Ítalo Gonçalves Sant'Ana Paiva 10853310

PME3380 - Modelagem
Lista B

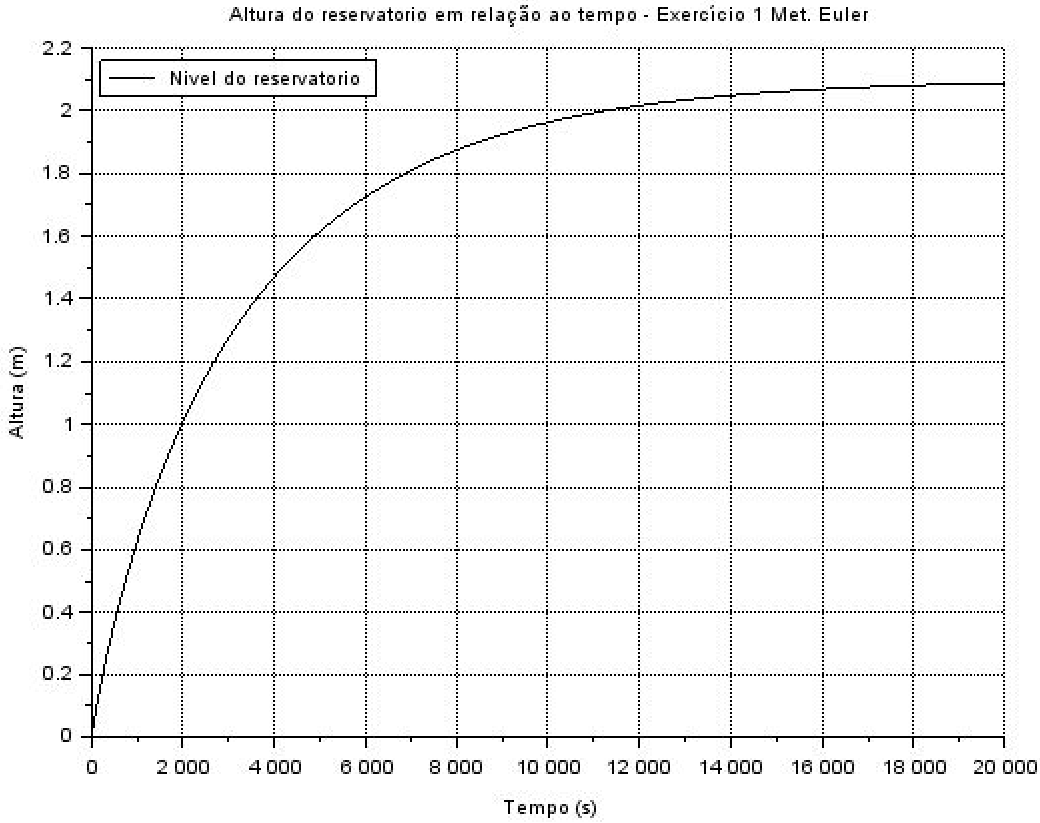
São Paulo

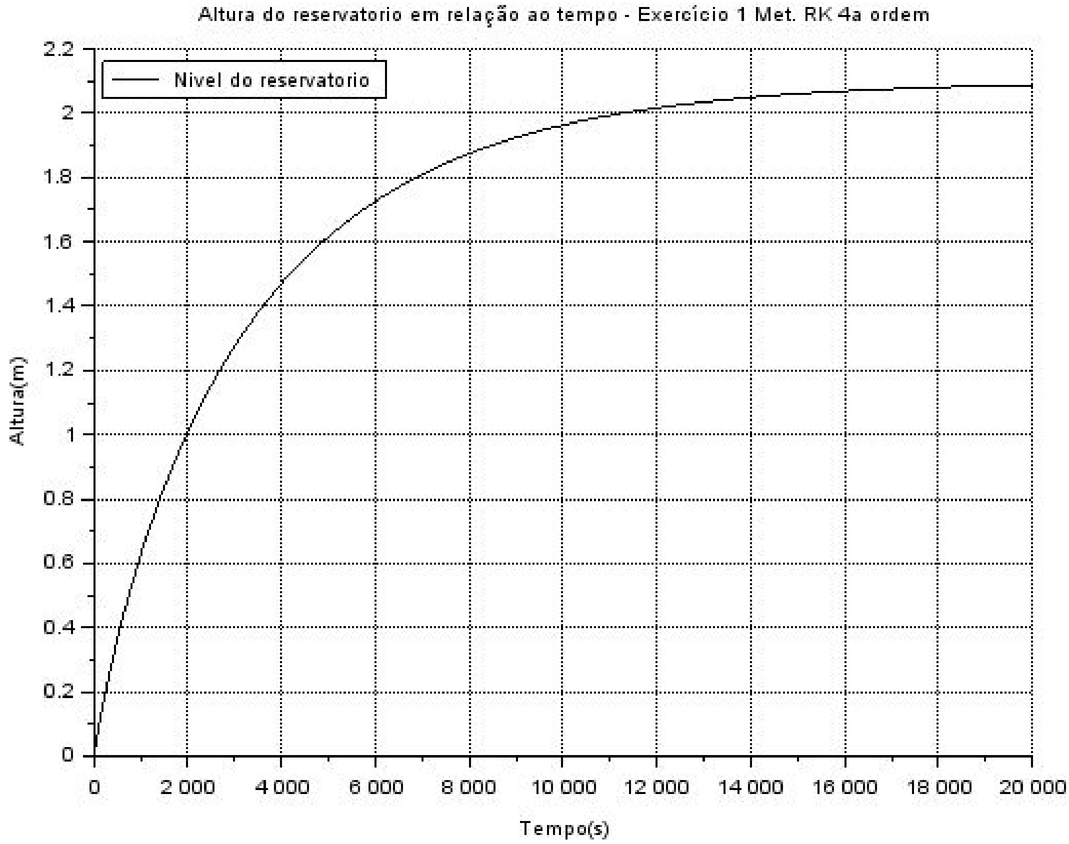
2020

Sumário

1 Exercício 1

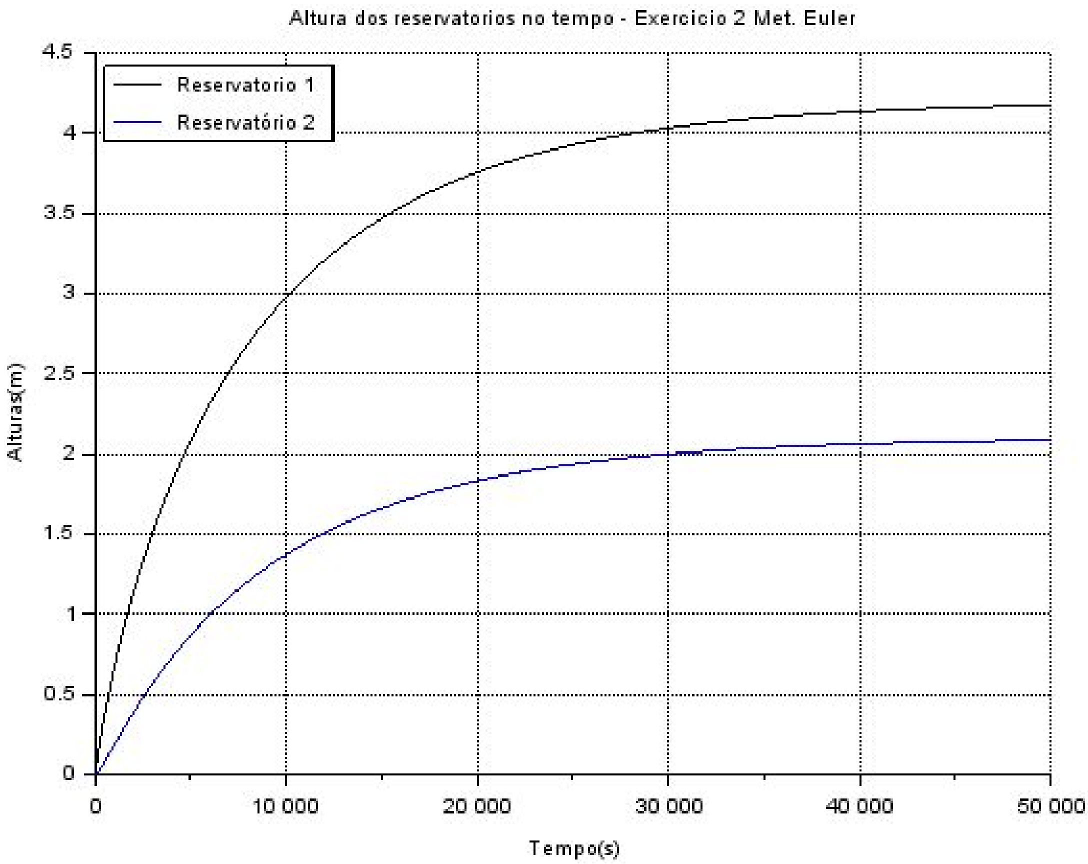
Para o Exercício 1, foi considerado altura inicial nula e um tempo final suficientemente grande para que se possa oberservar o fenômeno. A seguir, a resolução do problema com 1 Tanque será resolvido pelo método de Euler e pelo método de Runge Kutta de 4ª ordem, respectivamente.

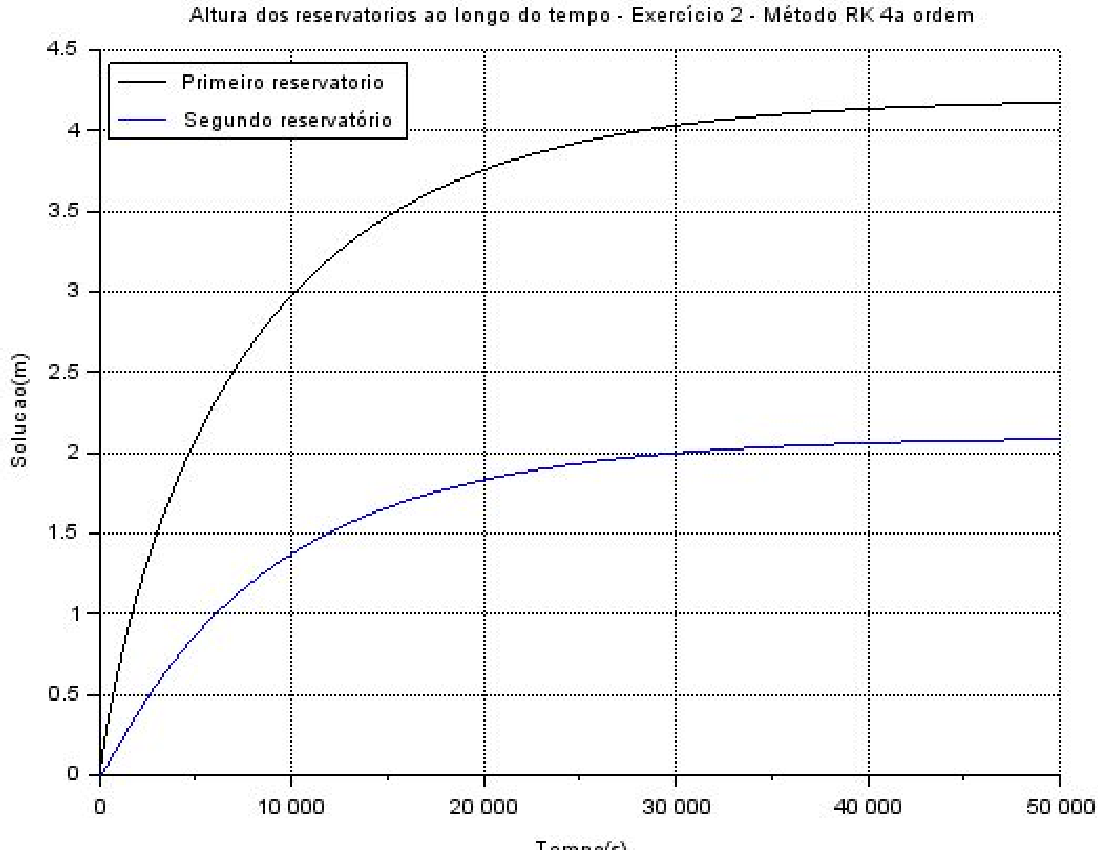



Altura do reservatorio em relação ao tempo - Exercicio 1 Met. RK 4a ordem
Nivel do reservatorio
Altura(m)
Tempo(s)
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
2 000
4 000
6 000
8 000
10 000
12 000
14 000
16 000
18 000
20 000

2 Exercício 2

No exercício 2 são considerados dois reservatórios. No segundo, a partir da imagem fornecida do exercício, foi considerado que sua área $S2$ é igual a $0,9 * S1$, assim, adotando alturas iniciais como zero e um tempo final suficientemente grande, foi possível observar o fenômeno.





Já ao que diz respeito à solução exata do problema, não foi possível obter valores para comparação com o método numérico, uma vez que não foi capaz de isolar a incógnita na equação obtida:

$$c_1 - \frac{t}{S} = \frac{2\left(Q \log\left(Q - \sqrt{C\, y(t)}\right) + \sqrt{C\, y(t)}\right)}{C}$$

A Código da simulação

```
EXERCCIO 1 - MTODO DE EULER
clear
//parametros de entrada
R = 2*10^8
Qe = 0.010247
rho = 1000
S = 10
g = 10

//funcao utilizada no problema, onde h é o parametro de entrada
function [hdot] = funcaoaltura(h)
hdot = (-sqrt(rho*g*h/R) + Qe)/S;
endfunction

// Instante inicial:
t(1) = 0;
// Instante final suficientemente grande para se observar bem o fenomeno:
tf = 20000;
// Passo de integracao (experimente alterar o passo):
dt = 0.1;
// Numero de passos):
n = round((tf - t(1))/dt);

// Condicao inicial da altura é zero, parte do tanque vazio
h(1) = 0;

// Integracao numerica usando o metodo de Euler:
// Comando for:
    for i = 1:n
        // Vetor de tempo:
        t(i+1) = t(i) + dt;
        // Solucao numerica:
        h(i+1) = h(i) + dt*funcaoaltura(h(i));
    // Termino do comando for:
    end

// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Plotando solucao numerica y versus vetor de tempo t
plot2d(t,h,[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Altura do reservatorio em relação ao tempo - Exercício 1 Met. Euler","Tempo (s)","Altura (m)","Nivel do reservatorio");
```

```
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends(T(4),[1,2],2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)

________________________________________________________________________
EXERCCIO 1 - METODO DE RUNGE KUTTA DE 4a ORDEM
clear
//parametros de entrada
R = 2*10^8
Qe = 0.010247
rho = 1000
S = 10
g = 10

//funcoes
function [hdot] = funcaoaltura(h)
hdot = (-sqrt(rho*g*h/R) + Qe)/S;
endfunction

// Instante inicial:
t(1) = 0;
// Instante final:
tf = 20000;
// Passo de integracao
dt = 0.1;
// Calculo de numero de passos:
n = round((tf - t(1))/dt);

// Condicao inicial:
h(1) = 0;

// Integracao numerica usando o metodo de Euler:
// Comando for:
    for i = 1:n
        // Vetor de tempo:
        t(i + 1) = t(i) + dt;
        // Solucao numerica:
        k1 = dt*funcaoaltura(h(i));
        k2 = dt*funcaoaltura(h(i) + k1/2);
        k3 = dt*funcaoaltura(h(i) + k2/2);
        k4 = dt*funcaoaltura(h(i) + k3);
        h(i+1)=h(i) + ((k1+2*k2+2*k3+k4)/6);

```

```
    // Termino do comando for:
end

// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Plotando solucao numerica y versus vetor de tempo t
plot2d(t,h,[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Altura do reservatorio em relação ao tempo - Exercício 1 Met. RK 4a
    ordem","Tempo(s)","Altura(m)","Nivel do reservatorio");
// Colocando uma legenda na parte superior esquerda da figura (parametro 2)
    :
legends(T(4),[1,2],2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)


EXERCCIO 2 - MTODO DE EULER
clear
//parametros de entrada
rho = 1000
g = 10
Qe = 0.010247
S1 = 10
S2 = 0.9*S1 // assumindo que a area de 2o reservatorio é 90% do 1o
Ra = 2*10^8
Rs = 2*10^8


//funcoes com parametros de entrada h1 e h2
function [hdot] = funcaoaltura1(h1, h2)
hdot = (-sqrt(rho*g*(h1 - h2)/Ra) + Qe)/S1;
endfunction

function [hdot] = funcaoaltura2(h1, h2)
hdot = (sqrt(rho*g*(h1 - h2)/Ra) - sqrt(rho*g*h2/Rs))/S2;
endfunction

// Instante inicial:
t(1) = 0;
// Instante final:
tf = 50000;
// Passo de integracao:
dt = 0.1;
// Cnumero de passos:
```

```
n = round((tf - t(1))/dt);

// Condicao inicial, as alturas partem de zero, ou seja, vazios:
h1(1) = 0;
h2(1) = 0;

// Integracao numerica usando o metodo de Euler:
// Comando for:
    for i = 1:n
        // Vetor de tempo:
        t(i + 1) = t(i) + dt;
        // Solucao numerica:
        h1(i+1) = h1(i) + dt*funcaoaltura1(h1(i),h2(i));
        h2(i+1) = h2(i) + dt*funcaoaltura2(h1(i),h2(i));
    // Termino do comando for:
    end

// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Plotando solucao numerica y versus vetor de tempo t
plot2d([t,t], [h1, h2], [1 2]);
// Usando a variavel do tipo 'lista':
T=list("Altura dos reservatorios no tempo - Exercicio 2 Met. Euler","Tempo(s)","Alturas(m)","Reservatorio 1", "Reservatório 2");
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends([T(4), T(5)], [1,2], 2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1), T(2), T(3));
// Colocando uma grade no grafico:
xgrid(1)
________________________________________________________________
EXERCICIO 2 - MTODO DE RUNGE KUTTA DE 4a ORDEM
clear
//parametros de entrada
rho = 1000
g = 10
Qe = 0.010247
S1 = 10
S2 = 0.9*S1 // assumindo que a area de 2o reservatorio é 90% do 1o
Ra = 2*10^8
Rs = 2*10^8

//funcoes
function [hdot] = funcaoaltura1(h1, h2)
hdot = (-sqrt(rho*g*(h1 - h2)/Ra) + Qe)/S1;
endfunction
```

```

function [hdot] = funcaoaltura2(h1, h2)
hdot = (sqrt(rho*g*(h1 - h2)/Ra) - sqrt(rho*g*h2/Rs))/S2;
endfunction

// Instante inicial:
t(1) = 0;
// Instante final:
tf = 50000;
// Passo de integracao:
dt = 0.5;
// numero de passos:
n = round((tf - t(1))/dt);

// Condicao inicial:
h1(1) = 0;
h2(1) = 0;

// Integracao numerica usando o metodo de Euler:
// Comando for:
    for i = 1:n
        // Vetor de tempo:
        t(i + 1) = t(i) + dt;
        // Solucao numerica:

        k1a = dt*funcaoaltura1(h1(i), h2(i));
        k1b = dt*funcaoaltura1(h1(i) + k1a/2, h2(i) + k1a/2);
        k1c = dt*funcaoaltura1(h1(i) + k1b/2, h2(i) + k1b/2);
        k1d = dt*funcaoaltura1(h1(i) + k1c/2, h2(i) + k1c/2);
        h1(i + 1) = h1(i) + ((k1a + 2*k1b + 2*k1c + k1d)/6);

        k2a = dt*funcaoaltura2(h1(i), h2(i));
        k2b = dt*funcaoaltura2(h1(i) + k2a/2, h2(i) + k2a/2);
        k2c = dt*funcaoaltura2(h1(i) + k2b/2, h2(i) + k2b/2);
        k2d = dt*funcaoaltura2(h1(i) + k2c/2, h2(i) + k2c/2);
        h2(i+1) = h2(i) + ((k2a + 2*k2b + 2*k2c + k2d)/6);
    // Termino do comando for:
    end

// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Plotando solucao numerica y versus vetor de tempo t
plot2d([t,t], [h1, h2], [1 2]);
// Usando a variavel do tipo 'lista':
T=list("Altura dos reservatorios ao longo do tempo - Exercício 2 - Método
    RK 4a ordem","Tempo(s)","Solucao(m)","Primeiro reservatorio", "Segundo
    reservatório");
```

```
// Colocando uma legenda na parte superior esquerda da figura (parametro 2)
    :
legends([T(4), T(5)], [1,2], 2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1), T(2), T(3));
// Colocando uma grade no grafico:
xgrid(1)
```