Vitória Menino Campos

Agenor de Toledo Fleury

PME 3380 - Modelagem de Sistemas Dinâmicos

02 de setembro de 2020

LISTA B

EXERCÍCIO 1:

Foi implementado um programa no Scilab que resolve a equação diferencial que modela o sistema de um tanque. Para o desenvolvimento do exercício foi utilizado um tempo que permite observar a estabilidade do nível do reservatório. Os gráficos a seguir mostram a solução numérica pelo método de Euler e pelo método de Runge-Kutta.

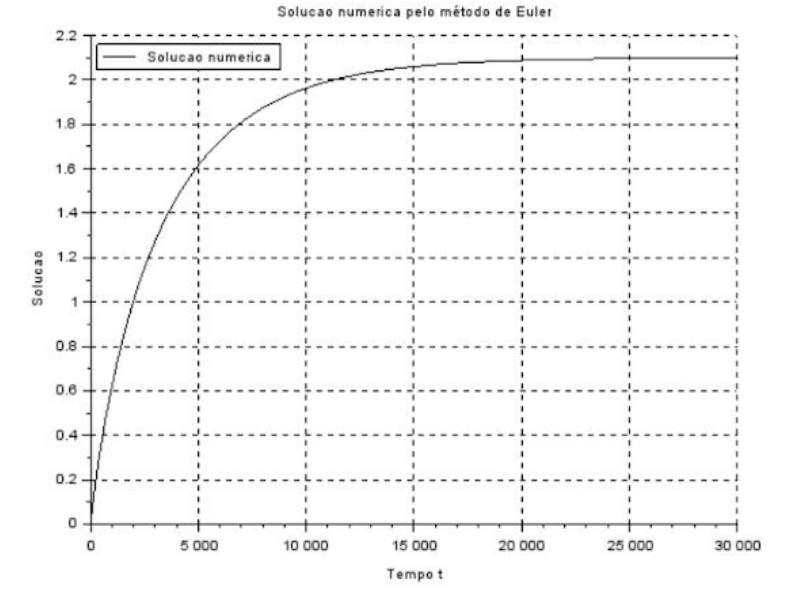

Figura 1: Gráfico solução numérica pelo método de Euler um tanque

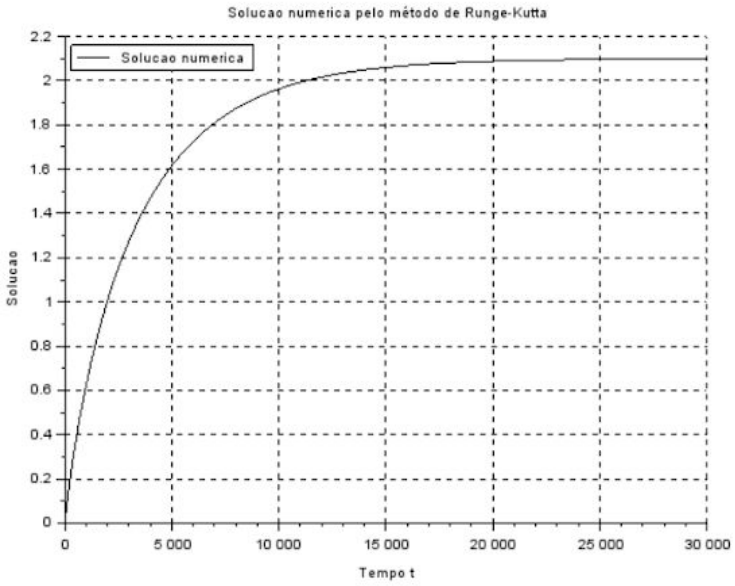

Figura 2: Gráfico solução numérica pelo método de Runge Kutta um tanque

A solução exata não é conhecida e nem pode ser comparada nos gráficos, pois não conseguimos isolar y(t) na seguinte equação solução:

$$c_1 - \frac{t}{S} = \frac{2\left(Q\log\left(Q - \sqrt{C\,y(t)}\right) + \sqrt{C\,y(t)}\right)}{C}$$

EXERCÍCIO 2:

Foi implementado um programa no Scilab que resolve as equações diferenciais que modela o sistema de dois tanques. Para o desenvolvimento do exercício foi utilizado um tempo que permite observar a estabilidade do nível do reservatório, também, como não estava especificado no enunciado, as áreas das seções transversais $S_1 \, e \, S_2$ foram consideradas $10\,m^2$ e $8\,m^2$

respectivamente e os valores de R_a *e* R_b foram considerados iguais ao R do primeiro exercício, $2 \times 10^8 Pa/(m^3/s)^2$. Os gráficos a seguir mostram a solução numérica pelo método de Euler e pelo método de Runge-Kutta.

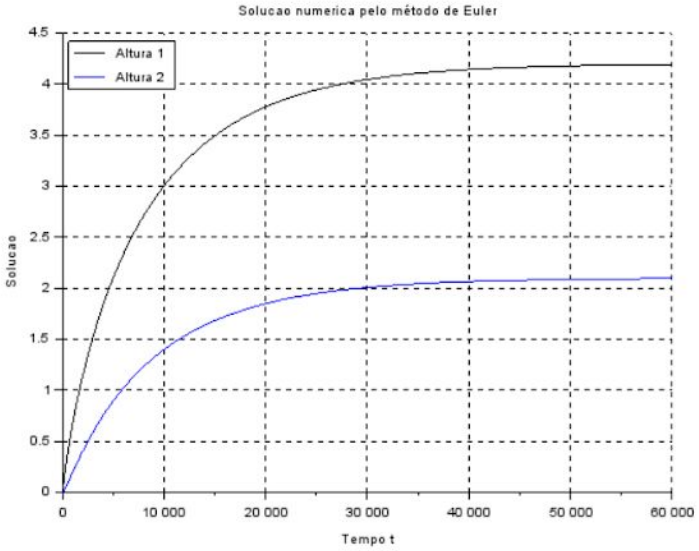

Figura 3: Gráfico solução numérica pelo método de Euler dois tanques

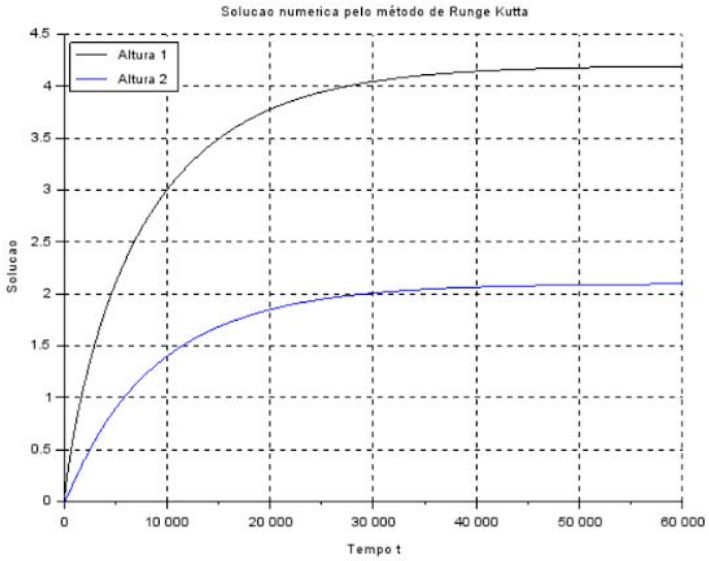

Figura 4: Gráfico solução numérica pelo método de Euler para dois tanques

CÓDIGOS:

- FUNÇÃO PARA EULER EXERCÍCIO 1:

```
function [ydot]=exel(y)
    ydot=(-((C*y)^(1/2))+Q)/S;
endfunction
save ('exel.sci','exel');
```

- EULER EXERCÍCIO 1:

```
// Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
// Apagando dados anteriores:
clear
// Carregando a equacao diferencial:
// Carregue a função usando o comando Load do Scilab
load('exel.sci','exel');
// Instante inicial:
C=(1000*10)/(2*10^8);
Q=0.010247;
S=10;
t(1)=0;
// Instante final:
tf=30000;
// 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 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*exel(y(i));
// Termino do comando for:
end
// Desenhando outro grafico com linhas diferentes:
plot2d([t],[y],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Solucao numerica pelo método de Runge-Kutta","Tempo t","Solucao","Solucao numerica");
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends(T(4),[1,2],2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
xgrid(1)
```

- RUNGE KUTTA EXERCÍCIO 1:

```
clear
//variáveis:
C=(1000*10)/(2*10^8);
Q=0.010247;
S=10;
// Instante inicial:
t(1)=0;
// Instante final:
tf=30000;
// 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*(-(C*(y(i)))^(1/2)+Q)/S;
    k2=h*(-(C*(y(i)+k1/2))^(1/2)+Q)/S;
    k3=h*(-(C*(y(i)+k2/2))^(1/2)+Q)/S;
    k4=h*(-(C*(y(i)+k3))^(1/2)+Q)/S;
    y(i+1)=y(i)+((k1+2*k2+2*k3+k4)/6);
// Termino do comando for:
end
// Desenhando outro grafico com linhas diferentes:
plot2d(t,y);
// Usando a variavel do tipo 'lista':
T=list("Solucao numerica pelo método de Euler","Tempo t","Solucao","Solucao numerica");
// Diminuindo a espessura das linhas:
xset('thickness',1)
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends(T(4),[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)
```

- FUNÇÃO PARA EULER EXERCÍCIO 2:

```
function[hdot]=exe2A(h1,h2)
    hdot=(Q-(C*(h1-h2))^(1/2))/S1;
endfunction
save ('exe2A.sci','exe2A');

function[hdot]=exe2B(h1,h2)
    hdot=((C*(h1-h2))^(1/2)-(W*h2)^(1/2))/S2;
endfunction
save ('exe2B.sci','exe2B');
```

- EULER PARA EXERCÌCIO 2:

```
// Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
// Apagando dados anteriores:
clear
load('exe2A.sci','exe2A');
load('exe2B.sci','exe2B');
//variáveis:
S1=10;
S2=8;
p=1000;
g=10;
Ra=2*10^8;
Rs=2*10^8;
Q=0.010247;
C=p*g/Ra
W=p*g/Rs
// Instante inicial:
t(1)=0;
// Instante final:
tf=60000;
// Condicao inicial:
h1(1)=0;
h2(1)=0;
ye(1)=0;
P=1;
n=round(tf/P);
for i=1:n
// Vetor de tempo:
    t(i+1)=t(i)+P;
    h1(i+1)=h1(i)+P*exe2A(h1(i),h2(i));
    h2(i+1)=h2(i)+P*exe2B(h1(i),h2(i));
end
// Desenhando outro grafico com linhas diferentes:
plot2d([t],[h1,h2],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Solucao numerica pelo método de Euler","Tempo t","Solucao","Altura 1","Altura 2");
// 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));
xgrid(1)
```

- RUNGE_KUTTA EXERCÍCIO 2:

```
clear
load('exe2A.sci','exe2A');
load('exe2B.sci','exe2B');
//variáveis:
S1=10
S2=8
p=1000
g=10
Ra=2*(10^8)
Rs=2*(10^8)
Q=0.010247
C=p*g/Ra;
W=p*g/Rs;
//Instante inicial:
t(1)=0;
//Instante final:
tf=50000;
//Condicao inicial:
h1(1)=0;
h2(1)=0;

//Passo de integracao (experimente alterar o passo):
l=0.5;
//Calculo de numero de passos):
n=round((tf-t(1))/l);
//Integracao numerica usando o metodo de Runge Kutta:
//Comando for:
for i=1:n
//Vetor de tempo:
    t(i+1)=t(i)+l;
//Solucao numerica:
    k1=l*exe2A(h1(i),h2(i));
    k2=l*exe2A(h1(i)+k1/2,h2(i)+k1/2);
    k3=l*exe2A(h1(i)+k2/2,h2(i)+k2/2);
    k4=l*exe2A(h1(i)+k3,h2(i)+k3);

    p1=l*exe2B(h1(i),h2(i));
    p2=l*exe2B(h1(i)+p1/2,h2(i)+p1/2);
    p3=l*exe2B(h1(i)+p2/2,h2(i)+p2/2);
    p4=l*exe2B(h1(i)+p3,h2(i)+p3);

    h1(i+1)=h1(i)+((k1+2*k2+2*k3+k4)/6);
    h2(i+1)=h2(i)+((k1+2*k2+2*k3+k4)/6);

//Termino do comando for:
end
//Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[h1,h2],[1 2]);
//Usando a variavel do tipo 'lista':
T=list("Solucao numerica pelo método de Runge Kutta","Tempo t","Solucao","Solucao numerica");
//Diminuindo a espessura das linhas:
xset('thickness',1)
//Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends(T(4),[1,2],2);
//Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
//Colocando uma grade no grafico:
```