Escola Politécnica da Universidade de São Paulo

Lista B – PME3380 – Modelagem de Sistemas Dinâmicos

Turma 1

Professor: Agenor de Toledo Fleury

Aluno: Henrique Silva Barbeta

Número USP: 10769323

São Paulo

Sumário

1. Exemplos

1.1. Método de Euler

Foi criada a função “funcao.sci”:

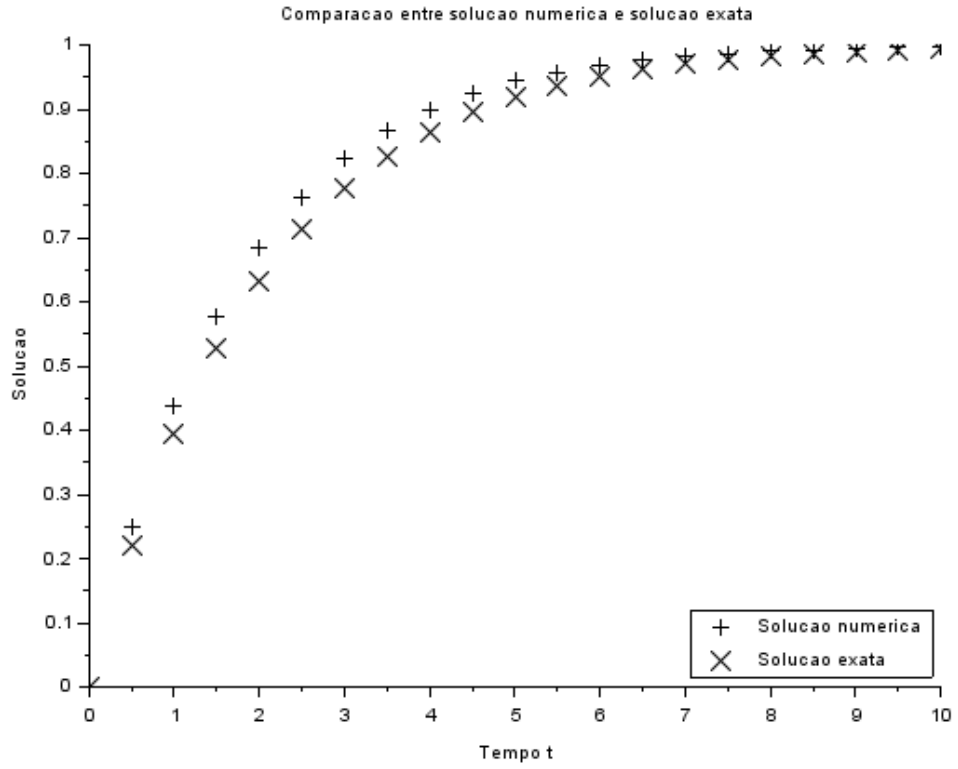
```
function [ydot]=funcao(y)
ydot=(1-y)/2;
endfunction
save('funcao.sci','funcao');
```

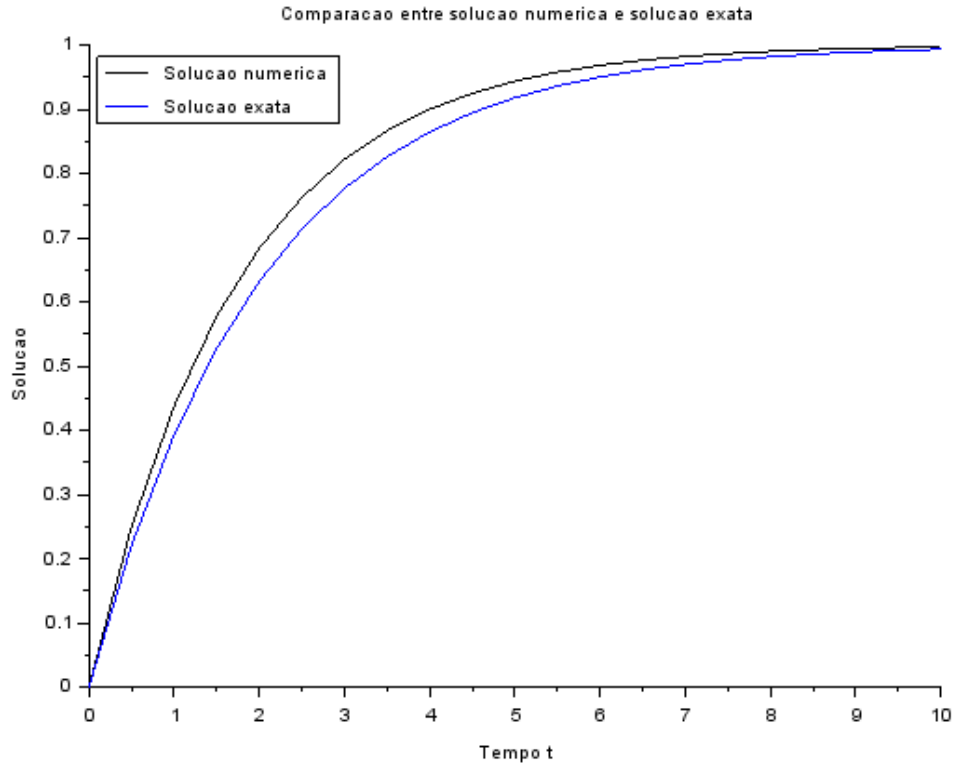
Depois criou-se “numericoE.sce”:

```
// 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:
```

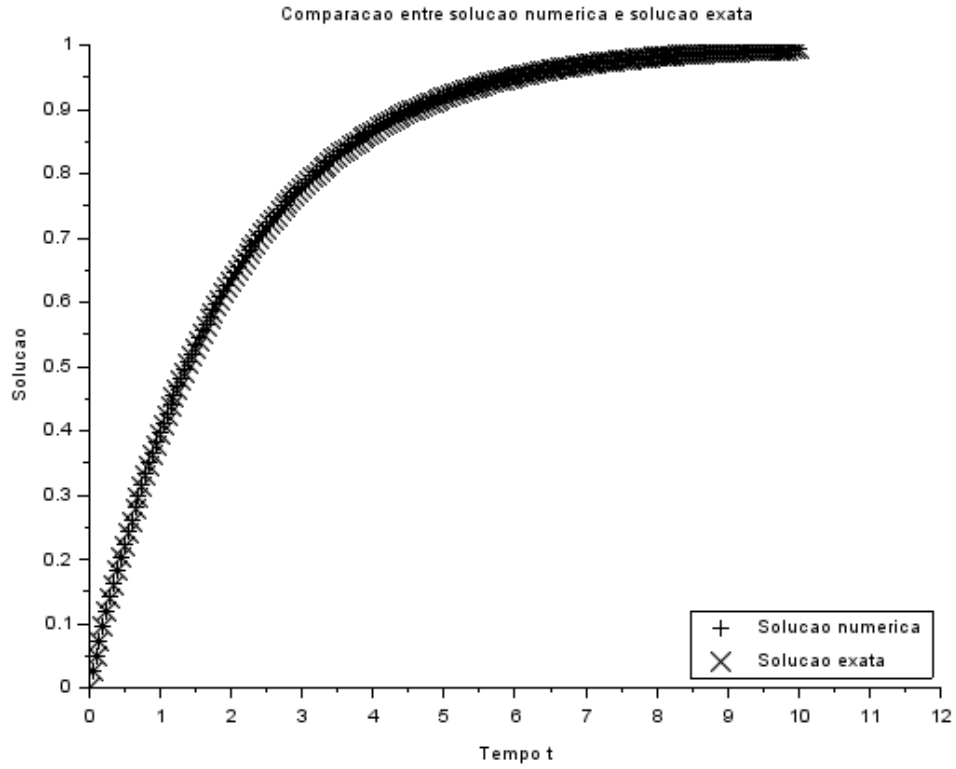
```
xtitle("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao")
// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Comparacao entre solucao numerica 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));
```

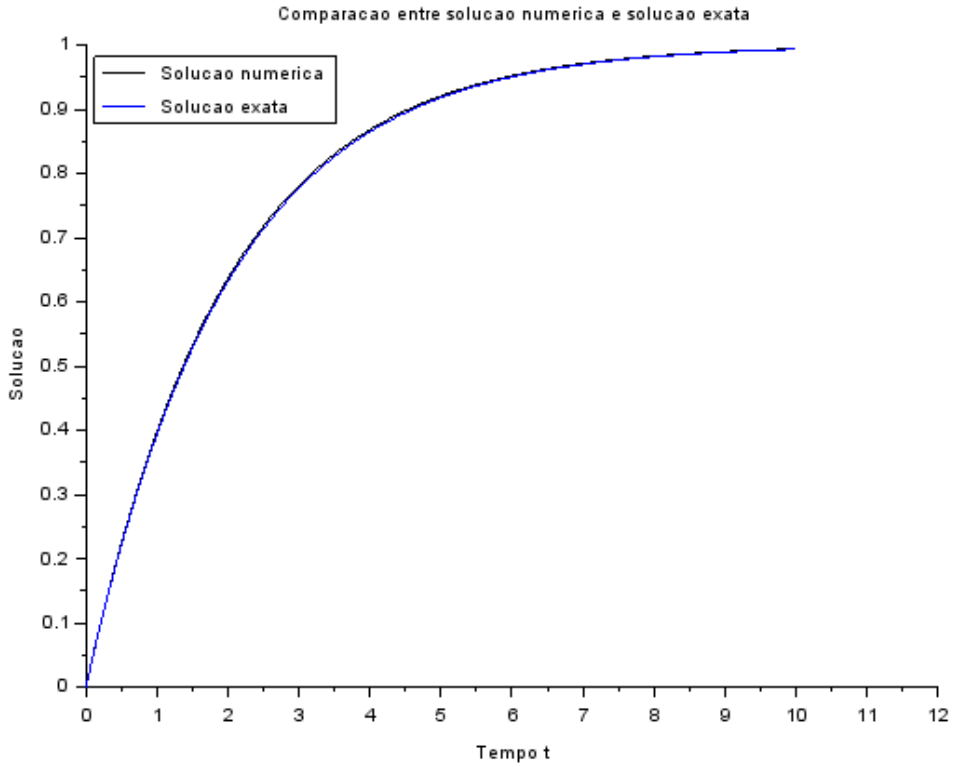
Executou-se primeiro “função.sci” e depois executou-se “numericoE.sce”, assim, obteve-se os seguintes gráficos:





Para fins de testes, foi trocado o passo “h” de 0,5 para 0,05.





Nota-se que há muito mais pontos e que a solução numérica é extremamente mais próxima à solução real. O passo diminuiu em 10x, e o efeito obtido foi extremamente significativo.

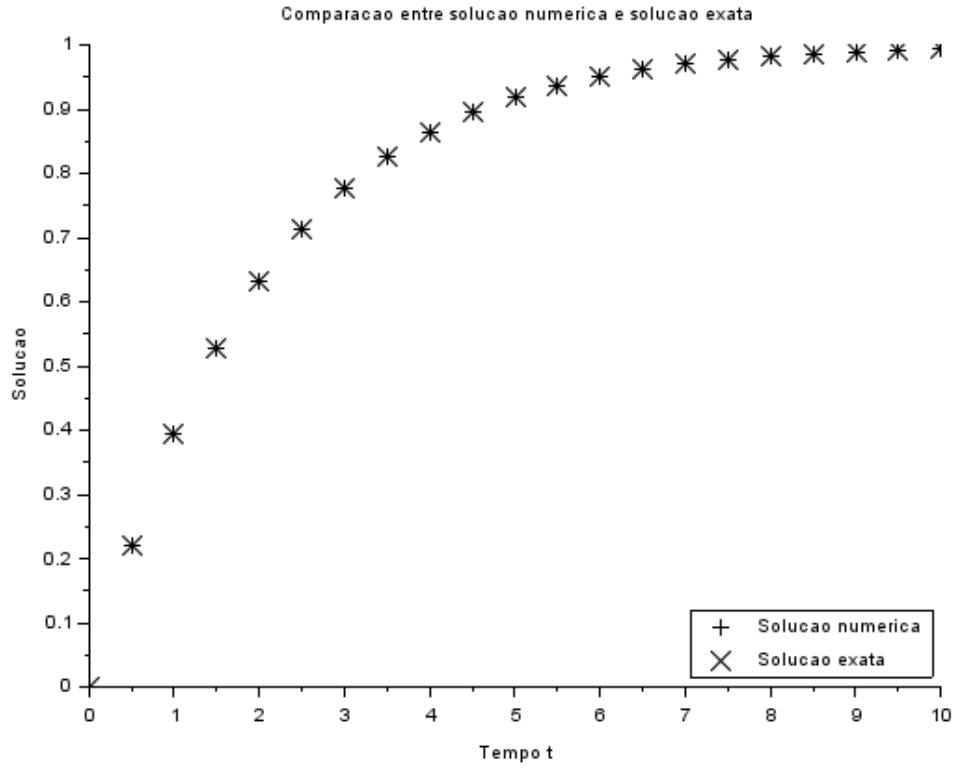
1.2. Método de Runge-Kutta

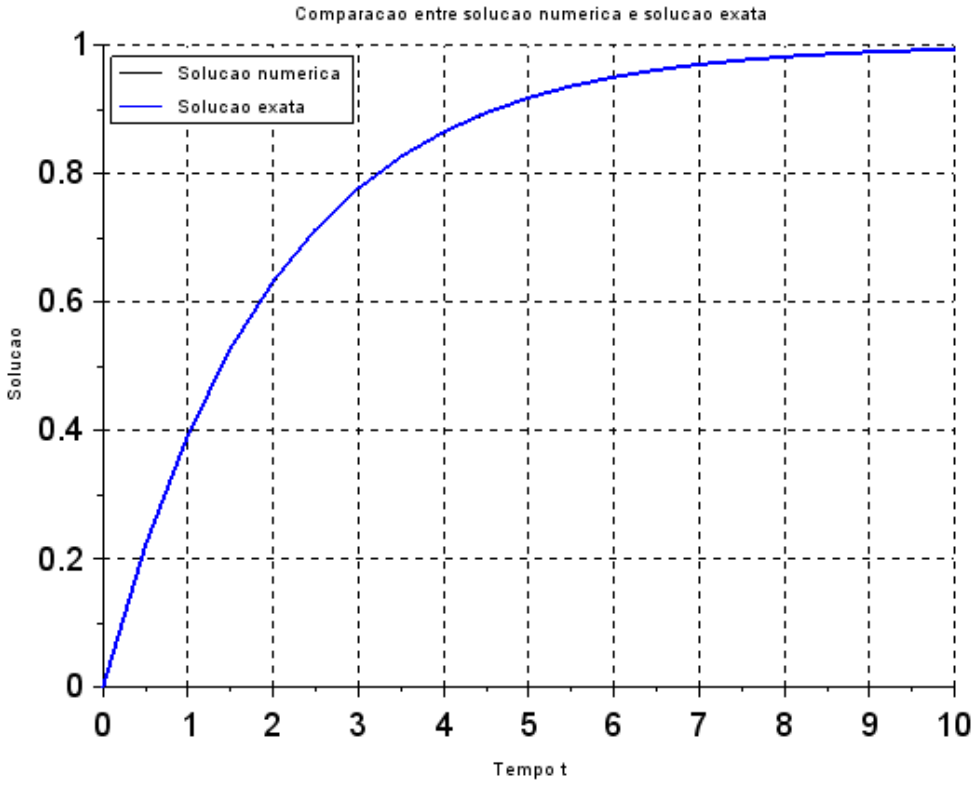
Implementou-se depois a função, para ser integrada, pelo método de Runge-Kutta, apartir do código abaixo:

```
// Conjunto de comandos para solucao numerica de equacao diferencial [1-y(i)]/2
// 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-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(["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);
// 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("Comparacao entre solucao numerica e solucao exata","Tempo
t","Solucao","Solucao numerica","Solucao 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 os eixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)
```

Foram plotados os gráficos, abaixo:





Podemos ver que a convergência para o valor real é muito melhor por esse método.

2. Exercício

2.1. Exercício 1 – reservatório único

Seguindo o padrão dos exemplos, foi criado um arquivo separado para a função, e outro arquivo para os métodos de integração.

O primeiro arquivos criado foi chamado de “func.sci” e o código dele está abaixo:

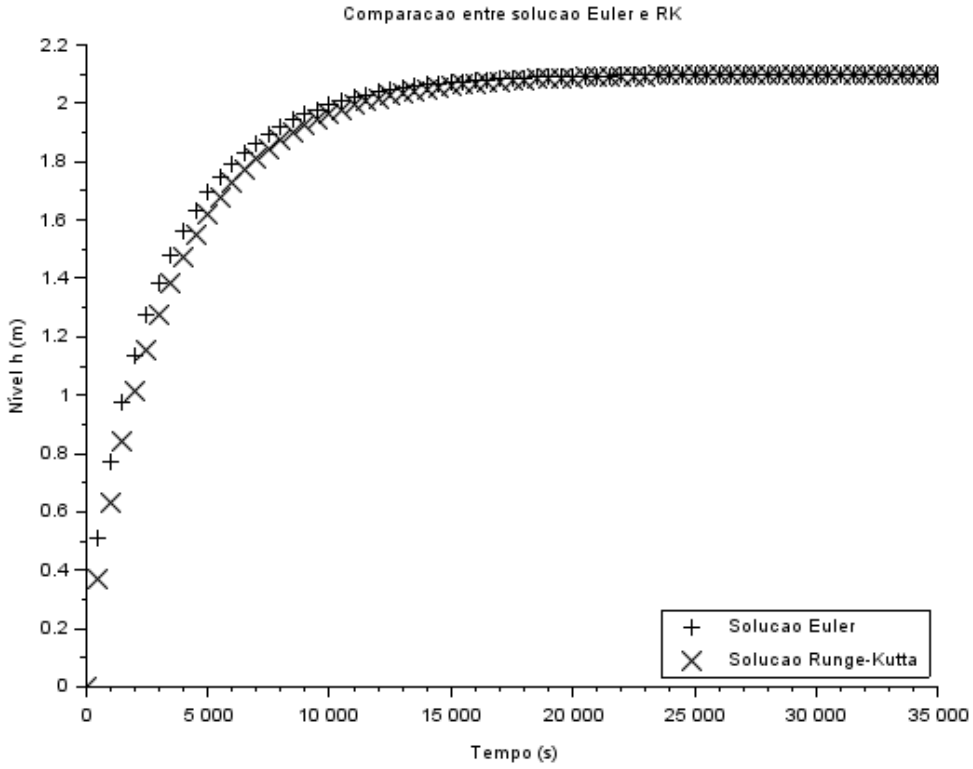
```
function [hdot]=func(h, ro, G, S, R, Qe)
hdot=(Qe-sqrt((ro*G*h)/R))/S;
endfunction
save('func.sci','func');
```

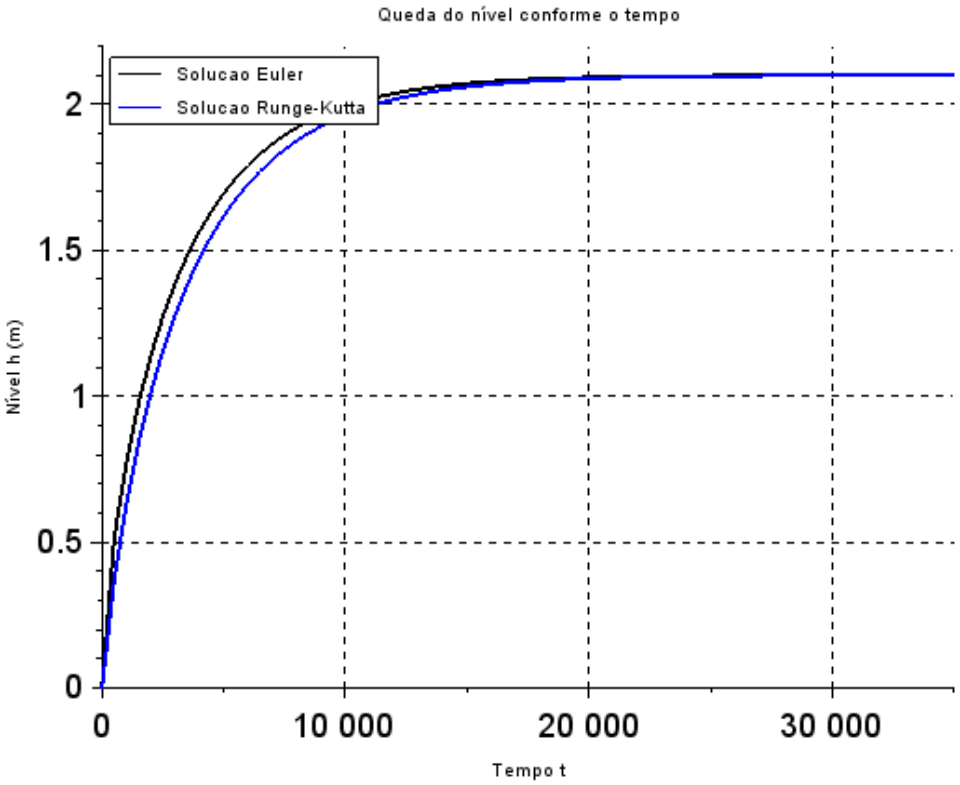
Depois foi criado o arquivo “NivelEeRK”, com o seguinte código:

```
// Conjunto de comandos para solucao numerica de equacao diferencial (Qe-
sqrt((ro*G*h)/R))/S
// Apagando dados anteriores:
clear
// Carregando a equacao diferencial:
// Carregue a função usando o comando Load do Scilab
load('func.sci','func');
//Definição dos parâmetros
S = 10;
R = 2*(10^8);
ro = 1000;
G = 10;
Qe = 0.010247;
// Instante inicial:
t(1)=0;
// Instante final:
tf=35000;
// Condicao inicial Euler:
yE(1)=0;
// Valor inicial Runge-Kutta:
yR(1)=0;
// Passo de integracao (experimente alterar o passo):
h=500;
// 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 Euler:
yE(i+1)=yE(i)+h*func(yE(i), ro, G, S, R, Qe);
// Solucao Runge-Kutta:
k1=func(yR(i), ro, G, S, R, Qe);
```

```
k2=func(yR(i)+h*0.5*k1, ro, G, S, R, Qe);
k3=func(yR(i)+h*0.5*k2, ro, G, S, R, Qe);
k4=func(yR(i)+h*k3, ro, G, S, R, Qe);
yR(i+1)=yR(i)+((k1+2*k2+2*k3+k4)*(h/6));
// Termino do comando for:
end
// Plotando solucao Euler yE versus vetor de tempo t e solucao RK yR versus vetor de
tempo t:
plot2d([t,t],[yE,yR],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solucao Euler","Solucao Runge-Kutta"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre solucao Euler e RK","Tempo (s)","Nível h (m)")
// 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],[yE,yR],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Queda do nível conforme o tempo","Tempo t","Nível h (m)","Solucao
Euler","Solucao Runge-Kutta");
// 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 os eixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)
```

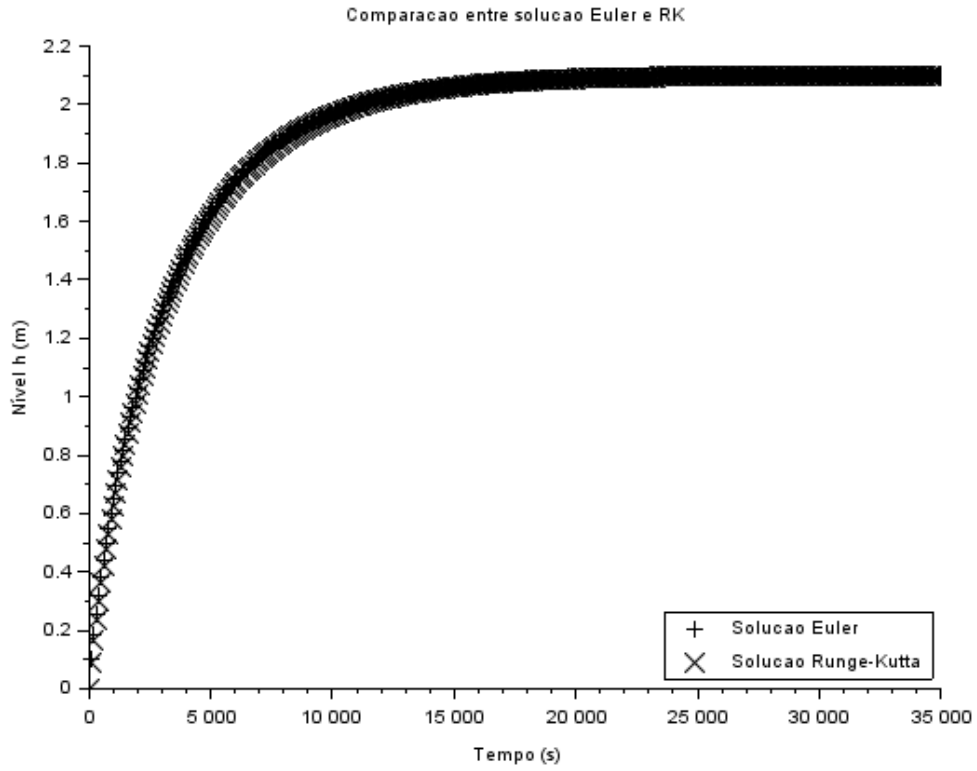
Usando esse passo de integração, obteve-se os seguinte gráficos:

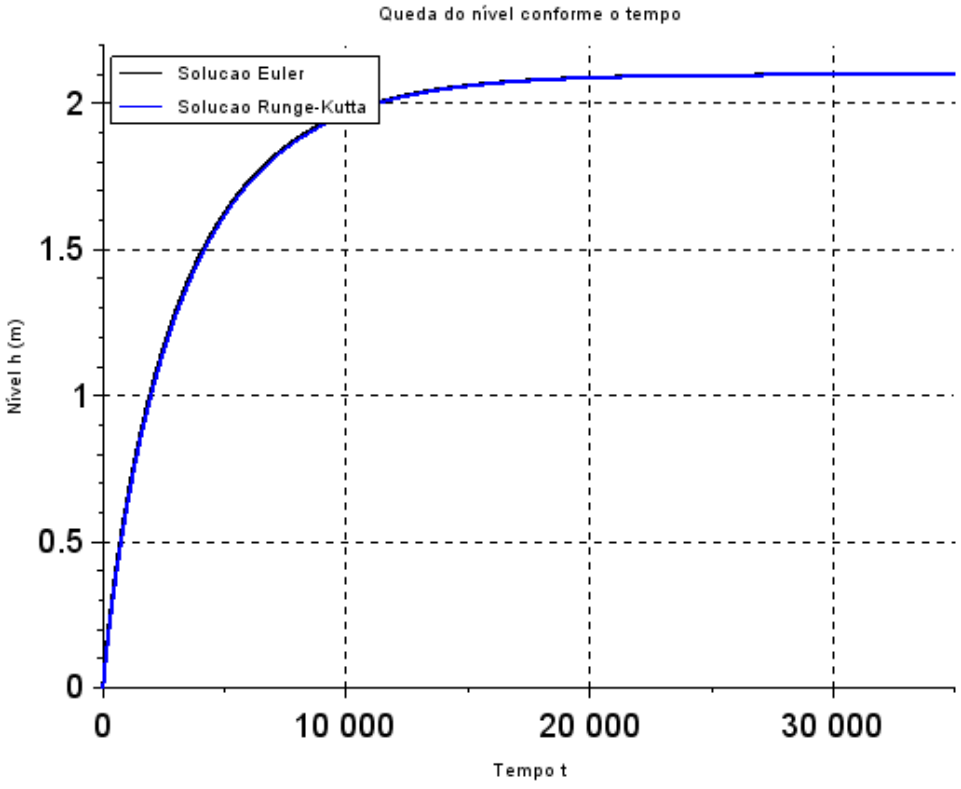




Pode-se perceber que com esse número de passos, é possivel ver algumas diferenças entre os dois gráficos.

Assim, para quetões de aproximação do valor real, diminuiu-se o passo de integração, de h=500, para h=100, obtendo-se os gráficos, seguintes:





Em, aproximadamente 2,1m, houve uma estabilização do nível do reservatório.

2.2. Exercício 2 – dois reservatórios

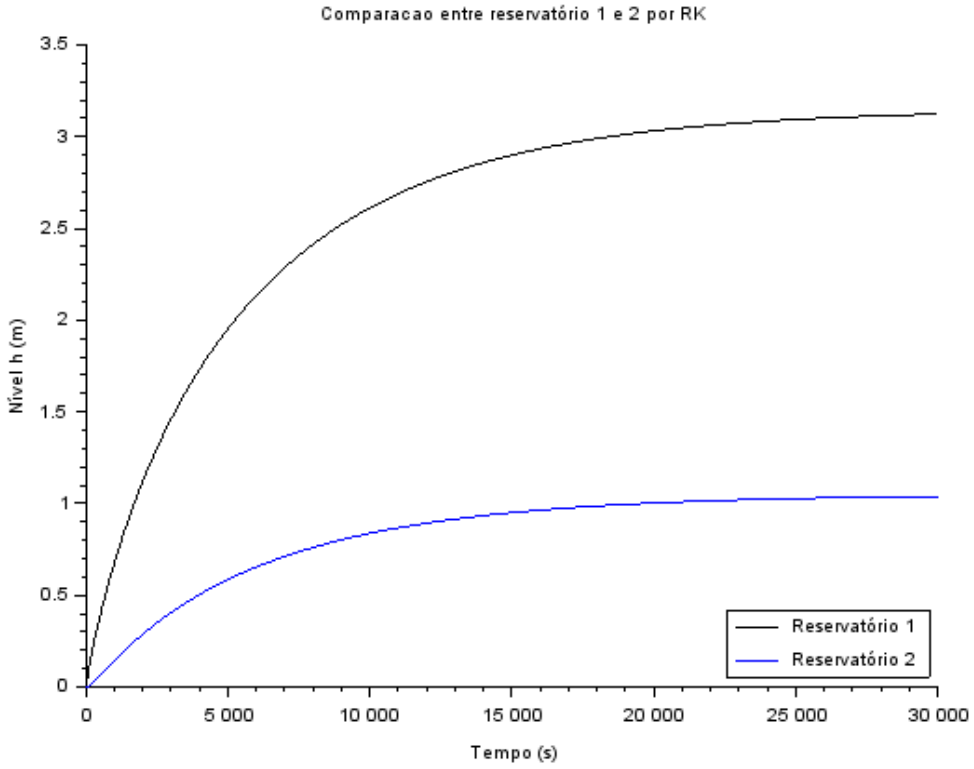
Agora foi feito o mesmo exercício, porém com 2 reservatórios.

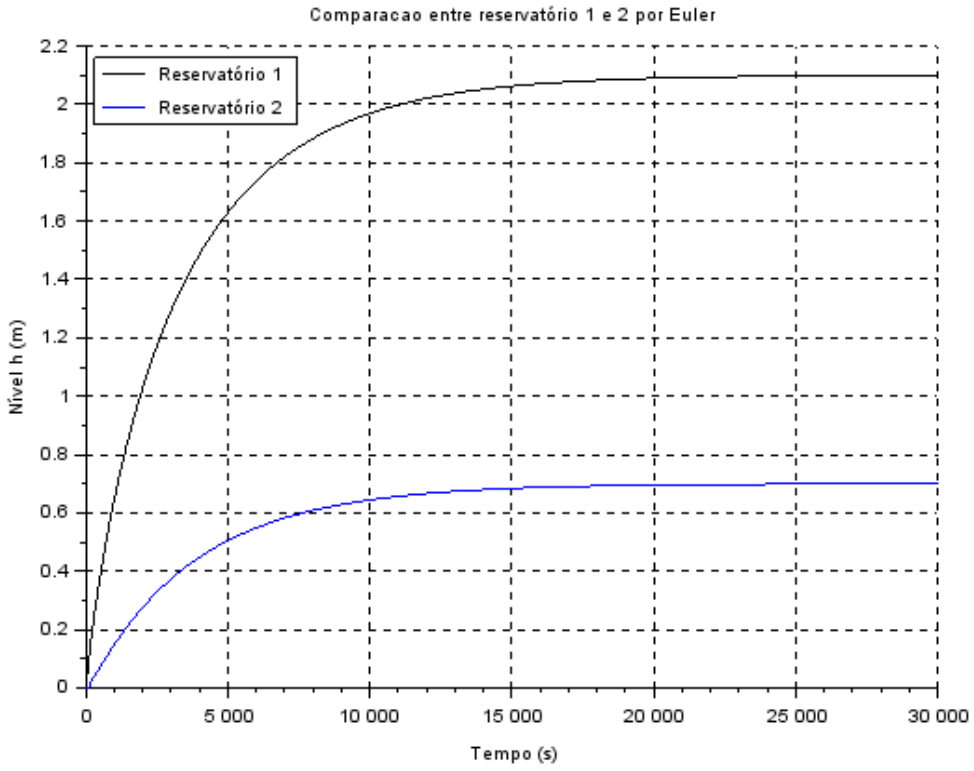
Assim, o código teve que ser levemente modificado:

```
// Conjunto de comandos para solucao numerica de equacao diferencial
// Apagando dados anteriores:
clear
//função 1
function [hdot1]=f1(h1, h2, ro, G, S1, Ra, Qe)
  hdot1=(Qe-sqrt((ro*G*(h1-h2))/Ra))/S1;
endfunction
//função 2
function [hdot2]=f2(h1, h2, ro, G, S2, Ra, Rs)
  hdot2=(sqrt(ro*G*(h1-h2)/Ra)-sqrt(ro*G*h2/Rs))/S2;
endfunction
//Definição dos parâmetros
S1 = 10;
S2 = 8;
Ra = 2*(10^8);
Rs = 1*(10^8);
ro = 1000;
G = 10;
Qe = 0.010247;
// Instante inicial:
t(1)=0;
// Instante final:
tf=30000;
// Condicao inicial 1:
r1(1)=0;
w1(1)=0;
// Condicao inicial 2:
r2(1)=0;
w2(1)=0;
// Passo de integracao (experimente alterar o passo):
h=100;
// Calculo de numero de passos):
n=round((tf-t(1))/h);
// Integracao numerica usando o metodo de Runge Kutta e Euler:
// Comando for:
for i=1:n
// Vetor de tempo:
  t(i+1)=t(i)+h;
// Solucao Runge-Kutta 1:
  k1=f1(r1(i), r2(i), ro, G, S1, Ra, Qe);
  k2=f1(r1(i)+h*0.5*k1, r2(i)+h*0.5*k1, ro, G, S1, Ra, Qe);
  k3=f1(r1(i)+h*0.5*k2, r2(i)+h*0.5*k2, ro, G, S1, Ra, Qe);
```

```
   k4=f1(r1(i)+h*k3, r2(i)+h*k3, ro, G, S1, Ra, Qe);
   r1(i+1)=r1(i)+((k1+2*k2+2*k3+k4)*(h/6));
// Solucao Runge-Kutta 2:
   k1=f2(r1(i), r2(i), ro, G, S2, Ra, Rs);
   k2=f2(r1(i)+h*0.5*k1, r2(i)+h*0.5*k1, ro, G, S2, Ra, Rs);
   k3=f2(r1(i)+h*0.5*k2, r2(i)+h*0.5*k2, ro, G, S2, Ra, Rs);
   k4=f2(r1(i)+h*k3, r2(i)+h*k3, ro, G, S2, Ra, Rs);
   r2(i+1)=r2(i)+((k1+2*k2+2*k3+k4)*(h/6));
// Solucao Euler 1:
   w1(i+1)=w1(i)+h*f1(w1(i),w2(1), ro, G, S1, Ra, Qe);
// Solucao Euler 2:
   w2(i+1)=w2(i)+h*f2(w1(i), w2(i), ro, G, S2, Ra, Rs);
// Termino do comando for:
end
// Plotando solucao Runge-Kutta
plot2d([t,t],[r1,r2],[1 2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Reservatório 1","Reservatório 2"],[1,2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre reservatório 1 e 2 por RK","Tempo (s)","Nível h (m)")
// Abrindo uma nova janela de graficos:
set("current_figure", 1);
// Plotando solucao Euler
plot2d([t,t],[w1,w2],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Comparacao entre reservatório 1 e 2 por Euler","Tempo (s)","Nível h
(m)","Reservatório 1","Reservatório 2");
// 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 os eixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)
```

Esse código nos deu como respostas os dois gráficos abaixo:





Nota-se que eles são extremamente parecidos, como já era esperado.