PME3380 - Modelagem de Sistemas Dinâmicos

Escola Politécnica da USP

Lista B

Professores: Dr. Décio Crisol Donha
Dr. Agenor T. Fleury

Aluno: Arthur Henrique Gomes de Pinho
N°USP:10379756

2020

EXEMPLO 1

Este exemplo implementa uma solução numérica para a equação diferencial $\dot{y}(t) = [1 - y(t)]/2$ segundo o Método de Euler. Para realização deste exemplo, foi implementado o seguinte código:

```
clear
// Carregando a equacao diferencial:
function [ydot]=funcao(y)
ydot=(1-y)/2;
endfunction
// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=0;
// Valor inicial da solucao exata:
ye(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.5;
// Calculo de numero de passos):
n=round(tf/h);
// Integracao numerica usando o method de Euler:
// Comando for:
for i=1:n
// Vetor de tempo:
t(i+1)=t(i)+h;
// Solucao numerica:
y(i+1)=y(i)+h* funcao(y(i));
// Solucao exata:
ye(i+1)=1-%e^(-t(i+1)/2);
// Termino do comando for:
end
// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo
t:
plot2d([t,t],[y,ye],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solucao numerica","Solucao exata"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao")
// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao","Solucao
numerica","Solucao exata");
// 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));
```

O resultado do programa implementado está representado pelas figuras 1 e 2. Pode-se constatar que a solução numérica pelo método de Euler apresenta resultados próximos da solução exata.

Figura 1 - Solução por Método de Euler com resultados discretos

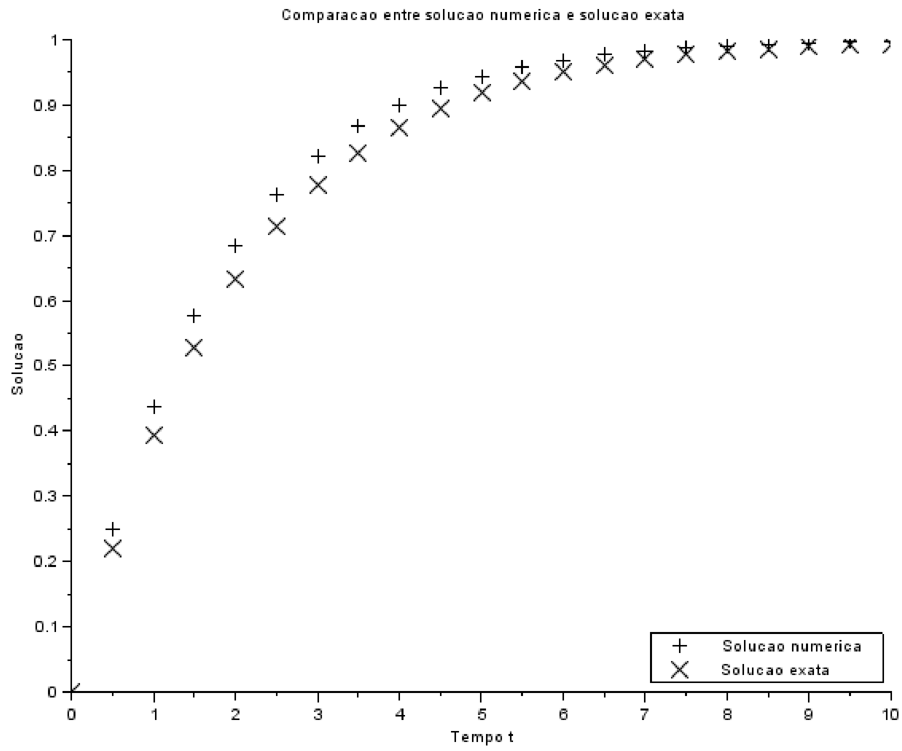
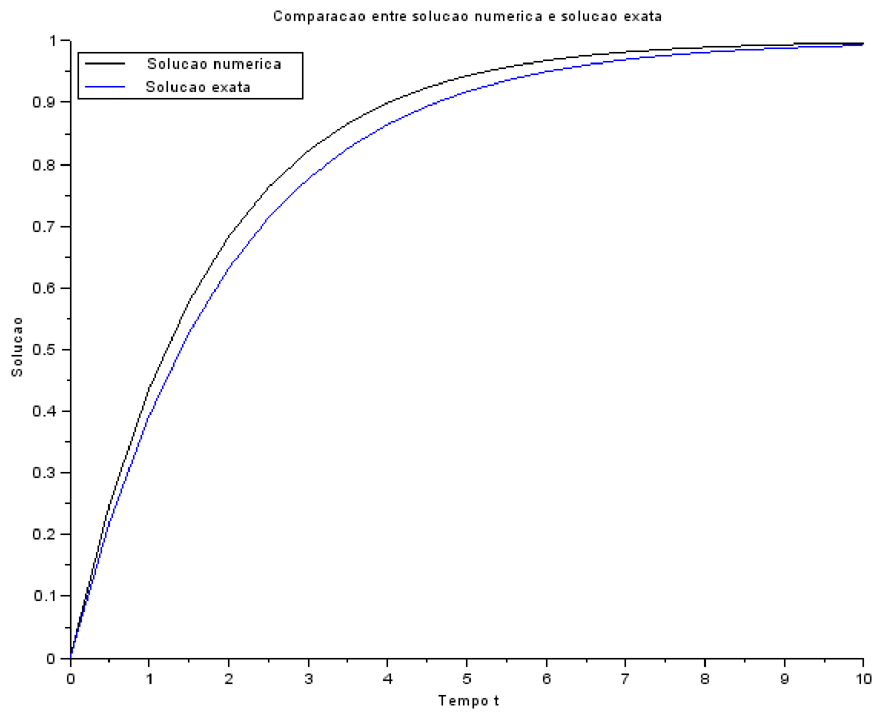


Figura 2 - Solução por Método de Euler com resultados linearizados



EXEMPLO 2

Este exemplo implementa uma solução numérica para a equação diferencial $\dot{y}(t) = [1 - y(t)]/2$ segundo o Método de Runge-Kutta. Para realização deste exemplo, foi implementado o seguinte código:

```
// Apagando dados anteriores:
clear
// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=0;
// Valor inicial da solucao exata:
ye(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.5;
// Calculo de numero de passos:
n=round((tf-t(1))/h);
// Integracao numerica usando o metodo de Runge Kutta:
// Comando for:
for i=1:n
// Vetor de tempo:
t(i+1)=t(i)+h;
// Solucao numerica:
k1=h*(1-(y(i)))/2;
k2=h*(1-(y(i)+k1/2))/2;
k3=h*(1-(y(i)+k2/2))/2;
k4=h*(1-(y(i)+k3))/2;
y(i+1)=y(i)+((k1+2*k2+2*k3+k4)/6);
// Solucao exata:
ye(i+1)=1-%e^(-t(i+1)/2);
// Termino do comando for:
end
// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de
tempo t:
plot2d([t,t],[y,ye],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solucao numerica","Solucao exata"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao")
// Abrindo uma nova janela de graficos:
set("current_figure", 1);
// Aumentando a espessura das linhas:
xset('thickness',2)
// Aumentando o tamanho da fonte:
xset('font size',4)
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao","Solucao
numerica","Solucao exata");
// Diminuindo a espessura das linhas:
xset('thickness',1)
// 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)
```

As figuras 3 e 4 apresentam os resultados obtidos através do programa implementado. Pode-se constatar que o Método Runge-Kutta apresenta grande precisão, dado que, graficamente, coincide com a solução exata.

Figura 3 - Solução por Método de Runge-Kutta com resultados discretos

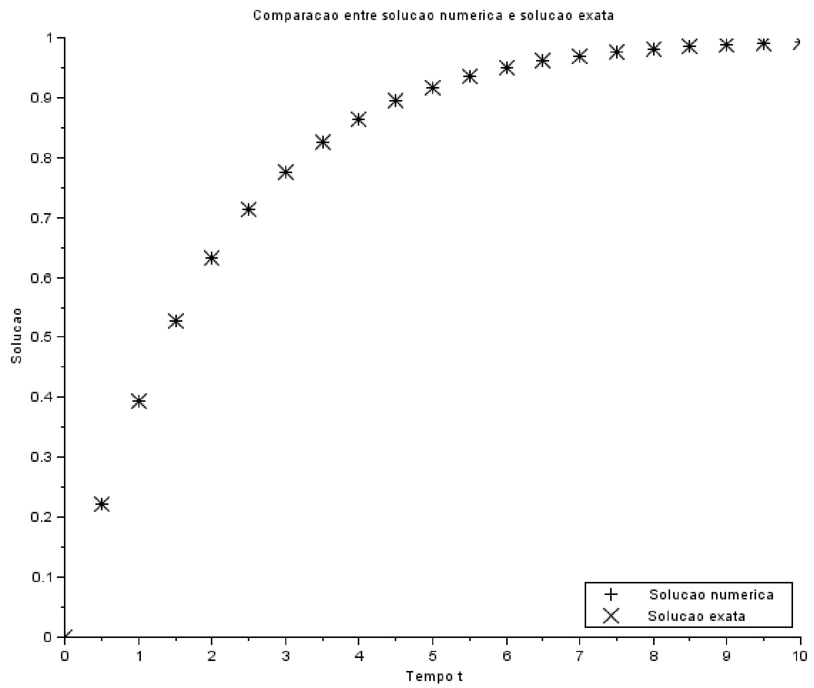
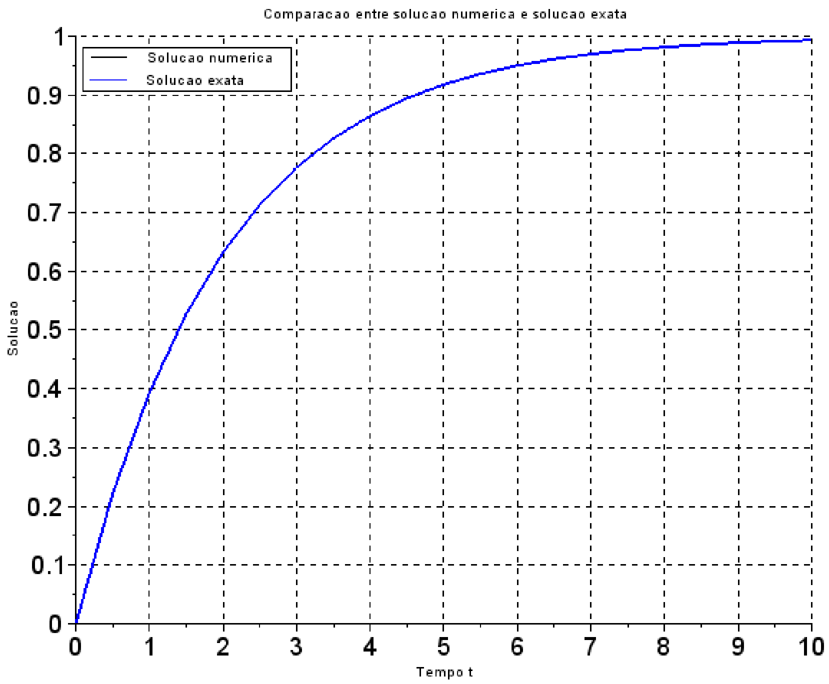


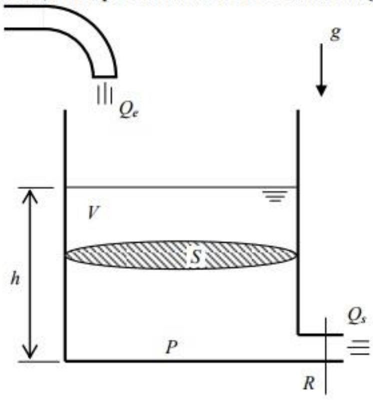
Figura 4 - Solução por Método de Runge-Kutta com resultados linearizados



EXERCÍCIO 1

Exercício:

Implemente um programa no Scilab que resolva numericamente a equação diferencial que modela o sistema abaixo, tanto pelo método de Euler como Runge Kutta.



Reservatório com água

Parâmetros:

S = 10 m^2 - área da seção transversal (constante)

$R = 2\times10^8$ Pa/(m^3/s)^2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)

ρ = 1000 kg/m^3 - massa específica da água

G = 10 m/s^2 - aceleração da gravidade na superfície da terra

Variáveis:

Q_e = 0,010247 m^3/s - vazão de entrada

h: nível do reservatório [m]

V: volume de água no reservatório [m^3]

P: pressão relativa à atmosférica, no fundo do reservatório [Pa]

Q_s: vazão de saída [m^3/s]

Admite-se que a água seja incompressível.

Pela equação da continuidade:

$$\frac{dV}{dt} = Q_e - Q_s$$

Vamos admitir que a perda de carga na saída é modelada pela expressão:

$$P = RQ_s^2 \quad \Rightarrow \quad Q_s = \sqrt{\frac{P}{R}}$$

Por outro lado, a pressão no fundo do reservatório é:

$$P = \rho g h$$

Volume de água no reservatório:

$$V = Sh \quad \Rightarrow \quad \dot{V} = S\dot{h}$$

Substituindo:

$$S\dot{h} = Q_e - \sqrt{\frac{\rho g h}{R}}$$

Resultando na seguinte equação difererencial ordinária não linear (modelo de 1 reservatório):

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

Considere uma entrada Q_e constante.

Utilizando, primeiramente, o método de Euler, chega-se nos resultados mostrados e plotados nos gráficos a seguir. O código é apresentado posteriormente no apêndice.

Figura 5 - Solução pelo Método de Euler da variação do nível do reservatório

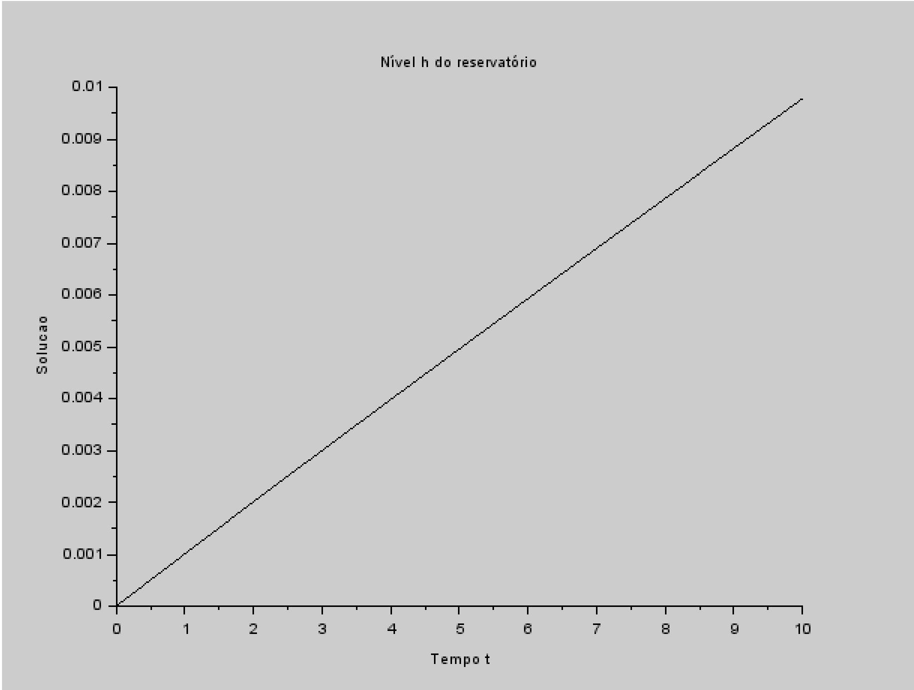


Figura 6 - Solução pelo Método de Euler da variação do volume de água do reservatório

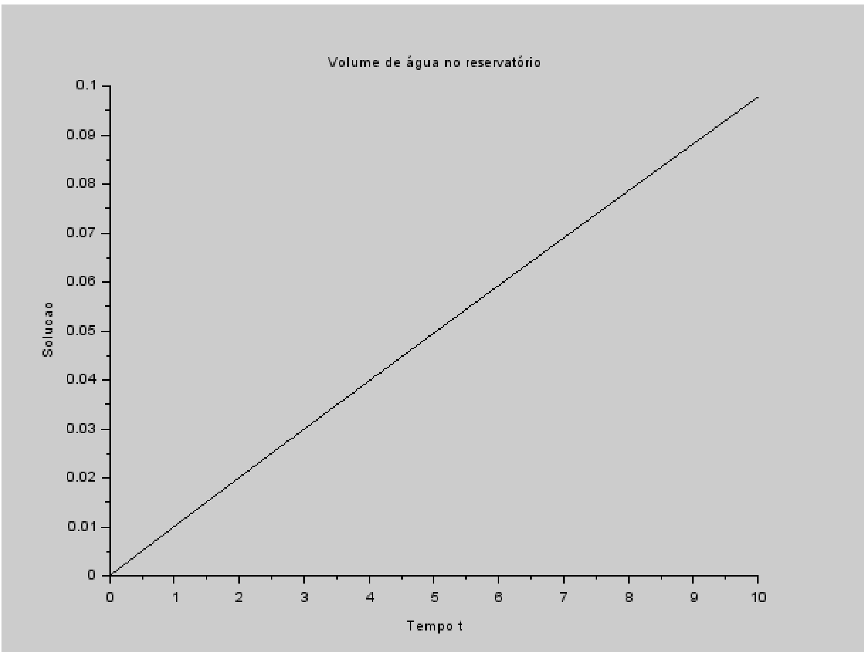


Figura 7 - Solução pelo Método de Euler da variação da pressão no fundo do reservatório

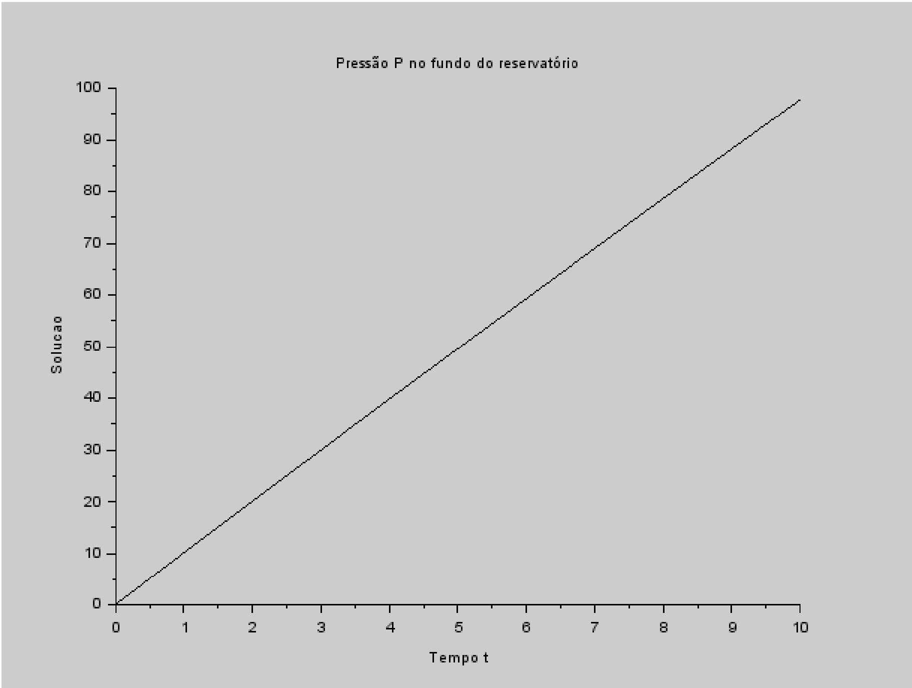
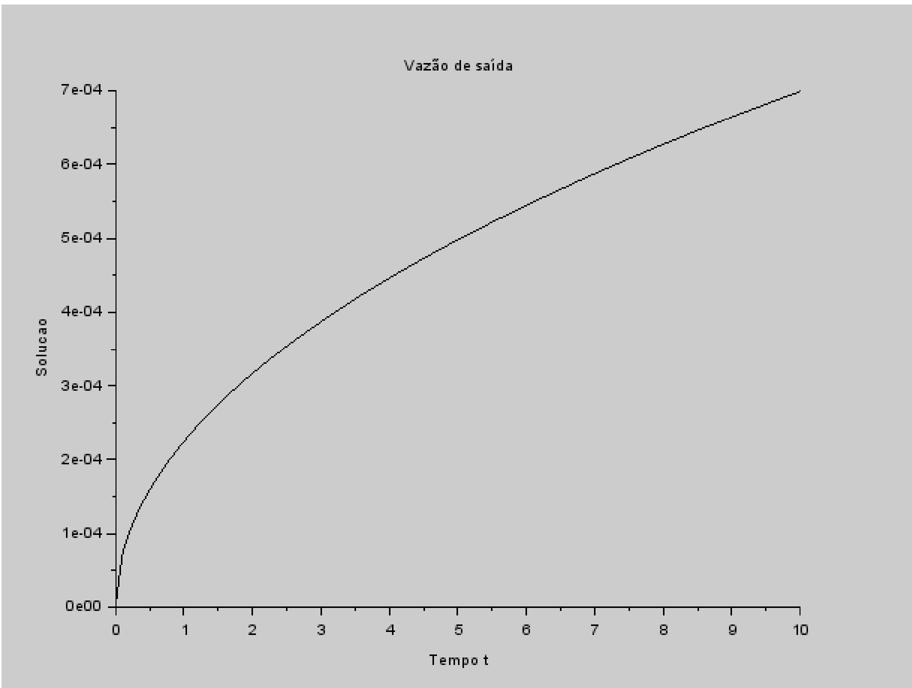


Figura 8 - Solução pelo Método de Euler da variação da vazão de saída



Agora, utilizando o método de Runge-Kutta, chega-se nos seguintes resultados.

Figura 9 - Solução pelo Método de Runge-Kutta da variação do nível do reservatório.

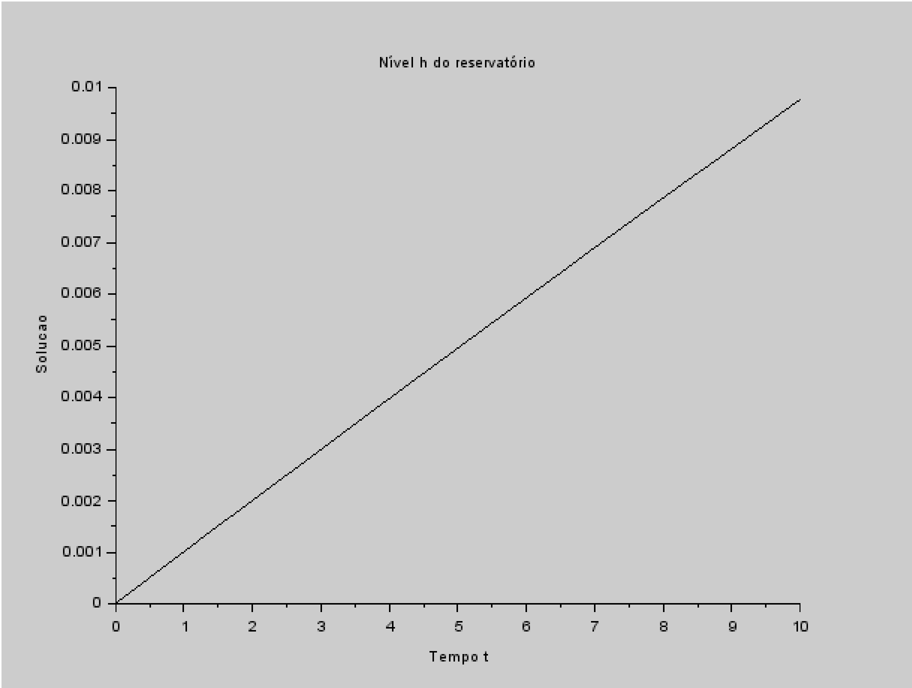


Figura 10 - Solução pelo Método de Runge-Kutta da variação do volume de água do reservatório.

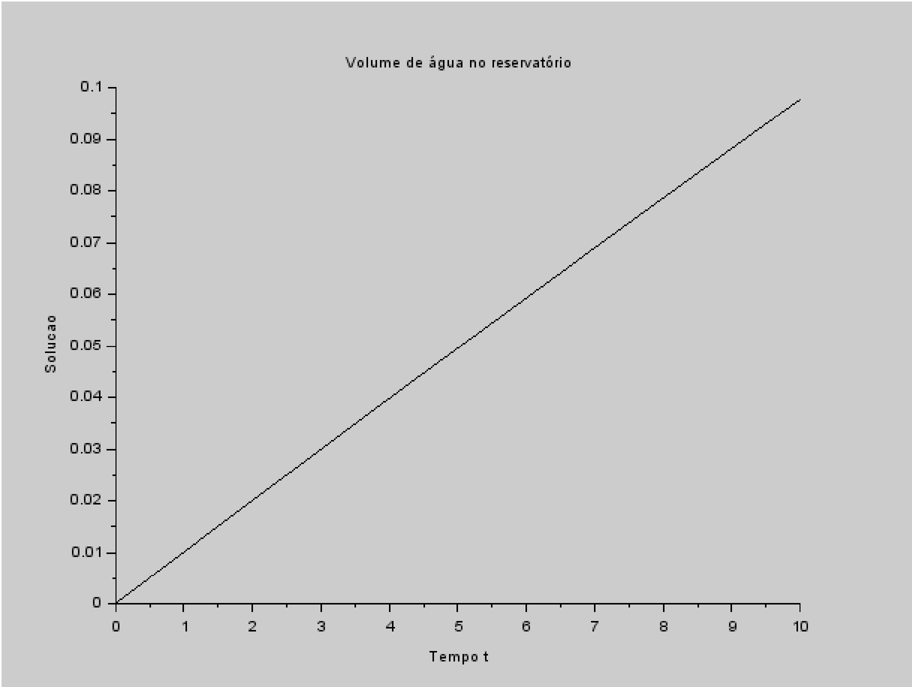


Figura 11 - Solução pelo Método de Runge-Kutta da variação da pressão no fundo do reservatório

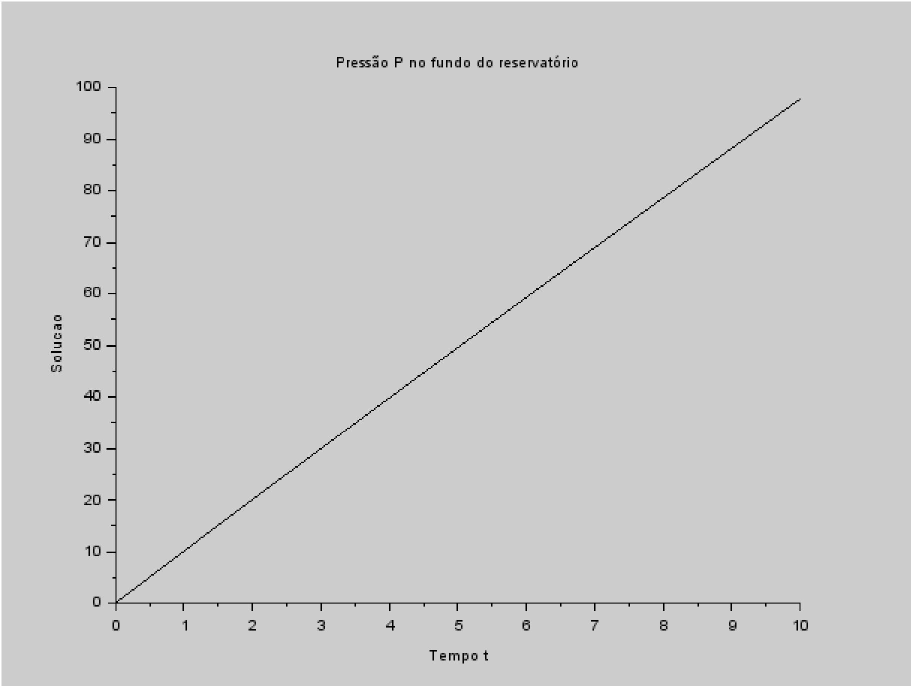
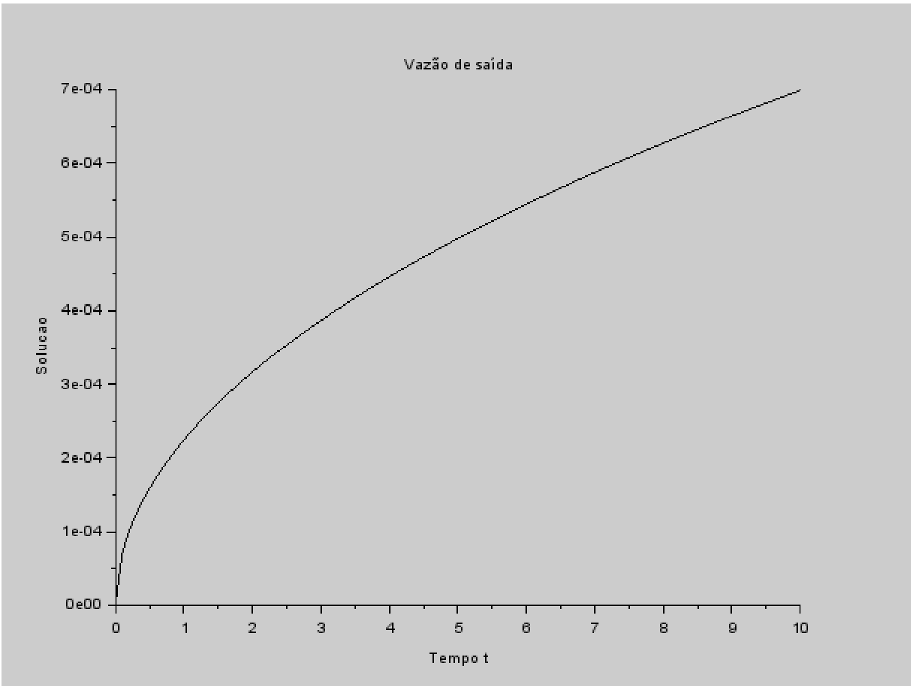


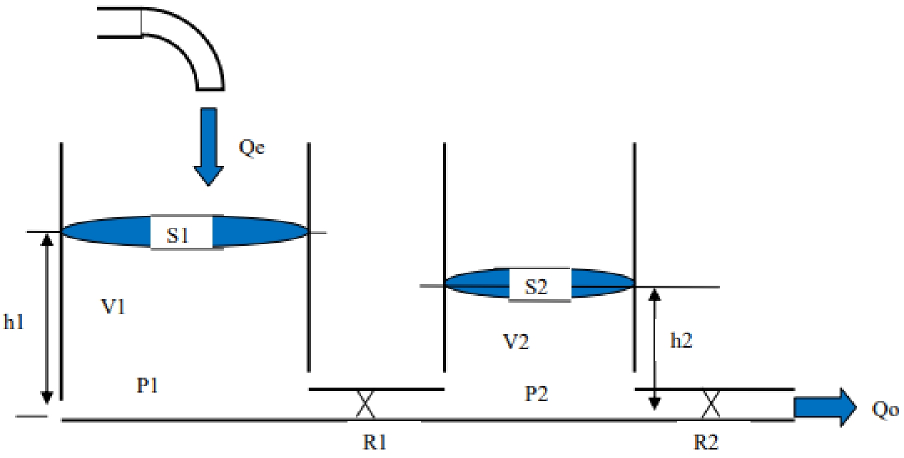
Figura 12 - Solução pelo Método de Runge-Kutta da variação da vazão de saída



Percebe-se que ambos os métodos resultam em integrações em que apresentam os mesmos resultados.

EXERCÍCIO 2

Desenvolva um programa em Scilab que resolva numericamente o sistema de equações diferencias que modela o sistema com dois reservatórios, usando tanto Euler como Runge Kutta. Dica: raciocine com vetores.



A seguir estão representados os gráficos do sistema com dois reservatórios, resolvido pelo método de Euler.

Figura 13 - Solução pelo Método de Euler comparação das alturas dos reservatórios

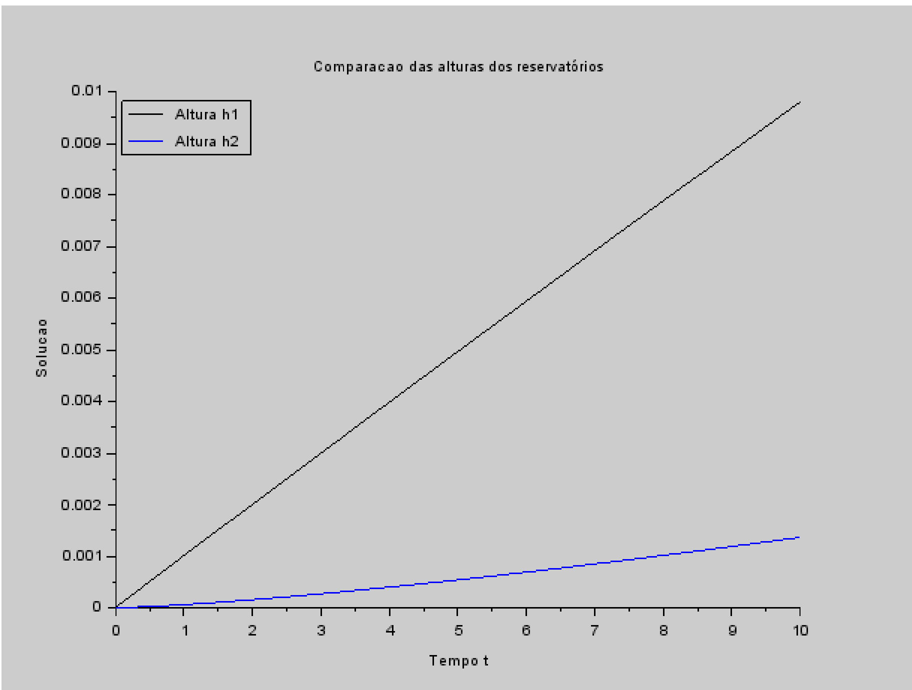


Figura 14 - Solução pelo Método de Euler comparação dos volumes dos reservatórios

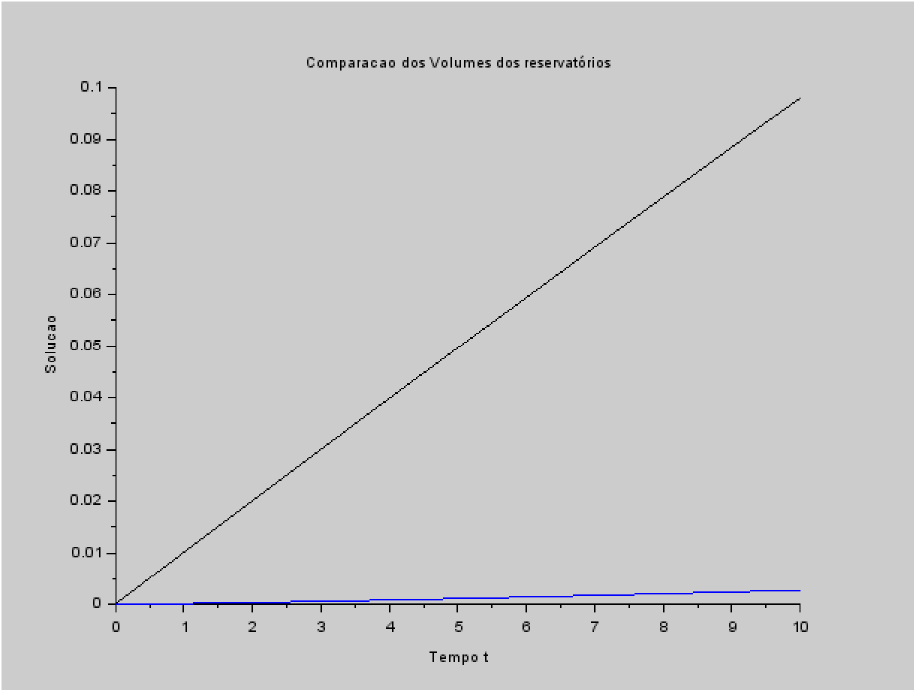


Figura 15 - Solução pelo Método de Euler comparação das pressões dos reservatórios

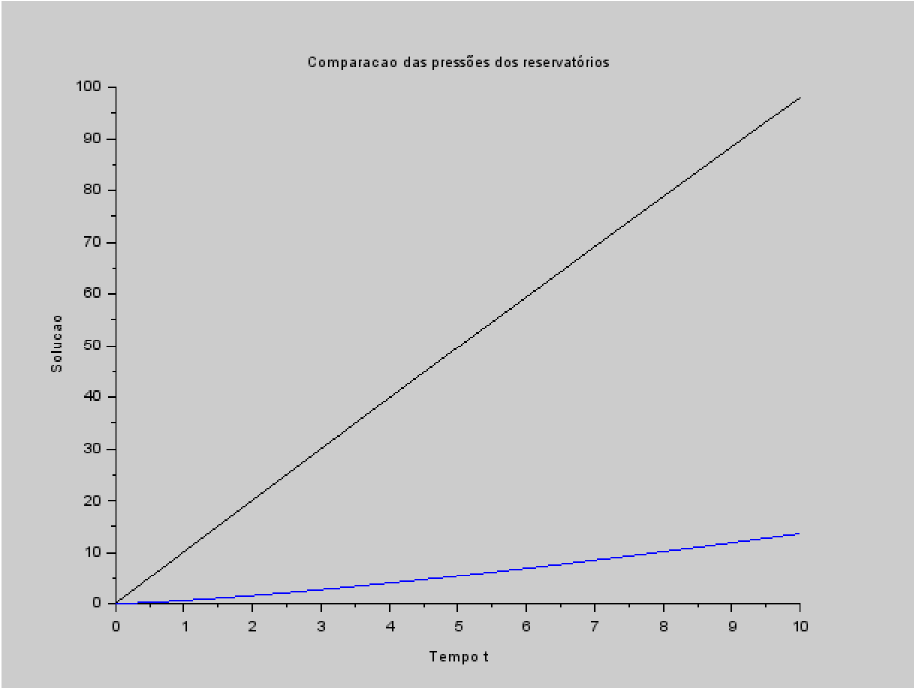
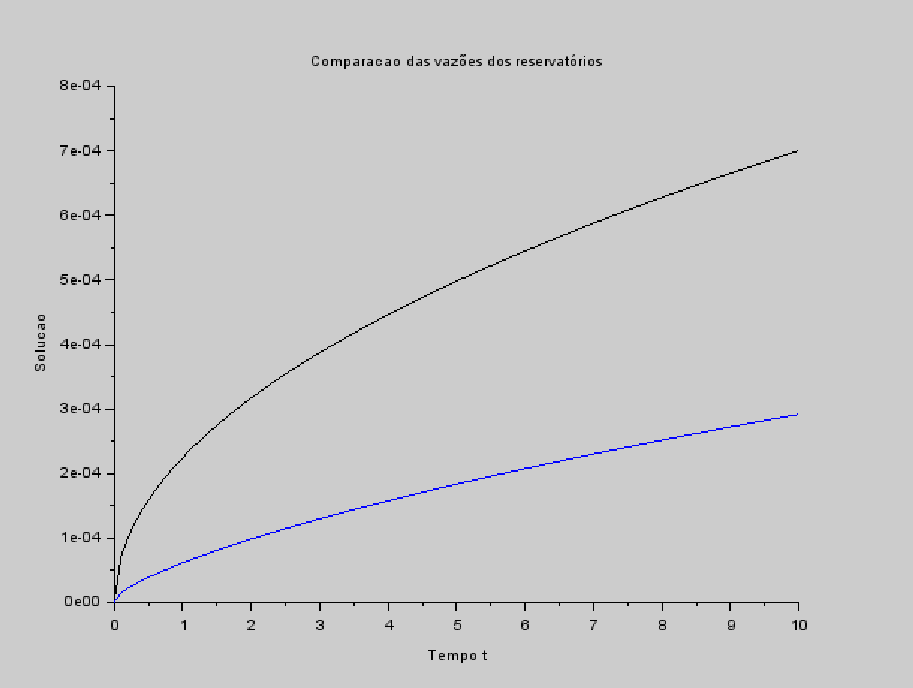


Figura 16 - Solução pelo Método de Euler comparação das vazões dos reservatórios



APÊNDICE

Método de Euler – Sistema com um reservatório

```
// Apagando dados anteriores:
clear

//Exercício

S = 10 //m2 - área da seção transversal (constante)
R = 2e8 //Pa/(m3/s)2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)
p = 1e3 //kg/m3 - massa específica da água
g = 10 //m/s2 - aceleração da gravidade na superfície da terra
Qe = 0.010247 //m3/s - vazão de entrada

//Achar
//h (nível do reservatório [m])
//V (volume de água no reservatório [m3])
//P (pressão relativa à atmosférica, no fundo do reservatório [Pa])
//Qs (vazão de saída [m3/s])
//Admite-se que a água seja incompressível.

function ydot=funcao(y)
ydot=(-sqrt(p*g*y/R)+Qe)/S;
endfunction

// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=0;
// Valor inicial da solucao exata:
ye(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.1;
// Calculo de numero de passos:
n=round(tf/h);
// Integracao numerica usando o metodo de Euler:
// Comando for:
for i=1:n
// Vetor de tempo:
t(i+1)=t(i)+h;
// Solucao numerica:
y(i+1)=y(i)+h*funcao(y(i));
// Termino do comando for:
end

V = S*y;
P = p*g*y;
Qs = sqrt(P/R);

// Plotando solucao numerica y versus vetor de tempo t:
figure(1);
plot2d([t],[y]);
xtitle("Nível h do reservatório","Tempo t","Solucao")

figure(2);
plot2d([t],[V]);
xtitle("Volume de água no reservatório","Tempo t","Solucao")

figure(3);
plot2d([t],[P]);
xtitle("Pressão P no fundo do reservatório","Tempo t","Solucao")
```

```
figure(4);
plot2d([t],[Qs]);
xtitle("Vazão de saída","Tempo t","Solucao")
```

Método de Runge-Kutta – Sistema com um reservatório

```
// Apagando dados anteriores:
clear

//Exercício

S = 10 //m2 - área da seção transversal (constante)
R = 2e8 //Pa/(m3/s)2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)
p = 1e3 //kg/m3 - massa específica da água
g = 10 //m/s2 - aceleração da gravidade na superfície da terra
Qe = 0.010247 //m3/s - vazão de entrada

//Achar
//h (nível do reservatório [m])
//V (volume de água no reservatório [m3])
//P (pressão relativa à atmosférica, no fundo do reservatório [Pa])
//Qs (vazão de saída [m3/s])
//Admite-se que a água seja incompressível.

// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=0;
// Valor inicial da solucao exata:
ye(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.1;
// Calculo de numero de passos:
n=round(tf/h);

// Integracao numerica usando o metodo de Runge Kutta:
// Comando for:
for i=1:n
// Vetor de tempo:
t(i+1)=t(i)+h;
// Solucao numerica:
k1=(-sqrt(p*g*y(i)/R)+Qe)/S
k2=(-sqrt(p*g*(y(i)+h*k1/2)/R)+Qe)/S
k3=(-sqrt(p*g*(y(i)+h*k2/2)/R)+Qe)/S
k4=(-sqrt(p*g*(y(i)+h*k3)/R)+Qe)/S

//k1=h*(1-(y(i)))/2;
//k2=h*(1-(y(i)+k1/2))/2;
//k3=h*(1-(y(i)+k2/2))/2;
//k4=h*(1-(y(i)+k3))/2;

y(i+1)=y(i)+((k1+2*k2+2*k3+k4)*h/6);

// Termino do comando for:
end

V = S*y
P = p*g*y
Qs = sqrt(P/R)

// Plotando solucao numerica y versus vetor de tempo t:
figure(1);
plot2d([t],[y]);
```

```
xtitle("Nível h do reservatório","Tempo t","Solucao")

figure(2);
plot2d([t],[V]);
xtitle("Volume de água no reservatório","Tempo t","Solucao")

figure(3);
plot2d([t],[P]);
xtitle("Pressão P no fundo do reservatório","Tempo t","Solucao")

figure(4);
plot2d([t],[Qs]);
xtitle("Vazão de saída","Tempo t","Solucao")
```

Método de Euler – Sistema com dois reservatórios

```
// Apagando dados anteriores:
clear

//Exercício

S1 = 10 //m2 - área da seção transversal (constante)
S2 = 2 //m2 - área da seção transversal (constante)
R1 = 2e8 //Pa/(m3/s)2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)
R2 = 1.6e8 //Pa/(m3/s)2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)
p = 1e3 //kg/m3 - massa específica da água
g = 10 //m/s2 - aceleração da gravidade na superfície da terra
Qe = 0.010247 //m3/s - vazão de entrada

//Achar
//h (nível do reservatório [m])
//V (volume de água no reservatório [m3])
//P (pressão relativa à atmosférica, no fundo do reservatório [Pa])
//Qs (vazão de saída [m3/s])
//Admite-se que a água seja incompressível.

function hdot1=hp1(h1, h2)
hdot1=(Qe-sqrt(p*g*(h1-h2)/R1))/S1;
endfunction

function hdot2=hp2(h1, h2)
hdot2=(sqrt(p*g*(h1-h2)/R1)-sqrt(p*g*h2/R2))/S2;
endfunction

// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicoes iniciais:
h1(1)=0;
h2(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.1;
// Calculo de numero de passos:
n=round(tf/h);

// Integracao numerica usando o metodo de Euler:
// Comando for:
for i=1:n
// Vetor de tempo:
t(i+1)=t(i)+h;
// Solucao numerica:
h1(i+1)=h1(i)+h*hp1(h1(i),h2(i));
h2(i+1)=h2(i)+h*hp2(h1(i+1),h2(i)); //colocando h1(i+1) para usar um dado mais atualizado
// Termino do comando for:
```

```
end

V1 = S1*h1;
P1 = p*g*h1;
Qs1 = sqrt(P1/R1);

V2 = S2*h2;
P2 = p*g*h2;
Qs2 = sqrt(P2/R2);

// Plotando solucao numerica y versus vetor de tempo t:
figure(1);
plot2d([t,t],[h1,h2],[1 2]);

legends(["Altura h1","Altura h2"],[1,2],2)
xtitle("Comparacao das alturas dos reservatórios","Tempo t","Solucao")

figure(2);
plot2d([t,t],[V1,V2], [1 2]);
xtitle("Comparacao dos Volumes dos reservatórios","Tempo t","Solucao")
figure(3);
plot2d([t,t],[P1,P2], [1 2]);
xtitle("Comparacao das pressões dos reservatórios","Tempo t","Solucao")
figure(4);
plot2d([t,t],[Qs1, Qs2], [1 2]);
xtitle("Comparacao das vazões dos reservatórios","Tempo t","Solucao")
```