

ALTERANDO O PASSO DE INTEGRAÇÃO

Na resolução de uma equação diferencial pode ser necessário guardar valores complementares, e não apenas resolver a EDO. Por exemplo, na resolução de:

$$\frac{dh(t)}{dt} = \frac{1}{A} \left(F_E - \frac{h}{R} \right)$$

Quero obter também o comportamento de:

$$Y = F_E + A + dh$$

Como ir calculando e armazenando esses valores ao longo do processo de resolução da equação diferencial?

O arquivo principal ganha três novas linhas, apresentadas em negrito abaixo. O comando *global* serve para fazer a passagem das variáveis novas necessárias entre o arquivo principal e a função “dhdt”. Os comandos “figure” e “plot”, por sua vez, são responsáveis por exibir o comportamento da variável Y ao longo do tempo.

```
% Definição das constantes do modelo  
clear all
```

```
global Y tvetor
```

```
R = 1;    % h/m2  
A = 2;    % m2  
Fe = 10; % m3/h  
% Tempo de simulação  
t = 0.0 : 0.01 : 10.0; % h  
% Simulação da altura de líquido  
[t,h] = ode45('dhdt',t, 0,[],[R A Fe]);  
% Visualização da simulação  
plot(t,h);  
title('Simulação do tanque de nível');  
xlabel('Tempo (h)');  
ylabel('Altura (m)');
```

```
figure(2)  
plot(tvetor,Y,'*b')
```

A função “dhdt” também sofrerá pequenas adaptações:

```
function dh = dhdt(t,h,flag,par)
global Y tvetor
R = par(1);
A = par(2);
Fe = par(3);
dh = (Fe-(h/R))/A;
Ylinha=Fe+A*dh;
Y=[Y Ylinha];
tvetor=[tvetor t];
```

Do mesmo modo que no arquivo principal, o “global” aqui serve para estabelecer a passagem das variáveis “Y” e “tvetor” de um arquivo para o outro. A variável “Y” será então calculada através da linha: “Ylinha=Fe+A*dh”.

Uma vez calculado o valor de Ylinha num determinado instante de tempo, é necessário armazená-lo, pois queremos obter Y para todos os instantes de tempo. O armazenamento é realizado através do comando: Y=[Y Ylinha]; Esse comando funciona do seguinte modo:

Inicialmente Y é vazio:

```
Y=[];
```

Quando Ylinha=1.34233 temos:

```
Y=[ Y Ylinha ]
Y=[ [ ] 1.342333 ]
Y=[ 1.342333 ]
```

Na passagem seguinte Ylinha=6.23 temos:

```
Y=[ Y Ylinha ]
Y=[ 1.342333 Ylinha ]
Y=[ 1.342333 6.23 ]
```

Na próxima rodada, Ylinha=9.17 temos:

```
Y=[ Y Ylinha ]
Y=[ 1.342333 6.23 Ylinha ]
Y=[ 1.342333 6.23 9.17 ]
```

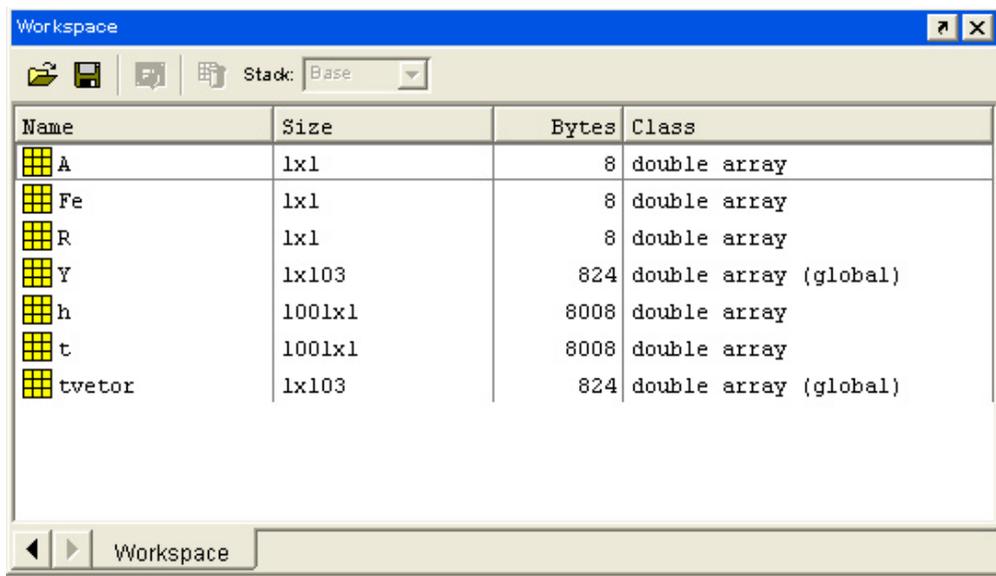
Observe que a cada passagem estamos armazenando o valor de Ylinha na matriz Y, que vai crescendo. É exatamente isso que foi feito na *function* “dhdt”.

A última linha da *function* armazena os valores de tempo (t) para os quais foi calculado cada valor de Y.

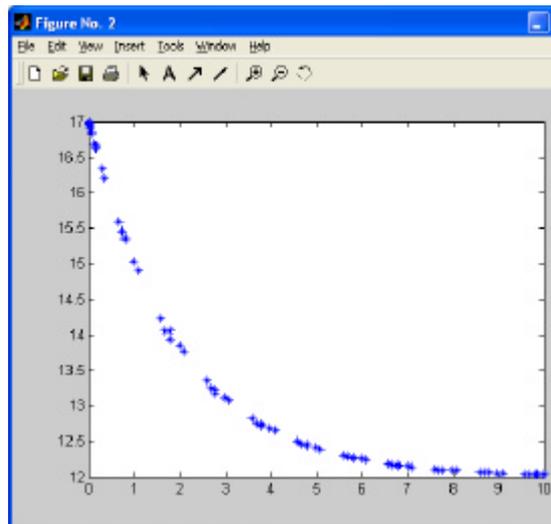
O resultado final com todas as variáveis é exibido abaixo. Observe que “Y” e “tveter” tem, cada uma, 103 elementos. Ou seja, foram calculados Y’s para 103 diferentes instantes de tempo. Porém, a resolução da equação diferencial promoveu um vetor “h” (variável dependente) e um vetor “t” (variável independente) com 1001 elementos. Lembre-se que a matriz “t” foi criada no programa principal pelo comando:

```
t = 0.0 : 0.01 : 10.0;
```

E esse comando gera uma matriz realmente de 1001 elementos. Logo, faz todo o sentido que variável independente (t) e dependente (h) tenham 1001 elementos.



Mas porque “Y” tem só 103? Porque apesar de precisar calcular “h” entre 0 e 10 com 1001 pontos, a rotina da ODE45 somente precisou chamar a *function* 103 vezes, pois a equação diferencial é simples de ser resolvida. Observe na figura abaixo o comportamento de Y com o tempo, e os instantes nos quais a ODE45 realizou o cálculo.



Como resolver esse problema e obter uma linha mais “contínua” de valores de Y?

Para isso precisamos forçar a rotina ODE45 a chamar mais vezes a *function* “dhdt”. Ou seja, torna-se necessário reduzir o passo máximo de integração. Os espaços em branco na figura acima foram causados por passos de integração muito elevados, que apesar de não prejudicarem o cálculo da variável dependente “h”, prejudicam o cálculo de “Y”. O comando dá poderes ao usuário de regular o passo de integração:

```
options = odeset('MaxStep',0.01);
[t,h] = ode45('dhdt',t, 0,options,[R A Fe]);
```

O passo de integração máximo ser de 0.01 melhorou bastante a aparência do gráfico de “Y”, mas também deixou o processo de cálculo bem mais lento. Agora, a rotina “dhdt” está sendo chamada 6025 vezes, contra apenas 103 do caso anterior.

