

PME 3380 – Modelagem

Lista B

Aluno: Luiz Ricardo de Sousa Cruz

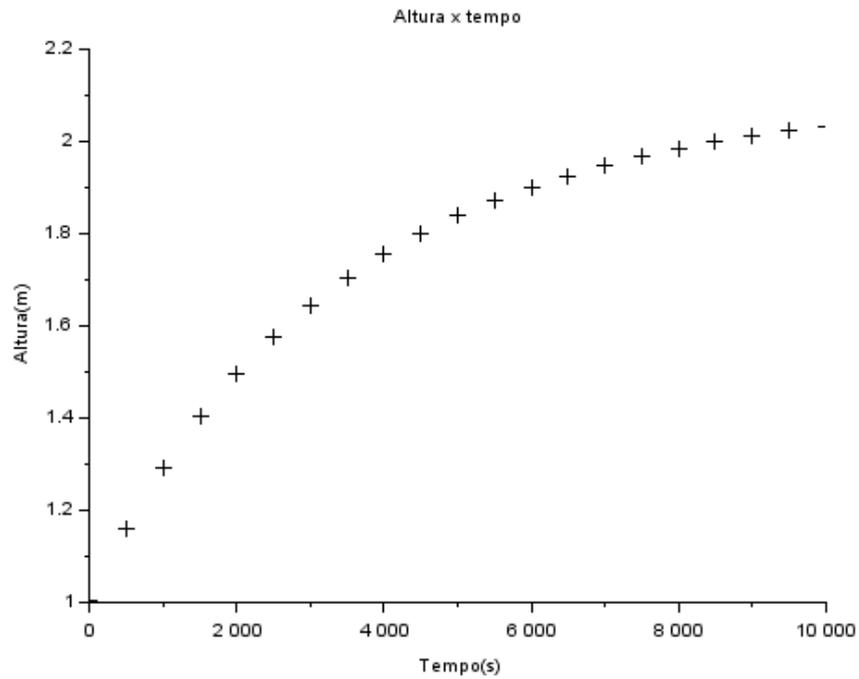
Nº USP: 10334961

1. Reservatório com Água

- Usando Euler

O código utilizado para simular o reservatório com água utilizando o método de Euler e o gráfico gerado mostrando a altura da coluna de água no reservatório por tempo de simulação estão explicitados abaixo:

```
1 //Euler
2 //Parâmetros
3 S = -10; //m²
4 R = -2*(10^8); //Pa/(m³/s)²
5 roh = 1000; //kg/m³
6 G = -10; //m/s²
7 Qe = -0.010247 //m³/s
8
9
10 function [ydot] = funcao(y)
11 ydot = (-sqrt(roh*G*y/R) + Qe)/S;
12 endfunction
13
14 //Estados iniciais
15 t(1) = 0;
16 tf = 10000;
17 y(1) = 1;
18
19 //Número de integrações do movimento
20 h = 500;
21 n = round((tf-t(1))/h);
22
23 //Método de Euler
24 for i = 1:n
25     t(i+1) = t(i) + h;
26     y(i+1) = y(i) + h*funcao(y(i));
27 end
28
29 plot2d(t,y,-1);
30 xtitle("Altura x tempo", "Tempo (s)", "Altura (m) ")
```



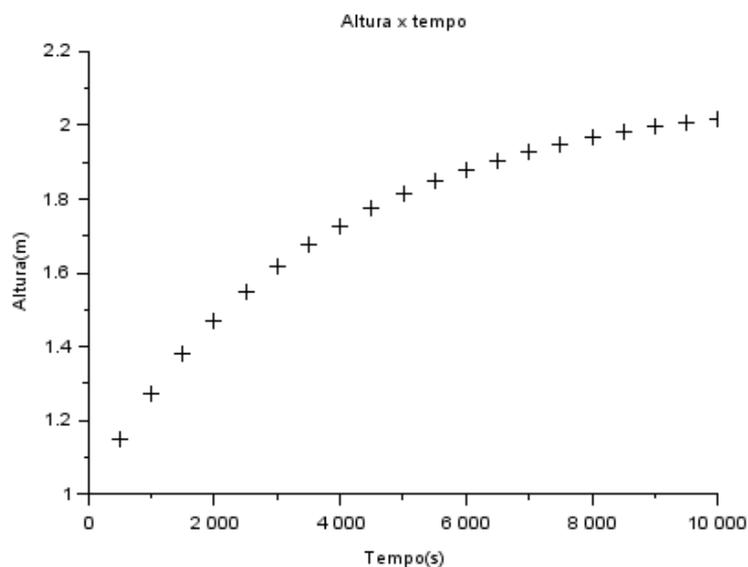
- **Runge - Kutta**

O código utilizado para simular o reservatório com água utilizando o método de Runge-Kutta e o gráfico gerado mostrando a altura da coluna de água no reservatório por tempo de simulação estão explicitados abaixo:

```

1 //Runge-Kutta
2 //Parâmetros
3 S = -10; //m²
4 R = -2*(10^8); //Pa/(m³/s)²
5 roh = 1000; //kg/m³
6 G = -10; //m/s²
7 Qe = -0.010247 //m³/s
8
9
10 function [ydot] = funcao(y)
11 ydot = (-sqrt(roh*G*y/R) + Qe)/S;
12 endfunction
13
14 //Estados iniciais
15 t(1) = 0;
16 tf = 10000;
17 y(1) = 1;
18
19 //Número de integrações do movimento
20 h = 500;
21 n = round((tf-t(1))/h);
22
23 //Método de Runge-Kutta
24 for i = 1:n
25     t(i+1) = t(i) + h;
26     k1 = h * (-sqrt(roh*G*y(i) / R) + Qe) / S;
27     k2 = h * (-sqrt(roh*G*(y(i) + k1/2) / R) + Qe) / S;
28     k3 = h * (-sqrt(roh*G*(y(i) + k2/2) / R) + Qe) / S;
29     k4 = h * (-sqrt(roh*G*(y(i) + k3/2) / R) + Qe) / S;
30     y(i+1) = y(i) + ((k1 + 2*k2 + 2*k3 + k4)/6);
31 end
32
33 plot2d(t, y, -1);
34 xtitle("Altura x tempo", "Tempo (s)", "Altura (m) ")

```

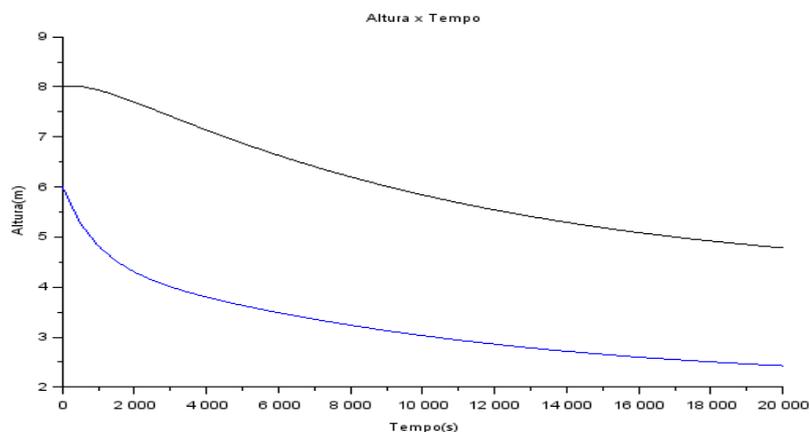


2. Dois Reservatórios com Água

- Usando Euler

O código utilizado para simular os reservatórios com água utilizando o método de Euler e o gráfico gerado mostrando a altura da coluna de água nos reservatórios por tempo de simulação estão explicitados abaixo:

```
1 roh = .1000; -//.kg/m^3
2 G = .10; -//.m/s^2
3 Ra = -.2*10^8; -//.Pa/(m^3/s^2)
4 Rs = -.2*10^8; -//.Pa/(m^3/s^2)
5 Qe = -.010247; -//.m^3/s
6 S1 = .10; -//.m^2
7 S2 = .5; -//.m^2
8
9
10 function [ydot] = funcao1(y1,y2)
11 ydot = -(Qe - sqrt(roh*G*(y1 - y2)/Ra))/S1
12 endfunction
13
14 function [ydot] = funcao2(y1,y2)
15 ydot = (sqrt(roh*G*(y1 - y2)/Ra) - sqrt(roh*G*(y2)/Rs))/S2
16 endfunction
17 //Estados iniciais
18 t(1)=0;
19 tf=20000;
20 y1(1)=8;
21 y2(1)=6;
22
23 //Número de integrações do movimento
24 h = .500;
25 n = round((tf-t(1))/h);
26
27 //Método de Euler
28 for i=1:n
29 ... t(i+1)=t(i)+h;
30 ... y1(i+1)=y1(i)+h*funcao1(y1(i),y2(i));
31 ... y2(i+1)=y2(i)+h*funcao2(y1(i),y2(i));
32 end
33
34 plot2d([t,t],[y1,y2],[1:2]);
35 xtitle("Altura x Tempo", "Tempo (s)", "Altura (m)");
```



- Usando Runge Kutta

O código utilizado para simular os reservatórios com água utilizando o método de Runge Kutta e o gráfico gerado mostrando a altura da coluna de água nos reservatórios por tempo de simulação estão explicitados abaixo:

```

1 roh = -1000; /// $\text{kg/m}^3$ 
2 G = -10; /// $\text{m/s}^2$ 
3 Ra = -2*10^8; /// $\text{Pa}/(\text{m}^3/\text{s}^2)$ 
4 Rs = -2*10^8; /// $\text{Pa}/(\text{m}^3/\text{s}^2)$ 
5 Qe = 0.010247; /// $\text{m}^3/\text{s}$ 
6 S1 = -10; /// $\text{m}^2$ 
7 S2 = 5; /// $\text{m}^2$ 
8
9
10 function [ydot] = funcao1(y1,y2)
11 ydot = (Qe - sqrt(roh*G*(y1 - y2)/Ra))/S1
12 endfunction
13
14 function [ydot] = funcao2(y1,y2)
15 ydot = (sqrt(roh*G*(y1 - y2)/Ra) - sqrt(roh*G*(y2)/Rs))/S2
16 endfunction
17
18 //Estados iniciais
19 t(1)=0;
20 tf=20000;
21 y1(1)=8;
22 y2(1)=6;
23
24 //Número de integrações do movimento
25 h = 500;
26 n = round((tf-t(1))/h);
27
28 //Método de Runge-Kutta
29 for i=1:n
30     t(i+1)=t(i)+h;
31     k1=h*((Qe - sqrt(roh*G*(y1(i) - y2(i))/Ra))/S1)
32     k2=h*((Qe - sqrt(roh*G*((y1(i) - y2(i)) + k1/2)/Ra))/S1)
33     k3=h*((Qe - sqrt(roh*G*((y1(i) - y2(i)) + k2/2)/Ra))/S1)
34     k4=h*((Qe - sqrt(roh*G*((y1(i) - y2(i)) + k3/2)/Ra))/S1)
35     y1(i+1)=y1(i)+((k1+2*k2+2*k3+k4)/6);
36     k1=h*((sqrt(roh*G*(y1(i) - y2(i))/Ra) - sqrt(roh*G*(y2(i))/Rs))/S2)
37     k2=h*((sqrt(roh*G*((y1(i) - y2(i)) + k1/2)/Ra) - sqrt(roh*G*(y2(i) + k1/2)/Rs))/S2)
38     k3=h*((sqrt(roh*G*((y1(i) - y2(i)) + k2/2)/Ra) - sqrt(roh*G*(y2(i) + k2/2)/Rs))/S2)
39     k4=h*((sqrt(roh*G*((y1(i) - y2(i)) + k3/2)/Ra) - sqrt(roh*G*(y2(i) + k3/2)/Rs))/S2)
40     y2(i+1)=y2(i)+((k1+2*k2+2*k3+k4)/6);
41 end
42
43 plot2d([t,t],[y1,y2],[1-2]);
44 legends(["h1","h2"],[1-2],4)
45 xtitle("Altura - x - tempo", "Tempo (s)", "Altura (m) ")

```

Altura x tempo

