

UNIVERSIDADE DE SÃO PAULO

ESCOLA POLITÉCNICA

MAP3121 - MÉTODOS NUMÉRICOS E APLICAÇÕES (2020)

# Exercício Programa 1

NOMES

Elias Daleffi Rodrigues Rayes  
Vinícius Marchioli

NUSP

10823848  
10774232

# Sumário

<b>1</b>	<b>Primeira tarefa</b>	<b>2</b>
1.1	Item A . . . . .	2
1.2	Item B . . . . .	6
1.3	Item C . . . . .	8
<b>2</b>	<b>Segunda tarefa</b>	<b>10</b>
2.1	Item A . . . . .	10
2.2	Item B . . . . .	11
2.3	Item C . . . . .	14
	<b>Referências</b>	<b>18</b>
	<b>Apêndices</b>	<b>19</b>
A	LEIAME . . . . .	19
B	Código em MATLAB para gerar as curvas . . . . .	21
C	Curvas do item 1A . . . . .	32
D	Curvas do item 1B . . . . .	51
E	Curvas do item 1C . . . . .	70
F	Curvas do item 2B . . . . .	77
G	Curvas do item 2C . . . . .	93

# 1 Primeira tarefa

Nesta tarefa, será empregado o método de Euler explícito, descrito pela fórmula (1) para resolução numérica da equação diferencial de calor, a modo de obter a distribuição de temperatura em uma barra.

$$u_i^{k+1} = u_i^k + \Delta t \left( \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + f(x_i, t_k) \right), \quad i = 1, \dots, N-1, \text{ e } k = 0, \dots, M-1 \quad (1)$$

O dimensionamento da malha é representado por uma matriz  $M \times N$ , na qual  $\Delta t = \frac{1}{M}$  é a discretização para o tempo e  $\Delta x = \frac{1}{N}$  para o espaço. Tais valores poderão ser escolhidos em tempo de execução, conforme pedido pelo enunciado. Todos as saídas dos testes são apresentadas em arquivos *.txt*, com nomes apropriados, e, especificamente, para o item A, são fornecidos dois resultados, que correspondem às fontes  $f_{teste}(t, x) = 10x^2(x - 1) - 60xt + 20t$  e  $f(t, x) = 10 \cos(10t)x^2(1 - x)^2 - (1 + \sin(10t))(12x^2 - 12x + 2)$ .

## 1.1 Item A

Primeiramente, no caso exemplo com fonte  $f_{teste}(t, x) = 10x^2(x - 1) - 60xt + 20t$ , para que se possa verificar que a solução exata é  $u(t, x) = 10tx^2(x - 1)$ , deve-se levar em consideração a condição vista na equação (2).

$$u_t(t, x) = u_{xx}(t, x) + f(t, x) \text{ em } [0, T] \times [0, 1] \quad (2)$$

$$u(0, x) = u_0(x) \text{ em } [0, 1] \quad (3)$$

$$u(t, 0) = g_1(t) \text{ em } [0, T] \quad (4)$$

$$u(t, 1) = g_2(t) \text{ em } [0, T] \quad (5)$$

Analiticamente, como  $u_t(t, x) = 10x^2(x - 1)$  e  $u_{xx} = 40tx + 20t(x - 1)$ , sabe-se que a condição (2) é satisfeita em  $[0, T] \times [0, 1]$  e, portanto,  $u(t, x)$  é a solução exata.

O erro de truncamento, em relação à  $f_{teste}$ , leva em consideração as equações (6) e (7).

$$\tau(\Delta t, \Delta x) \leq C_1 \Delta t + C_2 \Delta x^2 \quad (6)$$

$$C_1 = \max_{[0,T] \times [0,1]} \left| \frac{u_{tt}(t, x)}{2} \right| \quad \text{e} \quad C_2 = \max_{[0,T] \times [0,1]} \left| \frac{u_{xxxx}(t, \bar{x})}{12} \right| \quad (7)$$

Neste caso, para qualquer discretização  $\Delta t$  e  $\Delta x$ ,  $\tau(\Delta t, \Delta x) = 0$ , visto que as constantes  $C_1 = C_2 = 0$ , ou seja, o erro de truncamento será sempre zero para o caso teste (com  $u(t, x) = 10tx^2(x - 1)$  e  $f_{teste}$ ). De fato, comparando-se a função exata à calculada, conclui-se que os erros obtidos são apenas de arredondamento (verificar arquivo *Erro1ATeste.txt*).

Posteriormente, considerando

$$f(t, x) = 10 \cos(10t)x^2(1 - x)^2 - (1 + \sin(10t))(12x^2 - 12x + 2)$$

e  $u_0(x) = x^2(1 - x)^2$ , a fim de estudar a funcionalidade do método de Euler explícito, cujo termo genérico é apresentado pela equação (1), foram elaborados alguns testes, alterando-se os valores de  $N$  e  $M$  (e consequentemente  $\lambda$ ), mas fixando-se  $T = 1$ .

Vale lembrar que, o número de passos deve seguir uma complexidade  $\mathcal{O}(N \cdot M)$ , e, como  $M = \frac{N^2}{\lambda}$ , ou seja,  $M \propto N^2$ , a complexidade temporal do algoritmo é da ordem de  $k \cdot N^3$ , em que  $k = \frac{1}{\lambda}$ . Portanto, considerando  $N = 640$  e  $N = 1280$ , a solução será encontrada em, no mínimo,  $2 \cdot 640^3$  e  $2 \cdot 1280^3$  passos, respectivamente.

Foram geradas curvas (Apêndice C) com  $N$  variando entre 10 e 320 e com  $\lambda = 0,25$  ou  $\lambda = 0,5$ . Percebe-se que, a partir de  $N = 20$ , não é possível notar diferenças significantes nas curvas com menores discretizações. O código em *MATLAB* utilizado pode ser visto no Apêndice B. A figura 1 mostra a evolução da temperatura em relação ao tempo, com passo de 0,1. Desse modo, verifica-se, numericamente, que  $u(t, x) = (1 + \sin(10t))x^2(1 - x)^2$  é solução exata para  $f$ .

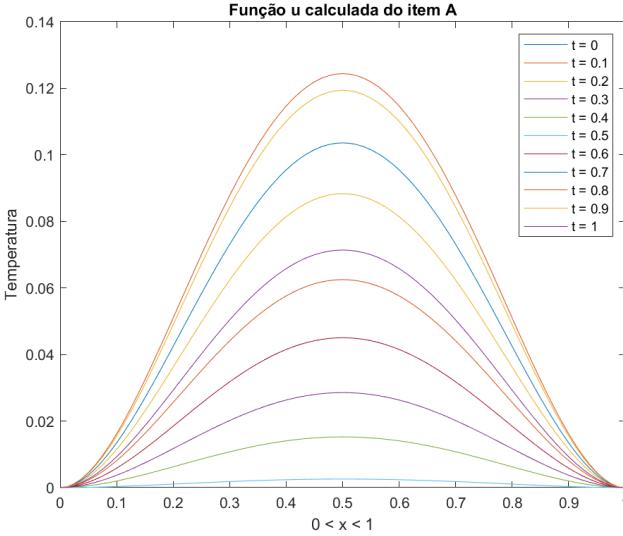


Figura 1: Solução do item A, com  $N = 80$  e  $\lambda = 0,5$

Respeitada a condição de  $\lambda \leq 0,5$ , nota-se que, com o refinamento da malha, o erro diminui, uma vez que o erro de truncamento cai proporcionalmente à soma referente à inequação (6). Ao tentar-se aplicar o método para  $\lambda = 0,51$ , percebe-se que o erro máximo encontrado cresce rapidamente, como visto na tabela 1. Ou seja, não há convergência para a solução exata. De fato, o método de euler explícito não se aplica para  $\lambda > 0,5$ .

$N$	Erro máximo
10	$5,312148 \cdot 10^{-3}$
20	$7,539084 \cdot 10^1$
40	$1,287565 \cdot 10^{40}$
80	$2,649958 \cdot 10^{198}$
160	$\infty$
320	$\infty$

Tabela 1: Erros máximos do item A para  $\lambda = 0,51$

Para que se possa observar o comportamento do erro para  $\lambda = 0,25$  e  $\lambda = 0,5$ , elaborou-se os gráficos das figuras 2 e 3. Percebe-se que o erro cai exponencialmente em ambos os casos, em função da discretização da malha. Dessa forma, observa-se fator de redução de aproximadamente 4 (para  $\lambda = 0,25$  e  $\lambda = 0,5$ ), conforme dobra-se  $N$ . Ou, ainda, de forma mais completa, pode-se inferir seu valor aproximado, para  $N$  genérico, a partir das fórmulas encontradas pelo ajuste exponencial.

Tais valores, em relação ao erro máximo encontrado partindo-se de  $f_{teste}$ , em que foi provado que não há erro de truncamento, são muito maiores. Isso ocorre pois o erro de ponto flutuante, quando são usados dados de 8 *bytes*, é muito pequeno, e, portanto, o truncamento da série de Taylor se manifesta de maneira mais evidente.

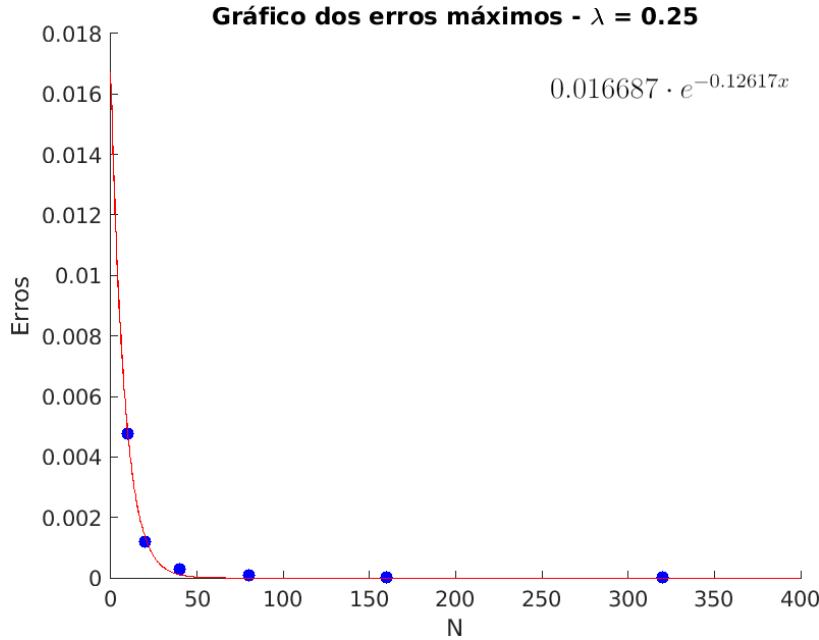


Figura 2: Ajuste exponencial com  $\lambda = 0,25$

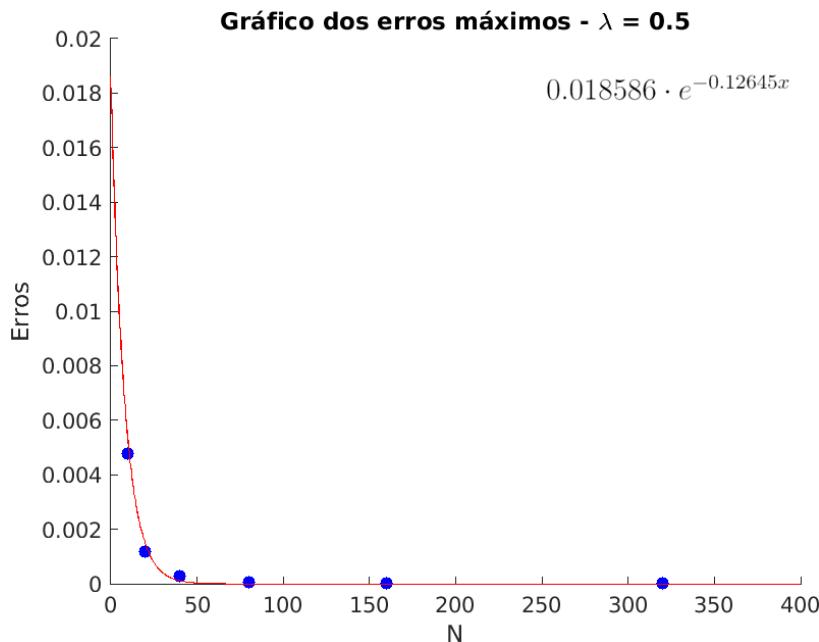


Figura 3: Ajuste exponencial com  $\lambda = 0.5$

## 1.2 Item B

Para que  $u(t, x) = e^{t-x} \cos(5tx)$  seja solução exata, deve-se respeitar as equações (2), (3), (4) e (5). Assim, temos que  $g_1(t) = e^t$  e  $g_2(t) = e^{t-1} \cos(5t)$  serão as condições de fronteira de Dirichlet e  $u_0(x) = e^{-x}$  a condição inicial do problema. Substituindo,

$$\begin{cases} u_t(t, x) = e^{t-x}(\cos(5tx) - 5x \sin(5tx)) \\ u_{xx}(t, x) = e^{t-x}(\cos(5tx) + 10t \sin(5tx) - 25t^2 \cos(5tx)) \end{cases} \quad \text{em (2)}$$

temos que  $f(t, x) = -5e^{t-x}(2t \sin(5tx) + x \sin(5tx) - 5t^2 \cos(5tx))$ .

Analogamente ao item A, considerando  $f(t, x)$  e as condições acima apontadas, pôde-se construir as curvas mostradas no apêndice D. Vê-se, novamente, que, a partir de  $N = 20$  e independentemente do valor de  $\lambda$  (desde que  $\lambda \leq 0,5$ ), os resultados encontrados não apresentam diferenças visuais significativas. A figura 4 mostra a evolução da temperatura em relação ao tempo, com passo de 0,1.

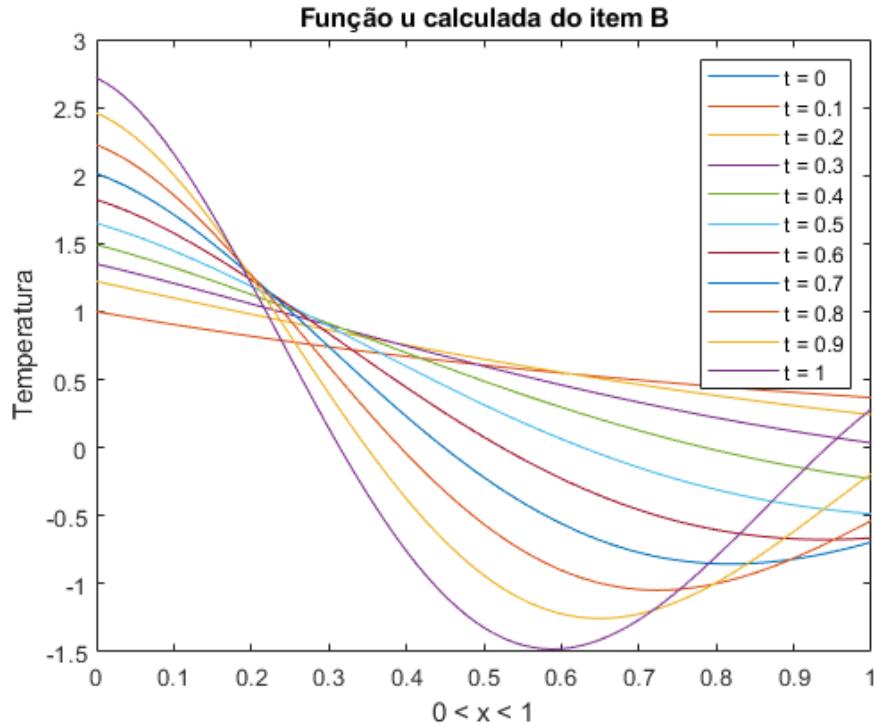


Figura 4: Solução do item B, com  $N = 80$  e  $\lambda = 0,5$

Para que se possa mais profundamente analisar os resultados obtidos, é possível visualizar, nas figuras 5 e 6, o comportamento do erro para  $\lambda = 0,25$  e  $\lambda = 0,5$ , respectivamente. Conforme dobra-se  $N$ , refinando-se a malha, têm-se que o erro máximo segue um fator de redução com valor 4, cuja forma para  $N$  genérico é uma exponencial decrescente.

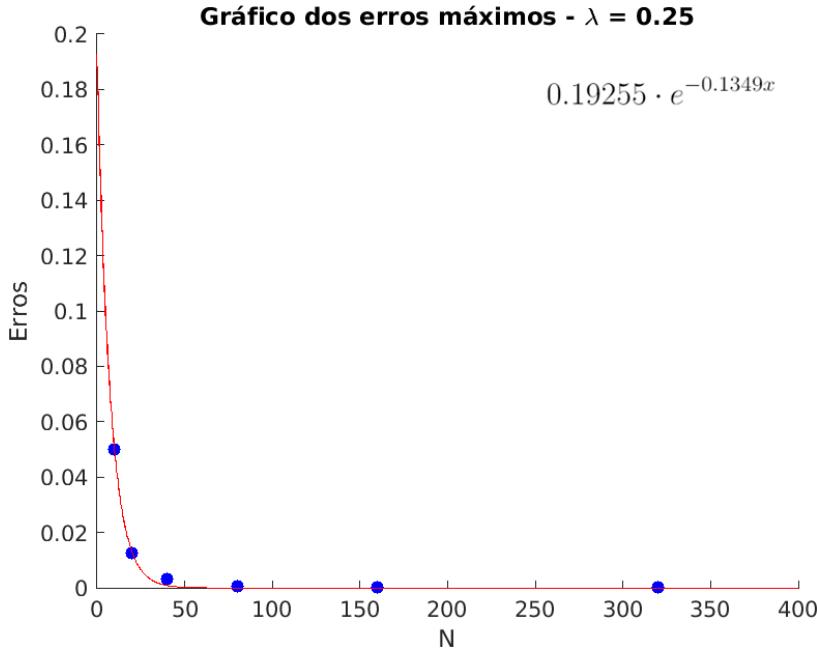


Figura 5: Ajuste exponencial com  $\lambda = 0,25$

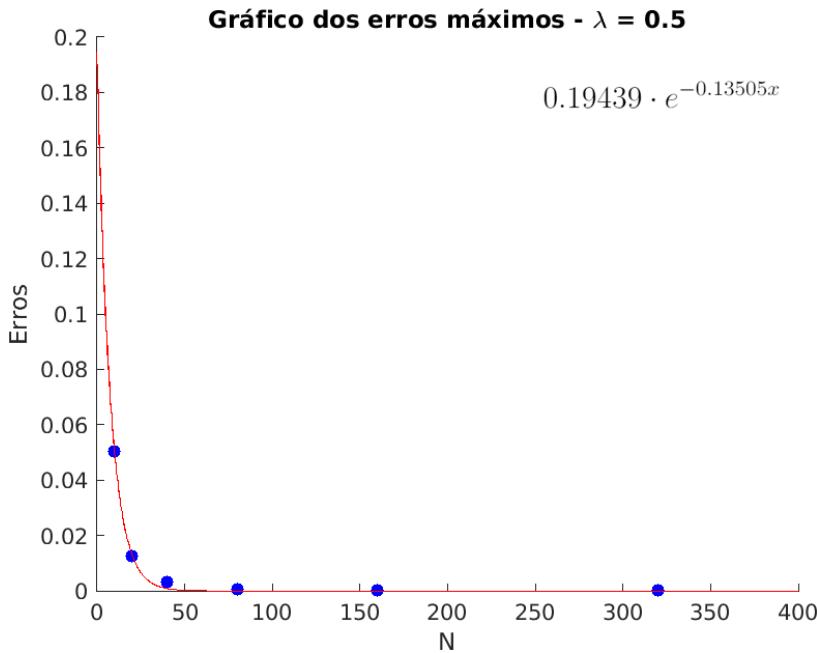


Figura 6: Ajuste exponencial com  $\lambda = 0,5$

Observando-se a tabela 2, nota-se que, caso  $\lambda > 0,5$ , conforme aumenta-se  $N$ , o erro máximo obtido aumenta significativamente, como ocorrido no item A.

N	Erro máximo
10	$5,046888 \cdot 10^{-2}$
20	$3,988129 \cdot 10^1$
40	$6,798130 \cdot 10^{39}$
80	$1,398343 \cdot 10^{198}$
160	$\infty$
320	$\infty$

Tabela 2: Erros máximos do item B para  $\lambda = 0,51$

### 1.3 Item C

No item C, deve-se encontrar solução exata com  $g_1(t) = g_2(t) = 0$ , implementando-se uma fonte pontual  $f(t, x) = r(t)g_h(x)$  em um ponto  $p$  do domínio com intensidade que varia conforme  $r(t) = 10000 * (1 - 2t^2)$ . Para tal, assume-se

$$g_h(x) = \begin{cases} \frac{1}{h}, & \text{se } p - \frac{h}{2} \leq x \leq p + \frac{h}{2} \\ 0, & \text{caso contrário.} \end{cases}$$

Adotando-se  $p = 0,25$  e  $h = \Delta x$ , tem-se que:

$$f(t, x) = \begin{cases} \frac{10000 \cdot (1 - 2t^2)}{\Delta x}, & \text{se } 0,25 - \frac{\Delta x}{2} \leq x \leq 0,25 + \frac{\Delta x}{2} \\ 0, & \text{caso contrário.} \end{cases}$$

Analogamente ao item A, considerando  $f(t, x)$  e as condições acima apontadas, pôde-se construir as curvas mostradas no apêndice E. Vê-se, novamente, que, a partir de  $N = 20$  e independentemente do valor de  $\lambda$  (desde que  $\lambda \leq 0,5$ ), os resultados encontrados não apresentam diferenças visuais significativas. A figura 7 mostra a evolução da temperatura em relação ao tempo, com passo de 0,1.

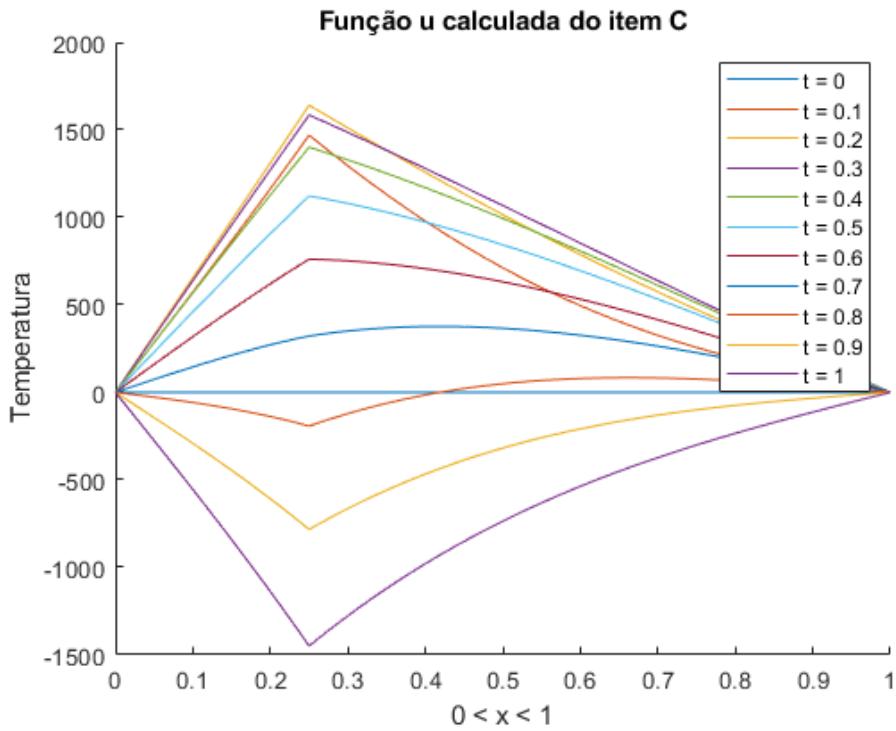


Figura 7: Solução do item C, com  $N = 80$  e  $\lambda = 0,5$

Desconhecida a solução exata, mesmo assumindo convergência do método numérico aplicado ( $\lambda \leq 0,5$ ), não é possível estabelecer uma análise de erros. De fato, caso houvesse a capacidade de estudá-lo, saberia-se, por consequência, a solução exata do problema.

## 2 Segunda tarefa

Considerando a impraticabilidade do método de Euler explícito desenvolvido na tarefa um, conforme faz-se o refinamento da malha, serão implementados métodos implícitos de maior eficiência, com  $\Delta t = \Delta x$ . Por definição, em um método implícito, a solução em um ponto de malha no novo instante depende também de outros valores no mesmo instante.

### 2.1 Item A

A fim de resolver os métodos implícitos para a tarefa dois, será necessário resolver um sistema tridiagonal simétrico. Deve-se, então, estabelecer uma rotina que calcula a decomposição  $A = LDL^t$  do sistema (8), em que  $L$  é uma matriz bidiagonal triangular unitária inferior e  $D$ , matriz diagonal.

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & l_{N-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{N-1} \end{bmatrix} \begin{bmatrix} 1 & l_2 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & l_{N-1} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \quad (8)$$

A função *solveLDLt*, implementada no programa, computa a solução  $x$  do sistema  $Ax = LDL^t x = b$ , desde que a matriz  $A$  seja positiva definida. É possível demonstrar<sup>1</sup> que a matriz de distribuição de temperatura, que envolve as condições estabelecidas pelo enunciado, obedece as condições para ser definida positivamente.

Primeiramente, a matriz  $A$  pode ser descrita por dois vetores, *diag* e *sub* de dimensão  $N-1$  (lembre-se que em C++, a indexação começa em 0). Assim, podemos determinar outros dois vetores, *d* e *l*, que descrevem as matrizes  $L$  e  $D$ , respectivamente, da seguinte maneira<sup>4</sup>:

$$d(i) = diag(i), \text{ para } i = 0 \quad (9)$$

$$d(i) = diag(i) - l^2(i-1) \cdot d(i-1), \text{ para } i \in [1, N-2] \quad (10)$$

$$l(i) = \frac{sub(i)}{d(i)}, \text{ para } i \in [0, N-3] \quad (11)$$

A solução da equação matricial pode ser feita em três partes, resolvendo sistemas lineares triangulares simples em cada passo. Podemos, então, decompor  $A = LDL^t$  em:

$$Lz = b$$

$$Dy = z$$

$$L^t x = y$$

Para a correta utilização da função implementada, deve-se criar um arquivo com nome *entrada.txt* cujas linhas seguem a formatação do apêndice A<sup>1</sup>. A saída será impressa em um arquivo nomeado *Output2A.txt*.

## 2.2 Item B

O método de Euler implícito, para um ponto específico da malha, pode ser visto na equação (12).

$$u_i^{k+1} = u_i^k + \lambda(u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}) + \Delta t f(x_i, t_{k+1}), \quad i = 1, \dots, N-1, \text{ e } k = 0, \dots, M-1 \quad (12)$$

Matricialmente, para cada instante, deve-se resolver a equação matricial vista em (13).

$$\begin{bmatrix} 1 + 2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1 + 2\lambda & -\lambda & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\lambda & 1 + 2\lambda & -\lambda \\ 0 & \cdots & 0 & -\lambda & 1 + 2\lambda \end{bmatrix} \begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ \vdots \\ u_{N-2}^{k+1} \\ u_{N-1}^{k+1} \end{bmatrix} = \begin{bmatrix} u_1^k + \Delta t f_1^{k+1} + \lambda g_1(t^{k+1}) \\ u_2^k + \Delta t f_2^{k+1} \\ \vdots \\ u_{N-2}^k + \Delta t f_{N-2}^{k+1} \\ u_{N-1}^k + \Delta t f_{N-1}^{k+1} + \lambda g_2(t^{k+1}) \end{bmatrix} \quad (13)$$

---

<sup>1</sup><https://github.com/eliasrr1/ep1-numerico>

Pode-se observar as soluções encontradas dos items A, B e C nos apêndices F, bem como os respectivos erros associados. As soluções exatas encontram-se nos apêndices C e D, uma vez que são as mesmas da primeira tarefa, geradas pelo MATLAB. Visualmente, a partir de  $N = M = 40$ , não é possível notar diferenças significativas nas curvas. Nas figuras 8, 9 e 10, a evolução da temperatura com passo de 0,1 pode ser observada.

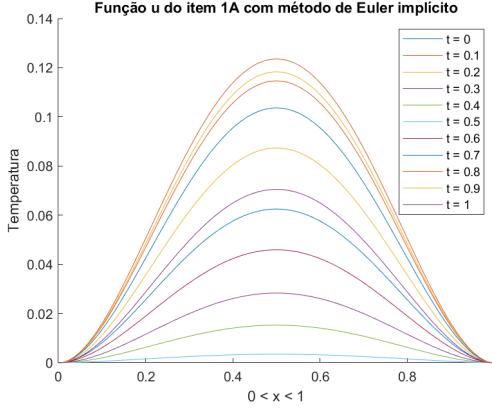


Figura 8: Solução para o item A, com  $N = M = 200$

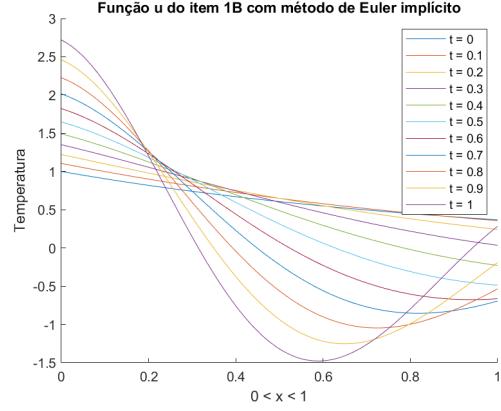


Figura 9: Solução para o item B, com  $N = M = 200$

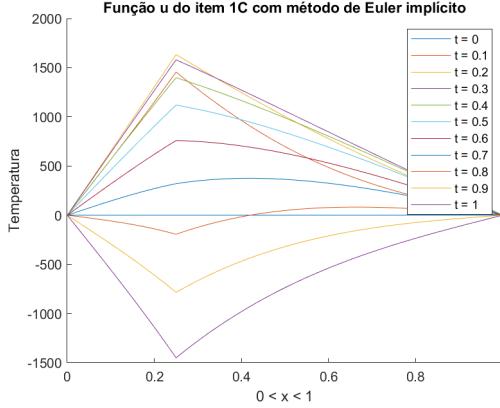


Figura 10: Solução para o item C, com  $N = M = 200$

Uma solução numérica é dita convergente de ordem  $n$  se o erro for da forma<sup>3</sup>, onde  $h$  é o passo da malha em uma direção:

$$E(h) := |u - u_{calc}| = Ch^n, \quad C \in \mathbb{R} \quad (14)$$

Portanto, quando observado que o erro de truncamento do método de Euler implícito é dado por (15), tem-se que a ordem de convergência em  $\Delta t$  é 1 e em  $\Delta x$  é 2.

$$\tau(\Delta t, \Delta x) = \max_{k,i} |\tau_i^k(\Delta t, \Delta x)| \leq C_1 \Delta t + C_2 \Delta x^2 \quad (15)$$

O comportamento do erro máximo está representado na figura 11. Novamente, vê-se decrescimento exponencial em função de N. Porém, não há fator de redução 4, como visto na primeira tarefa. Observando-se a tabela 3, pode-se ver seu comportamento.

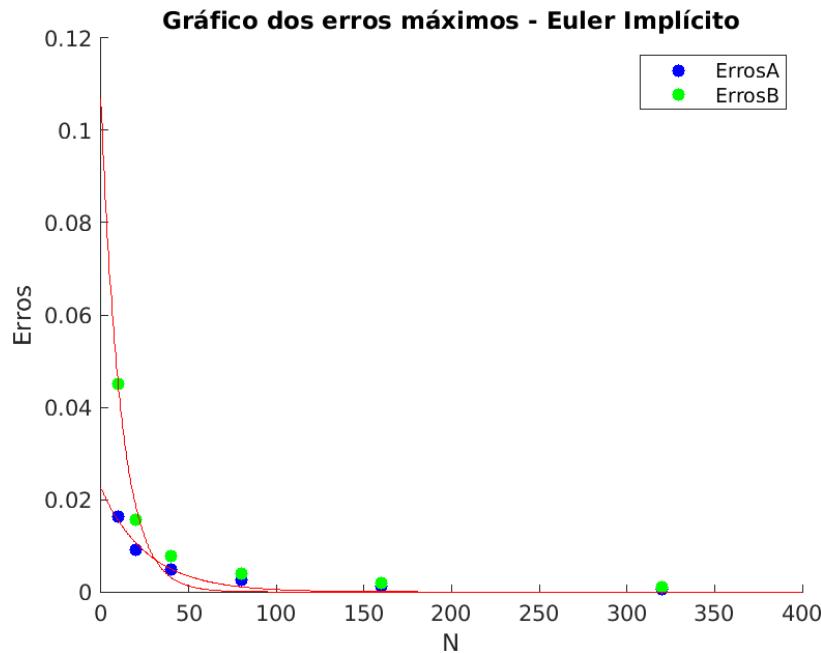


Figura 11: Ajuste exponencial para os erros

N=M	Erro máximo A	Fator de Redução A	Erro máximo B	Fator de Redução B
10	$1,626703 \cdot 10^{-2}$	-	$4,496773 \cdot 10^{-2}$	-
20	$9,194187 \cdot 10^{-3}$	$1,7692733$	$1,561894 \cdot 10^{-2}$	$2,8790513$
40	$4,944007 \cdot 10^{-3}$	$1,8596630$	$7,834641 \cdot 10^{-3}$	$1,9935744$
80	$2,581569 \cdot 10^{-3}$	$1,9151171$	$3,931040 \cdot 10^{-3}$	$1,9930199$
160	$1,319515 \cdot 10^{-3}$	$1,9564529$	$1,968850 \cdot 10^{-3}$	$1,9966173$
320	$6,673197 \cdot 10^{-4}$	$1,9773356$	$9,967567 \cdot 10^{-4}$	$1,9752563$

Tabela 3: Fator de redução para o método de Euler implícito, com  $T = 1$

## 2.3 Item C

O termo de malha para o método de Crank-Nicolson pode ser visto na equação (16).

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

Para cada instante de tempo, deve-se resolver o sistema matricial apresentado na equação (17).

$$\begin{bmatrix} 1 + \lambda & -\frac{\lambda}{2} & 0 & \cdots & 0 \\ -\frac{\lambda}{2} & 1 + \lambda & -\frac{\lambda}{2} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\frac{\lambda}{2} & 1 + \lambda & -\frac{\lambda}{2} \\ 0 & \cdots & 0 & -\frac{\lambda}{2} & 1 + \lambda \end{bmatrix} \begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ \vdots \\ u_{N-2}^{k+1} \\ u_{N-1}^{k+1} \end{bmatrix} = \begin{bmatrix} (1 - \lambda)u_1^k + \frac{\lambda}{2}(u_0^k + u_2^k) + \frac{\Delta t}{2}(f_1^k + f_1^{k+1}) + \frac{\lambda}{2}g_1(t^{k+1}) \\ (1 - \lambda)u_2^k + \frac{\lambda}{2}(u_1^k + u_3^k) + \frac{\Delta t}{2}(f_2^k + f_2^{k+1}) \\ \vdots \\ (1 - \lambda)u_{N-2}^k + \frac{\lambda}{2}(u_{N-3}^k + u_{N-1}^k) + \frac{\Delta t}{2}(f_{N-2}^k + f_{N-2}^{k+1}) \\ (1 - \lambda)u_{N-1}^k + \frac{\lambda}{2}(u_{N-2}^k + u_N^k) + \frac{\Delta t}{2}(f_{N-1}^k + f_{N-1}^{k+1}) + \frac{\lambda}{2}g_2(t^{k+1}) \end{bmatrix} \quad (17)$$

As soluções encontradas, bem como os erros associados a cada discretização, podem ser vistos nas figuras do apêndice G. As soluções exatas encontram-se nos apêndices C e D, uma vez que são as mesmas da primeira tarefa. Ainda, adotando-se passo temporal de 0,1, traçou-se as curvas das figuras 12, 13 e 14.

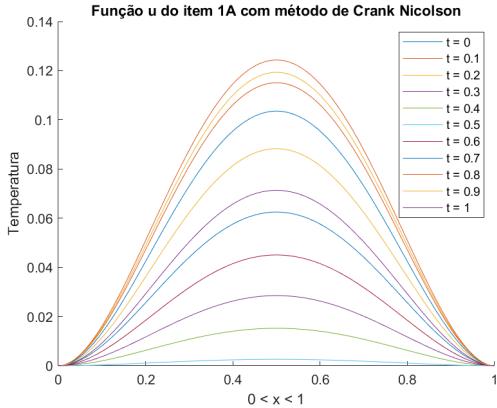


Figura 12: Solução para o item A, com  $N = M = 200$

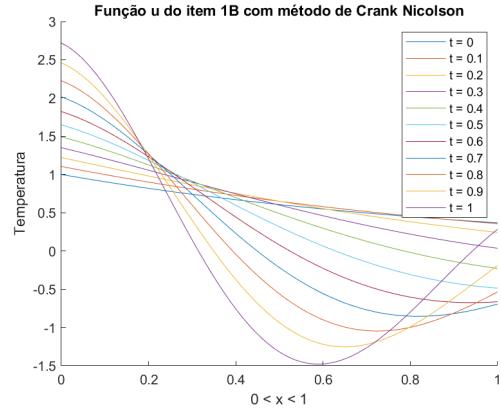


Figura 13: Solução para o item B, com  $N = M = 200$

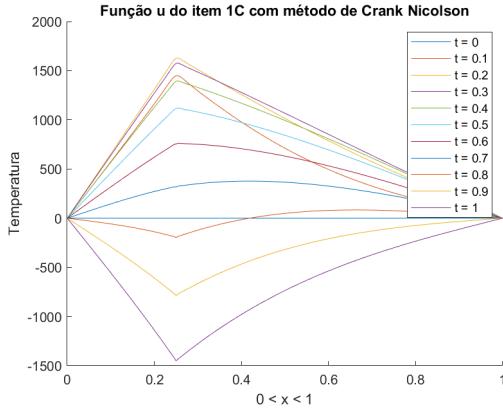


Figura 14: Solução para o item C, com  $N = M = 200$

Observam-se os resultados obtidos, para o método de Crank-Nicolson aplicado ao item C, notam-se oscilações significativas (vide figura 156 do apêndice G). Isso ocorre pois, como visto na figura 15, a intensidade da fonte apresenta descontinuidade em  $p$ .

Nessas condições, lembrando-se que  $\lambda > \frac{1}{2}$  (considerando que  $\Delta x = \Delta t$ ), o método tende a gerar oscilações espúrias<sup>5</sup>. É de se considerar, portanto, a utilização do método de Euler implícito, o qual é estável e imune à oscilações, para malhas pouco refinadas.

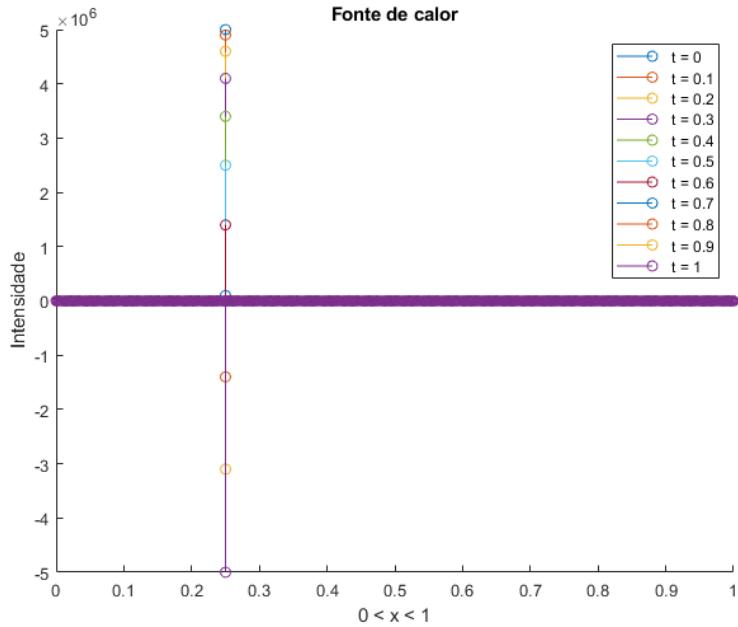


Figura 15: Descontinuidade da fonte pontual

O erro máximo, com  $T = 1$ , em função de  $N$ , pode ser visto na figura 16. Novamente, pode-se observar um comportamento exponencial decrescente para o erro, com fator de redução de aproximadamente 4, como na primeira tarefa.

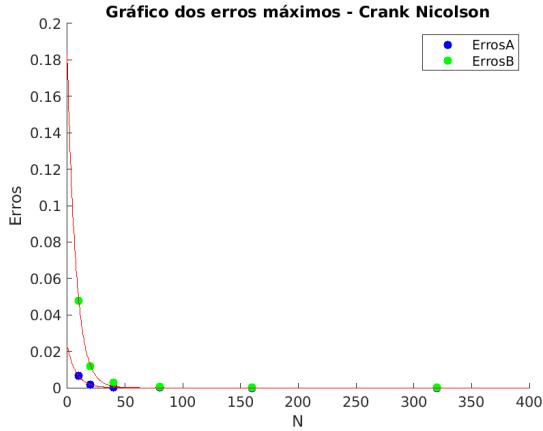


Figura 16: Ajuste exponencial

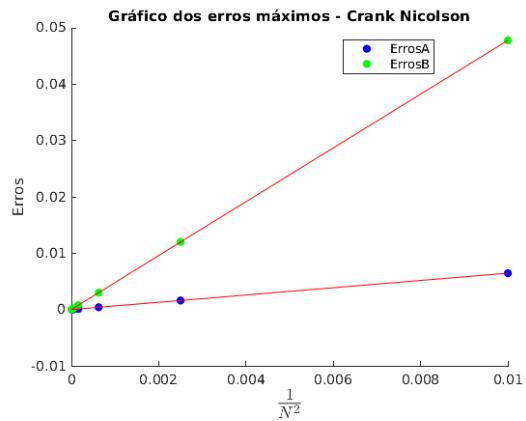


Figura 17: Ordem do erro

A análise de convergência analítica do método de Crank-Nicolson é complexa. Por essa razão, opta-se pela análise empírica. De acordo com a equação (14), pode-se determinar a ordem de convergência de um método a partir da sua relação de proporcionalidade com o passo.

Deve-se verificar que a ordem de convergência do método é 2, então repete-se o gráfico da figura 16, porém, troca-se a escala do eixo horizontal para o quadrado do passo. Observa-se, na figura 17, que tal relação é linear e, portanto, é possível concluir que o método possui, de fato, convergência de ordem 2 em  $\Delta x$  e  $\Delta t$ .

## Referências

- 1 ANDELIĆ, Milica; FONSECA, Carlos da. Sufficient conditions for positive definiteness of tridiagonal matrices revisited. **Positivity**, v. 15, mar. 2011. DOI: 10.1007/s11117-010-0047-y.
- 2 BEGOT, Vivien. **A possible solution to the spurious oscillations in finite difference methods for partial differential equations**. Ago. 2012. <https://www.lexifi.com/quant/possible-solution-spurious-oscillations-finite-difference-methods-partial-differential-equations/>. Acessado em 13 de Maio de 2020.
- 3 LEVEQUE, Randall J. **Finite difference methods for differential equations**. 2006.
- 4 NGUYEN, Duc. In: **Cholesky and LDL<sub>t</sub> Decomposition**. Jul. 2010. cap. 04.11.
- 5 ØSTERBY, Ole. Five Ways of Reducing the Crank–Nicolson Oscillations. **BIT Numerical Mathematics**, v. 43, n. 4, p. 811–822, nov. 2003. ISSN 1572-9125. DOI: 10.1023/B:BITN.0000009942.00540.94.

# Apêndices

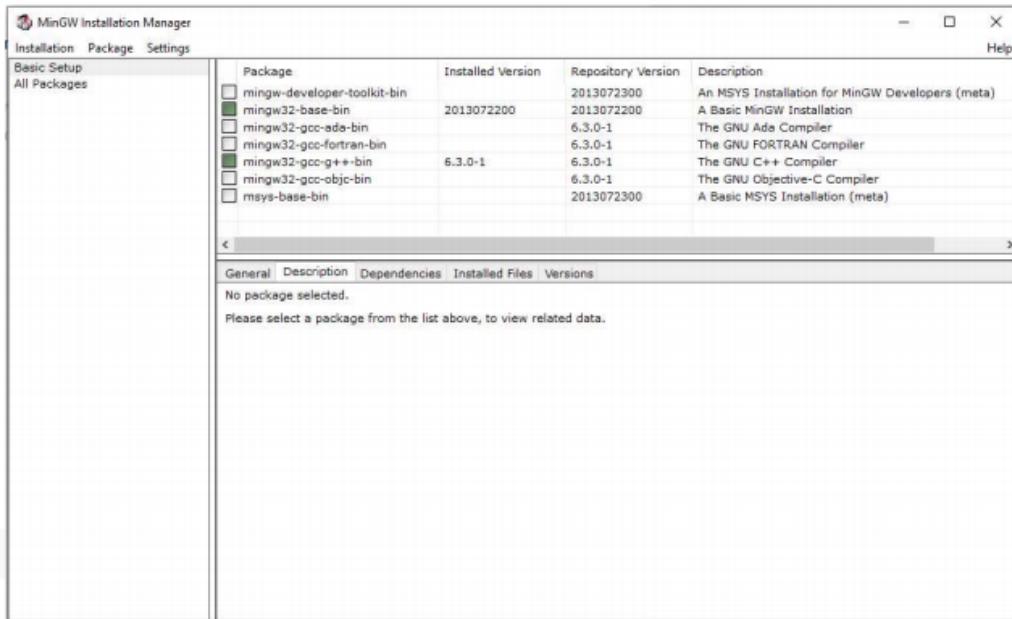
## A LEIAIME

MAP3121 - Métodos Numéricos e Aplicações (2020)

### Exercício Programa 1

#### Instruções de compilação

1. Instale o compilador de linguagem C/C++ disponível em [MINGW](#).
2. No instalador, selecione os seguintes pacotes:



3. Clique em *Installation* e selecione *Apply Changes*.
4. Para adicionar o compilador no PATH do Windows, siga as instruções do [link](#).
5. Coloque os seguintes arquivos em um mesmo diretório:
  - o main.cpp
  - o Tarefa.h
  - o Tarefa.cpp
6. Abra a janela de comando no caminho do diretório.
7. Execute o seguinte comando:

```
g++ -O2 -std=c++14 main.cpp Tarefa.cpp -o EP1.exe
```

---

### Instruções de uso

Para todos os itens o programa recebe os valores de N e M. Após execução, serão gerados arquivos texto com as saídas.

Tarefa 1		Tarefa 2	
Item	Arquivos de saída	Item	Arquivos de saída
1A	Output1A.txt Output1ATeste.txt Erro1A.txt Erro1ATeste.txt	2A (*)	Output2A.txt
1B	Output1B.txt Erro1B.txt	2B	Output2B.txt Erro2B.txt
1C	Output1C.txt	2C	Output2C.txt Erro2C.txt

(\*) Para o item 2A, considere um sistema linear na forma matricial  $Ax=b$ , onde A é uma matriz simétrica tridiagonal, positiva definida, de tamanho  $N-1 \times N-1$ , b é um vetor coluna de dimensão  $N-1$  e x é a solução do sistema. O programa recebe A e b e devolve a solução em um arquivo, para isso, siga os passos abaixo:

- Crie um arquivo de nome `entrada.txt`
- A primeira linha do arquivo deverá conter  $N-1$  elementos do vetor que representa a diagonal principal da matriz A.
- A segunda linha do arquivo deverá conter  $N-2$  elementos do vetor que representa a subdiagonal principal da matriz A.
- A terceira linha do arquivo deverá conter  $N-1$  elementos do vetor que representa o lado direito da equação, ou seja, o vetor coluna b.

Por exemplo, caso queira resolver o seguinte sistema 3x3, deve-se inserir o valor  $N = 4$ , e preencher o arquivo de entrada do modo descrito acima. O vetor resposta x será impresso no arquivo `Output2A.txt`

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

`entrada.txt`

```
2 2 2
-1 -1
1 0 0
```

`Output2A.txt`

```
7.500000e-001 5.000000e-001 2.500000e-001
```

## B Código em MATLAB para gerar as curvas

```
1 %% Tarefa 1
2
3 N = 10;
4 M = 200;
5
6 x = linspace(0, 1, N + 1);
7 t = linspace(0, 1, M + 1);
8
9 [X, T] = meshgrid(x, t);
10
11 % 1A Teste
12 uReal = (10 .* X .* X .* (X - 1) .* T);
13
14 figure(1)
15 surf(x, t, uReal)
16 title('Funcao u exata do item A de teste')
17 xlabel('0 < x < 1')
18 ylabel('0 < t < 1')
19 zlabel('Temperatura')
20 colormap('winter')
21 shading interp
22
23 figure(2)
24 R1aT = readmatrix('Output1ATeste.txt');
25 R1aT(:,end) = [];
26 surf(x, t, R1aT)
27 title('Funcao u calculada do item A de teste')
28 xlabel('0 < x < 1')
29 ylabel('0 < t < 1')
30 zlabel('Temperatura')
31 colormap('summer')
32 shading interp
33
```

```

34 figure(3)
35 errolaT = readmatrix('ErrolATeste.txt');
36 errolaT(:,end) = [];
37 surf(x, t, errolaT)
38 title('Erro do item A de teste')
39 xlabel('0 < x < 1')
40 ylabel('0 < t < 1')
41 zlabel('Temperatura')
42 colormap('spring')
43 shading interp
44
45 % 1A Real
46 uReal = ((1 + sin(10 .* T)) .* X .* X .* (1 - X) .* (1 - X));
47
48 figure(4)
49 surf(x, t, uReal)
50 title('Funcao u exata do item A')
51 xlabel('0 < x < 1')
52 ylabel('0 < t < 1')
53 zlabel('Temperatura')
54 shading interp
55
56 figure(5)
57 R1a = readmatrix('Output1A.txt');
58 R1a(:,end) = [];
59 surf(x, t, R1a)
60 title('Funcao u calculada do item A')
61 xlabel('0 < x < 1')
62 ylabel('0 < t < 1')
63 zlabel('Temperatura')
64 colormap('summer')
65 shading interp
66
67 figure(6)
68 errola = readmatrix('ErrolA.txt');
69 errola(:,end) = [];

```

```

70 surf(x, t, erro1a)
71 title('Erro do item A')
72 xlabel('0 < x < 1')
73 ylabel('0 < t < 1')
74 zlabel('Temperatura')
75 colormap('spring')
76 shading interp
77
78 figure(7)
79 hold
80 distancia = floor(M/10);
81 Ratual = R1a(1, :);
82 plot(x, Ratual)
83 for v = 1:10
84     Ratual = R1a(1+v*distancia, :);
85     plot(x, Ratual)
86 end
87 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
88 title('Funcao u calculada do item A')
89 xlabel('0 < x < 1')
90 ylabel('Temperatura')
91
92 % 1B
93 uReal = (exp(T - X) .* cos(5 .* T .* X));
94
95 figure(8)
96 surf(x, t, uReal)
97 title('Funcao u exata do item B')
98 xlabel('0 < x < 1')
99 ylabel('0 < t < 1')
100 zlabel('Temperatura')
101 shading interp
102
103 figure(9)
104 R1b = readmatrix('Output1B.txt');

```

```

105 R1b(:,end) = [];
106 surf(x, t, R1b)
107 title('Funcao u calculada do item B')
108 xlabel('0 < x < 1')
109 ylabel('0 < t < 1')
110 zlabel('Temperatura')
111 colormap('summer')
112 shading interp
113
114 figure(10)
115 erro1b = readmatrix('Erro1B.txt');
116 erro1b(:,end) = [];
117 surf(x, t, erro1b)
118 title('Erro do item B')
119 xlabel('0 < x < 1')
120 ylabel('0 < t < 1')
121 zlabel('Temperatura')
122 colormap('spring')
123 shading interp
124
125 figure(11)
126 hold
127 distancia = floor(M/10);
128 Ratual = R1b(1, :);
129 plot(x, Ratual)
130 for v = 1:10
131     Ratual = R1b(1+v*distancia, :);
132     plot(x, Ratual)
133 end
134 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
135     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
136 title('Funcao u calculada do item B')
137 xlabel('0 < x < 1')
138 ylabel('Temperatura')
139 % 1C

```

```

140
141 figure(12)
142 R1c = readmatrix('Output1C.txt');
143 R1c(:,end) = [];
144 surf(x, t, R1c)
145 title('Funcao u calculada do item C')
146 xlabel('0 < x < 1')
147 ylabel('0 < t < 1')
148 zlabel('Temperatura')
149 colormap('summer')
150 shading interp
151
152 figure(13)
153 hold
154 distancia = floor(M/10);
155 Ratual = R1c(1, :);
156 plot(x, Ratual)
157 for v = 1:10
158     Ratual = R1c(1+v*distancia, :);
159     plot(x, Ratual)
160 end
161 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
162     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
163 title('Funcao u calculada do item C')
164 xlabel('0 < x < 1')
165 ylabel('Temperatura')

```

```

1 %% Tarefa 2
2
3 N_M = 10;
4 x = linspace(0, 1, N_M + 1);
5 t = linspace(0, 1, N_M + 1);
6
7 %% 2B
8 R2b = readmatrix('Output2B.txt', 'CommentStyle', {'='});
9 R2b(:, end) = [];
10
11 R2b1 = R2b(2:N_M+2, :);
12 figure(1)
13 surf(x, t, R2b1)
14 title('Funcao u do item 1A com metodo de Euler implicito')
15 xlabel('0 < x < 1')
16 ylabel('0 < t < 1')
17 zlabel('Temperatura')
18 shading interp
19
20 R2b2 = R2b(N_M+4:2*N_M+4, :);
21 figure(2)
22 surf(x, t, R2b2)
23 title('Funcao u do item 1B com metodo de Euler implicito')
24 xlabel('0 < x < 1')
25 ylabel('0 < t < 1')
26 zlabel('Temperatura')
27 colormap('summer')
28 shading interp
29
30 R2b3 = R2b(2*N_M+6:3*N_M+6, :);
31 figure(3)
32 surf(x, t, R2b3)
33 title('Funcao u do item 1C com metodo de Euler implicito')
34 xlabel('0 < x < 1')
35 ylabel('0 < t < 1')

```

```

36 zlabel('Temperatura')
37 colormap('spring')
38 shading interp
39
40 % 2B - Graficos 2D
41
42 figure(4)
43 hold
44 distancia = floor(N_M/10);
45 Ratual = R2b1(1, :);
46 plot(x, Ratual)
47 for v = 1:10
48     Ratual = R2b1(1+v*distancia, :);
49     plot(x, Ratual)
50 end
51 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
52     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
53 title('Funcao u do item 1A com metodo de Euler implicito')
54 xlabel('0 < x < 1')
55 ylabel('Temperatura')
56
57 figure(5)
58 hold
59 distancia = floor(N_M/10);
60 Ratual = R2b2(1, :);
61 plot(x, Ratual)
62 for v = 1:10
63     Ratual = R2b2(1+v*distancia, :);
64     plot(x, Ratual)
65 end
66 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
67     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
68 title('Funcao u do item 1B com metodo de Euler implicito')
69 xlabel('0 < x < 1')
70 ylabel('Temperatura')

```

```

70 figure(6)
71 hold
72 distancia = floor(N_M/10);
73 Ratual = R2b3(1, :);
74 plot(x, Ratual)
75 for v = 1:10
76     Ratual = R2b3(1+v*distancia, :);
77     plot(x, Ratual)
78 end
79 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
    0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
80 title('Funcao u do item 1C com metodo de Euler implicito')
81 xlabel('0 < x < 1')
82 ylabel('Temperatura')
83
84 % Erros 2B
85
86 R2bErro = readmatrix('Erro2B.txt', 'CommentStyle', {'='});
87 R2bErro(:,end) = [];
88
89 R2bErro1 = R2bErro(2:N_M+2,:);
90 figure(7)
91 surf(x, t, R2bErro1)
92 title('Erro do item 1A com metodo de Euler implicito')
93 xlabel('0 < x < 1')
94 ylabel('0 < t < 1')
95 zlabel('Temperatura')
96 shading interp
97
98 R2bErro2 = R2bErro(N_M+4:2*N_M+4,:);
99 figure(8)
100 surf(x, t, R2bErro2)
101 title('Erro do item 1B com metodo de Euler implicito')
102 xlabel('0 < x < 1')
103 ylabel('0 < t < 1')
104 zlabel('Temperatura')

```

```

105 colormap('summer')
106 shading interp
107
108 %% 2C
109 R2c = readmatrix('Output2C.txt', 'CommentStyle', {'='});%
110 R2c(:,end) = [];
111
112 R2c1 = R2c(2:N_M+2,:);
113 figure(9)
114 surf(x, t, R2c1)
115 title('Funcao u do item 1A com metodo de Crank Nicolson')
116 xlabel('0 < x < 1')
117 ylabel('0 < t < 1')
118 zlabel('Temperatura')
119 shading interp
120
121 R2c2 = R2c(N_M+4:2*N_M+4,:);
122 figure(10)
123 surf(x, t, R2c2)
124 title('Funcao u do item 1B com metodo de Crank Nicolson')
125 xlabel('0 < x < 1')
126 ylabel('0 < t < 1')
127 zlabel('Temperatura')
128 colormap('summer')
129 shading interp
130
131 R2c3 = R2c(2*N_M+6:3*N_M+6,:);
132 figure(11)
133 surf(x, t, R2c3)
134 title('Funcao u do item 1C com metodo de Crank Nicolson')
135 xlabel('0 < x < 1')
136 ylabel('0 < t < 1')
137 zlabel('Temperatura')
138 colormap('spring')
139 shading interp
140

```

```

141 % 2C - Graficos 2D
142
143 figure(12)
144 hold
145 distancia = floor(N_M/10);
146 Ratual = R2c1(1, :);
147 plot(x, Ratual)
148 for v = 1:10
149     Ratual = R2c1(1+v*distancia, :);
150     plot(x, Ratual)
151 end
152 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
153     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
154 title('Funcao u do item 1A com metodo de Crank Nicolson')
155 xlabel('0 < x < 1')
156 ylabel('Temperatura')

157 figure(13)
158 hold
159 distancia = floor(N_M/10);
160 Ratual = R2c2(1, :);
161 plot(x, Ratual)
162 for v = 1:10
163     Ratual = R2c2(1+v*distancia, :);
164     plot(x, Ratual)
165 end
166 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
167     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
168 title('Funcao u do item 1B com metodo de Crank Nicolson')
169 xlabel('0 < x < 1')
170 ylabel('Temperatura')

171 figure(14)
172 hold
173 distancia = floor(N_M/10);
174 Ratual = R2c3(1, :);

```

```

175 plot(x, Ratual)
176 for v = 1:10
177     Ratual = R2c3(1+v*distancia, :);
178     plot(x, Ratual)
179 end
180 legend('t = 0', 't = 0.1', 't = 0.2', 't = 0.3', 't = 0.4', 't = 0.5', 't = ...
181     0.6', 't = 0.7', 't = 0.8', 't = 0.9', 't = 1')
182 title('Funcao u do item 1C com metodo de Crank Nicolson')
183 xlabel('0 < x < 1')
184 ylabel('Temperatura')
185
186
187 % Erros 2C
188
189
190 R2cErro = readmatrix('Erro2C.txt', 'CommentStyle', {'='});
191 R2cErro(:,end) = [];
192
193 R2cErro1 = R2cErro(2:N_M+2,:);
194 figure(15)
195 surf(x, t, R2cErro1)
196 title('Erro do item 1A com metodo de Crank Nicolson')
197 xlabel('0 < x < 1')
198 ylabel('0 < t < 1')
199 zlabel('Temperatura')
200 shading interp
201
202 R2cErro2 = R2cErro(N_M+4:2*N_M+4,:);
203 figure(16)
204 surf(x, t, R2cErro2)
205 title('Erro do item 1B com metodo de Crank Nicolson')
206 xlabel('0 < x < 1')
207 ylabel('0 < t < 1')
208 zlabel('Temperatura')
209 colormap('summer')
210 shading interp

```

## C Curvas do item 1A

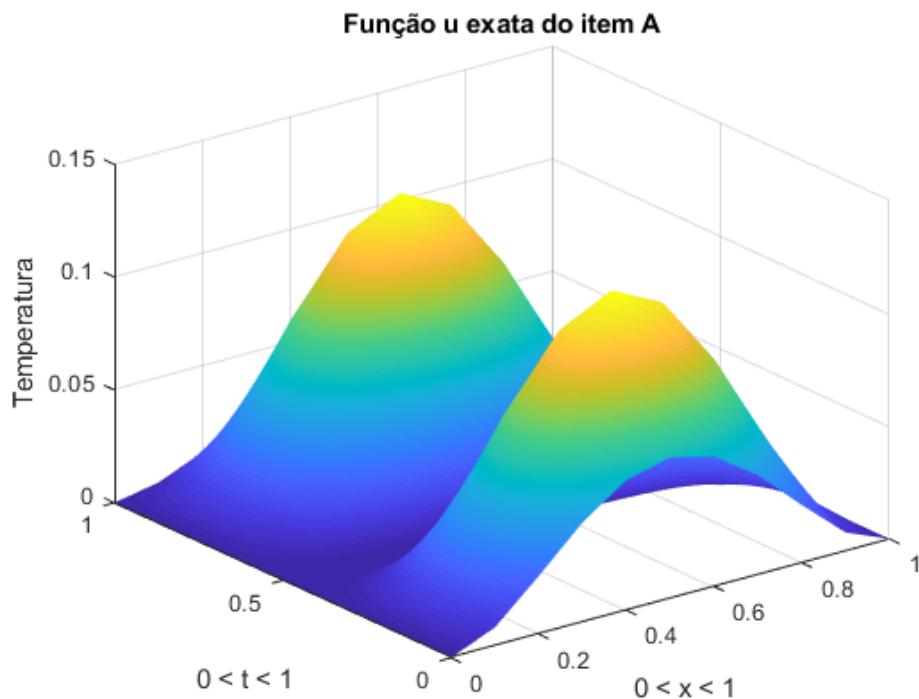


Figura 18: Curva da temperatura exata com  $N = 10$  e  $\lambda = 0.25$

**Função u calculada do item A**

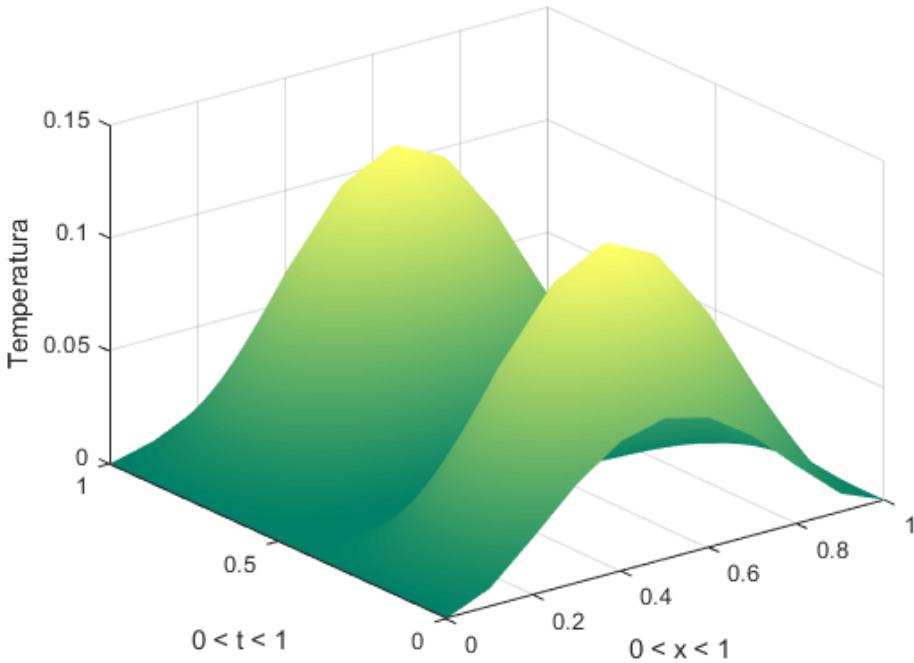


Figura 19: Curva da temperatura calculada com  $N = 10$  e  $\lambda = 0.5$

**Erro do item A**

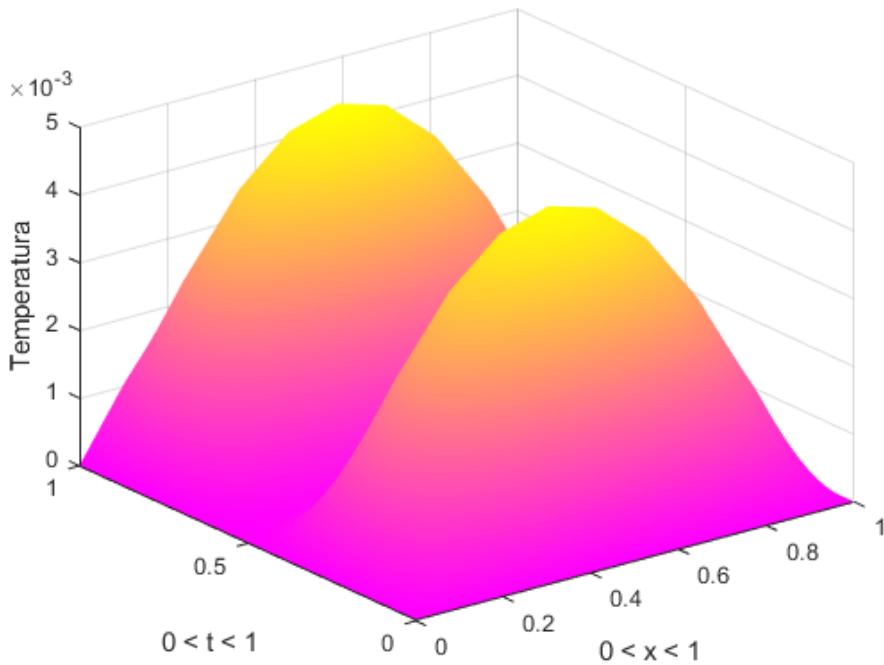


Figura 20: Erro com  $N = 10$  e  $\lambda = 0.5$

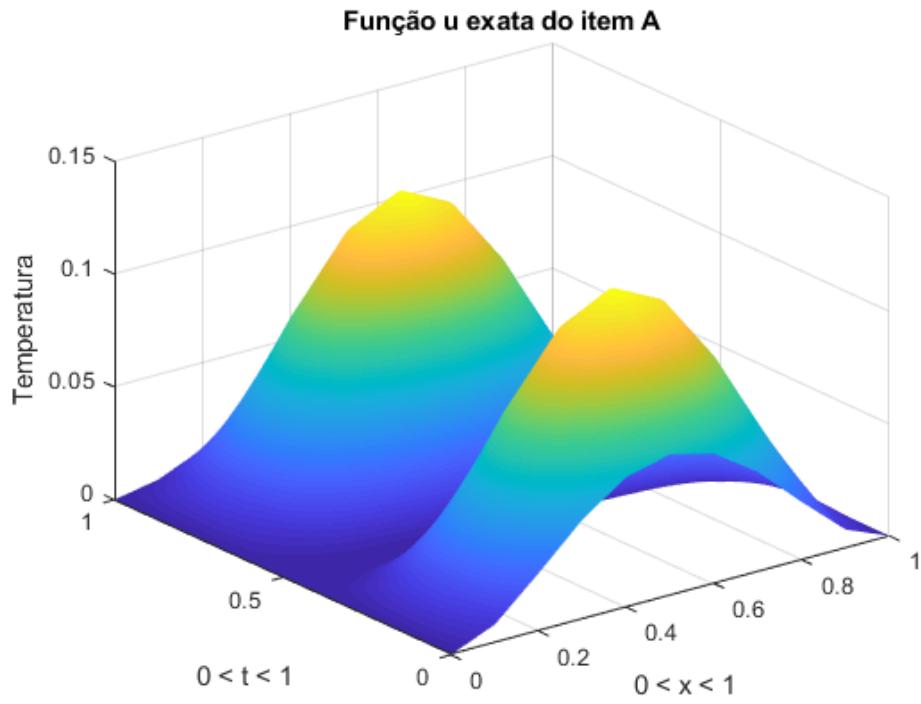


Figura 21: Curva da temperatura exata com  $N = 10$  e  $\lambda = 0.5$

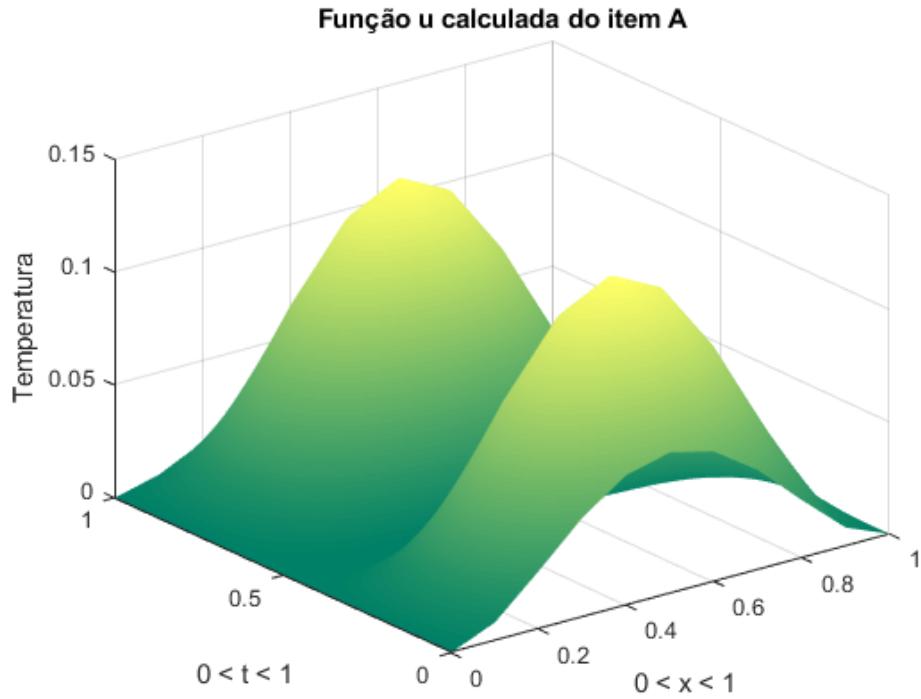


Figura 22: Curva da temperatura calculada com  $N = 10$  e  $\lambda = 0.5$

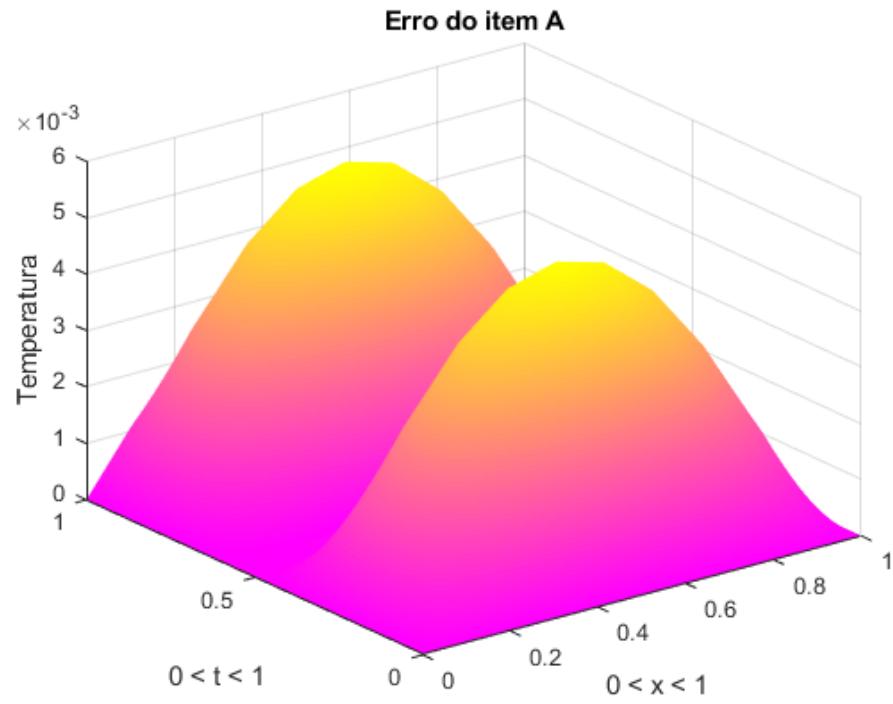


Figura 23: Erro com  $N = 10$  e  $\lambda = 0.5$

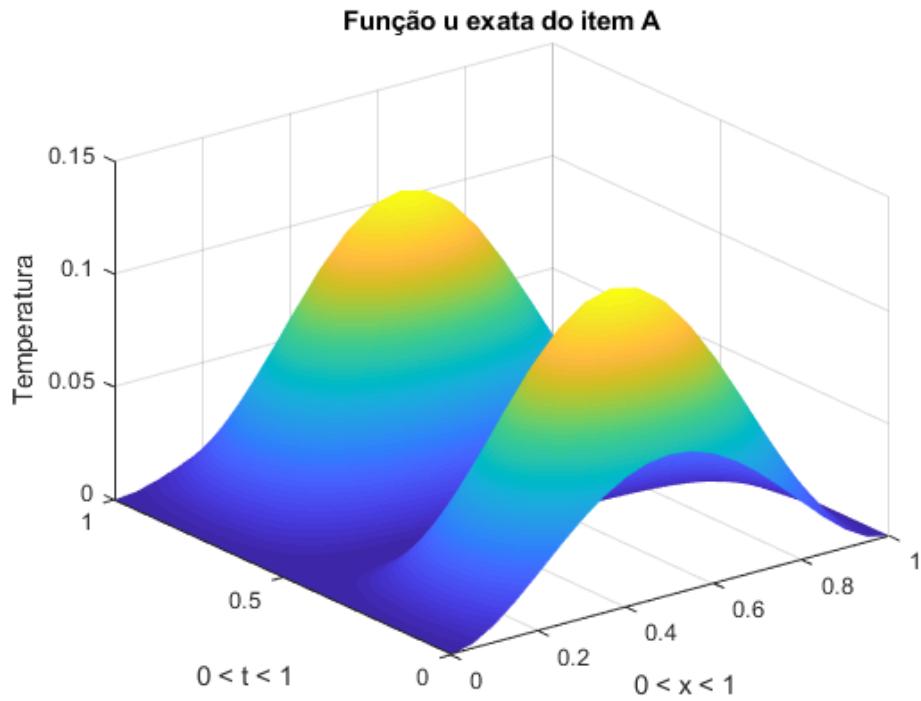


Figura 24: Curva da temperatura exata com  $N = 20$  e  $\lambda = 0.25$

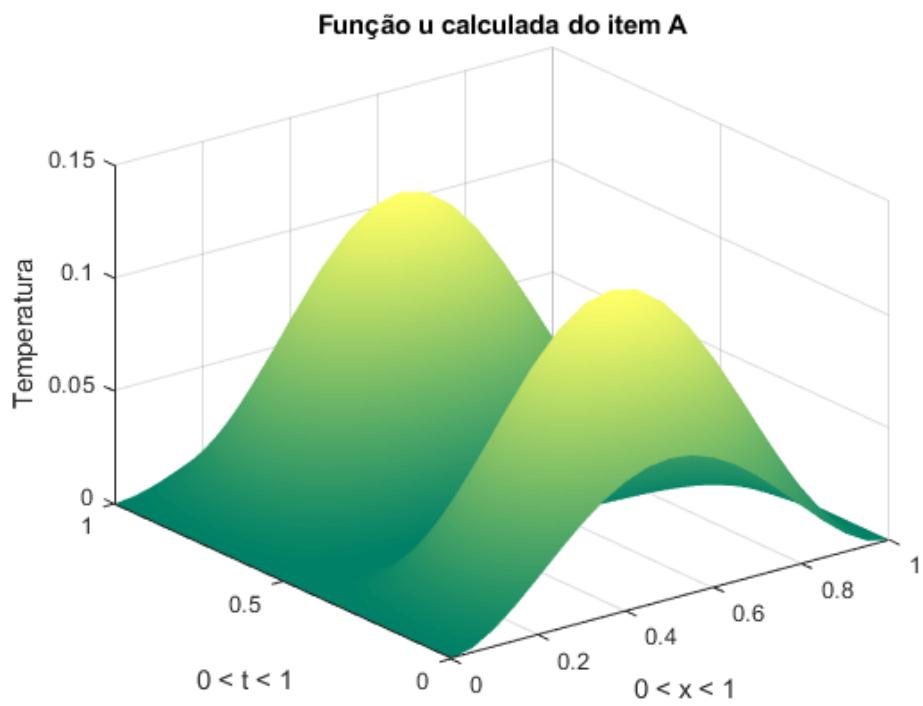


Figura 25: Curva da temperatura calculada com  $N = 20$  e  $\lambda = 0.25$

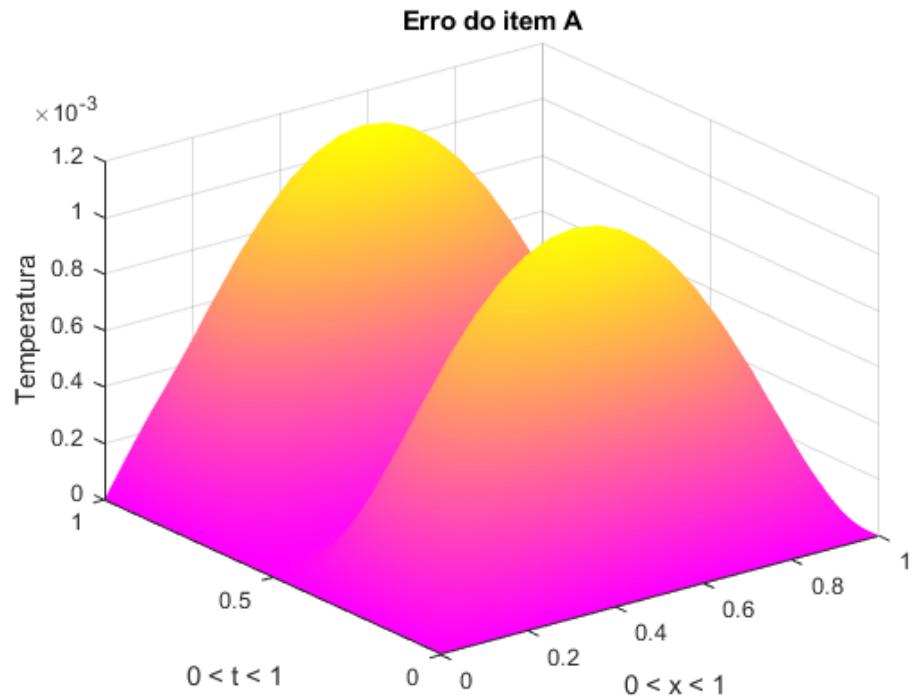


Figura 26: Erro com  $N = 20$  e  $\lambda = 0.25$

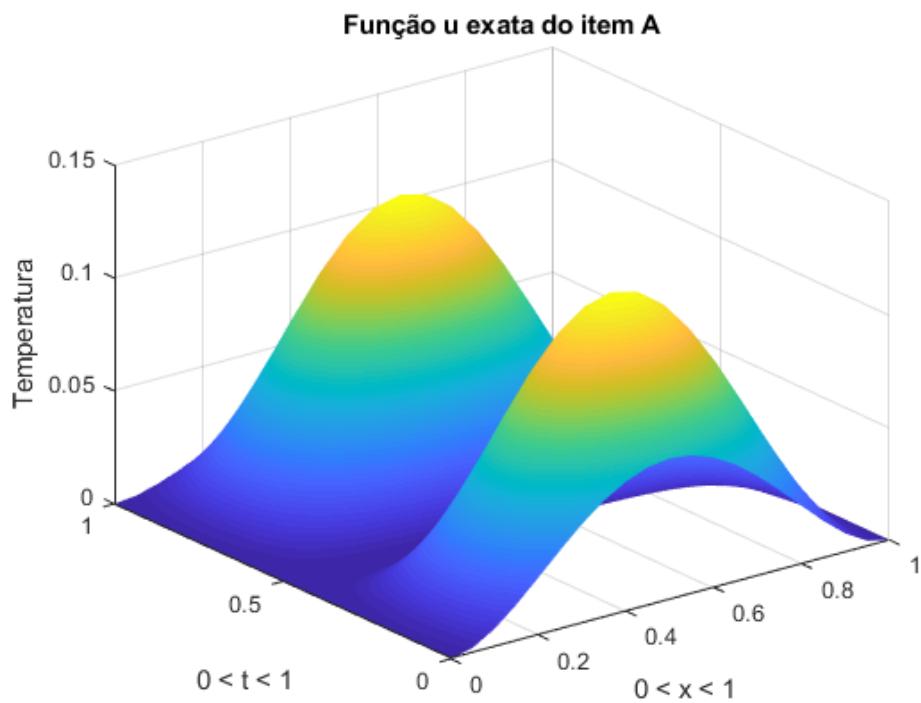


Figura 27: Curva da temperatura exata com  $N = 20$  e  $\lambda = 0.5$

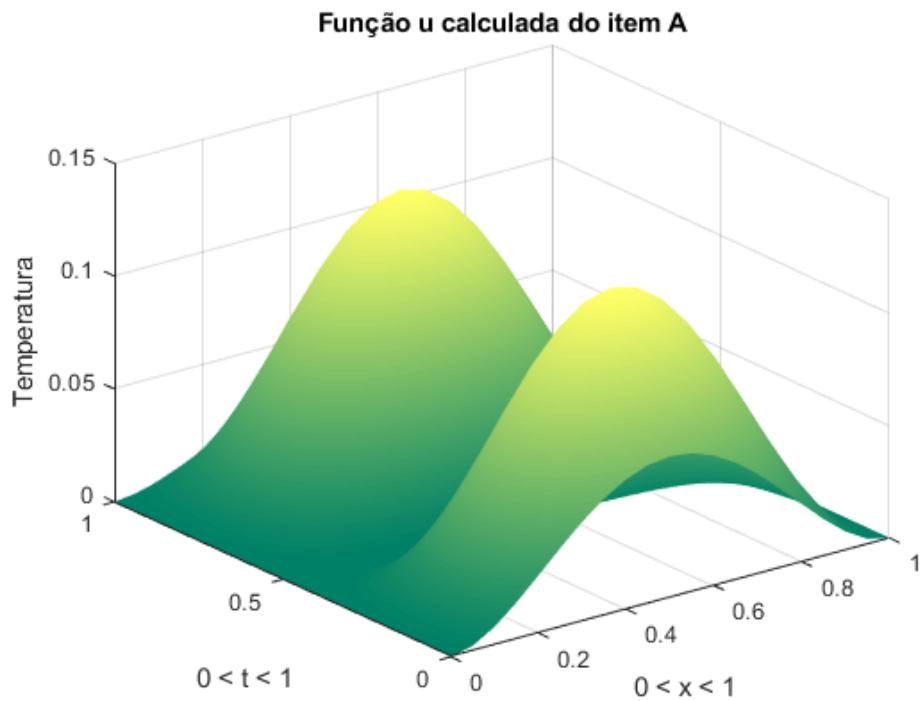


Figura 28: Curva da temperatura calculada com  $N = 20$  e  $\lambda = 0.5$

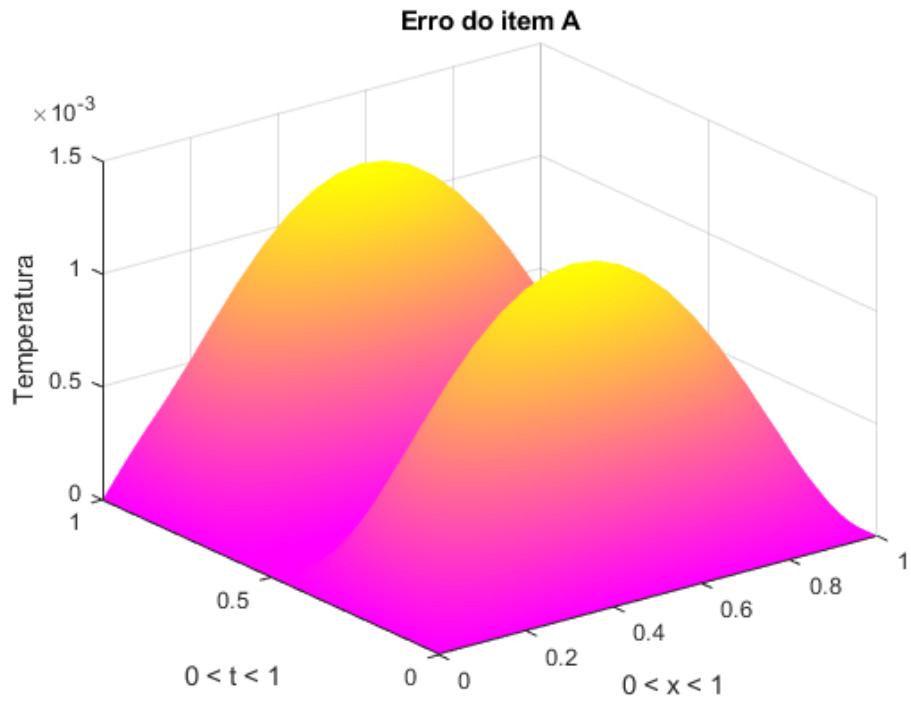


Figura 29: Erro com  $N = 20$  e  $\lambda = 0.5$

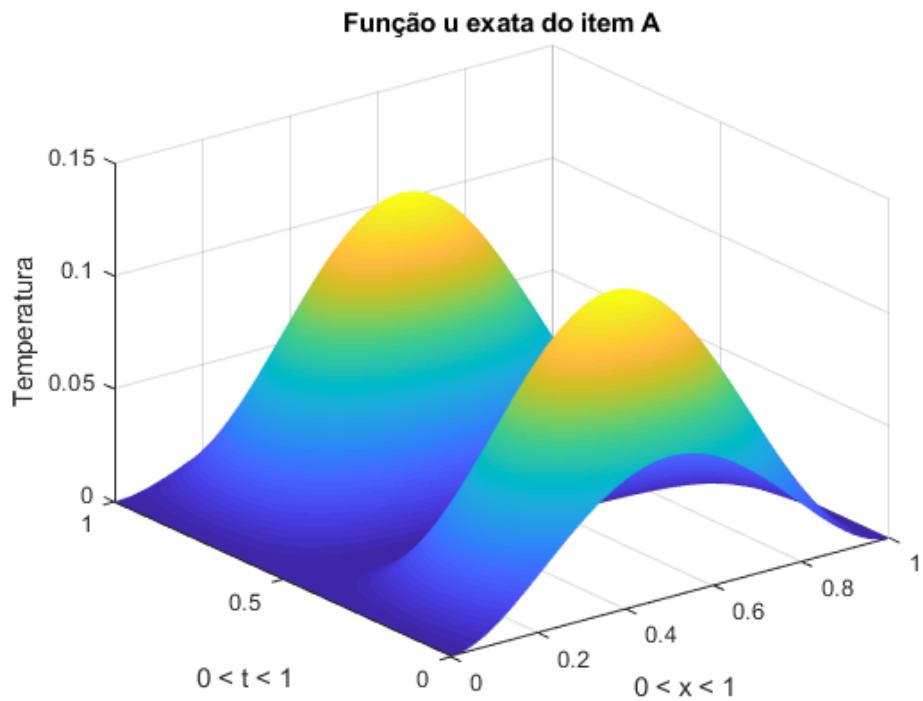


Figura 30: Curva da temperatura exata com  $N = 40$  e  $\lambda = 0.25$

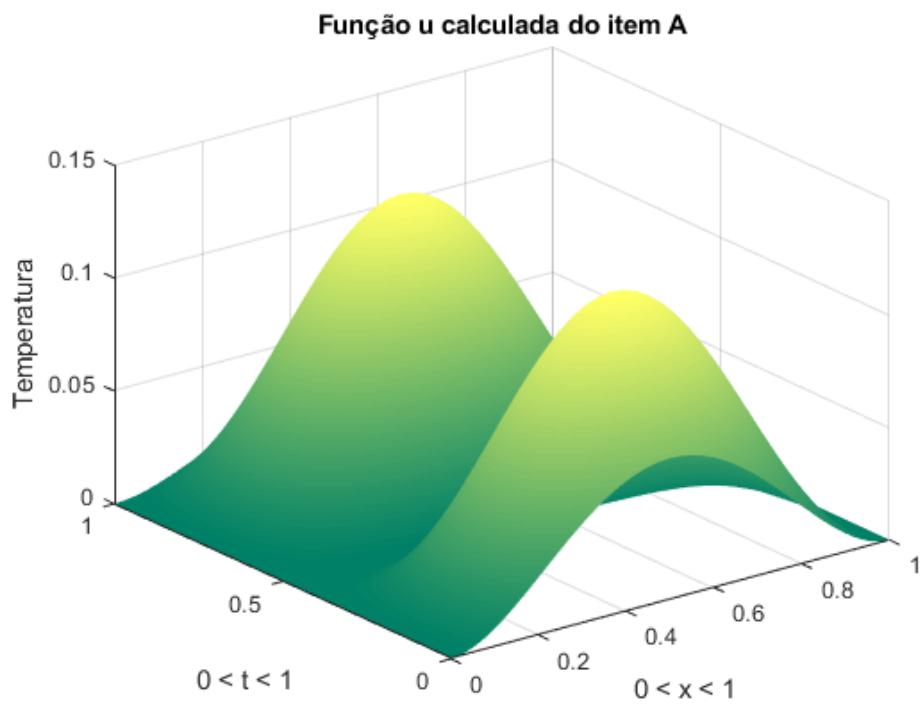


Figura 31: Curva da temperatura calculada com  $N = 40$  e  $\lambda = 0.25$

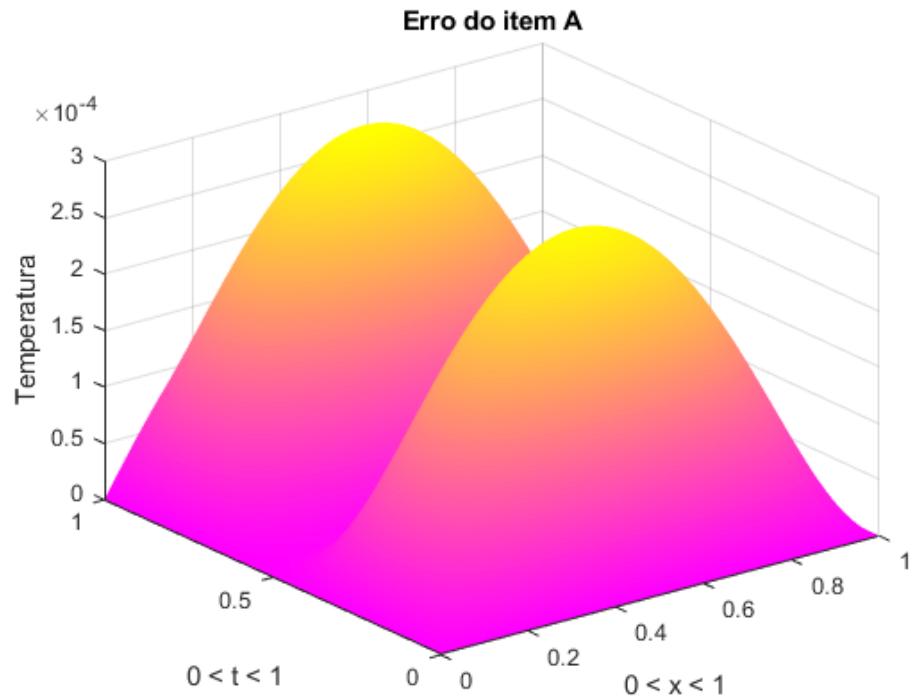


Figura 32: Erro com  $N = 40$  e  $\lambda = 0.25$

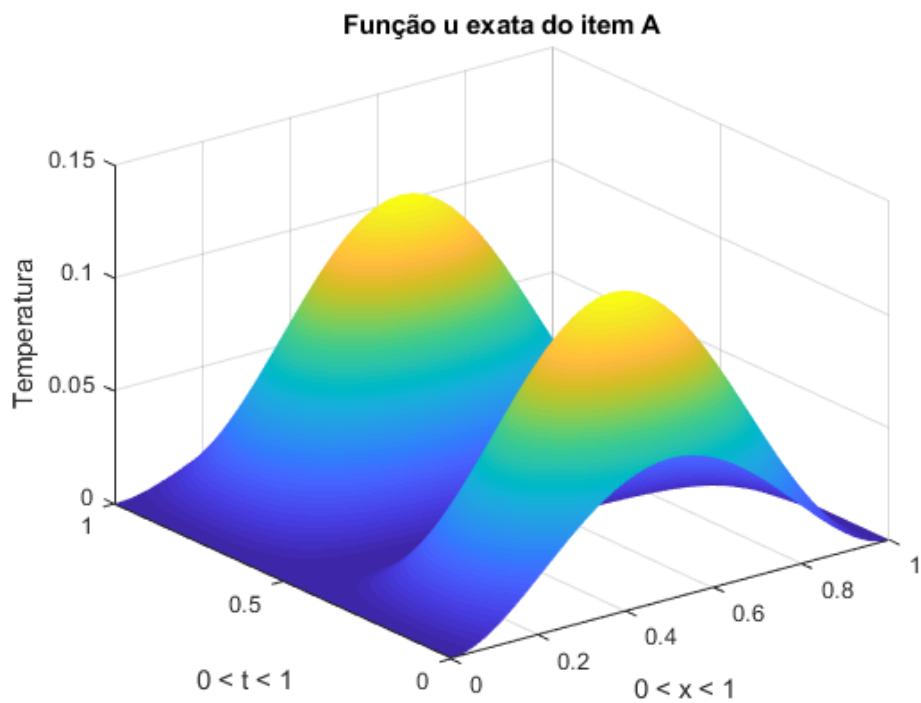


Figura 33: Curva da temperatura exata com  $N = 40$  e  $\lambda = 0.5$

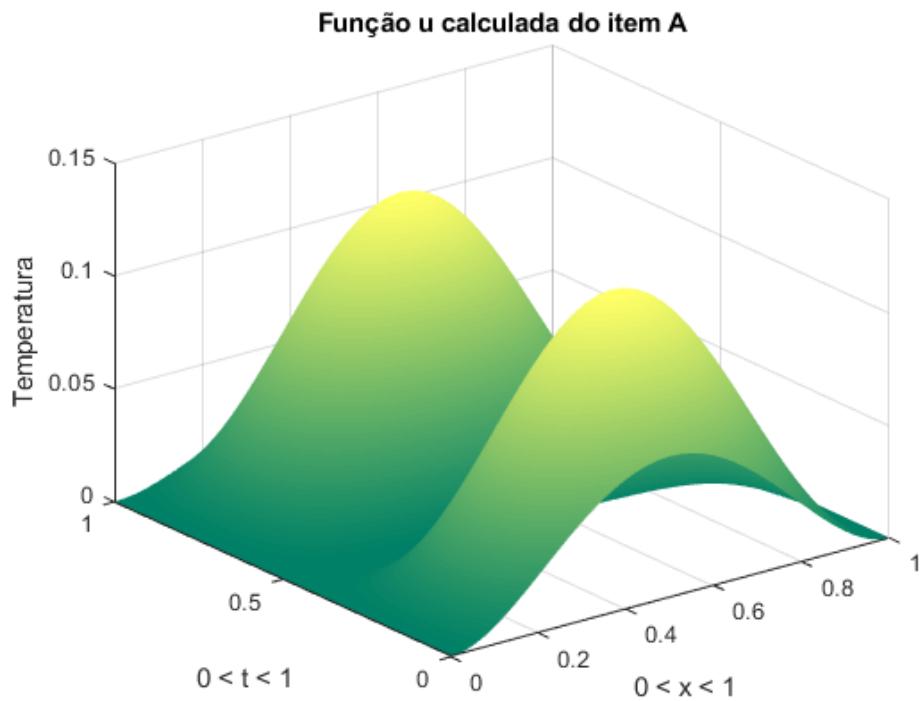


Figura 34: Curva da temperatura calculada com  $N = 40$  e  $\lambda = 0.5$

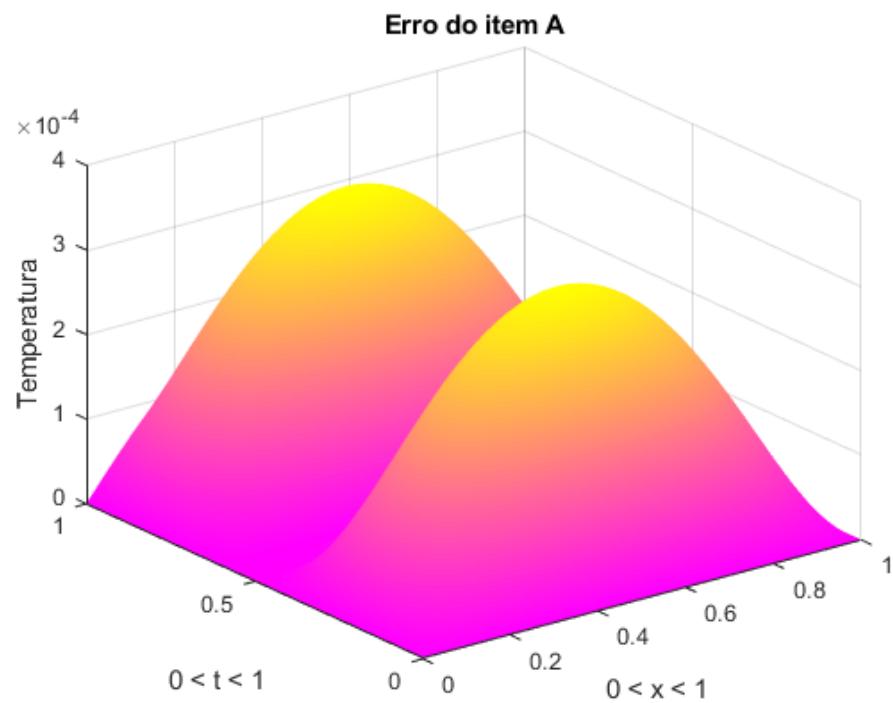


Figura 35: Erro com  $N = 40$  e  $\lambda = 0.5$

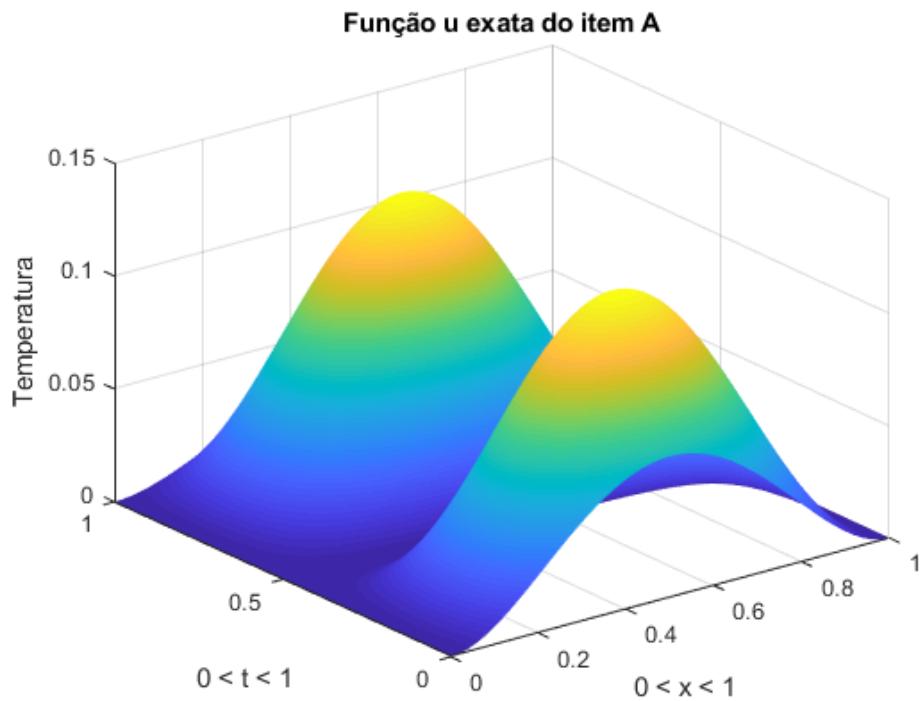


Figura 36: Curva da temperatura exata com  $N = 80$  e  $\lambda = 0.25$

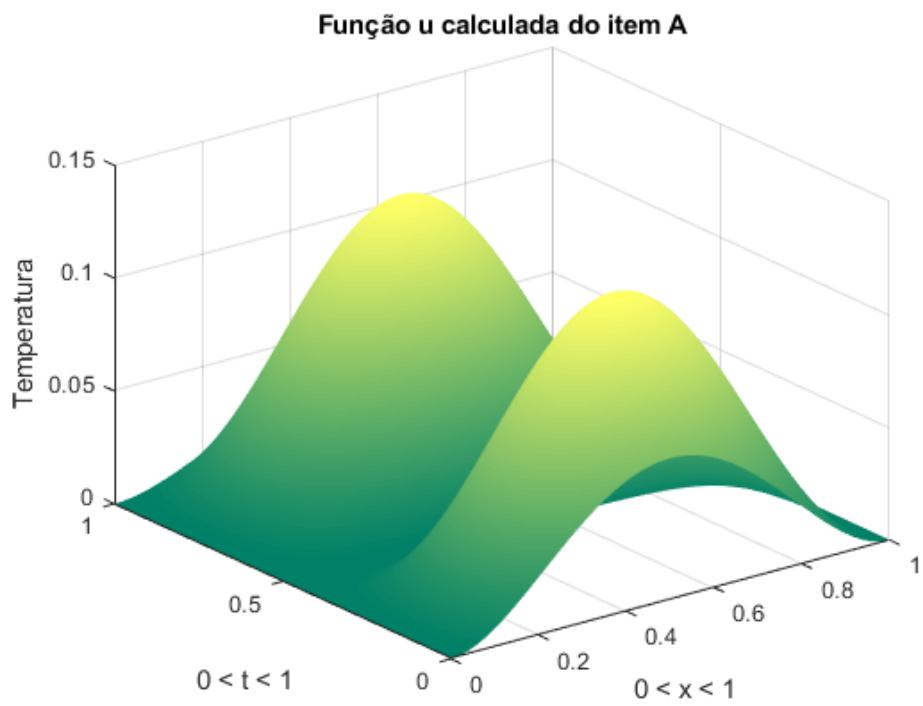


Figura 37: Curva da temperatura calculada com  $N = 80$  e  $\lambda = 0.25$

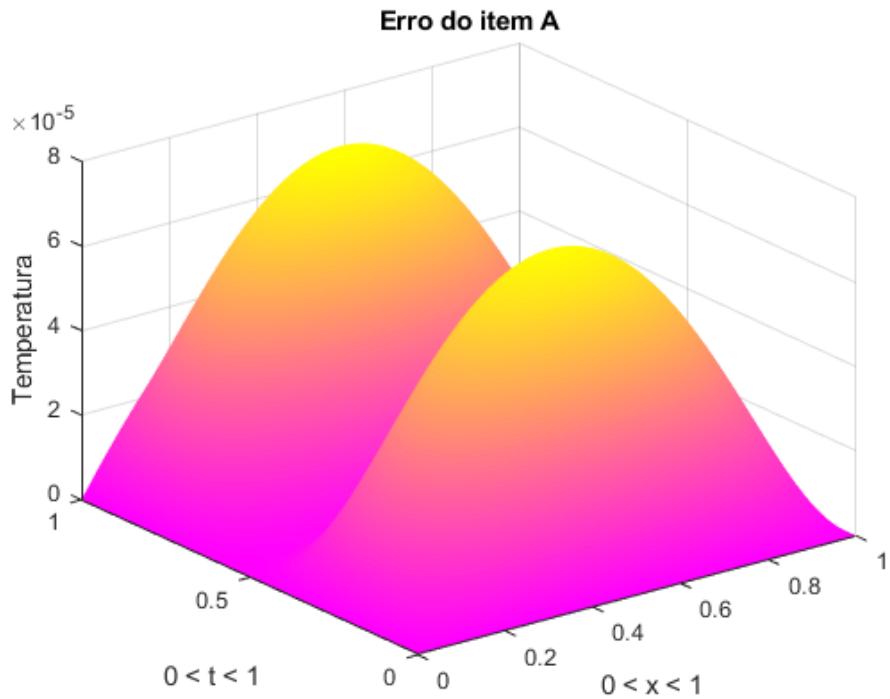


Figura 38: Erro com  $N = 80$  e  $\lambda = 0.25$

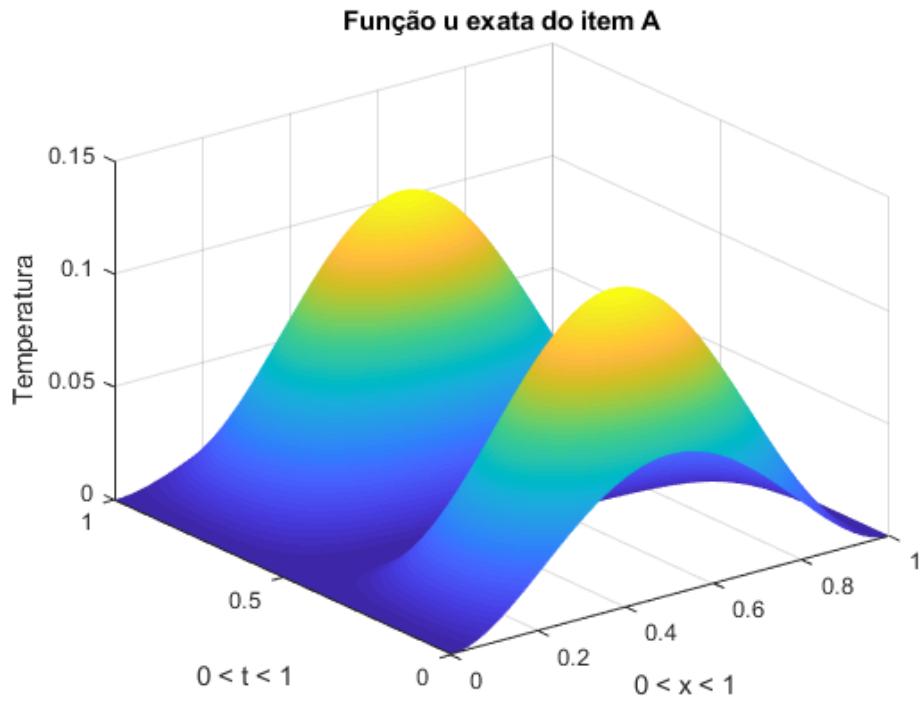


Figura 39: Curva da temperatura exata com  $N = 80$  e  $\lambda = 0.5$

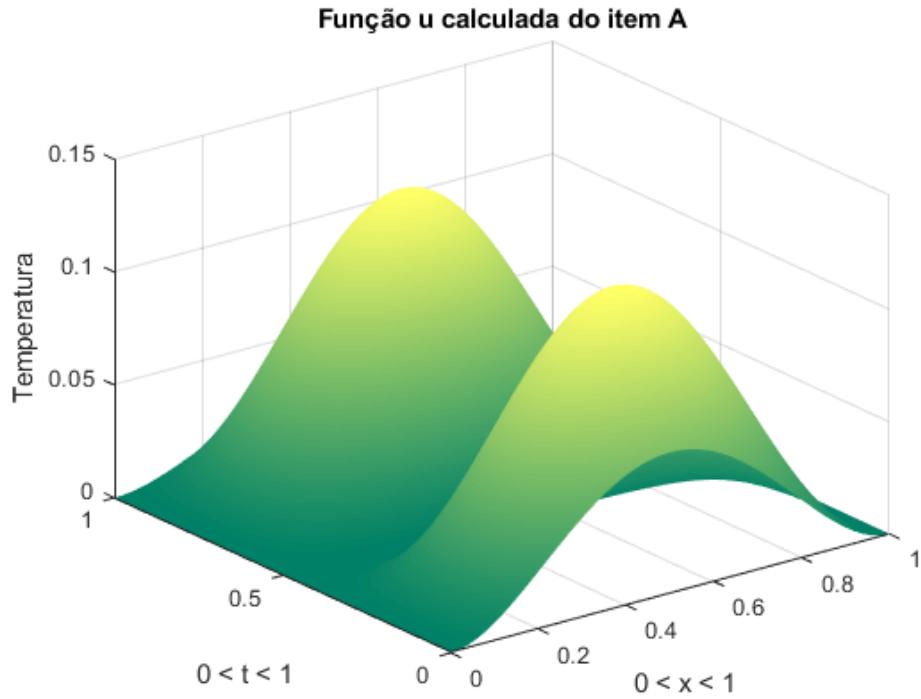


Figura 40: Curva da temperatura calculada com  $N = 80$  e  $\lambda = 0.5$

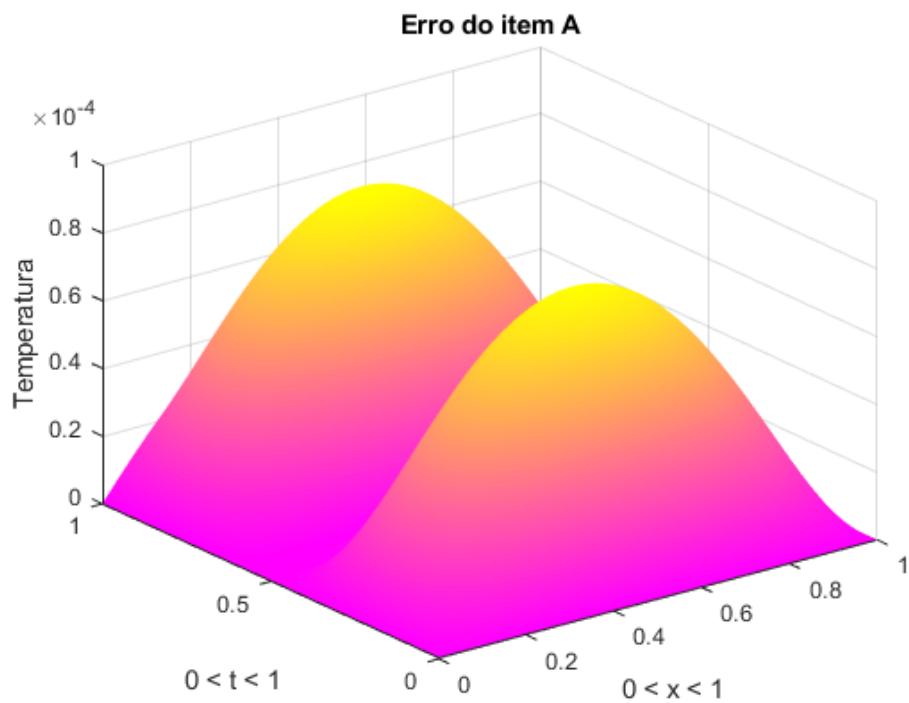


Figura 41: Erro com  $N = 80$  e  $\lambda = 0.5$

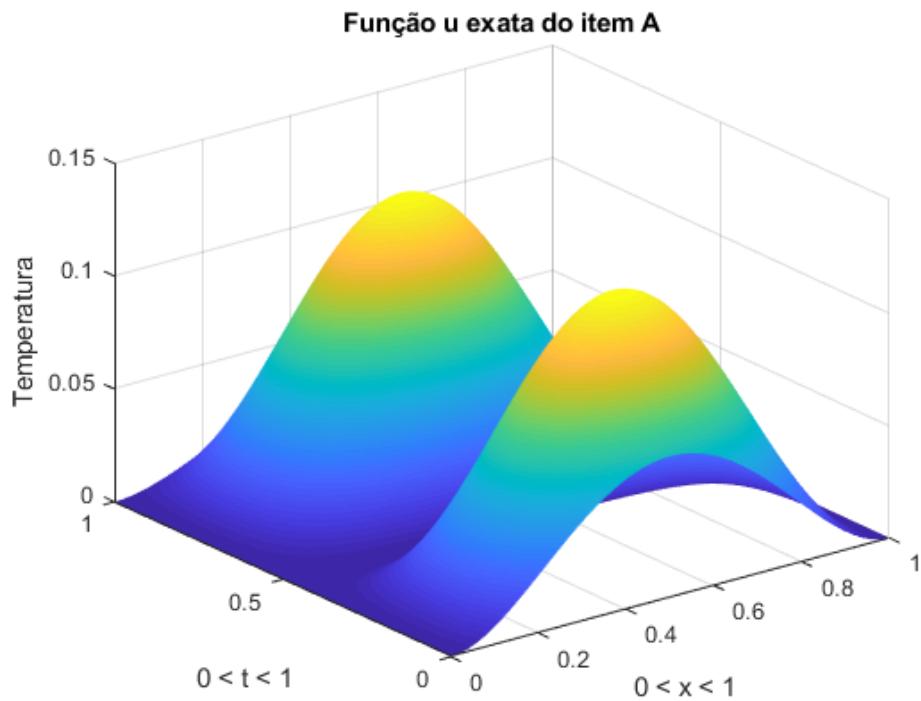


Figura 42: Curva da temperatura exata com  $N = 160$  e  $\lambda = 0.25$

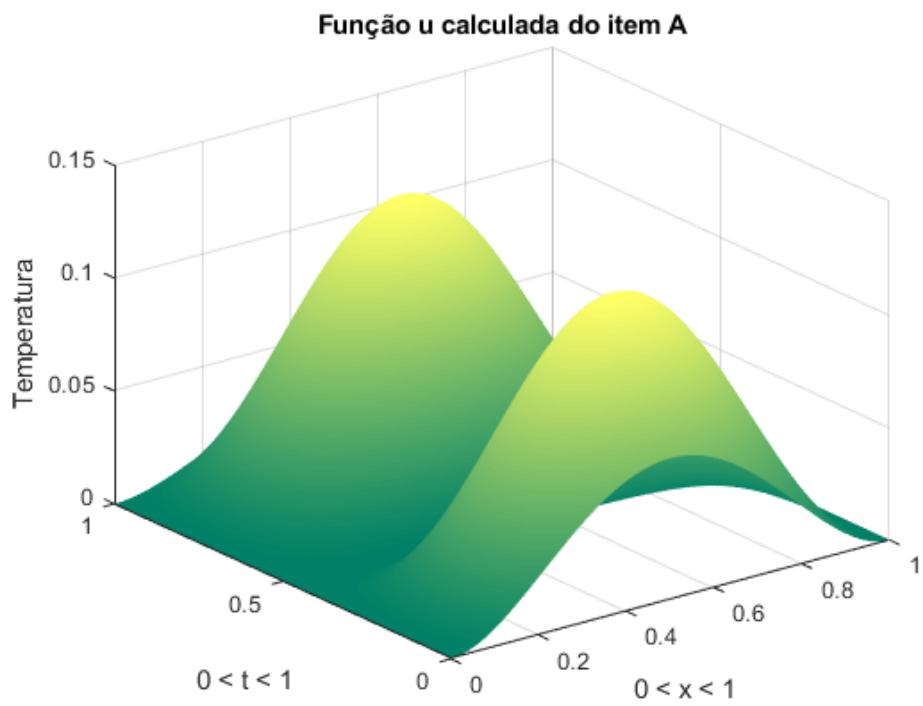


Figura 43: Curva da temperatura calculada com  $N = 160$  e  $\lambda = 0.25$

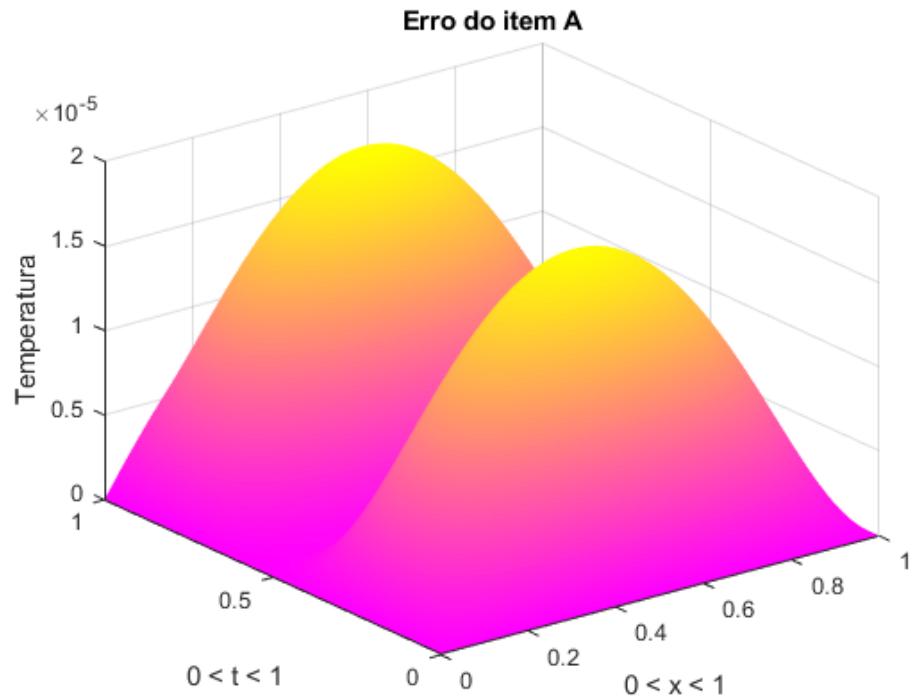


Figura 44: Erro com  $N = 160$  e  $\lambda = 0.25$

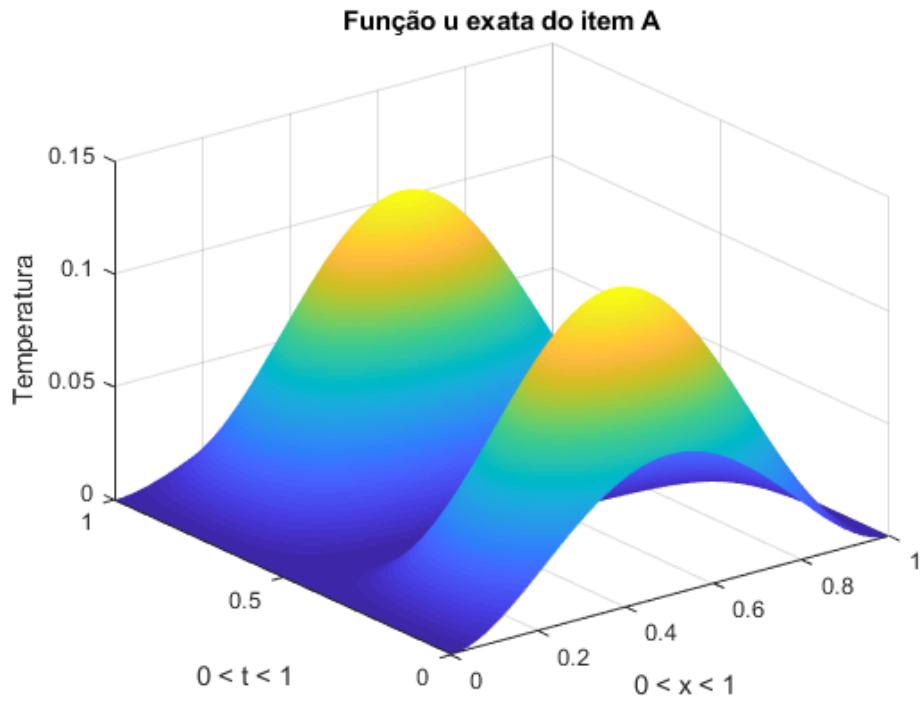


Figura 45: Curva da temperatura exata com  $N = 160$  e  $\lambda = 0.5$

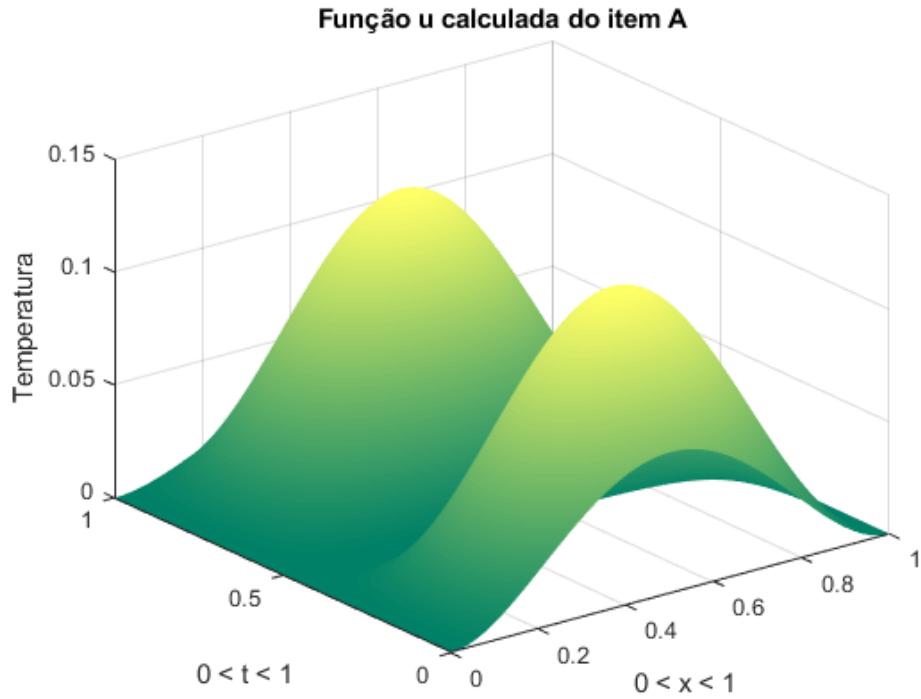


Figura 46: Curva da temperatura calculada com  $N = 160$  e  $\lambda = 0.5$

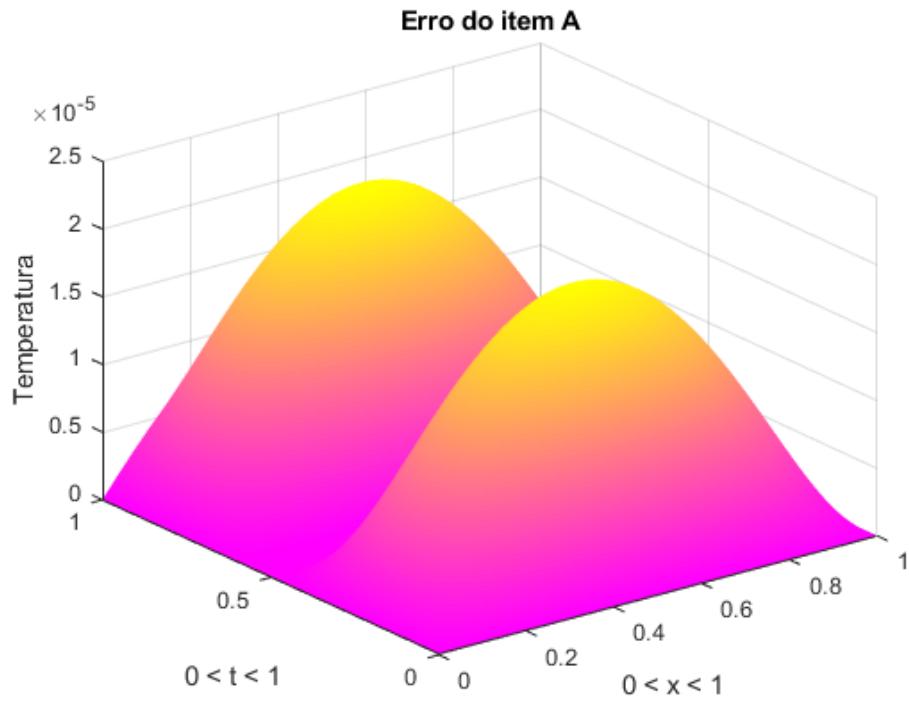


Figura 47: Erro com  $N = 160$  e  $\lambda = 0.5$

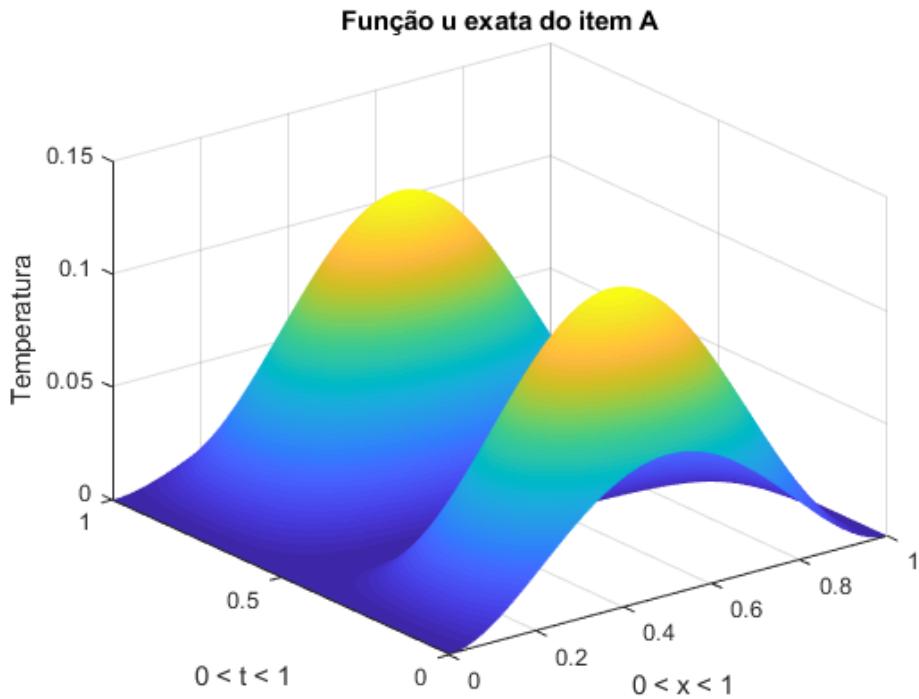


Figura 48: Curva da temperatura exata com  $N = 320$  e  $\lambda = 0.25$

**Função u calculada do item A**

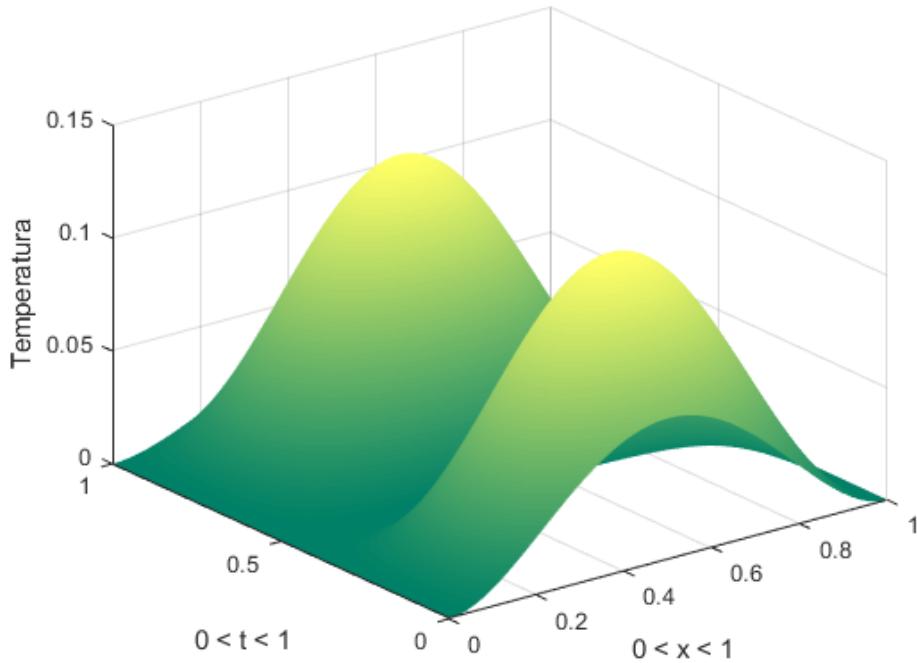


Figura 49: Curva da temperatura calculada com  $N = 320$  e  $\lambda = 0.25$

**Erro do item A**

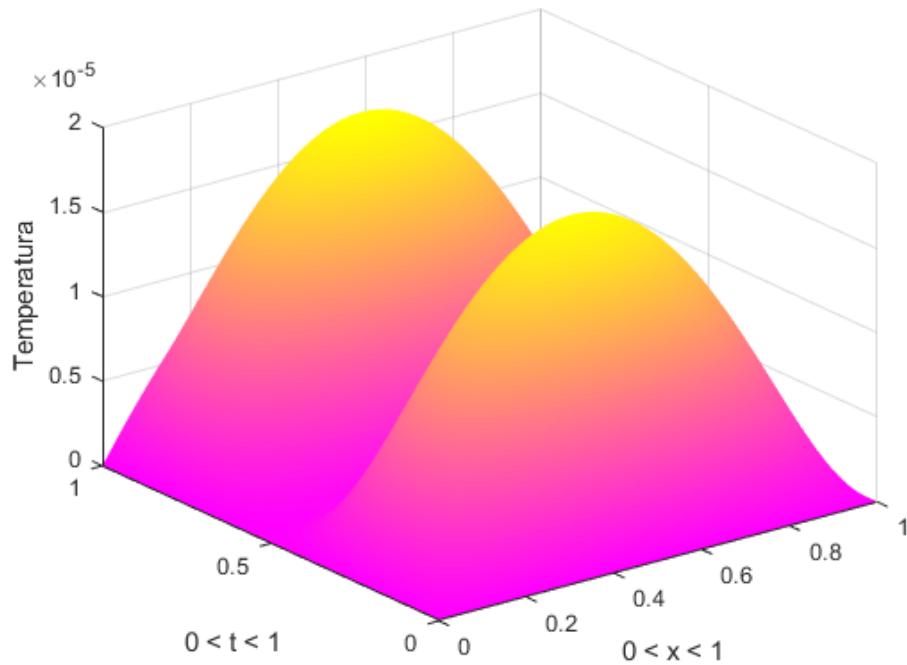


Figura 50: Erro com  $N = 320$  e  $\lambda = 0.25$

**Função u exata do item A**

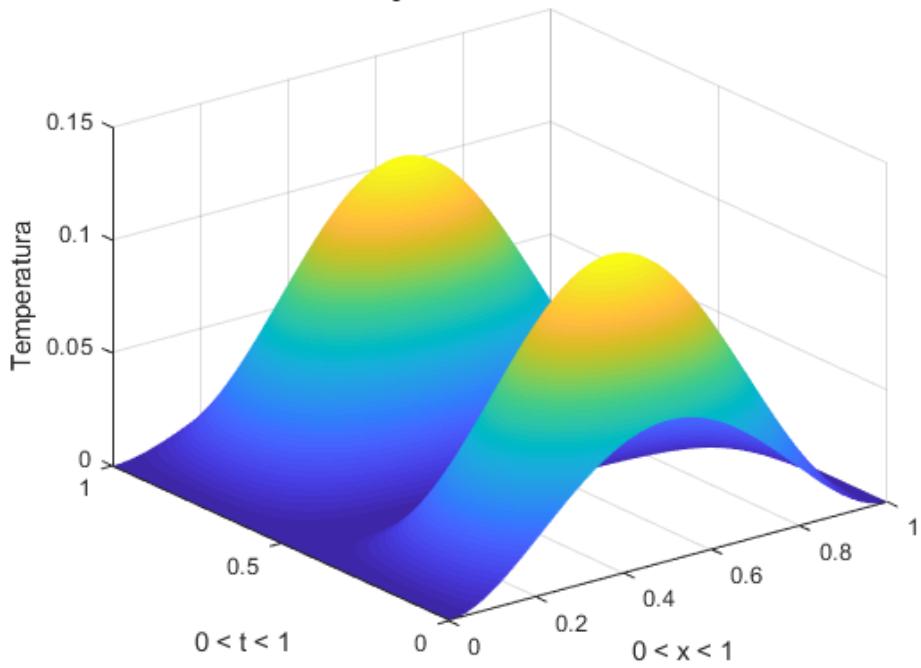


Figura 51: Curva da temperatura exata com  $N = 320$  e  $\lambda = 0.5$

**Função u calculada do item A**

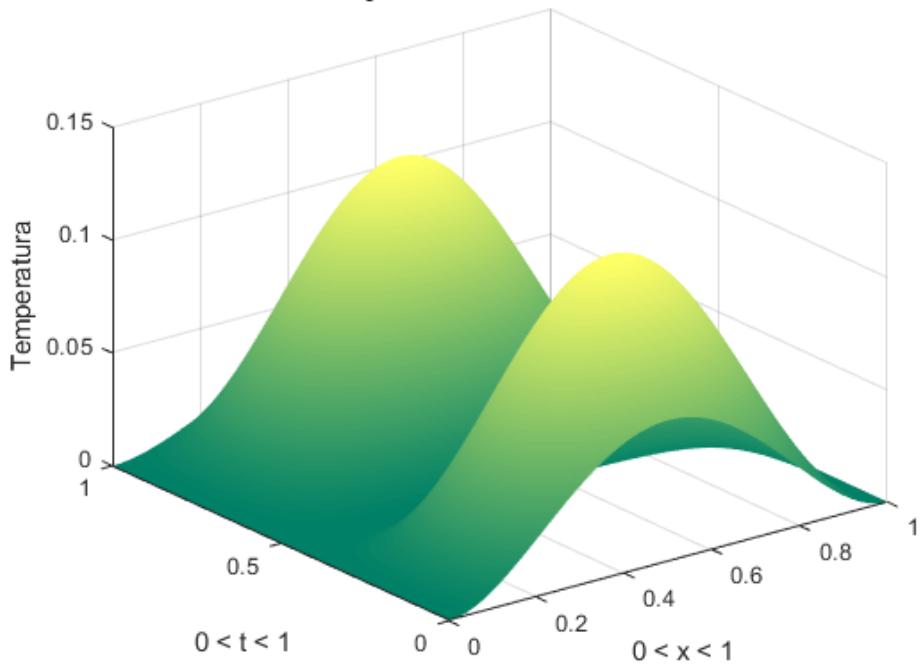


Figura 52: Curva da temperatura calculada com  $N = 320$  e  $\lambda = 0.5$

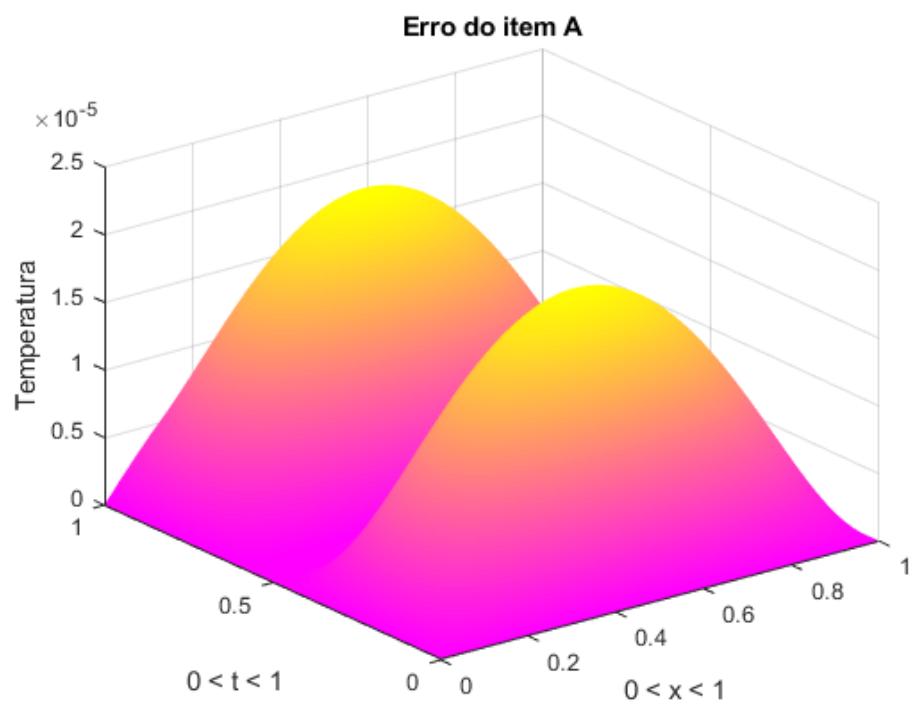


Figura 53: Erro com  $N = 320$  e  $\lambda = 0.5$

## D Curvas do item 1B

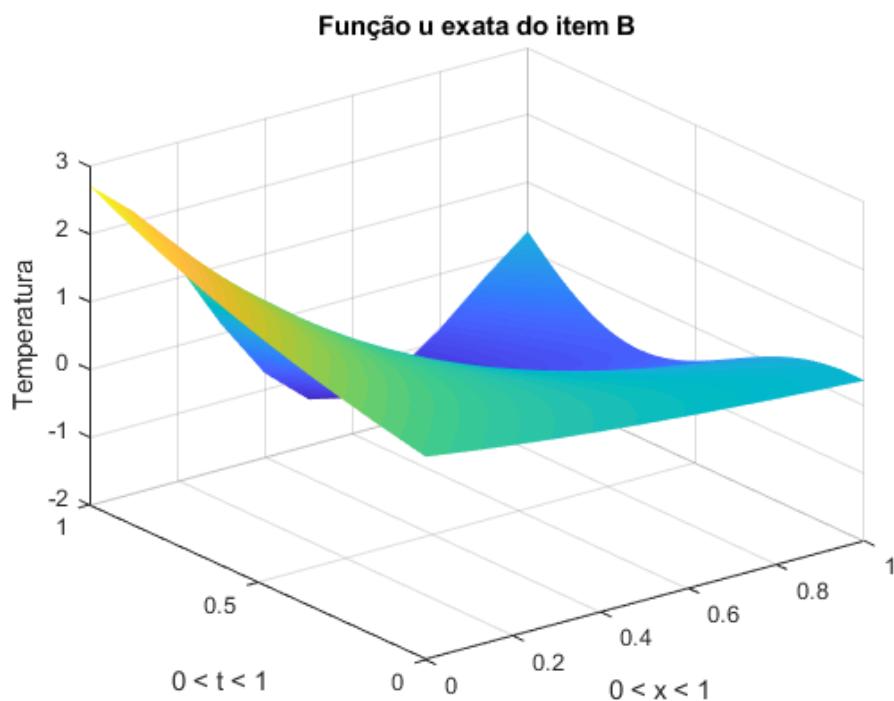


Figura 54: Curva da temperatura exata com  $N = 10$  e  $\lambda = 0.25$

**Função u calculada do item B**

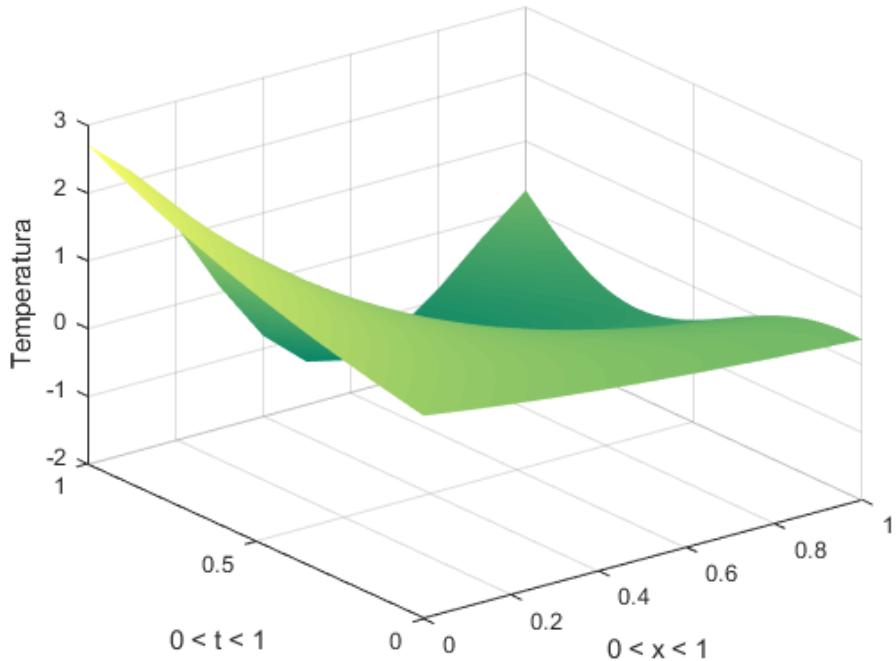


Figura 55: Curva da temperatura calculada com  $N = 10$  e  $\lambda = 0.5$

**Erro do item B**

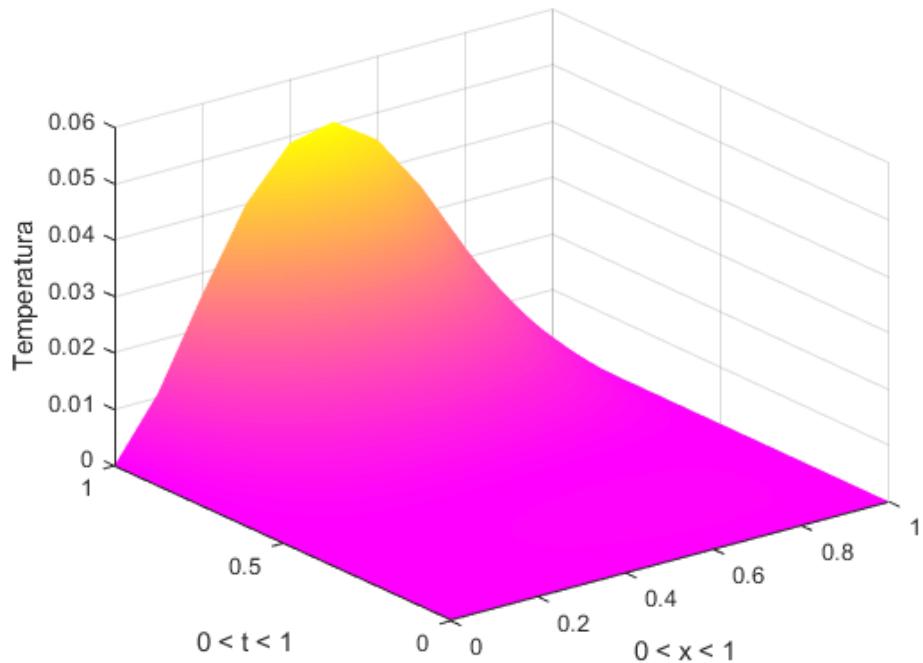


Figura 56: Erro com  $N = 10$  e  $\lambda = 0.5$

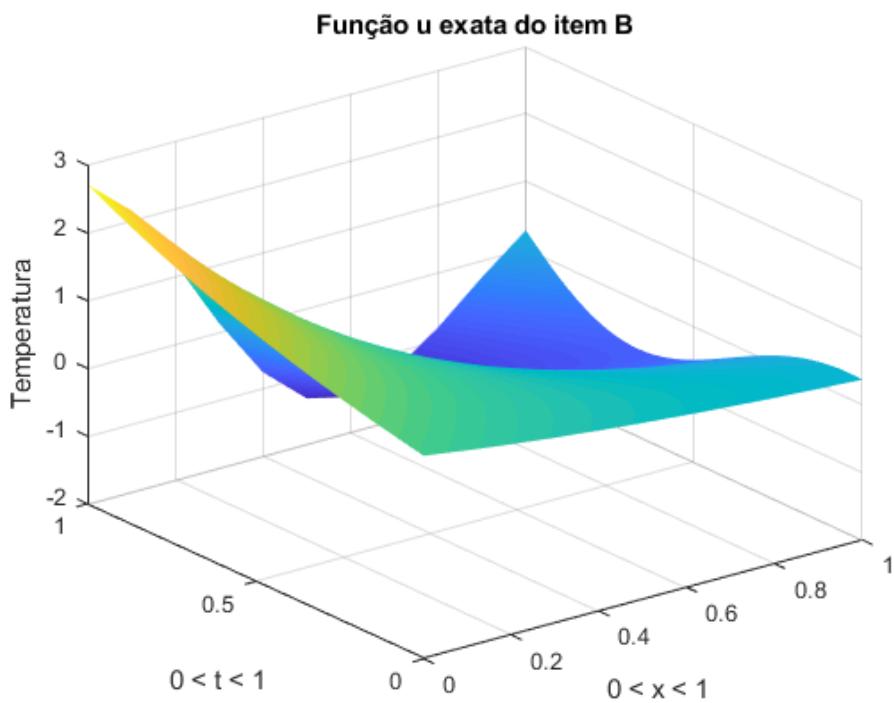


Figura 57: Curva da temperatura exata com  $N = 10$  e  $\lambda = 0.5$

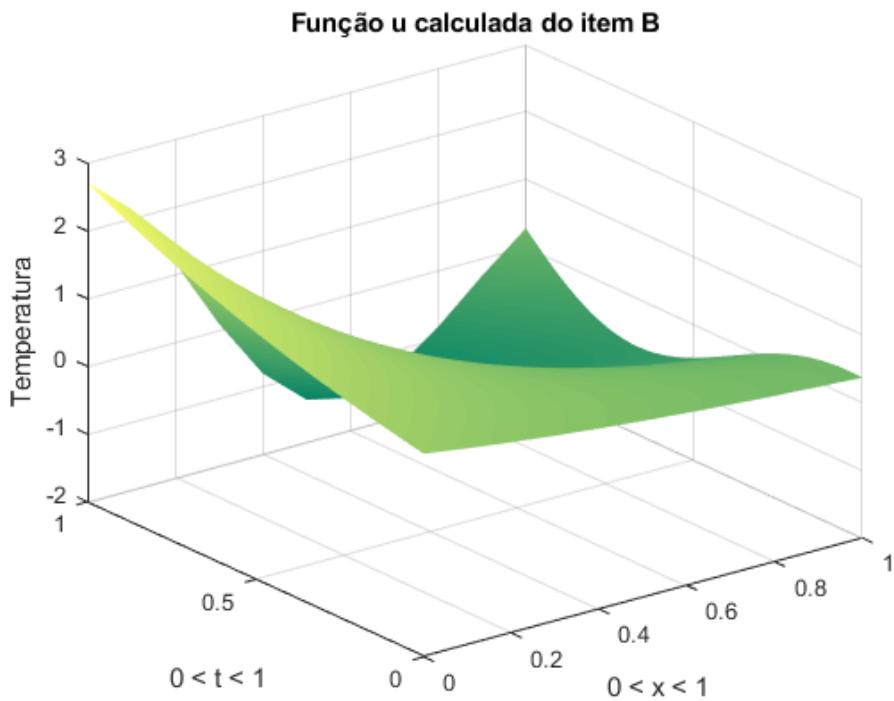


Figura 58: Curva da temperatura calculada com  $N = 10$  e  $\lambda = 0.5$

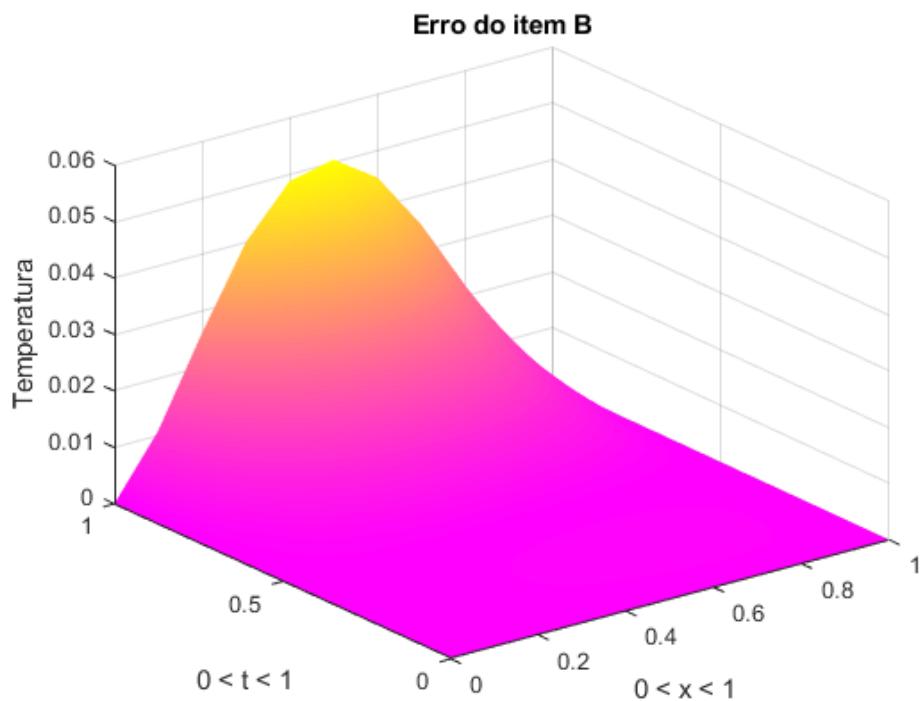


Figura 59: Erro com  $N = 10$  e  $\lambda = 0.5$

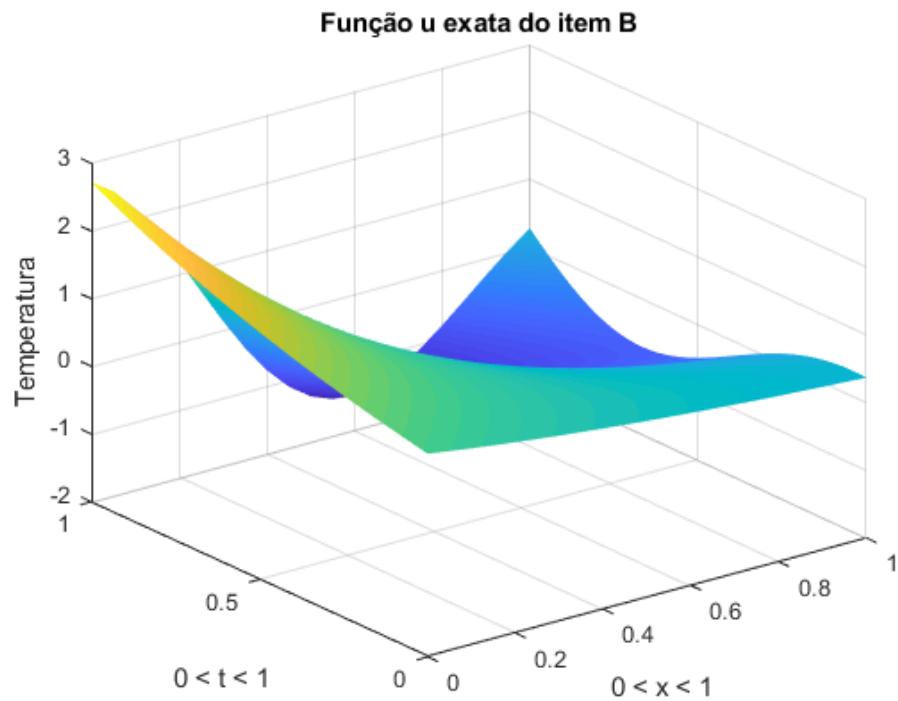


Figura 60: Curva da temperatura exata com  $N = 20$  e  $\lambda = 0.25$

**Função u calculada do item B**

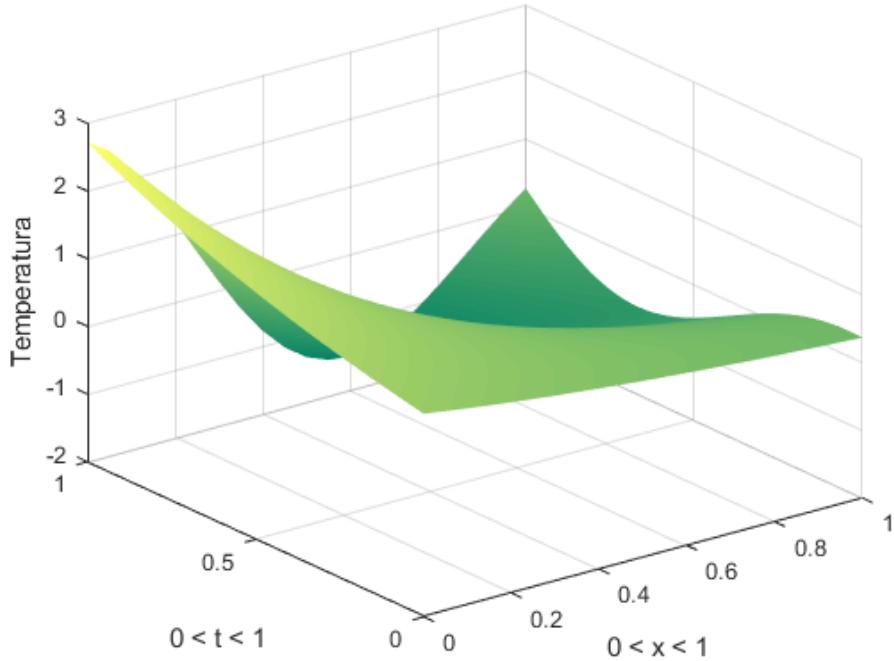


Figura 61: Curva da temperatura calculada com  $N = 20$  e  $\lambda = 0.25$

**Erro do item B**

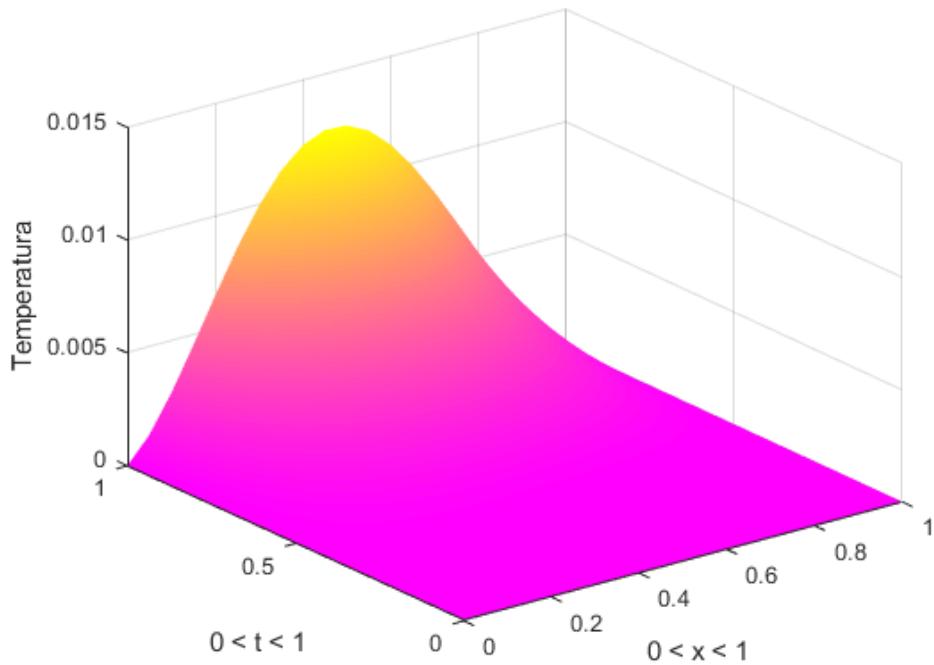


Figura 62: Erro com  $N = 20$  e  $\lambda = 0.25$

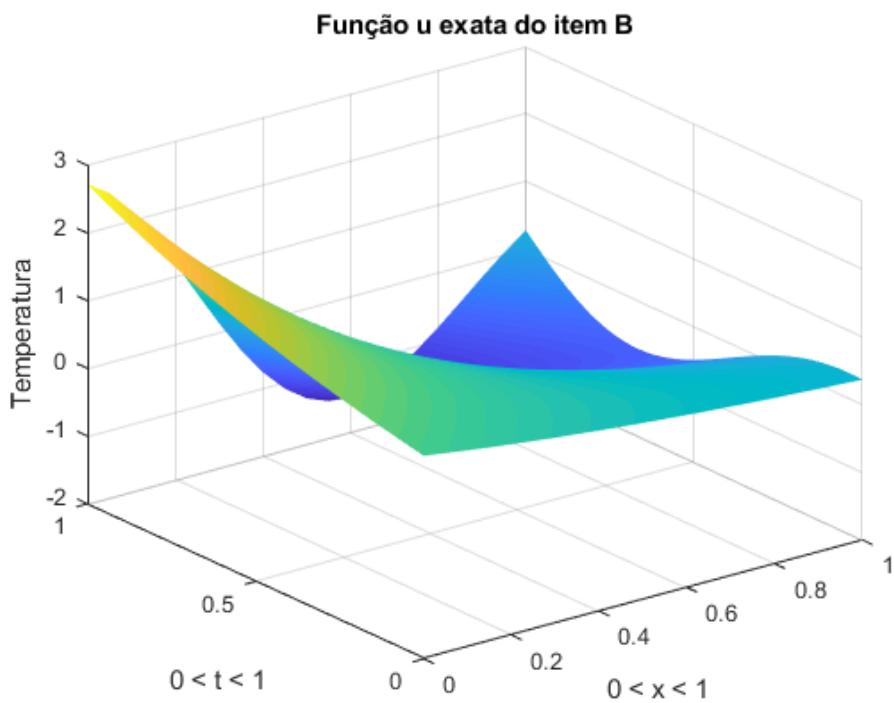


Figura 63: Curva da temperatura exata com  $N = 20$  e  $\lambda = 0.5$

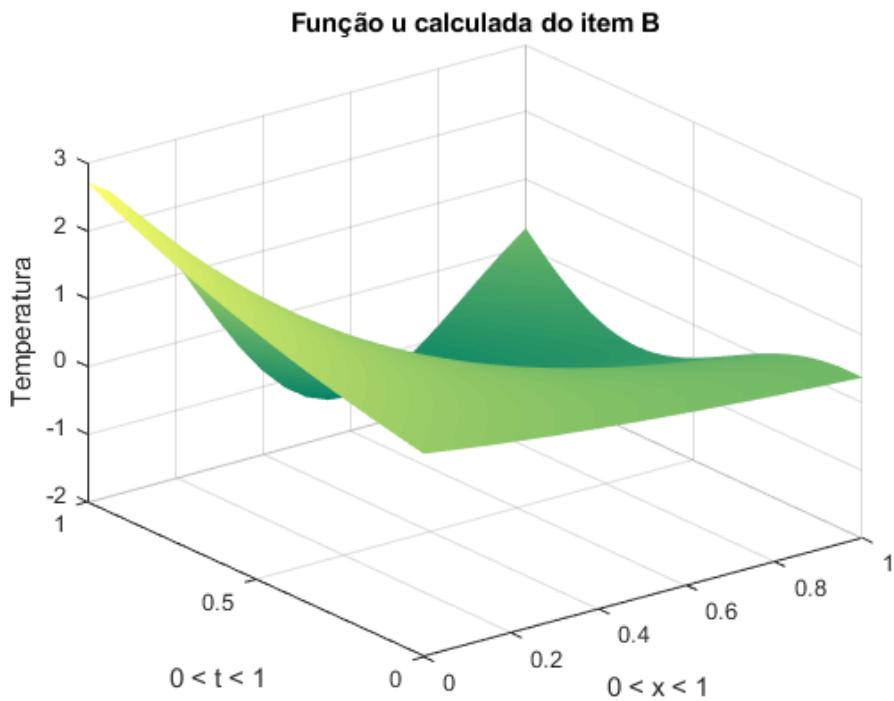


Figura 64: Curva da temperatura calculada com  $N = 20$  e  $\lambda = 0.5$

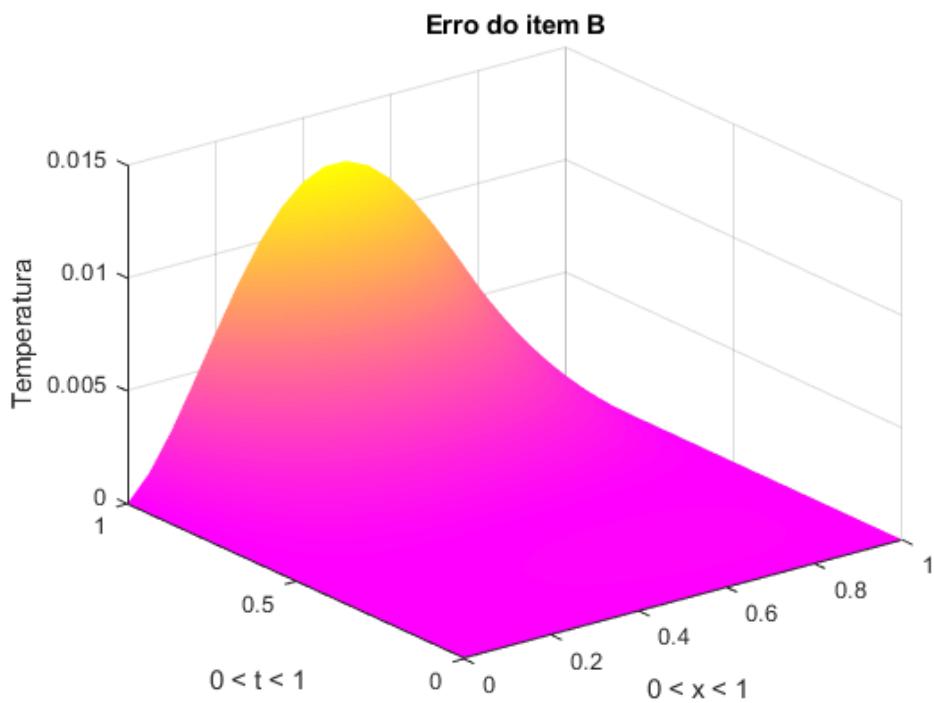


Figura 65: Erro com  $N = 20$  e  $\lambda = 0.5$

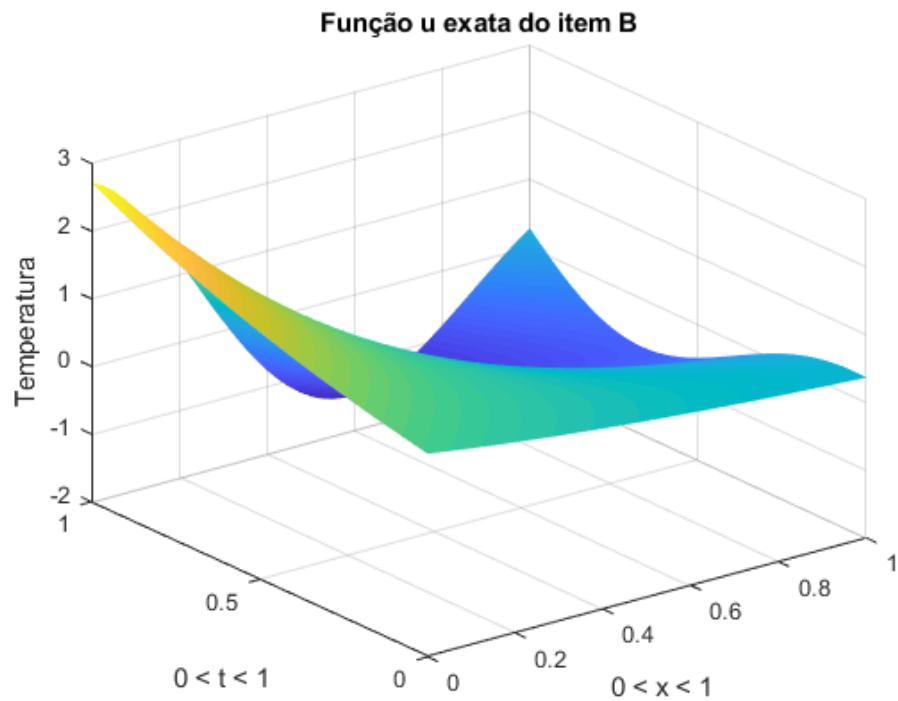


Figura 66: Curva da temperatura exata com  $N = 40$  e  $\lambda = 0.25$

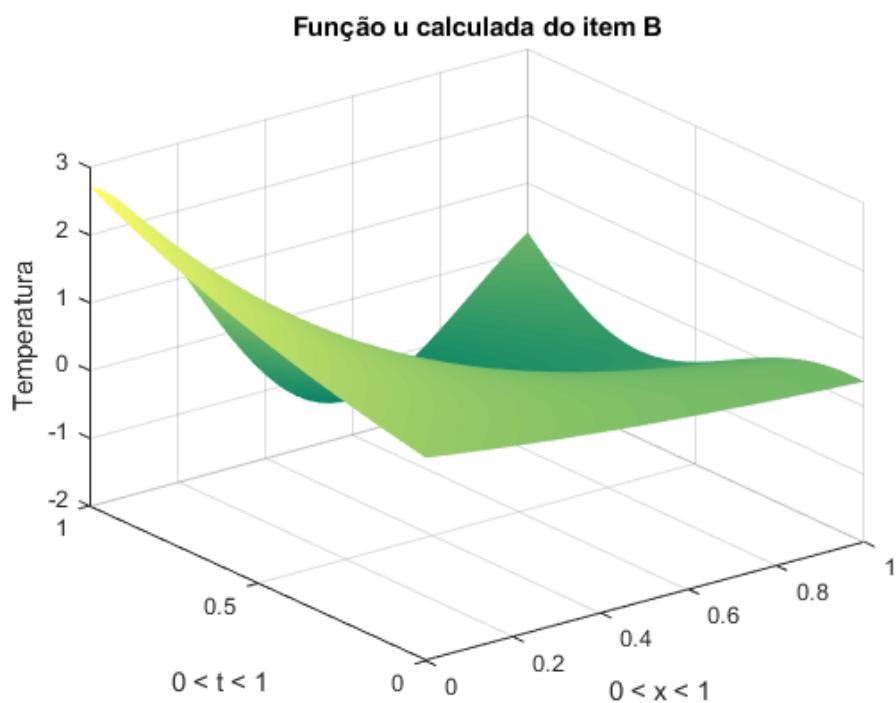


Figura 67: Curva da temperatura calculada com  $N = 40$  e  $\lambda = 0.25$

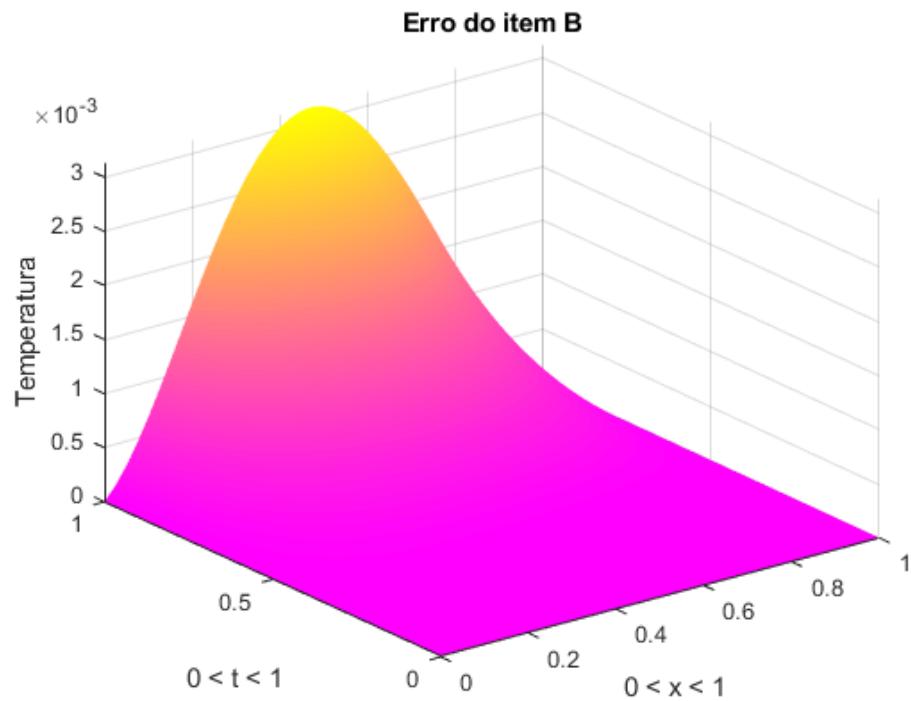


Figura 68: Erro com  $N = 40$  e  $\lambda = 0.25$

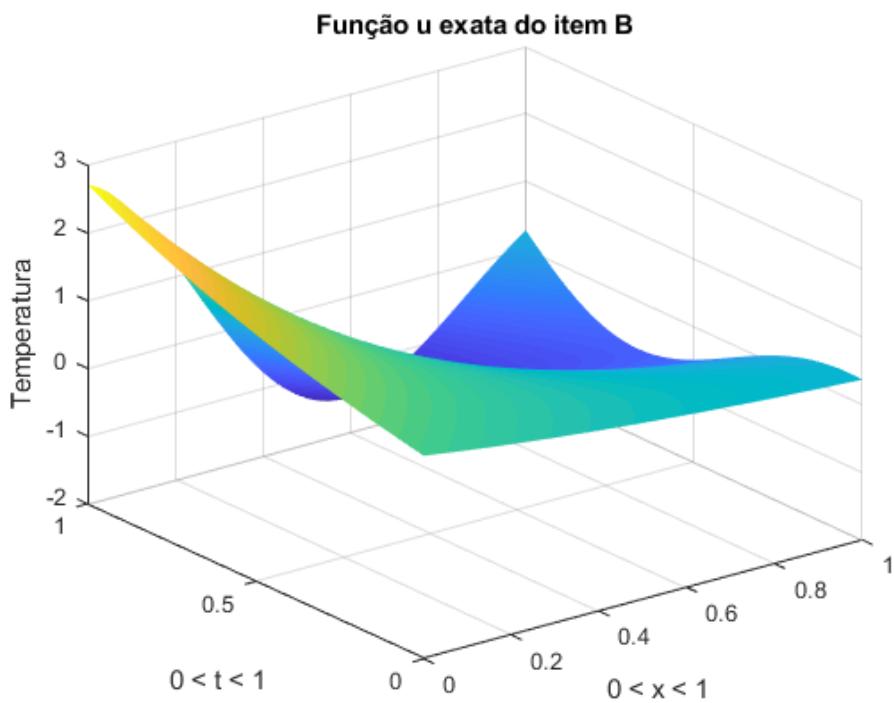


Figura 69: Curva da temperatura exata com  $N = 40$  e  $\lambda = 0.5$

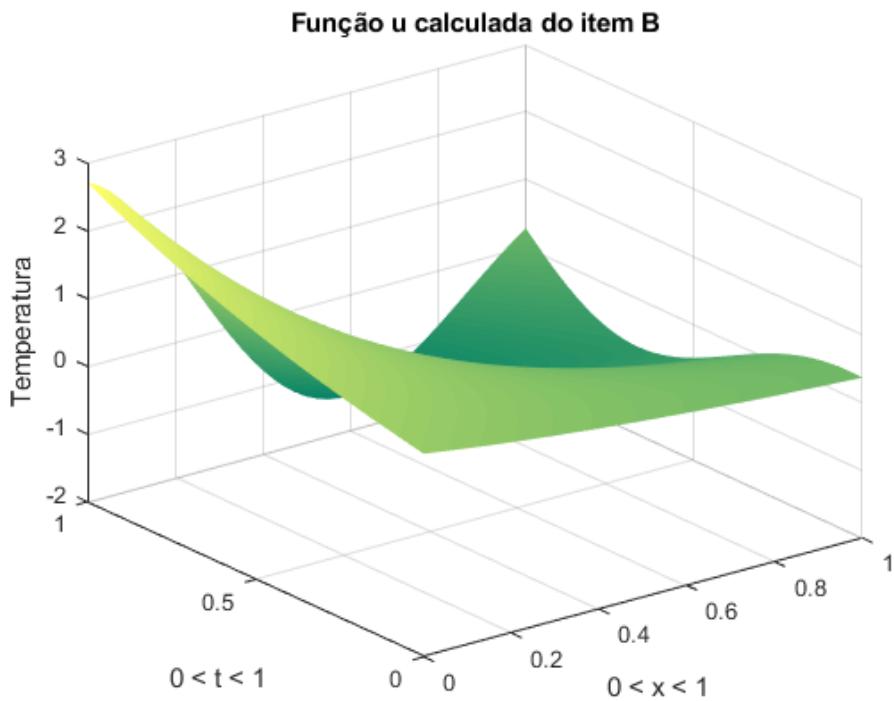


Figura 70: Curva da temperatura calculada com  $N = 40$  e  $\lambda = 0.5$

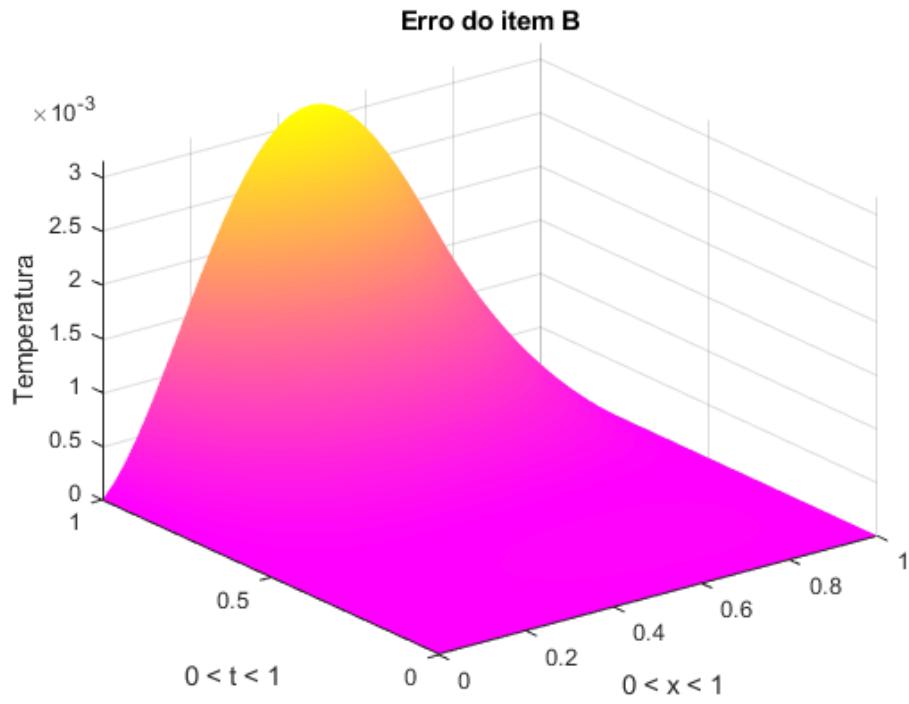


Figura 71: Erro com  $N = 40$  e  $\lambda = 0.5$

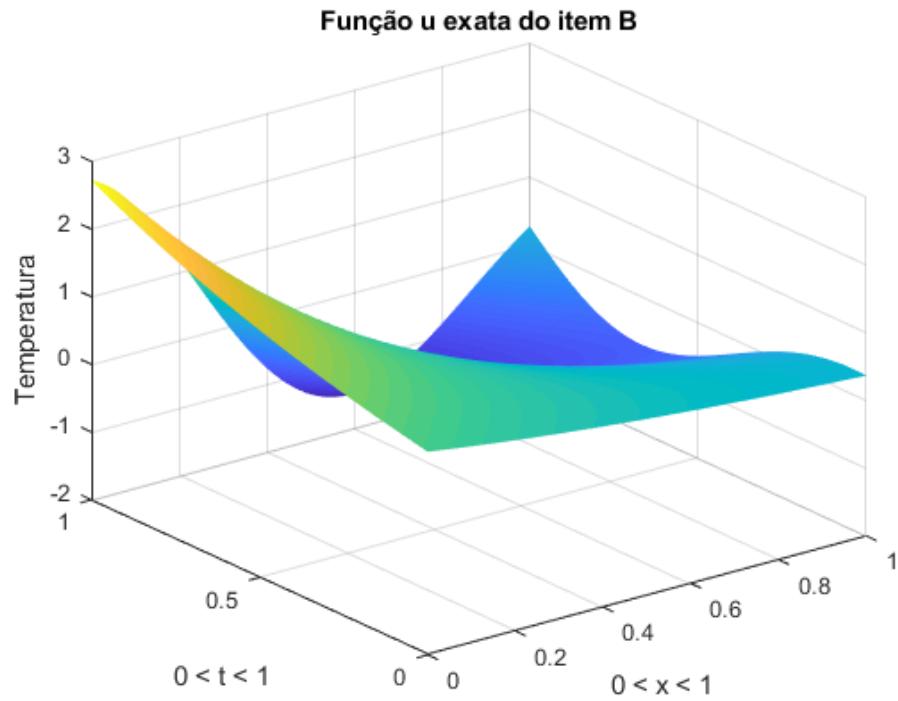


Figura 72: Curva da temperatura exata com  $N = 80$  e  $\lambda = 0.25$

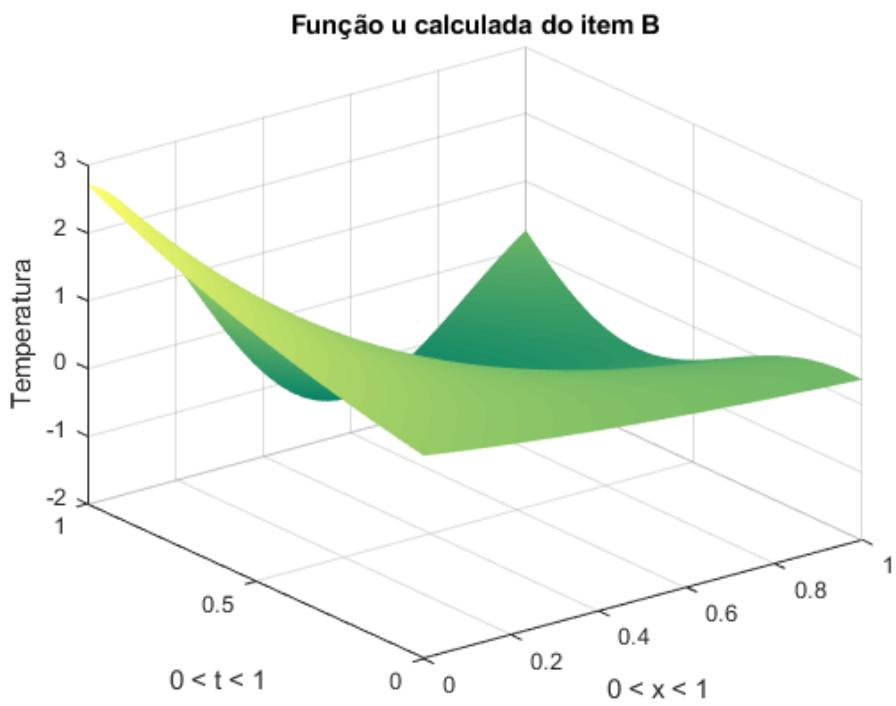


Figura 73: Curva da temperatura calculada com  $N = 80$  e  $\lambda = 0.25$

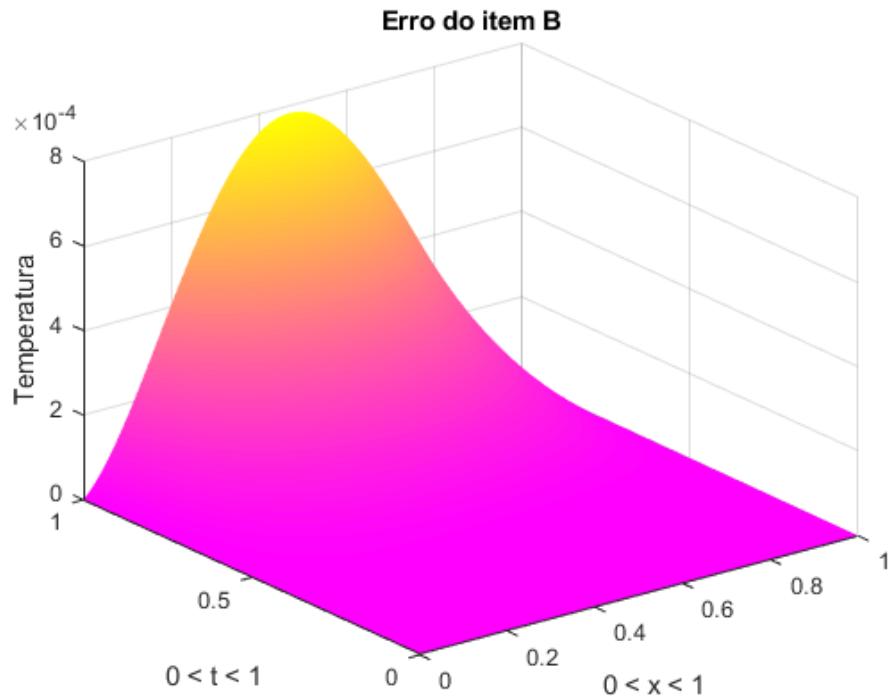


Figura 74: Erro com  $N = 80$  e  $\lambda = 0.25$

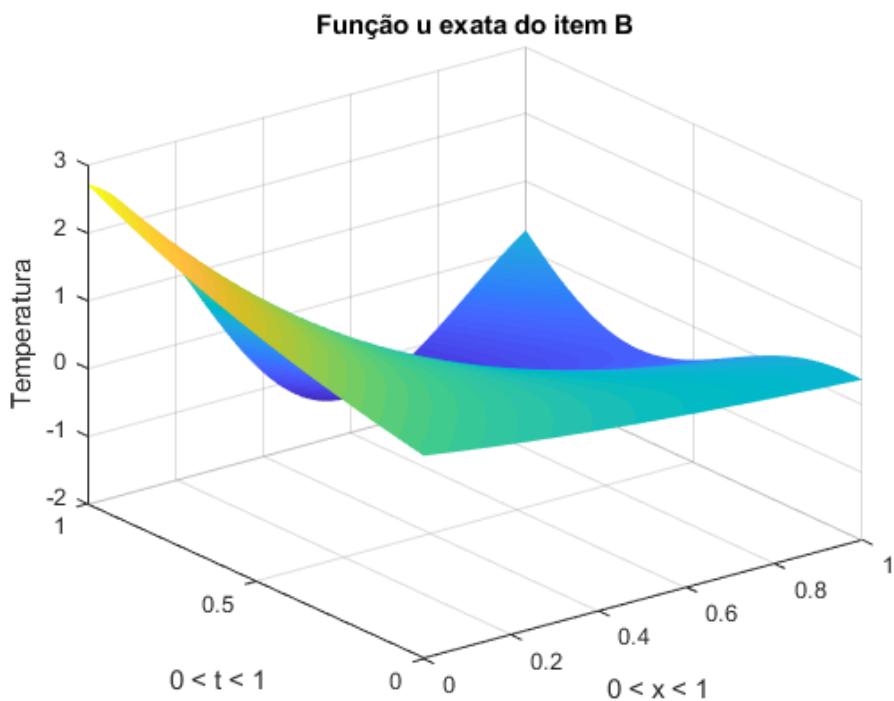


Figura 75: Curva da temperatura exata com  $N = 80$  e  $\lambda = 0.5$

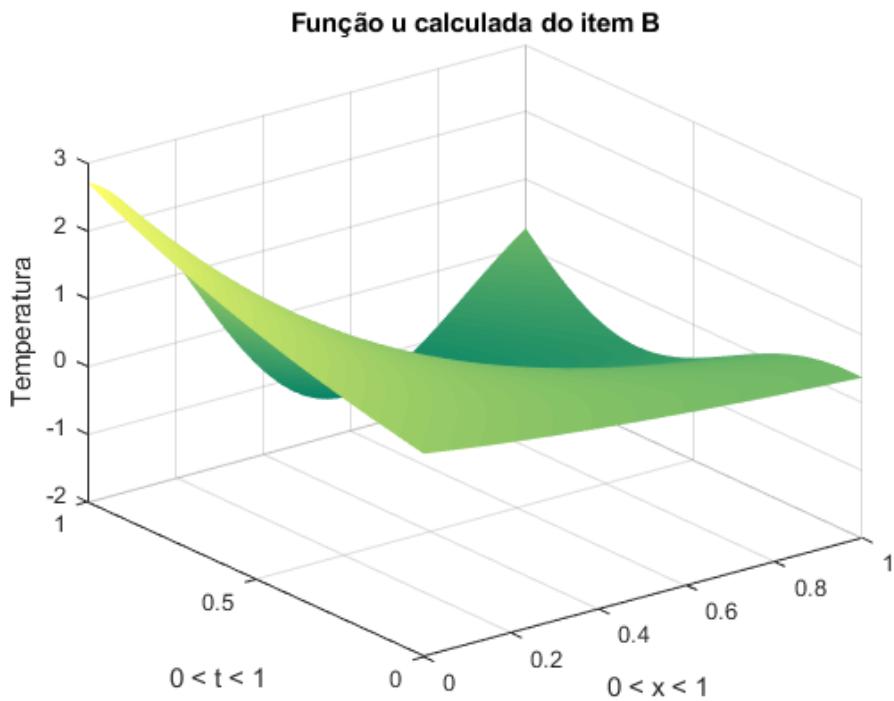


Figura 76: Curva da temperatura calculada com  $N = 80$  e  $\lambda = 0.5$

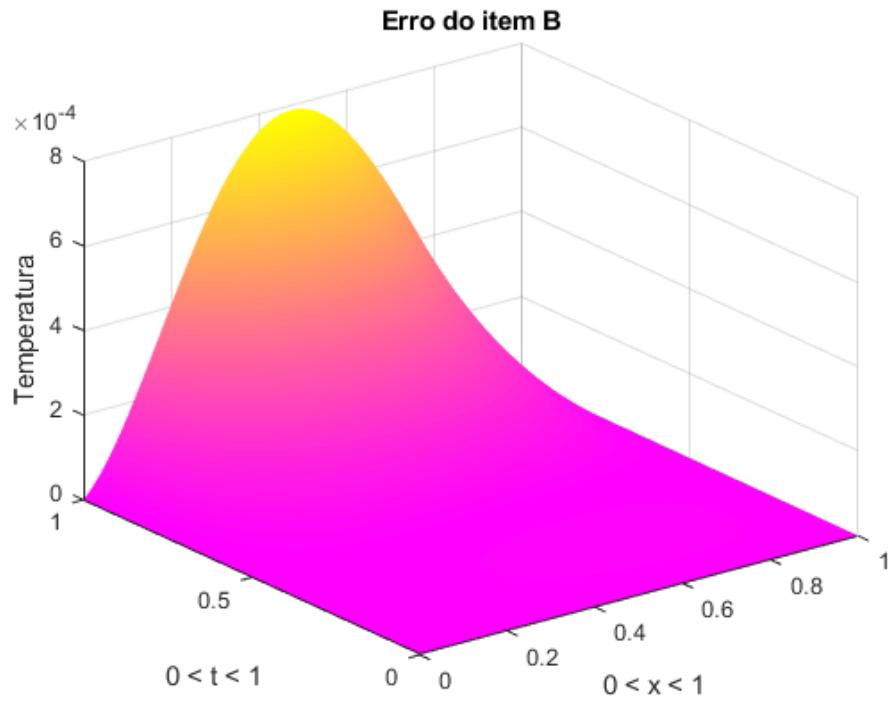


Figura 77: Erro com  $N = 80$  e  $\lambda = 0.5$

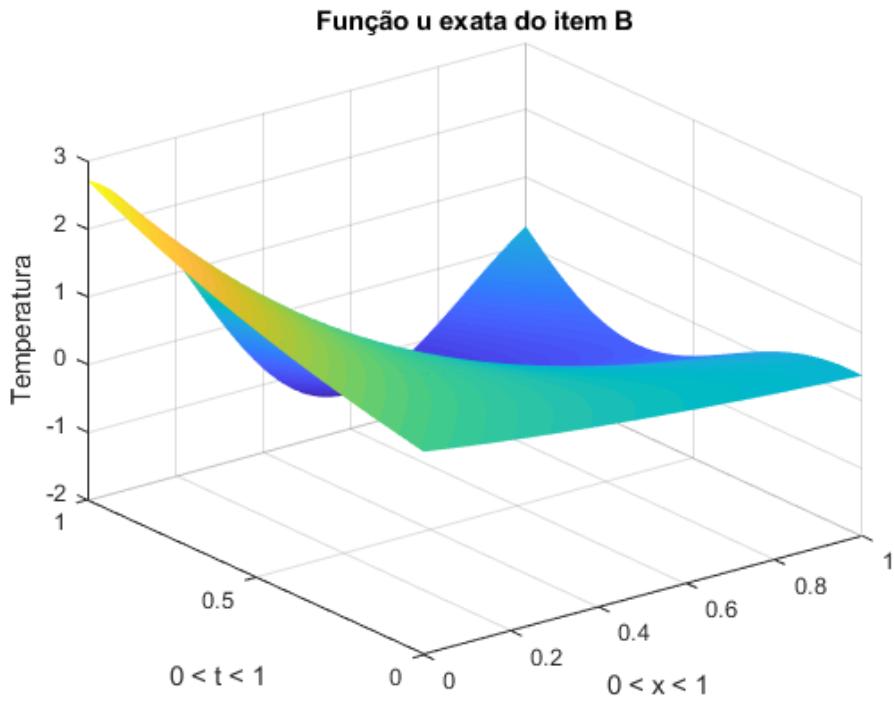


Figura 78: Curva da temperatura exata com  $N = 160$  e  $\lambda = 0.25$

**Função u calculada do item B**

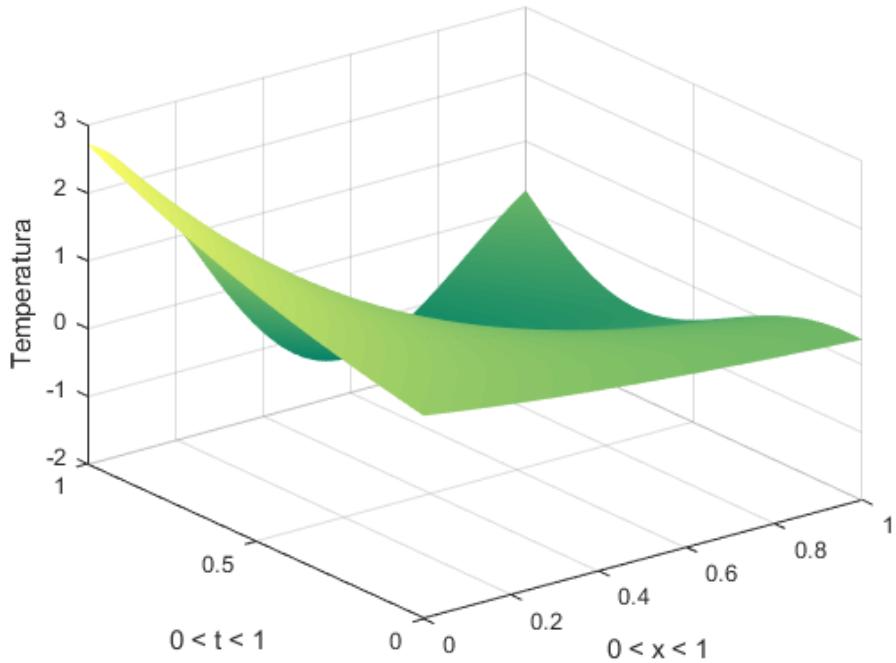


Figura 79: Curva da temperatura calculada com  $N = 160$  e  $\lambda = 0.25$

**Erro do item B**

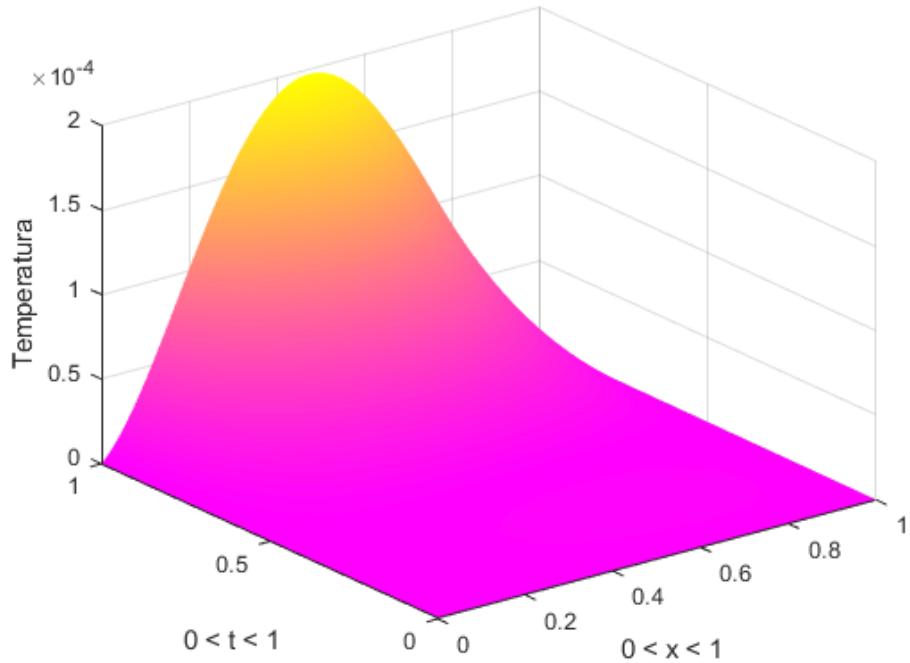


Figura 80: Erro com  $N = 160$  e  $\lambda = 0.25$

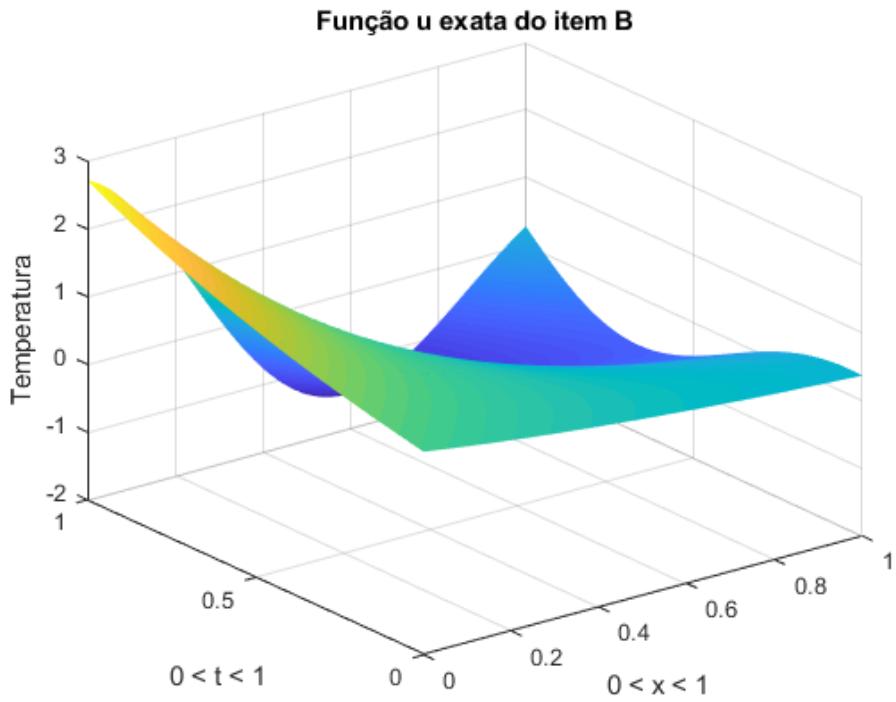


Figura 81: Curva da temperatura exata com  $N = 160$  e  $\lambda = 0.5$

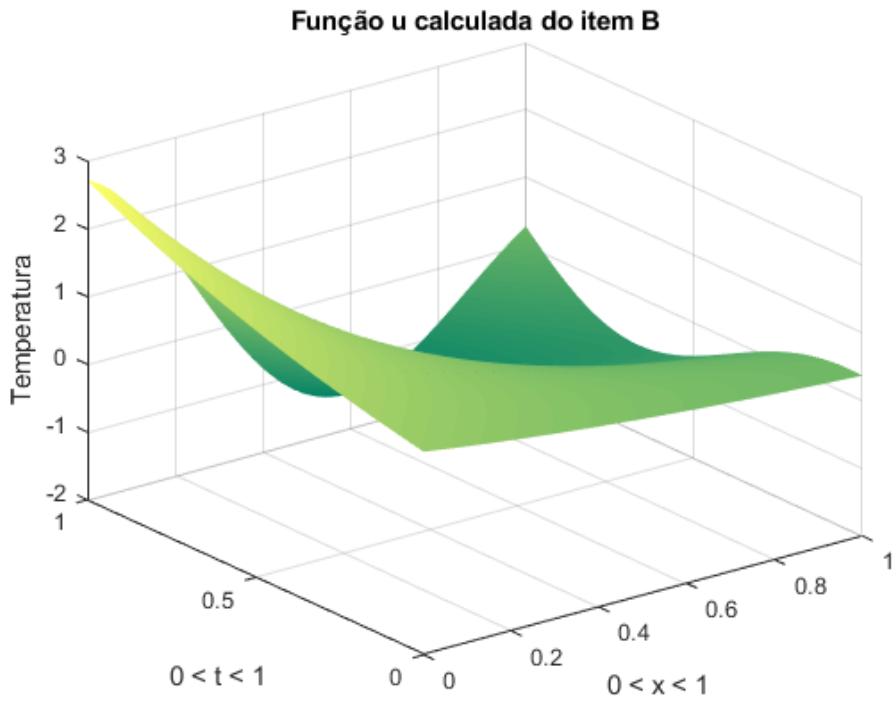


Figura 82: Curva da temperatura calculada com  $N = 160$  e  $\lambda = 0.5$

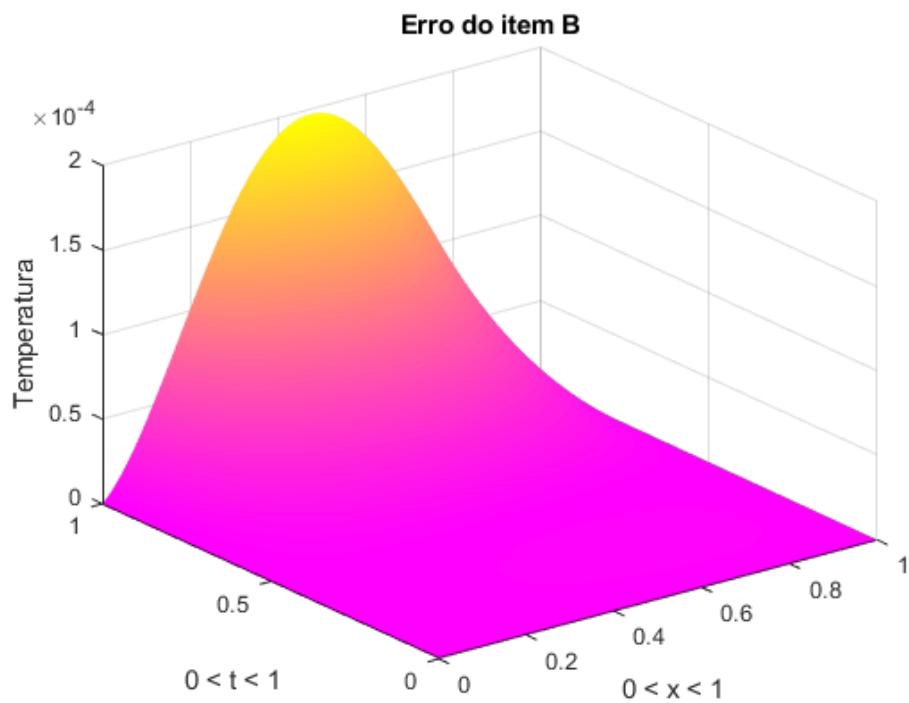


Figura 83: Erro com  $N = 160$  e  $\lambda = 0.5$

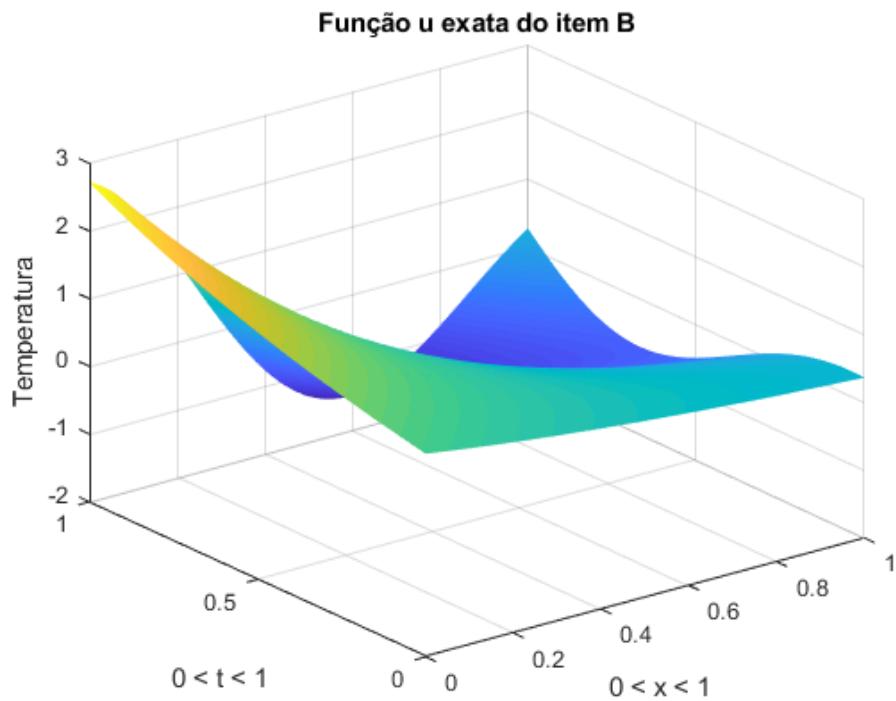


Figura 84: Curva da temperatura exata com  $N = 320$  e  $\lambda = 0.25$

**Função u calculada do item B**

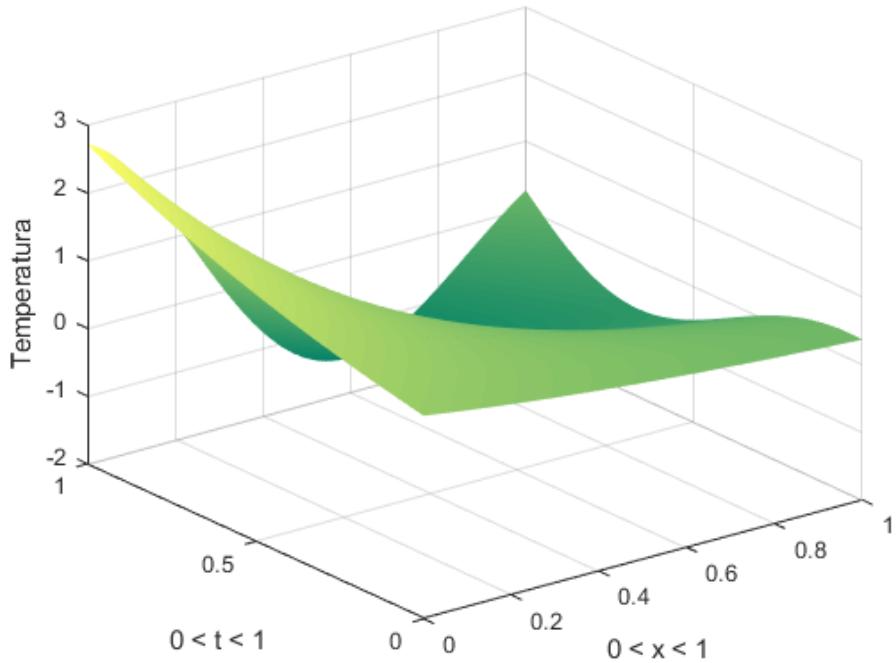


Figura 85: Curva da temperatura calculada com  $N = 320$  e  $\lambda = 0.25$

**Erro do item B**

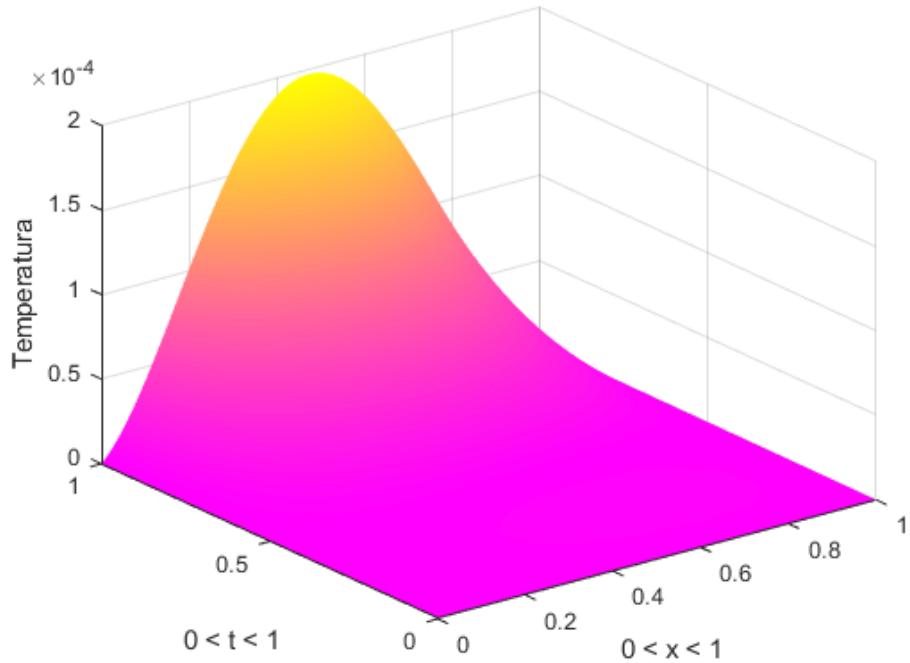


Figura 86: Erro com  $N = 320$  e  $\lambda = 0.25$

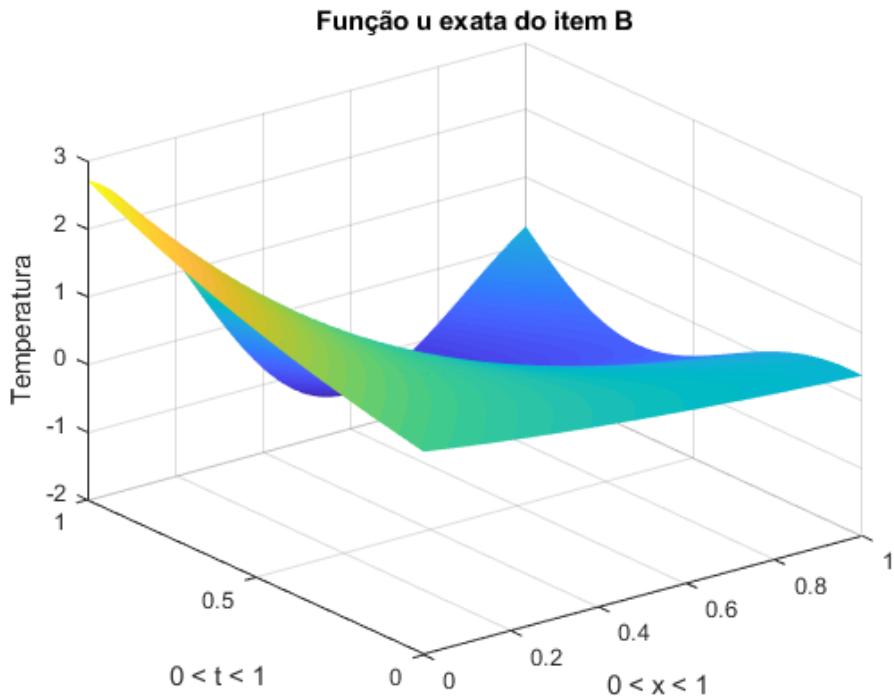


Figura 87: Curva da temperatura exata com  $N = 320$  e  $\lambda = 0.5$

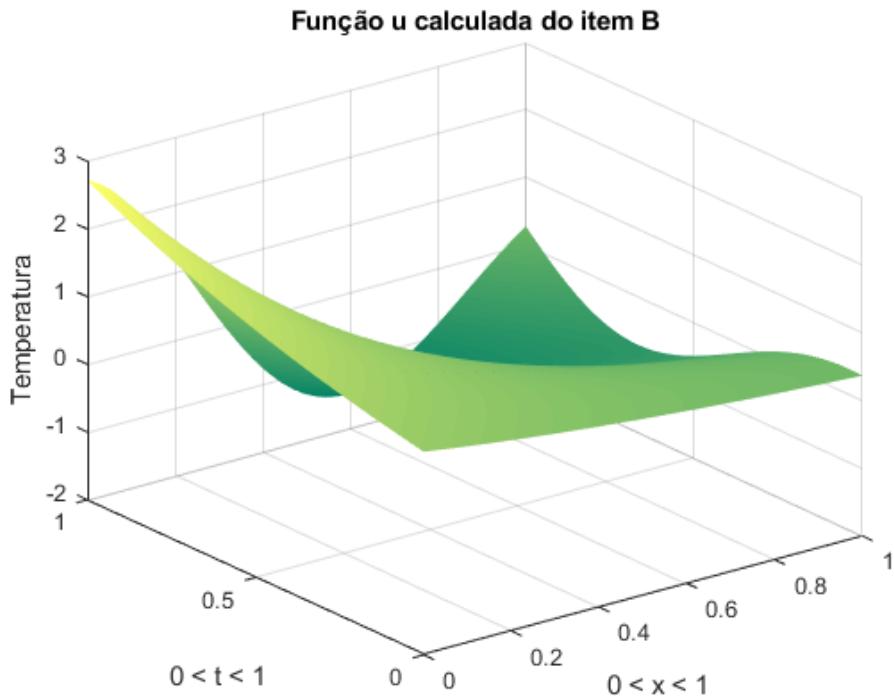


Figura 88: Curva da temperatura calculada com  $N = 320$  e  $\lambda = 0.5$

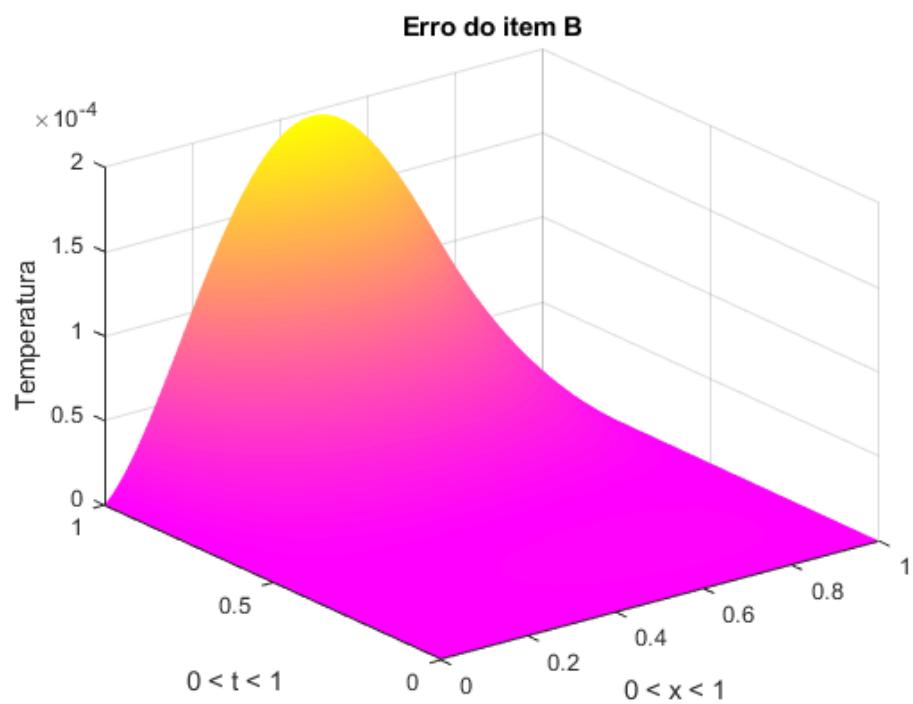


Figura 89: Erro com  $N = 320$  e  $\lambda = 0.5$

## E Curvas do item 1C

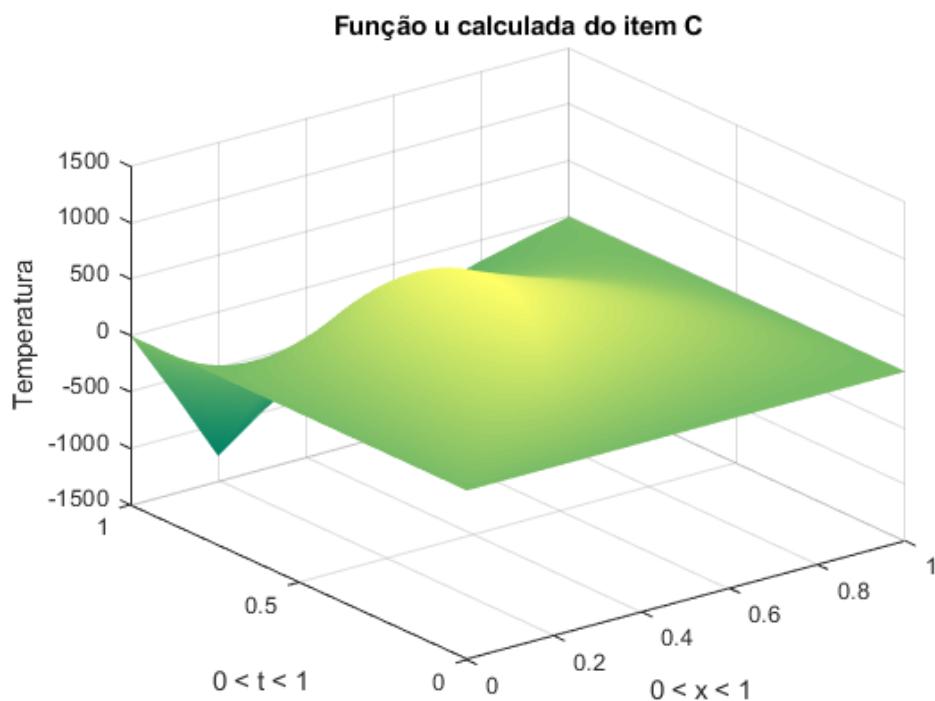


Figura 90: Curva da temperatura calculada com  $N = 10$  e  $\lambda = 0.25$

**Função u calculada do item C**

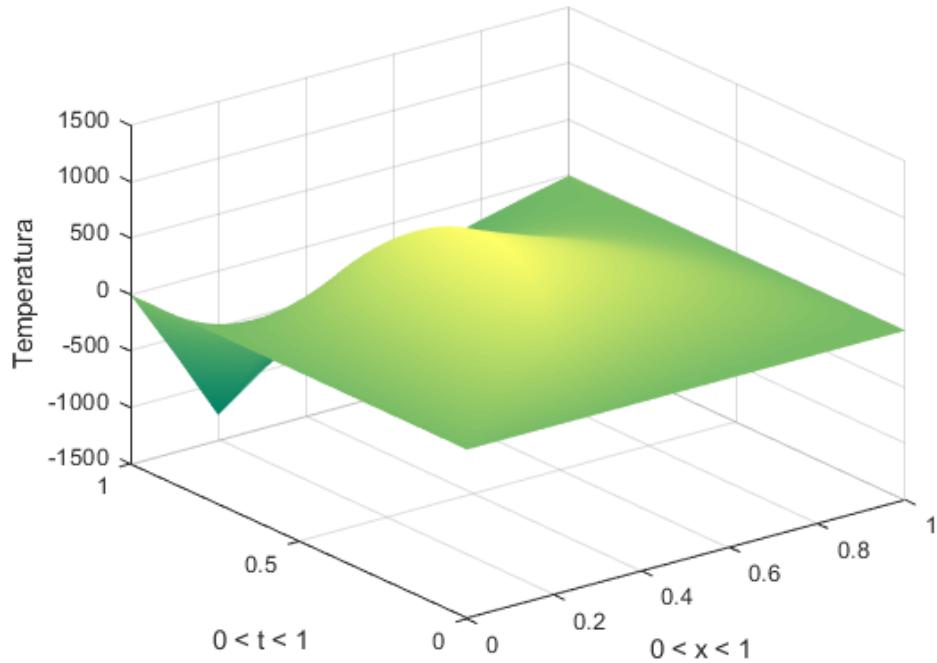


Figura 91: Curva da temperatura calculada com  $N = 10$  e  $\lambda = 0.5$

**Função u calculada do item C**

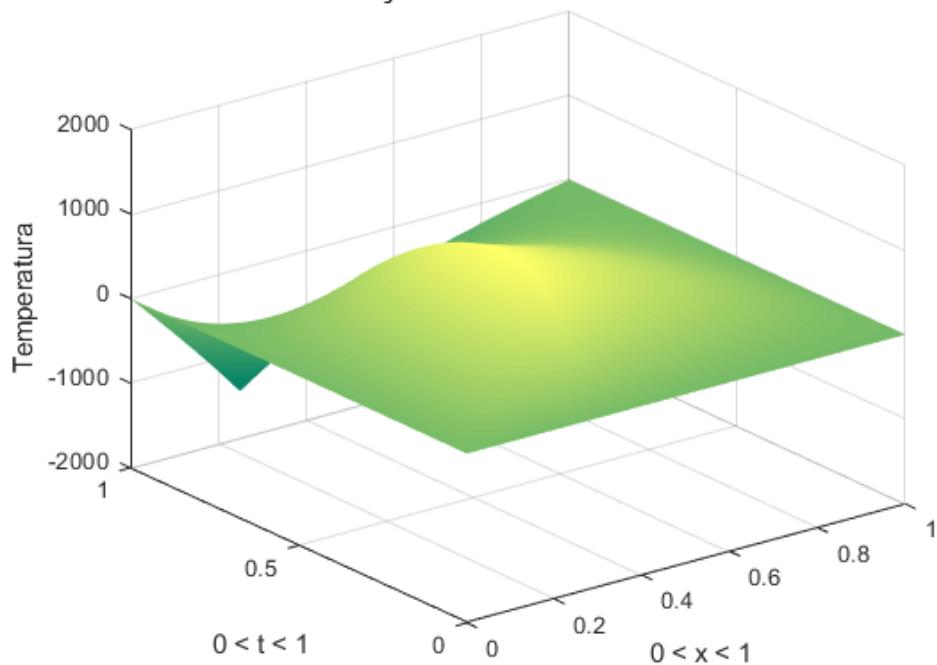


Figura 92: Curva da temperatura exata com  $N = 20$  e  $\lambda = 0.25$

**Função u calculada do item C**

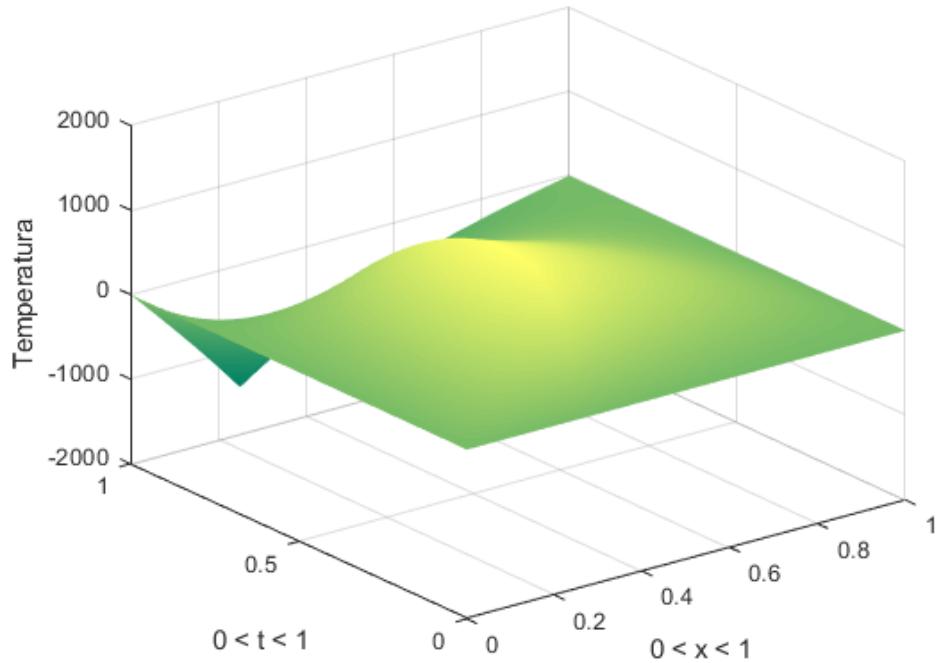


Figura 93: Curva da temperatura calculada com  $N = 20$  e  $\lambda = 0.5$

**Função u calculada do item C**

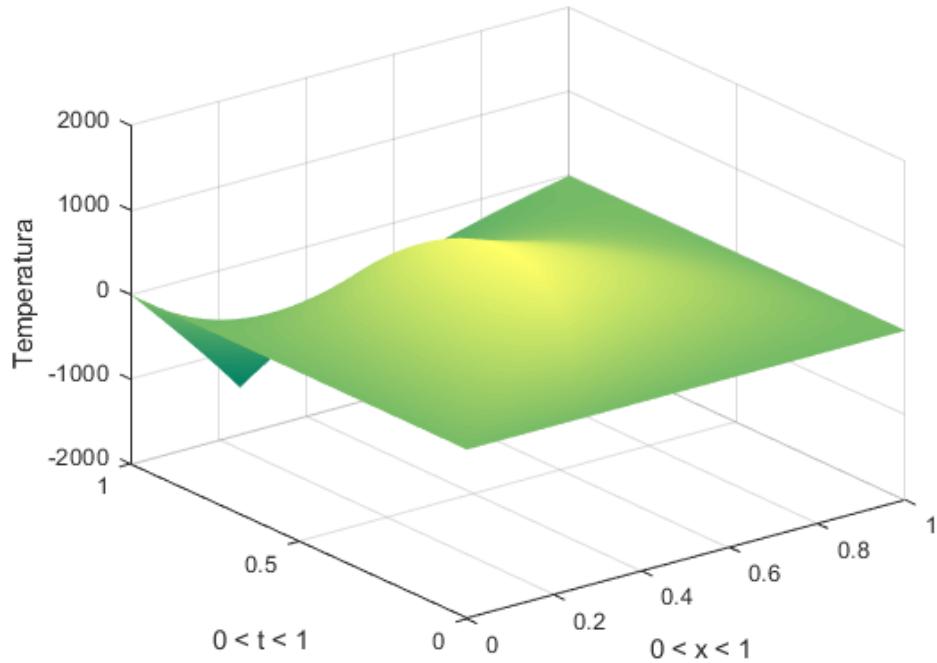


Figura 94: Curva da temperatura calculada com  $N = 40$  e  $\lambda = 0.25$

**Função u calculada do item C**

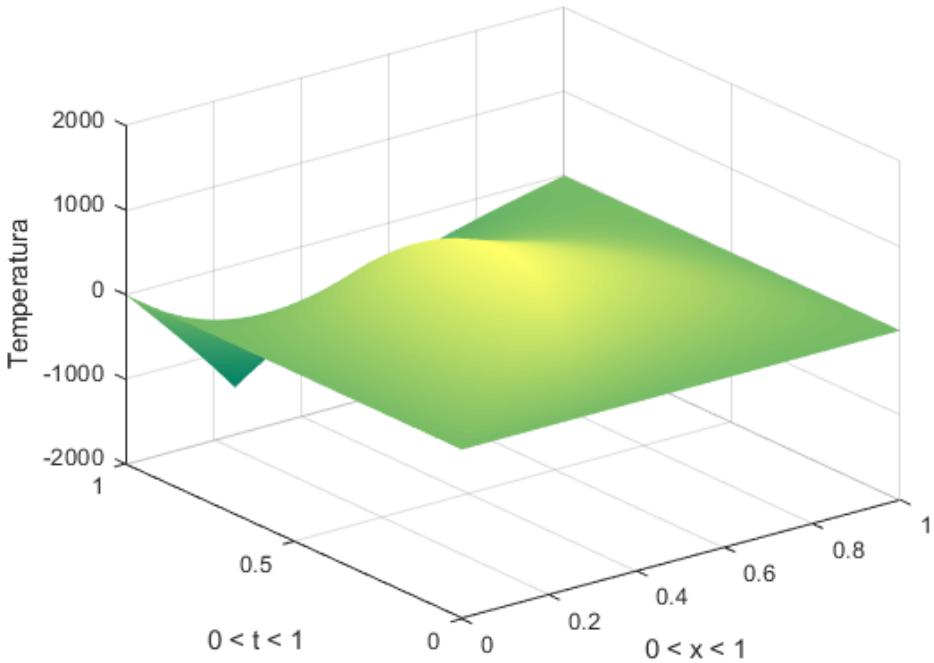


Figura 95: Curva da temperatura calculada com  $N = 40$  e  $\lambda = 0.5$

**Função u calculada do item C**

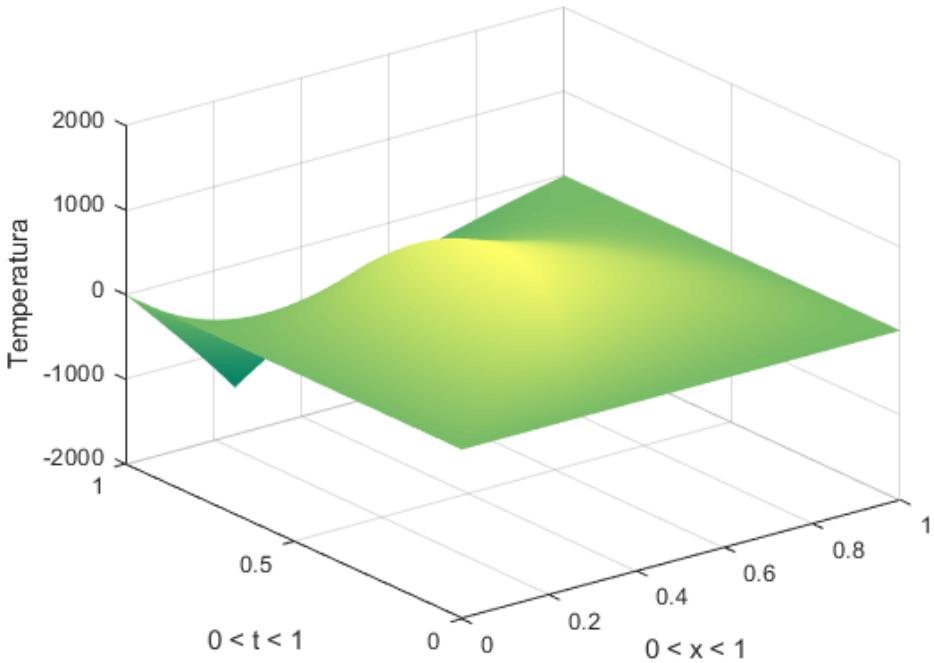


Figura 96: Curva da temperatura calculada com  $N = 80$  e  $\lambda = 0.25$

**Função u calculada do item C**

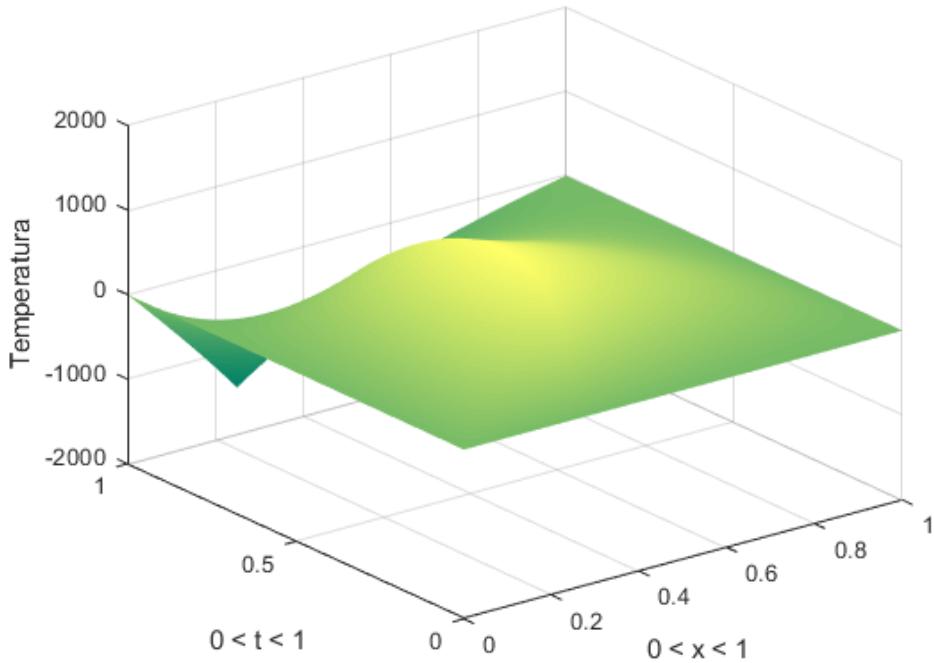


Figura 97: Curva da temperatura calculada com  $N = 80$  e  $\lambda = 0.5$

**Função u calculada do item C**

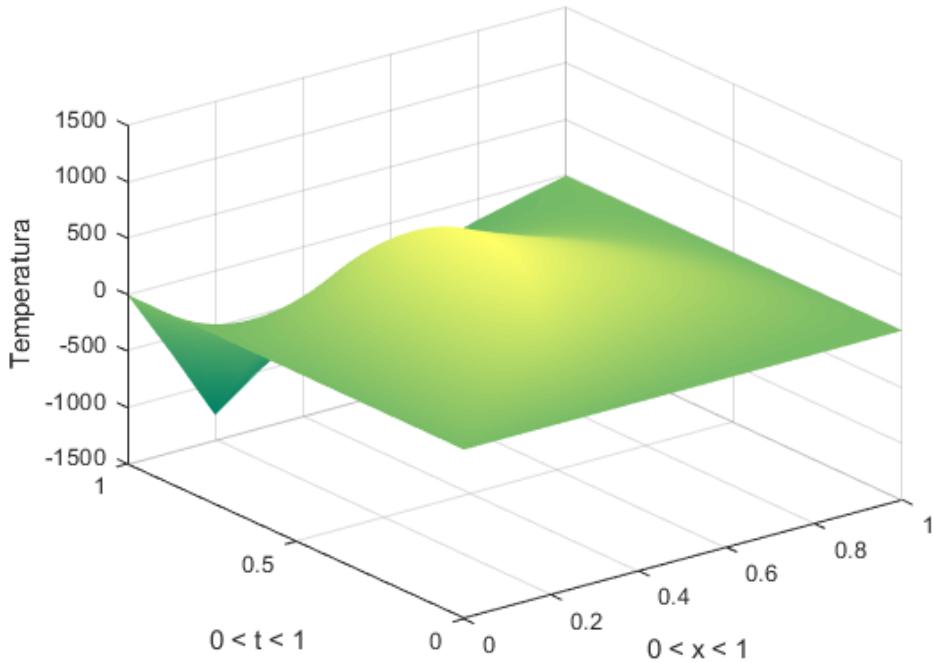


Figura 98: Curva da temperatura calculada com  $N = 160$  e  $\lambda = 0.25$

**Função u calculada do item C**

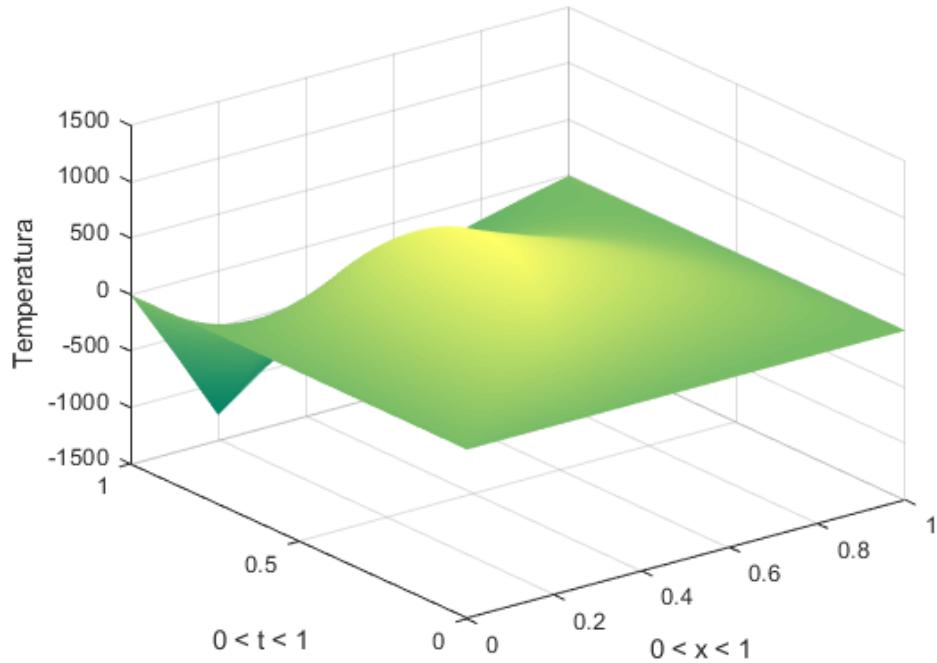


Figura 99: Curva da temperatura calculada com  $N = 160$  e  $\lambda = 0.5$

**Função u calculada do item C**

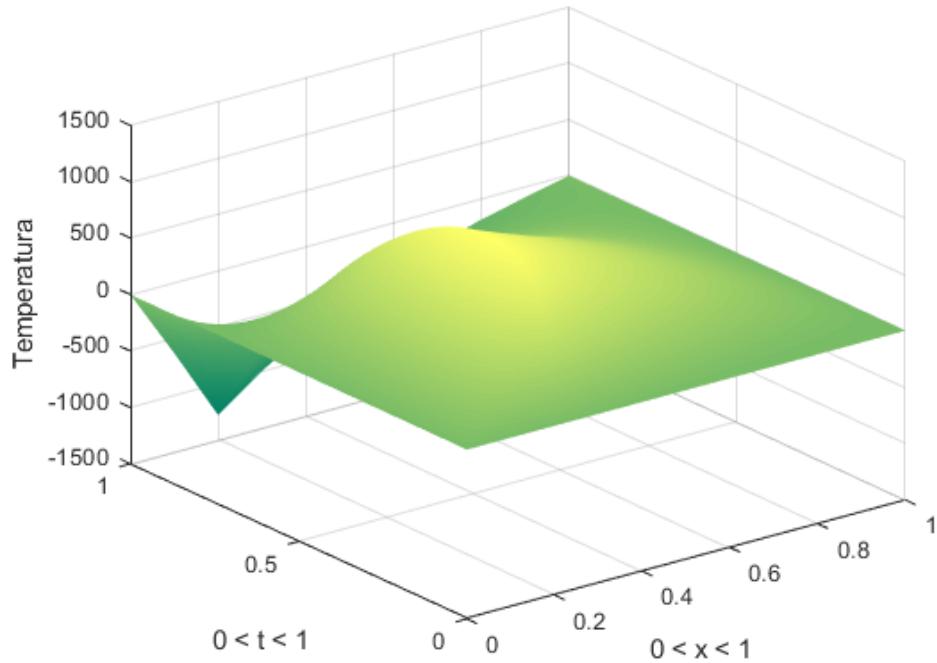


Figura 100: Curva da temperatura calculada com  $N = 320$  e  $\lambda = 0.25$

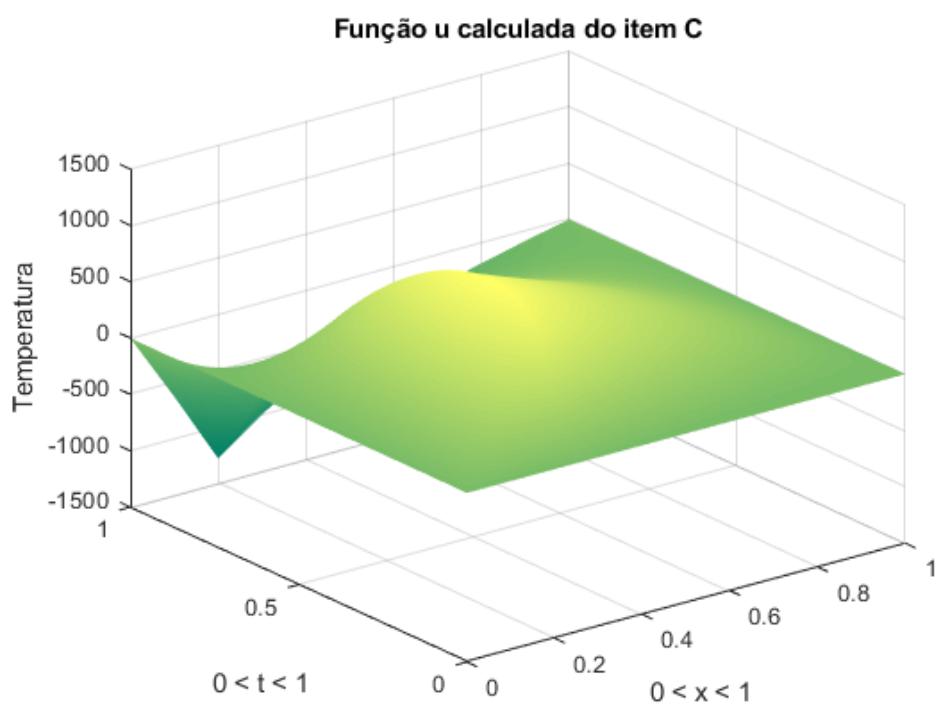


Figura 101: Curva da temperatura calculada com  $N = 320$  e  $\lambda = 0.5$

## F Curvas do item 2B

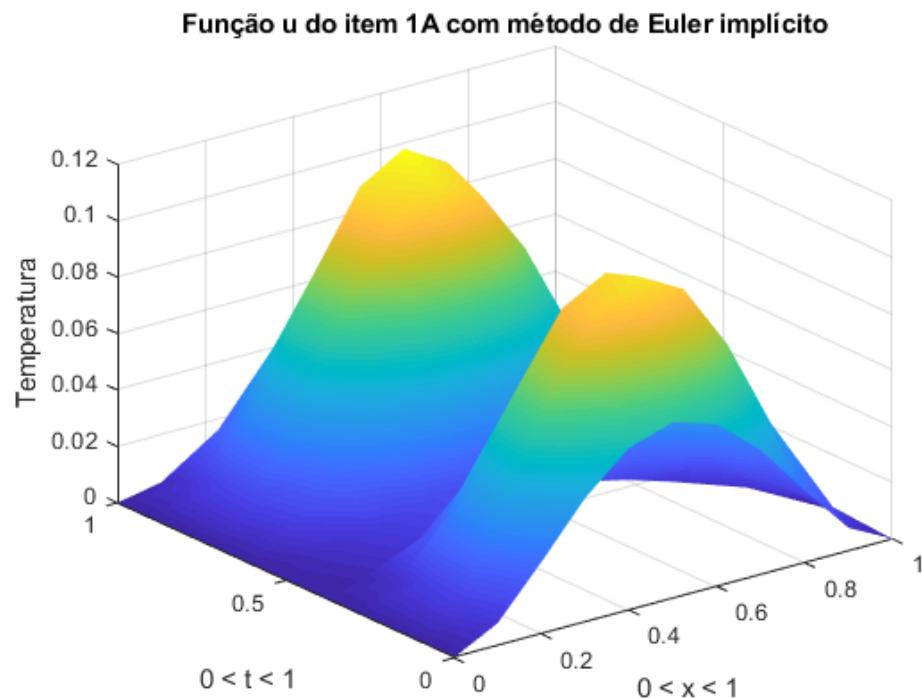


Figura 102: Curva da temperatura calculada com  $N = M = 10$

**Erro do item 1A com método de Euler implícito**

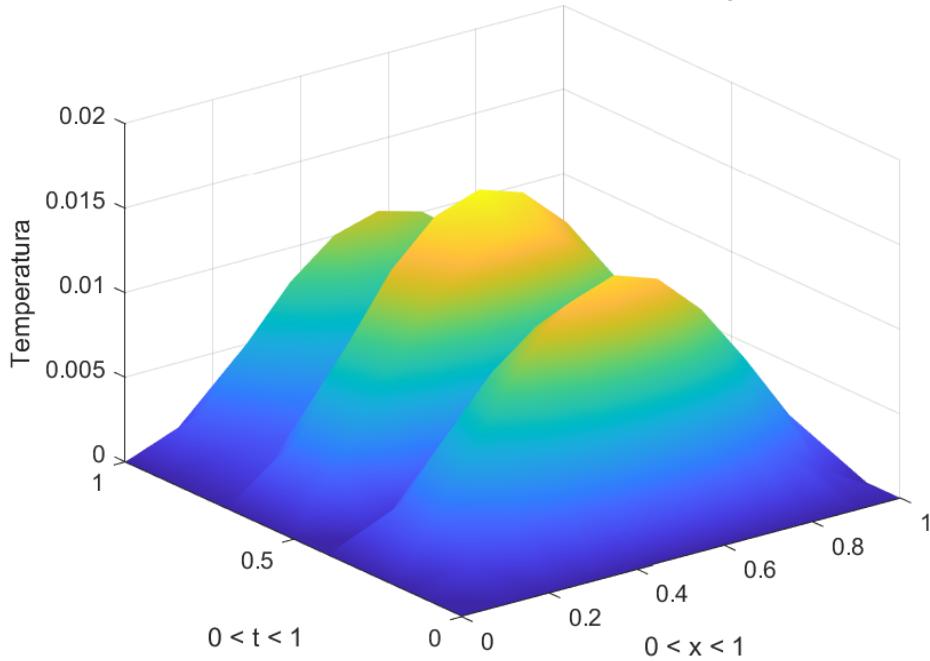


Figura 103: Erro com  $N = M = 10$

**Função u do item 1A com método de Euler implícito**

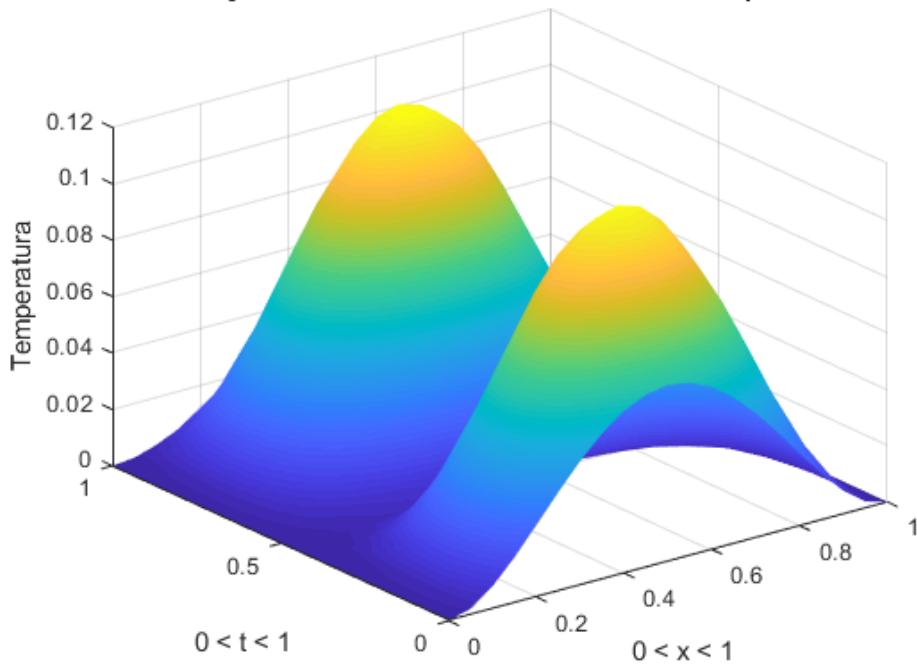


Figura 104: Curva da temperatura calculada com  $N = M = 20$

**Erro do item 1A com método de Euler implícito**

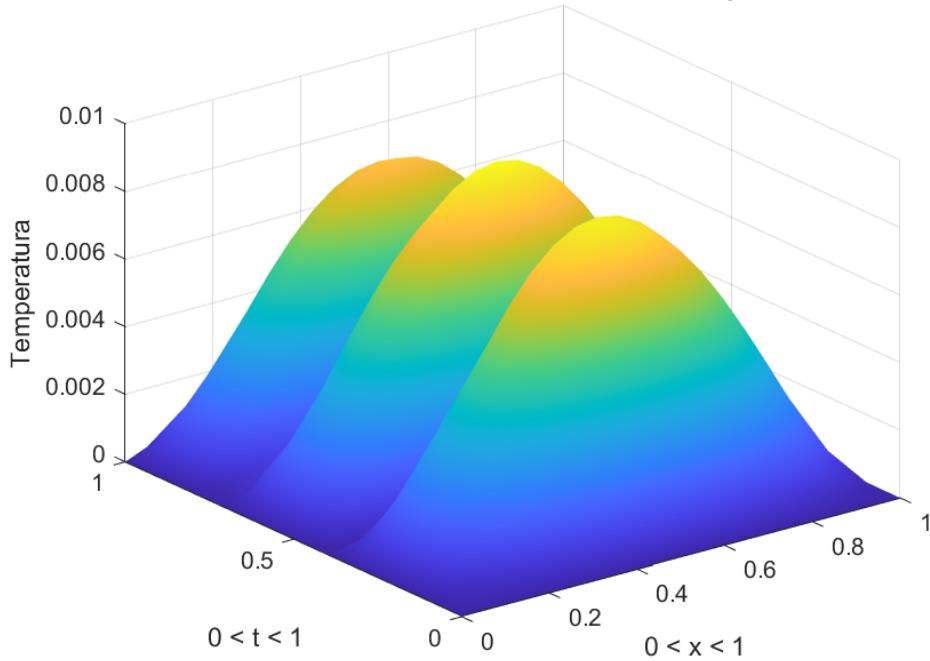


Figura 105: Erro com  $N = M = 20$

**Função u do item 1A com método de Euler implícito**

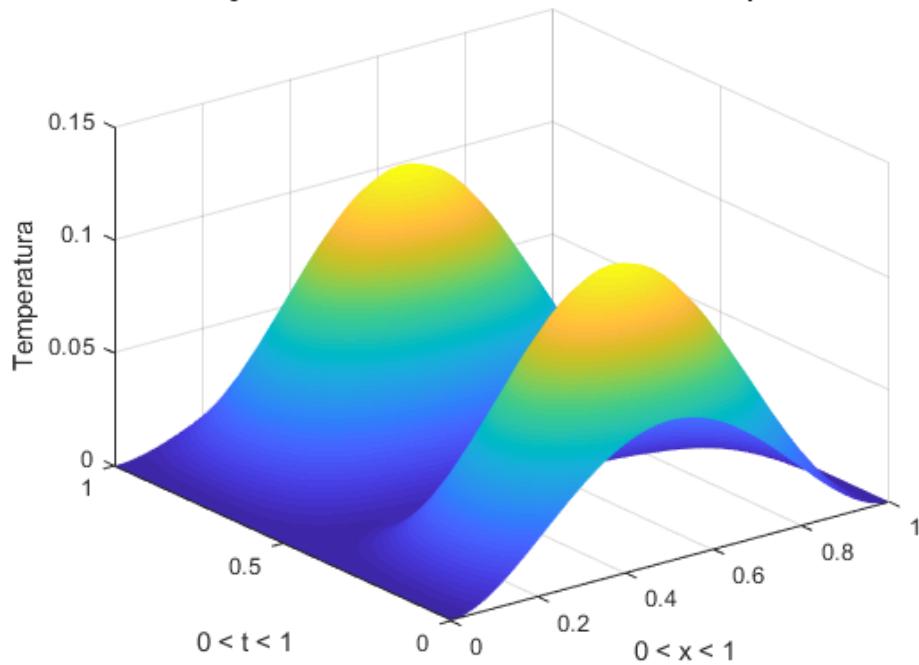


Figura 106: Curva da temperatura calculada com  $N = M = 40$

**Erro do item 1A com método de Euler implícito**

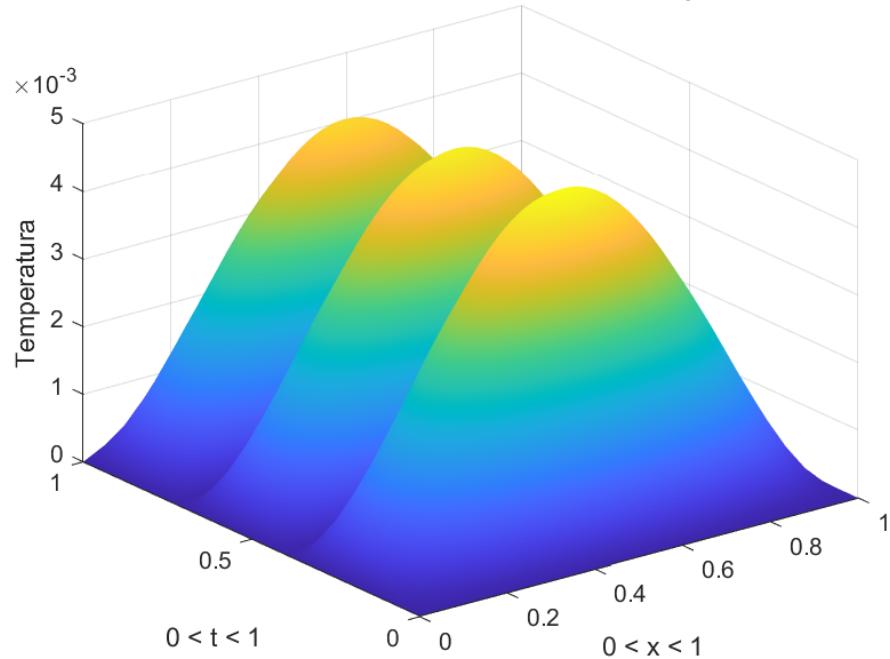


Figura 107: Erro com  $N = M = 40$

**Função u do item 1A com método de Euler implícito**

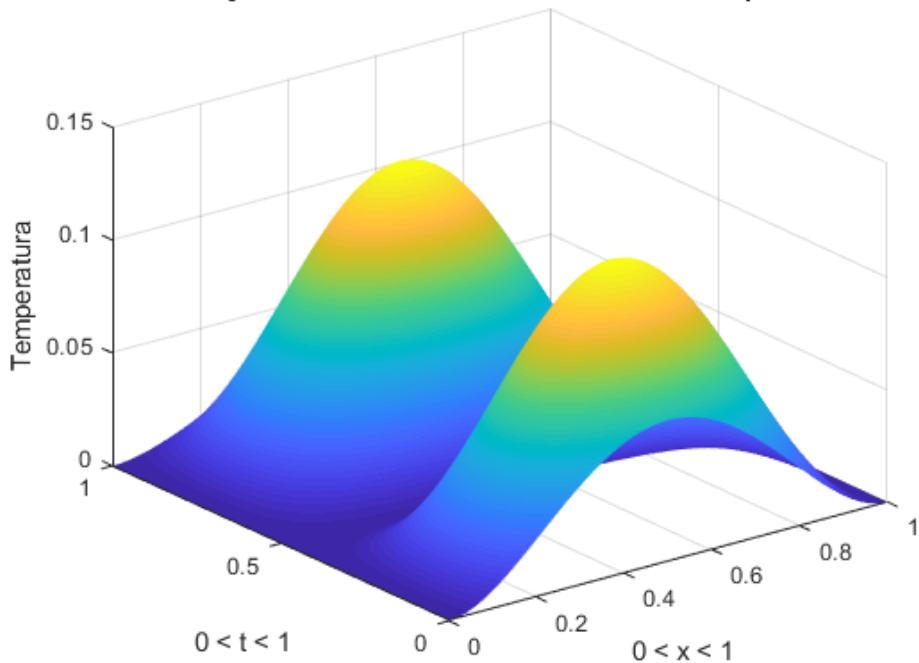


Figura 108: Curva da temperatura calculada com  $N = M = 80$

**Erro do item 1A com método de Euler implícito**

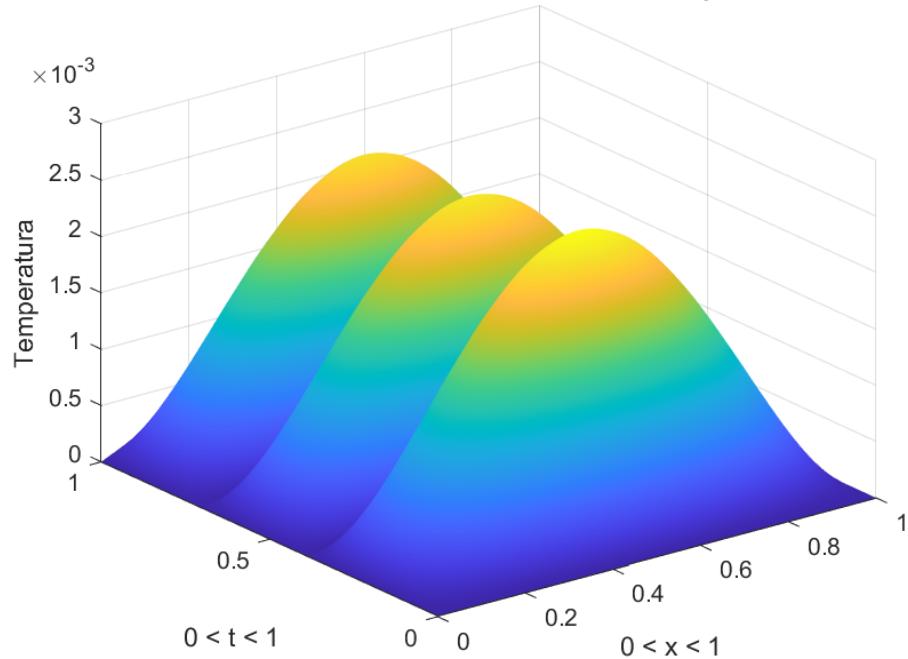


Figura 109: Erro com  $N = M = 80$

**Função u do item 1A com método de Euler implícito**

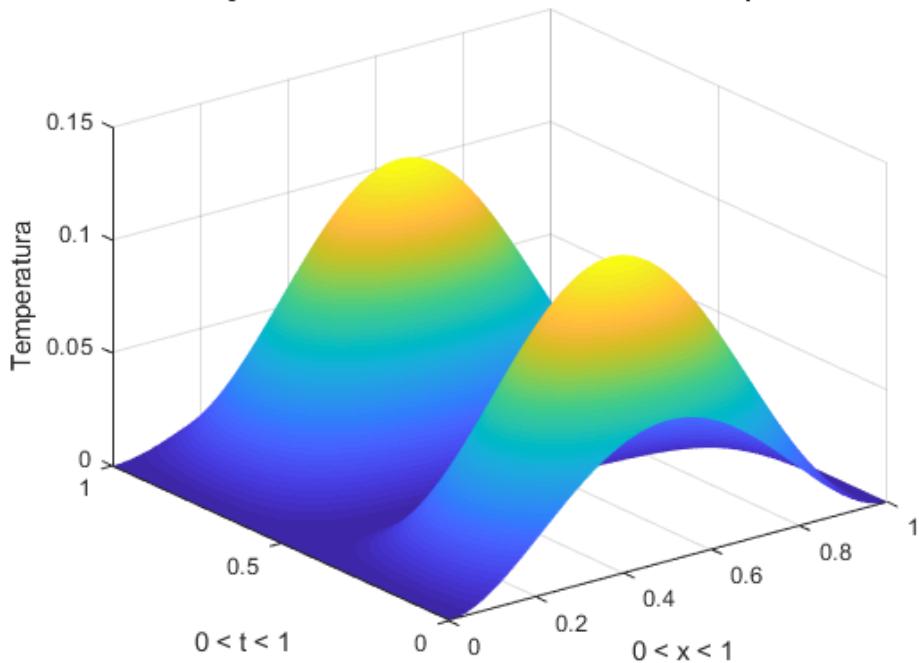


Figura 110: Curva da temperatura calculada com  $N = M = 160$

**Erro do item 1A com método de Euler implícito**

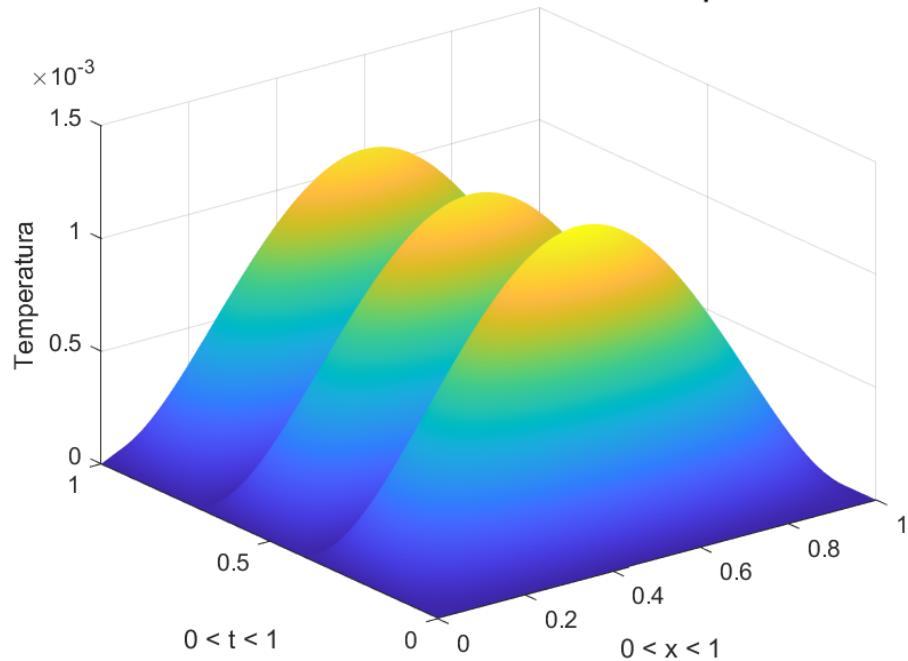


Figura 111: Erro com  $N = M = 160$

**Função u do item 1A com método de Euler implícito**

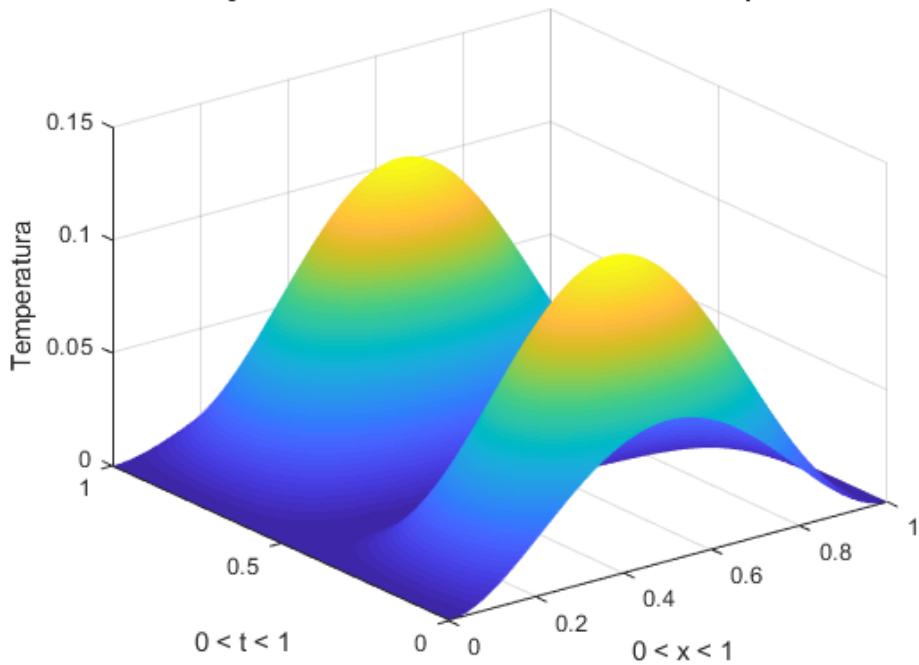


Figura 112: Curva da temperatura calculada com  $N = M = 320$

**Erro do item 1A com método de Euler implícito**

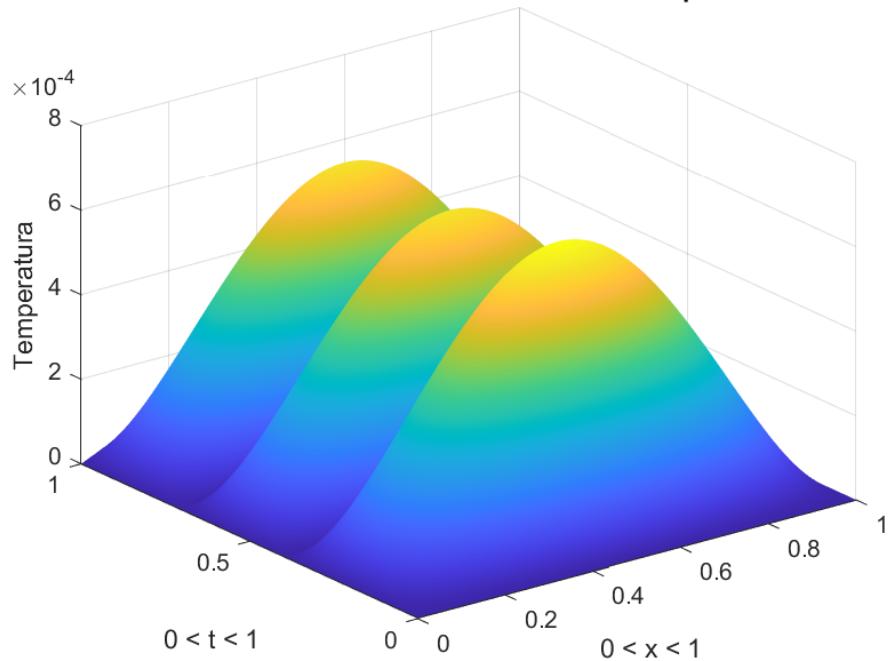


Figura 113: Erro com  $N = M = 320$

**Função u do item 1B com método de Euler implícito**

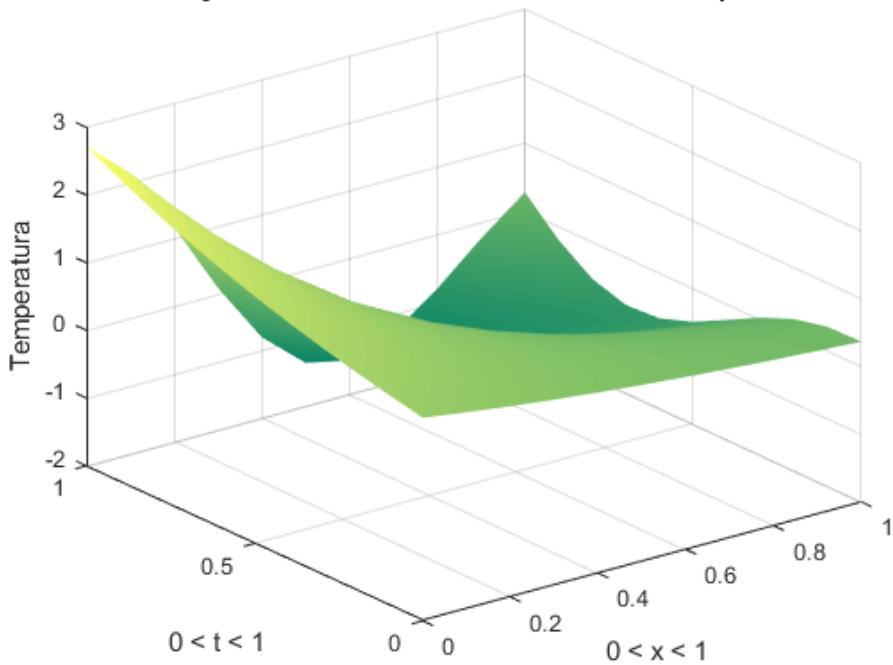


Figura 114: Curva da temperatura calculada com  $N = M = 10$

**Erro do item 1B com método de Euler implícito**

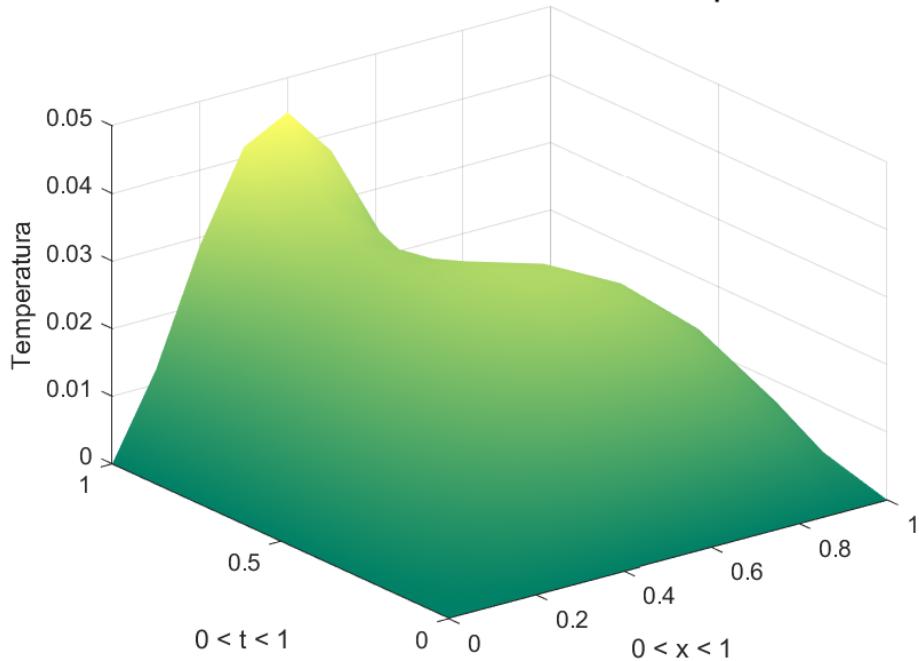


Figura 115: Erro com  $N = M = 10$

**Função u do item 1B com método de Euler implícito**

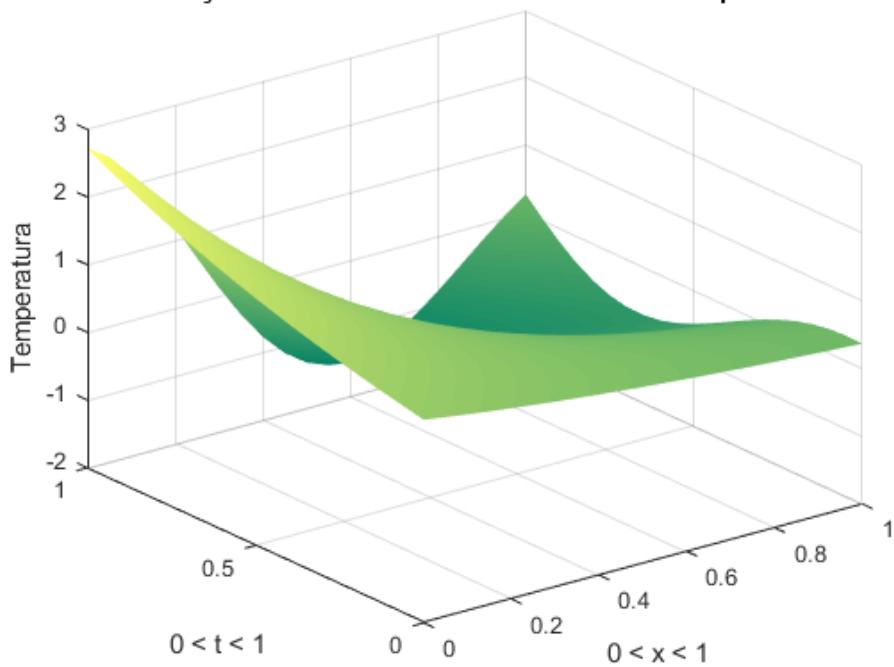


Figura 116: Curva da temperatura calculada com  $N = M = 20$

**Erro do item 1B com método de Euler implícito**

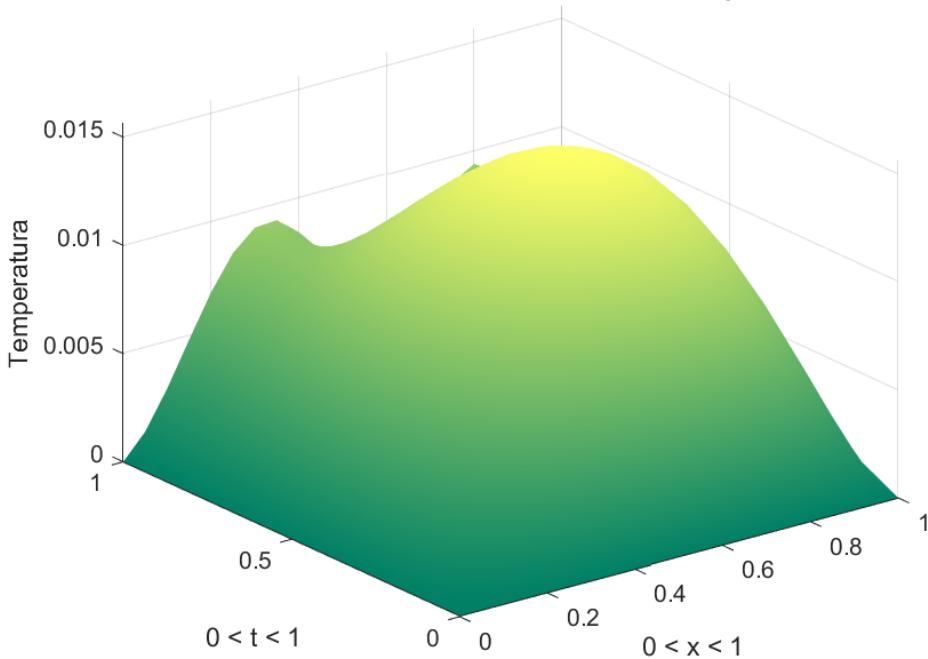


Figura 117: Erro com  $N = M = 20$

**Função u do item 1B com método de Euler implícito**

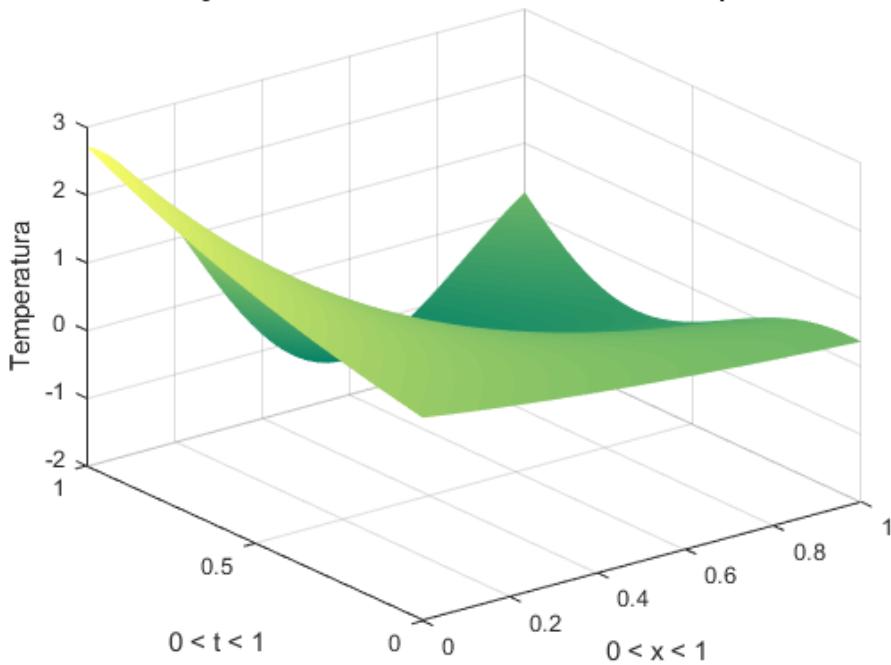


Figura 118: Curva da temperatura calculada com  $N = M = 40$

**Erro do item 1B com método de Euler implícito**

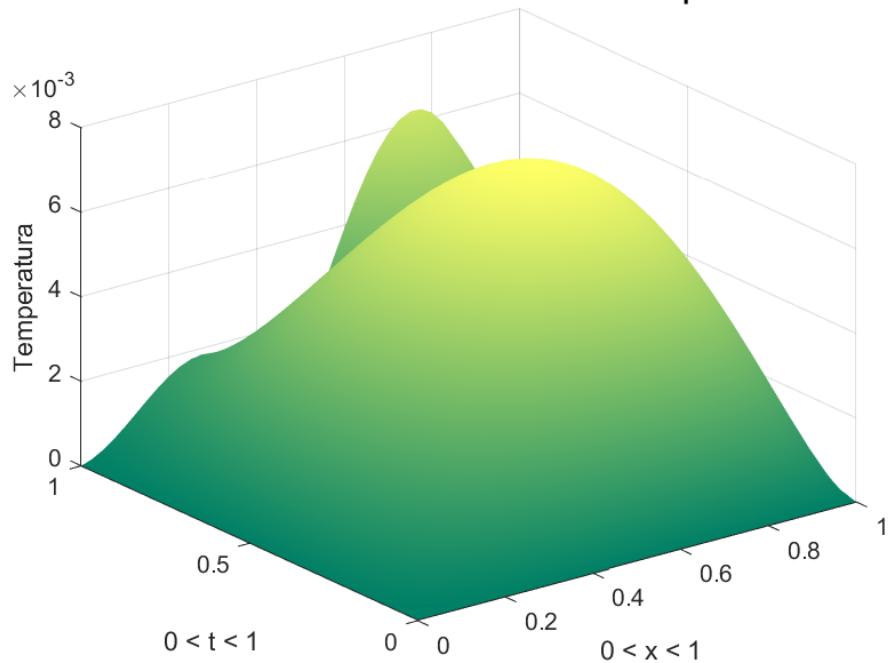


Figura 119: Erro com  $N = M = 40$

**Função u do item 1B com método de Euler implícito**

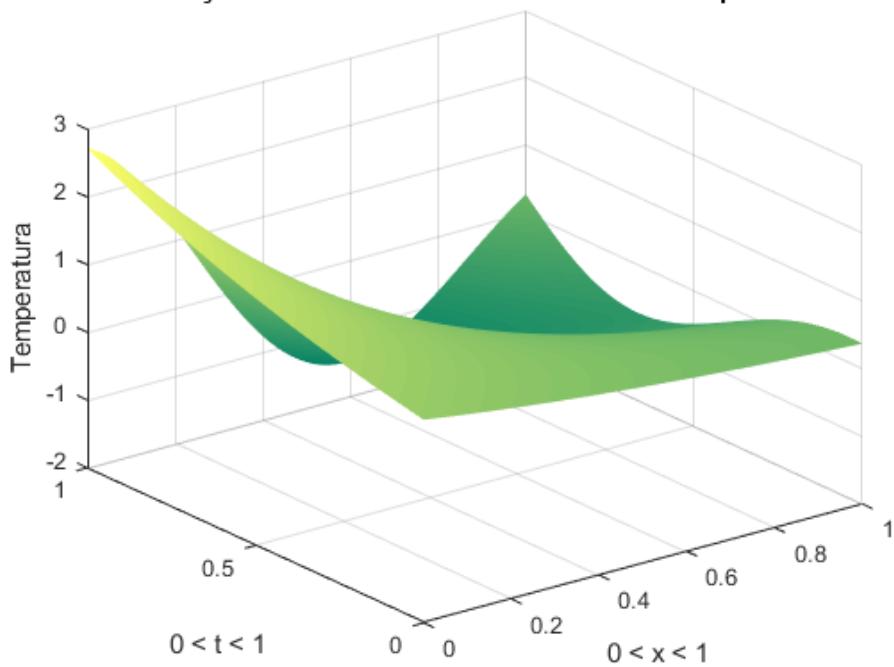


Figura 120: Curva da temperatura calculada com  $N = M = 80$

**Erro do item 1B com método de Euler implícito**

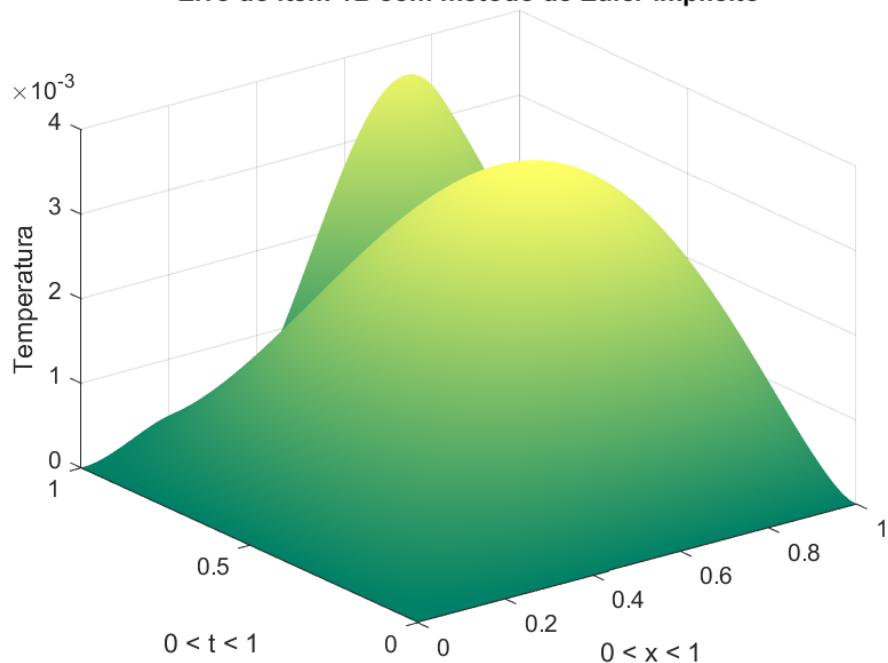


Figura 121: Erro com  $N = M = 80$

**Função u do item 1B com método de Euler implícito**

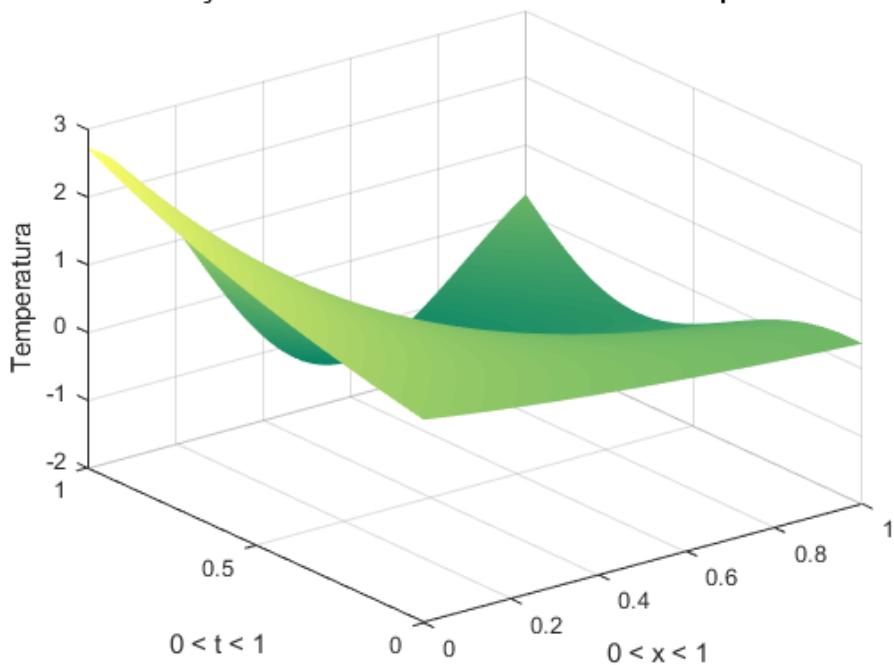


Figura 122: Curva da temperatura calculada com  $N = M = 160$

**Erro do item 1B com método de Euler implícito**

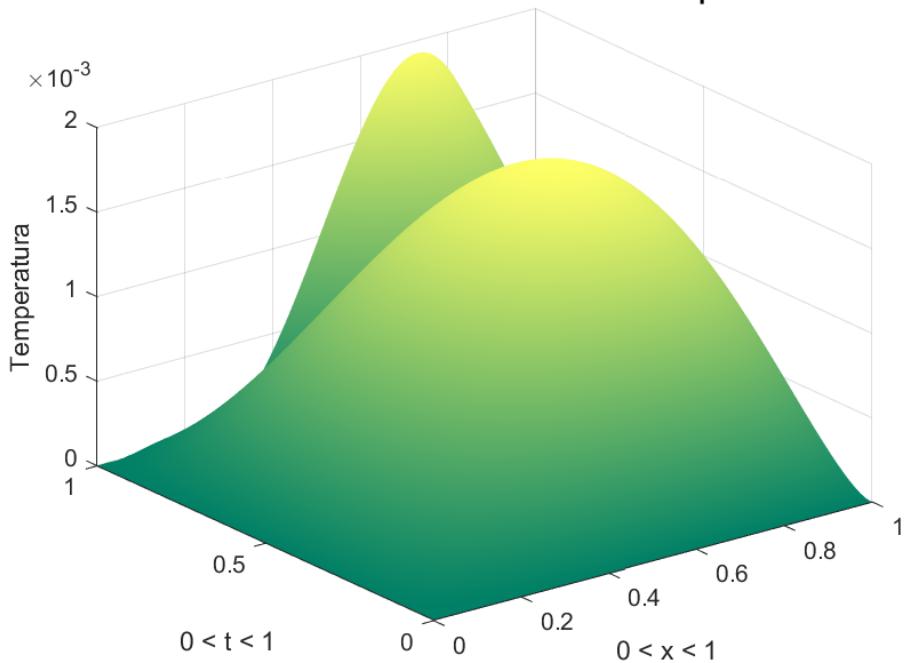


Figura 123: Erro com  $N = M = 160$

**Função u do item 1B com método de Euler implícito**

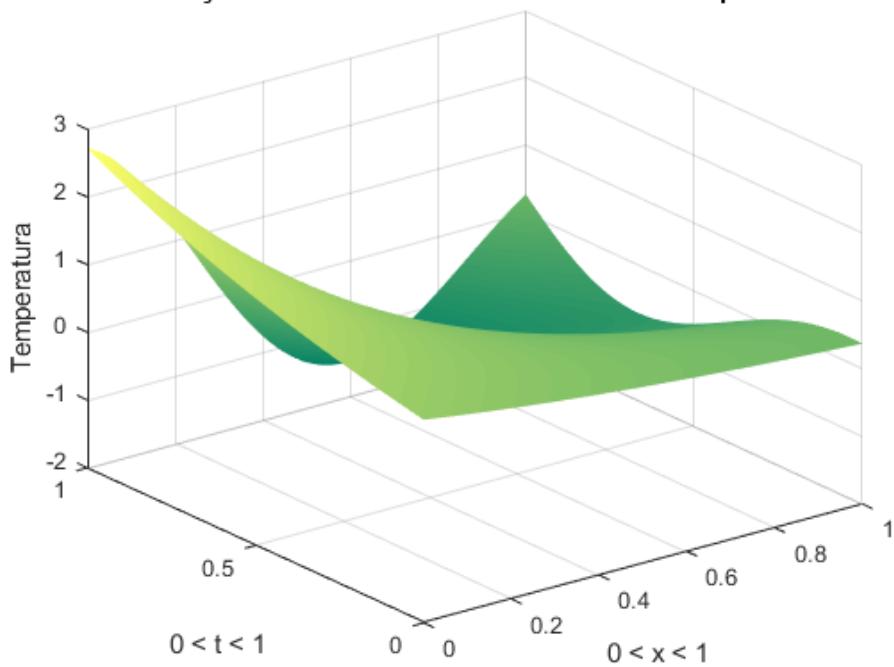


Figura 124: Curva da temperatura calculada com  $N = M = 320$

**Erro do item 1B com método de Euler implícito**

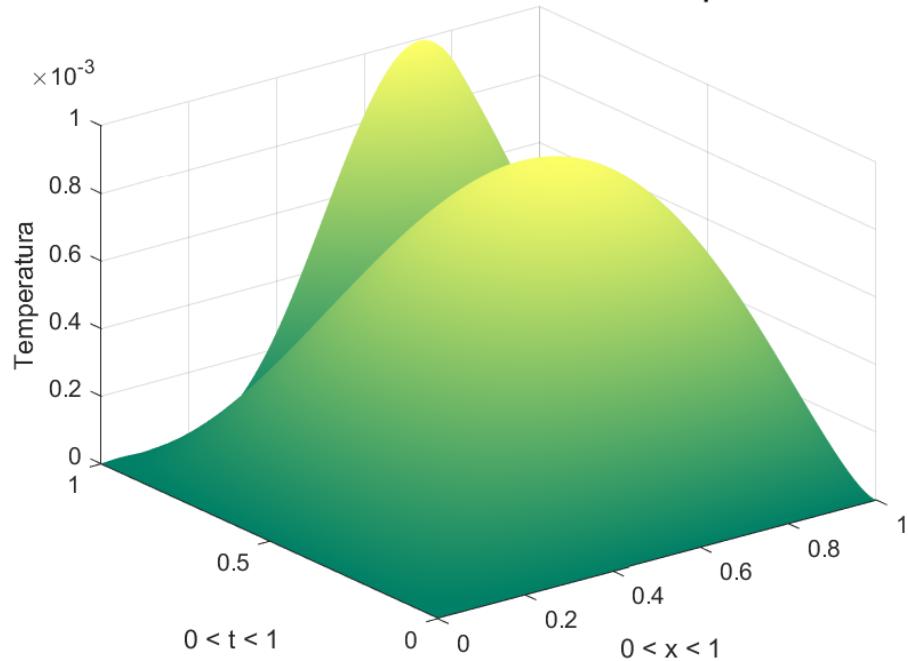


Figura 125: Erro com  $N = M = 320$

**Função u do item 1C com método de Euler implícito**

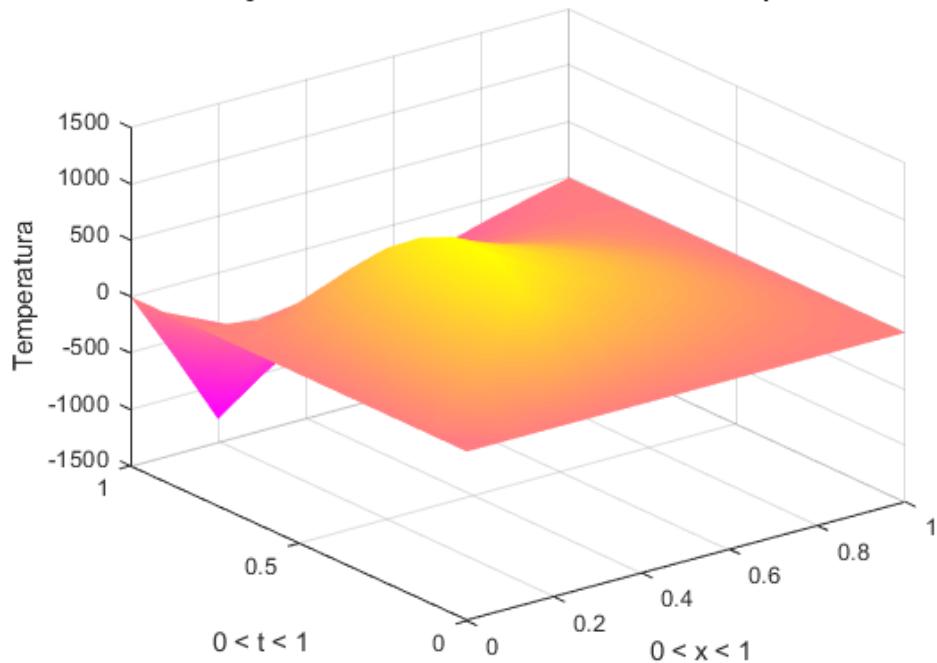


Figura 126: Curva da temperatura calculada com  $N = M = 10$

**Função u do item 1C com método de Euler implícito**

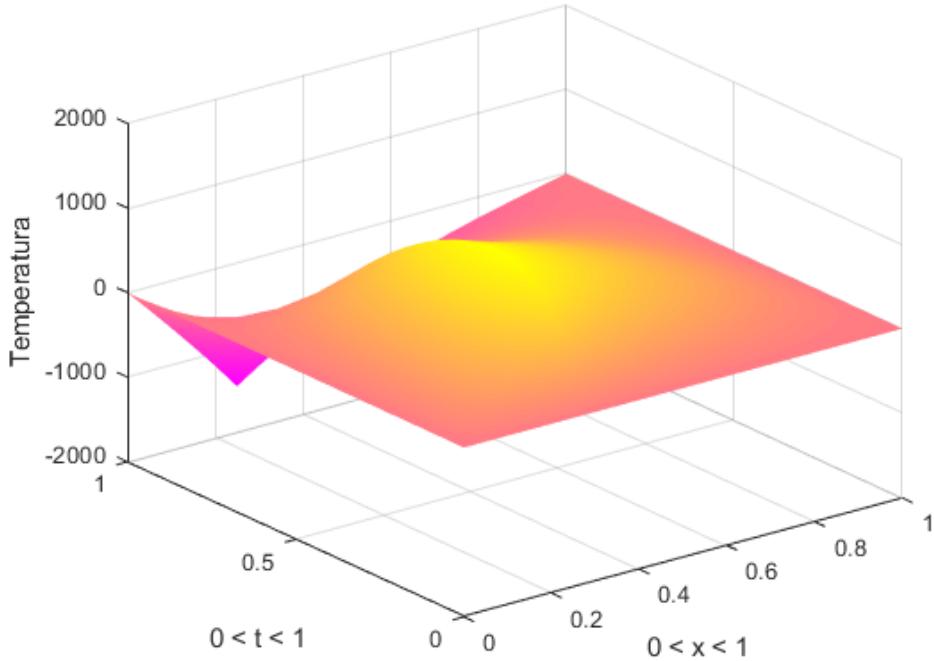


Figura 127: Curva da temperatura calculada com  $N = M = 20$

**Função u do item 1C com método de Euler implícito**

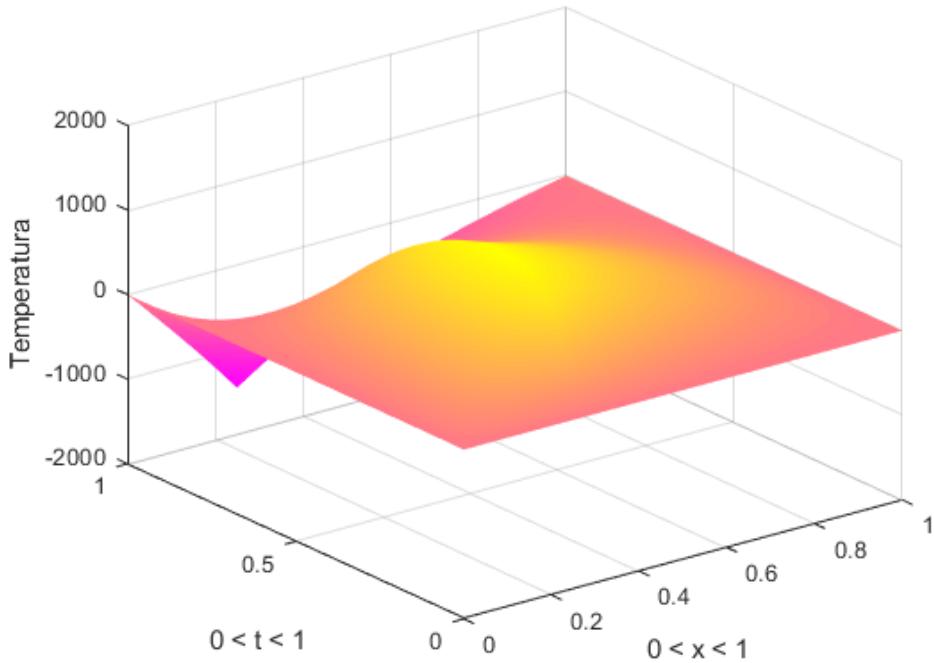


Figura 128: Curva da temperatura calculada com  $N = M = 40$

**Função u do item 1C com método de Euler implícito**

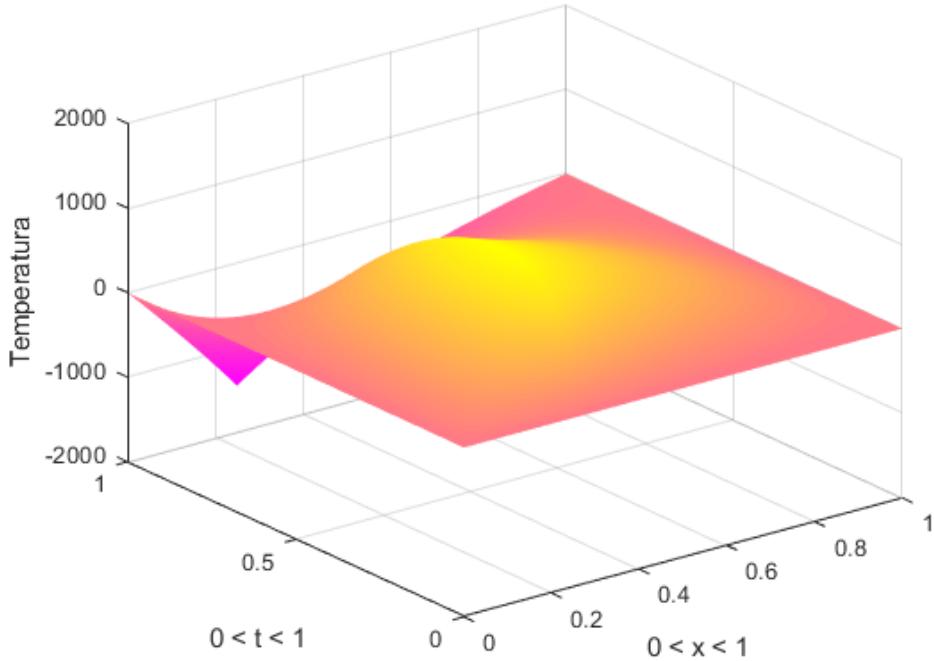


Figura 129: Curva da temperatura calculada com  $N = M = 80$

**Função u do item 1C com método de Euler implícito**

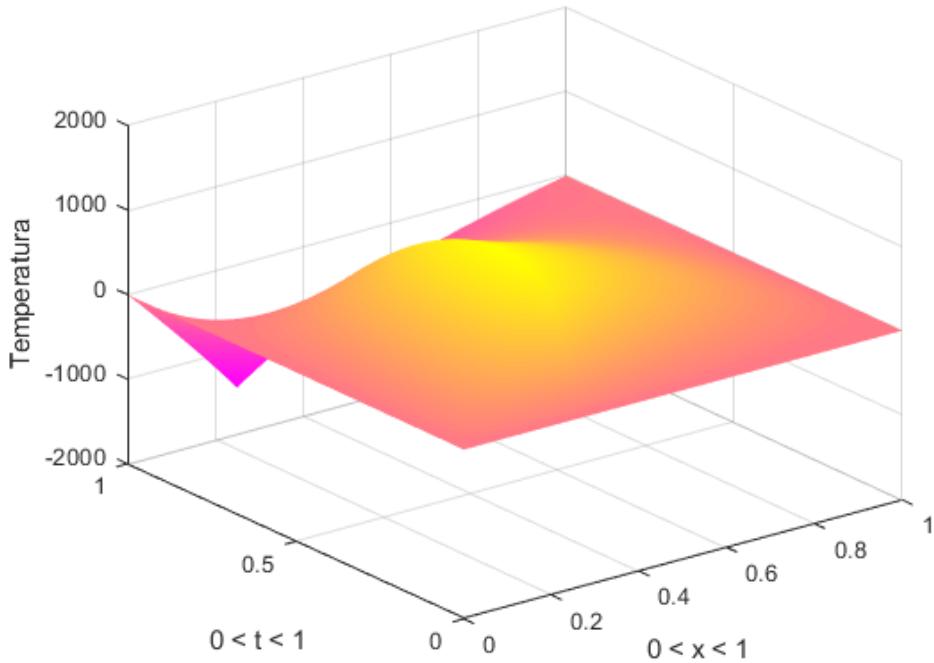


Figura 130: Curva da temperatura calculada com  $N = M = 160$

**Função u do item 1C com método de Euler implícito**

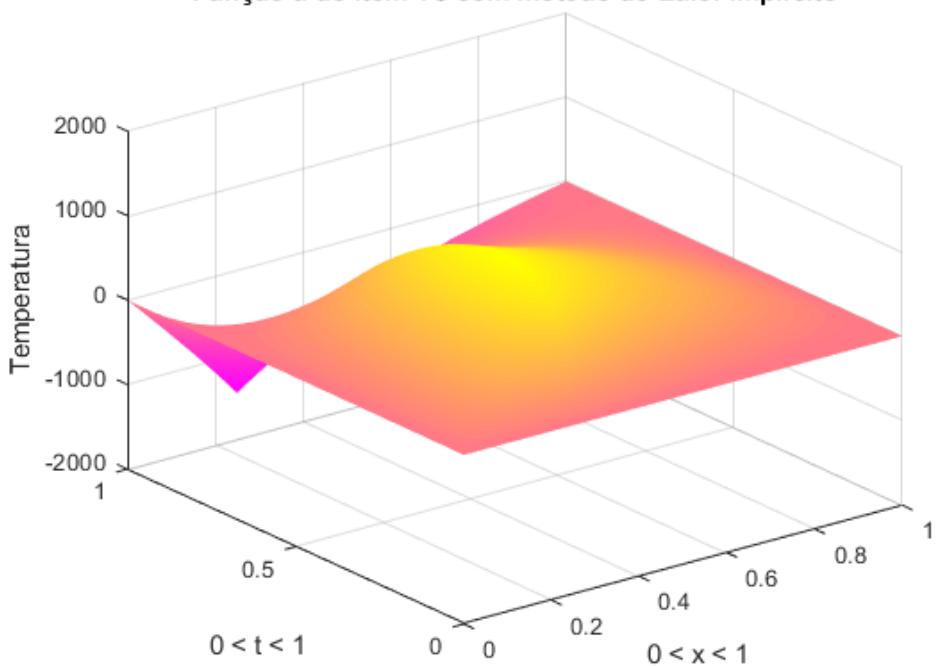


Figura 131: Curva da temperatura calculada com  $N = M = 320$

## G Curvas do item 2C

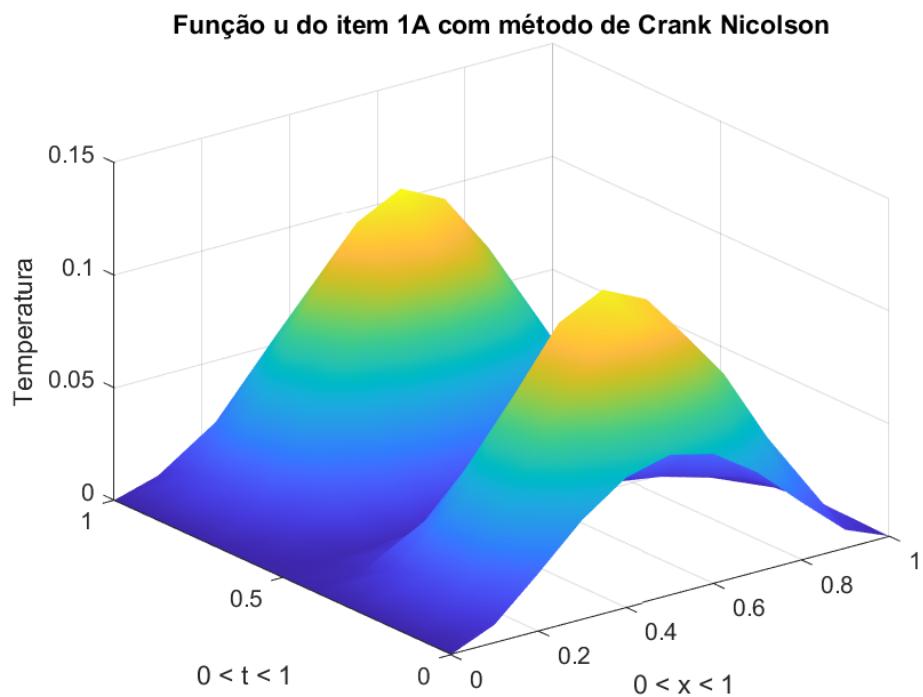


Figura 132: Curva da temperatura calculada com  $N = M = 10$

**Erro do item 1A com método de Crank Nicolson**

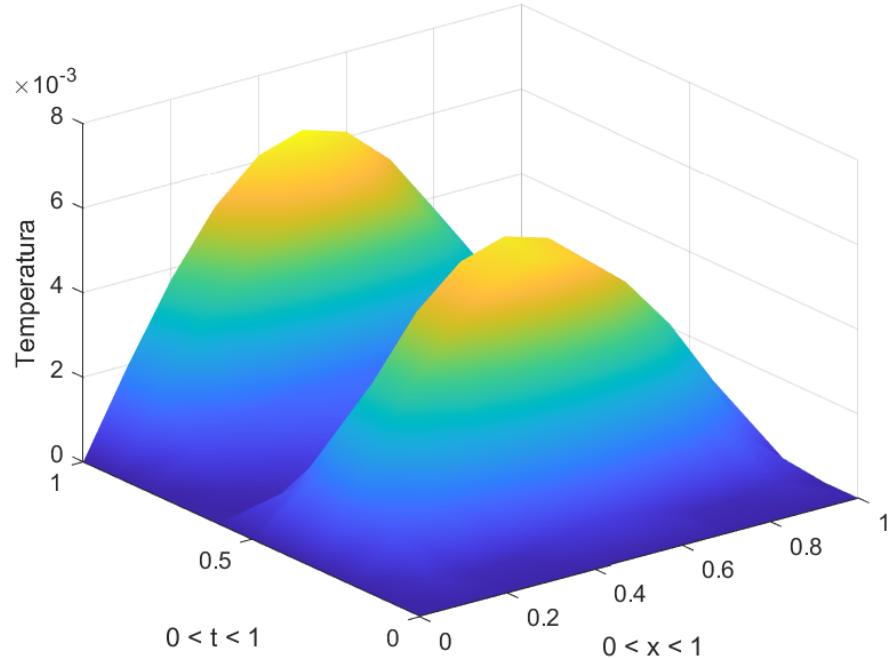


Figura 133: Erro com  $N = M = 10$

**Função u do item 1A com método de Crank Nicolson**

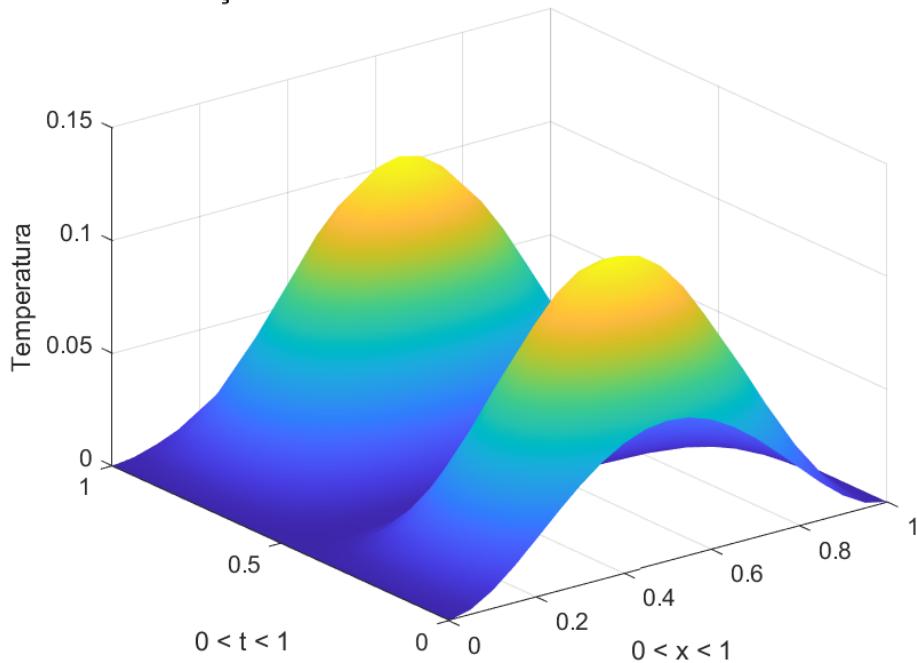


Figura 134: Curva da temperatura calculada com  $N = M = 20$

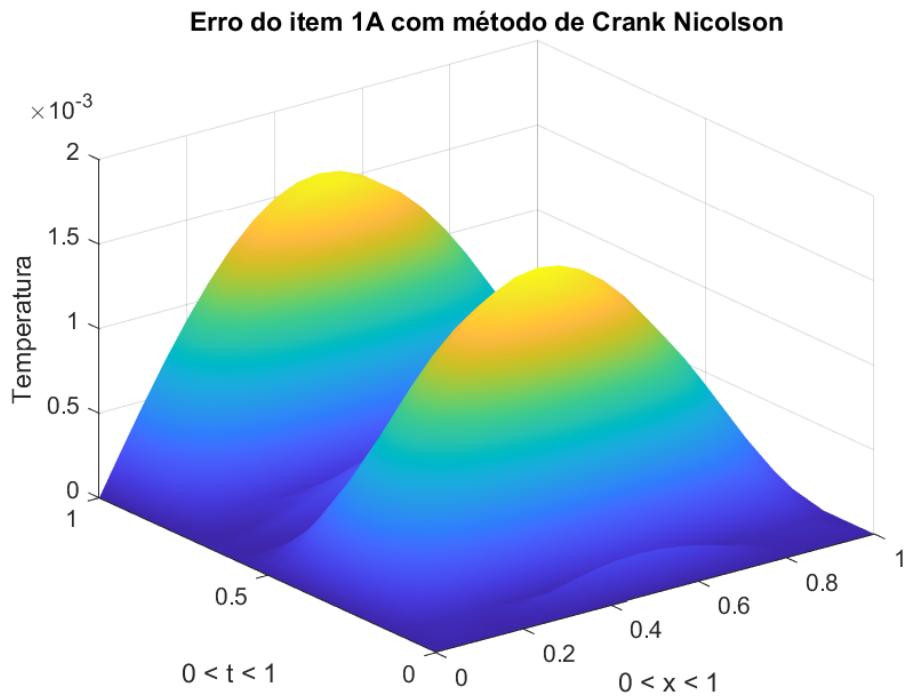


Figura 135: Erro com  $N = M = 20$

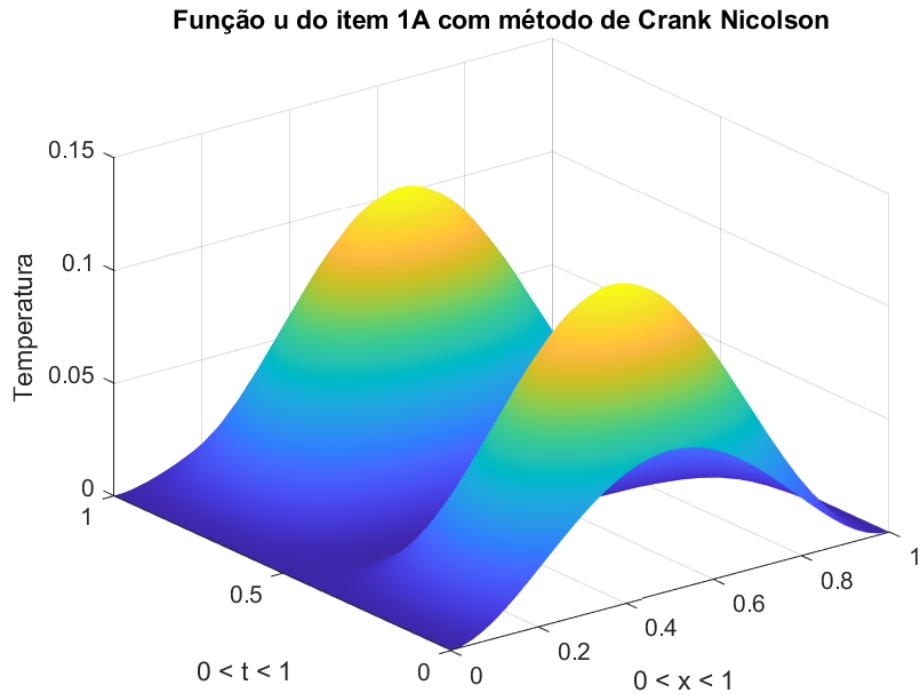


Figura 136: Curva da temperatura calculada com  $N = M = 40$

**Erro do item 1A com método de Crank Nicolson**

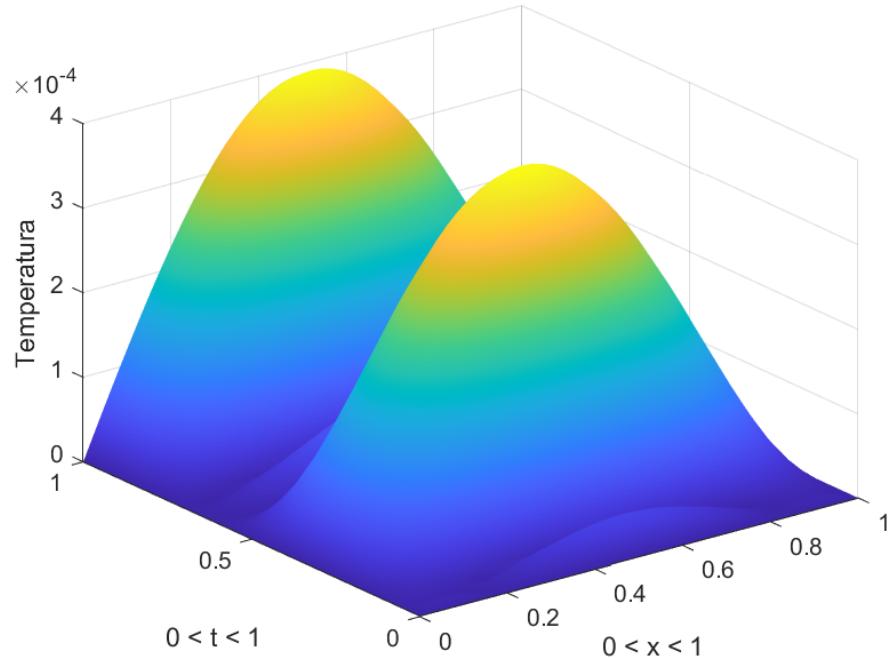


Figura 137: Erro com  $N = M = 40$

**Função u do item 1A com método de Crank Nicolson**

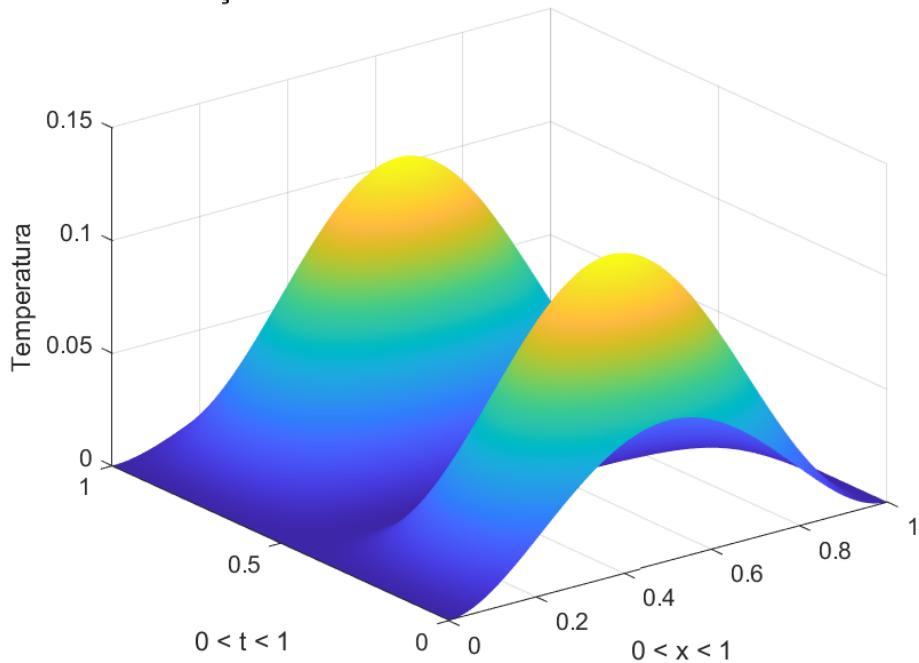


Figura 138: Curva da temperatura calculada com  $N = M = 80$

**Erro do item 1A com método de Crank Nicolson**

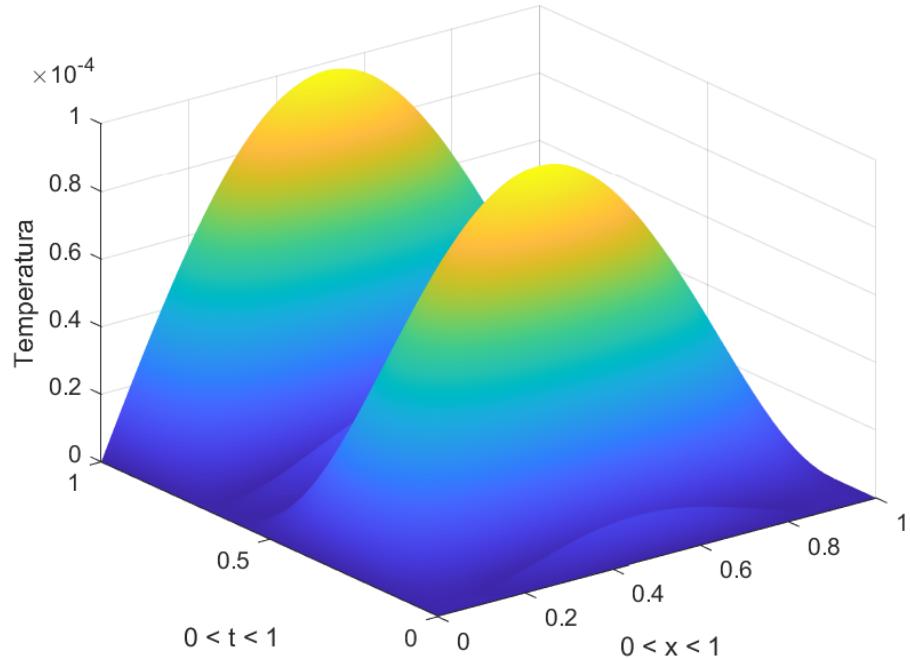


Figura 139: Erro com  $N = M = 80$

**Função u do item 1A com método de Crank Nicolson**

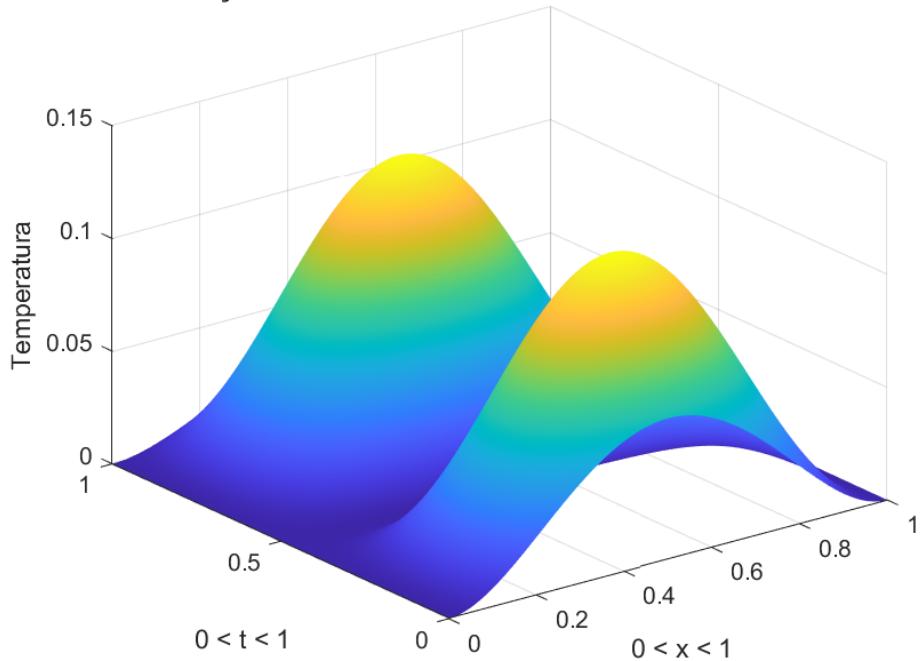


Figura 140: Curva da temperatura calculada com  $N = M = 160$

**Erro do item 1A com método de Crank Nicolson**

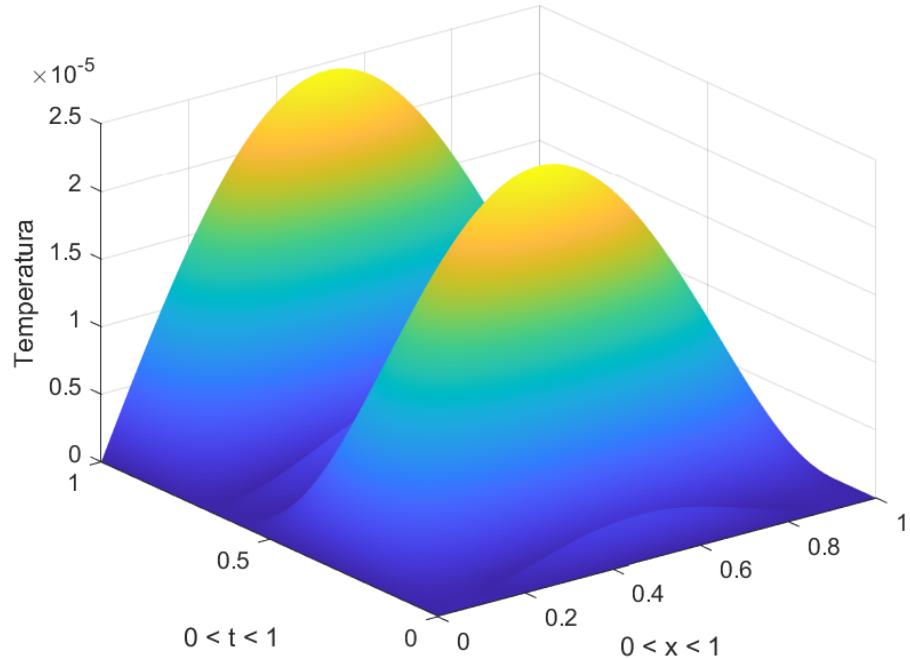


Figura 141: Erro com  $N = M = 160$

**Função u do item 1A com método de Crank Nicolson**

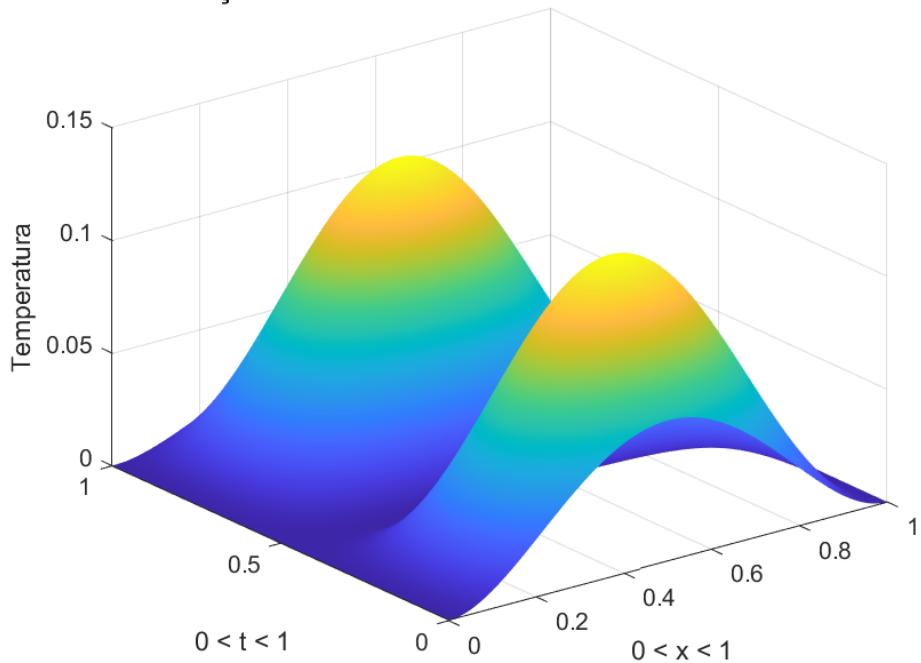


Figura 142: Curva da temperatura calculada com  $N = M = 320$

**Erro do item 1A com método de Crank Nicolson**

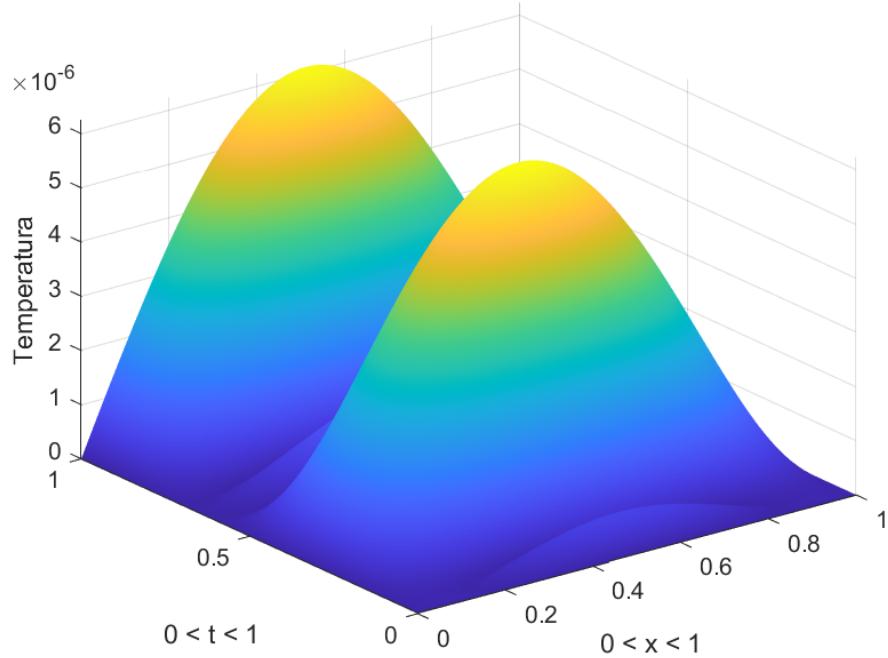


Figura 143: Erro com  $N = M = 320$

**Função u do item 1B com método de Crank Nicolson**

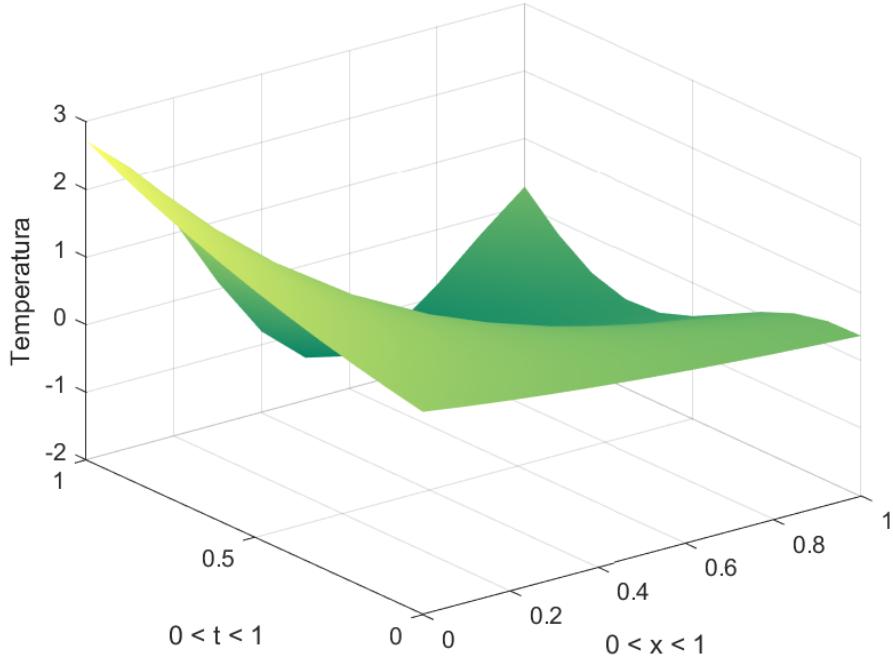


Figura 144: Curva da temperatura calculada com  $N = M = 10$

**Erro do item 1B com método de Crank Nicolson**

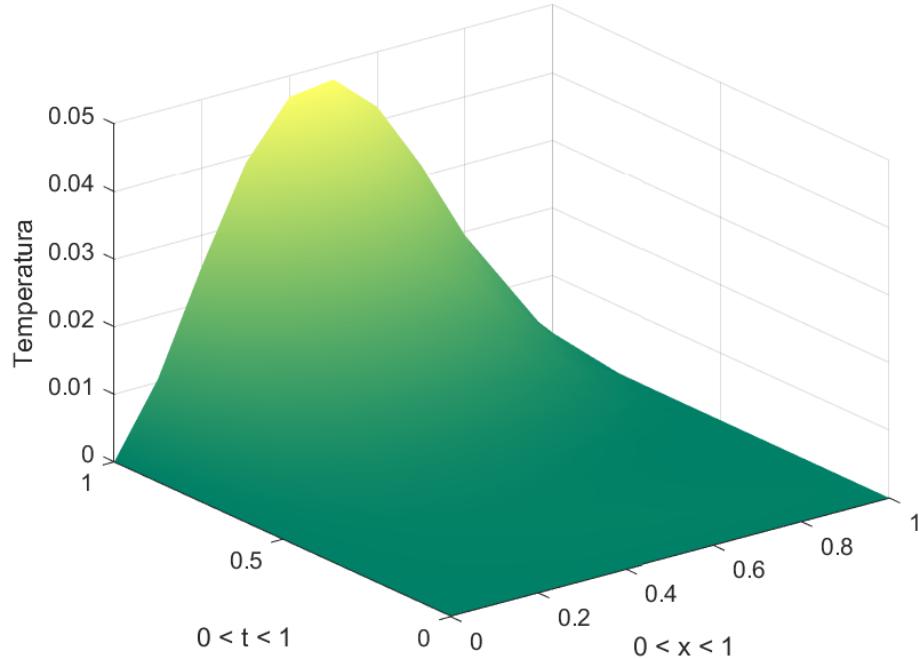


Figura 145: Erro com  $N = M = 10$

**Função u do item 1B com método de Crank Nicolson**

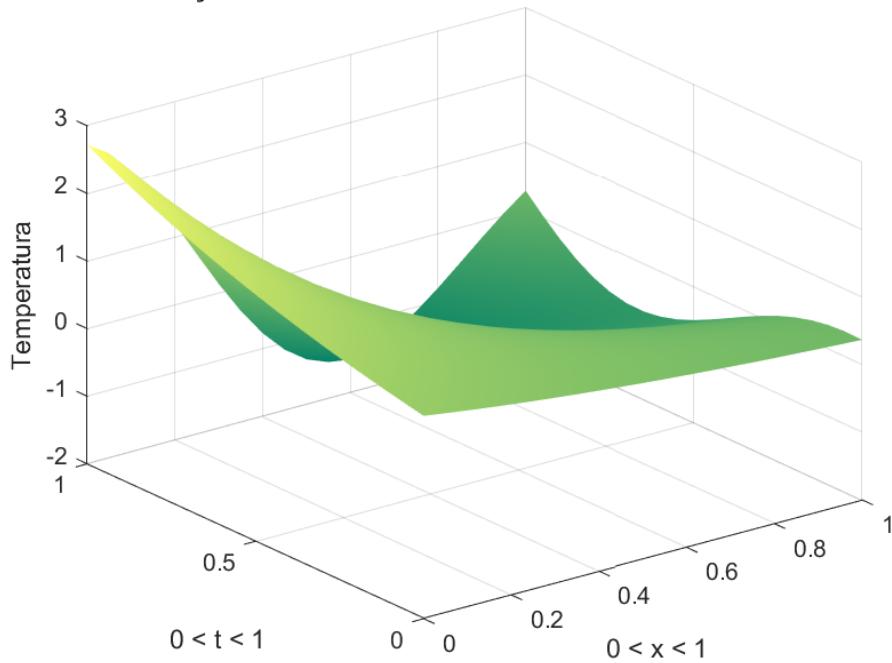


Figura 146: Curva da temperatura calculada com  $N = M = 20$

**Erro do item 1B com método de Crank Nicolson**

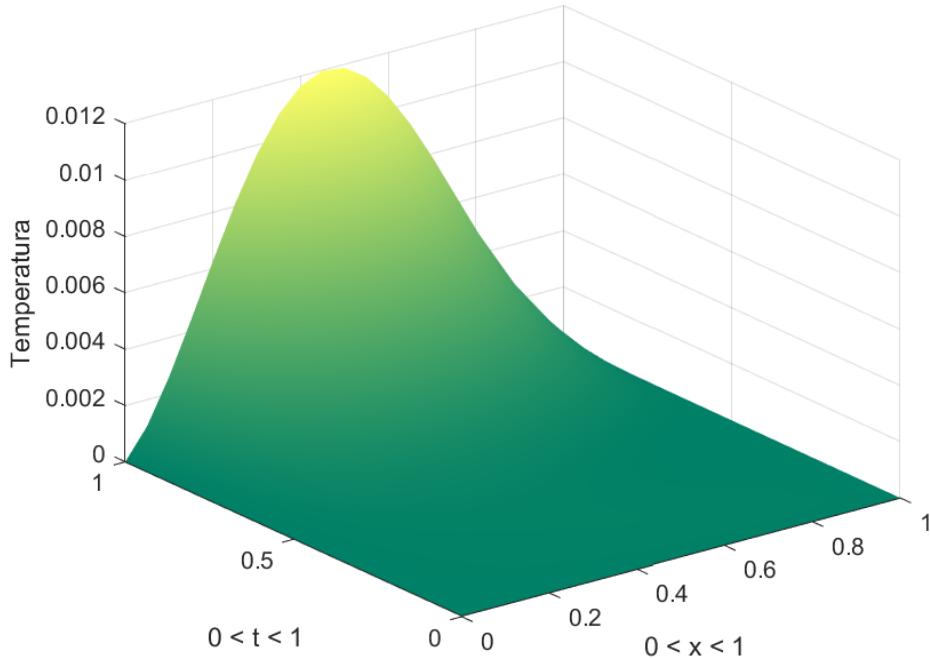


Figura 147: Erro com  $N = M = 20$

**Função u do item 1B com método de Crank Nicolson**

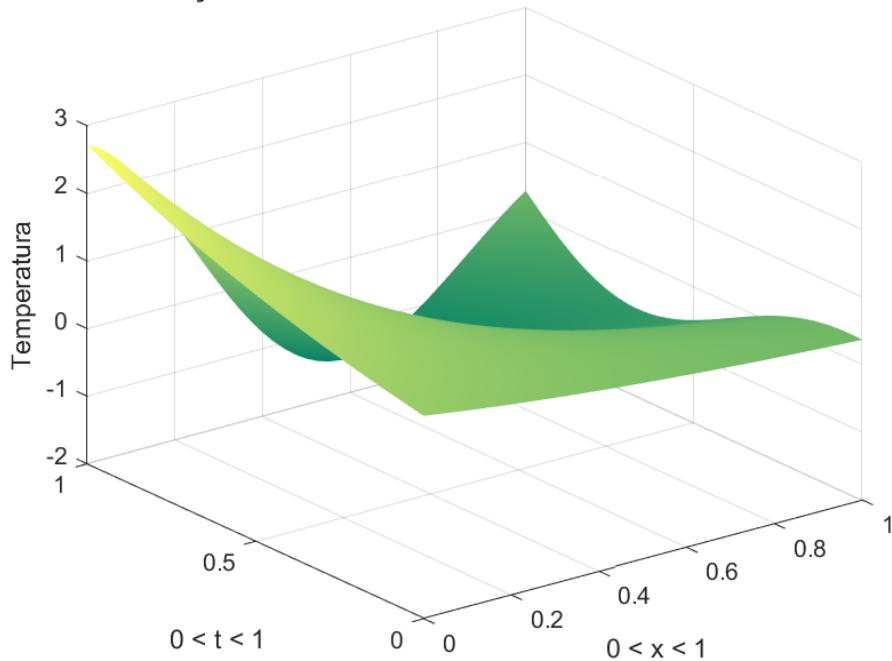


Figura 148: Curva da temperatura calculada com  $N = M = 40$

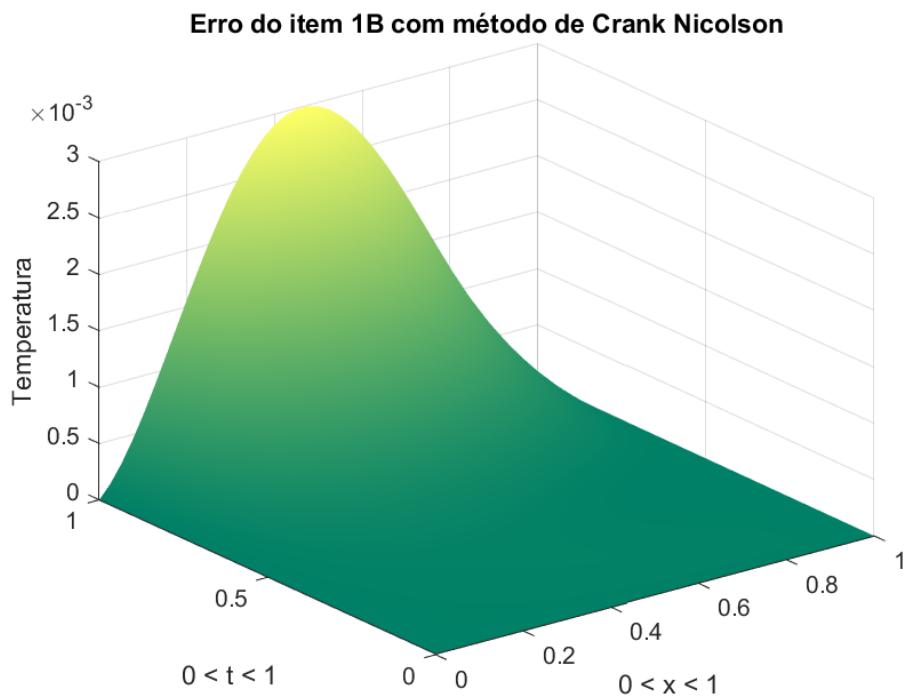


Figura 149: Erro com  $N = M = 40$

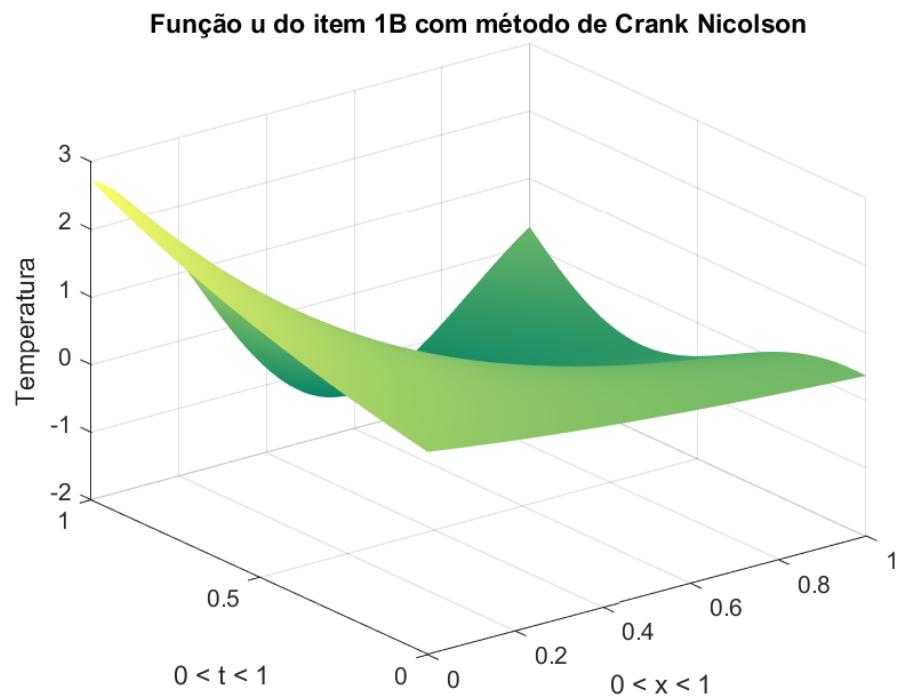


Figura 150: Curva da temperatura calculada com  $N = M = 80$

**Erro do item 1B com método de Crank Nicolson**

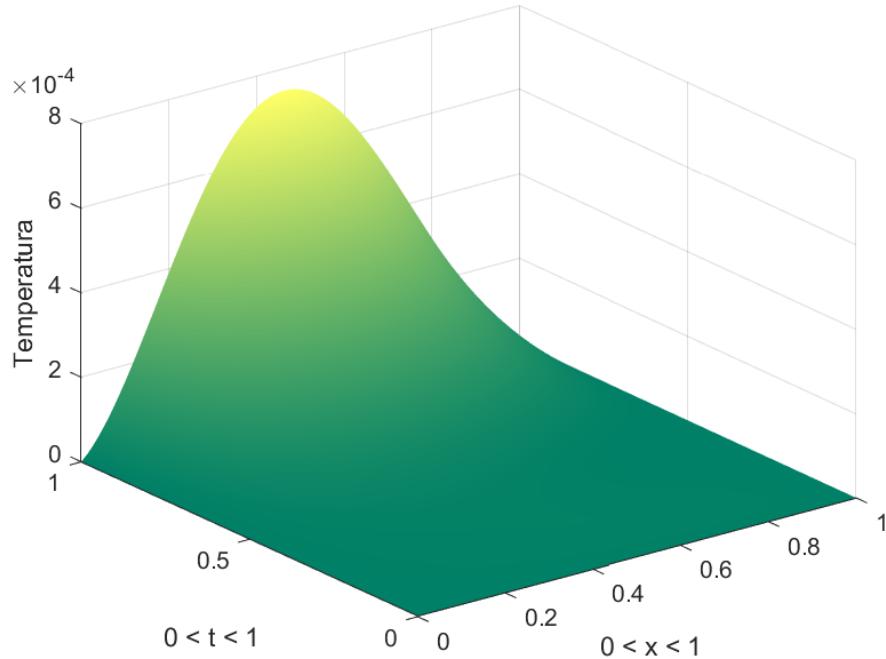


Figura 151: Erro com  $N = M = 80$

**Função u do item 1B com método de Crank Nicolson**

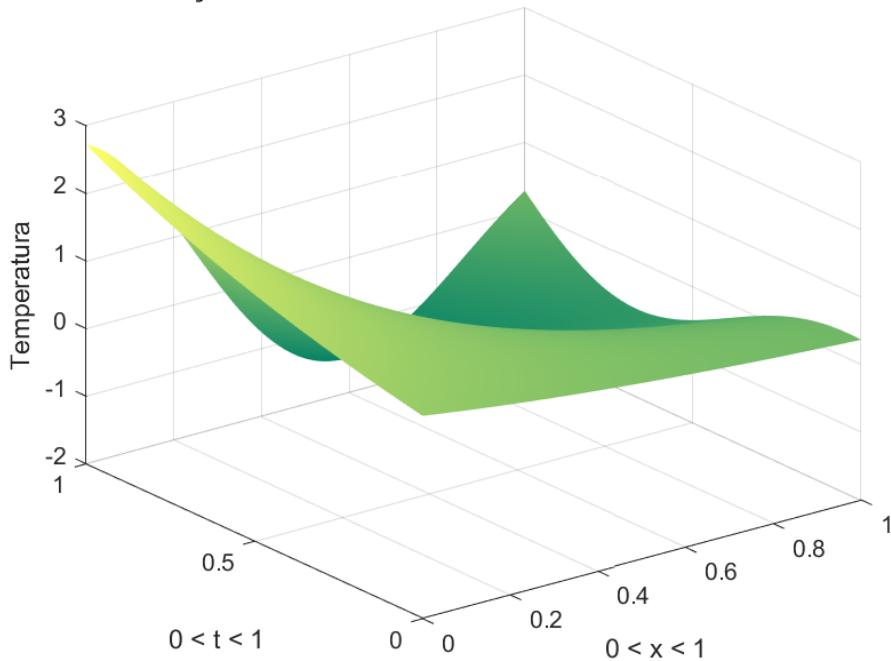


Figura 152: Curva da temperatura calculada com  $N = M = 160$

**Erro do item 1B com método de Crank Nicolson**

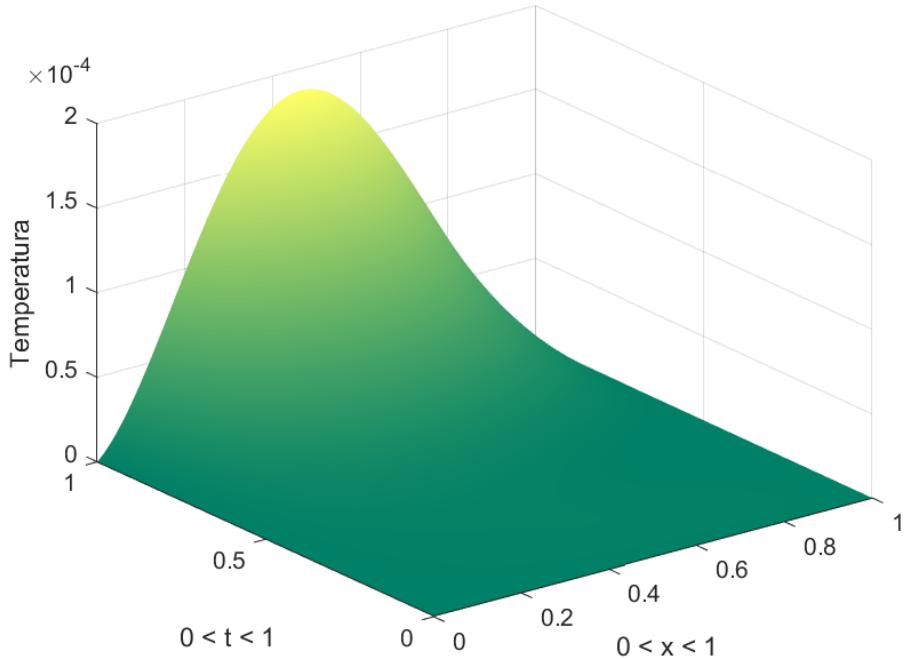


Figura 153: Erro com  $N = M = 160$

**Função u do item 1B com método de Crank Nicolson**

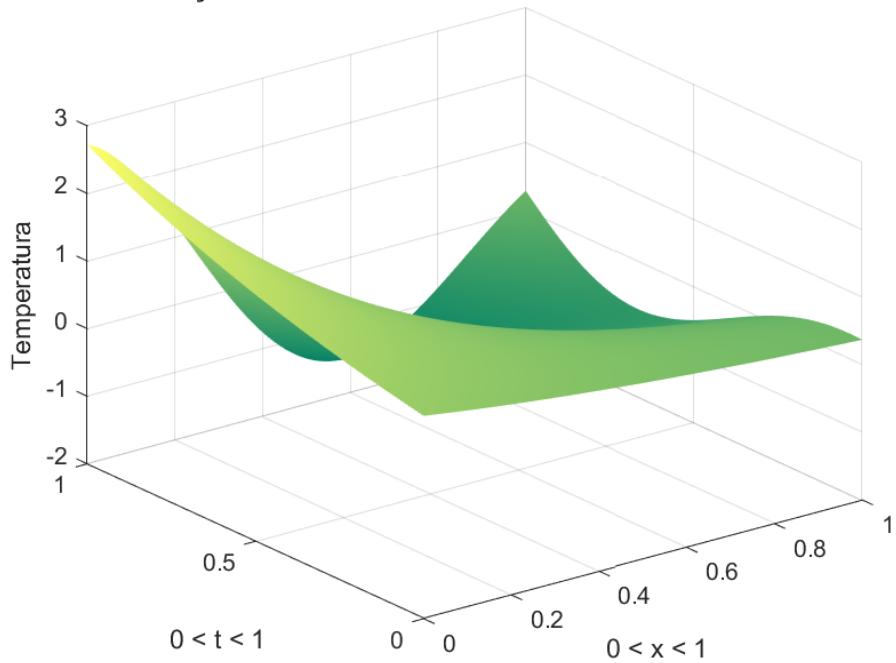


Figura 154: Curva da temperatura calculada com  $N = M = 320$

**Erro do item 1B com método de Crank Nicolson**

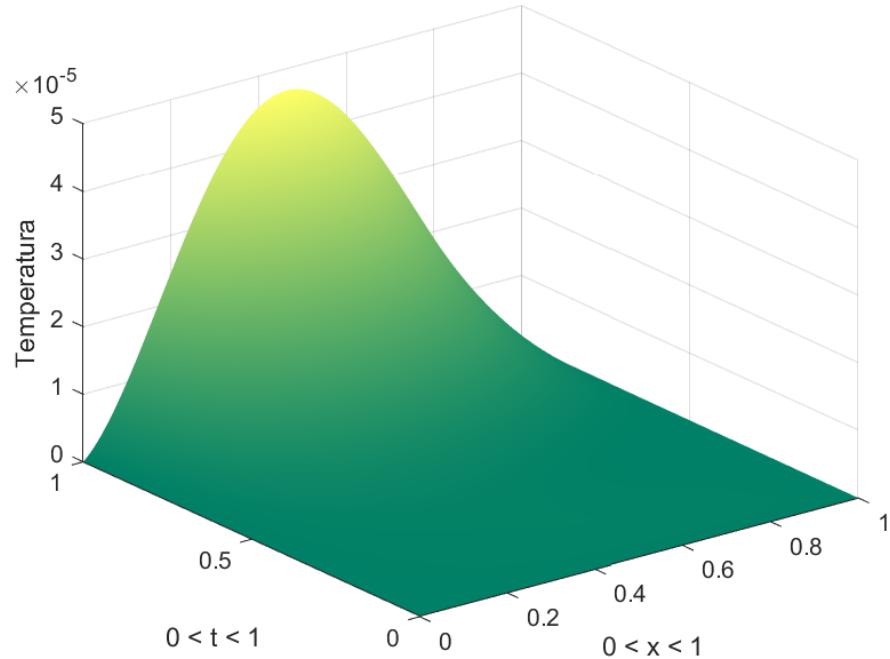


Figura 155: Erro com  $N = M = 320$

**Função u do item 1C com método de Crank Nicolson**

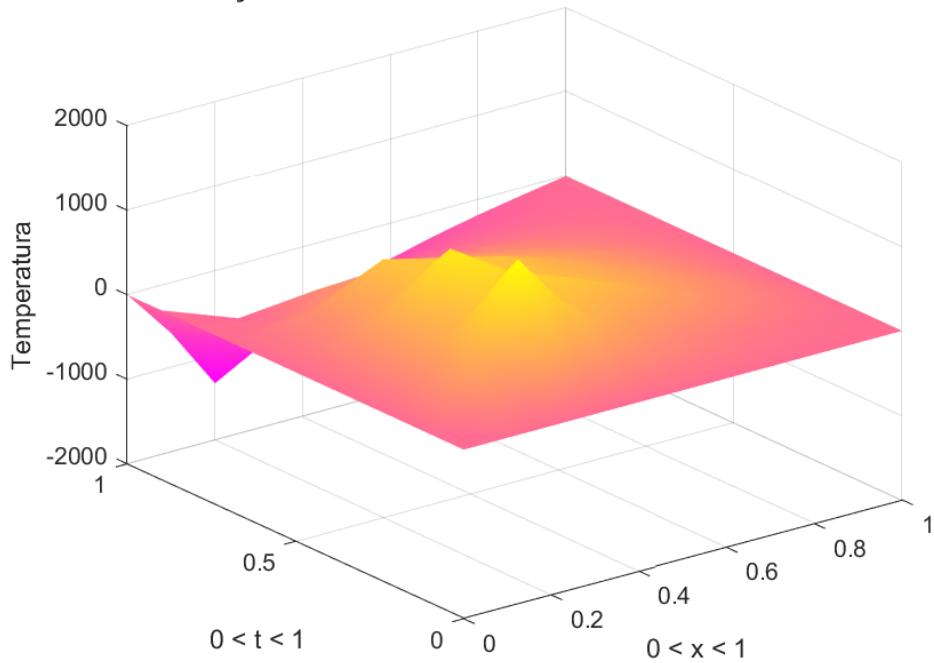


Figura 156: Curva da temperatura calculada com  $N = M = 10$

**Função u do item 1C com método de Crank Nicolson**

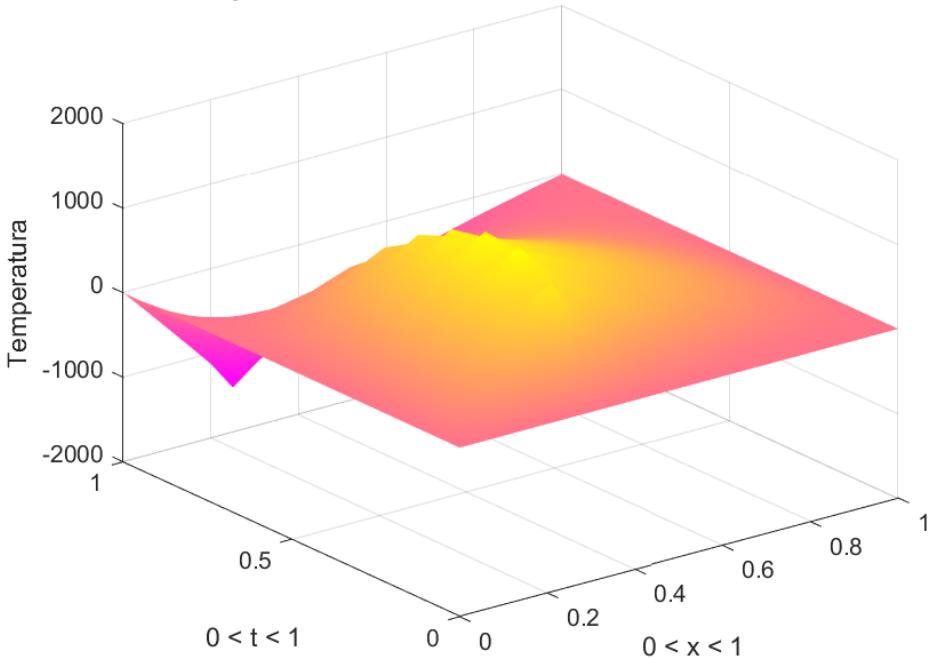


Figura 157: Curva da temperatura calculada com  $N = M = 20$

**Função u do item 1C com método de Crank Nicolson**

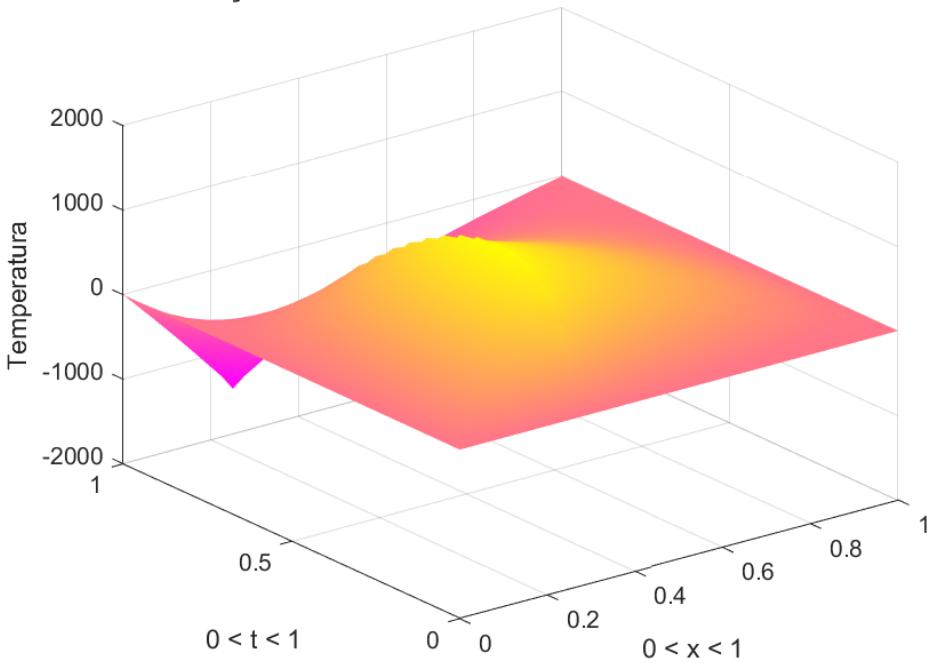


Figura 158: Curva da temperatura calculada com  $N = M = 40$

**Função u do item 1C com método de Crank Nicolson**

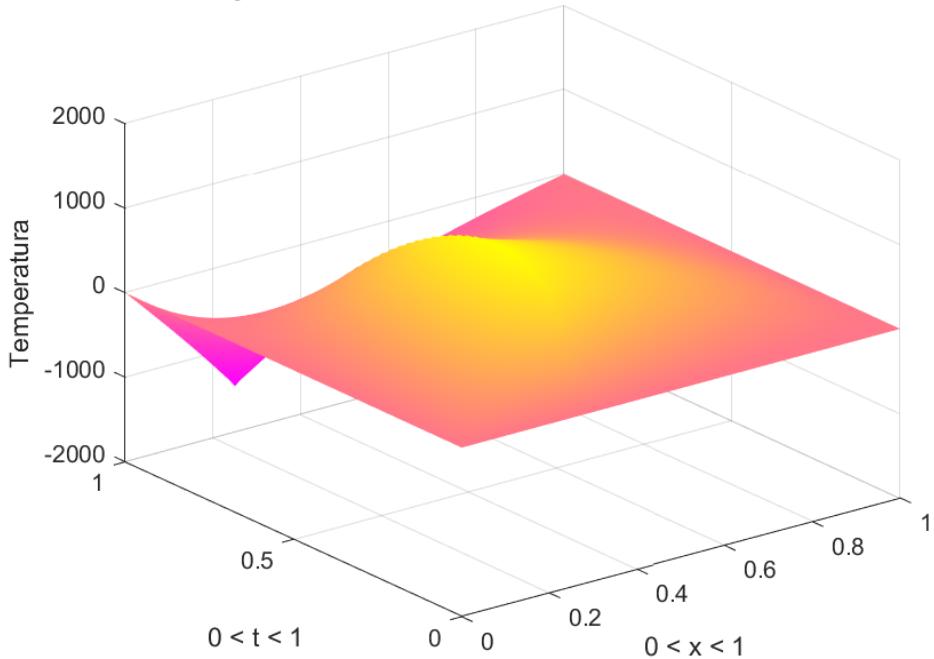


Figura 159: Curva da temperatura calculada com  $N = M = 80$

**Função u do item 1C com método de Crank Nicolson**

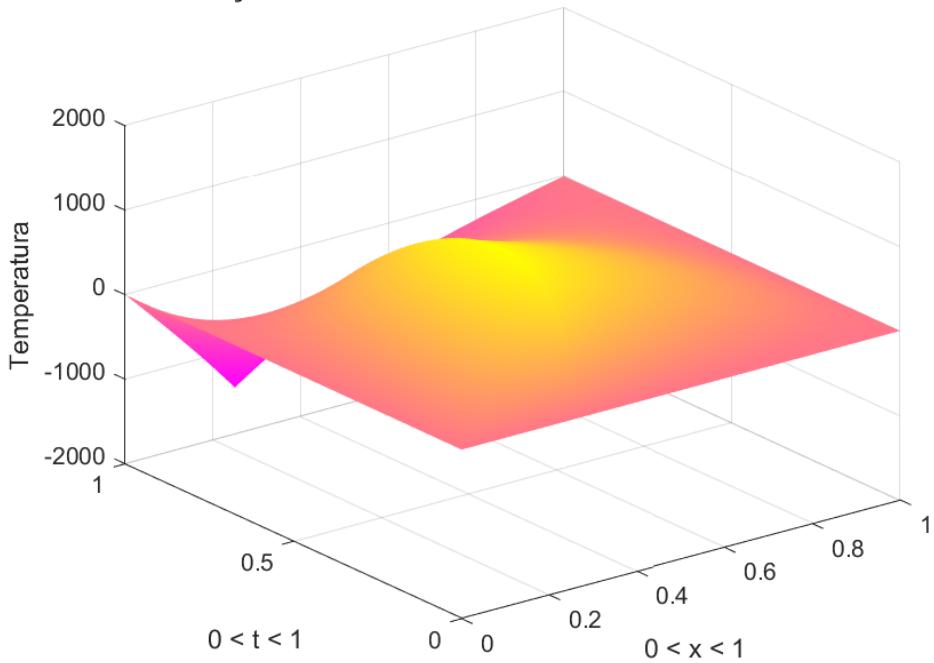


Figura 160: Curva da temperatura calculada com  $N = M = 160$

**Função u do item 1C com método de Crank Nicolson**

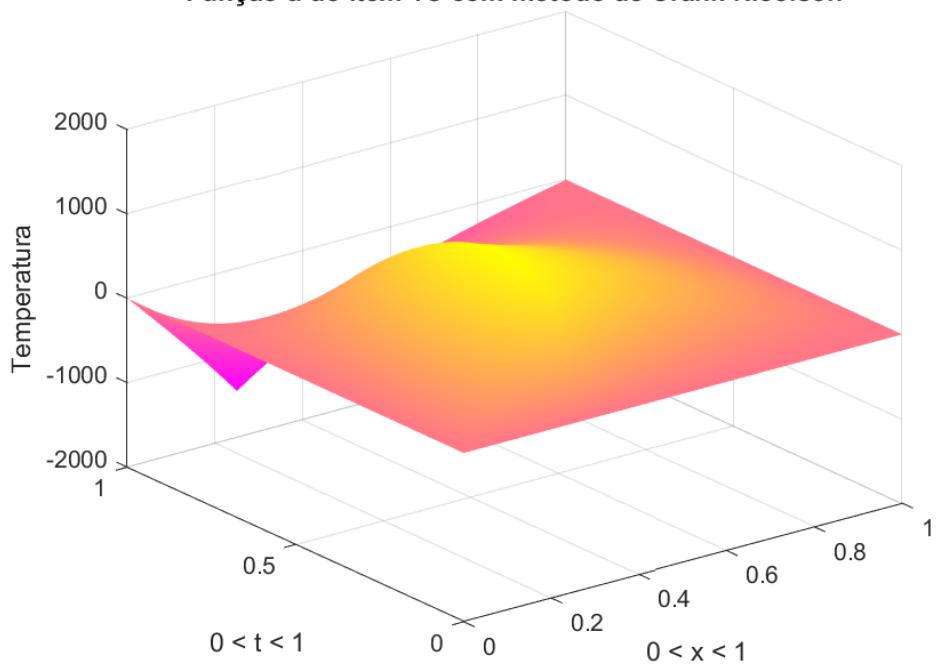


Figura 161: Curva da temperatura calculada com  $N = M = 320$