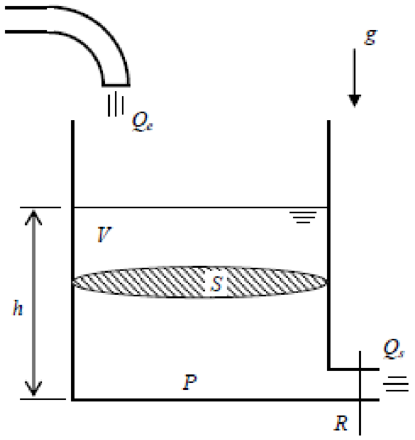
PME 3380 – Modelagem de Sistemas Dinâmicos

Lista B

Mariana Claudino Pin 9348644

1) Implemente um programa no *Scilab* que resolva numericamente a equação diferencial que modela o sistema abaixo, tanto pelo método de *Euler* como *Runge Kutta*.



Reservatório com água

Parâmetros:

$S = 10\ m^2$ - área da seção transversal (constante)

$R = 2x10^8 \frac{Pa}{\left(\frac{m^3}{s}\right)^2}$ - Parâmetro que relaciona vazão com queda de pressão (perda de carga)

$\rho = 1000\ kg/m3$ - Massa específica da água

$G = 10\ m/s^2$ - Aceleração da gravidade na superfície da terra

Variáveis:

$Qe = 0{,}010247\ m^3/s$ - Vazão de entrada

h: nível do reservatório $[m]$

V: volume de água no reservatório $[m^3]$

P: pressão relativa à atmosférica, no fundo do reservatório $[Pa]$

Qs: vazão de saída $[m^3/s]$

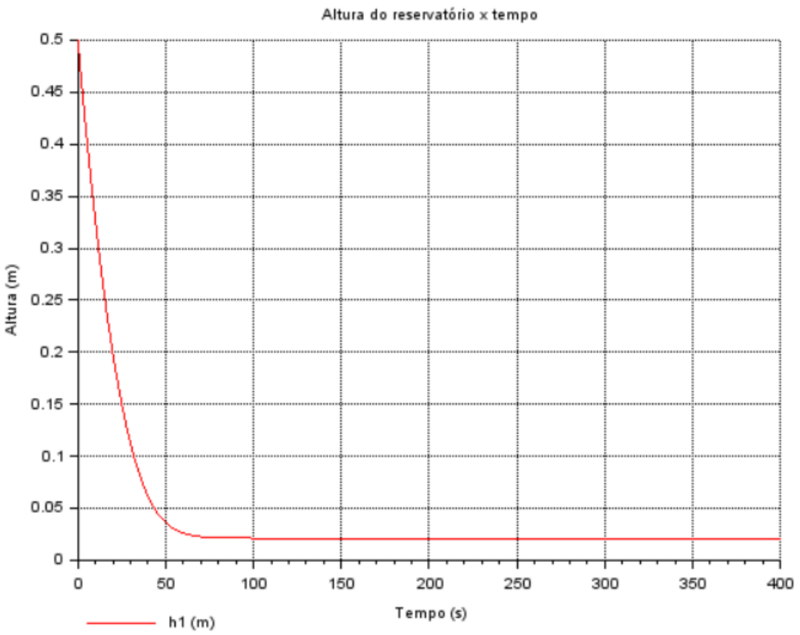
Admite-se que a água seja incompressível.

a) Euler

código:

```
// EULER

clear
// Parâmetros:
rho= 1000;
g= 10;
R= 2e8;
S= 0.2;
Qe=  0.0010247;

t(1)=0;
tf=400;
y(1)=0.5;
h=0.01;
n=round(tf/h);

// Integracao numerica usando o metodo de Euler:
function y_ponto=funcao(y)
    y_ponto=(-sqrt(rho*g*y/R)+Qe)/S
endfunction
for i=1:n
t(i+1)=t(i)+h;
y(i+1)=y(i)+h* funcao(y(i));
end

plot2d(t, y, [5], leg = "h1 (m)")
xtitle('Altura do reservatório x tempo', 'Tempo (s)', 'Altura (m)')
xgrid()

//
```

Gráfico:

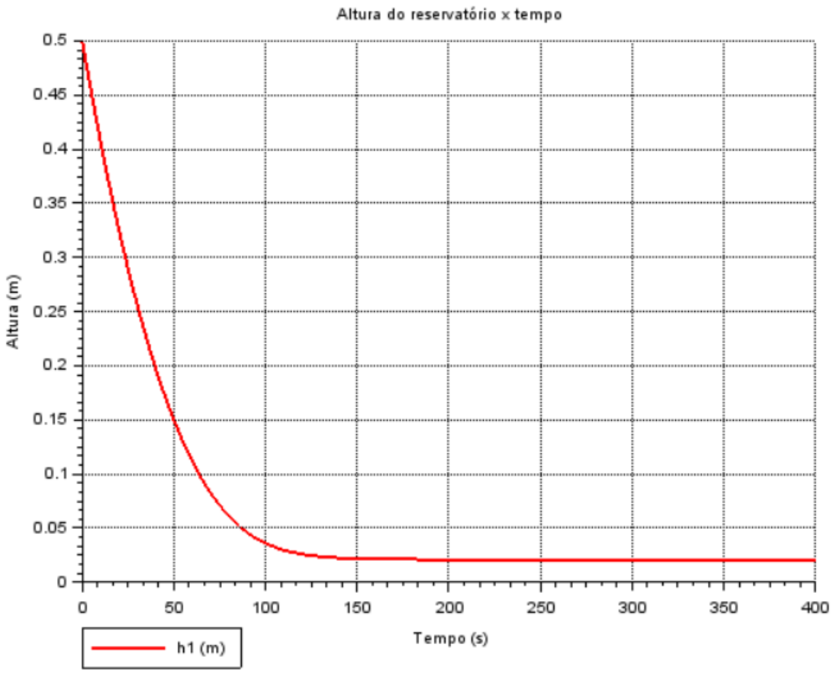


b) Runge Kutta

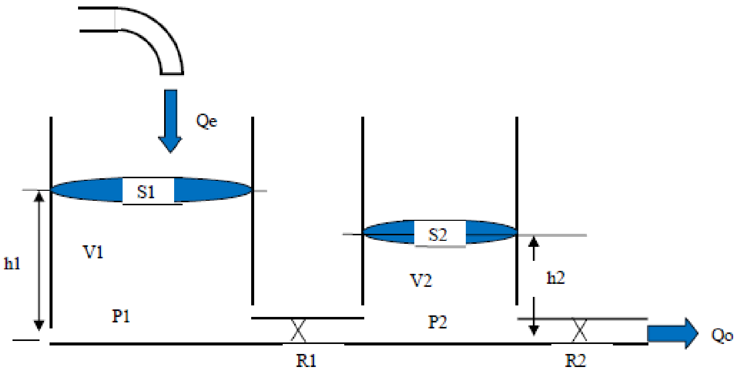
Código:

```
clc
clear

ti = 0; tf = 400; h = 0.01
t = [ti:h:tf]
hr = zeros(t);

// parâmetros do problema
S = 0.2;
R = 2D8;
ro = 1000;
g = 10;
Qe = 0.0010247;
C = ro*g/R

// valor inicial
hr(1) = 0.5;

// aplicação do Runge-Kutta:
for i = 1:(length(t) - 1)
k1 = (h/(2*S))*(Qe - sqrt((C*(hr(i)))));
k2 = (h/(2*S))*(Qe - sqrt((C*((hr(i) + k1/2)))));
k3 = (h/(2*S))*(Qe - sqrt((C*((hr(i) + k2/2)))));
k4 = (h/(2*S))*(Qe - sqrt((C*((hr(i) + k3)))));
hr(i + 1) = hr(i) + (k1 + 2*k2 + 2*k3 + k4)/6; // solução numérica de h1
end

plot2d(t, hr, [5], leg = "h1 (m)")
xtitle('Altura do reservatório x tempo', 'Tempo (s)', 'Altura (m)')
xgrid()
```

Gráfico:



2) Desenvolva um programa em *Scilab* que resolva numericamente o sistema de equações diferenciais que modela o sistema com dois reservatórios, usando tanto Euler como Runge Kutta.



Modelo do sistema de 2 reservatórios (considere a entrada constante e perdas de carga não lineares como no caso do ex. de 1 tanque).

$$\begin{cases} \dot{h}_1 = \left[Q_e - \sqrt{\dfrac{\rho g}{R_a}(h_1 - h_2)} \right] \dfrac{1}{S_1} \\ \dot{h}_2 = \left[\sqrt{\dfrac{\rho g}{R_a}(h_1 - h_2)} - \sqrt{\dfrac{\rho g}{R_s} h_2} \right] \dfrac{1}{S_2} \end{cases}$$

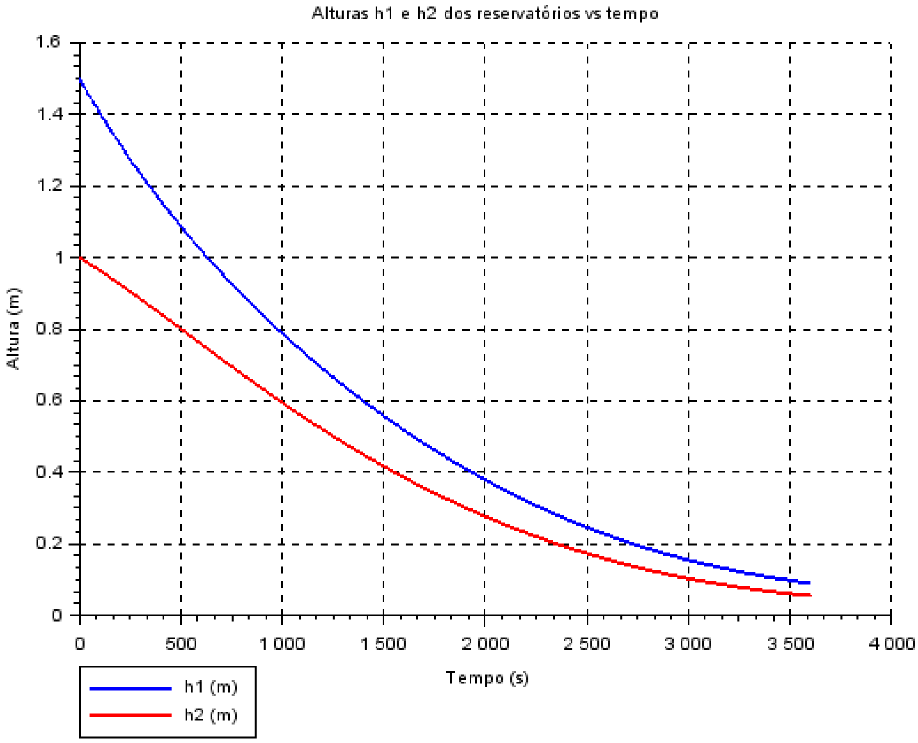
a) Euler

Código:

```
clear
// Parâmetros:
rho= 1000;
g= 10;
Ra= 2e8;
Rb= 2e8;
Sa= 2;
Sb= 3;
//Variáveis:
Qe=  0.0010247;

t(1)=0;
tf=3600;
y(1,1)=1.5;
y(1,2)=1;
h=0.01;
n=round(tf/h);
// Integracao numerica usando o metodo de Euler:

function y_pontoA=funcaoA(y)
   y_pontoA=(Qe - sqrt(rho*g*(y(1)-y(2))/Ra))/Sa
endfunction
function y_pontoB=funcaoB(y)
```

```
    y_pontoB=(sqrt(rho*g*(y(1)-y(2))/Ra)-sqrt(rho*g*y(2)/Rb))/Sb
endfunction
for i=1:n
t(i+1)=t(i)+h;

y(i+1,1)=y(i,1)+h* funcaoA(y(i,:));
y(i+1,2)=y(i,2)+h* funcaoB(y(i,:));
end

plot2d(t, [y(:,1)' y(:,2)'], [2 5], leg = "h1 (m) @ h2 (m)")
xtitle('Alturas h1 e h2 dos reservatórios vs tempo', 'Tempo (s)', 'Altura (m)')
xgrid()
```

Gráfico:



b) Runge Kutta

Código:

```
clc
clear

ti = 0; tf = 3600; h = 0.01
t = [ti:h:tf]
h1 = zeros(t); h2 = zeros(t);

// parâmetros do problema
S1 = 2;
S2 = 3;
R = 2D8;
```

```
ro = 1000;
g = 10;
Qe = 0.0010247;
C = ro*g/R

// valores iniciais
h1(1) = 1.5; h2(1) = 1

// aplicação do Runge-Kutta:
for i = 1:(length(t) - 1)
k1 = (h/(2*S1))*(Qe - sqrt((C*(h1(i) - h2(i)))));
q1 = (h/(2*S2))*(sqrt((C*(h1(i) - h2(i)))) - sqrt(C*h2(i)));
k2 = (h/(2*S1))*(Qe - sqrt((C*((h1(i) + k1/2) - (h2(i) + q1/2)))));
q2 = (h/(2*S2))*(sqrt((C*((h1(i) + k1/2) - (h2(i) + q1/2)))) - sqrt(C*(h2(i) + q1/2)));
k3 = (h/(2*S1))*(Qe - sqrt((C*((h1(i) + k2/2) - (h2(i) + q2/2)))));
q3 = (h/(2*S2))*(sqrt((C*((h1(i) + k2/2) - (h2(i) + q2/2)))) - sqrt(C*(h2(i) + q2/2)));
k4 = (h/(2*S1))*(Qe - sqrt((C*((h1(i) + k3) - (h2(i) + q3)))));
q4 = (h/(2*S2))*(sqrt((C*((h1(i) + k3) - (h2(i) + q3/2)))) - sqrt(C*(h2(i) + q3)));
h1(i + 1) = h1(i) + (k1 + 2*k2 + 2*k3 + k4)/6; // solução numérica de h1
h2(i + 1) = h2(i) + (q1 + 2*q2 + 2*q3 + q4)/6; // solução numérica de h2
end

plot2d(t, [h1' h2'], [2 5], leg = "h1 (m) @ h2 (m)")
xtitle('Alturas h1 e h2 dos reservatórios vs tempo', 'Tempo (s)', 'Altura (m)')
xgrid()
```

Gráfico: