



Escola Politécnica da Universidade de São Paulo

Lista 2

Nome: Yago Neves Yang

Número USP: 10772626

Disciplina: PME3380 – Modelagem de Sistemas Dinâmicos

Docentes: Décio Crisol e Agenor Fleury

São Paulo

2020

SUMÁRIO

1	INTRODUÇÃO	3
2	PROBLEMAS PROPOSTOS.....	3
2.1	PRIMEIRO EXERCÍCIO.....	3
2.2	SEGUNDO EXERCÍCIO.....	6
3	BIBLIOGRAFIA	9
4	APÊNDICE	9

1 INTRODUÇÃO

O objetivo da segunda lista proposta é o de implementar códigos no ambiente *Scilab* que permitam a solução de equações diferenciais através dos métodos de Euler e Runge-Kutta. Para comprovar o funcionamento dos códigos propostos ainda será proposta a implementação destes para a obtenção de gráficos referentes à solução para dois sistemas envolvendo a altura de água em reservatórios.

Nota-se ainda nesta introdução que todos os códigos empregados usarão os seguintes parâmetros em comum indicados na tabela logo abaixo.

Tabela 1 - Parâmetros comuns a todas as simulações

Instante de tempo inicial [s]	0
Instante de tempo final [s]	1000
Passo temporal [s]	0,01

2 PROBLEMAS PROPOSTOS

2.1 Primeiro exercício

O primeiro sistema proposto trata de um reservatório único e definido pelos seguintes parâmetros e equação diferencial listados abaixo.

Tabela 2 - Parâmetros para o primeiro exercício

Área da seção transversal - S [m ²]	10
Perda de carga - R [Pa/(m ³ s ²)]	2.10 ⁸
Massa específica da água - ρ [m ³ /kg]	1000
Aceleração da gravidade - g [m/s ²]	10
Vazão de entrada - Q_e [m ³ /s]	0,010247
Altura inicial da água no reservatório [m]	5

$$\dot{h} = \left(-\sqrt{\frac{\rho g h}{R}} + Q_e \right) \frac{1}{S} \rightarrow \text{Equação diferencial referente ao 1º exercício.}$$

A partir disso e dos códigos 1 e 2 apresentados no apêndice deste relatório, foi possível obter os seguintes resultados abaixo para a altura da água no reservatório no decorrer do tempo. Observe que a diferença visual entre os gráficos obtidos com os dois métodos é mínima.

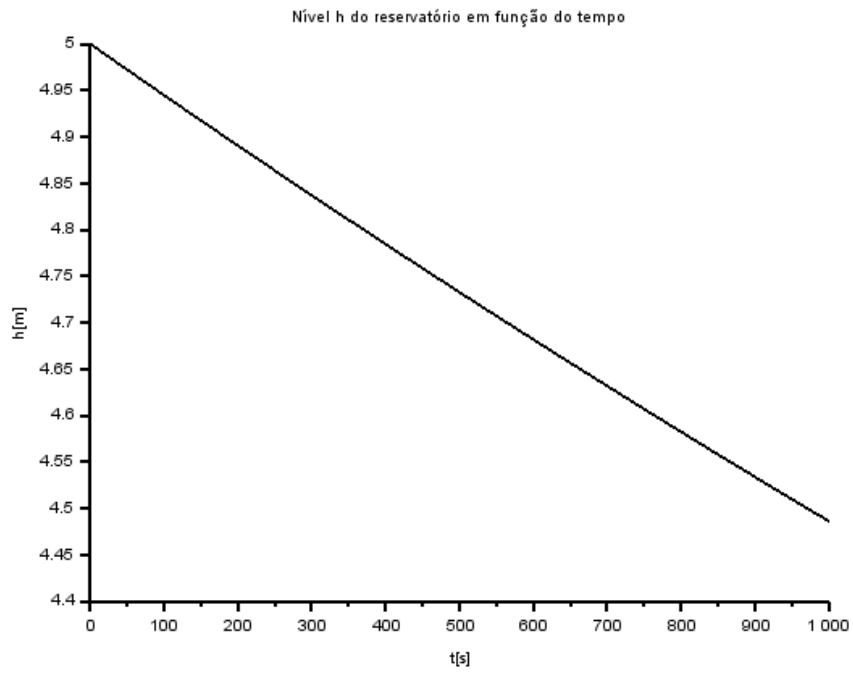


Figura 1 - Altura h do reservatório em função do tempo (Euler)

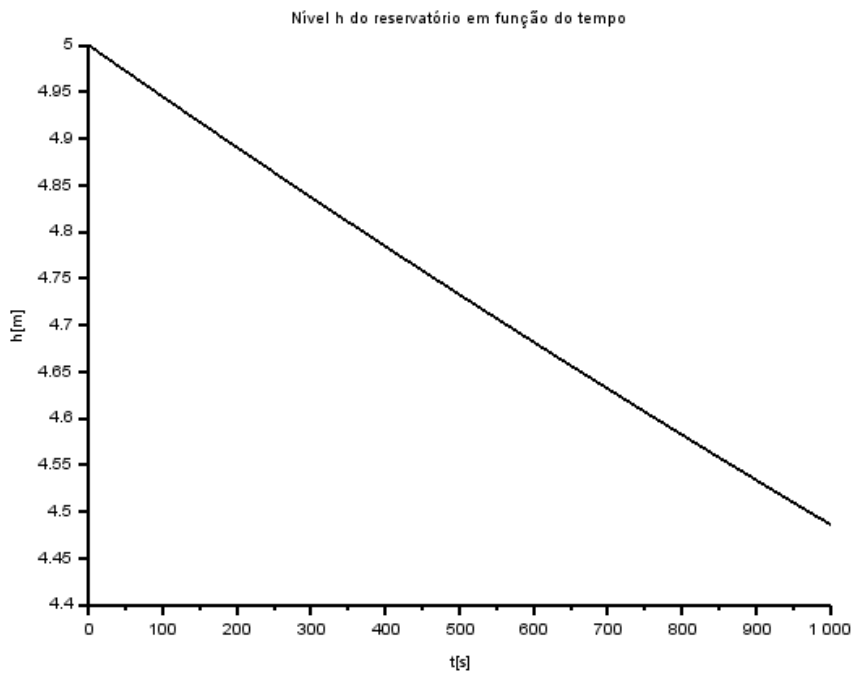


Figura 2 - Altura h do reservatório em função do tempo (Runge-Kutta)

2.2 Segundo exercício

O segundo exercício consiste em dois reservatórios acoplados entre si e definidos pelos seguintes parâmetros e equações diferenciais abaixo.

Tabela 3 - Parâmetros do segundo exercício

Área da seção transversal 1 – S_1 [m ²]	10
Área da seção transversal 2 – S_2 [m ²]	7,5
Perda de carga 1 – R_1 [Pa/(m ³ s) ²]	$2 \cdot 10^8$
Perda de carga 2 – R_2 [Pa/(m ³ s) ²]	10^8
Massa específica da água – ρ [m ³ /kg]	1000
Aceleração da gravidade – g [m/s ²]	10
Vazão de entrada – Q_e [m ³ /s]	0,010247
Altura inicial da água no reservatório 1 [m]	5
Altura inicial da água no reservatório 2 [m]	3

$$\left\{ \begin{array}{l} \dot{h}_1 = \left[Q_e - \sqrt{\frac{\rho g (h_1 - h_2)}{R_1}} \right] \frac{1}{S_1} \\ \dot{h}_2 = \left[\sqrt{\frac{\rho g (h_1 - h_2)}{R_1}} - \sqrt{\frac{\rho g h_2}{R_2}} \right] \frac{1}{S_2} \end{array} \right. \rightarrow \text{Equações diferenciais referentes ao 2º exercício}$$

Fazendo uso dos códigos 3 e 4 presentes no apêndice deste relatório foi possível chegar aos seguintes resultados para as alturas do nível da água nos dois reservatórios. Note que a diferença visual nos gráficos obtidos entre os métodos empregados, novamente, é mínima.

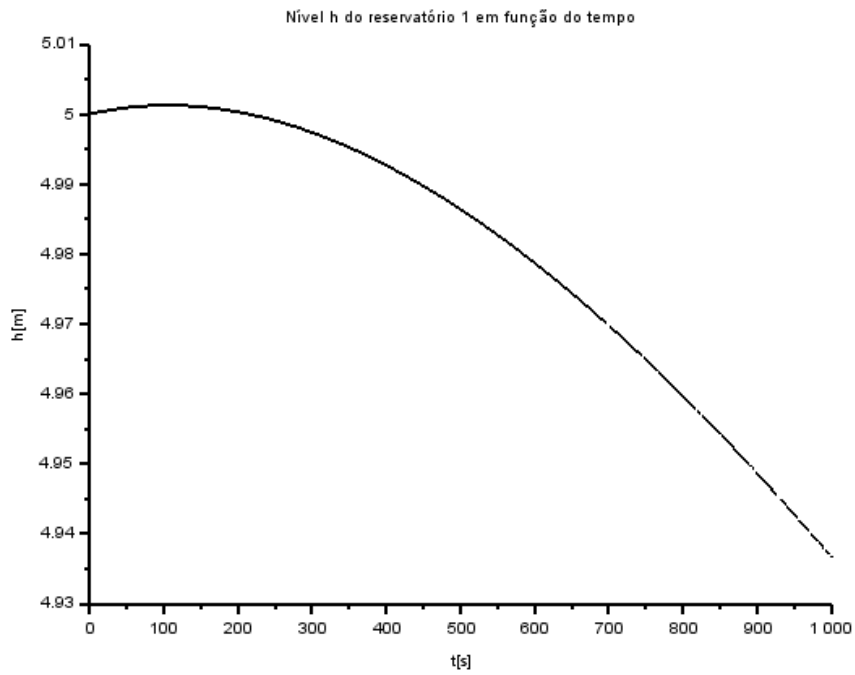


Figura 3 - Altura da água no reservatório 1 em função do tempo (Euler)

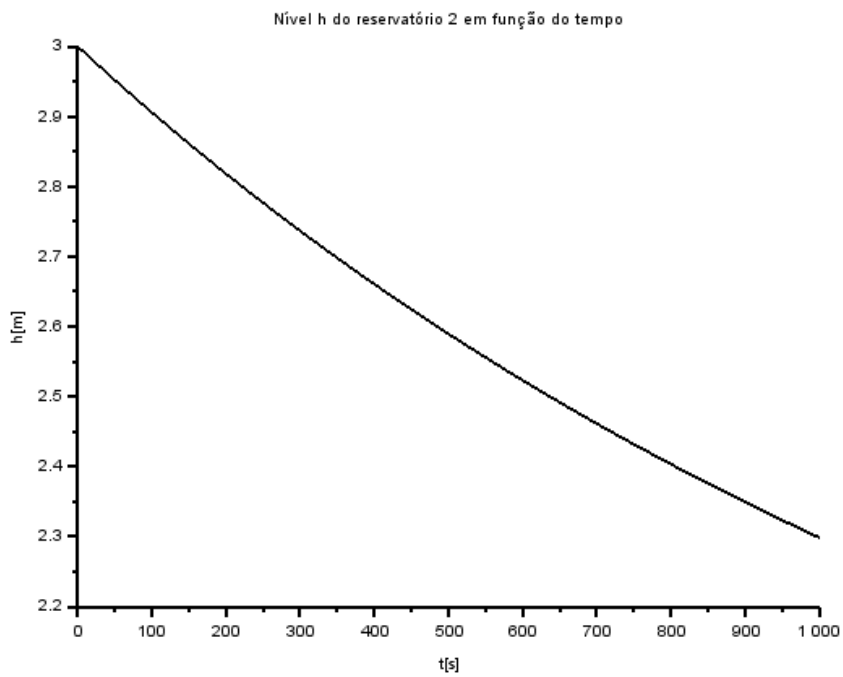


Figura 4 - Altura da água no reservatório 2 em função do tempo (Euler)

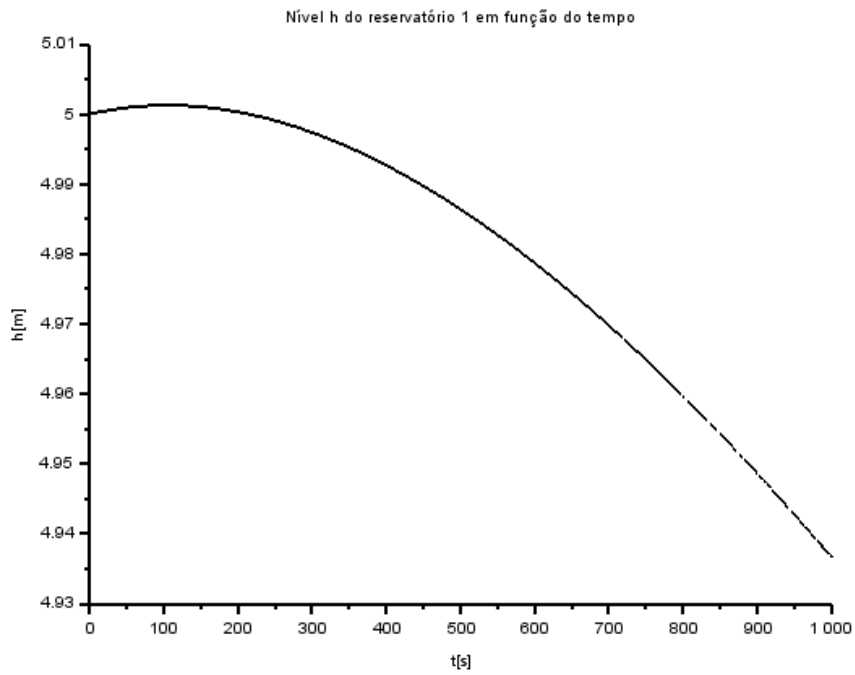


Figura 5 - Altura da água no reservatório 1 em função do tempo (Runge-Kutta)

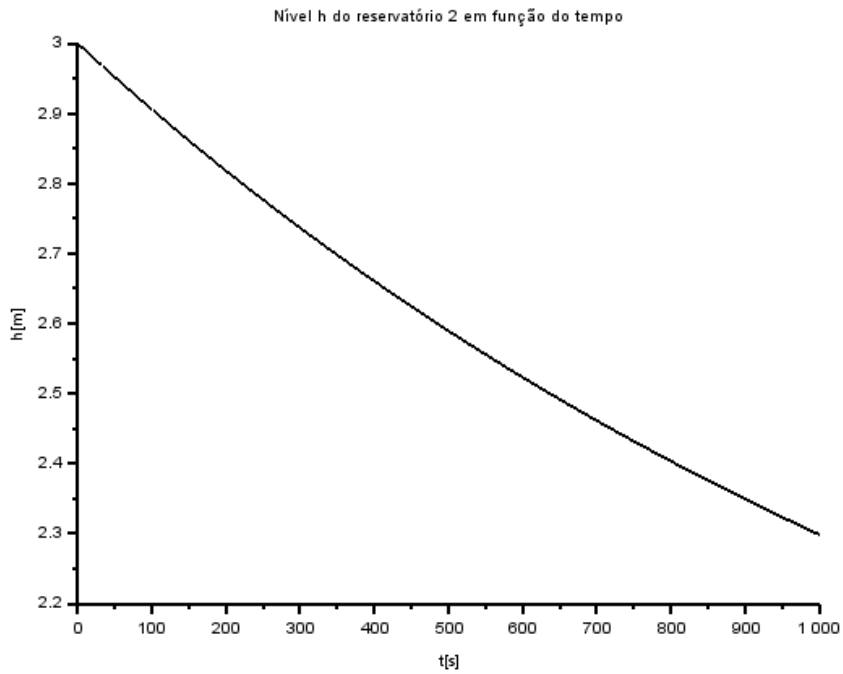


Figura 6 - Altura da água do reservatório 2 em função do tempo (Runge-Kutta)

3 BIBLIOGRAFIA

Autor desconhecido – **Lista B** – Acesso em 01/09/2020. Documento PDF disponível no site da disciplina de PME3380 – Modelagem de Sistemas Dinâmicos.

4 APÊNDICE

4.1 Código 1 – Euler (um reservatório)

```
// Parâmetros invariantes:  
S = 10.0; // Área da seção transversal [m^2]  
R = 2*10^8; // Perda de carga [Pa/((m^3*s)^2)]  
rho = 1000.0; // Massa específica da água [kg/m^3]  
g = 10.0; // Aceleração da gravidade [m/s^2]  
Qe = 0.010247; // Vazão de entrada [m^3/s]  
  
// Instante inicial:  
t(1) = 0;  
// Instante final:  
tf = 1000.0;  
// Condição inicial:  
y(1) = 5.0;  
// Passo de integração:  
h = 0.01;  
// Cálculo de número de passos:  
n = round(tf/h);  
  
// Integração numérica usando o método de Euler:  
for i = 1:n  
// Vetor temporal:  
t(i+1) = t(i)+h;  
// Solução numérica:  
y(i+1) = y(i)+h*(1/S)*(Qe-sqrt((rho*g*y(i))/R));  
end  
  
// Plot da solução em função do tempo:  
f1 = scf(1);  
p = gca();  
p.thickness = 2;  
plot2d(t',y');  
// Título da figura e nomes dos eixos:  
xtitle("Nível h do reservatório em função do tempo", "t[s]", "h[m]")
```

4.2 Código 2 – Runge-Kutta (um reservatório)

```
// Parâmetros invariantes:  
S = 10.0; // Área da seção transversal [m^2]  
R = 2*10^8; // Perda de carga [Pa/((m^3*s)^2)]  
rho = 1000.0; // Massa específica da água [kg/m^3]  
g = 10.0; // Aceleração da gravidade [m/s^2]  
Qe = 0.010247; // Vazão de entrada [m^3/s]  
  
// Instante inicial:  
t(1) = 0;  
// Instante final:
```

```

tf = 1000.0;
// Condição inicial:
y(1) = 5.0;
// Passo de integração:
h = 0.01;
// Cálculo de número de passos):
n = round(tf/h);

// Integração numérica pelo método de Runge Kutta:
for i = 1:n
// Vetor temporal:
t(i+1) = t(i)+h;
// Solução numérica:
k1 = h*(1/S)*(Qe-sqrt((rho*g*y(i))/R))
k2 = h*(1/S)*(Qe-sqrt((rho*g*(y(i)+k1/2))/R))
k3 = h*(1/S)*(Qe-sqrt((rho*g*(y(i)+k2/2))/R))
k4 = h*(1/S)*(Qe-sqrt((rho*g*(y(i)+k3))/R))
y(i+1) = y(i)+((k1+2*k2+2*k3+k4)/6);
end

// Plot da solução em função do tempo:
f1 = scf(1);
p = gca();
p.thickness = 2;
plot2d(t,'y');
// Título da figura e nomes dos eixos:
xtitle("Nível h do reservatório em função do tempo","t[s]","h[m]")

```

4.3 Código 3 – Euler (dois reservatórios)

```

// Parâmetros invariantes:
S1 = 10.0; // Área da seção transversal do reservatório 1 [m^2]
R1 = 2*10^8; // Perda de carga do reservatório 1 [Pa/((m^3*s)^2)]
S2 = 7.5; // Área da seção transversal do reservatório 2 [m^2]
R2 = 10^8; // Perda de carga do reservatório 2 [Pa/((m^3*s)^2)]
rho = 1000.0; // Massa específica da água [kg/m^3]
g = 10.0; // Aceleração da gravidade [m/s^2]
Qe = 0.010247; // Vazão de entrada [m^3/s]

// Instante inicial:
t(1) = 0;
// Instante final:
tf = 1000.0;
// Condição inicial:
y1(1) = 5.0;
y2(1) = 3.0;
// Passo de integração:
h = 0.01;
// Cálculo de número de passos):
n = round(tf/h);

// Integração numérica usando o método de Euler:
for i = 1:n
// Vetor temporal:
t(i+1) = t(i)+h;
// Solução numérica:
y1(i+1) = y1(i)+h*(1/S1)*(Qe-sqrt((rho*g*(y1(i)-y2(i))/R1)));

```

```
y2(i+1) = y2(i)+h*(1/S2)*((sqrt((rho*g*(y1(i)-y2(i)))/R1))-sqrt(rho*g*y2(i)/R2));
end
```

```
// Plot da solução em função do tempo:
```

```
f1 = scf(1);
p = gca();
p.thickness = 2;
plot2d(t,y1');
// Título da figura e nomes dos eixos:
xtitle("Nível h do reservatório 1 em função do tempo", "t[s]", "h[m]")
f2 = scf(2);
w = gca();
w.thickness = 2;
plot2d(t,y2');
// Título da figura e nomes dos eixos:
xtitle("Nível h do reservatório 2 em função do tempo", "t[s]", "h[m]")
```

4.4 Código 4 – Runge-Kutta (dois reservatórios)

```
// Parâmetros invariantes:
```

```
S1 = 10.0; // Área da seção transversal do reservatório 1 [m^2]
R1 = 2*10^8; // Perda de carga do reservatório 1 [Pa/((m^3*s)^2)]
S2 = 7.5; // Área da seção transversal do reservatório 2 [m^2]
R2 = 10^8; // Perda de carga do reservatório 2 [Pa/((m^3*s)^2)]
rho = 1000.0; // Massa específica da água [kg/m^3]
g = 10.0; // Aceleração da gravidade [m/s^2]
Qe = 0.010247; // Vazão de entrada [m^3/s]
```

```
// Instante inicial:
```

```
t(1) = 0;
```

```
// Instante final:
```

```
tf = 1000.0;
```

```
// Condição inicial:
```

```
y1(1) = 5.0;
```

```
y2(1) = 3.0;
```

```
// Passo de integração:
```

```
h = 0.01;
```

```
// Cálculo de número de passos:
```

```
n = round(tf/h);
```

```
// Integração numérica pelo método de Runge Kutta:
```

```
for i = 1:n
```

```
// Vetor temporal:
```

```
t(i+1) = t(i)+h;
```

```
// Solução numérica:
```

```
k11 = h*(1/S1)*(Qe-sqrt((rho*g*(y1(i)-y2(i)))/R1))
```

```
k21 = h*(1/S1)*(Qe-sqrt((rho*g*(y1(i)+k11/2-y2(i+h/2))/R1))
```

```
k31 = h*(1/S1)*(Qe-sqrt((rho*g*(y1(i)+k21/2-y2(i+h/2))/R1))
```

```
k41 = h*(1/S1)*(Qe-sqrt((rho*g*(y1(i)+k31-y2(i+h))/R1))
```

```
y1(i+1) = y1(i)+((k11+2*k21+2*k31+k41)/6);
```

```
k12 = h*(1/S2)*((sqrt((rho*g*(y1(i)-y2(i)))/R1))-sqrt(rho*g*y2(i)/R2))
```

```
k22 = h*(1/S2)*((sqrt((rho*g*(y1(i+h/2)-(y2(i)+k12/2))/R1))-sqrt(rho*g*(y2(i)+k12/2)/R2))
```

```
k32 = h*(1/S2)*((sqrt((rho*g*(y1(i+h/2)-(y2(i)+k22/2))/R1))-sqrt(rho*g*(y2(i)+k22/2)/R2))
```

```
k42 = h*(1/S2)*((sqrt((rho*g*(y1(i+h)-(y2(i)+k32))/R1))-sqrt(rho*g*(y2(i)+k32)/R2))
```

```
y2(i+1) = y2(i)+((k12+2*k22+2*k32+k42)/6)
```

```
end
```

```
// Plot da solução em função do tempo:
```

```
f1 = scf(1);  
p = gca();  
p.thickness = 2;  
plot2d(t,y1');  
// Título da figura e nomes dos eixos:  
xtitle("Nível h do reservatório 1 em função do tempo","t[s]","h[m]");  
f2 = scf(2);  
p = gca();  
p.thickness = 2;  
plot2d(t,y2');  
// Título da figura e nomes dos eixos:  
xtitle("Nível h do reservatório 2 em função do tempo","t[s]","h[m]");
```