Kevin Chu NUSP: 10705908

PME3380-Modelagem de Sistemas Dinâmicos
Lista B

Exercício 1

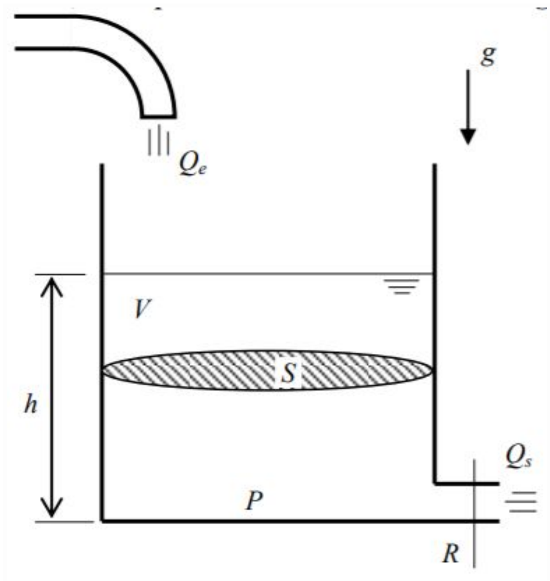

Figura 1 - Reservatório com água

Equação diferencial que modela o problema:

$$\dot{h} = \left(-\sqrt{\frac{\rho g h}{R}} + Q_e \right) \frac{1}{S}$$

Implementando um programa no Scilab que resolva numericamente a equação acima, obteve-se os seguintes resultados:

Resolvendo a equação diferencial pelo método de Euler:

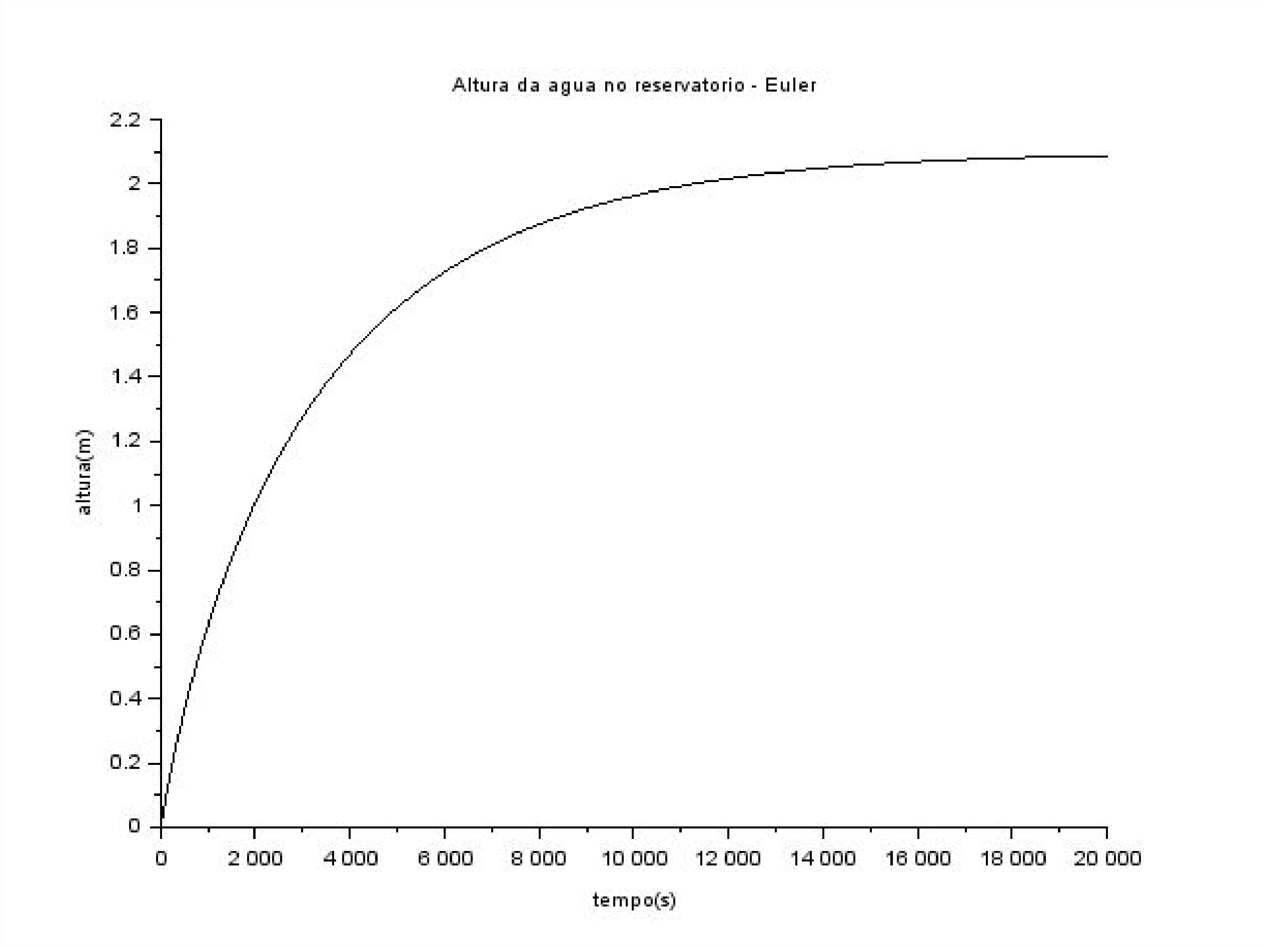

Gráfico 1 - Altura da água pelo tempo em um reservatório (Euler)

Resolvendo a equação diferencial pelo método de Runge-Kutta:

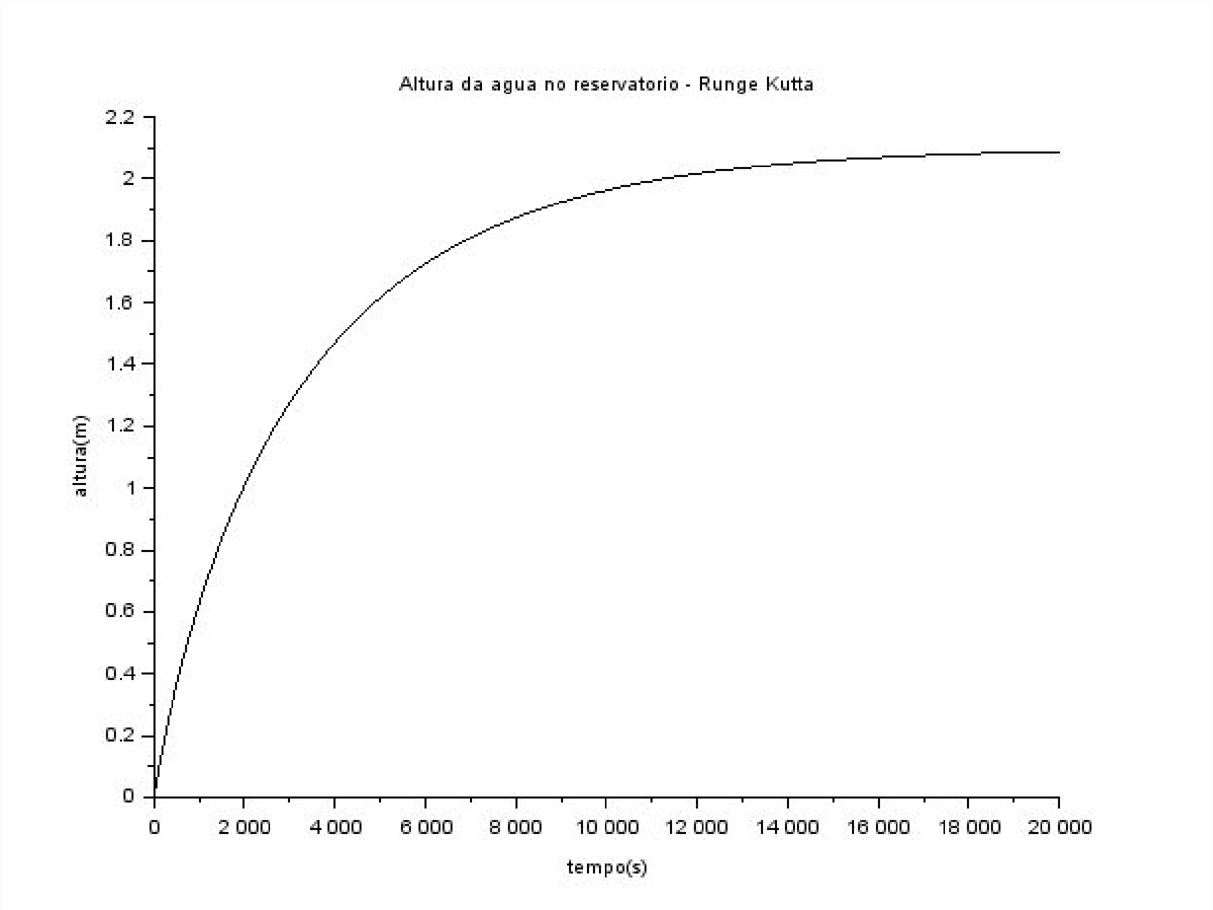

Gráfico 2 - Altura da água pelo tempo em um reservatório (Runge Kutta)

Exercício 2

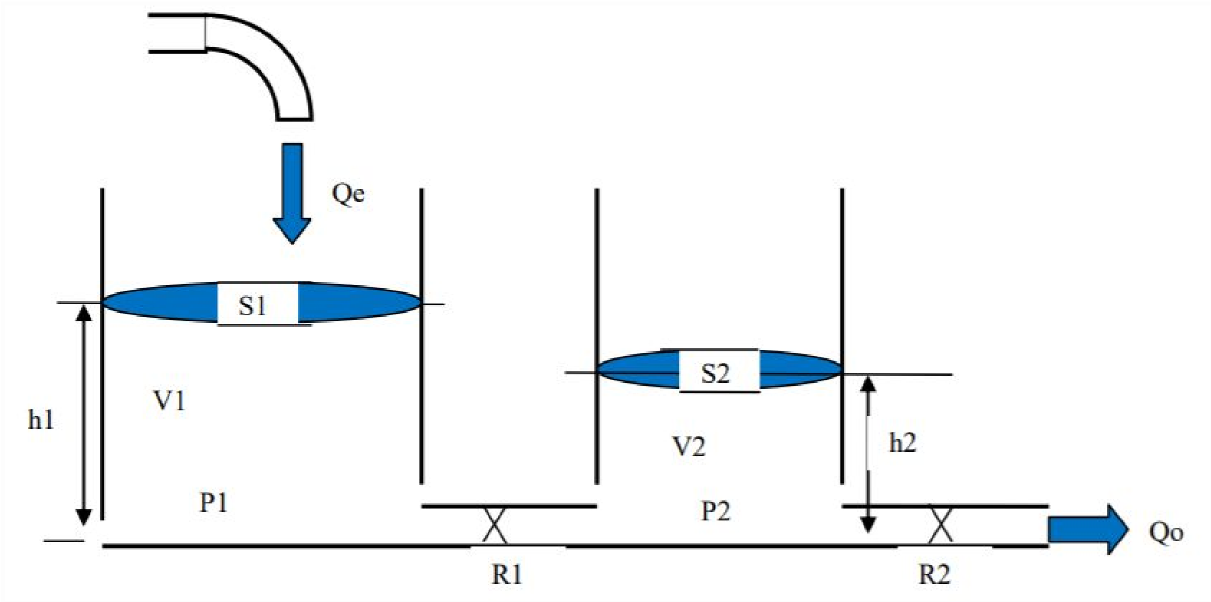

Figura 2 - Dois reservatórios com água

Equações diferenciais que modelam o problema:

$$\begin{cases} \dot{h}_1 = \left[Q_e - \sqrt{\dfrac{\rho g}{R_a}(h_1 - h_2)} \right] \dfrac{1}{S_1} \\ \dot{h}_2 = \left[\sqrt{\dfrac{\rho g}{R_a}(h_1 - h_2)} - \sqrt{\dfrac{\rho g}{R_s} h_2} \right] \dfrac{1}{S_2} \end{cases}$$

Desenvolvendo um programa no Scilab que resolva numericamente o sistema de equações acima, obteve-se os seguintes resultados:

Resolvendo o sistema pelo método de Euler:

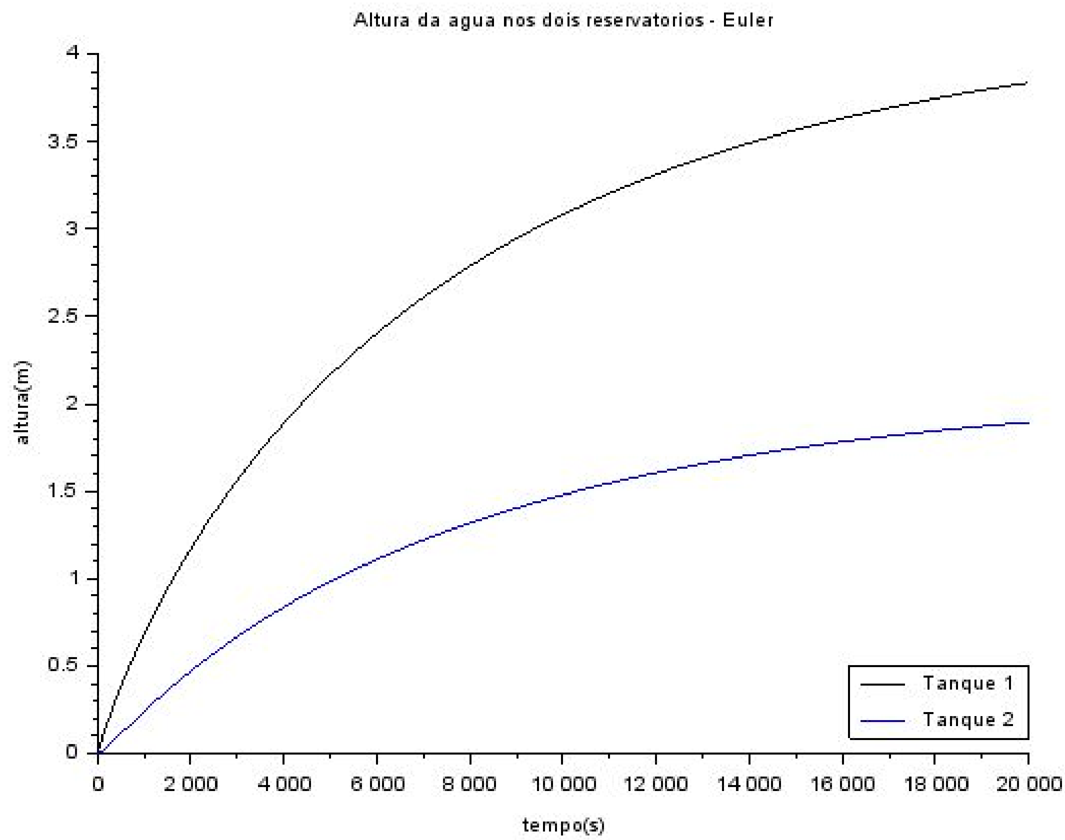

Gráfico 3 - Altura da água pelo tempo nos dois reservatórios (Euler)

Resolvendo o sistema por Runge Kutta:

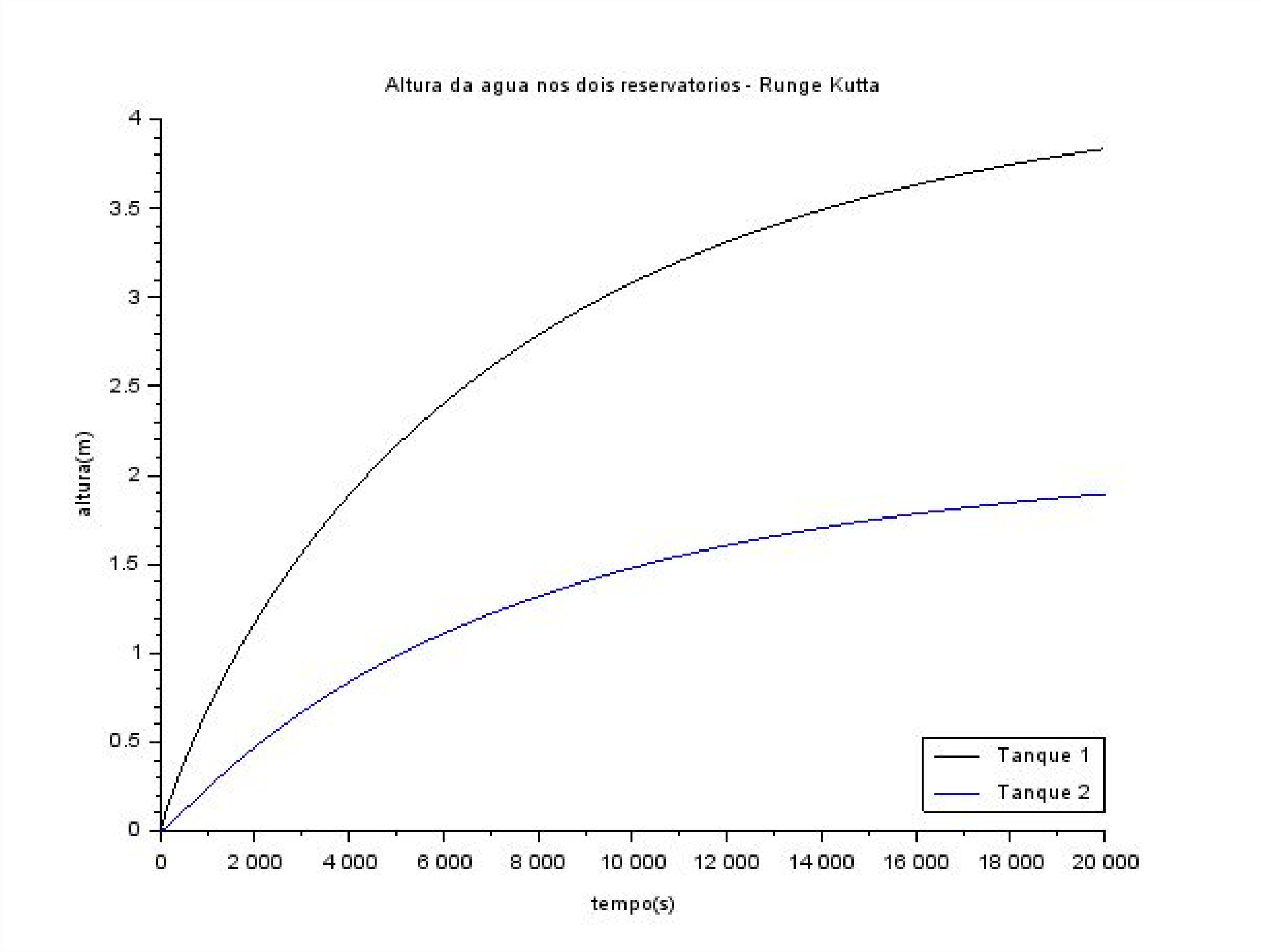

Gráfico 4 - Altura da água pelo tempo nos dois reservatórios (Runge Kutta)

Código 1

```
clear
// Parametros:
S=10
R=2*10^8
ro=1000
g=10
Qe=0.010247
// Instante inicial:
t(1)=0
// Instante final:
tf=20000
// Condicao inicial:
h(1)=0
// Passo de integracao:
y=0.5
// Calculo de numero de passos:
n=round((tf-t(1))/y)
// Definindo a funcao h:
function [hdot]=funcao(h)
hdot=(-sqrt(ro*g*h/R)+Qe)/S;
endfunction
// Integracao numerica usando o metodo de Euler:
   // Comando for:
   for i=1:n
   // Vetor de tempo:
   t(i+1)=t(i)+y;
   // Solucao numerica:
   h(i+1)=h(i)+y* funcao(h(i));
   // Termino do comando for:
end
// Plotando a solucao numerica h versus t para Euler:
scf(1)
plot2d(t,h)
xtitle('Altura da agua no reservatorio - Euler','tempo(s)','altura(m)')
// Integracao numerica usando o metodo de Runge Kutta:
   for i=1:n
   t(i+1)=t(i)+y;
   // Solucao numerica:
   k1=y*funcao(h(i));
   k2=y*funcao(h(i)+k1/2);
   k3=y*funcao(h(i)+k2/2);
   k4=y*funcao(h(i)+k3);
   h(i+1)=h(i)+((k1+2*k2+2*k3+k4)/6);
   end
// Plotando a solucao numerica h versus t para Runge Kuttta:
scf(2)
plot2d(t,h)
xtitle('Altura da agua no reservatorio - Runge Kutta','tempo(s)','altura(m)')
```

Código 2

```
clear
// Parametros:
S1=10
S2=5
Ra=2*10^8
Rs=2*10^8
ro=1000
g=10
Qe=0.010247
// Instante inicial:
t(1)=0
// Instante final:
tf=20000
// Condicao inicial:
h1(1)=0
h2(1)=0
// Passo de integracao:
y=0.5
// Calculo de numero de passos:
n=round((tf-t(1))/y)
n=round((tf-t(1))/y)
// Definindo a funcao h:
function [h1dot, h2dot]=funcao(h1, h2)
h1dot=(Qe-sqrt(ro*g*(h1-h2)/Ra))/S1;
h2dot=(sqrt(ro*g*(h1-h2)/Ra)-sqrt(ro*g*h2/Rs))/S2
endfunction
// Integracao numerica usando o metodo de Euler:
  for i=1:n
  t(i+1)=t(i)+y;
  [j,k]=funcao(h1(i),h2(i))
  h1(i+1)=h1(i)+y*j;
  h2(i+1)=h2(i)+y*k;
  end
// Plotando a solucao numerica h versus t para Euler:
scf(1)
plot2d([t,t],[h1,h2],[1,2])
xtitle('Altura da agua nos dois reservatorios - Euler','tempo(s)','altura(m)')
legends(['Tanque 1','Tanque 2'],[1,2])
// Integracao numerica usando o metodo de Runge Kutta:
  for i=1:n
  t(i+1)=t(i)+y;
  [k11,k12]=funcao(h1(i),h2(i));
  [k21,k22]=funcao(h1(i)+k11/2,h2(i)+k12/2);
  [k31,k32]=funcao(h1(i)+k21/2,h2(i)+k22/2);
  [k41,k42]=funcao(h1(i)+k31,h2(i)+k32);
  h1(i+1)=h1(i)+y*((k11+2*k21+2*k31+k41)/6);
  h2(i+1)=h2(i)+y*((k12+2*k22+2*k32+k42)/6);
  end
// Plotando a solucao numerica h versus t para Runge Kuttta:
scf(2)
plot2d([t,t],[h1,h2],[1,2])
xtitle('Altura da agua nos dois reservatorios - Runge Kutta','tempo(s)','altura(m)')
legends(['Tanque 1','Tanque 2'],[1,2])
```