

Universitat Autònoma de Barcelona

Facultat de Ciències



PRÀCTICA 4

Autors:

Gerard Lahuerta & Ona Sánchez

1601350 — 1601181

7 de Juny del 2022

Índex

1	Presentació de la llibreria i informació d'interès	3
2	Presentació dels programes	4
2.1	Fitxer <i>RK4.c</i>	4
2.1.1	RK4	4
3	Control d'errors	5
3.1	<i>RK4.c</i> : RK4	5
3.2	<i>pendol.c</i>	5
4	Compilació i execució	6
4.1	Compilació del Programa	6
4.1.1	Makefile	6
4.1.2	Compilació manual	7
4.2	Execució del Programa	7
5	Comprovacions de funcionament	8
6	Conclusions	11
6.1	Informació adicional	11

1 Presentació de la llibreria i informació d'interès

La llibreria de funcions ha estat programada en llenguatge *C* i consta de dos fitxers, un fitxer capcelera i un amb el codi de la funció.

Aquest fitxer amb codi precisa de les llibreries `math.h` i `RK4.h`.

També, el fitxer capcelera té definida la constant:

- `E_STEP`: missatge d'error per valor del nombre de passos inferior o igual a 0.

La llibreria té programada únicament la funció `RK4()` que permet calcular numèricament el valor d'equacions diferencials mitjançant n passos introduïts amb un *Runge Kutta* d'ordre 4.

Recalcar a més, que la funció conté el seu *CDocs* propi que conté la informació més rellevant de la funció de manera resumida: *input*, *output*, *descripció* i *informació rellevant*.

2 Presentació dels programes

2.1 Fitxer *RK4.c*

2.1.1 RK4

- Entrada:
 - Funció de la variable x: `double*` *f*
 - Funció de la variable y: `double*` *g*
 - * La funció *f* i *g* són del tipus: `double(double, double, double, void*)`
 - Valor inicial de x: `double` *x0*
 - Valor inicial de y: `double` *y0*
 - Valor inicial del temps: `double` *t0*
 - Valor final del temps: `double` *tf*
 - Nombre de passos per fer el Runge Kutta: `int` *n*
 - Vector de paràmetres necessaris per la funció f: `void*` *prm f*
 - Vector de paràmetres necessaris per la funció g: `void*` *prm g*
- Sortida: No retorna cap valor, `void`.
- Funcionament: Calcula el resultat numèric d'un sistema EDOs de dues variables usant un *Runge Kutta* d'ordre 4. Per trobar els valors *x* i *y*:

$$x_{n+1} = x_n + \frac{h}{6} \cdot (k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4})$$

$$y_{n+1} = y_n + \frac{h}{6} \cdot (l_{n1} + 2l_{n2} + 2l_{n3} + l_{n4})$$

3 Control d'errors

3.1 *RK4.c : RK4*

Per tal d'assegurar el correcte funcionament la funció, es confirmarà la correcta introducció dels paràmetres.

Es comprovarà que:

1. El valor del paràmetre n és major que 0.

En cas d'inclompir aquesta condició el programa actuarà de la següent manera:

1. Es mostrarà per pantalla el missatge d'error corresponent, guardat a la variable `E_STEPS` (explicada anteriorment a la secció 1).
2. No modificarà les variables x i y i finalitzarà immediatament la funció.

3.2 *pendol.c*

Per tal d'assegurar el correcte funcionament del programa i que l'equació que es vol resoldre mantingui el sentit físic, es comprovarà que:

1. Els valors introduïts per paràmetre s'han guardat correctament a la variable corresponent.
2. El valor dels paràmetres m i L son majors que 0, i $\alpha \geq 0$.
3. El valor del paràmetre n és major que 0.

En cas d'inclompir alguna d'aquestes condicions el programa actuarà de la següent manera:

1. Es mostrarà per pantalla el missatge d'error corresponent i es retornarà un -1.
2. Es mostrarà per pantalla el missatge d'error corresponent i es retornarà un -2.
3. Es mostrarà per pantalla el missatge d'error corresponent i es retornarà un -3.

4 Compilació i execució

4.1 Compilació del Programa

4.1.1 Makefile

Per facilitar la compilació del programa s'ha creat un fitxer *makefile* que inclou tant les comandes per crear l'executable com altres comandes associades (*clean* i *clean_all* que explicarem més endavant) per la correcta gestió dels fitxers que s'obtenen de l'execució del makefile.

Per compilar el programa i generar l'executable (*pendol*) només cal escriure a la terminal on es troben tots els fitxers (inclòs el *makefile*) la comanda make:

```
$make
```

Per tal de facilitar la generació del retrat de fase demanat i així comprovar que funcioni correctament, s'ha afegit una opció al makefile per obtenir l'executable que genera un gràfic (el del retrat de fase explicat a 2), usant la comanda:

```
$make graphics
```

Com hem comentat anteriorment, el fitxer makefile també conté comandes per la correcta gestió dels fitxers resultants d'obtenir l'executable, aquestes comandes són:

```
$make clean
```

```
$make clean_all
```

La comanda *clean* elimina tots els fitxers *.o* del directori mentre que la comanda *clean_all* elimina tant l'executable com els fitxers amb terminació *.o*.

Mencionar que no es poden utilitzar les dues comandes de manera seguida ja que donarà error.

Important: en cas de no utilitzar el sistema operatiu *Linux* (o semblants) o *IOS* cal modificar la variable *DELETE* de l'arxiu *makefile* per a poder utilitzar-lo (substituir per la comanda *del* en cas d'utilitzar Windows).

4.1.2 Compilació manual

En cas de voler compilar el programa de manera manual (comanda a comanda) utilitzar en ordre les següents comandes al terminal (una vegada ubicat al directori on es troben els fitxers).

```
$gcc -c RK4.c
$gcc -c pendol.c RK4.c -lm
$gcc pendol.o RK4.o -lm -o pendol
```

En cas de voler usar l'executable que genera un gràfic, cal donar-li permisos d'execució, usant les següent comandes:

```
$chmod 744 gnu_graphic.gp
$chmod 744 generator_graphic.sh
```

4.2 Execució del Programa

Adjuntem a continuació les diferents comandes que hi oferim amb la descripció de cada una:

Comanda	Descripció
<code>./pendol α m L x₀ y₀ t₀ t_f n</code>	Genera n valors del sistema x, y entre t_0 i t_f usant un Runge Kutta d'ordre 4 amb un pas de 100.
<code>./generator_graphics.sh</code>	Mitjançant l'executable <i>pendol</i> i <i>gnu_graphics.gp</i> guarda una imatge anomenada <i>graphic.png</i> mostrada a 2

Qualsevol argument de més que s'introdueixi al programa serà omés.

5 Comprovacions de funcionament

S'han decidit representar els resultats obtinguts mitjançant el programa `gnuplot` amb les dades obtingudes canviant el valors dels paràmetres i deduint si el gràfic obtingut és esperable segons el sentit físic.

Mitjançant l'execució del programa `generador_graphics.sh` s'obté el retrat de fase d'un cos amb massa 1 i una llargada de corda de 1 experimentant diverses velocitats i acceleracions sense fregament.

El grafic obtingut és el següent:

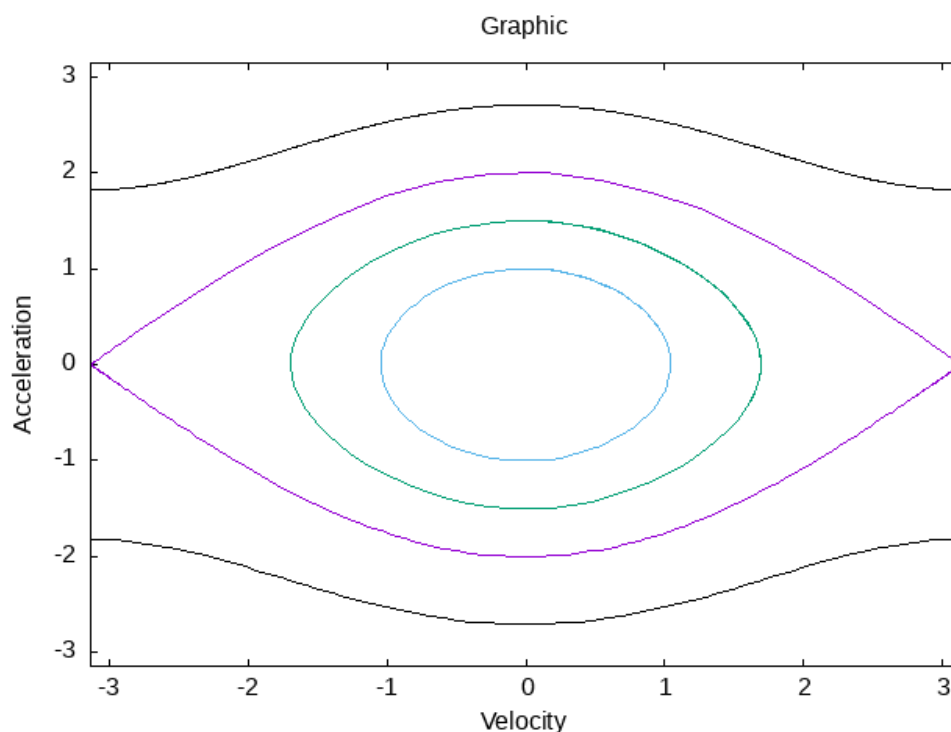


Figura 1: Retrat de Fase del sistema per als valors $(\alpha, m, L) = (0, 1, 1)$

Com s'observa, el gràfic té un sentit físic esperable ja que:

1. Si l'objecte comença amb una velocitat petita i no amb la suficient acceleració (es troba dins de la regió delimitada per les línies liles) l'objecte oscil·larà at infinitum. Això es reflexa fent que les corbes de nivell siguin tancades, no tendeixin a cap valor i reflexant que quan la velocitat és màxima (en mòdul) l'acceleració és 0 i que quan l'acceleració és màxima (en mòdul) la velocitat és 0.
2. Si l'objecte inicia amb la velocitat exacta i/o amb l'acceleració necessària com per trobar-se sobre la línia de color lila, l'objecte rota amb l'energia necessària com per a arribar a la part

més alta de la circumferència que traça amb velocitat 0. Aquest fenomen (similar al primer cas) es manifesta gràficament com una ona cosenoïdal.

3. Si a l'objecte se li dona una acceleració inicial suficient (es troba per fora de la regió delimitada per les línies liles) l'objecte donarà infinites voltes arribant a la part més alta de la trajectoria amb una velocitat i acceleració diferent de 0. Aquest fenomen es representa gràficament com una ona cosenoïdal similar al segon cas però desplaçada en l'eix d'ordenades reflexant que l'acceleració mai valdrà 0.

Per altre anda, si hi afegim una força de fregament, obtenim el següent gràfic:

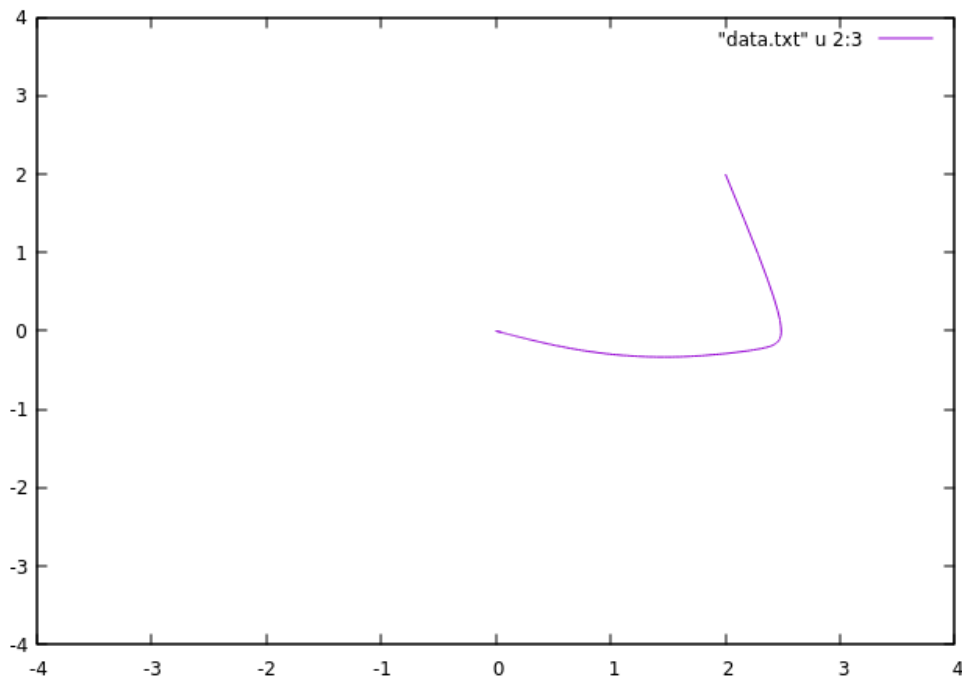


Figura 2: Retrat de Fase del sistema per als valors $(\alpha, m, L, x_0, y_0) = (3, 1, 1, 2, 2)$

S'observa d'aquí que tant l'acceleració com la velocitat tendeixen a anar cap a 0 (i de manera bastant ràpida) degut a que l'acceleració generada per la força de fregament és major que la que hi tenim inicialment.

Per altra banda, cal mencionar que si la introducció dels paràmetres no es fa amb cura i s'introdueixen valors de temps massa grans sense reajustar el valor d'interval a calcular el programa pot no donar resultats correctes; per exemple en aquest cas:

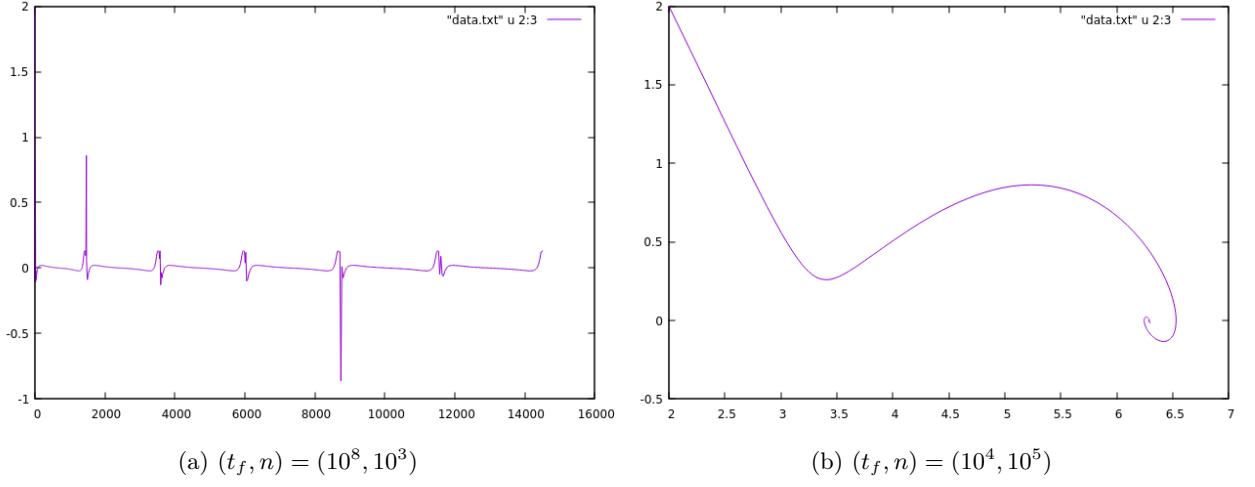


Figura 3: Gràfics generats per al sistema $(\alpha, m, L, x_0, y_0, t_0) = (2, 1, 1, 2, 2, 0)$

6 Conclusions

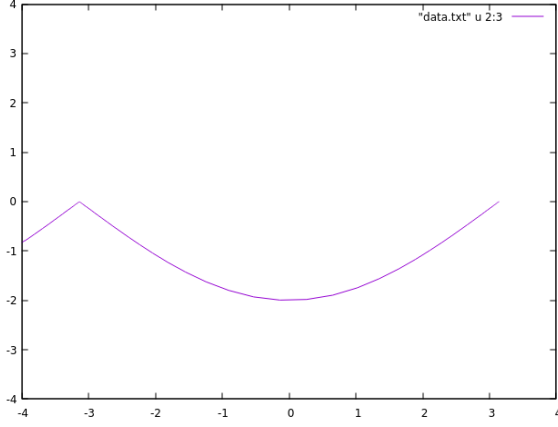
Finalitzem l'informe confirmant (a partir de les comprovacions de funcionament dutes a terme a l'apartat 5) que el programa *pendol.c* funciona de manera correcta, proporcionant els valors x i y al llarg d'un cert període de temps, resultants d'un sistema EDOs de dues variables que es resol de forma numèrica usant el mètode Runge Kutta d'ordre 4.

També, arrel de les comprovacions dutes a terme a l'apartat mencionat anteriorment, confirmem que el programa *generator_graphics.sh* funciona correctament, proporcionant l'exemple de retrat de fase ja explicat.

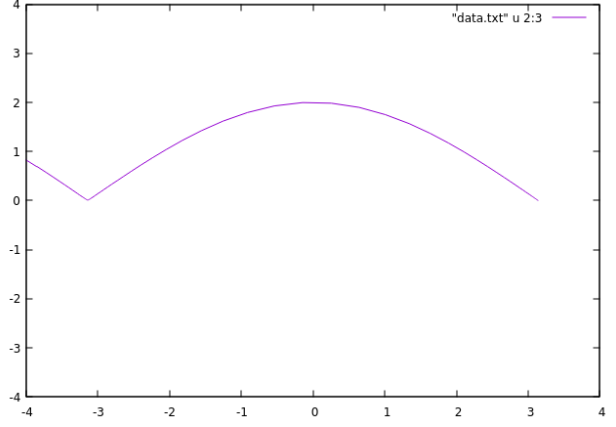
6.1 Informació adicional

- La complexitat del mètode és d'ordre lineal; és a dir, $O(n)$ (on n és el nombre de intervals que volem que utilitzi programa).
- Responem a la pregunta: **Quin paper juga el paràmetre α ?**
A l'equació del pèndol esmorteït $x'' + \alpha x' + \frac{m}{L} \sin(x) = 0$, el paràmetre α representa la força de fricció.
- Responem a la pregunta: **Si $\alpha = 0$, $m = 1$ i $L = 1$, intenteu obtenir el retrat de fase (gràfic en x i y) del pèndol.**
S'ha respòs aquesta pregunta a l'apartat 5.

- Responem a la pregunta: **Pel cas $\alpha = 0$, $m = 1$ i $L = 1$, els punts $((2k+1)\pi, 0)$, $k \in \mathbb{Z}$ són selles, i per tant inestables. Què passa si integreu el sistema amb condició inicial en un d'aquests punts amb t_f diferent?**

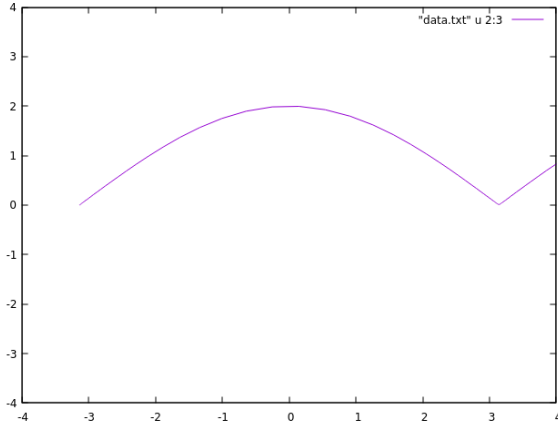


(a) $t_f = 200$

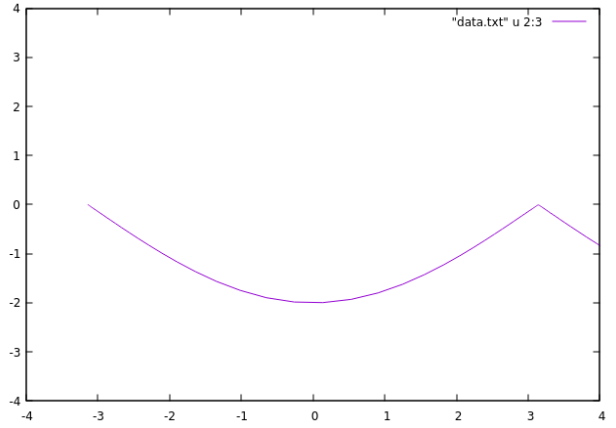


(b) $t_f = -200$

$$(\alpha, m, L, x_0, y_0, t_0, n) = (0, 1, 1, 3.14, 0, 0, 1000)$$

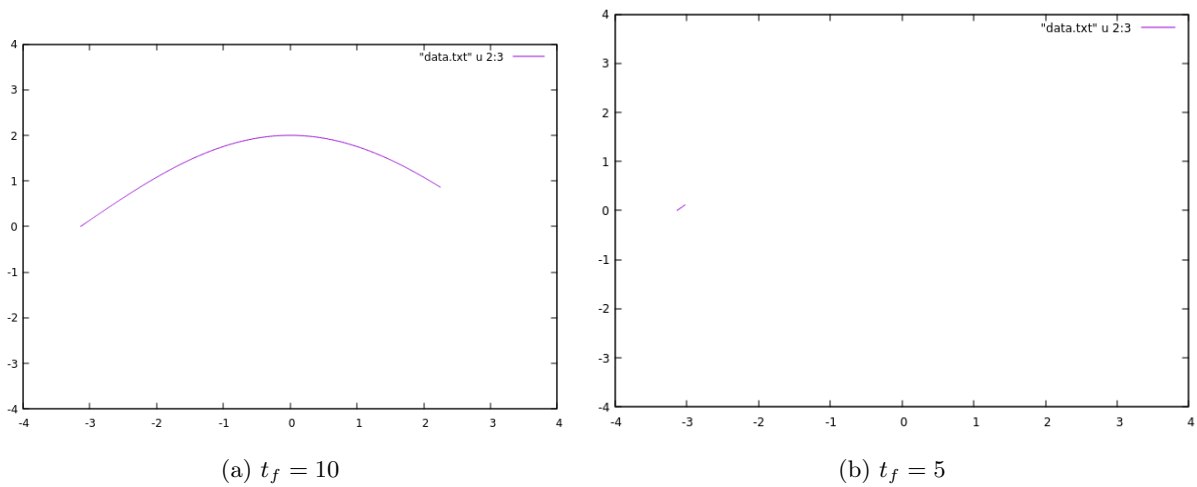


(a) $t_f = 200$



(b) $t_f = -200$

$$(\alpha, m, L, x_0, y_0, t_0, n) = (0, 1, 1, -3.14, 0, 0, 1000)$$



$$(\alpha, m, L, x_0, y_0, t_0, n) = (0, 1, 1, -3.14, 0, 0, 1000)$$

Observem comparant el recorregut dels últims dos gràfics amb els anteriors que, canviant el valor del temps el recorregut lineal canvia radicalment, en comparació amb els 4 gràfics anteriors.

Es dedueix que hi existeix un període en que la velocitat no varia molt davant el temps transcorrit i una altra on hi augmenta molt (procés al que arriba a la velocitat màxima).

Aquest fenomen correspon al fet que l'acceleració gravitatoria no accelera de la mateixa manera el pendòl en posicions diferents, només accelera una de les dues components del sistema, pel que l'acceleració que hi pateix l'objecte a l'hora d'oscil·lar no varia de forma lineal sino seguint un sinus.

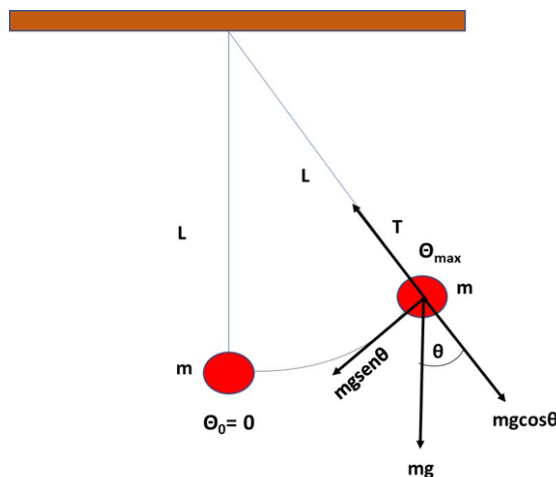


Figura 4: Diagrama de Sòlid Lliure d'un pendòl simple