Escola Politécnica da Universidade de São Paulo

PME 3380 – Modelagem de Sistemas Dinâmicos

Lista B

Gabriela Vasconcelos Araujo - 10771497

Turma 1
Prof. Agenor de Toledo Fleury
São Paulo, 2020

SUMÁRIO

Equações Diferenciais

Exemplo 1 – Método de Euler

funcao.sci

```
function [ydot]=funcao(y)
   ydot = (1 - y)/2;
endfunction
save ('funcao.sci','funcao');
```

numericoE.sce

```
// Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
// Apagando dados anteriores:
clear
// Carregando a equacao diferencial:
// Carregue a função usando o comando Load do Scilab
load('funcao.sci','funcao');
// Instante inicial:
t(1) = 0;
// Instante final:
tf = 10;
// Condicao inicial:
y(1) = 0;
// Valor inicial da solucao exata:
ye(1) = 0;
// Passo de integracao (experimente alterar o passo):
h = 0.5;
// Calculo de numero de passos):
n = round(tf/h);
// Integracao numerica usando o metodo de Euler:
// Comando for:
for i = 1:n
	// Vetor de tempo:
	t(i+1) = t(i) + h;
	// Solucao numerica:
	y(i+1) = y(i) + h * funcao(y(i));
	// Solucao exata:
	ye(i+1) = 1-%e^(-t(i+1)/2);
// Termino do comando for:
end
// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
plot2d([t,t],[y,ye],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solucao numerica","Solucao exata"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao")
// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T = list("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao","Solucao
numerica","Solucao exata");
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends([T(4),T(5)],[1,2],2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
```

SAÍDAS

Figura 1: Comparação entre solução numérica por Euler e solução exata - Dispersão

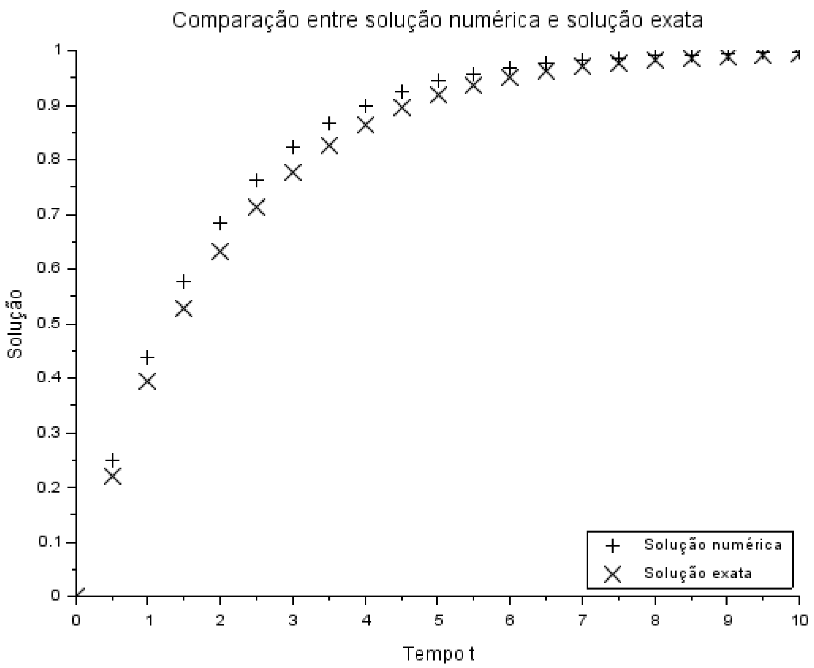
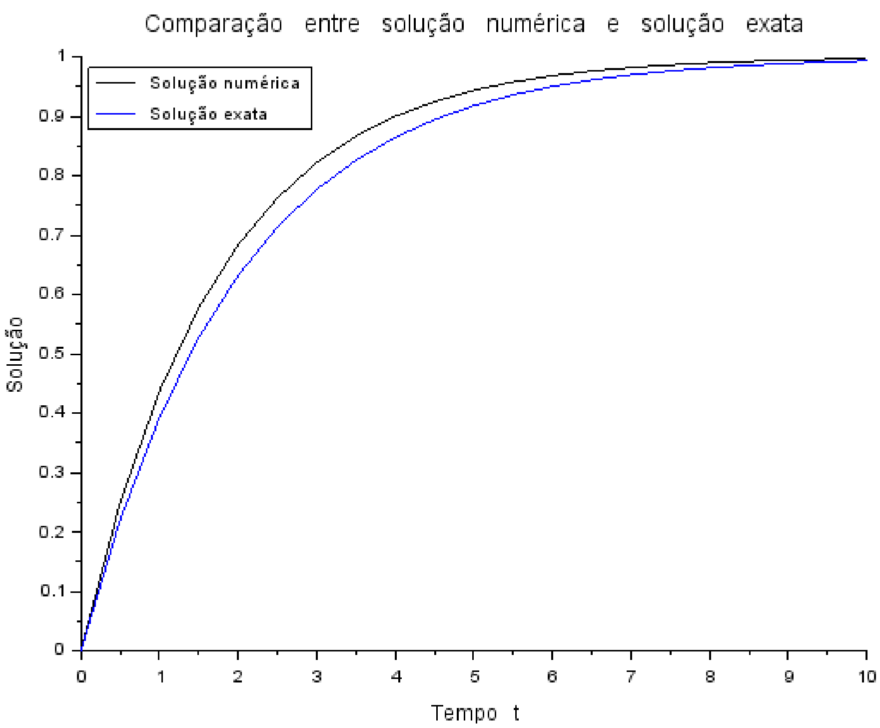


Figura 2: Comparação entre solução numérica por Euler e solução exata – Linha contínua



Exemplo 2 – Método de Runge Kutta de 4° Ordem

numericoR.sce

```
// Conjunto de comandos para solucao numerica de equacao diferencial [1-y(i)]/2

// Apagando dados anteriores:
clear
// Instante inicial:
t(1) = 0;
// Instante final:
tf = 10;
// Condicao inicial:
y(1) = 0;
// Valor inicial da solucao exata:
ye(1) = 0;
// Passo de integracao (experimente alterar o passo):
h = 0.5;
// Calculo de numero de passos):
n = round((tf-t(1))/h);
// Integracao numerica usando o metodo de Runge Kutta:

// Comando for:
for i = 1:n
    //Vetor de tempo:
    t(i+1) = t(i)+h;
    // Solucao numerica:
    k1 = h * (1-(y(i)))/2;
    k2 = h * (1-(y(i)+k1/2))/2;
    k3 = h * (1-(y(i)+k2/2))/2;
    k4 = h * (1-(y(i)+k3))/2;
    y(i+1) = y(i) + ((k1+2*k2+2*k3+k4)/6);
    // Solucao exata:
    ye(i+1) = 1 - %e^(-t(i+1)/2);
    // Termino do comando for:
end

// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
plot2d([t,t],[y,ye],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solução numérica","Solução exata"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparação entre solução numérica e solução exata","Tempo t","Solução")
// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Aumentando a espessura das linhas:
xset('thickness',2)
// Aumentando o tamanho da fonte:
xset('font size',4)
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T = list("Comparação entre solução numérica e solução exata","Tempo t","Solução","Solução numérica","Solução exata");
// Diminuindo a espessura das linhas:
xset('thickness',1)
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends([T(4),T(5)],[1,2],2);
// Colocando um titulo na figura e nomeando oseixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)
```

SAÍDAS

Figura 3: Comparação entre solução numérica por Runge Kutta e solução exata - Dispersão

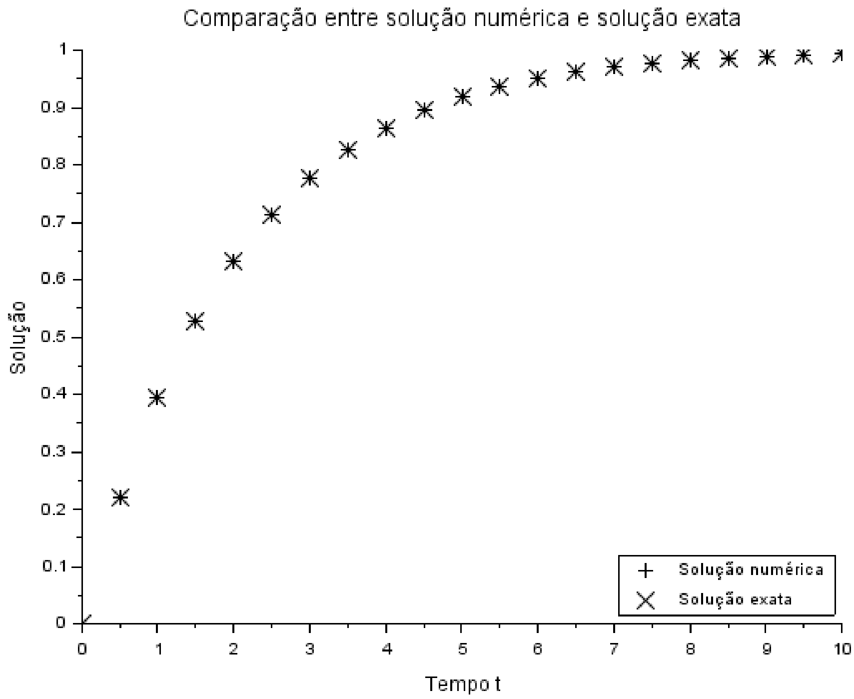
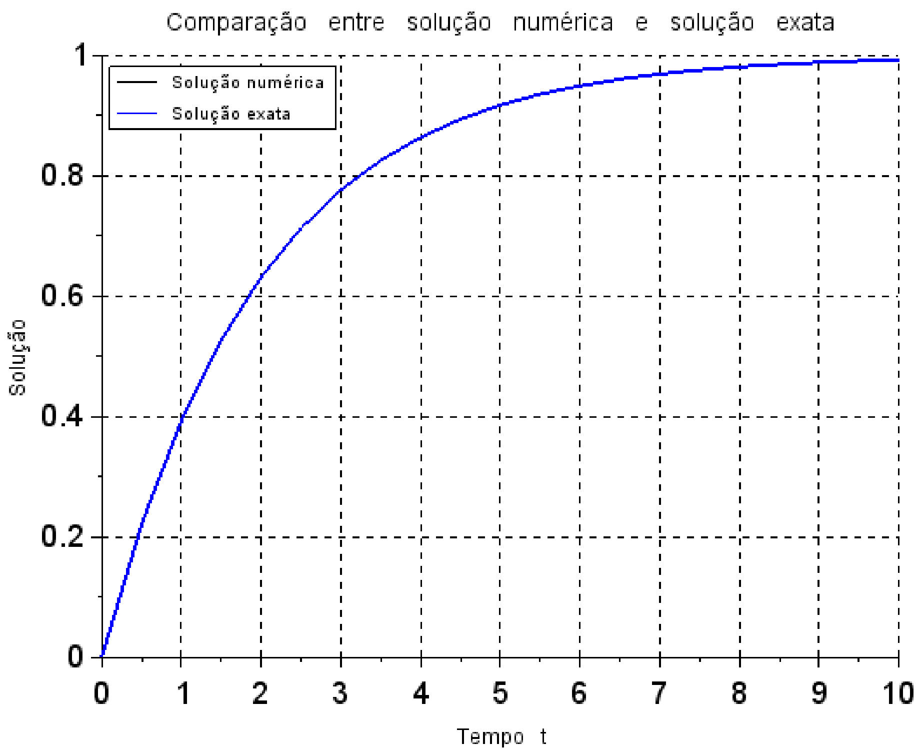


Figura 4: Comparação entre solução numérica por Runge Kutta e solução exata – Linha contínua



Exercício 1 – Reservatório Único

Para este exercício, ambos os métodos de integração serão aplicados em um único código, no arquivo nomeado como 1numerico.sce. Para modelar o problema, utilizou-se a seguinte equação diferencial:

$$\dot{h} = \frac{1}{S}\left(Q_c - \sqrt{\rho g h / R}\right)$$

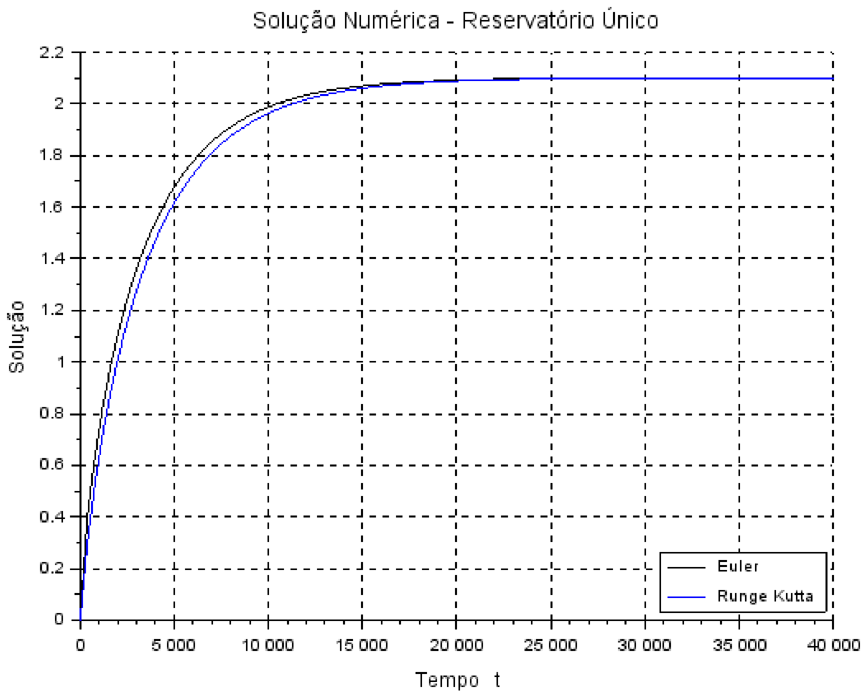
1numerico.sce

```
clear

function [hdot]=funcao(h, S, R, ro, G, Qe)
   hdot = (1/S) * (- sqrt(ro*G*h/R) + Qe);
endfunction

S = 10; // área da seção transversal
R = 2*(10^8); // relação de vazão com perda de carga
ro = 1000; // massa específica da água
G = 10; // aceleração da gravidade
Qe = 0.010247; // vazão de entrada

t(1) = 0; // instante inicial
tf = 40000; // instante final
HRK(1) = 0; // condição inicial - Runge Kutta
HE(1)= 0; // condição inicial - Euler
h = 400; // passo de integração
n = round((tf-t(1))/h); // cálculo do número de passos

// Integracao numérica usando o metodo de Runge Kutta:
for i = 1:n
      t(i+1) = t(i) + h; // vetor de tempo
      k1 = funcao(HRK(i),S,R,ro,G,Qe);
      k2 = funcao(HRK(i)+h*0.5*k1,S,R,ro,G,Qe);
      k3 = funcao(HRK(i)+h*0.5*k2,S,R,ro,G,Qe);
      k4 = funcao(HRK(i)+h*k3,S,R,ro,G,Qe);
      HRK(i+1) = HRK(i)+((k1+2*k2+2*k3+k4)*(h/6));
end

// Integracao numérica usando o método de Euler:
for i = 1:n
      t(i+1) = t(i) + h; // vetor de tempo
      HE(i+1) = HE(i) + h * funcao(HE(i),S,R,ro,G,Qe); // solução numérica
end

// Plotagem do gráfico:
plot2d([t,t],[HE,HRK],[1 2]);
T=list("Solução Numérica ","Tempo   t","Solução");
legends(["Euler","Runge Kutta"],[1,2],4)
xset('thickness',1); // diminuindo espessura das linhas;
xtitle(T(1),T(2),T(3)); // colocando título e nomeando os eixos
xgrid(1) // adicionando grade ao gráfico
```

SAÍDAS

Considerando a Figura 5 abaixo exposta, nota-se um caráter próximo do logarítmico das curvas. Além disso, percebe-se que as respostas obtidas pelos diferentes métodos apresentam grande similaridade.

Figura 5: Solução Numérica - Reservatório Único



EXERCÍCIO 2 – SISTEMA COM DOIS RESERVATÓRIOS

Para a análise desse sistema serão desenvolvidos dois programas distintos, o primeiro utilizando o método de Euler e o segundo utilizando Runge Kutta de 4° ordem. Para modelar o problema, utilizou-se o seguinte sistema de equações diferenciais:

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

Vale também ressaltar que, na simulação, os valores de área da seção transversal e de raio do reservatório 2 foram considerados menores quando comparados aos do reservatório 1.

2NUMERICOE.SCE

```
clear

function [h1dot]=freservatorio1(h1, h2, S1, Ra, ro, G, Qe)
   h1dot = (1/S1) * (- sqrt(ro*G*(h1-h2)/Ra) + Qe);
endfunction

function [h2dot]=freservatorio2(h1, h2, S2, Ra, Rs, ro, G, Qe)
   h2dot = (1/S2) * (sqrt(ro*G*(h1-h2)/Ra) - sqrt(ro*G*h2/Rs))
endfunction

S1 = 10; // área da seção transversal do reservatório 1
S2 = 7; //área da seção transversal do reservatório 2
Ra = 2*(10^8); // relação de vazão com perda de carga
Rs = 1.5*(10^8); // relação de vazão com perda de carga
ro = 1000; // massa específica da água
G = 10; // aceleração da gravidade
Qe = 0.010247; // vazão de entrada

t(1) = 0; // instante inicial
tf = 40000; // instante final
H1(1) = 0; // condição inicial reservatório 1
H2(1) = 0; // condição inicial reservatório 2
h = 100; // passo de integração
n = round((tf-t(1))/h); // cálculo do número de passos

// Integração numérica usando o método de Euler:
for i = 1:n
      t(i+1) = t(i) + h; // vetor de tempo
      H1(i+1) = H1(i) + h * freservatorio1(H1(i),H2(i),S1,Ra,ro,G,Qe); // solução numérica do reservatório 1
      H2(i+1) = H2(i) + h * freservatorio2(H1(i),H2(i),S2,Ra,Rs,ro,G,Qe); // solução numérica do reservatório 2
end

// Plotagem dos gráficos:
plot2d([t,t],[H1,H2],[1 2]);
T = list("Solução Numérica - Método de Euler ","Tempo   t","Solução");
legends(["Reservatório 1","Reservatório 2"],[1,2],4)
xset('thickness',1); // diminuindo espessura das linhas;
xtitle(T(1),T(2),T(3)); // colocando título e nomeando os eixos
xgrid(1) // adicionando grade ao gráfico
```

2NUMERICOR.SCE

```
clear

function [h1dot]=freservatorio1(h1, h2, S1, Ra, ro, G, Qe)
   h1dot = (1/S1) * (- sqrt(ro*G*(h1-h2)/Ra) + Qe);
endfunction

function [h2dot]=freservatorio2(h1, h2, S2, Ra, Rs, ro, G, Qe)
   h2dot = (1/S2) * (sqrt(ro*G*(h1-h2)/Ra) - sqrt(ro*G*h2/Rs))
endfunction

S1 = 10; // área da seção transversal do reservatório 1
S2 = 7; //área da seção transversal do reservatório 2
Ra = 2*(10^8); // relação de vazão com perda de carga
Rs = 1.5*(10^8); // relação de vazão com perda de carga
ro = 1000; // massa específica da água
G = 10; // aceleração da gravidade
```

```
Qe = 0.010247; // vazão de entrada

t(1) = 0; // instante inicial
tf = 40000; // instante final
H1(1) = 0; // condição inicial reservatório 1
H2(1) = 0; // condição inicial reservatório 2
h = 100; // passo de integração
n = round((tf-t(1))/h); // cálculo do número de passos

// Integração numérica usando o método de Euler:
for i = 1:n
    t(i+1) = t(i) + h; // vetor de tempo
    k1 = freservatorio1(H1(i),H2(i),S1,Ra,ro,G,Qe);
    k2 = freservatorio1(H1(i)+h*0.5*k1,H2(i)+h*0.5*k1,S1,Ra,ro,G,Qe);
    k3 = freservatorio1(H1(i)+h*0.5*k2,H2(i)+h*0.5*k2,S1,Ra,ro,G,Qe);
    k4 = freservatorio1(H1(i)+h*k3,H2(i)+h*k3,S1,Ra,ro,G,Qe);
    H1(i+1) = H1(i)+((k1+2*k2+2*k3+k4)*(h/6));
    k1 = freservatorio2(H1(i),H2(i),S2,Ra,Rs,ro,G,Qe);
    k2 = freservatorio2(H1(i)+h*0.5*k1,H2(i)+h*0.5*k1,S2,Ra,Rs,ro,G,Qe);
    k3 = freservatorio2(H1(i)+h*0.5*k2,H2(i)+h*0.5*k2,S2,Ra,Rs,ro,G,Qe);
    k4 = freservatorio2(H1(i)+h*k3,H2(i)+h*k3,S2,Ra,Rs,ro,G,Qe);
    H2(i+1) = H2(i)+((k1+2*k2+2*k3+k4)*(h/6));
end

// Plotagem dos gráficos:
plot2d([t,t],[H1,H2],[1 2]);
T = list("Solução Numérica - Método de Runge Kutta ","Tempo   t","Solução");
legends(["Reservatório 1","Reservatório 2"],[1,2],4)
xset('thickness',1); // diminuindo espessura das linhas;
xtitle(T(1),T(2),T(3)); // colocando título e nomeando os eixos
xgrid(1) // adicionando grade ao gráfico
```

Saídas

Figure 6: Solução Numérica - Método de Euler

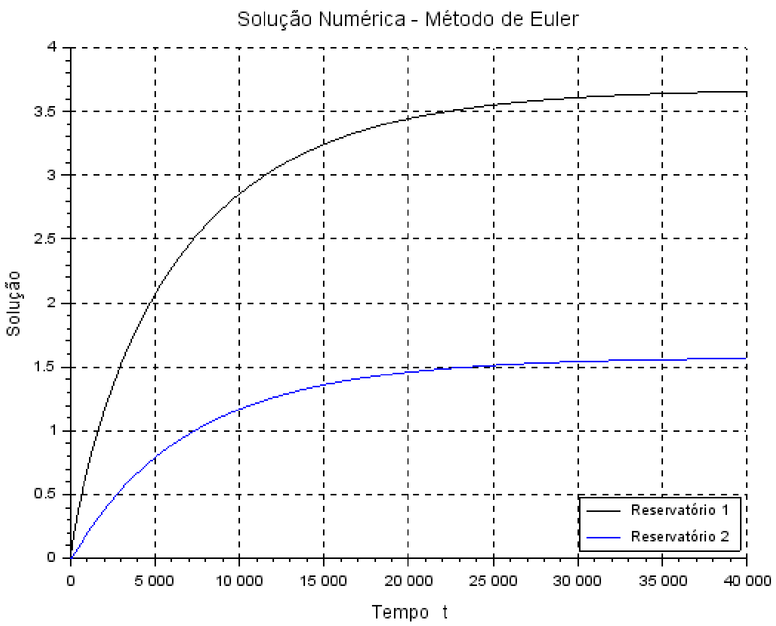
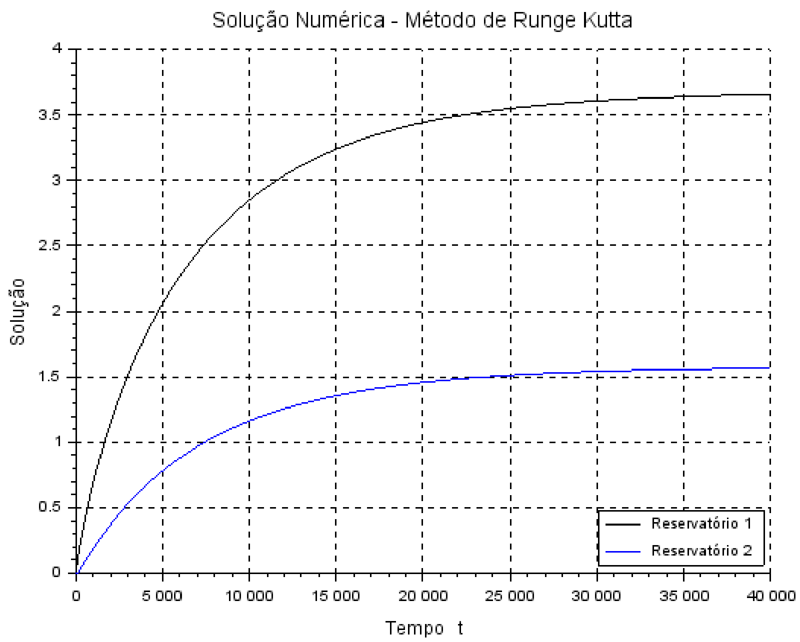


Figure 7: Solução Numérica - Método de Runge Kutta



Como esperado, a altura do reservatório 1 permanece superior à do reservatório 2 ao longo do tempo. Além disso, nota-se também a similaridade dos resultados obtidos por ambos os métodos numéricos.