

Ítalo Gonçalves Sant'Ana Paiva 10853310

PME3380 - Modelagem
Lista B

São Paulo

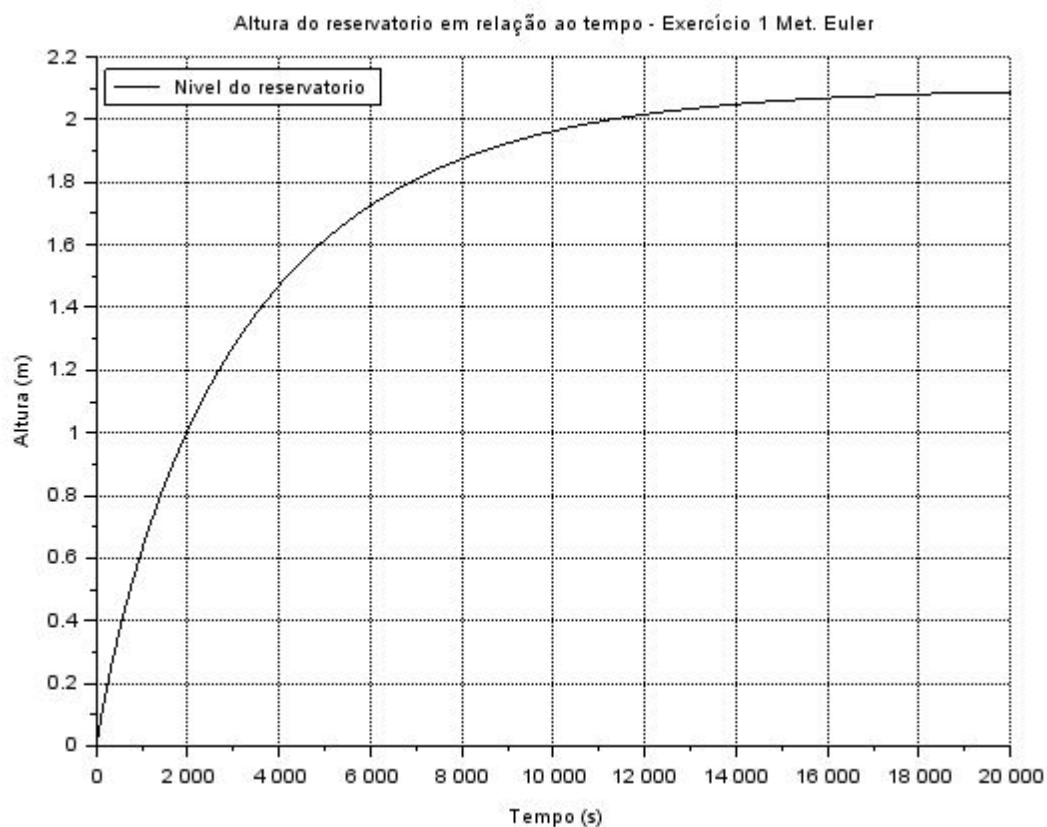
2020

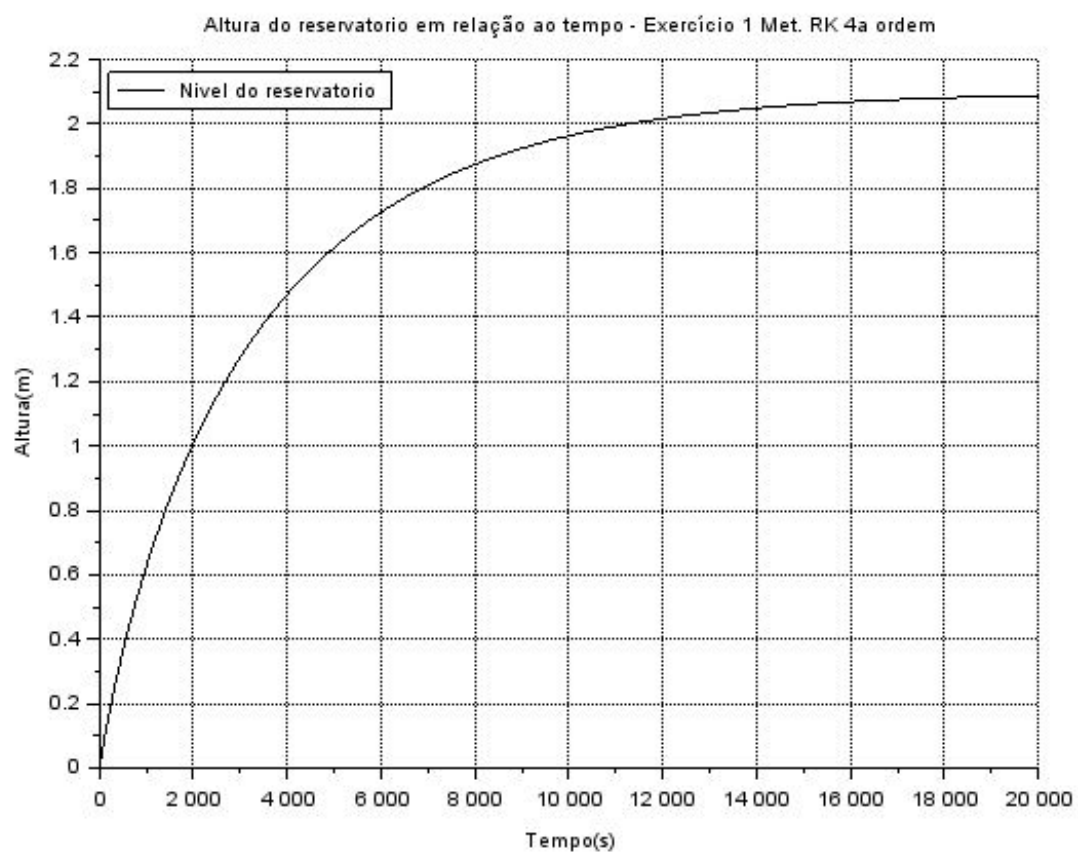
Sumário

1	EXERCÍCIO 1	1
2	EXERCÍCIO 2	3
A	CÓDIGO DA SIMULAÇÃO	5

1 Exercício 1

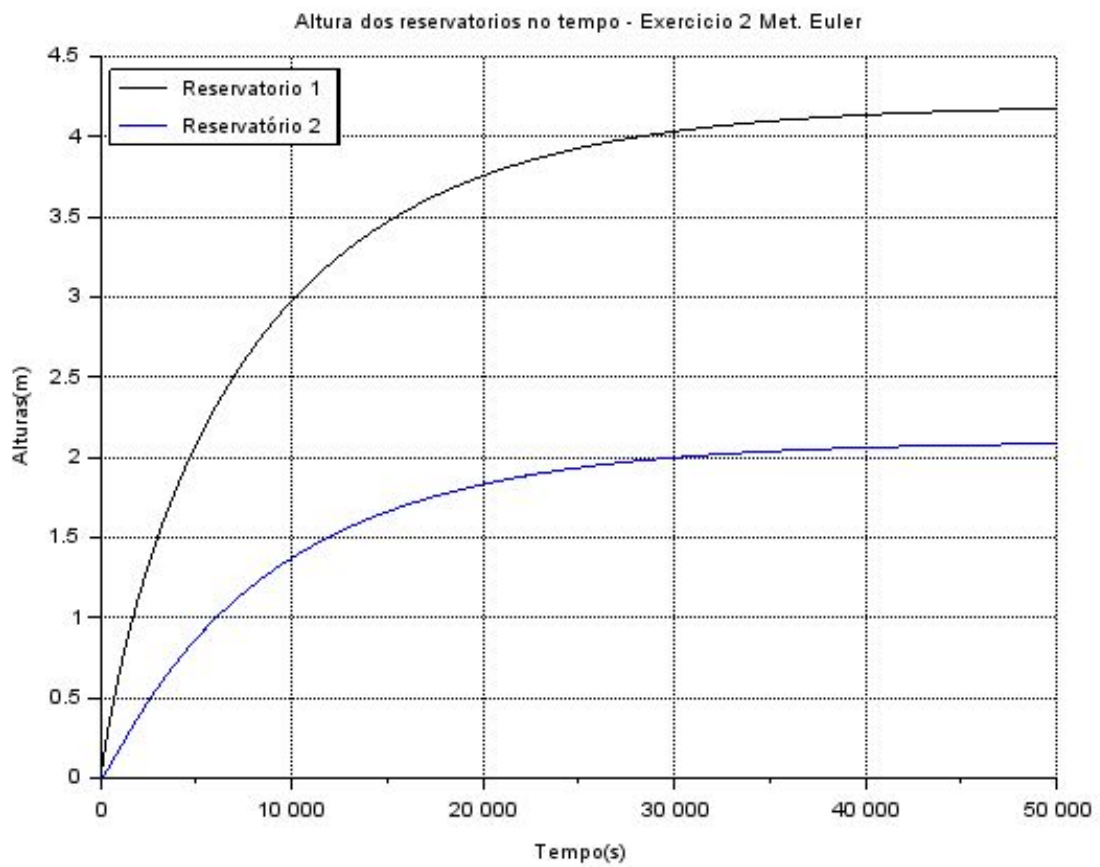
Para o Exercício 1, foi considerado altura inicial nula e um tempo final suficientemente grande para que se possa observar o fenômeno. A seguir, a resolução do problema com 1 Tanque será resolvido pelo método de Euler e pelo método de Runge Kutta de 4ª ordem, respectivamente.

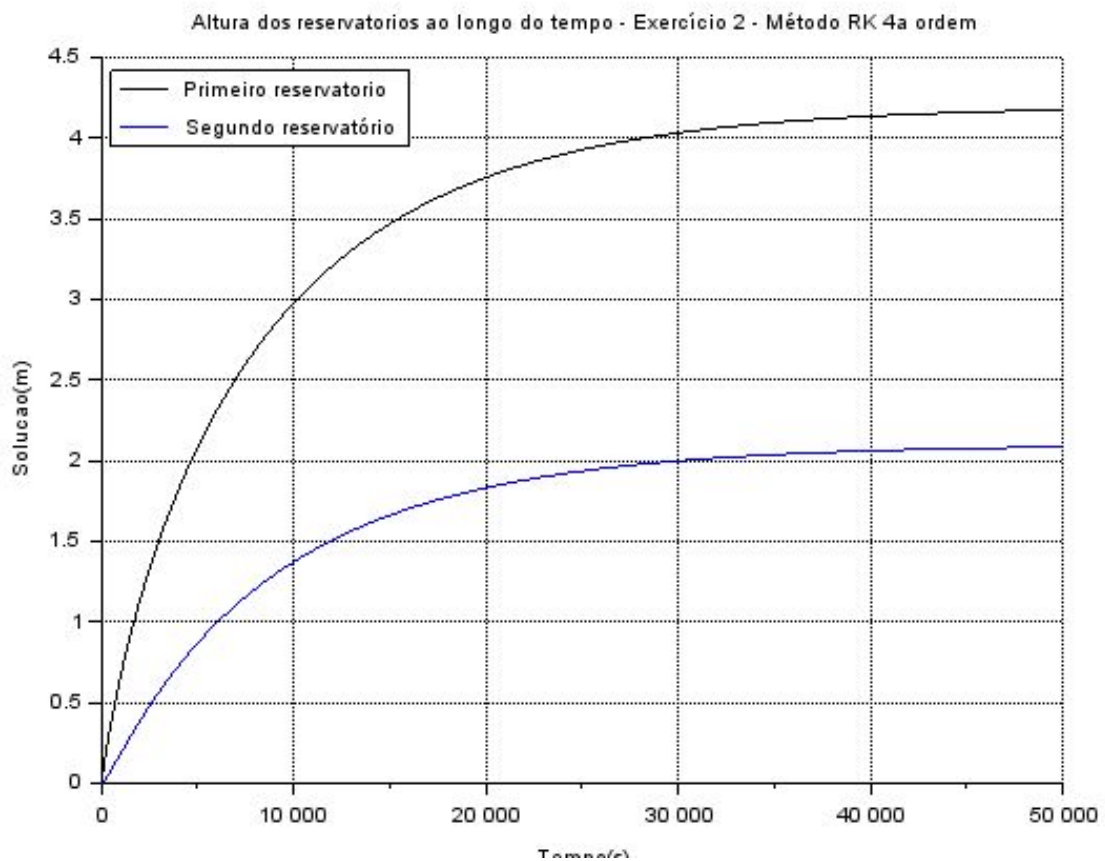




2 Exercício 2

No exercício 2 são considerados dois reservatórios. No segundo, a partir da imagem fornecida do exercício, foi considerado que sua área S_2 é igual a $0,9 * S_1$, assim, adotando alturas iniciais como zero e um tempo final suficientemente grande, foi possível observar o fenômeno.





Já ao que diz respeito à solução exata do problema, não foi possível obter valores para comparação com o método numérico, uma vez que não foi capaz de isolar a incógnita na equação obtida:

$$c_1 - \frac{t}{S} = \frac{2 \left(Q \log \left(Q - \sqrt{C y(t)} \right) + \sqrt{C y(t)} \right)}{C}$$

A Código da simulação

```

1 EXERCICIO 1 - MTODO DE EULER
2 clear
3 //parametros de entrada
4 R = 2*10^8
5 Qe = 0.010247
6 rho = 1000
7 S = 10
8 g = 10
9
10 //funcao utilizada no problema, onde h é o parametro de entrada
11 function [hdot] = funcaoaltura(h)
12 hdot = (-sqrt(rho*g*h/R) + Qe)/S;
13 endfunction
14
15 // Instante inicial:
16 t(1) = 0;
17 // Instante final suficientemente grande para se observar bem o fenomeno:
18 tf = 20000;
19 // Passo de integracao (experimente alterar o passo):
20 dt = 0.1;
21 // Numero de passos:
22 n = round((tf - t(1))/dt);
23
24 // Condicao inicial da altura é zero, parte do tanque vazio
25 h(1) = 0;
26
27 // Integracao numerica usando o metodo de Euler:
28 // Comando for:
29 for i = 1:n
30     // Vetor de tempo:
31     t(i+1) = t(i) + dt;
32     // Solucao numerica:
33     h(i+1) = h(i) + dt*funcaoaltura(h(i));
34 // Termina do comando for:
35 end
36
37 // Abrindo uma nova janela de graficos:
38 set('current_figure',1);
39 // Plotando solucao numerica y versus vetor de tempo t
40 plot2d(t,h,[1 2]);
41 // Usando a variavel do tipo 'lista':
42 T=list("Altura do reservatorio em relação ao tempo – Exercício 1 Met. Euler",
    "Tempo (s)", "Altura (m)", "Nivel do reservatorio");

```

```
43 // Colocando uma legenda na parte superior esquerda da figura (parametro 2)
44 :
45 legends(T(4),[1,2],2);
46 // Colocando um titulo na figura e nomeando os eixos:
47 xtitle(T(1),T(2),T(3));
48 // Colocando uma grade no grafico:
49 xgrid(1)
50
51 EXERCICIO 1 – METODO DE RUNGE KUTTA DE 4a ORDEM
52 clear
53 //parametros de entrada
54 R = 2*10^8
55 Qe = 0.010247
56 rho = 1000
57 S = 10
58 g = 10
59
60 //funcoes
61 function [hdot] = funcaoaltura(h)
62 hdot = (-sqrt(rho*g*h/R) + Qe)/S;
63 endfunction
64
65 // Instante inicial:
66 t(1) = 0;
67 // Instante final:
68 tf = 20000;
69 // Passo de integracao
70 dt = 0.1;
71 // Calculo de numero de passos:
72 n = round((tf - t(1))/dt);
73
74 // Condicao inicial:
75 h(1) = 0;
76
77 // Integracao numerica usando o metodo de Euler:
78 // Comando for:
79 for i = 1:n
80     // Vetor de tempo:
81     t(i + 1) = t(i) + dt;
82     // Solucao numerica:
83     k1 = dt*funcaoaltura(h(i));
84     k2 = dt*funcaoaltura(h(i) + k1/2);
85     k3 = dt*funcaoaltura(h(i) + k2/2);
86     k4 = dt*funcaoaltura(h(i) + k3);
87     h(i+1)=h(i) + ((k1+2*k2+2*k3+k4)/6);
88
```



```
89     // Termino do comando for:
90 end
91
92 // Abrindo uma nova janela de graficos:
93 set("current_figure",1);
94 // Plotando solucao numerica y versus vetor de tempo t
95 plot2d(t,h,[1 2]);
96 // Usando a variavel do tipo 'lista':
97 T=list("Altura do reservatorio em relação ao tempo – Exercício 1 Met. RK 4a
    ordem","Tempo(s)","Altura(m)","Nivel do reservatorio");
98 // Colocando uma legenda na parte superior esquerda da figura (parametro 2)
    :
99 legends(T(4),[1,2],2);
100 // Colocando um titulo na figura e nomeando os eixos:
101 xtitle(T(1),T(2),T(3));
102 // Colocando uma grade no grafico:
103 xgrid(1)
104
105


---


106 EXERCICIO 2 – MTODO DE EULER
107 clear
108 //parametros de entrada
109 rho = 1000
110 g = 10
111 Qe = 0.010247
112 S1 = 10
113 S2 = 0.9*S1 // assumindo que a area de 2o reservatorio é 90% do 1o
114 Ra = 2*10^8
115 Rs = 2*10^8
116
117
118 //funcoes com parametros de entrada h1 e h2
119 function [hdot] = funcaoaltura1(h1, h2)
120 hdot = (-sqrt(rho*g*(h1 - h2)/Ra) + Qe)/S1;
121 endfunction
122
123 function [hdot] = funcaoaltura2(h1, h2)
124 hdot = (sqrt(rho*g*(h1 - h2)/Ra) - sqrt(rho*g*h2/Rs))/S2;
125 endfunction
126
127 // Instante inicial:
128 t(1) = 0;
129 // Instante final:
130 tf = 50000;
131 // Passo de integracao:
132 dt = 0.1;
133 // Cnumero de passos:
```

```

134 n = round((tf - t(1))/dt);
135
136 // Condição inicial, as alturas partem de zero, ou seja, vazios:
137 h1(1) = 0;
138 h2(1) = 0;
139
140 // Integração numérica usando o método de Euler:
141 // Comando for:
142     for i = 1:n
143         // Vetor de tempo:
144         t(i + 1) = t(i) + dt;
145         // Solução numérica:
146         h1(i+1) = h1(i) + dt*funcaoaltura1(h1(i),h2(i));
147         h2(i+1) = h2(i) + dt*funcaoaltura2(h1(i),h2(i));
148     // Termina o comando for:
149     end
150
151 // Abrindo uma nova janela de gráficos:
152 set("current_figure",1);
153 // Plotando solução numérica y versus vetor de tempo t
154 plot2d([t,t], [h1, h2], [1 2]);
155 // Usando a variável do tipo 'lista':
156 T=list("Altura dos reservatórios no tempo – Exercício 2 Met. Euler","Tempo(
        s)","Alturas(m)","Reservatório 1", "Reservatório 2");
157 // Colocando uma legenda na parte superior esquerda da figura (parâmetro 2)
        :
158 legends([T(4), T(5)], [1,2], 2);
159 // Colocando um título na figura e nomeando os eixos:
160 xtitle(T(1), T(2), T(3));
161 // Colocando uma grade no gráfico:
162 xgrid(1)
163
164 EXERCÍCIO 2 – MÉTODO DE RUNGE KUTTA DE 4ª ORDEM
165 clear
166 //parâmetros de entrada
167 rho = 1000
168 g = 10
169 Qe = 0.010247
170 S1 = 10
171 S2 = 0.9*S1 // assumindo que a área de 2º reservatório é 90% do 1º
172 Ra = 2*10^8
173 Rs = 2*10^8
174
175 //funções
176 function [hdot] = funcaoaltura1(h1, h2)
177 hdot = (-sqrt(rho*g*(h1 - h2)/Ra) + Qe)/S1;
178 endfunction

```

```

179
180 function [hdot] = funcaoaltura2(h1, h2)
181 hdot = (sqrt(rho*g*(h1 - h2)/Ra) - sqrt(rho*g*h2/Rs))/S2;
182 endfunction
183
184 // Instante inicial:
185 t(1) = 0;
186 // Instante final:
187 tf = 50000;
188 // Passo de integracao:
189 dt = 0.5;
190 // numero de passos:
191 n = round((tf - t(1))/dt);
192
193 // Condicao inicial:
194 h1(1) = 0;
195 h2(1) = 0;
196
197 // Integracao numerica usando o metodo de Euler:
198 // Comando for:
199 for i = 1:n
200     // Vetor de tempo:
201     t(i + 1) = t(i) + dt;
202     // Solucao numerica:
203
204     k1a = dt*funcaoaltura1(h1(i), h2(i));
205     k1b = dt*funcaoaltura1(h1(i) + k1a/2, h2(i) + k1a/2);
206     k1c = dt*funcaoaltura1(h1(i) + k1b/2, h2(i) + k1b/2);
207     k1d = dt*funcaoaltura1(h1(i) + k1c/2, h2(i) + k1c/2);
208     h1(i + 1) = h1(i) + ((k1a + 2*k1b + 2*k1c + k1d)/6);
209
210     k2a = dt*funcaoaltura2(h1(i), h2(i));
211     k2b = dt*funcaoaltura2(h1(i) + k2a/2, h2(i) + k2a/2);
212     k2c = dt*funcaoaltura2(h1(i) + k2b/2, h2(i) + k2b/2);
213     k2d = dt*funcaoaltura2(h1(i) + k2c/2, h2(i) + k2c/2);
214     h2(i+1) = h2(i) + ((k2a + 2*k2b + 2*k2c + k2d)/6);
215     // Termina do comando for:
216 end
217
218 // Abrindo uma nova janela de graficos:
219 set("current_figure",1);
220 // Plotando solucao numerica y versus vetor de tempo t
221 plot2d([t,t], [h1, h2], [1 2]);
222 // Usando a variavel do tipo 'lista':
223 T=list("Altura dos reservatorios ao longo do tempo - Exercício 2 - Método
      RK 4a ordem", "Tempo(s)", "Solucao(m)", "Primeiro reservatorio", "Segundo
      reservatório");

```

```
224 // Colocando uma legenda na parte superior esquerda da figura (parametro 2)
      :
225 legends([T(4), T(5)], [1,2], 2);
226 // Colocando um titulo na figura e nomeando os eixos:
227 xtitle(T(1), T(2), T(3));
228 // Colocando uma grade no grafico:
229 xgrid(1)
```