

# Volumes Finitos e Algoritmos de Acoplamento

## Tópicos Especiais em Sistemas Térmicos: CFD

Professor: Adriano Possebon Rosa

Laboratório de Energia e Ambiente  
Departamento de Engenharia Mecânica  
Universidade de Brasília

Nesta aula, o nosso objetivo é entender um pouco melhor como funciona o processo de discretizar e resolver as equações governantes da mecânica dos fluidos e da transferência de calor.

Vamos começar com uma discussão sobre os métodos de discretização mais utilizados na engenharia. Depois, vamos dar uma olhada no sistema de equações algébricas resultante e em como resolvê-lo. Iremos abordar alguns exemplos simples.

Por fim, vamos ver os algoritmos de acoplamento pressão-velocidade mais utilizados em CFD e vamos fechar com uma análise de como tudo isso é implementado no OpenFOAM.

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

Frequentemente (muito frequentemente), as Equações Diferenciais Parciais (EDPs) que governam problemas de dinâmica dos fluidos são **não-lineares** e **acopladas**.

Soluções analíticas são 'impossíveis' de se obter em muitos casos. Portanto, as equações precisam ser discretizadas, transformadas em um conjunto de equações algébricas e resolvidas numericamente.

Os métodos de discretização mais importantes são

- o Método das Diferenças Finitas (FDM),
- o Método dos Elementos Finitos (FEM) e
- o Método dos Volumes Finitos (FVM).

O **FDM** é um dos métodos mais antigos para discretizar EDPs. Ele aproxima as derivadas parciais com expansões de Taylor.

Os nós são espaçados em uma grade quadrada e as equações são resolvidas nos nós.

O FDM funciona bem para geometrias simples com malhas estruturadas, mas as expansões de Taylor precisam ser ajustadas para formas mais irregulares.

O **FEM** é usado em alguns softwares de CFD, como o COMSOL Multiphysics, onde o domínio é dividido em elementos.

Em 2D, os elementos são frequentemente triangulares, com 3 nós associados a cada elemento. As variáveis desconhecidas são calculadas nos nós e usadas para aproximar os valores dentro dos elementos.

O FEM funciona tanto para malhas não estruturadas quanto para geometrias complexas. A desvantagem deste método é que ele é uma abordagem altamente matemática, e as equações algébricas obtidas não têm significado físico.

O **FVM** é o método mais usado em softwares de CFD (OpenFOAM e ANSYS Fluent, por exemplo).

Pontos positivos do FVM.

1) Flexibilidade Geométrica e Adaptabilidade de Malhas.

O FVM pode ser aplicado a malhas estruturadas e não estruturadas, oferecendo flexibilidade na representação de geometrias complexas. Além disso, é eficaz para malhas adaptativas, permitindo a alocação eficiente de recursos computacionais.

## 2) Tratamento Natural de Condições de Contorno e Interfaces

O FVM lida de forma natural com condições de contorno complexas e interfaces entre diferentes meios, permitindo a modelagem eficaz de problemas com geometrias complicadas e múltiplas fases.

## 3) Conservação de Quantidades Físicas.

O FVM conserva quantidades físicas, como massa, energia e quantidades extensivas, tornando-o ideal para simulações precisas de fenômenos físicos onde a conservação é crucial.



Pontos negativos do FVM.

1) Interpolação e Erros de Discretização e Custo Computacional em 3D.

A precisão do FVM depende das técnicas de interpolação usadas para calcular os fluxos nas faces dos volumes, podendo introduzir erros de discretização, especialmente em malhas não uniformes. Além disso, em problemas 3D, o FVM pode ser computacionalmente intensivo, exigindo recursos significativos de computação.

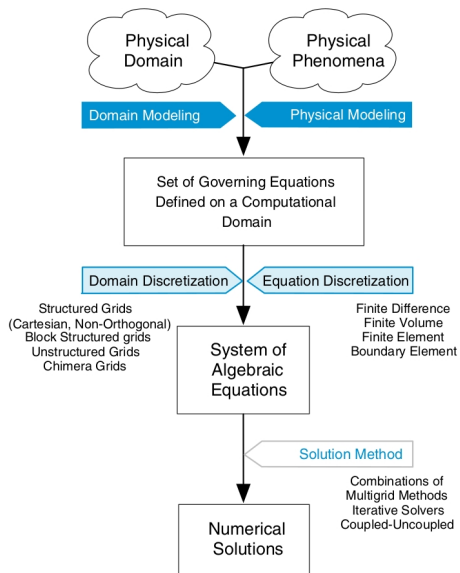
## 2) Perda de Detalhes em Malhas Grosseiras e Difusão Numérica.

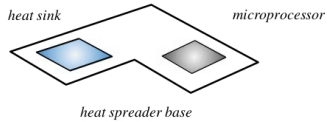
Em malhas grosseiras, detalhes finos do fluxo podem ser perdidos devido à representação média das propriedades dentro dos volumes. Além disso, o FVM pode introduzir difusão numérica, levando a resultados imprecisos, especialmente em gradientes abruptos ou fronteiras.

Um outro ponto positivo, em **FVM**, é que todos os termos que são aproximados têm um significado físico.

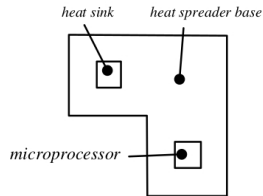
Cada nó é cercado por um pequeno volume. As equações diferenciais são integradas sobre cada volume e transformadas em integrais de superfície.

As variáveis desconhecidas são armazenadas em cada nó (centro da célula) e a interpolação linear é usada para aproximar os fluxos no centro de cada face.

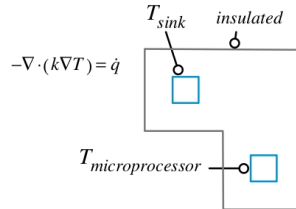




## Domain Modeling

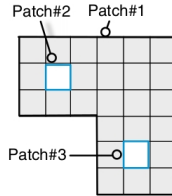


## Physical Modeling

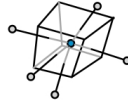




## Domain Discretization



## Equations Discretization

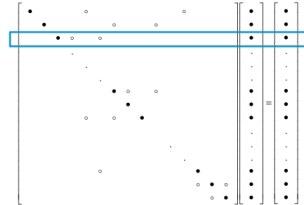


$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\text{transient term}} + \underbrace{\nabla \cdot (\rho \mathbf{v} \phi)}_{\text{convection term}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{diffusion term}} + \underbrace{Q}_{\text{source term}}$$

$$a_C \phi_C + \sum_{F \sim \text{NB}(C)} a_F \phi_F = b_C$$



## Solution Method



## Comentários.

- O objetivo de todo método numérico em CFD é transformar algumas poucas EPDs extremamente complexas em milhares (ou milhões ou bilhões) de equações algébricas simples.
- No final, independente do método, sempre temos que resolver um sistema de equações algébricas.
- Nesta aula eu vou apresentar de forma bem introdutória o FDM e os métodos de solução de equações algébricas. Eu recomendo, para quem quiser ver mais detalhes sobre esses temas, as aulas 04 e 05 do meu curso [Métodos Numéricos em Termofluidos](#).

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

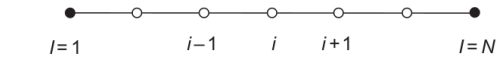


Vamos ver alguns detalhes dos métodos FDM e FVM.

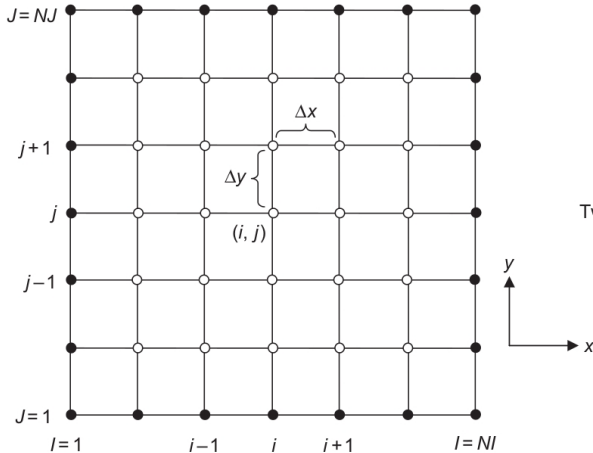
Sobre o FEM eu não sei nada, então esse eu vou pular.

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo



One-dimensional



Two-dimensional

Expansão em série de Taylor para uma variável  $\phi = \phi(x, y)$  no ponto  $(i + 1, j)$  em torno do ponto  $(i, j)$ :

$$\phi_{i+1,j} = \phi_{i,j} + \left( \frac{\partial \phi}{\partial x} \right)_{i,j} \Delta x + \left( \frac{\partial^2 \phi}{\partial x^2} \right)_{i,j} \frac{\Delta x^2}{2} + \left( \frac{\partial^3 \phi}{\partial x^3} \right)_{i,j} \frac{\Delta x^3}{3!} + \dots \quad (1)$$

No ponto  $(i - 1, j)$ :

$$\phi_{i-1,j} = \phi_{i,j} - \left( \frac{\partial \phi}{\partial x} \right)_{i,j} \Delta x + \left( \frac{\partial^2 \phi}{\partial x^2} \right)_{i,j} \frac{\Delta x^2}{2} - \left( \frac{\partial^3 \phi}{\partial x^3} \right)_{i,j} \frac{\Delta x^3}{3!} + \dots \quad (2)$$

Podemos aproximar a derivada de  $\phi$  com relação a  $x$  em  $(i, j)$  usando a equação (1):

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} + \left(\frac{\partial^2 \phi}{\partial x^2}\right)_{i,j} \frac{\Delta x}{2} + \left(\frac{\partial^3 \phi}{\partial x^3}\right)_{i,j} \frac{\Delta x^2}{3!} + \dots \quad (3)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} + O(\Delta x) \quad (4)$$

A expressão  $O(\Delta x)$  representa o erro de truncamento da aproximação por diferenças finitas, o qual mede a precisão da aproximação e determina a taxa na qual o erro diminui à medida que o espaçamento entre os pontos é reduzido.

Desprezando esse termo, temos:

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} \approx \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \quad (5)$$

Essa aproximação é chamada de diferença avançada (*forward difference*).

Outra forma de aproximar a derivada de  $\phi$  com relação a  $x$  em  $(i, j)$  é usando a equação (2):

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} + \left(\frac{\partial^2 \phi}{\partial x^2}\right)_{i,j} \frac{\Delta x}{2} - \left(\frac{\partial^3 \phi}{\partial x^3}\right)_{i,j} \frac{\Delta x^2}{3!} + \dots \quad (6)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} + O(\Delta x) \quad (7)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} \approx \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} \quad (8)$$

Essa aproximação é chamada de diferença atrasada (*backward difference*).



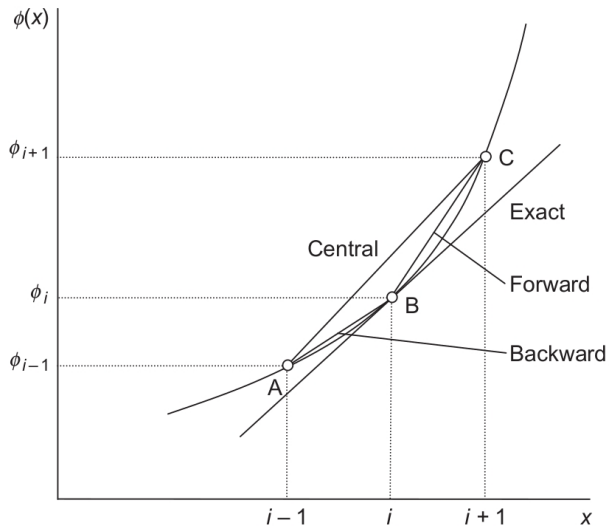
Podemos ainda usar uma combinação das equações (1) e (2) para obter uma aproximação diferente da derivada de  $\phi$  em  $(i, j)$ . Faça equação (1) menos a equação (2). Isolando a derivada, resulta

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{\Delta x} + \left(\frac{\partial^3 \phi}{\partial x^3}\right)_{i,j} \frac{\Delta x^2}{3} + \dots \quad (9)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} + O(\Delta x^2) \quad (10)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{i,j} \approx \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} \quad (11)$$

Essa aproximação é chamada de diferença centrada (*central difference*).



Para a aproximação da derivada segunda, temos

$$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_{i,j} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2} + O(\Delta x^2) \quad (12)$$

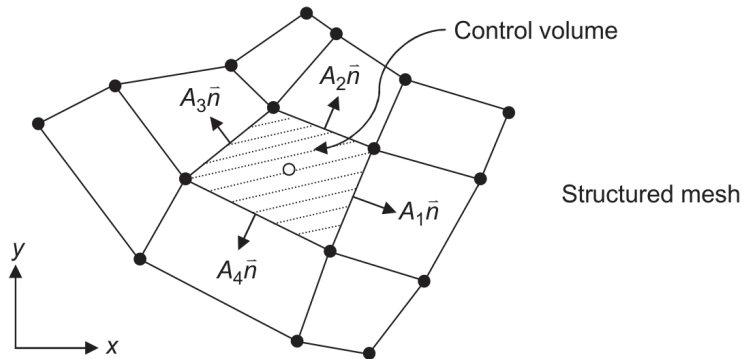
Quando o problema envolve derivada no tempo, também podemos aproximar essas derivadas usando séries de Taylor.

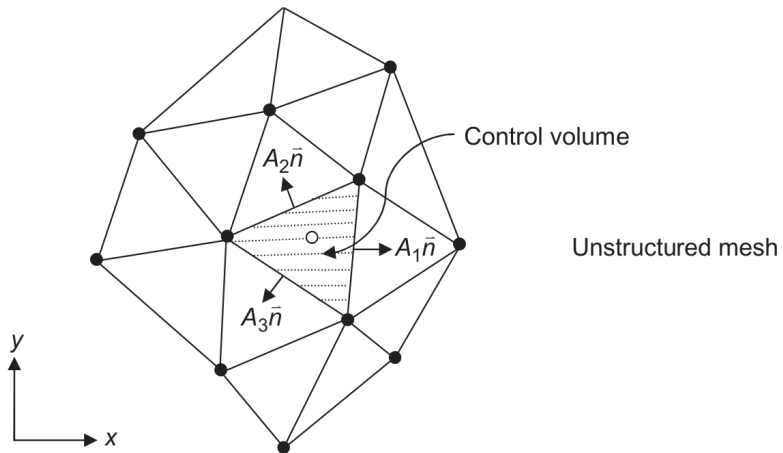
$$\left(\frac{\partial \phi}{\partial t}\right)_{i,j}^n = \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t} + O(\Delta t) \quad (13)$$

$$\left(\frac{\partial \phi}{\partial t}\right)_{i,j}^n \approx \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t} \quad (14)$$

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - **FVM**
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo







Os valores das variáveis são armazenados no centroide de cada um dos volumes de controle.

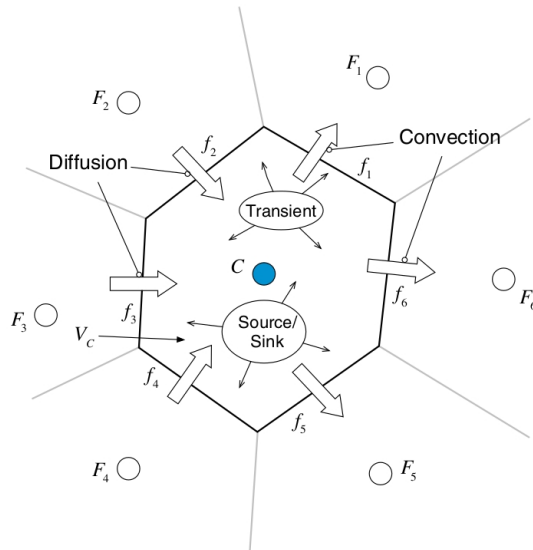
Uma interpolação é usada para expressar os valores das variáveis na superfície do volume de controle em termos dos valores no centro, e fórmulas de quadratura adequadas são aplicadas para aproximar as integrais de superfície e volume.

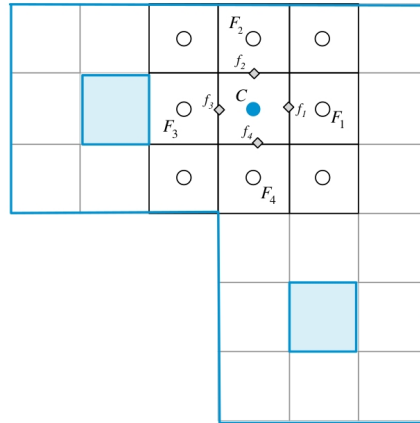
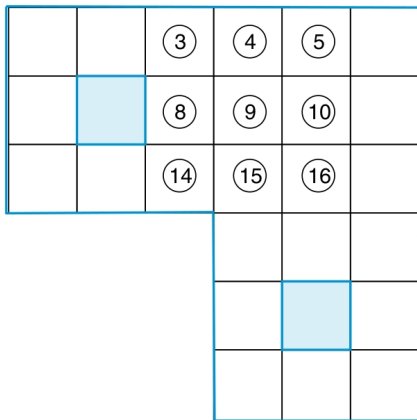
Uma equação algébrica para cada um dos volumes de controle pode ser obtida, na qual aparecem os valores dos vizinhos.

O método é conservativo, contanto que as integrais de superfície calculadas em uma face sejam iguais nos volumes de controle que compartilham essa face.

Uma desvantagem deste método, em comparação com os esquemas de diferenças finitas, é que aproximações de diferenciação de ordem superior à segunda são mais difíceis de desenvolver em três dimensões.

Isso se deve à necessidade de dois níveis de aproximação, que são: interpolação e integração.





$$-\nabla \cdot (k \nabla T) = \dot{q} \quad (15)$$

$$-\int_{VC} \nabla \cdot (k \nabla T) dV = \int_{VC} \dot{q} dV \quad (16)$$

$$-\int_{SC} (k \nabla T) \cdot d\mathbf{S} = \dot{q}_C V \quad (17)$$

$$-\sum_f (k \nabla T)_f \cdot \mathbf{S}_f = \dot{q}_C V \quad (18)$$

$$\begin{aligned}
 & - (k \nabla T)_{f_1} \cdot \mathbf{S}_{f_1} - (k \nabla T)_{f_2} \cdot \mathbf{S}_{f_2} \\
 & - (k \nabla T)_{f_3} \cdot \mathbf{S}_{f_3} - (k \nabla T)_{f_4} \cdot \mathbf{S}_{f_4} = \dot{q}_C V
 \end{aligned} \tag{19}$$

$$\begin{aligned}
 \mathbf{S}_{f_1} &= \Delta y_{f_1} \mathbf{i} \\
 \delta x_{f_1} &= x_{F_1} - x_C \\
 (\nabla T)_{f_1} &= \left( \frac{\partial T}{\partial x} \right)_{f_1} \mathbf{i} + \left( \frac{\partial T}{\partial y} \right)_{f_1} \mathbf{j}
 \end{aligned} \tag{20}$$

$$(\nabla T)_{f_1} \cdot \mathbf{S}_{f_1} = \left( \frac{\partial T}{\partial x} \right)_{f_1} \Delta y_{f_1} \quad (21)$$

$$(\nabla T)_{f_1} \cdot \mathbf{S}_{f_1} = \frac{T_F - T_C}{\delta x_{f_1}} \quad (22)$$

$$- (k \nabla T)_{f_1} \cdot \mathbf{S}_{f_1} = -k_{f_1} \left( \frac{T_F - T_C}{\delta x_{f_1}} \right)_{f_1} \Delta y_{f_1} = a_{F_1} (T_F - T_C) \quad (23)$$

$$- (k \nabla T)_{f_1} \cdot \mathbf{S}_{f_1} = a_{F_1} (T_F - T_C) \quad (24)$$

$$a_{F_1} = -k \frac{\Delta y_{f_1}}{\delta x_{f_1}} \quad (25)$$

$$a_{F_2} = -k_{f_2} \frac{\Delta x_{f_2}}{\delta y_{f_2}} \quad (26)$$

$$a_{F_3} = -k_{f_3} \frac{\Delta y_{f_3}}{\delta x_{f_3}} \quad (27)$$

$$a_{F_4} = -k_{f_4} \frac{\Delta x_{f_4}}{\delta y_{f_4}} \quad (28)$$



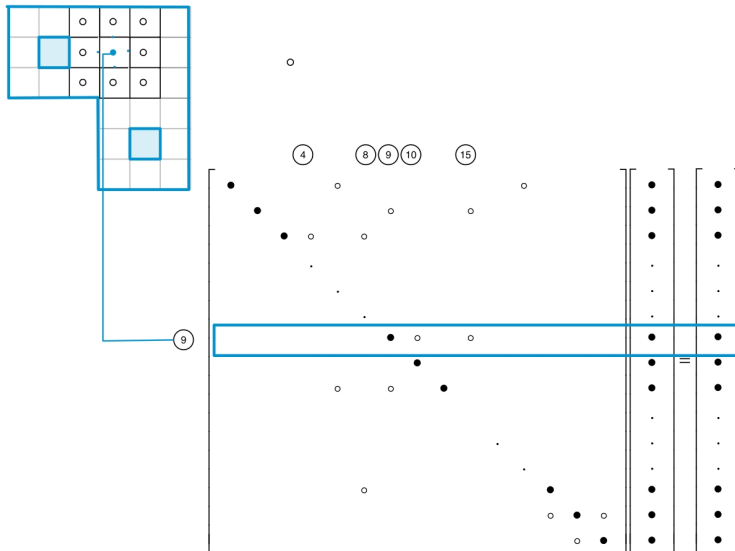
$$\begin{aligned} - \sum_f (k \nabla T)_f \cdot \mathbf{S}_f &= -(a_{F_1} + a_{F_2} + a_{F_3} + a_{F_4})T_C \\ &+ a_{F_2}T_{F_2} + a_{F_3}T_{F_3} + a_{F_4}T_{F_4} = \dot{q}_C V \end{aligned} \quad (29)$$

Ou, na forma final:

$$a_C T_C + \sum_i a_i T_{F_i} = b_C \quad (30)$$

$$a_C = -(a_{F_1} + a_{F_2} + a_{F_3} + a_{F_4}) \quad (31)$$

$$b_C = \dot{q}_C V \quad (32)$$



Montamos um sistema de equações algébricas da forma

$$\mathbf{A}\mathbf{T} = \mathbf{b} , \quad (33)$$

em que  $\mathbf{A}$  é a matriz dos coeficientes,  $\mathbf{T}$  é o vetor das variáveis desconhecidas e  $\mathbf{b}$  é o vetor fonte.

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM**
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

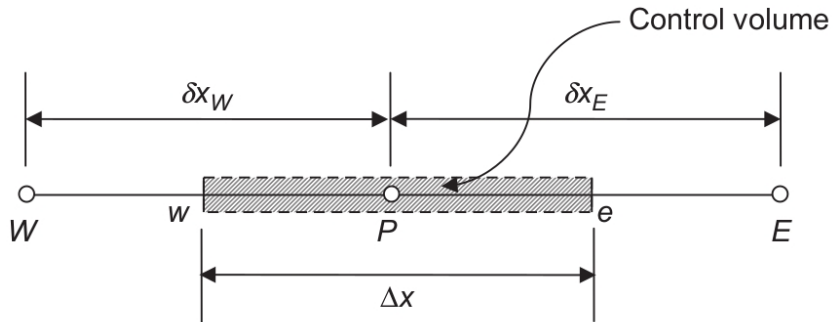
Como exemplo, para podermos ver na prática a aplicação do método de volumes finitos, vamos considerar o processo puramente difusivo de uma propriedade  $\phi$  em um domínio unidimensional.

A equação governante para este problema é

$$\frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) + S_\phi = 0 , \quad (34)$$

em que  $\Gamma$  é o coeficiente de difusão e  $S_\phi$  é o termo fonte.

Essa é uma equação que aparece no problema de condução de calor com geração de calor.



Pontos nodais:  $W$ ,  $P$  e  $E$ .

Faces:  $w$  e  $e$ .

Volume do elemento  $P$ :  $\Delta V = \Delta x \Delta y \Delta z$ . Área da face  $w$ :  $A_w = \Delta y \Delta z$ .  
 Área da face  $e$ :  $A_e = \Delta x \Delta z$ .

Vamos integrar a equação governante no volume que tem  $P$  como centro. Esse volume tem faces  $e$  e  $w$ .

Integrando a equação, temos

$$\int_{\Delta V} \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) dV + \int_{\Delta V} S_\phi dV = 0 . \quad (35)$$

$$\int_{\Delta V} \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) dV = \int_A \left( \Gamma \frac{\partial \phi}{\partial x} \right) dA^x \approx \sum_{i=1}^2 \left( \Gamma \frac{\partial \phi}{\partial x} \right)_i A_i^x \quad (36)$$

$$\sum_{i=1}^2 \left( \Gamma \frac{\partial \phi}{\partial x} \right)_i A_i^x = \left( \Gamma \frac{\partial \phi}{\partial x} \right)_e A_e - \left( \Gamma \frac{\partial \phi}{\partial x} \right)_w A_w \quad (37)$$

$$\int_{\Delta V} S_\phi dV \approx S_P \Delta V \quad (38)$$



$$\left(\Gamma \frac{\partial \phi}{\partial x}\right)_e A_e - \left(\Gamma \frac{\partial \phi}{\partial x}\right)_w A_w + S_P \Delta V = 0 \quad (39)$$

$$\left(\Gamma \frac{\partial \phi}{\partial x}\right)_e A_e \approx \Gamma_e A_e \left(\frac{\phi_E - \phi_P}{\delta x_E}\right) \quad (40)$$

$$\left(\Gamma \frac{\partial \phi}{\partial x}\right)_w A_w \approx \Gamma_w A_w \left(\frac{\phi_P - \phi_W}{\delta x_W}\right) \quad (41)$$

$$\Gamma_e A_e \left( \frac{\phi_E - \phi_P}{\delta x_E} \right) - \Gamma_w A_w \left( \frac{\phi_P - \phi_W}{\delta x_W} \right) + S_P \Delta V = 0 \quad (42)$$

$$a_E = -\frac{\Gamma_e A_e}{\Delta V \delta x_E} \quad a_W = -\frac{\Gamma_w A_w}{\Delta V \delta x_W} \quad (43)$$

$$a_P = \frac{\Gamma_e A_e}{\Delta V \delta x_E} + \frac{\Gamma_w A_w}{\Delta V \delta x_W} = -(a_E + a_W) \quad (44)$$

$$b = S_P \quad (45)$$

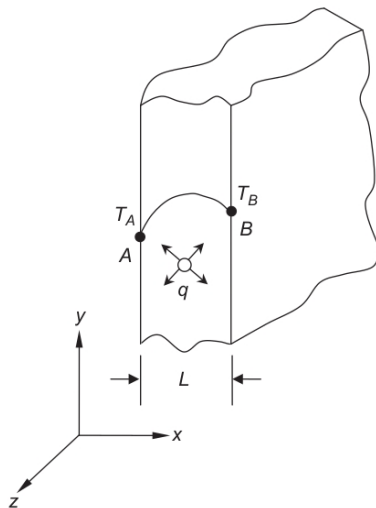
$$\boxed{a_P \phi_P + a_E \phi_E + a_W \phi_W = b} \quad (46)$$

Considerando essa equação governante e metodologia, vamos resolver um problema completo.

Considere o problema de condução de calor unidimensional e permanente em uma parede com geração uniforme de calor. A equação governante é dada por

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + q = 0 , \quad (47)$$

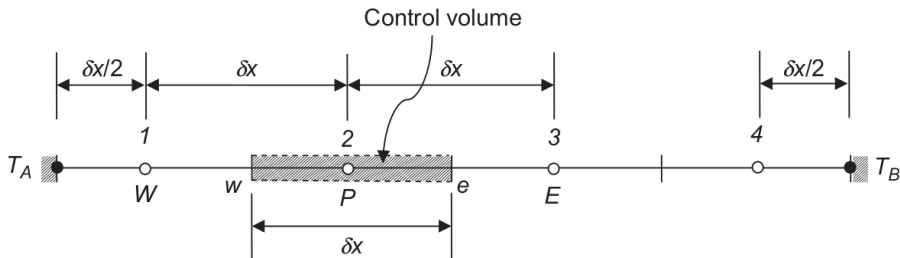
em que  $k = 5 \text{ W}/(\text{m} \cdot ^\circ\text{C})$  e  $q = 500 \text{ kW}/\text{m}^3$ . A parede tem espessura  $L = 2 \text{ cm}$  e as temperaturas nas duas extremidades são  $100^\circ\text{C}$  e  $400^\circ\text{C}$ .



Condições de contorno:  $T(x = 0) = T_A$  e  $T(x = L) = T_B$ .

Vamos dividir o domínio em 4 volumes de controle.

$$\delta x = 0.005 \text{ m}$$



Resolver esse problema significa encontrar o valor da temperatura nos 4 pontos (que são os centros de seus volumes de controle), ou seja, encontrar  $T_1$ ,  $T_2$ ,  $T_3$  e  $T_4$ .

Para os pontos 2 e 3, podemos usar a fórmula geral. Começando pelo ponto 2:

$$a_2 T_2 + a_3 T_3 + a_1 T_1 = b_2 \quad (48)$$

$$a_1 = -\frac{kA}{A\delta x^2} = a_3 = -2 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (49)$$

$$a_2 = 4 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (50)$$

$$b_2 = 5 \times 10^5 \text{ W}/\text{m}^3 \quad (51)$$

Para o ponto 3:

$$a_3 T_3 + a_4 T_4 + a_2 T_2 = b_3 \quad (52)$$

$$a_2 = -\frac{kA}{A\delta x^2} = a_4 = -2 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (53)$$

$$a_3 = 4 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (54)$$

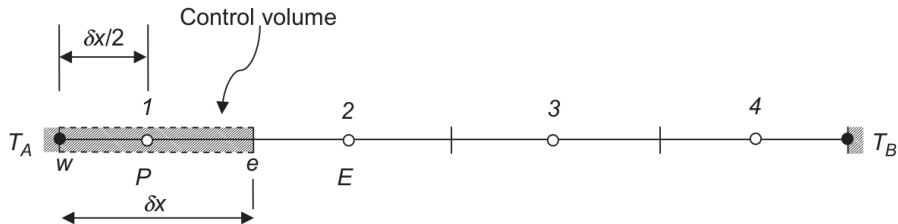
$$b_3 = 5 \times 10^5 \text{ W}/\text{m}^3 \quad (55)$$

Devemos tomar cuidado com os volumes 1 e 4, pois esses volumes têm uma face que está na fronteira do domínio.

As condições de contorno serão aplicadas a partir das equações para esses dois volumes.

Vamos considerar primeiro o volume 1.





Abrindo a equação neste volume, temos:

$$\frac{k}{\delta x} \left( \frac{\partial T}{\partial x} \right)_e - \frac{k}{\delta x} \left( \frac{\partial T}{\partial x} \right)_w + q = 0 \quad (56)$$

$$\frac{k}{\delta x} \left( \frac{T_2 - T_1}{\delta x} \right) - \frac{k}{\delta x} \left( \frac{T_1 - T_A}{\delta x/2} \right) + q = 0 \quad (57)$$

$$a_1 T_1 + a_2 T_2 = b_1 \quad (58)$$

$$a_1 = \frac{3k}{\delta x^2} = 6 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (59)$$

$$a_2 = -\frac{k}{\delta x^2} = -2 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (60)$$

$$b_3 = q + \frac{2kT_A}{\delta x^2} = 4.05 \times 10^7 \text{ W}/\text{m}^3 \quad (61)$$

Fazendo uma análise semelhante para o volume 4, temos

$$a_4 T_4 + a_3 T_3 = b_4 \quad (62)$$

$$a_4 = \frac{3k}{\delta x^2} = 6 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (63)$$

$$a_3 = -\frac{k}{\delta x^2} = -2 \times 10^5 \text{ W}/(\text{m}^3 \cdot ^\circ\text{C}) \quad (64)$$

$$b_4 = q + \frac{2kT_B}{\delta x^2} = 1.605 \times 10^8 \text{ W}/\text{m}^3 \quad (65)$$

Juntando as 4 equações (e dividindo os valores por  $10^5$ ), temos o seguinte sistema de equações algébricas:

$$\begin{bmatrix} 6 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 6 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 405 \\ 5 \\ 5 \\ 1605 \end{bmatrix} \quad (66)$$

Resolvendo:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 140 \\ 217.5 \\ 292.5 \\ 365 \end{bmatrix} \quad (67)$$

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão**
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

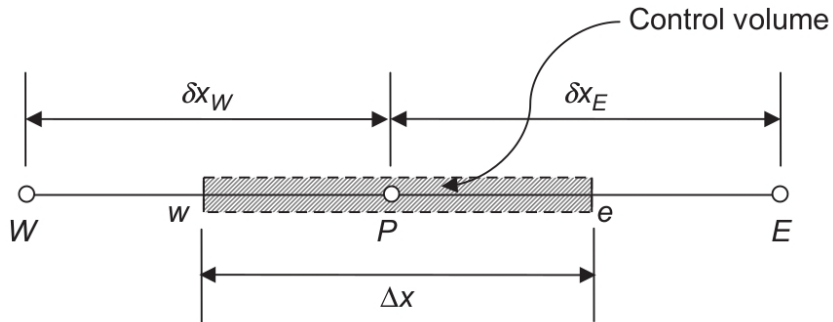
Seja o problema permanente de advecção-difusão.

Em uma dimensão, a equação governante desse processo para uma variável  $\phi$  em um campo de velocidade  $u$  é dada por

$$\frac{\partial}{\partial x} (\rho u \phi) = \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) , \quad (68)$$

em que  $\rho$  é a densidade e  $\Gamma$  é o coeficiente de difusão.

Note que nós temos dois fluxos: o fluxo advectivo e o fluxo difusivo.





Integrando essa equação e usando o teorema da divergência, nós temos

$$(\rho u \phi)_e A_e - (\rho u \phi)_w A_w = \left( \Gamma \frac{\partial \phi}{\partial x} \right)_e A_e - \left( \Gamma \frac{\partial \phi}{\partial x} \right)_s A_s \quad (69)$$

$$\begin{aligned} & (\rho u)_e \phi_e A_e - (\rho u)_w \phi_w A_w = \\ & = \Gamma_e A_e \left( \frac{\phi_E - \phi_P}{\delta x_E} \right) - \Gamma_w A_w \left( \frac{\phi_P - \phi_W}{\delta x_W} \right) \end{aligned} \quad (70)$$

$(\rho u A)$  é o fluxo de massa.

Como calcular  $\phi_e$  e  $\phi_w$  ?

A primeira alternativa seria usar uma interpolação linear, usando os dois pontos vizinhos. Dessa forma, temos

$$\phi_e = \frac{1}{2} (\phi_P + \phi_E) \quad (71)$$

e

$$\phi_w = \frac{1}{2} (\phi_P + \phi_W) \quad (72)$$

Essa aproximação é de segunda ordem e não tem nenhuma tendência quanto à direção do escoamento.

A equação para o ponto  $P$  se torna

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (73)$$

$$a_E = \frac{\Gamma_e A_e}{\delta x_E} - \frac{(\rho u)_e A_e}{2} \quad a_W = \frac{\Gamma_w A_w}{\delta x_W} + \frac{(\rho u)_w A_w}{2} \quad (74)$$

$$a_P = a_E + a_W + (\rho u)_e A_e - (\rho u)_w A_w \quad (75)$$

A aproximação linear funciona bem com a difusão, mas pode levar a resultados ruins para a advecção.

O problema é que a difusão acontece da mesma forma em todas as direções, enquanto a advecção é altamente direcional.

A variação da propriedade, devido à advecção, depende fortemente da direção do escoamento.

É aí que entram os métodos *upwind*.

A ideia do ***upwind*** é interpolar o valor da propriedade na face de forma desbalanceada, seguindo a direção do escoamento.

No nosso caso, se  $u_w > 0$  e  $u_e > 0$ , significa que o escoamento está para a direita. As informações do escoamento viajam da esquerda para a direita. Dessa forma, os valores nas faces são calculados como

$$\phi_w = \phi_W \quad (76)$$

e

$$\phi_e = \phi_P \quad (77)$$

$$\begin{aligned}
 & (\rho u)_e \phi_P A_e - (\rho u)_w \phi_W A_w = \\
 & = \Gamma_e A_e \left( \frac{\phi_E - \phi_P}{\delta x_E} \right) - \Gamma_w A_w \left( \frac{\phi_P - \phi_W}{\delta x_W} \right)
 \end{aligned} \tag{78}$$

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \tag{79}$$

$$a_E = \frac{\Gamma_e A_e}{\delta x_E} \quad a_W = \frac{\Gamma_w A_w}{\delta x_W} + (\rho u)_w A_w \tag{80}$$

$$a_P = a_E + a_W + (\rho u)_e A_e - (\rho u)_w A_w \tag{81}$$

Se  $u_w < 0$  e  $u_e < 0$ , significa que o escoamento está para a esquerda. Agora as informações do escoamento viajam da direita para a esquerda. Dessa forma, os valores nas faces são calculados como

$$\phi_w = \phi_P \quad (82)$$

e

$$\phi_e = \phi_E \quad (83)$$

$$\begin{aligned}
 & (\rho u)_e \phi_E A_e - (\rho u)_w \phi_P A_w = \\
 & = \Gamma_e A_e \left( \frac{\phi_E - \phi_P}{\delta x_E} \right) - \Gamma_w A_w \left( \frac{\phi_P - \phi_W}{\delta x_W} \right)
 \end{aligned} \tag{84}$$

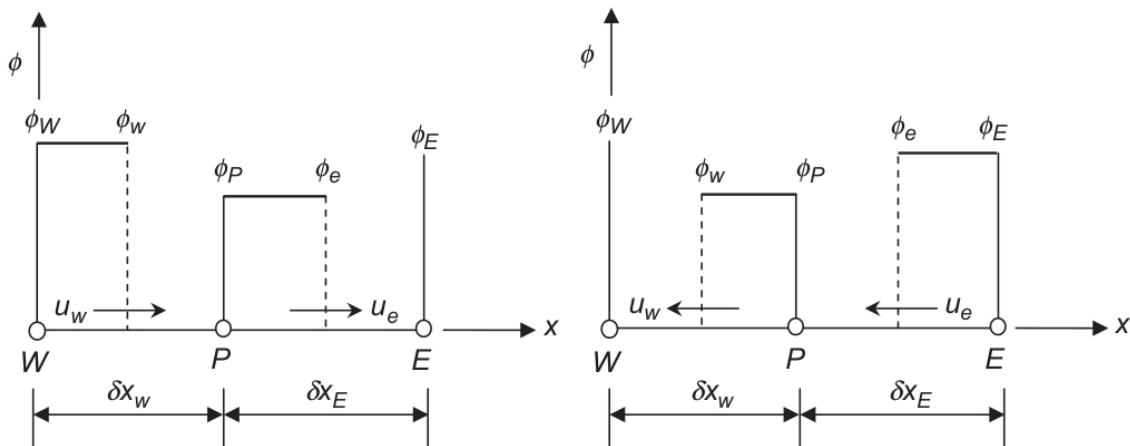
$$a_P \phi_P = a_E \phi_E + a_W \phi_W \tag{85}$$

$$a_E = \frac{\Gamma_e A_e}{\delta x_E} - (\rho u)_e A_e \quad a_W = \frac{\Gamma_w A_w}{\delta x_W} \tag{86}$$

$$a_P = a_E + a_W + (\rho u)_e A_e - (\rho u)_w A_w \tag{87}$$



Esse método é o *upwind* de primeira ordem.

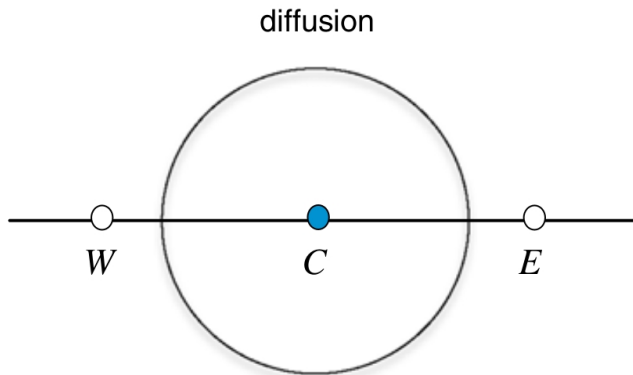


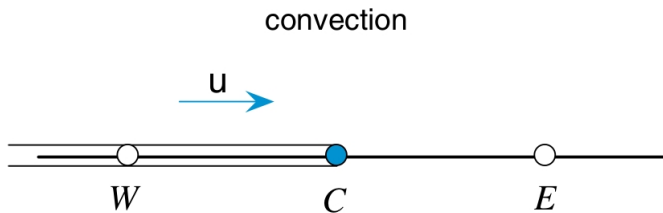
O método usando a aproximação centrada se torna instável, no caso de escoamento para a direita, quando o termo  $a_E$  se torna negativo. Assim:

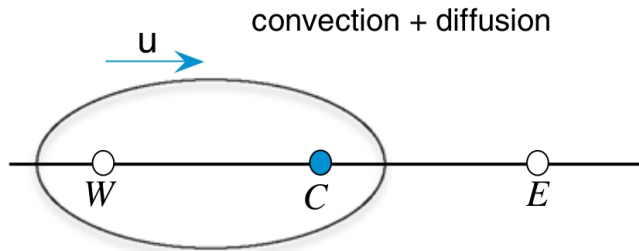
$$a_E < 0 \quad \frac{\Gamma_e A_e}{\delta x_E} - \frac{(\rho u)_e A_e}{2} < 0 \quad (88)$$

$$Pe = \frac{\rho u_e (\delta x_E)}{\Gamma_e} > 2 \quad (89)$$

$Pe$  é o número de Péclet e representa uma razão local (na célula) entre advecção e difusão.  $Pe = 0$  indica difusão pura.







Considerando agora a equação da advecção-difusão em regime transiente, temos

$$\frac{\partial}{\partial t} (\rho\phi) + \frac{\partial}{\partial x} (\rho u\phi) = \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \phi}{\partial x} \right) \quad (90)$$

Considerando  $\rho$  constante:

$$\frac{\partial}{\partial t} (\phi) + \frac{\partial}{\partial x} (u\phi) = \frac{\partial}{\partial x} \left( \frac{\Gamma}{\rho} \frac{\partial \phi}{\partial x} \right) \quad (91)$$

Fazendo a discretização espacial, para o volume que tem o ponto  $P$  como centroide, temos:

$$\begin{aligned} \frac{\partial \phi}{\partial t} = & -u_e \phi_e A_e + u_w \phi_w A_w + \\ & + \frac{\Gamma_e}{\rho} A_e \left( \frac{\phi_E - \phi_P}{\delta x_E} \right) - \frac{\Gamma_w}{\rho} A_w \left( \frac{\phi_P - \phi_W}{\delta x_W} \right) \end{aligned} \quad (92)$$

Usando uma expansão em série de Taylor para  $\phi_P^{n+1}$  em torno de  $\phi_P^n$ , temos:

$$\left(\frac{\partial \phi}{\partial t}\right)^n = \frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} + O(\Delta t) \quad (93)$$

Aqui,  $\Delta t$  é o incremento no tempo (diferença entre dois tempos consecutivos) e  $n$  é o número de passos de tempo dados.  $n$  é o tempo atual (conhecido) e  $n + 1$  é o próximo passo de tempo (desconhecido).

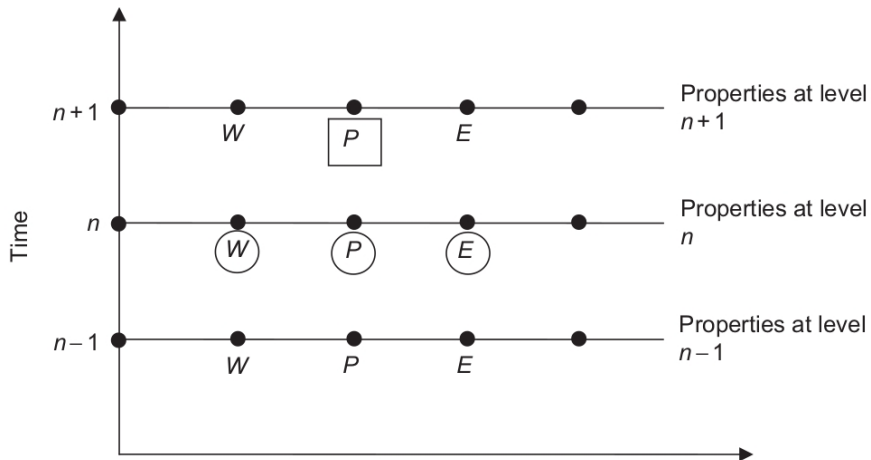


Substituindo:

$$\begin{aligned} \frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} = & -u_e^n \phi_e^n A_e + u_w^n \phi_w^n A_w + \\ & + \frac{\Gamma_e^n}{\rho} A_e \left( \frac{\phi_E^n - \phi_P^n}{\delta x_E} \right) - \frac{\Gamma_w^n}{\rho} A_w \left( \frac{\phi_P^n - \phi_W^n}{\delta x_W} \right) \end{aligned} \quad (94)$$

Esse é o método de **Euler explícito**, que tem primeira ordem no tempo.

É chamado de explícito porque a variável desconhecida ( $\phi_P^{n+1}$ ) pode ser calculada diretamente.



Também podemos ter um método de primeira ordem implícito.

Usando uma expansão em série de Taylor para  $\phi_P^n$  em torno de  $\phi_P^{n+1}$ , temos:

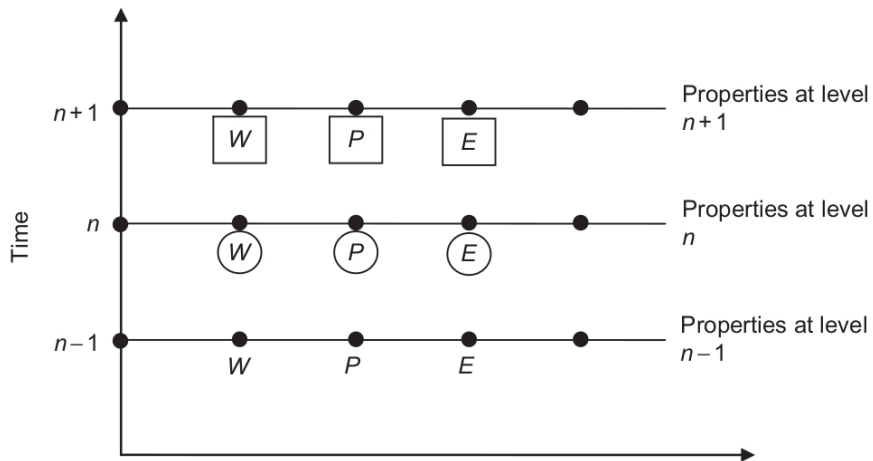
$$\left(\frac{\partial \phi}{\partial t}\right)^{n+1} = \frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} + O(\Delta t) \quad (95)$$

Substituindo:

$$\begin{aligned} \frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} = & -u_e^{n+1} \phi_e^{n+1} A_e + u_w^{n+1} \phi_w^{n+1} A_w + \\ & + \frac{\Gamma_e^{n+1}}{\rho} A_e \left( \frac{\phi_E^{n+1} - \phi_P^{n+1}}{\delta x_E} \right) - \frac{\Gamma_w^{n+1}}{\rho} A_w \left( \frac{\phi_P^{n+1} - \phi_W^{n+1}}{\delta x_W} \right) \end{aligned} \quad (96)$$

Esse é o método de **Euler implícito**, que também tem primeira ordem no tempo.

É chamado de implícito porque a variável desconhecida ( $\phi_P^{n+1}$ ) não pode ser obtida diretamente. Neste caso, devemos resolver um sistema de equações algébricas para obter todos os  $\phi_P^{n+1}$  simultaneamente.



## Comentários.

O método de Euler explícito exige um  $\Delta t$  muito baixo para ser estável, muitas vezes impraticável. Por isso quase sempre utilizamos o implícito, que permite o uso de um  $\Delta t$  maior.

No entanto, o método de Euler implícito é de primeira ordem. O mais interessante é, sempre que possível, buscar por métodos de segunda ordem (*backward* ou Crank-Nicolson, no OpenFOAM).

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas**
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

Em muitas metodologias numéricas (Volumes Finitos, Diferenças Finitas, Elementos Finitos), a etapa final da solução consiste na resolução de um sistema de equações lineares.

**Por isso, o estudo de métodos para resolver esses sistemas é muito importante.**

Temos dois tipos de métodos usados para resolver sistemas lineares: os **Métodos Diretos** e os **Métodos Iterativos**.



**Métodos Diretos** são aqueles que dão a **solução exata** (a menos do erro de máquina) após um certo número de passos.

Exemplos de Métodos Diretos:

- 1 Eliminação Simples
- 2 Eliminação de Gauss
- 3 Decomposição LU
- 4 Eliminação de Gauss-Jordan
- 5 Algoritmo de Thomas

**Métodos Iterativos**, por sua vez, obtêm uma **solução aproximada** para o sistema de equações, a partir de uma resposta inicial (*chute*) e de um algoritmo de iteração.

Alguns exemplos de Métodos Iterativos:

- 1 Gauss-Jacobi
- 2 Gauss-Seidel
- 3 SOR
- 4 Gradiente Conjugado
- 5 Multigrid

Em simulações CFD quase sempre são usados métodos iterativos.

Temos pelo menos 3 vantagens:

- no método iterativo não é necessário montar a matriz  $\mathbf{A}$  (esse é um problema sério em duas e três dimensões);
- com um *chute* inicial bom, o método iterativo converge com algumas poucas iterações;
- o método direto nos dá a solução exata, mas muitas vezes uma solução aproximada já é o suficiente, pois já existe um erro associado à própria aproximação das derivadas no problema.

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade**
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

No caso de um escoamento incompressível, as equações governantes do problema são a equação da continuidade,

$$\nabla \cdot \mathbf{U} = 0 ,$$

e a equação de Navier-Stokes,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} = -\nabla p .$$

Temos 4 equações e 4 incógnitas (as 3 componentes da velocidade e a pressão).

O problema é que não temos uma equação para a pressão.

Uma forma de contornar esse problema é por meio dos métodos de correção da pressão, que são os mais usados em CFD.

A ideia do método é criar uma equação para a pressão a partir da equação do momento, com a restrição da incompressibilidade.

Considere novamente a equação

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} = -\nabla p$$

Aplicando o divergente na equação, temos

$$\nabla \cdot \left( \frac{\partial \mathbf{U}}{\partial t} \right) + \nabla \cdot (\nabla \cdot (\mathbf{U}\mathbf{U})) - \nabla \cdot (\nu \nabla^2 \mathbf{U}) = -\nabla \cdot (\nabla p)$$

Considerando que o divergente da velocidade deve ser nulo,  $\nabla \cdot \mathbf{U} = 0$ , resulta

$$\cancel{\nabla \cdot \left( \frac{\partial \mathbf{U}}{\partial t} \right)} + \nabla \cdot (\nabla \cdot (\mathbf{U}\mathbf{U})) - \cancel{\nabla \cdot (\nu \nabla^2 \mathbf{U})} = -\nabla \cdot (\nabla p)$$

Assim, temos uma equação para a pressão, dada por

$$\nabla^2 p = \nabla \cdot \mathbf{H}(\mathbf{U}) \ ,$$

em que

$$\mathbf{H}(\mathbf{U}) = -\nabla \cdot (\mathbf{U}\mathbf{U}) \ .$$



Dessa forma, a ideia desses métodos é resolver a pressão

$$\nabla^2 p = \nabla \cdot \mathbf{H}(\mathbf{U})$$

e em seguida usar essa pressão para calcular uma nova velocidade

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} = -\nabla p .$$

Esse processo pode ter iterações internas, para melhorar/facilitar a convergência.

No OpenFOAM, a equação a ser resolvida é ligeiramente modificada.

Para deixar a equação para a pressão mais convergente, um termo  $A\mathbf{U}$  (em que  $A$  representa os termos da diagonal da matriz dos coeficientes na equação do momento) é adicionado e subtraído da equação de Navier-Stokes:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} - A\mathbf{U} + A\mathbf{U} = -\nabla p .$$

Seja agora

$$-\mathbf{H}(\mathbf{U}) = \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} - A\mathbf{U}$$

Dividindo por  $A$  e aplicando o divergente, temos:

$$-\nabla \cdot \left( \frac{\mathbf{H}(\mathbf{U})}{A} \right) = -\nabla \cdot \left( \frac{1}{A} \nabla p \right) .$$

Assim, temos a equação para a pressão:

$$\nabla \cdot \left( \frac{1}{A} \nabla p \right) = \nabla \cdot \left( \frac{\mathbf{H}(\mathbf{U})}{A} \right) .$$

Calculada a pressão, a velocidade é obtida com a equação

$$\mathbf{U} = \frac{\mathbf{H}(\mathbf{U})}{A} - \frac{1}{A} \nabla p .$$

Temos, então, 3 passos básicos.

1) *momentum predictor* (opcional):

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} = -\nabla p$$

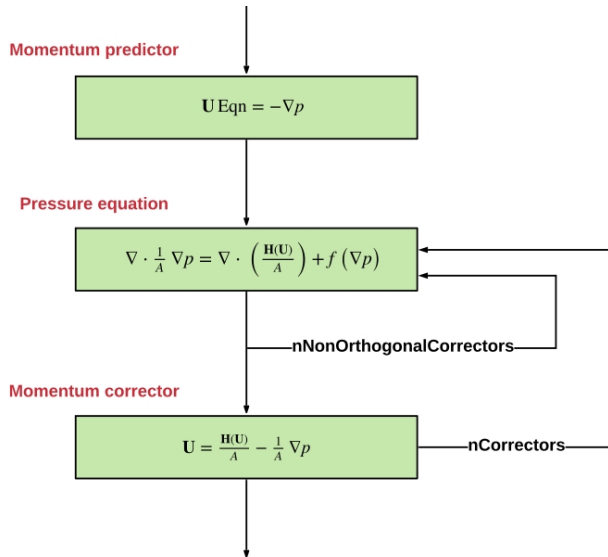
2) *pressure equation*:

$$\nabla \cdot \left( \frac{1}{A} \nabla p \right) = \nabla \cdot \left( \frac{\mathbf{H}(\mathbf{U})}{A} \right)$$

3) *momentum corrector*:

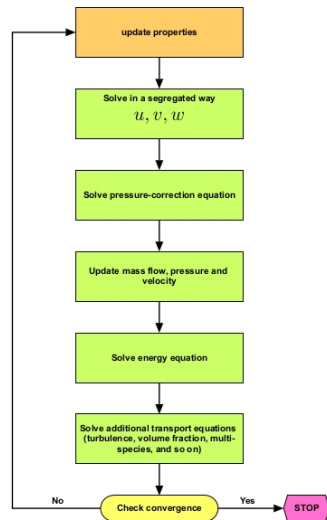
$$\mathbf{U} = \frac{\mathbf{H}(\mathbf{U})}{A} - \frac{1}{A} \nabla p$$

Esses são os passos básicos dos métodos de correção da pressão.



Os métodos de correção da pressão mais utilizados são:

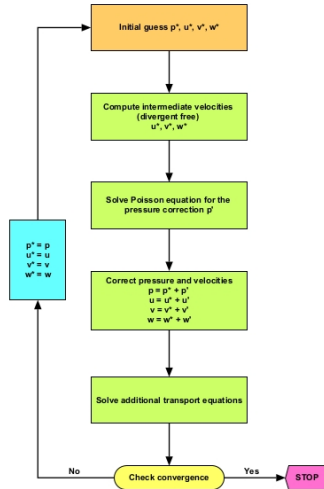
- **SIMPLE** (Semi-Implicit Method for Pressure-Linked Equations)
- **SIMPLEC** (SIMPLE Corrected/Consistent)
- **PISO** (Pressure Implicit with Splitting Operators)
- **PIMPLE** (PISO + SIMPLE)



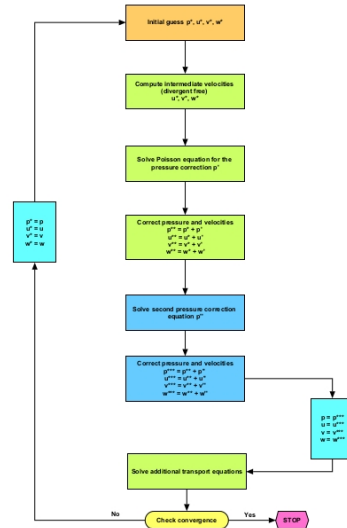
Os métodos SIMPLE e SIMPLEC são usados em simulações em regime permanente, enquanto os métodos PISO e PIMPLE são usados em simulações transientes.

Os métodos SIMPLE e PISO são apresentados no próximo slide.

## Pressure-velocity coupling using the SIMPLE method



## Pressure-velocity coupling using the PISO method





# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 **OpenFOAM**
  - fvSchemes
  - fvSolution
  - Exemplo

Um caso típico no OpenFOAM tem 3 pastas:

- *0*
- *constant*
- *system*

Na pasta *0* nós entramos com as condições iniciais e de contorno do problema.

Na pasta *constant* nós temos as propriedades físicas do fluido, no arquivo *physicalProperties*, e as informações sobre a modelagem do transporte de momento, no arquivo *momentumTransport*.

Na pasta *system*, temos as informações sobre a geometria e a malha, no arquivo *blockMeshDict*, e sobre as configurações gerais da simulação, no arquivo *controlDict*.

Temos, ainda na pasta *system*, os arquivos ***fvSchemes*** e ***fvSolution***.

No arquivo ***fvSchemes***, nós especificamos os esquemas de discretização usados no método de volumes finitos empregado para resolver as equações governantes.

As especificações sobre os métodos de resolução dos sistemas de equações lineares e sobre o algoritmo de acoplamento pressão-velocidade, por sua vez, são informadas no arquivo ***fvSolution***.

Vamos analisar esses dois arquivos, começando com o *fvSchemes*.

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - **fvSchemes**
  - fvSolution
  - Exemplo

No dicionário ***fvSchemes***, nós vamos especificar os esquemas de discretização utilizados nas aproximações do método de volumes finitos.

Assim, precisaremos definir os seguintes subdicionários:

- ***ddtSchemes***
- ***gradSchemes***
- ***divSchemes***
- ***laplacianSchemes***
- ***interpolationSchemes***
- ***snGradSchemes***

**Observação:** o material desta seção foi retirado de *“Introduction to the FVM method: Standard practices in general CFD with applications to Open-FOAM”*, da Wolf Dynamics. Recomendo muito a leitura.

***ddtSchemes*** se refere à forma como a derivada no tempo é discretizada.

As principais opções são:

- *steadyState*: para simulações em regime permanente.
- *Euler*: para simulações em regime transiente, primeira ordem.
- *backward*: para simulações em regime transiente, segunda ordem.
- *CrankNicolson*: para simulações em regime transiente, entre primeira e segunda ordem.

No caso transiente, *Euler* é o mais robusto, mas é de primeira ordem. *Crank-Nicolson 1* é o método de Crank-Nicolson puro (segunda ordem, mas pode apresentar oscilações indesejadas), enquanto *CrankNicolson 0* recupera o método de *Euler*. A opção *CrankNicolson 0.7* é uma boa escolha em várias situações, já que é estável e preciso.

```
1 ddtSchemes
2 {
3     default    CrankNicolson 0.7;
4 }
```



No ***divSchemes*** nós definimos como são calculados os termos advectivos, ou seja, os termos do tipo  $\nabla \cdot (\mathbf{U}\phi)$ .

As principais opções são:

- *upwind*: primeira ordem, limitado.
- *linear*: central, segunda ordem, ilimitado.
- *linearUpwind*: segunda ordem, ilimitado.
- *vanLeer*: segunda ordem, TVD, limitado.
- *Minmod*: segunda ordem, TVD, limitado.
- *limitedLinear*: segunda ordem, ilimitado.

O método *upwind* é estável e robusto, mas é de primeira ordem e é muito difusivo. Os métodos de segunda ordem possuem uma precisão maior, mas podem se tornar oscilatórios. O ideal é um método de segunda ordem estável. Podemos começar a simulação com o método de primeira ordem e, depois de ver que tudo está funcionando bem, passar para um de segunda ordem. O melhor método vai depender do problema.

O tratamento dos termos advectivos é um dos maiores desafios em CFD.

```
1 ddtSchemes
2 {
3     default      none;
4     div(phi,U)   Gauss linearUpwind grad(U);
5 }
```

O dicionário ***gradSchemes*** indica a forma como o gradiente é calculado ( $\nabla(p)$  e  $\nabla(\mathbf{U})$ , por exemplo).

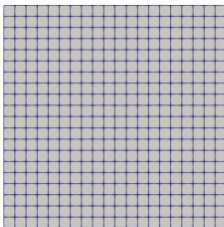
A opção mais usada é a *Gauss linear*. A opção *cellLimited Gauss linear 1* também é usada, principalmente em simulações com malhas não tão boas. A opção *cellLimited* limita o valor do gradiente de acordo com os valores da variável nas células mais próximas, aumentando a estabilidade da simulação.

```
1 gradSchemes
2 {
3     default      Gauss linear;
4     grad(U)      cellLimited Gauss linear 1;
5 }
```

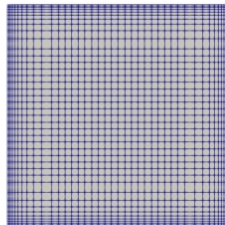
O dicionário ***laplacianSchemes*** está relacionado aos termos difusivos das equações, da forma  $\nabla \cdot (\Gamma \nabla \phi)$ .

As principais opções aqui são:

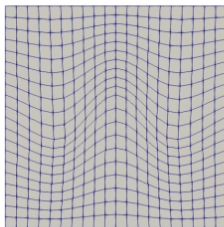
- *orthogonal*: para malhas perfeitas (ortogonal e com células idênticas), segunda ordem.
- *corrected*: para malhas não ortogonais e com diferentes tamanhos de célula.
- *limited*: para malhas não ortogonais e com diferentes tamanhos de célula.



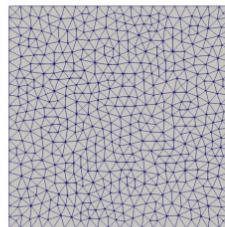
Perfect orthogonal mesh with no stretching  
**laplacianSchemes** → **orthogonal**



Orthogonal mesh with stretching  
**laplacianSchemes** → **limited 1** or **corrected**



Mesh with some degree of non-orthogonality (low to medium)  
**laplacianSchemes** → **limited 1** to **limited 0.5**



General unstructured meshes  
**laplacianSchemes** → **limited 0.5**

Para malhas com não ortogonalidade menor que  $70^\circ$ , você pode deixar o fator *limited* como sendo 1.

Para malhas com não ortogonalidade entre  $70^\circ$  e  $85^\circ$ , defina o fator como 0.5.

Para malhas com não ortogonalidade maior que  $85^\circ$ , o melhor é tentar fazer uma nova malha. Mas caso você queira usar essa malha, defina o fator entre 0.333 e 0.5.

```
1  laplacianSchemes
2  {
3      default      Gauss linear limited 1;
4  }
```

Os termos de gradiente normal às superfícies, ***snGradSchemes***, são usados no cálculo dos termos laplacianos quando o teorema da divergência é aplicado. Normalmente eles são configurados da mesma forma que os termos no *laplacianSchemes*.

```
1 snGradSchemes
2 {
3     default        limited 1;
4 }
```

Por fim, o subdicionário ***interpolationSchemes*** apresenta o método utilizado para interpolar os valores do centro de uma célula nas faces dessa célula.

A opção mais utilizada neste caso é a *linear*.

```
1 interpolationSchemes
2 {
3     default      linear;
4 }
```



## Recommended setup for most cases

```

ddtSchemes
{
    default      CrankNicolson 0.333; //0-1
}
gradSchemes
{
    default      cellLimited Gauss linear 1;
    grad(p)      cellLimited Gauss linear 0.5;
}
divSchemes
{
    default      none;
    div(phi,U)   Gauss linearUpwindV default;
    div(phi,omega) Gauss linearUpwind default;
    div(phi,k)   Gauss linearUpwind default;
    div((nuEff*dev(T(grad(U)))) Gauss linear;
}
laplacianSchemes
{
    default      Gauss linear limited 1;
}
interpolationSchemes
{
    default      linear;
}
snGradSchemes
{
    default      limited 1;
}

```

• This setup is recommended for most of the cases.



- It is very similar to the default method you will find in commercial solvers.
- In overall, this setup is second order accurate and fully bounded.
- According to the quality of your mesh, you will need to change the blending factor of the **laplacianSchemes** and **snGradSchemes** keywords.
- To keep time diffusion to a minimum, use a CFL number less than 2, and preferably below 1.
- If during the simulation the turbulence quantities become unbounded, you can safely change the discretization scheme to upwind. After all, turbulence is diffusion.
- For gradient discretization the **leastSquares** method is more accurate. But we have found that it is a little bit oscillatory in tetrahedral meshes.

## A very stable but too diffusive numerics

```

ddtSchemes
{
    default      Euler;
}
gradSchemes
{
    default      cellLimited Gauss linear 1;
}
divSchemes
{
    default      none;
    div(phi,U)   Gauss upwind;
    div(phi,omega) Gauss upwind;
    div(phi,k)   Gauss upwind;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default      Gauss linear limited 0.5;
}
interpolationSchemes
{
    default      linear;
}
snGradSchemes
{
    default      limited 0.5;
}

```

- If you are looking for extra stability, you can use this method.
- This setup is very stable but too diffusive.
- This setup is first order in space and time.
- You can use this setup to start the solution in the presence of bad quality meshes or strong discontinuities.
- Remember, you can start using a first order method and then switch to a second order method.
- According to the quality of your mesh, you will need to change the blending factor of the **laplacianSchemes** and **snGradSchemes** keywords.
- You can use this method for troubleshooting. If the solution diverges, you better check boundary conditions, physical properties, and so on.
- **Start robustly, end with accuracy.**

# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - **fvSolution**
  - Exemplo

No arquivo ***fvSolution*** nós vamos definir os métodos de solução dos sistemas de equações lineares (os *solvers*) e os detalhes do algoritmo de acoplamento pressão-velocidade.

Com relação aos métodos de solução dos sistemas de equações, as principais opções são

- *smoothSolvers* (método de relaxação iterativo). Principais opções: DIC, DILU, DILUGaussSeidel, GaussSeidel, symGaussSeidel.
- Gradiente Conjugado e suas variações (PCG, PBiCG, PBiCGStab, ...). Principais pré-condicionadores: diagonal, DIC, DILU, GAMG.
- Multigrid (GAMG). Principais opções de smoother: GaussSeidel e DICGaussSeidel.

## Exemplo 1: *incompressibleFluid/motorBike* .

```
1 solvers
2 {
3     p
4     {
5         solver      GAMG;
6         tolerance    1e-07;
7         relTol       0.1;
8         smoother     GaussSeidel;
9     }
10
11     pFinal
12     {
13         $p;
14         tolerance    1e-6;
15         relTol        0;
16     }
```

```
17     Phi
18     {
19         $p;
20     }
21
22     "(U|nuTilda)"
23     {
24         solver      smoothSolver;
25         smoother     GaussSeidel;
26         tolerance    1e-08;
27         relTol       0.1;
28     }
29 }
```

## Exemplo 2: *fluid/BenardCells* .

```
1 solvers
2   "rho.*"
3   {
4       solver      diagonal;
5   }
6
7   p_rgh
8   {
9       solver      GAMG;
10      smoother     DIC;
11      tolerance     1e-8;
12      relTol        0.01;
13  }
14
15  p_rghFinal
16  {
17      $p_rgh;
18      relTol        0;
19  }
```

```
20   "(U|e|k|epsilon)"
21   {
22       solver      PBiCGStab;
23       preconditioner DILU;
24       tolerance     1e-8;
25       relTol        0.01;
26   }
27
28   "(U|e|k|epsilon)Final"
29   {
30       $U;
31       relTol        0;
32   }
33
34 }
```

Quando se trata de *solvers*, há muitas opções e combinações disponíveis no OpenFOAM.

Ao escolher os *solvers* lineares, não há uma teoria geral. Isso depende do problema e do hardware (tipo de malha, física envolvida, memória do processador, conectividade de rede, método de particionamento, entre outros).

Na maioria das vezes, o método GAMG é a melhor escolha para matrizes simétricas (por exemplo, pressão). O método GAMG deve convergir rapidamente (menos de 100 iterações). Se estiver levando mais iterações, tente alterar algumas das opções do solucionador.

Se estiver demorando muito ou for instável, use o solucionador PCG com um bom pré-condicionador. Ao executar com muitos núcleos (mais de 1000), usar o PCG pode ser uma escolha melhor.



Para matrizes assimétricas, o método PBiCGStab com o pré-condicionador DILU é uma boa escolha.

O *solver smoothSolver* com Gauss-Seidel também apresenta um desempenho muito bom.

Se o método PBiCGStab com o pré-condicionador DILU falhar misteriosamente com um erro relacionado ao pré-condicionador, utilize o *smoothSolver* ou altere o pré-condicionador.

Em geral, o solucionador PBiCGStab deve ser mais rápido do que o *smoothSolver*.

Matrizes assimétricas são montadas a partir da velocidade ( $\mathbf{U}$ ) e das quantidades transportadas ( $k$ ,  $\omega$ ,  $\epsilon$ ,  $e$ ,  $T$ , etc.). A pressão possui uma matriz simétrica.

Geralmente, calcular a velocidade e as quantidades transportadas é barato e rápido, então é uma boa ideia usar uma tolerância rigorosa ( $10^{-8}$ ) para esses campos. Resolver a pressão geralmente é a parte mais cara.

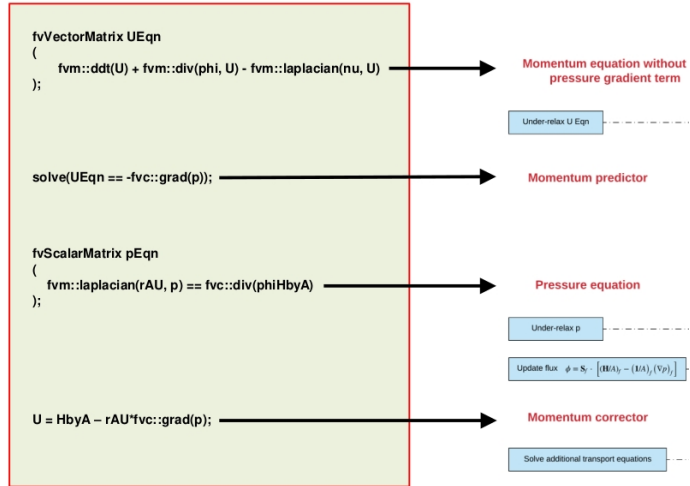
O solucionador *diagonal* é utilizado para retrossubstituição, por exemplo, ao calcular a densidade usando a equação de estado.

A configuração do algoritmo de acoplamento pressão-velocidade também é feita no *fvSolution*.

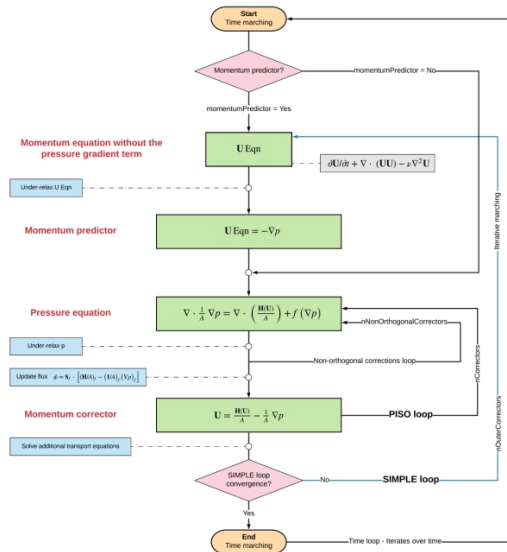
Em simulações em *steadyState*, temos que definir o subdicionário SIMPLE ou o PIMPLE.

Em simulações transientes, temos que definir o dicionário PIMPLE.

# PIMPLE.



# PIMPLE.



Com relação ao *nOrthogonalCorrectors*, recomenda-se um valor de 3 para não ortogonalidade entre  $70^\circ$  e  $85^\circ$ , 2 para não ortogonalidade entre  $60^\circ$  e  $70^\circ$  e 1 para não ortogonalidade menor que  $60^\circ$ .

Exemplo.

```
1 PIMPLE
2 {
3     nOuterCorrectors 1;
4     nCorrectors      3;
5     nNonOrthogonalCorrectors 1;
6 }
```

# Exemplo.

```
1 SIMPLE
2 {
3     nNonOrthogonalCorrectors 0;
4     consistent    yes;
5     residualControl
6     {
7         p          1e-2;
8         U          1e-3;
9         "(k|epsilon|omega|f|v2)" 1e-3;
10    }
11 }
12
13 relaxationFactors
14 {
15     equations
16     {
17         U          0.9; // 0.9 is more stable but 0.95 more convergent
18         ".*"       0.9; // 0.9 is more stable but 0.95 more convergent
19     }
20 }
```



# Sumário

- 1 Introdução
- 2 Métodos de Discretização
  - FDM
  - FVM
- 3 Exemplos de Aplicação de FVM
- 4 Problema de Advecção-Difusão
- 5 Resolvendo Sistemas de Equações Algébricas
- 6 Algoritmos de Acoplamento Pressão-Velocidade
- 7 OpenFOAM
  - fvSchemes
  - fvSolution
  - Exemplo

Para entender na prática como essas configurações funcionam, vamos analisar um caso bem simples: transporte advectivo e difusivo de uma propriedade escalar em uma dimensão.

Abra o caso *temperatureScalarTransport*. Para rodar este caso, execute os comandos *blockMesh*, *setFields* e *foamRun*, nessa ordem.

Rode diferentes simulações com diferentes parâmetros e diferentes configurações (modifique as configurações nos arquivos *fvSchemes* e *fvSolution*).

Veja quais são as melhores combinações para este caso. Observe os casos em que a simulação diverge. Observe também os casos em que aparecem oscilações indesejadas na simulação.

Por fim, dê uma olhada com calma em diferentes tutoriais do OpenFOAM, focando nos arquivos *fvSchemes* e *fvSolution*.