

UNIVERSIDADE DE SÃO PAULO

ESCOLA POLITÉCNICA

VICTOR MANOEL FERREIRA ROSA DA COSTA

10772713

Modelagem de Sistemas Dinâmicos- Lista B

São Paulo

2020

Nestes exercícios pretende-se simular as soluções de equações diferenciais utilizando os métodos numéricos de Euler e Runge-Kutta de 4ª ordem.

#### 1- Sistema com um reservatório

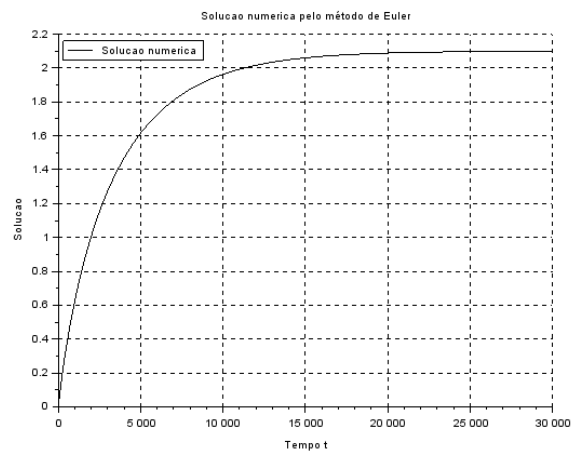
Ao estudar o sistema chega-se a seguinte equação diferencial:

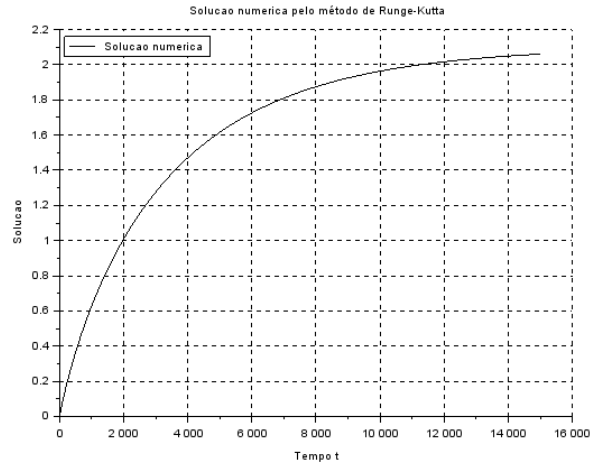
$$\dot{h} = \frac{1}{S} \left( Q_e - \sqrt{\frac{\rho g h}{R}} \right)$$

Onde:

- $h$  é o nível do reservatório [m]
- $S$  é a área da seção transversal do reservatório [ $10 \text{ m}^2$ ]
- $Q_e$  é a vazão de entrada [ $\text{m}^3/\text{s}$ ]
- $\rho$  é a massa específica da água [ $1000 \text{ kg/m}^3$ ]
- $g$  é a aceleração da gravidade [ $10 \text{ m/s}^2$ ]
- $R$  é a perda de carga [ $2 \cdot 10^8 \text{ Pa}/(\text{m}^3/\text{s})^2$ ]

Usando-se os algoritmos dos métodos numéricos, têm-se os seguintes resultados:





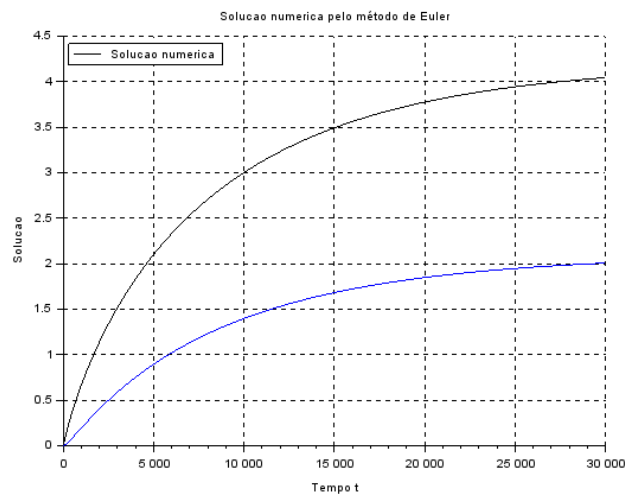
## 2- Sistema com dois reservatórios

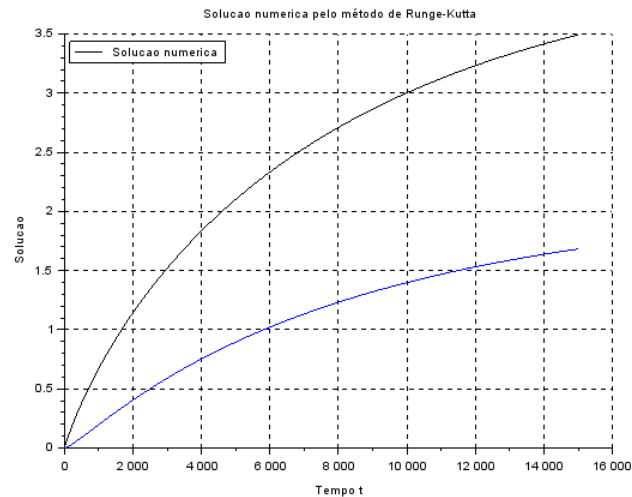
Adotando-se o mesmo procedimento do caso anterior, encontram-se duas equações diferenciais:

$$\dot{h}_1 = \frac{1}{S_1} \left( Q_e - \sqrt{\frac{\rho g (h_1 - h_2)}{R_a}} \right)$$

$$\dot{h}_2 = \frac{1}{S_2} \left( \sqrt{\frac{\rho g (h_1 - h_2)}{R_a}} - \sqrt{\frac{\rho g h_2}{R_s}} \right)$$

Os resultados foram:





### 3- Códigos:

Exercício 1 pelo método de Euler:

```
1 function [ydot]=ex(y)
2 .....ydot=(1/S)*(Qe-sqrt(C*y(i)));
3 endfunction
4 save('ex.sci','ex');
```

```
1 clear;
2
3 load('ex.sci','ex');
4
5 S=10;
6 R=2*(10^8);
7 g=10;
8 p=1000;
9 Qe=0.010247;
10 C=p*g/R;
11
12 t(1)=0;
13 tf=30000;
14 y(1)=0;
15 h=0.5;
16 n=round(tf/h);
17
18 for i=1:n
19 .....t(i+1)=t(i)+h;
20 .....y(i+1)=y(i)+h*ex(y);
21 end
22
23 plot2d([t],[y],[1:2]);
24 T=list("Solucao-numerica","Tempo-t","Solucao","Solucao-numerica");
25 xset("thickness",1);
26 legends(T(4),[1,2],2);
27 xtitle(T(1),T(2),T(3));
28 xgrid(1);
```

Exercício 1 pelo método de Runge-Kutta:

```
1 clear;
2
3 S=10;
4 R=2*(10^8);
5 g=10;
6 p=1000;
7 Qe=0.010247;
8 C=p*g/R;
9
10 t(1)=0;
11 tf=30000;
12 y(1)=0;
13 h=0.5;
14 n=round((tf-t(1)/h));
15
16 for i=1:n
17     t(i+1)=t(i)+h;
18     k1=h*(1/S)*(Qe-sqrt(C*y(i)));
19     k2=h*(1/S)*(Qe-sqrt(C*(y(i)+k1/2)));
20     k3=h*(1/S)*(Qe-sqrt(C*(y(i)+k2/2)));
21     k4=h*(1/S)*(Qe-sqrt(C*(y(i)+k3)));
22     y(i+1)=y(i)+((k1+2*k2+2*k3+k4)/6);
23 end
24
25 plot2d(t,y);
26 T=list("Solucao-numerica-pelo-metodo-de-Runge-Kutta","Tempo-t","Solucao","Solucao-numerica");
27 xset("thickness",1);
28 legends(T(4),[1,2],2);
29 xtitle(T(1),T(2),T(3));
30 xgrid(1);
```

Exercício 2 pelo método de Euler:

```
1 clear;
2
3 load('ex2A.sci','ex2A');
4 load('ex2B.sci','ex2B');
5
6 S1=10;
7 S2=8;
8 R=2*(10^8);
9 g=10;
10 p=1000;
11 Qe=0.010247;
12 C=p*g/R;
13
14 t(1)=0;
15 tf=30000;
16 y1(1)=0;
17 y2(1)=0;
18 h=0.5;
19 n=round(tf/h);
20
21 for i=1:n
22     t(i+1)=t(i)+h;
23     y1(i+1)=y1(i)+h*ex2A(y1,y2);
24     y2(i+1)=y2(i)+h*ex2B(y1,y2);
25 end
26
27 plot2d([t],[y1,y2],[1:2]);
28 T=list("Solucao-numerica-pelo-metodo-de-Euler","Tempo-t","Solucao","Solucao-numerica");
29 xset("thickness",1);
30 legends(T(4),[1,2],2);
31 xtitle(T(1),T(2),T(3));
32 xgrid(1);
```

Exercício 2 pelo método de Runge-Kutta:

```
1 clear;
2
3 S1=10;
4 S2=8;
5 R=2*(10^8);
6 g=10;
7 p=1000;
8 Qe=0.010247;
9 C=p*g/R;
10
11 t(1)=0;
12 tf=30000;
13 y1(1)=0;
14 y2(1)=0;
15 h=0.5;
16 n=round((tf-t(1)/h));
17
18 for i=1:n
19     t(i+1)=t(i)+h;
20     k1=h*(1/S1)*(Qe-sqrt(C*(y1(i)-y2(i)))));
21     k2=h*(1/S1)*(Qe-sqrt(C*((y1(i)-y2(i))+k1/2))));
22     k3=h*(1/S1)*(Qe-sqrt(C*((y1(i)-y2(i))+k2/2))));
23     k4=h*(1/S1)*(Qe-sqrt(C*((y1(i)-y2(i))+k3))));
24     y1(i+1)=y1(i)+((k1+2*k2+2*k3+k4)/6);
25
26     q1=h*(1/S2)*(sqrt(C*(y1(i)-y2(i)))-sqrt(C*y2(i))));
27     q2=h*(1/S2)*(sqrt(C*((y1(i)-y2(i))+q1/2)))-sqrt(C*(y2(i)+q1/2)));
28     q3=h*(1/S2)*(sqrt(C*((y1(i)-y2(i))+q2/2)))-sqrt(C*(y2(i)+q2/2)));
29     q4=h*(1/S2)*(sqrt(C*((y1(i)-y2(i))+q3)))-sqrt(C*(y2(i)+q3)));
30     y2(i+1)=y2(i)+((q1+2*q2+2*q3+q4)/6);
31 end
```