

# Trabalho 3

Professor: Grazione Souza

- Nome do aluno: Pedro Henrique Couto Silva;
- Matrícula: 202020466311
- Nome do aluno : Vinicius Carvalho Monnerat Bandeira;
- Matrícula: 202020466711

O estudo aqui proposto pode ser encontrado em [https://github.com/ViniciusCMB/Metodos\\_Num.git](https://github.com/ViniciusCMB/Metodos_Num.git)

---

## Bibliotecas

In [171]...

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

## Função para resolver o problema

```
def funcao_C(T, Nx, Lx, alpha, k, dt, CE):
    # Inicializando a matriz de coeficientes e o vetor de solução
    dx = Lx / Nx # passo no espaço
    Nt = int(T / dt)
    A = np.zeros((Nx, Nx))
    C = np.zeros(Nx)

    # Preenchendo a matriz de coeficientes A
    for i in range(1, Nx-1):
        A[i, i-1] = -alpha / dx**2
        A[i, i] = 1 / dt + 2 * alpha / dx**2 + k
        A[i, i+1] = -alpha / dx**2

    # Condições de contorno
    A[0, 0] = 1.0
    A[-1, -1] = 1.0
    A[-1, -2] = -1.0

    # Inicializando a solução
    C[0] = CE

    # Loop no tempo
    for n in range(Nt):
        # Vetor de termos independentes
        b = C / dt
        b[0] = CE
        b[-1] = 0.0

        # Resolvendo o sistema linear
        C = np.linalg.solve(A, b)

    x = np.linspace(0, Lx, Nx)

    return C, x
```

## Parâmetros de Entrada:

- T: Tempo total de simulação.
- Nx: Número de pontos no espaço.
- Lx: Comprimento do domínio.
- alpha: Constante de difusão.
- k: Constante de reação.
- dt: Passo no tempo.
- CE: Condição de contorno em  $x = 0$ .

## Inicialização:

- dx: Passo no espaço, calculado dividindo o comprimento do domínio pelo número de pontos no espaço.
- Nt: Número de passos no tempo, calculado dividindo o tempo total pelo passo no tempo.
- A: Matriz de coeficientes, inicializada com zeros.
- C: Vetor de solução, inicializado com zeros.
- Preenchimento da Matriz de Coeficientes A: -- Para cada ponto no espaço (exceto os extremos), a matriz A é preenchida com os coeficientes baseados na equação de difusão.  $A[i, i-1]$ ,  $A[i, i]$  e  $A[i, i+1]$  são preenchidos com valores que dependem de alpha, dx, dt e k.

## Condições de Contorno:

As condições de contorno são aplicadas nos extremos da matriz A. Inicialização da Solução:

O vetor C é inicializado com a condição de contorno em  $x = 0$ .

## Loop no Tempo:

Para cada passo no tempo, o vetor de termos independentes b é calculado. A condição de contorno é aplicada ao vetor b. O sistema linear  $A * C = b$  é resolvido para obter a nova solução C. Retorno:

A função retorna o vetor de solução C e o vetor de posições x.

## Função funcao\_C:

De maneira geral função resolve uma equação de difusão-reação em uma dimensão usando o método das diferenças finitas. Ela calcula a distribuição de concentração C ao longo do domínio Lx após um tempo T, considerando as condições de contorno e os parâmetros fornecidos.

## Parâmetros para a solução

```
In [173... # Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 100 # número de pontos no espaço
alpha = 0.01 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo
T = 1.0 # tempo total

# Condições de contorno
CE = 5 # valor de C em  $x = 0$ 
```

## Obtém a solução e o domínio

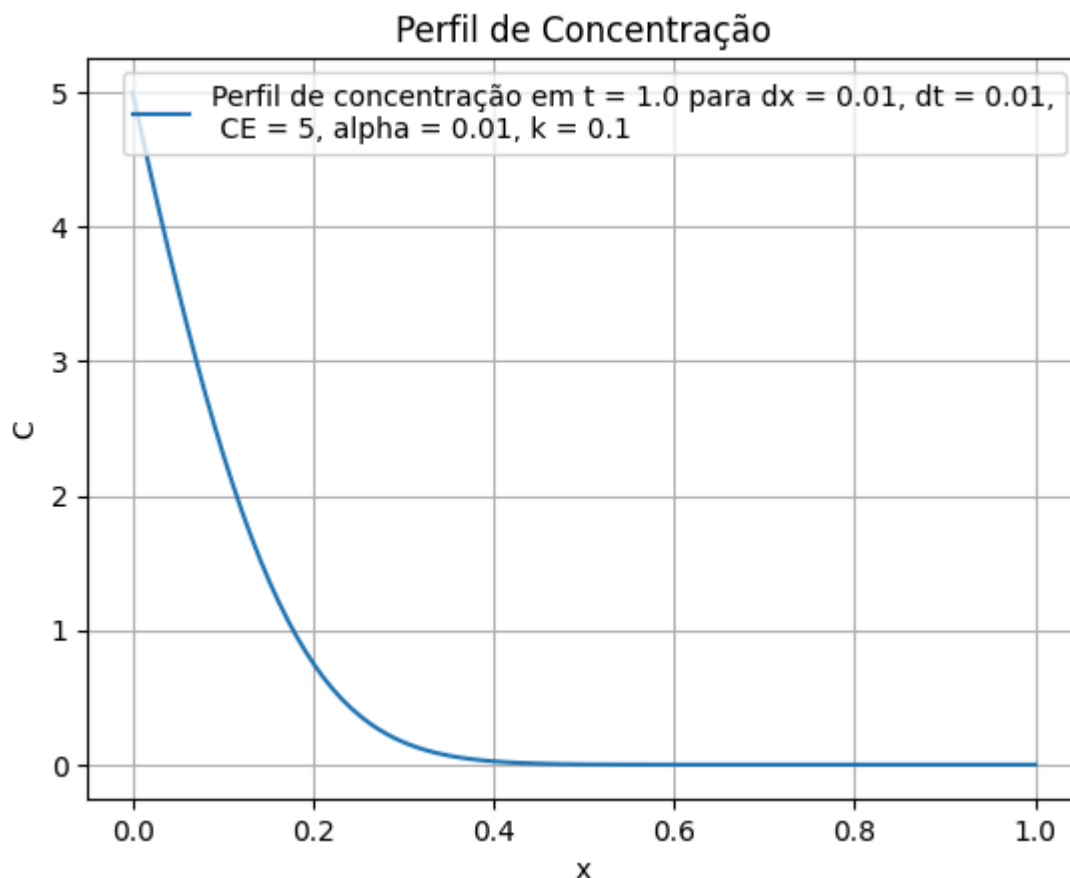
```
In [181... C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)

C
```

```
Out[181... array([5.          , 4.84498835, 4.69482169, 4.54934986, 4.40842737,
 4.27191331, 4.13967116, 4.01156868, 3.88747778, 3.76727435,
 3.65083819, 3.53805288, 3.42880561, 3.32298715, 3.22049168,
 3.1212167 , 3.02506294, 2.93193424, 2.84173748, 2.75438245,
 2.66978181, 2.58785094, 2.50850793, 2.43167343, 2.3572706 ,
 2.28522504, 2.21546471, 2.14791984, 2.08252289, 2.01920846,
 1.95791325, 1.89857594, 1.84113722, 1.78553963, 1.73172758,
 1.67964725, 1.62924658, 1.58047515, 1.5332842 , 1.48762653,
 1.44345649, 1.4007299 , 1.35940405, 1.3194376 , 1.28079059,
 1.24342437, 1.20730157, 1.17238608, 1.13864297, 1.1060385 ,
 1.07454008, 1.04411619, 1.01473642, 0.98637139, 0.95899273,
 0.93257306, 0.90708597, 0.88250596, 0.85880846, 0.83596977,
 0.81396704, 0.79277829, 0.77238231, 0.75275872, 0.73388788,
 0.71575094, 0.69832974, 0.68160688, 0.66556562, 0.65018993,
 0.63546443, 0.62137439, 0.60790573, 0.59504498, 0.58277927,
 0.57109634, 0.5599845 , 0.54943265, 0.53943024, 0.52996725,
 0.52103424, 0.51262226, 0.5047229 , 0.49732826, 0.49043095,
 0.48402408, 0.47810123, 0.47265648, 0.46768438, 0.46317997,
 0.45913875, 0.45555666, 0.45243013, 0.44975603, 0.44753168,
 0.44575487, 0.44442382, 0.44353718, 0.44309409, 0.44309409])
```

## Plota o resultado obtido

```
In [175... # Plotando o resultado
plt.plot(x, C, label=f'Perfil de concentração em t = {T} para dx = {Lx / (Nx)}, dt = {dt}')
plt.xlabel('x')
plt.ylabel('C')
plt.grid()
titulo = 'Perfil de Concentração'
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')
```



# Estudando valores para $Nx = \{10, 100, 1000\}$ com $Lx = 1, \alpha = 0.01, k = 0.1, dt = 0.01$ , e tempo de simulação $T = 1.0$

O que se espera de valores crescentes de  $Nx$

Valores crescentes de ( $Nx$ ) representam um aumento no número de pontos no espaço. Isso significa que a discretização do domínio ( $Lx$ ) está se tornando mais fina. Em termos de concentração ( $C$ ), espera-se que:

- A solução ( $C$ ) se torne mais precisa e suave, capturando melhor as variações ao longo do domínio.
- A resolução espacial aumentada permita uma melhor representação dos gradientes de concentração.
- A estabilidade e a precisão do método numérico melhorem, resultando em perfis de concentração mais detalhados e realistas.

Em resumo, com valores crescentes de ( $Nx$ ), a discretização espacial se torna mais refinada, levando a uma solução mais precisa e detalhada da concentração ao longo do domínio.

In [176...

```
# Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 10 # número de pontos no espaço
alpha = 0.01 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo
T = 1.0 # tempo total

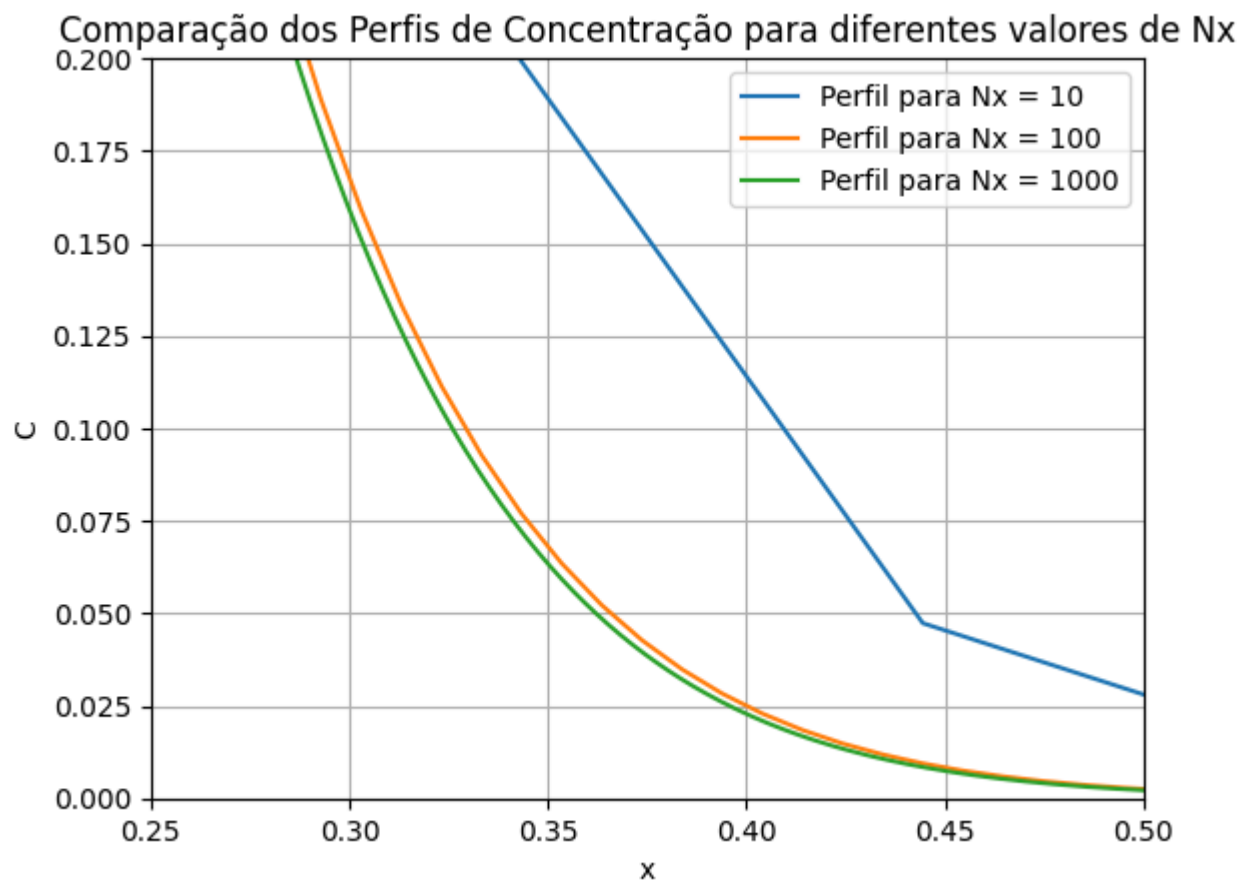
# Condições de contorno
CE = 5 # valor de C em x = 0

# Resultado
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para Nx = {Nx}')

Nx = 100 # número de pontos no espaço
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para Nx = {Nx}')

Nx = 1000 # número de pontos no espaço
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para Nx = {Nx}')

# Plotando o resultado
plt.xlabel('x')
plt.ylabel('C')
plt.axis([0.25, 0.5, 0, 0.2])
plt.grid()
titulo = 'Comparação dos Perfis de Concentração para diferentes valores de Nx'
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')
```



Como é possível observar, valores maiores para  $N_x$  se aproximam melhor de um resultado consistente.

Estudando valores para  $k = \{0.1, 1, 10\}$  com  $Lx = 1$ ,  $Nx = 100$ ,  $\alpha = 0.01$ ,  $dt = 0.01$ , e tempo de simulação  $T = 1.0$

O que se espera de valores crescentes de  $k$

Valores crescentes de  $(k)$  representam um aumento na constante de reação. Isso significa que a taxa de reação química está aumentando. Em termos de concentração  $(C)$ , espera-se que:

- A concentração  $(C)$  diminua mais rapidamente ao longo do domínio  $(Lx)$ .
- A solução  $(C)$  se estabilize em valores mais baixos conforme  $(k)$  aumenta.
- A difusão terá menos influência comparada à reação, resultando em perfis de concentração mais acentuados.

Em resumo, com valores crescentes de  $(k)$ , a reação domina o processo, levando a uma redução mais rápida da concentração ao longo do tempo e do espaço.

```
In [183... # Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 100 # número de pontos no espaço
alpha = 0.1 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo
T = 1.0 # tempo total

# Condições de contorno
CE = 5 # valor de C em x = 0
```

```

# Resultado
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para k = {k}')

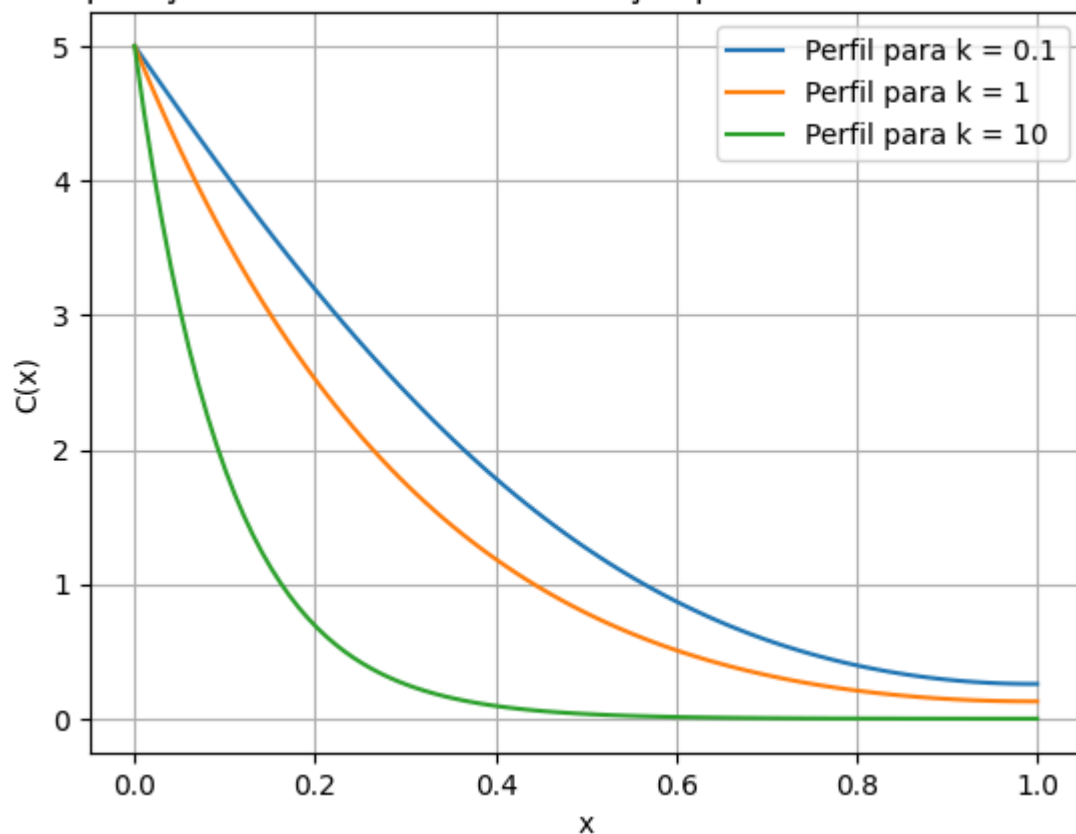
k = 1 # constante k
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para k = {k}')

k = 10 # constante k
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para k = {k}')

plt.xlabel('x')
plt.ylabel('C(x)')
plt.grid()
titulo = 'Comparação dos Perfis de Concentração para diferentes valores de k'
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')

```

### Comparação dos Perfis de Concentração para diferentes valores de k



Como é possível observar, valores crescentes de  $k$  tornam a reação mais acentuada.

Estudando valores para  $\alpha = \{0.01, 0.1, 1, 10\}$  com  $Lx = 1$ ,  $Nx = 100$ ,  $k = 0.1$ ,  $dt = 0.01$ , e tempo de simulação  $T = 1.0$

Valores crescentes de ( $\alpha$ ) representam um aumento na constante de difusão. Isso significa que a taxa de difusão está aumentando. Em termos de concentração ( $C$ ), espera-se que:

- A concentração ( $C$ ) se espalhe mais rapidamente ao longo do domínio ( $Lx$ ).
- A solução ( $C$ ) se torne mais uniforme ao longo do domínio conforme ( $\alpha$ ) aumenta.

- A influência da difusão aumentada resulte em perfis de concentração mais suaves e menos acentuados.

Em resumo, com valores crescentes de ( $\alpha$ ), a difusão domina o processo, levando a uma distribuição mais uniforme da concentração ao longo do tempo e do espaço.

In [178...

```
# Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 100 # número de pontos no espaço
alpha = 0.01 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo
T = 1.0 # tempo total

# Condições de contorno
CE = 5 # valor de C em x = 0

# Resultado
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para  $\alpha = \{alpha\}$ ')

alpha = 0.1 # constante alpha
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para  $\alpha = \{alpha\}$ ')

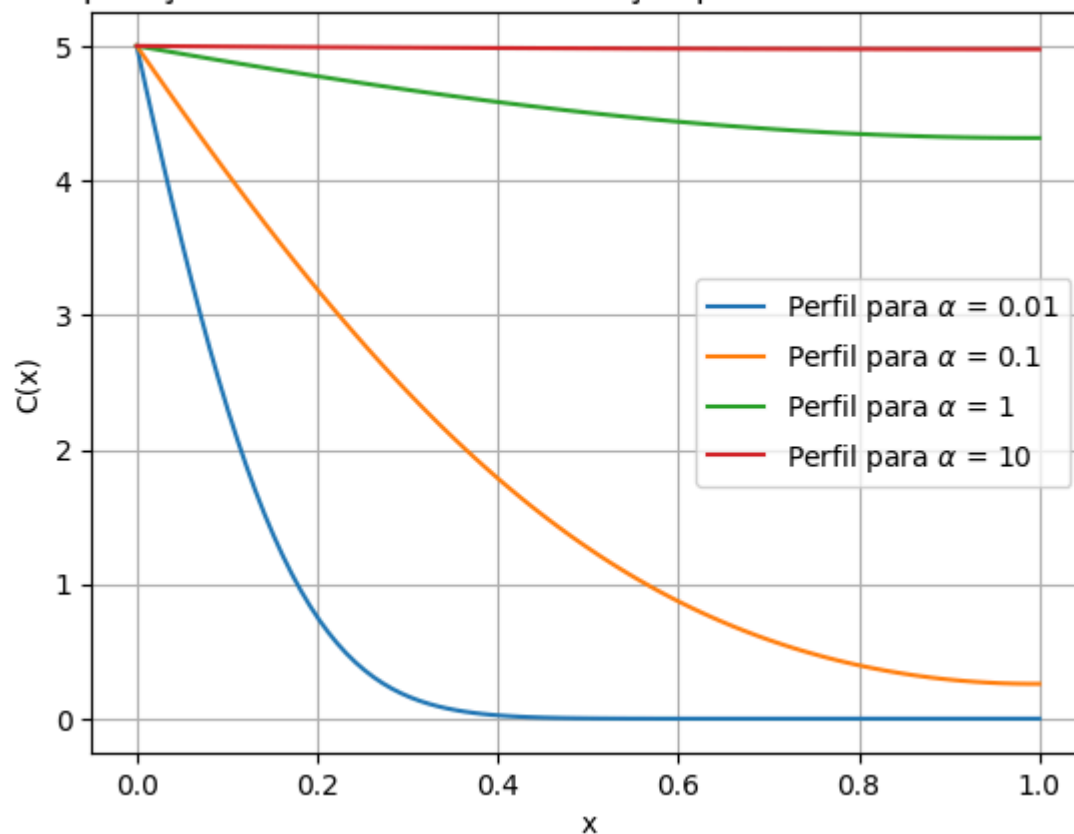
alpha = 1 # constante alpha
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para  $\alpha = \{alpha\}$ ')

alpha = 10 # constante alpha
C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
plt.plot(x, C, label=f'Perfil para  $\alpha = \{alpha\}$ ')

plt.xlabel('x')
plt.ylabel('C(x)')
plt.grid()
titulo = 'Comparação dos Perfis de Concentração para diferentes valores de  $\alpha$ '
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')
```



## Comparação dos Perfis de Concentração para diferentes valores de $\alpha$



É possível observar que maiores valores de  $\alpha$  difundem mais rapidamente a concentração como esperado.

## Concentração na fronteira esquerda do domínio

A condição de contorno (CE) influencia diretamente a concentração ( $C$ ) no ponto ( $x = 0$ ) ao longo do tempo. Especificamente:

- **Valor Inicial de Concentração:** (CE) define o valor inicial da concentração no ponto ( $x = 0$ ). Valores maiores de (CE) resultam em uma concentração inicial mais alta nesse ponto.
- **Distribuição de Concentração:** A concentração ( $C$ ) ao longo do domínio ( $L_x$ ) será afetada pela condição de contorno (CE). Um (CE) maior pode levar a uma concentração mais alta ao longo do domínio, dependendo dos parâmetros de difusão ( $\alpha$ ) e reação ( $k$ ).
- **Perfis de Concentração:** Diferentes valores de (CE) resultam em diferentes perfis de concentração ao longo do domínio. Isso pode ser observado nos gráficos de concentração, onde perfis com (CE) maiores tendem a ter valores de concentração mais altos.

Em resumo, (CE) é um parâmetro crucial que define a condição de contorno no ponto ( $x = 0$ ) e influencia a distribuição e os perfis de concentração ao longo do domínio.

```
In [187... # Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 100 # número de pontos no espaço
alpha = 0.01 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo
T = 1.0 # tempo total

# Condições de contorno
CE_values = np.arange(5, 36, 6) # valor de C em x = 0
```

```

valores_finais = []

for CE in CE_values:
    C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
    plt.plot(x, C, label=f'CE = {CE}')
    valores_finais.append(C[-1])

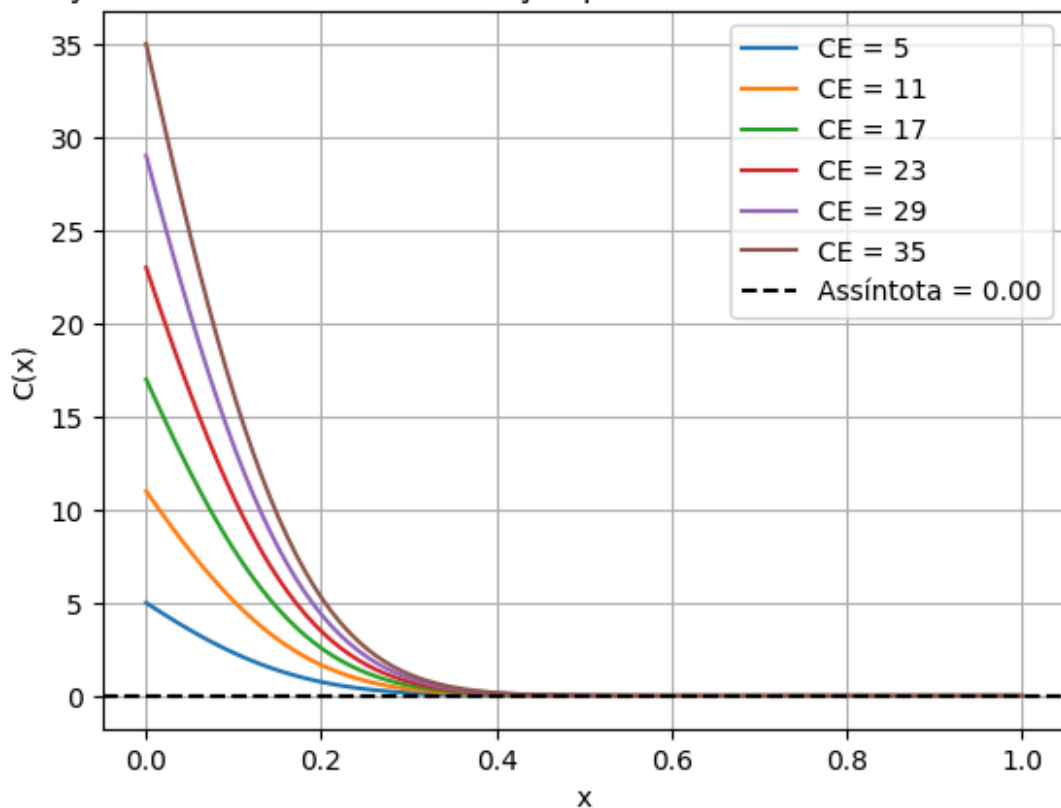
# Calculando a média dos valores finais
media_valores_finais = np.mean(valores_finais)

# Plotando a assíntota
plt.axhline(y=media_valores_finais, color='black', linestyle='--', label=f'Assíntota = {media_valores_finais}')

plt.xlabel('x')
plt.ylabel('C(x)')
plt.grid()
titulo = 'Comparação dos Perfis de Concentração para diferentes valores de CE em T = 1.0'
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')

```

Comparação dos Perfis de Concentração para diferentes valores de CE em T = 1.0



Observa-se que para todas as concentrações simuladas, a tendência era uma concentração 0 na fronteira direita do domínio. Entretanto, ao simular um intervalo maior de tempo, o comportamento muda.

In [186...

```

# Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 100 # número de pontos no espaço
alpha = 0.01 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo
T = 10.0 # tempo total

# Condições de contorno
CE_values = np.arange(5, 36, 6) # valor de C em x = 0

valores_finais = []

```

```

for CE in CE_values:
    C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
    plt.plot(x, C, label=f'CE = {CE}')
    valores_finais.append(C[-1])

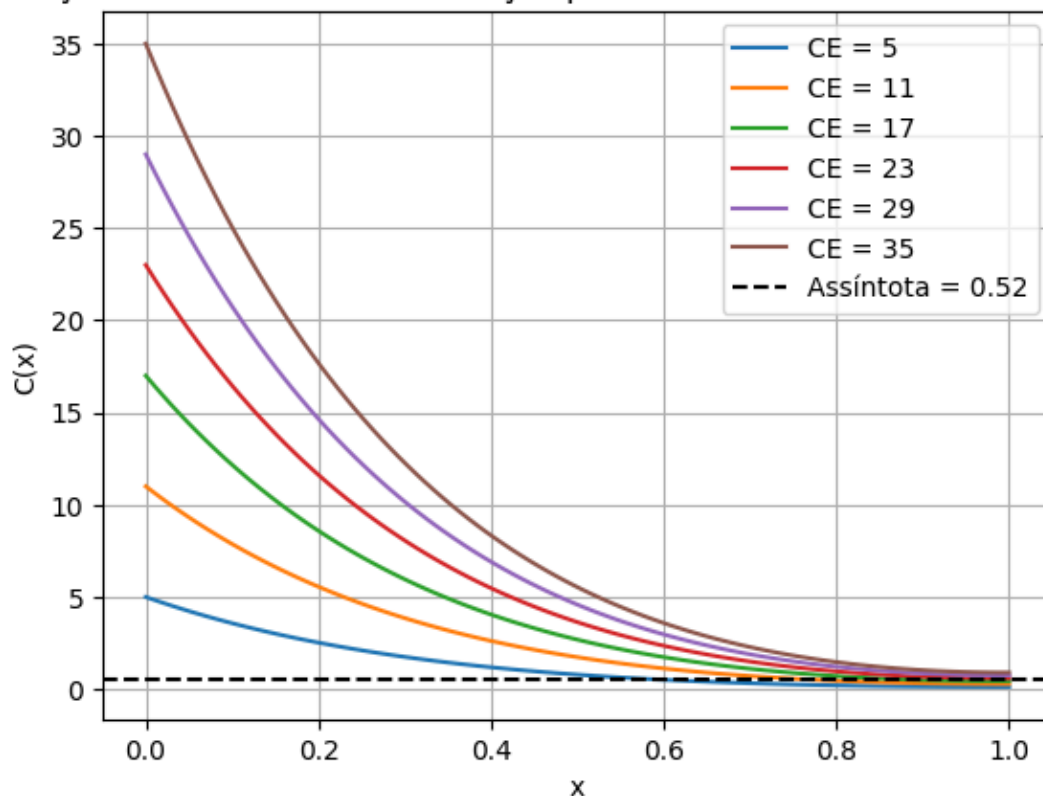
# Calculando a média dos valores finais
media_valores_finais = np.mean(valores_finais)

# Plotando a assíntota
plt.axhline(y=media_valores_finais, color='black', linestyle='--', label=f'Assíntota = {media_valores_finais}')

plt.xlabel('x')
plt.ylabel('C(x)')
plt.grid()
titulo = 'Comparação dos Perfis de Concentração para diferentes valores de CE em T = 10.0'
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')

```

Comparação dos Perfis de Concentração para diferentes valores de CE em T = 10.0



Isso se deve pois a tendência é que após determinado tempo, cada condição de (  $CE$  ) se torne um valor constante ao longo do domínio.

## Tempo de simulação

O tempo de simulação (  $T$  ) influencia diretamente a evolução da concentração (  $C$  ) ao longo do domínio (  $Lx$  ). Especificamente:

- **Evolução Temporal:** À medida que (  $T$  ) aumenta, a concentração (  $C$  ) tem mais tempo para evoluir, permitindo que os efeitos da difusão e da reação se manifestem mais plenamente.
- **Distribuição de Concentração:** Para tempos (  $T$  ) maiores, a concentração (  $C$  ) tende a se estabilizar, atingindo um estado estacionário onde as mudanças ao longo do tempo são mínimas.

- **Perfis de Concentração:** Diferentes valores de  $(T)$  resultam em diferentes perfis de concentração ao longo do domínio. Perfis para tempos menores podem mostrar variações mais acentuadas, enquanto perfis para tempos maiores tendem a ser mais suaves e uniformes.

Em resumo, o tempo de simulação  $(T)$  é um parâmetro crucial que define a duração da simulação e influencia a distribuição e os perfis de concentração ao longo do domínio.

```
In [ ]: # Definindo parâmetros
Lx = 1.0 # comprimento do domínio
Nx = 100 # número de pontos no espaço
alpha = 0.01 # constante alpha
k = 0.1 # constante k
dt = 0.01 # passo no tempo

# Condições de contorno
CE = 5 # valor de C em x = 0

# Condições de contorno
T_values = [1, 5, 15, 30, 45, 60, 120] # valores de T variando de 1 até 60

# Valores de T para calcular a média dos valores finais
T_values_assintota = [30, 45, 60, 120]

valores_finais = []

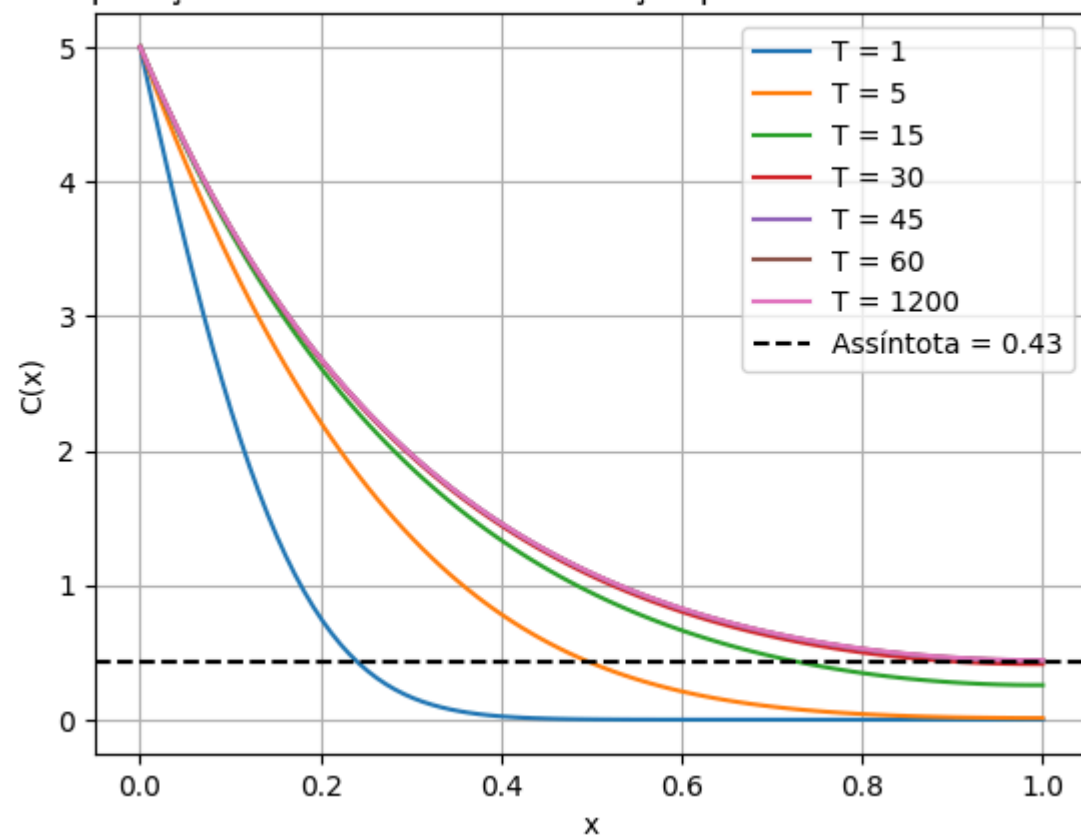
# Plotando os perfis de concentração para os valores de T
for T in T_values:
    C, x = funcao_C(T, Nx, Lx, alpha, k, dt, CE)
    plt.plot(x, C, label=f'T = {T}')
    if T in T_values_assintota:
        valores_finais.append(C[-1])

# Calculando a média dos valores finais
media_valores_finais = np.mean(valores_finais)

# Plotando a assíntota
plt.axhline(y=media_valores_finais, color='black', linestyle='--', label=f'Assíntota :

plt.xlabel('x')
plt.ylabel('C(x)')
plt.grid()
titulo = 'Comparação dos Perfis de Concentração para diferentes valores de T'
plt.title(titulo)
plt.legend()
# plt.show()
plt.savefig('docs/img/trab3/'+titulo+'.png')
```

## Comparação dos Perfis de Concentração para diferentes valores de T



A assíntota ( $y = 0.43$ ) observada ocorre porque, após um tempo suficiente, a concentração ( $C$ ) ao longo do domínio se estabiliza em um valor médio constante devido ao equilíbrio entre difusão e reação.