



Trabalho 1 de Métodos Numéricos para Equações Diferenciais II

Ariel Nogueira Kovaljski

Nova Friburgo, XX de setembro de 2020

Sumário

1	Introdução	2
1.1	A Equação de Advecção-Difusão	2
1.2	Método dos Volumes Finitos	3
2	Desenvolvimento	6
2.1	Condições Inicial e de Contorno	6
2.2	Consistência, Convergência e Estabilidade	7
2.2.1	Consistência	7
2.2.2	Convergência	7
2.2.3	Estabilidade	8
2.3	Programação	9
3	Resultados	11
3.1	Seção 1	11
3.2	Seção 2	11
3.3	Seção 3	11
4	Conclusão	12
5	Código Computacional	13

1. Introdução

Neste trabalho foi implementado um método computacional de maneira a resolver a equação de Advecção-Difusão de forma numérica.

Para melhor entender o desenvolvimento, é necessária introdução de alguns conceitos-chave utilizados.

1.1 A Equação de Advecção-Difusão

A equação de advecção-difusão possibilita a solução de problemas envolvendo variações espaciais e temporais da concentração de uma substância escoando em um fluido. Um exemplo bastante didático consiste no despejo de esgoto em um afluente: o contaminante sofrerá efeitos difusivos — concentrando-se ao redor da saída — e efeitos advectivos — sendo carregado no sentido da correnteza.



Figura 1.1: Efeitos difusivos e advectivos observados no despejo de esgoto no Rio Bengala

Para um problema unidimensional tem-se a seguinte forma,

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(uc) - \frac{\partial}{\partial x} \left(D \frac{\partial c}{\partial x} \right) = 0 \quad (1.1)$$

onde c indica a concentração, u a velocidade e D o coeficiente de difusão.

Considerando que para u e D constantes, tem-se \bar{u} e α , respectivamente, é possível reescrever Eq. 1.1 como,

$$\frac{\partial c}{\partial t} + \bar{u} \frac{\partial c}{\partial x} - \alpha \frac{\partial^2 c}{\partial x^2} = 0 \quad (1.2)$$

1.2 Método dos Volumes Finitos

O método dos volumes finitos tem como finalidade a discretização do domínio espacial. Este é subdividido em um conjunto de volumes finitos e as variáveis dependentes são determinadas como médias volumétricas sobre estes volumes, avaliadas nos centros dos mesmos.

A partir da Eq. 1.1, para adaptá-la ao métodos dos volumes finitos, é possível reescrevê-la como

$$\frac{\partial \Phi}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (1.3)$$

onde,

$$\Phi = c \quad (1.4) \quad f = f(c) = uc - D \frac{\partial c}{\partial x} \quad (1.5)$$

Para a solução, considera-se um domínio discretizado e subdividido conforme figuras abaixo:

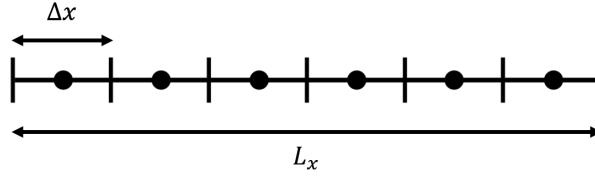


Figura 1.2: Partição do domínio da solução

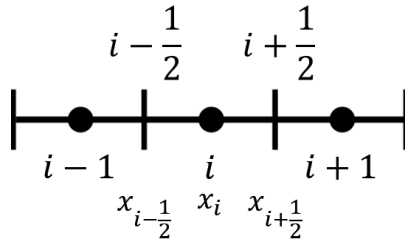


Figura 1.3: Posições ao redor do i -ésimo volume da malha

Para obter-se a discretização do domínio, integra-se a Eq. 1.3 no intervalo de tempo de t^n a t^{n+1} e no espaço de $x_{i-\frac{1}{2}}$ a $x_{i+\frac{1}{2}}$. Além disso, define-se os incrementos no tempo e no espaço como

$$\Delta t = t^{n+1} - t^n \quad \text{e} \quad \Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$$

Tem-se, assim, uma integração dupla

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\int_{t^n}^{t^{n+1}} \frac{\partial \Phi}{\partial t} dt \right) dx + \int_{t^n}^{t^{n+1}} \left(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f}{\partial x} dx \right) dt = 0$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} [\Phi(x, t^{n+1}) - \Phi(x, t^n)] dx + \int_{t^n}^{t^{n+1}} [f(x_{i+\frac{1}{2}}, t) - f(x_{i-\frac{1}{2}}, t)] dt = 0 \quad (1.6)$$

Define-se uma variável Q_i^k que consiste no valor médio aproximado de Φ no espaço, dado um tempo k onde $k = t^n$ ou $k = t^{n+1}$

$$Q_i^n \approx \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Phi(x, t^n) dx \quad (1.7)$$

$$Q_i^{n+1} \approx \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Phi(x, t^{n+1}) dx \quad (1.8)$$

Analogamente, define-se os fluxos como um F_k^n que consiste no valor médio aproximado de F no tempo, dada uma posição k onde $k = x_{i-\frac{1}{2}}$ ou $k = x_{i+\frac{1}{2}}$

$$F_{i-\frac{1}{2}}^n \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(x_{i-\frac{1}{2}}, t) dt \quad (1.9)$$

$$F_{i+\frac{1}{2}}^n \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(x_{i+\frac{1}{2}}, t) dt \quad (1.10)$$

Substituindo as Eq. 1.7, 1.8, 1.9 e 1.10 na Eq. 1.6 e dividindo todos os termos por Δx tem-se que

$$\frac{1}{\Delta x} \left\{ \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} [\Phi(x, t^{n+1}) - \Phi(x, t^n)] dx \right\} + \frac{1}{\Delta x} \left\{ \int_{t^n}^{t^{n+1}} [f(x_{i+\frac{1}{2}}, t) - f(x_{i-\frac{1}{2}}, t)] dt \right\} = 0$$

$$Q_i^{n+1} - Q_i^n + \frac{1}{\Delta x} \left\{ \int_{t^n}^{t^{n+1}} [f(x_{i+\frac{1}{2}}, t) - f(x_{i-\frac{1}{2}}, t)] dt \right\} = 0$$

$$Q_i^{n+1} = Q_i^n + \frac{\Delta t}{\Delta t \Delta x} \left\{ \int_{t^n}^{t^{n+1}} [f(x_{i+\frac{1}{2}}, t) - f(x_{i-\frac{1}{2}}, t)] dt \right\}$$

Por fim, obtém-se

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n) \quad (1.11)$$

onde o valor médio de Φ no volume de controle, Q , deve ser atualizado iterativamente para o próximo passo de tempo.

Para a aproximação dos fluxos, dentre as possíveis metodologias, foi escolhido o uso de uma aproximação *upwind* para a concentração c :

$$c \approx \frac{(Q_i^n - Q_{i-1}^n)}{\Delta x} \quad (1.12)$$

e uma aproximação centrada para a derivada espacial da concentração $\frac{\partial c}{\partial x}$:

$$\frac{\partial c}{\partial x} \approx \frac{(Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n)}{\Delta x^2} \quad (1.13)$$

que, ao serem substituídas na Eq. 1.11, resultam em

$$\begin{aligned} Q_i^{n+1} &= Q_i^n - \frac{\Delta t}{\Delta x} \left[\bar{u}Q_i^n - \alpha \frac{(Q_{i+1}^n - Q_i^n)}{\Delta x} - \bar{u}Q_{i-1}^n + \alpha \frac{(Q_i^n - Q_{i-1}^n)}{\Delta x} \right] \\ Q_i^{n+1} &= Q_i^n - \frac{\Delta t}{\Delta x} \left[\bar{u}(Q_i^n - Q_{i-1}^n) - \alpha \frac{(Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n)}{\Delta x} \right] \\ Q_i^{n+1} &= Q_i^n - \Delta t \left[\bar{u} \frac{(Q_i^n - Q_{i-1}^n)}{\Delta x} - \alpha \frac{(Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n)}{\Delta x^2} \right] \end{aligned} \quad (1.14)$$

2. Desenvolvimento

Neste capítulo serão abordados os passos e métodos utilizados para se obter a solução numérica do problema proposto.

2.1 Condições Inicial e de Contorno

A resolução de qualquer equação diferencial parcial (EDP) requer a determinação de sua condição(ões) inicial(ais) e de contorno. No caso da EDP discretizada (Eq. 1.14), o mesmo se aplica.

$$Q_i^{n+1} = Q_i^n - \Delta t \left[\bar{u} \frac{(Q_i^n - Q_{i-1}^n)}{\Delta x} - \alpha \frac{(Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n)}{\Delta x^2} \right]$$

Nota-se que há uma dependência temporal do termo futuro Q_i^{n+1} em relação aos termos Q presentes em n , e seus vizinhos espaciais i , $i \pm 1$. A partir desta relação é possível perceber que, para se calcular o primeiro termo, Q^1 , é necessário um termo de partida, Q^0 . Se tem, assim, a necessidade do estabelecimento de uma condição inicial. Neste trabalho, considera-se uma concentração inicial, c_{ini} , constante para toda a malha.

Mudando o foco para os volumes da malha, nos volumes $i = 0$ e $i = 6$ há uma dependência de termos localizados além do seu domínio — Q_0^n e Q_{nx+1}^n , respectivamente. Para resolver este problema, neste trabalho foram adotadas as seguintes condições de contorno:

$$c(x = 0, t) = c_{inj} \quad (2.1) \quad \left(\frac{\partial c}{\partial x} \right)_{x=L_x}^t = 0 \quad (2.2)$$

Aliado a estas condições, utiliza-se o conceito de *volumes fantasmas*, para a definição destas condições no discreto. É possível redefinir as condições de contorno como uma média entre os dois volumes adjacentes. Ao se realizar tal construção para a fronteira esquerda, obtém-se,

— *Inserir gráfico sobre células fantasmas* —

$$\begin{aligned} c_{inj} &= \frac{Q_0^n + Q_1^n}{2} \\ Q_0^n &= 2c_{inj} - Q_1^n \end{aligned} \quad (2.3)$$

Analogamente, para a fronteira direita, obtém-se,

$$\left(\frac{\partial c}{\partial x} \right)_{x=L_x}^t \approx \frac{Q_{nx+1}^n - Q_{nx}^n}{\Delta x} = 0$$

$$Q_{nx+1}^n = Q_{nx}^n \quad (2.4)$$

A partir de ambas as relações, definem-se as equações discretas. Para o contorno esquerdo,

$$\begin{aligned} Q_1^{n+1} &= Q_1^n - \Delta t \left[\bar{u} \frac{(Q_1^n - (2c_{\text{inj}} - Q_1^n))}{\Delta x} - \alpha \frac{(Q_2^n - 2Q_1^n + (2c_{\text{inj}} - Q_1^n))}{\Delta x^2} \right] \\ Q_1^{n+1} &= Q_1^n - \Delta t \left[\bar{u} \frac{(2Q_1^n - 2c_{\text{inj}})}{\Delta x} - \alpha \frac{(Q_2^n - 3Q_1^n + 2c_{\text{inj}})}{\Delta x^2} \right] \end{aligned} \quad (2.5)$$

e para o contorno direito,

$$\begin{aligned} Q_{nx}^{n+1} &= Q_{nx}^n - \Delta t \left[\bar{u} \frac{(Q_{nx}^n - Q_{nx-1}^n)}{\Delta x} - \alpha \frac{(Q_{nx}^n - 2Q_{nx}^n + Q_{nx-1}^n)}{\Delta x^2} \right] \\ Q_{nx}^{n+1} &= Q_{nx}^n - \Delta t \left[\bar{u} \frac{(Q_{nx}^n - Q_{nx-1}^n)}{\Delta x} - \alpha \frac{(Q_{nx-1}^n - Q_{nx}^n)}{\Delta x^2} \right] \end{aligned} \quad (2.6)$$

2.2 Consistência, Convergência e Estabilidade

A análise da consistência, convergência e estabilidade de uma EDP tem como finalidade garantir que a solução numérica do problema — calculada por algoritmos — se aproxime o máximo possível da solução real, com algumas observações.

2.2.1 Consistência

Se trata da equivalência da forma algorítmica da EDP em relação a sua forma analítica. Um método numérico é dito *consistente* quando, através de operações algébricas, é possível recuperar a EDP original.

2.2.2 Convergência

Se trata da aproximação dos valores numéricos do algoritmo à solução analítica da EDP, dado um certo número de iterações. Um método numérico é dito *convergente* quando este sempre irá tender aos valores da solução, não importando o número de iterações.

A análise direta da convergência de algoritmo é muito difícil, mesmo para os casos mais fáceis. Uma possível saída para esse problema é utilizar o Teorema da Equivalência de Lax, que diz:

“Para um problema linear de valor inicial bem-posto e um método de discretização consistente, estabilidade é condição necessária e suficiente para a convergência.” (Peter Lax)

Como a Eq. 1.14 foi obtida a partir da EDP analítica, é certa a sua consistência, restando assim, a determinação de sua estabilidade para a garantia da convergência.

2.2.3 Estabilidade

Se trata do comportamento do algoritmo e seus valores numéricos frente aos parâmetros de entrada. Um algoritmo *estável* se comporta de maneira esperada frente a uma faixa específica de valores de entrada.

Para se determinar a estabilidade da Eq. 1.14, é possível reescrevê-la para a análise através do método de Von Neumann.

$$(Q^*)_i^{n+1} = (s + C)(Q^*)_{i-1}^n + (1 - 2s + C)(Q^*)_i^n + s(Q^*)_{i+1}^n \quad (2.7)$$

onde Q^* é solução numérica da EDP, sujeita aos erros de arredondamento do computador, $s = \frac{\alpha \Delta t}{\Delta x^2}$ e $C = \frac{\bar{u} \Delta t}{\Delta x}$ (número de Courant). Além disso, define-se uma nova equação,

$$\xi_i^{n+1} = (s + C)\xi_{i-1}^n + (1 - 2s + C)\xi_i^n + s\xi_{i+1}^n \quad (2.8)$$

onde $\xi_i^n = Q_i^n - (Q^*)_i^n$ é o erro numérico introduzido em cada ponto da malha.

Segundo o método de Von Neumann, é possível expandir cada termo ξ_i^n em uma série de Fourier,

$$\xi_i^n = (G)^n e^{j\theta i} \quad (2.9)$$

onde j é a unidade imaginária e G é um fator de amplificação, de forma que,

$$\xi_i^{n+1} = G\xi_i^n \quad (2.10)$$

e a estabilidade é garantida se $|G| \leq 1$. Substituindo esses termos em Eq. 2.8, obtém-se a seguinte equação:

$$G^{n+1} e^{j\theta i} = (s + C)G^n e^{j\theta(i-1)} + (1 - 2s + C)G^n e^{j\theta i} + sG^n e^{j\theta(i+1)} \quad (2.11)$$

Dividindo-a por $G^n e^{j\theta i}$,

$$G = (s + C)e^{-j\theta} + (1 - 2s + C) + se^{j\theta}$$

A partir das relações de Euler, $e^{-j\theta} = \cos \theta - j \sin \theta$ & $e^{j\theta} = \cos \theta + j \sin \theta$, é possível reescrevê-la como,

$$G = (s + C)e^{-j\theta} + (1 - 2s + C) + se^{j\theta}$$

$$G = 1 - (2s + C)(1 - \cos \theta) - jC \sin \theta$$

tendo assim,

$$|G| = \sqrt{[1 - (2s + C)(1 - \cos \theta)]^2 + C^2 \sin^2 \theta} \leq 1 \quad (2.12)$$

O caso limite para esta construção ocorre quando $\sin \theta = \pm 1$ e $\cos \theta = 0$. É necessário analisar quais são os valores de s e C que mantém a veracidade da inequação.

$$|G| = \sqrt{[1 - (2s + C)(1)]^2 + C^2} \leq 1$$

Se $2s + C \leq 1$ a condição é satisfeita. O que acontece, porém, no novo caso limite de $2s + C = 1$?

$$|G| = \sqrt{(1 - 1)^2 + C^2} = \sqrt{C^2} = C$$

Como $2s + C = 1$, C tem que ser menor que 1 para um $s \neq 0$. Desta forma,

$$\begin{aligned} 2\frac{\alpha\Delta t}{\Delta x^2} + \frac{\bar{u}\Delta t}{\Delta x} &\leq 1 \\ \Delta t \left(\frac{2\alpha}{\Delta x^2} + \frac{\bar{u}\Delta t}{\Delta x} \right) &\leq 1 \\ \Delta t &\leq \frac{1}{\frac{2\alpha}{\Delta x^2} + \frac{\bar{u}}{\Delta x}} \end{aligned} \quad (2.13)$$

Tem-se, então, um método *condicionalmente estável*, ou seja, que depende de uma faixa de valores para garantir a estabilidade de seu funcionamento.

2.3 Programação

Aliado destes conceitos, foi possível construir um programa em linguagem C que calcula as concentrações para cada célula ao longo do tempo, exportando um arquivo de texto com os resultados; este arquivo, então, é lido por um *script* Python que gera um gráfico correspondente.

O programa principal possui a seguinte estrutura, descrita em pseudo-código:

```
// Vetores para concentração no tempo 'n' e no tempo 'n+1', respectivamente
vetor Q_antigo
vetor Q_novo

// Inicializa ambos os vetores com os valores da concentração inicial (c_ini)
inicializaVetor(Q_antigo)
inicializaVetor(Q_novo)

// Cálculo de Q, iterado ao longo do tempo
faça {

// Cálculo do volume da fronteira esquerda
Q_novo[0] := Q_antigo[0] - Delta_t/Delta_x * ( u_bar * (2*Q_antigo[0] - 2*c_inj)
          - alpha * (Q_antigo[1] - 3*Q_antigo[0] + 2*c_inj) / Delta_x)

// Cálculo de volumes do centro da malha
para (x := 1 até nx - 1) {
    Q_novo[x] := Q_antigo[x] - Delta_t/Delta_x * (u_bar * (Q_antigo[x]
          - Q_antigo[x-1]) - alpha * (Q_antigo[x+1] - 2*Q_antigo[x]
          - Q_antigo[x-1]) / Delta_x )
}

// Cálculo do volume da fronteira direita
Q_novo[x] := Q_antigo[x] - Delta_t/Delta_x * ( u_bar * (Q_antigo[x] -
          Q_antigo[x-1]) - alpha
          * (Q_antigo[x-1] - Q_antigo[x]) / Delta_x)

// Atualiza vetores antigos para a próxima iteração
para (x := 0 até nx) {
```

```
        Q_antigo[x] = Q_novo[x];  
    }  
  
    // Incrementa passo de tempo  
    t := t + Delta_t  
} enquanto (t < t_final)
```

3. Resultados

Os resultados entram aqui.

3.1 Seção 1

A seção 1 entra aqui.

3.2 Seção 2

A seção 2 entra aqui.

3.3 Seção 3

A seção 3 entra aqui.

4. Conclusão

As conclusões entram aqui.

5. Código Computacional

O código computacional entra aqui.