

Lista B SCILAB Modelagem - PME3380

Wilson Siou Kan Chow, 10769938

27/08/2020

EXERCÍCIO 1

Foi feito um arquivo .sci e um .sce .

funcao1B.sci

```
clear

rho = 1000; // kg/m^3
g = 10; // m/s^2
R = 2*10^8; // Pa/(m^3/s^2)
Qe = 0.010247; // m^3/s
S = 10; // m^2

function [ydot] = funcao(y)
    ydot = (-sqrt(rho * g * y / R) + Qe) / S;
endfunction
```

exerc1B.sce

```
// tempos e estados iniciais. y1=EULER , y2=KUTTA
t(1)=0;
tf=10000;
y1(1)=1;
y2(1)=1;

// numero de passos
h=500;
n=round((tf-t(1))/h);

//rotina numerica

for i=1:n
    t(i+1)=t(i)+h;

    //EULER
    y1(i+1)=y1(i)+h* funcao(y1(i));

    //KUTTA
    k1=h*( (-sqrt(rho * g * y2(i) / R) + Qe) / S );
    k2=h*( (-sqrt(rho * g * (y2(i) + k1/2) / R) + Qe) / S );
    k3=h*( (-sqrt(rho * g * (y2(i) + k2/2) / R) + Qe) / S );
    k4=h*( (-sqrt(rho * g * (y2(i) + k3/2) / R) + Qe) / S );
```

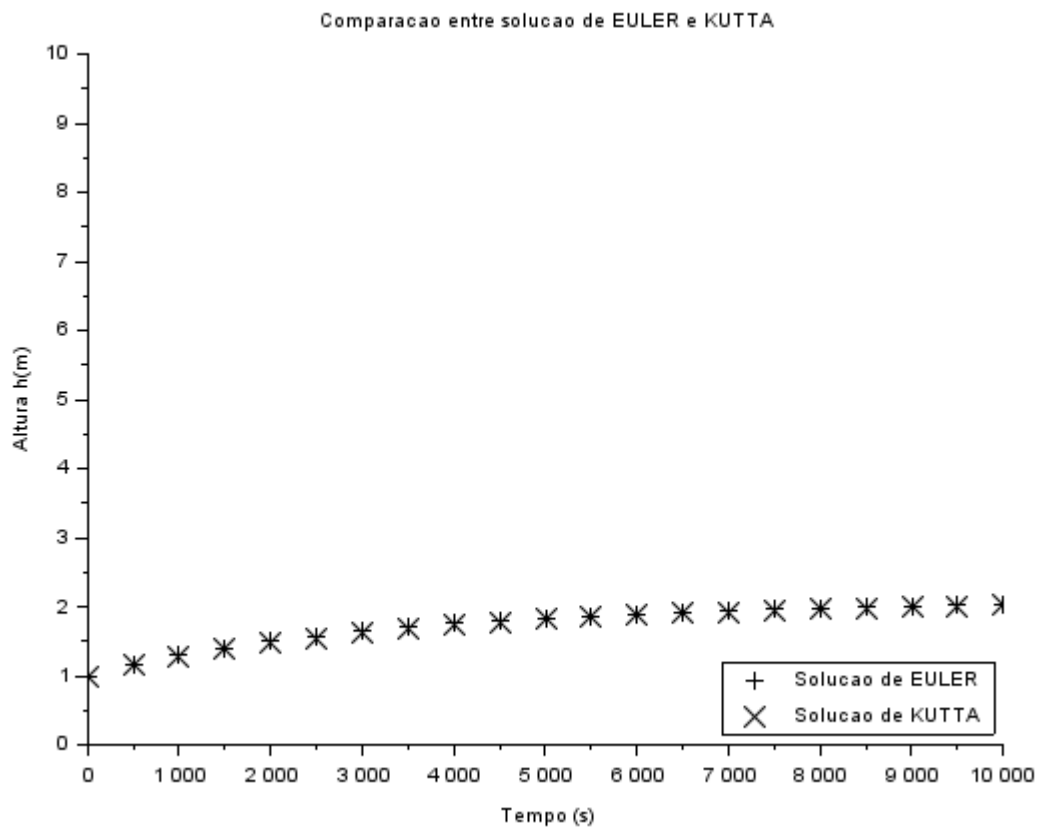
```
y2(i+1)=y2(i)+((k1+2*k2+2*k3+k4)/6);
```

```
end
```

```
plot2d([t,t],[y1,y2],[-1 -2]);  
legends(["Solucao de EULER","Solucao de KUTTA"],[-1 -2],4)  
xlabel("Comparacao entre solucao de EULER e KUTTA","Tempo (s)","Altura h(  
↪ m)");
```

```
//plot2d(t,y2,2);  
//legends(["Solucao numrica de Kutta"],[2],[1])  
//xlabel("Solucao numrica de Kutta")  
h=gca(); //get current axes  
h.data_bounds = [0, 0;tf , 10]
```

Com isso, foi gerado o seguinte plot:



EXERCÍCIO 2

Foram feitos dois arquivos .sce, um pra cada tipo de integração. Não usei .sci, pois as funções estavam dando problemas.

exerc2B Euler.sce

```
xdel(winsid())
clear

rho = 1000; // kg/m^3
g = 10; // m/s^2
Ra = 2*10^8; // Pa/(m^3/s^2)
Rs = 2*10^8; // Pa/(m^3/s^2)
Qe = 0.010247; // m^3/s
S1 = 10; // m^2
S2 = 5; // m^2

function [ydot] = funcao1(y1,y2)
    ydot = ( Qe - sqrt( rho * g *(y1 - y2) /Ra )) /S1
endfunction

function [ydot] = funcao2(y1,y2)
    ydot = ( sqrt( rho * g *(y1 - y2) /Ra ) - sqrt( rho * g *(y2) /Rs )) /
        ↪ S2
endfunction

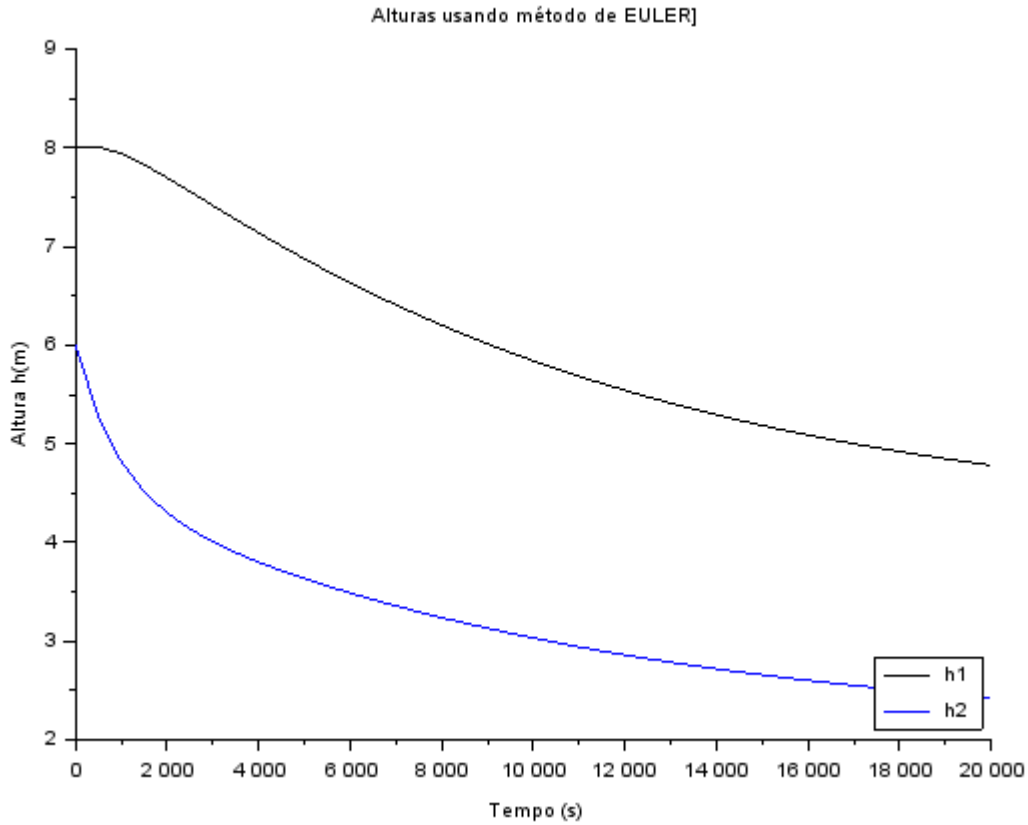
//tempos
t(1)=0;
tf=20000;
y1(1)=8;
y2(1)=6;

//calculo do numero de passos
h=500;
n=round((tf-t(1))/h);

//rotina numerica EULER
for i=1:n
    t(i+1)=t(i)+h;
    y1(i+1)=y1(i)+h* funcao1(y1(i),y2(i));
    y2(i+1)=y2(i)+h* funcao2(y1(i),y2(i));
end

plot2d([t,t],[y1,y2],[1 2]);
legends(["h1","h2"],[1 2],4)
xtitle("Alturas usando mtodo de EULER","Tempo (s)","Altura h(m)")
```

Isso produziu o seguinte plot:



————— exerc2B Kutta.sce —————

```

xdel(winsid())
clear

rho = 1000; // kg/m^3
g = 10; // m/s^2
Ra = 2*10^8; // Pa/(m^3/s^2)
Rs = 2*10^8; // Pa/(m^3/s^2)
Qe = 0.010247; // m^3/s
S1 = 10; // m^2
S2 = 5; // m^2

//tempos
t(1)=0;
tf=20000;
y1(1)=8;
y2(1)=6;

//calculo do numero de passos
h=500;
n=round((tf-t(1))/h);

//rotina numerica EULER
for i=1:n

```

```

t(i+1)=t(i)+h;

k1=h*( (Qe - (sqrt(rho * g * (y1(i)-y2(i)) / Ra) ) )/S1 )
k2=h*( (Qe - (sqrt(rho * g * ((y1(i)-y2(i)) + k1/2) / Ra) ) )/S1 )
k3=h*( (Qe - (sqrt(rho * g * ((y1(i)-y2(i)) + k2/2) / Ra) ) )/S1 )
k4=h*( (Qe - (sqrt(rho * g * ((y1(i)-y2(i)) + k3/2) / Ra) ) )/S1 )

y1(i+1)=y1(i)+((k1+2*k2+2*k3+k4)/6);

k1=h*( ((sqrt(rho * g * (y1(i)-y2(i)) / Ra) - sqrt( rho * g *(y2(i)))/
↪ Rs ) ) )/S2 )
k2=h*( ((sqrt(rho * g * ((y1(i)-y2(i)) + k1/2) / Ra) - sqrt( rho * g
↪ *(y2(i) + k1/2)/Rs ) ) )/S2 )
k3=h*( ((sqrt(rho * g * ((y1(i)-y2(i)) + k2/2) / Ra) - sqrt( rho * g
↪ *(y2(i) + k2/2)/Rs ) ) )/S2 )
k4=h*( ((sqrt(rho * g * ((y1(i)-y2(i)) + k3/2) / Ra) - sqrt( rho * g
↪ *(y2(i) + k3/2)/Rs ) ) )/S2 )

y2(i+1)=y2(i)+((k1+2*k2+2*k3+k4)/6);

end

plot2d([t,t],[y1,y2],[1 2]);
legends(["h1","h2"],[1 2],4)
xtitle("Alturas usando mtodo de KUTTA","Tempo (s)","Altura h(m)")

```

Isso gerou o seguinte plot:

Alturas usando método de KUTTA]

