Escola Politécnica da Universidade de São Paulo

Modelagem de sistemas dinâmicos

Lista B

Vítor Facchini
10772605

Professor: Décio Crisol e Agenor Fleury

São Paulo
2020

Sumário

Lista de Figuras

1 Introdução

Neste relatório serão descritos os procedimentos feitos para a realização da Lista B da disciplina de modelagem de sistemas dinâmicos. Esta lista teve como objetivo simular um sistema de reservatórios de água, contendo 2 reservatórios, 1 entrada e 1 saída de água. Para a integração numérica, utilizou-se inicialmente o método de Euler, posteriormente seguindo para o método de Runge-Kutta de segunda ordem.

2 Execução da tarefa

O problema a ser resolvido é obter a altura do nível da água em ambos os reservatórios (h_1, h_2), ilustrado na Figura 2.1.

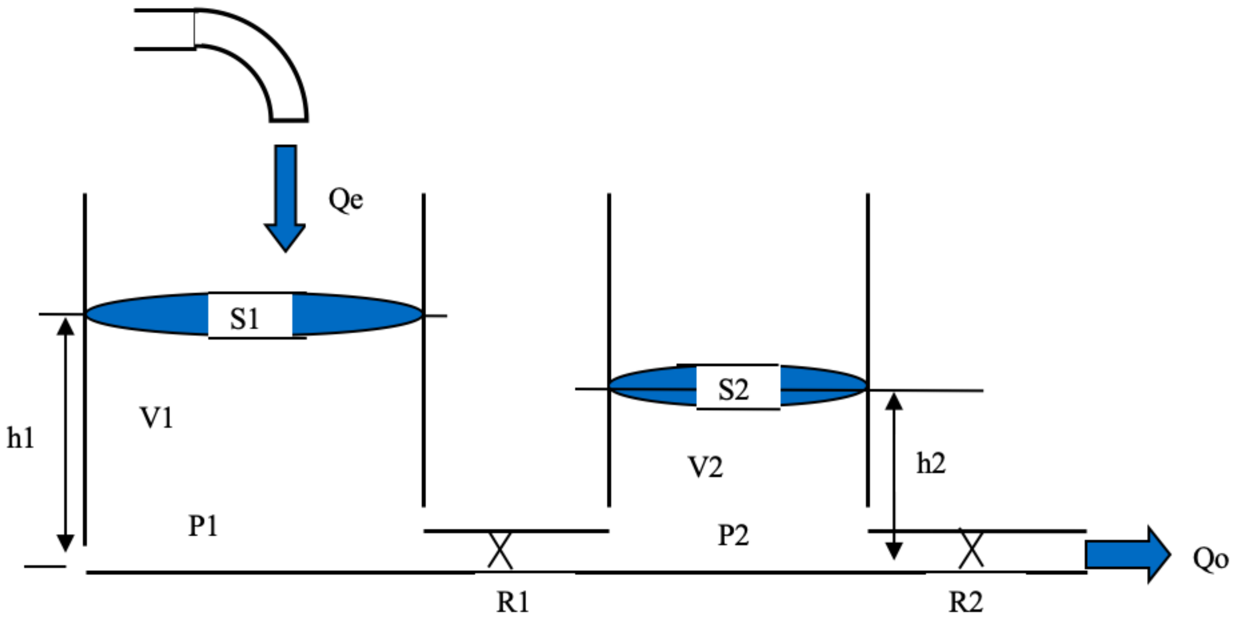

Figura 2.1: Ilustração do problema

Para resolver o problema, as equações fornecidas na lista foram resolvidas por meio de uma integração numérica, utilizando o método de Euler e Runge-Kutta. Para ambas as integrações foi utilizado o seguinte vetor de estados e sua derivada:

$$y = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \quad \dot{y} = \begin{bmatrix} \dfrac{Q_e - \sqrt{\dfrac{\rho g}{R_1}(h_1 - h_2)}}{S_1} \\ \\ \dfrac{\sqrt{\dfrac{\rho g}{R_1}(h_1 - h_2)} - \sqrt{\dfrac{\rho g}{R_1} h_2}}{S_1} \end{bmatrix}$$

E as condições iniciais adotadas foram:

- $R_1 = R_2 = 2 \cdot 10^8$;
- $h_{0_1} = 2\,m$;
- $h_{0_2} = 1\,m$
- Tempo de integração: 10 s

2.1 Método de Euler

Para o método de Euler, utilizou-se um $dt = 0,5$ s. Os resultados obtidos estão ilustrados na Figura 2.2

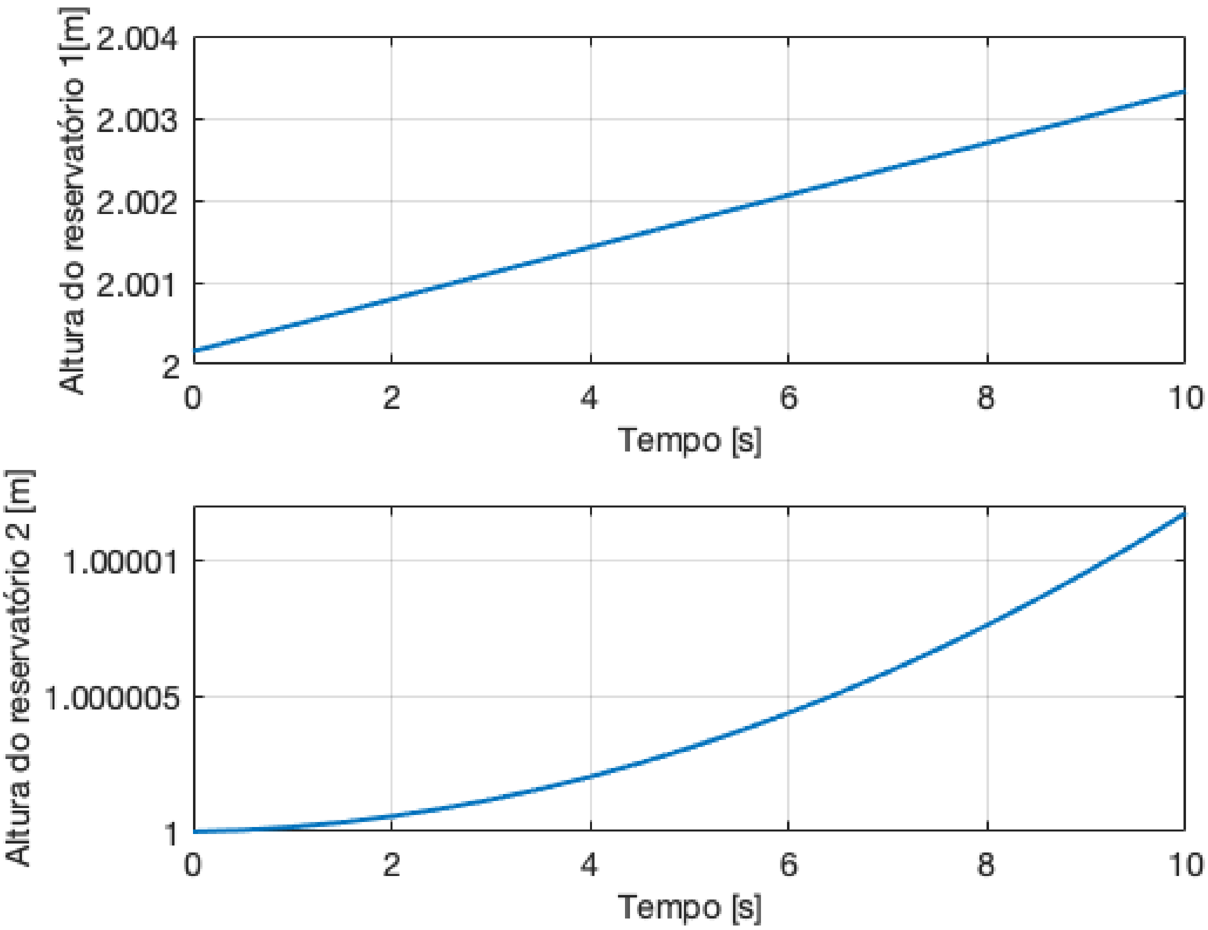

Figura 2.2: Alturas dos reservatórios pelo método de Euler

2.2 Método de Runge-Kutta

Para o método de Runge-Kutta de segunda ordem, também foi utilizado um $dt = 0,5$ s. Os resultados obtidos estão ilustrados na Figura 2.3

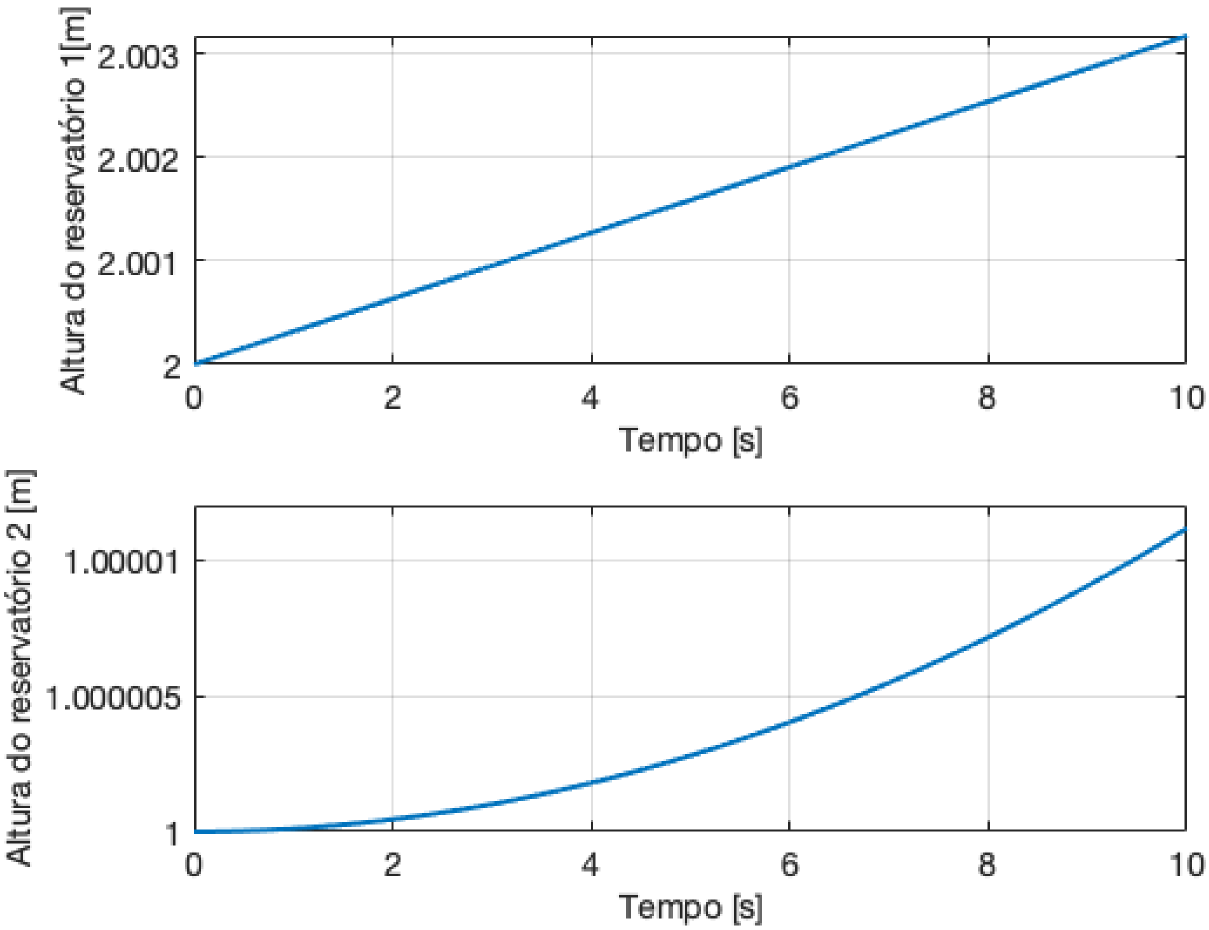

Figura 2.3: Alturas dos reservatórios pelo método de Runge-Kutta

3 Conclusão

Com os resultados obtidos, conclui-se que, os valores de resistência são bem elevados, acarrentando uma dificuldade em drenar a água dos reservatórios, fazendo com que o nível da água suba. Também verificou-se que a altura inicial dos reservatórios é muito influente, visto que caso a altura seja bem elevada (como 10 m, por exemplo, vide a Figura 3.1), o nível da água irá abaixar, visto a grande pressão da água, vencendo a elevada resistência de saída.

Também foi possível verificar que a variação da altura da água no segundo reservatório foi bem inferior à do primeiro, ou seja, as vazões de entrada e saída de água no segundo reservatório eram próximas, visto que o principal limitante à vazão era o coeficiente R, que tinha uma ordem de grandeza muito superior a qualquer outro parâmetro.

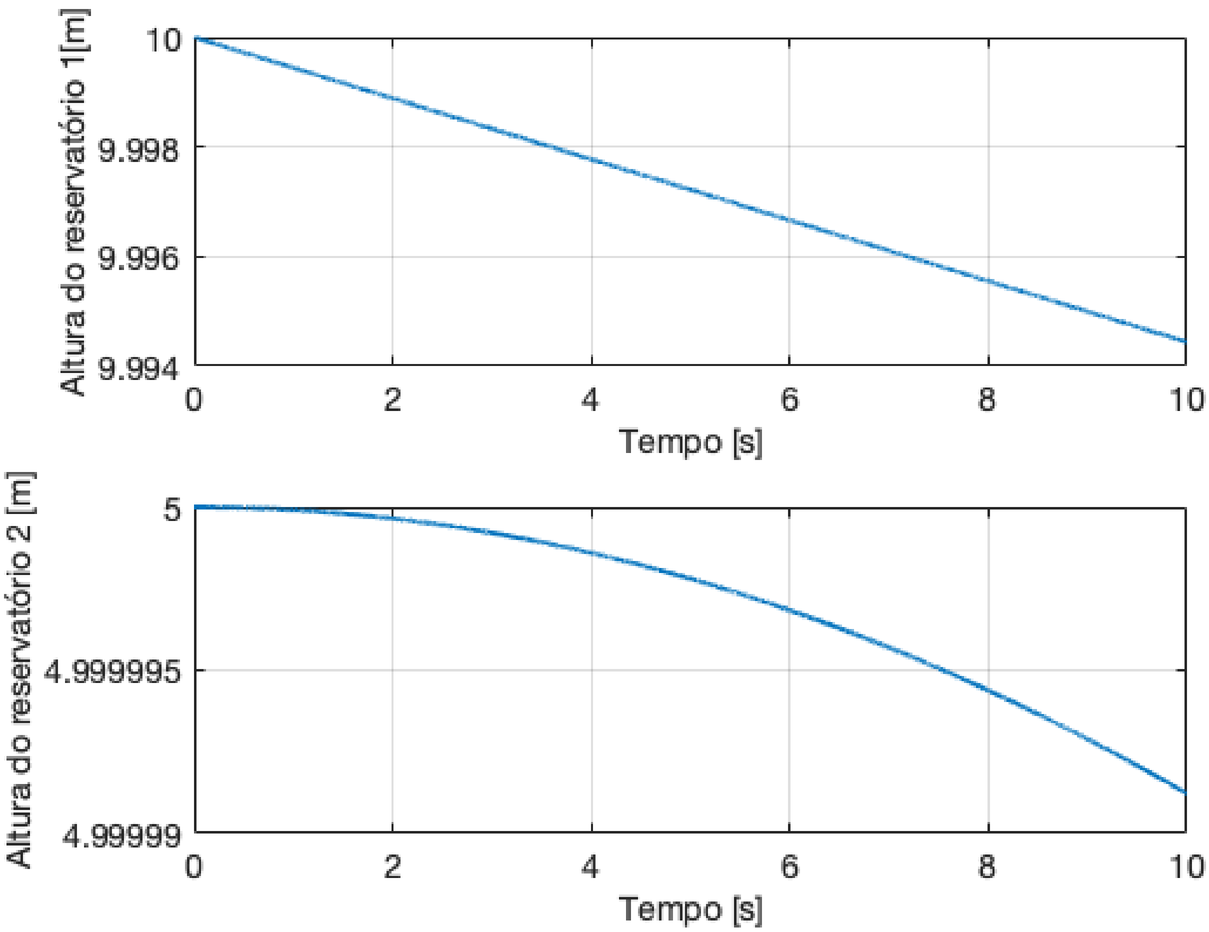

Figura 3.1: Altura dos reservatórios com $h_{0_1} = 10$ m e $h_{0_2} = 5$ m

4 Apêndice

Nesta seção será apresentado o código utilizado, a primeira metade foi usada para o método de Euler e a segunda para o método de Runge-Kutta.

```
function numericoE
euler=true;
if euler
    close all

    dt = .5;
    tempo = 0:dt:10;

    Y=[];

%% Variaveis
    Qe = 0.010247;
    h01 = 10;
    h02 = 5;

    %% Constantes
    g = 10;
    rho = 1000;
    Ra = 2E8;
    S1 = 10;
    S2 = S1/2;

    %% Integracao por Euler
    y = [h01;h02];
    for t=tempo
        yp = [(Qe - sqrt(rho*g*(y(1)-y(2))/Ra))/S1
            (sqrt(rho*g*(y(1)-y(2))/Ra) - sqrt(rho*g*(y(2))/Ra))/S2];

        y = y+yp*dt;
        Y = [Y y];
    end
    h1 = Y(1,:);
    h2 = Y(2,:);
    subplot(2,1,1)
    plot(tempo,h1,'LineWidth',2)
    xlabel('Tempo [s]')
    ylabel('Altura do reservatorio 1[m]')
    grid on
    subplot(2,1,2)
```

```
    plot(tempo,h2,'LineWidth',2)
    xlabel('Tempo [s]')
    ylabel('Altura do reservatorio 2 [m]')
    grid on
else
    close all

    dt = .5;
    tempo = 0:dt:10;

    Y=[];

    %% Variaveis
    Qe = 0.010247;
    h01 = 2;
    h02 = 1;

    %% Constantes
    g = 10;
    rho = 1000;
    Ra = 2E8;
    S1 = 10;
    S2 = S1/2;

    %% Integracao por Euler
    y = [h01;h02];
    for t=tempo

        k1 = dt*y;
        k2 = dt*(y+k1/2);
        k3 = dt*(y+k2/2);
        k4 = dt*(y+k3);
        y = y+(k1+2*k2+2*k3+k4)/6;


        yp = [(Qe - sqrt(rho*g*(y(1)-y(2))/Ra))/S1
            (sqrt(rho*g*(y(1)-y(2))/Ra) - sqrt(rho*g*(y(2))/Ra))/S2];

        y = y+yp*dt;
        Y = [Y y];
    end
    h1 = Y(1,:);
    h2 = Y(2,:);
    subplot(2,1,1)
```

```
    plot(tempo,h1,'LineWidth',2)
    xlabel('Tempo [s]')
    ylabel('Altura do reservatorio 1[m]')
    grid on
    subplot(2,1,2)
    plot(tempo,h2,'LineWidth',2)
    xlabel('Tempo [s]')
    ylabel('Altura do reservatorio 2 [m]')
    grid on

end
```