ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

SAMUEL ALVES DA S. JUNIOR   NºUSP: 10769639

PME3380 – MODELAGEM DE SISTEMAS DINÂMICOS

(LISTA B)

SÃO PAULO, SP

2020

Sumário

LISTA DE GRÁFICOS

Exercício 1

Códigos

Nesta seção são apresentados os códigos empregados para a simulação do exercício 1 pedido.

Método de Euler

```
clear
clc
xdel(winsid())

function [f]=x(h, rho, g, R, Qe, S)
 f = (Qe - sqrt(rho*g*h/R))/S;
endfunction

rho = 1000;
Qe = 0.010247;
S = 10;
g = 10;
R = 2e8;

ti = 0;
tf = 50000;
n = 100000;
h = (tf-ti)/n;
t = zeros(n+1,1)

for i = 2:n+1
 t(i,1) = t(i-1,1) + h
end
H = zeros(n+1,1);
for i = 2:n+1
 H(i,1) = H(i-1,1) + h*x(H(i-1,1), rho, g, R, Qe, S)
end

plot(t,H);
title("Nível do reservatório - Método de Euler", "fontsize", 4)
xlabel("tempo (s)", "fontsize", 4)
ylabel("Nível do reservatório (m)", "fontsize", 4)
xgrid(1);
```

Método de Runge-Kutta

```
clear
clc
xdel(winsid())
function [f]=x(h, rho, g, R, Qe, S)
 f = (Qe - sqrt(rho*g*h/R))/S;
endfunction

rho = 1000;
Qe = 0.010247;
S = 10;
g = 10;
```

```
R = 2e8;

ti = 0;
tf = 50000;
n = 100000;
h = (tf-ti)/n;
t = zeros(n+1,1)

for i = 2:n+1
 t(i,1) = t(i-1,1) + h
end
H = zeros(n+1,1);
for i = 2:n+1
 k1 = x(H(i-1,1), rho, g, R, Qe, S);
 k2 = x(H(i-1,1) + h*0.5*k1, rho, g, R, Qe, S);
 k3 = x(H(i-1,1) + h*0.5*k2, rho, g, R, Qe, S);
 k4 = x(H(i-1,1) + h*k3, rho, g, R, Qe, S);
 H(i,1) = H(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4);
end

plot(t,H);
title("Nível do reservatório - Método de Runge-Kutta", "fontsize", 4)
xlabel("tempo (s)", "fontsize", 4)
ylabel("Nível do reservatório (m)", "fontsize", 4)
xgrid(1);
```

Resultados

Para o exercício 1 foi simulado o evento com um tempo de cerca de 50000 segundos, tanto com o método de Euler, como com o método de Runge Kutta de 4º ordem, nota-se a partir dos gráficos 1 e 2 listados abaixo, que os métodos apresentam resultados semelhantes, garantindo assim a validade do modelo.

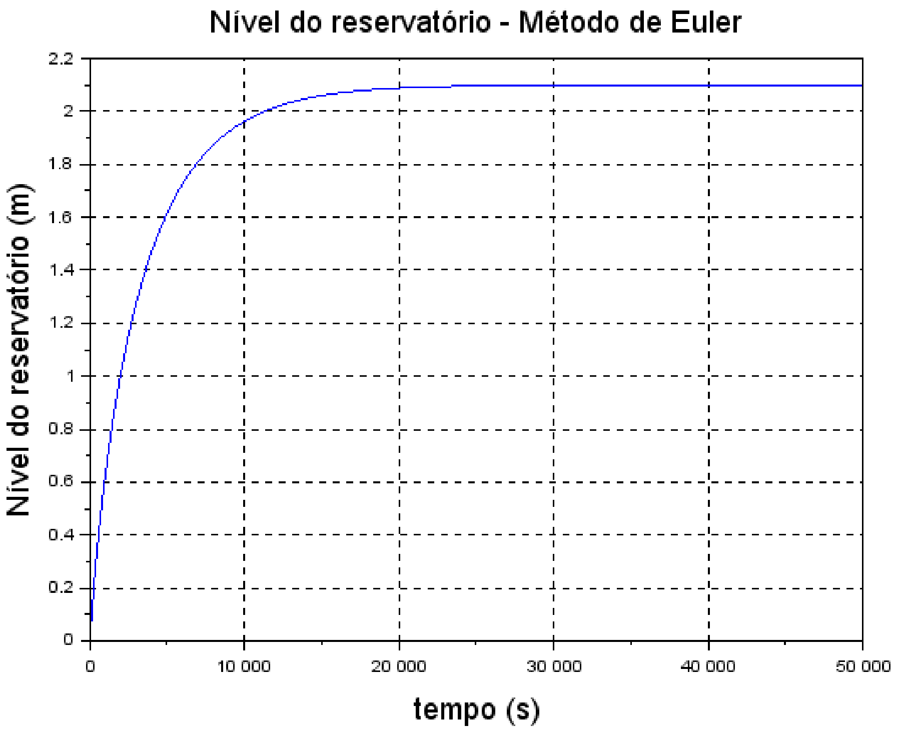

Gráfico 1: Problema com 1 reservatório - Método de Euler

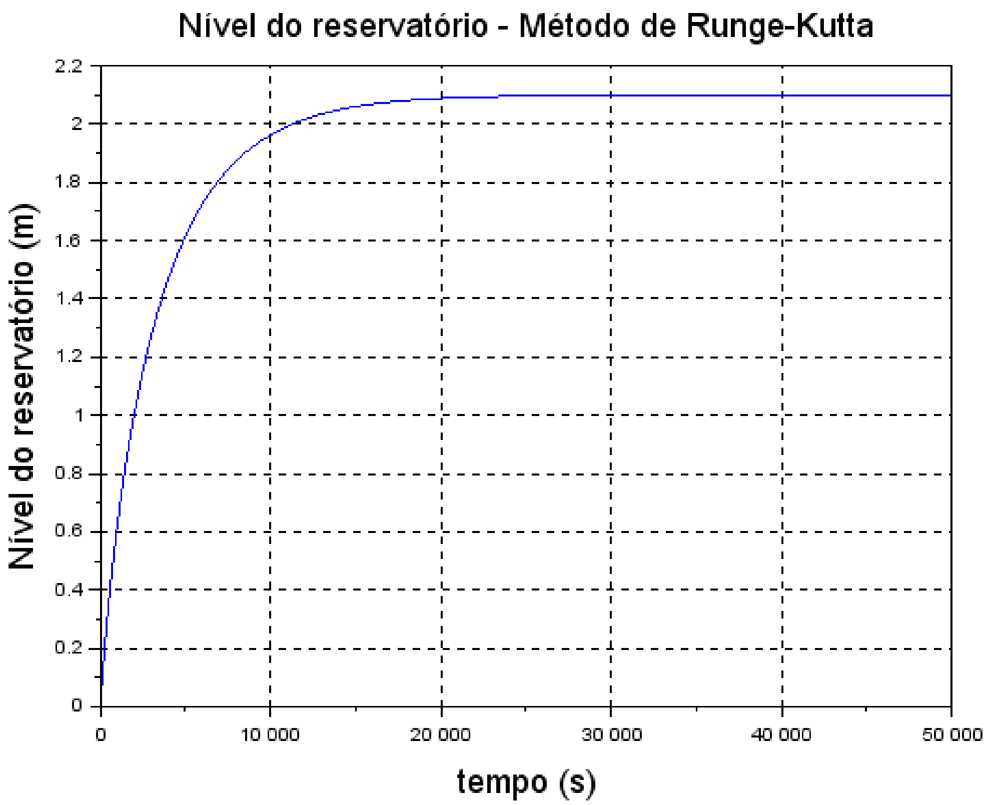

Gráfico 2: : Problema com 1 reservatório - Método de Runge-Kutta

Exercício 2

Códigos

Nesta seção são apresentados os códigos empregados para a simulação do exercício 2 pedido.

Método de Euler

```
clear
clc
xdel(winsid())
rho = 1000;
Qe = 0.010247;
S1 = 10 ;
S2 = 10 ;
g = 10;
R1 = 2e8;
R2 = 2e8;

h10 = 1 ;
h20 = 0.5 ;

tf = 50000;
n = 100000;

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,'black' ) ;
title(" h1 em função de t ", "fontsize", 4)
xlabel("tempo (s) " , "fontsize", 4)
ylabel("Nível do reservatório(m) ", "fontsize", 4)

f2 = scf ( 2 ) ;
plot (t , h2 ) ;
title(" h2 em função de t ", "fontsize", 4)
xlabel("tempo (s) " , "fontsize", 4)
ylabel("Nível do reservatório(m) ", "fontsize", 4)

f3 = scf(3);
plot (t , h1,'black',t,h2,'b' ) ;
legends(["Reservatório 1", "Reservatório 2"],[1,2],2)
title("Nível dos reservatórios - Método de Euler", "fontsize", 4)
xlabel("Tempo(s)", "fontsize", 4)
```

```
ylabel("Nível do reservatório(m) ", "fontsize", 4)
xgrid(1);
```

Método de Runge-Kutta

```
clear
clc
xdel(winsid())

function [f1]=x1(h1, h2, Qe, rho, g, Ra, S1)
 f1 = (Qe-sqrt(rho*g*(h1-h2)/Ra))/S1;
endfunction
function [f2]=x2(h1, h2, rho, g, Ra, Rs, S2)
 f2 = (sqrt(rho*g*(h1-h2)/Ra)-sqrt(rho*g*h2/Rs))/S2;
endfunction

rho = 1000;
Qe = 0.010247;
S1 = 10 ;
S2 = 10 ;
g = 10;
R1 = 2e8;
R2 = 2e8;

ti = 0;
tf = 50000;
n = 100000;

h = (tf-ti)/n;
t = zeros(n+1,1)

for i = 2:n+1
 t(i,1) = t(i-1,1) + h;
end
h1 = zeros(n+1,1);
h2 = zeros(n+1,1);
for i = 2:n+1
 k1 = x1(h1(i-1,1), h2(i-1,1), Qe, rho, g, R1, S1);
 k2 = x1(h1(i-1,1) + h*0.5*k1, h2(i-1,1) + h*0.5*k1, Qe, rho, g, R1, S1);
 k3 = x1(h1(i-1,1) + h*0.5*k2, h2(i-1,1) + h*0.5*k2, Qe, rho, g, R1, S1);
 k4 = x1(h1(i-1,1) + h*k3, h2(i-1,1) + h*k3, Qe, rho, g, R1, S1);
 h1(i,1) = h1(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4);
 k1 = x2(h1(i-1,1), h2(i-1,1), rho, g, R1, R2, S2);
 k2 = x2(h1(i-1,1) + h*0.5*k1, h2(i-1,1) + h*0.5*k1, rho, g, R1, R2, S2);
 k3 = x2(h1(i-1,1) + h*0.5*k2, h2(i-1,1) + h*0.5*k2, rho, g, R1, R2, S2);
 k4 = x2(h1(i-1,1) + h*k3, h2(i-1,1) + h*k3, rho, g, R1, R2, S2);
 h2(i,1) = h2(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4);
end
f1 = scf ( 1 ) ;
plot (t , h1,'black' ) ;
title(" h1 em função de t ", "fontsize", 4)
xlabel("tempo (s) " , "fontsize", 4)
ylabel("Nível do reservatório(m) ", "fontsize", 4)

f2 = scf ( 2 ) ;
```

```
plot (t , h2 ) ;
title(" h2 em função de t ", "fontsize", 4)
xlabel("tempo (s) " , "fontsize", 4)
ylabel("Nível do reservatório(m) ", "fontsize", 4)

f3 = scf(3);
plot (t , h1,'black',t,h2,'b' ) ;
legends(["Reservatório 1", "Reservatório 2"],[1,2],2)
title("Nível dos reservatórios - Método de Runge-Kutta", "fontsize", 4)
xlabel("Tempo(s)", "fontsize", 4)
ylabel("Nível do reservatório(m) ", "fontsize", 4)
xgrid(1);
```

Resultados

Para o exercício 2, novamente o evento foi simulado numericamente por dois métodos, Euler e Runge-Kutta, e com os mesmos 50000 segundos, mas agora o problema apresenta dois reservatórios, nota-se novamente grande semelhança nos resultados obtidos entre os métodos, como pode ser observado nos gráficos 3 e 4.

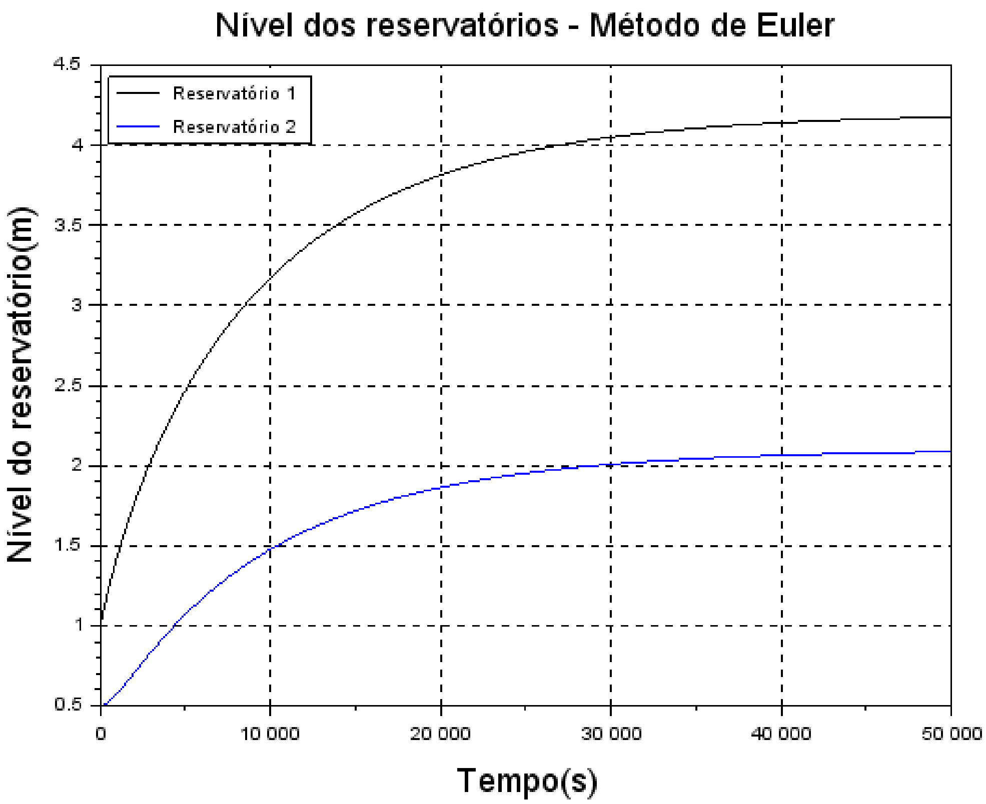

Gráfico 3: : Problema com 2 reservatórios - Método de Euler

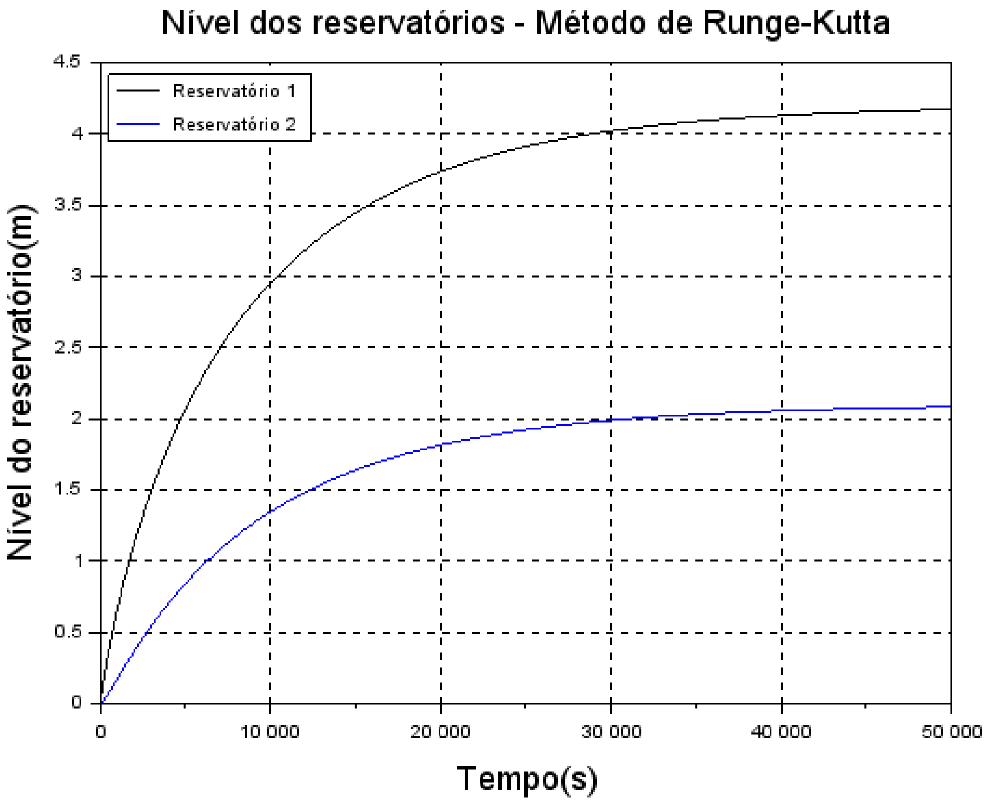

Gráfico 4: : Problema com 2 reservatórios - Método de Runge-Kutta