

# Simulação do Escoamento Bidimensional de Óleo em um Reservatório Empregando a Biblioteca Numérica PETSc

---

Fábio Bertolino Vasconcellos

Pós-graduação em Modelagem Computacional, Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Brasil.

Paulo de Tarço Honório Jr.

Unidade de Desenvolvimento Tecnológico Singrar, Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Brasil.

Grazione de Souza

Departamento de Modelagem Computacional, Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Brasil.

Helio Pedro Amaral Souto

Departamento de Modelagem Computacional, Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Brasil.

**Abstract:** We used the numerical library Portable Extensible Toolkit for Scientific Computation (PETSc) to obtain the solution of the system of algebraic equations arising from the discretization of the governing equation of a two-dimensional oil flow in a porous medium, where both are slightly compressible. Typically, this step is responsible for the highest computational cost compared to the others. Thus, in addition to the search for an accurate solution, we also aimed to achieve better computational performance. We determined the numerical solution in terms of the pressures in the reservoir and the producing well (via a well-reservoir coupling technique), using a Picard method linearization and a time-implicit formulation. We obtained the pressure variations as a function of time by using different computational meshes, modifying the physical properties, and using the methods for solving algebraic systems available at PETSc. The solutions presented the expected physical behavior, and we concluded that this numerical library was successful for the accomplished simulations.

**Palavras-chave:** algebraic systems of equations, numerical reservoir simulation, PETSc, wellbore pressure.

Corresponding author: Helio Pedro Amaral Souto, helio@iprj.uerj.br

Received: 15 October 2024; Accepted: 04 November 2024; Published: 29 December 2024.

## 1 Introdução

Com o surgimento dos computadores, após a segunda guerra mundial, uma série de problemas de engenharia puderam ser estudados a partir das soluções obtidas numericamente com a utilização de recursos computacionais. Nesta conjuntura, os avanços alcançados, em termos de capacidade de memória e de processamento, têm permitido que problemas cada vez mais realísticos sejam considerados como, por exemplo, os da dinâmica da tensão-deformação em equipamentos, da transferência de

calor, das reações químicas, da dinâmica dos fluidos computacional e da simulação de escoamentos em subsuperfície. Nessa última categoria, é possível se destacar a simulação de reservatórios de petróleo, imprescindível na proposição de um planejamento que resulte na produção otimizada, para que se aumente o fator de recuperação de hidrocarbonetos. Este trabalho é dedicado à simulação de reservatórios empregando a biblioteca Portable Extensible Toolkit for Scientific Computation (PETSc) [3], tendo em vista a necessidade de se resolver sistemas de equações algébricas de grande porte, presentes nos casos de interesse.

## 1.1 Resolução de sistemas de equações algébricas

Na simulação de reservatórios de petróleo na escala de campo [27, 39, 40, 42], frequentemente necessita-se resolver sistemas de equações algébricas não-lineares com mais de 1 milhão de incógnitas quando de problemas tridimensionais, mesmo no caso do escoamento monofásico isotérmico monocomponente. Quando se trata de simulações envolvendo a presença de poços horizontais, por exemplo, tipicamente é necessária uma discretização com malhas refinadas para capturar, adequadamente, o escoamento sujeito ao aparecimento dos diferentes efeitos de fronteira [1, 43]. Em geral, na simulação de escoamentos monofásicos em reservatórios de petróleo, mais de 70% do esforço computacional é despendido no processo de resolução desses sistemas.

A questão do tempo de execução é crucial na simulação de reservatórios, levando em consideração que, para casos reais, as simulações visam a prever a produção ao longo de décadas e podem, em certas situações, durar semanas, mesmo em máquinas destinadas à computação de alto desempenho [36, 37]. Dessa forma, é essencial que a indústria de óleo e gás busque empregar métodos numéricos eficientes para a resolução de sistemas de equações algébricas, além da implementação de técnicas de paralelização, de modo que seja possível testar, mais rapidamente, diferentes cenários de produção, a fim de se definir o melhor plano de operação.

A fim de se resolver de forma computacionalmente eficiente esses sistemas de equações, diferentes grupos têm, ao longo de décadas, atuado na elaboração de métodos e na construção de códigos numéricos que conduziram à criação de bibliotecas voltadas para a álgebra linear [12, 16]. Elas são constituídas por códigos eficientes e flexíveis destinados à resolução de sistemas do tipo  $\mathbf{Ax} = \mathbf{b}$ , onde  $\mathbf{A}$  é a matriz dos coeficientes,  $\mathbf{x}$  é o vetor incógnita e  $\mathbf{b}$  um vetor cujos elementos são conhecidos. Portanto, necessita-se conhecer corretamente de que forma essas informações devem ser repassadas para tais bibliotecas numéricas. Outro ponto importante, no que diz respeito à eficiência computacional, é o hardware disponível para a realização das simulações [19].

## 1.2 A biblioteca PETSc na simulação numérica

A biblioteca Portable, Extensible Toolkit for Scientific Computation fornece um conjunto de estrutura de dados e rotinas para aplicações em larga escala, com recursos para a execução em paralelo ou em série, favorecendo a implementação de códigos destinados à computação científica, os quais podem ser escritos, por exemplo, nas linguagens C, C++, Fortran ou Python. Ela contém a biblioteca Toolkit for Advanced Optimization (TAO) voltada para a resolução de equações diferenciais parciais e problemas que demandam um elevado desempenho computacional. Por padrão, todos os programas são executados utilizando a Message Passing Interface (MPI) [3]. A sua estrutura computacional permite uma integração com a linguagem escolhida pelo usuário, possibilitando

o aproveitamento de um programa computacional pré-existente, flexibilizando a reutilização de códigos numéricos.

Na sequência, foca-se na sua utilização nas engenharias e na busca pela melhoria do desempenho computacional, uma breve revisão bibliográfica das aplicações da biblioteca PETSc. Destaca-se que ela possui, conforme relatado em [29], métodos numéricos específicos para a resolução de equações diferenciais e algébricas não-lineares, dependentes do tempo, e para a otimização numérica.

Croucher et al. (2020) [12], por exemplo, utilizaram o simulador Waiwera que foi projetado para resolver problemas envolvendo sistemas geotérmicos. Esse código, segundo os autores, foi paralelizado e fez uso intensivo da PETSc [25, 29, 35]. O referido simulador foi escrito em linguagem Fortran 2003, orientada a objetos, e é do tipo open-source. Os autores relataram que os resultados com ele obtidos apresentaram consistência com aqueles alcançados com outros simuladores reconhecidos na literatura, tais como o TOUGH2 (Transport Of Unsaturated Groundwater and Heat), além de possuir uma boa escalabilidade quando do uso da versão paralelizada.

Segundo Wenqing et al. (2015) [35], as tecnologias destinadas à estocagem de energia térmica são de interesse atual e buscam melhorar a integração da geração de energia renovável e a eficiência energética. As simulações numéricas do armazenamento termoquímico de calor são desafiadoras e computacionalmente custosas, devido à complexidade em se descrever os processos acoplados e não lineares característicos desses sistemas. Tais dificuldades aumentam quando são consideradas geometrias complexas. Para se viabilizar simulações computacionalmente eficientes, um código utilizando o método dos elementos finitos foi paralelizado, usando as rotinas da biblioteca PETSc, como parte da estrutura do código aberto OpenGeoSys. De acordo com os autores, na simulação da descarga de um armazenador de calor, o tempo de execução foi reduzido significativamente.

As simulações abrangendo a geodinâmica são também caracterizadas pela existência de não linearidades e efeitos tridimensionais, impondo desafios à ciência da computação. Nesse caso específico, Katz et al. (2007) [25] discutiram como o uso da PETSc pode facilitar o desenvolvimento dos simuladores. Como ilustração, os autores realizaram simulações do escoamento estacionário de um fluido não-newtoniano; de ondas magmáticas no manto terrestre; e bifásicos, em meios com de alta porosidade, sob cisalhamento simples. Eles destacaram dois dos recursos adicionais da PETSc: o armazenamento estruturado de parâmetros e a saída autodocumentada.

Cuomo et al. (2015) [13] apresentaram uma estrutura, utilizando a computação paralela, direcionada à simulação do fluxo óptico usando um cluster de Graphics Processing Unit (GPUs) e a PETSc. Para tanto, na modelagem foi considerada uma equação diferencial parcial (EDP) parabólica e foi implementado um algoritmo paralelizado, adequado para ambientes de computação heterogêneos (multiprocessado, uma ou várias GPUs).

Como resultado, o software proposto possibilitou um ganho de cerca de 95% em relação à implementação sequencial.

Por sua vez, a suíte TOUGH, composta por simuladores direcionados ao escoamento multifásico não isotérmico e ao transporte de calor em meios porosos, vem sendo atualizada, por vários desenvolvedores, ao longo dos anos. Em função da crescente complexidade dos processos, bem como da extensão dos domínios de resolução, surgiu a necessidade da realização de melhorias no que diz respeito à eficiência computacional. Jung et al. (2017) [23] relataram que os métodos de resolução de sistemas lineares, paralelizados, da PETSc foram adicionados nas versões TOUGH2 e TOUGH2-MP. Como conclusão, eles constataram que os métodos da PETSc apresentaram, em geral, um melhor desempenho do que os do Aztec paralelizado (original) e do solver linear interno da TOUGH3.

Um outro problema desafiador é a simulação numérica da eletrofisiologia cardíaca. O sistema Bidomain, um modelo matemático da atividade bioelétrica cardíaca, constituído por uma equação diferencial parcial elíptica e uma parabólica do tipo reação-difusão, descreve a propagação da excitação elétrica no tecido cardíaco. As duas equações encontram-se acopladas a um sistema de equações diferenciais ordinárias (EDOs), que simulam as correntes iônicas através da membrana cardíaca. Na busca pela redução do custo computacional, o desenvolvimento de pré-condicionadores eficientes e escalonáveis para a resolução dos sistemas lineares, provenientes da discretização dessas equações, é fundamental. Centofanti e Scacchi (2024) [10] estudaram o mesmo problema utilizando a PETSc e arquiteturas modernas de computação de alto desempenho, usando várias Central Processing Units (CPUs) e GPUs, e verificaram que houve melhorias no desempenho computacional. Ainda segundo os autores, a portabilidade da PETSc engloba a utilização de GPUs e enfatiza a flexibilidade e a extensibilidade, permitindo que os desenvolvedores usem o seu modelo de programação predileto.

De fato, há uma série de problemas de interesse na Física, na Matemática, na Computação e na Engenharia onde a biblioteca PETSc tem sido utilizada. Em se tratando do trabalho desenvolvido por Kempe et al. (2015) [26], foi abordado o problema do escoamento no interior de uma cavidade, cuja tampa superior é móvel. Foi feita uma comparação do desempenho computacional quando da obtenção da solução das equações de Navier-Stokes, utilizando o método dos volumes finitos, considerando os fluidos como sendo incompressíveis e empregando um método de projeção. O código numérico foi executado em CPUs e GPUs, juntamente com a PETSc.

Por outro lado, Kang et al. (2023) [24] utilizaram os métodos de Runge-Kutta e a biblioteca PETSc para resolver um problema de interação fluido-fluido. Eles usaram as equações de Navier-Stokes para fluidos compressíveis e realizaram uma série de experimentos numéricos em larga escala.

Prosseguindo, Bonart et al. (2017) [6] trabalharam no acoplamento entre as bibliotecas numéricas da PETSc e ADOL-C. Trata-se de um pacote que facilita a avaliação de derivadas primeiras e superiores de funções vetoriais que são definidas em códigos escritos nas linguagens C ou C++. As suas rotinas podem ser chamadas via programas que usem linguagens de programação vinculadas à linguagem C. Os autores empregaram a computação paralela para estudar um modelo de coluna de destilação em condições de não equilíbrio.

Um novo esquema, massivamente paralelo, foi desenvolvido para estudar escoamentos em superfícies livres usando o método livre de malha Incompressible Smoothed Particle Hydrodynamics (ISPH), para simulações envolvendo o uso de mais de 100 milhões de partículas [20]. Com esse propósito, é requerido que seja resolvido um sistema no qual a matriz esparsa, associada à equação de Poisson escrita em termos da pressão, possui uma ordem elevada. Essa não é uma tarefa trivial para problemas que usam tal quantidade de partículas, que se movem em evolução contínua. A biblioteca PETSc e o pré-condicionador HYPRE BoomerAMG foram utilizados a fim de se garantir uma rápida convergência.

Já Boffi et al. (2024) [5] tiveram como objetivo desenvolver um solver paralelizado direcionado para problemas de interação fluido-estrutura. Eles empregaram multiplicadores de Lagrange e o método dos elementos finitos para a discretização espacial. Por outro lado, é utilizado um esquema de primeira ordem, do tipo diferenças finitas, na aproximação temporal. O sistema linear resultante do processo de discretização é resolvido, a cada passo de tempo, usando a versão paralelizada do método Generalized Minimum Residual (GMRES) e pré-condicionadores disponíveis na biblioteca PETSc.

No escopo dos aplicativos de geração de imagens, Carracciolo et al. (2006) [9] analisaram o esforço computacional requerido para o desenvolvimento de um ambiente de software distribuído. Tal iniciativa teve por finalidade possibilitar o uso de computadores de alto desempenho e sistemas de armazenamento, para a redução do ruído das sequências ecocardiográficas tridimensionais, assim como a utilização da PETSc.

Em contrapartida, Hovland e McInnes (2001) [21] utilizaram a biblioteca PETSc em aplicações aeroespaciais envolvendo o uso da paralelização e um método de Newton-Krylov-Schwarz. Eles estudaram o escoamento ao redor de um aerofólio do tipo M6 utilizando as equações tridimensionais de Euler e uma formulação totalmente implícita no tempo.

No caso das equações incompressíveis de Navier-Stokes, Clarke e Krishnamoorthy (2019) [11] relataram que há um custo computacional expressivo quando da resolução numérica de escoamentos multifásicos. Ele é decorrente, em geral, do emprego de uma formulação implícita quando do cálculo da correção de pressão. Os autores utilizaram o método iterativo dos Gradientes Biconjugados Estabilizado e o pré-condicionador de Jacobi, ambos da biblioteca PETSc.

Por fim, destaca-se que em grande parte dos simuladores contemporâneos, aplicados na engenharia, é necessário que se resolva sistemas de equações algébricas. Assim sendo, Jolivet et al. (2021) [22] discutem como as bibliotecas PETSc e High Performance Domain Decomposition Methods (HPDDM) podem ser usadas, em conjunto, de modo a propiciar aos usuários a utilização dos métodos avançados do subespaço de Krylov (para a resolução de sistemas de equações) e de Schwarz para o particionamento de domínios. São também fornecidos exemplos de aplicações envolvendo a análise de autovaleores, a elasticidade, a combustão e o eletromagnetismo.

Aos leitores interessados destaca-se, dentre outras contribuições que fizeram uso da biblioteca PETSc, os seguintes trabalhos: [4, 7, 28] e [41].

## 2 Escoamento Monofásico de Óleo em um Reservatório

A modelagem físico-matemática, tendo-se em vista a simulação do escoamento monofásico isotérmico em um reservatório de petróleo, depende das características do escoamento, das propriedades físicas do reservatório e do fluido e, principalmente, das hipóteses adotadas. Esta seção é dedicada à apresentação do modelo adotado, das propriedades de fluido e de rocha, das equações de balanço, da equação diferencial parcial governante, cuja variável dependente é a pressão do óleo, e das condições inicial e de contorno.

As seguintes hipóteses são utilizadas na derivação do modelo de escoamento:

1. O campo de permeabilidades é heterogêneo e isotrópico;
2. A compressibilidade da rocha é pequena e constante;
3. O fluido é newtoniano;
4. Não ocorrem reações químicas;
5. O escoamento é bidimensional, no plano  $xy$ , e em regime laminar;
6. O escoamento é monofásico e isotérmico;
7. Os efeitos gravitacionais são desconsiderados;
8. O poço de produção é vertical e penetra totalmente na formação;
9. Ausências de dano à formação e de estocagem no poço.

Inicia-se pela introdução do fator-volume-formação do óleo [32],  $B = \rho_{osc}/\rho_o$ , de modo que a equação de conservação da massa seja escrita na forma [18]

$$\frac{\partial}{\partial t} \left( \frac{\phi}{B} \right) + \nabla \cdot \left( \frac{\mathbf{v}_o}{B} \right) - \frac{q_m}{V_b \rho_{osc}} = 0, \quad (1)$$

onde  $\phi$  é a porosidade,  $\mathbf{v}_o$  é a velocidade superficial do óleo,  $q_m$  é o termo fonte,  $V_b$  é o volume total (rocha mais poros),  $\rho_o$  é a massa específica do óleo nas condições de reservatório e  $\rho_{osc}$  representa a massa específica do óleo em condições padrão de temperatura e pressão,  $T_{sc}$  e  $p_{sc}$ , respectivamente.

Para escoamentos a baixas velocidades, a equação que expressa a conservação da quantidade de movimento, no escoamento de fluidos em meios porosos, é a lei de Darcy clássica [14, 18, 38],

$$\mathbf{v}_o = -\frac{\mathbf{k}}{\mu_o} (\nabla p - \rho_o g \nabla D), \quad (2)$$

onde  $\mathbf{k}$  é o tensor de permeabilidade absoluta,  $\mu_o$  é a viscosidade do óleo,  $g$  é a magnitude da aceleração da gravidade e  $D$  é a profundidade [32].

Agora, após substituição da Eq. (2) na Eq. (1), obtém-se

$$\frac{\partial}{\partial t} \left( \frac{\phi}{B} \right) = \nabla \cdot \left[ \frac{\mathbf{k}}{B\mu_o} (\nabla p - \rho_o g \nabla D) \right] + \frac{q_m}{V_b \rho_{osc}}. \quad (3)$$

Se os efeitos gravitacionais não forem levados em consideração e introduzindo-se a relação  $q_m = q_{sc} \rho_{osc}$  tem-se, a partir da Eq. (3), que

$$\frac{\partial}{\partial t} \left( \frac{\phi}{B} \right) - \nabla \cdot \left( \frac{\mathbf{k}}{B\mu_o} \nabla p \right) - \frac{q_{sc}}{V_b} = 0. \quad (4)$$

Aqui, considera-se que [18]

$$B = \frac{B^0}{1 + c_o(p - p^0)}, \quad (5)$$

$$\phi = \phi^0 [1 + c_\phi(p - p^0)], \quad (6)$$

onde  $B^0$  e  $\phi^0$  são, respectivamente, o fator-volume-formação e a porosidade nas condições de referência para a pressão  $p^0$ , e  $c_o$  e  $c_\phi$  representam as compressibilidades do óleo e da rocha, respectivamente. As compressibilidades de óleo e de rocha são consideradas pequenas e constantes, de forma que óleo e rocha são ligeiramente compressíveis.

Na Eq. (4), o primeiro termo, do lado esquerdo do sinal de igualdade, pode ser reescrito de forma a se obter explicitamente as derivadas parciais da pressão em relação ao tempo [17],

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\phi}{B} \right) &= \frac{1}{B} \frac{\partial \phi}{\partial t} + \phi \frac{\partial}{\partial t} \left( \frac{1}{B} \right) \\ &= \frac{1}{B} \frac{d\phi}{dp} \frac{\partial p}{\partial t} + \phi \frac{d}{dp} \left( \frac{1}{B} \right) \frac{\partial p}{\partial t} \\ &= \left[ \frac{1}{B} \frac{d\phi}{dp} + \phi \frac{d}{dp} \left( \frac{1}{B} \right) \right] \frac{\partial p}{\partial t}. \end{aligned} \quad (7)$$

Em seguida, introduz-se as Eqs. (5)-(6) na Eq. (7), de modo que

$$\frac{\partial}{\partial t} \left( \frac{\phi}{B} \right) = \left( \frac{\phi c_o}{B^0} + \frac{\phi^0 c_\phi}{B} \right) \frac{\partial p}{\partial t}. \quad (8)$$

Finalmente, a partir das Eqs. (4) e (8) é possível se obter a equação governante

$$\Gamma_p \frac{\partial p}{\partial t} - V_b \nabla \cdot \left( \frac{\mathbf{k}}{B\mu_o} \nabla p \right) - q_{sc} = 0, \quad (9)$$

onde

$$\Gamma_p = V_b \left( \frac{\phi c_o}{B^0} + \frac{\phi^0 c_\phi}{B} \right). \quad (10)$$

Quando da determinação da viscosidade do fluido, utiliza-se a correlação frequentemente aplicada ao caso de fluidos ligeiramente compressíveis em escoamentos isotérmicos [18],

$$\mu_o = \frac{\mu^0}{1 - c_\mu(p - p^0)}, \quad (11)$$

onde  $\mu^0$  é o valor da viscosidade medida na pressão de referência  $p^0$  e  $c_\mu$  é o coeficiente de variação da viscosidade em relação à pressão.

## 2.1 Condições auxiliares

A Eq. (9) é uma equação diferencial parcial não-linear, utilizada na determinação da pressão do óleo. Para se resolvê-la, é ainda necessário que sejam fornecidas as condições auxiliares (inicial e de contorno) apropriadas.

Como condição inicial utiliza-se

$$p(x, y, t = 0) = p_{ini}(x, y) = p_{inic}, \quad (12)$$

onde  $p_{inic}$  é a pressão inicial antes de começar a produção de óleo.

No que diz respeito às condições de contorno, considera-se fluxo nulo nas fronteiras do reservatório, logo,

$$\left( \frac{\partial p}{\partial x} \right)_{x=0, L_x} = \left( \frac{\partial p}{\partial y} \right)_{y=0, L_y} = 0, \quad (13)$$

onde  $L_x$  e  $L_y$  são os respectivos comprimentos do reservatório nas direções dos eixos  $x$  e  $y$ .

## 2.2 Acoplamento poço-reservatório

Se o termo fonte  $q_{sc}$  for utilizado para fornecer a vazão de produção na representação do poço, via uma técnica de acoplamento poço-reservatório, é possível escrever [18]

$$q_{sc} = -J_w (p - p_{wf}), \quad (14)$$

onde  $J_w$  é o índice de produtividade e  $p_{wf}$  é a pressão no poço.

A determinação numérica do índice de produtividade, no contexto da simulação de reservatórios, será discutida na seção dedicada à resolução numérica. Por hora, ressalta-se que esse acoplamento permite o cálculo da pressão no poço se a sua vazão for prescrita e vice-versa. Neste trabalho, optou-se por utilizar uma condição de vazão de produção prescrita.

## 3 Resolução Numérica

A simulação numérica é amplamente utilizada nas engenharias e, principalmente, na resolução de problemas de escoamento em reservatórios de petróleo, visto que apenas em alguns casos particulares, simplificados, é possível se obter soluções analíticas para os mesmos [18]. Agora, tratar-se-á das questões relativas à obtenção da solução da equação governante introduzida previamente. Para tanto, utiliza-se os métodos dos Volumes Finitos e de Picard, um método iterativo de resolução de sistemas algébricos, e uma técnica específica para a determinação do índice de produtividade.

Os valores da pressão são obtidos no centro dos volumes finitos e o sistema algébrico de equações resultante do processo de discretização é resolvido mediante o emprego de um método iterativo, uma vez que utiliza-se uma formulação totalmente implícita no tempo.

Para um problema bidimensional, a equação governante (9) pode ser posta na forma

$$\frac{\partial}{\partial x} \left( T_x \frac{\partial p}{\partial x} \right) dx + \frac{\partial}{\partial y} \left( T_y \frac{\partial p}{\partial y} \right) dy = \Gamma_p \frac{\partial p}{\partial t} + q_{sc}, \quad (15)$$

onde foram introduzidas as novas variáveis

$$T_x \equiv \frac{A_x k_x}{\mu B}, \quad (16)$$

$$T_y \equiv \frac{A_y k_y}{\mu B}, \quad (17)$$

sendo que  $A_x = dyL_z$ ,  $A_y = dxL_z$  e  $V_b = dxdyL_z$ , considerando-se que  $L_z$  é a espessura do reservatório e que  $A_x$  e  $A_y$  não são, respectivamente, funções de  $x$  e de  $y$  [18].

No processo de discretização da Eq. (15), emprega-se uma formulação totalmente implícita no tempo, diferenças centradas a três pontos na aproximação das derivadas espaciais e uma aproximação do tipo diferença recuada para a derivada temporal. Portanto, feitas todas as substituições, a forma final discretizada da equação governante é dada por [34]

$$\begin{aligned} & \mathbb{T}_{x, i+1/2, j}^{n+1} (p_{i+1, j}^{n+1} - p_{i, j}^{n+1}) - \mathbb{T}_{x, i-1/2, j}^{n+1} (p_{i, j}^{n+1} - p_{i-1, j}^{n+1}) \\ & + \mathbb{T}_{y, i, j+1/2}^{n+1} (p_{i, j+1}^{n+1} - p_{i, j}^{n+1}) - \mathbb{T}_{y, i, j-1/2}^{n+1} (p_{i, j}^{n+1} - p_{i, j-1}^{n+1}) \\ & = \Gamma_{i, j}^{n+1} (p_{i, j}^{n+1} - p_{i, j}^n) + (q_{sc})_{i, j}^{n+1}, \end{aligned} \quad (18)$$

onde  $i - 1/2$  e  $i + 1/2$  representam a posição das faces dos volumes finitos na direção do eixo  $x$ , cujo nó central é  $i, j$  (com definições análogas para  $j$  e a direção do eixo  $y$ ).

Na Eq. (18), foram introduzidas as novas variáveis conhecidas como as transmissibilidades:

$$\mathbb{T}_{x, i \pm \frac{1}{2}, j}^{n+1} = \left( \frac{A_x k_x}{\mu B \Delta x} \right)_{i \pm \frac{1}{2}, j}^{n+1}, \quad (19)$$

$$\mathbb{T}_{y,i,j \pm \frac{1}{2}}^{n+1} = \left( \frac{A_y k_y}{\mu B \Delta y} \right)_{i,j \pm \frac{1}{2}}^{n+1}. \quad (20)$$

Quando da determinação delas, emprega-se uma média harmônica em se tratando das propriedades de rocha e de geometria, e uma média aritmética (sendo a malha uniforme) com respeito às do fluido. Assim, obtém-se os seus valores em  $(i \pm 1/2, j)$  e  $(i, j \pm 1/2)$ , a partir daqueles conhecidos em  $(i, j)$ ,  $(i \pm 1, j)$  e  $(i, j \pm 1)$ .

No que diz respeito ao coeficiente do termo transitente, ele é calculado a partir de uma expansão conservativa [18],

$$\Gamma_{i,j}^{n+1} = \left[ \frac{\phi^n c_o}{B^0} + \frac{\phi^0 c_\phi}{B^{n+1}} \right] \frac{V_{b,i,j}}{\Delta t}, \quad (21)$$

onde  $\Delta t = t^{n+1} - t^n$  representa o incremento de tempo.

### 3.1 Linearização do sistema de equações algébricas não-lineares

A Equação (18) representa um conjunto de equações algébricas não-lineares. Portanto, para que se possa utilizar as técnicas desenvolvidas especificamente para a resolução de sistemas de equações lineares, elas devem ser linearizadas. Então, tendo em vista a utilização do Método de Picard, para a determinação da transmissibilidade na direção do eixo  $x$  [18], introduz-se as aproximações

$$\mathbb{T}_{x,i \pm \frac{1}{2},j}^{n+1} \cong \mathbb{T}_{x,i \pm \frac{1}{2},j}^{n+1,v} = \left( \frac{A_x k}{\mu B \Delta x} \right)_{i \pm \frac{1}{2},j}^{n+1,v}, \quad (22)$$

$$\mathbb{T}_{y,i,j \pm \frac{1}{2}}^{n+1} \cong \mathbb{T}_{y,i,j \pm \frac{1}{2}}^{n+1,v} = \left( \frac{A_y k}{\mu B \Delta y} \right)_{i,j \pm \frac{1}{2}}^{n+1,v}. \quad (23)$$

Portanto, a Equação (18) pode ser reescrita como

$$\begin{aligned} & \mathbb{T}_{x,i+\frac{1}{2},j}^{n+1,v} \left( p_{i+1,j}^{n+1,v+1} - p_{i,j}^{n+1,v+1} \right) \\ & - \mathbb{T}_{x,i-\frac{1}{2},j}^{n+1,v} \left( p_{i,j}^{n+1,v+1} - p_{i-1,j}^{n+1,v+1} \right) \\ & + \mathbb{T}_{y,i,j+\frac{1}{2}}^{n+1,v} \left( p_{i,j+1}^{n+1,v+1} - p_{i,j}^{n+1,v+1} \right) \\ & - \mathbb{T}_{y,i,j-\frac{1}{2}}^{n+1,v} \left( p_{i,j}^{n+1,v+1} - p_{i,j-1}^{n+1,v+1} \right) \\ & = \Gamma_{i,j}^{n+1,v} \left( p_{i,j}^{n+1,v+1} - p_{i,j,k}^n \right) + (q_{sc})_{i,j}^{n+1,v}, \quad (24) \end{aligned}$$

onde os níveis de iteração são indicados por  $v$  (valores conhecidos) e  $v + 1$  (valores desconhecidos). O mesmo procedimento é utilizado para os termos  $\Gamma_{i,j}$  e  $(q_{sc})_{i,j}$ .

### 3.2 Determinação do índice de produtividade

No caso do acoplamento poço-reservatório, também se lineariza o termo que contém o índice de produtividade  $J_w$ . Aqui, apenas poços verticais são considerados, negligenciando-se as perdas de carga e os efeitos inerciais nos seus interiores. A vazão de produção no poço,  $q_{sc}$ , está relacionada com a diferença de pressão por intermédio do índice de produtividade e a sua forma discretizada é dada por [34]

$$(q_{sc})_{i,j}^{n+1,v} = -J_{w,i,j}^{n+1,v} \left[ p_{i,j}^{n+1,v+1} - (p_{wf})_{i,j}^{n+1,v+1} \right]. \quad (25)$$

Assim sendo, em se tratando desse caso em particular, o índice de produtividade é calculado por [1]

$$J_{w,i,j}^{n+1,v} = \left[ \frac{2\pi \sqrt{k_x k_y} \Delta z}{\mu B \ln \left( \frac{r_{eq}}{r_w} \right)} \right]_{i,j}^{n+1,v}, \quad (26)$$

onde

$$r_{eq} = 0,28 \left[ \frac{\sqrt{\frac{k_y}{k_x}} (\Delta x)^2 + \sqrt{\frac{k_x}{k_y}} (\Delta y)^2}{\left( \frac{k_y}{k_x} \right)^{1/4} + \left( \frac{k_x}{k_y} \right)^{1/4}} \right]_{i,j,k}, \quad (27)$$

com  $r_{eq}$  e  $r_w$  representando os raios equivalente [31] e do poço, respectivamente.

### 3.3 Metodologia de solução do sistema de equações algébricas

Os sistemas oriundos da linearização das equações algébricas não-lineares têm, em geral, como características o fato da matriz dos coeficientes ser esparsa e a elevada quantidade de incógnitas. Portanto, necessita-se de muita memória e velocidade de processamento quando da resolução deles.

O método dos Gradientes Conjugados (GC) foi o escolhido para ser empregado na resolução do sistema linearizado de equações algébricas. Ele é considerado como um dos métodos iterativos mais eficientes para a resolução de sistemas lineares não-estacionários, pois herda informações das iterações precedentes e as considera quando da realização da subsequente. Resumidamente, as suas etapas consistem em executar passos, a cada iteração, na direção oposta a do gradiente da função a ser minimizada, de tal forma que a informação proveniente da direção já pesquisada não seja repetida, até que seja encontrado o mínimo estrito e global. A minimização ocorre nos chamados subespaços de pesquisa (espaço de Krylov), gerados a partir dos resíduos de cada iteração [8].

Supondo-se que não sejam introduzidos erros de arredondamento, se a matriz dos coeficientes  $\mathbf{A}$  possui  $m$  autovalores distintos, o método deve convergir em no

máximo  $m$  iterações. Por outro lado, se ela possuir muitos autovalores distintos cujas magnitudes variam enormemente, o método provavelmente necessitará de um grande número de iterações para que ela seja alcançada [33].

Por sua vez, a convergência pode ser acelerada mediante o pré-condicionamento da matriz  $\mathbf{A}$ . Ele consiste em transformar o sistema original em um outro que tem a mesma solução, porém, possuindo propriedades mais favoráveis para que a convergência seja atingida mais rapidamente [33]. A matriz pré-condicionadora é escolhida de tal forma que  $\mathbf{A}' = \mathbf{C}\mathbf{A}\mathbf{C}^T$ , onde a matriz  $\mathbf{C}$  é dita ser não-singular e pré-condicionadora. A biblioteca PETSc disponibiliza diferentes métodos de pré-condicionamento como, por exemplo, o Symmetric Successive Over-Relaxation (SSOR).

#### 4 Implementação da Biblioteca PETSc

De modo a viabilizar a utilização da biblioteca PETSc, é necessário instalá-la juntamente com um conjunto de “pacotes de programas” que proporcionam a compilação e a execução do programa computacional criado pelo usuário. As orientações para a sua instalação e utilização encontram-se disponíveis no endereço eletrônico <https://petsc.org> [3]. Neste trabalho, a distribuição empregada é a PETSc 3.18 de 2022 e o código numérico foi desenvolvido usando a linguagem de programação C.

Após a discretização da equação governante, a matriz dos coeficientes  $\mathbf{A}$ , associada ao sistema de equações algébricas linearizadas, é esparsa, simétrica, pentadiagonal e com predominância da diagonal principal. As componentes do vetor  $\mathbf{b}$  contém as pressões obtidas no passo de tempo anterior, acrescidas do termo fonte.

No Algoritmo 1, lista-se as principais etapas a serem efetivadas para a implementação da utilização de um dos métodos de resolução do pacote Krylov Subspace Methods (KSP) contidos na sua biblioteca <petscksp.h>.

Algoritmo 1: Resolução básica utilizando a PETSc

- 1 Inicialização da biblioteca PETSc adequada ao problema;
- 2 Declaração das variáveis do tipo PETSc;
- 3 Obtenção dos parâmetros do problema;
- 4 Inicialização das matrizes e vetores;
- 5 Configuração do solver;
- 6 Resolução do sistema linear  $\mathbf{Ax} = \mathbf{b}$  pelo método escolhido;
- 7 Verificação dos resultados e/ou arquivamento de dados;
- 8 Liberação da memória e finalização do uso da biblioteca PETSc.

A função `PetscInitialize` é a encarregada por iniciar a aplicação da biblioteca, sendo a primeira função da PETSc dentro do código.

#### 4.1 Declaração das variáveis

Os principais tipos de dados da PETSc podem ser vistos na Tabela 1. Um fato interessante é a possibilidade de se alterar o tamanho das variáveis. Por exemplo, é possível mudar a configuração de uma variável do tipo `PetscInt` de 32 bits para 64 bits.

Tabela 1 Tipos de dados da biblioteca PETSc

Tipo	Descrição
<code>PetscInt</code>	Variável para números inteiros
<code>PetscScalar</code>	Variável para números reais com precisão dupla
<code>PetscReal</code>	Representa um numero Real de um <code>PetscScalar</code>
<code>PetscBool</code>	Variável lógica: 0 ou 1
<code>Vec</code>	Declaração de um vetor
<code>Mat</code>	Declaração de uma matriz
<code>KSP</code>	Declaração de soluções da família KSP
<code>PC</code>	Declaração dos pré-condicionadores

Os parâmetros de entrada do simulador encontram-se organizados em arquivos, facilitando a realização das simulações numéricas e evitando a necessidade da recompilação do código numérico a cada alteração. Na sequência, é fornecido um exemplo de declaração de variáveis, onde  $N_x$  e  $N_y$  representam os números de células na malha computacional, nas direções dos eixos  $x$  e  $y$ , respectivamente:

```
//Declaração das variáveis principais
Vec          x,b;
Mat          A;
Ksp          ksp;
PC           pc;
PetscInt     Nx,Ny,col[5];
PetscScalar  *resultado,v[5];
//Variáveis comuns
int          i,j;
double      Lx,Ly,Lz;
```

As variáveis seguem as mesmas regras da sintaxe da linguagem C, podendo ser declaradas como global ou local. Os vetores e matrizes são apenas apontados, sendo necessário realizar a alocação de memória e as configurações dos parâmetros.

#### 4.2 Inicialização das matrizes e vetores

As matrizes e os vetores necessitam de uma atenção especial em função do tipo de método que se deseja utilizar, influenciando diretamente nas configurações e alocações de memória. Por padrão, as matrizes são alocadas em uma forma compactada via a técnica conhecida como Compressed Sparse Row (CSR) [37]. Segue um exemplo de configuração da matriz e dos vetores:

```
//Parametrização da matriz
  MatCreate(PETSC_COMM_WORLD, &A);
  MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE,
Nx * Ny, Nx * Ny);
  MatSetFromOptions(A);
  MatSetUp(A);
//Parametrização do vetor
  VecCreate(PETSC_COMM_WORLD, &x);
  VecSetSizes(x, PETSC_DECIDE, Nx * Ny);
  VecSetFromOptions(x);
//Função para configurar vetores com as mesmas
dimensões
  VecDuplicate(x, &b);
```

Após a locação de memória, pode-se passar para a etapa de definição dos valores dos elementos dos vetores e da matriz. É fornecido, agora, um exemplo com a inicialização do vetor **b** e da matriz **A**. Para inserir cada linha da matriz, é necessário fornecer um array com a posição e outro contendo os valores, *col* e *v* respectivamente, enquanto que a variável *i* informa qual é a linha:

```
//Função para alocar um valor a todos os
elementos do vetor
  VecSet(b, P_ref);
//Função para inserir os valores de uma linha
na matriz pentadiagonal "A"
  MatSetValues(A, 1, &i, 5, col, v,
INSERT_VALUES)
//Funções de construção devem ser chamadas
  MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY);
  MatAssemblyEnd(A, MAT_FINAL_ASSEMBLY);
```

### 4.3 Configuração do método de resolução

Uma vez escolhido o método KSP, é necessário se configurar as suas opções. Quando uma função de configuração não é chamada, a PETSc utiliza as definidas como padrão. Por exemplo, caso não esteja determinado que o KSPSetType é o do método dos Gradientes Conjugados (KSPCG), por padrão seria escolhido o GMRES. Por intermédio da função KSPSetTolerances são estabelecidos os critérios de tolerância:

```
//Criação do solver linear e configurações das
opções
  KSPCreate(PETSC_COMM_WORLD, &ksp);
  KSPSetType(ksp, KSPCG);
  KSPSetOperators(ksp, A, A);
  KSPSetTolerances(ksp, 1.e-10, 1.e-50,
PETSC_DEFAULT, PETSC_DEFAULT);
```

### 4.4 Resolução do sistema linear

A função KSPSolve é a responsável por dar início ao processo de resolução do sistema algébrico  $\mathbf{Ax} = \mathbf{b}$ , sendo

imprescindível fornecer os vetores **x** e **b**, sabendo-se que a matriz **A** já foi configurada via a função KSPSetOperators. Após ter sido alcançada a convergência, a solução do sistema é armazenada no vetor **x**.

A função KSPSetInitialGuessNonzero permite informar a estimativa inicial. Quando ela não for configurada, o padrão é o de se começar com  $\mathbf{x} = \mathbf{0}$ , que pode aumentar consideravelmente o número de iterações e o tempo de execução até a convergência, de acordo com a tolerância estipulada e o tipo de problema físico-matemático considerado,

```
//Resolução do sistema linear
  KSPSetInitialGuessNonzero(ksp,
PETSC_TRUE);
  KSPSetFromOptions(ksp);
  KSPSolve(ksp, b, x);
```

### 4.5 Verificação dos resultados

Para se ter acesso às informações de cada componente do vetor solução é necessário utilizar um ponteiro e a função VecGetArray. Também pode-se manipular diretamente o próprio vetor, mediante o uso de diversas funções pré-existentes como, por exemplo, a função VecView, que exhibe os elementos do vetor:

```
//Função para apontar os dados do vetor para o
ponteiro "resultado"
  VecGetArray(x, &resultado);
//Imprimir na tela o vetor solução
  VecView(x, PETSC_VIEWER_STDOUT_WORLD);
```

### 4.6 Liberação de memória e finalização da execução

Para que a simulação seja finalizada, é indispensável que seja feita a liberação da memória alocada. A cada tipo de dado corresponde uma função específica para tal finalidade. A última função chamada em um código PETSc é a PetscFinalize, responsável por concluir a execução do programa:

```
VecRestoreArray(&resultado);
VecDestroy(&x); VecDestroy(&b);
MatDestroy(&A);
KSPDestroy(&ksp);
PetscFinalize();
```

## 5 Resultados Numéricos

Nas simulações de reservatório, quando se tem por meta a estimativa da pressão no poço, utiliza-se um passo de

tempo variável para melhor capturar os efeitos que se desenrolam nos instantes iniciais da produção. Portanto, um passo de tempo inicial,  $\Delta t_i$ , é multiplicado, sequencialmente, pelo fator  $F_{\Delta t}$ , a cada passo de tempo, até se atingir o passo de tempo final,  $\Delta t_{final}$  [34].

Os parâmetros gerais, as propriedades da rocha, do fluido e geométricas, para a construção do caso de referência, podem ser encontrados na Tabela 2, tendo sido considerado que a viscosidade é mantida constante.

Tabela 2 Parâmetros gerais para a rocha, o fluido e a geometria para o caso de referência

Parâmetro	Valor	Unidade
$B_0$	1,3	$\text{m}^3/\text{std m}^3$
$c_o$	$8,0 \times 10^{-7}$	$\text{kPa}^{-1}$
$c_r$	$5,0 \times 10^{-7}$	$\text{kPa}^{-1}$
$F_{\Delta t}$	1,1	-
$k_x = k_y = k$	0,05	$\mu\text{m}^2$
$L_x = L_y$	1.000	m
$L_z$	50	m
$p_{inic} = p_0$	69.000	kPa
$q_{sc}$	-1.000	$\text{m}^3/\text{dia}$
$t_{final}$	100	dia
$tol$	$1 \times 10^{-6}$	kPa
$\Delta t_i$	0,001	dia
$\Delta t_{final}$	5	dia
$\mu^0$	0,001	$\text{Pa} \cdot \text{s}$
$\rho_{ref}$	840	$\text{kg}/\text{m}^3$
$\phi_{ini} = \phi_0$	0,2	-

Por outro lado, a Tabela 3 exibe a quantidade de células, nas direções dos eixos  $x$  e  $y$ , respectivamente  $N_x$  e  $N_y$ , consideradas na proposição das diferentes malhas computacionais utilizadas na investigação do comportamento da solução numérica, ao se implementar o estudo do refinamento de malha. Ele é crucial para que se possa garantir a obtenção de soluções numéricas acuradas, além de se reduzir a estocagem numérica associada ao uso da técnica de acoplamento poço-reservatório [15, 30, 31].

Tabela 3 Malhas consideradas no estudo de refinamento

Malha	$N_x$	$N_y$
1	11	11
2	33	33
3	65	65
4	129	129
5	257	257
6	513	513
7	1.025	1.025

As simulações foram executadas em um nó do cluster do Laboratório de Computação de Alto Desempenho (LCAD) do Instituto Politécnico da Universidade do Estado do Rio de Janeiro (UERJ), cujas especificações são: Processador Intel(R) Xeon Silver 4210 2,2 GHz; Memória de 128 Gb; Instruction Set: 64 bits; Capacidade de armazenamento de 0,5 TB; Número de núcleos igual a 10;

Número de threads igual a 20; e Sistema operacional: Linux OpenSuse 15.5.

## 5.1 Refinamento de malha

Neste estudo, foram empregadas sete malhas de modo a se determinar a mais adequada tendo em vista a convergência numérica do método [2, 18]. Elas foram refinadas nas direções dos eixos  $x$  e  $y$ , mantendo-se inalteradas as propriedades físicas e as condições de operação do caso de referência.

Na Figura 1, mostra-se os valores obtidos sob o refinamento de malha, sendo perceptível, nos instantes iniciais da produção, o surgimento de patamares provocados pelo aparecimento do artefato numérico [15]. Ele é chamado de estocagem numérica, devido à semelhança qualitativa da variação de  $p_{wf}$ , em função do tempo, quando da presença da estocagem física no poço [15]. Eles já eram esperados e são consequência da escolha da técnica de acoplamento poço-reservatório aqui empregada. Ela foi introduzida por Peaceman (1978) [30], que considerou que o escoamento, na região de interação entre o poço e o reservatório, se dava em condições de regime permanente. A magnitude do artefato depende, por exemplo, da malha computacional empregada e das propriedades de fluido e de rocha.

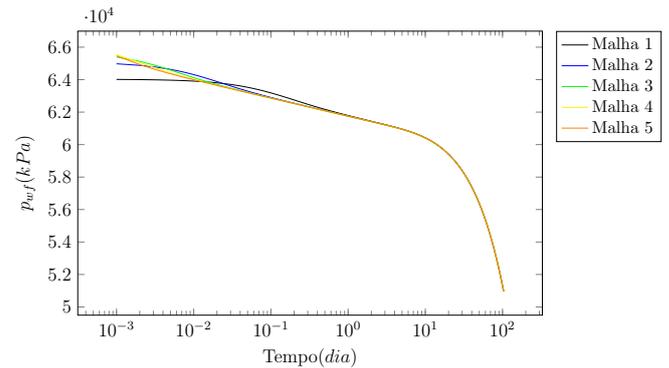


Figura 1: Pressão no poço em função do tempo de produção. Fonte: os autores.

É possível verificar, na Figura 2, a redução do efeito do artefato numérico à medida que as malhas são refinadas. Isso mostra que as soluções, a partir da Malha 5, não apresentam alterações significativas nos valores da pressão no poço e pode-se considerar que a convergência numérica foi estabelecida. Vale destacar que a mitigação dela não faz parte dos objetivos deste trabalho. Em conclusão, reteve-se a Malha 5 como sendo aquela que deve ser usada em conjunto com os demais parâmetros e valores já estipulados para o caso de referência.

Na Figura 3, a título de ilustração, é exibida a superfície que fornece a distribuição de pressão no interior do reservatório, computada já utilizando-se a Malha 5. A distribuição simétrica obtida exibe o comportamento físico esperado, tendo em vista as propriedades de rocha consideradas (campos homogêneos de permeabilidade absoluta e de porosidade inicial), com as menores

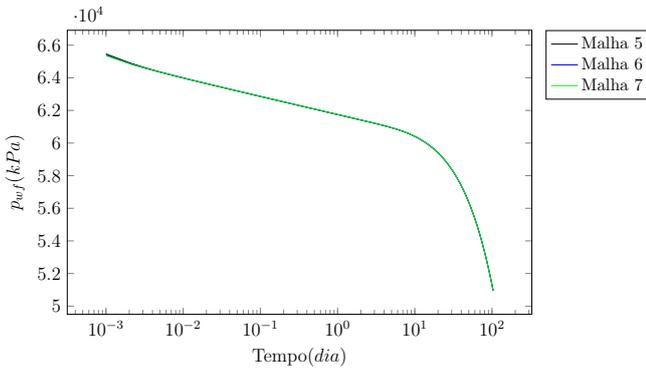


Figura 2: Pressão no poço em função do tempo de produção. Fonte: os autores.

pressões localizadas na região onde está localizado o poço produtor.

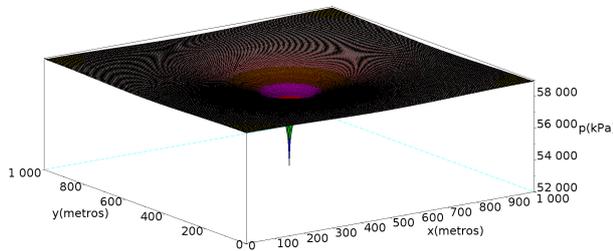


Figura 3: Distribuição do campo de pressões no reservatório após transcorridos 100 dias de simulação. Fonte: os autores.

## 5.2 Comparação de alguns dos métodos de solução e pré-condicionadores da PETSc

A versão da biblioteca PETSc 3.18 permite ao usuário escolher pelo menos um dentre os 51 métodos de resolução de sistemas de equações, disponíveis na biblioteca KSP. Em alguns casos, pode-se também trabalhar em conjunto com um dos 55 tipos de pré-condicionadores. Essa diversidade mostra a versatilidade da biblioteca e, dentre as possibilidades de escolha, foram selecionados os métodos Generalized Minimal Residual Method (GMRES), Conjugate Gradient Method (CG) e Biconjugate Gradient Method BICG, em conjunto com os pré-condicionadores de Jacobi, LU, Incomplete LU (ILU) e Successive Over-Relaxation Method (SSOR).

Inicia-se mostrando os valores da pressão no poço,  $p_{wf}$ , obtidos sem o emprego de pré-condicionadores (Figura 4). Percebe-se que é difícil se distinguir as curvas umas das outras, sendo que o erro relativo calculado foi inferior a 0,001%. No entanto, existem variações consideráveis no que se refere aos tempos de execução, sendo que o menor valor foi alcançado com o uso do método CG, demandando 76,09 s para executar a simulação do caso de referência, seguido pelo GMRES com 154,49 s e, por fim, o BICG com 148,89 s.

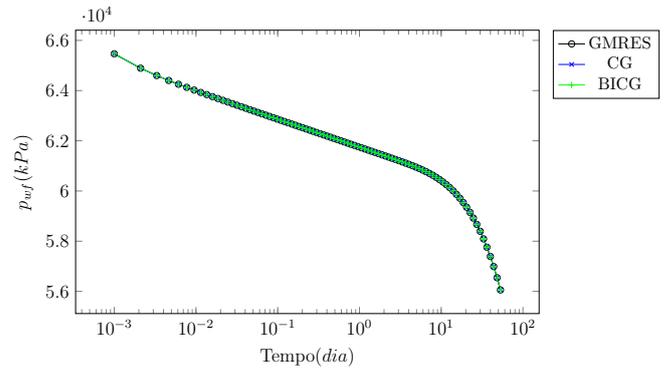


Figura 4: Pressão no poço em função do tempo de produção empregando diferentes métodos: GMRES, CG e BICG. Fonte: os autores.

Chama-se a atenção para o fato de que o método BICG não teria o seu emprego recomendado para o problema estudado, uma vez que a matriz dos coeficientes é simétrica. Como consequência, é natural que se tenha um tempo de execução aproximadamente duas vezes maior do que o do CG. No entanto, ele foi também utilizado no intuito de se validar os resultados.

Na Figura 5, tem-se as curvas de pressão determinadas com o método dos Gradientes Conjugados. Entretanto, agora, fez-se uso dos pré-condicionadores. Pode-se afirmar, uma vez mais, que as curvas encontram-se sobrepostas para os casos calculados sem ( $CG_{ref}$ ) e com os pré-condicionadores Jacobi, SSOR, ILU, LU.

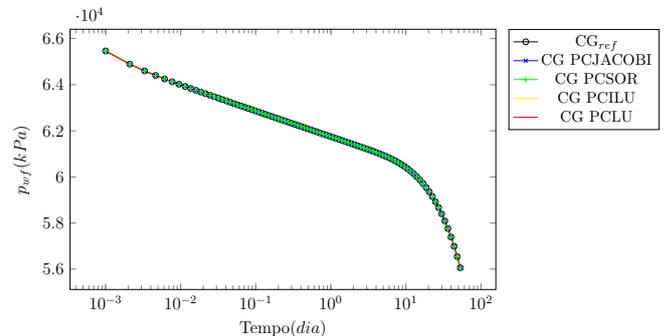


Figura 5: Pressão no poço em função do tempo de produção empregando o método CG, sem ( $CG_{ref}$ ) e com pré-condicionadores. Fonte: os autores.

A Tabela 4 contém os valores dos tempos de execução quando da utilização dos pré-condicionadores. O uso do pré-condicionador SSOR conseguiu melhorar o tempo de simulação em 6,57%. No entanto, essa conclusão não deve servir como base para uma generalização, pois a eficiência deles depende das propriedades da matriz dos coeficientes (dos seus autovalores).

## 5.3 Variação de algumas das propriedades físicas

A alteração de alguns dos valores das propriedades físicas pode ajudar a se constatar se os resultados, numericamente calculados, estão de acordo com aqueles amplamente conhecidos na literatura.

Tabela 4 Tempo de execução - CG

Método	Pré-condicionador	Tempo (s)
	—	76,09
CG	Jacobi	101,02
	LU	746,49
	ILU	73,8
	SSOR	71,0

Por exemplo, na Figura 6 encontram-se os correspondentes à diminuição e à elevação dos valores da permeabilidade absoluta, tomando-se como ponto de partida o caso de referência. É sabido, de acordo com a lei de Darcy, que a velocidade de escoamento, de um fluido no interior de um meio poroso, depende diretamente da permeabilidade absoluta e inversamente da viscosidade. Além disso, averigua-se que para as menores valores da permeabilidade absoluta obtém-se as maiores variações de pressão, devido ao fato do aumento da resistência ao escoamento através do meio poroso [18].

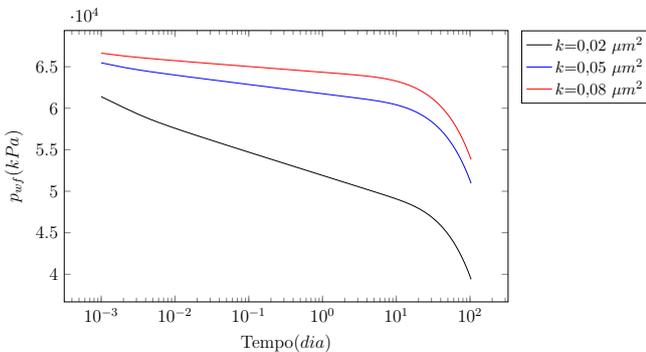


Figura 6: Pressão no poço em função do tempo de produção: variação da permeabilidade. Fonte: os autores.

Por outro lado, na Figura 7, tem-se os resultados das simulações quando aumenta-se e diminui-se os valores da porosidade do caso de referência. Como é de conhecimento geral, a porosidade é uma propriedade relacionada à capacidade de armazenamento do volume de fluido no reservatório. Da figura, pode-se atestar que para os maiores valores de porosidade correspondem as menores quedas de pressão, em função do aumento do volume de fluido estocado nos poros. A porosidade também influencia na capacidade do meio poroso em transportar o fluido. Quanto menor o seu valor, mais rápido se dará o seu deslocamento e, dessa forma, os efeitos de fronteira se farão sentir mais cedo. Tal fato pode ser observado na mudança, mais acentuada, no perfil das curvas de pressão para os tempos tardios [32]. Portanto, o efeito da sua variação é inverso ao da permeabilidade absoluta.

Embora o foco tenha sido o entendimento da física do problema, investigou-se também se essas mudanças nas propriedades poderiam afetar o tempo de execução do simulador numérico.

Na Tabela 5, encontram-se os tempos de simulação quando da variação da permeabilidade absoluta. Ela está

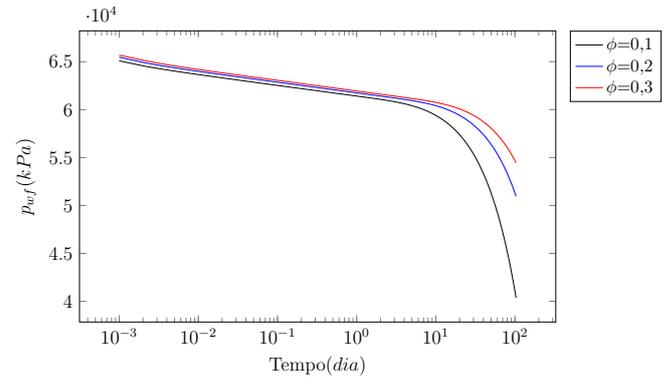


Figura 7: Pressão no poço em função do tempo de produção: variação da porosidade. Fonte: os autores.

diretamente relacionada com os valores das transmissibilidades, que aparecem nas diagonais principais da matriz dos coeficientes, alterando a dominância da diagonal principal, influenciando, assim, no número de iterações para que se obtenha a convergência e, com isso, o tempo de execução. Vê-se que os respectivos tempos aumentam conforme os seus valores crescem.

Tabela 5 Tempo de execução variando-se os valores da permeabilidade

Permeabilidade	Tempo (s)
$k= 0,02 \mu m^2$	63,03
$k= 0,05 \mu m^2$	71,04
$k= 0,08 \mu m^2$	84,15

Por sua vez, a mudança dos valores da porosidade leva a modificações apenas na diagonal principal, com a consequente alteração na sua dominância. Portanto, tem-se uma redução do tempo de execução à medida que os seus valores são aumentados, vide a Tabela 6.

Tabela 6 Tempo de execução variando-se os valores da porosidade

Porosidade	Tempo (s)
$\phi=0,1$	92,32
$\phi=0,2$	71,04
$\phi=0,3$	68,46

#### 5.4 Escoamento em um meio heterogêneo

Finalmente, incluiu-se uma discussão tratando do escoamento em meios heterogêneos. As heterogeneidades aumentam a dificuldade de obtenção das soluções e introduzem complicações adicionais quando das suas implementações. Entretanto, esse é um caminho a ser percorrido, a fim de que se possa tratar de problemas mais próximos dos encontrados na realidade. Com o objetivo de se verificar o comportamento da solução numérica, no caso de um meio heterogêneo, alterou-se a permeabilidade absoluta,  $k_{fx}$ , em uma dada região (faixa) do reservatório (região em vermelho apresentada na Figura 8),

com o seu comprimento perfazendo um total de 61 células. Ela foi posicionada a uma distância de 3 células do poço produtor.

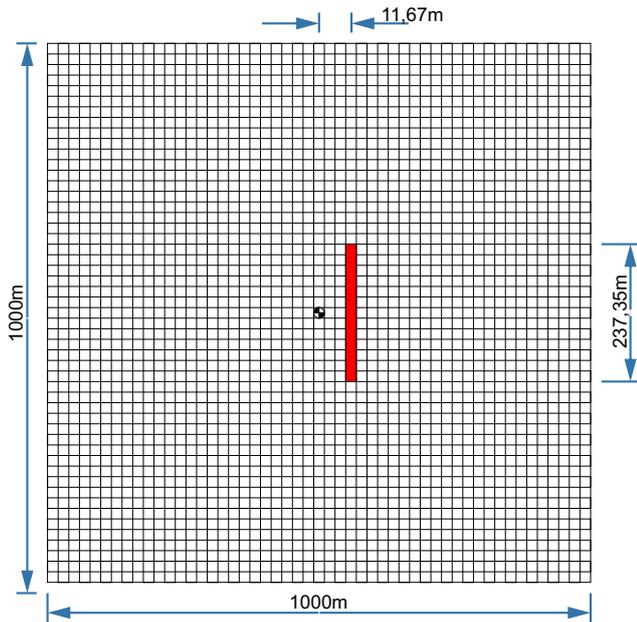


Figura 8: Representação do meio heterogêneo. Fonte: os autores.

Na Figura 9, são mostradas as curvas de pressão no poço quando do escoamento nos meios homogêneo ( $k_{fx} = k$ ) e heterogêneo. Elas estão, qualitativamente, de acordo com a física do escoamento, com as maiores pressões no poço estando relacionadas à existência de uma região de maior permeabilidade. A faixa do reservatório, contendo os maiores valores da permeabilidade absoluta, se comporta como um canal que oferece uma menor resistência ao escoamento e, também, é responsável pelo aumento da permeabilidade equivalente do sistema compreendendo as diferentes regiões do reservatório. A inclinação da curva da pressão no poço versus o tempo, no gráfico especializado, também é uma função do valor da permeabilidade, como pode ser observado na figura.

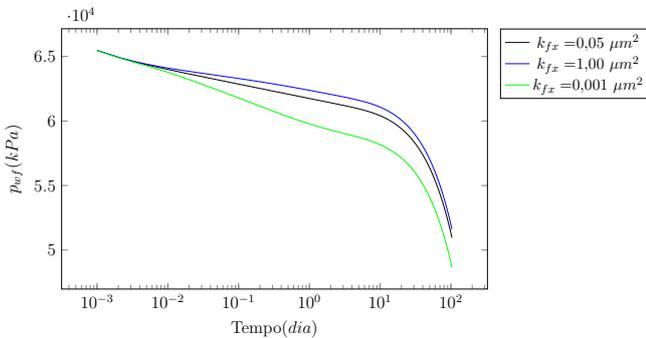


Figura 9: Pressão no poço em função do tempo de produção: variação de  $k_{fx}$ . Fonte: os autores.

Verifica-se, vide a Tabela 7, que a modificação do campo de permeabilidade absoluta, na região especifi-

cada, ou seja, a inclusão de uma heterogeneidade no domínio do problema físico-matemático, implicou o maior número de iterações para que a convergência fosse atin- gida, influenciando, assim, diretamente no tempo de execução.

Tabela 7 Tempo de execução variando-se os valores de  $k_{fx}$  da faixa heterogênea

Meio	$k_{fx}$	Tempo (s)
Homogêneo	$0,05 \mu m^2$	71,04
Heterogêneo	$0,001 \mu m^2$	88,41
Heterogêneo	$1,0 \mu m^2$	101,15

## 6 Conclusões

O uso de bibliotecas científicas, como a PETSc, é uma prática moderna no campo da modelagem computacio- nal. Trata-se de uma ferramenta versátil, que pode ser utilizada nas simulações de escoamentos em reservató- rios de petróleo. Aliada ao seu emprego, encontra-se a análise cuidadosa da escolha do método de solução de sistemas de equações, de modo a possibilitar a redução do tempo de simulação mantendo, concomitantemente, a confiabilidade dos resultados. Assim sendo, a utilização dessas bibliotecas pode ser um diferencial no desenvol- vimento de simuladores de reservatórios computacional- mente mais eficientes.

Os resultados que aqui foram apresentados, em con- junto com os procedimentos realizados visando à viabi- lização do uso da biblioteca PETSc, foram as principais contribuições deste trabalho.

Portanto, é possível concluir que a PETSc é uma fer- ramenta indicada para o seu uso na simulação de reser- vatórios, favorecendo a determinação da melhor estraté- gia de solução a ser empregada e, com isso, viabilizando a otimização do simulador. Aos pesquisadores interessa- dos, a biblioteca TAO pode contribuir para que se al- cance um maior desempenho computacional a partir do simulador previamente desenvolvido pelo programador, criado para a resolução de um dado problema específico.

Por fim, verificou-se que todos os resultados obtidos apresentaram, qualitativamente, o comportamento físico esperado com base naqueles disponíveis em outros tra- balhos.



## Referências

- [1] N.S. Al-Mohannadi. Simulation of Horizontal Well Tests. PhD thesis, Colorado School of Mines, Gol- den, USA, 2004.

- [2] K. Aziz and A. Settari. Petroleum Reservoir Simulation. Elsevier Applied Science, New York, USA, 1990.
- [3] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussovitch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, and J. Zhang. PETSc Web page, 2024.
- [4] S. Bhowmick, D. Kaushik, L. McInnes, B. Norris, and P. Raghavan. Parallel adaptive solvers in compressible PETSc-FUN3D simulations. In Anil Deane, Akin Ecer, James McDonough, Nobuyuki Satofuka, Gunther Brenner, David R. Emerson, Jacques Periaux, and Damien Tromeur-Dervout, editors, Parallel Computational Fluid Dynamics 2005, pages 277–284. Elsevier, Amsterdam, 2006.
- [5] D. Boffi, F. Credali, L. Gastaldi, and S. Scacchi. A parallel solver for fluid–structure interaction problems with Lagrange multiplier. Mathematics and Computers in Simulation, 220:406–424, 2024.
- [6] H. Bonart, S. Fillinger, E. Esche, G. Wozny, and J.-U. Repke. Source code generation for parallelized simulations of large-scale nonlinear equation systems on a supercomputer using MOSAIC, PETSc, and ADOL-C. In Antonio Espuña, Moisès Graells, and Luis Puigjaner, editors, 27th European Symposium on Computer Aided Process Engineering, volume 40 of Computer Aided Chemical Engineering, pages 2083–2088. Elsevier, 2017.
- [7] P. D. Boom, A. Seepujak, O. Kosmas, L. Margetts, and A. Jivkov. Parallelized discrete exterior calculus for three-dimensional elliptic problems. Computer Physics Communications, 279:108456, 2022.
- [8] A. P. Canal. Paralelização de métodos de resolução de sistemas lineares esparsos com o deck em um cluster de PCs. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- [9] L. Carracciolo, L. D’Amore, and A. Murli. Towards a parallel component for imaging in PETSc programming environment: a case study in 3-D echocardiography. Parallel Computing, 32(1):67–83, 2006.
- [10] E. Centofanti and S. Scacchi. A comparison of algebraic multigrid bidomain solvers on hybrid CPU–GPU architectures. Computer Methods in Applied Mechanics and Engineering, 423:116875, 2024.
- [11] L. E. Clarke and G. Krishnamoorthy. Preconditioning strategies to accelerate the convergence of iterative methods in multiphase flow simulations. Mathematics and Computers in Simulation, 165:200–222, 2019.
- [12] A. Croucher, M. O’Sullivan, J. O’Sullivan, A. Yeh, J. Burnnell, and W. Kissiling. Waiwera: A parallel open-source geothermal flow simulator. Computers & Geosciences, 141:104529, 2020.
- [13] S. Cuomo, A. Galletti, G. Giunta, and L. Marcelino. Toward a multi-level parallel framework on GPU cluster with PETSc-CUDA for PDE-based optical flow computation. Procedia Computer Science, 51:170–179, 2015.
- [14] H. Darcy. Les fontaines publiques de la ville de Dijon. Exposition et application des principes à suivre et des formules à employer dans les questions de distribution d’eau. Victor Dalmont, 1856.
- [15] G. de Souza. Acoplamento Poço-reservatório na Simulação Numérica de Reservatórios de Gás. PhD thesis, Universidade Estadual do Norte Fluminense, Macaé, Brasil, 2013.
- [16] D. Demidov. AMGCL-A C++ library for efficient solution of large sparse linear systems. Software Impacts, 6:100037, 2020.
- [17] J. Dyrdaahl. Thermal flow in fractured porous media and operator splitting. Master’s thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2014.
- [18] T. Ertekin, J. H. Abou-Kassem, and G. R. King. Basic Applied Reservoir Simulation. Society of Petroleum Engineers, Richardson, USA, 2001.
- [19] J.-M. Gratien. A robust and scalable multi-level domain decomposition preconditioner for multi-core architecture with large number of cores. Journal of Computational and Applied Mathematics, 373:112614, 2020.
- [20] X. Guo, B. D. Rogers, S. Lind, and P. K. Stansby. New massively parallel scheme for Incompressible Smoothed Particle Hydrodynamics (ISPH) for highly nonlinear and distorted flow. Computer Physics Communications, 233:16–28, 2018.
- [21] P. D. Hovland and L. C. McInnes. Parallel simulation of compressible flow using automatic differentiation and PETSc. Parallel Computing, 27(4):503–519, 2001.
- [22] P. Jolivet, J. E. Roman, and S. Zampini. KSPHPDDM and PCHPDDM: Extending PETSc with advanced Krylov methods and robust multilevel overlapping Schwarz preconditioners. Computers & Mathematics with Applications, 84:277–295, 2021.

- [23] Y. Jung, George S. H. Pau, S. Finsterle, and R. M. Pollyea. TOUGH3: A new efficient version of the TOUGH suite of multiphase flow and transport simulators. *Computers & Geosciences*, 108:2–7, 2017.
- [24] S. Kang, A. Dener, A. Hamilton, H. Zhang, E. M. Constantinescu, and R. L. Jacob. Multirate partitioned Runge–Kutta methods for coupled Navier–Stokes equations. *Computers & Fluids*, 264:105964, 2023.
- [25] R. F. Katz, M. G. Knepley, B. Smith, M. Spiegelman, and E. T. Coon. Numerical simulation of geodynamic processes with the portable extensible toolkit for scientific computation. *Physics of the Earth and Planetary Interiors*, 163(1):52–68, 2007.
- [26] T. Kempe, A. Aguilera, W. Nagel, and J. Fröhlich. Performance of a projection method for incompressible flows on heterogeneous hardware. *Computers & Fluids*, 121:37–43, 2015.
- [27] Y. Ma and Z. Chen. Parallel computation for reservoir thermal simulation of multicomponent and multiphase fluid flow. *Journal of Computational Physics*, 201:224–237, 2004.
- [28] B. Mantravadi, P. Jagad, and R. Samtaney. A hybrid discrete exterior calculus and finite difference method for Boussinesq convection in spherical shells. *Journal of Computational Physics*, 491:112397, 2023.
- [29] R. T. Mills, M. F. Adams, S. Balay, J. Brown, A. Dener, M. Knepley, S. E. Kruger, H. Morgan, T. Munson, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and Junchao Zhang. Toward performance-portable PETSc for GPU-based exascale systems. *Parallel Computing*, 108:102831, 2021.
- [30] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation. *Society of Petroleum Engineers Journal*, 18(3):183–194, 1978.
- [31] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *Society of Petroleum Engineers Journal*, 23(3):531–543, 1983.
- [32] A. J. Rosa, R. S. Carvalho, and J. A. D. Xavier. *Engenharia de reservatórios de petróleo*. Interciência, 2006.
- [33] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, EUA, 2003.
- [34] F. B. Vasconcellos. Utilização da biblioteca numérica PETSc na solução de sistemas algébricos na simulação de reservatórios. Master’s thesis, Universidade do Estado do Rio de Janeiro, Nova Friburgo, Brasil, 2024.
- [35] W. Wang, O. Kolditz, and T. Nagel. A parallel FEM scheme for the simulation of large scale thermochemical energy storage with complex geometries using PETSc routines. *Energy Procedia*, 75:2080–2086, 2015.
- [36] L. F. Werneck. Implementação paralelizada de métodos de resolução de sistemas algébricos na simulação de reservatórios de gás. Master’s thesis, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2016.
- [37] L. F. Werneck, M. M. Freitas, G. Souza, L. F. C. Jatobá, and H. P. A. Souto. An OpenMP parallel implementation using a coprocessor for numerical simulation of oil reservoirs. *Computational and Applied Mathematics*, 38:1–30, 2019.
- [38] S. Whitaker. *The Method of Volume Average. Theory and Applications of Transport in Porous Media*. Kluwer Academic Publishers, 1999.
- [39] H. Yang, S. Sun, Y. Li, and C. Yang. A scalable fully implicit framework for reservoir simulation on parallel computers. *Computer Methods in Applied Mechanics and Engineering*, 330:334–350, 2018.
- [40] H. Yang, S. Sun, Y. Li, and C. Yang. Parallel reservoir simulators for fully implicit complementarity formulation of multicomponent compressible flows. *Computer Physics Communications*, 244:2–12, 2019.
- [41] H. Zhang and E. M. Constantinescu. Optimal checkpointing for adjoint multistage time-stepping schemes. *Journal of Computational Science*, 66:101913, 2023.
- [42] M. Zhang, H. Yang, S. Wu, and S. Sun. Parallel multilevel domain decomposition preconditioners for monolithic solution of non-isothermal flow in reservoir simulation. *Computers & Fluids*, 232:105183, 2022.
- [43] Y. Zhao, L. Zhang, and B. Shan. Mathematical model of fractured horizontal well in shale gas reservoir with rectangular stimulated reservoir volume. *Journal of Natural Gas Science and Engineering*, 59:67–79, 2018.