[i] = Temp_2[i] + (dt/(dx**2.0))*(Temp_2[i+1]-2.
↪ 0*Temp_2[i]+Temp_2[i-1])

    # Atualizando o ponto N.
    Temp_2_new[N] = (4.0/3.0)*Temp_2_new[N-1] - (1.0/3.0)*Temp_2_new[N-2]

    # Atualizando o vetor Temp_1.
    Temp_2 = np.copy(Temp_2_new)

    # Atualizando o tempo.
    t += dt

plt.axis((-0.1, 1.1, -0.1, 1.1))
plt.plot(x,Temp_2,'bo')
plt.show()
```



### 3.3 Opção 3: Aproximação de Segunda Ordem com *Ghost Point*

$$T_5^{k+1} = T_5^k + \frac{\Delta t}{\Delta x^2} \left( 2T_4^k - 2T_5^k \right) . \quad (18)$$

```
[8]: t = 0.0

# Criando o vetor de temperaturas.
# Temp_3 se refere à opção 3.
Temp_3 = np.zeros(N+1,float)

# Condição inicial.
Temp_3 = np.linspace(0.0,1.0,N+1)

Temp_3_new = np.copy(Temp_3)

while t < t_final-0.01*dt:

    # Loop no espaço.
    for i in range(1,N):
        # Calculando a nova temperatura.
        Temp_3_new[i] = Temp_3[i] + (dt/(dx**2.0))*(Temp_3[i+1]-2.
↪ 0*Temp_3[i]+Temp_3[i-1])

    # Atualizando o ponto N.
```

```

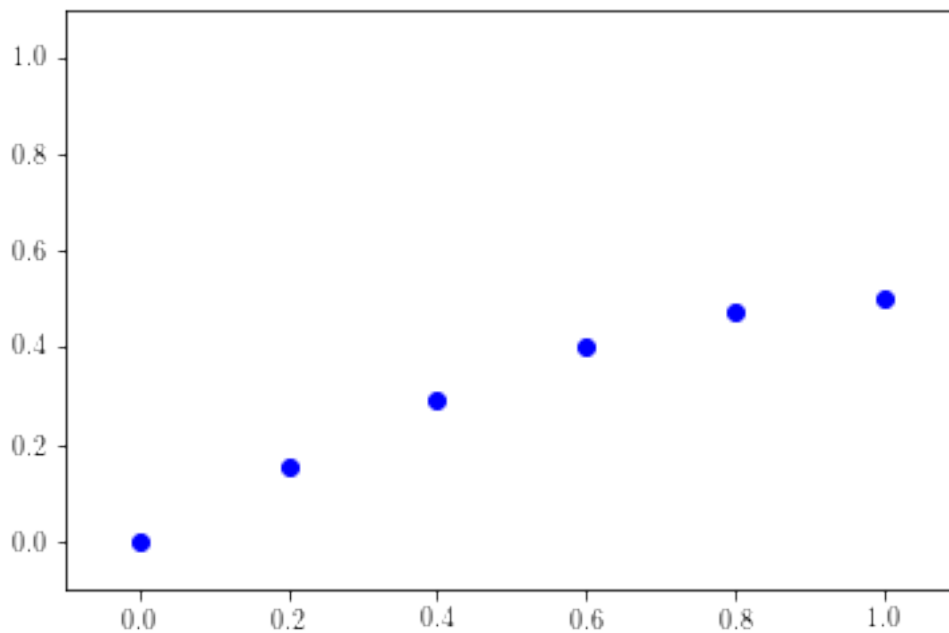
Temp_3_new[N] = Temp_3[N] + (dt/(dx**2.0))*(2.0*Temp_3[N-1]-2.0*Temp_3[N])

# Atualizando o vetor Temp_1.
Temp_3 = np.copy(Temp_3_new)

# Atualizando o tempo.
t += dt

plt.axis((-0.1, 1.1, -0.1, 1.1))
plt.plot(x,Temp_3,'bo')
plt.show()

```



Plotando as 3 soluções no mesmo gráfico.

```

[9]: plt.axis((-0.1, 1.1, -0.1, 1.1))
plt.plot(x,Temp_1,'ko')
plt.plot(x,Temp_2,'bo')
plt.plot(x,Temp_3,'ro')
plt.show()

```

