Física Computacional

2022/2023

Universidade de Aveiro

Departamento de Física

Trabalho Prático 1

Métodos de Euler e Euler-Cromer - Revisões

Problema 1.1 **Queda de uma pedra** (movimento a 1 Dimensão)

Uma pedra é largada 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 peso.
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 ainda que a pedra é largada com velocidade inicial nula. A altura típica de um andar é de 3 m.

Como fazer? E de onde partir?

Precisamos de relembrar alguns conhecimentos!

► 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.
No nosso caso, a integração vai ser feita *numericamente* pelo método de Euler.

► 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.

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

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

c) Aplique o algorítmo de Euler às duas equações diferenciais de 1ª ordem que resultam da equação de 2ª ordem obtida em a).

d) Obtenha uma estimativa numérica da velocidade, **v**, entre os instantes inicial, t_0 = 0 s, e final, t_f = 2 s. Use o método de Euler para integrar a equação do movimento, com passo *h= 0.2 s*. Considere que o sentido positivo do eixo vertical (o eixo dos ZZ) aponta para cima. Use um ciclo **For.**

Para a construção do código é necessário:

► Aplicar o algoritmo de Euler à equação de evolução para a velocidade, de modo a obter uma equação de recorrência. (*Discretização da equação*).

► Esta equação de recorrência permite obter a velocidade num instante *k+1* à custa da velocidade e da aceleração no instante *k*.

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(N,1);
v(1) = V0; %Pode ser definido antes de v? Porquê?
```

- criar um ciclo que vai calculando v(k+1) à custa de v(k) e a(k)

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

- represente 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) Repita para o ciclo **`while`.**

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

$$vz(t) = v_0 - gt$$

g) Use o método de Euler para integrar numericamente as duas equações de 1ª ordem obtidas em c). Represente graficamente a altura da pedra em função do tempo.

Obtenha uma estimativa numérica do instante em que a pedra cai no chão e da sua velocidade. Use o **`interp1`** do Matlab.

<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 ou while) 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 novamente os vetores,   v=v(1:k); etc… t=t(1:k)….

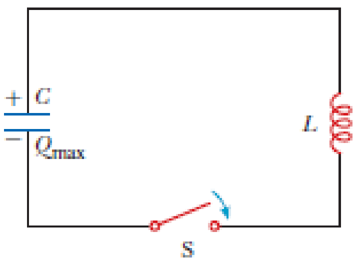
► Para obter a estimativa numérica do instante em que a pedra cai no chão, deve usar a função **interp1** do MATLAB, para z = 0.

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

$$z(t) = z_0 - \frac{1}{2} g t^2$$

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

Problema 1.2 <u>Oscilador harmónico simples</u> (circuito LC)



Considere o circuito da figura, contituido por um condensador e uma bobina (um indutor) em série. (Circuito LC). O condensador C encontra-se carregado com a carga Q_0, e com uma diferença de potencial aos seus terminais de Vc (t = 0) = 5 V.

No instante t = 0 s o interruptor S é fechado. O condensador começa a descarregar-se, oscilando a carga entre + Q_0 e - Q_0. Como consequência a tensão Vc, aos seus terminais, oscila também, entre um valor máximo de +5 V e um valor mínimo de -5V.

Considere que a indutância, ***L*** tem o valor de 0.25 *H* e a capacidade ***C*** é igual a $1 \times 10^{-3}\ F$.

a) Escreva a equação diferencial que resulta da aplicação das leis de Kirchhoff ao circuito.

<u>Recorde</u>: a variação do potencial aos terminais do condensador é dada por Vc = Q/C, e no indutor L é dada por V_L= L dI/dt. A corrente elétrica no circuito é definida como I = dQ/dt.

b) Mostre que a equação diferencial obtida em **a)** se pode escrever como:

$$\frac{d^2 V_c}{dt^2} = -a\, V_c$$

Sendo $a = 1/(LC)$.

c) Aplique o algoritmo de Euler à equação obtida em **b)**. Realize as transformações que achar necessárias para o poder fazer.

d) Use o método de Euler para integrar numericamente o sistema de equações obtido em c), para um tempo final t_f = 0.5s, e h= *0.001 s*. Represente graficamente a tensão *Vc*, a carga *Q*, e a corrente *I*, em função do tempo. Estão em fase?

e) Compare os resultados obtidos com as soluções analíticas, sabendo que o problema admite as seguintes soluções :

$$\boldsymbol{V_c}(t) = V_o \cdot cos(\omega t) \qquad \boldsymbol{Q}(t) = Q_o \cdot cos(\omega t)$$

$$\boldsymbol{I}(t) = -Q_0 \cdot \omega \cdot sen(\omega t) = -I_{max} \cdot sen(\omega t)$$

Sendo ω a frequência angular das oscilações, dada por $\sqrt{\frac{1}{LC}}$, (o período é T= 2π/ω). V_0, *e* Q_0 representam a tensão e a carga no condensador no instante inicial, respetivamente. I_{max} representa o valor máximo da corrente, e é dado por - ω Q_0. Considerou-se que a fase inicial era nula.

f) Repita os cálculos considerando *h= 0.0001s*. O que observa?

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

g) Determine o período T.
Para tal use a função **`islocalmax`** do Matlab. Esta função permite localizar máximos locais.
Obtenha o valor do período sem e com interpolação dos instantes em que ocorrem os máximos locais (**`interp1`**). Compare-os com o valor esperado.

h) Para o OHS sabe-se que a energia total se conserva. Porquê?
Verifique este resultado para o circuito LC.
A energia elétrica armazenada no condensador é dada por $U_c = ½\ Q^2/C$. A energia magnética armazenada na bobina é dada por $U_m = ½\ LI^2$. Pode mostrar-se que $U_c + U_m = ½\ Q_0^2/C$.

Problema 1.3 **<u>Oscilador harmónico simples</u>** (Euler-Cromer)

a) Repita as alíneas d) e h) do problema anterior mas agora usando o método de Euler-Cromer.

b) O método de Euler-Cromer pode ser aplicado a ODEs de qualquer ordem? Qual é a ordem deste método?

Problema 1.4 **<u>Oscilador harmónico simples</u>** (Erro Gobal do método de Euler)

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

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

<u>Vamos verificar este resultado!</u>

Como fazer?

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

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

$$\textbf{\textit{erro global}} = |\ V_0 \textbf{\textit{-Amplitude máxima}}\ |$$

Vamos considerar valores de *h entre* 10^{-3}*s* e 10^{-7}*s* .
Note que a sua escolha deve ser feita de tal forma que 10 s seja um múltiplo inteiro de todos os valores de *h*.
<u>Faça um gráfico do logaritmo do módulo do erro em função do logaritmo de *h*.</u>

<u>Note que</u>: os valores de h considerados têm diferentes ordens de grandeza. E quando se lida com diferentes ordens de grandeza é habitual logaritmizar a equação.

►*<u>Linearize a equação obtida, calcule o declive e compare-o com o valor esperado.</u>*

Para verificarmos a validade deste resultado é necessário construir um novo código, a partir do anterior (Problema 1.2 (d)) .

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 diversos 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); dVc=zeros(1,N);
                dVc(1) = 0; Vc=zeros(1,N); Vc(1)=V0;
                for k=1:N
                   dVc(k+1) = dVc(k) + …. *h  ;
                   Vc(k+1) = Vc(k) + dVc(k)*h;
                end
    …….
            Vmax= max(Vc);
     ERROGLOBAL(j) = abs(V0-Vmax);
End
ETC…  plot(log(Hhs), log(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)?