

Simulação Numérica De Escoamentos Dispersos Em Turbomáquinas Utilizando Método De Elementos Finitos

Lucas Carvalho De Sousa
Gustavo Rabello Dos Anjos

Universidade do Estado do Rio de Janeiro

encarvlucas@hotmail.com

25 de Junho de 2019



Sumário

1 Introdução

- Simulação de Escoamentos Bidimensionais com Partículas
- Escoamentos em Turbomáquinas

2 Equações de Governo

- Formulação Corrente-Vorticidade
- Equação de Basset–Boussinesq–Oseen (BBO)

3 Métodos Numéricos

- Método dos Elementos Finitos
- Discretizações dos Modelos
- Definição das Matrizes

4 Código

- Montagem das Matrizes Globais
- Estrutura de Uso da Biblioteca
- Estrutura de Solução

5 Validações e Resultados

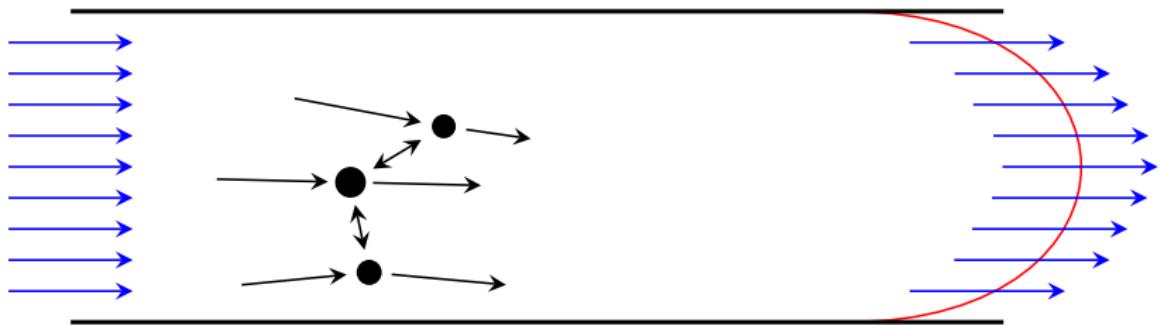
- Validações
- Resultados de Simulações

Introdução

Simulação de Escoamentos Bidimensionais com Partículas

Objetivos deste trabalho:

Desenvolver uma biblioteca de Python para a simulação de escoamentos particulados.

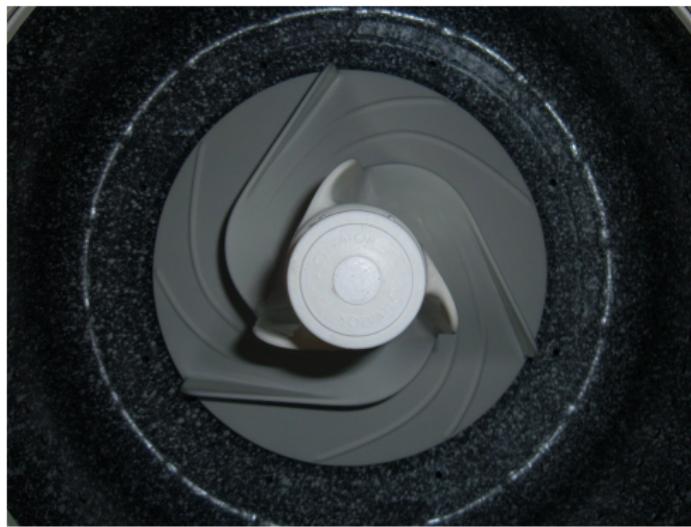


Escoamento entre placas, Hagen-Poiseuille.

Escoamentos em Turbomáquinas

Objetivos deste trabalho:

Estudar como partículas se comportam dentro de uma turbomáquina em funcionamento.



Fonte: © BrokenSphere / Wikimedia Commons.

Equações de Governo

Formulação Corrente-Vorticidade

Hipóteses tomadas

- Fluído incompressível
- Fluído newtoniano

Equação de Navier-Stokes

$$\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \vec{v}_f = -\frac{1}{\rho_f} \vec{\nabla} p + \frac{\mu_f}{\rho_f} \nabla^2 \vec{v}_f + \vec{g}$$

Desvantagens

- Requer o uso da equação auxiliar da continuidade devido à presença da pressão
- Acoplamento da pressão e velocidade
- Exige elementos de ordem elevada

Formulação Corrente-Vorticidade

Equação da Vorticidade

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \vec{\omega} = \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega}$$

Equação da Corrente

$$\nabla^2 \psi = -\omega_z$$

Equações Auxiliares

$$\vec{v}_f = (v_{f,x}, v_{f,y})$$

$$v_{f,x} = \frac{\partial \psi}{\partial y}$$

$$v_{f,y} = -\frac{\partial \psi}{\partial x}$$

$$\omega_z = \frac{\partial v_{f,x}}{\partial y} - \frac{\partial v_{f,y}}{\partial x}$$

Equação de Basset–Boussinesq–Oseen (BBO)

Equação que representa as forças exercidas sobre as partículas.
Sua expressão é a soma das forças separadamente.

Equação de Basset–Boussinesq–Oseen

$$\vec{F}_p = \sum \vec{F} = \vec{F}_{grav} + \vec{F}_{drag} + \vec{F}_{lift} + \vec{F}_{mass}$$

Restrição

A equação BBO é somente válida para Reynolds da partícula menores que 1. $Re_p < 1$

Reynolds de Partícula

$$Re_p = \frac{\rho_p}{\mu_f} |(\vec{v}_f - \vec{v}_p)|_{max} d_p$$

Equação de Basset–Boussinesq–Oseen (BBO)

Força Gravitacional

$$\vec{F}_{grav} = m_p \vec{g}$$

Força de Sustentação

$$\vec{F}_{lift} = 1.61 \mu_f d_p (\vec{v}_f - \vec{v}_p) \sqrt{Re_G}$$

Força de Arrasto

$$\vec{F}_{drag} = 3\pi \mu_f d_p (\vec{v}_f - \vec{v}_p)$$

Força de Massa Virtual

$$\vec{F}_{mass} = \frac{1}{2} \rho_f V_p \frac{d}{dt} (\vec{v}_f - \vec{v}_p)$$

Reynolds de Cisalhamento

$$Re_G = \frac{\rho_f}{\mu_f} d_p^2 \nabla \vec{v}_f$$

Métodos Numéricos

Método dos Elementos Finitos

Domínio

Equações são definidas em um domínio Ω com contorno Γ .

Forma forte com as funções peso

$$\int_{\Omega} \left(\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \vec{\omega} - \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \right) \cdot \vec{\delta} d\Omega = 0$$

$$\int_{\Omega} (\nabla^2 \psi + \omega_z) \cdot \vec{\phi} d\Omega = 0$$

$$\int_{\Omega} \left(\vec{v}_f - \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \right) \cdot \vec{\xi} d\Omega = 0$$

Condições de contorno

$$\omega = \omega_{\Gamma} \text{ em } \Gamma$$

$$\psi = \psi_{\Gamma} \text{ em } \Gamma$$

$$\vec{v}_f = \vec{v}_{f\Gamma} \text{ em } \Gamma$$

$\vec{\delta}$, $\vec{\phi}$ e $\vec{\xi}$ são as funções de peso de cada equação.

Método dos Elementos Finitos

Forma fraca

$$m_1 \left(\frac{\partial \vec{\omega}}{\partial t}, \delta \right) + g_1(\vec{v}_f, \vec{\delta}) + \frac{\mu_f}{\rho_f} k_1(\vec{\omega}, \vec{\delta}) = 0$$

$$-k_2(\psi, \vec{\phi}) + m_2(\omega_z, \vec{\phi}) = 0$$

$$m_3(\vec{v}_f, \vec{\xi}) - g_3(\psi, \vec{\xi}) = 0$$

Onde:

$$m_1 \left(\frac{\partial \vec{\omega}}{\partial t}, \delta \right) = \int_{\Omega} \frac{\partial \vec{\omega}}{\partial t} \cdot \vec{\delta} d\Omega$$

$$g_1(\vec{v}_f, \vec{\delta}) = \int_{\Omega} \vec{v}_f \cdot \vec{\nabla} \vec{\omega} \cdot \vec{\delta} d\Omega$$

$$k_1(\vec{\omega}, \vec{\delta}) = \int_{\Omega} \vec{\nabla} \vec{\omega} \cdot \vec{\nabla} \vec{\delta} d\Omega$$

$$k_2(\psi, \vec{\phi}) = \int_{\Omega} \vec{\nabla} \psi \cdot \vec{\nabla} \vec{\phi} d\Omega$$

$$m_2(\omega_z, \vec{\phi}) = \int_{\Omega} \omega_z \cdot \vec{\phi} d\Omega$$

$$m_3(\vec{v}_f, \vec{\xi}) = \int_{\Omega} \vec{v}_f \cdot \vec{\xi} d\Omega$$

$$g_3(\psi, \vec{\xi}) = \int_{\Omega} \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \cdot \vec{\xi} d\Omega$$

Discretização do Modelo de Escoamentos

Formulação de Galerkin

Funções de peso são definidas com valor igual às funções interpoladoras.

$$\omega(\vec{x}, t) = \sum_{i=1}^{n_p} \omega_i(t) N_i(\vec{x})$$

$$\delta(\vec{x}, t) = \sum_{j=1}^{n_p} \delta_i(t) N_j(\vec{x})$$

$$\psi(\vec{x}, t) = \sum_{i=1}^{n_p} \psi_i(t) N_i(\vec{x})$$

$$\phi(\vec{x}, t) = \sum_{j=1}^{n_p} \phi_i(t) N_j(\vec{x})$$

$$v_{f,x}(\vec{x}, t) = \sum_{i=1}^{n_p} v_{f,x,i}(t) N_i(\vec{x})$$

$$\xi(\vec{x}, t) = \sum_{j=1}^{n_p} \xi_i(t) N_j(\vec{x})$$

$$v_{f,y}(\vec{x}, t) = \sum_{i=1}^{n_p} v_{f,y,i}(t) N_i(\vec{x})$$

Discretização do Modelo de Escoamentos

Função de Aproximação

$N(x)$ é a função de aproximação de cada elemento:

$$N_i(\vec{x}) = [N_1(\vec{x}), \dots, N_{n_p}(\vec{x})]$$

Matrizes locais dos elementos

Surgem os termos locais, para cada elemento e :

$$\mathbf{m}^e = \int_{\Omega^e} N_i^e N_j^e d\Omega^e$$

$$\mathbf{g}_x^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} N_j^e d\Omega^e$$

$$\mathbf{g}_y^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} N_j^e d\Omega^e$$

$$\mathbf{k}_{xx}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} d\Omega^e$$

$$\mathbf{k}_{yy}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} d\Omega^e$$

Discretização do Modelo de Escoamentos

Discretização no tempo

Para os termos temporais é utilizada o Método de Diferenças Finitas:

$$\frac{\partial \omega}{\partial t} \approx \frac{\omega(t + dt) - \omega(t)}{dt} = \frac{\omega^{t_{n+1}} - \omega^{t_n}}{dt}$$

Equações na forma global

$$\left(\mathbf{M} v_{f,x}^{t_n} \mathbf{G}_x + v_{f,y}^{t_n} \mathbf{G}_y + \frac{\mu_f}{\rho_f} (\mathbf{K}_{xx} + \mathbf{K}_{yy}) \right) \omega^{t_{n+1}} = \mathbf{M} \omega^{t_n}$$

$$(\mathbf{K}_{xx} + \mathbf{K}_{yy}) \psi = \mathbf{M} \omega^{t_{n+1}}$$

$$\mathbf{M} v_{f,x}^{t_n} \omega^{t_{n+1}} = \mathbf{G}_y \psi$$

$$\mathbf{M} v_{f,y}^{t_n} \omega^{t_{n+1}} = -\mathbf{G}_x \psi$$

Discretização do Modelo de Partículas

Equações das forças nas partículas

$$\vec{F}_{grav}^{t_n} = m_p \vec{g}$$

$$\vec{F}_{drag}^{t_n} = 3\pi\mu_f d_p \left(\vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}} \right)$$

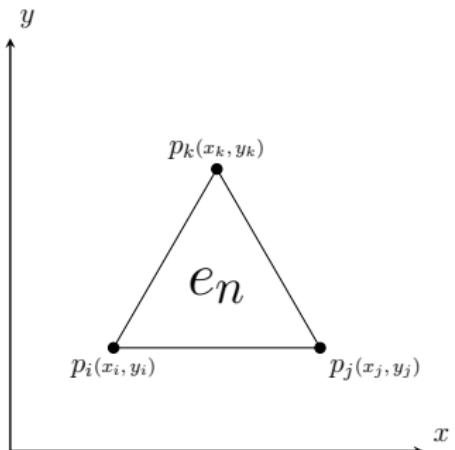
$$\vec{F}_{lift}^{t_n} = 1.61\mu_f d_p \left(\vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}} \right) \sqrt{Re_G^{t_n}}$$

$$\vec{F}_{mass}^{t_n} = \frac{1}{2}\rho_f V_p \frac{\left(\vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}} \right) - \left(\vec{v}_f^{t_{n-1}} - \vec{v}_p^{t_{n-2}} \right)}{dt}$$

Reynolds específicos

$$Re_p^{t_n} = \frac{\rho_p}{\mu_f} d_p \left| \vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}} \right|_{max} \quad Re_G^{t_n} = \frac{d_p^2 \rho_f}{\mu_f} \left(\frac{d \vec{v}_f}{d \vec{r}} \right)^{t_n}$$

Matrizes dos Elementos Triangulares



Elemento triangular linear.

Coordenadas relativas

$$\mathbf{b} \begin{cases} b_i = y_j - y_k \\ b_j = y_k - y_i \\ b_k = y_i - y_j \end{cases} \quad \mathbf{c} \begin{cases} c_i = x_k - x_j \\ c_j = x_i - x_k \\ c_k = x_j - x_i \end{cases}$$

Matrizes de Gradiente

$$\mathbf{g}_x^e = \frac{1}{6} \begin{bmatrix} b_i & b_j & b_k \\ b_i & b_j & b_k \\ b_i & b_j & b_k \end{bmatrix} \quad \mathbf{g}_y^e = \frac{1}{6} \begin{bmatrix} c_i & c_j & c_k \\ c_i & c_j & c_k \\ c_i & c_j & c_k \end{bmatrix}$$

Matriz de Massa

$$\mathbf{m}^e = \frac{A^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Matrizes de Rígidez

$$\mathbf{k}_{xx}^e = \frac{t_h}{4A} \begin{bmatrix} b_i b_i & b_j b_i & b_k b_i \\ b_i b_j & b_j b_j & b_k b_j \\ b_i b_k & b_j b_k & b_k b_k \end{bmatrix} \quad \mathbf{k}_{yy}^e = \frac{t_h}{4A} \begin{bmatrix} c_i c_i & c_j c_i & c_k c_i \\ c_i c_j & c_j c_j & c_k c_j \\ c_i c_k & c_j c_k & c_k c_k \end{bmatrix}$$

Código

Montagem das Matrizes Globais

Algoritmo de montagem

$$\mathbf{m}_{en} = \begin{bmatrix} m_{ii} & m_{ij} & m_{ik} \\ m_{ji} & m_{jj} & m_{jk} \\ m_{ki} & m_{kj} & m_{kk} \end{bmatrix} \xrightarrow[\text{l=i,j,k}]{\text{loop}} \mathbf{M} = \begin{bmatrix} M_{0,0} & M_{0,1} & \dots & M_{0,np} \\ M_{1,0} & \ddots & & M_{1,np} \\ \vdots & & & \vdots \\ M_{I,q} + m_{I,q} & \ddots & & \vdots \\ M_{np,0} & & & M_{np,np} \end{bmatrix}$$

```

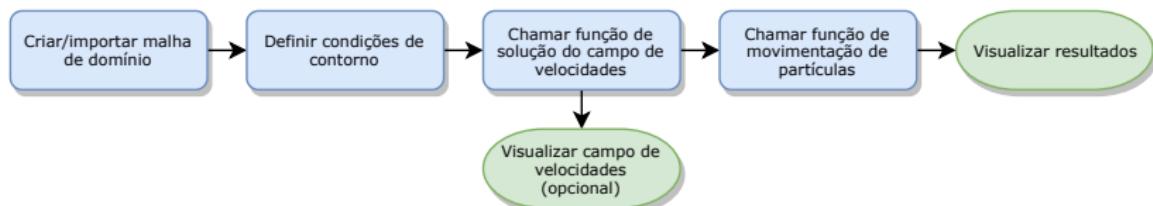
# Loop em cada elemento na lista da malha
for elem in malha.i.en:
    x = malha.x[elem] # = [x_i, x_j, x_k]
    y = malha.y[elem] # = [y_i, y_j, y_k]

    # Criação das matrizes locais
    ...

# Registro das matrizes locais nas matrizes globais
for i in range(3):
    for j in range(3):
        kx_global[elem[i], elem[j]] += k_x[i][j]
        ky_global[elem[i], elem[j]] += k_y[i][j]
        m_global[elem[i], elem[j]] += m[i][j]
        gx_global[elem[i], elem[j]] += g_x[i][j]
        gy_global[elem[i], elem[j]] += g_y[i][j]

```

Estrutura de Uso da Biblioteca



Fluxograma da lógica de uso da biblioteca pelo usuário.

Exemplo de uso da biblioteca

```
# Importação da biblioteca
import TccLib

# Importação da malha ou coordenadas de uma nova
malha = TccLib.Mesh("arquivo_da_malha.msh")
# ou malha = TccLib.Mesh([coordenadas (x, y)])

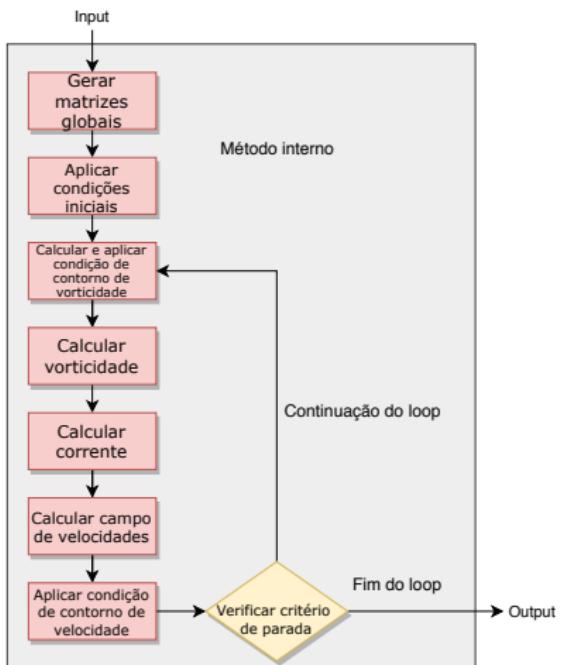
# Adição de partículas
malha.add_particle(propriedades da partícula)
```

```
# Definição das condições de contorno
malha.new_boundary_condition("nome da propriedade",
                             [índices dos nós],
                             [valor da condição no nó],
                             [1 para Dirichlet ou 0 para Neumann])

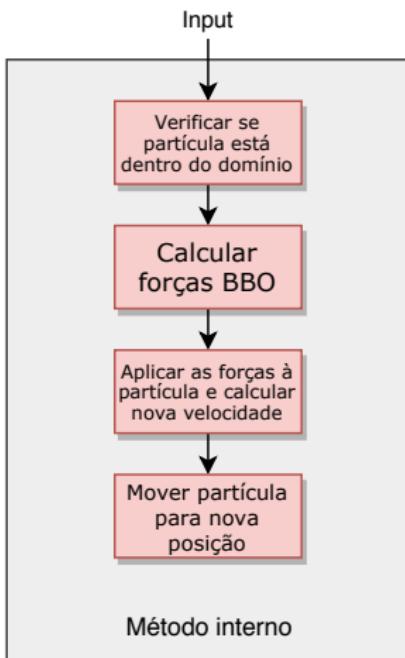
# Chamada para a função de solução
v_x, v_y = TccLib.solve_velocity_field(malha)

# Loop de movimentação das partículas
for t in time_list:
    TccLib.move_particles(malha, (v_x, v_y))
```

Estrutura de Solução



Algoritmo de solução do sistema de corrente-vorticidade.



Algoritmo de solução da posição das partículas.

Validações e Resultados

Configurações

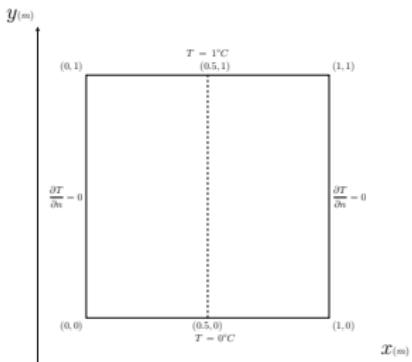
Especificação da Máquina Utilizada

- Dell Latitude E6410.
- Processador Intel® Core™ i5 CPU M 520 2.40GHz com 4 núcleos.
- 4Gb de memória RAM.
- O sistema operacional Ubuntu 16.04 LTS.
- Compilador Python 3.5.

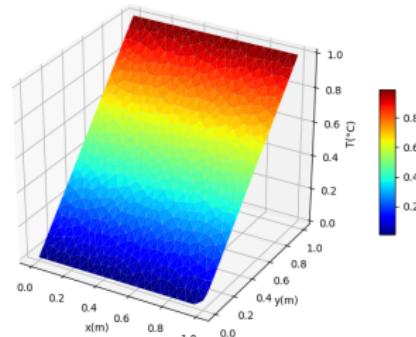
Especificação da Máquina Utilizada

Foram inseridas 5 partículas igualmente espaçadas por simulação.
As partículas foram colocadas próximas a entrada dos escoamentos.

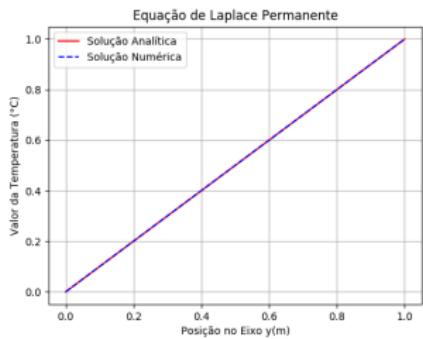
Validações de Problemas em Sólidos



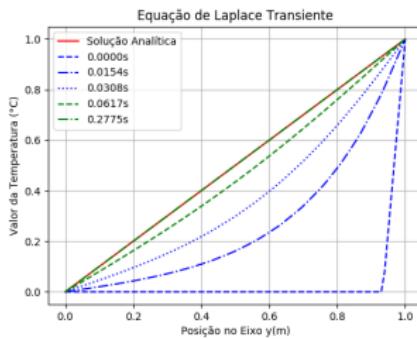
Condições de contorno em uma placa sólida.



Resultado da simulação na placa.

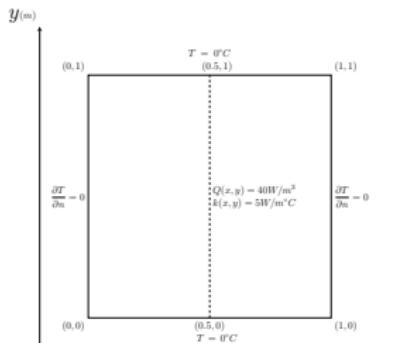


Comparação do resultado permanente.

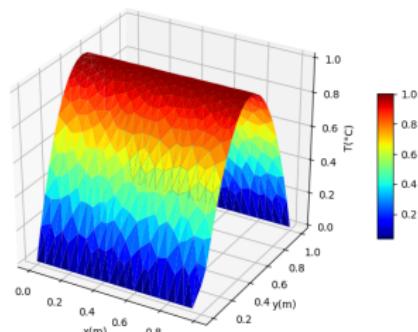


Comparação do resultado transitório.

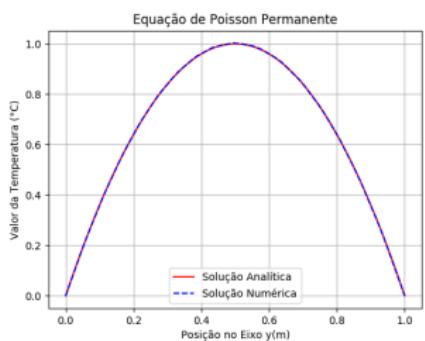
Validações de Problemas em Sólidos



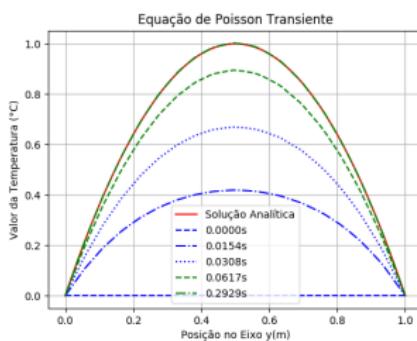
Placa com geração de calor.



Resultado da simulação na placa.

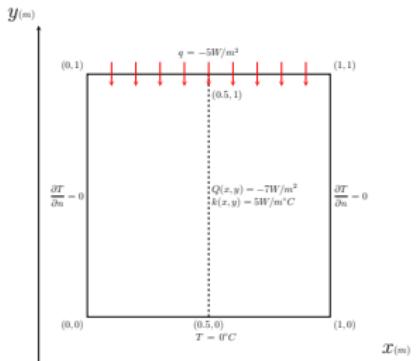


Comparação do resultado permanente.

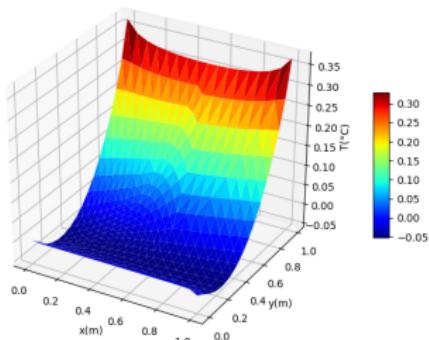


Comparação do resultado transitório.

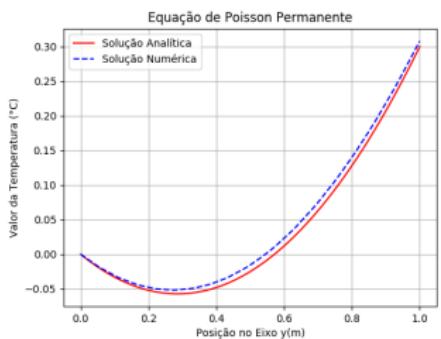
Validações de Problemas em Sólidos



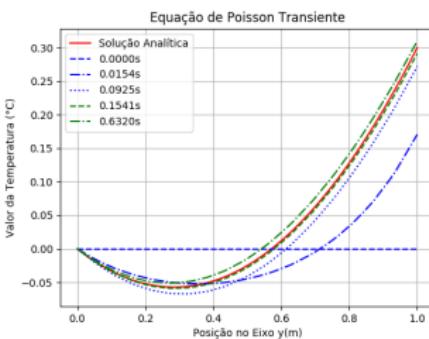
Placa com fluxo e geração de calor.



Resultado da simulação na placa.

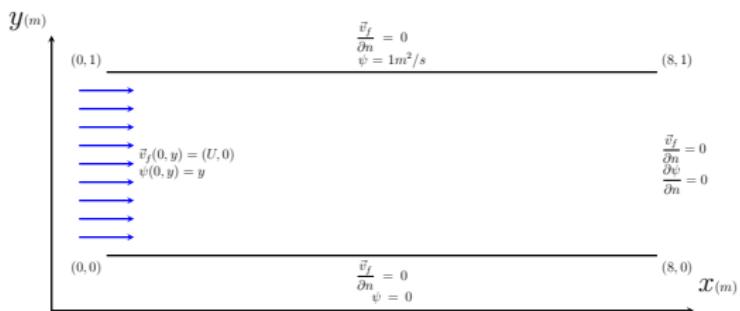


Comparação do resultado permanente.

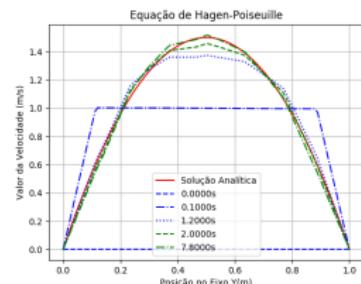


Comparação do resultado transitório.

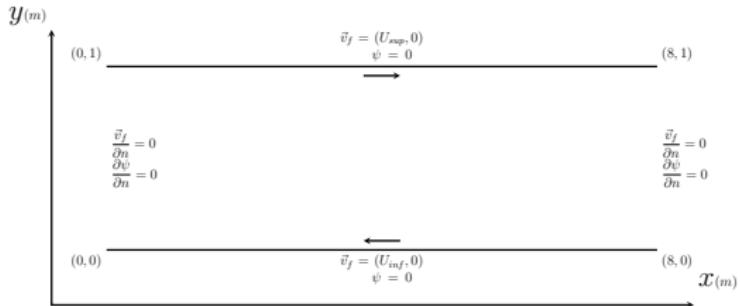
Validações do Modelo Corrente-Vorticidade



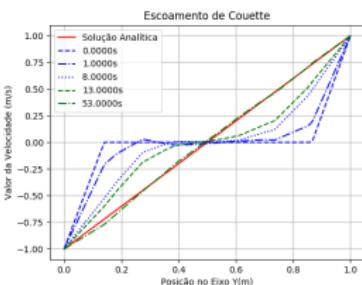
Escoamento entre placas estacionárias (Poiseuille).



Comparação com solução analítica.

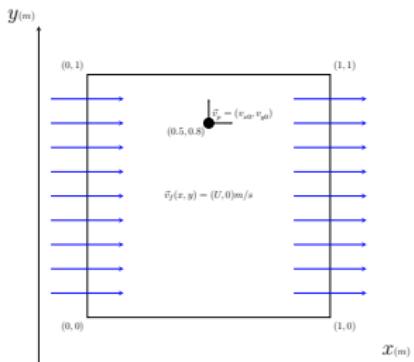


Escoamento entre placas em movimento (Couette).

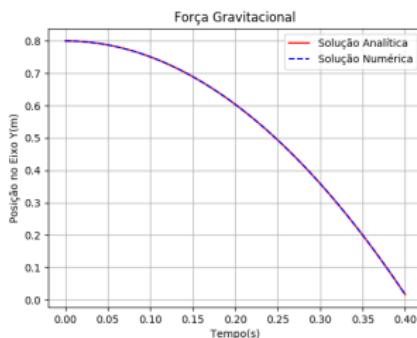


Comparação com solução analítica.

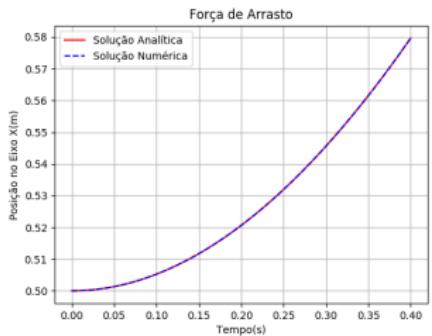
Validações das Forças nas Partículas



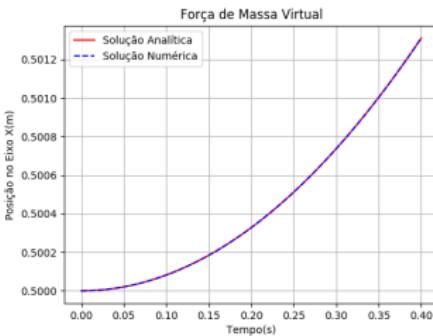
Condições de contorno das forças individuais.



Partícula sob efeito da força gravitacional.

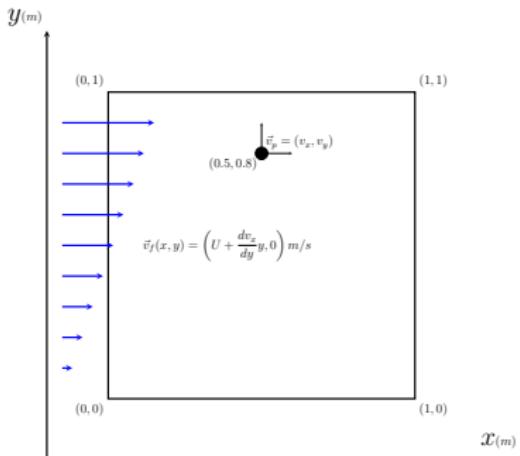


Partícula sob efeito da força de arrasto.

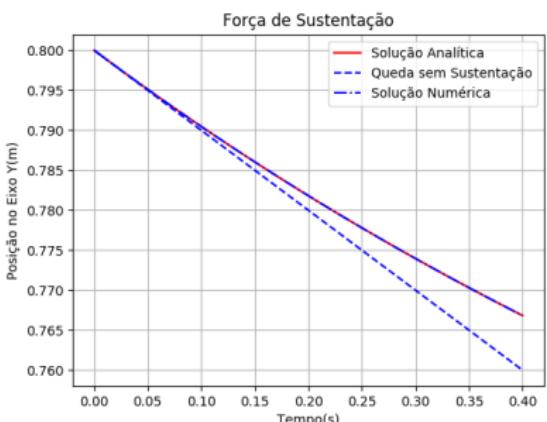


Partícula sob efeito da força de massa virtual.

Validações das Forças nas Partículas



Condições de contorno das força de sustentação.



Partícula sob efeito da força de sustentação.

Simulação em um Canal

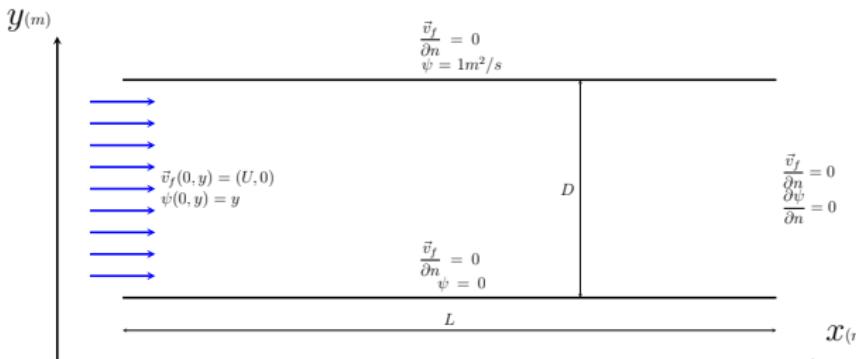
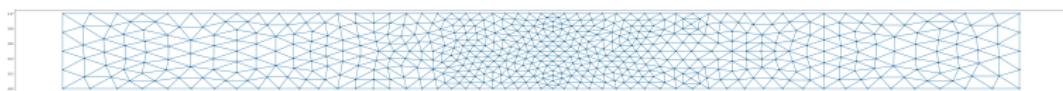


Diagrama da simulação.

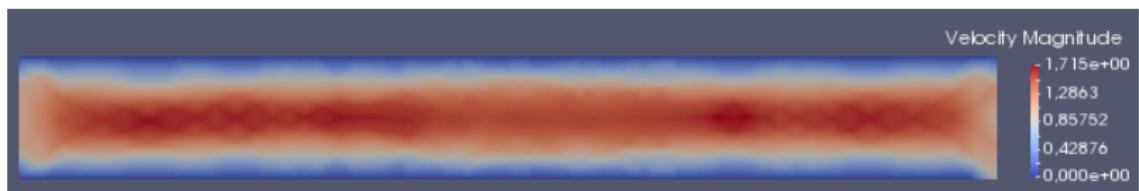
Parâmetros

$L = 8m$, $D = 1m$, $U = 1m/s$, $\mu_f = 50Pa.s$, $\rho_f = 50kg/m^3$,
 $d_p = 0.001m$, $\rho_p = 20000kg/m^3$.

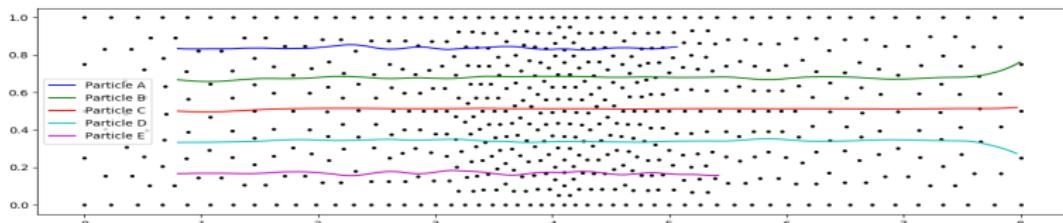
Simulação em um Canal



Malha do canal com 593 nós e 1072 elementos.



Campo de velocidades resultante, com $dt = 0.1s$ e tempo total $t = 20.0s$, realizada em torno de 1h.



Percorso das partículas na simulação, com $dt = 5E^{-5}$ e tempo total de $t = 6.0s$, realizada em torno de 6h.

Simulação em um Canal com Obstáculo

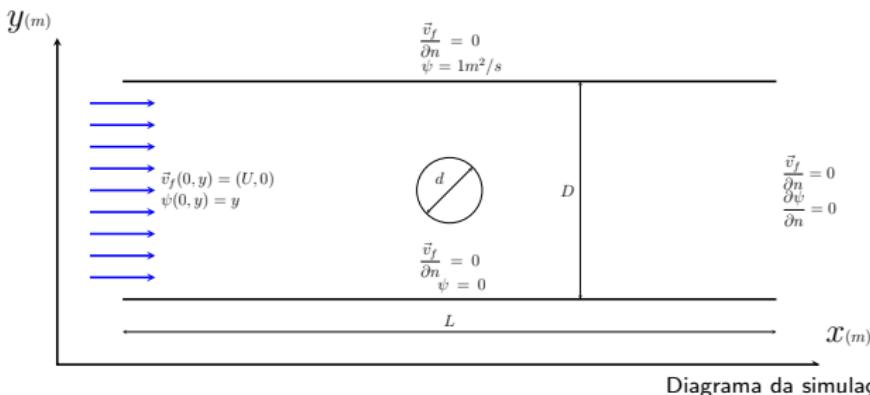
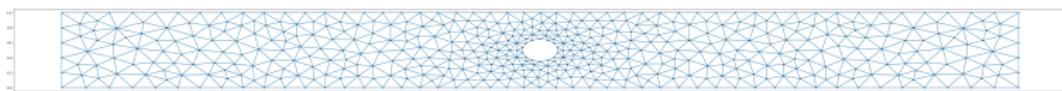


Diagrama da simulação.

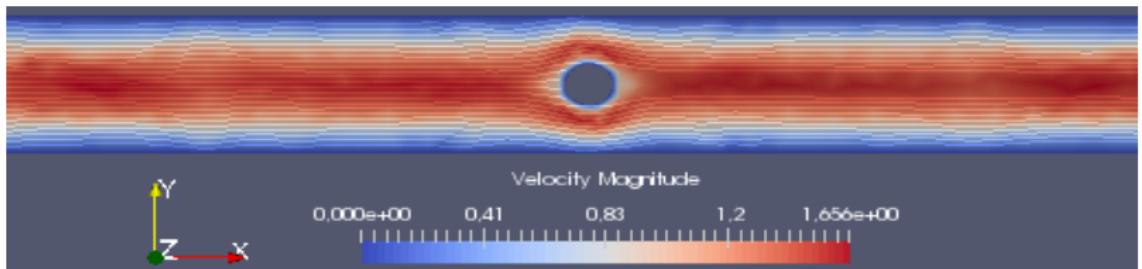
Parâmetros

$L = 8m$, $D = 1m$, $d = 0.3m$, $U = 1m/s$, $\mu_f = 50Pa.s$,
 $\rho_f = 50kg/m^3$, $d_p = 0.001m$, $\rho_p = 20000kg/m^3$.

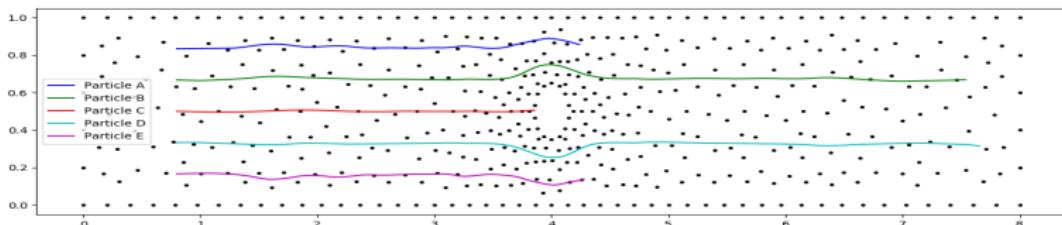
Simulação em um Canal com Obstáculo



Malha do canal com 494 nós e 868 elementos.



Campo de velocidades resultante, com $dt = 0.1s$ e tempo total $t = 20.0s$, realizada em torno de 1h.



Percorso das partículas na simulação, com $dt = 5E^{-5}$ e tempo total de $t = 5.0s$, realizada em torno de 4h.

Simulação em um Canal em Degrau

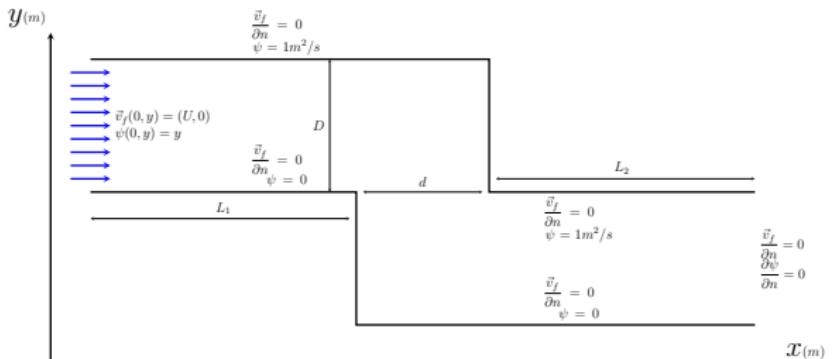
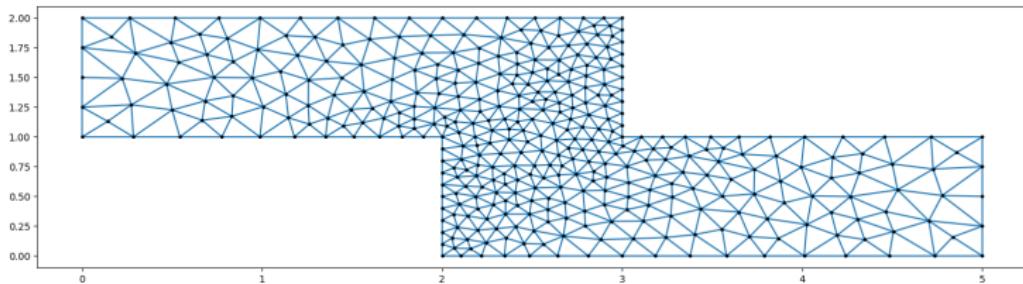


Diagrama da simulação.

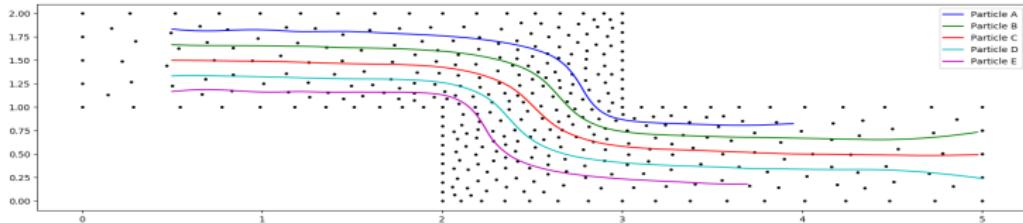
Parâmetros

$L_1 = 2m$, $L_2 = 2m$, $d = 1m$, $D = 1m$, $U = 1m/s$, $\mu_f = 50Pa.s$,
 $\rho_f = 50kg/m^3$, $d_p = 0.001m$, $\rho_p = 20000kg/m^3$.

Simulação em um Canal em Degrau

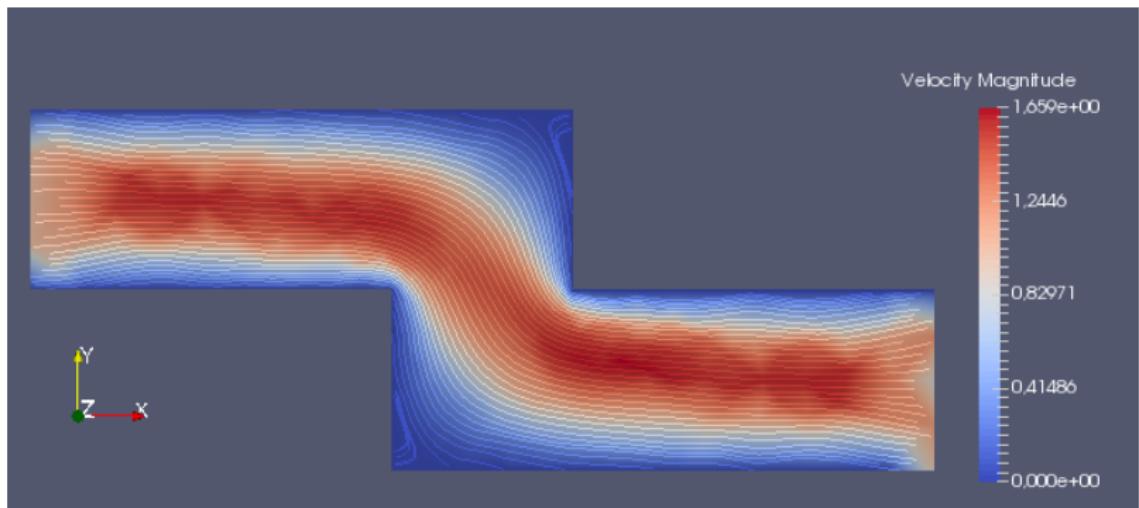


Malha do canal com 380 nós e 676 elementos.



Percorso das partículas na simulação, com $dt = 5E^{-5}$ e tempo total de $t = 5.0s$, realizada em torno de 4h.

Simulação em um Canal em Degrau



Campo de velocidades resultante, com $dt = 0.1s$ e tempo total $t = 20.0s$, realizada em torno de 1h.

Simulação em um Canal com Restrição

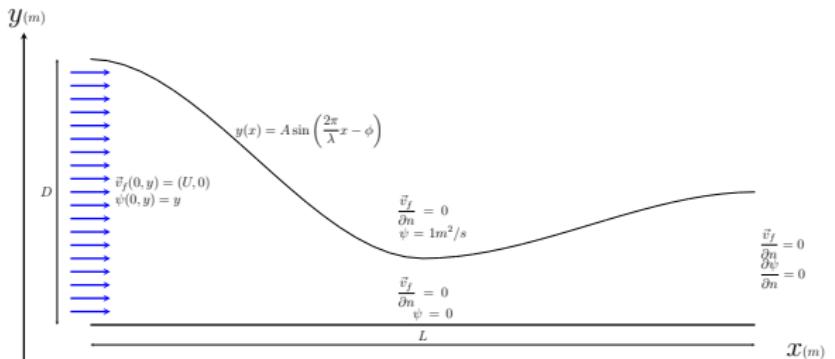
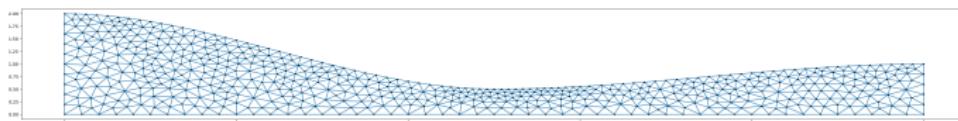


Diagrama da simulação.

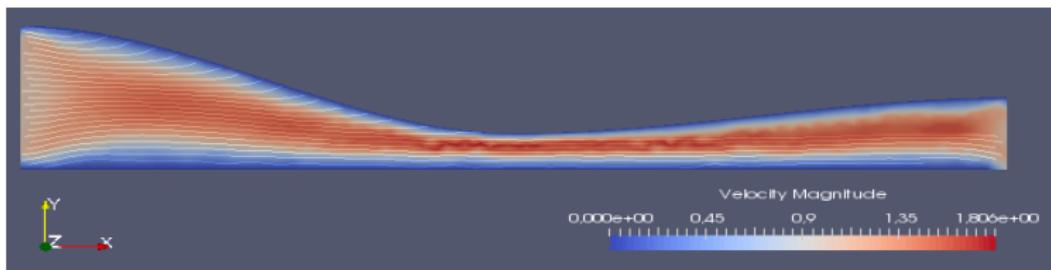
Parâmetros

$L = 8m$, $D = 2m$, $A = 0.004m$, $\lambda = 0.0006m$, $\phi = 0$, $U = 1m/s$,
 $\mu_f = 50Pa.s$, $\rho_f = 50kg/m^3$, $d_p = 0.001m$, $\rho_p = 20000kg/m^3$.

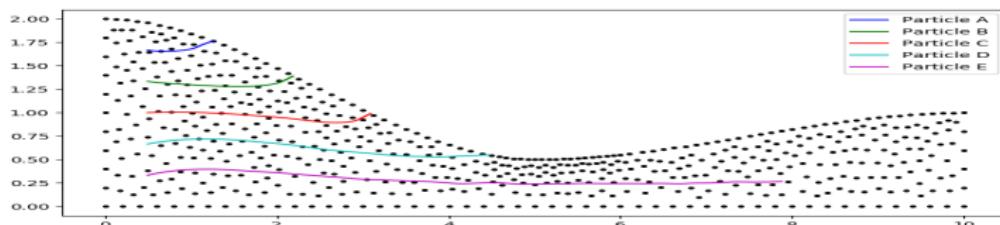
Simulação em um Canal com Restrição



Malha do canal com 600 nós e 1047 elementos.

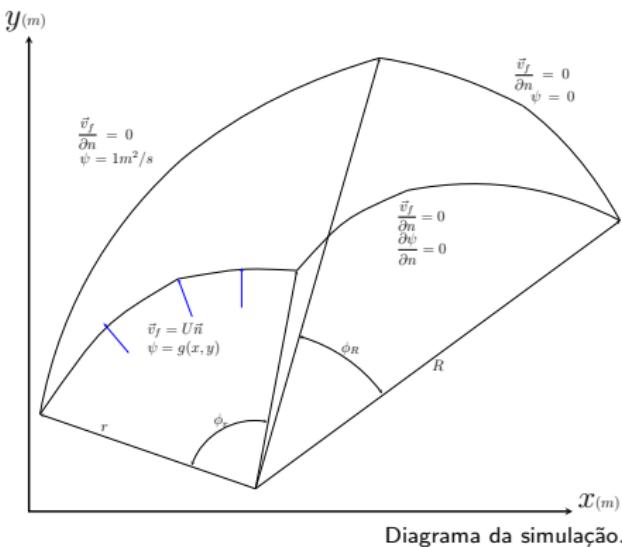


Campo de velocidades resultante, com $dt = 0.1s$ e tempo total $t = 20.0s$, realizada em torno de $2h$.



Percorso das partículas na simulação, com $dt = 5E^{-5}$ e tempo total de $t = 15.0s$, realizada em torno de $24h$.

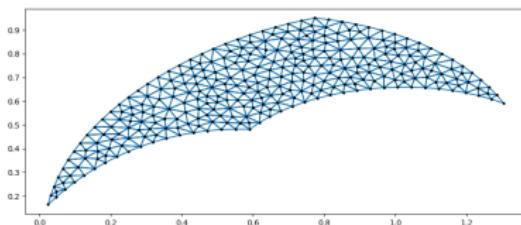
Simulações em um Impelidor



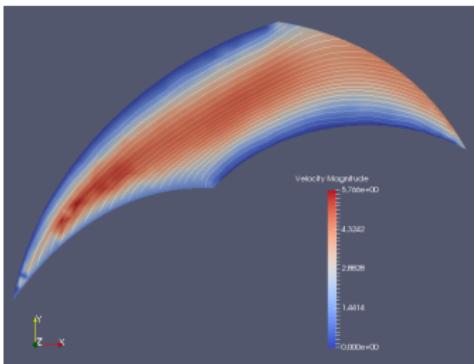
Parâmetros

$R = 1m$, $r = 0.75m$, $\phi_R = 34^\circ$, $\phi_r = 80^\circ$, $U = 1m/s$,
 $\mu_f = 50Pa.s$, $\rho_f = 50kg/m^3$, d_p, ρ_p = por caso.

Simulações em um Impelidor



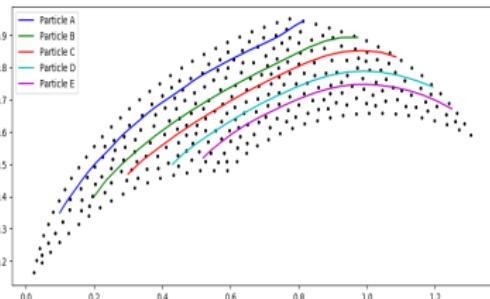
Malha do canal com 531 nós e 1058 elementos.



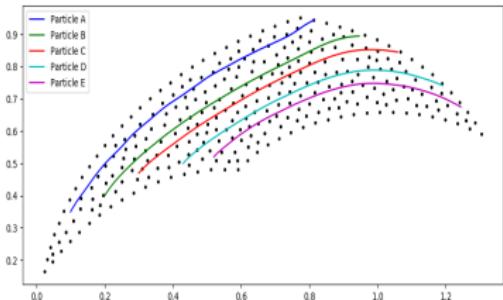
Campo de velocidades resultante, com $dt = 0.1s$ e tempo total $t = 20.0s$, realizada em torno de $2h$.

Simulações em um Impelidor

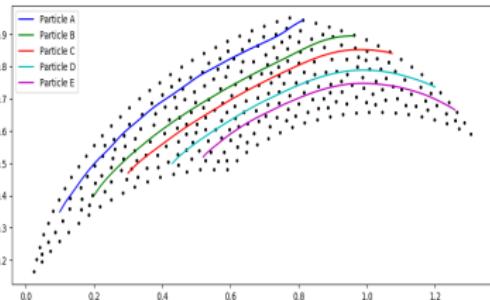
Simulações dos percursos das partículas efetuadas com $dt = 5E^{-5}$ e tempo total de $t = 5.0s$, realizadas em torno de 12h.



Partículas de ouro, $d_p = 1E^{-3} m$ e $\rho_{Au} = 2E^4 kg/m^3$.



Partículas de ferro, $d_p = 1E^{-3} m$ e $\rho_{Fe} = 7.3E^3 kg/m^3$.



Partículas de areia, $d_p = 5E^{-4} m$ e $\rho_p = 1.6E^3 kg/m^3$.

Conclusão

Conclusões

- A linguagem Python é uma excelente ferramenta para criar simuladores, ela permite realizar cálculos complexos rapidamente.
- O modelo e a biblioteca representam bem o fenômeno esperado.
- Para os casos estudados, força de arrasto possuí a maior contribuição ao movimento.
- Embora a biblioteca utilize o armazenamento de matrizes esparsas, as simulações levam bastante tempo.

Pontos fracos

- O simulador requer um grande esforço computacional.
- Não foi implementada a funcionalidade de *multithreading* na biblioteca.
- Restrição de uso para valores de Reynolds < 1 .

Agradecimentos



Muito Obrigado!