Lista B

Lucas Souza Vieira
N° USP: 10772863
PME3380 - Modelagem de Sistemas Dinâmicos
Escola Politécnica da Universidade de São Paulo

27 de agosto de 2020

1 Um reservatório

Para obtenção da solução numérica do primeiro problema, basta implementar o algoritmo de integração numérica presente nas equações apresentadas na parte teórica da lista. Fixando os parâmetros do problema e variando a condição inicial $h(t = 0) = h_0$, são obtidas diferentes soluções. O tamanho do passo de integração também interfere na semelhança entre as soluções pelo método de Euler e pelo método de Runge-Kutta de quarta ordem.

Arbitrou-se um tempo de simulação de 10000 segundos, com instante inicial em 0 e passo de integração $\Delta t = 10\ s$. Os demais parâmetros do sistema são dados pelo enunciado, com exceção da altura inicial do reservatório. A equação diferencial é dada por (1). Trata-se de um problema de valor inicial.

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \left(-\sqrt{\frac{\rho g h}{R}} + Q_e \right) \frac{1}{S} \tag{1}$$

Veja que, se escolhermos $h_0 = \frac{R{Q_e}^2}{\rho g} \approx 2,1\ m$, a taxa de variação da altura é nula e o reservatório mantém nível constante ao longo do tempo. Simulamos, portanto, a resposta do sistema para condição inicial de altura menor e maior que o valor de $2,1\ m$. Arbitrando $h_0 = 3,0\ m$ e $h_0 = 1,0\ m$, são apresentados os resultados de execução do código, respectivamente nas Figuras 1 e 2. Dado o passo de integração pequeno em relação ao intervalo de tempo simulado, os resultados dos métodos de Euler e Runge-Kutta são visualmente indistinguíveis na escala adotada para os gráficos.

Figura 1: Resultado de integração com um reservatório para $h_0 = 3,0\ m$

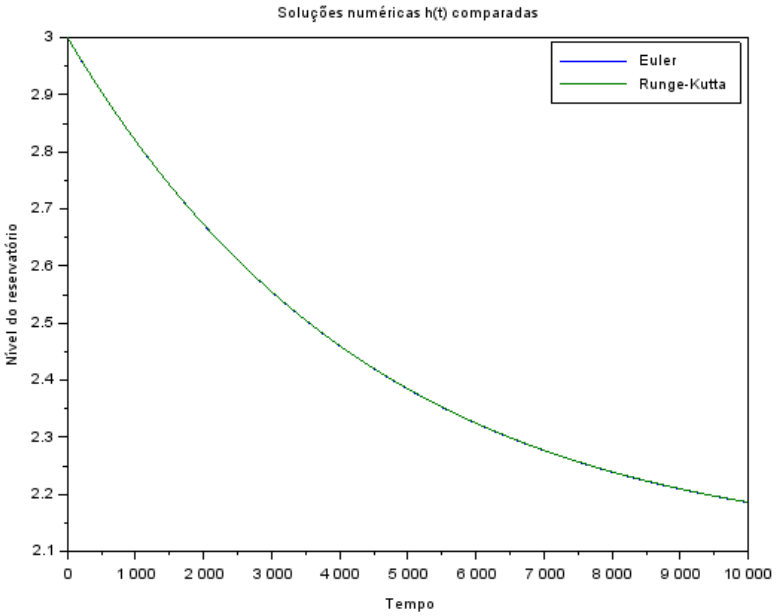
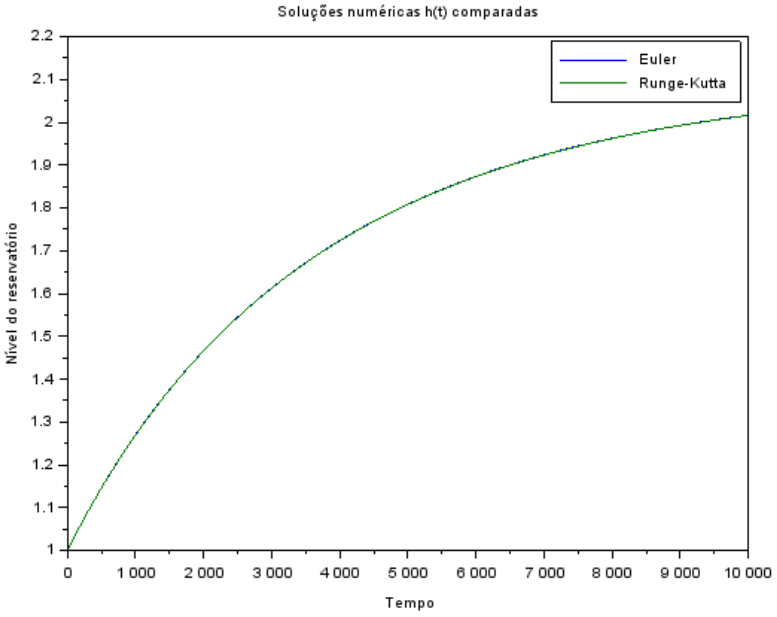


Figura 2: Resultado de integração com um reservatório para $h_0 = 1,0\ m$



Como pode-se perceber, a diferença entre as vazões de entrada e saída é responsável pela variação no nível do reservatório. Sendo a vazão Q_e constante, a vazão de saída Q_s varia ao longo do tempo de modo que, para um tempo tendendo ao infinito, as vazões se equilibram. Simultaneamente, o nível do reservatório tende ao nível de equilíbrio de aproximadamente $2,1\ m$, diminuindo no caso em que $h_0 > 2,1\ m$ e aumentando se $h_0 < 2,1\ m$.

2 Dois reservatórios

Para o problema com dois reservatórios, temos um sistema com dois graus de liberdade. Desta forma, são obtidas duas equações diferenciais acopladas, indicadas na forma de vetores em (2). A solução descreve a evolução temporal dos níveis dos reservatórios 1 e 2. Resolvemos o sistema pelos métodos de Euler e de Runge-Kutta. Para facilitar a descrição algébrica do algoritmo de integração, reescreve-se o lado direito da igualdade (2) como feito em (3). As funções f_i poderiam ser também funções do tempo, mas no sistema tratado são apenas funções das variáveis h_1 e h_2.

$$\begin{pmatrix} \dot{h}_1 \\ \dot{h}_2 \end{pmatrix} = \begin{pmatrix} \left[Q_e - \sqrt{\frac{\rho g}{R_a}(h_1 - h_2)} \right] \frac{1}{S_1} \\ \left[\sqrt{\frac{\rho g}{R_a}(h_1 - h_2)} - \sqrt{\frac{\rho g}{R_s} h_2} \right] \frac{1}{S_2} \end{pmatrix} \tag{2}$$

$$\begin{pmatrix} \dot{h}_1 \\ \dot{h}_2 \end{pmatrix} = \begin{pmatrix} f_1(h_1, h_2) \\ f_2(h_1, h_2) \end{pmatrix} \tag{3}$$

Para o método de Euler, os níveis dos reservatório em um instante t_{i+1} são calculados segundo a equação (4):

$$\begin{pmatrix} h_1^{i+1} \\ h_2^{i+1} \end{pmatrix} = \begin{pmatrix} h_1^{i} \\ h_2^{i} \end{pmatrix} + \begin{pmatrix} \Delta t f_1(h_1, h_2) \\ \Delta t f_2(h_1, h_2) \end{pmatrix} \tag{4}$$

No método de Runge-Kutta, é preciso calcular, a cada i-ésima iteração, os fatores $k_{m,1}$ e $k_{m,2}$, sendo $m = 1, ..., 4$ como no caso de apenas uma equação diferencial. No código, esses fatores são organizados em uma matriz $[k]_i$, indicada em (5), cujas entradas são atualizadas a cada iteração. Estas são calculadas segundo (6). Calculadas todas as entradas de $[k]_i$, determinam-se os níveis dos reservatórios em t_{i+1} pela equação (7).

$$[k]_i = \begin{pmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \\ k_{3,1} & k_{3,2} \\ k_{4,1} & k_{4,2} \end{pmatrix} \tag{5}$$

$$[k]_i = \Delta t \begin{pmatrix} f_1(h_1^i, h_2^i) & f_2(h_1^i, h_2^i) \\ f_1(h_1^i + \frac{1}{2}k_{1,1}, h_2^i + \frac{1}{2}k_{1,2}) & f_2(h_1^i + \frac{1}{2}k_{1,1}, h_2^i + \frac{1}{2}k_{1,2}) \\ f_1(h_1^i + \frac{1}{2}k_{2,1}, h_2^i + \frac{1}{2}k_{2,2}) & f_2(h_1^i + \frac{1}{2}k_{2,1}, h_2^i + \frac{1}{2}k_{2,2}) \\ f_1(h_1^i + k_{3,1}, h_2^i + k_{3,2}) & f_2(h_1^i + k_{3,1}, h_2^i + k_{3,2}) \end{pmatrix} \tag{6}$$

$$\begin{pmatrix} h_1^{i+1} \\ h_2^{i+1} \end{pmatrix} = \begin{pmatrix} h_1^i \\ h_2^i \end{pmatrix} + \frac{1}{6} \begin{pmatrix} k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1} \\ k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2} \end{pmatrix} \tag{7}$$

Ambos os métodos de integração foram implementados no código em Scilab pedido. Novamente arbitraram-se alguns parâmetros do sistema para verificar o comportamento de sua resposta. Tomou-se a mesma área de seção transversal para os dois tanques, iguais à dimensão do tanque do exercício anterior. Arbitrou-se escolha análoga para a perda de carga e para a vazão constante de entrada. Para as condições iniciais, escolheu-se $h_1^0 = 5,0\ m$ e $h_2^0 = 2,0\ m$. O tempo de simulação e passo de integração são os mesmos do exercício anterior. Os resultados são apresentados nas Figuras 3 e 4, em que o a primeira se refere ao reservatório 1 e a segunda, ao reservatório 2. A diferença entre as soluções por Euler e Runge-Kutta são pouco perceptíveis nesta escala.

Figura 3: Resultado de integração para o reservatório 1 com $h_0^1 = 5,0\ m$ e $h_0^2 = 2,0\ m$

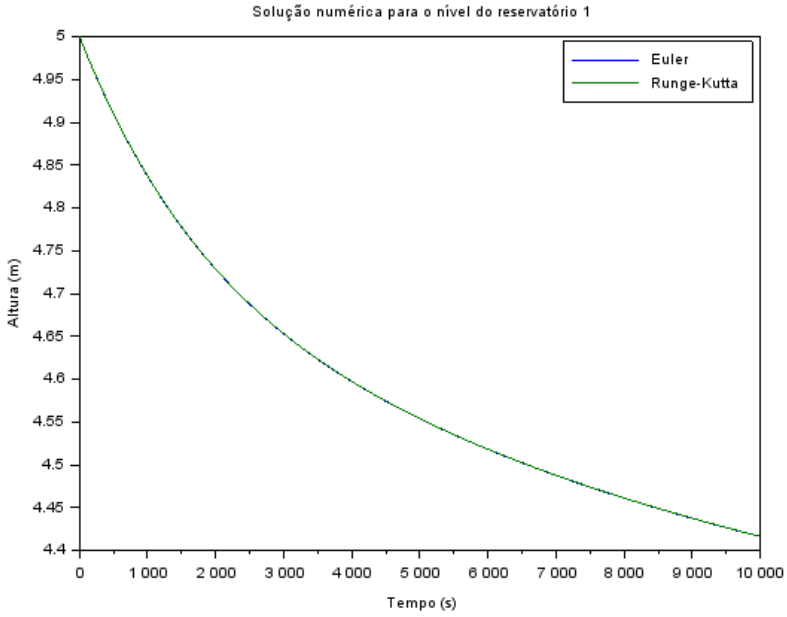
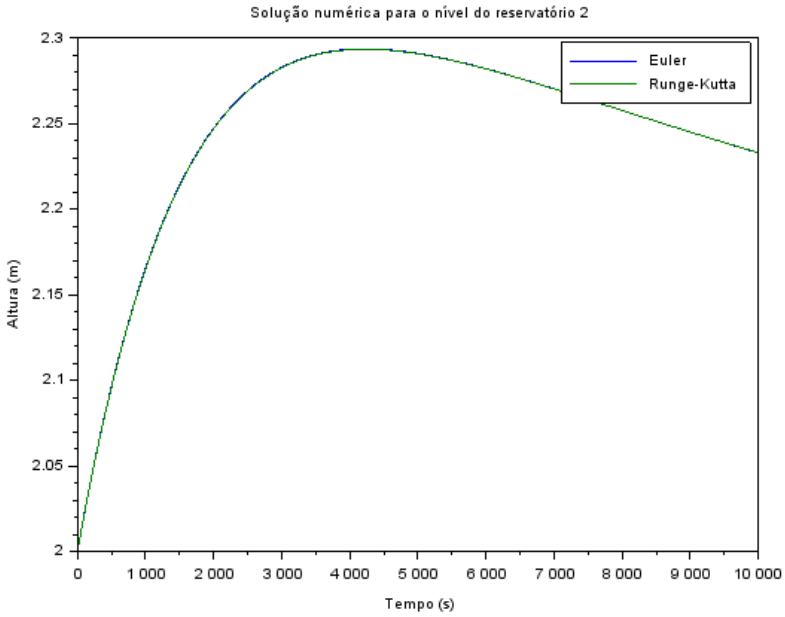


Figura 4: Resultado de integração para o reservatório 2 com $h_0^1 = 5,0\ m$ e $h_0^2 = 2,0\ m$



O volume cerca de 16 vezes maior do reservatório 1 ocasiona uma pressão maior em seu fundo e, portanto, seu nível segue diminuindo ao longo da simulação. A grande diferença de pressões propicia uma vazão considerável para o reservatório 2, que é maior do que a sua vazão de saída até atingir um máximo. Após esse máximo, a pressão obtida no fundo do reservatório 2 (devida ao aumento do nível), em conjunto com a dminuição do nível do reservatório 1, faz com que o balanço de vazões seja maior no sentido da vazão de saída do sistema. Para o tempo tendendo ao infinito, os níveis de cada reservatório tendem a seus respectivos valores de equilíbrio, supondo que Q_e é constante.

Anexo

Código usado para obtenção da solução numérica do problema com um reservatório:

```
// Lucas Souza Vieira
// Num. USP: 10772863
// PME3380 - Modelagem de Sistemas Dinamicos
// Codigo para resolucao do primeiro exercicio da Lista B

// Apagando dados anteriores
clear
xdel(winsid())

// Definicao das constantes (unidades no SI)
S = 10.0;
R = 2.0*10^8;                   // Perda de carga do reservatorio
rho = 1000.0;                   // Area da secao transversal do reservatorio
g = 10.0;                       // Aceleracao da gravidade na superficie da Terra
Qe = 0.010247;                  // Vazao de entrada
```

```
16
17 // Definicao de tempo de simulacao e numero de iteracoes
18 ti = 0.0;                          // Instante de tempo inicial
19 tf = 10000;                        // Instante de tempo final
20 Delta_t = 10;                      // Passo temporal de integracao
21 n_iterat = round((tf-ti)/Delta_t);  // Numero de iteracoes de integracao
22 vec_size = n_iterat + 1;           // Numero de entradas dos vetores integrados
23 t = linspace(ti,tf,n_iterat+1); // Vetor com os instantes de tempo
24
25 // Definicao de condicoes iniciais
26 h0 = 1.;                           // Altura inicial de agua no reservatorio
27 h_euler = zeros(vec_size);       // Inicializacao do vetor com a solucao por Euler
28 h_euler(1) = h0;                 // Condicao inicial para Euler
29 h_rk = zeros(vec_size);          // Inicializacao do vetor com a solucao por RK
30 h_rk(1) = h0;                    // Condicao inicial para RK
31
32
33 // Laco de integracao
34 for i = 1:n_iterat;
35     h_euler(i+1) = h_euler(i) + Delta_t*(-sqrt(rho*g*h_euler(i)/R)+Qe)/S;
36     k1 = Delta_t*(-sqrt(rho*g*h_rk(i)/R)+Qe)/S;
37     k2 = Delta_t*(-sqrt(rho*g*(h_rk(i)+k1/2.)/R)+Qe)/S;
38     k3 = Delta_t*(-sqrt(rho*g*(h_rk(i)+k2/2.)/R)+Qe)/S;
39     k4 = Delta_t*(-sqrt(rho*g*(h_rk(i)+k3)/R)+Qe)/S;
40     h_rk(i+1) = h_rk(i) + (k1+2.*k2+2.*k3+k4)/6;
41 end
42
43 //Plot das solucoes de Euler e RK para o nivel do reservatorio
44 f0 = scf(0);                       // Inicia a janela grafica 0
45 plot(t,h_euler',t,h_rk');          // Plota os dados
46 legend(["Euler","Runge-Kutta"]); // Legenda das curvas
47 //Define o titulo do grafico
48 xtitle("Solu  es num ricas h(t) comparadas","Tempo","N vel do reservat rio");
```

Código usados para obtenção da solução numérica do problema com dois reservatórios:

```
1  // Lucas Souza Vieira
2  // Num. USP: 10772863
3  // PME3380 - Modelagem de Sistemas Dinamicos
4  // Codigo para resolucao do segundo exercicio da Lista B
5
6  // Apagando dados anteriores
7  clear
8  xdel(winsid())
9
10 // Definicao das constantes (unidades no SI)
11 Ra = 2*10^8;                      // Perda de carga do reservatorio 1
12 Rs = 2*10^8;                      // Perda de carga do reservatorio 1
13 S1 = 10.0;                        // Area da secao transversal do reservatorio 1
14 S2 = 10.0;                        // Area da secao transversal do reservatorio 2
15 rho = 1000.0;                     // Densidade da agua
16 g = 10.0;                         // Aceleracao da gravidade na superficie da Terra
17 Qe = 0.010247;                    // Vazao de entrada
18
19 // Definicao de tempo de simulacao e numero de iteracoes
20 ti = 0.0;                          // Instante de tempo inicial
21 tf = 10000;                        // Instante de tempo final
22 Delta_t = 10.;                     // Passo temporal de integracao
23 n_iterat = round((tf-ti)/Delta_t);  // Numero de iteracoes de integracao
24 vec_size = n_iterat + 1;           // Numero de entradas dos vetores integrados
25 t = linspace(ti,tf,n_iterat+1);     // Vetor com os instantes de tempo
26
27 //Funcao para o calculo da derivada temporal de h1 em funcao de h1 e h2
28 function dw1 = hp1(h1,h2)
29     dw1 = (Qe-sqrt(rho*g*(h1-h2)/Ra))/S1;
30 endfunction
31
```

```
//Funcao para o calculo da derivada temporal de h2 em funcao de h1 e h2
function dw2 = hp2(h1,h2)
    dw2 = (sqrt(rho*g*(h1-h2)/Ra)-sqrt(rho*g*h2/Rs))/S2
endfunction

// Definicao de condicoes iniciais
h01 = 5.0;                  // Nivel inicial do reservatorio 1
h02 = 2.0;                   // Nivel inicial do reservatorio 2
h0 = [h01,h02];             // Vetor com as condicoes iniciais

// Matriz com as variaveis auxiliares de integracao kij do metodo RK
k = zeros(4,2);

//Inicializacao da matris h_e cujas colunas representam
    //os vetores integrados de h1 e h2 pelo metodo de Euler
h_e = zeros(vec_size,2);

//Inicializacao da matris h_e cujas colunas representam
    //os vetores integrados de h1 e h2 pelo metodo de RK
h_rk = zeros(vec_size,2);

h_e(1,1) = h0(1);       //Altura inicial h1(t=0) para Euler
h_e(1,2) = h0(2);       //Altura inicial h2(t=0) para Euler
h_rk(1,1)= h0(1);       //Altura inicial h1(t=0) para Runge-Kutta
h_rk(1,2)= h0(2);       //Altura inicial h2(t=0) para Runge-Kutta

// Laco de integracao
for i=1:n_iterat;
    h_e(i+1,1) = h_e(i,1)+Delta_t*hp1(h_e(i,1),h_e(i,2));
    h_e(i+1,2) = h_e(i,2)+Delta_t*hp2(h_e(i,1),h_e(i,2));
    k(1,1) = Delta_t*hp1(h_rk(i,1),h_rk(i,2));
    k(1,2) = Delta_t*hp2(h_rk(i,1),h_rk(i,2));
    k(2,1) = Delta_t*hp1(h_rk(i,1)+k(1,1)/2,h_rk(i,2)+k(1,2)/2);
    k(2,2) = Delta_t*hp2(h_rk(i,1)+k(1,1)/2,h_rk(i,2)+k(1,2)/2);
    k(3,1) = Delta_t*hp1(h_rk(i,1)+k(2,1)/2,h_rk(i,2)+k(2,2)/2);
    k(3,2) = Delta_t*hp2(h_rk(i,1)+k(2,1)/2,h_rk(i,2)+k(2,2)/2);
    k(4,1) = Delta_t*hp1(h_rk(i,1)+k(3,1),h_rk(i,2)+k(3,2));
    k(4,2) = Delta_t*hp2(h_rk(i,1)+k(3,1),h_rk(i,2)+k(3,2))
    h_rk(i+1,1) = h_rk(i,1)+(k(1,1)+2*k(2,1)+2*k(3,1)+k(4,1))/6.;
    h_rk(i+1,2) = h_rk(i,2)+(k(1,2)+2*k(2,2)+2*k(3,2)+k(4,2))/6.;
end

//Plot das solucoes de Euler e RK para o reservatorio 1
scf(0);                             //Inicia a janela grafica 0
plot(t,h_e(:,1)',t,h_rk(:,1)');     //Plota os dados
legend(["Euler","Runge-Kutta"]);    //Define a legenda das curvas
//Define o titulo do grafico
xtitle("Solu  o num rica para o n vel do reservat rio 1","Tempo (s)","Altura (m)");

//Plot das solucoes de Euler e RK para o reservatorio 2
scf(1);                             //Inicia a janela grafica 1
plot(t,h_e(:,2)',t,h_rk(:,2)');     //Plota os dados
legend(["Euler","Runge-Kutta"]);    //Define a legenda das curvas
//Define o titulo do grafico
xtitle("Solu  o num rica para o n vel do reservat rio 2","Tempo (s)","Altura (m)");
```