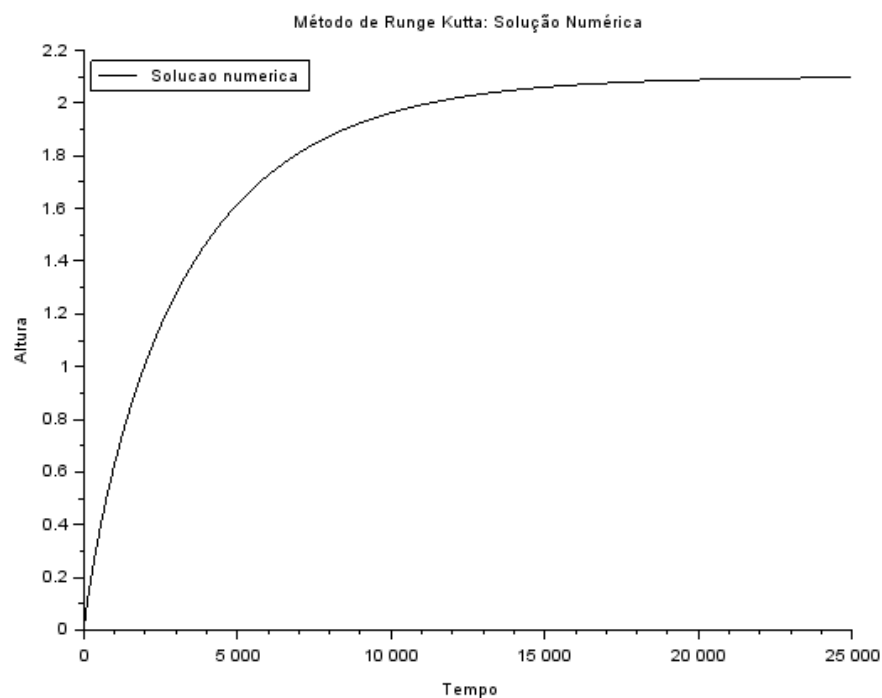
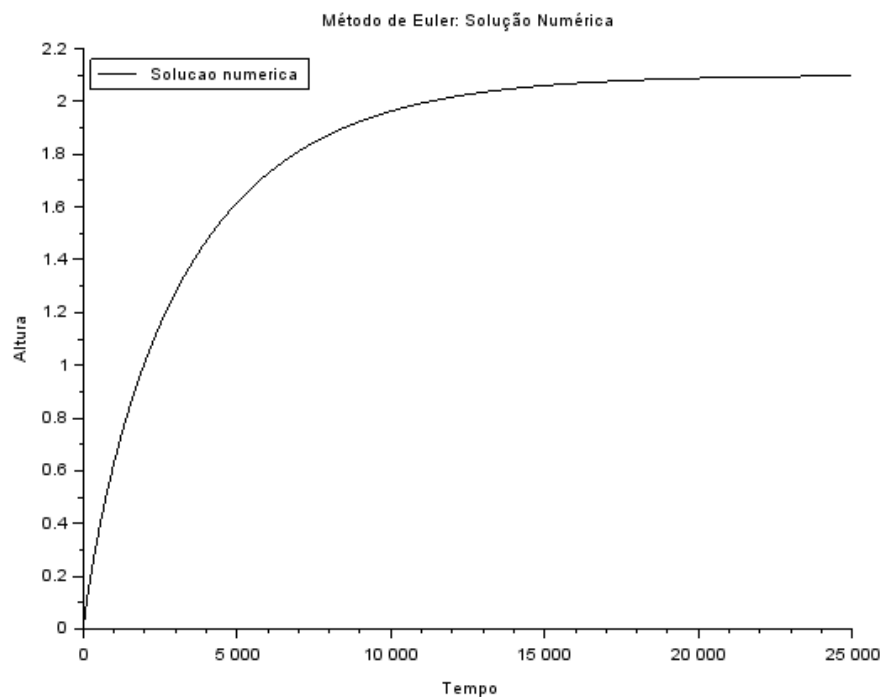


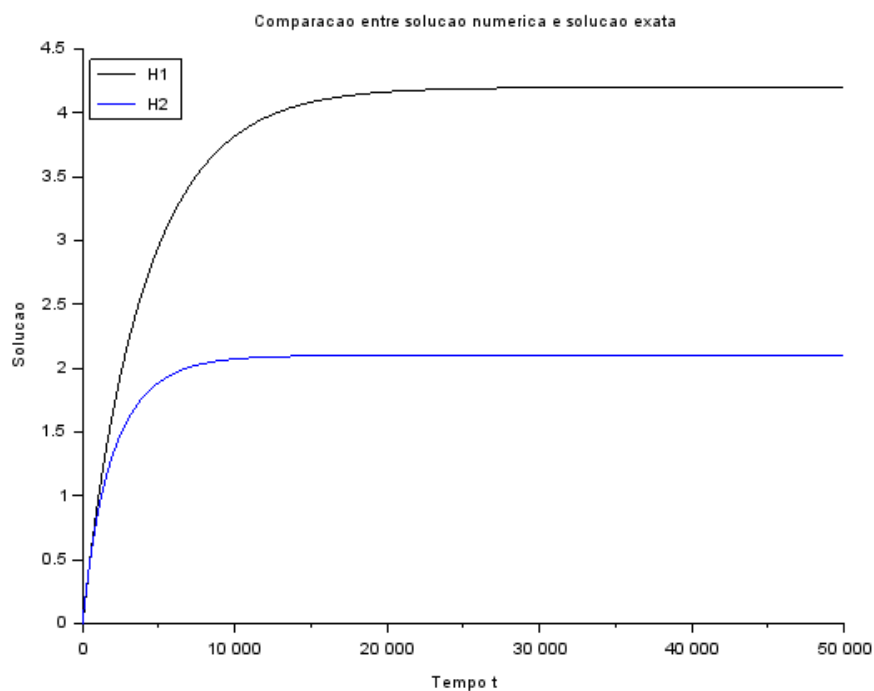
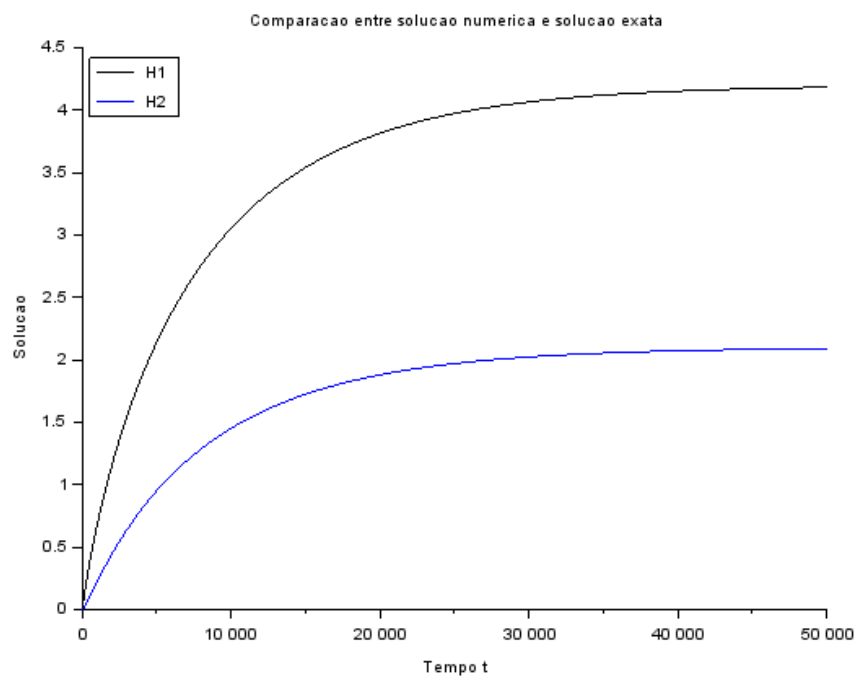
Exercício 1

Nesse problema, foi adotado que o reservatório com água começa completamente vazio e vai enchendo com o passar do tempo, até que o sistema fique estável com a altura tornando-se praticamente constante ao final do processo. Estão representados os gráficos abaixo com a modelagem baseada nos métodos de integração de Euler e de Runge-Kutta, respectivamente:



Exercício 2

Nesse segundo exercício foram considerados dois tanques, onde o primeiro tinha área de seção transversal 10 e o segundo 6 (SI), com os mesmos raios. Seguem abaixo os gráficos utilizando-se os métodos de Euler e Runge-Kutta, respectivamente:



Código

Segue abaixo o código utilizado para a modelagem dos exercícios:

Exercício 1 – Euler:

```
1 //Apagando dados anteriores:
2 clear
3 //EULER-SOLUÇÃO-NUMÉRICA
4 //Parametros
5 S=-10
6 R=-2*10^8
7 p=-1000
8 g=-10
9 Qe=-0.010247
10
11 function hpto = funcao(h)
12 ... hpto = (Qe - sqrt((p*g*h)/R))/S
13 endfunction
14
15 //Instante inicial:
16 t(1)=0;
17 //Instante final:
18 tf=25000;
19 //Condicao inicial:
20 y(1)=0;
21
22 //Passo de integracao (experimente alterar o passo):
23 h=0.5;
24 //Calculo de numero de passos:
25 n=round(tf/h);
26 //Integracao numerica usando o metodo de Euler:
27 //Comando for:
28 for i=1:n
29 //Vetor de tempo:
30 t(i+1)=t(i)+h;
31 //Solucao numerica:
32 y(i+1)=y(i)+h*funcao(y(i));
33 //Termino do comando for:
34 end
35
36 //Desenhando outro grafico com linhas diferentes:
37 plot2d([t],[y],[1:2]);
38 //Usando a variavel do tipo 'lista':
39 T=list("Método de Euler: Solução Numérica","Tempo-","Altura","Solucao numerica","Solucao exata");
40 //Colocando uma legenda na parte superior esquerda da figura (parametro 2):
41 legends(T(4),[1,2],2);
42 //Colocando um titulo na figura e nomeando os eixos:
43 xtitle(T(1),T(2),T(3));
```

Exercício 1 – Runge-Kutta:

```
1 clear();
2 clc;
3
4 //Parametros
5 S = -10
6 R = -2*10^8
7 p = -1000
8 g = -10
9 Qe = -0.010247
10
11 function hpto = funcao(y)
12 ... hpto = (Qe - sqrt((p*g*h)/R))/S
13 endfunction
14
15
16 //Conjunto de comandos para solucao numerica de equacao diferencial [1-y(i)]/2
17 //Apagando dados anteriores:
18 //Instante inicial:
19 t(1)=0;
20 //Instante final:
21 tf=25000;
22 //Condicao inicial:
23 y(1)=0;
24 //Valor inicial da solucao exata:
25 ye(1)=0;
26 //Passo de integracao (experimente alterar o passo):
27 h=0.5;
28 //Calculo de numero de passos:
29 n=round((tf-t(1))/h);
30 //Integracao numerica usando o metodo de Runge-Kutta:
31 //Comando for:
32 for i=1:n
33 //Vetor de tempo:
34 t(i+1)=t(i)+h;
35
36 //Solucao numerica:
37 k1=h*((-sqrt(p*g*y(i)/R) + Qe)/S);
38 k2=h*((-sqrt(p*g*(y(i) + k1/2)/R) + Qe)/S);
39 k3=h*((-sqrt(p*g*(y(i) + k2/2)/R) + Qe)/S);
40 k4=h*((-sqrt(p*g*(y(i) + k3/2)/R) + Qe)/S);
41 y(i+1)=y(i)+((k1+2*k2+2*k3+k4)/6);
42 //Termino do comando for:
43 end
44
45 //Desenhando outro grafico com linhas diferentes:
46 plot2d([t],[y],[1:2]);
47 //Usando a variavel do tipo 'lista':
48 T=list("Método de Runge-Kutta: Solução Numérica","Tempo","Altura","Solucao numerica","Solucao exata");
49 //Colocando uma legenda na parte superior esquerda da figura (parametro 2):
50 legends(T(4),[1,2],2);
51 //Colocando um titulo na figura e nomeando os eixos:
52 xtitle(T(1),T(2),T(3));
```

Exercício 2 – Euler:

```
1 //Apagando dados anteriores:
2 clear
3 //EX2 -- EULER SOLUÇÃO NUMÉRICA
4 //Parametros
5 S1 = 10;
6 S2 = 6;
7 R1 = 2*10^8;
8 R2 = 2*10^8;
9 p = 1000;
10 g = 10;
11 Qe = 0.010247;
12
13 //Definindo as funções
14 function hpto = funcao1(h1,h2)
15 ... hpto = (Qe - sqrt(p*g*(h1-h2)/R1))/S1
16 endfunction
17
18 function hpto = funcao2(h1,h2)
19 ... hpto = (sqrt(p*g*(h1-h2)/R1) - sqrt(p*g*h2/R2))/S2
20 endfunction
21
22 //Instante inicial:
23 t(1)=0;
24 //Instante final:
25 tf=50000;
26 //Condicao inicial:
27 h1(1)=0;
28 h2(1)=0;
29 ye(1)=0
30 //Passo de integracao (experimente alterar o passo):
31 h=0.5;
32 //Calculo de numero de passos:
33 n=round(tf/h);
34 //Integracao numerica usando o metodo de Euler:
35 //Comando for:
36 for i=1:n
37 //Vetor de tempo:
38 t(i+1)=t(i)+h;
39 h1(i+1) = h1(i) + h*(funcao1(h1(i),h2(i)));
40 h2(i+1) = h2(i) + h*(funcao2(h1(i),h2(i)));
41
42 //Termino do comando for:
43 end
44
45 //Desenhando outro grafico com linhas diferentes:
46 plot2d([t,t],[h1,h2],[1,2]);
47 //Usando a variavel do tipo 'lista':
48 T=list("Comparacao entre solucao numerica e solucao exata","Tempo-t","Solucao","H1","H2");
49 //Diminuindo a espessura das linhas:
50 xset("thickness",1)
51 //Colocando uma legenda na parte superior esquerda da figura (parametro 2):
52 legends([T(4),T(5)],[1,2],2);
53 //Colocando um titulo na figura e nomeando os eixos:
54 xtitle(T(1),T(2),T(3));
55
```

Exercício 2 – Runge-Kutta:

```
1 //Apagando dados anteriores:
2 clear
3 //EX2 -- RUNGE - SOLUÇÃO NUMÉRICA
4 //Parâmetros
5 R1 = 2*10^8;
6 R2 = 2*10^8;
7 Qe = 0.010247;
8 p = 1000;
9 S1 = 10;
10 S2 = 6;
11 g = 10;
12
13 //Definindo as funções
14 function [hpto] = funcao1(h1,h2)
15 ... hpto = (-sqrt(p*g*(h1-h2)/R1) + Qe)/S1 ;
16 endfunction
17 function [h2pto]=funcao2(h1,h2)
18 ... h2pto = (-sqrt(p*g*(h2-h1)/R1) -sqrt(-p*g*h2/R2) + Qe)/S2 ;
19 endfunction
20
21 //Instante inicial:
22 t(1)=0;
23 //Instante final:
24 tf=50000;
25 //Condicao inicial:
26 h1(1)=0;
27 h2(1)=0;
28
29 //Passo de integracao (experimente alterar o passo):
30 h=0.5;
31 //Calculo de numero de passos:
32 n=round((tf-t(1))/h);
33 //Integracao numerica usando o metodo de Runge-Kutta:
34 //Comando for:
35 for i=1:n
36 ... //Vetor de tempo:
37 ... t(i+1)=t(i)+h;
38 ... //Solucao numerica:
39 ... k1=h*funcao2(h1(i),h2(i));
40 ... k2=h*funcao2(h1(i)+k1/2,h2(i)+k1/2); ...
41 ... k3=h*funcao2(h1(i)+k2/2,h2(i)+k2/2);
42 ... k4=h*funcao2(h1(i)+k3,h2(i)+k3);
43 ... h2(i+1)=h2(i)+((k1+2*k2+2*k3+k4)/6);
44 ...
45 ... a1=h*funcao1(h1(i),h2(i));
46 ... a2=h*funcao1(h1(i)+a1/2,h2(i)+a1/2);
47 ... a3=h*funcao1(h1(i)+a2/2,h2(i)+a2/2);
48 ... a4=h*funcao1(h1(i)+a3,h2(i)+a3);
49 ... h1(i+1)=h1(i)+((a1+2*a2+2*a3+a4)/6);
50 ...
51 //Termino do comando for:
52 end
53
54 //Desenhando outro grafico com linhas diferentes:
55 plot2d([t,t],[h1,h2],[1,2]);
56 //Usando a variavel do tipo 'lista':
57 T=list("Comparacao entre solucao numerica e solucao exata","Tempo-t","Solucao","H1","H2");
58 //Diminuindo a espessura das linhas:
59 xset("thickness",1)
60 //Colocando uma legenda na parte superior esquerda da figura (parametro 2):
61 legenda([T(4),T(5)],[1,2],2);
62 //Colocando um titulo na figura e nomeando os eixos:
63 xtitle(T(1),T(2),T(3));
64
65
```