Lista B

PME 3380 - Modelagem

Nathan Daleffi Rodrigues Rayes

10772585

Escola Politécnica

Universidade de São Paulo

São Paulo

2020

1

O problema do reservatório duplo a seguir foi resolvido em *Scilab* através de dois métodos iterativos: Euler e Runge-Kutta de 2ª ordem:

$$\begin{cases} \dot{h_1} = \left[Q_e - \sqrt{\frac{\rho g}{R_1}(h_1 - h_2)}\right] \frac{1}{S_1} \\ \dot{h_2} = \left[\sqrt{\frac{\rho g}{R_1}(h_1 - h_2)} - \sqrt{\frac{\rho g}{R_1} h_2}\right] \frac{1}{S_2} \end{cases} \tag{1}$$

Como condições iniciais foram adotadas alturas $h_1 = 1m$ e $h_2 = 0.5m$

Os resultados gráficos para o método de Euler e para essas condições e valores de $R_1 = R_2 = 2 \cdot 10^8 Pa/(m^3/s)^2$ foram os seguintes:

Figura 1 – Altura h_1 calculada por Euler

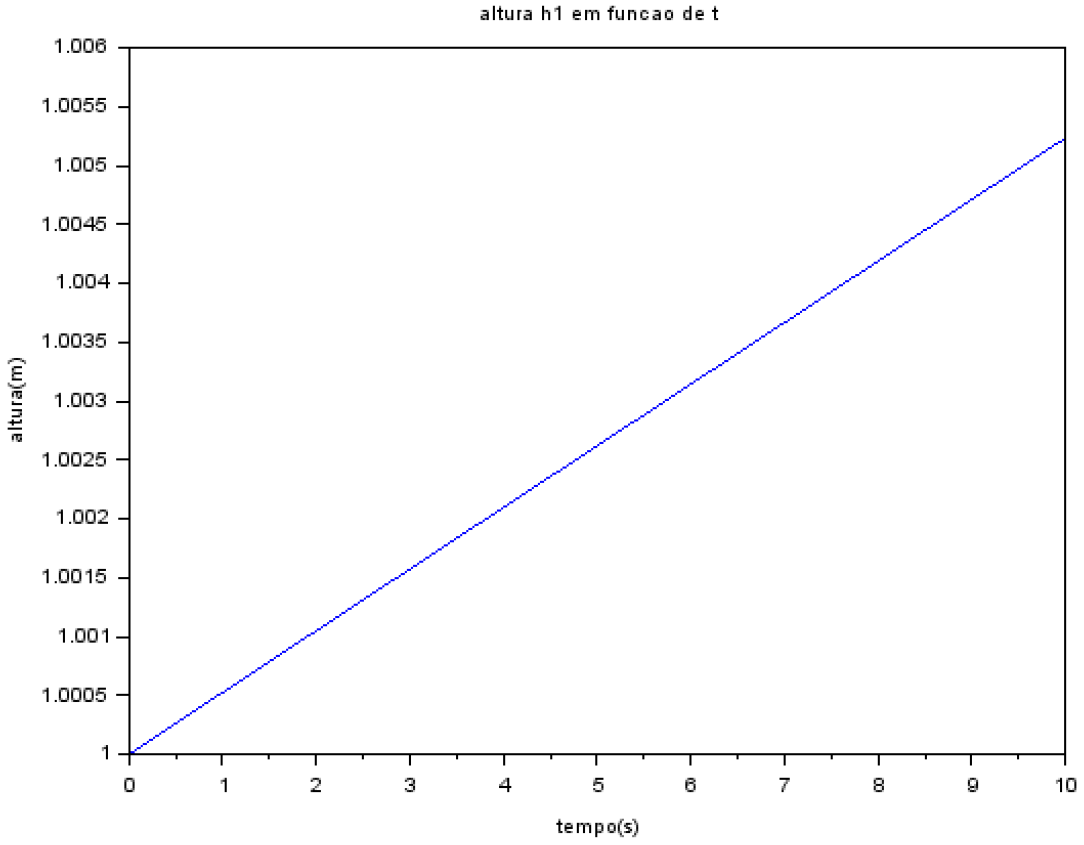
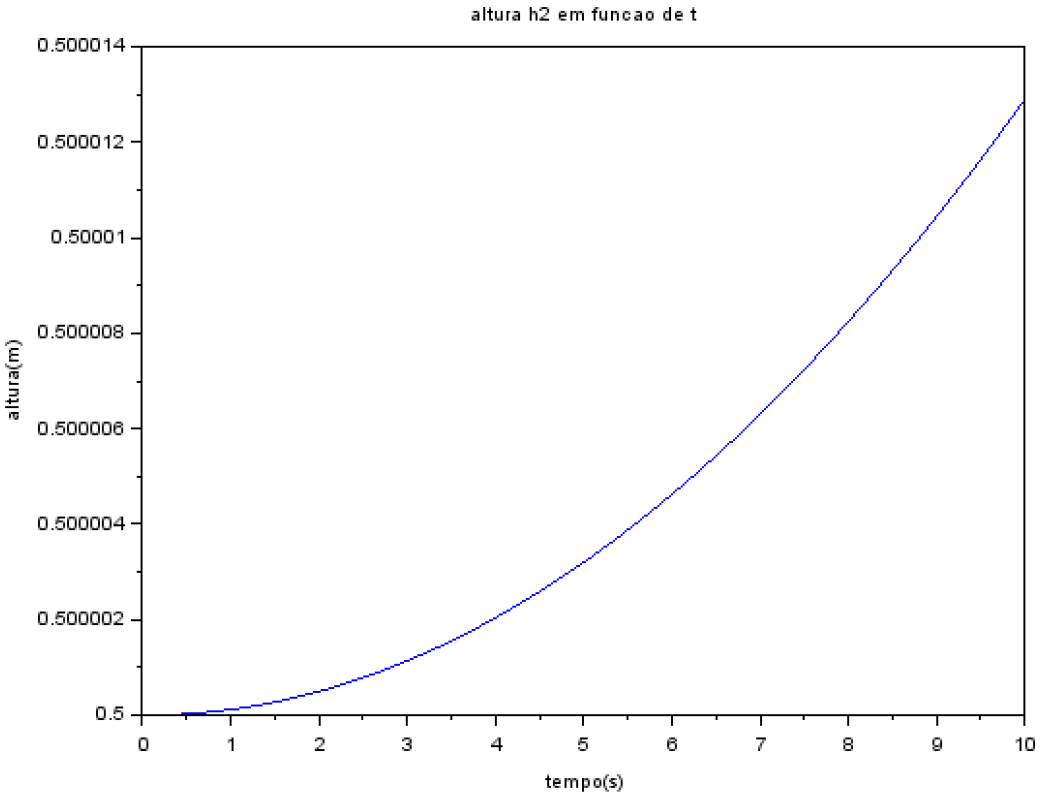


Figura 2 – Altura h_2 calculada por Euler



Pode-se perceber que o valor das Resistências R_1 e R_2 é elevado, o que leva a um aumento da altura de água em cada reservatório.

Os gráficos a seguir mostram esse mesmo resultado calculado a partir do método iterativo de Runge-Kutta de 2ª ordem:

Figura 3 – Altura h_1 calculada por Runge-Kutta de 2ª ordem

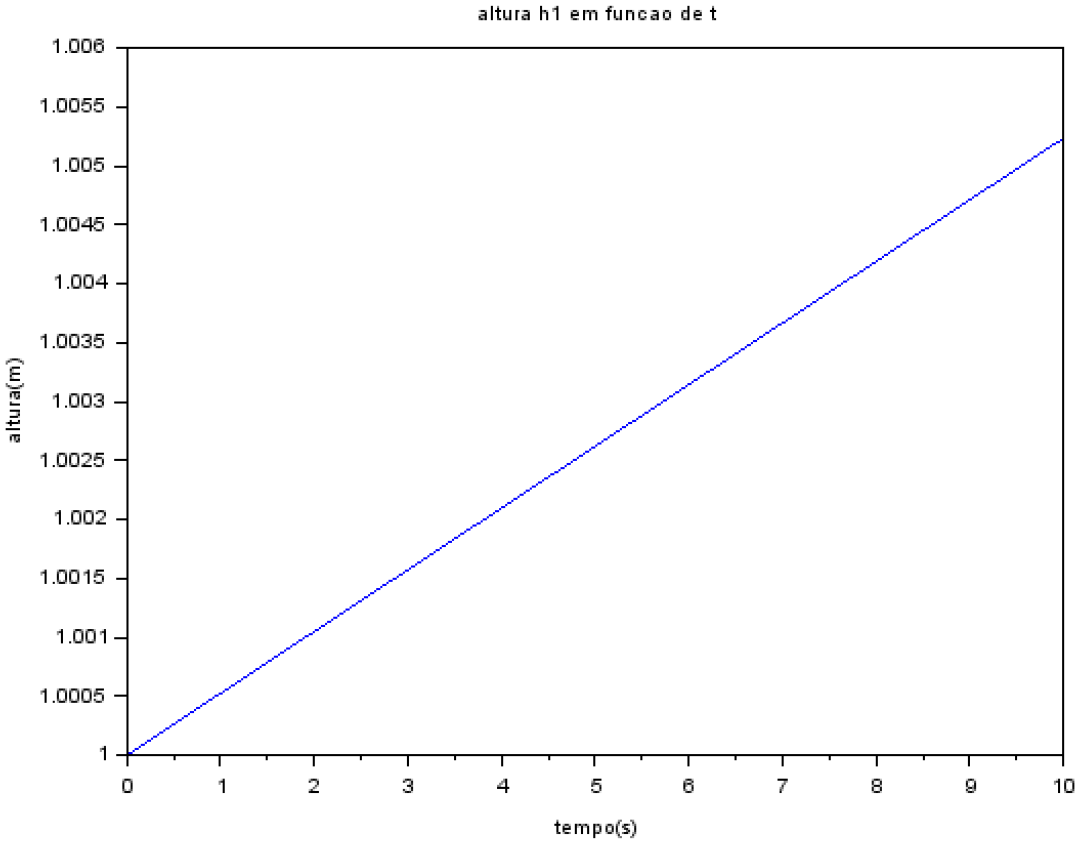
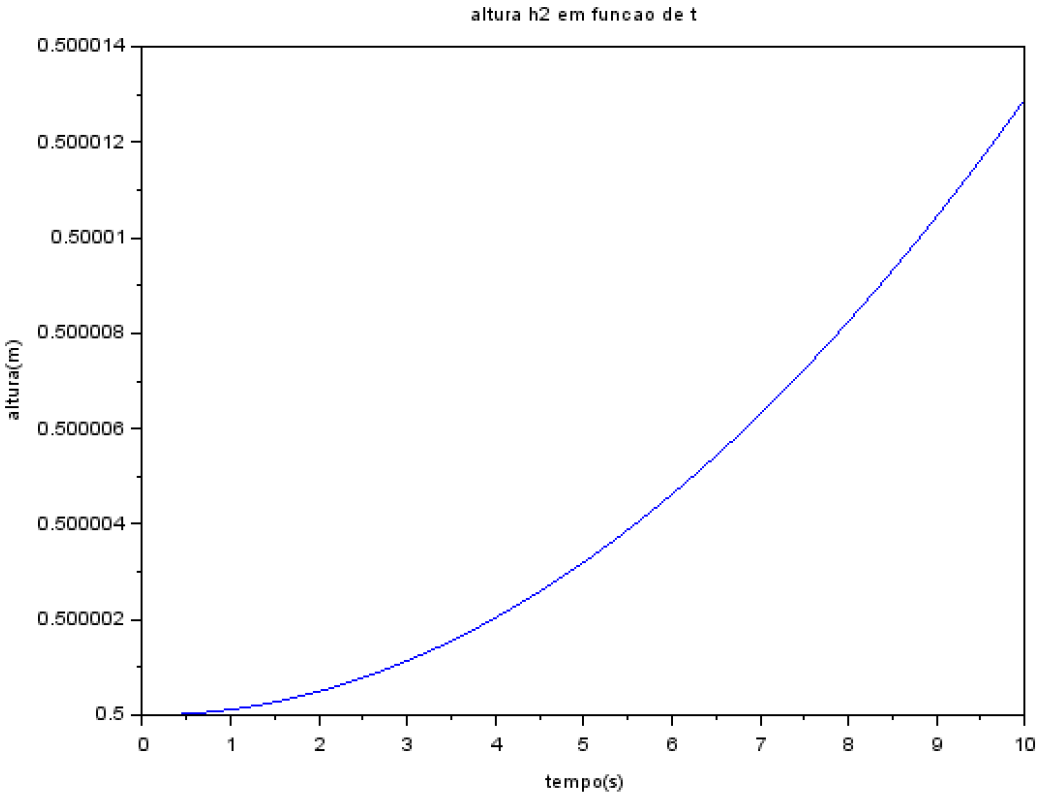


Figura 4 – Altura h_2 calculada por Runge-Kutta de 2ª ordem



Pode-se perceber que não há diferença significa entre estes dois métodos, apesar do método de Runge-Kutta ter uma ordem de convergência de 2º grau em comparação com a convergência linear do método de Euler.

Os códigos produzidos para cada método de resolução desse problema encontram-se abaixo:

a) Euler

```
clear;
rho = 1000;
g = 10;
R = 2e8;
R1 = R;
R2 = R;
S = 10;
Qe = 0.010247;
S1 = S;
S2 = S;

h10 = 1;
```

```
h20 = 0.5;

tf = 10;
n = 100;

t = linspace(0,tf,n);
dt = t(2) - t(1);
h1 = zeros(1,n);
h2 = zeros(1,n);
h1(1) = h10;
h2(1) = h20;
for i=2:n
    h1(i) = h1(i-1) + dt*(Qe - sqrt(rho*g*(h1(i-1)-h2(i-1))/R1))/S1;
    h2(i) = h2(i-1) + dt*(sqrt(rho*g*(h1(i-1)-h2(i-1))/R1) - sqrt(rho*g*h2(
    i-1)/R2))/S2;
end

f1 = scf(1);
plot(t,h1);
xtitle("altura h1 em funcao de t","tempo(s)","altura(m)")
f2 = scf(2);
plot(t,h2);
xtitle("altura h2 em funcao de t","tempo(s)","altura(m)")
plot(t,h2)
```

b) Runge-Kutta de 2ª ordem

```
clear;
rho = 1000;
g = 10;
R = 2e8;
R1 = R;
R2 = R;
S = 10;
Qe = 0.010247;
S1 = S;
S2 = S;

h10 = 1;
h20 = 0.5;

tf = 10;
n = 100;

t = linspace(0,tf,n);
dt = t(2) - t(1);
h1 = zeros(1,n);
h2 = zeros(1,n);
```

```
22 h1(1) = h10;
23 h2(1) = h20;
24 function y = f1(i)
25     y = (Qe - sqrt(rho*g*(h1(i)-h2(i))/R1))/S1;
26 endfunction
27
28 function y = f2(i)
29     y = (sqrt(rho*g*(h1(i)-h2(i))/R1) - sqrt(rho*g*h2(i)/R2))/S2
30 endfunction
31
32 for i=2:n
33     k11 = dt*f1(i-1);
34     k21 = dt*f1(i-1 + k11/2);
35     k31 = dt*f1(i-1 + k21/2);
36     k41 = dt*f1(i-1 + k31);
37
38     h1(i) = h1(i-1) + (k11 + 2*k21 + 2*k31 + k41)/6
39
40     k12 = dt*f2(i-1);
41     k22 = dt*f2(i-1 + k12/2);
42     k32 = dt*f2(i-1 + k22/2);
43     k42 = dt*f2(i-1 + k32);
44
45     h2(i) = h2(i-1) + (k12 + 2*k22 + 2*k32 + k42)/6
46 end
47
48 f1 = scf(1);
49 plot(t,h1);
50 xtitle("altura h1 em funcao de t","tempo(s)","altura(m)")
51 f2 = scf(2);
52 plot(t,h2);
53 xtitle("altura h2 em funcao de t","tempo(s)","altura(m)")
54 plot(t,h2)
```