Métodos Numéricos — Lista 01

André Paladini 14182390
Tiago F. Oliva Costa 8004408

16 de outubro de 2023

1 Questão 01

Classifique as EDPs abaixo quanto à ordem, a linearidade / não-linearidade, a homogeneidade e ao tipo.

A 2a ordem; Linear; Homogênea; Hiperbólica.

B 2a ordem; Linear; Não-Homogênea; Parabólica.

C 1a ordem; Linear; Homogênea; Parabólica.

D 2a ordem; Linear; Homogênea; Elíptica.

E 2a ordem; Não-Linear; Homogênea; Parabólica.

F 2a ordem; Não-Linear; Homogênea; Parabólica.

2 Questão 02

Qual a diferença entre as condições de contorno de Dirichlet, Neumann e Robin?

A condição de contorno de Dirichlet (ou primeiro tipo) especifica valores que a variável dependente $y(x)$ toma ao longo da fronteira do domínio. Ou seja

$$y(a) = \alpha, \quad y(b) = \beta.$$

A condição de contorno de Neumann (ou segundo tipo) especifica valores que a derivada $y'(x)$ da variável dependente toma ao longo da fronteira do domínio. Ou seja

$$y'(a) = \alpha, \quad y'(b) = \beta.$$

A condição de contorno de Robin (ou terceiro tipo) especifica valores que tanto a variável dependente $y(x)$, como a sua derivada $y'(x)$, tomam ao longo da fronteira do domínio. Ou seja, para um domínio Ω e sua fronteira representada por $\partial\Omega$, têm-se

$$ay + b\frac{\partial y}{\partial x} = g \qquad \text{em} \qquad \partial\Omega.$$

3 Suplemento

Para as questões 03 e 04, considere como
forward difference

$$D_+ f(x) = f(x+h) - f(x),$$

central difference

$$D_0 f(x) = f(x + \frac{h}{2}) - f(x - \frac{x}{2}),$$

e *backward difference*

$$D_- f(x) = f(x) - f(x-h).$$

A metodologia exposta na Seção 1.5 de LeVeque[1], é adotada para calcular os coeficientes gerais para diferenças finitas

$$\frac{1}{(i-1)!} \sum_{j=1}^{n} c_j (x_j - \bar{x})^{(i-1)} \begin{cases} 1 & \text{se } i-1 = k \\ 0 & \text{caso contrário} \end{cases} \tag{1}$$

implementada segundo o seguinte excerto de código em Python

```
"""
Find finite difference coefficients

k    - kth derivative order
xbar - target point to approximate around
x    - vector of N stencil points
"""
def fdcoeffV(k, xbar, x):
    if isinstance(x, list):
        x = np.array(x)

    n = len(x)
    A = np.ones((n, n))
    xrow = np.transpose(x - xbar)  # displacements

    for i in range(1, n+1):
        A[i-1, :] = np.divide(np.power(xrow, (i - 1)),
                              math.factorial(i - 1))

    b = np.zeros((n, 1))  # b is right hand side,
    b[k] = 1  # so kth derivative term remains
    c = np.linalg.solve(A, b)  # solve for coefficients
    return np.transpose(c)  # row vector
```

4 Questão 03

Pede-se $\frac{\mathrm{d}J_0(x)}{\mathrm{d}x}$ em $x = 3$, onde

$$J_\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,(m+\alpha)!} \left(\frac{x}{2}\right)^{2m+\alpha},$$

e por sua vez

$$J_0(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\, m!} \left(\frac{x}{2}\right)^{2m}.$$

Considerando que a função de Bessel converge, podemos aplicar a derivada da série infinita obtendo

$$J_0'(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,(m-1)!} \left(\frac{x}{2}\right)^{2m-1} = J_{(-1)}(x) = -J_1(x).$$

Dessa forma temos a solução exata

$$f'(x) = -0.339059$$

e as aproximações
(a) Backward com 2 pontos

$$D_{-2} = -0.3836, \quad error = 4.46\text{e}{-2}$$

(b) Backward com 3 pontos

$$D_{-3} = -0.3439, \quad error = 4.89\text{e}{-2}$$

(c) Forward com 2 pontos

$$D_{+2} = -0.2908, \quad error = -4.83\text{e}{-2}$$

(d) Forward com 3 pontos

$$D_{+3} = -0.3414, \quad error = 2.38\text{e}{-3}$$

(e) Central com 2 pontos

$$D_{0,2} = -0.3372, \quad error = -1.84\text{e}{-3}$$

(f) Central com 4 pontos

$$D_{0,4} = -0.3390, \quad error = -1.53\text{e}{-5}$$

5 Questão 04

Considere a função

$$f(x) = \mathrm{e}^x \sin(x).$$

Temos, pela regra do produto,

$$f'(x) = \mathrm{e}^x \sin(x) + \mathrm{e}^x \cos(x),$$

e aplicando a regra do produto novamente

$$f''(x) = 2\mathrm{e}^x \cos(x).$$

Considerando, por exemplo, $h = \Delta x = 1$ e utilizando a aproximação obtida na Eq.(1) para $f'(x = 2)$ e $f''(x = 2)$, obtemos os seguintes valores para os casos solicitados

(i) Central com 2 e 3 pontos obtemos os coeficientes

(ii) Central com 4 e 5 pontos

caso desejemos plotar a dependência do erro de aproximação com relação ao passo dediscretização h, vide Fig. 2, percebemos que as aproximações com mais pontos, 3 pontos para a primeira derivada e 5 pontos para a segunda, resultam em um erro menor independente do valor de h. Além disso, percebe-se que o valor de h influencia positivamente o erro, com menores valores de h resultando em erros menores, efeito observado independente do número de pontos.

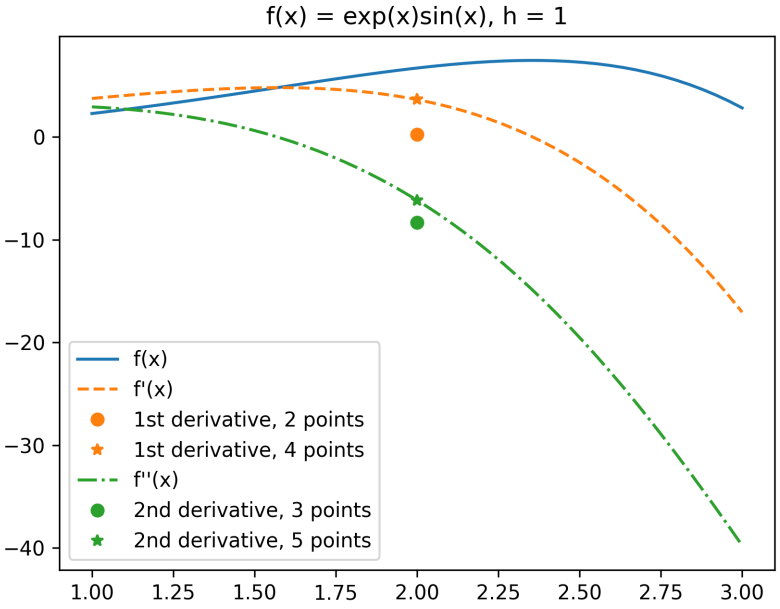


Figura 1: Aproximação para 1a e 2a derivada.

6 Questão 5

Q5 took 12.47940993309021 seconds

7 Questão 6

```
## Input data
# Geometry and material properties
L = 1
lamb = 0.01
# Boudary conditions
q = 1
# Initial conditions
T0 = 1
# Final time
tend = 30

## Discretization
N = 50  # spatial
Nt = 3000  # temporal
x = np.linspace(0, L, N)
t = np.linspace(0, tend, Nt)
```

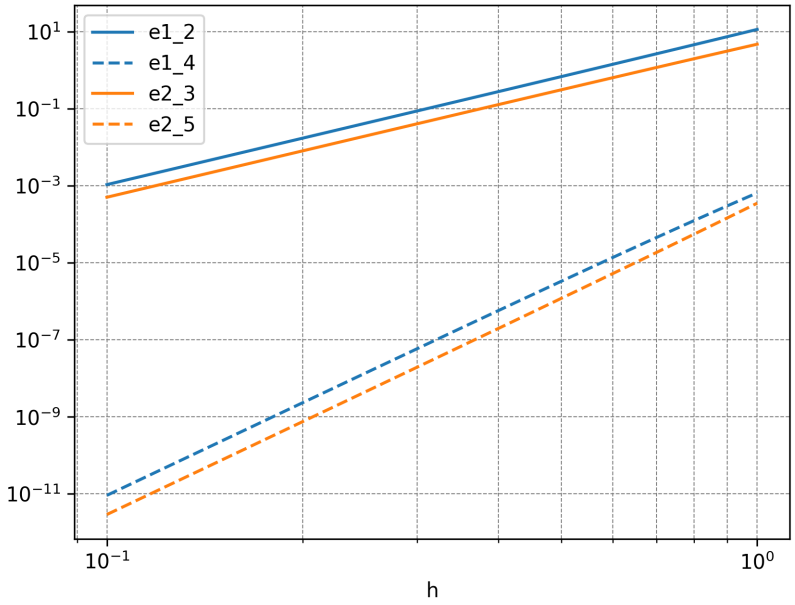


Figura 2: Evolução do erro de aproximação com variação de h.

```
dx = x[1] - x[0]
dt = t[1] - t[0]

beta = lamb * dt / (dx**2)

T = np.zeros((N, Nt))
for k, tk in enumerate(t):
    if tk == 0:
        for i, xi in enumerate(x):
            T[i, k] = T0
    else:
        for i, xi in enumerate(x):
            if np.isclose(xi, 0):
                if t[k] <= 10:
                    T[i, k] = T[i, k - 1] + beta * (
                        2 * T[i + 1, k - 1] - 2 * T[i, k - 1] + 2 * dx * q
                    )
                else:
                    T[i, k] = T[i, k - 1] + beta * (
                        2 * T[i + 1, k - 1] - 2 * T[i, k - 1]
                    )
            elif np.isclose(xi, L):
                T[i, k] = T[i, k - 1] + beta * (2 * T[i - 1, k - 1] - 2 * T[i, k - 1])
```

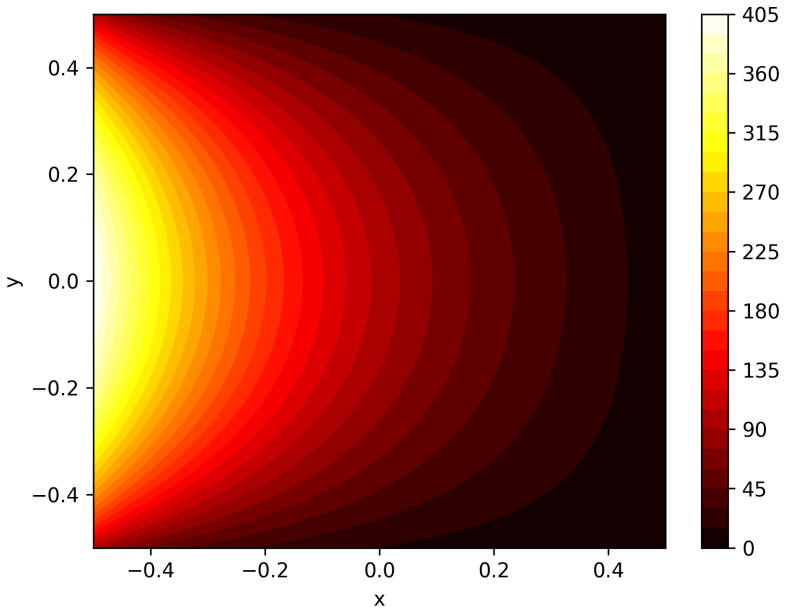


Figura 3: Heatmap para a solução explícita em $T = 0$ segundos

```
        else:
            T[i, k] = T[i, k - 1] + beta * (
                T[i - 1, k - 1] - 2 * T[i, k - 1] + T[i + 1, k - 1]
            )

## Input data
# Geometry and material properties
L = 1
lamb = 0.01
# Boudary conditions
q = 1
# Initial conditions
T0 = 1
# Final time
tend = 30

## Discretization
N = 50  # spatial
Nt = 3000  # temporal
x = np.linspace(0, L, N)
t = np.linspace(0, tend, Nt)
dx = x[1] - x[0]
dt = t[1] - t[0]
```

```
beta = lamb * dt / (dx**2)

T = np.zeros((N, Nt))

# Matrix A assembly
A = np.zeros((N, N))

for i, xi in enumerate(x):
    if np.isclose(xi, 0):
        A[i, i] = 2 * beta + 1
        A[i, i + 1] = -2 * beta
    elif np.isclose(xi, L):
        A[i, i - 1] = -2 * beta
        A[i, i] = 2 * beta + 1
    else:
        A[i, i - 1] = -beta
        A[i, i + 1] = -beta
        A[i, i] = 2 * beta + 1

# print(A)

b = np.zeros(N)
# Solution
for k, tk in enumerate(t):
    if tk == 0:
        T[:, k] = T0
    else:
        b = list(T[:, k - 1])
        if t[k] <= 10:
            b[0] = b[0] + 2 * beta * dx * q
            T[:, k] = np.linalg.solve(A, b)
        else:
            T[:, k] = np.linalg.solve(A, b)
```

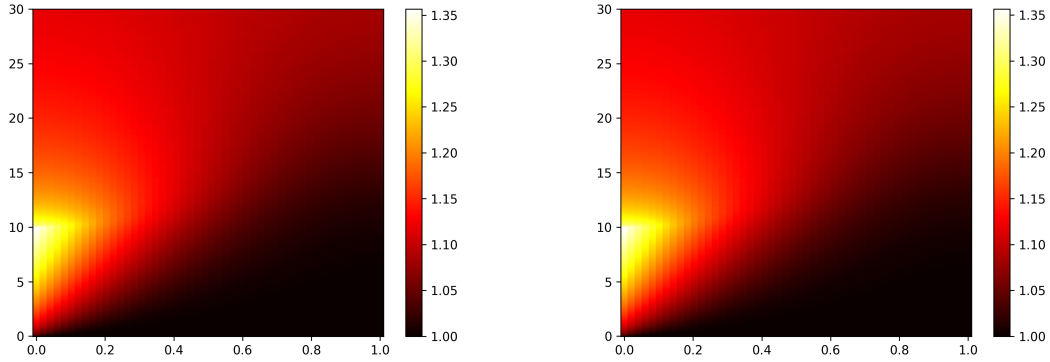
Q6a took 8.867920160293579 seconds Q6b took 0.04085826873779297 seconds

Quanto à discretização adotada: para as derivadas no espaço, foi usada a aproximação de derivada central de 3 pontos e para as derivadas no tempo, aproximação de dois pontos.

Para aplicar as condições de contorno, foram utilizados pontos fantasma nas aproximações das derivadas espaciais, já que se tratavam de condições de contorno de Neumann.

8 Questão 7

Q7a took 2.9745030403137207 seconds Q7b took 4.0168750286102295 seconds



(a) solução explícita

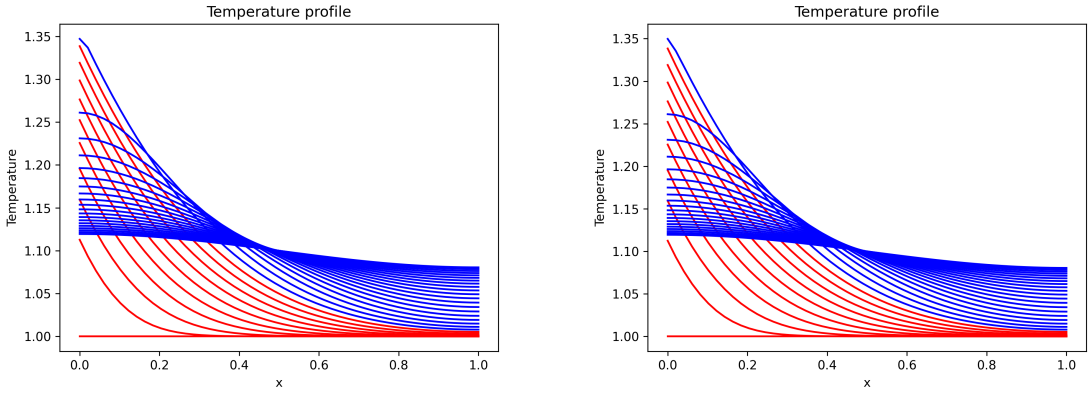
(b) solução implícita

Figura 4: Colobar para Questão 6

Referências

[1] Randall J. LeVeque (2007) *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, Society for Industrial and Applied Mathematics

[2] Francisco Chinesta, Amine Ammar, Adrien Leygue, Roland Keunings. An overview of the proper generalized decomposition with applications in computational rheology. Journal of Non-Newtonian Fluid Mechanics, 2011, 166 (11), pp.578-592. hal-01061441



(a) solução explícita

(b) solução implícita

Figura 5: Perfis de temperature Questão 6

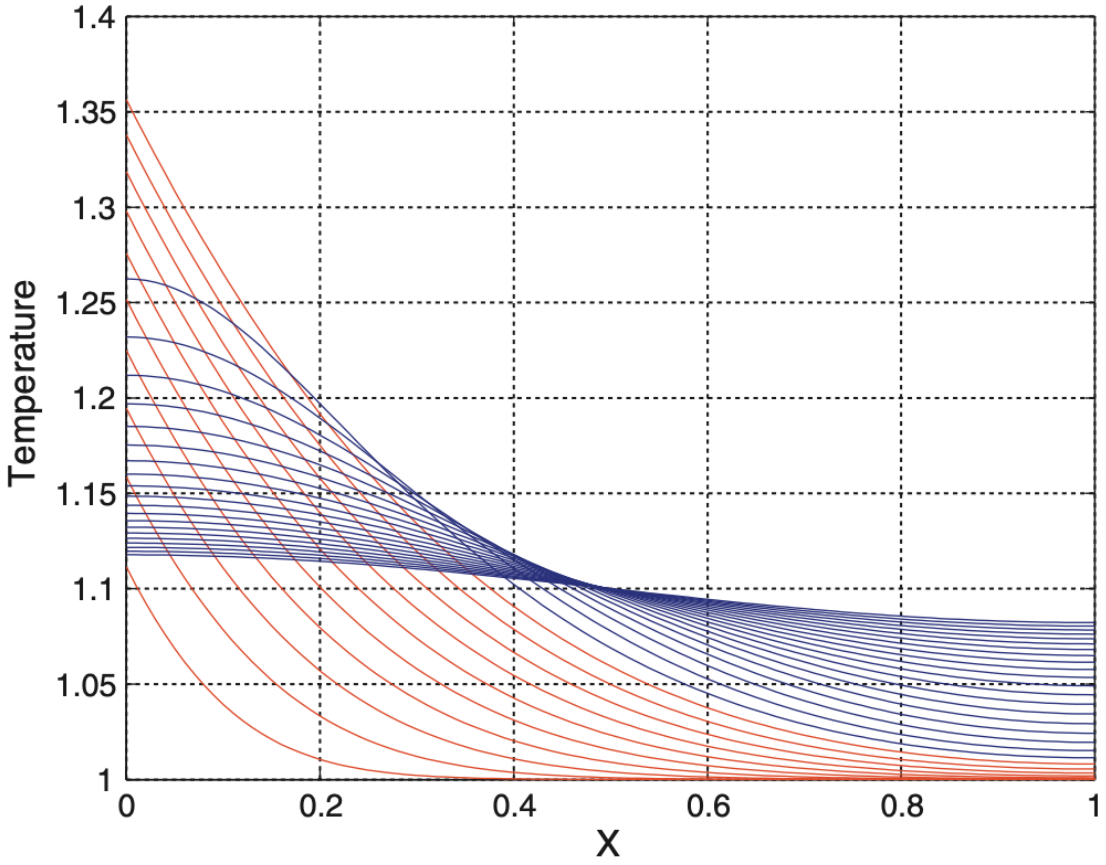
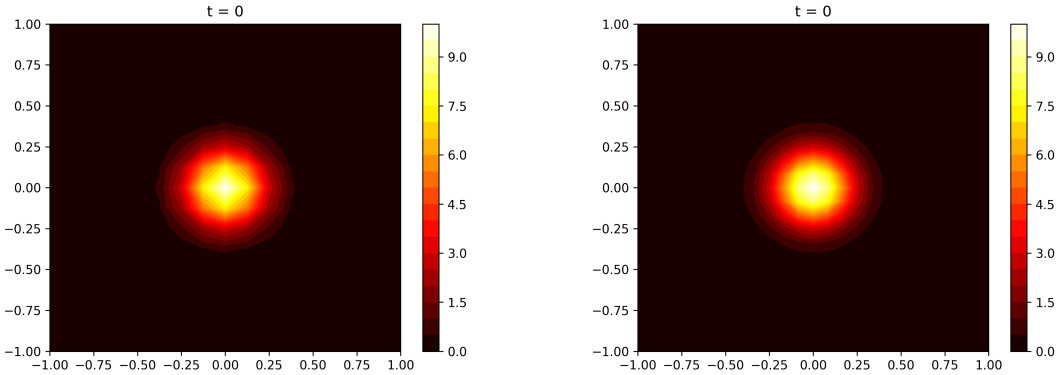


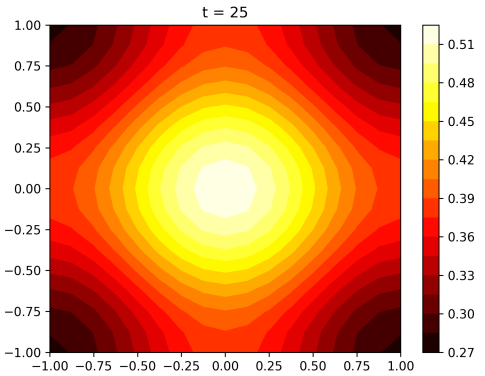
Figura 6: Figura 01 de [2], "Temperature profiles corresponding to the source term (13) at discrete times $t_m = m$,for $m = 1, 2, ..., 30$. The red curves correspond to the heating stagel up to $t = 10$, while the blue curves for $t > 10$ illustrate the heat transfer by conduction from the warmest zones towards the coldest ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)"



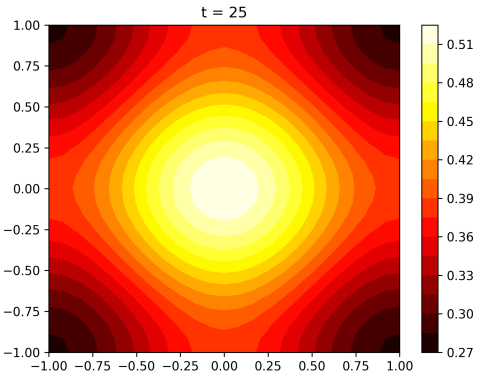
(a) $T = 0$ s, solução explícita

(b) $T = 0$ s, solução implícita

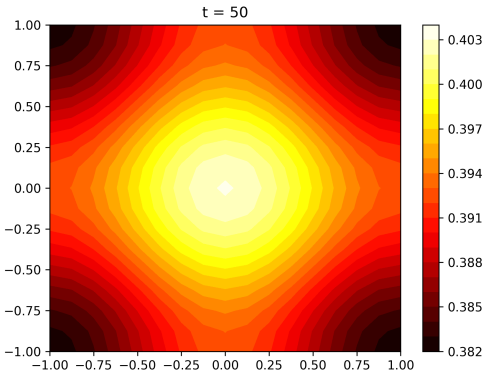
Figura 7: Distribuição de temperatura inicial para a questão 7.



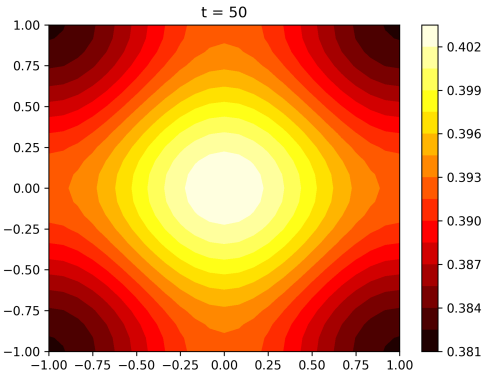
(a) $T = 25\ s$, solução explícita



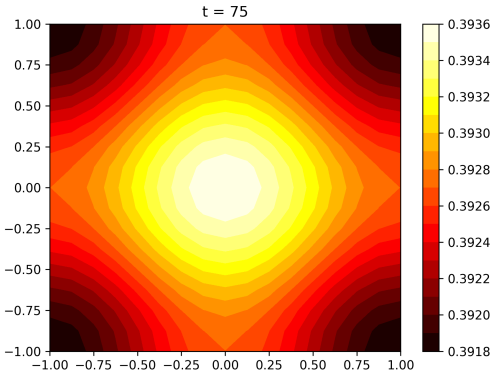
(b) $T = 25\ s$, solução implícita



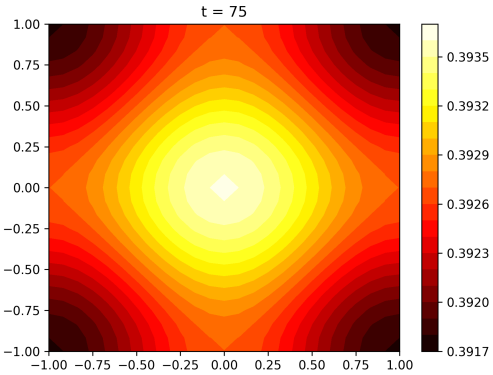
(c) $T = 50\ s$, solução explícita



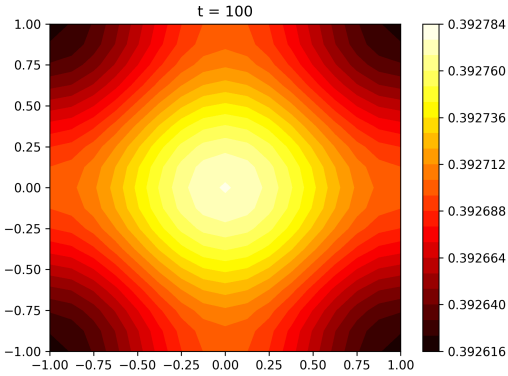
(d) $T = 50\ s$, solução implícita



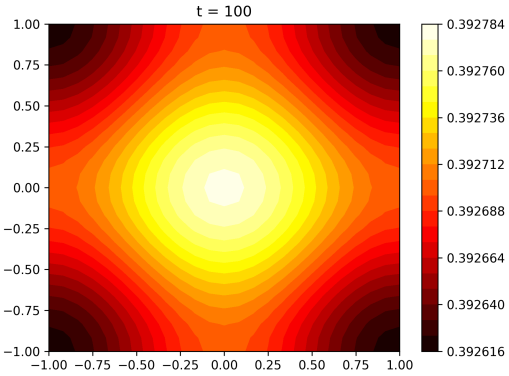
(e) $T = 75\ s$, solução explícita



(f) $T = 75\ s$, solução implícita

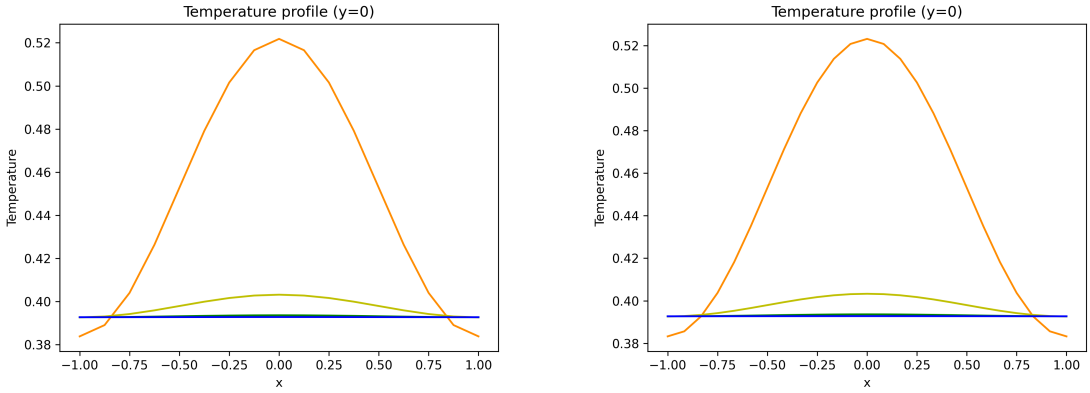


(g) $T = 100\ s$, solução explícita



(h) $T = 100\ s$, solução implícita

Figura 8: Distribuição de temperatura em evoluindo em função do tempo para a Questão 7.



(a) solução explícita

(b) solução implícita

Figura 9: Perfis de temperatura para questão 7.