

Performance dos Métodos do Ponto Fixo e de Newton-Raphson em Problemas de Fluxo em Meio Poroso Incompressíveis*

Tiago C. A. Amorim²

^aPetrobras, Av. Henrique Valadares, 28, Rio de Janeiro, 20231-030, RJ, Brasil

Abstract

O método usualmente empregado para resolver os sistemas de equações não lineares de simulações de fluxo em meio poroso é o de Newton-Raphson. Em sistemas incompressíveis a matriz jacobiana associada ao problema é próxima da matriz de coeficientes. Foi comparada a performance do Método de Newton-Raphson contra o Método do Ponto Fixo. Os testes realizados mostraram que o Método de Newton-Raphson é sempre melhor que o Método do Ponto Fixo. A performance dos dois métodos se aproximam à medida que o problema fica mais linear.

Keywords: Método de Newton-Raphson, Método do Ponto Fixo, Fluxo em Meio Poroso

1. Introdução

Em simuladores de fluxo em meio poroso o usual é empregar o Método de Newton-Raphson para resolver os sistemas de equações não lineares resultantes[1][2]. Em problemas incompressíveis os termos não lineares do sistema de equações se resumem às permeabilidades relativas. Com um sistema de equações *quase* linear, o Método do Ponto Fixo se mostra como uma alternativa viável. Este trabalho se propõe a avaliar a diferença entre empregar o Método de Newton-Raphson versus o Método do Ponto Fixo.

2. Metodologia

2.1. Método do Ponto Fixo

Dado um sistema de equações:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (2.1)$$

No Método do Ponto Fixo o sistema 2.1 é resolvido transformando cada função $f_i(\vec{x}) = 0$ em $g_i(\vec{x}) = x_i$. Nesta forma o problema é resolvido de forma iterativa (Equação 2.2)[3].

$$\vec{x}^{k+1} = G(\vec{x}^k) \quad (2.2)$$

onde

$$G(\vec{x}) = \{g_1(\vec{x}), g_2(\vec{x}), \dots, g_n(\vec{x})\} \quad (2.3)$$

O algoritmo do Método do Ponto Fixo é apresentado em 1.

Algorithm 1 Método do Ponto Fixo

Entrada: \vec{x}^0

$k \leftarrow 0$

Repete

$\vec{x}^{k+1} \leftarrow G(\vec{x}^k)$

$k \leftarrow k + 1$

Até $\|\vec{x}^k - \vec{x}^{k-1}\| < \epsilon$

Retorna: \vec{x}^k

Diferentes critérios de convergência foram discutidos no relatório anterior[4].

2.2. Método de Newton-Raphson

O método de Newton-Raphson para função de uma variável é um método iterativo que depende do valor da função e de sua derivada para gerar uma série de estimativas da raiz de $f(x)$ (Equação 2.4).

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)} \quad (2.4)$$

A *versão* do método para encontrar a raiz de um sistema de equações $F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x})\}$ faz uso da matriz jacobiana (Equação 2.5)[3].

$$\vec{x}^{k+1} = \vec{x}^k - J(\vec{x}^k)^{-1} F(\vec{x}^k) \quad (2.5)$$

*Relatório número 12 como parte dos requisitos da disciplina IM253: Métodos Numéricos para Fenômenos de Transporte.

**Atualmente cursando doutorado no Departamento de Engenharia de Petróleo da Faculdade de Engenharia Mecânica da UNICAMP (Campinas/SP, Brasil).

Email address: t100675@dac.unicamp.br (Tiago C. A. Amorim)

onde

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (2.6)$$

Definindo $\Delta \vec{x}^k = \vec{x}^{k+1} - \vec{x}^k$, é possível reescrever a Equação 2.5 como um sistema de equações lineares (Equação 2.7). Desta forma, ao invés de inverter a matriz jacobiana, é resolvido um sistema de equações lineares para encontrar \vec{x}^{k+1} .

$$J(\vec{x}^k) \Delta \vec{x}^k = -F(\vec{x}^k) \quad (2.7)$$

O algoritmo do Método de Newton Raphson é apresentado em 2.

Algorithm 2 Método de Newton-Raphson

Entrada: \vec{x}^0

$k \leftarrow 0$

Repete

 Resolve $J(\vec{x}^k) \Delta \vec{x}^k = -F(\vec{x}^k)$

$\vec{x}^{k+1} \leftarrow \vec{x}^k + \Delta \vec{x}^k$

$k \leftarrow k + 1$

Até $\|\Delta \vec{x}^{k-1}\| < \epsilon$

Retorna: \vec{x}^k

2.3. Fluxo em Meio Poroso Incompressível

Em um relatório anterior[5] é feita uma rápida discussão sobre as equações que governam o fluxo em meio poroso incompressível bifásico em uma dimensão (Equações 2.8).

$$\begin{aligned} & \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_w \right)_{i+\frac{1}{2}}^{t_{s+1}} (p_{i+1} - p_i - \gamma_w \Delta D_{i+\frac{1}{2}})^{t_{s+1}} \\ & + \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_w \right)_{i-\frac{1}{2}}^{t_{s+1}} (p_{i-1} - p_i - \gamma_w \Delta D_{i-\frac{1}{2}})^{t_{s+1}} \\ & = \frac{1}{\Delta t} \left(\frac{V \phi S_w}{B_w} \right)_i^{t_{s+1}} - \left(\frac{V \phi S_w}{B_w} \right)_i^{t_s} + q_w^{std} \end{aligned} \quad (2.8a)$$

$$\begin{aligned} & \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_o \right)_{i+\frac{1}{2}}^{t_{s+1}} (p_{i+1} - p_i - \gamma_o \Delta D_{i+\frac{1}{2}})^{t_{s+1}} \\ & + \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_o \right)_{i-\frac{1}{2}}^{t_{s+1}} (p_{i-1} - p_i - \gamma_o \Delta D_{i-\frac{1}{2}})^{t_{s+1}} \\ & = \frac{1}{\Delta t} \left(\frac{V \phi (1 - S_w)}{B_o} \right)_i^{t_{s+1}} - \left(\frac{V \phi (1 - S_w)}{B_o} \right)_i^{t_s} + q_o^{std} \end{aligned} \quad (2.8b)$$

com:

$$\lambda_p = \frac{k k_{rp}}{B_p \mu_p}$$

Para passar para problemas bi e tridimensionais basta somar termos de transferência de massa entre as células vizinhas (à esquerda da igualdade em 2.8).

As variáveis do problema são as pressões (p) e saturações de água (S_w) das células. É possível colocar este conjunto de equações na forma $Kx = f$. As equações 2.9, 2.10 e 2.11 assumem problema unidimensional e plano¹.

Por simplicidade, os termos fonte ($q_{p,i}^{std}$) foram colocados em f . Estes termos podem aparecer tanto em f como em K . Será função do tipo do controle do poço. Poços controlados por vazão terão seus termos em f , e poços controlados por pressão de fundo terão seus termos em f e K .

$$x = \begin{bmatrix} p_1 \\ Sw_1 \\ p_2 \\ Sw_2 \\ \vdots \\ p_n \\ Sw_n \end{bmatrix} \quad (2.10)$$

$$f = \begin{bmatrix} - \left(\frac{V \phi S_w}{B_w} \right)_1^{t_s} + q_{w,1}^{std} \\ \left(\frac{V \phi S_w}{B_o} \right)_1^{t_s} + q_{o,1}^{std} \\ - \left(\frac{V \phi S_w}{B_w} \right)_2^{t_s} + q_{w,2}^{std} \\ \left(\frac{V \phi S_w}{B_o} \right)_2^{t_s} + q_{o,2}^{std} \\ \vdots \\ - \left(\frac{V \phi S_w}{B_w} \right)_n^{t_s} + q_{w,n}^{std} \\ \left(\frac{V \phi S_w}{B_o} \right)_n^{t_s} + q_{o,n}^{std} \end{bmatrix} \quad (2.11)$$

Para utilizar o Método de Newton-Raphson é preciso encontrar a matriz jacobiana do sistema de equações não lineares a ser resolvido. Os únicos termos não lineares das equações são as permeabilidades relativas: k_{rw} e k_{ro} . O termo fonte também pode aparecer, caso seja controlado por pressão de fundo.

É possível observar que a matriz jacobiana será a matriz de coeficientes (K , Equação 2.9), acrescida das derivadas parciais dos termos não lineares das transmissibilidades (J_{tr} , Equação 2.12) e dos termos fonte (J_q , Equação 2.13): $J = K + J_{tr} + J_q$. A equação 2.12 assume que o termo $i - \frac{1}{2}$ é avaliado na célula $i - 1$ (esquema implícito, com maior pressão em $i - 1$).

2.4. Problema proposto

O problema proposto é a resolução de uma simulação de um modelo de fluxo em meio poroso incompressível. O problema é bidimensional quadrado ($ni = nj$), em uma

¹ $\Delta D = 0$.

$$K = \begin{bmatrix} -\left(\frac{A}{\Delta x}\lambda_w\right)_{1.5} & \left(\frac{V\phi}{\Delta t B_w}\right)_1 & \left(\frac{A}{\Delta x}\lambda_w\right)_{1.5} & 0 & \cdots & 0 & 0 \\ -\left(\frac{A}{\Delta x}\lambda_o\right)_{1.5} & -\left(\frac{V\phi}{\Delta t B_o}\right)_1 & \left(\frac{A}{\Delta x}\lambda_o\right)_{1.5} & 0 & \cdots & 0 & 0 \\ \left(\frac{A}{\Delta x}\lambda_w\right)_{1.5} & 0 & -\left(\frac{A}{\Delta x}\lambda_w\right)_{1.5} - \left(\frac{A}{\Delta x}\lambda_w\right)_{2.5} & \left(\frac{V\phi}{\Delta t B_w}\right)_2 & \cdots & 0 & 0 \\ \left(\frac{A}{\Delta x}\lambda_o\right)_{1.5} & 0 & -\left(\frac{A}{\Delta x}\lambda_o\right)_{1.5} - \left(\frac{A}{\Delta x}\lambda_o\right)_{2.5} & -\left(\frac{V\phi}{\Delta t B_o}\right)_2 & \cdots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & \cdots & \left(\frac{A}{\Delta x}\lambda_w\right)_{n-\frac{1}{2}} & 0 & -\left(\frac{A}{\Delta x}\lambda_w\right)_{n-\frac{1}{2}} & \left(\frac{V\phi}{\Delta t B_w}\right)_n \\ 0 & 0 & \cdots & \left(\frac{A}{\Delta x}\lambda_o\right)_{n-\frac{1}{2}} & 0 & -\left(\frac{A}{\Delta x}\lambda_o\right)_{n-\frac{1}{2}} & -\left(\frac{V\phi}{\Delta t B_o}\right)_n \end{bmatrix} \quad (2.9)$$

$$J_{tr} = \begin{bmatrix} 0 & -\left(\frac{A}{\Delta x}\frac{\partial\lambda_w}{\partial Sw_1}\right)_{1.5} & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\left(\frac{A}{\Delta x}\frac{\partial\lambda_o}{\partial Sw_1}\right)_{1.5} - \frac{\partial q_{o,1}^{std}}{\partial Sw_1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \left(\frac{A}{\Delta x}\frac{\partial\lambda_w}{\partial Sw_1}\right)_{1.5} & 0 & -\left(\frac{A}{\Delta x}\frac{\partial\lambda_w}{\partial Sw_2}\right)_{1.5} - \left(\frac{A}{\Delta x}\frac{\partial\lambda_w}{\partial Sw_2}\right)_{2.5} & \cdots & 0 & 0 \\ 0 & \left(\frac{A}{\Delta x}\frac{\partial\lambda_o}{\partial Sw_1}\right)_{1.5} & 0 & -\left(\frac{A}{\Delta x}\frac{\partial\lambda_o}{\partial Sw_2}\right)_{1.5} - \left(\frac{A}{\Delta x}\frac{\partial\lambda_o}{\partial Sw_2}\right)_{2.5} & \cdots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & & \left(\frac{A}{\Delta x}\frac{\partial\lambda_w}{\partial Sw_{n-1}}\right)_{n-\frac{1}{2}} & 0 & 0 \\ 0 & 0 & \cdots & 0 & & \left(\frac{A}{\Delta x}\frac{\partial\lambda_o}{\partial Sw_{n-1}}\right)_{n-\frac{1}{2}} & 0 & 0 \end{bmatrix} \quad (2.12)$$

$$J_q = \begin{bmatrix} -\frac{\partial q_{w,1}^{std}}{\partial p_1} & -\frac{\partial q_{w,1}^{std}}{\partial Sw_1} & 0 & 0 & \cdots & 0 & 0 \\ -\frac{\partial q_{o,1}^{std}}{\partial p_1} & -\frac{\partial q_{o,1}^{std}}{\partial Sw_1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -\frac{\partial q_{w,2}^{std}}{\partial p_2} & -\frac{\partial q_{w,2}^{std}}{\partial Sw_2} & \cdots & 0 & 0 \\ 0 & 0 & -\frac{\partial q_{o,2}^{std}}{\partial p_2} & -\frac{\partial q_{o,2}^{std}}{\partial Sw_2} & \cdots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & -\frac{\partial q_{w,n}^{std}}{\partial p_n} & -\frac{\partial q_{w,n}^{std}}{\partial Sw_n} \\ 0 & 0 & \cdots & 0 & 0 & -\frac{\partial q_{o,n}^{std}}{\partial p_n} & -\frac{\partial q_{o,n}^{std}}{\partial Sw_n} \end{bmatrix} \quad (2.13)$$

malha uniforme (Δx , Δy e Δz constantes). As propriedades de fluido (Bw , Bo , μ_w e μ_o) e de rocha (ϕ e k) são constantes.

As curvas de permeabilidade relativa foram construídas com a formulação de Corey (Equações 2.14)². Ao utilizar esta formulação é possível calcular analiticamente as derivadas de k_{rw} e k_{ro} com relação a Sw .

$$k_{rw} = k_{rw}^{(Sw=1-Sor)} Sw_d^{nw} \quad (2.14a)$$

$$k_{ro} = k_{ro}^{(Sw=Swi)} (1 - Sw_d)^{no} \quad (2.14b)$$

com:

$$Sw_d = \frac{Sw - Swi}{1 - Sor - Swi}$$

O modelo tem dois poços: um injetor de água controlado por vazão constante (q_w^{std}) na célula $[0, 0]$ e um produtor controlado por pressão de fundo (p_{wf}) na célula $[ni, nj]$.

Os valores dos parâmetros do modelo proposto estão no Apêndice B.

3. Implementação

Todo o código utilizado nesta análise foi desenvolvido em C++.

O modelo de fluxo em meio poroso é implementado em uma classe: *SimModel*. Nesta classe são informadas as propriedades de reservatório, fluido, permeabilidade relativa e parâmetros dos poços. Existe a opção de usar Newton-Raphson ou Ponto Fixo para resolver os sistema de equações não lineares. O principal método da classe é *run_simulation*, que faz a simulação do modelo de fluxo em meio poroso.

As condições iniciais ($p^{t=0}$ e $Sw^{t=0}$) e as condições de contorno (limites do reservatório e controles dos poços) são dadas. A simulação de fluxo assume um passo de tempo (Δt) e tenta resolver as Equações 2.8. Ao final de cada resolução de um passo de tempo é avaliado se o resultado é aceito ou se é reduzido o passo de tempo e refeita a resolução das Equações 2.8.

Para controlar os erros de balanço de massa, é avaliado se as máximas variações de p e Sw estão dentro dos limites estabelecidos. Se os limites não forem satisfeitos, o passo de tempo é reduzido e os cálculos refeitos. Se os limites forem atendidos o valor do passo de tempo será incrementado para o próximo cálculo. O algoritmo implementado é apresentado em 3.

Duas funções foram implementadas para resolver o comando **Resolve** $p^{t+\Delta t}$, $Sw^{t+\Delta t}$: Newton-Raphson e Ponto Fixo. O algoritmo de Newton-Raphson segue 2.

Algorithm 3 Simulação em Meio Poroso

Entrada: $p^{t=0}$, $Sw^{t=0}$, t_{fim} e Δt

$t \leftarrow 0$

Repete

Resolve $p^{t+\Delta t}$, $Sw^{t+\Delta t}$

$\Delta p_{max} \leftarrow \max |p^{t+\Delta t} - p^t|$

$\Delta Sw_{max} \leftarrow \max |Sw^{t+\Delta t} - Sw^t|$

$convergiu \leftarrow \Delta p_{max} < \epsilon_{\Delta p}$ e $\Delta Sw_{max} < \epsilon_{\Delta Sw}$

Se $convergiu$ ou $\Delta t = \Delta t_{min}$ **então**

$t \leftarrow t + \Delta t$

$\Delta t \leftarrow \min(1.2\Delta t, \Delta t_{max})$

Caso contrário

$\Delta t \leftarrow \max(0.5\Delta t, \Delta t_{min})$

Até $t \geq t_{fim}$

Retorna: \vec{x}^k

O Método do Ponto Fixo implementado é ligeiramente diferente do apresentado em 1. Os termos não lineares seguem a proposta do método, e são avaliados com os valores dos parâmetros da iteração anterior (\vec{x}^k). Os termos lineares são resolvidos como um sistema de equações lineares. O algoritmo modificado é apresentado em 4.

Algorithm 4 Método do Ponto Fixo Modificado

Entrada: $\vec{x}^0 = \{p^t, Sw^t\}$

$k \leftarrow 0$

Repete

$k_{ro} \leftarrow f k_{ro}(\vec{x}^k)$

$k_{rw} \leftarrow f k_{rw}(\vec{x}^k)$

Monta matriz K (2.9)

Monta vetor f (2.11)

Resolve $K \vec{x}^{k+1} = f$

$k \leftarrow k + 1$

Até $\|\vec{x}^k - \vec{x}^{k-1}\| < \epsilon$

Retorna: $\vec{x}^k = \{p^{t+\Delta t}, Sw^{t+\Delta t}\}$

Em ambos métodos foi utilizado o mesmo critério de parada, discutido anteriormente[4]. A proposta foi de utilizar o máximo erro relativo por variável:

$$\epsilon = \left\| \frac{x_i^k - x_i^{k-1}}{x_i^k} \right\|_{\infty} \quad (3.1)$$

Em 3.1 a divisão é feita elemento a elemento (*piecewise*).

Todas as resoluções de sistemas lineares são feitas com o Método de Eliminação de Gauss com Pivotamento Parcial e Escala[5].

4. Resultados

Um mesmo modelo de reservatório com $n_i = n_j = 10$ foi simulado por 5 anos, utilizando os métodos de Newton-Raphson e do Ponto Fixo para resolver as equações não-

²Por simplicidade, foi assumido que as saturações de água inicial, crítica e conata coincidem.

lineares. Observa-se que os resultados foram idênticos (Gráfico 1).

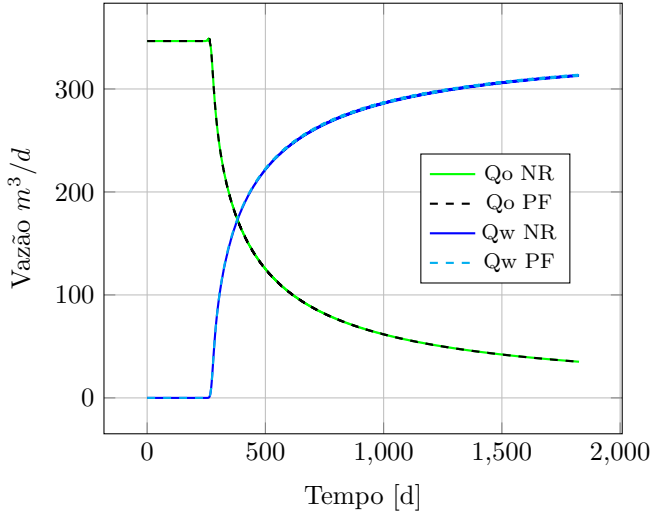


Figura 1: Vazão de óleo e água para a resolução com os métodos de Newton-Raphson (NR) e do Ponto Fixo (PF).

Foi implementado um contador de chamadas à rotina de solução de sistemas de equações lineares para ser um indicador da performance de cada método. O Gráfico 2 mostra que enquanto a produção é apenas de óleo os dois métodos tem a mesma performance. À medida que mais água chega do produtor, o método de Newton-Raphson consegue convergir com um número menor de iterações. O Método de Newton-Raphson usou 24% menos iterações que o Método do Ponto Fixo.

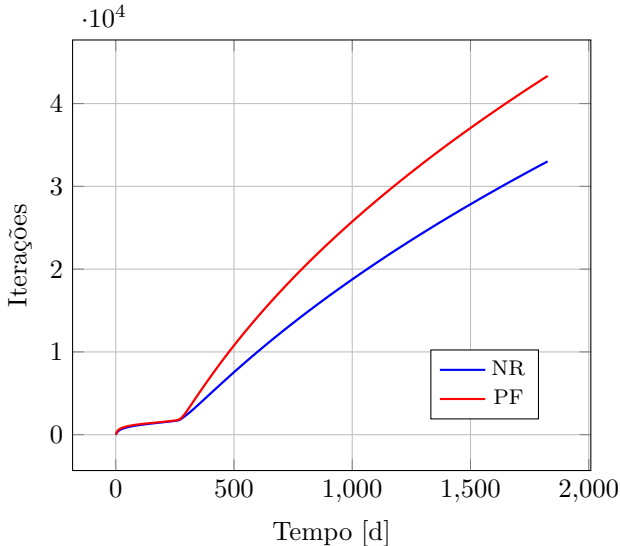


Figura 2: Número total de iterações para a resolução com os métodos de Newton-Raphson (NR) e do Ponto Fixo (PF).

Foi feito um estudo de sensibilidade para avaliar o impacto de diferentes parâmetros no número total de iterações que cada método precisou para resolver a simulação

(Tabela 3). A sensibilidade mostra que o número de iterações não é diretamente proporcional aos valores dos parâmetros testados (exceto o número de células), mas que o Método de Newton-Raphson sempre tem um resultado melhor. O parâmetro que mais influenciou a diferença entre a performance dos métodos foi a viscosidade.

O código foi implementado em C++ e em um único arquivo. Pode ser encontrado em <https://github.com/TiagoCAAmorim/numerical-methods>.

5. Conclusão

Os resultados mostraram que o Método de Newton-Raphson é sempre melhor que o Método do Ponto Fixo, necessitando de menos iterações para convergir. Para problemas mais simples, com menos células, curvas de permeabilidade relativa mais próximas do linear ou razão de mobilidade mais próxima da unidade, os dois métodos tem performance parecida. À medida que os problemas ficam *mais não lineares*, a diferença entre os métodos aumenta.

Apêndice A. Lista de Variáveis

B_o : Fator volume de formação do óleo no reservatório (m^3/m^3).

B_w : Fator volume de formação da água no reservatório (m^3/m^3).

D : Profundidade.

Δx : Discretização espacial na direção i .

Δy : Discretização espacial na direção j .

Δz : Discretização espacial na direção k .

Δt : Discretização temporal.

γ_p : Peso específico da fase p ($\gamma_p = \rho_p g$).

k : Permeabilidade absoluta do meio poroso.

k_{rp} : Permeabilidade relativa da fase p .

μ_p : Viscosidade da fase p .

n_i : Número de células na direção i .

n_j : Número de células na direção j .

p : Pressão.

p_{wf} : Pressão de fundo do poço.

ϕ : Porosidade da rocha.

q_p : Vazão volumétrica da fase p .

q_p^{std} : Vazão volumétrica da fase p medida em condições padrão (*standard*).

ρ_p : Densidade da fase p .

Figura 3: Sensibilidade do número total de iterações.

Parâmetro	Iterações		Redução ao usar Newton-Raphson
	Ponto Fixo	Newton-Raphson	
Modelo Proposto	43 350	33 007	24%
$\mu_o = 0.1 \text{ cP}$	10 822	10 490	3%
$\mu_o = 1.0 \text{ cP}$	176 682	176 185	0%
$\mu_o = 10 \text{ cP}$	85 058	80 633	5%
$\mu_o = 100 \text{ cP}$	46 877	37 224	19%
$n_i = n_j = 3$	4 927	3 867	22%
$n_i = n_j = 9$	35 511	26 983	24%
$n_o = n_w = 1.5$	63 159	50 185	21%
$n_o = n_w = 2.0$	44 684	34 786	22%
$n_o = n_w = 3.0$	28 155	20 175	28%

r_w : Raio do poço.

S_p : Saturação da fase p no meio poroso.

S_{wi} : Saturação de água inicial (imóvel).

S_{or} : Saturação de óleo residual (imóvel).

$$q_w^{std} = 350 \text{ m}^3/d \text{ (injetor de água)}$$

$$r_w = 4 \text{ pol}$$

$$S_{wi} = 20\%$$

$$S_{or} = 15\%$$

$$t_{final} = 5 \text{ anos}$$

Apêndice B. Parâmetros do Problema Proposto

$$Bo = 1.01$$

$$Bw = 1.00$$

$$\Delta x = 50 \text{ m}$$

$$\Delta y = 50 \text{ m}$$

$$\Delta z = 30 \text{ m}$$

$$k = 1000 \text{ mD}$$

$$k_{ro}^{Sw=S_{wi}} = 1.00$$

$$k_{rw}^{Sw=1-S_{or}} = 0.60$$

$$\mu_o = 130 \text{ cP}$$

$$\mu_w = 1.0 \text{ cP}$$

$$n_i = 10$$

$$n_j = 10$$

$$n_o = 2.0$$

$$n_w = 3.0$$

$$p^{t=0} = 340 \text{ bar}$$

$$p_{wf} = 330 \text{ bar (produtor)}$$

$$\phi = 15 \%$$

Referências

- [1] C. M. G. L. (CMG), Cmg imex user's manual, CMG Calgary, Canada, 2022.
- [2] E. U. M. SCHLUMBERGER, Technical description, Schlumberger Ltd (2009) 519–538.
- [3] R. L. Burden, J. D. Faires, A. M. Burden, Análise numérica, Cengage Learning, 2016.
- [4] T. C. A. Amorim, Performance dos métodos de jacobi, gauss-siedel e srs para resolver sistemas de equações de problemas de fluxo em meio poroso incompressíveis, Relatório número 11 da disciplina IM253: Métodos Numéricos para Fenômenos de Transporte (12 2023).
- [5] T. C. A. Amorim, Performance do método de eliminação de gauss para resolver sistema de equações de problemas de fluxo em meio poroso incompressíveis, Relatório número 10 da disciplina IM253: Métodos Numéricos para Fenômenos de Transporte (12 2023).