Análise por Elementos Finitos da Eficiência de Varrido de Colchões Lavadores em Anulares Afetados por Excentricidade e Erosões

Silas Aguiar Melo

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2022

Silas Aguiar Melo

Análise por Elementos Finitos da Eficiência de Varrido de Colchões Lavadores em Anulares Afetados por Excentricidade e Erosões

TCC apresentado ao curso de Matemática Computacionaldo Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Matemática Computacional.

Orientador: Gustavo Charles Peixoto de Oliveira

Outubro de 2022

Ficha catalográfica: elaborada pela biblioteca do CI.

Será impressa no verso da folha de rosto e não deverá ser contada.
Se não houver biblioteca, deixar em branco.

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão do Curso de Matemática Computacionalintitulado ***Análise por Elementos Finitos da Eficiência de Varrido de Colchões Lavadores em Anulares Afetados por Excentricidade e Erosões***, de autoria de Silas Aguiar Melo, aprovado por Banca Examinadora constituída pelos seguintes membros:

Prof. Dr. Gustavo Charles Peixoto de Oliveira
Universidade Federal da Paraíba

Prof. Dr. Moisés Dantas dos Santos
Universidade Federal da Paraíba

Prof. Dr. Gustavo Rabello dos Anjos
Universidade Federal do Rio de Janeiro

Dr. Bruno Leonardo de Sena Costa
Universidade Federal do Rio Grande do Norte

Coordenador(a) do Curso
Prof. Dr. Roberto Quirino do Nascimento

João Pessoa, 12 de outubro de 2022

RESUMO

A cimentação primária de poços é uma fase crucial das atividades de *upstream* na indústria de petróleo e gás para proteger o revestimento metálico instalado no poço, impedir a migração de fluidos por meio de fraturas e impedir efeitos adversos no espaço anular. Em particular, colchões lavadores são aplicados com a finalidade de remover impurezas remanescentes após a perfuração e facilitar a aderência do cimento à parede porosa. Neste trabalho, analisamos o comportamento hidrodinâmico de um colchão lavador com propriedades físicas equivalentes a de uma emulsão à base de óleo vegetal no espaço anular existente entre o revestimento (*casing*) e a formação rochosa. O escoamento é simulado de maneira simplificada usando a hipótese de fluido Newtoniano, modelado pelas equações de Navier-Stokes para o caso incompressível. Simulações numéricas são executadas via método de elementos finitos a partir da biblioteca FEniCs para configurações bidimensionais representativas de poços submetidos a excentricidades (taxas de *standoff* menores do que 100%) e erosões. Por fim, comparamos os efeitos de *standoff* e de erosões com modelos homólogos não erodidos e calculamos a eficiência de varrido para caso particular. Os resultados mostram que o perfil de velocidade é alterado quando consideramos malhas cujos contornos assemelham-se a paredes erodidas.

Palavras-chave: petróleo e gás, cimentação primária, limpeza de poços, dinâmica dos fluidos computacional.

ABSTRACT

Primary well cementing is a crucial stage of upstream activities in the oil and gas industry to protect the metallic casing inside the well, prevent fluid migration through fractures and avoid adverse effects on the annular space. In particular, flusher fluids are applied to clean post-drilling impurities and facilitate cement's adhesion to the porous wall. In this report, we analyzed the hydrodynamic behavior of a flusher with physical properties equivalent to a vegetable oil-based emulsion moving through the annular space between the casing and the rock formation wall. The incompressible fluid flow was simulated under a Newtonian hypothesis and modeled by the Navier-Stokes equations. Numerical simulations were performed via finite element method from the FEniCs library for two-dimensional configurations representing wells subjected to standoff rates less than 100% and erosion. Finally, we compared the effects of eccentricity and erosions against homologous non-eroded models and calculated the sweep efficiency for each case. Our findings show that the velocity profile is changed when one consider meshes whose boundaries mimic eroded walls.

Keywords: oil and gas, primary cementing, well cleaning, computational fluid dynamics.

LISTA DE FIGURAS

LISTA DE TABELAS

Sumário

1 INTRODUÇÃO

O primeiro poço de petróleo foi perfurado na Pensilvânia, Estados Unidos, em meados do século XIX. Esse poço tinha apenas 23 metros de profundidade quando começou seu ciclo produtivo. Naquela época, o petróleo, até então utilizado como combustível para lamparinas a óleo, gradativamente passou a ser destilado para produzir combustíveis, como o querosene. A descoberta de novas jazidas faria surgir cidades em pleno deserto americano, caracterizando o início da febre do "ouro negro". Dali em diante, a tecnologia envolvida na perfuração de poços de petróleo evoluiu consideravelmente e, com o advento da computação, houve grandes melhorias nos processos de extração [**?**].

A história do petróleo no Brasil começa com Eugênio Ferreira de Camargo, quem, em 1892, tentou encontrar petróleo na cidade paulista de Bofete após perfurar um poço de 488 metros. Todavia, achou-se apenas água. Essa foi a primeira tentativa para encontrar petróleo no Brasil. Entretanto, apenas em 1939 descobriu-se a primeira jazida de petróleo na cidade de Salvador, Bahia [**?**].

A demanda por petróleo ainda é elevada, visto que é matéria-prima para muitos produtos derivados, tais como plástico, asfalto e gasolina. Com o avanço tecnológico dos processos de recuperação, a indústria de petróleo e gás passou a explorar o recurso em lugares cada vez mais profundos, às vezes ultrapassando os 5000 metros, a exemplo do petróleo retirado do Pré-Sal brasileiro [**?**]. Entretanto, essa enorme profundidade leva a desafios operacionais no âmbito da perfuração, completação e estabilidade dos poços.

Durante a atividade de perfuração, injeta-se um fluido por dentro da coluna de perfuração, que avança desde a superfície até à broca. Na broca há um orifício por onde ele é expelido e segue o seu caminho de retorno à superfície passando pelo espaço anular existente entre a coluna de perfuração e a parede do poço. Durante o movimento de retorno, o fluido transporta cascalhos ou fragmentos de rocha que são gerados pela perfuração da formação. Esse fluido é chamado de *fluido de perfuração* [**?**].

No processo de perfuração, introduz-se um revestimento metálico com o intuito de proteger e isolar a formação rochosa [1]. Em seguida, o espaço anular compreendido entre o revestimento e a formação necessita de preenchimento. Esse preenchimento é realizado por meio do processo de cimentação [15]. A cimentação é uma das fases mais importantes na construção de poços de petróleo, a qual tem por função proteger o revestimento metálico instalado no poço e assim impedir o movimento de fluidos através do espaço anular [14].

O fluido de perfuração e a pasta de cimento são na maioria das vezes incompatíveis. Por isso, é necessário remover o fluido de perfuração para que a pasta de cimento tenha aderência à formação rochosa. Para tanto, são injetados fluidos intermediários antes da injeção do cimento que são chamados de "colchões" (*flushers*), a saber, o "colchão lavador"

e o “colchão espaçador”. Esses fluidos têm como função remover completamente o fluido de perfuração da parede do poço [**?**].

Há inúmeros desafios na extração do petróleo. Quanto mais profundo o poço, mais problemas surgem. Um desses problemas é a ocorrência de deslocamento por excentricidade, fenômeno usualmente chamado de *standoff*. O *standoff* ocorre quando o revestimento é inserido no poço e não se conforma perfeitamente centralizado até o fundo do poço, causando má eficiência no processo de limpeza. Consequentemente, a aderência do cimento à formação torna-se comprometida, provocando potenciais problemas de infiltração de fluidos indesejáveis para o interior do poço, além de problemas mecânicos e estruturais, tais como fraturas que podem induzir a demolição do poço [3].

O presente trabalho analisa o comportamento hidrodinâmico de um colchão lavador com propriedades físicas equivalentes às de uma emulsão a base de óleo vegetal no espaço anular existente entre o revestimento (*casing*) e a formação rochosa. Realizamos simulações de escoamento de maneira simplificada usando a hipótese de fluido newtoniano, modelado pelas equações de Navier-Stokes para o caso incompressível. Os experimentos numéricos são executados via método de elementos finitos a partir da biblioteca FEniCS para configurações bidimensionais representativas de poços submetidos a excentricidades (taxas de *standoff* menores do que 100%) e erosões. Por fim, os efeitos de *standoff* e de erosões foram comparados com modelos homólogos não erodidos e foi calculada a eficiência de varrido para o caso particular.

2 REVISÃO BIBLIOGRÁFICA

2.1 Mecânica dos Fluidos Aplicada à Engenharia de Poços de Petróleo

Fluidos são substâncias capazes de escoar que se deformam com facilidade. Gases, líquidos e inclusive o plasma são classificados como fluidos. A mecânica dos fluidos está presente em uma vasta gama de situações: sistema respiratório e circulatório, esportes náuticos, bombas hidráulicas, ventiladores, turbinas, aviões, navios, rios, moinhos de vento, mísseis supersônicos, icebergs, motores, filtros, jatos e sprinklers. Quase tudo no mundo é um fluido ou se move dentro ou perto de um fluido [5].

Os fluidos podem ser newtonianos ou não newtonianos. Fluido newtoniano é aquele que a viscosidade, ou atrito interno, é constante para diferentes taxas de cisalhamento. Nos fluidos newtonianos, a tensão é diretamente proporcional à taxa de deformação. Apesar de não existir um fluido perfeitamente newtoniano, fluidos considerados homogêneos, como a água e o ar, costumam ser estudados como newtonianos para muitas finalidades práticas [**?**].

Fluido não-newtoniano é aquele cuja viscosidade varia proporcionalmente à energia cinética que se imprime a esse mesmo fluido, respondendo de forma quase instantânea. Para exemplo temos a mistura do amido de milho com água que, dependendo da pressão que recebe, pode ser um sólido ou um líquido, apresentando característica viscosa. Com pressão suficiente, torna-se um sólido e com menor pressão volta ao estado líquido [**?**].

Outra característica de um fluido diz respeito à sua compressibilidade. Um fluido *compressível* é aquele que possui a habilidade de ser comprimido em um recipiente quando seu volume diminui por ação de uma força aplicada. A massa específica do fluido muda em relação ao grau de compressão sofrido . Já um fluido *incompressível* é aquele cuja massa específica permanece relativamente constante, ou seja, que resiste à redução de seu volume quando submetido a uma força compressiva [**?**].

2.1.1 Colchões

Para que haja remoção eficiente do fluido de perfuração na parede da formação no interior dos poços de petróleo, são injetados fluidos denominados colchões lavadores e colchões espaçadores antes da pasta de cimento, como dito anteriormente. Estes colchões têm distintas finalidades durante o processo de limpeza do poço.

Colchão lavador é um fluido com baixa viscosidade e baixa massa específica compatível com o fluido de perfuração. Por meio de ações químicas e mecânicas, ele dilui e remove o reboco (camada de baixa permeabilidade formado pelos fluidos de perfuração nas paredes do poço).

Colchão espaçador é, geralmente, um fluido com viscosidade e massa específica altas que atua apenas de forma mecânica para a remoção do reboco. Semelhantemente aos colchões lavadores, também são injetados antes da pasta de cimento com o intuito de evitar o contato dela com o fluido de perfuração. Eles ajudam a preservar eventuais problemas de cimentação, a exemplo da formação de canais de fluido no interior do cimento que prejudicam o isolamento hidráulico [13].

A Fig. 1 ilustra o funcionamento dos colchões na engenharia de poços. Podemos observar que o colchão lavador (cor azul clara), depois de injetado pelo interior do revestimento, retorna à superfície pelo espaço anular empurrando o fluido de perfuração responsável pelo carreamento de detritos (cor marrom). Em seguida, o colchão espaçador (cor verde), remove o restante dos fluidos anteriormente injetados para que a pasta de cimento (cor cinza) se adira às paredes da formação adequadamente [13].

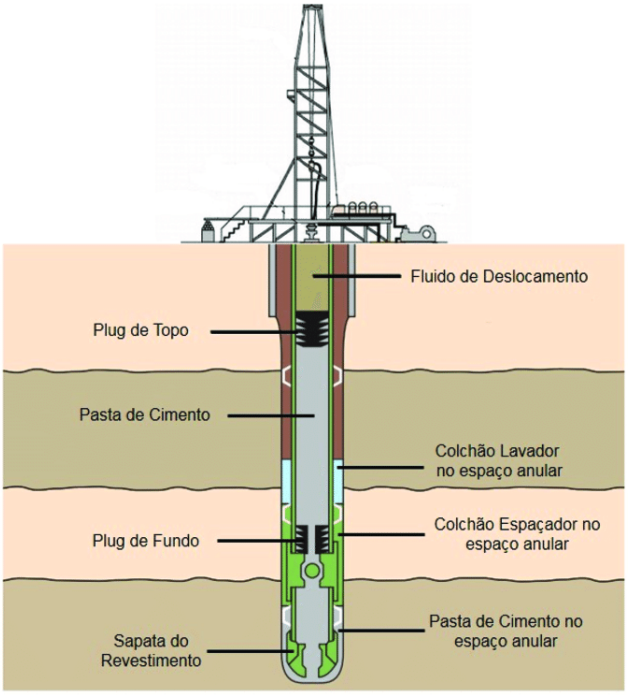

Figura 1: Remoção de fluido de perfuração por ação de colchões (lavador e espaçador) durante a cimentação primária de um poço de petróleo. Fonte: [10].

2.1.2 Excentricidade do Tubo de Revestimento

Quando o eixo do tubo de revestimento (*casing*) coincide com o eixo do poço perfurado, diz-se que o arranjo está sob uma configuração anular concêntrica. Neste caso, os fluidos deslocam-se no espaço anular de maneira relativamente simétrica, proporcionando aos processos de limpeza e cimentação certo grau de uniformidade na parede do poço. Entretanto, quando o tubo de revestimento está deslocado do centro, a configuração do anular torna-se excêntrica, os colchões tendem a fluir de maneira não uniforme. A presença

de excentricidade (*standoff*) no tubo de revestimento é um fator que influencia fortemente o deslocamento de fluidos durante os processos de perfuração e completação de poços de petróleo.

O comprometimento da eficiência de varredura pode ocorrer com qualquer fluido injetado, inclusive com a pasta de cimento. Em geral, para que não haja pontos da formação com lacunas de cimentação, centralizadores são instalados no tubo para manter o espaço anular o mais uniforme possível. A excentricidade entre poço e revestimento é medida, em geral, por um parâmetro popularmente conhecido como *standoff*. A taxa de standoff pode ser representada por um percentual que varia entre 0% (revestimento em contato com a parede) e 100% (revestimento afastado ao máximo da parede). Entretanto, os limites inferior e máximo são praticamente inatingíveis, permanecendo a taxa em valores intermediários. Obter valores genuinamente altos de *standoff* é uma tarefa extremamente difícil para a engenharia de poços. Além disso, o próprio revestimento pode dobrar ou ceder em alguns pontos no interstício entre um centralizador e outro, resultando em um *standoff* notadamente menor (*sag point*). Por isso, durante o trabalho de limpeza, o colchão lavador encontrará maior resistência para deslocar o fluido de perfuração na porção anular que estiver espremida, deixando para trás resquícios de detritos não carreados. Um exemplo desse problema é mostrado na Fig. 2, relativo a um incidente ocorrido com o poço de Macondo, no Golfo do México, em 2011 [8, **?**].

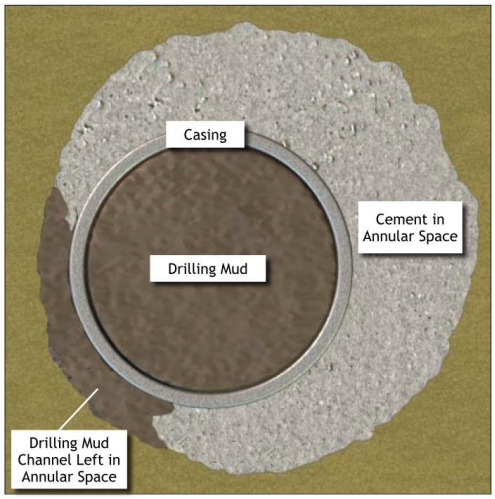

Figura 2: Seção transversal de um revestimento excêntrico.

A Fig. 3 mostra um caso de *standoff* menor entre dois centralizadores [**?**].

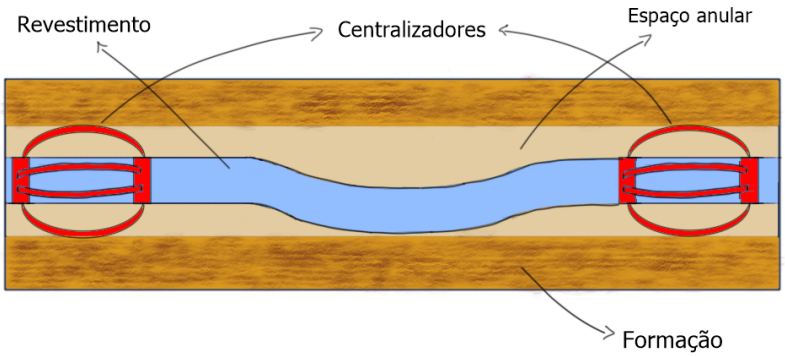

Figura 3: *Standoff* menor do que 100% entre dois centralizadores. Fonte: autor.

Considerando a seção transversal de um anel excêntrico, como se vê na Fig. 4, a taxa de *standoff* pode ser calculada como [3]:

$$\varsigma = (1 - \alpha)100\%, \quad \text{com} \quad \alpha = \frac{e}{r_o - r_i}, \tag{2.1}$$

onde e é a distância entre o centro do poço e o centro do tubo de revestimento, r_o é o raio do poço e r_i é o raio do tubo de revestimento. O cálculo do *standoff* usa um ponto de referência na parede do poço para definir o afastamento, especificamente na interseção inferior que se acha entre a circunferência do poço aberto e a linha de simetria que passa pelo ângulo desdobrado de π radianos (Fig. 4). Dessa forma, $\varsigma = 0\%$ equivale a dizer que o revestimento toca a parede do poço no tal ponto, enquanto que $\varsigma = 100\%$ equivale a dizer que o revestimento dista o máximo possível desse ponto. Em outras palavras, $\varsigma = 100\%$ significaria concentricidade "perfeita".

$$R_{STO} = \left[\left(1-\varepsilon\right)\times 100\right]. \qquad (4\text{-}163)$$

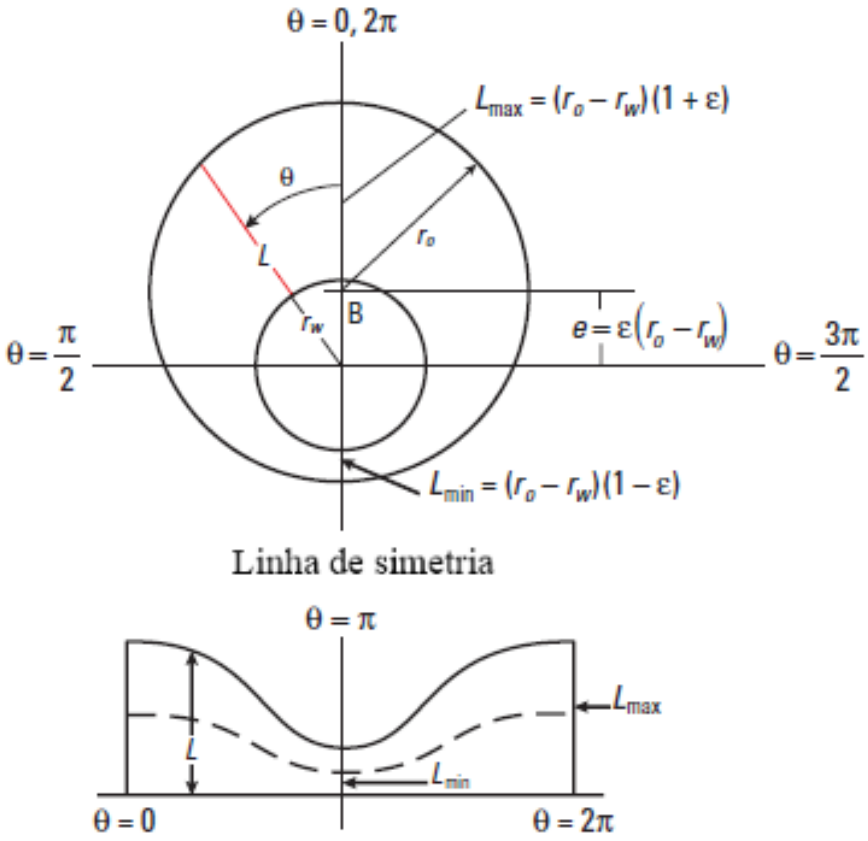

Figura 4: Seção transversal equivalente a um revestimento excêntrico. Fonte: [3]

Neste trabalho, consideramos simulações em que $0\% < \varsigma \leq 100\%$. Em particular, o caso em que ς é máximo é utilizado como modelo ideal de referência.

2.2 Objetivos de Pesquisa

A Matemática Computacional é capaz de propor modelos matemáticos para resolver problemas reais de alta complexidade. Aqui, procuramos dar enfoque à melhoria de processos industriais relacionados à perfuração de poços de petróleo, um tema interdisciplinar que integra diversas áreas de conhecimento, tais como matemática, física, computação e as engenharias química, mecânica e de petróleo.

Estudar o comportamento do escoamento de colchões lavadores e sua eficiência de limpeza em diferentes configurações de poços traz benefícios para o desenvolvimento científico e tecnológico nacional, além de motivar talentos para a pesquisa. A seguir, destacamos os principais objetivos estabelecidos para a pesquisa.

2.2.1 Objetivo Geral

Simular numericamente o escoamento de colchões lavadores caracterizados por propriedades constituintes similares às de óleos biodegradáveis não inflamáveis que se desenvolvem durante a etapa de pré-cimentação de poços de petróleo.

2.2.2 Objetivos Específicos

- Aplicar o método dos elementos finitos para solucionar numericamente as equações de Navier-Stokes para fluidos incompressíveis;
- Implementar códigos computacionais tomando como base a biblioteca FEniCS;
- Gerar malhas numéricas para diferentes características geométricas de poço e de espaço anular, simulando efeitos de erosão e excentricidade (*standoff*);
- Definir um parâmetro para quantificar a eficiência de varredura do escoamento em regime laminar;
- Analisar resultados de simulação para configurações de poço bidimensionais;

3 METODOLOGIA

3.1 Geração de Malhas

Para que seja possível aplicar o método dos elementos finitos, é necessário discretizar o domínio em elementos. O domínio discretizado chama-se malha, sobre a qual as simulações numéricas de escoamento devem ser executadas. A Fig. 5 ilustra a perfuração de um poço de petróleo, onde a região com hachuras em amarelo caracteriza o domínio a ser modelado.

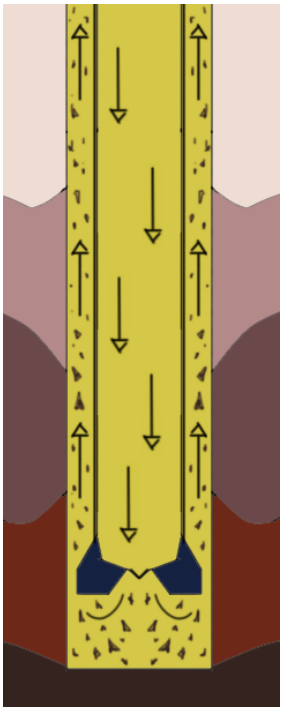

Figura 5: Ilustração de uma perfuração de um poço de petróleo. Fonte: autor

A Fig. 6 é uma representação esquemática de um poço de petróleo, cuja ilustração foi utilizada como referência para criar o domínio geométrico no software AutoCad e, a partir dele, gerar o malhamento triangular. As setas representam o sentido do escoamento. As dimensões do revestimento são: 8,66 pol (aproximadamente 0,220 m) de largura e 39,37 polegadas (aproximadamente 1 m) de comprimento [9]. Simplificadamente, o comprimento do revestimento foi limitado a 1 m de extensão e a rugosidade na parede da formação simulada como uma interpolação por *spline* de pontos definidos "quase-aleatórios".

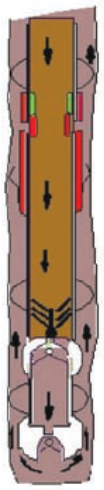

Figura 6: Representação esquemática de um poço de petróleo [3].

A Fig. 7 é o resultado do desenho no AutoCad do modelo simétrico com ς máximo, posteriormente editada no software Paint para evidenciar o sentido do escoamento dos fluidos, bem como os contornos de entrada (*inflow*) e saída (*outflow*) de fluido – em verde e azul –, e também as paredes (*wall*) – em marrom.

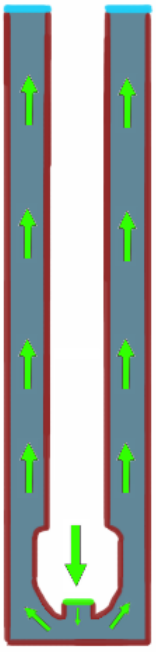

Figura 7: Representação dos contornos da malha. Fonte: autor.

Utilizamos o software Gmsh para gerar a malha [**?**]. Gmsh é um programa de código aberto com o objetivo de gerar malha de elementos finitos 1D, 2D e 3D com suporte integrado a CAD. A filosofia por detrás do Gmsh é contribuir com a facilidade e rapidez na construção de uma malha. Para criar modelos, pode-se usar a interface gráfica ou a sua

própria linguagem nativa por meio de *scripts*. Há também uma API para as linguagens de programação C/C++, Python e Julia [**?**]. Apesar disso, a facilidade de se trabalhar e a quantidade de recursos superior do AutoCad motivaram-nos escolhê-lo para construir o modelo. O modelo geométrico é dividido em vários elementos, tais como pontos, linhas, triângulos, quadriláteros, tetraedros e hexaedros, conectados entre si por vértices, ou nós. O Gmsh implementa vários algoritmos para gerar malhas automaticamente. As malhas em geral são não estruturadas. isto é, não existe nenhuma relação predefinida entre quaisquer dois elementos [**?**].

O Gmsh pode ler muitos tipos de arquivos. Um deles é o formato IGES, exportável pelo AutoCad. Portanto, para que haja integração entre o Gmsh e o AutoCad, trabalhamos com arquivos do tipo IGES. Normalmente, é útil combinar alguns elementos geométricos em grupos com o objetivo de definir propriedades matemáticas, tais como domínio, condições de contorno, funcionais ou materiais. Esse agrupamento pode ser feito no módulo *Geometry* do Gmsh através do estabelecimento de "grupos físicos".

Para as malhas consideradas aqui, definimos 4 grupos físicos: `inflow`, `outflow`, `wall` e `flow`. Como já dissemos, o primeiro e o segundo grupo correspondem à entrada e saída de fluido. As paredes do poço, a saber, a formação rochosa nua, bem como as paredes do revestimento são tratadas pela condição de não escorregamento (*no-slip*), onde a velocidade é assumida nula. Por fim, o interior do domínio, `flow`, é tratado como uma superfície física. Todas as malhas são não estruturadas e compostas de elementos triangulare com refinamento adaptativo na região de `inflow` próxima à sapata, conforme mostra a Fig. 8. Como fase final, exportamas as malhas no formato `.msh` para, em seguida, serem processadas no *solver* da biblioteca FEniCS.

Consideramos 4 modelos, aqui denominados A_1, A_2, B_1 e B_2, cujas dimensões são iguais, com 1 metro de altura e 0.22 metros de comprimento, as condições iniciais e de contorno são idênticas. As propriedades do fluido simulado também são as mesmas para todas as geometrias, assim como os parâmetros de simulação. Os modelos A possuem paredes lisas, ao passo que os modelos B admitem rugosidades. Para os modelos A_1 e B_1, $\varsigma = 100\%$; para A_2 e B_2, $\varsigma < 100\%$. As malhas A são representadas na figura Figs. 8 com expansão da região da base do poço. As malhas B são representadas na Fig. 9 também com expansão da região da base do poço, ilustrando as diferentes malhas utilizadas.

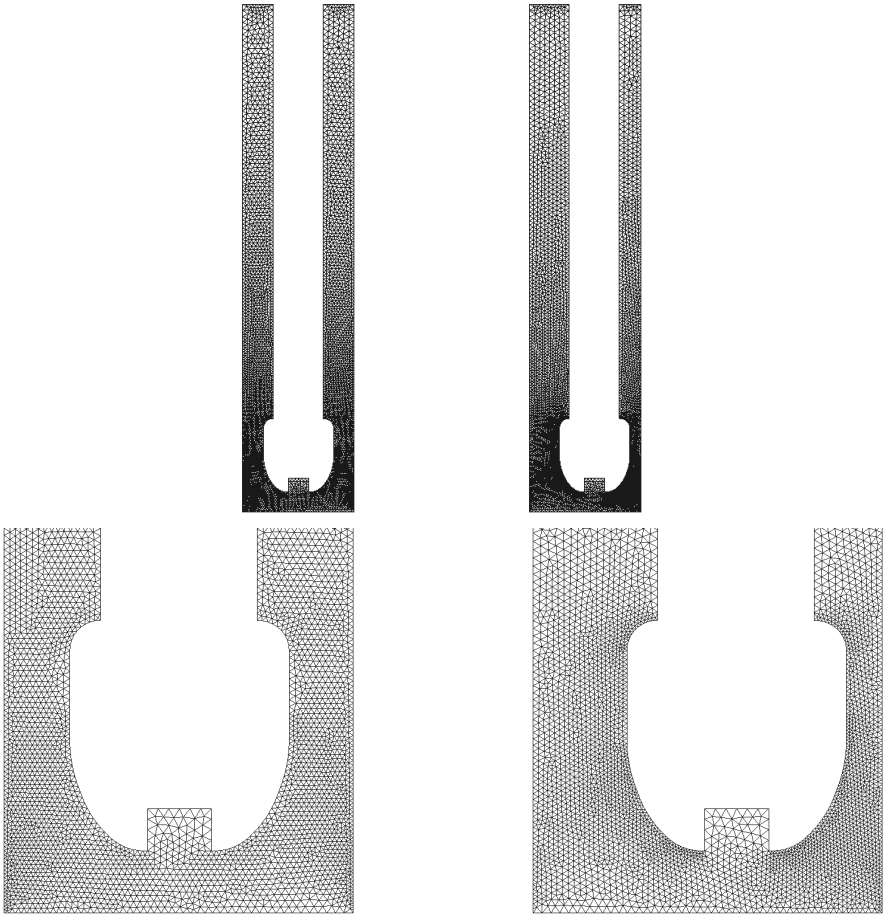

Figura 8: Malhas da configuração A: anular e sapata. Fonte: autor.

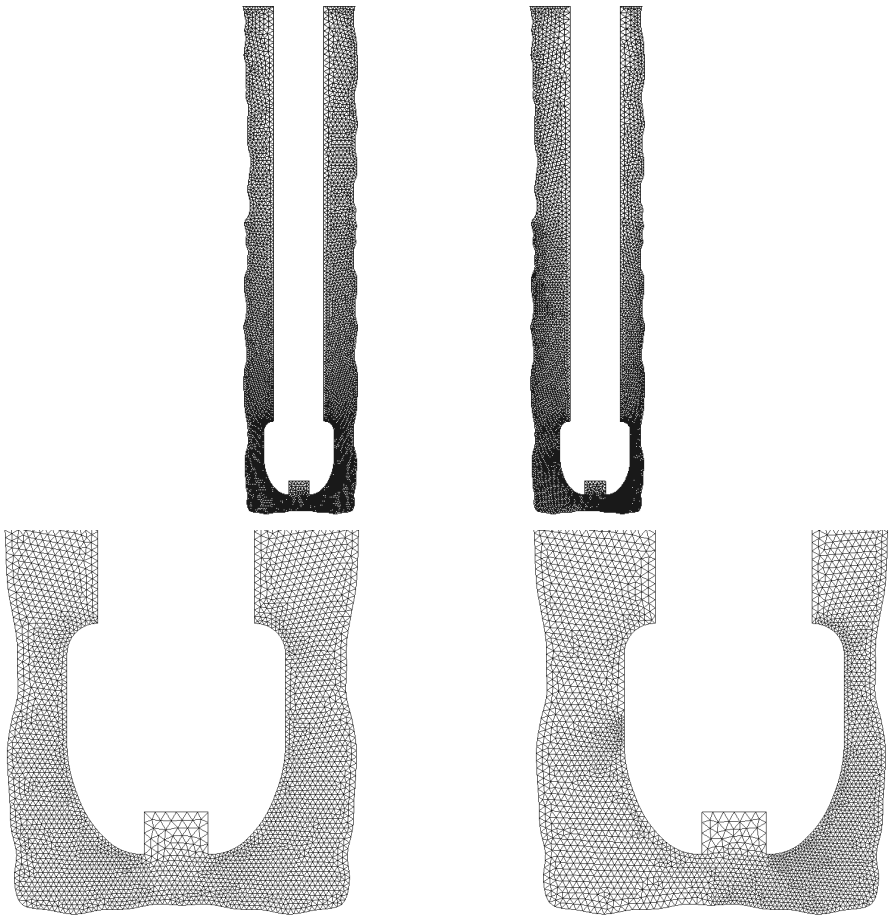

Figura 9: Malhas da configuração B: anular e sapata. Fonte: autor.

A Tab. 1 resume os principais parâmetros de malha e a taxa de *standoff* considerados para cada modelo. Onde n_{elem}, n_{ver} e n_{nos} é o número de elementos, de vértice e de nós da malha, respectivamente.

Tabela 1: Principais parâmetros de malha.

modelo	n_{elem}	n_{ver}	n_{nos}	ς
A_1	12520	6698	44828	100%
A_2	12986	6931	40458	50%
B_1	14706	7810	53278	100%
B_2	14550	7732	52654	50%

foi utilizado elementos triangulares de Taylor-Hood [11] com interpolação quadrática para a velocidade e interpolação linear para a pressão. Esses elementos são denotados como P2P1. Além de estáveis, esses elementos também convergem quadraticamente. Cada componente de velocidade será caracterizada por 6 nós enquanto a pressão será caracterizada por 3 nós Fig. 10

Figura 10: Elementos de triângulo linear e quadrático. A figura (a) corresponde aos elementos lineares de pressão e a figura (b) corresponde aos elementos quadráticos para cada componente da velocidade. Fonte: [12]

3.2 A Biblioteca FEniCS

FEniCS é uma ferramenta computacional de código aberto que tem por objetivo a resolução de equações diferenciais parciais (EDPs). O FEniCS oferece rapidez para traduzir modelos físico-matemáticos em códigos de elementos finitos escritos nas linguagens de programação Python e C++ [**?**]. No programa estão implementadas a biblioteca fundamental DOLFIN e outras funcionalidades especializadas.

O DOLFIN fornece um ambiente de solução de problemas para modelos baseados em EDPs, incluindo estruturas de dados e algoritmos para malhas computacionais e abstrações de elementos finitos. O DOLFIN fornece classes para operar com matrizes, vetores, elementos diversos e espaços funcionais, entidades importantes para computação de elemento finitos. A interface foi projetada com o propósito de ser simples. Para resolver EDPs usando a interface DOLFIN, os usuários devem expressar os problemas na forma variacional usando a linguagem específica UFL [6]. A Fig. 11 exemplifica os componentes mais relevantes da biblioteca DOLFIN.

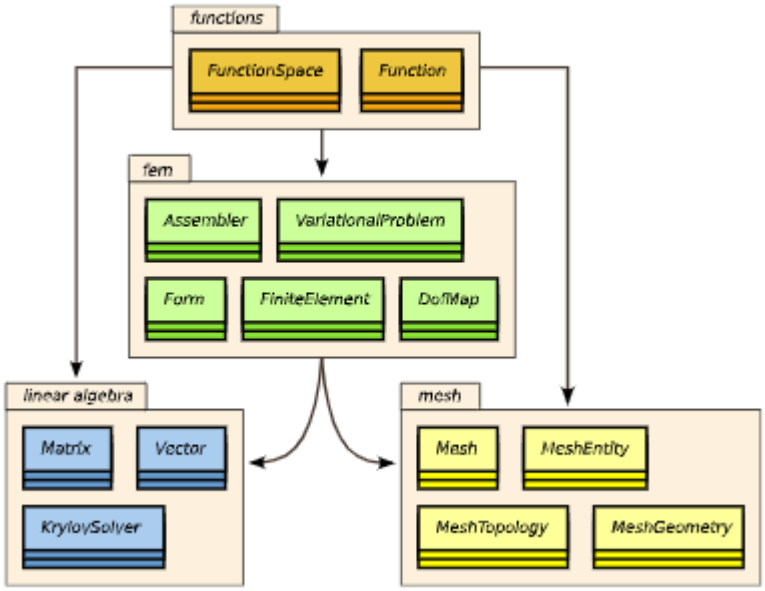

Figura 11: Componentes mais relevantes da biblioteca DOLFIN [6].

3.3 Equações de Navier Stokes

As equações de Navier Stokes (ENS) descrevem o movimento de fluidos viscosos. Estas equações foram derivadas originalmente na década de 1840 por Claude-Louis Navier e George Gabriel Stokes com base nas leis de conservação de massa, momento linear e energia. Elas permitem determinar os campos de pressão e de velocidade em um determinado escoamento [**?**]. As ENS podem ser utilizadas para modelar correntes oceânicas, escoamentos em dutos, escoamentos externos sobre aerofólios e aeronaves, hemodinâmica, entre outras aplicações [2].

Assumiremos neste trabalho que os fluidos escoantes são incompressíveis. Assim, consideraremos a equação da quantidade de movimento escritas na seguinte forma:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) + \nabla\cdot\boldsymbol{\sigma} + \boldsymbol{f} = \boldsymbol{0} \tag{3.1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{3.2}$$

onde ρ é a massa específica, $\boldsymbol{u}$ a velocidade, t o tempo, p a pressão e $\boldsymbol{f}$ a força de corpo por unidade de volume, a ser especificada adiante. Chamado de tensor de tensões de Cauchy, $\boldsymbol{\sigma}$ relaciona-se à pressão pela equação constitutiva para fluidos newtonianos dada por

$$\boldsymbol{\sigma} = 2\mu\left(\boldsymbol{u}\right) - p\boldsymbol{I}, \tag{3.3}$$

onde $\boldsymbol{I}$ é o tensor identidade. As tensões cisalhantes e a taxa de deformação em fluidos newtonianos relacionam-se de modo linear pela constante de proporcionalidade μ, a

viscosidade cinemática. O tensor de deformação $\boldsymbol{\epsilon}$ é dado por [**?**]:

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T). \tag{3.4}$$

Tratando a força de corpo como a gravidade, isto é, $\boldsymbol{f} = \rho \boldsymbol{g}$, a substituição das Eqs. (3.3) e (3.4) na Eq. (3.5) permite-nos reescrever a equação da quantidade de movimentos como as ENS:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) - \nabla p + \nabla \cdot [\mu(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)] + \rho \boldsymbol{g} = \boldsymbol{0} \tag{3.5}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{3.6}$$

tendo em vista que $\nabla \cdot p\boldsymbol{I} = \nabla p$.

3.3.1 Formulação Variacional e Discretização das Equações

A formulação padrão de Galerkin para as ENS incompressíveis, baseada em polinômios interpoladores de mesma ordem para velocidade e pressão, apresenta instabilidades. Portanto, consideramos neste trabalho espaços de elementos finitos do tipo Taylor-Hood (P2/P1) para a discretização espacial, com polinômios interpoladores quadráticos para a velocidade e lineares para a pressão, além de um método de projeção [**?**] conhecido como *Esquema de Correção Incremental da Pressão*, ou IPCS, do inglês *Incremental Pressure Correction Scheme* [**?**].

A seguir, explicamos brevemente a formulação variacional, bem como a discretização temporal associadas às ENS descritas na subseção 3.3. Detalhes podem ser encontrados em [7]. Primeiramente, definimos o produto interno genérico

$$\langle v, w \rangle_\Phi = \int_\Phi vw \, d\Phi, \tag{3.7}$$

, para funções contínuas v e w definidas em Φ. Neste texto, $\Phi = \Omega$ representa o interior do domínio onde as ENS são resolvidas e $\Phi = \partial\Omega$ representa o contorno.

Para estabelecer a forma variacional para as ENS, algumas hipóteses matemáticas devem ser feitas sobre os campos de velocidade e pressão, entre elas a de continuidade, variação limitada e integrabilidade. Em linhas gerais, devemos buscar uma solução aproximada para as incógnitas $\boldsymbol{u}$ e p a partir de uma forma "enfraquecida" que é obtida por meio de integrais ponderadas.

Escolhendo-se funções $\boldsymbol{v}$ e q, a primeira vetorial e a segunda escalar, para agirem como funções de ponderação, multiplicamos as Eqs. (3.5) e (3.5), respectivamente, por $\boldsymbol{v}$ e

q. Após integração por partes e algumas operações algébricas, chegamos à forma compacta

$$\langle \rho(\boldsymbol{u}^* - \boldsymbol{u}^n)/\Delta t, \boldsymbol{v}\rangle_\Omega + \langle \rho \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n, \boldsymbol{v}\rangle_\Omega + \langle \sigma(\boldsymbol{u}^{n+\frac{1}{2}}, p^n), \boldsymbol{\epsilon}(\boldsymbol{v})\rangle_\Omega + \\ -\langle \boldsymbol{n} p^n, \boldsymbol{v}\rangle_{\partial\Omega} + \langle \mu \boldsymbol{n} \nabla \boldsymbol{u}^{n+\frac{1}{2}}, \boldsymbol{v}\rangle_{\partial\Omega} + \langle \rho^{n+1}, \boldsymbol{v}\rangle_{\partial\Omega} = \boldsymbol{0}, \tag{3.8}$$

que caracteriza a forma variacional já discretizada no tempo para um passo de tempo Δt. A Eq. (3.8) é, além disso, o primeiro passo do método ICPS, em que a velocidade tentativa $\boldsymbol{u}^*$ é uma estimativa que não obedece à restrição de divergência nula dada pela Eq.(3.2).

A notação $\boldsymbol{u}^{n+\frac{1}{2}}$ sugere uma aproximação implícita para $\boldsymbol{u}$ avaliada no ponto médio temporal, ou seja,

$$\boldsymbol{u}^{n+\frac{1}{2}} \approx (\boldsymbol{u}^n + \boldsymbol{u}^{n+1})/2.$$

No segundo passo do método IPCS, calculamos a estimativa para a pressão resolvendo uma equação tipo Poisson usando a velocidade tentativa então calculada

$$\langle \nabla p^{n+1}, \nabla q\rangle_\Omega = \langle \nabla p^n, \nabla q\rangle_\Omega - \Delta t^{-1} \langle \nabla \cdot \boldsymbol{u}^*, q\rangle_\Omega, \tag{3.9}$$

Agora que temos a nova pressão, basta calcular a velocidade corrigida $\boldsymbol{u}^{n+1}$, tal que $\nabla \cdot \boldsymbol{u}^{n+1} = 0$. Este é o terceira e último passo do método ICPS:

$$\langle \boldsymbol{u}^{n+1}, \boldsymbol{v}\rangle_\Omega = \langle \boldsymbol{u}^*, \boldsymbol{v}\rangle_\Omega - \Delta t \langle \nabla(p^{n+1} - p^n), \boldsymbol{v}\rangle_\Omega. \tag{3.10}$$

Em suma, para cada passo de tempo no processo iterativo de solução das ENS, as 3 equações acima são resolvidas por um processo de *splitting* [7].

3.3.2 Implementação Computacional

A implementação computacional das equações discretas foi realizada na linguagem Python. A seguir, inserimos porções compactas do código aplicado. A versão completa está disponível no Apêndice. Após importar todas as bibliotecas necessárias, importar a malha que foi gerada via Gmsh como .msh como mostra o cód no apendice. As funções `create_mesh` e `mvc_mf`, servem para extrair os grupos físicos definidos no Gmsh, para identificação direta pelo FEniCS.

A partir da malha importada e da identificação dos grupos físicos, as propriedades de fluido, os parâmetros de simulação, tempo, quantidade de iteração, assim como as condições de contorno e iniciais podem ser definidos na função `dados_entrada`.

A função `def` descreve o modelo matemático das equações de Navier-Stokes. As linhas 5-7 definem a velocidade inicial e as linhas 9-14 as condições de contorno. As linhas 35-39 referem-se às equações 3.4 e 3.3, que representam, respectivamente, o tensor de

tensões de Cauchy e o tensor de deformação.

Depois de importar a biblioteca, definir os dados do problema, importar a malha, informar as condições de contorno, implementa-se o método IPCS. As linhas 41-48 referem-se às equações 3.8, 3.9 e 3.10. As linhas 50-52 realizam a correção da pressão. Com a pressão corrigida, calcula-se a velocidade corrigida.

```
1  def modelo(mesh, dados_problema,dados_fluidos, cfl, mf):
2      V = VectorFunctionSpace(mesh, 'P', 2)
3      Q = FunctionSpace(mesh, 'P', 1)
4
5      inflow_profile = ('0' +
6      str(dados_problema['velocidade_fluido_inflowX']), '0' +
7      str(dados_problema['velocidade_fluido_inflowY']))
8
9      bcu_inflow = DirichletBC(V, Expression(inflow_profile,
10     degree=2), mf, 1)
11     bcp_outflow = DirichletBC(Q, Constant(0), mf, 2)
12     bcp_paredes = DirichletBC(V, Constant((0, 0)), mf, 3)
13     bcu = [bcu_inflow, bcp_paredes]
14     bcp = [bcp_outflow]
15
16     u = TrialFunction(V)
17     v = TestFunction(V)
18     p = TrialFunction(Q)
19     q = TestFunction(Q)
20
21     u0 = Function(V)
22     u_ = Function(V)
23     p_n = Function(Q)
24     p_ = Function(Q)
25
26     U = 0.5 * (u0 + u)
27     n = FacetNormal(mesh)
28     f = Constant((0, dados_problema['grav']))
29     k = Constant(cfl['dt'])
30
31     viscosidadeCinematica =
32         Constant(dados_fluidos['viscosidade'])
```

```
33      massa_especifica =
34          Constant(dados_fluidos['massa_especifica'])
35      beta = 1
36
37      def epsilon(u):
38          return (1 / 2) * (nabla_grad(u) + nabla_grad(u).T)
39
40      def sigma(u, p, viscosidadeCinematica):
41          return 2 * viscosidadeCinematica \
42          * epsilon(u) - p * Identity(len(u))
43
44      F1 = (1.0 / k) * inner(u - u0, v) * df.dx \
45          + inner(grad(u0) * u0, v) * df.dx \
46          + inner(sigma(U, p_n, viscosidadeCinematica),
47          epsilon(v)) * df.dx \
48          + inner(p_n * n,
49          v) * df.ds \
50          - beta * viscosidadeCinematica * inner(grad(U).T * n,
51          v) * df.ds \
52          - viscosidadeCinematica * inner(f, v) * df.dx
53      a1 = lhs(F1)
54      L1 = rhs(F1)
55
56      a2 = inner(grad(p), grad(q)) * df.dx
57      L2 = inner(grad(p_n), grad(q)) * df.dx \
58          - (1.0 / k) * div(u_) * q * df.dx
59
60      a3 = inner(u, v) * df.dx
61      L3 = inner(u_, v) * df.dx \
62          - k * inner(grad(p_ - p_n), v) * df.dx
63
64      A1 = assemble(a1)
65      A2 = assemble(a2)
66      A3 = assemble(a3)
67
68      [bc.apply(A1) for bc in bcu]
69      [bc.apply(A2) for bc in bcp]
70
71      modelo = {'A1': A1,
```

```
72          'A2': A2,
73          'A3': A3,
74          'bcu': bcu,
75          'bcp': bcp,
76          'cfl': cfl,
77          'L1': L1,
78          'L2': L2,
79          'L3': L3,
80          'u_': u_,
81          'p_': p_,
82          'u0': u0,
83          'p_n': p_n}
84
85      return modelo
```

A função `solucao` soluciona as equações para cada passo de tempo dentro de uma laço de repetição. As linhas 4-6 aplicam as condições iniciais de contorno às matrizes. Nas linhas 8-19, resolve-se o método IPCS.

```
1
2  def solucao(modelo):
3      passos = 0
4      t = 0
5      for n in tqdm(range(modelo['cfl']['num_steps'])):
6          [bc.apply(modelo['A1']) for bc in modelo['bcu']]
7          [bc.apply(modelo['A2']) for bc in modelo['bcp']]
8
9          b1 = assemble(modelo['L1'])
10         [bc.apply(modelo['A1'], b1) for bc in modelo['bcu']]
11         solve(modelo['A1'], modelo['u_'].vector(), b1, 'bicgstab',
12         'hypre_amg')
13
14         b2 = assemble(modelo['L2'])
15         [bc.apply(b2) for bc in modelo['bcp']]
16         solve(modelo['A2'], modelo['p_'].vector(), b2, 'bicgstab',
17         'hypre_amg')
18
19         b3 = assemble(modelo['L3'])
20         solve(modelo['A3'], modelo['u_'].vector(), b3, 'cg', 'sor')
```

```
21
22          if n \% 200 == 0:
23              pvd_file = File('velocidade-{0}.pvd'.format(passos))
24              pvd_file << modelo['u_']
25              pvd_file = File('pressao-{0}.pvd'.format(passos))
26              pvd_file << modelo['p_']
27              passos = passos + 1
28
29          modelo['u0'].assign(modelo['u_'])
30          modelo['p_n'].assign(modelo['p_'])
31          t = t + modelo['cfl']['dt']
32      return modelo
```

3.4 Quantificação da Eficiência de Varrido

A fim de realizar a quantificação do efeito das erosões na parede da formação sobre a frente de propagação do escoamento, desenvolvemos um método comparativo de eficiência. Comumente, a determinação da eficiência de colchões lavadores depende do cálculo de áreas "molhadas" do espaço anular que são ocupadas pelos colchões em relação à área total disponível. Há diversas fórmulas na literatura para conceitos próximos, tais como, *eficiência de limpeza*, *eficiência de deslocamento* [**?**] e *eficiência de remoção* [**?**]. Aqui, adotaremos o termo *eficiência de varrido* para um coeficiente adimensional que quantifica o desvio relativo da vazão aproximada do escoamento de um colchão lavador por um espaço anular em uma configuração de paredes livres de erosão para com a configuração de paredes erodidas. A seguir, descrevemos a construção teórica do parâmetro de eficiência que empregaremos.

Consideremos o domínio do espaço anular descrito no plano cartesiano x, y com y orientado positivamente para cima (Fig. 12). A parede da formação e do tubo de revestimento na configuração lisa (modelo A) possui extensão vertical invariável e está limitada na porção esquerda (direita) pelo intervalo $[x_0^E, x_f^E]$ ($[x_0^D, x_f^D]$). Na configuração erodida, assume-se que a parede da formação varia lateralmente ao longo da extensão vertical por pequenas perturbações de acordo com uma função $\xi^E(y)$ ($\xi^D(y)$) que descreve um perfil de contorno médio em relação a $x = x_0^E$ ($x = x_f^D$) fixado.

No segmento $[x_0^E, x_f^E]$ ($[x_0^D, x_f^D]$), escolhemos x_l^E (x_l^D) como um ponto que particiona a porção anular esquerda em dois subsegmentos tais que $[x_0^E, x_f^E] = [x_0^E, x_l^E] \cup (x_l^E, x_f^E]$ ($[x_0^D, x_f^D] = [x_0^D, x_l^D] \cup (x_l^D, x_f^D]$), onde l indica que x_l^E (x_l^D) está afastado de x_f^E (x_f^D) aproximadamente por um comprimento de $l\%$ da largura da porção anular esquerda (direita).

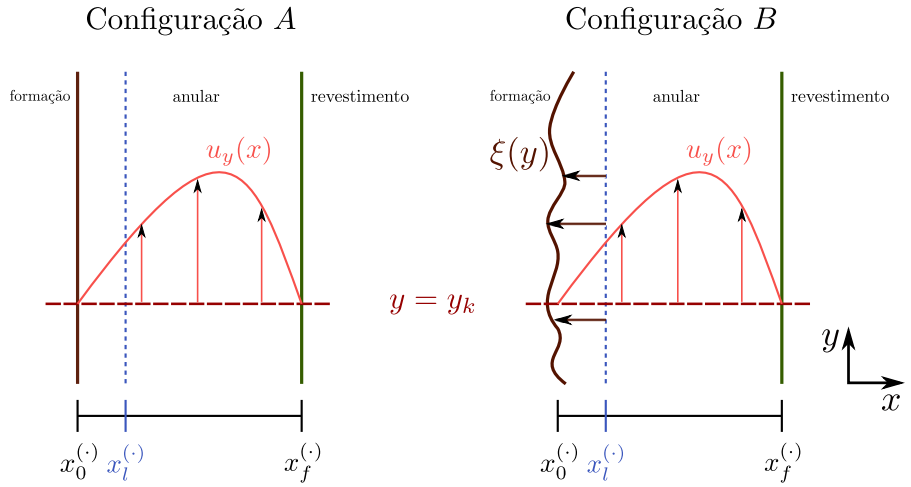

Figura 12: Esquema representativo para cálculo de vazões nas configurações de poço não erodido (A) e poço erodido (B). Fonte: Autor.

Definimos

$$Q_p^{(q)}(y_k) = \int_{x_0^q}^{x_f^q} u_y(x)\,dx\Big|_{y=y_k}, \quad q = D, E \text{ e} \tag{3.11}$$

$$Q_{p,l}^{(q)}(y_k) = \int_{x_l^q}^{x_f^q} u_y(x)\,dx\Big|_{y=y_k}, \quad q = D, E, \tag{3.12}$$

respectivamente, como a *vazão planar total na porção anular* q e a *vazão planar percentual na porção anular* q na configuração p à profundidade $y = y_k$. Isto é, tais quantidades representam medidas da frente de propagação do escoamento ao longo da direção y, levando em conta a variação da componente u_y da velocidade sobre o corte transversal (direção x). Quando $p = A$, as quantidades referem-se ao caso de poço não erodido; quando $p = B$, ao caso de poço erodido.

A partir das Eqs. (3.11) e (3.12), definimos

$$\eta_l^{(q)}(y_k) = \frac{Q_{B,l}^{(q)}(y_k)}{Q_{A,l}^{(q)}(y_k)} \times 100\%, \quad q = D, E \tag{3.13}$$

como o *coeficiente de eficiência de varrido efetivo a* $l\%$ *do revestimento* na porção anular q, de maneira que

$$\eta^{(q)}(y_k) = \frac{Q_B^{(q)}(y_k)}{Q_A^{(q)}(y_k)} \times 100\%, \quad q = D, E \tag{3.14}$$

é o *coeficiente de eficiência de varrido total* na porção anular q para a profundidade y_k. Nesses termos, à medida que $\eta_l^{(q)}(y_k)$ ($\eta^{(q)}(y_k)$) tende a 100%, o efeito das erosões sobre a propagação do escoamento é praticamente desprezível; opostamente, à medida que tende a 0%, esse efeito é considerável.

Ambos os coeficientes usam a configuração não erodida como modelo de referência para a configuração erodida, permitindo, dessa maneira, a quantificação da influência das erosões na perda de vazão planar. Entretanto, o coeficiente de eficiência de varrido efetivo captura essa influência de modo indireto através da alteração detectada no escoamento apenas na porção livre do anular. Já o coeficiente de eficiência de varrido total mede a influência diretamente.

4 RESULTADOS E DISCUSSÃO

Nesta seção, apresentamos os resultados das simulações numéricas obtidas para as configurações de poço dos tipos A e B processados via biblioteca FEniCs e pós-processados no software Paraview.

4.1 Posicionamento do Problema-Base

Os experimentos levam em consideração um único tipo de colchão lavador à base de óleo vegetal com viscosidade cinemática de 0,18 Pa.s e massa específica de 900 kg.m^{-3} [**?**]. Para definir a velocidade de entrada, assumimos um diâmetro de 0,4 m para o bocal de saída de fluido no fundo de poço como comprimento de referência e $Re = 500$, que mantém o escoamento dentro do regime laminar [**?**]. A profundidade dos poços é assumida pequena, de modo que a aceleração da gravidade permanece fixada em 9.8 m.s^{-2}. A extensão do domínio na direção normal ao escoamento (largura) equivale a de um poço com 8,66 pol. (aproximadamente 0,22 m) de diâmetro. Na direção tangente ao escoamento, o domínio limita-se a 1 m de comprimento.

Consideramos Ω_d o domínio, Γ_p o contorno da parede da formação, Γ_r o contorno do revestimento, Γ_e o contorno do orifício por onde o fluido é expelido e Γ_s o contorno do anular por onde o fluido escapa. Reescrevendo as equações de Navier-Stokes de maneira compacta, podemos definir o problema de valor de contorno e inicial da seguinte forma: determinar os campos $\boldsymbol{u}$ e p no espaço anular, tais que:

$$\mathcal{L}[\boldsymbol{u}, p; \rho, \mu, \boldsymbol{g}] = \mathbf{0} \tag{4.1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{4.2}$$

sujeito às condições de contorno e inicial

$$\boldsymbol{u}|_{\Gamma_p} = \mathbf{0} \tag{4.3}$$

$$p|_{\Gamma_s} = 0 \tag{4.4}$$

$$\boldsymbol{u}|_{\Gamma_e} = \boldsymbol{u_0}. \tag{4.5}$$

4.2 Configurações do Resolvedor Numérico

A fim de resolver os sistemas lineares, utilizamos o *método do gradiente biconjugado estabilizado* (`bicgstab`) para o cálculo da velocidade tentativa e da pressão. De forma geral, o `bicgstab` utiliza menos memória que outros métodos. Para a correção de velocidade, foi utilizado o *método do gradiente conjugado* `cg`. Como pré-condicionador, utilizamos o `hypre_amg`.

4.3 Simulações Numéricas

Consideramos 4 configurações de poço para as simulações: A_1 (anular concêntrico sem erosões); A_2 (anular excêntrico sem erosões); B_1 (anular concêntrico com erosões); A_2 (anular excêntrico com erosões). Para calcular o coeficiente de eficiência de varrido do colchão lavador, escolhemos 4 cotas de profundidade, a saber $y_k = \{0,1; 0,3; 0,7; 1.0\}$, assim retornando os respectivos valores $\eta(y_k)$, para $k = 1, 2, 3, 4$ para as porções esquerda e direita do anular. O *loop* temporal considerou um passo de tempo $\Delta t = 0.00025$ com 40 mil iterações. Cada simulação durou, em média, 3 horas, em um computador com processador Intel Core i5 8a. geração e 8 GB RAM.

4.3.1 Configurações sem Erosões

Considerando a geometria sem nenhum tipo de erosão na parede do poço e com valor de *standoff* 100% A_1, é possível observar na Fig. 13 o efeito de escoamento parabólico na saída do espaço anular.

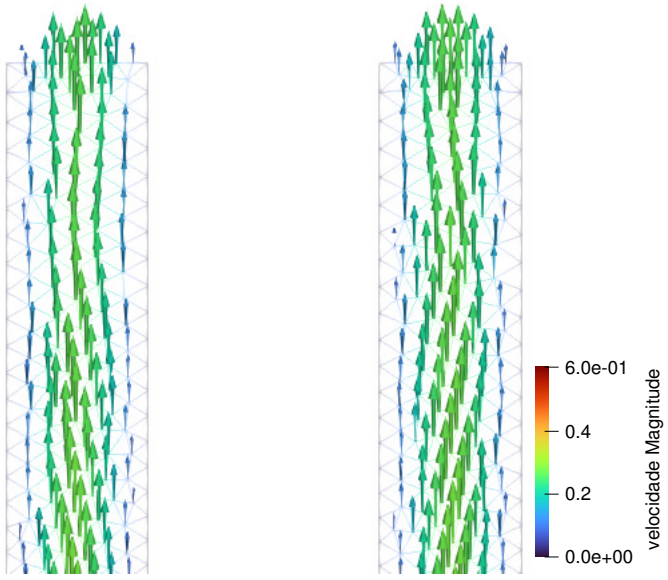

Figura 13: Campo de velocidade na saída do espaço anular no tempo de 10s na geometria A_1. Fonte: autor

A Fig. 14 é o perfil de velocidade no tempo de 0,5 segundo. Podemos observar a simetria do escoamento por se tratar de um caso com excentricidade nula.

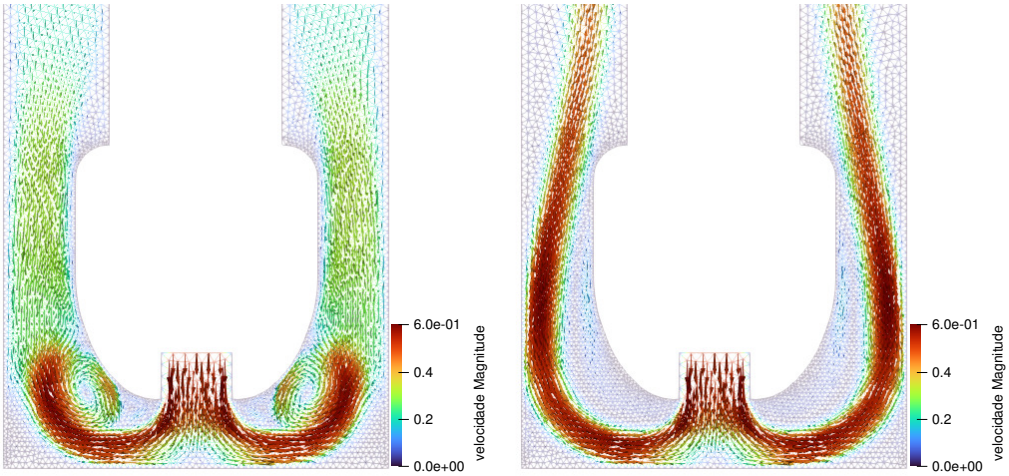

Figura 14: Campo de velocidade na sapata no tempo de 0.5s e 10s na geometria A_1. Fonte: autor

A Fig. 15 mostra o perfil de velocidade na geometria A_1 para os 4 níveis distintos.

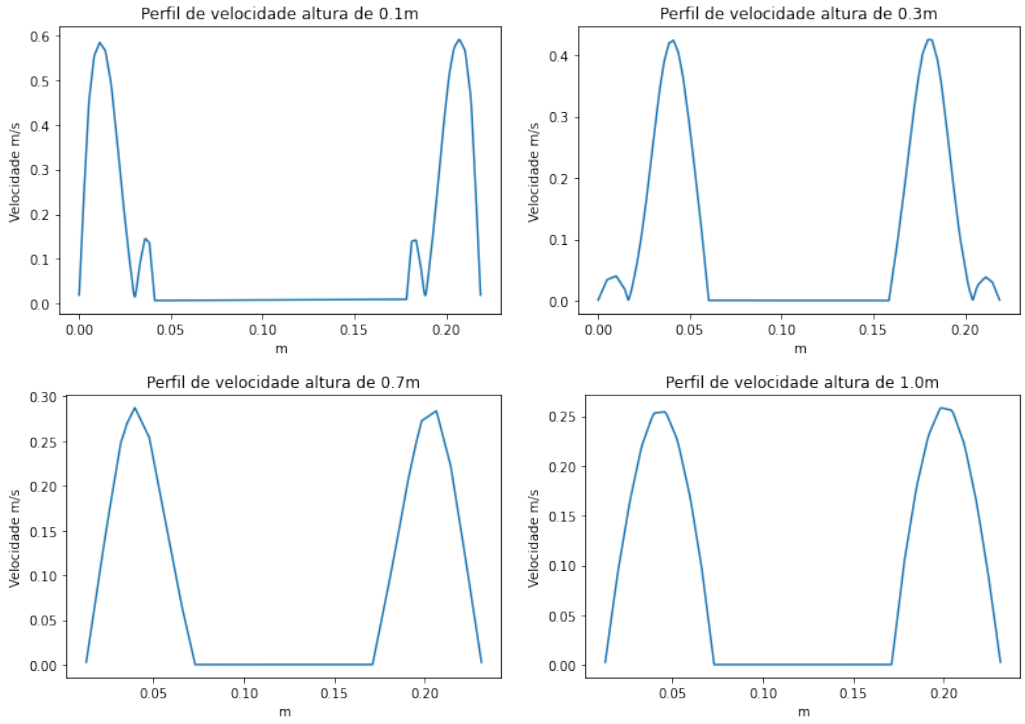

Figura 15: Perfis de velocidade da geometria A_1 em 4 níveis distintos

Considerando agora a geometria A_2, de parede lisa e valor de *standoff* de 50%, o campo de velocidade na saída é mostrado na Fig. 16.

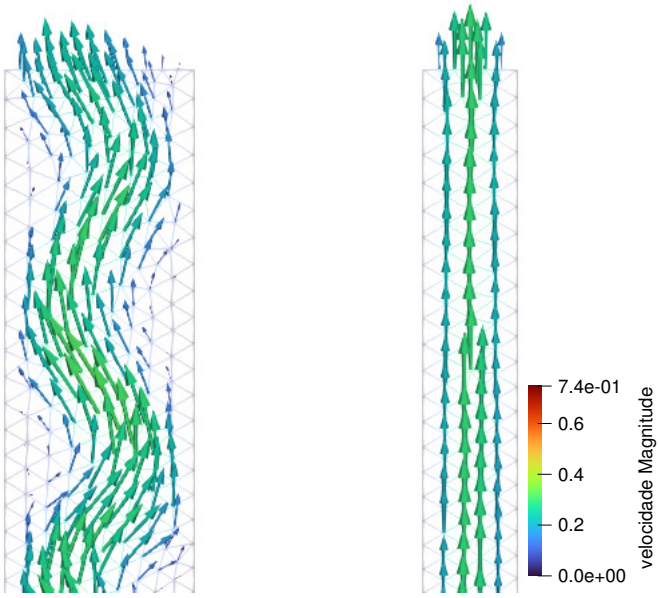

Figura 16: Campo de velocidade na saída do espaço anular no tempo de 10s na geometria A_2. Fonte: autor

É evidente o efeito do standoff no escoamento, o fluxo teve um comportamento ondulatório. Na região da sapata, perto do fundo do poço Fig. 17 é possível observar que do lado direito a velocidade teve um perfil mais regular que do lado esquerdo.

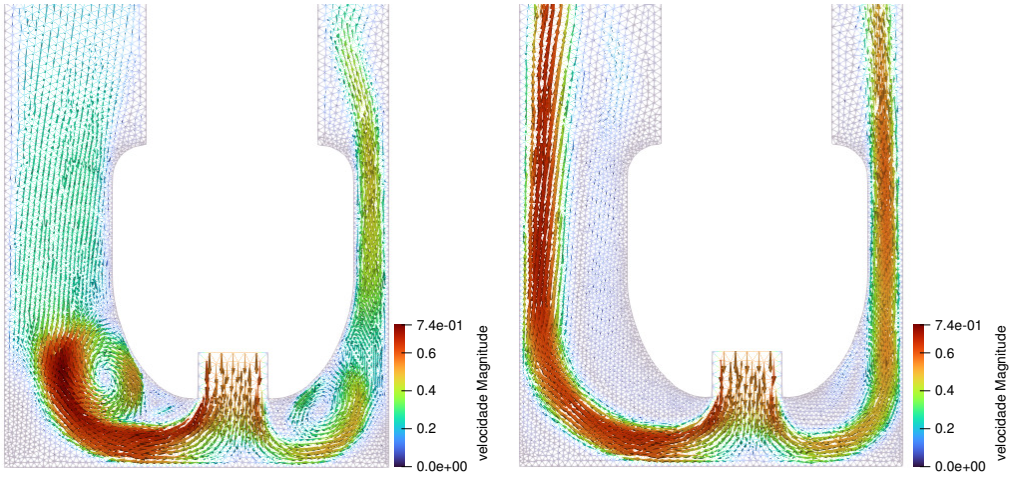

Figura 17: Campo de velocidade na sapata no tempo de 0.5s e 10s na geometria A_2. Fonte: autor

Os perfis de velocidade para os 4 níveis selecionados na geometria onde não há nenhuma rugosidade na parede do poço e um valor de *standoff* menor que 100% podem ser visualizados na figura

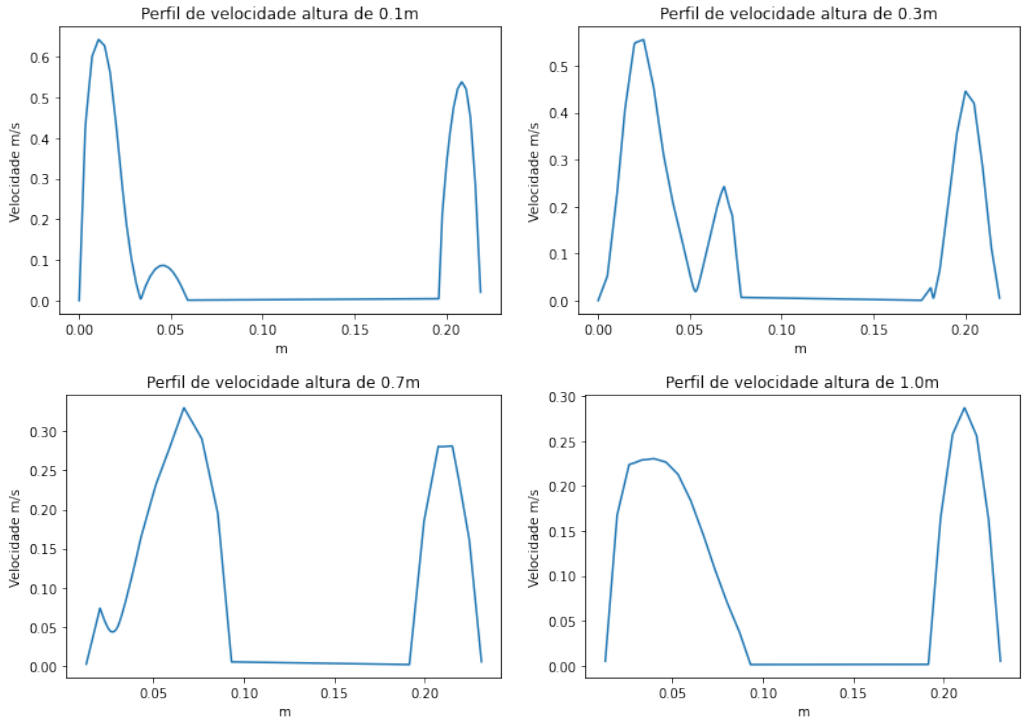

Figura 18: Perfis de velocidade da geometria A_2 em 4 níveis distintos

Recortes do campo de pressão das geometrias A podem ser vistos na Fig. 19

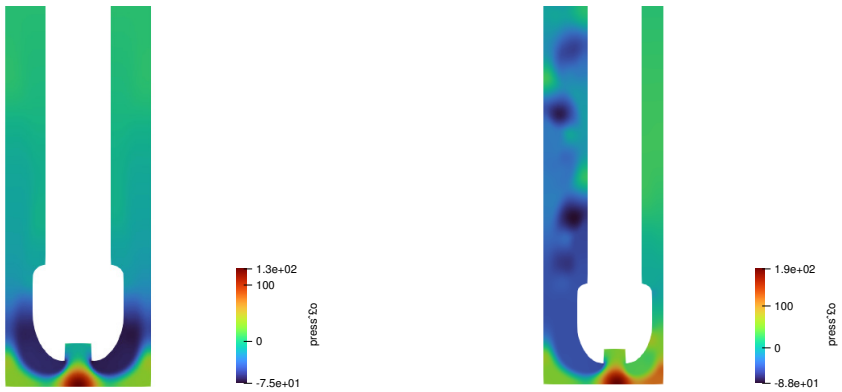

Figura 19: Campo de pressão nas geometrias A. Fonte: autor

4.3.2 Configurações com Erosões

Para o caso das geometrias B, em que há um certo grau de erosão na parede da formação, o perfil de velocidade na saída do anular observado na Fig. 23 mostra um escoamento parabólico. O mesmo comportamento foi visto nas geometrias A. Logo, a erosão não causou interferências.

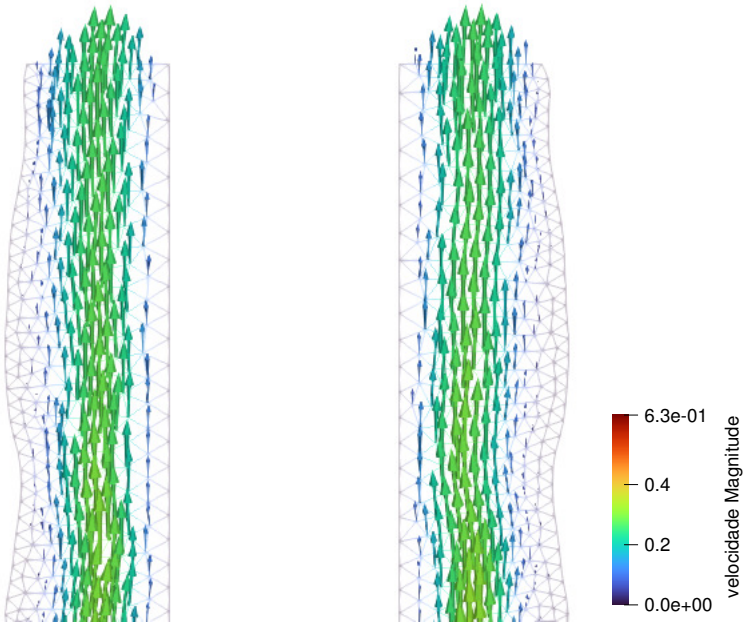

Figura 20: Campo de velocidade na saída do espaço anular no tempo de 10s na geometria B_1. Fonte: autor

A Fig. 21 mostra o comportamento do fluido na sapata no tempo de 0.5s e 10s evidenciando o efeito da erosão se comparado a Fig. 14 pode-se observar que o comportamento do fluido foi simétrico como na geometria A_1 e concluir que o efeito da erosão nesse caso não teve impacto expressivo

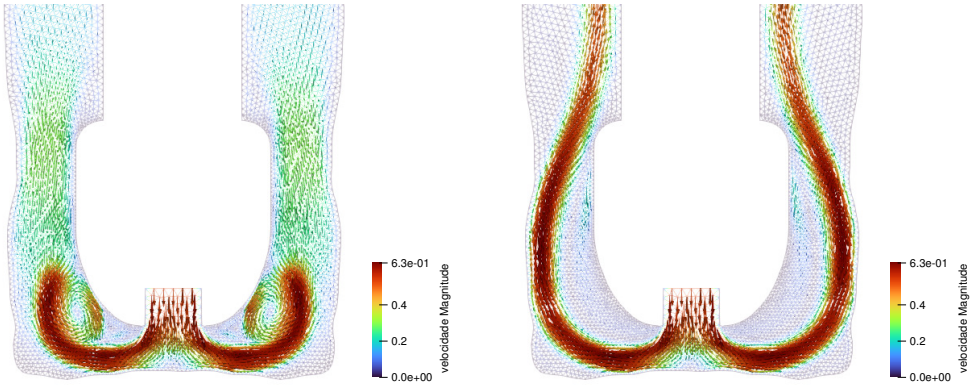

Figura 21: Campo de velocidade na sapata no tempo de 0.5s e 10s na geometria B_1. Fonte: autor

Os perfis de velocidade para os 4 níveis selecionados na geometria onde há rugosidade na parede do poço e um valor de *standoff* igual a 100% (B_1) podem ser visualizados na figura 22.

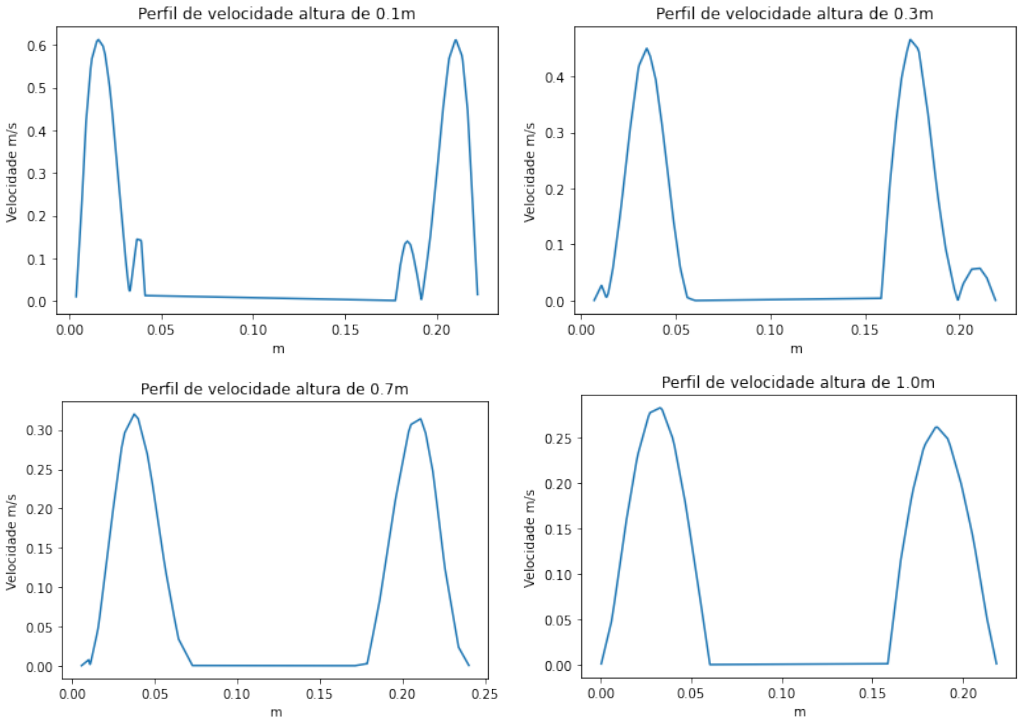

Figura 22: Perfis de velocidade da geometria B_1 em 4 níveis distintos

Para o caso da geometria B_2, onde a taxa de *standoff* é de 50%, o perfil de velocidade na saída do anular observado na Fig. 23 mostra um escoamento parabólico, esse mesmo comportamento foi visto nas geometrias A, portanto, a erosão não causou interferência nesse sentido, porém a erosão causou um efeito de irregularidade no escoamento com pequenas zonas de recirculação.

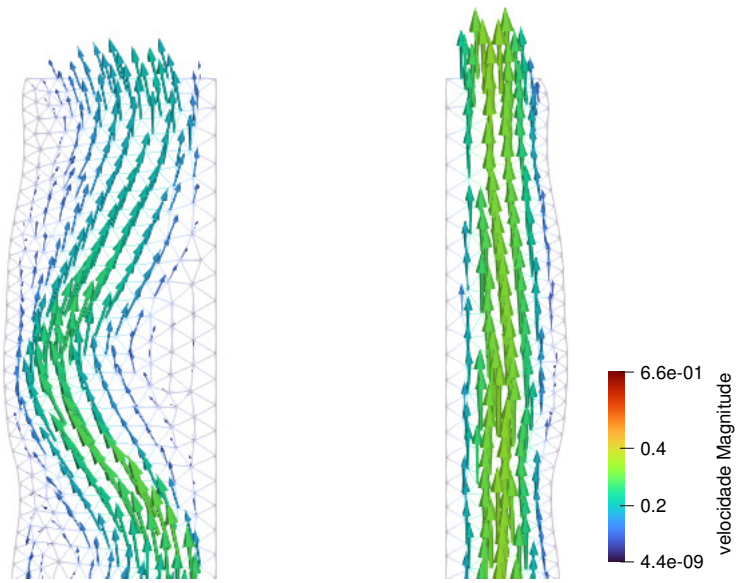

Figura 23: Campo de velocidade na saída do espaço anular no tempo de 10s na geometria B_2. Fonte: autor

A Fig. 24 mostra o comportamento do fluido na sapata nos tempos de 0.5s e 10s evidenciando o efeito da erosão se comparado a Fig. 17 pode-se observar que o comportamento do fluido não foi simétrico como na geometria A_2

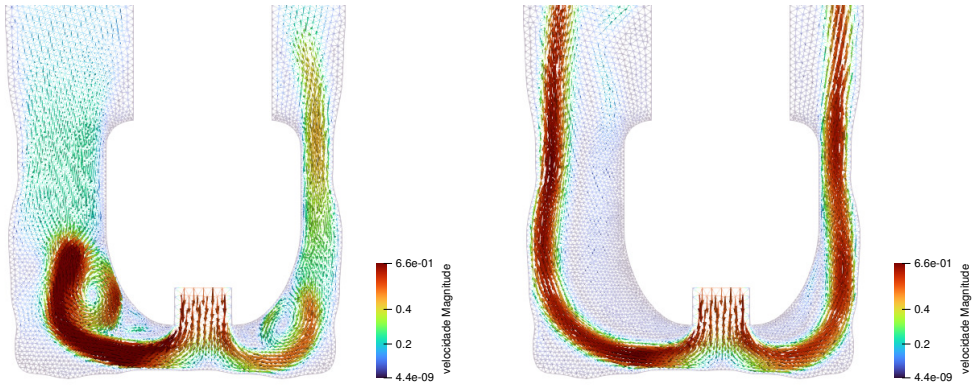

Figura 24: Campo de velocidade na sapata no tempo de 0.5s e 10s na geometria B_2. Fonte: autor

Os perfis de velocidade para os 4 níveis selecionados na geometria onde há rugosidade na parede do poço e um valor de *standoff* menor que 100% B_2 podem ser visualizado na figura 25.

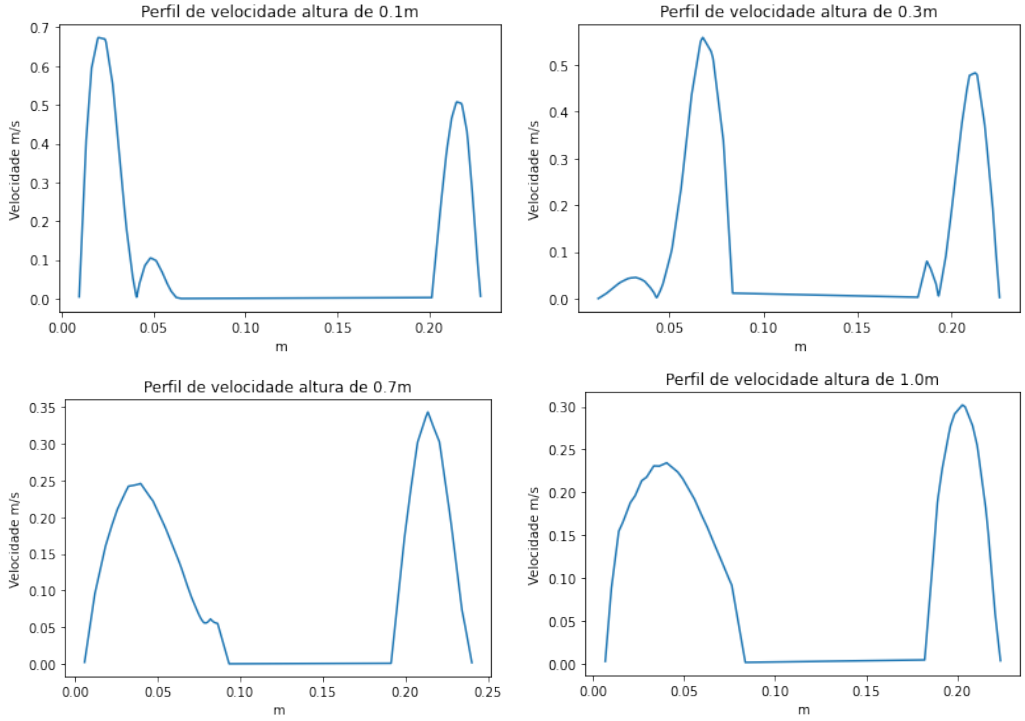

Figura 25: Perfis de velocidade da geometria B_2 para 4 níveis distintos

Recortes do campo de pressão das geometrias B podem ser vistos na Fig. 26.

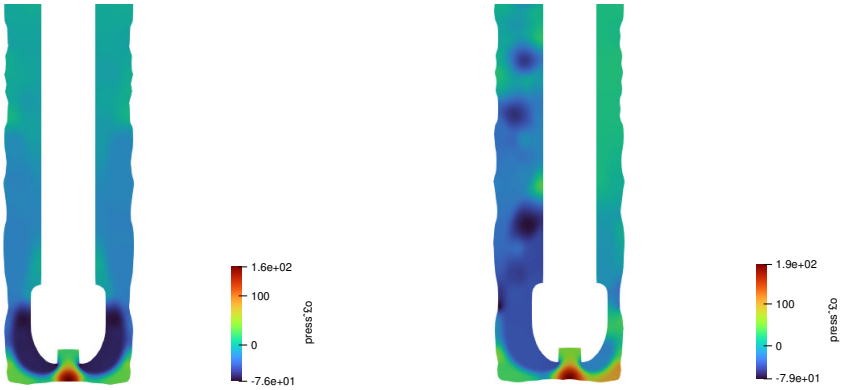

Figura 26: Campo de pressão nas geometrias B. Fonte: autor

4.3.3 Análise de eficiência de varrido

A fim de comparar a eficiência de varrido, as tabelas 2, 3, 4 e 5 abaixo, descrevem as vazões Q em cada geometria e o parâmetro η dos respectivos lados do espaço anular. É descrito também os valores da vazão Q e do η no espaço delimitado em 90% e no espaço total.

O parâmetro de eficiência de varrido η do lado esquerdo do espaço anular levando em consideração as geometrias A_1 e B_1 ficou em torno de 0.9 a exceção do valor delimitado em 90% perto do fundo do poço em $0.1m$ onde foi de 0.53, tabela 2, isto é, há uma eficiência, no caso em que há erosão, de 90% do caso ideal (liso) em níveis acima de $0.3m$, em $0.1m$ a erosão causou uma ineficiência de aproximadamente 50% em relação ao caso ideal (liso) se levarmos em consideração o espaço delimitado em 90%. Esta análise é análoga ao lado direito, tabela 3 das mesmas geometrias por se tratar de um caso de *standoff* de 100%.

Tabela 2: Valor do parâmetro η para cada nível das geometrias A_1 e $B1$ do lado esquerdo

k	y_k	Q_A^E	$Q_{A,90}^E$	Q_B^E	$Q_{B,90}^E$	η_{90}^E	η^E
1	0.1	45.18	41.42	41.64	22.06	53.27%	92.17%
2	0.3	45.28	45.89	41.42	42.78	93.23%	91.46%
3	0.7	44.92	43.68	41.22	40.00	91.57%	91.77%
4	1.0	44.93	43.74	41.22	40.00	91.44%	91.74%

Tabela 3: Valor do parâmetro η para cada nível das geometrias A_1 e $B1$ do lado direito

k	y_k	Q_A^D	$Q_{A,90}^D$	Q_B^D	$Q_{B,90}^D$	η_{90}^D	η^D
1	0.1	46.14	42.24	41.63	17.17	40.64%	90.21%
2	0.3	46.10	46.69	41.54	43.22	92.57%	90.12%
3	0.7	45.63	44.42	41.08	39.68	89.34%	90.03%
4	1.0	45.72	44.52	41.08	39.68	89.12%	89.85%

O parâmetro de eficiência de varrido η do lado esquerdo do espaço anular levando em consideração as geometrias A_2 e B_2 ficou em torno de 0.6 a 0.8 a exceção do valor perto ao fundo do poço em $0.1m$ onde foi de 0.37, tabela 2, isto é, há uma eficiência no caso da erosão de 90% do caso ideal (liso) em níveis acima de $0.3m$, em $0.1m$ a erosão causou uma ineficiência de aproximadamente 37% em relação ao caso ideal (liso), considerando apenas o espaço delimitado em 90%.

Como o valor de *standoff* não é 100% aqui, então há uma grande diferença dos valores de η no espaço anular de menor distância, visto que o escoamento possui menos espaço para percorrer, ele preenche todo o espaço direito, e a eficiência aumenta, o inverso acontece no lado esquerdo, onde os valores foram menos que 0.9 no caso simétrico. No lado direito a eficiência de varrido teve uma melhoria quando a parede foi erodida com um valor de 115% do caso ideal (liso).

Tabela 4: Valor do parâmetro η para cada nível das geometrias A_2 e $B2$ do lado esquerdo

k	y_k	Q_A^E	$Q_{A,90}^E$	Q_B^E	$Q_{B,90}^E$	η_{90}^E	η^E
1	0.1	55.58	44.20	42.11	16.72	37.82%	75.77%
2	0.3	56.63	54.88	43.35	42.99	78.34%	76.55%
3	0.7	54.83	56.20	42.17	35.74	63.58%	76.91%
4	1.0	55.74	51.93	42.08	41.05	79.05%	75.49%

Tabela 5: Valor do parâmetro η para cada nível das geometrias A_2 e $B2$ do lado direito

k	y_k	Q_A^D	$Q_{A,90}^D$	Q_B^D	$Q_{B,90}^D$	η_{90}^D	η^D
1	0.1	34.74	33.24	40.95	25.33	76.20%	117.86%
2	0.3	35.37	34.56	40.78	29.86	86.40%	115.27%
3	0.7	33.95	32.92	40.58	35.48	107.78%	119.53%
4	1.0	34.32	33.27	39.06	37.59	113.00%	113.82%

Os gráficos 27 e 28 ilustra esse comportamento analisado nas 4 tabelas.

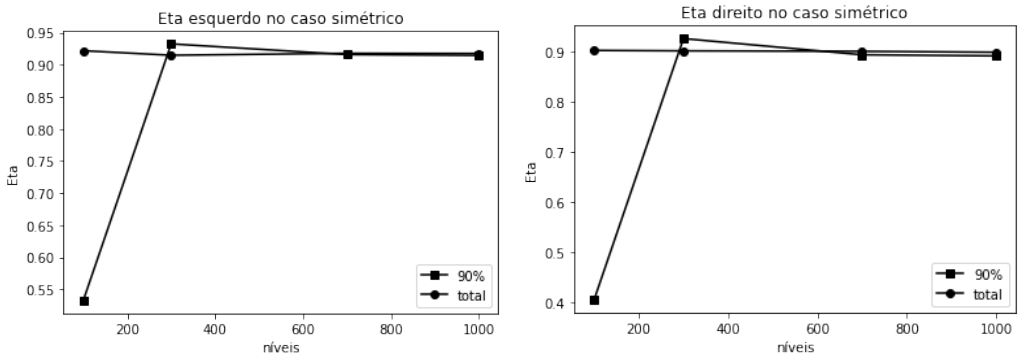

Figura 27: Valores de η para as geometrias com o valor de *standoff* 100% (A_1 e B_1). Fonte: autor

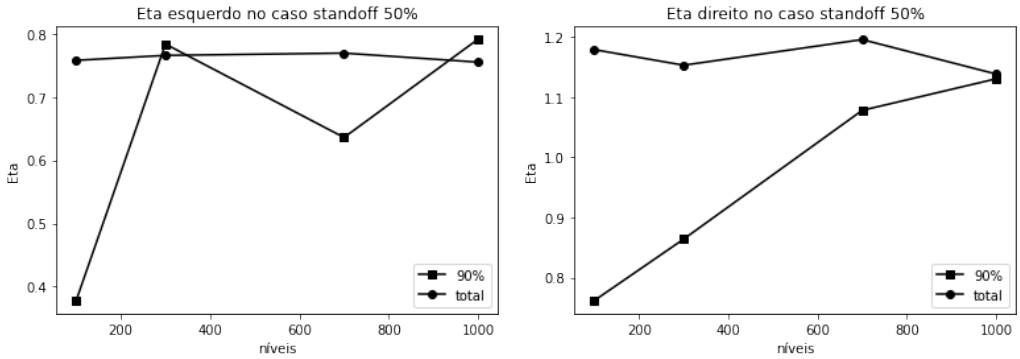

Figura 28: Valores de η para as geometrias com o valor de *standoff* 50% (A_2 e B_2). Fonte: autor

5 CONCLUSÃO

Este trabalho apresentou um estudo sobre o comportamento de colchões lavadores a base de óleo vegetal. O escoamento desse tipo de fluido ocorre durante a fase de pré-cimentação de poços de petróleo. Simulamos a dinâmica em diferentes tipos de geometrias de admitindo erosões e excentricidade do revestimento (*standoff*). A fim de estudar a eficiência de varrido, um parâmetro tomando como referência a configuração de poço ideal, sem rugosidade nas paredes da formação.

Algumas limitações que tornam as simulações um pouco aquém da realidade tiveram de ser tratadas, tais como o nível do refinamento de malha, cuja captura dos perfis de velocidade não tiveram suavidade suficiente. Em segundo lugar, vale dizer que fluidos newtonianos não representam com exatidão o escoamento de colchões lavadores, visto que, em geral, modelos não-newtonianos, tais como o de Herschel-Bulkley são comumente utilizados.

A extensão de estudo do poço foi limitada a $1m$, além de considerarmos a hipótese de poço raso, onde variações da força gravitacional praticamente são desprezíveis. Outro fator a ser levado em consideração é que em configurações com *standoff*, geometrias bidimensionais são limitadas ao plano de simetria da configuração. Em ambas as configurações, tanto concêntricas como excêntricas, as "vazões" calculadas também perdem o seu sentido original, passando a medir uma espécie de "fluxo planar".

Apesar disso, este trabalho destaca o potencial do método dos elementos finitos para tratar de problemas relacionados à cimentação primária de poços de petróleo, bem como desafios relacionados associados à engenharia de poços.

Como trabalhos futuros, listamos algumas sugestões:

- implementar modelo não-newtoniano que represente melhor o tipo de escoamento de colchões lavadores a base de óleo vegetal;
- criar geometrias tridimensionais para melhor representar a formação rochosa, a topologia das erosões, a frente de propagação do fluido, interfaces e vazões reais;
- melhorar a definição do coeficiente que mede a eficiência de varrido para levar em consideração outras características essenciais do escoamento;
- realizar simulações de fluidos com outras propriedades.

REFERÊNCIAS

[1] KOEHLER, Leonardo Pereira. **Projeto de revestimento de popos e suas especificações.** 2018.

[2] ELIAS, R. N. **Estrutura de dados por arestas para a simulação paralela de escoamentos incompressíveis pelo método estabilizado de elementos finitos.** UFRJ, 2007.

[3] NELSON, Erik B.; GUILLOT, Dominique. **Well Cementing**. 2ª Edition.

[4] COTTA, Pery. **O petróleo é nosso?**. Guavira, Rio de Janeiro, 3 de outubro de 1953.

[5] WHITE, FRANK M. . **Fluid Mechanics**. 7ª edição. University of Rhode Island: McGraw-Hill, 2008.

[6] LOGG, Anders. **Automated solution of differential equations by the finite element method:** The FEniCS book. 2011.

[7] LANGTANGEN, Hans Petter; LOGG, Anders. **Solving PDEs in Python**: The FEniCS Tutorial Volume I. Center for Biomedical Computing, Simula Research Laboratory and Department of Informatics, University of Oslo. Springer. 2017.

[8] COMMISSION, National. **Macondo: The Gulf Oil Disaster, Chief Counsel's Report**. National Commission. 2011.

[9] ROCHA, Luiz; AZEVEDO, Cecilia. **Projetos De Poços De Petróleo**: Geopressões e assentamento de colunas de revestimento. 3ª edição. Interciência, 2019.

[10] CURBELO F. D. S.; ARANHA R. M.; MOCHIZUKI V. L.; GUARNICA A. I. C.; FREITAS J. C. O.; SILVA R. K. P. **Lavadores compostos por óleo vegetal, tensoativo e salmoura.** XXI Congresso Brasileiro de Engenharia Química, Fortaleza: Cobeq, 2016.

[11] TAYLOR, C.; Hood, P. **A numerical solution of the Navier-Stokes equations using the finite element technique**, Comp. and Fluids 1 (1973), 73-100.

[12] CORNTHWAITE, JOHN P. **Pressure poison method for the incompressible Navier-Stokes equations using galerkin finite elements.** Faculty of Georgia Southern University in Partial Ful llment. 2003.

[13] CAMPOS, G. **PROCELAB – Procedimentos e métodos de laboratório destinados à cimentação de poços de Petróleo.** 2001

[14] BOURGOYNE, Adam T. **Applied drilling engineering.** 1991

[15] SANTOS, Jaíne Lima Dos et al. **Revestimento e cimentação de poços de petróleo.** Anais II CONEPETRO. Campina Grande: Realize, agosto, 2016.

[16] ARANHA, Rayanne Macêdo et al. **OBTENÇÃO DE COLCHÃO LAVADOR A BASE DE TENSOATIVO E ÓLEO VEGETAL PARA REMOÇÃO DE FLUIDO DE PERFURAÇÃO NÃO AQUOSO.** 2015. Disponível em: <https://www.editorarealize.com.br/artigo/visualizar/10365>. Acesso em: 08/11/2021 10:41

[17] TIKHONOV, Vadim S. Tikhonov; BUKASHKINA, Olga S.; GANDIKOTA, Raju. **NUMERICAL SIMULATION OF CASING CENTRALIZATION.**. p. 667-675, July, 2014.

[18] GODA, K. **A multistep technique with implicit difference schemes for calculating two-dimensional or three-dimensional cavity flows**. Journal of Computational Physics, v. 30, n. 1, p. 76 – 95, 1979.

[19] FOROUSHAN, Hanieh K.;LUND, Bjornar; YTREHUS, Jan David;SAASEN, Arild. **Cement Placement: An Overview of Fluid Displacement Techniques and Modelling.** Energies 2021, 14,v. 573. 01/2021.

[20] NACCACHE, Mônica. Flow displacement in eroded regions inside annular ducts. Springer. Journal of the Brazilian Society of Mechanical Sciences and Engineering, v.420, p. 1-14, 03/2018.

[21] ESCUDIER, M.P.; OLIVEIRA, P.J.; PINHO, F.T. **Fully developed laminar flow of purely viscous non-Newtonian liquids through annuli, including the effects of eccentricity and inner-cylinder rotation**.Elsevier. International Journal of Heat and Fluid Flow, v. 23, p. 52–73, 04/2002.

[22] LIMA, Jaíne. **Revestimento e cimentação de poços de petróleo**: 2006. Dissertação – Física, Universidade Federal de Sergipe, Sergipe, 2006.

[23] CHORIN, A. **Numerical solution of the navier–stokes equations.** 1968

[24] MOLON, Fernando. **Simulações numéricas de problemas descritos pelas equações de Navier-Stokes incompressíveis via biblioteca FEniCS.** – Engenharia Mecânica, Universidade Federal do Espírito Santo, Espírito Sant, 2017.

[25] LUPYANA, Samwel Daud. **The Influence of Velocity Profile on Cement Displacement Efficiency**. Norwegian University of Science and Technology, Department of Petroleum Engineering and Applied Geophysics, 2015.

[26] PORDEUS, Roberto **Fenômenos de transporte mecânica dos fluidos:** Considerações e Propriedades dos Fluidos. Universidade Federal Rural do Semi-Árido - UFERSA. 2017.

[27] JANÚNCIO, Danilo. História: 'Petróleo é nosso' leva à criação do monopólio. Folha, 03/10/2003. Disponível em: https://www1.folha.uol.com.br/folha/especial/2003/petrobras50anos/fj0310200303.shtml. Acesso em: 15/05/2022.

[28] Geuzaine, C. J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009. Disponível em: http://gmsh.info. Acesso em 20/05/2022

[29] ALTMAN, Max. **Hoje na História: 1859 - Perfurado o primeiro poço de petróleo nos EUA.** 2020. Disponível em: <https://operamundi.uol.com.br/hoje-na-historia/5976/hoje-na-historia-1859-perfurado-o-primeiro-poco-de-petroleo-nos-eua>. Acesso em: 8 11 2021.

[30] petrobras. **Batemos sucessivos recordes de produção no pré-sal.** Cidade: Organização, ano. Disponível em: <https://petrobras.com.br/fatos-e-dados/batemos-sucessivos-recordes-de-producao-no-pre-sal.htm>. Acesso em: 8 11 2021.

[31] WOLFRAM, Stephen. **Implications for Everyday Systems**. A New Kind of Science. Page 996.

[32] https://gmsh.info/

[33] https://fenicsproject.org/

[34] https://gmsh.info/doc/texinfo/gmsh.html

APÊNDICE 1

Neste apêndice, incluímos o código-fonte utilizado para o desenvolvimento deste trabalho para futuras consultas.

```
import meshio as mio
import matplotlib.pyplot as plt
import h5py
from fenics import *
import dolfin as df
from tqdm import tqdm
```

```
def menu_fluido():
    while True:
        print("informe o fluido\n\n"
              "1 - oleo\n")
        fluido_type = int(input())
        if fluido_type == 1:
            return fluido_type

def carregar_malha(caminho):
    mesh_from_file = mio.read(caminho)
    return mesh_from_file


def create_mesh(mesh, cell_type, prune_z=False):
    cells = mesh.get_cells_type(cell_type)
    cell_data = mesh.get_cell_data("gmsh:physical", cell_type)
    out_mesh = mio.Mesh(points=mesh.points,
    cells={cell_type: cells},
                        cell_data={"name_to_read": [cell_data]})
    if prune_z:
        out_mesh.prune_z_0()
    return out_mesh


def mvc_mf(mesh_from_file):
```

```
26      line_mesh = create_mesh(mesh_from_file, "line", prune_z=True)
27      mio.write("facet_mesh.xdmf", line_mesh)
28
29      triangle_mesh = create_mesh(mesh_from_file, "triangle",
30      prune_z=True)
31      mio.write("mesh.xdmf", triangle_mesh)
32
33      mesh = Mesh()
34
35      with XDMFFile("mesh.xdmf") as infile:
36          infile.read(mesh)
37      mvc = MeshValueCollection("size_t", mesh, 2)
38
39      with XDMFFile("facet_mesh.xdmf") as infile:
40          infile.read(mvc, "name_to_read")
41      mf = MeshFunction("size_t", mesh, mvc)
42      plot(mesh)
43      plt.show()
44      return mesh, mvc, mf
45
46
47  def dados_entrada(type_fluido):
48      dados_problema = {'diametro_saida':40,
49                        'Tempo': 10,
50                        'grav':-9.85,
51                        'num_steps': 40000}
52      Reynolds = 50
53      dados_problema['velocidade_fluido_inflowX'] = 0
54      dados_problema['velocidade_fluido_inflowY'] =
55      -1000*(Reynolds*0.0381)/(900*0.04)
56      dados_problema['dt'] =
57      int(dados_problema['Tempo'])/int(dados_problema['num_steps'])
58
59      oleo = {
60          'nome': "oleo",
61          'viscosidade': 38.1,
62          'densidade': 0.000900
63      }
64
```

```
65      dados_fluidos = [oleo]
66      Reynolds = (Reynolds*0.0381)/(900*40)
67      return dados_fluidos[type_fluido - 1],dados_problema,Reynolds
68
69
70  def delete_pasta_simulacao():
71      import shutil
72
73      try:
74          shutil.rmtree('TesteIPCS1')
75          shutil.rmtree('navier_stokes_cylinder')
76      except OSError as e:
77          print(f"Error:{ e.strerror}")
78
79
80  def modelo(mesh, dados_problema,dados_fluidos, mf):
81      V = VectorFunctionSpace(mesh, 'P', 2)
82      Q = FunctionSpace(mesh, 'P', 1)
83
84      inflow_profile = ('0' +
85      str(dados_problema['velocidade_fluido_inflowX']),
86      '0' + str(dados_problema['velocidade_fluido_inflowY']))
87
88      bcu_inflow = DirichletBC(V, Expression(inflow_profile,
89      degree=2), mf, 1)
90      bcp_outflow = DirichletBC(Q, Constant(0), mf, 2)
91      bcp_paredes = DirichletBC(V, Constant((0, 0)), mf, 3)
92      bcu = [bcu_inflow, bcp_paredes]
93      bcp = [bcp_outflow]
94
95      u = TrialFunction(V)
96      v = TestFunction(V)
97      p = TrialFunction(Q)
98      q = TestFunction(Q)
99
100     u0 = Function(V)
101     u_ = Function(V)
102     p_n = Function(Q)
103     p_ = Function(Q)
```

```
104
105     U = 0.5 * (u0 + u)
106     n = FacetNormal(mesh)
107     f = Constant((0, dados_problema['grav']*
108     dados_fluidos['densidade']))
109     k = Constant(dados_problema['dt'])
110
111     viscosidadeCinematica =
112     Constant(dados_fluidos['viscosidade'])
113     densidade = Constant(dados_fluidos['densidade'])
114     beta = 1
115
116     def epsilon(u):
117         return (1 / 2) * (nabla_grad(u) + nabla_grad(u).T)
118
119     def sigma(u, p, viscosidadeCinematica):
120         return 2 * viscosidadeCinematica * epsilon(u) - p *
121         Identity(len(u))
122
123     # tentativa de velocidade
124     F1 = (1.0 / k) * inner(u - u0, v) * df.dx \
125          + inner(grad(u0) * u0, v) * df.dx \
126          + inner(sigma(U, p_n, viscosidadeCinematica),
127          epsilon(v)) * df.dx \
128          + inner(p_n * n, v) * df.ds \
129          - beta * viscosidadeCinematica * inner(grad(U).T * n, v)
130          * df.ds \
131          - viscosidadeCinematica * inner(f, v) * df.dx
132     a1 = lhs(F1)
133     L1 = rhs(F1)
134
135     # Correcao da pressao
136     a2 = inner(grad(p), grad(q)) * df.dx
137     L2 = inner(grad(p_n), grad(q)) * df.dx \
138          - (1.0 / k) * div(u_) * q * df.dx
139
140     # Correcao da velocidade
141     a3 = inner(u, v) * df.dx
142     L3 = inner(u_, v) * df.dx \
```

```
143             - k * inner(grad(p_ - p_n), v) * df.dx
144
145     A1 = assemble(a1)
146     A2 = assemble(a2)
147     A3 = assemble(a3)
148
149     [bc.apply(A1) for bc in bcu]
150     [bc.apply(A2) for bc in bcp]
151
152     modelo = {'A1': A1,
153               'A2': A2,
154               'A3': A3,
155               'bcu': bcu,
156               'bcp': bcp,
157               'dados_problema':dados_problema,
158               'L1': L1,
159               'L2': L2,
160               'L3': L3,
161               'u_': u_,
162               'p_': p_,
163               'u0': u0,
164               'p_n': p_n}
165     return modelo
166
167
168 def solucao(modelo):
169     passos = 0
170     t = 0
171     for n in tqdm(range(modelo['dados_problema']['num_steps'])):
172
173         [bc.apply(modelo['A1']) for bc in modelo['bcu']]
174         [bc.apply(modelo['A2']) for bc in modelo['bcp']]
175
176         # Tentativa de velocidade
177         b1 = assemble(modelo['L1'])
178         [bc.apply(modelo['A1'], b1) for bc in modelo['bcu']]
179         solve(modelo['A1'], modelo['u_'].vector(), b1,
180         'bicgstab', 'hypre_amg')
181
```

```
182         # Correcao de Pressao
183         b2 = assemble(modelo['L2'])
184         [bc.apply(b2) for bc in modelo['bcp']]
185         solve(modelo['A2'], modelo['p_'].vector(), b2,
186         'bicgstab', 'hypre_amg')
187
188         # Correcao da velocidade
189         b3 = assemble(modelo['L3'])
190         solve(modelo['A3'], modelo['u_'].vector(), b3, 'cg',
191         'sor')
192
193         if n % 200 == 0:
194             pvd_file = File('TesteIPCS1/
195             velocidade-{0}.pvd'.format(passos))
196             pvd_file << modelo['u_']
197             pvd_file = File('TesteIPCS1/
198             pressao-{0}.pvd'.format(passos))
199             pvd_file << modelo['p_']
200             passos = passos + 1
201
202         modelo['u0'].assign(modelo['u_'])
203         modelo['p_n'].assign(modelo['p_'])
204         t = t + modelo['dados_problema']['dt']
205     return modelo
```

```
1 if __name__ == "__main__":
2     fluido_type = menu_fluido()
3     caminho = "lisav2.msh"
4     mesh_from_file = carregar_malha(caminho)
5     mesh, mvc, mf = mvc_mf(mesh_from_file)
6     dados_fluidos, dados_problema, Reynolds  =
7     dados_entrada(fluido_type)
8     modelo = modelo(mesh, dados_problema,dados_fluidos, mf)
9     resultado = solucao(modelo)
```

ANEXO B – ANEXOS E APÊNDICES 2

```
name = 100
caminho_lisa    =
str("liso/standoff/"+str(name)+".csv")
caminho_rugoso =
str("rugoso/standoff/"+str(name)+".csv")
df_rugoso = pd.read_csv(caminho_rugoso)
df_lisa = pd.read_csv(caminho_lisa)

df_lisa = df_lisa.dropna().reset_index(drop=True)
df_rugoso = df_rugoso.dropna().reset_index(drop=True)

df_lisa['f_24:0'] = df_lisa['f_24:0']/1000
df_lisa['f_24:1'] = df_lisa['f_24:1']/1000
df_lisa['Points:0'] = df_lisa['Points:0']/1000
df_rugoso['f_24:0']=df_rugoso['f_24:0']/1000
df_rugoso['f_24:1']=df_rugoso['f_24:1']/1000
df_rugoso['Points:0']=df_rugoso['Points:0']/1000

vx_lisa = df_lisa['f_24:0']
vx_rugoso = df_rugoso['f_24:0']
vy_lisa = df_lisa['f_24:1']
vy_rugoso = df_rugoso['f_24:1']
v_lisa = np.sqrt(vx_lisa**2 + vy_lisa**2).values
v_rugoso = np.sqrt(vx_rugoso**2 + vy_rugoso**2).values
v_lisa = vy_lisa
v_rugoso = vy_rugoso

x_lisa = df_lisa['Points:0'].values
x_rugoso = df_rugoso['Points:0'].values

nao_nulo_esq_lisa = np.argwhere((x_lisa < 100/1000)).flatten()
nao_nulo_esq_rugoso = np.argwhere((x_rugoso < 100/1000)).flatten()
nao_nulo_dir_lisa = np.argwhere((x_lisa > 100/1000)).flatten()
nao_nulo_dir_rugoso = np.argwhere((x_rugoso > 100/1000)).flatten()

E_lisa = x_lisa[nao_nulo_esq_lisa];
```

```
E_rugoso = x_rugoso[nao_nulo_esq_rugoso];
D_lisa = x_lisa[nao_nulo_dir_lisa];
D_rugoso= x_rugoso[nao_nulo_dir_rugoso];
h_e_lisa = np.max(E_lisa) - np.min(E_lisa)
h_e_rugoso = np.max(E_rugoso) - np.min(E_rugoso)
h_d_lisa = np.max(D_lisa) - np.min(D_lisa)
h_d_rugoso = np.max(D_rugoso) - np.min(D_rugoso)

v_e_lisa = v_lisa[nao_nulo_esq_lisa];
v_e_rugoso = v_rugoso[nao_nulo_esq_rugoso];
v_d_lisa = v_lisa[nao_nulo_dir_lisa]
v_d_rugoso = v_rugoso[nao_nulo_dir_rugoso]
Q_e_lisa = trapezoid(v_e_lisa)
Q_e_rugoso = trapezoid(v_e_rugoso)

Q_d_lisa = trapezoid(v_d_lisa)
Q_d_rugoso = trapezoid(v_d_rugoso)

deltax = 0.1
delta_e = np.min(E_lisa)+(h_e_lisa*deltax)
delta_d = np.max(D_lisa)-(h_d_lisa*deltax)

limite_zona_limpa_e = (delta_e,np.max(E_lisa))
limite_zona_limpa_d = (np.min(D_lisa),delta_d)

regiao_limpa_e_lisa    = np.argwhere(((x_lisa <
limite_zona_limpa_e[1])&(x_lisa >
limite_zona_limpa_e[0]))).
flatten()
regiao_limpa_d_lisa    = np.argwhere(((x_lisa <
limite_zona_limpa_d[1])&(x_lisa > limite_zona_limpa_d[0]))).
flatten()

regiao_limpa_e_rugoso = np.argwhere(((x_rugoso <
limite_zona_limpa_e[1])&
(x_rugoso > limite_zona_limpa_e[0]))).flatten()
regiao_limpa_d_rugoso = np.argwhere(((x_rugoso <
limite_zona_limpa_d[1])&(x_rugoso > limite_zona_limpa_d[0]))).flatten()

```

```
76 Qlimpo_e_lisa    = trapezoid(v_lisa[regiao_limpa_e_lisa])
77 Qlimpo_e_rugoso = trapezoid(v_rugoso[regiao_limpa_e_rugoso])
78
79 Qlimpo_d_lisa    = trapezoid(v_lisa[regiao_limpa_d_lisa])
80 Qlimpo_d_rugoso = trapezoid(v_rugoso[regiao_limpa_d_rugoso])
81
82 Q_A_E = Q_e_lisa
83 Q_A90_E = Qlimpo_e_lisa
84 Q_B_E = Q_e_rugoso
85 Q_B90_E = Qlimpo_e_rugoso
86 EtaE90 = Q_B90_E/Q_A90_E
87 EtaE = Q_B_E/Q_A_E
88
89 Q_A_D = Q_d_lisa
90 Q_A90_D = Qlimpo_d_lisa
91 Q_B_D = Q_d_rugoso
92 Q_B90_D = Qlimpo_d_rugoso
93 EtaD90 = Q_B90_D/Q_A90_D
94 EtaD = Q_B_D/Q_A_D
```