Programa de Engenharia Civil (PEC)

Disciplina:CPC-777 Modelagem e simulação de reservatórios
Professor (es): Paulo Couto
Aluno (a): Vivian de Carvalho Rodrigues
DRE: 121010011

Relatório: Lista 2

Histórico de revisões

Rev.	Data	Motivo
0	02-out-2021	Primeira emissão (disponível na atividade Lista 2- Google classroom)
1	09-out-2021	Ajustes na metodologia implícita (código e resultados)

Sumário

1. Introdução

Este relatório tem por objetivo apresentar a resolução dos problemas referentes a Lista 2 da disciplina Modelagem e simulação de reservatórios (CPC 777).

2. Problema 1

Provar a através de Séries de Taylor que as seguintes aproximações são verdadeiras [1]:

$$\frac{df}{dx} = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12\Delta x} + \varepsilon(\Delta x^4)$$

$$\frac{df}{dx} = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2\Delta x} + \varepsilon(\Delta x^2)$$

$$\frac{d^2f}{dx^2} = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{\Delta x^2} + \varepsilon(\Delta x)$$

Figura 1 – Enunciado do problema 1 da lista 2 [1]

2.1 Resolução

[3] A série de Taylor é definida da seguinte forma, para um ponto x em torno de 'a' e 'x>a':

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^n}{n!} f^n(a) =$$

$$f(a) + (x-a)\left.\frac{df}{dx}\right]_{x=a} + \frac{(x-a)^2}{2!}\left.\frac{d^2f}{dx^2}\right]_{x=a} + \frac{(x-a)^3}{3!}\left.\frac{d^3f}{dx^3}\right]_{x=a} + \cdots + \frac{(x-a)^n}{n!}\left.\frac{d^nf}{dx^n}\right]_{x=a} \quad \text{(Eq.1)[2]}$$

Para um ponto em torno de 'a' e 'x < a':

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^n}{n!} f^n(a) =$$

$$f(a) - (x-a)\left.\frac{df}{dx}\right]_{x=a} + \frac{(x-a)^2}{2!}\left.\frac{d^2f}{dx^2}\right]_{x=a} - \frac{(x-a)^3}{3!}\left.\frac{d^3f}{dx^3}\right]_{x=a} + \cdots + \frac{(-1)^n(x-a)^n}{n!}\left.\frac{d^nf}{dx^n}\right]_{x=a} \quad \text{(Eq.2)[2]}$$

Fazendo

$$f(x_{i+1}) = f(x + \Delta x)$$
$$f(x_{i-1}) = f(x - \Delta x)$$
$$f(x_{i+2}) = f(x + 2\Delta x)$$
$$f(x_{i-2}) = f(x - 2\Delta x)$$

Onde

$$x_{i+1} - x_i = \Delta x$$
$$x_{i-1} - x_i = -\Delta x$$
$$\left.\frac{d^nf}{dx^n}\right]_{x_i} = f^n(x_i)$$

Então seguindo a expansão da ST no entorno de xi:

$$(i) \qquad f(x+\Delta x) = f(x_i) + \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + \frac{\Delta x^3}{3!} f'''(x_i) + \frac{\Delta x^4}{4!} f''''^{(x_i)} + \cdots + \frac{\Delta x^n}{n!} f^{n(x_i)}$$

$$(ii) \qquad f(x-\Delta x) = f(x_i) - \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) - \frac{\Delta x^3}{3!} f'''(x_i) + \frac{\Delta x^4}{4!} f''''^{(x_i)} + \cdots + (-1)^n \frac{\Delta x^n}{n!} f^{n(x_i)}$$

$$(iii) \qquad f(x+2\Delta x) = f(x_i) + 2\Delta x\, f'(x_i) + \frac{(2\Delta x)^2}{2!} f''(x_i) + \frac{(2\Delta x)^3}{3!} f'''(x_i) + \frac{(2\Delta x)^4}{4!} f''''^{(x_i)} + \cdots + \frac{(2\Delta x)^n}{n!} f^{n(x_i)}$$

$$(iv) \qquad f(x-2\Delta x) = f(x_i) - 2\Delta x\, f'(x_i) + \frac{(2\Delta x)^2}{2!} f''(x_i) - \frac{(2\Delta x)^3}{3!} f'''(x_i) + \frac{(2\Delta x)^4}{4!} f''''^{(x_i)} + \cdots + (-1)^n \frac{(2\Delta x)^n}{n!} f^{n(x_i)}$$

Subtraindo (ii) de (i)

$$f(x+\Delta x) - f(x-\Delta x) = \qquad (I)$$
$$2\Delta x\, f'(x_i) + \frac{2\,\Delta x^3}{3!} f'''(x_i) + \frac{2\,\Delta x^5}{5!} f'''''(x_i) + \cdots + \frac{2\,\Delta x^{2n+1}}{(2n+1)!} f^{2n+1}(x_i)$$

Subtraindo (iv) de (iii)

$$f(x+2\Delta x) - f(x-2\Delta x) = \qquad (II)$$
$$4\Delta x\, f'(x_i) + \frac{2\,(2\Delta x)^3}{3!} f'''(x_i) + \frac{2\,(2\Delta x)^5}{5!} f'''''(x_i) + \cdots + \frac{2\,(2\Delta x)^{2n+1}}{(2n+1)!} f^{2n+1}(x_i)$$

Fazendo 8 (I) – II

$$8\, f(x+\Delta x) - 8f(x-\Delta x) - f(x+2\Delta x) + f(x-2\Delta x) = 12\,\Delta x\, f'(x_i) - \frac{48\,\Delta x^5}{5!} f'''''(x_i) + \cdots$$

Isolando f’(xi), obtém-se, portanto:

$$\boxed{f'(x_i) = \frac{8\, f(x+\Delta x) - 8f(x-\Delta x) - f(x+2\Delta x) + f(x-2\Delta x)}{12\,\Delta x} + \left\{\frac{48\,\Delta x^4}{5!} f'''''(x_i) + \cdots\right\}}$$

Fazendo 4 (i) – (iii)

$$4\, f(x+\Delta x) - f(x+2\Delta x) = 3\, f(x_i) + 2\,\Delta x\, f'(x_i) - \frac{4\,\Delta x^3}{3!} f'''(x_i) - \frac{12\,\Delta x^4}{4!} f''''(x_i) - \cdots$$

Isolando f’(xi), obtém-se, portanto:

$$\boxed{f'(x_i) = \frac{4\, f(x+\Delta x) - f(x+2\Delta x) - 3\, f(x_i)}{2\,\Delta x} + \left\{\frac{2\,\Delta x^2}{3!} f'''(x_i) - \frac{6\,\Delta x^3}{4!} f''''(x_i) - \cdots\right\}}$$

Fazendo (iv) – 2(ii)

$$f(x-2\Delta x)-\ 2f(x-\Delta x)=\ -f(x_i)+\Delta x^2 f''(x_i)-\frac{6\,\Delta x^3}{3!}f'''(x_i)+\frac{14\,\Delta x^4}{4!}f''''(x_i)+\cdots$$

Isolando f''(xi), obtém-se, portanto:

$$\boxed{f''(x_i)=\frac{f(x-2\Delta x)-\ 2f(x-\Delta x)+\ f(x_i)}{\Delta x^2}+\left\{\frac{6\,\Delta x}{3!}f'''(x_i)-\frac{14\,\Delta x^2}{4!}f''''(x_i)+\cdots\right\}}$$

3. Problema 2

2. O fluxo radial unidimensional em regime transiente é dado por:

$$\frac{\partial P_D}{\partial t_D}=\frac{1}{r_D}\frac{\partial}{\partial r_D}\left(r_D\frac{\partial P_D}{\partial r_D}\right)$$

$$P_D(r_D,t_D=0)=0$$

$$\left(r_D\frac{\partial P_D}{\partial r_D}\right)_{r_D=1}=-1$$

$$\left(\frac{\partial P_D}{\partial r_D}\right)_{r_D=r_{eD}=r_e/r_w}=0$$

onde P_D, r_D e t_D são as variáveis adimensionais

$$P_D=\frac{2\pi kh}{q_{o,std}B_o\mu_o}(P_i-P)\qquad t_D=\frac{kt}{\phi\mu_o c_t r_w^2}\qquad r_D=\frac{r}{r_w}$$

Utilizando os dados da Tabela 1, resolva o problema dado pelas Eqs. (1.4) e (1.5) por diferenças finitas utilizando:

a) Formulação explícita;

b) Formulação totalmente implícita; e

obtenha e plote os perfis radiais de pressão (dimensional) no meio poroso entre r_w e r_e para t = 1 minuto, t = 1 hora, t = 1 dia, t = 30 dias, t = 1 ano e t = 10 anos. Compare seus resultados com aqueles obtidos na lista 1. Como você interpreta seus resultados?

Tabela 1. Dados para o problema 2.

Parâmetro	***Sistema de unidades***	
	API	***SI***
ϕ	20 %	20%
c_t	1,5 × 10⁻⁵ psi⁻¹	2,18 × 10⁻⁹ Pa⁻¹
k	150 mD	148 × 10⁻¹⁵ m²
h	10 ft	3,048 m
$q_{o,std}$	3000 STB/dia	518 × 10⁻⁵ m³/seg.
B_o	1,5 bbl/STB	1,5 m³/ m³std
μ_o	0,33 cp	3,3 × 10⁻⁴ Pa.s
P_i	2.200 psi	15,17 × 10⁶ Pa
r_w	3,5 pol.	0,0889 m
r_e	2.000 ft	609,6 m

Figura 2 - Enunciado do problema 2 da lista 2 [1]

3.1 Resolução

Seguindo os passos de resolução descrito em [2], a aproximação *forward* da derivada primeira de f (x_i) é

$$\left.\frac{df}{dx}\right]_{x_i} = \frac{f(x_{i+1}) - f(x_i)}{\Delta x} - \xi\,(\Delta x)$$

ou

$$f'(x_i) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Equação 1 - Aproximação *foward* da derivada primeira [2]

A aproximação centrada da derivada primeira é dada por:

$$\left.\frac{df}{dx}\right]_{x_i} = \frac{f(x_{i+1}) - f(x_{i-1})}{2\,\Delta x} - \xi\,(\Delta x^2)$$

ou

$$f'(x_i) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\,\Delta x}$$

Equação 2 - Aproximação *central* da derivada primeira [2]

Ainda, para a aproximação centrada a derivada de segunda ordem fica da seguinte maneira:

$$\left.\frac{d^2f}{dx^2}\right]_{x_i} = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{\Delta x^2} - \xi\,(\Delta x^2)$$

ou

$$f''(x_i) = \frac{f(x + \Delta x) + 2f(x) - f(x - \Delta x)}{\Delta x^2}$$

Equação 3 - Aproximação *central* da derivada segunda [2]

De acordo com [2], a equação do problema de fluxo unidimensional em regime transiente a ser resolvido:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\,\frac{\partial P}{\partial r}\right) = \frac{\phi\mu c_t}{k}\,\frac{\partial P}{\partial t}$$

Equação 4 – Equação da difusividade para fluxo radial em regime transiente - analítico [2]

Onde $\eta = {}^{k}/_{\phi\mu c_t}$

Desenvolvendo a equação acima pela regra da cadeia e, substituindo as derivadas pelas aproximações da Série de Taylor *forward*, para a derivada temporal, e centrada para as derivadas espaciais:

$$\frac{1}{\eta}\frac{\partial P}{\partial t} = \frac{1}{r}\left(\frac{\partial P}{\partial r} + r\,\frac{d^2 P}{dr^2}\right)$$

$$\frac{1}{\eta}\left(\frac{P_i - P_i^0}{\Delta t}\right) = \frac{1}{r_i}\left[\frac{P_{i+1}^{\theta} - P_i^{\theta}}{2\,\Delta r}\right] + \frac{P_{i+1}^{\theta} - 2P_i^{\theta} + P_{i-1}^{\theta}}{\Delta r^2}$$

Equação 5 – Equação da difusividade para fluxo radial em regime transiente – discreto

O termo θ identifica o tempo t_0 e t no qual o termo difusivo é calculado[4].

$$\boxed{P_{i,j}^{\theta} = \theta P_{i,j} + (1-\theta)P_{i,j}^{0}}$$

- Se $\theta = 0$ esquema de discretização totalmente explícita;
- Se $\theta = 1$ esquema de discretização totalmente implícita;
- Se $0 < \theta < 1$ esquema de discretização parcialmente implícita.

Figura 3 – Opções de esquema de discretização [4]

3.2 Esquema explícito

As informações para o cálculo da variável dependente (pressão) no nó i são provenientes do tempo anterior (θ = 0, condição inicial):

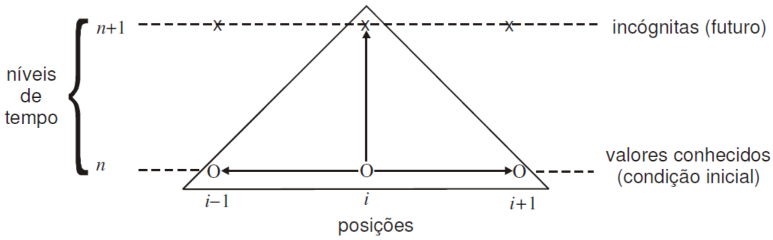


Figura 4 – Discretização usando o método explícito[2]

Então:

$$\frac{1}{\eta}\left(\frac{P_i - P_i^0}{\Delta t}\right) = \frac{1}{r_i}\left[\frac{P_{i+1}^{0} - P_i^{0}}{2\,\Delta r}\right] + \frac{P_{i+1}^{0} - 2P_i^{0} + P_{i-1}^{0}}{\Delta r^2}$$

Equação 6 – Equação da difusividade para fluxo radial em regime transiente – esquema explícito

Resolvendo a equação para única incógnita P_i

$$P_i = \left(\frac{\eta\Delta t}{2\,r_i\,\Delta r} + \frac{\eta\Delta t}{\Delta r^2}\right)P_{i+1}^0 + \left(\frac{\eta\Delta t}{\Delta r^2} - \frac{\eta\Delta t}{2\,r_i\,\Delta r}\right)P_{i-1}^0 + \left(1 - \frac{2\,\eta\Delta t}{\Delta r^2}\right)P_i^0$$

Equação 7 –Pressão no nó i- regime transiente – esquema explícito

Onde os termos entre parênteses são chamados de "transmissibilidade do nó" (τ), assim:

$$P_i = \tau_{i+1}\,P_{i+1}^0 + \tau_{i-1}\,P_{i-1}^0 + \tau_i\,P_i^0$$

Onde

- $\tau_{i+1} = \left(\frac{\eta \Delta t}{2\, r_i\, \Delta r} + \frac{\eta \Delta t}{\Delta r^2}\right)$
- $\tau_{i-1} = \left(\frac{\eta \Delta t}{\Delta r^2} - \frac{\eta \Delta t}{2\, r_i\, \Delta r}\right)$
- $\tau_i = \left(1 - \frac{2\, \eta \Delta t}{\Delta r^2}\right)$

[2] Na fronteira r = r_w (r → 0 ou i =0), a condição de contorno é:

$$\lim_{r \to 0}\left(r\,\frac{\partial P}{\partial r}\right) = -\,\frac{q_0 B_0 \mu}{2\pi k h}$$

Substituindo na equação original do problema

$$\frac{1}{\eta}\,\frac{\partial P}{\partial t} = \frac{1}{r}\left(\frac{\partial P}{\partial r} + r\,\frac{d^2 P}{dr^2}\right) = -\frac{1}{r^2}\,\frac{q_0 B_0 \mu}{2\pi k h} + \frac{d^2 P}{dr^2}$$

Equação 8 – Equação do problema de fluxo radial unidimensional para r~0

Então

$$\frac{1}{\eta}\left(\frac{P_i - P_i^0}{\Delta t}\right) = -\frac{1}{r_i^2}\,\frac{q_0 B_0 \mu}{2\pi k h} + \frac{P_{i+1}^0 - 2P_i^0 + P_{i-1}^0}{\Delta r^2}$$

ou

$$\frac{1}{\eta}\left(\frac{P_0 - P_0^0}{\Delta t}\right) = -\frac{1}{r_0^2}\,\frac{q_0 B_0 \mu}{2\pi k h} + \frac{P_1^0 - 2P_0^0 + P_{-1}^0}{\Delta r^2}$$

Equação 9 – Equação do problema de fluxo radial unidimensional – explícito para r~0

De acordo com [5] e [4], para resolver P_{-1} fictício, utiliza-se a condição de simetria no nó i = 0

$$\left.\frac{dP}{dr}\right]_{r=0} = \frac{{P_1}^0 - {P_{-1}}^0}{2\,\Delta r} = 0$$

então

$${P_1}^0 = {P_{-1}}^0$$

Portanto, substituindo a igualdade acima na equação 9

$$P_0 = \left(\frac{2\,\eta \Delta t}{\Delta r^2}\right) P_1^0 + \left(1 - \frac{2\,\eta \Delta t}{\Delta r^2}\right) P_0^0 - \frac{\eta \Delta t}{r_0^2}\,\frac{q_0 B_0 \mu}{2\pi k h}$$

Equação 10 – Equação do problema de fluxo radial unidimensional – explícito para r~0

ou

$$P_0 = \tau_1\, P_1^0 + \tau_0\, P_0^0 + B_0$$

- $\tau_1 = \frac{2\eta\Delta t}{\Delta r^2}$
- $\tau_0 = \left(1 - \frac{2\,\eta\Delta t}{\Delta r^2}\right)$
- $B_0 = -\frac{\eta\Delta t}{r_0^2}\frac{q_0 B_0 \mu}{2\pi kh}$

[2] Na fronteira r = r_e (r → ∞), a condição de contorno é a própria pressão inicial (condição inicial de pressão em t =0).

3.3 Esquema implícito

[4]As informações de cálculo da variável dependente (pressão) no nó i são provenientes inteiramente do tempo atual (θ = 0, as equações resultantes que compõem o sistema linear serão todas acopladas):

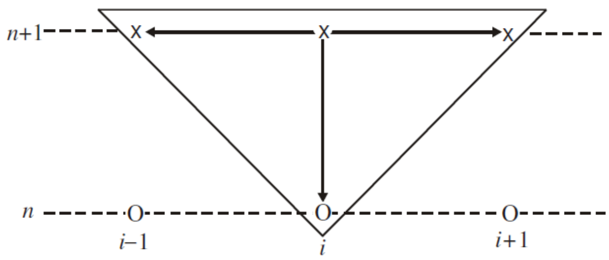


Figura 5 - Discretização usando o método implícito [2]

Então:

$$\frac{1}{\eta}\left(\frac{P_i - P_i^0}{\Delta t}\right) = \frac{1}{r_i}\left[\frac{P_{i+1} - P_i}{2\,\Delta r}\right] + \frac{P_{i+1} - 2P_i + P_{i-1}}{\Delta r^2}$$

Equação 11 – Equação da difusividade para fluxo radial em regime transiente – esquema implícito

Resolvendo a equação para a incógnita P_i

$$\left(1 + \frac{2\,\eta\Delta t}{\Delta r^2}\right)P_i = \left(\frac{\eta\Delta t}{2\,r_i\,\Delta r} + \frac{\eta\Delta t}{\Delta r^2}\right)P_{i+1} + \left(\frac{\eta\Delta t}{\Delta r^2} - \frac{\eta\Delta t}{2\,r_i\,\Delta r}\right)P_{i-1} + P_i^0$$

Equação 12 –Equação acoplada da pressão no nó i - regime transiente – esquema implícito

Ou

$$\tau_i\ P_i = \tau_{i+1}\ P_{i+1} + \tau_{i-1}\,P_{i-1} + P_i^0$$

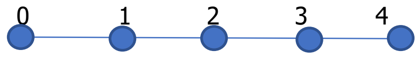
- $\tau_{i+1} = \left(\frac{\eta\Delta t}{2\,r_i\,\Delta r} + \frac{\eta\Delta t}{\Delta r^2}\right)$
- $\tau_{i-1} = \left(\frac{\eta\Delta t}{\Delta r^2} - \frac{\eta\Delta t}{2\,r_i\,\Delta r}\right)$
- $\tau_i = \left(1 + \frac{2\,\eta\Delta t}{\Delta r^2}\right)$

Portanto é possível organizar a equação 12 da seguinte forma (em termos das transmissibilidades):

$$-\tau_{i-1}\,P_{i-1} + \tau_i\,P_i - \tau_{i+1}\,P_{i+1} = P_i^0$$

Equação 13 –Equação acoplada da pressão no nó i - regime transiente – esquema implícito

Considerando, por exemplo uma malha com 5 pontos:



$$0 \;+\; \tau_0\,P_0 \;-\; \tau_1\,P_1 = P_0^0$$

$$-\,\tau_0\,P_0 \;+\tau_1\,P_1 \;-\; \tau_2\,P_2 = P_1^0$$

$$-\,\tau_1\,P_1 \;+\tau_2\,P_2 \;-\; \tau_3\,P_3 = P_2^0$$

$$-\,\tau_2\,P_2 \;+\tau_3\,P_3 \;-\; \tau_4\,P_4 = P_3^0$$

$$-\,\tau_3\,P_3 \;+\tau_4\,P_4 \;-\; 0 \;= P_4^0$$

Em outras palavras:

$$\begin{bmatrix} \tau_0 & -\tau_1 & 0 & 0 & 0 \\ -\tau_0 & \tau_1 & -\tau_2 & 0 & 0 \\ 0 & -\tau_1 & \tau_2 & -\tau_3 & 0 \\ 0 & 0 & -\tau_2 & \tau_3 & -\tau_4 \\ 0 & 0 & 0 & -\tau_3 & \tau_4 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} \begin{bmatrix} P_0^0 \\ P_1^0 \\ P_2^0 \\ P_3^0 \\ P_4^0 \end{bmatrix}$$

Portanto, para uma malha de n pontos:

$$[\tau]_{n\,x\,n}\;\;[P\,]_{n\,x\,1}\;\; = \;[P^0]_{n\,x1}$$

O campo de pressões a ser calculado será:

$$[P\,]_{n\,x\,1}\;\; = \;[\tau]_{n\,x\,n}{}^{-1}\,[P^0]_{n\,x1}$$

[2] Na fronteira r = r_w (r → 0 ou i =0), a condição de contorno é:

$$\lim_{r\to 0}\left(r\,\frac{\partial P}{\partial r}\right) = -\,\frac{q_0 B_0 \mu}{2\pi k h}$$

Substituindo na equação original do problema

$$\frac{1}{\eta}\,\frac{\partial P}{\partial t} = \frac{1}{r}\left(\frac{\partial P}{\partial r} + r\,\frac{d^2 P}{dr^2}\right) = -\frac{1}{r^2}\,\frac{q_0 B_0 \mu}{2\pi k h} + \frac{d^2 P}{dr^2}$$

Equação 14 – Equação do problema de fluxo radial unidimensional para r~0

Então

$$\frac{1}{\eta}\left(\frac{P_i - P_i^0}{\Delta t}\right) = -\frac{1}{r_i^2}\,\frac{q_0 B_0 \mu}{2\pi k h} + \frac{P_{i+1} - 2P_i + P_{i-1}}{\Delta r^2}$$

ou

$$\frac{1}{\eta}\left(\frac{P_0 - P_0^0}{\Delta t}\right) = -\frac{1}{r_0^2}\frac{q_0 B_0 \mu}{2\pi k h} + \frac{P_1 - 2P_0 + P_{-1}}{\Delta r^2}$$

Equação 15 – Equação do problema de fluxo radial unidimensional – numérico para r~0

De acordo com [5] e [4], para resolver P_{-1} fictício, utiliza-se a condição de simetria no nó i = 0

$$\left.\frac{dP}{dr}\right]_{r=0} = \frac{{P_1}^0 - {P_{-1}}^0}{2\,\Delta r} = 0$$

então

$${P_1}^0 = {P_{-1}}^0$$

Portanto, substituindo a igualdade acima na equação 15

$$\left(1 + \frac{2\,\eta\Delta t}{\Delta r^2}\right)P_0 = \left(\frac{2\eta\Delta t}{\Delta r^2}\right)P_1 + P_0^0 - \frac{\eta\Delta t}{r_0^2}\frac{q_0 B_0 \mu}{2\pi k h}$$

Equação 16 – Equação do problema de fluxo radial unidimensional – implícito para r~0

ou

$$\tau_0\ P_0 - \tau_1\ P_1 = B_0$$

- $\tau_1 = \frac{2\eta\Delta t}{\Delta r^2}$
- $\tau_0 = \left(1 + \frac{2\,\eta\Delta t}{\Delta r^2}\right)$
- $B_0 = P_0^0 - \frac{\eta\Delta t}{r_0^2}\frac{q_0 B_0 \mu}{2\pi k h}$

De forma análoga na fronteira r = r_e (r → ∞), fazendo ($P_{n+1} = P_{n-1}$) na equação 12

$$\left(1 + \frac{2\,\eta\Delta t}{\Delta r^2}\right)P_n = \left(\frac{2\eta\Delta t}{\Delta r^2}\right)P_{n-1} + P_n^0$$

Equação 17 – Equação do problema de fluxo radial unidimensional – implícito para r~r_e

$$\tau_n\ P_n - \tau_{n-1}\ P_{n-1} = P_n^0$$

- $\tau_{n-1} = \frac{2\eta\Delta t}{\Delta r^2}$
- $\tau_n = \left(1 + \frac{2\,\eta\Delta t}{\Delta r^2}\right)$

4. Resultados

Para a implementação numérica, foi utilizada a linguagem de programação em Python 3.8 (gerenciador *Anaconda*). O problema analítico foi implementado junto ao código para comparação dos resultados.

Os códigos estão nos anexos.

4.1 Analítico

O resultado do problema analítico (Exercício 1 - Lista 1) está reproduzido na imagem abaixo.

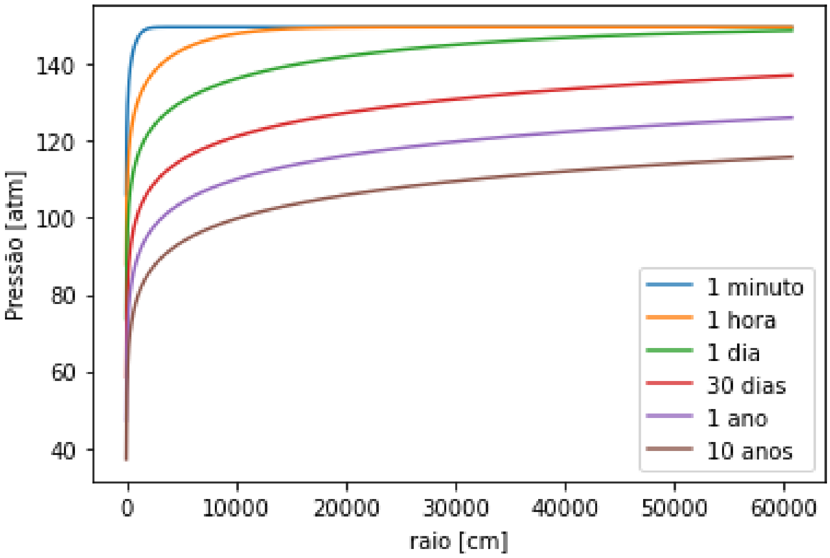


Figura 6 – Caso analítico (re = 609.6m)

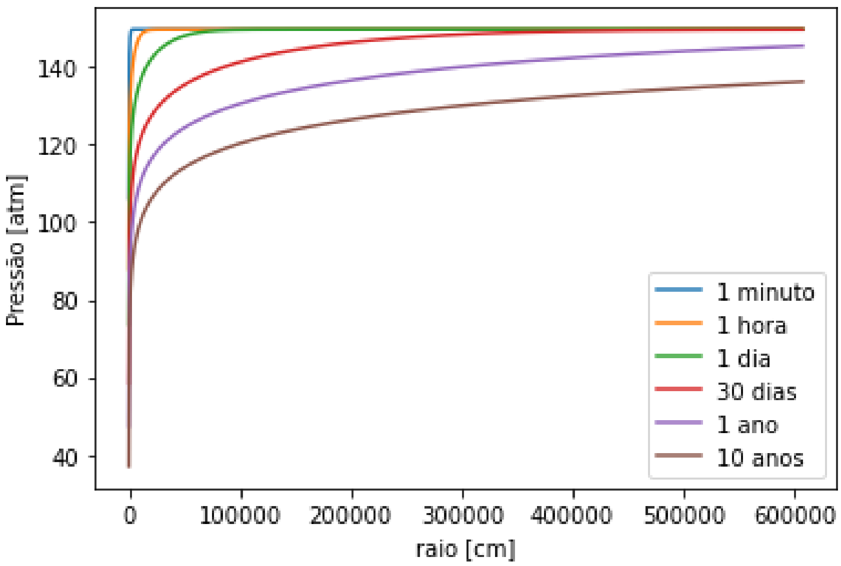


Figura 7 - Caso analítico (re = 6 km)

4.2 Numérico – esquema explícito

Os resultados numéricos no esquema explícito no regime transiente estão apresentados nas figuras abaixo:

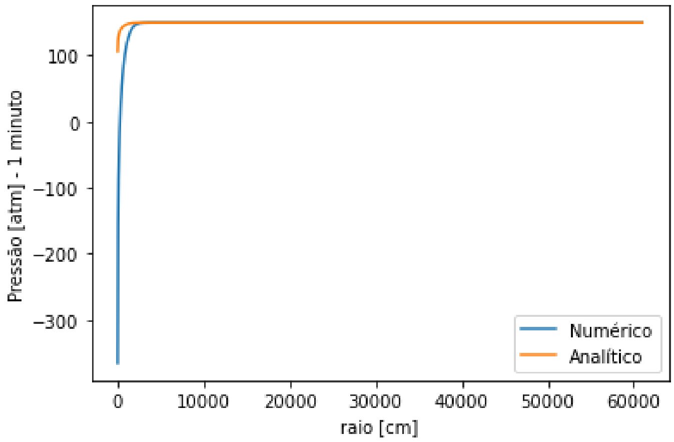


Figura 8 -Numérico - explícito – 1 minuto (re = 609.6m)

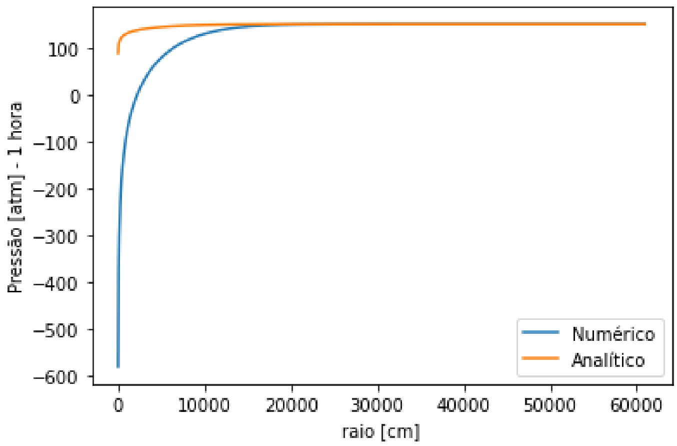


Figura 9 - Numérico - explícito – 1 hora (re = 609.6m)

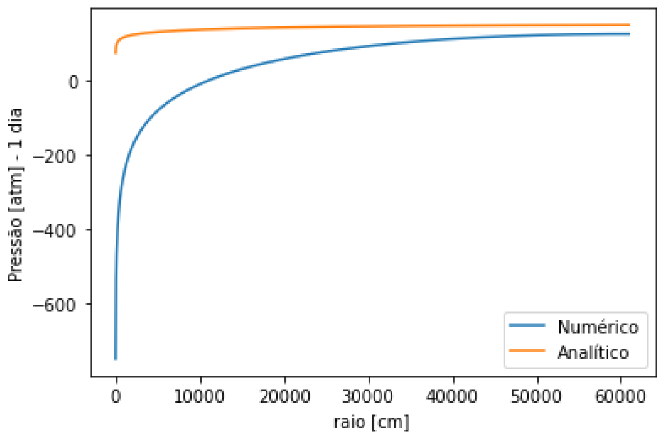


Figura 10 - Numérico - explícito – 1 dia (re = 609.6m)

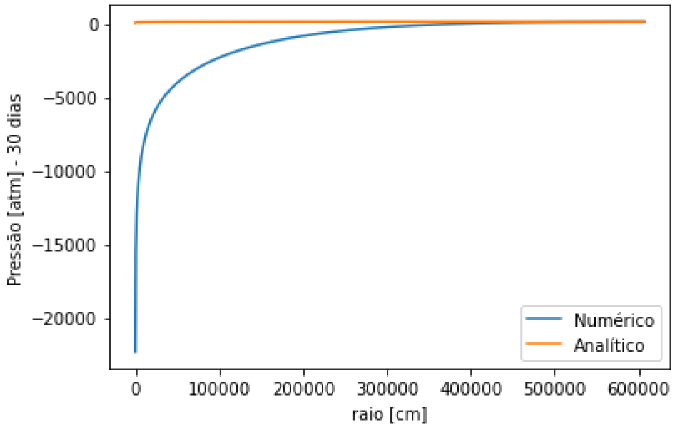


Figura 11 - Numérico - explícito – 30 dias (re = 6 km)

O programa possui limitações de tempo de rodada para análises maiores que 30 dias (4 horas de simulação). Por isso, não foram apresentados os resultados numéricos para 1 ano e 10 anos.

4.3 Numérico – esquema implícito

Os resultados numéricos no esquema implícito no regime transiente estão apresentados nas figuras abaixo:

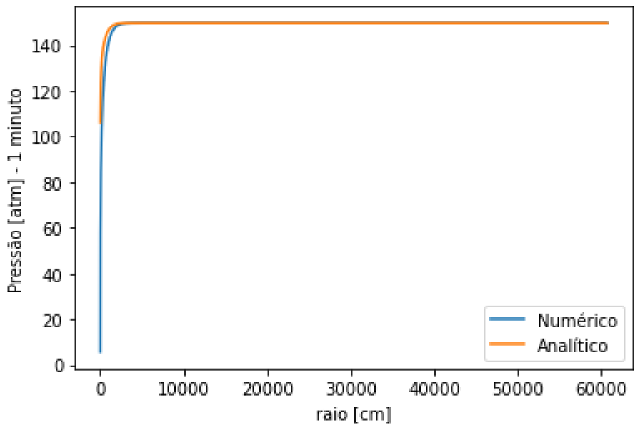


Figura 12- Numérico - implícito – 1 minuto (re = 609.6m)

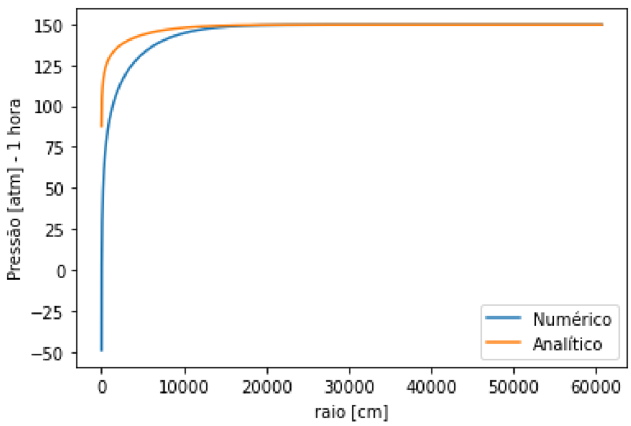


Figura 13 - Numérico - implícito – 1 hora (re = 609.6m)

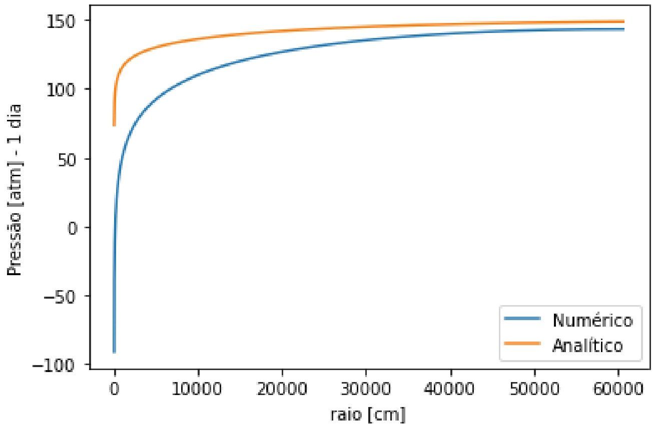


Figura 14 - Numérico - implícito – 1 dia (re = 609.6m)

O programa possui limitações de tempo de rodada para análises maiores que 1 dia (4 horas de simulação). Por isso, não foram apresentados os resultados numéricos para 30 dias, 1 ano e 10 anos.

5. Conclusão

Os resultados mostram que, para o esquema explícito os resultados tendem a divergir com o tempo. Já no esquema implícito o erro é menor, há uma tendência de convergir para os resultados analíticos.

Conclui-se que o esquema implícito é mais estável que o explícito pelo fato de os coeficientes das variáveis dependentes serem positivos.

O programa Python é um bom compilador (ou interpretador) para uma primeira implementação de análise numérica, no entanto, pode ter limitações de tempo de simulação para análises mais robustas.

Vale a pena reproduzir os exercícios com outros interpretadores como o *Wolfram Mathematica*.

6. Referências bibliográficas

[1] Lista 2
[2] Adalberto J. Rosa, Renato de S.Carvalho e José A. Daniel Xavier, Engenharia de Reservatórios de Petróleo
[3] Aula 5 - Séries de Taylor
[4] Aula 6 – Método das diferenças finitas
[5] Osik M,Necati. Heat Conduction

Anexos

Analítico

```
#Resolução do Problema 1 - Lista 1:Fluxo radial unidimensional - analítico

import math as m
import matplotlib.pyplot as plt
import scipy.special as sc

#Dados de entrada
re = 60690                      #raio externo [cm]
rw = 8.9                        #raio do poço [cm]
h = 304.8                       #espessura do reservatório [cm]
pi = 149.7                      #pressão inicial [atm]
q0 = 5180                       #vazão [cm^3 std/ s]
k = 0.15                        #permeabilidade [Darcy]
phi= 0.2                        #porosidade [x100%]
ct = 0.0002204                  #compressibilidade total [1/atm]
mu = 0.33                       #viscosidade [cp]
Bo = 1.5                        # fato volume formação [m^3/m^3 std]
g = 1.78108                     #gamma para p em rw


C = Bo*q0*mu/(4*m.pi*k*h)       #parte constante da equação 3.225 de Rosa et al.

t1 = 60                          #[s] t = 1 min
t2 = 60*60                       #[s] t = 1 hora
t3 = 60*60*24                    #[s] t = 1 dia
t4 = 60*60*24*30                 #[s] t = 30 dias
t5 = 60*60*24*365                #[s] t = 1 ano
t6 = 60*60*24*365*10             #[s] t = 10 anos


#Definindo a lista de raios - r[0] = 0 metros. A correção será aplicada no cálculo da pressão.
r = []                                                  #lista a ser montada
n = 1000                                                #quantidade de termos

for i in range(n+1):
    r = r + [i*(re-rw)/n]

print(r[-1])                                            #teste para saber o último valor: r[n+1]

#Perfil de pressão para t = 1min
p1 = []                                                 #lista a ser montada

for i in range(n+1):
    p1 = p1 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t1))]  #montagem da lista para p1@1min

p1[0]= pi+C*sc.expi(-phi*mu*ct*rw**2/(4*k*t1))          #correção para r[0] = rw

print(p1[-1])                                           #teste para saber o último valor: p1[n+1]

```

```
#Perfil de pressão para t = 1hora
p2 = []

for i in range(n+1):
    p2 = p2 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t2))]

p2[0]= pi+C*sc.expi(-phi*mu*ct*rw**2/(4*k*t2))

print(p2[-1])


#Prerfil de pressão para t = 1 dia
p3 = []

for i in range(n+1):
    p3 = p3 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t3))]

p3[0]= pi+C*sc.expi(-phi*mu*ct*rw**2/(4*k*t3))

print(p3[-1])

#Perfil de pressão para t = 30 dias
p4 = []

for i in range(n+1):
    p4 = p4 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t4))]

p4[0]= pi+C*sc.expi(-phi*mu*ct*rw**2/(4*k*t4))

print(p4[-1])

#Perfil de pressão para t = 1 ano
p5 = []

for i in range(n+1):
    p5 = p5 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t5))]

p5[0]= pi+C*sc.expi(-phi*mu*ct*rw**2/(4*k*t5))

print(p5[-1])

#Perfil de pressão para t = 10 anos
p6 = []

for i in range(n+1):
    p6 = p6 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t6))]

p6[0]= pi+C*sc.expi(-phi*mu*ct*rw**2/(4*k*t6))

print(p6[-1])

```

```
#Gráficos
plt.plot(r,p1, label="1 minuto")
plt.plot(r,p2, label="1 hora")
plt.plot(r,p3, label="1 dia")
plt.plot(r,p4, label="30 dias")
plt.plot(r,p5, label="1 ano")
plt.plot(r,p6, label="10 anos")
plt.xlabel('raio [cm]')
plt.ylabel('Pressão [atm]')
plt.legend()
plt.show()
```

Numérico -Explícito

```python
#Resolução do Problema 2 - Lista 2:Fluxo radial unidimensional - numérico
#Método explícito

import math as m
import scipy.special as sc
import matplotlib.pyplot as plt


#1 - Dados de entrada
re = 60960                          #raio externo [cm]
rw = 8.9                            #raio do poço [cm]
h = 304.8                           #espessura do reservatório [cm]
pi = 149.7                          #pressão inicial [atm]
q0 = 5180                           #vazão [cm^3 std/ s]
k = 0.15                            #permeabilidade [Darcy]
phi= 0.2                            #porosidade [x100%]
ct = 0.0002204                      #compressibilidade total [1/atm]
mu = 0.33                           #viscosidade [cp]
Bo = 1.5                            # fato volume formação [m^3/m^3 std]
g = 1.78108                         #gamma para p em rw


#2 - Cálculo das constantes

dt = 0.05                              #passo de tempo [s]
dr = 60.96                         #espassamento da malha [cm]

ni = k/(phi*mu*ct)                  #constante de difusividade hidráulica - eq.3.113 de Rosa et al.

#parcelas constantes da transmissibilidade do nó
tau_c1 = ni*dt/(dr**2)
tau_c2 = ni*dt/(2*dr)               #parcela que multiplica 1/ri no cálculo da pressão

#parcelas constantes B - relativo às pressões no poço
cc = Bo*q0*mu/(2*m.pi*k*h)


#3 - Definição da lista de raios - pontos i - que serão calculados o campos de pressão

r = []
n = int (re/dr)
for i in range(n+1):
    r += [i*dr]
r[0] = rw                           #correção r1 = 0 para r1 = rw

print(r[-1])                        #teste para saber o último valor: r[n+1]

Nr = len(r)                         #numero de pontos no espaço


#4 - Condições iniciais

t1 = 60*60*24 #*30                  #[s] tempo total de análise
Nt = int(t1/dt)                       #número de passos no tempo

Pi = []                             #campo de pressões iniciais
for i in range(Nr):                 #Pi = [pi for i in range(Nr)]
    Pi += [pi]

#5 - Cálculo do campos de pressões no tempo (numérico)

P = []                              #pressões no tempo presente
P_aux = []                          #pressões no tempo futuro
for i in range (Nt):
    if i == 0:
        P = Pi.copy()
        print('it =', i,'// t[s] =', round(i*dt, 2), '// horas =', round(i*dt/3600, 2), ' // dias =', round(i*dt/3600/24, 2),
              '// mes = ', round(i*dt/3600/24/30.2, 1), '// ano =', round(i*dt/3600/24/365.5, 1), '// pw[atm] =',round(P[0], 2))
    else:
        P = P_aux.copy()
        P_aux = []
        print('it =', i,'// t[s] =', round(i*dt, 2), '// horas =', round(i*dt/3600, 2), ' // dias =', round(i*dt/3600/24, 2),
              '// mes = ', round(i*dt/3600/24/30.2, 1), '// ano =', round(i*dt/3600/24/365.5, 1), '// pw[atm] =',round(P[0], 2))
    for j in range (Nr):
        if j == 0:
            P_aux += [2*tau_c1*P[j+1]+(1-2*tau_c1)*P[j] - (ni*dt/(r[j]**2))*cc]
        elif j == Nr-1:
            P_aux += [P_aux[j-1]]
        else:
            P_aux += [((1/r[j])*tau_c2 +tau_c1)*P[j+1] + (tau_c1-(1/r[j])*tau_c2)*P[j-1] + (1 - 2*tau_c1)*P[j]]


print (P_aux[-1])



#6 - Perfil de pressão  (analítico)
C = Bo*q0*mu/(4*m.pi*k*h)                #parte constante da equação 3.225 de Rosa et al.

p1 = []                                  #lista a ser montada

for i in range(Nr):
    p1 = p1 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*k*t1))]       #montagem da lista para p1@t1


print(p1[-1])                            #teste para saber o último valor: p1[n+1]


#7 - Gráficos
plt.plot(r, P_aux, label="Numérico")
plt.plot(r, p1, label="Analítico")
plt.xlabel('raio [cm]')
plt.ylabel('Pressão [atm] - 1 dia')
plt.legend()
plt.show()
```

Numérico – Implícito

```python
#Resolução do Problema 2 - Lista 2:Fluxo radial unidimensional - numérico
#Método implícito

import math as m
import scipy.special as sc
import matplotlib.pyplot as plt

#1 - Dados de entrada
re = 60690                                    #raio externo [cm]
rw = 8.9                                      #raio do poço [cm]
h = 304.8                                     #espessura do reservatório [cm]
pi = 149.7                                    #pressão inicial [atm]
q0 = 5180                                     #vazão [cm^3 std/ s]
kk = 0.15                                     #permeabilidade [Darcy]
phi= 0.2                                      #porosidade [x100%]
ct = 0.0002204                                #compressibilidade total [1/atm]
mu = 0.33                                     #viscosidade [cp]
Bo = 1.5                                      # fato volume formação [m^3/m^3 std]
g = 1.78108                                   #gamma para p em rw


#2 - Cálculo das constantes

dt = 15                                       #passo de tempo [s]
dr = 30.345                                    #espassamento da malha [cm]

ni = kk/(phi*mu*ct)                            #constante de difusividade hidráulica - eq.3.113 de Rosa et al.

#parcelas constantes da transmissibilidade do nó
tau_c1 = ni*dt/(dr**2)
tau_c2 = ni*dt/(2*dr)                          #parcela que multiplica 1/ri no cálculo da pressão


# parcelas constantes relativo ao pw
cc = Bo*q0*mu/(2*m.pi*kk*h)

#3 - Definição da lista de raios - pontos i - que serão calculados o campos de pressão
#A quantidade de nós, Nr, irá definir o tamanho da matriz das transmissibiliades - Nr x Nr

r = []
n = int (re/dr)
for i in range(n+1):
    r += [i*dr]
r[0] = rw                                     #correção r1 = 0 para r1 = rw

print(r[-1])                                  #teste para saber o último valor: r[n+1]

Nr = len(r)                                   #numero de pontos no espaço


#4 - Condições iniciais

t1 = 60 #*60*24                                 #[s] tempo total de análise
Nt = int(t1/dt)                                #número de passos no tempo

Pi = []                                       #campo de pressões iniciais
for i in range(Nr):                           #Pi = [pi for i in range(Nr)]
    Pi += [pi]
#5 - Montagem da matriz de transmissibilidades

#matriz zerada
tr = []
for i in range (Nr):
    tr += [[]]
for i in range (Nr):
    tr[i] += [0 for j in range(Nr)]                  #matriz = [[0 for i in range(Nr)] for j in range (Nr)].Montagem de ma

#matriz tri - diagonal
for i in range (Nr):
    for j in range (Nr):
        if i==j:                                     #parcela diagonal
            tr [i][j] = 1 + 2*tau_c1
        elif j-i == 1:                               #diagonal superior
            if i == 0:
                tr [i][j] = -2*tau_c1
            else:
                tr [i][j] = -(tau_c2*(1/r[i]) + tau_c1)
        elif j-i == -1:                              #diagonal inferior
             if i == (Nr-1):
                tr [i][j] = -2*tau_c1
             else:
                tr [i][j] = -(tau_c1 - tau_c2*(1/r[i]))



#6 - Inversão da matriz de transmissibiliade
#https://numpy.org/doc/stable/reference/generated/numpy.linalg.inv.html?highlight=inv#numpy.linalg.inv


from numpy.linalg import inv
tr_inv = inv(tr)

#7 - Cálculo do campos de pressões no tempo (numérico)

P = []                                               #pressões no tempo presente
P_aux = []                                           #pressões no tempo futuro

for k in range (Nt):
    if k == 0:
        P = Pi.copy()
        print('it =', k,'// t[s] =', round(k*dt, 2), '// horas =', round(k*dt/3600, 2), ' // dias =', round(k*dt/3600/24, 2),
              '// mes = ', round(k*dt/3600/24/30.2, 1), '// ano =', round(k*dt/3600/24/365.5, 1), '// pw[atm] =',round(P[0], 2))
    else:
        P = P_aux.copy()
        P_aux = []
        print('it =', k,'// t[s] =', round(k*dt, 2), '// horas =', round(k*dt/3600, 2), ' // dias =', round(k*dt/3600/24, 2),
              '// mes = ', round(k*dt/3600/24/30.2, 1), '// ano =', round(k*dt/3600/24/365.5, 1), '// pw[atm] =',round(P[0], 2))
    for i in range (Nr):
        P_aux += [0]
        for j in range (Nr):
            if j == 0:
                P_aux [i] = P_aux[i] + tr_inv[i][j]*(P[j] - (ni*dt/(r[j]**2))*cc)
            else:
                P_aux [i] = P_aux[i] + tr_inv[i][j]*P[j]

print (P_aux[-1])
```

```
#8 - Perfil de pressão (analítico)
C = Bo*q0*mu/(4*m.pi*kk*h)                                      #parte constante da equação 3.225 de Rosa et al.

p1 = []                                                         #lista a ser montada

for i in range(Nr):
    p1 = p1 + [pi+C*sc.expi(-phi*mu*ct*r[i]**2/(4*kk*t1))]      #montagem da lista para p1@t1


print(p1[-1])                                                   #teste para saber o último valor: p1[n+1]


#7 - Gráficos
plt.plot(r, P_aux, label="Numérico")
plt.plot(r, p1, label="Analítico")
plt.xlabel('raio [cm]')
plt.ylabel('Pressão [atm] - 1 minuto')
plt.legend()
plt.show()
```