Física Computacional

2020/2021

Universidade de Aveiro

Departamento de Física

Trabalho Prático 1

Método de Euler - Revisões

Parte I

Problema 1.1 <u>Movimento a uma dimensão</u>

A. <u>Queda de uma pedra</u> *(aceleração constante)*

Uma pedra é largada, com velocidade inicial nula, de um segundo andar de um edifício. Pretende-se conhecer a velocidade **v**, e a posição **z**, da pedra em função do tempo t, desde o momento em que é largada até atingir o solo. A pedra está unicamente sujeita ao seu pêso.

Como fazer? E de onde partir?

<u>Precisamos de relembrar alguns conhecimentos</u>!

► Como sabemos da Mecânica, parte-se da equação do movimento, (2ª Lei de Newton) , e integra-se:

A 1ª integração, da aceleração em ordem ao tempo, dá-nos a velocidade;

A 2ª integração, da velocidade em ordem ao tempo, dá-nos a posição.

<u>No nosso caso, a integração vai ser feita *numericamente* pelo método de Euler.</u>

► Por outro lado, sabemos do Cálculo que uma equação diferencial de ordem *n* pode ser convertida num sistema de *n* equações diferenciais de 1ª ordem.

Considere que a pedra tem massa *m* igual a 150 g, e que a aceleração da gravidade **g** é igual a 9.8 (m/s^2). Considere também que um andar tem uma altura típica de 3 m.

a) Escreva a equação diferencial que resulta da aplicação da segunda lei de Newton ao problema.

b) Escreva o algoritmo de Euler.
(Recorde que este algoritmo permite obter a solução numérica para uma equação diferencial do tipo $\frac{dz}{dt} = f(t,z)$. Pode aplicá-lo diretamente à equação diferencial que obteve em a)? Porquê?

c) Se necessário modifique a equação de modo a poder aplicar o algorítmo de Euler.

d) Vamos começar por usar o método de Euler para se obter uma estimativa numérica da velocidade, entre o instante inicial $t_0 = 0$ s e um tempo final $t_f = 1.5$ s, com passo $h = 0.2$ s. Considere que o sentido positivo do eixo vertical (o eixo dos ZZ) aponta para cima.

Para a construção do código é necessário:

► Aplicar o algoritmo de Euler à equação de evolução para a velocidade (i.e., é preciso discretizar a equação!!!), para obter a equação de recorrência.

► Esta equação de recorrência permite obter a velocidade num instante *k+1* à custa da velocidade e da aceleração no instante *k* *(no nosso caso a aceleração é constante, mas pode não o ser!).* Pelo que, para obter a velocidade no intervalo de tempo requerido, é necessário:

- definir um incremento temporal, $h = t_{k+1} - t_k$ (o passo);
- criar um vetor para a variável independente t, $t = t_0{:}h{:}t_f$
- criar um vetor de zeros para a variável depente v, com a mesma dimensão do vetor t (vai obter a solução numérica nos instantes t_k):

```
N = length(t); v=zeros(1,N);
 %É necessário definir a velocidade inicial depois de definir o vetor v. E porque não antes de definir v?
   v(1) = V0;
```

• criar um ciclo que vai calculando v(k+1) à custa de v(k) e a(k) (no caso presente a(z) é constante).
Pode ser um ciclo ***for*** ou ***while***, por exemplo:

```
for k=1:N
   v(k+1) = v(k) + ….    ;
end
```

- pode representar o resultado com **plot(t,v).**

- O código corre? Não. Porquê?
- Para correr tem que definir <u>previamente</u> todas as constantes!
  ($h=$ … ; t_0 = … ; t_f = …; V0 =… ; m = 0.150 e g = 9.8;)

- No topo do código deve incluir <u>**sempre**</u> as seguintes intruções:

```
clc;
clear all;
close all;
```

e) Compare o resultado com a solução analítica, sabendo que a equação diferencial para a velocidade admite a seguinte solução:

$$vz(t) = v0 - gt$$

f) Use o método de Euler e a forma dinâmica para as ODE e represente graficamente a estimativa numérica da <u>altura</u> instantânea da pedra em função do tempo.

Obtenha uma estimativa numérica do instante em que a pedra cai no chão e da sua velocidade.

<u>Para a construção de um novo código a partir do código obtido em d) é preciso ter em conta que:</u>

▶ O ciclo for deve ser interrompido logo que *z* toma valores negativos. Não faz sentido do ponto de vista físico ter **z<0**!
Como fazer? Pode incluir um **break** no programa sempre que **z<0** !

▶ Depois de inserir o **break** no programa, analise os novos vetores **v** e **z** no editor do Matlab. O que observa?
Deve verificar que os vetores, a partir de um dado índice, só contêm zeros. Estes zeros não têm qualquer significado físico. Têm que se descartar!!!!!
Uma forma de os eliminar é a seguinte:
Identificar o índice k para o qual isso acontece, e redefinir o os vetores novamente, v=v(1:k); etc… t=t(1:k)….

▶ Para obter a estimativa numérica do instante em que a pedra cai no chão, deve usar a função **interp1** do MATLAB, para z = 0.

g) Compare o resultado com a solução analítica, sabendo que a equação diferencial para a posição admite a seguinte solução:

$$z(t) = z0 - \frac{1}{2}gt^2$$

O que observa? O comportamento é idêntico ao observado em **e)?** Interprete os resultados.

B. <u>Oscilador harmónico simples</u> (aceleração variável)

Um <u>sistema massa-mola</u> é posto a oscilar, tendo sido aplicada uma perturbação de pequena amplitude para tal. Pretende-se estudar o movimento e conhecer a posição $\boldsymbol{x}$ e a velocidade $\boldsymbol{vx}$, da massa em função do tempo.
Uma das extremidades da mola está presa a uma parede, e a outra extremidade tem acoplada uma massa pontual *m*. A massa está apenas sujeita à força restauradora do equilíbrio devido á mola, $F= -Kx$, sendo K a contante da mola, e x o deslocamento da massa em relação à posição de equilíbrio.
Considere K = 2.5 N/m, m = 0.5 kg, e um deslocamento inicial $x0$= 10 cm. Desprezou-se o atrito.

a) Escreva a equação diferencial que resulta da aplicação da segunda lei de Newton ao problema.

b) Aplique o algoritmo de Euler à equação obtida. Realize as transformações que achar necessárias para o poder fazer. (Use o mesmo procedimento que em A. c)).

c) Use o método de Euler para integrar numericamente o sistema de equaçoes obtido, e represente graficamente a posição $\boldsymbol{x}$ e a velocidade $\boldsymbol{vx}$ da massa, em função do tempo. *(Repare que a aceleração é variável. Tem que ser calculada em cada iteração!).* Considere o instante inicial t_0 = 0 s e um tempo final t_f = 10 s, com passo h= *0.1 s.*

d) Compare o resultado com a solução analítica, sabendo que o sistema de equações diferenciais admite a seguinte solução:

$$\boldsymbol{x}(t) = x0 \cdot \cos(\omega t) \quad e \quad \boldsymbol{vx}(t) = -x0 \cdot \omega \cdot sen(\omega t)$$

Sendo ω a frequência angular do movimento (T= 2π/ω), e *x0* o deslocamento inicial da massa. Considerou-se que a fase inicial era nula.

(Sugestão: represente graficamente as soluções analíticas e as númericas para ***x(t)*** e para a velocidade ***vx(t)***). O que pode concluir?

Repita os cálculos considerando *h= 0.01s*. O que observa?

Discuta o impacto da variação do valor de *h* na solução, em particular na amplitude e no período da mesma.

e) Sabe-se que para o método de Euler, o erro global, ie, o módulo da diferença entre a solução analítica e a numérica no final da simulação é directamente proporcional a *h:*

$$\boldsymbol{Erro\ Global = constante * h}$$

Vamos verificar este resultado!

Para tal é necessário calcular o erro global para vários valores de h.

No nosso caso, vamos considerar como *erro global* o valor absoluto da diferença entre a amplitude esperada e a amplitude máxima observada durante o tempo total da simulação,

erro global = | x0-Amplitude máxima |

Vamos considerar valores de *h entre* 10^{-1s} *e* 10^{-4s} . Note que a sua escolha deve ser feita de tal forma que 10 s seja um múltiplo inteiro de todos os valores de *h*. Faça um gráfico do logaritmo do módulo do erro em função do logaritmo de *h*.

(*Note que: os valores de h considerados têm diferentes ordens de grandeza. E quando se lida com diferentes ordens de grandeza é habitual logaritmizar a equação.*

►*Linearize a equação obtida, e identifique o valor do declive esperado.)*

Para verificarmos a validade deste resultado é necessário construir um novo código, a partir do anterior.

Para a construção de um novo código a partir do anterior é preciso ter em conta que:

- Para cada valor de h, e para o mesmo tempo final de simulação t_f , a dimensão dos vetores muda.

- Para cada valor de h obtém-se um valor para o Erro Gobal.
- Há que correr o programa para vários valores de h, e calcular o respectivo Erro Global para cada caso.

▶ É necessário construir um vector com vários valores de h.
▶ É necessário construir um vector Erro Global, com a mesma dimensão do vetor dos valores de h.

```
Hhs=[10^-1 ...  ....  .... 10^-4]
Nh=length(Hhs);
```

```
for j=1:Nh  %este ciclo permite variar os valores de h…
        h= Hhs(j);
          % OLD

                t = t0:h:tf

                N = length(t); vx=zeros(1,N);
               vx(1) = V0; x=zeros(1,N); x(1)=1;

                for k=1:N
                   a(k) = --- ;
                   vx(k+1) = v(k) + ….   ;
                   x(k+1) = x(k) + vx(k)*h;
                end
     …….
          Xmax= max(x);
     ERROGLOBAL(j) = abs(x0-Xmax);

End

ETC…  plot(ln(Hhs), ln(ERROGLOBAL))
```

Pode usar o comando `lsline` do MATLAB para traçar uma recta sobre a curva que obteve.

Use a função `polyfit` do MATLAB para determinar o valor aproximado do expoente. (Qual é o valor esperado)?