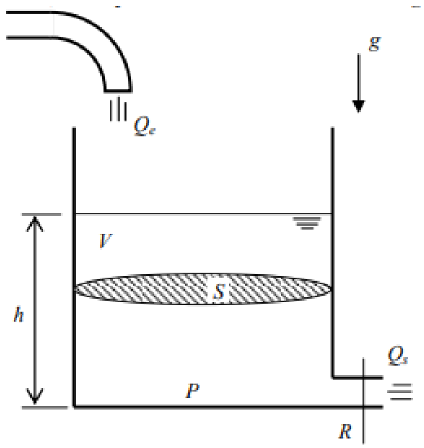
PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

1. Reservatório único

Para o primeiro exercício considerou-se o reservatório esquematizado a seguir:



Reservatório com água

Parâmetros:

S = 10 m^2 - área da seção transversal (constante)

$R = 2\times10^8$ Pa/(m^3/s)^2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)

ρ = 1000 kg/m^3 - massa específica da água

G = 10 m/s^2 - aceleração da gravidade na superfície da terra

Variáveis:

Q_e = 0,010247 m^3/s - vazão de entrada

h: nível do reservatório [m]

V: volume de água no reservatório [m^3]

P: pressão relativa à atmosférica, no fundo do reservatório [Pa]

Q_s: vazão de saída [m^3/s]

Admite-se que a água seja incompressível.

Além disso, assim como demonstrado no enunciado, considerou-se:

$$Q_s = \sqrt{\frac{P}{R}} \quad Vazão\ de\ saída$$

$$P = \rho g h \quad Pressão\ no\ ponto\ mais\ baixo$$

$$\dot{h} = \frac{\left(-\sqrt{\frac{\rho g h}{R}} + Q_e\right)}{S} \quad derivada\ de\ h$$

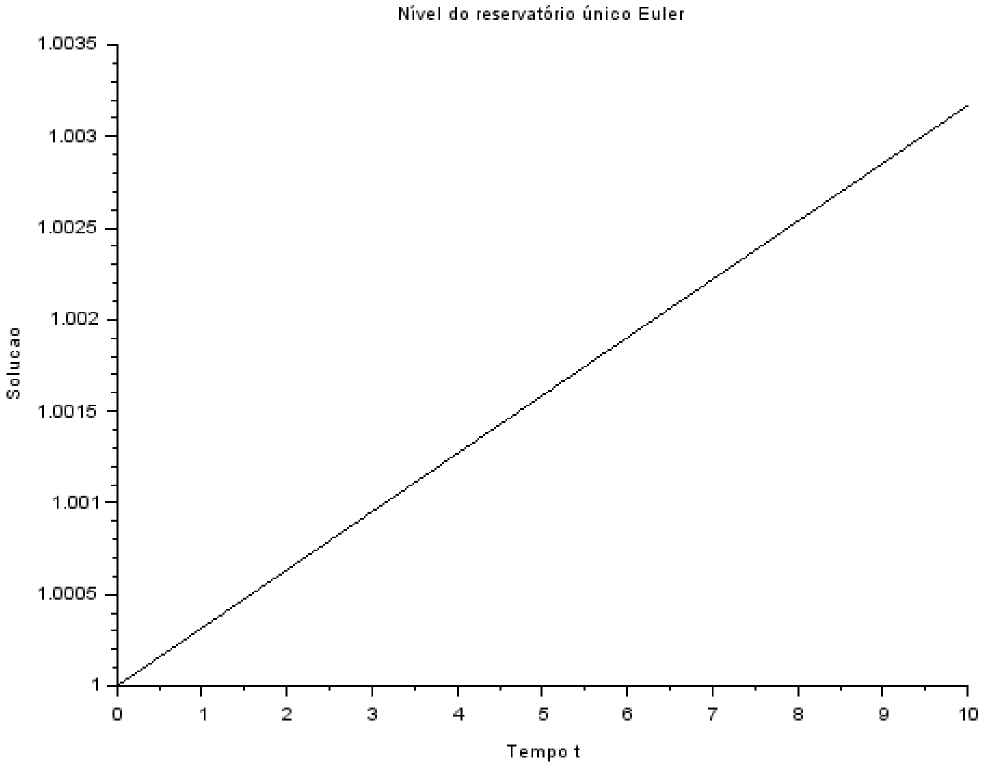
Assim, resolveu-se o problema através de métodos numéricos: Euler e Runge Kutta

1.1. Método de Euler

Conhecendo a derivada e o valor inicial de h, podemos calcular o método de Euler da seguinte maneira:

$$h(i+1) = h(i) + dt * \dot{h}(h(i))$$

O resultado obtido para $h_0 = 1$ foi o seguinte:



1.2. Método de Runge-Kutta 4ª ordem

Assim como o método de Euler, por meio da derivada de h, pode-se implementar uma solução numérica pelo método de runge-kutta da seguinte maneira:

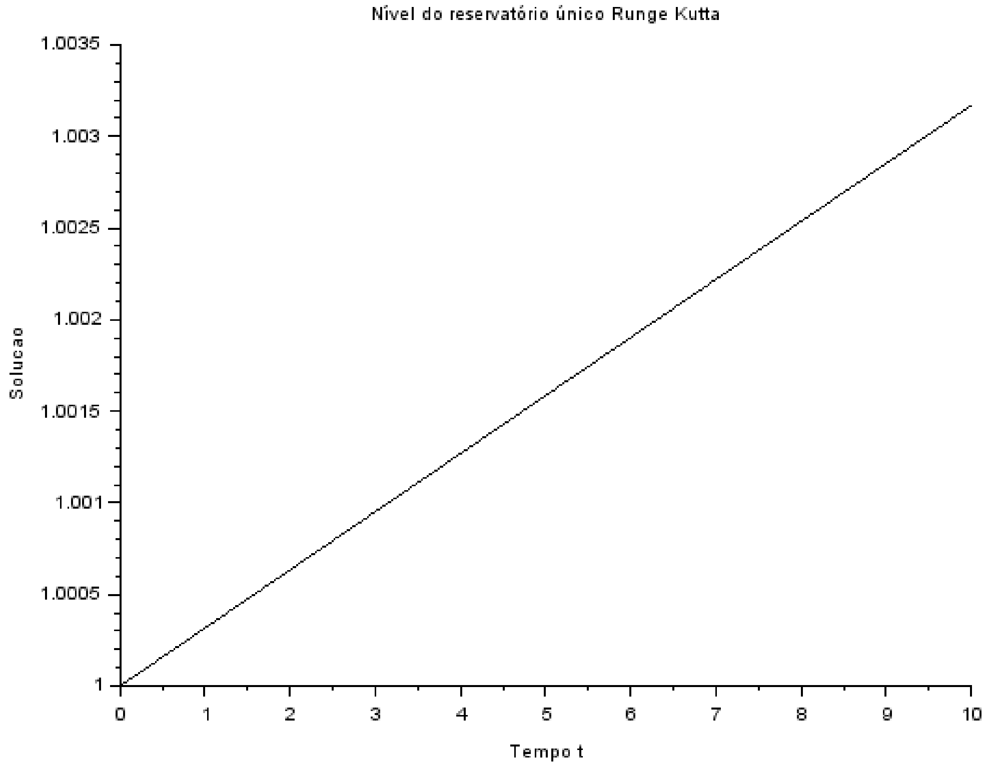
$$K_1(i) = \dot{h}(h(i))$$
$$K_2(i) = \dot{h}(h(i) + \frac{dt}{2} * K_1)$$
$$K_3(i) = \dot{h}(h(i) + \frac{dt}{2} * K_2)$$
$$K_4(i) = \dot{h}(h(i) + dt * K_3)$$
$$h(i+1) = h(i) + \frac{dt}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

O resultado, para $h_0 = 1$,encontrado foi o seguinte:



Percebe-se que o resultado é muito semelhante ao encontrado no método de Euler. Dessa forma, valida-se os programas utilizados e seus resultados.

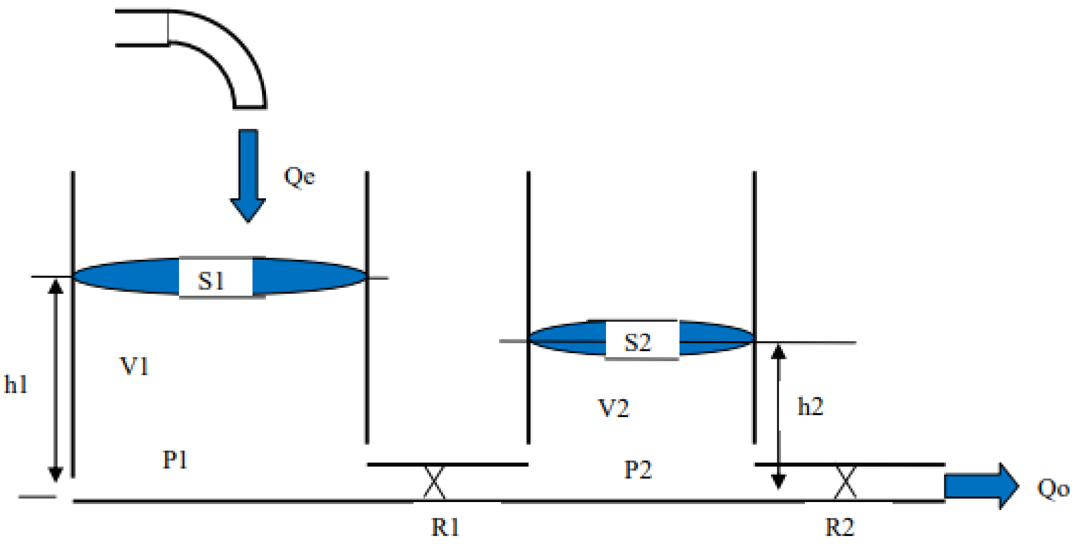
PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

2. Dois reservatórios unidos

No segundo exercício resolve-se o problema para o seguinte esquema:



Os parâmetros considerados forma replicados do primeiro exercício para os dois tanques:

$$Q_e = 0.010247$$
$$\rho = 1000$$
$$g = 10$$
$$R_a = 2 * 10\text{^}8$$
$$R_s = 2 * 10\text{^}8$$
$$S_1 = 10$$
$$S_2 = 10$$

Por meio do enunciado também sabemos que:

$$\dot{h_1} = \frac{Q_e - \sqrt{\rho \frac{g}{R_a}(h_1 - h_2)}}{S1}$$

$$\dot{h_2} = \frac{\sqrt{\rho \frac{g}{Ra}(h1 - h2)} - \sqrt{\rho g \frac{h_2}{R_s}}}{S2}$$

Sabendo as derivadas, pode-se calcular as soluções numéricas

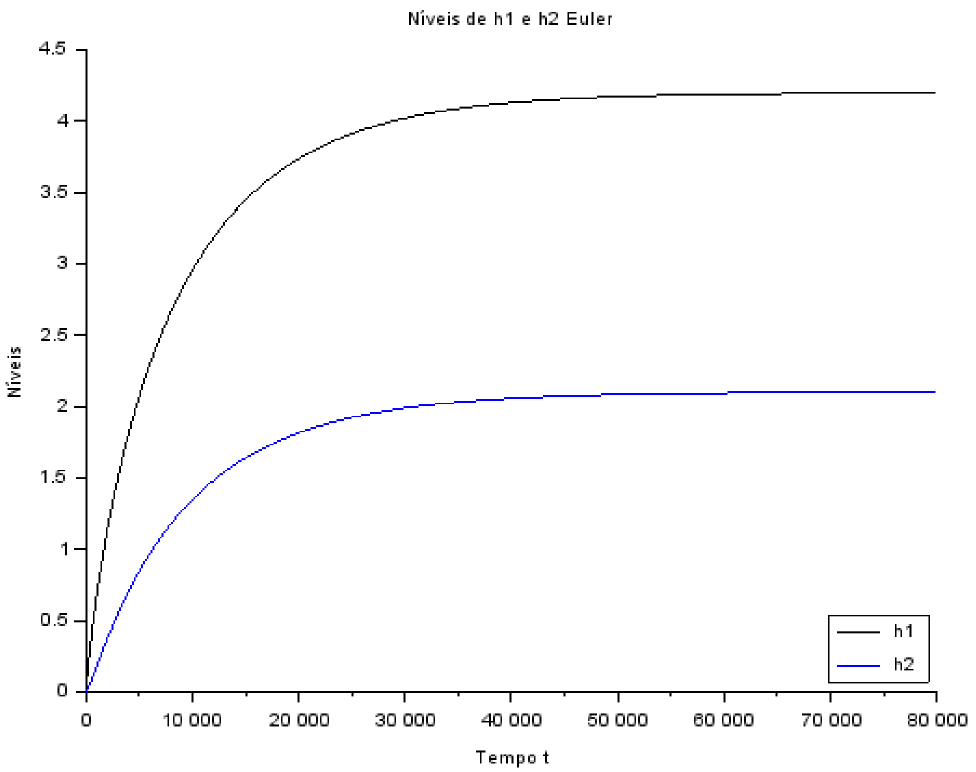
2.1. Método de Euler

O método de Euler não possui grandes modificações em relação ao primeiro caso. Basicamente, é realizado o mesmo processo nas duas equações.

$$h_1(i+1) = h_1(i) + \frac{Q_e - \sqrt{\rho \frac{g}{R_a}(h_1(\mathrm{i}) - h_2(i))}}{S1} * \mathrm{dt}$$

$$h_2(i+1) = h_2(i) + \frac{\sqrt{\rho \frac{g}{Ra}(h1(\mathrm{i}) - h2(\mathrm{i}))} - \sqrt{\rho g \frac{h_2(i)}{R_s}}}{S2} * dt$$

Os resultados encontrados para $h_{1_0} = 0$ e $h_{2_0} = 0$ são mostrados a seguir:



Observa-se que a Vazão de entrada é maior que a de saída do primeiro tanque. E também vale notar que após algum tempo as alturas se estabilizam. Ou seja a partir de, aproximadamente, 30.000 segundos as vazões de entrada e saída se igualam.

2.2. Método de Runge-Kutta 4ª ordem

O método de runge-kutta tem algumas peculiaridades nesse exercício. Todos os coeficientes do método devem ser calculados para as duas variáveis simultaneamente, pois elas são dependentes entre si. Assim, calcula-se a solução numérica da seguinte maneira:

$$K_{1_{\mathrm{h1}}}(i) = \dot{h}_1(h_1(i), h_2(i))$$
$$K_{1_{\mathrm{h2}}}(i) = \dot{h}_2(h_1(i), h_2(i))$$

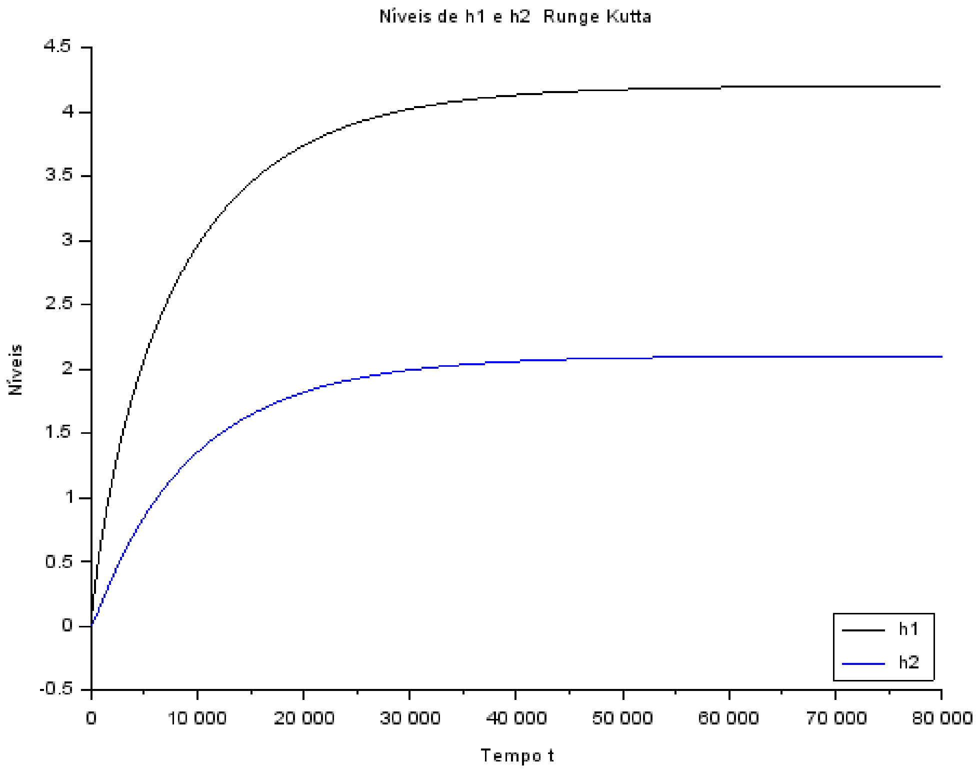
$$K_{2_{\mathrm{h1}}}(i) = \dot{h}_{\mathrm{h1}}(h_{h1}(i) + \frac{dt}{2} * K_{1_{\mathrm{h1}}}, h_{h2}(i) + \frac{dt}{2} * K_{1_{\mathrm{h2}}})$$
$$K_{2_{\mathrm{h2}}}(i) = \dot{h}_{\mathrm{h2}}(h_{h1}(i) + \frac{dt}{2} * K_{1_{\mathrm{h1}}}, h_{h2}(i) + \frac{dt}{2} * K_{1_{\mathrm{h2}}})$$

$$K_{3_{\mathrm{h1}}}(i) = \dot{h}_{\mathrm{h1}}(h_{h1}(i) + \frac{dt}{2} * K_{2_{\mathrm{h1}}}, h_{h2}(i) + \frac{dt}{2} * K_{2_{\mathrm{h2}}})$$
$$K_{3_{\mathrm{h2}}}(i) = \dot{h}_{\mathrm{h2}}(h_{h1}(i) + \frac{dt}{2} * K_{2_{\mathrm{h1}}}, h_{h2}(i) + \frac{dt}{2} * K_{2_{\mathrm{h2}}})$$

$$K_{4_{\mathrm{h1}}}(i) = \dot{h}_{\mathrm{h1}}(h_{h1}(i) + dt * K_{3_{\mathrm{h1}}}, h_{h2}(i) + \mathrm{dt} * K_{3_{\mathrm{h2}}})$$
$$K_{4_{\mathrm{h2}}}(i) = \dot{h}_{\mathrm{h2}}(h_{h1}(i) + dt * K_{3_{\mathrm{h1}}}, h_{h2}(i) + \mathrm{dt} * K_{3_{\mathrm{h2}}})$$

$$h_1(i+1) = h_1(i) + \frac{dt}{6}(K_{1_{\mathrm{h1}}} + 2K_{2_{\mathrm{h1}}} + 2K_{3_{\mathrm{h1}}} + K_{4_{\mathrm{h1}}})$$
$$h_2(i+1) = h_2(i) + \frac{dt}{6}(K_{1_{\mathrm{h2}}} + 2K_{2_{\mathrm{h2}}} + 2K_{3_{\mathrm{h2}}} + K_{4_{\mathrm{h2}}})$$

Os resultados para $h_{1_0} = 0$ e $h_{2_0} = 0$ são mostrados a seguir:



Observa-se que o resultado é muito semelhante ao encontrado pelo método de Euler. Dessa forma, validam-se os programas utilizados.

3. Códigos

3.1. Euler para o exercício 1

```
// Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
// Apagando dados anteriores:
clear
// Carregando a equacao diferencial:
function [ydot]=funcao(y)
ydot=(-sqrt(rho*g*y/R)+Qe)*1/S;
endfunction

//parâmetros do exercício
Qe=0.010247
rho=1000
g=10
R=2*10^8
S=10

```

```
// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=1;
// Valor inicial da solucao exata:
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;
//pressão
P=rho*g*y(i);
//Vazao de saída
Qs=sqrt(P/R);
// Solucao numerica:
y(i+1)=y(i)+h* funcao(y(i));

// 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,y);
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Nível do reservatório único Euler","Tempo t","Solucao")
```

3.2. Runge-Kutta para o exercício 1

```
// Conjunto de comandos para solucao numerica de equacao diferencial [1-y(i)]/2
// Apagando dados anteriores:
clear
function [ydot]=funcao(y)
ydot=(-sqrt(rho*g*y/R)+Qe)*1/S;
endfunction

//parâmetros do exercício
Qe=0.010247
rho=1000
g=10
R=2*10^8
S=10

// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=1;
// 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=funcao(y(i));
k2=funcao(y(i)+k1*h/2);
k3=funcao(y(i)+k2*h/2);
k4=funcao(y(i)+k3*h);

y(i+1)=y(i)+(h*(k1+2*k2+2*k3+k4)/6);
// 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,y);
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Nível do reservatório único Runge Kutta","Tempo t","Solucao")
```

3.3. Euler para o exercício 2

```
// Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
// Apagando dados anteriores:
clear
function [h1_dot, h2_dot]=funcao(h1, h2)
h1_dot=(Qe-sqrt(rho*g/Ra*(h1-h2)))/S1
h2_dot=(sqrt(rho*g/Ra*(h1-h2))-sqrt(rho*g*h2/Rs))/S2
endfunction

// Instante final:
tf=80000;
// Condicao inicial:
h1(1)=0;
h2(1)=0;

//parâmetros do exercício
Qe=0.010247
rho=1000
g=10
Ra=2*10^8
Rs=2*10^8
S1=10
S2=10

// Instante inicial:
t(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.25;
// 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;
//calculando as derivada
[h1_dot,h2_dot]=funcao(h1(i),h2(i))
// Solucao numerica:
h1(i+1)=h1(i)+h* h1_dot;
```

```
h2(i+1)=h2(i)+h* h2_dot;
// Termino do comando for:
end
plot2d(t,h1)
// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
plot2d(t,[h1,h2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["h1","h2"],[1 2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Níveis de h1 e h2 Euler","Tempo t","Níveis")
```

3.4. Runge-Kutta para o exercício 2

```
// Conjunto de comandos para solucao numerica de equacao diferencial [1-y1(i)]/2
// Apagando dados anteriores:
clear

//funções que calculam as derivadas
function [h1_dot]=Dh1(h1, h2)
h1_dot=(Qe-sqrt(rho*g/Ra*(h1-h2)))/S1
endfunction

function [h2_dot]=Dh2(h1, h2)
h2_dot=(sqrt(rho*g/Ra*(h1-h2))-sqrt(rho*g*h2/Rs))/S2
endfunction

// Instante final:
tf=80000;
// Condicao inicial:
h1(1)=0;
h2(1)=0;

//parâmetros do exercício
Qe=0.010247
rho=1000
g=10
Ra=2*10^8
Rs=2*10^8
S1=10
S2=10

// Instante inicial:
t(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_h1=Dh1(h1(i),h2(i));
k1_h2=Dh2(h1(i),h2(i));

k2_h1=Dh1(h1(i)+k1_h1*h/2,h2(i)+k1_h2*h/2);
k2_h2=Dh2(h1(i)+k1_h1*h/2,h2(i)+k1_h2*h/2);
```

```

k3_h1=Dh1(h1(i)+k2_h1*h/2,h2(i)+k2_h1*h/2);
k3_h2=Dh2(h1(i)+k2_h1*h/2,h2(i)+k2_h1*h/2);

k4_h1=Dh1(h1(i)+k3_h1*h,h2(i)+k3_h1*h);
k4_h2=Dh2(h1(i)+k3_h1*h,h2(i)+k3_h1*h);

h1(i+1)=h1(i)+(h*(k1_h1+2*k2_h1+2*k3_h1+k4_h1)/6);
h2(i+1)=h2(i)+(h*(k1_h2+2*k2_h2+2*k3_h2+k4_h2)/6);

// 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,[h1,h2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["h1","h2"],[1 2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Níveis de h1 e h2  Runge Kutta","Tempo t","Níveis")
```