revista principia
Vol. 61, N. 3
REVISTA DE DIVULGAÇÃO CIENTÍFICA E TECNOLÓGICA DO INSTITUTO FEDERAL DE CIÊNCIA, EDUCAÇÃO E TECNOLOGIA DA PARAÍBA

DOI: *http://dx.doi.org/10.18265/1517-0306a2022id7140*
ARTIGO ORIGINAL

SUBMETIDO 02/08/2022
APROVADO 02/12/2022
PUBLICADO ON-LINE 05/12/2022
PUBLICADO 10/07/2024
EDITOR ASSOCIADO
José Carlos Lima Júnior

Simulação numérica do escoamento em reservatórios de gás natural não convencionais utilizando o processamento em paralelo (OpenMP)

Grazione de Souza [1]*
Breno Luís Dutra [2]
Lucas Barros Lima [3]
Helio Pedro Amaral Souto [4]

[1] gsouza@iprj.uerj.br
[2] breno.ld@outlook.com
[3] lucas.barros@iprj.uerj.br
[4] helio@iprj.uerj.br

Universidade do Estado do Rio de Janeiro (UERJ), Brasil

*Autor para correspondência.

RESUMO: Este trabalho é dedicado à simulação numérica do escoamento de gás natural em reservatórios não convencionais na presença dos efeitos de adsorção e escorregamento. Uma formulação numérica totalmente implícita, utilizando o método *Control Volume Finite Difference* (CVFD), foi adotada tendo em vista a resolução numérica da equação diferencial parcial não linear que governa o escoamento da fase gás. O método dos Gradientes Conjugados foi utilizado na solução do sistema de equações algébricas obtidas a partir do processo de discretização. A fim de melhorar a eficiência computacional, a *Application Programming Interface* (API) *Open Multi-Processing* (OpenMP) foi empregada na paralelização do método dos Gradientes Conjugados. Uma análise de sensibilidade foi realizada e mostrou-se que foi possível alcançar, para as simulações consideradas, valores de *speedup* superiores a 5.

Palavras-chave: gás natural; método dos Gradientes Conjugados; OpenMP; paralelização; simulação de reservatórios.

Numerical simulation of flow in unconventional natural gas reservoirs using parallel processing (OpenMP)

ABSTRACT: *This article is dedicated to the numerical simulation of natural gas flow in unconventional reservoirs in the presence of adsorption and slippage effects. We adopted a fully implicit numerical formulation, using the Control Volume Finite Difference (CVFD) method, to resolve the non-linear partial differential equation governing the flow of gas phase. We used the Conjugated Gradients method to solve the system of algebraic equations obtained from the discretization process. To improve the computational efficiency, we employed the Application Programming Interface (API) Open Multi-Processing (OpenMP) to*

ISSN (on-line): 2447-9187

parallelize the Conjugate Gradients method. A sensitivity analysis was carried out, and it was shown that it was possible to achieve, for the simulations considered, speedup values greater than 5.

Keywords: *conjugate gradient method; natural gas; OpenMP; parallelization; reservoir simulation.*

1 Introdução

A indústria de óleo e gás natural utiliza as simulações numéricas para estudar o comportamento dos reservatórios portadores de hidrocarbonetos desde a década de 1950. Elas permitem, por exemplo, testar várias estratégias de recuperação de hidrocarbonetos de modo a escolher o melhor plano de desenvolvimento para o campo produtor. Portanto, mediante o uso das ferramentas da computação de alto desempenho, pode-se otimizar o processo decisório no gerenciamento da produção dos reservatórios. Nesse contexto, emprega-se a *Application Programming Interface* (API) *Open Multi-Processing* – OpenMP (Carvalho *et al*., 2020; Redondo; Rubio; Valero, 2018; Salles *et al*., 2018; Werneck *et al*., 2019) na simulação de reservatórios não convencionais de gás natural.

A simulação numérica de reservatórios é uma área da engenharia de petróleo dedicada ao uso de métodos numéricos na obtenção das soluções das equações governantes dos escoamentos nas jazidas de hidrocarbonetos. Trata-se de uma área multidisciplinar que envolve conhecimentos, por exemplo, da geologia, do escoamento em meios porosos, de equações diferenciais parciais, de métodos numéricos e da programação computacional. O seu principal objetivo é viabilizar a escolha da melhor estratégia de produção, maximizando o fator de recuperação de hidrocarbonetos (Chen; Huan; Ma, 2006; Ertekin; Abou-Kassem; King, 2001).

Tem-se constatado, em muitas pesquisas científicas, *softwares* na área de engenharia como ferramenta padrão na indústria do petróleo e gás natural e as suas aplicações na simulação numérica de problemas de escoamento em reservatórios não convencionais (Chen *et al*., 2020; Li *et al*., 2015; Wang *et al*., 2017). Nesses reservatórios, técnicas especiais de recuperação que fogem às práticas convencionais (poço vertical, por exemplo) são necessárias para se ter uma produção economicamente viável. Entre os reservatórios não convencionais, existem os de gás de folhelho (*shale gas*), os de óleo de folhelho (*shale oil*), os de gás aderido a veios de carvão (*coalbed methane*), os de óleo pesado (*heavy oil*) e os depósitos de hidratos de gás (*gas-hydrate deposits*). No caso dos reservatórios do tipo folhelho, considerados neste trabalho, existem estudos em andamento destinados a viabilizar o aproveitamento desses recursos no Brasil (Aba; Parente; Santos, 2022; Lenhard; Andersen; Coimbra-Araújo, 2018), contribuindo para a manutenção do perfil diversificado da matriz energética brasileira no futuro.

Classifica-se como computação de alto desempenho a utilização de computadores em processamento de dados e cálculos complexos em altas velocidades. Utilizando, por exemplo, um computador pessoal com um processador de 3 GHz, pode-se realizar cerca de 3 bilhões de cálculos por segundo. Algumas soluções em *High Performance Computing* (HPC) podem levar a quadrilhões de cálculos por segundo. Uma das mais conhecidas soluções em HPC é o supercomputador, o qual contém milhares de nós que trabalham juntos para completar uma ou mais tarefas, no que se chama de processamento em paralelo. Hoje, empresas do ramo de tecnologia da informação conseguem processar dados na ordem

de petabytes usando HPC, *clusters* (conjunto de diversos computadores trabalhando em conjunto) e placas de vídeo (Mohammed *et al*., 2020; Rousset *et al*., 2016; Wright, 2019).

Entre as técnicas de HPC, pode-se citar a *Message Passing Interface* (MPI), a *Compute Unified Device Architecture* (CUDA), a *Open Accelerators* (OpenACC), a *Open Computing Language* (OpenCL) e o OpenMP. A MPI (Wu *et al*., 2013) possibilita a solução de um problema usando um conjunto de processadores ou nós de um *cluster*. Por outro lado, a CUDA (Jimale; Ridzuan; Zainon, 2019) é uma extensão da linguagem C, na qual se utiliza as unidades de processamento gráfico (*Graphics Processing Unit*, GPU) para a computação paralela. A API OpenACC (Wolfe *et al*., 2018) também permite o uso das placas gráficas para acelerar o processamento, mediante o uso de diretivas de compilação, além de também poder utilizar os recursos da *Central Processing Unit* (CPU). No caso da OpenCL (Pennycook *et al*., 2013), tem-se um padrão de programação para a execução em paralelo em plataformas heterogêneas. Por fim, com a API OpenMP (Werneck *et al*., 2019) é possível usar a memória de forma distribuída, permitindo que várias *threads* (linhas de execução) atuem na execução de uma tarefa. Sugere-se aos leitores o trabalho de Aldinucci *et al*. (2021) sobre a utilização e comparação das técnicas MPI, OpenMP, OpenACC e CUDA.

Um componente fundamental no OpenMP são as *threads*. Segundo Akhter e Roberts (2006), elas são processos que se autodividem em diversas tarefas que são executadas de forma independente. O OpenMP permite paralelizar processos usando a memória compartilhada (Chapman; Jost; Van Der Pas, 2007). Essa API atualmente oferece suporte às linguagens de programação C/C++, Fortran e Python. A paralelização se dá mediante o uso de diretivas (comandos que instruem como o compilador deve processar o código). Com elas, é possível, por exemplo, estabelecer o número de *threads* que o processador utilizará, as variáveis que se deseja compartilhar entre as *threads* e quais trechos do código se deseja paralelizar.

No restante do artigo, a seção 2 é dedicada à modelagem do escoamento de gás natural em meios porosos, abordando as propriedades do fluido e da rocha, a equação diferencial parcial governante utilizada para a determinação da pressão, além das condições iniciais e de contorno. Em seguida, na seção 3, discorre-se sobre a avaliação numérica da pressão da fase gasosa, incluindo o processo de discretização, a construção do sistema de equações algébricas não lineares e a paralelização de parte do código numérico utilizando a API OpenMP. A seção 4 é voltada para a análise e a discussão dos resultados obtidos, sob o ponto de vista do desempenho computacional e dos efeitos físicos advindos do escorregamento e da adsorção. Finalmente, a seção 5 contém as conclusões alcançadas e algumas perspectivas futuras.

2 Modelagem do escoamento de gás em meios porosos

A capacidade de armazenamento de hidrocarbonetos em um reservatório depende diretamente da sua porosidade. A porosidade total é aquela que inclui os poros interconectados e não interconectados. A porosidade efetiva (Equação 1), considerada apenas como porosidade daqui em diante, contabiliza apenas os poros interconectados (Dandekar, 2013). Considera-se que a rocha possui uma compressibilidade pequena e constante, de forma que (Ertekin; Abou-Kassem; King, 2001):

$$\phi = \phi^0 \left[1 + c_\phi \left(p - p^0 \right) \right] \tag{1}$$

onde ϕ é a porosidade, p é a pressão, ϕ^0 é a porosidade na condição de referência para a pressão p^0, e c_ϕ é o coeficiente de compressibilidade da rocha.

2.1 Adsorção e escorregamento

Nas últimas duas décadas, houve um aumento do interesse sobre os reservatórios do tipo *shale gas*, devido aos avanços tecnológicos que permitiram o melhor aproveitamento econômico dessas jazidas, sendo que nesses reservatórios, tipicamente, estão presentes os efeitos de escorregamento e/ou de adsorção. O gás em reservatórios pode estar armazenado de três maneiras: na forma de gás livre nos poros e nas fraturas, dissolvido no óleo e na água e adsorvido na superfície dos poros. Um modelo de adsorção fornece a quantidade de gás que se encontra adsorvido na superfície sólida a uma determinada pressão e temperatura. Para o caso isotérmico, essa quantidade depende apenas da pressão. Um desses modelos é o da Isoterma de Langmuir (Li *et al*., 2016), expresso pela Equação 2:

$$V_{ads} = \frac{V_L p}{p_L + p} \tag{2}$$

onde V_{ads} é o volume de gás adsorvido, V_L é o volume máximo de adsorção (volume de Langmuir) e P_L é a pressão quando a adsorção atinge metade da sua máxima capacidade (pressão de Langmuir).

A permeabilidade absoluta fornece uma medida da resistência que o meio impõe ao escoamento do fluido no seu interior. Em certos casos de escoamentos de gás em meios porosos, utiliza-se uma permeabilidade aparente de modo a se levar em conta o efeito Klinkenberg (Klinkenberg, 1941), ou seja, o escorregamento do gás na superfície dos poros. O efeito Klinkenberg acontece quando o livre caminho médio do gás não pode ser desprezado quando comparado ao raio hidráulico (Florence *et al*., 2007).

Segundo Jiang e Younis (2015), o tensor permeabilidade aparente k_a pode ser definido mediante a incorporação dos efeitos da adsorção do gás e do escorregamento, na forma da Equação 3:

$$k_a = \left(1 + \frac{4\,Kn}{1+Kn}\right)\left(1 - \frac{d_m}{R_h}\frac{p/p_L}{1+p/p_L}\right)^4 k = \left(1 + \frac{4\,Kn}{1+Kn}\right)\left(\frac{R_{eff}}{R_h}\right)^4 k \tag{3}$$

onde $Kn = \lambda / R_{eff}$ é o número de Knudsen, λ é o livre caminho médio, R_{eff} é o raio hidráulico efetivo, d_m é o diâmetro da molécula de gás, R_h é o raio hidráulico equivalente e k é o tensor de permeabilidade absoluta, considerado como sendo diagonal. Além disso, $\lambda = \mu / p\sqrt{\pi ZRT/2M}$ (Florence *et al*., 2007), onde μ é a viscosidade, Z é o fator de compressibilidade dos gases, R é a constante universal dos gases, T é a temperatura e M é a massa molecular do gás, e $R_h = 2\sqrt{2\tau}\sqrt{k_m/\phi}$, onde τ representa a tortuosidade do meio e k_m, a média geométrica das componentes principais do tensor diagonal de permeabilidade absoluta k (Chen; Huan; Ma, 2006).

2.2 Equação governante

Quando se trata de escoamento monofásico de um gás, levando em conta os efeitos de escorregamento e de adsorção, a velocidade superficial é dada pela lei de Darcy modificada, dada pela Equação 4:

$$v = \frac{-k_a}{\mu}\left(\nabla p - \rho g \nabla D\right) \tag{4}$$

onde ρ é a massa específica, g é a magnitude da aceleração da gravidade e D é a profundidade (Carvalho *et al*., 2020).

Segundo Li *et al*. (2016), a conservação de massa para o escoamento em meios porosos, incorporando efeitos da adsorção, pode ser escrita como a Equação 5:

$$\frac{\partial}{\partial t}\left(\frac{\rho_{sc}\phi}{B}\right) + \frac{\partial}{\partial t}\left[\left(\frac{\rho_s \rho_{sc} V_{ads}}{B}\right)\right] + \nabla \cdot \left(\frac{\rho_{sc} v}{B}\right) - \frac{q_{sc}\rho_{sc}}{V_b} = 0 \tag{5}$$

onde *sc* indica as condições padrão de pressão e temperatura (na superfície), $B = \rho_{sc}/\rho$ é o fator-volume-formação, q_{sc} é o termo fonte em condições padrão e V_b representa o volume total.

Agora, sabendo que ρ_{sc} é constante, assumindo que os efeitos gravitacionais podem ser desprezados e utilizando a regra da cadeia, as Equações 4 e 5 em conjunto passam a ser escritas da seguinte forma:

$$\nabla \cdot \left(\frac{k_a}{\mu B}\nabla p\right) + \frac{q_{sc}}{V_b} = \rho_s\left[\frac{1}{B}\frac{d}{dp}V_{ads} + V_{ads}\frac{d}{dp}\left(\frac{1}{B}\right)\right]\frac{\partial p}{\partial t} + \left[\frac{1}{B}\frac{d\phi}{dp} + \phi\frac{d}{dp}\left(\frac{1}{B}\right)\right]\frac{\partial p}{\partial t} \tag{6}$$

ou ainda

$$\nabla \cdot \left(\frac{k_a}{\mu B}\nabla p\right) + \frac{q_{sc}}{V_b} = \left(\Gamma_p + \Gamma_s\right)\frac{\partial p}{\partial t} \tag{7}$$

onde Γ_p, que inclui efeitos de compressibilidades da rocha e do fluido, e Γ_s, que inclui efeitos de adsorção e de compressibilidade do fluido, são, respectivamente, apresentados nas Equações 8a e 8b:

$$\Gamma_p = \frac{c_\phi \phi^0}{B} + \phi\frac{d}{dp}\left(\frac{1}{B}\right) \tag{8a}$$

$$\Gamma_s = \rho_s\left[\frac{1}{B}\frac{d}{dp}V_{ads} + V_{ads}\frac{d}{dp}\left(\frac{1}{B}\right)\right] \tag{8b}$$

2.3 Acoplamento poço-reservatório

No presente estudo, utiliza-se um modelo de acoplamento poço-reservatório na representação do poço produtor de comprimento L_{wf}, localizado no centro do plano xy (Peaceman, 1983). Portanto, a vazão de produção q_{sc} é expressa em função da pressão no poço p_{wf} e é dada pela Equação 9:

$$q_{sc} = -J_w\left(p - p_{wf}\right) \tag{9}$$

onde J_w é o índice de produtividade (diferente de zero onde houver poço). A determinação do índice de produtividade no contexto da simulação numérica de reservatórios será discutida mais adiante.

2.4 Condições auxiliares

Conforme visto, a equação governante (Equação 7) é uma equação diferencial parcial não linear cuja variável dependente é a pressão do gás. Então, para que se tenha um problema matematicamente bem posto, deve-se fornecer as condições inicial e de contorno apropriadas. Como o foco é o escoamento bidimensional em regime transiente, em um reservatório paralelepipédico, utiliza-se a condição inicial (Equação 10):

$$p\left(x, y, t=0\right) = p_{ini}\left(x, y\right) = p_{inic} \tag{10}$$

onde P_{inic} é a pressão do reservatório antes da produção ser iniciada. Como condições de contorno (Equação 11), impõe-se que não há escoamento através das fronteiras externas do reservatório:

$$\left(\frac{\partial p}{\partial x}\right)_{x=0,L_x} = \left(\frac{\partial p}{\partial y}\right)_{y=0,L_y} = 0 \tag{11}$$

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

2.5 Determinação das propriedades do gás

Concluindo esta seção, cabe destacar que as propriedades do fluido são funções da pressão, da temperatura e da sua densidade $y = M/M_{ar} = M/28{,}96$, onde M_{ar} é a massa molecular do ar. Ademais, quando da determinação das propriedades do gás natural, tomou-se como base o trabalho de Carvalho *et al*. (2020), sendo aplicadas as correlações de Dranchuk e Abou-Kassem (1975) para o cálculo do fator de compressibilidade Z, as de Lee, Gonzalez e Eakin (1966) para o cômputo da viscosidade, as correlações de Sutton (1985) para a avaliação das propriedades pseudorreduzidas do gás natural e a equação de estado para gás real $\rho = pM/ZRT$, utilizada na determinação de B.

A motivação para tratar desse caso específico se deve ao aumento da produção e do consumo de gás natural. Como pode ser visto, o modelo que descreve o escoamento de gás natural considerado é constituído por uma equação diferencial parcial não linear, o que leva a um aumento do custo computacional quando do processo da sua resolução numérica. Tal fato é agravado em função da não linearidade ser mais pronunciada devido à inclusão dos efeitos de escorregamento e de adsorção, tipicamente encontrados em reservatórios do tipo *shale gas* (Pessanha *et al*., 2020).

3 Resolução numérica

Como, em geral, não é possível obter uma solução analítica para escoamentos reais em reservatórios, a não ser que hipóteses simplificadoras sejam introduzidas, opta-se pela resolução numérica. O método escolhido para tal finalidade foi o *Control Volume Finite Difference* (CVFD), e empregou-se uma malha computacional de blocos centrados (Ertekin; Abou-Kassem; King, 2001). A solução numérica é determinada nos nós da malha, que estão localizados nos centros das células (blocos). O número total de células nas direções dos eixos x e y são n_x e n_y, nessa ordem, sendo que os índices inteiros i e j enumeram os nós centrais das células nessas respectivas direções.

Portanto, discretiza-se a Equação 7 a fim de se determinar a pressão do fluido no instante de tempo $n+1$ no nó (i, j), ou seja, pela Equação 12,

$$\left[\frac{\partial}{\partial x}\left(T_x^p\frac{\partial p}{\partial x}\right)dx+\frac{\partial}{\partial y}\left(T_y^p\frac{\partial p}{\partial y}\right)dy\right]_{i,j}^{n+1}=\left[\left(\Gamma_p+\Gamma_s\right)\frac{\partial p}{\partial t}+q_{sc}\right]_{i,j}^{n+1} \tag{12}$$

sabendo que $V_b=dxdy\,L_z$, com constante ao longo do eixo x e dx constante ao longo do eixo y, de modo que foi possível introduzir as novas variáveis $T_x^p\equiv A_x k_{a,x}/\mu B$ e $T_y^p\equiv A_y k_{a,y}/\mu B$, sendo $A_x=dy\,L_z$, $A_y=dx\,L_z$ e L_z o comprimento do reservatório na direção z.

Empregando-se uma aproximação do tipo diferença centrada, obtém-se a Equação 13 para a direção x

$$\frac{\partial}{\partial x}\left(T_x^p\frac{\partial p}{\partial x}\right)_{i,j}^{n+1}\approx\frac{1}{\Delta x_{i,j}}\left[\left(T_x^p\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j}-\left(T_x^p\frac{\partial p}{\partial x}\right)_{i-\frac{1}{2},j}\right]^{n+1} \tag{13}$$

onde $\Delta x_{i,j}$ corresponde ao espaçamento da malha na direção x na célula (i,j).

Na Equação 13, sabe-se que as derivadas parciais da pressão em relação a x, nas faces das células $i\pm1/2$ e $j\pm1/2$, também podem ser aproximadas por diferenças centradas (Equação 14):

$$\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j}^{n+1}\approx\frac{p_{i+1,j}^{n+1}-p_{i,j}^{n+1}}{\Delta x_{i+\frac{1}{2},j}},\quad\left|\quad\left(\frac{\partial p}{\partial x}\right)_{i-\frac{1}{2},j}^{n+1}\approx\frac{p_{i,j}^{n+1}-p_{i-1,j}^{n+1}}{\Delta x_{i-\frac{1}{2},j}}\right. \tag{14}$$

Expressões análogas podem ser obtidas para a discretização dos termos associados à direção y. Em se tratando do termo transiente, utiliza-se a expansão conservativa proposta por Li *et al*. (2016), de forma que:

$$\Gamma_p^{n+1} = \left(V_b\right)_{i,j}\left[\frac{1}{B^n}\frac{\partial \phi}{\partial p} + \phi^{n+1}\frac{\partial}{\partial p}\left(\frac{1}{B}\right)\right] \tag{15}$$

$$\Gamma_s^{n+1} = \left(V_b\right)_{i,j}\left[\rho_s V_{ads}^n \frac{\partial}{\partial p}\left(\frac{1}{B}\right) + \frac{\rho_s}{B^{n+1}}\frac{\partial}{\partial p}V_{ads}\right] \tag{16}$$

onde Δt é o incremento de tempo, e optou-se por aproximar o termo da derivada parcial em relação ao tempo por um esquema do tipo diferença recuada (Equação 17),

$$\left(\frac{\partial p}{\partial t}\right)_{i,j}^{n+1} \approx \frac{p_{i,j}^{n+1} - p_{i,j}^{n}}{\Delta t} \tag{17}$$

Finalmente, obtém-se a Equação 18 para a formulação totalmente implícita para a equação governante discretizada:

$$\begin{aligned} &T_x^r\Big|_{i+1/2,j}^{n+1}\left(p_{i+1,j}^{n+1} - p_{i,j}^{n+1}\right) + T_x^r\Big|_{i-1/2,j}^{n+1}\left(p_{i,j}^{n+1} - p_{i-1,j}^{n+1}\right) + T_y^r\Big|_{i,j+1/2}^{n+1}\left(p_{i,j+1}^{n+1} - p_{i,j}^{n+1}\right) \\ &+ T_y^r\Big|_{i,j-1/2}^{n+1}\left(p_{i,j}^{n+1} - p_{i,j-1}^{n+1}\right) = \frac{\left(\Gamma_p + \Gamma_s\right)_{i,j}^{n+1}}{\Delta t}\left(p_{i,j}^{n+1} - p_{i,j}^{n}\right) + \left(q_{sc}\right)_{i,j}^{n+1} \end{aligned} \tag{18}$$

Na Equação 18, foram introduzidas as novas variáveis T^r_x e T^r_y, chamadas de transmissibilidades, que são definidas como $T_x^r \equiv A_x k_{a,x}/\mu B \Delta x$ e $T_y^r \equiv A_y k_{a,y}/\mu B \Delta y$, respectivamente. Como as transmissibilidades devem ser avaliadas nas faces das células, elas são calculadas a partir de uma média harmônica empregando os seus valores conhecidos nos nós vizinhos. Quanto às propriedades do gás, elas são determinadas via média aritmética (Ertekin; Abou-Kassem; King, 2001).

3.1 Acoplamento poço-reservatório

A versão discretizada da equação do modelo de acoplamento, Equação 19, é dada por (Peaceman, 1983):

$$\left(q_{sc}\right)_{i,j}^{n+1} = -\left(J_w\right)_{i,j}^{n+1}\left[p_{i,j}^{n+1} - \left(p_{wf}\right)_{i,j}^{n+1}\right] \tag{19}$$

e o índice de produtividade é determinado, para um poço vertical, por intermédio da Equação 20:

$$\left(J_w\right)_{i,j}^{n+1}=\left\{2\pi\sqrt{k_{a,x}k_{a,y}}L_z\left[B\mu ln\left(\frac{r_{eq}}{r_w}\right)\right]^{-1}\right\}_{i,j}^{n+1} \tag{20}$$

onde r_w é o raio do poço e r_{eq} é o raio equivalente adaptado, dado pela Equação 21:

$$r_{eq}=0{,}28\left[\sqrt{\sqrt{\frac{k_{a,y}}{k_{a,x}}}\left(\Delta x\right)^2+\sqrt{\frac{k_{a,x}}{k_{a,y}}}\left(\Delta y\right)^2}\right]_{i,j}\left[\sqrt[4]{\frac{k_{a,y}}{k_{a,x}}}+\sqrt[4]{\frac{k_{a,x}}{k_{a,y}}}\right]_{i,j}^{-1} \tag{21}$$

3.2 Resolução numérica do sistema algébrico

A Equação 18 representa um sistema algébrico não linear. Aqui, optou-se por utilizar uma técnica de linearização via método de Picard (Nick *et al*., 2013). Então, a sua forma discretizada linearizada é dada por:

$$\begin{aligned}&T_y^r\Big|_{i,j-1/2}^{v,n+1}p_{i,j-1}^{v+1,n+1}+T_x^r\Big|_{i-1/2,j}^{v,n+1}p_{i-1,j}^{v+1,n+1}\\&+\left[T_y^r\Big|_{i,j-1/2}^{v,n+1}+T_x^r\Big|_{i-1/2,j}^{v,n+1}+\frac{\left(\Gamma_p+\Gamma_s\right)_{i,j}^{v,n+1}}{\Delta t}+T_x^r\Big|_{i+1/2,j}^{v,n+1}+T_y^r\Big|_{i,j+1/2}^{v,n+1}\right]p_{i,j}^{v+1,n+1}\\&+T_x^r\Big|_{i+1/2,j}^{v,n+1}p_{i+1,j}^{v+1,n+1}+T_y^r\Big|_{i,j+1/2}^{v,n+1}p_{i,j+1}^{v+1,n+1}=\frac{\left(\Gamma_p+\Gamma_s\right)_{i,j}^{v,n+1}}{\Delta t}p_{i,j}^{n}+\left(q_{sc}\right)_{i,j}^{n+1}\end{aligned} \tag{22}$$

onde o sobrescrito v indica as iterações do método de Picard para a determinação da pressão.

As propriedades do fluido e da rocha são determinadas no nível iterativo $n+1$, v e são utilizadas, a seguir, no cálculo dos coeficientes do sistema de equações algébricas, de modo a se obter as pressões em $v+1$, $n+1$. Em seguida, atualiza-se os coeficientes e os termos fontes das equações e determina-se p^{n+1}. Prosseguindo, testa-se a convergência da etapa de solução (teste de convergência interno – tol_i) e, também, para a etapa de atualização dos coeficientes do sistema de equações algébricas (teste de convergência externo – tol_e). Portanto, o método de Picard é utilizado na iteração envolvendo a avaliação dos coeficientes, enquanto um método iterativo, o método dos Gradientes Conjugados, é aplicado na solução do sistema de equações algébricas.

3.3 Paralelização do método dos Gradientes Conjugados

O método indireto dos Gradientes Conjugados é classificado como não estacionário e pode ser aplicado quando a matriz dos coeficientes do sistema algébrico for simétrica positiva definida (Ertekin; Abou-Kassem; King, 2001; Saad, 2003; Salles *et al*., 2018; Werneck *et al*., 2019). O sistema algébrico linearizado possui

um elevado número de incógnitas. Então, torna-se necessária a utilização de uma grande quantidade de memória e de recursos de processamento, para a obtenção da solução numérica de casos realísticos. Entre as ferramentas que podem auxiliar na melhoria do desempenho computacional, tem-se o uso de técnicas de compressão de matrizes, as quais permitem o armazenamento de matrizes esparsas em vetores, de forma que somente os elementos não nulos são guardados. Entre as diversas técnicas de compressão, utiliza-se a *Compressed Sparse Row* (CSR) – Linha Esparsa Comprimida (Saad, 2003; Werneck *et al*., 2019). Werneck *et al*. (2016) utilizaram a API OpenMP na simulação numérica do escoamento de gás natural em conjunto com a técnica CSR, no entanto, não foram considerados reservatórios do tipo *shale gas*. Por outro lado, Carvalho *et al*. (2020) focaram o escoamento de gás natural e o efeito do escorregamento. Os autores também utilizaram a API OpenMP e obtiveram resultados empregando métodos iterativos estacionários.

O OpenMP utiliza o modelo *fork-join* e tem como particularidade o fato de utilizar uma *thread* mestre que, quando da execução do código, ao encontrar a diretiva *parallel*, divide as tarefas entre outras *threads* (*fork*). Quando ao final da execução da região paralelizada ocorre a reunião das *threads* (*join*) e a *thread* mestre novamente comandará a execução. A paralelização de um trecho de código se dá pelo uso da diretiva *#pragma omp parallel*, o qual é colocado entre chaves.

Com o intuito de preservar o resultado final do código paralelizado, o OpenMP possui opções conhecidas como construções de sincronização. Uma forma de garantir a consistência dos resultados é definir as variáveis como *privada* ou *compartilhada*. Uma variável é privada quando cada *thread* tem a sua própria cópia dela, que é invisível para as demais. Já uma variável é compartilhada quando todas elas acessam a mesma cópia dela. Para definir uma variável privada, basta acrescentar a cláusula *private* na diretiva: *#pragma omp parallel private(var1, var2,...,varn)* onde *var1, var2,...,varn* representam os nomes das variáveis. Por padrão, o OpenMP define todas as variáveis como compartilhadas, e isso pode ser feito explicitamente através da cláusula *shared(var)*.

Quando existem trechos dos códigos que não podem ser paralelizados, faz-se uso de algumas cláusulas (*single*, *atomic*, *critical*, *master*) que indicam que ele será executado por uma única *thread*. Uma cláusula importante é a *barrier*, que define um ponto específico no programa no qual nenhuma *thread* pode continuar a execução até que as demais atinjam aquele ponto. Implicitamente, ao final de cada região paralela, tem-se uma cláusula *barrier*, mas ela também pode ser introduzida via *#pragma omp barrier* ou removida pela cláusula *nowait*. Ao leitor, sugere-se o trabalho de Chapman, Jost e Van Der Pas (2007) para mais informações sobre todas as cláusulas do OpenMP.

No caso de laços aninhados independentes, eles podem ser paralelizados utilizando a diretiva *#pragma omp parallel for*. A cláusula *collapse(m)*, onde *m* é a quantidade de laços existentes, especifica quantos *loops* em um *loop* aninhado devem ser colapsados em um grande espaço de iteração e divididos de acordo com a cláusula *schedule*. Quando os *loops* não são independentes, pode ocorrer uma queda de desempenho, pois as variáveis entrarão na chamada condição de corrida (Chapman; Jost; Van Der Pas, 2007).

Em regiões possuindo variáveis privadas, quando se paraleliza um *loop* contendo um somatório, tem-se que, a cada iteração, uma *thread* executa uma parte desse somatório. Então, ao término dos cálculos, necessita-se que uma única variável contenha o valor do somatório e, portanto, adiciona-se a diretiva *reduction*(*operador: var1,var2,...,varn*) de modo a garantir a consistência do resultado.

A opção de se paralelizar o método dos Gradientes Conjugados baseou-se na constatação de que a maior parte do esforço computacional é despendida na resolução do sistema algébrico. No método, realizam-se operações do tipo: produto de um vetor por uma matriz; produto escalar entre vetores; e a soma e a multiplicação de um vetor por um escalar. Portanto, a paralelização concentrou-se nessas operações algébricas. Em seguida, apresenta-se trechos do código numérico (em linguagem C) exemplificando o uso da técnica de compressão de matrizes CSR em conjunto com a paralelização empregando o OpenMP.

No caso do produto entre uma matriz e um vetor:

```
1 void produto_matriz_vetor_csr(int NX, int NY, double *Matrixvalues,
2 int *Mtxindj, int *Nonzerosinrow, double *x, double *b)
3 {
4    int i,j,k,N=nxy=NX*NY;
5    double soma;
6
7    #pragma omp parallel for private (i,k) reduction(+:sum)
8    for (i = 1; i <= N; i++)
9    {
10              sum = 0.0;
11              for (k = Nonzerosinrow[i]; k < Nonzerosinrow[i+1]; k++)
12              {
13                        soma=soma + Matrixvalues[k]*x[Mtxindj[k]];
14              }
15   b[i]=soma;
16   }
17 }
```

Para o produto escalar entre dois vetores:

```
1 double produto_escalar(int N,double *a,double *b)
2 {
3    int i;
4    double soma;
5
6    soma=0.0;
7    #pragma omp parallel for private(i) reduction(+:soma)
8    for(i=1;i<=N;i++)
9.   {
10              soma=soma+(a[i]*b[i]);
11   }
12   return soma;
13 }
```

Por último, soma e multiplicação por escalar envolvendo vetores:

```
1 void atualizar_vetor(int N,double *a,double *b,double *c,double d)
2 {
3    int i;
4    #pragma omp parallel for private(i)
5    for(i=1;i<=N;i++)
6    {
7              a[i]= b[i]+(d*c[i]);
8    }
9 }
```

A API OpenMP foi escolhida devido ao fato de ela poder ser aplicada em códigos executados em *hardwares* multiprocessados e com memória compartilhada, além de ser uma técnica que torna bastante favorável a implementação de uma versão paralelizada a partir de um código escrito originalmente para a execução serial. Ademais, ela também permite a paralelização incremental como, por exemplo, de laços (*loops*). Essas estruturas são muito utilizadas na realização de operações na álgebra linear numérica, tais como o produto matriz-vetor e o produto escalar, presentes nas técnicas de resolução de sistemas de equações algébricas como aquela aqui utilizada. O módulo responsável pelo método dos Gradientes Conjugados foi paralelizado, pois ele é o responsável pelo maior custo computacional nesse tipo de aplicação em simulação de reservatórios de petróleo, representando tipicamente mais de 70% do tempo de execução.

4 Resultados numéricos

Os casos estudados se fundamentam na simulação de testes de pressão, como no contexto abordado por Al-Mohannadi (2004). No presente trabalho, são considerados poços verticais em reservatórios do tipo *shale gas*. Simulações mais realísticas de produção deveriam considerar o uso de poços horizontais hidraulicamente fraturados (Wang *et al*., 2017). No entanto, o objetivo principal deste trabalho é a paralelização do método dos Gradientes Conjugados utilizando a API OpenMP. Todas as simulações numéricas foram efetuadas em um servidor PowerEdge R730: sistema operacional CentOS 7.3.1611; 2 Processadores Intel Xeon E5-2620 com 2,4 GHz e 6 núcleos (12 *threads*); e 48 GB de memória RAM.

A Tabela 1 apresenta as propriedades e os valores estipulados para o caso de referência e deve-se salientar que o incremento de tempo varia segundo a fórmula $\Delta t^{n+1} = R_c \Delta t^n$, onde R_c é a razão de crescimento, e que os seus valores encontram-se limitados entre Δt_{inic} e Δt_{max}. Nem todos os valores mostrados na tabela são realísticos, uma vez que o objetivo do trabalho é a proposição de um caso sintético que permita simulações não muito longas, tendo em vista a duração dos casos executados sem que o código seja paralelizado (em série). Nesse sentido, as dimensões do reservatório e a vazão de produção são inferiores às que podem ser observadas em casos de campo. Por outro lado, o tempo máximo de produção (t_{max}) está de acordo com o utilizado na prática para um teste de pressão de longa duração.

Tabela 1 ▼
Parâmetros para o caso de referência.
Fonte: dados da pesquisa

Parâmetro	Valor	Unidade	Parâmetro	Valor	Unidade
c_ϕ	$1{,}0\times10^{-6}$	psi^{-1}	R	10,73	ft^3psi/R lbm-mol
d_m	$2{,}3\times10^{-10}$	ft	R_c	1,5	-
$k_x = k_y$	$5{,}0\times10^{-6}$	darcy	$tol_i = tol_e$	$1{,}0\times10^{-7}$	psi
$L_x = L_y$	16.000	ft	t_{max}	300	dia
L_z	50	ft	T	609,67	R
L_{wf}	40	ft	T_{sc}	519,67	R
$n_x = n_y$	2.561	-	V_L	0,0005	ft^3/lbm
$p_{inic} = p^0$	6.000	psi	γ	0,6	-
p_{sc}	14,65	psi	τ	1,41	-
p_L	1.100	psi	$\phi_{inic} = \phi^0$	0,12	-
$Q_{sc} = \sum q_{sc}$	$-2{,}0\times10^{4}$	scf/dia	Δt_{inic}	0,001	dia
r_w	0,3	ft	Δt_{max}	20	dia

4.1 Desempenho computacional

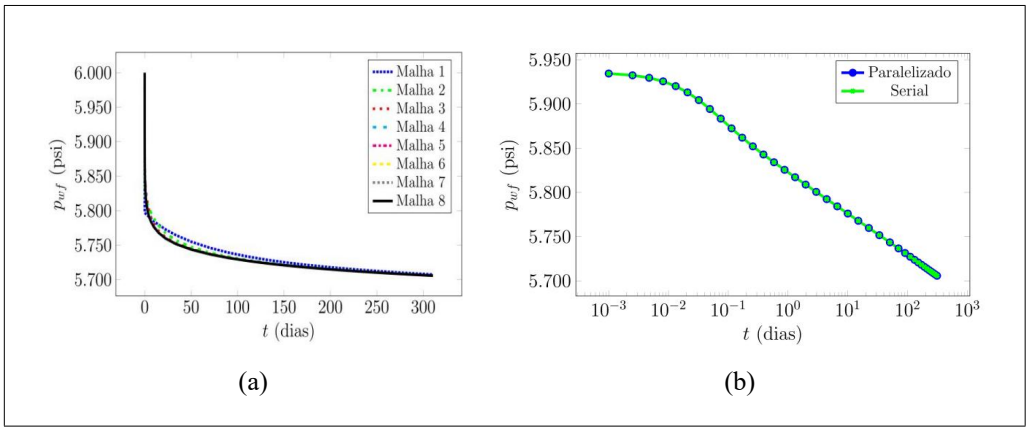
As simulações foram realizadas empregando-se oito diferentes malhas computacionais, cujo número de células pode ser visto na Tabela 2, e um tempo máximo de produção de 300 dias.

Tabela 2 ▶
Parâmetros para o caso de referência.
Fonte: dados da pesquisa

Malha	1	2	3	4	5	6	7	8
$n_x = n_y$	81	161	321	641	1.281	2.561	5.121	10.241

O ganho em desempenho computacional é auferido mediante o valor do *speedup*, S, definido como sendo a relação entre os tempos de execução em série e em paralelo. Na prática, o tempo de execução da versão paralelizada é obtido como sendo a média dos valores de três simulações, e é calculado também o desvio padrão percentual. Dessa forma, um estudo de refinamento de malha foi realizado e pôde-se atestar que os resultados numéricos convergem à medida que a malha computacional é refinada (Dutra *et al*., 2021). Este estudo foi realizado considerando as diferentes malhas apresentadas na Tabela 2. Conforme pode ser visto na Figura 1a, que contém o gráfico da variação da pressão no poço em função do tempo, constata-se que os valores numéricos convergem à medida que as malhas são refinadas e que elas se encontram praticamente sobrepostas a partir da Malha 5. Em função desses resultados, a Malha 6 foi aqui escolhida como sendo a de referência para as simulações tendo-se em vista a convergência dos resultados.

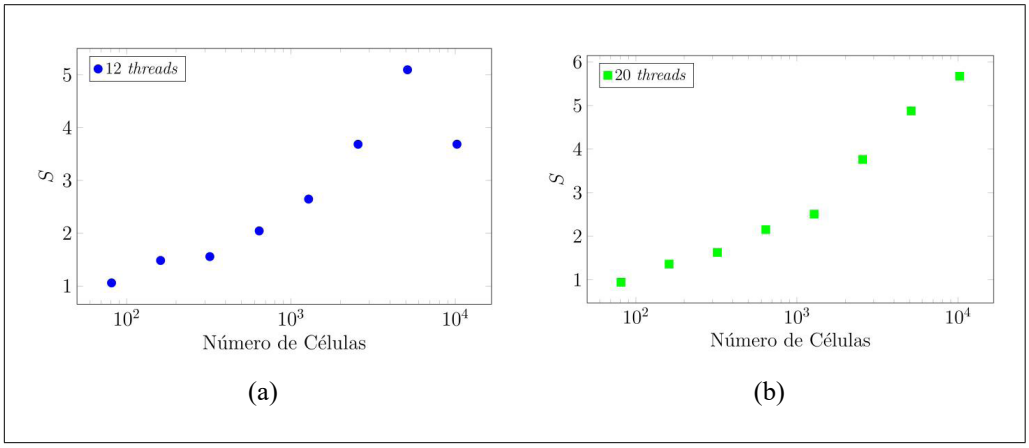
Figura 1 ▶
Verificações numéricas: (a) Refinamento de malha e (b) Comparação dos valores da pressão no poço obtidos com as versões paralelizada e serial.
Fonte: dados da pesquisa



Inicialmente, a fim de verificar se a versão paralelizada forneceria os mesmos resultados que os obtidos com a versão serial, comparou-se as curvas de pressão no poço produtor, calculadas usando a Malha 6 e 20 *threads*. Como é possível observar, vide Figura 1b, as curvas estão sobrepostas e pode-se afirmar que ambas as versões proveram o mesmo resultado, condizente fisicamente com o escoamento estudado. Foi feito um gráfico especializado, tipicamente utilizado na área de Análise de Testes de Pressão na indústria de óleo & gás natural (Al-Mohannadi, 2004). Destaca-se que a versão serial do simulador já havia sido verificada, considerando a lei de Darcy clássica, quando do desenvolvimento do trabalho de Pessanha *et al*. (2020).

Verificados os resultados da versão paralelizada, passou-se ao estudo do ganho em eficiência computacional. Foram feitas simulações utilizando 2, 4, 6, 8, 10, 12, 14, 16, 18 e 20 *threads* e as oito malhas apresentadas na Tabela 2 (Dutra *et al*., 2021). Os melhores desempenhos foram alcançados para 12 e 20 *threads*, vide Figuras 2a e 2b. Embora os valores do *speedup* aumentem progressivamente para as 8 malhas quando se usa 20 *threads* (Figura 2b), o mesmo não pode ser dito em relação aos resultados obtidos com 12 *threads* (Figura 2a). Entretanto, trata-se de um comportamento já relatado em outros trabalhos, que decorre das características do OpenMP e da maneira como o código foi programado (Werneck *et al*., 2019).

Figura 2 ▶
Variação do *speedup* em função do número de células: (a) 12 *threads* e (b) 20 *threads*.
Fonte: dados da pesquisa



É importante destacar que foram obtidos *speedups* maiores do que cinco para a Malha 7 e 12 *threads* e para a Malha 8 e 20 *threads*. Tratando da Malha 7 (12 *threads*), o tempo médio de execução foi de 7.630 s para a versão serial e de 1.497 s para a paralelizada, com desvios padrão iguais a 0,11% e 5,27%, respectivamente. Os correspondentes valores para a Malha 8 (20 *threads*) foram 55.729 s (0,72%) e 9.824 s (3,4%). Para a malha

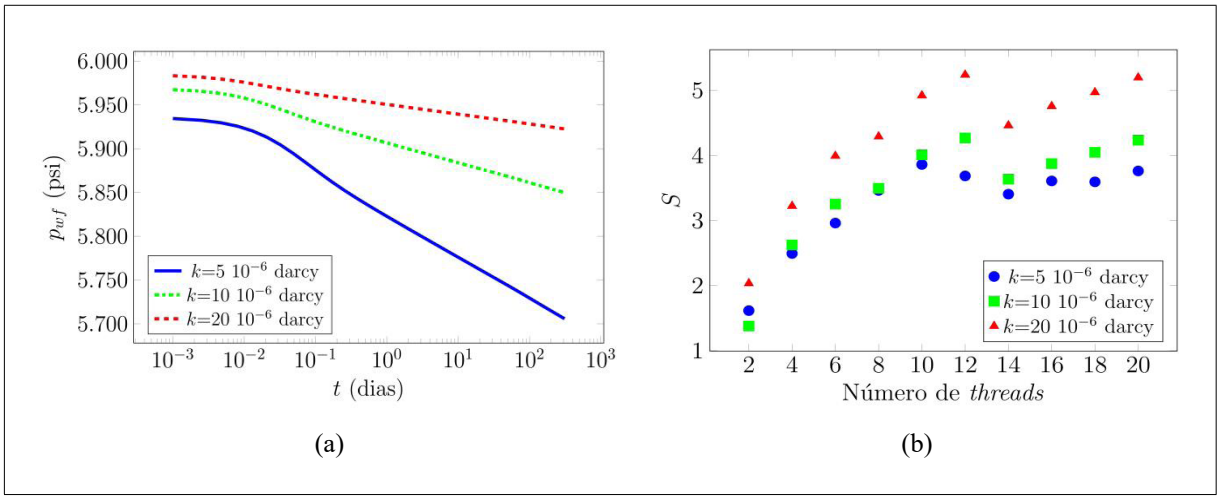
de referência, Malha 6, o maior *speedup* foi obtido com 10 *threads* (3,86) e os valores médios são 1.074 s (0,17%) e 278 s (4,94%). Tal fato atesta que a paralelização do método dos Gradientes Conjugados foi bem-sucedida e que ela proporcionou um ganho de desempenho não desprezível, uma redução de mais de 500% no tempo de execução.

4.2 Análise de sensibilidade

Além do ganho de desempenho proporcionado pela variação do número de células da malha computacional e de *threads*, também foi feito um estudo para verificar a influência da variação da permeabilidade absoluta, da densidade e da vazão de produção na redução do tempo de execução. Todas as simulações foram empreendidas com a Malha 6, 10 *threads* (melhor desempenho para a referida malha) e t_{max} igual a 300 dias.

Inicia-se com as simulações considerando diferentes valores da permeabilidade absoluta $k_x = k_y = k$. Como os efeitos do escorregamento do gás são levados em consideração no modelo físico, a mudança dos valores da permeabilidade absoluta vai levar a variações nos valores da permeabilidade aparente, vide Equação 3. Quanto maior for o valor da permeabilidade aparente, menor será a resistência do meio ao escoamento e, portanto, menor será a queda de pressão no poço produtor para uma vazão constante. Tal constatação pode ser verificada na Figura 3a, que apresenta a variação da pressão no poço produtor em função do tempo para três distintas permeabilidades absolutas e 10 *threads*. No que diz respeito ao desempenho computacional, na Figura 3b tem-se o comportamento do *speedup* em função do número de *threads* e do valor da permeabilidade absoluta. Observa-se que os ganhos são dependentes dos valores da permeabilidade absoluta. Para o mais baixo valor de k, 5×10^{-6} darcy, o maior *speedup* (3,86) foi atingido com 10 *threads*, enquanto para os valores intermediário (10×10^{-6} darcy) e mais alto (20×10^{-6} darcy), eles foram alcançados com 12 *threads* (4,27) e 20 *threads* (5,2), respectivamente. Os tempos médios de execução associados foram de 278,4 s, 311,9 s e 374,17 s. No entanto, apesar da variação, atesta-se que os valores são da mesma ordem de grandeza.

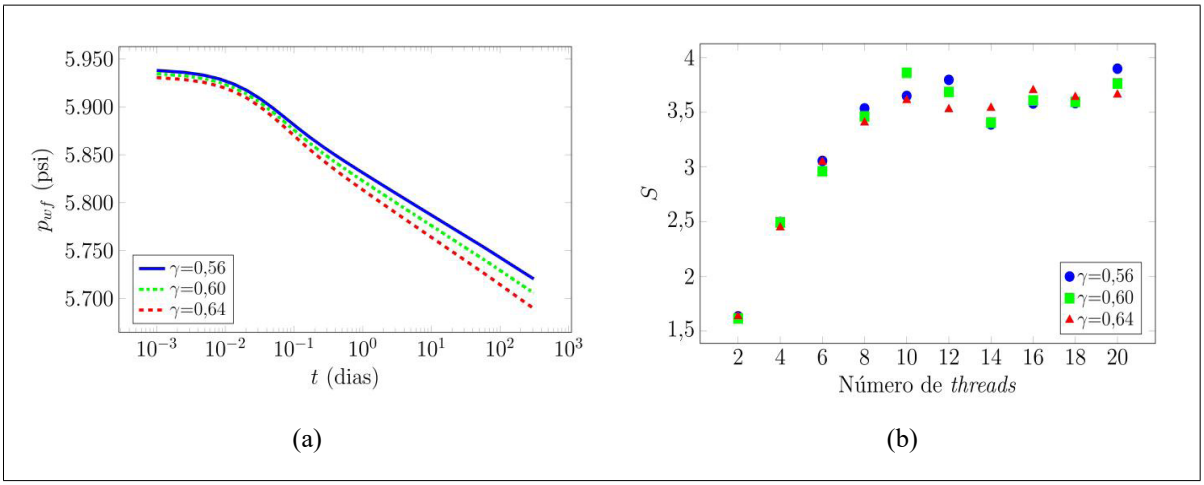
Figura 3 ▼
Influência da variação da permeabilidade absoluta: (a) Variação da pressão no poço produtor em função do tempo e (b) Variação do *speedup* em função do número de *threads*.
Fonte: dados da pesquisa



Na sequência, aborda-se o caso em que os valores da densidade do gás são variados. Trata-se de um problema interessante, uma vez que a densidade afeta os valores da viscosidade, que, por sua vez, alteram o valor do número de Knudsen, da permeabilidade aparente e da velocidade superficial. Tal afirmação é verificada a partir das Equações 3, 4 e 6. No entanto, vê-se que, para os valores de densidade escolhidos,

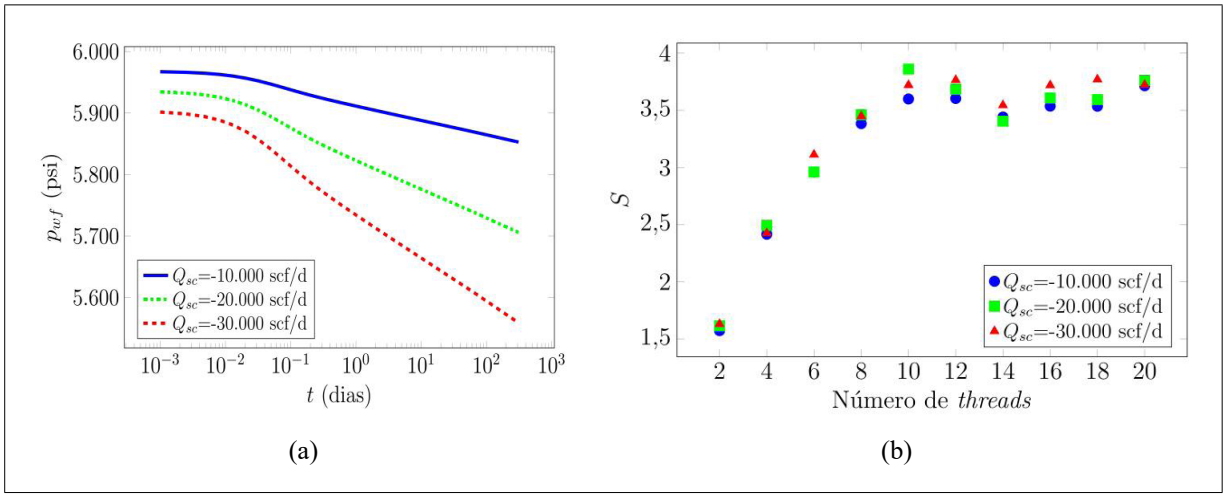
Figura 4 ▼
Influência da variação da densidade: (a) Variação da pressão no poço produtor em função do tempo e (b) Variação do *speedup* em função do número de *threads*.
Fonte: dados da pesquisa

as curvas de pressão no poço produtor mostram uma menor variabilidade do que no caso precedente, conforme mostra a Figura 4a. Assim, o efeito na queda de pressão foi menos pronunciado do que o da permeabilidade absoluta. De modo semelhante, Figura 4b, o desempenho computacional também mudou em função dos valores da densidade. Para a densidade igual a 0,56, o valor mais elevado do *speedup* foi obtido para 20 *threads*, enquanto, para 0,60 e 0,64, eles ocorreram para 10 e 16 *threads*. Em termos absolutos, alcançou-se uma redução do tempo de execução igual a 3,9, 3,61 e 3,7, nessa ordem, sendo que os respectivos tempos médios de execução foram de 278,2 s, 295,7 s e 269,8 s.



O último caso focou a alteração da vazão de produção, mantida fixa nas simulações realizadas. Assim como anteriormente, três valores foram considerados: –10.000 scf/dia, –20.000 scf/dia e –30.000 scf/dia. As respectivas curvas de pressão no poço produtor podem ser vistas na Figura 5a. Diferentemente do caso precedente, a mudança na vazão de produção alterou significativamente a variação de pressão. No caso em análise, esse comportamento já era esperado, visto que uma maior energia é necessária para que o reservatório produza mais gás. Portanto, os menores valores de pressão estarão associados às maiores vazões. Na Figura 5b, encontram-se os valores do *speedup* determinados para as três vazões e fazendo-se variar o número de *threads*. Assim como anteriormente, os maiores ganhos em desempenho mudam em função de e do número de processadores. Da menor para a maior vazão, obtém-se máximos iguais a 3,72 (20 *threads*), 3,86 (10 *threads*) e 3,77 (18 *threads*). Quanto ao tempo médio de execução, tem-se, respectivamente, 269,8 s, 278,4 s e 295,8 s.

Figura 5 ▼
Influência da variação da vazão de produção: (a) Variação da pressão no poço produtor em função do tempo e (b) Variação do *speedup* em função do número de *threads*.
Fonte: dados da pesquisa



Em função dos resultados apresentados, é possível constatar que o desempenho computacional melhora à medida que se aumenta o número de processadores utilizados quando da execução da versão paralelizada via OpenMP. No entanto, conforme relatado na literatura, o ganho em eficiência pode não aumentar continuamente, ou mesmo diminuir e flutuar em torno de um valor (Chapman; Jost; Van Der Pas, 2007; Werneck *et al*., 2019). Consequentemente, com base nos casos aqui abordados, recomenda-se que testes específicos sejam realizados com o número de processadores variando entre 10 e 20, a fim de se determinar a quantidade mais eficiente para o problema que se deseja resolver numericamente. Salienta-se que essa observação se aplica ao caso do computador utilizado, que possui um número máximo de 24 *threads*.

5 Conclusões

Nesta pesquisa, utilizou-se uma das ferramentas voltadas para a computação de alto desempenho – a API OpenMP. Ela foi empregada na paralelização do método dos Gradientes Conjugados, usado na resolução do sistema linearizado de equações algébricas oriundo da discretização da equação governante, com o objetivo de resolver eficientemente o escoamento de gás em reservatórios do tipo *shale gas*. O intuito foi obter um ganho de desempenho computacional, via aplicação de uma API, que permitisse a paralelização dos códigos preexistentes, sem a necessidade do uso de uma programação avançada. Entretanto, cuidados especiais devem ser tomados, a fim de não se comprometer a acurácia dos resultados nem desenvolver uma versão paralelizada que não resulte em um aumento significativo da eficiência computacional.

Os resultados mostraram que, mediante o uso conjugado da técnica de compressão de matrizes CSR e do OpenMP, foi possível alcançar valores do *speedup* próximos a seis, para a Malha 8 e o processamento empregando 20 *threads*. Empreendeu-se, também, uma análise de sensibilidade, com a Malha 6, visando a quantificar a influência da variação da permeabilidade absoluta, da densidade e da vazão de produção na eficiência computacional. Constatou-se que, além dos resultados apresentarem o comportamento físico esperado, os maiores ganhos em eficiência computacional foram atingidos com o uso de 10 a 20 *threads*. No entanto, os valores do *speedup* sofrem alterações, embora não significativas, em função da variabilidade dos valores dessas variáveis, assim como mudam o número de *threads* associadas aos seus valores máximos. Concluindo, pode-se afirmar que a paralelização do método dos Gradientes Conjugados permitiu que ganhos substanciais fossem auferidos no que diz respeito ao tempo total de execução, quando comparado àquele obtido com a versão serial.

Ao se aumentar o número de células na malha computacional, elevando-se assim o esforço computacional, os valores do *speedup* crescem. Entretanto, para a faixa considerada, os ganhos não continuam a aumentar à medida que mais *threads* são empregadas. Depreende-se, então, para os casos aqui abordados, que 12 *threads* representam um número adequado para a obtenção dos *speedups* mais expressivos. Atualmente, um trabalho em andamento enseja a paralelização de outras partes do código numérico, além da correspondente ao método de resolução do sistema de equações algébricas. Por exemplo, foca-se a determinação dos coeficientes da equação governante e, obviamente, uma nova análise de desempenho será necessária. Por último, a API OpenACC também está sendo implementada para a resolução de um outro problema.

Financiamento

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) - Código de Financiamento 001 e da Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), processos E-26/010.001790/2019, E-26/010.002226/2019 e E-26/211.208/2021.

Conflito de interesses

Os autores declaram não haver conflito de interesses.

Referências

ABA, M. M.; PARENTE, V.; SANTOS, E. M. Estimation of water demand of the three major Brazilian shale-gas basins: Implications for water availability. **Energy Policy**, v. 168, 113170, 2022. DOI: https://doi.org/10.1016/j.enpol.2022.113170.

AKHTER, S.; ROBERTS, J. **Multi-core programming**: increasing performance through software multi-threading. Santa Clara, USA: Intel Press, 2006.

ALDINUCCI, M.; CESARE, V.; COLONNELLI, I.; MARTINELLI, A. R.; MITTONE, G.; CANTALUPO, B.; CAVAZZONI, C.; DROCCO, M. Practical parallelization of scientific applications with OpenMP, OpenACC and MPI, **Journal of Parallel and Distributed Computing**, v. 157, p. 13-29, 2021. DOI: https://doi.org/10.1016/j.jpdc.2021.05.017.

AL-MOHANNADI, N. S. **Simulation of horizontal well tests**. 2004. Thesis (Doctorate in Petroleum Engineering) – Colorado School of Mines, Golden, 2004. Disponível em: https://repository.mines.edu/handle/11124/12255. Acesso em: 3 dez. 2022.

CARVALHO, L. T. M.; WERNECK, L. F.; SOUZA, G.; SOUTO, H. P. A. Uma implementação paralelizada via a API OpenMP para a simulação numérica de reservatórios de gás natural. **Revista Brasileira de Computação Aplicada**, v. 12, n. 2, p. 103-121, 2020. DOI: https://doi.org/10.5335/rbca.v12i2.10158.

CHAPMAN, B.; JOST, G.; VAN DER PAS, R. **Using OpenMP**: portable shared memory parallel programming. London: The MIT Press, 2007.

CHEN, Z.; CHU, H.; CHEN, H.; ZHANG, J. A comprehensive model for production data analysis in unconventional reservoirs of hydrocarbons. **International Journal of Hydrogen Energy**, v. 45, n. 33, p. 16432-16446, 2020. DOI: https://doi.org/10.1016/j.ijhydene.2020.04.151.

CHEN, Z.; HUAN, G.; MA, Y. **Computational methods for multiphase flows in porous media**. Philadelphia: Society of Industrial and Applied Mathematics, 2006. DOI: https://doi.org/10.1137/1.9780898718942.

DANDEKAR, A. Y. **Petroleum reservoir rock and fluid properties**. 2nd. ed. Boca Raton: CRC Press, 2013.

DRANCHUK, P. M.; ABOU-KASSEM, J. H. Calculation of Z factors for natural gases using equations of state. **Journal of Canadian Petroleum Technology**, v. 14, n. 3, p. 34-36, 1975. DOI: https://doi.org/10.2118/75-03-03.

DUTRA, B. L.; LIMA, L. B.; SOUZA, G.; AMARAL SOUTO, H. P. Simulação numérica do escoamento em reservatórios do tipo *shale gas* usando o OpenMP. *In*: ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL, 24.; ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS, 12., 2021, Uberlândia. **Anais** [...]. Uberlândia: Universidade Federal de Uberlândia, 2021. Disponível em: https://even3.blob.core.windows.net/anais/422015.pdf. Acesso em: 3 nov. 2022.

ERTEKIN, T.; ABOU-KASSEM, J. H.; KING, G. R. **Basic applied reservoir simulation**. Richardson: Society of Petroleum Engineers, 2001. (SPE Textbook Series, 7).

FLORENCE, F. A.; RUSHING, J. A.; NEWSHAM, K. E.; BLASINGAME, T. A. Improved permeability prediction relations for low permeability sands. *In*: ROCK MOUNTAIN OIL & GAS TECHNOLOGY SYMPOSIUM, 2007, Denver. **Proceedings** [...]. Denver: Society of Petroleum Engineers, 2007. DOI: https://doi.org/10.2118/107954-MS.

JIANG, J.; YOUNIS, R. M. A multimechanistic multicontinuum model for simulating shale gas reservoir with complex fractured system. **Fuel**, v. 161, p. 333-344, 2015. DOI: https://doi.org/10.1016/j.fuel.2015.08.069.

JIMALE, A. O.; RIDZUAN, F.; ZAINON, W. M. N. W. Square matrix multiplication using CUDA on GP-GU. **Procedia Computer Science**, v. 161, p. 398-405, 2019. DOI: https://doi.org/10.1016/j.procs.2019.11.138.

KLINKENBERG, L. J. The permeability of porous media to liquids and gases. *In*: MID-YEAR MEETING, 11., 1941, Tulsa. **Proceedings** [...]. Tulsa: American Petroleum Institute, 1941.

LEE, A. L.; GONZALEZ, M. H.; EAKIN, B. E. The viscosity of natural gases. **Journal of Petroleum Technology**, v. 18, n. 8, p. 997-1000, 1966. DOI: https://doi.org/10.2118/1340-PA.

LENHARD, L. G.; ANDERSEN, S. M.; COIMBRA-ARAÚJO, C. H. Energy-Environmental Implications of Shale Gas Exploration in Paraná Hydrological Basin, Brazil. **Renewable and Sustainable Energy Reviews**, v. 90, p. 56-69, 2018. DOI: https://doi.org/10.1016/j.rser.2018.03.042.

LI, D.; ZHANG, L.; WANG. J. Y.; LU, D.; DU, J. Effect of adsorption and permeability correction on transient pressures in organic rich gas reservoirs: vertical and hydraulically fractured horizontal wells. **Journal of Natural Gas Science and Engineering**, v. 31, p. 214-225, 2016. DOI: https://doi.org/10.1016/j.jngse.2016.02.033.

LI, Q.; XING, H.; LIU, J.; LIU, X. A review on hydraulic fracturing of unconventional reservoir. **Petroleum**, v. 1, n. 1, p. 8-15, 2015. DOI: https://doi.org/10.1016/j.petlm.2015.03.008.

MOHAMMED, A.; ELELIEMY, A.; CIORBA, F. M.; KASIELKE, F.; BANICESCU, I. An approach for realistically simulating the performance of scientific applications on

high performance computing systems. **Future Generation Computer Systems**, v. 111, p. 617-633, 2020. DOI: https://doi.org/10.1016/j.future.2019.10.007.

NICK, H. M.; RAOOF, A.; CENTLER, F.; THULLNER, M.; REGNIER, P. Reactive dispersive contaminant transport in coastal aquifers: numerical simulation of a reactive Henry problem. **Journal of Contaminant Hydrology**, v. 145, p. 90-104, 2013. DOI: https://doi.org/10.1016/j.jconhyd.2012.12.005.

PEACEMAN, D. W. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. **Society of Petroleum Engineers Journal**, v. 23, n. 3, p. 531-543, 1983. DOI: https://doi.org/10.2118/10528-PA.

PENNYCOOK, S. J.; HAMMOND, S. D.; WRIGHT, S. A.; HERDMAN, J. A.; MILLER, I.; JARVIS, S. A. An investigation of the performance portability of OpenCL. **Journal of Parallel and Distributed Computing**, v. 73, n. 11, p. 1439-1450, 2013. DOI: https://doi.org/10.1016/j.jpdc.2012.07.005.

PESSANHA, M. L. O.; ROSÁRIO, R. C. D.; SOUZA, G.; SOUTO, H. P. A. comparative study and sensitivity analysis in simulation of non-Darcy flow in shale gas reservoirs. **International Journal of Advanced Engineering Research and Science**, v. 7, n. 11, p. 109-121, 2020. Disponível em: http://journal-repository.com/index.php/ijaers/article/view/2697. Acesso em: 3 nov. 2022.

REDONDO, C.; RUBIO, G.; VALERO, E. On the efficiency of the IMPES method for two-phase flow problems in porous media. **Journal of Petroleum Science and Engineering**, v. 164, p. 427-436, 2018. DOI: https://doi.org/10.1016/j.petrol.2018.01.066.

ROUSSET, A.; HERRMANN, B.; LANG, C.; PHILIPPE, L. A survey on parallel and distributed multi-agent systems for high performance computing simulations. **Computer Science Review**, v. 22, p. 27-46, 2016. DOI: https://doi.org/10.1016/j.cosrev.2016.08.001.

SAAD, Y. **Iterative methods for sparse linear systems**. 2nd. ed. Philadelphia: Society of Industrial and Applied Mathematics, 2003.

SALLES, R. M.; WERNECK, L. F.; SOUZA, G.; SOUTO, H. P. A. Aplicação de células inativas, *Compressed Sparse Row* e OpenMP na simulação numérica paralelizada de reservatórios de petróleo. **Revista Brasileira de Computação Aplicada**, v. 10, n. 2, p. 64-79, 2018. DOI: https://doi.org/10.5335/rbca.v10i2.8055.

SUTTON, R. P. Compressibility factors for high-molecular-weight reservoir gases. *In*: SPE ANNUAL TECHNICAL CONFERENCE AND EXHIBITION, 1985, Las Vegas. **Proceedings** […]. Las Vegas: Society of Petroleum Engineers, 1985. DOI: https://doi.org/10.2118/14265-MS.

WANG, L.; WANG, S.; ZHANG, R.; WANG, C.; XIONG, Y.; ZHENG, X.; LI, S.; JIN, K.; RUI, Z. Review of multi-scale and multi-physical simulation technologies for shale and tight gas reservoirs. **Journal of Natural Gas Science and Engineering**, v. 37, p. 560-578, 2017. DOI: https://doi.org/10.1016/j.jngse.2016.11.051.

WERNECK, L. F.; FREITAS, M. M.; SILVA JUNIOR, H. G.; SOUZA, G.; SOUTO, H. P. A. An OpenMP parallel implementation for numerical simulation of gas reservoirs using Intel Xeon Phi coprocessor. **RIPE: Revista Interdisciplinar de**

Pesquisa em Engenharia, v. 2, n. 21, p. 37-56, 2016. Disponível em: https://periodicos.unb.br/index.php/ripe/article/view/21697. Acesso em: 3 nov. 2022.

WERNECK, L. F.; FREITAS, M. M.; SOUZA, G.; JATOBÁ, L. F. C.; AMARAL SOUTO, H. P. An OpenMP parallel implementation using a coprocessor for numerical simulation of oil reservoirs. **Computational & Applied Mathematics**, v. 38, 33, 2019. DOI: https://doi.org/10.1007/s40314-019-0788-6.

WOLFE, M.; LEE, S.; KIM, J.; TIAN, X.; XU, R., CHAPMAN, B.; CHANDRASEKARAN, S. The OpenACC data model: preliminary study on its major challenges and implementations. **Parallel Computing**, v. 78, p. 15-27, 2018. DOI: https://doi.org/10.1016/j.parco.2018.07.003.

WRIGHT, S. A. Performance modeling, benchmarking and simulation of high performance computing systems. **Future Generation Computer Systems**, v. 92, p. 900-902, 2019. DOI: https://doi.org/10.1016/j.future.2018.11.020.

WU, Y.; LI, T.; SUN, L.; CHEN, J. Parallelization of a hydrological model using the Message Passing Interface. **Environmental Modelling & Software**, v. 43, p. 124-132, 2013. DOI: https://doi.org/10.1016/j.envsoft.2013.02.002.