Trabalho 4 - Análise de Variações nos Parâmetros

Renato

Introdução

Este relatório apresenta a solução numérica da equação diferencial parcial:

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} - \alpha\frac{\partial^2 C}{\partial x^2} = 0, \quad 0 < x < L_x, \quad t > 0$$

com as condições de contorno:

$$C(x = 0, t) = C_E, \quad \left.\frac{\partial C}{\partial x}\right|_{x=L_x} = 0$$

O objetivo é determinar o perfil de concentração $C(x, t)$ ao longo do domínio espacial e temporal, utilizando o método de diferenças finitas explícito, considerando variações nos parâmetros físicos (α e u) e numéricos (n_x) para analisar diferentes cenários.

Discretização

Utilizamos as mesmas aproximações das derivadas conforme descrito anteriormente, resultando na equação discretizada:

$$C_i^{n+1} = C_i^n - \Delta t \left(u\frac{C_i^n - C_{i-1}^n}{\Delta x} - \alpha\frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta x^2} \right)$$

A condição de estabilidade permanece:

$$\Delta t \leq \frac{1}{\dfrac{2\alpha}{\Delta x^2} + \dfrac{u}{\Delta x}}$$

Parâmetros dos Testes

Para explorar diferentes cenários, variamos os seguintes parâmetros:

- **Coeficiente de difusão** (α):
  - $\alpha = 0.001$
  - $\alpha = 0.01$
  - $\alpha = 0.1$
- **Velocidade de advecção** (u):

  - $u = 0.5$
  - $u = 1.0$
  - $u = 2.0$
- **Número de pontos espaciais** (n_x):
  - $n_x = 25$
  - $n_x = 50$
  - $n_x = 100$

Os demais parâmetros permanecem constantes:

- Comprimento do domínio: $L_x = 1.0$
- Condição de contorno em $x = 0$: $C_E = 1.0$
- Tempo final da simulação: $T_{\text{final}} = 0.5$

Implementação em Python

Para acomodar as variações nos parâmetros, modificamos o código para incluir loops sobre os valores de α, u e n_x, permitindo executar múltiplas simulações.

Código

```
import numpy as np
import matplotlib.pyplot as plt

# Par metros constantes
Lx = 1.0        # Comprimento do dom nio
CE = 1.0        # Condi  o de contorno em x = 0
T_final = 0.5   # Tempo final da simula  o

# Valores a serem testados
alfa_values = [0.001, 0.01, 0.1]
u_values = [0.5, 1.0, 2.0]
nx_values = [25, 50, 100]

# Loop sobre os valores de alfa, u e nx
for alfa in alfa_values:
    for u in u_values:
        for nx in nx_values:
            dx = Lx / (nx - 1)          # Tamanho do passo
               espacial
            # Condi  o de estabilidade para o passo de tempo
            dt_estabilidade = 1.0 / (2 * alfa / dx**2 + u / dx)
            dt = 0.9 * dt_estabilidade  # Um pouco menor que o
               m ximo permitido
            nt = int(np.ceil(T_final / dt))  # N mero de passos
               no tempo
```

```
dt = T_final / nt                  # Recalcular dt para ajustar exatamente em T_final

# Malhas espacial e temporal
x = np.linspace(0, Lx, nx)
t = np.linspace(0, T_final, nt+1)

# Condição inicial: C(x, t=0) = 0
C = np.zeros(nx)
C_todos = np.zeros((nt+1, nx))
C_todos[0, :] = C.copy()

# Função para aplicar a condição de Neumann
def aplicar_condicao_neumann(C):
    C[-1] = C[-2]
    return C

# Loop no tempo
for n in range(nt):
    # Aplicar condições de contorno
    C[0] = CE
    C = aplicar_condicao_neumann(C)

    # Criar uma cópia de C para armazenar os novos valores
    C_novo = C.copy()

    # Atualizar os pontos interiores
    for i in range(1, nx-1):
        # Calcular as diferenças finitas
        dCdx = (C[i] - C[i-1]) / dx
        d2Cdx2 = (C[i+1] - 2*C[i] + C[i-1]) / dx**2

        # Atualizar usando o esquema explícito
        C_novo[i] = C[i] - dt * (u * dCdx - alfa * d2Cdx2)

    # Atualizar no último ponto
    i = nx - 1
    dCdx = (C[i] - C[i-1]) / dx
    d2Cdx2 = (C[i-1] - 2*C[i] + C[i-1]) / dx**2
    C_novo[i] = C[i] - dt * (u * dCdx - alfa * d2Cdx2)

    # Atualizar a solução
    C = C_novo.copy()
    C_todos[n+1, :] = C.copy()

# Plotar os resultados
plt.figure(figsize=(10, 6))
for i in range(0, nt+1, nt//5):
```

```
                plt.plot(x, C_todos[i, :], label=f"t = {t[i]:.2f}
                    s")
            plt.xlabel('Posi  o x')
            plt.ylabel('Concentra  o C')
            plt.title(f'Perfil de C - alfa={alfa}, u={u}, nx={nx}
                ')
            plt.legend()
            plt.grid(True)
            # Salvar a figura
            filename = f'perfil_C_alfa_{alfa}_u_{u}_nx_{nx}.png'
            plt.savefig(filename, dpi=300)
            plt.close()
```

Descrição do Código

O código foi modificado para incluir loops sobre os valores de α, u e n_x. Para cada combinação desses parâmetros, o código:

1. Calcula o tamanho do passo espacial Δx e o passo de tempo Δt conforme a condição de estabilidade.
2. Inicializa as malhas espacial e temporal, e define a condição inicial.
3. Executa o loop no tempo, atualizando a concentração C em cada ponto espacial.
4. Plota e salva os gráficos dos perfis de concentração em diferentes tempos para cada conjunto de parâmetros.

As figuras são salvas com nomes que identificam os parâmetros utilizados, facilitando a organização e análise dos resultados.

Resultados da Simulação

Os resultados obtidos mostram como as variações nos parâmetros influenciam o comportamento da concentração $C(x, t)$. A seguir, apresentamos uma análise dos efeitos de cada parâmetro.

Efeito do Coeficiente de Difusão (α)

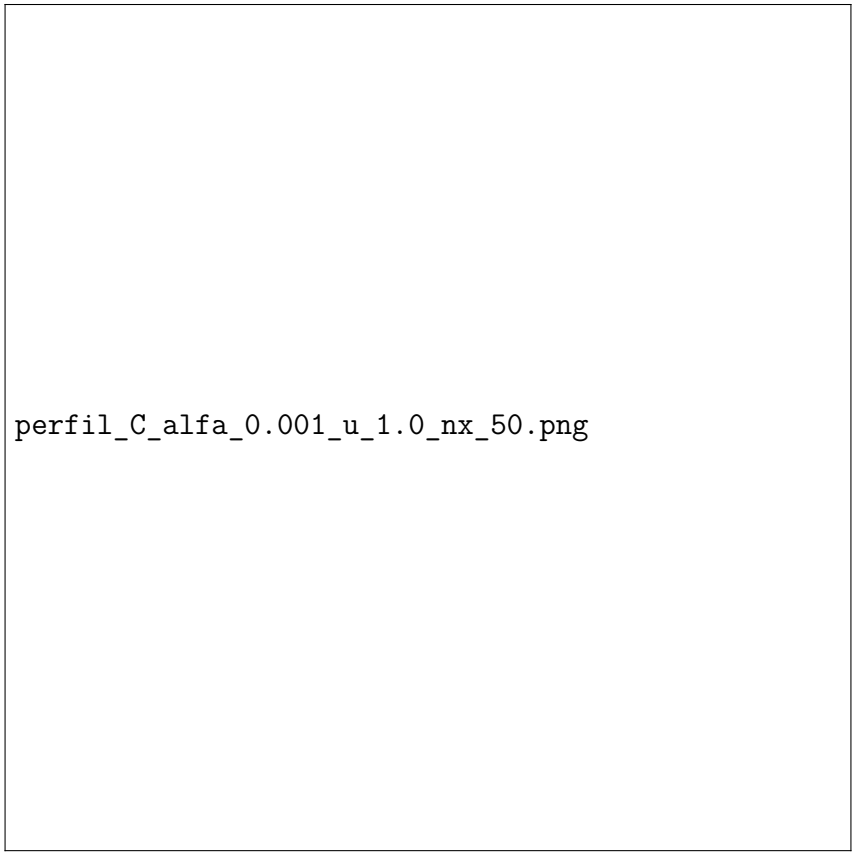

Figura 1: Perfil de C para $\alpha = 0.001$, $u = 1.0$, $n_x = 50$.

Com um valor baixo de α, a difusão é menos significativa, resultando em um perfil de concentração com frente mais abrupta, conforme mostrado na Figura 1. A advecção domina o processo, transportando a concentração na direção positiva de x sem muito espalhamento.

Efeito da Velocidade de Advecção (u)

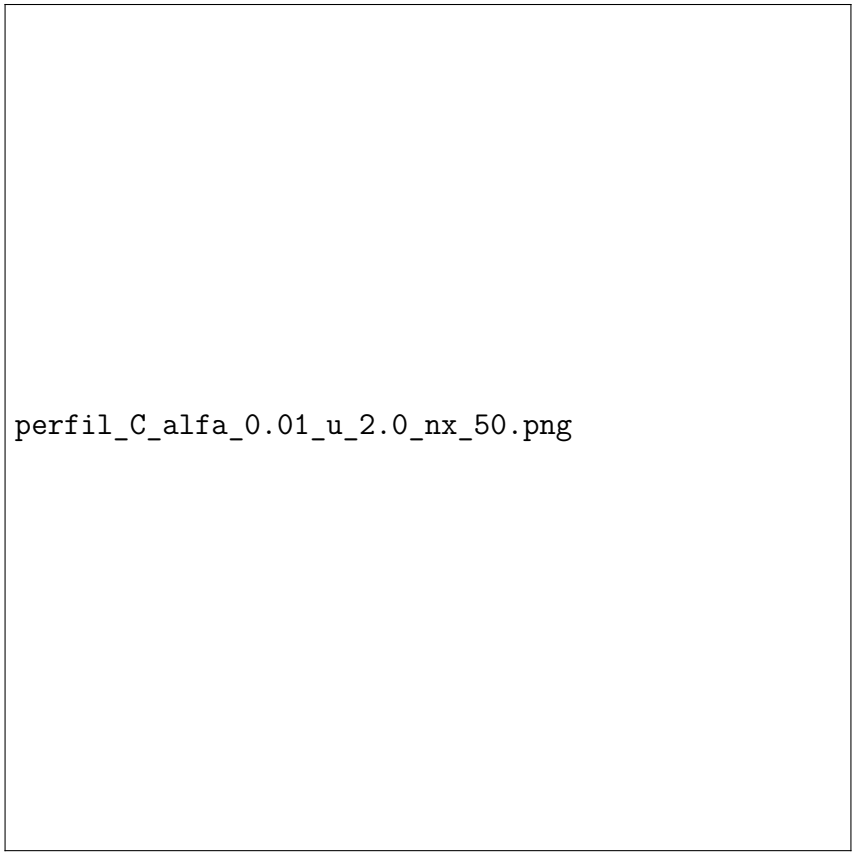

Figura 2: Perfil de C para $\alpha = 0.01$, $u = 2.0$, $n_x = 50$.

Ao aumentar a velocidade de advecção, a concentração é transportada mais rapidamente ao longo do domínio, como visto na Figura 2. Isso resulta em um deslocamento mais acentuado do perfil de concentração em tempos menores.

Efeito do Número de Pontos Espaciais (n_x)

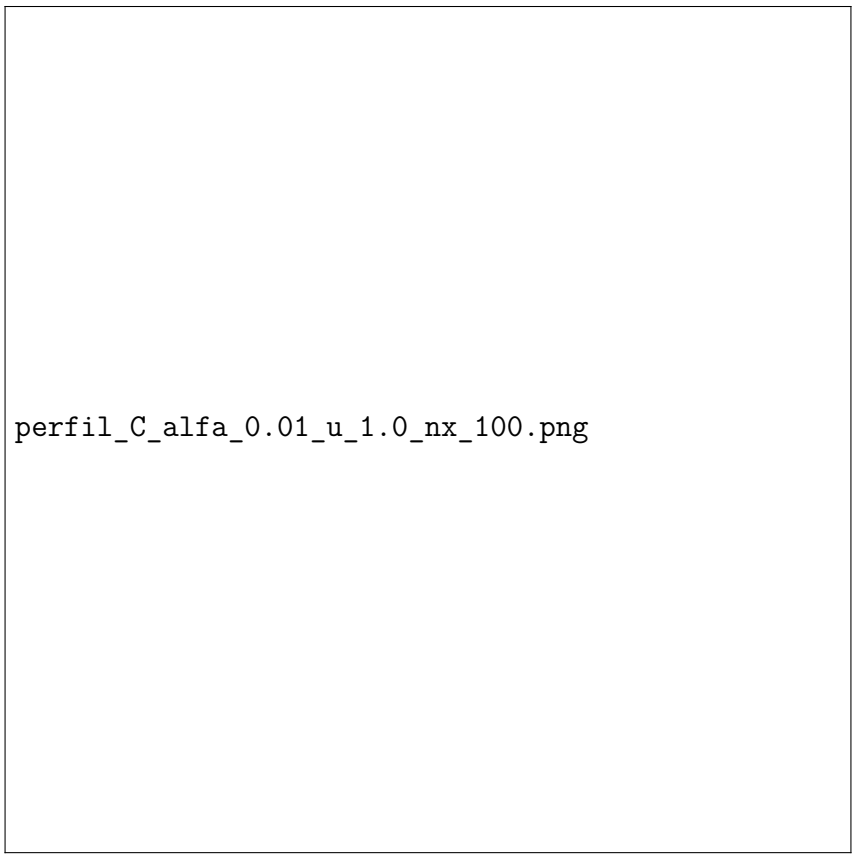

Figura 3: Perfil de C para $\alpha = 0.01$, $u = 1.0$, $n_x = 100$.

Com um maior número de pontos espaciais, a malha fica mais refinada, permitindo capturar detalhes mais precisos do perfil de concentração, como mostrado na Figura 3. Observa-se que o perfil é mais suave e apresenta melhor resolução espacial.

Análise dos Resultados

As variações nos parâmetros demonstram claramente seus impactos no comportamento da solução:

- **Coeficiente de Difusão** (α):
  - Valores baixos de α resultam em menor difusão, mantendo a frente de concentração mais definida.
  - Valores altos de α aumentam o espalhamento da concentração, suavizando o perfil.
- **Velocidade de Advecção** (u):
  - Valores altos de u aceleram o transporte da concentração, deslocando o perfil mais rapidamente.

  - Valores baixos de u reduzem o efeito advectivo, permitindo que a difusão tenha maior influência.

- **Número de Pontos Espaciais (n_x)**:
  - Um maior n_x proporciona uma malha mais refinada, aumentando a precisão da solução numérica.
  - Malhas mais grosseiras (n_x baixo) podem não capturar adequadamente variações rápidas na concentração.

Conclusão

A análise das variações nos parâmetros físicos e numéricos permitiu compreender melhor os diferentes cenários do problema. Observou-se que os parâmetros α e u controlam, respectivamente, os processos de difusão e advecção, influenciando significativamente o perfil de concentração ao longo do tempo.

Além disso, o refinamento da malha espacial, controlado por n_x, é crucial para a precisão da solução numérica. Ajustes adequados nos parâmetros permitem modelar diferentes situações físicas e obter resultados confiáveis.