Nome: Gabriel Pinheiro **NUSP**: 10336595

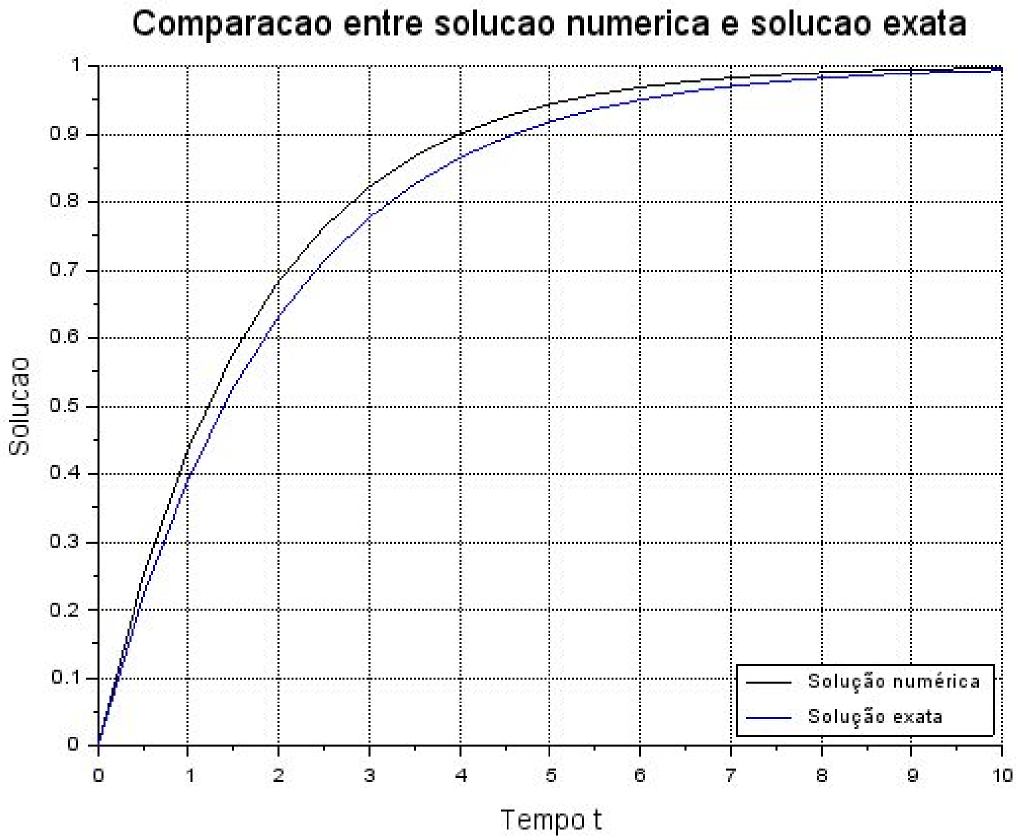
RESOLUÇÃO LISTA B

Exemplo 1:

1. Algoritmo

```
clear

ti = 0;
tf = 10;
n = 20;
h = (tf-ti)/n;

t = zeros(n+1,1)
for i = 2:n+1
   t(i,1) = t(i-1,1) + h;
end

y = zeros(n+1,1);
ye = zeros(n+1,1);
//f = (1-y)/2

for i = 2:n+1
   f = (1 - y(i-1,1))/2;
   y(i,1) = y(i-1,1) + h*f;
   ye(i,1) = 1 - %e**(-t(i,1)/2)
end

plot2d([t,t],[y,ye],[1 2]);
legends(["Solução numérica", "Solução exata"],[1,2],4)
title("Comparacao entre solucao numerica e solucao exata", "fontsize", 4)
xlabel("Tempo t", "fontsize", 3)
ylabel("Solucao", "fontsize", 3)
xgrid(1);
```

2. Resultados

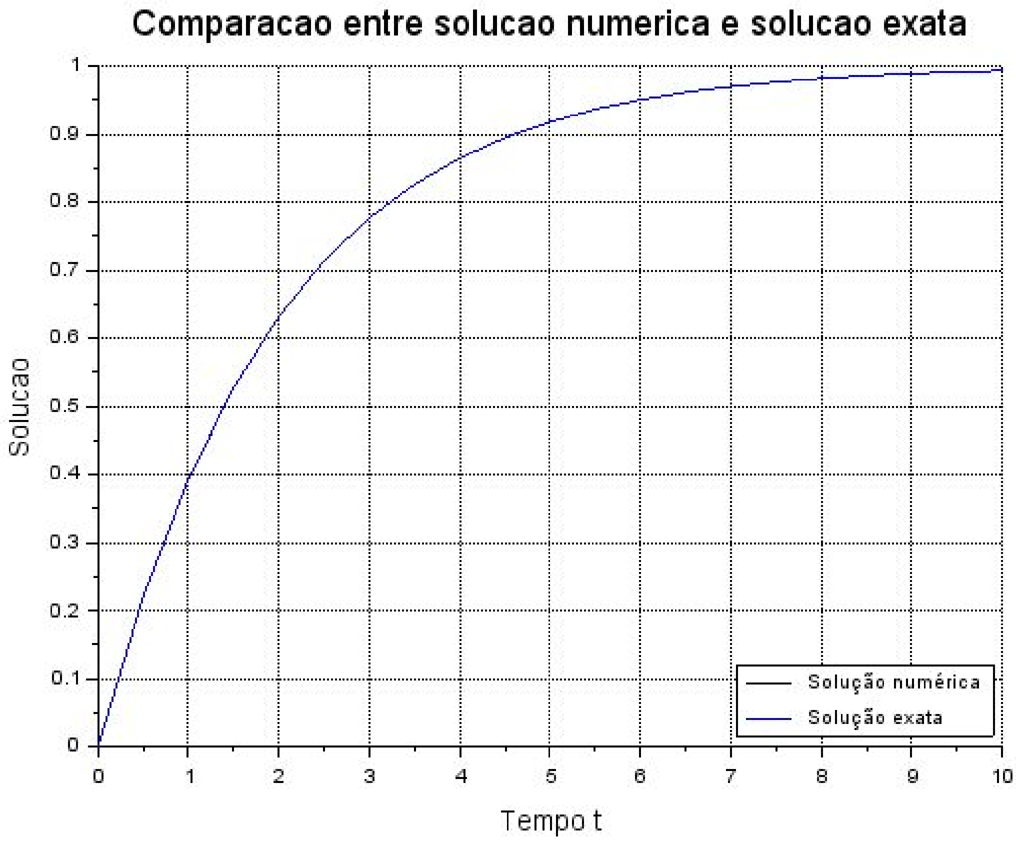


Exemplo 2:

1. Algoritmo

```
clear

function [yp]=f(y)
  yp = (1-y)/2;
endfunction

ti = 0;
tf = 10;
n = 20;
h = (tf-ti)/n;

t = zeros(n+1,1)
for i = 2:n+1
  t(i,1) = t(i-1,1) + h;
end

y = zeros(n+1,1);
ye = zeros(n+1,1);
//f = (1-y)/2

for i = 2:n+1
  k1 = f(y(i-1,1));
  k2 = f(y(i-1,1) + h*0.5*k1);
  k3 = f(y(i-1,1) + h*0.5*k2);
  k4 = f(y(i-1,1) + h*k3);

  y(i,1) = y(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4)
  ye(i,1) = 1 - %e**(-t(i,1)/2)
end

plot2d([t,t],[y,ye],[1 2]);
legends(["Solução numérica", "Solução exata"],[1,2],4);
title("Comparacao entre solucao numerica e solucao exata", "fontsize", 4);
xlabel("Tempo t", "fontsize", 3);
ylabel("Solucao", "fontsize", 3);
xgrid(1);
```

2. Resultados

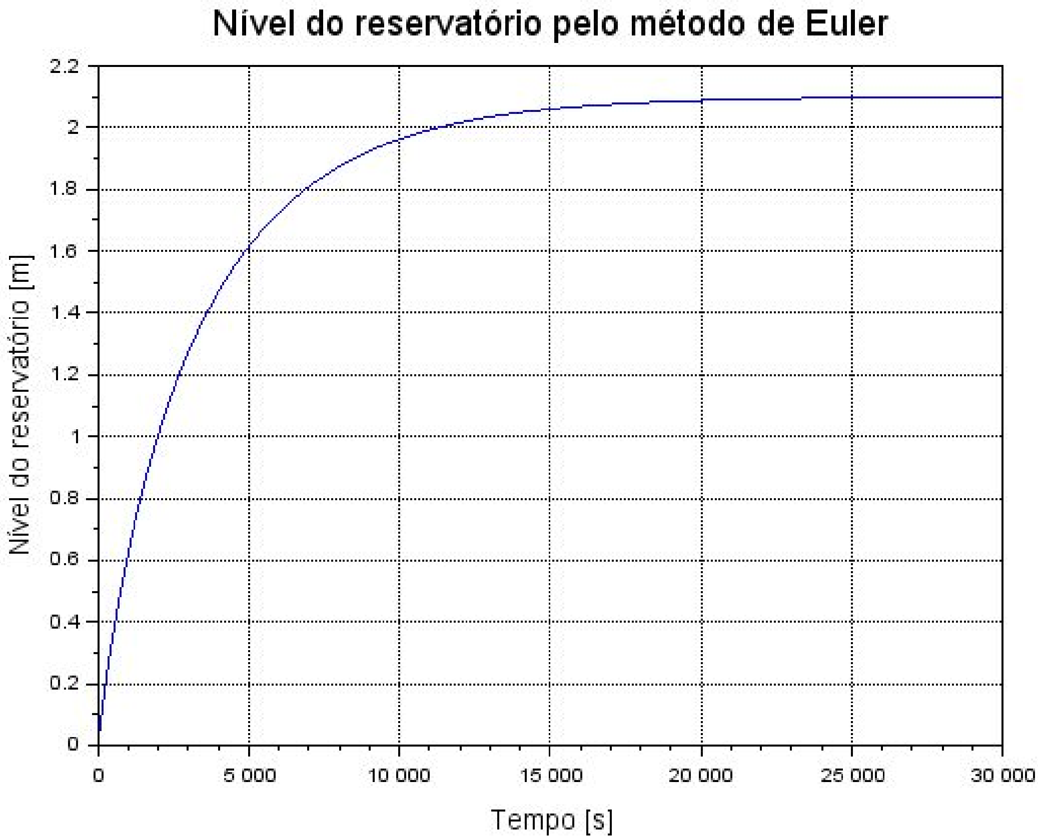


Exercício 1 Reservatório (Método de Euler):

1. Algoritmo

```
clear

function [hp]=f(h, rho, g, R, Qe, S)
   hp = (Qe - sqrt(rho*g*h/R))/S;
endfunction

Qe = 0.010247;
rho = 1000;
g = 10;
R = 2e+8;
S = 10;

ti = 0;
tf = 30000;
n = 60000;
h = (tf-ti)/n;

t = zeros(n+1,1)
for i = 2:n+1
   t(i,1) = t(i-1,1) + h
end

H = zeros(n+1,1);
for i = 2:n+1
   H(i,1) = H(i-1,1) + h*f(H(i-1,1), rho, g, R, Qe, S)
end

plot(t,H);
title("Nível do reservatório pelo método de Euler", "fontsize", 4)
xlabel("Tempo [s]", "fontsize", 3)
ylabel("Nível do reservatório [m]", "fontsize", 3)
xgrid(1);
```

2. Resultados

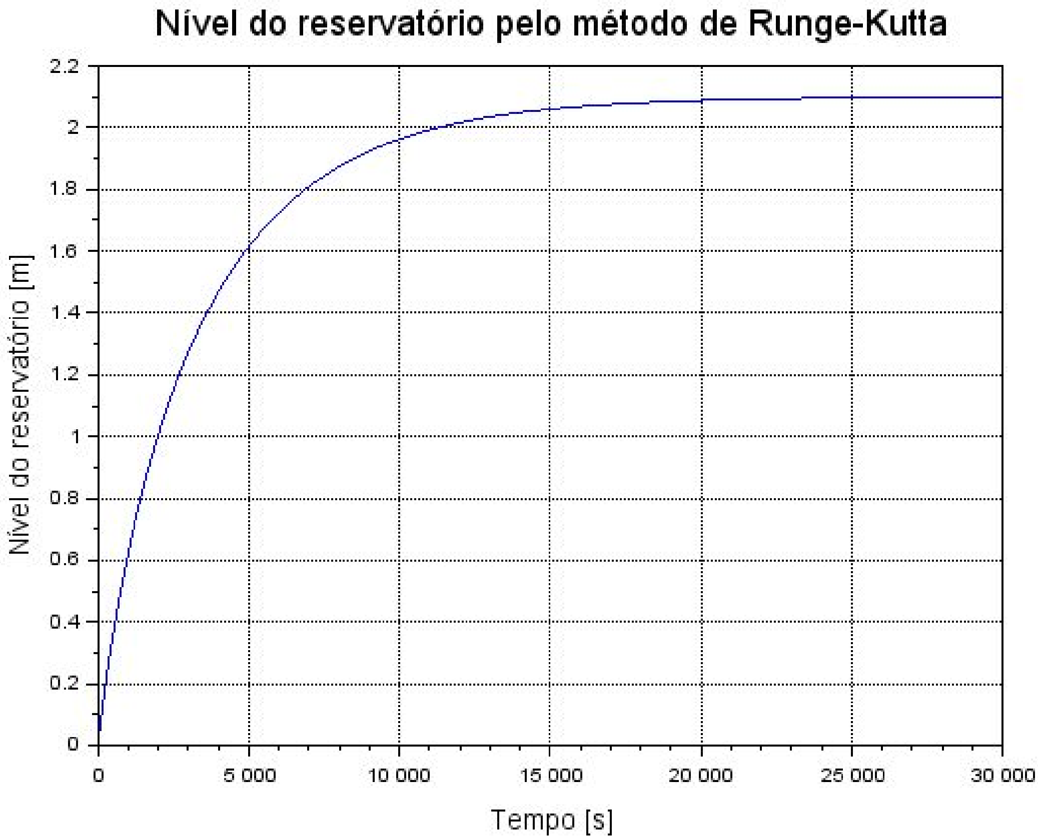


Exercício 1 Reservatório (Método de Runge-Kutta de 4° Ordem):

1. Algoritmo

```
clear

function [hp]=f(h, rho, g, R, Qe, S)
   hp = (Qe - sqrt(rho*g*h/R))/S;
endfunction

Qe = 0.010247;
rho = 1000;
g = 10;
R = 2e+8;
S = 10;

ti = 0;
tf = 30000;
n = 60000;
h = (tf-ti)/n;

t = zeros(n+1,1)
for i = 2:n+1
   t(i,1) = t(i-1,1) + h
end

H = zeros(n+1,1);
for i = 2:n+1
   k1 = f(H(i-1,1), rho, g, R, Qe, S);
   k2 = f(H(i-1,1) + h*0.5*k1, rho, g, R, Qe, S);
   k3 = f(H(i-1,1) + h*0.5*k2, rho, g, R, Qe, S);
   k4 = f(H(i-1,1) + h*k3, rho, g, R, Qe, S);
   H(i,1) = H(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4);
end

plot(t,H);
title("Nível do reservatório pelo método de Runge-Kutta", "fontsize", 4)
xlabel("Tempo [s]", "fontsize", 3)
ylabel("Nível do reservatório [m]", "fontsize", 3)
xgrid(1);
```

2. Resultados



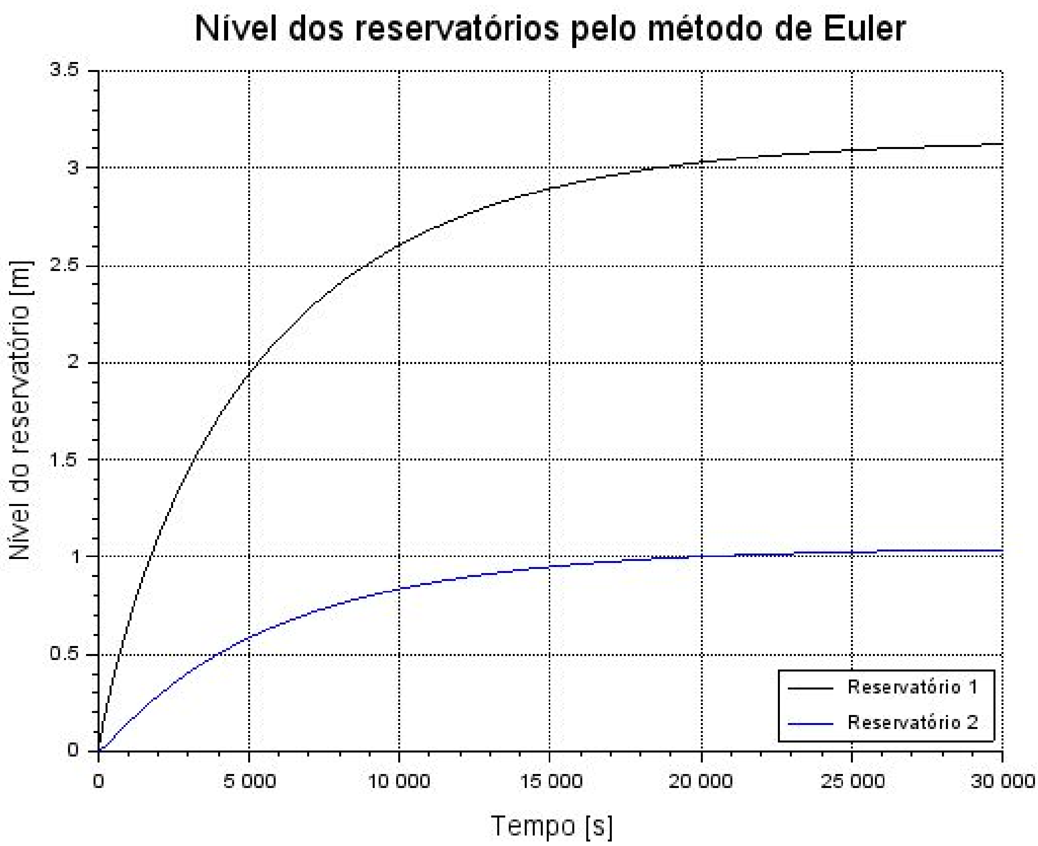
Exercício 2 Reservatórios (Método de Euler):

1. Algoritmo

```
clear

function [hp1]=f1(h1, h2, Qe, rho, g, Ra, S1)
   hp1 = (Qe-sqrt(rho*g*(h1-h2)/Ra))/S1;
endfunction
function [hp2]=f2(h1, h2, rho, g, Ra, Rs, S2)
   hp2 = (sqrt(rho*g*(h1-h2)/Ra)-sqrt(rho*g*h2/Rs))/S2;
endfunction

Qe = 0.010247;
rho = 1000;
g = 10;
Ra = 2e+8;
S1 = 10;
Rs = 1e+8;
S2 = 8;

ti = 0;
tf = 30000;
n = 60000;
h = (tf-ti)/n;

t = zeros(n+1,1)
for i = 2:n+1
   t(i,1) = t(i-1,1) + h;
end

h1 = zeros(n+1,1);
h2 = zeros(n+1,1);

for i = 2:n+1
   h1(i,1) = h1(i-1,1) + h*f1(h1(i-1,1), h2(i-1,1), Qe, rho, g, Ra, S1);
   h2(i,1) = h2(i-1,1) + h*f2(h1(i-1,1), h2(i-1,1), rho, g, Ra, Rs, S2)
end

plot2d([t,t],[h1,h2],[1 2]);
legends(["Reservatório 1", "Reservatório 2"],[1,2],4)
title("Nível dos reservatórios pelo método de Euler", "fontsize", 4)
```

```
xlabel("Tempo [s]", "fontsize", 3)
ylabel("Nível do reservatório [m]", "fontsize", 3)
xgrid(1);
```

2. Resultados



Exercício 2 Reservatórios (Método de Runge-Kutta de 4° ordem):

1. Algoritmo

```
clear

function [hp1]=f1(h1, h2, Qe, rho, g, Ra, S1)
   hp1 = (Qe-sqrt(rho*g*(h1-h2)/Ra))/S1;
endfunction
function [hp2]=f2(h1, h2, rho, g, Ra, Rs, S2)
   hp2 = (sqrt(rho*g*(h1-h2)/Ra)-sqrt(rho*g*h2/Rs))/S2;
endfunction

Qe = 0.010247;
rho = 1000;
g = 10;
Ra = 2e+8;
S1 = 10;
Rs = 1e+8;
S2 = 8;

ti = 0;
tf = 30000;
n = 60000;
h = (tf-ti)/n;

t = zeros(n+1,1)
for i = 2:n+1
   t(i,1) = t(i-1,1) + h;
end

h1 = zeros(n+1,1);
h2 = zeros(n+1,1);

for i = 2:n+1
   k1 = f1(h1(i-1,1), h2(i-1,1), Qe, rho, g, Ra, S1);
   k2 = f1(h1(i-1,1) + h*0.5*k1, h2(i-1,1) + h*0.5*k1, Qe, rho, g, Ra, S1);
   k3 = f1(h1(i-1,1) + h*0.5*k2, h2(i-1,1) + h*0.5*k2, Qe, rho, g, Ra, S1);
   k4 = f1(h1(i-1,1) + h*k3, h2(i-1,1) + h*k3, Qe, rho, g, Ra, S1);
   h1(i,1) = h1(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4);

   k1 = f2(h1(i-1,1), h2(i-1,1), rho, g, Ra, Rs, S2);
```

```
    k2 = f2(h1(i-1,1) + h*0.5*k1, h2(i-1,1) + h*0.5*k1, rho, g, Ra, Rs, S2);
    k3 = f2(h1(i-1,1) + h*0.5*k2, h2(i-1,1) + h*0.5*k2, rho, g, Ra, Rs, S2);
    k4 = f2(h1(i-1,1) + h*k3, h2(i-1,1) + h*k3, rho, g, Ra, Rs, S2);
    h2(i,1) = h2(i-1,1) + (h/6)*(k1+2*k2+2*k3+k4);
end

plot2d([t,t],[h1,h2],[1 2]);
legends(["Reservatório 1", "Reservatório 2"],[1,2],4)
title("Nível dos reservatórios pelo método de Runge-Kutta", "fontsize", 4)
xlabel("Tempo [s]", "fontsize", 3)
ylabel("Nível do reservatório [m]", "fontsize", 3)
xgrid(1);
```

2. Resultados