



UNIVERSITAT DE  
BARCELONA

Facultat de Matemàtiques  
i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

---

OBJECTES INVARIANTS EN  
UN PÈNDOL PERTORBAT

---

Autor: Salvador Font i Bergadà

Director: Dr. Àngel Jorba

Realitzat a: Departament de Matemàtica Aplicada

Barcelona, 17 de gener de 2024



## Abstract

The disturbed pendulum's systems have some invariant objects, which in this project are studied in some cases. In order to do it, we start by understanding the Floquet's theorem and some numerical methods for integrating differential equations. These notions are applied afterwards in pinpointing periodic orbits, studying its stability and, were that to happen, the invariant manifolds around them.

## Resum

Els sistemes del pèndol pertorbat presenten uns objectes invariants els quals en aquest treball s'estudien en alguns casos. Per a fer-ho, es realitza un treball previ per entendre el teorema de Floquet i certs mètodes d'integració numèrica d'equacions diferencials. Aquests conceptes són aplicats posteriorment en la localització d'òrbites periòdiques, en un estudi de la seva estabilitat i, si es dona el cas, de les varietats invariants al voltant d'aquestes.

## Agraïments

Vull agrair la feina realitzada per tots els professors que m'he trobat en les diferents assignatures del grau de matemàtiques.

Especialment, agraeixo al Dr. Àngel Jorba el suport essencial i l'ajuda que m'ha donat com a tutor d'aquest treball.

També vull agrair a la meva família haver-me acompanyat durant tota l'etapa universitària i el procés de realitzar aquest treball.

# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
<b>2</b>	<b>El teorema de Floquet</b>	<b>2</b>
<b>3</b>	<b>Integració numèrica de EDOs</b>	<b>7</b>
3.1	El mètode d'Euler . . . . .	7
3.2	Convergència dels mètodes . . . . .	9
3.3	Els mètodes de Taylor . . . . .	9
3.4	La diferenciació automàtica . . . . .	10
3.5	Mètodes de Taylor amb diferenciació automàtica . . . . .	16
3.6	Ordre i pas òptims . . . . .	17
<b>4</b>	<b>El pèndol pertorbat</b>	<b>19</b>
4.1	Motivació de l'estudi . . . . .	19
4.2	Òrbites periòdiques . . . . .	21
4.3	Estabilitat de les òrbites periòdiques . . . . .	26
4.4	Mètode de continuació respecte $\omega$ . . . . .	29
<b>5</b>	<b>Intersecció de les separatius</b>	<b>36</b>
5.1	Aproximació lineal . . . . .	38
5.2	Parametrització de la varietat . . . . .	39
5.3	Resultats . . . . .	41
<b>6</b>	<b>Conclusions</b>	<b>44</b>
	<b>Referències</b>	<b>45</b>
	<b>Annex: Programes</b>	<b>47</b>
	Continuació . . . . .	47
	Angle entre separatius amb aproximació lineal de la varietat . . . . .	53
	Angle entre separatius amb parametrització de la varietat . . . . .	60



# 1 Introducció

L'objectiu principal d'aquest treball és, tal i com suggereix el seu títol, estudiar objectes invariants en els sistemes d'equacions diferencials de pèndols pertorbats. Això ens portarà a realitzar un estudi de les òrbites periòdiques d'aquests sistemes i la seva estabilitat.

Pel que fa a l'estructura, podem destacar-ne quatre punts principals. D'aquests, els dos primers presenten uns conceptes i eines fonamentals per a poder desenvolupar els dos darrers.

Comencem en l'apartat 2 amb un estudi teòric al voltant del teorema de Floquet, el qual es servirà quan haguem de tractar amb sistemes caracteritzats per matrius periòdiques. En l'apartat 3, es presenten uns mètodes d'integració numèrica de EDOs, començant pel mètode d'Euler i el de Taylor. També s'hi explica l'algorisme de la diferenciació automàtica i com es pot incorporar al mètode de Taylor per a fer-lo més eficient.

En l'apartat 4 ens trobem amb el primer dels dos sistemes de pèndol pertorbat que estudiarem. En aquest cas, ens centrarem en les òrbites periòdiques prop de l'origen. Veurem com ens serà de gran utilitat l'anomenada aplicació estroboscòpica per a localitzar-les i analitzar-ne l'estabilitat. També, degut a que el sistema que estudiem depèn de paràmetres, aplicarem un mètode de continuació per a localitzar més òrbites periòdiques i extreure més conclusions sobre el comportament d'aquestes respecte els paràmetres presents en el sistema.

Pel que fa a l'apartat 5, estudiarem un sistema de pèndol pertorbat lleugerament diferent a l'anterior. Aquí, el que ens interessarà serà l'estudi de les varietats invariants del punt amb òrbita periòdica inestable. Ens centrarem en calcular l'angle d'intersecció d'aquestes, el qual, degut a la pertorbació, deixa de ser nul. Per a fer-ho, necessitarem molts dels conceptes tractats en els apartats anteriors així com presentar un parell de mètodes per a poder arribar al resultat.

Com a referències principals d'aquest treball, més enllà de les cites de resultats i conceptes específics presents al llarg d'aquesta memòria, calen destacar els següents:

- L'apartat 2 d'aquest treball s'ha basat en el capítol 3 de [9], més concretament, en la resolució dels problemes 33 a 36.
- En l'apartat 3 s'ha agafat informació de [10] i [6] per a presentar els mètodes d'Euler i de Taylor i de [7] per a tot el relacionat amb la diferenciació automàtica i la seva aplicació en el mètode de Taylor.
- Pel que fa als apartats 4 i 5, els raonaments generals han sorgit durant les reunions amb el tutor del treball.
- Tots els programes necessaris per a obtenir els resultats i gràfics dels apartats 4 i 5, han estat fets per l'autor i implementen l'integrador d'equacions diferencials descrit en [5], el qual està basat en el mètode de Taylor amb la diferenciació automàtica exposat en l'apartat 3.

## 2 El teorema de Floquet

Estudiar les solucions d'un sistema d'equacions diferencials  $x' = A(t)x + b(t)$  ens resulta un exercici relativament senzill en el cas de ser un sistema homogeni ( $b(t) \equiv 0$ ) amb la matriu  $A(t)$  amb coeficients constants ( $A(t) \equiv A$ ).

Si estem en aquest cas, amb  $A$  una matriu  $n \times n$ , sent  $\lambda_i$  un valor propi de  $A$  associat al vector propi  $v_i$ , es verifica que  $x_i(t) = v_i e^{\lambda_i t}$  és solució de  $x' = Ax$ .

També, si tenim  $n$  solucions linealment independents d'un sistema d'equacions diferencials podem definir la matriu fonamental del sistema com aquella matriu formada, per columnes, per les  $n$  solucions linealment independents. Aleshores, en el cas d'estar treballant amb un sistema homogeni definit per una matriu amb coeficients constants la qual tingui els vectors propis linealment independents, la matriu fonamental està formada per columnes per les solucions, ja definides, associades a cada vector propi:

$$\phi(t) = (x_1 | \cdots | x_n).$$

Al llarg d'aquest treball es tractaran sistemes d'equacions diferencials que no verifiquen aquests requisits. De fet, en els sistemes que voldrem estudiar, la matriu  $A$  no només dependrà de  $t$  sinó que, a més, ho farà de manera periòdica. És a dir,  $A(t)$  verificarà  $A(t+T) = A(t) \forall t$ . Direm, doncs, que  $A(t)$  és una matriu  $T$ -periòdica.

Per a poder extreure conclusions basades en l'estudi de les seves solucions, ens cal presentar primer un resultat que ens relacioni aquestes solucions amb les d'un sistema definit per una matriu amb coeficients constants: el teorema de Floquet.

Abans de enunciar el teorema de Floquet, començem per presentar uns resultats que ens ajudaran a l'hora de demostrar-lo.

**Lema 2.1** (Logaritme d'una matriu). *Sigui  $C$  una matriu  $n \times n$  complexa amb  $\det C \neq 0$ . Aleshores, existeix una matriu complexa  $B$  tal que  $C = e^B$ .*

*Demostració:*

Abans de començar pròpiament amb la demostració del lema, recordem que podem considerar l'exponencial d'una matriu amb la següent expressió:

$$e^B = \sum_{k=0}^{\infty} \frac{B^k}{k!}.$$

Ara, observem com podem simplificar la demostració al relacionar el cas general que presenta el lema, per a una matriu  $C$  qualsevol, amb un cas més concret, les matrius en forma de Jordan.

Sigui  $J$  la forma de Jordan de  $C$ . Aleshores, podem expressar  $J = P^{-1}CP$ , per una certa matriu  $P$  ([1, p.138]). Observem que, suposant  $C = e^B$ , es verifica:

$$J = P^{-1}CP = P^{-1}e^B P = e^{P^{-1}BP}.$$



Per tant, demostrar l'existència d'una matriu  $D$  tal que  $J = e^D$  és suficient per a demostrar el lema, ja que si  $J$  és la forma de Jordan de  $C$  i existeix la matriu  $D$  anterior, també existirà la matriu  $B = PDP^{-1}$ .

Observem també que la matriu  $J$  és diagonal per blocs, amb blocs de dimensió  $j \times j$  ( $j \leq n$ ) de la forma  $J_i = \lambda_i Id + E_i$ , sent  $\lambda_i$  un valor propi de  $C$  i  $E_i$  una matriu nilpotent de grau  $j$  ( $E_i^j = 0$ ) formada per tot de zeros i la diagonal immediatament superior a la principal plena totalment per uns.

Aleshores, n'hi ha prou en demostrar el resultat per alguna d'aquestes  $J_i$ , ja que si demostrem l'existència d'una matriu  $D_i$  que verifica  $J_i = e^{D_i}$ , haurem demostrat l'existència de la matriu  $D$  que buscàvem. La matriu  $D$  és també diagonal per blocs amb les matrius  $D_i$  com a blocs.

Centrem-nos, doncs, en aquest cas.

Sigui  $J = \lambda Id + E$  de dimensió  $j \times j$ . Com  $\lambda \neq 0$  ( $\det C \neq 0 \rightarrow C$  no singular), podem expressar  $J = \lambda(Id + \frac{1}{\lambda}E)$ . Aleshores, de l'expressió:

$$\log(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k+1} x^{k+1},$$

en traiem:

$$\log(J) = \log(\lambda)Id + \log\left(Id + \frac{1}{\lambda}E\right) = \log(\lambda)Id + \sum_{i=0}^{\infty} \frac{(-1)^i}{i+1} \left(\frac{1}{\lambda}E\right)^{i+1}.$$

Tenint en compte la nilpotència de  $E$  ( $E^j = 0$ ), obtenim:

$$\log(J) = \log(\lambda)Id + \sum_{i=0}^{j-2} \frac{(-1)^i}{i+1} \left(\frac{1}{\lambda}E\right)^{i+1}.$$

Ara bé, ens falta demostrar que  $\log(J) = D$ , la matriu que buscàvem. Per això, comprovarem que es verifica  $J = e^D$  o, el que és el mateix,  $J = e^{\log(J)}$ .

Definim:

$$S := \sum_{i=0}^{j-2} \frac{(-1)^i}{i+1} \left(\frac{1}{\lambda}E\right)^{i+1},$$

aleshores,  $e^{\log(J)} = e^{\log(\lambda)Id+S} = e^{\log(\lambda)Id} \cdot e^S$ .

En l'última igualtat, s'ha utilitzat la següent propietat de l'exponencial de matrius:

$$\text{Si } AB = BA, \text{ aleshores, per a tot } t \ e^{t(A+B)} = e^{tA} \cdot e^{tB} \text{ ([8, p.60]).}$$

Es pot demostrar ([3, p.18]) que  $e^S = Id + \frac{1}{\lambda}E$ , amb el que podem concloure:

$$e^{\log(J)} = \lambda Id \cdot \left[Id + \frac{1}{\lambda}E\right] = \lambda \cdot \left[Id + \frac{1}{\lambda}E\right] = J.$$

Queda, per tant, demostrada l'existència de el logaritme de la matriu  $J$  i, conseqüentment, l'existència del logaritme d'una matriu  $C$  qualsevol.

□

Seguim enunciant i demostrant un corol·lari relacionat amb el lema anterior.

**Corol·lari 2.2.** *Sigui  $D$  una matriu real no singular. Aleshores, existeix una matriu real  $E$  tal que  $D^2 = e^E$ .*

*Demostració:*

Comencem recordant els següents resultats ([9]):

$$M \in \mathbb{R}^{n \times n} \leftrightarrow M = \overline{M},$$

$$e^{\overline{A}} = \overline{(e^A)}.$$

Pel lema anterior, sabem que existeix una matriu  $B$  tal que  $D = e^B$ . Com  $D$  és una matriu real, es verifica  $D^2 = D \cdot D = D \cdot \overline{D}$ . Llavors:

$$\left. \begin{array}{l} D \cdot D = e^B \cdot e^B \\ D \cdot \overline{D} = e^B \cdot \overline{(e^B)} = e^B \cdot e^{\overline{B}} \\ D \cdot D = D \cdot \overline{D} \end{array} \right\} \rightarrow e^B \cdot e^B = e^B \cdot e^{\overline{B}} \rightarrow$$

$$\rightarrow B = \overline{B} \rightarrow B \in \mathbb{R}^{n \times n}.$$

També es cert que  $D^2 = e^{2B}$  així que, si definim  $E := 2B$ , ja tenim demostrat l'enunciat ja que,  $D^2 = e^E$  i  $E \in \mathbb{R}^{n \times n}$ .

□

Per últim, una proposició que ens presenta una propietat de la matriu fonamental d'un sistema d'equacions diferencials caracteritzat per una matriu  $\tau$ -periòdica.

**Proposició 2.3.** *Sigui  $\phi(t)$  la matriu fonamental del sistema  $x' = A(t)x$  amb  $A(t)$   $\tau$ -periòdica. Aleshores, existeix una matriu  $C$  no singular tal que  $\phi(t+\tau) = \phi(t)C$ . A més,  $\phi(t+\tau)$  també és matriu fonamental del mateix sistema.*

*Demostració:*

Comencem veient que  $\varphi(t) := \phi(t+\tau)$  és efectivament matriu fonamental:

$$\phi'(t+\tau) = A(t+\tau)\phi(t+\tau) = A(t)\phi(t+\tau),$$

per tant,  $\varphi'(t) = A(t)\varphi(t)$ , és a dir,  $\varphi(t)$  és matriu fonamental del sistema  $x' = A(t)x$ .

Definim  $C(t) := \phi^{-1}(t)\phi(t + \tau) = \phi^{-1}(t)\varphi(t)$  i sigui  $C_0 := C(t_0)$ . Observem que  $\varphi_0(t) := \phi(t)C_0$  també és matriu fonamental:

$$(\phi(t)C_0)' = \phi'(t)C_0 = A(t)\phi(t)C_0 \quad \rightarrow \quad \varphi_0'(t) = A(t)\varphi_0(t).$$

Ara bé,  $\varphi_0(t_0) = \varphi(t_0)$  així que, per la unicitat de la solució, tenim  $\varphi_0(t) = \varphi(t)\forall t$ . Aleshores,  $C(t) \equiv C_0$ , constant en el temps.

També, com hem definit la matriu  $C(t)$  a partir de matrius fonamentals del sistema, aquesta  $C_0$  serà no singular.

Hem demostrat, doncs, l'existència de la matriu  $C$ , ja que  $C = C_0$ .

□

Ja tenim ara els resultats necessaris per a demostrar el teorema de Floquet.

**Teorema 2.4** (Teorema de Floquet). *Signi  $A(t)$  una matriu  $\tau$ -periòdica. Aleshores, existeix una matriu  $P(t)$   $\tau$ -periòdica i una matriu  $B$ , en general complexa, tals que per a la matriu  $\phi(t)$  (matriu fonamental del sistema  $x' = A(t)x$ ) es verifica  $\phi(t) = P(t) \cdot e^{Bt}$ .*

*Demostració:*

Per la proposició 2.2, coneixem l'existència d'una matriu no singular  $C$  tal que  $\phi(t + \tau) = \phi(t)C$ . Com  $C$  és constant,  $\phi(\tau) = \phi(0)C = Id \cdot C = C$ .

Pel lema 2.1, aplicable a aquesta matriu  $C$ , existeix una matriu  $\tilde{B}$  tal que  $C = e^{\tilde{B}}$  i, per tant, podem expressar  $C = e^{B\tau}$  (sent  $\tilde{B} = B\tau$ ).

Ara, definim  $P(t) = \phi(t)e^{-Bt}$ . Observem que verifica  $\phi(t) = P(t) \cdot e^{Bt}$ , falta comprovar que es  $\tau$ -periòdica.

$$P(t + \tau) = \phi(t + \tau)e^{-B(t+\tau)} = \phi(t + \tau)e^{-B\tau}e^{-Bt},$$

$$\phi(t + \tau) = \phi(t)C = \phi(t)e^{B\tau},$$

↓

$$P(t + \tau) = \phi(t)e^{B\tau}e^{-B\tau}e^{-Bt} = \phi(t)e^{-Bt} = P(t).$$

□

Ara que ja hem presentat i demostrat el teorema de Floquet, pot semblar que no ens serveixi per a l'objectiu que perseguíem: relacionar les solucions d'un sistema d'equacions diferencials amb una matriu  $\tau$ -periòdica amb les solucions d'un sistema d'equacions diferencials amb una matriu a coeficients constants.

Aquesta relació, però, queda demostrada en el corollari següent:

**Corollari 2.5.** *Si sigui  $A(t)$  una matriu  $\tau$ -periòdica. Aleshores, existeix una matriu periòdica  $P(t)$  tal que la transformació  $\varphi(t) \rightarrow P(t)\varphi(t)$  transforma biunívocament les solucions de  $x' = A(t)x$  a les d'una equació lineal a coeficients constants  $x' = Bx$ .*

*Demostració:*

Si sigui el canvi de variables  $x = P(t)y$ , on la matriu  $P(t)$  és la definida en el teorema de Floquet ( $P(t) = \phi(t)e^{-Bt}$ ). Aleshores, de derivar el canvi de variables i considerar el sistema de la matriu  $A(t)$  obtenim la següent expressió:

$$\left. \begin{array}{l} x' = P(t)'y + P(t)y' \\ x' = A(t)x \end{array} \right\} \rightarrow A(t)x = P'(t)y + P(t)y'.$$

Substituïm  $x = P(t)y$  i calculem la derivada de  $P(t)$  a partir de la seva expressió ( $P'(t) = \phi'(t)e^{-Bt} - B\phi(t)e^{-Bt}$ ):

$$A(t)P(t)y = \phi'(t)e^{-Bt}y - B\phi(t)e^{-Bt}y + P(t)y'.$$

Ara, tenint en compte que  $\phi'(t) = A(t)\phi(t)$ :

$$A(t)P(t)y = A(t)\phi(t)e^{-Bt}y - B\phi(t)e^{-Bt}y + P(t)y'.$$

Apliquem de nou l'expressió de  $P(t)$  per a reescriure la igualtat:

$$A(t)P(t)y = A(t)P(t)y - BP(t)y + P(t)y'.$$

Reordenant els termes arribem a poder cancel·lar  $A(t)P(t)y$  i  $P(t)$  de tal manera que ens quedem amb un sistema d'equacions diferencials lineal a coeficients constants:

$$y' = By.$$

□

Ara ja hem trobat el que volíem. La matriu  $P(t)$  definida en la demostració del teorema de Floquet ens permet realitzar la transformació de solucions entre el sistema  $\tau$ -periòdic que estudiarem i un sistema lineal a coeficients constants.

### 3 Integració numèrica de EDOs

Quan estem davant d'una equació diferencial:

$$x'(t) = f(t, x(t)),$$

amb  $x(t) \in \mathbb{R}^n$  i  $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ , un dels principals problemes que se'ns plantegen és trobar una funció  $\varphi : \mathbb{R} \rightarrow \mathbb{R}^n$  que verifiqui  $\varphi'(t) = f(t, \varphi(t))$ .

En general, donada una equació diferencial, existeixen infinites solucions  $\varphi(t)$  d'aquesta. D'entre totes elles, sabem que només una verificarà el conegut problema de Cauchy o de valor inicial:

$$\begin{cases} x'(t) = f(t, x(t)), \\ x(t_0) = x_0. \end{cases}$$

Del qual suposem que  $f$  és una funció analítica en el seu domini i que  $x(t)$  està definida per a tot  $t \in I = [t_0, t_f]$ .

La solució d'aquest problema forma part del conjunt de solucions de  $x'(t) = f(t, x(t))$  però, a més, verifica que a l'avaluar-la en  $t_0$  ens retorna  $x_0$ . La distingirem, doncs, de la resta de solucions utilitzant la següent expressió:

$$\varphi(t; t_0, x_0),$$

tot i que, en ocasions, també en farem referència com  $\varphi(t)$  quan sigui clar que ens estem referint no a la solució d'una equació diferencial, sinó a la solució d'un problema de valor inicial donat.

Aquesta solució al problema de valor inicial es pot trobar analíticament només en certs casos. És per aquest motiu que necessitem mètodes numèrics aplicables a un conjunt major de casos per a poder aproximar-la.

El nostre objectiu és, per tant, trobar una aproximació de la solució del problema ( $\varphi(t)$  o  $\varphi(t; t_0, x_0)$ ) en  $I = [t_0, t_f]$ .

#### 3.1 El mètode d'Euler

Aquest mètode, potser el més elemental en quant a integració de equacions diferencials, ens permetrà calcular una aproximació de  $\varphi(t_0 + h)$ , la qual anomenarem  $\psi(t_0 + h)$ , sent  $h$  un valor que considerem positiu i petit (en comparació amb l'amplada de l'interval a integrar:  $I = [t_0, t_f]$ ).

Per aplicar aquest mètode, així com els que s'explicaran posteriorment, hem de suposar que podem calcular tantes derivades temporals de la solució  $\varphi(t; t_0, x_0)$  com ens calgui i que aquestes seràn contínues en  $I = [t_0, t_f]$ .

Podem deduir aquest mètode a partir de l'aproximació de la derivada de la solució en  $t_0$  ( $\varphi(t_0)$ ) com a diferències dividides:

$$\frac{\varphi(t_0 + h) - \varphi(t_0)}{(t_0 + h) - t_0} \simeq \varphi'(t_0),$$

↓

$$\frac{\varphi(t_0 + h) - \varphi(t_0)}{h} \simeq f(t_0, \varphi(t_0)).$$

Per a obtenir la segona equació, hem aplicat que  $\varphi'(t) = f(t, \varphi(t))$ . Obtenim doncs, aïllant-la d'aquesta segona expressió, l'aproximació de  $\varphi(t_0 + h)$  que buscàvem:

$$\varphi(t_0 + h) \simeq \psi(t_0 + h) := \varphi(t_0) + hf(t_0, \varphi(t_0)).$$

Els mètodes que descriurem no ens serviran per a trobar l'aproximació de la solució  $\psi(t) \forall t \in I = [t_0, t_f]$ , sinó que el que trobarem seran els valors de  $\psi(t)$  per a un conjunt indeterminadament gran de punts, normalment equidistants, inclosos en l'interval  $I = [t_0, t_f]$ .

Per al mètode d'Euler, si definim  $h := (t_f - t_0)/n$  per un  $n \in \mathbb{N}$ ,  $t_i := t_0 + ih$  per  $i \in [0, n]$  i  $\psi_i := \psi(t_i)$ , calcularem les aproximacions a la solució  $\psi_i$  seguint la recurrència ([10, p.435]):

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \psi_i + hf(t_i, \psi_i) \end{cases} \quad \forall i \in [0, n-1].$$

Aquesta resulta d'aplicar  $\psi(t_0 + h) := \varphi(t_0) + hf(t_0, \varphi(t_0))$  tenint en compte que per a  $i \geq 1$  desconeixem el valor de  $\varphi(t_i)$ . Per tant, hem de substituir-lo per al valor aproximat  $\psi_i$  que sí tenim calculat.

Una altra manera d'entendre aquesta recurrència és considerar que estem calculant, per a cada  $t_i$ , l'aproximació a la solució del problema de valor inicial:

$$\begin{cases} x'(t) = f(t, x(t)), \\ x(t_i) = \psi_i, \end{cases}$$

en  $t_i + h$ .

Aquesta, segons els càlculs anteriors, és:

$$\psi(t_i + h) := \psi_i + hf(t_i, \psi_i),$$

coincidint, així, amb l'expressió de la recurrència, ja que  $\psi(t_i + h) = \psi(t_{i+1}) = \psi_{i+1}$ .

### 3.2 Convergència dels mètodes

Una característica que es raonable demanar que verifiquin els mètodes d'integració com el d'Euler és que, en el límit  $h \rightarrow 0$ , la solució aproximada que obtenim ( $\psi(t)$ ) convergeixi a la solució real del problema de valor inicial que estiguem estudiant ( $\phi(t)$ ). D'aquesta manera, podrem aconseguir, teòricament, qualsevol precisió desitjada prenent un valor de  $h$  suficientment petit.

En tal cas, direm que el mètode és convergent.

**Definició 3.1** (Mètode convergent ). ([6, p.26])

*Un mètode numèric s'anomena convergent a la solució  $\phi(t)$  d'un problema de valor inicial donat en  $t = t^* \in I = [t_0, t_f]$  si l'error global:  $e_n = \psi(t_n) - \phi(t_n)$ , en  $t_n = t^*$ , verifica:*

$$|e_n| \rightarrow 0,$$

quan  $h \rightarrow 0$ .

*Direm que convergeix a  $p$ -èsim ordre si  $e_n = \mathcal{O}(h^p)$  per algun  $p > 0$ . Farem referència al major valor de  $p$  que ho verifiqui com a **ordre del mètode**.*

Com a norma general, vàlida en els casos en els quals els aplicarem al llarg d'aquest treball, el mètode d'Euler així com els mètodes de Taylor que descriurem a continuació són mètodes convergents.

### 3.3 Els mètodes de Taylor

Observem que l'aproximació que obtenim amb el mètode d'Euler ( $\psi_{i+1} = \psi_i + hf(t_i, \psi_i)$ ) és també l'expressió dels primers termes de la sèrie de Taylor de  $\psi(t)$  a prop de  $t_i$ :

$$\psi_{i+1} = \psi_i + hf(t_i, \psi_i) = \psi_i + h\psi'_i.$$

D'aquesta manera, veiem que l'error que cometem en cada pas d'Euler (error local), que és l'error de truncar la sèrie de Taylor, és proporcional a  $h^2$ . Després de  $n$  passos, però, l'error global comès és:

$$e_n \propto nh^2 \propto h.$$

Ja que, recordem que hem definit  $h$  de la següent forma:  $h := (t_f - t_0)/n$ .

El mètode d'Euler és, per tant, un mètode d'ordre 1.

Vist així, sembla clar que si volem reduir aquest error i, per tant, augmentar la precisió de les nostres aproximacions de  $\varphi(t)$ , hauríem de treballar amb més termes de la sèrie de Taylor. Obtindríem així el mètode següent ([6, p.37]):

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \psi_i + h\psi'_i + h^2\frac{\psi''_i}{2!} + \dots + h^p\frac{\psi_i^{(p)}}{p!} \end{cases} \quad \forall i \in [0, n-1].$$

Aquest plantejament, però, ens genera uns nous inconvenients. Per començar, necessitem conèixer l'expressió de les derivades de  $\psi_i$  per a poder avaluar-les:

$$\begin{aligned} \psi'_i &= f(t_i, \psi_i), \\ \psi''_i &= f_t(t_i, \psi_i) + f_{\psi_i}(t_i, \psi_i)\psi'_i, \\ \psi'''_i &= (\psi_i)^2 f_{\psi_i\psi_i}(t_i, \psi_i) + \psi_i[2f_{t\psi_i}(t_i, \psi_i) + (f_{\psi_i}(t_i, \psi_i))^2] + f_t(t_i, \psi_i)f_{\psi_i}(t_i, \psi_i) + \\ &\quad + f_{tt}(t_i, \psi_i), \\ &\vdots \end{aligned}$$

Aquestes expressions cada vegada més intricades faran que no només ens suposi un esforç major a l'hora de implementar el mètode, sinó que també tardarem més temps a l'hora d'executar-lo.

### 3.4 La diferenciació automàtica

Per a poder treballar amb més termes de la sèrie de Taylor corresponent sense trobar-nos amb el problema presentat anteriorment, podem utilitzar un procediment anomenat diferenciació automàtica. Aquest ens permetrà avaluar ràpidament les derivades d'una funció en un punt donat fins a un ordre arbitràriament alt.

Podrem aplicar la diferenciació automàtica a qualsevol funció elemental, és a dir, que estigui formada per sumes, productes, quocients i composicions de les funcions potència, exponencial, logarítmica i trigonomètriques. Per a fer-ho, ens cal primer presentar el concepte de la derivada normalitzada, el qual té un paper fonamental en la diferenciació automàtica.

**Definició 3.2** (Derivada n-èssima normalitzada). *Sigui  $f : t \in I \subset \mathbb{R} \mapsto \mathbb{R}$ .*

*Definim la seva derivada n-èssima normalitzada de la següent manera:*

$$f^{[n]}(t) := \frac{1}{n!}f^{(n)}(t),$$

on  $f^{(n)}(t)$  és la derivada n-èssima de  $f$  en  $t \in I$ .

Podem observar que la derivada n-èssima normalitzada de les funcions potència, exponencial i logarítmica és fàcilment calculable a partir de la definició:

$$\bullet f(t) = t^\alpha \Rightarrow f^{[n]}(t) = \begin{cases} \frac{1}{n!} \prod_{j=0}^{n-1} (\alpha - j)t^{\alpha-n} & \text{si } n \leq \alpha, \\ 0 & \text{si } n > \alpha. \end{cases}$$



- $f(t) = e^t \Rightarrow f^{[n]}(t) = \frac{1}{n!}e^t$ .
- $f(t) = \ln(t) \Rightarrow f^{[n]}(t) = \frac{1}{n!} \frac{(n-1)!(-1)^{n-1}}{t^n}$ .

Anem a veure ara com podem calcular la derivada n-èsima normalitzada de funcions que són formades a partir de les anteriors mitjançant la composició i les operacions elementals, així com la derivada n-èsima normalitzada de les funcions trigonomètriques.

**Proposició 3.3.** *Siguin  $a$ ,  $b$  i  $c$  funcions de classe  $C^n$  i sigui  $\alpha \in \mathbb{R} \setminus \{0\}$ .*

1. Si  $a(t) = b(t) \pm c(t)$ , aleshores:

$$a^{[n]}(t) = b^{[n]}(t) \pm c^{[n]}(t).$$

2. Si  $a(t) = b(t)c(t)$ , aleshores:

$$a^{[n]}(t) = \sum_{j=0}^n b^{[n-j]}(t)c^{[j]}(t).$$

3. Si  $a(t) = \frac{b(t)}{c(t)}$ , aleshores:

$$a^{[n]}(t) = \frac{1}{c^{[0]}(t)} \left[ b^{[n]}(t) - \sum_{j=1}^n c^{[j]}(t)a^{[n-j]}(t) \right].$$

4. Si  $a(t) = b(t)^\alpha$ , aleshores:

$$a^{[n]}(t) = \frac{1}{nb^{[0]}(t)} \sum_{j=0}^{n-1} (n\alpha - j(\alpha + 1))b^{[n-j]}(t)a^{[j]}(t).$$

5. Si  $a(t) = e^{b(t)}$ , aleshores:

$$a^{[n]}(t) = \frac{1}{n} \sum_{j=0}^{n-1} (n-j)a^{[j]}(t)b^{[n-j]}(t).$$

6. Si  $a(t) = \log b(t)$ , aleshores:

$$a^{[n]}(t) = \frac{1}{b^{[0]}(t)} \left[ b^{[n]}(t) - \frac{1}{n} \sum_{j=1}^{n-1} (n-j)b^{[j]}(t)a^{[n-j]}(t) \right].$$

7. Si  $a(t) = \cos c(t)$  i  $b(t) = \sin c(t)$ , aleshores:

$$a^{[n]}(t) = -\frac{1}{n} \sum_{j=1}^n j b^{[n-j]}(t)c^{[j]}(t),$$

$$b^{[n]}(t) = \frac{1}{n} \sum_{j=1}^n j a^{[n-j]}(t) c^{[j]}(t).$$

*Demostració:*

1. Aquest és trivial, partint de la definició de derivada n-èsima, de la propietat que la derivada d'una suma de funcions es igual a la suma de derivades i de la propietat distributiva de la suma vers el producte.

2. Recordem que  $a^{[n]}(t) = \frac{1}{n!} a^{(n)}(t)$ .

Apliquem la fórmula de Leibniz a  $a^{(n)}(t)$  i obtenim:

$$a^{[n]}(t) = \frac{1}{n!} \sum_{j=0}^n \binom{n}{j} b^{(n-j)}(t) c^{(j)}(t) = \frac{1}{n!} \sum_{j=0}^n \frac{n!}{(n-j)!j!} b^{(n-j)}(t) c^{(j)}(t).$$

Introduint  $\frac{1}{n!}$  dins el sumatori, podem cancel·lar el  $n!$  i, amb  $(n-j)!$  i  $j!$  podem recuperar les expressions de la derivada normalitzada:

$$a^{[n]}(t) = \sum_{j=0}^n b^{[n-j]}(t) c^{[j]}(t).$$

3. Apliquem (2) a  $b(t) = a(t)c(t)$ :

$$b^{[n]}(t) = \sum_{j=0}^n a^{[n-j]}(t) c^{[j]}(t) = a^{[n]}(t) c^{[0]}(t) + \sum_{j=1}^n a^{[n-j]}(t) c^{[j]}(t).$$

Aïllem  $a^{[n]}(t)$  per a obtenir el resultat:

$$a^{[n]}(t) = \frac{1}{c^{[0]}(t)} \left[ b^{[n]}(t) - \sum_{j=1}^n c^{[j]}(t) a^{[n-j]}(t) \right].$$

4. Si prenem logaritmes a banda i banda de l'expressió  $a(t) = b(t)^\alpha$  i després derivem, podem arribar a:  $a'(t)b(t) = \alpha a(t)b'(t)$ .

Abans de continuar, ens cal tenir en compte que:

$$(a')^{[n]}(t) = \frac{1}{n!} a^{(n)}(t) = \frac{n+1}{(n+1)!} a^{(n+1)}(t) = (n+1) a^{[n+1]}(t).$$

Ara apliquem (2) a  $a'(t)b(t) = \alpha a(t)b'(t)$  per  $n-1$ .

Per la banda dreta ens queda:

$$\sum_{j=0}^{n-1} (a')^{[j]}(t) b^{[n-1-j]}(t) = \sum_{j=0}^{n-1} j a^{[j+1]}(t) b^{[n-1-j]}(t).$$

Fem un canvi de variable  $j \rightarrow j + 1$  per a obtenir un sumatori que puguem unir amb el que calcularem a continuació per a la banda esquerra de l'equació:

$$\sum_{j=0}^n j a^{[j]}(t) b^{[n-j]}(t).$$

Observem que podem introduir el cas  $j = 0$  en aquest últim sumatori sense alterar-ne el resultat.

Per a la banda esquerra ens queda:

$$\alpha \sum_{j=0}^{n-1} a^{[j]}(t) (b')^{[n-1-j]}(t) = \alpha \sum_{j=0}^{n-1} (n-j) a^{[j]}(t) b^{[n-j]}(t).$$

Unim les dues expressions:

$$\sum_{j=0}^n j a^{[j]}(t) b^{[n-j]}(t) = \alpha \sum_{j=0}^{n-1} (n-j) a^{[j]}(t) b^{[n-j]}(t).$$

Extraiem el cas  $n$  del sumatori de l'esquerra:

$$n a^{[n]}(t) b^{[0]}(t) + \sum_{j=0}^{n-1} j a^{[j]}(t) b^{[n-j]}(t) = \alpha \sum_{j=0}^{n-1} (n-j) a^{[j]}(t) b^{[n-j]}(t).$$

Ja podem aïllar  $a^{[n]}(t)$ :

$$a^{[n]}(t) = \frac{1}{n b^{[0]}(t)} \sum_{j=0}^{n-1} (n\alpha - j(\alpha + 1)) b^{[n-j]}(t) a^{[j]}(t).$$

5. Prenent logaritmes i derivant  $a(t) = e^{b(t)}$  podem arribar a:  $a'(t) = a(t)b'(t)$ . Ara apliquem (2) per a  $n - 1$ :

$$(a')^{[n-1]} = \sum_{j=0}^{n-1} a^{[j]}(b')^{[n-1-j]}.$$

Com  $(a')^{[n-1]} = n a^{[n]}$  i  $(b')^{[n-1-j]} = (n-j) b^{[n-j]}$ , obtenim:

$$a^{[n]}(t) = \frac{1}{n} \sum_{j=0}^{n-1} (n-j) a^{[j]}(t) b^{[n-j]}(t).$$

6. Derivant podem arribar a l'expressió:  $a'(t)b(t) = b'(t)$ . Ara apliquem (2) per a  $n - 1$  i, anàlogament al cas anterior, obtenim:

$$n b^{[n]}(t) = \sum_{j=0}^{n-1} (n-j) b^{[j]}(t) a^{[n-j]}(t).$$

Traiem el cas  $j = 0$  del sumatori:

$$nb^{[n]}(t) = na^{[n]}(t)b^{[0]}(t) + \sum_{j=0}^{n-1} (n-j)b^{[j]}(t)a^{[n-j]}(t).$$

Ara aïllem per a obtenir:

$$a^{[n]}(t) = \frac{1}{b^{[0]}(t)} \left[ b^{[n]}(t) - \frac{1}{n} \sum_{j=1}^{n-1} (n-j)b^{[j]}(t)a^{[n-j]}(t) \right].$$

7. Derivant podem arribar a  $a'(t) = -b(t)c'(t)$  i  $b'(t) = a(t)c'(t)$ . Apliquem (2) a les dues expressions per a  $n - 1$ . Fent uns càlculs anàlegs als del cas (5), ens queda:

$$a^{[n]}(t) = -\frac{1}{n} \sum_{j=1}^n j b^{[n-j]}(t) c^{[j]}(t),$$

i

$$b^{[n]}(t) = \frac{1}{n} \sum_{j=1}^n j a^{[n-j]}(t) c^{[j]}(t).$$

□

Veiem ara un exemple de com aplicar les fórmules de la proposició anterior per a calcular la derivada n-èsima normalitzada d'una funció donada. Ho farem a partir de les funcions que formen un sistema d'equacions diferencials. Més endavant, veurem com podrem incloure aquestes derivades normalitzades a l'hora d'implementar un mètode de Taylor i els beneficis que això ens comportarà.

**Exemple 3.4.** Calcularem la derivada n-èsima normalitzada de les funcions del sistema que descriu el moviment d'un pèndol pertorbat:

$$x''(t) + \omega^2 \sin(x(t)) = \varepsilon \sin(t) \rightarrow \begin{cases} x'(t) & = y(t), \\ y'(t) & = -\omega^2 \sin(x(t)) + \varepsilon \sin(t). \end{cases}$$

Per a poder aplicar la proposició, ens cal modificar el sistema fent ús de variables auxiliars de la següent manera:

$$\left\{ \begin{array}{l} u_1(t) = x(t), \\ u_2(t) = y(t), \\ u_3(t) = t, \\ u_4(t) = \sin(u_1(t)), \\ u_5(t) = \cos(u_1(t)), \\ u_6(t) = -\omega^2 u_4(t), \\ u_7(t) = \sin(u_3(t)), \\ u_8(t) = \cos(u_3(t)), \\ u_9(t) = \varepsilon u_7(t), \\ u_{10}(t) = u_6(t) + u_9(t), \\ x'(t) = u_2(t), \\ y'(t) = u_{10}(t). \end{array} \right.$$

Les variables  $u_5(t)$  i  $u_8(t)$  ens caldràn per a poder aplicar el punt 7 de la proposició.

Ara, calculem els valors  $u_i^{[n]}(t)$  segons la proposició anterior:

$$\left\{ \begin{array}{l} u_1^{[n]}(t) = x^{[n]}(t), \\ u_2^{[n]}(t) = y^{[n]}(t), \\ u_3^{[n]}(t) = \begin{cases} t & \text{si } n = 0, \\ 1 & \text{si } n = 1, \\ 0 & \text{si } n > 1, \end{cases} \\ u_4^{[n]}(t) = \frac{1}{n} \sum_{j=1}^n j u_5^{[n-j]}(t) u_1^{[j]}(t), \\ u_5^{[n]}(t) = -\frac{1}{n} \sum_{j=1}^n j u_4^{[n-j]}(t) u_1^{[j]}(t), \\ u_6^{[n]}(t) = -\omega^2 u_4^{[n]}(t), \\ u_7^{[n]}(t) = \frac{1}{n} \sum_{j=1}^n j u_8^{[n-j]}(t) u_3^{[j]}(t), \\ u_8^{[n]}(t) = -\frac{1}{n} \sum_{j=1}^n j u_7^{[n-j]}(t) u_3^{[j]}(t), \\ u_9^{[n]}(t) = \varepsilon u_7^{[n]}(t), \\ u_{10}^{[n]}(t) = u_6^{[n]}(t) + u_9^{[n]}(t), \\ x'^{[n]}(t) = u_2^{[n]}(t), \\ y'^{[n]}(t) = u_{10}^{[n]}(t). \end{array} \right.$$

Per a obtenir aquest resultat hem utilitzat l'expressió de la derivada n-èsima normalitzada de la funció potència, els punts 1 i 7 de la proposició i el fet que, degut a la definició de derivada n-èsima normalitzada, es manté la propietat que les constants "surten" fora la derivada.

Podem obtenir els valors de  $x^{[n+1]}(t)$  i  $y^{[n+1]}(t)$  necessaris per a reinicialitzar la recurrència:

$$x^{[n+1]}(t) = \frac{1}{(n+1)!} x^{(n+1)}(t) = \frac{1}{(n+1)!} u_2^{(n)}(t) = \frac{n!}{(n+1)!} u_2^{[n]}(t) = \frac{1}{n+1} u_2^{[n]}(t).$$

Així, doncs:

$$\begin{cases} x^{[n+1]}(t) &= \frac{1}{n+1}u_2^{[n]}(t), \\ y^{[n+1]}(t) &= \frac{1}{n+1}u_{10}^{[n]}(t). \end{cases}$$

Fixem-nos que el que hem construït és una recurrència per a calcular  $x^{[n]}(t)$  i  $y^{[n]}(t)$  per a una  $n$  qualsevol a un temps  $t$  donat, sempre i quant existeixin i estiguin ben definides les  $x^{[j]}(t)$  i  $y^{[j]}(t)$  per a  $0 < j < n$ . Aquest procés recursiu és el que anomenem **diferenciació automàtica**.

### 3.5 Mètodes de Taylor amb diferenciació automàtica

Hem vist que plantejar els mètodes de Taylor de la següent manera:

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \psi_i + h\psi'_i + h^2\frac{\psi''_i}{2!} + \dots + h^p\frac{\psi_i^{(p)}}{p!} \end{cases} \quad \forall i \in [0, n-1],$$

ens generava problemes a l'hora de calcular les derivades. Ara bé, fent ús de la diferenciació automàtica, podem plantejar-los:

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \sum_{j=0}^p \psi_i^{[j]}(t_i)h^j \end{cases} \quad \forall i \in [0, n-1].$$

Podem ara, doncs, augmentar l'ordre del mètode ( $p$ ) sense necessitat de realitzar més càlculs explícits de les successives derivades de  $\psi_i$ , només fent més iteracions de la recurrència que calcula les derivades normalitzades. D'aquesta manera, ja no se'ns presenta el problema de calcular i avaluar intricades expressions. De totes maneres, l'algoritme de la diferenciació automàtica ens pot suposar un esforç computacional, el qual en alguns casos pot ser innecessari.

Prenem com a referència la descripció del mètode de Taylor amb les derivades normalitzades:

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \sum_{j=0}^p \psi_i^{[j]}(t_i)h^j \end{cases} \quad \forall i \in [0, n-1].$$

Tal i com hem comentat en la descripció del mètode d'Euler, es pot interpretar aquesta recurrència com la resolució d'un problema de valor inicial relacionat amb el sistema de equacions diferencials corresponent. Així, en cada pas coneixem l'aproximació  $\psi_i$  a  $t_i$  i ens preguntem, donada la condició inicial  $\varphi(t_i) = \psi_i$ , quina és l'aproximació a  $t_i + h$ .

Fixem-nos que, d'aquesta manera, al llarg de les  $n$  iteracions estem mantenint un pas  $h$  i un ordre  $p$  constants. Aquest fet resulta contraproduent ja que es pot donar el cas que per a un  $t_i$ , no ens sigui necessari arribar a ordre  $p$  per a garantir la precisió desitjada, o que per a un  $t_j$  ens calgui augmentar aquest ordre. També, si podem augmentar el pas  $h$  entre dues iteracions, podrem acabar reduint el nombre total d'iteracions necessàries per arribar a integrar tot l'interval  $I = [t_0, t_f]$ .

Per tant, en termes de reduir el nombre de càlculs a realitzar i obtenir així un algoritme el més eficient possible, en cada iteració  $i$  del mètode entre  $t_0$  i  $t_f$  hauríem de trobar un ordre ( $p_i$ ) i un pas ( $h_i$ ) òptims. És a dir, un ordre el més baix possible i un pas el més gran possible que ens permetin mantenir la precisió que necessitem.

### 3.6 Ordre i pas òptims

Una condició que han de mantenir aquests valors  $p_i$  i  $h_i$  és:

$$\|\psi_i(t_{i+1}) - \psi_{i+1}\| \leq \varepsilon.$$

**Proposició 3.5.** *Suposem que la funció  $z \mapsto \psi(t_i + z)$  és analítica en un disc de radi  $\rho_i$ . Sigui  $A_i$  una constant positiva tal que:*

$$|\psi_i^{[j]}| \leq \frac{A_i}{\rho_i^j} \quad \forall j \in \mathbb{N}.$$

*Aleshores, si la precisió requerida  $\varepsilon$  tendeix a 0, els valors de  $h_i$  i  $p_i$  que ens donen la precisió  $\varepsilon$  i minimitzen el nombre global d'operacions tendeixen a:*

$$h_i = \frac{\rho_i}{e^2} \quad , \quad p_i = -\frac{1}{2} \ln\left(\frac{\varepsilon}{A_i}\right) - 1.$$

Cal notar que els valors òptims de pas i ordre que ens dona la proposició anterior, ho són només si la cota  $\frac{A_i}{\rho_i^j}$  no pot ser millorada. Si, per exemple, podem reduir el valor de  $A_i$  o si  $\psi(t)$  és una funció entera, els valors deixen de ser òptims.

Aquesta proposició tampoc ens serveix a nivell pràctic, ja que els valors de  $A_i$  i  $\rho_i$  no són fàcilment calculables. Per tant, determinarem els valors de  $h_i$  i  $p_i$  de la següent manera:

Siguin  $\varepsilon_a$  i  $\varepsilon_r$  les toleràncies absoluta i relativa de l'error. Si  $\varepsilon_r \|\psi_i\| \leq \varepsilon_a$ , controlarem l'error absolut amb  $\varepsilon_a$  i, en cas contrari, controlarem l'error relatiu amb  $\varepsilon_r$ . Cal notar que en tots dos casos estem igualment controlant l'error absolut per  $\max\{\varepsilon_a, \varepsilon_r \|\psi_i\|\}$ .

Comencem calculant l'ordre:

Definim  $\varepsilon_i$ :

$$\varepsilon_i = \begin{cases} \varepsilon_a & \text{si } \varepsilon_r \|\psi_i\|_\infty \leq \varepsilon_a, \\ \varepsilon_r & \text{altrament.} \end{cases}$$

Aleshores,

$$p_i = \left\lceil -\frac{1}{2} \ln \varepsilon_i + 1 \right\rceil.$$

Pel que fa al càlcul del pas, definim  $\rho_i^{(j)}$ :

$$\rho_i^{(j)} = \begin{cases} \left( \frac{1}{\|\psi_i^{[j]}\|_\infty} \right)^{\frac{1}{j}} & \text{per } 1 \leq j \leq p \quad \text{si } \varepsilon_r \|\psi_i\|_\infty \leq \varepsilon_a, \\ \left( \frac{\|\psi_i\|_\infty}{\|\psi_i^{[j]}\|_\infty} \right)^{\frac{1}{j}} & \text{per } 1 \leq j \leq p \quad \text{altrament.} \end{cases}$$

Per tant, el radi de convergència és:  $\rho_i = \min\{\rho_i^{(p-1)}, \rho_i^p\}$  i tenim:

$$h_i = \frac{\rho_i}{e^2}.$$

Així, a l'introduir la diferenciació automàtica en el mètode de Taylor, per a obtenir un mètode eficient en quant a nombre de càlculs a realitzar, substituïm la descripció amb un ordre  $p$  i un pas  $h$  constants:

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \sum_{j=0}^p \psi_i^{[j]}(t_i) h^j \end{cases} \quad \forall i \in [0, n-1].$$

Per una en la qual l'ordre i el pas són calculats en cada iteració, tal i com s'ha explicat, per a obtenir els seus valors òptims ( $p_i$  i  $h_i$ ):

$$\begin{cases} \psi_0 = \varphi(t_0) = x_0, \\ \psi_{i+1} = \sum_{j=0}^{p_i} \psi_i^{[j]}(t_i) h_i^j \end{cases} \quad \forall i \in [0, n-1].$$



## 4 El pèndol pertorbat

En aquest apartat buscarem i estudiarem les òrbites periòdiques d'un pèndol pertorbat aprofitant els conceptes presentats en els dos primers apartats.

Considerem l'equació d'un pèndol pertorbat:

$$x'' + \omega^2 \sin(x) = \varepsilon \sin(t).$$

Per a poder treballar aplicant els conceptes presentats anteriorment, ens convé reescriure aquesta equació en forma de sistema d'equacions diferencials ordinàries de primer ordre:

$$\begin{cases} x' = y, \\ y' = -\omega^2 \sin(x) + \varepsilon \sin(t). \end{cases}$$

### 4.1 Motivació de l'estudi

Analitzem la següent equació d'un pèndol:

$$x'' + \omega^2 \sin(x) = 0,$$

la qual resulta de considerar l'equació presentada a l'inici d'aquest apartat amb un valor de  $\varepsilon = 0$ . Prop de l'origen ( $x = 0$ ), podem aproximar-la linealment i obtenir:

$$x'' + \omega^2 x = 0.$$

Aquesta equació diferencial té solucions de la forma  $A \cos(\omega t) + B \sin(\omega t)$ , les quals tenen una freqüència  $\omega$ . A l'introduir la pertorbació  $\varepsilon \sin(t)$  amb una  $\varepsilon \neq 0$ , hi estem incorporant una funció de freqüència 1. Aquest fet ens genera un comportament peculiar quan estem en el cas  $\omega = 1$ , quan les dues freqüències coincideixen.

Per a poder veure-ho millor, comencem per buscar les solucions de:

$$x'' + \omega^2 x = \varepsilon \sin(t),$$

en el cas  $\omega \neq 1$ . Com hem comentat, les solucions de l'equació diferencial homogènia associada (la que obtenim quan  $\varepsilon = 0$ ) són de la forma  $A \cos(\omega t) + B \sin(\omega t)$ . També podem veure que la següent és una solució particular de l'equació diferencial que estem estudiant:

$$\frac{\varepsilon}{\omega^2 - 1} \sin(t).$$

Per tant, donat que si a les solucions de l'equació homogènia associada els hi sumem una solució particular de l'equació que estem estudiant obtenim totes les solucions d'aquesta, ens resulta que  $x'' + \omega^2 x = \varepsilon \sin(t)$  té solucions periòdiques de la forma:

$$A \cos(\omega t) + B \sin(\omega t) + \frac{\varepsilon}{\omega^2 - 1} \sin(t).$$

Ara bé, en el cas  $\omega = 1$ , tot i que es mantenen les solucions de l'equació homogènia associada ( $A \cos(t) + B \sin(t)$ ), canvia la solució particular. Aquest fet era esperable ja que  $\varepsilon/(\omega^2 - 1)$  està només ben definit en el cas  $\omega \neq 1$ . Per  $\omega = 1$ , una solució particular és:

$$\frac{-\varepsilon}{2} t \cos(t).$$

D'aquesta manera, la solució general per a  $\omega = 1$  no és periòdica:

$$A \cos(\omega t) + B \sin(\omega t) - \frac{\varepsilon}{2} t \cos(t).$$

Ens trobem així, pel que fa a les solucions periòdiques, que el sistema del pèndol pertorbat estudiat en la seva aproximació resultant de linealitzar-lo prop de l'origen, presenta una singularitat en  $\omega = 1$ . Mentre que per a altres valors de  $\omega$  totes les solucions són periòdiques, per a  $\omega = 1$  cap solució ho és.

Aquest fet ens convida a investigar quin comportament tenen les òrbites del sistema del pèndol pertorbat prop de l'origen i per a valors de  $\omega$  propers a 1.

## 4.2 Òrbites periòdiques

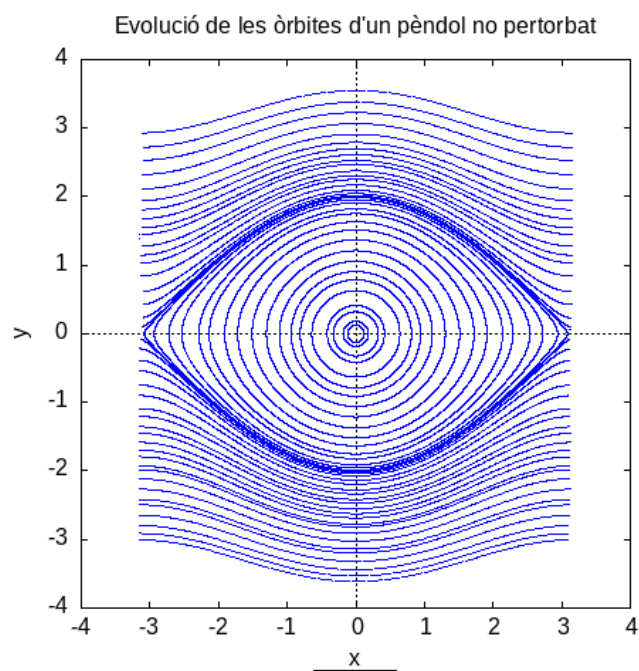


Figura 1

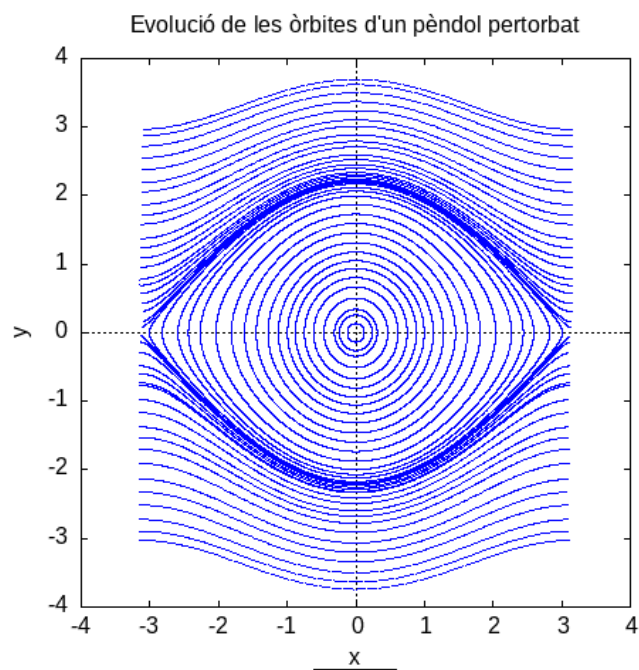


Figura 2

Si analitzem les dues imatges anteriors, les qual mostren, respectivament, les òrbites del sistema del pèndol sense pertorbacions ( $\omega = 1$  i  $\varepsilon = 0$ ) i les del sistema del pèndol pertorbat (per  $\omega = 1.1$  i  $\varepsilon = 0.01$ ) projectades al pla  $(x, y)$ , podem observar poques diferències. De fet, principalment, només podem apreciar el que sembla una lleugera dil·latació en la direcció  $y$  en la Figura 2 respecte la Figura 1, la qual és conseqüència d'introduir un valor  $\omega > 1$ .

Així doncs, aquests gràfics, així com l'estudi de l'evolució de les òrbites de diferents punts, no ens poden proporcionar informació sobre l'existència d'òrbites periòdiques en cada un dels casos.

Necessitem, per tant, una eina que ens permeti determinar si un punt pertany, o no, a una òrbita periòdica.

Fixem-nos que si considerem un punt  $p \in \mathbb{R}^2$ ,  $p = (x_0, y_0)$  i denotem  $\varphi(t; t_0, p)$  a la solució del problema de Cauchy associat al sistema de equacions diferencials del pèndol pertorbat amb condicions inicials  $t_0$  i  $p$ , aleshores  $p$  estarà en una òrbita periòdica de període  $2\pi$  si:

$$\varphi(t_0 + 2\pi; t_0, p) = p.$$

Ja que, si es verifica la condició anterior, passat el període de  $2\pi$  tornem a trobar-nos en les mateixes condicions inicials i, per tant, si seguim avançant en el temps tornarem a descriure la mateixa òrbita.

Definim ara la coneguda com **Aplicació Estroboscòpica** o **Aplicació de Poincaré**:

$$P : \Sigma \rightarrow \Sigma,$$

sent  $\Sigma$  un subespai transversal al flux associat al sistema que estem estudiant ([1, p.71]).

En el nostre cas, podem considerar  $\Sigma = \{(x, y, t) \in \mathbb{R}^3 | t = t_0\}$  i, degut a la periodicitat de  $\sin(t)$ , també es verifica que  $\Sigma = \{(x, y, t) \in \mathbb{R}^3 | t = t_0 + 2\pi\}$ .

Així, l'aplicació estroboscòpica és una aplicació que, donat un punt  $p \in \mathbb{R}^2$ , la imatge que ens retorna sigui un punt de l'òrbita de  $p$  que estigui a  $2\pi$  unitats de temps més avançat. És a dir:

$$P(p) = \varphi(t_0 + 2\pi; t_0, p).$$

**Observació 4.1.** El valor  $2\pi$  s'aplica en aquest cas ja que el sistema del pèndol pertorbat està format per funcions que tenen un període respecte  $t$  de  $2\pi$  ( $\sin(t)$ ). En un cas general, si les funcions del sistema tinguessin un període  $T$ , tindríem:  $P(p) = \varphi(t_0 + T; t_0, p)$ .

Ara, fixant-nos en la definició d'òrbita periòdica i de l'aplicació estroboscòpica, podem afrontar el problema de trobar òrbites periòdiques del pèndol pertorbat com el problema de trobat punts fixos de la aplicació estroboscòpica. Aquets punts fixos de  $P$  tindran una òrbita de període  $2\pi$ .

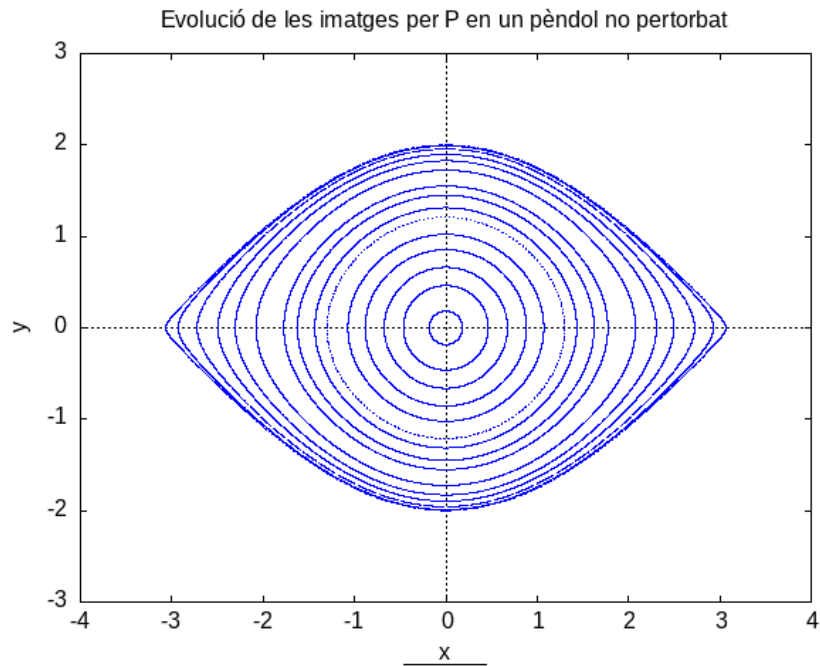


Figura 3

Si observem ara els gràfics resultants de representar imatges consecutives de l'aplicació estroboscòpica en el cas del pèndol sense pertorbar (Figura 3) i el pèndol pertorbat (Figura 4), sí s'aprecien clares diferències. Podem intuir l'aparició de dos nous punts fixos de  $P$  a l'introduir la pertorbació, situats sobre l'eix  $x = 0$ . Aquests dos gràfics corresponen a valors de  $\omega$  i  $\varepsilon$  respectivament iguals que en els gràfics de les Figures 1 i 2.

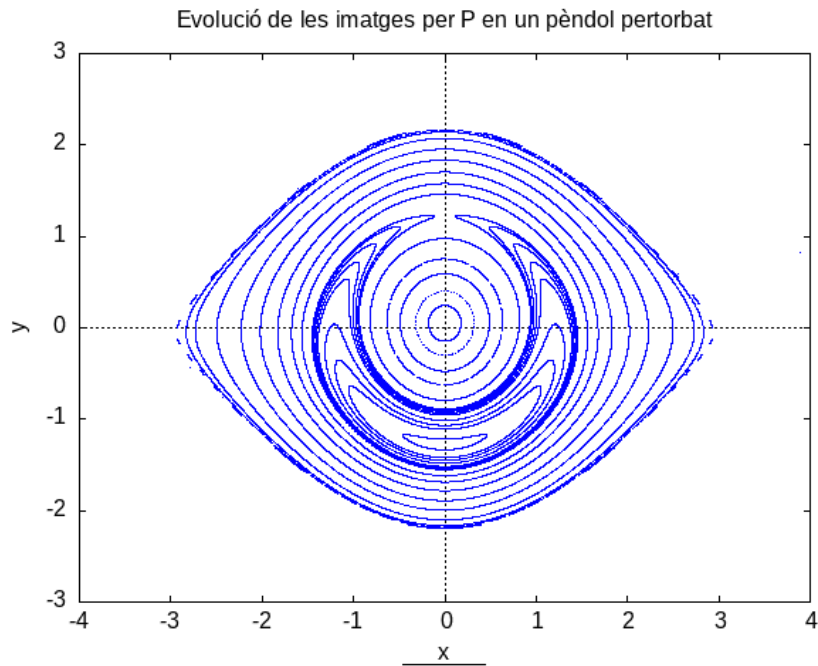


Figura 4

El nostre objectiu, ara, és determinar quins són aquests punts fixos. Per a fer-ho, aplicarem el mètode de Newton per a buscar zeros de la funció:

$$F(z) = P(z) - z.$$

Recordem que, per a aplicar el mètode de Newton, ens cal una aproximació inicial  $z^{(0)}$  a partir de la qual calcularem les següents aproximacions reiterant el procediment:

$$z^{(k+1)} = z^{(k)} + w.$$

Sent  $w$  solució de:

$$DF(z^{(k)})w = -F(z^{(k)}).$$

Necessitem, doncs, determinar  $DF(z)$ .

Partim de l'expressió de  $F(z)$  i obtenim que  $DF(z) = DP(z) - Id$ . Així, el que hem de trobar és  $DP(z)$ . Per a aconseguir-ho, farem ús dels següents resultats:

**Teorema 4.2** (Equacions variacionals). *Sigui  $f : \Omega \subset \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ , amb  $\Omega$  obert, una funció que depèn d'un paràmetre  $\lambda \in \mathbb{R}^m$ .*

*Suposem que  $f$  és contínua i  $\mathcal{C}^1(\Omega)$  respecte  $x \in \mathbb{R}^n$  i  $\lambda$ .*

*Aleshores,  $\varphi(t; t_0, x_0, \lambda)$  (solució del problema de Cauchy:  $x' = f(t, x, \lambda)$ ;  $x(t_0) = x_0$ ) és  $\mathcal{C}^1$  respecte totes les variables.*

**Proposició 4.3** (Equacions variacionals per la condició inicial  $x_0$ ). ([12, p.46]) *Sigui  $M(t) \in \mathbf{M}^{n \times n}$  la matriu de derivades parcials de  $\varphi(t)$  respecte la condició inicial  $x_0 \in \mathbb{R}^n$ .*

*Aquesta és solució del problema matricial:*

$$\begin{cases} X' = A(t)X, \\ X(t_0) = Id. \end{cases}$$

*On  $A(t) = D_x f(t, \varphi(t; t_0, x_0, \lambda), \lambda)$ .*

**Observació 4.4.** La matriu  $M(t)$  de la proposició anterior és la matriu fonamental principal del sistema  $x' = f(t, x, \lambda)$ .

Per al nostre cas, podem considerar que el sistema del pèndol pertorbat té la forma:

$$\begin{cases} x' = f_1(t, x, y) = y, \\ y' = f_2(t, x, y) = -\omega^2 \sin(x) + \varepsilon \sin(t). \end{cases}$$

Aleshores, la matriu que busquem,  $DP(z)$ , és la matriu  $M(t)$  de la proposició anterior avaluada en  $t_f = 2\pi$  i, per a trobar-la, necessitem la matriu  $A$ , la qual, segons la descripció de la proposició, és:

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega^2 \cos(x) & 0 \end{pmatrix}.$$

A partir d'aquesta forma de la matriu  $A$ , no podem obtenir directament la matriu  $DP(z)$ . Per a obtenir-la, ens caldrà fet ús del mètode de Taylor per a evolucionar de  $t_0 = 0$  fins a  $t_f = 2\pi$  el sistema:

$$\begin{cases} x' = f_1(t, x, y) = y, \\ y' = f_2(t, x, y) = -\omega^2 \sin(x) + \varepsilon \sin(t), \\ V' = AV, \\ x(t_0) = x_0, \\ y(t_0) = y_0, \\ V(t_0) = Id. \end{cases}$$

Podem reescriure el sistema anterior com a un sistema d'equacions diferencials de

primer ordre si considerem  $V = \begin{pmatrix} a(t) & b(t) \\ c(t) & d(t) \end{pmatrix}$ . Obtenint, d'aquesta manera, el sistema següent:

$$\begin{cases} x' = f_1(t, x, y) = y, \\ y' = f_2(t, x, y) = -\omega^2 \sin(x) + \varepsilon \sin(t), \\ a' = c, \\ b' = d, \\ c' = -\omega^2 \cos(x)a, \\ d' = -\omega^2 \cos(x)b, \\ x(t_0) = x_0, \\ y(t_0) = y_0, \\ a(t_0) = 1 = d(t_0), \\ b(t_0) = 0 = c(t_0). \end{cases}$$

Aleshores, si , com hem dit,  $t_0 = 0$  i  $z = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ , temin:

$$P(z) = \begin{pmatrix} x(2\pi) \\ y(2\pi) \end{pmatrix} \text{ i } DP(z) = \begin{pmatrix} a(2\pi) & b(2\pi) \\ c(2\pi) & d(2\pi) \end{pmatrix}.$$

Ja tenim, doncs, tot el necessari per a aplicar el mètode de Newton i trobar els punts fixos de l'aplicació estroboscòpica  $P(z)$ .

### 4.3 Estabilitat de les òrbites periòdiques

Una vegada hem trobat els punts fixos ( $z_0$ ) de l'aplicació estroboscòpica  $P(z)$ , els quals tenen òrbites periòdiques en el sistema del pèndol pertorbat, ens interessa conèixer quina és l'estabilitat d'aquestes òrbites trobades. Comencem presentant la definició d'òrbita estable.

**Definició 4.5** (Òrbita estable). ([12, p.315]) *L'òrbita d'un punt  $z_0$  (la qual denotarem per  $\gamma(z_0)$ ) s'anomena estable si, donat un entorn qualsevol  $U(\gamma(z_0))$  d'aquesta, existeix un altre entorn  $V(\gamma(z_0)) \subseteq U(\gamma(z_0))$  tal que qualsevol solució que comenci en  $V(\gamma(z_0))$ , es mantindrà en  $U(\gamma(z_0)) \forall t \geq 0$ .*

Per a estudiar l'estabilitat de les òrbites periòdiques, estudiarem primer l'estabilitat dels punts fixos de  $P(z)$ .

Sigui  $z_0$  un punt fix de  $P$ . Si fem una pertorbació ( $z_0 + h$ ), podem considerar la seva imatge per  $P$ :

$$P(z_0 + h) = P(z_0) + DP(z_0)h + \mathcal{O}_2(\|h\|).$$

Tenint en compte que  $P(z_0) = z_0$  i quedant-nos amb l'aproximació lineal, després de  $n$  iteracions tenim:



$$P^n(z_0 + h) = z_0 + DP(z_0)^n h.$$

D'aquí en traiem un criteri per a l'estabilitat de  $z_0$  com a punt fix de  $P$ :

- $z_0$  és un **punt estable** si tots els valors propis de  $DP(z_0)$  tenen mòdul menor o igual a 1.
- $z_0$  és un **punt inestable** si  $DP(z_0)$  té algun valor propi amb mòdul major a 1.

Analitzem, doncs, com són els valors propis de  $DP(z_0)$ .

**Teorema 4.6** (Fórmula de Liouville). ([12, p.83]) *Sigui  $M$  la matriu de solucions de  $x' = A(t)x$  per  $t \in I \subset \mathbb{R}$ . Aleshores,  $\forall t \in I$ :*

$$\det(M(t)) = \det(M(t_0)) \exp \int_{t_0}^t \text{tr}(A(s)) ds.$$

A l'aplicar la Fórmula de Liouville al sistema d'equacions per a trobar  $DP(z_0)$  (el que ens resulta de les equacions variacionals), ens trobem que  $\text{tr}(A(s)) = 0$ . Recordem que:

$$A = \begin{pmatrix} 0 & 1 \\ -\omega^2 \cos(x) & 0 \end{pmatrix}.$$

Per tant, tenim:

$$\det V(2\pi) = \det V(0),$$

↓

$$\det DP(z_0) = \det Id = 1.$$

Tenint en compte que el producte dels valors propis és igual al determinant, en el nostre cas se'ns presenten dues possibilitats:

1. Els valors propis són reals:

En aquest cas, l'un és invers de l'altre:  $\lambda_1 = \frac{1}{\lambda_2}$ . Si no són iguals a 1, aquest és el cas inestable, ja que un dels dos tindrà mòdul major a 1.

2. Els valors propis són complexos conjugats:

En aquest cas:  $\lambda_1 = a + bi$  i  $\lambda_2 = a - bi$ , per tant:

$$\lambda_1 \lambda_2 = 1, \quad \rightarrow \quad (a + bi)(a - bi) = 1, \quad \rightarrow \quad a^2 + b^2 = 1.$$

Els dos valors propis tenen mòdul igual a 1. És, per tant, un cas estable.

Veiem ara com el Teorema de Floquet (Teorema 2.4) presentat en la secció 2 ens serà de gran utilitat per a relacionar l'estabilitat de les òrbites amb els valors propis de la matriu  $DP(z_0)$ .

El sistema de les variacionals del pèndol pertorbat està format per equacions amb període  $2\pi$ . Això és així degut a que  $x(t)$  i, per tant, també la matriu del sistema  $A(t)$ , tenen aquesta periodicitat, en estar estudiant-ho en una òrbita periòdica. Aleshores, per el Corol·lari 2.5, sabem que podem transformar biunívocament, a través d'una matriu  $Q(t)$   $2\pi$ -periòdica, les solucions d'aquest sistema a les d'un sistema lineal a coeficients constants definit per una matriu  $B$ :  $\mu' = B\mu$ .

El Teorema de Floquet ens dona una relació entre aquesta matriu  $B$  i la matriu fonamental del sistema del pèndol pertorbat,  $\phi(t)$ :

$$\phi(t) = Q(t) \cdot e^{Bt}.$$

Sabem que  $\phi(2\pi) = DP(z_0)$ , així que, per a relacionar les matrius  $B$  i  $DP(z_0)$ , ens cal conèixer  $Q(2\pi)$ . Per a fer-ho, partim de la proposició 2.3:

$$\phi(t + 2\pi) = \phi(t)C,$$

per una matriu  $C$  no singular. D'aquí podem deduir:

$$\phi(0 + 2\pi) = \phi(2\pi) = \phi(0)C = IdC = C.$$

Ara, reordenem l'equació del Teorema de Floquet i l'avaluem a  $2\pi$ :

$$Q(t) = \phi(t) \cdot e^{-Bt},$$

↓

$$Q(2\pi) = \phi(2\pi) \cdot e^{-B2\pi} = C \cdot e^{-B2\pi} = e^{B2\pi} \cdot e^{-B2\pi} = Id.$$

Ja que, recordem que en la demostració del Teorema de Floquet estipulem que  $C = e^{B2\pi}$ . Així doncs, hem trobat la relació entre les matrius  $B$  i  $DP(z_0)$ :

$$e^{B2\pi} = DP(z_0).$$

D'aquí n'extraiem que:

$$\lambda \text{ vap de } DP(z_0), \quad \rightarrow \quad \frac{\log(\lambda)}{2\pi} \text{ vap de } B.$$

Podem concloure, doncs, que un valor propi de  $DP(z_0)$  de mòdul menor a 1 correspon a un valor propi amb part real negativa de  $B$  i un valor propi de  $DP(z_0)$  de mòdul major a 1 correspon a un valor propi amb part real positiva de  $B$ .

Tornant als dos casos possibles que teníem de valors propis de  $DP(z_0)$  i aplicant el resultat anterior, obtenim les dues possibilitats següents:

1. Si els valors propis de  $DP(z_0)$  són reals, la matriu  $B$  tindrà un valor propi amb part real positiva i un amb part real negativa, és el cas inestable.
2. Si els valors propis de  $DP(z_0)$  són complexos conjugats, els valors propis de la matriu  $B$  tindran part real 0, és el cas estable.

Podem concloure, per tant, que el criteri d'estabilitat dels punts fixos de  $DP(z_0)$  coincideix amb el criteri d'estabilitat en el sistema de les variacionals i, consegüentment, amb el criteri d'estabilitat de les òrbites periòdiques.

#### 4.4 Mètode de continuació respecte $\omega$

El sistema d'equacions diferencials ordinàries de primer ordre que estem estudiant:

$$\begin{cases} x' = y, \\ y' = -\omega^2 \sin(x) + \varepsilon \sin(t), \end{cases}$$

presenta una clara dependència de dos paràmetres ( $\varepsilon$  i  $\omega$ ). Aquest fet ens permet, una vegada trobada una òrbita periòdica per a un valor donat de  $\varepsilon$  i  $\omega$ , aplicar un mètode de continuació respecte un dels paràmetres per a veure com es modifiquen aquestes òrbites respecte la modificació del paràmetre.

El mètode de continuació que aplicarem és el **mètode predictor-corrector amb pseudo-paràmetre arc** ([11, p.63]) i l'aplicarem respecte el paràmetre  $\omega$ .

El que estem buscant és una corba en l'espai  $\mathbb{R}^3$  parametritzada amb el paràmetre arc  $s$  tal que  $c(s) = (x, y, \omega)$  compleixi que  $(x, y)$  dona una òrbita periòdica del sistema del pèndol pertorbat per al valor  $\omega$  corresponent, amb un valor de  $\varepsilon$  fixat. Aleshores, el mètode predictor-corrector amb pseudo-paràmetre arc consisteix en el següent:

1. Predicció.

Signi  $c^0 = c(s_0)$  el punt de partida (una òrbita periòdica ja coneguda) i  $v = c'(s_0)$  el vector tangent a la corba en  $c^0$ . Aleshores, donat un pas  $h$ , la predicció serà:  $c^{(1,0)} = c^0 \pm hv$ .

2. Correcció.

Partint del punt  $c^{(1,0)}$  i, aplicant el mètode de Newton, busquem un punt que verifiqui:

- a) Ser un punt de la corba.

b) Estar a distància  $h$  de  $c^0$ .

Abans d'aplicar el mètode de continuació descrit, com ens cal partir d'una òrbita periòdica coneguda, haurem d'aplicar el mètode de Newton per a trobar-la, partint, tal i com s'ha explicat, d'una aproximació inicial.

Una vegada coneguda una òrbita periòdica, analitzem com implementem la predicció. Donada la funció:

$$F(z) = P(z) - z,$$

la qual ja hem utilitzat en el mètode de Newton per a trobar el punt fix, ens n'adonem que derivant l'expressió  $F(c(s_0)) = 0$ , obtenim:

$$DF(c(s_0)) \cdot c'(s_0) = 0,$$

on  $DF(c(s_0))$  és una matriu  $2 \times 3$  la qual conté, per columnes, la diferencial de  $F$  respecte  $x$ ,  $y$ , i  $w$ . Les dues primeres columnes ja hem vist que són iguals a  $DP(z) - Id$  i que les hem de calcular fent ús de les equacions variacionals per a la condició inicial.

Per a calcular l'última columna, necessitarem fer ús de la proposició següent:

**Proposició 4.7** (Equacions variacionals per al paràmetre  $\lambda \in \mathbb{R}^m$ ). *Sigui  $N(t) \in \mathbf{M}^{n \times m}$  la matriu de derivades parcials de  $\varphi(t)$  respecte al paràmetre  $\lambda \in \mathbb{R}^m$ .*

*Aquesta és solució del sistema no homogeni:*

$$\begin{cases} X' = A(t)X + B(t), \\ X(t_0) = 0. \end{cases}$$

On  $A(t) = D_x f(t, \varphi(t; t_0, x_0, \lambda), \lambda)$  i  $B(t) = D_\lambda f(t, \varphi(t; t_0, x_0, \lambda), \lambda)$ .

En el nostre cas,  $m = 1$ . El sistema que tenim és:

$$\begin{pmatrix} e(t) \\ f(t) \end{pmatrix}' = A \begin{pmatrix} e(t) \\ f(t) \end{pmatrix} + \begin{pmatrix} 0 \\ -2\omega \sin(x) \end{pmatrix},$$

$$\begin{pmatrix} e(t) \\ f(t) \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -\omega^2 \cos(x) & 0 \end{pmatrix} \begin{pmatrix} e(t) \\ f(t) \end{pmatrix} + \begin{pmatrix} 0 \\ -2\omega \sin(x) \end{pmatrix},$$

$$\begin{pmatrix} e(t) \\ f(t) \end{pmatrix}' = \begin{pmatrix} f(t) \\ -\omega^2 \cos(x)e(t) - 2\omega \sin(x) \end{pmatrix}.$$

Amb la condició inicial  $\begin{pmatrix} e(t_0) \\ f(t_0) \end{pmatrix}' = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ .

Aquestes equacions les inclourem al sistema que teníem per a calcular  $P(z)$  i  $DP(z)$  i al qual apliquem el mètode de Taylor. Així doncs, ara haurem de treballar amb el següent sistema:

$$\left\{ \begin{array}{l} x' = f_1(t, x, y) = y, \\ y' = f_2(t, x, y) = -\omega^2 \sin(x) + \varepsilon \sin(t), \\ a' = c, \\ b' = d, \\ c' = -\omega^2 \cos(x)a, \\ d' = -\omega^2 \cos(x)b, \\ e' = f, \\ f' = -\omega^2 \cos(x)e - 2\omega \sin(x), \\ x(t_0) = x_0, \\ y(t_0) = y_0, \\ a(t_0) = 1 = d(t_0), \\ b(t_0) = 0 = c(t_0), \\ e(t_0) = 0 = f(t_0). \end{array} \right.$$

A partir del qual, si  $t_0 = 0$  i tant la condició inicial per a  $x$  i  $y$  com el valor de  $\omega$  corresponen a la imatge de  $s_0$  per  $c$  (corresponen a una òrbita periòdica, al ser un punt de la corba que estem buscant), obtenim:

$$DF(c(s_0)) = \begin{pmatrix} a(2\pi) & b(2\pi) & e(2\pi) \\ c(2\pi) & d(2\pi) & f(2\pi) \end{pmatrix}.$$

Ara, tornant a l'expressió  $DF(c(s_0)) \cdot c'(s_0) = 0$ , podem veure que si  $DF(c(s_0))$  té rang 2, podem calcular  $c'(s_0)$  fent el producte vectorial de les dues files de  $DF(c(s_0))$ . Per a fer la predicció, ens caldrà normalitzar-lo i multiplicar-lo per el pas  $h$  escollit.

Estudiem ara com realitzar la correcció. Com ja hem comentat, consisteix en aplicar el mètode de Newton a:

$$\left\{ \begin{array}{l} F(z) = 0, \\ \|z - c^0\|_2^2 - h^2 = 0. \end{array} \right.$$

Sent  $z = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ . Per la qual cosa tornarem a necessitar l'anterior càlcul de  $DF(z)$ .

Com hem dit, aquest mètode de continuació ens permet anar trobant nous punts  $(x_0, y_0)$  que ens donen òrbites periòdiques per al respectiu valor de  $\omega$  que anem variant. Una vegada trobades aquestes òrbites, les podem estudiar amb els criteris d'estabilitat presentats en el punt anterior.

Per al cas particular de  $\varepsilon = 0.01$ , si realitzem dues vegades el mètode de de continuació ja descrit, partint de  $z_0 = (0, 0)$  com a primera aproximació per a trobar la primera òrbita

en els dos casos, per a valors inicials de  $\omega = \sqrt{2}, \frac{1}{\sqrt{2}}$  respectivament i dirigint-nos en els dos casos cap a valors de  $\omega$  més propers a 1, obtindrem els resultats presentats en el gràfic posterior (Figura 5). Per obtenir-lo, hem utilitzau un valor de  $h = 0.001$  i hem realitzat  $N = 2000$  repeticions del mètode predictor-corrector.

Cal notar que en aquest gràfic només hi són representats els valors de  $y_0$  que ens donen una òrbita periòdica per al valor de  $\omega$  corresponent. Pel que fa al valor de  $x_0$ , aquest es manté constant i igual a 0.

També cal destacar que el nombre d'òrbites periòdiques no es manté constant, mentre que per a valors de  $\omega$  inferiors a 1 només en tenim una i és estable, a mesura que anem a valors de  $\omega$  més grans ens trobem amb una bifurcació del tipus sella-node: apareixen dues òrbites periòdiques més, una d'estable i una d'instable.

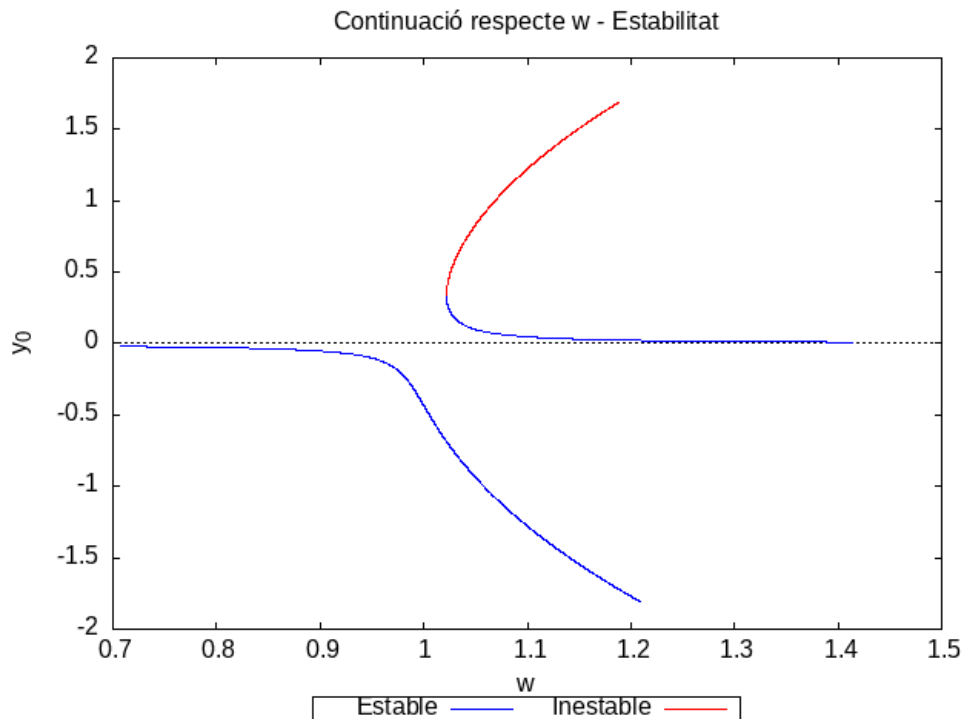
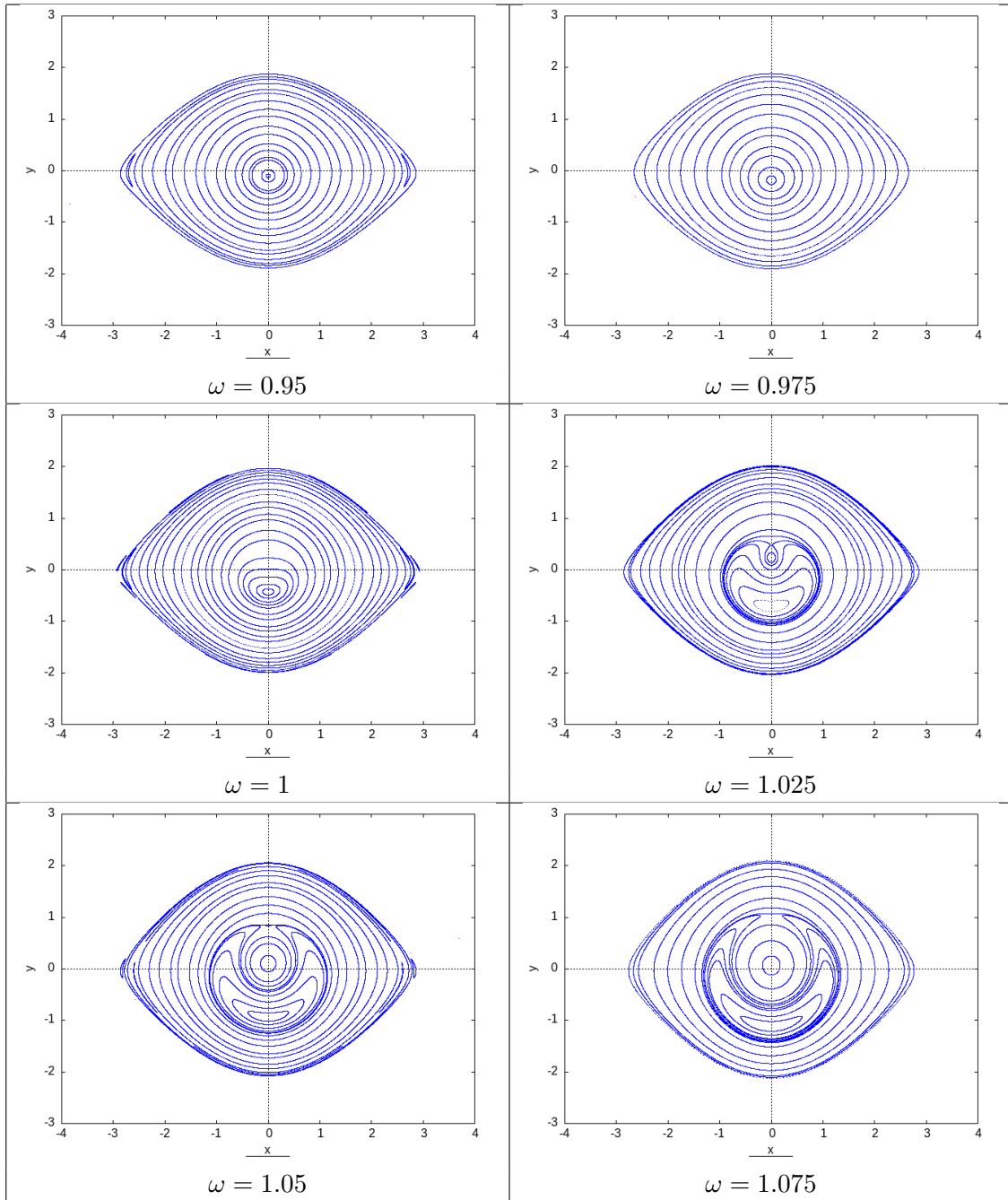


Figura 5

En la taula de la pàgina següent (Taula 1) es pot apreciar, mitjançant gràfics que mostren l'evolució de les imatges per l'aplicació estroboscòpica, com es modifiquen i apareixen nous punts fixos, seguint el que hem vist a la Figura 5.

Podem observar com el punt fix de la primera gràfica, evoluciona cap a valors de  $y_0$  cada vegada més baixos, situant-se per sota dels nous dos punts fixos que apareixen, tots sobre l'eix  $x = 0$ . També veiem com el punt fix estable que apareix evoluciona cap a tenir un valor de  $y_0$  proper a 0 i com el punt fix inestable el trobem a valors de  $y_0$  creixents.



Taula 1

També, si modifiquem el valor de  $\varepsilon$ , apropant-lo més a 0, podem observar com es veu afectat el gràfic presentat en la Figura 5. Tenim els resultats a la pàgina següent. Els dos gràfics s'han fet amb els mateixos paràmetres que el de la Figura 5, modificant únicament el valor de  $\varepsilon$ . Obtenim així el de la Figura 6 per  $\varepsilon = 0.0025$  i el de la Figura 7 per  $\varepsilon = 0.0005$ .

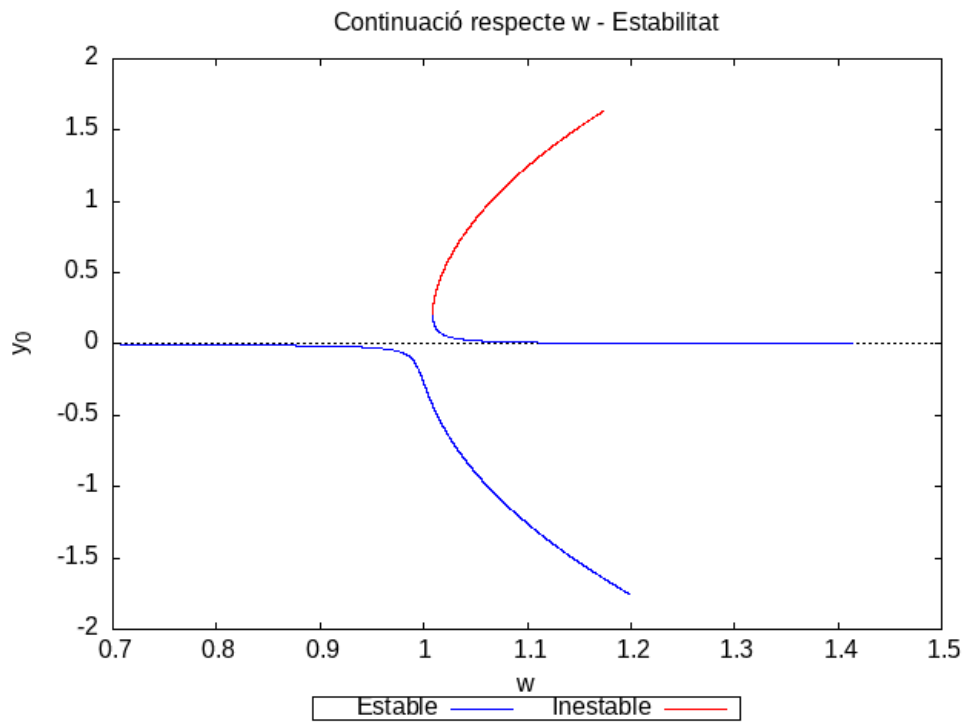


Figura 6

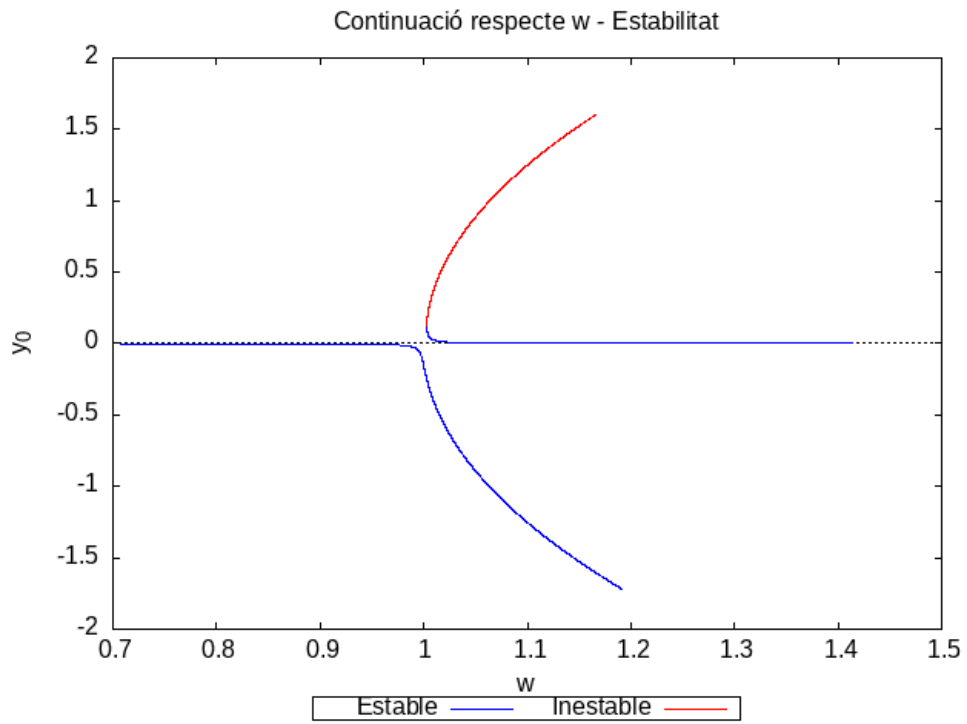


Figura 7



Aquests resultats plasmats en els gràfics de les Figures 5, 6 i 7, a més, concorden amb el següent teorema:

**Teorema 4.8.** *Sigui la funció  $F(z) = P(z) - z$ , sent  $P(z)$  l'aplicació estroboscòpica associada al sistema del pèndol pertorbat (dependent de  $\omega$  i  $\varepsilon$ ).*

*Si  $\omega \neq 1$  i  $\varepsilon$  és prou petit, aquesta funció  $F(z)$  té una única solució a prop de  $z = (0, 0)$ .*

*Demostració:*

Durant aquesta demostració, per emfatitzar la dependència de  $F(z)$  del paràmetre  $\varepsilon$ , la denotarem per  $F_\varepsilon(z)$ .

Sabem que si  $\varepsilon = 0$ ,  $z_0 = (0, 0)$  és solució de  $F_0(z)$ . El que volem és intentar continuar aquesta solució.

Per a fer-ho, utilitzarem el teorema de la funció implícita, el qual podem aplicar partint de les condicions anteriors. Aquest teorema ens implica que si  $D_z F(z_0)$  és regular, aleshores  $\forall \varepsilon$  tal que  $|\varepsilon - 0| < tol$  existeix un  $z_\varepsilon$  el qual és solució de  $F_\varepsilon(z)$ .

Aleshores, per a demostrar el teorema, només ens falta veure que  $D_z F(z_0)$  és regular.

Ja hem vist anteriorment que  $DF(z) = DP(z) - Id$ . Per saber si és regular, doncs, podem analitzar quins són els valors propis de  $DP(z_0)$ . Recordem que  $DP(z_0)$  l'obtenim del sistema de les variacionals, el qual en  $z_0 = (0, 0)$  presenta la següent forma:

$$V' = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} V.$$

Hem vist que  $DP(z_0)$  serà la matriu fonamental principal del sistema avaluada en  $2\pi$ . Per tant:

$$DP(z_0) = \exp \left[ \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} (2\pi) \right].$$

Els seus valors propis seràn, per tant,  $\{e^{2\pi\omega i}, e^{-2\pi\omega i}\}$  i, com estem suposant el cas  $\omega \neq 1$ , aquests valors propis no seràn iguals a 1. D'aquí en traiem que  $DF(z_0)$  no tindrà valors propis nuls, sent, per tant, una matriu regular.

□

## 5 Intersecció de les separatius

Considerem, ara, un pèndol pertorbat amb la següent equació:

$$x'' + \sin(x) = \mu \sin\left(\frac{t}{\varepsilon}\right),$$

la qual podem reescriure en forma de sistema d'equacions diferencials ordinàries de primer ordre:

$$\begin{cases} x' = y, \\ y' = -\sin(x) + \mu \sin\left(\frac{t}{\varepsilon}\right). \end{cases}$$

Per a poder seguir un procediment anàleg a l'exposat en la secció anterior per a localitzar òrbites periòdiques i estudiar-ne la seva estabilitat, hem de tenir en compte una sèrie de qüestions.

Per començar, aquest sistema no té una periodicitat de  $2\pi$ , sinó de  $2\pi\varepsilon$ .

A l'hora de calcular les variacionals respecte la condició inicial, necessàries per a l'estudi de l'estabilitat, ens trobem amb una lleugera modificació del sistema a solucionar. En aquest cas:

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\cos(x) & 0 \end{pmatrix}.$$

Així doncs, el sistema a resoldre és:

$$\begin{cases} x' = y, \\ y' = -\sin(x) + \mu \sin\left(\frac{t}{\varepsilon}\right), \\ a' = c, \\ b' = d, \\ c' = -\cos(x)a, \\ d' = -\cos(x)b, \\ x(t_0) = x_0, \\ y(t_0) = y_0, \\ a(t_0) = 1 = d(t_0), \\ b(t_0) = 0 = c(t_0). \end{cases}$$

Obtenim així el mateix sistema per a les variacionals que en l'equació anterior, però amb  $\omega = 1$ .

I, si  $t_0 = 0$  i  $z = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$ , temin:

$$P(z) = \begin{pmatrix} x(2\pi\varepsilon) \\ y(2\pi\varepsilon) \end{pmatrix} \text{ i } DP(z) = \begin{pmatrix} a(2\pi\varepsilon) & b(2\pi\varepsilon) \\ c(2\pi\varepsilon) & d(2\pi\varepsilon) \end{pmatrix}.$$

L'equació del pèndol pertorbat que estem estudiant es converteix en l'equació del pèndol sense pertorbar en el cas  $\mu = 0$ . Recordem que el sistema del pèndol sense pertorbar:

$$\begin{cases} x' = y, \\ y' = -\sin(x), \end{cases}$$

té un punt fix en  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ,  $\begin{pmatrix} -\pi \\ 0 \end{pmatrix}$  i  $\begin{pmatrix} \pi \\ 0 \end{pmatrix}$  els qual són un centre i dos punts de sella respectivament. També hi ha dues separatius que uneixen els dos punts de sella.

Per a valors prou petits del paràmetre  $\mu$ , el sistema del pèndol pertorbat que estem estudiant té un punt  $(z_0)$  amb òrbita periòdica inestable el qual podem trobar partint de la condició inicial  $z = \begin{pmatrix} -\pi \\ 0 \end{pmatrix}$ . En aquest punt,  $DP(z_0)$  té, tal i com s'ha explicat, dos valors propis reals sent l'un l'invers de l'altre. Tenim, doncs, dues varietats, una d'estable i una d'inestable, al voltant d'aquest punt.

El que ens interessa estudiar és l'angle d'intersecció de la varietat inestable d'aquest punt amb la varietat estable d'un punt amb òrbita periòdica inestable proper a  $z = \begin{pmatrix} \pi \\ 0 \end{pmatrix}$ .

Recordem la definició de varietat estable:

**Definició 5.1** (Varietat estable de  $z_0$ ). *Conjunt de punts  $z \in \mathbb{R}^2$  tals que:*

$$\lim_{n \rightarrow \infty} P^n(z) = z_0.$$

I la definició anàloga de varietat inestable:

**Definició 5.2** (Varietat inestable de  $z_0$ ). *Conjunt de punts  $z \in \mathbb{R}^2$  tals que:*

$$\lim_{n \rightarrow -\infty} P^n(z) = z_0.$$

Com hem comentat, per  $\mu = 0$  hi ha una separatiu contínua, però a l'introduir la pertorbació, aquesta també es veu afectada i, com es manté la simetria del sistema (el sistema es manté invariant amb el canvi  $x \rightarrow -x$ ,  $t \rightarrow -t$ ), les dues varietats s'intersequen en l'eix  $x = 0$ .

El que farem a continuació serà calcular l'angle de tall entre aquestes dues varietats. Com a referència, tenim la següent fórmula pel valor d'aquest angle:

**Teorema 5.3.** ([2, p.433]) *Per a  $\varepsilon > 0$ ,  $\varepsilon \rightarrow 0$  i  $\mu \rightarrow 0$ , considerem la següent fórmula per al valor de l'angle entre les separatius:*

$$\alpha = \frac{\pi}{2\varepsilon} \frac{\mu}{\cosh\left(\frac{\pi}{2\varepsilon}\right)} [1 + \mathcal{O}(\mu, \varepsilon^2)].$$

## 5.1 Aproximació lineal

Per a determinar numèricament aquest angle, el primer que necessitem és trobar un **interval fonamental** de la varietat inestable, és a dir, un conjunt de punts a partir dels quals i mitjançant iteracions de l'aplicació estroboscòpica podem obtenir tots els punts de la varietat.

L'ideal seria conèixer un punt  $q$  de la varietat proper al punt fix, així calcularíem  $P(q)$ , que també formaria part de la varietat. Podríem considerar, per tant, tots els punts pertanyents a la varietat i situats entre  $q$  i  $P(q)$  com a interval fonamental, ja que les seves imatges successives per l'aplicació estroboscòpica ens donarien tota la varietat. El problema, però, és que no coneixem a priori cap punt d'aquesta varietat. El que farem, doncs, serà descriure dos procediments que ens permetran aproximar un punt  $q$  de la varietat inestable així com un conjunt de punts pertanyents a l'interval fonamental.

Un primer mètode per a determinar aquest interval consisteix en aprofitar l'aproximació lineal de la varietat que ens dona el vector propi associat al valor propi de mòdul més gran que 1 (inestable). Sigui  $z_0$  el punt fix i siguin  $\lambda_1$  i  $v_1$  el valor i vector propi inestable, donat un  $\delta > 0$  calculem el punt  $q$ :

$$q = z_0 + \delta v_1.$$

A partir d'aquest punt calculem els següents dos punts:

$$P(q) \quad \text{i} \quad q' = z_0 + \lambda_1 \delta v_1.$$

L'objectiu és trobar una  $\delta$  prou petita tal que es verifiqui:

$$\|P(q) - q'\|_2 < tol,$$

per una tolerància ( $tol$ ) fixada.

Si això es compleix, podem considerar  $q' \approx P(q)$  i, per tant, el segment entre  $q$  i  $q'$  ens serveix com a interval fonamental de la varietat.

Una vegada tenim un interval fonamental de la varietat, ens cal trobar quina és la potència ( $pot$ ) de l'aplicació estroboscòpica ( $P$ ) que trasllada aquest interval proper al punt fix a un interval que interseca l'eix  $x = 0$ . Com hem dit, l'interval és el segment entre  $q$  i  $q'$ . Per tant, per a trobar aquesta potència és suficient anar calculant imatges consecutives de  $q'$  per l'aplicació estroboscòpica i veure quina és la primera vegada que la component  $x$  de la imatge pren un valor positiu. El nombre d'imatges que haguem calculat serà el valor de la potència que buscàvem.

Ara sabem que, aplicant  $P^{pot}$  a l'interval, hi ha punts que tenen imatge amb component  $x$  tant positiva com negativa. Per a trobar l'angle entre les varietats, necessitem trobar el punt  $q_0$  de l'interval fonamental que verifiqui que  $P^{pot}(q_0)$  tingui component  $x$  nul·la.

Per a obtenir una primera aproximació, comencem amb una partició de l'interval amb  $N$  punts:

$$q_i = q + \lambda_1 \delta v_1 \frac{i}{N} \quad \forall i \in \{0, 1, \dots, N\}.$$

I busquem d'entre aquest punt  $q_i$  el que tingui la imatge per  $P^{pot}$  amb component  $x$  més propera a 0 pels positius i el que la tingui més propera a 0 pels negatius. Per a fer-ho, podem anar calculant  $P^{pot}(q_i)$  per  $i = 0, 1, 2, 3, \dots$  i, quan detectem un canvi de signe en el valor de la component  $x$  ens quedem amb aquell punt i l'anterior, el qual tindrà imatge amb component  $x$  negativa. Anomenem  $a$  i  $b$  a aquests dos punts.

Per a millorar aquesta aproximació, apliquem el mètode de la secant, buscant el valor del paràmetre  $\gamma$  tal que  $P^{pot}(a(1-\gamma) + b\gamma)$  tingui component  $x$  nul·la. D'aquesta manera tenim una aproximació molt més refinada del punt  $q_0$ .

Hem arribat, doncs, a trobar un punt de la varietat amb component  $x$  nul·la:  $P^{pot}(q_0)$ . Per a trobar l'angle de tall entre les varietats, necessitem el vector tangent a la varietat en  $P^{pot}(q_0)$ . Per a obtenir-lo, partim del vector tangent a la varietat en  $q_0$ , el qual podem aproximar al vector propi  $v_1$  o, equivalentment, al vector entre els punts  $a$  i  $b$  que hem utilitzat com a condicions inicials del mètode de la secant.

Amb aquest vector tangent en  $q_0$  podem calcular el vector tangent en  $P^{pot}(q_0)$ , al multiplicar-lo per  $DP^{pot}(q_0)$ . Amb el vector tangent a  $P^{pot}(q_0)$  ens és fàcil calcular l'angle de tall entre les varietats el qual, per simetria, serà el doble de l'angle que aquest vector tangent forma amb l'horitzontal.

## 5.2 Parametrització de la varietat

Un procediment anàleg per a calcular l'angle d'intersecció entre les varietats consisteix en partir d'un interval fonamental no basat en l'aproximació lineal de la varietat al voltant del punt  $z_0$ , sinó en una aproximació en sèrie de potències d'ordre superior. Per a poder aplicar-lo, ens cal una manera de determinar aquesta nova aproximació.

**El mètode de la parametrització** ([4, p.9]) ens permetrà trobar-la:

Considerem  $K(u) = a_0 + a_1u + a_2u^2 + a_3u^3 + \dots$  la parametrització de la varietat. Anem a determinar els coeficients  $a_i$  d'aquesta parametrització.

Com  $K(0) = z_0$ , tenim  $a_0 = z_0$ .

També, com  $a_0 + a_1u$  correspon a l'aproximació lineal, tenim  $a_1 = v_1$ , vector propi unitari de valor propi  $\lambda_1$ .

La resta dels coeficients de la parametrització queden determinats en imposar la següent propietat (o condició d'invariància):

$$P(K(u)) = K(\lambda_1 u).$$

Si definim  $K_j = \sum_{i=0}^j a_i u^i$  i suposem coneguts tots els coeficients  $a_i \forall i = 0, 1, 2, \dots, j$ , podem determinar  $a_{j+1}$ :

$$\begin{aligned}
P(K_j(u) + a_{j+1}u^{j+1}) &= P(K_j(u)) + DP(K_j(u))(a_{j+1}u^{j+1}) + \mathcal{O}(u^{j+2}) = \\
&= P(K_j(u)) + DP(a_0 + \sum_{i=1}^j a_i u^i)(a_{j+1}u^{j+1}) + \mathcal{O}(u^{j+2}) = \\
&= P(K_j(u)) + DP(z_0)(a_{j+1}u^{j+1}) + \mathcal{O}(u^{j+2}).
\end{aligned}$$

Tenint en compte que  $P(K_j(u)) = a_0 + \lambda_1 a_1 u + \lambda_1^2 a_2 u^2 + \dots + \lambda_1^j a_j u^j + b_{j+1} u^{j+1} + \mathcal{O}(u^{j+2})$ , obtenim:

$$P(K_j(u) + a_{j+1}u^{j+1}) = \sum_{i=0}^j \lambda_1^i a_i u^i + b_{j+1} u^{j+1} + DP(z_0)(a_{j+1}u^{j+1}) + \mathcal{O}(u^{j+2}).$$

Ara imposem,  $P(K_j(u) + a_{j+1}u^{j+1}) = a_0 + \lambda_1 a_1 u + \lambda_1^2 a_2 u^2 + \dots + \lambda_1^{j+1} a_{j+1} u^{j+1} + \mathcal{O}(u^{j+2})$  (la condició d'invariància).

Igualant els coeficients de  $u^{j+1}$  obtenim:

$$b_{j+1} + DP(z_0)a_{j+1} = \lambda_1^{j+1} a_{j+1}.$$

Així doncs, podem trobar el coeficient  $a_{j+1}$  resolent:

$$(DP(z_0) - \lambda_1^{j+1} Id)a_{j+1} = -b_{j+1}.$$

En aquest sistema,  $DP(z_0)$  i  $\lambda_1$  són elements coneguts. Pel que fa a  $b_{j+1}$ , aquest és el coeficient de grau  $j + 1$  del desenvolupament en sèrie de Taylor de  $P(K_j(u))$  i el podem obtenir amb el *jet transport* ([4, p.5]). A la pràctica, aquest coeficient el calcularem amb el programari descrit en [5].

Una vegada tenim parametritzada la varietat al voltant de  $z_0$  (fins a un cert ordre  $k$ ), podem procedir a trobar un interval fonamental.

Comencem buscant un valor de  $u$  tal que  $\|a_k u^k\| \leq 10^{-m}$ , per un  $m \in \mathbb{N}$ :

$$\|a_k u^k\| \leq 10^{-m}, \quad \rightarrow \quad \|a_k\| |u|^k \leq 10^{-m}, \quad \rightarrow \quad \log_{10} \|a_k\| + k \log_{10} |u| \leq -m.$$

Per tant,

$$|u| \leq 10^{\frac{-m - \log_{10} \|a_k\|}{k}} = \eta.$$

A la pràctica, els coeficients  $a_k$  amb  $k$  senar i  $k$  parell tenen un decaïment diferent. Per evitar tenir problemes de precisió, demanarem la condició anterior per a  $k$  i  $k - 1$  i, dels dos valors de  $\eta$  obtinguts, ens quedarem amb el més petit.

Ara, sigui  $u_0 = \frac{\eta}{\lambda_1}$ , es verifica que  $\|P(K_k(u_0)) - K_k(\lambda_1 u_0)\|$  és suficientment petit com per a poder considerar els punts  $K_k(u)$  amb  $u \in [u_0, \eta]$  interval fonamental de la varietat.

Una vegada determinat aquest interval fonamental, procedim de manera anàloga a l'explicat anteriorment.

Comencem buscant la potència (*pot*) de  $P$  que ens permetrà traslladar aquest interval fins a intersecar l'eix  $x = 0$ . Ho fem, com s'ha explicat anteriorment, analitzant el signe de la component  $x$  de la imatge de  $K_k(\eta)$  per diferents iteracions de l'aplicació estroboscòpica.

Per a trobar el punt de la corba que té component  $x = 0$ , comencem amb una partició de l'interval fonamental. En aquest cas, fem la partició de  $N$  punts a  $[u_0, \eta]$ , la qual quedarà reflectida en l'interval en aplicar  $K_k$ :

$$K_k(u_i) = K_k\left(u_0 + \eta - u_0\right) \frac{i}{N} \quad \forall i \in 0, 1, \dots, N.$$

Una vegada determinats dos punts  $a$  i  $b$  ( $a = u_{i-1}$ ,  $b = u_i$ ) amb imatges per  $P^{pot} \circ K_k$  amb component  $x$  negativa i positiva, apliquem el mètode de la secant per a trobar el valor ( $\gamma_0$ ) del paràmetre  $\gamma$  tal que  $P^{pot}(K_k(a(1 - \gamma) + b\gamma))$  tingui component  $x$  nul·la.

Per a trobar el vector tangent en  $P^{pot}(K_k(a(1 - \gamma_0) + b\gamma_0))$ , hem de partir del vector tangent en  $K_k(a(1 - \gamma_0) + b\gamma_0)$ . Aquest, donat que  $K(u)$  és una sèrie de potències, el calculem derivant aquesta sèrie:

$$K'_k(u) = \sum_{i=1}^j i a_i u^{i-1}.$$

Com hem fet abans, el traslladarem multiplicant-lo per  $DP^{pot}(K_k(a(1 - \gamma_0) + b\gamma_0))$  i així ja podrem calcular l'angle d'intersecció entre les dues varietats.

### 5.3 Resultats

Aquests són els resultats d'implementar els mètodes descrits en els apartats anteriors.

Comencem amb **l'aproximació lineal de la varietat**.

Partim d'una aproximació inicial del punt fix  $z_0 = (-\pi, 0)$  i el trobem amb el mètode de Newton amb una precisió de  $10^{-14}$ . La tolerància per a determinar la  $\delta$  la fixem en  $10^{-14}$ , fem una partició de l'interval fonamental amb  $N = 500$  punts i refinem la primera aproximació del punt amb imatge amb component  $x$  nul·la amb el mètode de la secant amb una precisió de  $10^{-11}$ . En tots els passos utilitzem el mètode de Taylor amb un ordre màxim de 16.

Obtenim així els resultats que es mostren en la taula 2.

$\varepsilon$	Angle Trobat	Angle fórmula assimptòtica	Factor $L$
1/4	1.4663172552198997e-03	1.4666809769146745e-03	4.876e+05
1/8	1.3694713151729969e-06	1.3694761408356832e-06	1.013e+05
1/16	1.2291584129470704e-12	2.3879160759059986e-12	3.121e+04
1/32	9.5650360792339978e-13	1.4520388387220488e-23	2.107e+04

Taula 2: Aproximació lineal de la varietat

En aquesta taula, així com en les següents, s'hi mostren el valor del paràmetre  $\varepsilon$  que s'ha utilitzat (no es mostra el valor del paràmetre  $\mu$  ja que en tots els casos s'ha fixat  $\mu = \varepsilon^2$ ), l'angle d'intersecció de les varietats que s'ha trobat i l'angle d'intersecció de les varietats que ens dona la fórmula asimptòtica del Teorema 5.3. El factor  $L$ , ens indica, en cada cas, per a quin factor s'ha multiplicat la norma del vector tangent a la varietat al comparar-lo en  $q_0$  i en  $P^{pot}(q_0)$ .

Presentem ara els resultats del **mètode de la parametrització** per a poder compararlos amb els anteriors.

Per aplicar el mètode de la parametrització, comencem igualment amb una aproximació inicial del punt fix  $z_0 = (-\pi, 0)$ , a partir de la qual apliquem el mètode de Newton amb una precisió de  $10^{-14}$ . Calculem la parametrització de la varietat fins a ordre 16 ( $K_{16}$ ), busquem el valor  $u_0$  amb una  $m = 16$ , la partició de l'interval la fem, en aquest cas, amb  $N = 50$  punts i el mètode de la secant l'apliquem amb una precisió de  $10^{-13}$ . En tots els passos utilitzem el mètode de Taylor amb un ordre màxim de 16.

Obtenim els resultats de la taula 3.

$\varepsilon$	Angle Trobat	Angle fórmula assimptòtica	Factor $L$
1/4	1.4663172625543489e-03	1.4666809769146745e-03	8.212e+00
1/8	1.3694680465579333e-06	1.3694761408356832e-06	8.212e+00
1/16	2.3997784622377963e-12	2.3879160759059986e-12	5.569e+00
1/32	1.0027283467998178e-15	1.4520388387220488e-23	4.594e+00

Taula 3: Parametrització de la varietat

Podem veure que amb el mètode de la parametrització aconseguim més precisió en el càlcul de l'angle, sobretot millorant el valor per a  $\varepsilon = 1/16$ . Aquest fet és degut fonamentalment a que, al tenir la parametrització de la varietat al voltant del punt amb òrbita periòdica inestable ( $z_0$ ), hem pogut realitzar els càlculs partint d'un interval fonamental més allunyat d'aquest punt  $z_0$ . Per aquest motiu, hem necessitat, en general, menys iteracions (*pot*) de l'aplicació estroboscòpica per fer que aquest interval fonamental intersequi l'eix  $x = 0$  i així ha quedat reflectit en valors del factor  $L$  significativament menors. Aquest fet ens ha suposat un augment en la precisió dels valors obtinguts al traslladar l'interval fins a creuar  $x = 0$  i, per tant, hem pogut aplicar el mètode de la



secant demanant una precisió major.

Veiem, però, que encara no tenim tota la precisió necessària per a poder donar una bona aproximació de l'angle en el cas  $\varepsilon = 1/32$ . Per a solucionar-ho, podem realitzar els càlculs del mètode de la parametrització amb una aritmètica de quàdruple precisió en comptes de la de doble precisió que hem utilitzat en els casos anteriors.

Al fer aquest pas, podem demanar més precisió en els processos que realitzem. Comencem igualment amb una aproximació inicial del punt fix  $z_0 = (-\pi, 0)$ , a partir de la qual apliquem el mètode de Newton ara amb una precisió de  $10^{-32}$ . Calculem la parametrització de la varietat fins a ordre 30 ( $K_{30}$ ), busquem el valor  $u_0$  amb una  $m = 32$ , la partició de l'interval la mantenim amb  $N = 50$  punts i el mètode de la secant l'apliquem amb una precisió de  $10^{-30}$ . En tots els passos utilitzem el mètode de Taylor amb un ordre màxim de 30.

Obtenim els resultats de la taula 4.

$\varepsilon$	Angle Trobat	Angle fórmula assimptòtica	Factor $L$
1/4	1.4663172625545283e-03	1.4666809769146744e-03	8.212e+00
1/8	1.3694680508393990e-06	1.3694761408356826e-06	8.212e+00
1/16	2.3879159970462720e-12	2.3879160759059964e-12	5.569e+00
1/32	1.4520388384020293e-23	1.4520388387220459e-23	5.569e+00

Taula 4: Parametrització de la varietat - Float128

Observem que hem aconseguit una millora en la precisió dels resultats per a  $\varepsilon = 1/16$  i una bona aproximació per a  $\varepsilon = 1/32$ .

En aquesta última taula, podem apreciar que els valors de l'angle trobat són més exactes, més semblants als calculats amb la fórmula del teorema 5.3, a mesura que anem disminuint el valor de  $\varepsilon$ . Aquest fet pot ser explicat, ja que el teorema ens presenta la fórmula com assimptòtica i, per tant, ens proporciona un valor més exacte com més proper a 0 és el valor de  $\varepsilon$ .

## 6 Conclusions

Finalitzat ja el treball, puc considerar que l'objectiu d'estudiar objectes invariants en els sistemes d'equacions diferencials de pèndols pertorbats ha estat assolit en els casos en els quals m'he centrat: localitzar òrbites periòdiques, estudiar-ne l'estabilitat i aproximar varietats invariants al voltant d'un punt amb òrbita inestable.

Durant el temps que m'ha portat completar aquest treball, he estat aplicant coneixements obtinguts en diverses assignatures del grau de matemàtiques en noves situacions. També he ampliat coneixements previs, com en l'àmbit de la integració numèrica d'equacions diferencials i n'he obtingut de nous, com la diferenciació automàtica i la programació amb aritmètica de quàdruple precisió.

Una part important del treball ha estat la programació dels procediments teòrics descrits en aquesta memòria. Aquest procés m'ha servit, sobretot en l'apartat 5, per a entendre la importància de tenir cura de la precisió amb la que es treballa en situacions que presenten caos.

## Referències

- [1] Carmen Chicone. *Ordinary Differential Equations with Applications*, volume 34 of *Texts in Applied Mathematics*. Springer-Verlag, 1999.
- [2] Amadeu Delshams and Teresa M. Seara. An Asymptotic Expression for the Splitting of Separatrices of the Rapidly Forced Pendulum. *Communications in Mathematical Physics*, 150:433–463, 1992.
- [3] Jean Gallier. Logarithms and Square Roots of Real Matrices Existence, Uniqueness and Applications in Medical Imaging. Philadelphia, PA 19104, USA, September 2019.
- [4] Joan Gimeno, Àngel Jorba, Marc Jorba-Cuscó, Narcís Miguel, and Maorong Zou. Numerical Integration of High-Order Variational Equations of ODEs. *Applied Mathematics and Computation*, 442:1–20, 2022.
- [5] Joan Gimeno, Àngel Jorba, and Maorong Zou. *Taylor User's Manual: Version 2.1*, 2023.
- [6] David Francis Griffiths and Desmond J Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. Springer Undergraduate Mathematics Series. Springer, 2010.
- [7] Àngel Jorba and Maorong Zou. A Software Package for the Numerical Integration of ODEs by Means of High-Order Taylor Methods. *Experimental Mathematics*, 14(1):99–111, 2004.
- [8] Mohammed Abdullah Saleh Salman and Dr. V.C. Borkar. Exponential Matrix and their Properties. *International Journal of Scientific and Innovative Mathematical Research (IJSIMR)*, 4:53–63, January 2016.
- [9] Jorge Manuel Sotomayor. *Lições de Equações Diferenciais Ordinárias*. Inst. de Matemática Pura e Aplicada, 1979.
- [10] Josef Stoer, Roland Bulirsch, R Bartels, Walter Gautschi, and Christoph Witzgall. *Introduction to Numerical Analysis*. Springer-Verlag, 2nd edition, 1992.
- [11] Joan Carles Tatjer. Mètodes Numèrics II (curs 22-23). 08007, Barcelona, Spain, January 2023.
- [12] Gerald Teschl. *Ordinary Differential Equations and Dynamical Systems*, volume 140. American Mathematical Soc., 2012.



## Annex: Programes

El que es presenta en aquest annex és una selecció dels programes que s'han fet per obtenir els gràfics i resultats presents en aquest treball.

Se'n pot trobar una més àmplia col·lecció en el següent enllaç: [Programes](#).

Tots els programes utilitzen l'integrador d'equacions diferencials descrit en [5].

### Continuació

El programa que realitza la continuació treballa junt amb un programa generat pel programari [5] que integra les equacions presents en el següent fitxer:

```
extern double w, eps; /* declare some external vars */

/* Sistema del pendol */
diff(x,t) = y;
diff(y,t) = - w * w * sin(x) + eps * sin(t);
/* Variacionals respecte condicio inicial */
diff(a,t) = c;
diff(b,t) = d;
diff(c,t) = - w * w * cos(x) * a;
diff(d,t) = - w * w * cos(x) * b;
/* Variacionals respecte parametre w */
diff(E,t) = F;
diff(F,t) = - w * w * cos(x) * E - 2 * w * sin(x);
```

Equacions

Aquest és el programa que realitza el mètode de continuació descrit en l'apartat 4.4 d'aquest treball.

```
/* Salvador Font i Bergada
Curs 2023 - 2024
Contnuacio amb predictor-corrector
Distingirem entre punts fixos estables i inestables
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "taylor.h"

double w, eps;

void constructvec(double z[2], double M[2][3], double v[8]);
void destructvec(double v[8], double z[2], double M[2][3]);
void F(double z[2], double DF[2][3]);
int prediccio (double h, double z[2], double vtanterior[3]);
int correccio (double h, double z0[2], double w0, double z[2], int kmax, double prec);
int resoldre (double A[3][3], double b[3], double x[3]);
int stabilitytest(double z[2]);

int main (void){
```

```

double t, tfinal, h_return, w0, h, N, prec, kmax;
double err, precNewton = 1e-12, det, detx, dety, tr, discriminant;
int parada = 0, test, iterNewton=0, i;
double z[2], DF[2][3], zpost[2], z0[2], vectan[3];
    double nb;
    char sresul[]="res.resul.stable", uresul[]="res.resul.unstable";
FILE *sfresul, *ufresul;

/* Obertura fitxers sortida */
sfresul = fopen (sresul, "w");
ufresul = fopen (uresul, "w");

/* Parametres */
//w = 1./sqrt(2.0);
w=sqrt(2.0);
eps = 0.0025;

h = 0.001;
N = 2000;
prec = 1e-12;
kmax = 8;

/* Partint de condicions inicials , busquem un punt fix */
zpost[0] = 0.0;
zpost[1] = 0.0;

tfinal = 2*M.PI;

    /* Metode Newton */
while (parada == 0){
    iterNewton++;

    t = 0.0;

    z[0] = zpost[0];
    z[1] = zpost[1];

    F(zpost, DF);

    /* Solucio sistema 2x2 */
    det = DF[0][0]*DF[1][1] - DF[0][1]*DF[1][0];
    detx = -zpost[0]*DF[1][1] + DF[0][1]*zpost[1];
    dety = -DF[0][0]*zpost[1] + zpost[0]*DF[1][0];

    zpost[0] = z[0] + (detx / det);
    zpost[1] = z[1] + (dety / det);

    /* Criteri Parada */
    err = sqrt( pow(z[0]-zpost[0], 2) + pow(z[1]-zpost[1], 2) );
    if(err < precNewton){
        parada = 1;
    }
}

z[0] = zpost[0];
z[1] = zpost[1];

/* Ja tenim un punt fix */
/* Apliquem el metode predictor-corrector per a fer continuacio de w */

    /* Cal inicialitzar el vector tangent de manera que dirigim la
    * correccio cap a w mes propers a 1 */
vectan[0] = 0;
vectan[1] = 0;
vectan[2] = -1; /* Sentit de la continuacio */

    /* Repetim el proces N vegades */
for(i=1; i<=N; i++){

    z0[0] = z[0];

```

```

    z0[1] = z[1];
    w0 = w;

    test = prediccio(h, z, vectan);
    if(test == 1){
        printf("ERROR: ha fallat la prediccio\n");
        return 1;
    }
    test = correccio(h, z0, w0, z, kmax, prec);
    if(test == 1){
        printf("ERROR: ha fallat la correccio\n");
        return 1;
    }

    if(stabilitytest(z)==1) {
        fprintf(sfresul, "%24.15le %24.15le\n", w, z[1]);
    } else {
        fprintf(ufresul, "%24.15le %24.15le\n", w, z[1]);
    }
}

fclose(sfresul);
fclose(ufresul);
return 0;
}

/* Funcio que omple un vector v amb la info de z i M
 * en l'ordre pertinent segons les equacions de Taylor */
void constructvec(double z[2], double M[2][3], double v[8]){
    v[0] = z[0];
    v[1] = z[1];
    v[2] = M[0][0];
    v[3] = M[0][1];
    v[4] = M[1][0];
    v[5] = M[1][1];
    v[6] = M[0][2];
    v[7] = M[1][2];
}

/* Funcio que realitza el proces invers a la funcio anterior */
void destructvec(double v[8], double z[2], double M[2][3]){
    z[0] = v[0];
    z[1] = v[1];
    M[0][0] = v[2];
    M[0][1] = v[3];
    M[1][0] = v[4];
    M[1][1] = v[5];
    M[0][2] = v[6];
    M[1][2] = v[7];
}

/* Funcio que evalua F(z):= P(z)-z, on P(z) es l'aplicacio de Poincare
 * la qual s'evalua evolucionant les condicions inicials amb el metode
 * de Taylor
 * Tambe obtemin els valors de les derivades de F respecte les condicions
 * inicials i el parametre w */
void F(double z[2], double DF[2][3]){

    double t, tfinal;
    int test;
    double v[8], pz[2], det;

    /* Condicions inicials per aplicar Taylor */
    pz[0] = z[0];
    pz[1] = z[1];

    DF[0][0] = 1;
    DF[0][1] = 0;
    DF[1][0] = 0;
    DF[1][1] = 1;

```

```

DF[0][2] = 0;
DF[1][2] = 0;

tfinal = 2*M_PI;
t = 0;

constructvec(pz, DF, v);
/* Taylor */
while(test=taylor_step_pendol(&t, v, 1, 1, -16, -16, &tfinal, NULL, NULL, NULL)
== 0);
if(test == -1){
printf("ERROR en taylor\n");
}
destructvec(v, pz, DF);

/* F(z)=P(z)-z */
z[0] = pz[0] - z[0];
z[1] = pz[1] - z[1];

/* DF = DP - Id */
DF[0][0] = DF[0][0] - 1;
DF[1][1] = DF[1][1] - 1;
}

/* Funcio que implementa la prediccio */
int prediccio (double h, double z[2], double vanterior[3]){
double tol = 1e-5, DHx[2][3], co, v[3], Fz[2], normav, product;

Fz[0] = z[0];
Fz[1] = z[1];
F(Fz, DHx);

/* Comprovacio rang de DHx */
co = DHx[0][0]*DHx[1][0] + DHx[0][1]*DHx[1][1] + DHx[0][2]*DHx[1][2];
co = co / (sqrt(pow(DHx[0][0],2)+pow(DHx[0][1],2)+pow(DHx[0][2],2)));
co = co / (sqrt(pow(DHx[1][0],2)+pow(DHx[1][1],2)+pow(DHx[1][2],2)));
if (fabs(co-1.)<tol){
return 1;
}

/* Vector tangent (producte vectorial files DHx) */
v[0] = DHx[0][1]*DHx[1][2] - DHx[0][2]*DHx[1][1];
v[1] = DHx[0][2]*DHx[1][0] - DHx[0][0]*DHx[1][2];
v[2] = DHx[0][0]*DHx[1][1] - DHx[0][1]*DHx[1][0];

normav = sqrt(pow(v[0],2)+pow(v[1],2)+pow(v[2],2));

/* Verificacio del sentit de la prediccio
* Si vanterior * v < 0 hem de conviar el signe de v*/
product = vanterior[0]*v[0] + vanterior[1]*v[1] + vanterior[2]*v[2];
if(product < 0.){
v[0] = -v[0];
v[1] = -v[1];
v[2] = -v[2];
}

/* Prediccio */
z[0] = z[0] + h * v[0] / normav;
z[1] = z[1] + h * v[1] / normav;
w = w + h * v[2] / normav;

/* Reescribim vanterior amb v */
vanterior[0] = v[0];
vanterior[1] = v[1];
vanterior[2] = v[2];

return 0;
}

```



```

/* Funcio que implementa la correccio (Newton) */
int correccio (double h, double z0[2], double w0c, double z[2], int kmax, double
prec){
    int iter=0, i;
    double DA[3][3], b[3], error, Fz[2], DHx[2][3], x[3];
    double nb;

    do{
        Fz[0] = z[0];
        Fz[1] = z[1];

        F(Fz, DHx);

        b[0] = -Fz[0];
        b[1] = -Fz[1];
        b[2] = -(pow(z[0]-z0[0],2) + pow(z[1]-z0[1],2) + pow(w-w0c,2) - pow(h,2));
        for(i=0; i<3; i++){
            DA[0][i] = DHx[0][i];
            DA[1][i] = DHx[1][i];
        }
        for(i=0; i<2; i++){
            DA[2][i] = 2*(z[i]-z0[i]);
        }
        DA[2][2] = 2*(w-w0c);

        nb=sqrt(b[0]*b[0]+b[1]*b[1]+b[2]*b[2]);

        resoldre(DA, b, x);
        error = sqrt(pow(x[0],2)+pow(x[1],2)+pow(x[2],2));

        for(i=0; i<2; i++){
            z[i] = x[i] + z[i];
        }
        w = x[2] + w;
        iter = iter + 1;

    }while((iter < (kmax+1)) && (error > prec));

    if((iter == (kmax+1)) && (error > prec)){
        /* no s'ha convergit */
        return 1;
    }
    return 0;
}

/* Funcio per a resoldre sistemes lineals 3x3 Ax=b */
int resoldre (double A[3][3], double b[3], double x[3]){
    double detA, det0, det1, det2, tol = 1e-5;
    double r[3]; int i, j;
    /* Resolucio per Cramer: calculem els determinants necessaris */
    detA = A[0][0]*A[1][1]*A[2][2] + A[2][1]*A[1][0]*A[0][2] + A[2][0]*A[0][1]*A
    [1][2];
    detA = detA - A[2][0]*A[1][1]*A[0][2] - A[1][0]*A[0][1]*A[2][2] - A[0][0]*A
    [1][2]*A[2][1];

    if (fabs(detA)<tol){
        return 1;
    }

    det0 = b[0]*A[1][1]*A[2][2] + A[2][1]*b[1]*A[0][2] + b[2]*A[0][1]*A[1][2];
    det0 = det0 - b[2]*A[1][1]*A[0][2] - b[1]*A[0][1]*A[2][2] - b[0]*A[1][2]*A
    [2][1];

    det1 = A[0][0]*b[1]*A[2][2] + b[2]*A[1][0]*A[0][2] + A[2][0]*b[0]*A[1][2];
    det1 = det1 - A[2][0]*b[1]*A[0][2] - A[1][0]*b[0]*A[2][2] - A[0][0]*A[1][2]*b
    [2];

```

```

det2 = A[0][0]*A[1][1]*b[2] + A[2][1]*A[1][0]*b[0] + A[2][0]*A[0][1]*b[1];
det2 = det2 - A[2][0]*A[1][1]*b[0] - A[1][0]*A[0][1]*b[2] - A[0][0]*b[1]*A
  [2][1];

x[0] = det0/detA;
x[1] = det1/detA;
x[2] = det2/detA;

return 0;
}

/* Funcio que ens diu si el punt fix es estable o inestable
   Retorna 1 si es estable i 0 si es inestable */
int stabilitytest(double z[2]){
double t, tfinal, det, tr, discriminant;
int test;
double DP[2][3], v[8];

DP[0][0] = 1;
DP[0][1] = 0;
DP[1][0] = 0;
DP[1][1] = 1;
DP[0][2] = 0;
DP[1][2] = 0;

tfinal = 2*M_PI;
t = 0;

constructvec(z, DP, v);
/* Taylor */
while(test=taylor_step_pendol(&t, v, 1, 1, -16, -16, &tfinal, NULL, NULL, NULL)
== 0);
if(test == -1){
printf("ERROR en taylor\n");
}
destructvec(v, z, DP);

/* Estudiem la estabilitat de l'orbita */
/* Busquem els vaps de DP(z) */

det = DP[0][0]*DP[1][1] - DP[0][1]*DP[1][0];
tr = DP[0][0] + DP[1][1];
discriminant = pow(tr,2) - 4*det ;

if(discriminant < 0.0){
/* Cas estable */
return 1;
}else{
/* Cas inestable */
return 0;
}
}
}

```

## Angle entre separatius amb aproximació lineal de la varietat

El càlcul de l'angle entre separatius amb una aproximació lineal de la varietat s'ha realitzat amb 3 programes executats de forma consecutiva. Per a obtenir el resultat per al valor de  $\varepsilon$  desitjat, només cal modificar-lo en el primer programa, els altres dos no requereixen cap modificació.

Els tres necessiten de l'integrador per a les equacions del sistema:

```
extern double mu, eps; /* declare some external vars */

/* Sistema del pendol */
diff(x,t) = y;
diff(y,t) = - sin(x) + mu * sin(t/eps);
/* Variacionals respecte condicio inicial */
diff(a,t) = c;
diff(b,t) = d;
diff(c,t) = - cos(x) * a;
diff(d,t) = - cos(x) * b;
```

Equacions

Aquests són els tres programes.

```
/* Salvador Font i Bergada
Curs 2023 - 2024
Trobar punts fixos i analitzar la seva estabilitat
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "taylor.h"

double mu, eps;

int main (void){

    double t, tfinal;
    double err, precNewton = 1e-14, det, detx, dety, tr, discriminant;
    double vap1, norm;
    int parada = 0, test, iterNewton=0, i;
    double z[2], pz[6], zpost[2], vep[2];
    double DP[2][2];
    char resul[]="res.resulp1";
    FILE *fresul;

    /* Obertura fitxer sortida */
    fresul = fopen (resul, "w");

    /*Parametres */
    eps = 1./32;
    mu = eps*eps;

    /* Condicions inicials */
    zpost[0] = -M_PI;
    zpost[1] = 0.0;

    tfinal = 2*M_PI*eps;

    /* Metode NEWTON */
```

```

while (parada == 0){
    iterNewton++;

    t = 0.0;

    z[0] = zpost[0];
    z[1] = zpost[1];

    pz[0] = z[0];
    pz[1] = z[1];
    pz[2] = 1.0;
    pz[3] = 0.0;
    pz[4] = 0.0;
    pz[5] = 1.0;

    /* Taylor */
    while(test=taylor_step_pendol( &t, pz, 1, 1, -16, -16, &tfinal, NULL, NULL,
    NULL) == 0);
    if(test == -1){
        printf("ERROR en taylor\n");
        break;
    }

    DP[0][0] = pz[2];
    DP[0][1] = pz[3];
    DP[1][0] = pz[4];
    DP[1][1] = pz[5];

    /* Solucio sistema 2x2 - Cramer*/
    det = ((DP[0][0]-1)*(DP[1][1]-1)) - (DP[0][1]*DP[1][0]);
    detx = ((z[0]-pz[0])*(DP[1][1]-1)) - (DP[0][1]*(z[1]-pz[1]));
    dety = ((DP[0][0]-1)*(z[1]-pz[1])) - ((z[0]-pz[0])*DP[1][0]);

    zpost[0] = z[0] + (detx / det);
    zpost[1] = z[1] + (dety / det);

    /* Criteri Parada */
    err = sqrt( pow(z[0]-zpost[0], 2) + pow(z[1]-zpost[1], 2) );
    if(err < precNewton){
        parada = 1;
    }
}

printf(" Newton Iter: %d \n", iterNewton);
printf(" Newton Err : %27.16le \n", err);
printf(" (x, x') = (%27.16le , %27.16le ) \n", zpost[0], zpost[1]);
printf(" \n\n");

/* Estudiem la estabilitat de l'orbita */
/* Busquem els vaps de DP(z)*/
det = DP[0][0]*DP[1][1] - DP[0][1]*DP[1][0];
tr = DP[0][0] + DP[1][1];
discriminant = pow(tr,2) - 4*det ;

printf(" Det de DP(z): %27.16le \n", det);

if(discriminant < 0.0){
    printf(" El punt fix trobat NO es inestable \n");
    exit(1);
}else{
    printf(" Els vaps de DP(z): \n");
    printf(" %27.16le \n", (tr + sqrt(discriminant))/2.0);
    printf(" %27.16le \n", (tr - sqrt(discriminant))/2.0);
}
printf(" \n");

/* Trobem els veps de DP(z) en el cas del equilibri inestable:

```

```

    * Coneixem els vaps vap1 i vap2 aleshores ,
    * vepi = ( -DP[0][1] , DP[0][0] - vapi )
    * */
vap1 = (tr + sqrt(pow(tr,2) - 4*det))/2.0;
vep[0] = -(-DP[0][1]);
vep[1] = -(DP[0][0] - vap1);

    /*Normalitzem el vep*/
norm = sqrt(pow(vep[0],2)+pow(vep[1], 2));
vep[0] = vep[0] / norm;
vep[1] = vep[1] / norm;

printf(" vap inestable: %27.16le \n", vap1 );
printf(" vep inestable: %27.16le , %27.16le \n", vep[0], vep[1] );

    /* Dades pel fitxer de sortida:
    * eps i mu; punt fix; vap i vep inestable; DP */
fprintf(fresul, " %27.16le %27.16le \n", eps, mu);
fprintf(fresul, " %27.16le %27.16le \n", zpost[0], zpost[1]);
fprintf(fresul, " %27.16le \n", vap1);
fprintf(fresul, " %27.16le %27.16le \n", vep[0], vep[1]);
fprintf(fresul, " %27.16le %27.16le \n", DP[0][0], DP[0][1]);
fprintf(fresul, " %27.16le %27.16le \n", DP[1][0], DP[1][1]);

fclose(fresul);
return 0;
}

```

programal.c

```

/* Salvador Font i Bergada
Curs 2023 - 2024
Trobar varietat inestable
(dos punts que seran extrems de l'interval fonamental)
Trobar la potencia de P que trasllada aquest interval a
interseccar l'eix x=0
Trobar dos punts dins aquest interval, relativamet propers,
amb imatges per P^pot a banda i banda de l'eix x=0
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "taylor.h"

double mu, eps;

int main (void){

    double pfix[2], q[2], qlineal[2], pq[2], pz[6], vep[2];
    double a[2], b[2], pma[2], pmb[2];
    double vap, delta, t, tfinal, err;
    int i, test, j, potencia;
    char in[]="res.resulp1", resul[]="res.resulp2";
    FILE *fin, *fresul;

    /* Obertura fitxers entrada i sortida */
    fin = fopen (in, "r");
    fresul = fopen (resul, "w");

    /* Lectura dades fitxer entrada */
    fscanf(fin, " %lf %lf", &eps, &mu);
    fscanf(fin, " %lf %lf", &pfix[0], &pfix[1]);
    fscanf(fin, " %lf ", &vap);

```

```

fscanf(fin , " %lf %lf" , &vep[0] , &vep[1]);

tfinal = 2*M.PI*eps;

/* sigui q = pfix + delta * vep */
/* Busco un delta tq l'error entre pfix + delta * vap * vep i P(q)
 * sigui menor que 1e-14*/
test = 0;
delta = 1e-4;
err = 1.;
printf(" vap inestable: %27.16le \n", vap );
printf(" vep inestable: %27.16le , %27.16le \n", vep[0] , vep[1] );
i=0;
while (err > 1e-14){
    delta = delta*0.9;
    q[0] = pfix[0] + delta * vep[0];
    q[1] = pfix[1] + delta * vep[1];
    qlineal[0] = pfix[0] + delta * vap * vep[0];
    qlineal[1] = pfix[1] + delta * vap * vep[1];
    pz[0] = q[0];
    pz[1] = q[1];
    pz[2] = 1.0;
    pz[3] = 0.0;
    pz[4] = 0.0;
    pz[5] = 1.0;
    /* Taylor */
    t = 0.0;
    while(test=taylor_step_pendol( &t , pz , 1 , 1 , -16 , -16 , &tfinal , NULL , NULL,
    NULL) == 0);
    if(test == -1){
        printf("ERROR en taylor\n");
        break;
    }
    pq[0] = pz[0];
    pq[1] = pz[1];
    err = sqrt( pow(qlineal[0]-pq[0],2) + pow(qlineal[1]-pq[1],2) );
    printf("%27.16le %27.16le \n", err , delta);
    i++;
}
printf(" iterdelta: %d\n", i);
printf(" q: %27.16le , %27.16le \n", q[0] , q[1]);
printf(" ql: %27.16le , %27.16le \n", qlineal[0] , qlineal[1]);

/* Ara podem considerar el segment entre q i qlineal com a interval
 * fonamental de la varietat.
 * Iterarem un nombre de punts d'aquest segment per a trobar-la */

/* Busquem la potencia de P que fa que el segment tingui imatges amb
 * x negatives i positives (creui eix de les y)
Pasara quan la imatge de qlineal per una potencia de P tingui x positiva */
pz[0] = pfix[0] + delta * vap * vep[0] ;
pz[1] = pfix[1] + delta * vap * vep[1] ;
pz[2] = 1.0;
pz[3] = 0.0;
pz[4] = 0.0;
pz[5] = 1.0;
j=0;
while(pz[0]<0.){
    j++;
    /*Taylor*/
    t = 0.0;
    while(test=taylor_step_pendol( &t , pz , 1 , 1 , -16 , -16 , &tfinal , NULL , NULL,
    NULL) == 0){
        if(test == -1){
            printf("ERROR en taylor\n");
            break;
        }
    }
}
}

```

```

printf(" pot: %d \n",j);
potencia = j;

/* Iterant 'potencia' vegades P(z) ja trobem punts amb valors de x (pz[0])
positius
* Busquem els dos punts (a i b) tals que la seva imatge per 'potencia'
iteracions de P(z)
* sigui mes propera a 0 tant pels positius com pels negatius */
pma[0] = -10;
pmb[0] = 10;
for(i=1; i<500; i++){
    pz[0] = q[0] + delta * vap * vep[0] * i / 500;
    pz[1] = q[1] + delta * vap * vep[1] * i / 500;
    pma[0] = pmb[0];
    pma[1] = pmb[1];
    for(j=1; j<=potencia; j++){
        pz[2] = 1.0;
        pz[3] = 0.0;
        pz[4] = 0.0;
        pz[5] = 1.0;
        /*Taylor*/
        t = 0.0;
        while(test=taylor_step_pendol( &t, pz, 1, 1, -16, -16, &tfinal, NULL, NULL
, NULL) == 0){
            if(test == -1){
                printf("ERROR en taylor\n");
                break;
            }
        }
    }
}

pmb[0] = pz[0];
pmb[1] = pz[1];
if(pz[0]> 0.0){
    /*Hem trobat el punt que creua x=0.
Aquest i l'anterior son els que busquem */
    b[0] = q[0] + delta * vap * vep[0] * i / 500;
    b[1] = q[1] + delta * vap * vep[1] * i / 500;
    a[0] = q[0] + delta * vap * vep[0] * (i-1) / 500;
    a[1] = q[1] + delta * vap * vep[1] * (i-1) / 500;
    break;
}
}

printf(" a: %27.16le , %27.16le \n", a[0], a[1]);
printf("pma: %27.16le , %27.16le \n", pma[0], pma[1]);
printf(" b: %27.16le , %27.16le \n", b[0], b[1]);
printf("pmb: %27.16le , %27.16le \n", pmb[0], pmb[1]);
printf(" j: %d \n",potencia);

/* Dades pel fitxer de sortida: */
fprintf(fresul, " %27.16le %27.16le \n", eps, mu);
fprintf(fresul, " %d \n",potencia);
fprintf(fresul, " %27.16le %27.16le \n", a[0], a[1]);
fprintf(fresul, " %27.16le %27.16le \n", b[0], b[1]);

fclose(fin);
fclose(fresul);
return 0;
}

```

programa2.c

```

/* Salvador Font i Bergada
Curs 2023 – 2024
Trobar l'angle entre les separatius

Partint dels punts a i b trobats en el programa2.c, amb el
metode de la secant, trobarem un punt c, del
segment ab, tal que P^pot(c) tingui component x
"nul·la"
Amb això i DP^pot(c), poderem calcular l'angle
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "taylor.h"

double mu, eps;

int main (void){

    double pz[6];
    double a[2], b[2], c1[2], c2[2], vtan[2], vtanfinal[2];
    double lambda1, lambda2, f1, f2, aux, angle, t, tfinal, err = 1;
    double normvt1, normvt2;
    int i, test, j, potencia;
    char in[]="res.resulp2";
    FILE *fin;

    /* Obertura fitxers entrada i sortida */
    fin = fopen (in, "r");

    /* Lectura dades fitxer de entrada */
    fscanf(fin, " %lf %lf", &eps, &mu);
    fscanf(fin, " %d", &potencia);
    fscanf(fin, " %lf %lf", &a[0], &a[1]);
    fscanf(fin, " %lf %lf", &b[0], &b[1]);

    printf(" a: %27.16le , %27.16le \n", a[0], a[1]);
    printf(" b: %27.16le , %27.16le \n", b[0], b[1]);
    printf("pot: %d \n", potencia);

    tfinal = 2*M_PI*eps;

    lambda1 = 0.;
    lambda2 = 1.;

    /* Metode de la secant */
    /* Busquem el lambda tq P^10(a*lambda + b*(1-lambda)) sigui 0 */
    while ( err > 1e-11){
        c1[0] = a[0]*(1.-lambda1) + b[0]*(lambda1);
        c1[1] = a[1]*(1.-lambda1) + b[1]*(lambda1);
        c2[0] = a[0]*(1.-lambda2) + b[0]*(lambda2);
        c2[1] = a[1]*(1.-lambda2) + b[1]*(lambda2);

        /* Calculem les seves imatges per P^potencia */
        pz[0] = c1[0];
        pz[1] = c1[1];
        pz[2] = 1.0;
        pz[3] = 0.0;
        pz[4] = 0.0;
        pz[5] = 1.0;
        for(j=1; j<=potencia; j++){
            /*Taylor*/
            t = 0.0;
            while(test=taylor_step_pendol( &t, pz, 1, 1, -16, -16, &tfinal, NULL, NULL
, NULL) == 0){
                if(test == -1){
                    printf("ERROR en taylor\n");

```



```

        break;
    }
}
f1 = pz[0];

pz[0] = c2[0];
pz[1] = c2[1];
pz[2] = 1.0;
pz[3] = 0.0;
pz[4] = 0.0;
pz[5] = 1.0;
for(j=1; j<=potencia; j++){
    /*Taylor*/
    t = 0.0;
    while(test=taylor_step_pendol( &t, pz, 1, 1, -16, -16, &tfinal, NULL, NULL
, NULL) == 0){
        if(test == -1){
            printf("ERROR en taylor\n");
            break;
        }
    }
}
f2 = pz[0];

aux = lambda2;
lambda2 = aux - f2*(aux-lambda1) / (f2-f1);
lambda1 = aux;

err = fabs(f2);
}
printf(" lambda final: %27.16le \n", aux);
printf(" c: %27.16le , %27.16le \n", c2[0], c2[1]);
printf(" pc: %27.16le , %27.16le \n", pz[0], pz[1]);

/* Busquem el vector tangent a la varietat en x=0 */
/* En c es b-a: */
vtan[0] = b[0]-a[0];
vtan[1] = b[1]-a[1];
normvt1 = sqrt(pow(vtan[0], 2) + pow(vtan[1], 2));

/* En P^potencia(c) es DP^potencia*(b-a) */
vtanfinal[0] = pz[2] * vtan[0] + pz[3] * vtan[1];
vtanfinal[1] = pz[4] * vtan[0] + pz[5] * vtan[1];
normvt2 = sqrt(pow(vtanfinal[0], 2) + pow(vtanfinal[1], 2));

printf(" DP vtani\n ");
printf(" %27.16le , %27.16le | %27.16le \n", pz[2], pz[3], vtan[0]);
printf(" %27.16le , %27.16le | %27.16le \n", pz[4], pz[5], vtan[1]);

printf(" vtan: %27.16le , %27.16le \n", vtanfinal[0], vtanfinal[1]);

angle = atan(vtanfinal[1] / vtanfinal[0]);
angle = 2.0 * angle;

printf(" angle : %27.16le \n", angle);
printf(" angle formula: %27.16le \n", (M_PI/(2*eps))*(mu/cosh(M_PI/(2*eps))));

printf(" la norma del vtan s'ha multiplicat per: %27.16le \n", normvt2/normvt1);

fclose(fin);
return 0;
}

```

programa3.c

## Angle entre separatius amb parametrització de la varietat

Per a calcular l'angle entre separatius fent una parametrització de la varietat, també s'han realitzat 3 programes. Els que es presenten a continuació són els que implementen l'aritmètica de quàdruple precisió (float128) i són anàlegs als que treballen amb aritmètica de doble precisió. Per a obtenir el resultat per al valor de  $\varepsilon$  desitjat, només cal modificar-lo en el primer programa, els altres dos no requereixen cap modificació.

Aquests programes necessiten de dos integradors diferents. Un per a les equacions:

```
extern MY_FLOAT mu, eps; /* declare some external vars */

/* Sistema del pendol */
diff(x,t) = y;
diff(y,t) = - sin(x) + mu * sin(t/eps);
/* Variacionals respecte condicio inicial */
diff(a,t) = c;
diff(b,t) = d;
diff(c,t) = - cos(x) * a;
diff(d,t) = - cos(x) * b;
```

### Equacions

I un altre que, a part d'integrar les equacions del sistema, realitza els càlculs del *jet transport*:

```
extern MY_FLOAT mu, eps; /* declare some external vars */

/* Sistema del pendol */
diff(x,t) = y;
diff(y,t) = - sin(x) + mu * sin(t/eps);

jet x,y symbols 1 deg 30;
```

### Equacions i jet

Aquests són els tres programes.

```
/* Salvador Font i Bergada
Curs 2023 - 2024
Trobar punts fixos i analitzar la seva estabilitat
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <quadmath.h>

#include "taylor.h"

__float128 mu, eps;

int main (void){

    __float128 t, tfinal;
    __float128 err, precNewton = 1e-32q, det, detx, dety, tr, discriminant;
    __float128 vap1, norm;
```

```

int parada = 0, test, iterNewton=0, i;
_float128 z[2], pz[6], zpost[2], vep[2];
_float128 DP[2][2];
char resul[]="res.resulp1";
char buf[128];
FILE *fresul;

/* Obertura fitxer sortida */
fresul=fopen(resul,"wb");
if (fresul == NULL) {puts("no puc obrir prova.res per escriure"); exit(1);}

/* Parametres */
eps = 1.0q/32.0q;
mu = eps*eps;

/* Condicions inicials */
zpost[0] = -M.PIq;
zpost[1] = 0.0q;

tfinal = 2.0q*M.PIq*eps;

/* Metode NEWTON*/
while (parada == 0){
    iterNewton++;

    t = 0.0q;

    z[0] = zpost[0];
    z[1] = zpost[1];

    pz[0] = z[0];
    pz[1] = z[1];
    pz[2] = 1.0q;
    pz[3] = 0.0q;
    pz[4] = 0.0q;
    pz[5] = 1.0q;

    /* Taylor */
    while(test=taylor_step_pendol(&t, pz, 1, 1, -34, -34, &tfinal, NULL, NULL,
    NULL) == 0);
    if(test == -1){
        printf("ERROR en taylor\n");
        break;
    }

    DP[0][0] = pz[2];
    DP[0][1] = pz[3];
    DP[1][0] = pz[4];
    DP[1][1] = pz[5];

    /* Solucio sistema 2x2 - Cramer*/
    det = ((DP[0][0]-1.0q)*(DP[1][1]-1.0q)) - (DP[0][1]*DP[1][0]);
    detx = ((z[0]-pz[0])*(DP[1][1]-1.0q)) - (DP[0][1]*(z[1]-pz[1]));
    dety = ((DP[0][0]-1.0q)*(z[1]-pz[1])) - ((z[0]-pz[0])*DP[1][0]);

    zpost[0] = z[0] + (detx / det);
    zpost[1] = z[1] + (dety / det);

    /* Criteri Parada */
    err = sqrtq((z[0]-zpost[0])*(z[0]-zpost[0]) + (z[1]-zpost[1])*(z[1]-zpost[1])
);
    if(err < precNewton){
        parada = 1;
    }
}

```

```

printf(" Newton Iter: %d \n", iterNewton);

    printf(" Punt fix: (");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qf", zpost[0]);
    printf("%s, ", buf);
quadmath_snprintf(buf, sizeof(buf), "%35.30Qf", zpost[1]);
    printf("%s )\n", buf);

/* Estudiem la estabilitat de l'orbita */
/* Busquem els vaps de DP(z)*/
det = DP[0][0]*DP[1][1] - DP[0][1]*DP[1][0];
tr = DP[0][0] + DP[1][1];
discriminant = tr*tr - 4.0q*det ;

if(discriminant < 0.0q){
    printf(" El punt fix trobat NO es inestable \n");
    exit(1);
}

/* Trobem els veps de DP(z) en el cas del equilibri inestable:
 * Coneixem els vaps vap1 i vap2 aleshores ,
 * vepi = ( -DP[0][1] , DP[0][0] - vapi )
 * */
vap1 = (tr + sqrtq(tr*tr - 4.0q*det))/2.0q;
vep[0] = -(-DP[0][1]);
vep[1] = -(DP[0][0] - vap1);

/*Normalitzem el vep*/
norm = sqrtq(vep[0]*vep[0] + vep[1]*vep[1]);
vep[0] = vep[0] / norm;
vep[1] = vep[1] / norm;

printf(" Vap inestable: ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qf", vap1);
    printf("%s\n", buf);
printf(" Vep inestable: ( ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qf", vep[0]);
    printf("%s, ", buf);
quadmath_snprintf(buf, sizeof(buf), "%35.30Qf", vep[1]);
    printf("%s )\n", buf);

/* Dades pel fitxer de sortida: */

fwrite(&eps, sizeof(_float128), 1, fresul);
fwrite(&mu, sizeof(_float128), 1, fresul);
fwrite(&zpost[0], sizeof(_float128), 1, fresul);
fwrite(&zpost[1], sizeof(_float128), 1, fresul);
fwrite(&vap1, sizeof(_float128), 1, fresul);
fwrite(&vep[0], sizeof(_float128), 1, fresul);
fwrite(&vep[1], sizeof(_float128), 1, fresul);
fwrite(&DP[0][0], sizeof(_float128), 1, fresul);
fwrite(&DP[0][1], sizeof(_float128), 1, fresul);
fwrite(&DP[1][0], sizeof(_float128), 1, fresul);
fwrite(&DP[1][1], sizeof(_float128), 1, fresul);

fclose(fresul);
return 0;
}

```

programa1.c

```

/* Salvador Font i Bergada
   Curs 2023 – 2024
   Calcular coeficients parametrizacio
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <quadmath.h>

#include "taylor_pdj.h"

__float128 mu,eps;

void sistema (__float128 M[2][2], __float128 a[2], __float128 b[2]);

int main(void)
{
    int i, j, ordre;
    __float128 t, tf, vap, det, detx, dety;
    __float128 x[2], A[2][31], pfix[2], vep[2], DP[2][2], M[2][2], b[2], a[2];
    MYJET xjet[2]; /* declarar el vector de 2 jets */
    char in[]="res.resulp1", resul[]="res.resulp2";
    char buf[128];
    FILE *fin, *fresul;

    taylor_initialize_jet_library(); /* inicialitzar la llibreria de jets */
    taylor_initialize_jet_variable(xjet); /*inicialitzar variables */
    taylor_initialize_jet_variable(xjet+1); /*inicialitzar variables */

    /* Obertura fitxers entrada i sortida */
    fin=fopen(in,"rb");
    if (fin == NULL) {puts("no puc obrir prova.res per llegir"); exit(1);}
    fresul=fopen(resul,"wb");
    if (fresul == NULL) {puts("no puc obrir prova.res per escriure"); exit(1);}

    /* Lectura dades fitxer entrada */
    fread(&eps, sizeof(__float128),1,fin);
    fread(&mu, sizeof(__float128),1,fin);
    fread(&pfix[0], sizeof(__float128),1,fin);
    fread(&pfix[1], sizeof(__float128),1,fin);
    fread(&vap, sizeof(__float128),1,fin);
    fread(&vep[0], sizeof(__float128),1,fin);
    fread(&vep[1], sizeof(__float128),1,fin);
    fread(&DP[0][0], sizeof(__float128),1,fin);
    fread(&DP[0][1], sizeof(__float128),1,fin);
    fread(&DP[1][0], sizeof(__float128),1,fin);
    fread(&DP[1][1], sizeof(__float128),1,fin);

    /* Primers coeficients */
    A[0][0] = pfix[0];
    A[1][0] = pfix[1];

    A[0][1] = vep[0];
    A[1][1] = vep[1];

    for(ordre = 2; ordre < 31; ordre++){
        /* Amb la crida a la següent funcio escollim el grau al que anem a
           treballar, amb un maxim igual al que hem posat al fitxer
           pendol.eq (es a dir, 30). Aquest valor anira variant mentre usem
           el metode de la parametrizacio.
        */
        taylor_set_jet_variable_degree(ordre);
        for (j=0; j<2; j++){
            {
                for(i=0; i<ordre; i++){
                    xjet[j][i]=A[j][i]; /* termes constants del jet */
                }
            }
        }
    }
}

```

```

    xjet[j][ordre]=0.0q;
}

x[0]=pfix[0];
x[1]=pfix[1];
t=0.0q;
tf=2.0q*M_PIq*eps;
while (taylor_step_pdj(&t,x,1,1,-34,-34,&tf,NULL,NULL,xjet)==0);

/* Hem de resoldre (DP(pfix) - vap^ordre * Id) a_ordre = - xjet_ordre
 * per a trobar els components columna _ordre_ de la matriu A */
/* Solucio sistema 2x2 */
M[0][0] = DP[0][0] - powq(vap, ordre);
M[0][1] = DP[0][1];
M[1][0] = DP[1][0];
M[1][1] = DP[1][1] - powq(vap, ordre);

b[0] = -xjet[0][ordre];
b[1] = -xjet[1][ordre];

sistema(M, a, b);

A[0][ordre] = a[0];
A[1][ordre] = a[1];
}

/* Impresio resultats */
fwrite(&eps, sizeof(__float128), 1, fresul);
fwrite(&mu, sizeof(__float128), 1, fresul);
fwrite(&vap, sizeof(__float128), 1, fresul);

for(i=0; i<31; i++){
    for(j=0; j<2; j++){
        quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", A[j][i]);
        printf("%s\n", buf);
        fwrite(&A[j][i], sizeof(__float128), 1, fresul);
    }
    printf("\n");
}

fclose(fin);
fclose(fresul);

return 0;
}

/* funcio que resol un sistema 2x2 per Cramer */
void sistema (__float128 M[2][2], __float128 a[2], __float128 b[2]){
    __float128 detx, dety, det;

    det = M[0][0] * M[1][1] - M[0][1] * M[1][0];
    detx = b[0] * M[1][1] - M[0][1] * b[1];
    dety = M[0][0] * b[1] - b[0] * M[1][0];

    a[0] = (detx / det);
    a[1] = (dety / det);
}

```

programa2.c

```

/* Salvador Font i Bergada
Curs 2023 – 2024
Calcular Angle entre separatius

Trobar varietat inestable
(u0 i alpha tq. K(u0) i K(alpha) son extrems de
l'interval fonamental)
Trobar la potencia de P que trasllada aquest interval a
interseccar l'eix x=0
Trobar dos punts dins aquest interval, relativamet propers,
amb imatges per P^pot a banda i banda de l'eix x=0.
Despres aplicar el metode de la secant per a
refinar aquests valors
Amb això i DP^pot(c), poderem calcular l'angle
Fem tambe un test per a la component d'ordre 2 de la
parametrizacio de la varietat
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <quadmath.h>

#include "taylor_pdj.h"
#include "taylor.h"

void K(--float128 Ku[2], --float128 A[2][31], --float128 u, int grau);
void dK(--float128 Ku[2], --float128 A[2][31], --float128 u, int grau);

--float128 mu, eps;

int main(void)
{
    int i, j, p, ordre = 30, potencia;
    --float128 t, tf, vap, u0, alpha, u, ua, ub, norm1, norm2, angle;
    --float128 lambda1, lambda2, f1, f2, aux, err = 1., alpha2;
    --float128 x[2], A[2][31], Ku0[2], Kvu0[2], Ku[2], dKu[2];
    --float128 pz[6], vtanfinal[2];
    MY_JET xjet[2]; /* declarar el vector de 2 jets */
    char in[]="res.resulp2";
    char buf[128];
    FILE *fin;

    taylor_initialize_jet_library(); /* inicialitzar la llibreria de jets */
    taylor_initialize_jet_variable(xjet); /*inicialitzar variables */
    taylor_initialize_jet_variable(xjet+1); /*inicialitzar variables */

    /* Obertura fitxers entrada i sortida */
    fin=fopen(in,"rb");
    if (fin == NULL) {puts("no puc obrir prova.res per llegir"); exit(1);}

    /* Lectura dades fitxer de entrada */
    fread(&eps, sizeof(--float128), 1, fin);
    fread(&mu, sizeof(--float128), 1, fin);
    fread(&vap, sizeof(--float128), 1, fin);

    for(i=0; i<31; i++){
        for(j=0; j<2; j++){
            fread(&A[j][i], sizeof(--float128), 1, fin);
        }
    }

    /* Busco u tq ||A30*u^30|| <= 1e-32
    * em quedo amb la cota alpha tq |u|<= alpha*/

    alpha = -32.0q - log10q(sqrtq(A[0][30]*A[0][30] + A[1][30]*A[1][30]));
    alpha = alpha / 30.0q;

```

```

alpha = powq(10.0q, alpha);

alpha2 = -32.0q - log10q(sqrtq(A[0][29]*A[0][29] + A[1][29]*A[1][29]));
alpha2 = alpha2 / 29.0Q;
alpha2 = powq(10.0q, alpha2);

if(alpha2 < alpha){
    alpha = alpha2;
}

u0 = alpha / vap;

K(Ku0, A, u0, 30);
K(Kvapu0, A, alpha, 30);

printf(" Interval fonamental (u): [ ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", u0);
    printf("%s, ", buf);
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", alpha);
    printf("%s ]\n\n", buf);

taylor_set_jet_variable_degree(0);

x[0]=Ku0[0];
x[1]=Ku0[1];
xjet[0][0]=x[0];
xjet[1][0]=x[1];
t=0.0q;
tf=2.0q*M_PIq*eps;
while (taylor_step_pdj(&t, x, 1, 1, -34, -34, &tf, NULL, NULL, xjet)==0);

printf(" err P(K(u.0)) - K(alpha): ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", sqrtq((x[0]-Kvapu0[0])*(x[0]-Kvapu0[0]) + (x[1]-Kvapu0[1])*(x[1]-Kvapu0[1])));
    printf("%s\n\n", buf);

/* A partir d'aquí, puc utilitzar K(u) amb u de [u0, alpha] com a interval
fonamental
* i procedir a buscar l'angle de tall entre les separatius */

/*Busquem la potencia de P que fa que l'interval tingui imatges amb
* x negatives i positives (creui eix de les y)
* Pasara quan la imatge de K(alpha) per una potencia de P tingui x positiva */
K(Ku, A, alpha, 30);
p=0;
x[0]=Ku[0];
x[1]=Ku[1];
while (x[0]<0.q){
    p++;
    xjet[0][0]=x[0];
    xjet[1][0]=x[1];
    t=0.0q;
    tf=2.0q*M_PIq*eps;
    while (taylor_step_pdj(&t, x, 1, 1, -34, -34, &tf, NULL, NULL, xjet)==0);
}
potencia = p;
printf(" potencia: %d\n", potencia);

/* Busquem els dos punts (K(ua) i K(ub)) de l'interval fonamental
tals que la seva imatge per 'potencia' iteracions de P(z)
sigui mes propera a 0 tant pels positius com pels negatius */
ub = u0;
for (i=1; i<50; i++){
    u = u0 + (alpha - u0) * i / 50.0q;
    K(Ku, A, u, 30);
    aux = x[0];
    x[0]=Ku[0];

```



```

x[1]=Ku[1];
ua = ub;
ub = u;
for (j=1; j<=potencia; j++){
    xjet [0][0]=x [0];
    xjet [1][0]=x [1];
    t=0.0q;
    tf=2.0q*M.PIq*eps;
    while ( taylor_step_pdj(&t ,x,1,1,-34,-34,&tf ,NULL,NULL,xjet )==0);
}
if (x[0] > 0.0){
    /*Hem trobat el punt que creua x=0.
    Aquest i l'anterior son els que busquem */
    break;
}
}

printf(" Interval reduit: [ ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe",ua);
    printf("%s , ",buf);
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe",ub);
    printf("%s ]\n\n",buf);

    printf(" Component x imatges per P^pot dels extrems de l'interval: \n");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe",aux);
    printf(" %s , ",buf);
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe",x[0]);
    printf("%s \n\n",buf);

/* Metode de la secant */
/* Busquem el lambda tq P^potencia(K(ua*(1.0q-lambda) + ub*(lambda))) sigui 0 */
lambda1 = 0.0q;
lambda2 = 1.0q;
while ( err > 1e-30q){
    K(Ku, A, ua*(1.0q-lambda1) + ub*(lambda1), 30);

    /* Calculem les seves imatges per P^potencia */
    x[0] = Ku[0];
    x[1] = Ku[1];
    for (j=1; j<=potencia; j++){
        xjet [0][0]=x [0];
        xjet [1][0]=x [1];
        t=0.0q;
        tf=2.0q*M.PIq*eps;
        while ( taylor_step_pdj(&t ,x,1,1,-34,-34,&tf ,NULL,NULL,xjet )==0);
    }
    f1 = x[0];

    u = ua*(1.0q-lambda2) + ub*(lambda2);
    K(Ku, A, u, 30);
    x[0] = Ku[0];
    x[1] = Ku[1];
    for (j=1; j<=potencia; j++){
        xjet [0][0]=x [0];
        xjet [1][0]=x [1];
        t=0.0q;
        tf=2.0q*M.PIq*eps;
        while ( taylor_step_pdj(&t ,x,1,1,-34,-34,&tf ,NULL,NULL,xjet )==0);
    }
    f2 = x[0];
    /*
    printf(" lambda1:");
    quadmath_snprintf(buf, sizeof(buf), "%35.30Qe",lambda1);
        printf("%s\n",buf);
        printf(" f1:");
    quadmath_snprintf(buf, sizeof(buf), "%35.30Qe",f1);
        printf("%s\n",buf);
        printf(" lambda2:");

```

```

quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", lambda2);
    printf("%s\n", buf);
    printf(" f2:");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", f2);
    printf("%s\n", buf);
*/
aux = lambda2;
lambda2 = aux - f2*(aux-lambda1) / (f2-f1);
lambda1 = aux;

err = fabsq(f2);

}
/* Final metode sacant */
printf(" secant_resul(u): \n");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", u);
    printf("%s\n\n", buf);

K(Ku, A, u, 30);

printf("x P^pot(K(u_0)) : ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", x[0]);
    printf("%s\n\n", buf);

/* Angle */

/* Calcul DP^potencia */
pz[0] = Ku[0];
pz[1] = Ku[1];
pz[2] = 1.0q;
pz[3] = 0.0q;
pz[4] = 0.0q;
pz[5] = 1.0q;
for(j=1; j<=potencia; j++){
    /*Taylor amb DP*/
    t = 0.0q;
    tf=2.0q*M_PIq*eps;
    while(taylor_step_pendol(&t, pz, 1, 1, -34, -34, &tf, NULL, NULL, NULL) ==
0);
}

/* Vector tangent al punt de sortida*/
dK(dKu, A, u, 30);
norm1 = sqrtq(dKu[0]*dKu[0] + dKu[1]*dKu[1]);

/* Vector tangent a l'arribada */
vtanfinal[0] = pz[2] * dKu[0] + pz[3] * dKu[1];
vtanfinal[1] = pz[4] * dKu[0] + pz[5] * dKu[1];
norm2 = sqrtq(vtanfinal[0]*vtanfinal[0] + vtanfinal[1]*vtanfinal[1]);

angle = atanq(vtanfinal[1] / vtanfinal[0]);
angle = 2.0q * angle;

printf(" eps: ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", eps);
    printf("%s\n\n", buf);

printf(" Angle: ");
quadmath_snprintf(buf, sizeof(buf), "%24.16Qe", angle);
    printf("%s\n\n", buf);

printf(" Angle teoric: ");
quadmath_snprintf(buf, sizeof(buf), "%24.16Qe", (M_PIq/(2.0q*eps))*(mu/coshq(
M_PIq/(2.0q*eps))));
    printf("%s\n\n", buf);

printf(" la norma del vtan s'ha multiplicat per: ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", norm2/norm1);

```

```

printf("%s\n\n", buf);

/* TEST per a verificar els components d'ordre 2 de la parametrizacio
el resultat, errA/errB, hauria de ser 2^3 = 8 */
alpha = 1.0q;
__float128 errA = 1, errB;
while(errA > 1e-5q){
    alpha = alpha * 0.99q;
    K(Ku0, A, alpha, 2);
    K(Kvapu0, A, alpha*vap, 2);

    x[0]=Ku0[0];
    x[1]=Ku0[1];
    xjet[0][0]=x[0];
    xjet[1][0]=x[1];
    t=0.0q;
    tf=2.0q*M_PIq*eps;
    while (taylor_step_pdj(&t, x, 1, 1, -34, -34, &tf, NULL, NULL, xjet)==0);

    errA = (x[0]-Kvapu0[0])*(x[0]-Kvapu0[0]) + (x[1]-Kvapu0[1])*(x[1]-Kvapu0[1]);
    errA = sqrtq(errA);
}

K(Ku0, A, alpha/2.0q, 2);
K(Kvapu0, A, alpha*vap/2.0q, 2);

x[0]=Ku0[0];
x[1]=Ku0[1];
xjet[0][0]=x[0];
xjet[1][0]=x[1];
t=0.0q;
tf=2.0q*M_PIq*eps;
while (taylor_step_pdj(&t, x, 1, 1, -34, -34, &tf, NULL, NULL, xjet)==0);

errB = (x[0]-Kvapu0[0])*(x[0]-Kvapu0[0]) + (x[1]-Kvapu0[1])*(x[1]-Kvapu0[1]);
errB = sqrtq(errB);

printf(" errA/errB: ");
quadmath_snprintf(buf, sizeof(buf), "%35.30Qe", errA/errB);
printf("%s\n", buf);

fclose(fin);

return 0;
}

/* Funcio que retorna el valor de K(u) fins a un cert grau */
void K(__float128 Ku[2], __float128 A[2][31], __float128 u, int grau){
    int i;
    /* Avaluacio amb Horner */
    Ku[0] = A[0][grau];
    Ku[1] = A[1][grau];
    for(i=grau-1; i>=0; i--){
        Ku[0] = A[0][i] + Ku[0] * u;
        Ku[1] = A[1][i] + Ku[1] * u;
    }
}

/* Funcio que retorna el valor de K'(u) fins a un cert grau */
void dK(__float128 Ku[2], __float128 A[2][31], __float128 u, int grau){
    int i;
    /* Avaluacio amb Horner */
    Ku[0] = grau * A[0][grau];
    Ku[1] = grau * A[1][grau];
    for(i=grau-1; i>=1; i--){

```

```
    Ku[0] = i * A[0][i] + Ku[0] * u;  
    Ku[1] = i * A[1][i] + Ku[1] * u;  
  }  
}
```

programa3.c