Pràctica 4: Resolució Numèrica d'EDOs

Júlia Aguilera Ayuso - NIU: 1533697

Guillermo Raya Garcia - NIU: 1568864

7 de juny de 2022

1 Introducció

L'objectiu d'aquesta pràctica és resoldre amb el mètode Runge-Kutta d'ordre 4 l'equació del pèndol esmorteït, que és la següent:

$$x'' + \alpha x' + \frac{m}{L}\sin x = 0$$

Aquesta EDO no té solució analítica i tampoc representa un sisteme Hamiltonià, per tant, és molt important l'ús de mètodes numètics per resoldre-la.

Veiem que aquest problema es pot tractar com un problema 2-dimensional, és a dir, es pot reescriure com un sistema de dues equacions diferencials de primer ordre. Direm que tenim dues variables; x que serà la posició, i y que serà la velocitat.

$$\begin{aligned} x &: \text{ posició} \\ x' = y &: \text{ velocitat} \\ x'' = y' &: \text{ acceleració} \end{aligned}$$

De manera que aïllant x'' de l'equació del pèndol tenim:

$$x'' = -\alpha x' - \frac{m}{L}\sin x$$

Considerant les definicions anteriors, substituïnt tenim:

$$y' = -\alpha y - \frac{m}{L}\sin x$$

El que ens porta al següent sistema d'equacions:

$$\begin{cases} x' &= y \\ y' &= -\alpha y - \frac{m}{L}\sin x \end{cases}$$

Que correspon a les següents funcions:

$$\begin{cases} \dot{x} = f_1(t; x, y) &= y \\ \dot{y} = f_2(t; x, y) &= -\alpha y - \frac{m}{L}\sin x \end{cases}$$

2 Manual de software

2.1 Codi

En aquesta pràctica la part de codi consta de dos fitxers.c i un fitxer.h

2.1.1 `RK4`

Per programar el mètode Runge-Kutta 4, hem creat una arxiu capçalera RK4.h on es troba el prototip de la funció RK4. Aquesta ha estat programada dins de l'arxiu RK4.c, i segueix la següent estructura:

- **Entrada**:
  - `(double (*f1)(double, double, double, void*)`: Funció f1, primera equació del nostre sistema.
  - `double (*f2)(double, double, double, void*)`: Funció f2, segona equació del nostre sistema.
  - `double *x`: Punter on guardarem l'estimació numèrica del valor de x per a $t = t_f$.
  - `double *y`: Punter on guardarem l'estimació numèrica del valor de y per a $t = t_f$.
  - `double x0`: Variable que conté la condició inicial per x amb el que començar el mètode.
  - `double y0`: Variable que conté la condició inicial per y amb el que començar el mètode.
  - `double t0`: Temps inicial t_0.
  - `double tf`: Temps final t_f.
  - `double n`: Número de passos amb que s'aplica RK4.
  - `double *prm`: Paràmetres de la funció `f1`.
  - `double *prm2`: Paràmetres de la funció `f2`.
- **Funcionament**: Aplica el mètode Runge-Kutta 4 amb un pas de $h = \frac{t_f - t_0}{n}$.
- **Sortida**: No en té perquè es una funció de tipus `void`. La sortida es va guardant a les direccions dels punters `x` i `y`.

2.1.2 `pendol.c`

Aquest arxiu és el main del nostre programa, té dins definides $f_1(t; x, y)$ i $f_2(t; x, y)$. El programa resol el problema de dues equacions diferencials de primer ordre establert per

$$\begin{cases} \frac{dx}{dt} = f_1(t; x, y) \\ \frac{dy}{dt} = f_2(t; x, y) \end{cases}$$

tenint en compte les condicions inicials que li donem en cridar-lo a la consola de linux.

2.2 Utilitats

En aquest apartat explicarem com compilar i executar el programa que utilitza la biblioteca que hem creat.

2.2.1 Generant l'executable

Per tal de generar l'executable vam crear un fitxer Makefile per facilitar la compilació del programa. Per tant, el que hem de fer per generar l'executable és:

Un cop estem a la carpeta dels arxius del programa, escriure la comanda `make all` des de la terminal, de manera que el Makefile generi l'executable i els objectes necessaris de cada arxiu. En cas de voler esborrar tots els arxius generats, podeu fer ús de la comanda `make clean`.

2.2.2 Ús de les utilitats

Per fer us del programa `pendol.c`, hem d'executar la següent comanda:

`./pendol` α `m L x0 y0 t0 tf n`

Aquest programa ens retornarà la solució numèrica de l'equació diferencial de segon ordre del pèndol esmorteït.

3 Resultats i anàlisi

3.1 Quin paper juga el paràmetre α ?

El paràmetre α és el coeficient de fregament, el que fa és frenar el pèndol fins al repos després d'un temps t, per tant, aquest paràmetre provoca variacions importants en el moviment del pèndol.

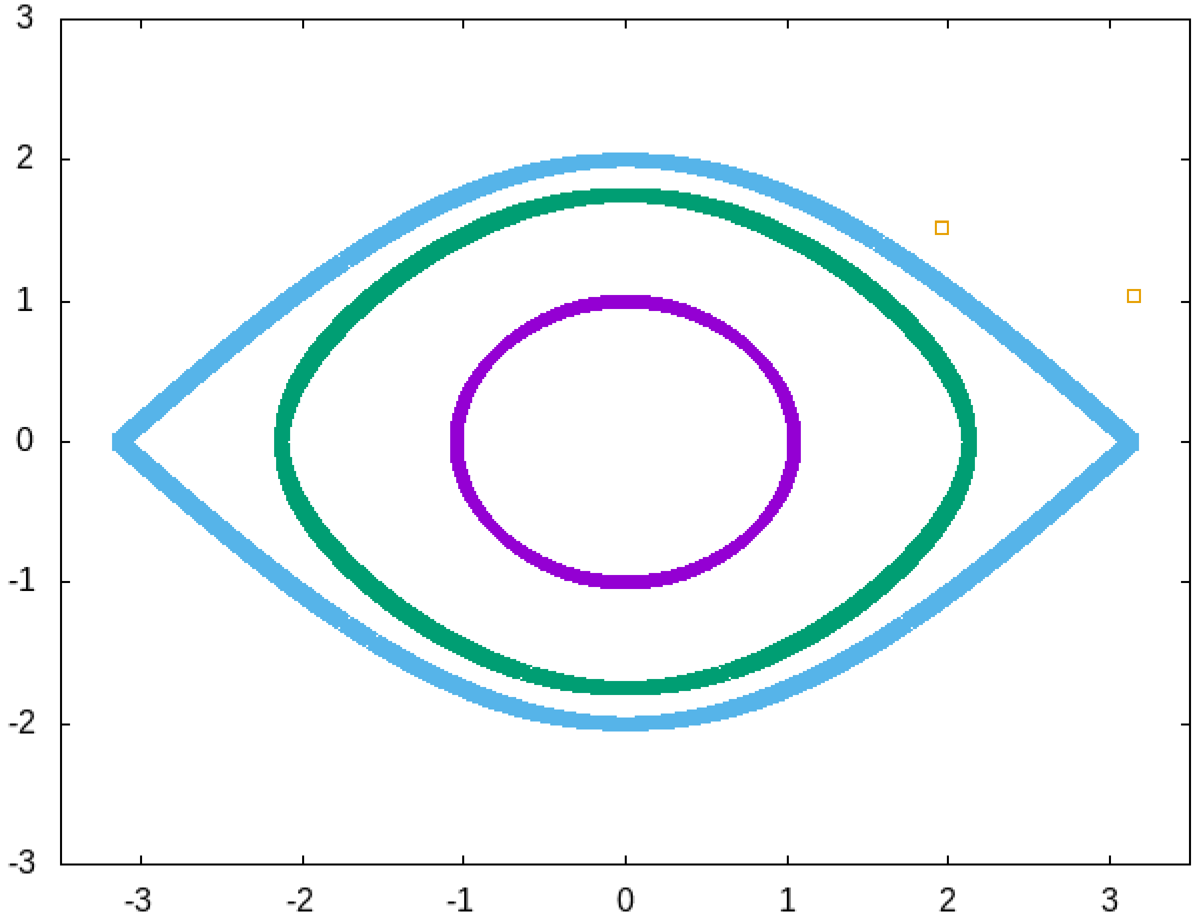
3.2 Retrat de fase del pèndol amb $\alpha = 0$, $m = 1$ i $L = 1$:

Per tal de crear retrat de fase del pèndol desitjat, hem creat un script de `bash` que segueix els següents passos:

1. Compila el programa `pendol.c` en un executable `pendol` utilitzant la comanda `make all`.
2. Executa el programa `pendol` diverses vegades amb els arguments $\alpha = 0$, $m = 1$ i $L = 1$ seguits de diferents valors per a les condicions inicials:
   - $(x, y) = (0, 1)$
   - $(x, y) = (0, 1.75)$
   - $(x, y) = (0, 2)$
   - $(x, y) = (0, 2.25)$
3. Guarda els resultats per als diferents valors inicials dins diversos fitxers de texte `.txt`.
4. Executa gnuplot i crea la gràfica dels diferents fitxers junts.
5. Guarda la gràfica en un fitxer anomenat `plot.png`.
6. Esborra els fitxers de texte auxiliars i l'executable.

Per a cridar-lo, hi ha prou amb entrar a la carpeta de l'entrega, obrir una terminal de linux i executar la comanda `./plotGraph.sh`. En cas de no funcionar per falta de permís, podeu fer servir `chmod 744 ./plotGraph.sh`.

El fitxer `plot.png` hauria de contenir una imatge com aquesta:



3.3 Pel cas $\alpha = 0$, $m = 1$ i $L = 1$, els punts $((2k+1)\pi, 0)$, $k \in Z$ són selles, i per tant inestables. Què passa si integreu el sistema amb condició inicial en un d'aquests punts amb amb tf diferent ?

Per aquests casos a mesura que augmentem el temps el que veiem és que cada cop oscil·lem més aprop del punt de sella, segurament a partir d'un valor concret de tf, no aconseguim arribar a sortir del punt.

4 Conclusions

En aquesta pràctica hem aconseguit programar amb èxit el mètode de Runge-Kutta4 per a sistemes de dues equacions diferencials, i, tot i que hem tingut alguns problemes (no hem aconseguit fer un plot com el que sortia a l'enunciat, i no hem acabat d'entendre bé el comportament del sistema en alguns punts), pensem que hem assolit els objectius essencials de la pràctica.