

## 5 Introducció a la resolució numèrica d'edos

El propòsit d'aquestes notes és introduir l'alumne en alguns dels problemes bàsics que hom es troba quan vol simular numèricament molts sistemes dinàmics.

No dedicarem temps a la presentació i programació (que sempre resta com a possibilitat per a l'estudiant que ho desitgi) dels mètodes clàssics d'integració numèrica d'edos (d'un pas o múltiples) sino que farem servir directament un integrador, **Taylor**, dissenyat per n'À. Jorba (Universitat de Barcelona) i M. Zou (University of Texas). Aquest software, de lliure accés, amb codi obert i pensat per a nivell de recerca més que no pas de docència, ha demostrat els darrers anys un nivell molt d'alt de competitivitat (tenint en compte que no deixa de ser general i que hom pot crear sempre integradors més eficients *ad-hoc* per a un sistema concret) en quant a velocitat i precisió. Els algorismes que presentem en aquestes pàgines han estat directament estrets d'un article (un clàssic) d'en C. Simó [3] a on es mostra un ventall d'eines numèriques i analítiques essencials per a afrontar molts problemes en sistemes dinàmics.

### 5.1 El mètode de Taylor

Els darrers anys hi ha hagut un desenvolupament de programes que generen de manera automàtica rutines d'integració numèrica d'edos fent servir el mètode de Taylor, posant de manifest la seva potència computacional en comparació amb altres mètodes (Runge-Kutta, Adams-Bashforth, ...). Un d'aquests programes és l'anomenat **Taylor**, creat pels professors À. Jorba (Universitat de Barcelona) i M. Zou (University of Texas). El podeu descarregar des de l'adreça ([Taylor-home](#)). En el cas que no estigüés instal·lat al pc de l'aula o volguéssim instal·lar-lo a qualsevol altre ordinador seguiríem les instruccions indicades allà.

Seguirem la documentació de **Taylor** [1, 2], centrant-nos en un primer contacte amb ell i deixant per a l'estudiant interessat una lectura més profunda. Per facilitar-ne la comprensió ho aplicarem al càlcul (i dibuix) d'una òrbita d'un sistema concret. En el nostre cas, hem escollit **l'equació de Lorenz**.

El sistema de Lorenz és un exemple de sistema 3D amb comportament caòtic:

$$\begin{cases} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= xy - \beta z. \end{cases} \quad (1)$$

És ben coneguda l'existència d'un atractor per a determinats valors dels seus paràmetres:  $\sigma = 10$ ,  $\rho = 28$ ,  $\beta = 8/3$ . Aquest atractor va ser una de les primeres mostres de l'anomenat *butterfly effect* (efecte papallona). Podeu trobar fàcilment a internet vídeos que el mostren. Una ràpida tria podria ser, per exemple, aquesta: [video-1](#), [video-2](#), [video-3](#).

---

<sup>0</sup>T. Lázaro, J. R. Pacha i M. Ollé. Departament Matemàtica Aplicada I

Usarem Taylor per integrar-ho numèricament. El primer que hem de fer és definir un programa amb el cap de vectors. L'anomenarem `lorenz.eq1` i constarà de les següents línies:

```
RR=28.0;
diff(x,t)=10.0*(y-x);
diff(y,t)=RR * x - y - x*z;
diff(z,t)=x*y -(8.0/3.0)*z;
```

Graveu-lo a la vostra carpeta de treball. Ara hem de cridar a Taylor per a que creï el programa que efectuarà un pas simple d'integració numèrica del nostre sistema. Per fer-ho, escrivim:

```
perico@delospalotes: taylor -name lrnz -o lorenz.c -jet -step lorenz.eq1
perico@delospalotes: taylor -name lrnz -o taylor.h -header
```

La primera línia crea el fitxer `lorenz.c`, que conté el codi que calcula les derivades necessàries fins a l'ordre convenient del nostre camp (l'anomenat *jet of derivatives*, amb l'opció `-jet`) i el control de pas (`-step`). L'opció `-name` indica a Taylor que afegixi el suffix `lrnz` al nom del fitxer resultant, `taylor_step_lrnz`. La segona línia produeix el *header file* `taylor.h`, necessari per compilar `lorenz.c`. Un cop executades aquestes comandes tindreu a la vostra carpeta els següents fitxers:

```
perico@delospalotes: ls
lorenz.c lorenz.eq1 taylor.h
```

Tot i que Taylor també treballa amb precisió *estesa* nosaltres farem servir l'opció per defecte, la precisió doble del nostre ordinador. Podem cridar Taylor de diverses maneres:

### 5.1.1 Execució directa de Taylor

Taylor permet crear un programa simple que ens permet calcular els punts d'una òrbita concreta demanada per línia de comandes. Per fer-ho, escrivim

```
perico@delospalotes: taylor -name lrnz -o main_lrnz.c -main_only lorenz.eq1
```

el compilem

```
perico@delospalotes: gcc -O3 main_lrnz.c lorenz.c -lm -s
```

generant-se l'executable `a.out`. Si l'executem veurem que ens demana les condicions inicials de la nostra òrbita i les toleràncies desitjades:

```
perico@delospalotes: ./a.out
Enter Initial xx[0]: 0.1
Enter Initial xx[1]: 0.2
Enter Initial xx[2]: 0.3
```

```
Enter start time: 0.0
Enter stop time: 0.5
Enter absolute error tolerance: 0.1e-16
Enter relative error tolerance: 0.1e-16
```

La sortida per pantalla serà una taula amb els valors obtinguts  $x(t), y(t), z(t), t$ :

```
0.1 0.2 0.3 0
0.169464 0.36274 0.265313 0.0484829
0.309703 0.671984 0.236866 0.100502
0.539572 1.17353 0.225961 0.147831
0.88444 1.92431 0.245377 0.189871
1.38449 3.01108 0.322239 0.227996
2.0651 4.48536 0.503345 0.262065
2.96612 6.42411 0.86911 0.29303
4.11167 8.85801 1.53644 0.321165
5.50045 11.743 2.64885 0.346567
7.12416 14.9901 4.37751 0.369644
8.95632 18.4314 6.90055 0.39078
10.9434 21.7988 10.363 0.410287
13.0013 24.7287 14.8242 0.428432
15.005 26.7883 20.1692 0.445362
16.8064 27.5863 26.0763 0.46119
18.2881 26.8643 32.1801 0.476329
19.3086 24.5336 37.9112 0.490913
```

Observeu que la darrera dada que treu per pantalla correspon a temps 0.490913 i no pas 0.5, com havíem seleccionat. Això és degut a que el càlcul del nou punt es fa dins d'un bucle `do/while` amb la condició de que el temps emprat en aquell moment no passi el valor marcat a `stop_time`. Per a obtenir el valor corresponent a temps `stop_time` (0.5 en el nostre exemple) caldrà fer-li escriure un iterat més de l'integrador.

Si volguéssim representar-ho gràficament amb l'ajut del `gnuplot` podríem demanar que la sortida anés a un fitxer de dades:

```
perico@delospalotes: ./a.out > exemple-orbita-lrnz.dat
```

A més, per a que l'òrbita a representar sigui més llarga repetirem l'exemple anterior però amb `stop_time` igual a tt 30. Un cop fet això cridem a `gnuplot`:

```
perico@delospalotes$ gnuplot
```

```
G N U P L O T
```

```
Version 4.4 patchlevel 2
last modified Wed Sep 22 12:10:34 PDT 2010
System: Linux 2.6.38-9-generic-pae
```

```
Copyright (C) 1986-1993, 1998, 2004, 2007-2010
Thomas Williams, Colin Kelley and many others
```

```
gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:    type "help seeking-assistance"
immediate help:    type "help"
plot window:       hit 'h'
```

```
Terminal type set to 'wxt'
gnuplot> set term postscript
Terminal type set to 'postscript'
Options are 'landscape noenhanced defaultplex \
  leveldefault monochrome colortext \
  dashed dashlength 1.0 linewidth 1.0 butt noclip \
  palfuncparam 2000,0.003 \
  "Helvetica" 14 '
```

```
gnuplot> set output "./exemple-orbita.ps"
gnuplot> splot "exemple-orbita-lrnz.dat" u 1:2:3 with lines
gnuplot> quit
perico@delospalotes:
```

Si obrim el fitxer `exemple-orbita.ps` amb l'okular, per exemple o qualsevol visualitzador de postscripts, obtenim:

**Comentari 5.1** *Un manera alternativa de generar un petit programa principal `lorenz.c` que contingui el header, el codi de control de pas i el programa que generi el jet de derivades és fer*

```
perico@delospalotes: taylor -o lorenz.c lorenz.eq1
```

*Compilant-lo (linkant-hi la llibreria matemàtica):*

```
perico@delospalotes: gcc -O3 lorenz.c -lm
```

*obtenim un fitxer executable `a.out` com abans.*

### 5.1.2 Execució de Taylor dins d'un programa

Per fer-ho, cal inicialment demanar a Taylor que generi l'integrador i el fitxer *header* `taylor.h`:

"exemple-orbita-lrnz.dat" u 1:2:3 ———

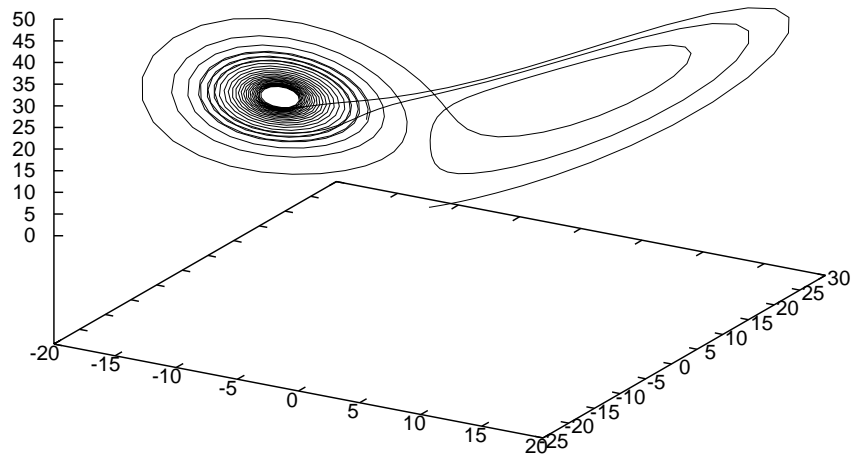


Figure 1: Una òrbita del sistema de Lorenz

```
perico@delospalotes: taylor -o lorenz.c -jet -step -name lorenz lorenz.eq1
perico@delospalotes: taylor -o taylor.h -header
```

Aquesta primera línia genera un fitxer `lorenz.c` que haurem de *linkar* i compilar juntament amb el fitxer que contingui l'algorisme de crida a l'integrador i que anomenarem, per exemple, `main1.c`:

```
#include "math.h"
#include "stdio.h"
#include "taylor.h"

#define max(a,b) ((a)<(b) ? (b) : (a))
#define sgn(a)   ((a)<0 ? -1 : 1 )
#define abs(a)   ((a)<0 ? -(a) : (a))

// Taylor control parameters
#define H_INICIAL 0.001
#define ABS_ERR 1.0e-16
#define REL_ERR 1.0e-16

int main(int argc, char **argv)
```

```
{
// xx[]: variables espaials, t: temps
double xx[3], t;
// Constants al Taylor
double h, abs_err, rel_err, h_return, log10abs_err;
double log10rel_err, endtime;
int nsteps = 20, step_ctrl_method = 2, direction = 1;
int order = 10;
//
int nombre_passos; // nombre de passos d'integraci

// Fixem les condicions inicials
xx[0] = 0.1;
xx[1] = 0.2;
xx[2] = 0.3;
t = 0.0;

// Assignem parametres de control de Taylor
h= (double) H_INICIAL;
abs_err = (double) ABS_ERR ;
rel_err = (double) REL_ERR;
log10abs_err = log10(abs_err);
log10rel_err = log10(rel_err);

// =====
// Exemple 1: integrem nombre_passos passos de l'edo fins a arribar, com a maxim,
// a temps endtime

endtime = 100.;

nombre_passos=20;
printf("\n Exemple 1: integrem la nostra edo %d passos \n\n",nombre_passos);

h_return=h;
while( -- nsteps >= 0 && h_return != 0) {
// Per exemple treiem els resultats per pantalla
printf("x:%f \ty:%f \tz:%f \tt:%f \th:%e \tpas:%d\n",
      xx[0],xx[1],xx[2],t,h_return,(nombre_passos-nsteps));

taylor_step_lorenz(&t, &xx[0], direction,
```

```
    step_ctrl_method,log10abs_err, log10rel_err,
    &endtime, &h_return, &order);
}

// =====
// Exemple 2: integrem fins a arribar a endtime

endtime = 2.0;
printf("\n\n\n Exemple 2: integrem la nostra edo fins a temps %g \n\n",endtime);
// Tornem a les condicions inicials
xx[0] = 0.1;
xx[1] = 0.2;
xx[2] = 0.3;
t = 0.0;
h= (double) H_INICIAL;

h_return=h;
while( (t<endtime) && h_return != 0) {
    // Per exemple treiem els resultats per pantalla
    printf("x:%f \ty:%f \tz:%f \tt:%f \th:%e\n", xx[0],xx[1],xx[2],t,h_return);

    taylor_step_lorenz(&t, &xx[0], direction,
    step_ctrl_method,log10abs_err, log10rel_err,
    &endtime, &h_return, &order);
}
printf("x:%f \ty:%f \tz:%f \tt:%f \th:%e\n", xx[0],xx[1],xx[2],t,h_return);

return(0);
}
```

Un cop gravat aquest fitxer `main1.c`, els linkem i compilem, generant el nostre executable `IntLorenz`:

```
perico@delospalotes: gcc -o IntLorenz lorenz.c main1.c -lm
```

Recordeu que si no especifiquem l'executable de sortida, per defecte el nom que dona `gcc` és `a.out`. Per executar el programa només cal fer

```
perico@delospalotes: ./IntLorenz
```

i sortirà per pantalla una taula similar a l'obtinguda a la secció anterior.

Taylor també permet modificar des del fitxer principal els paràmetres que puguin aparèixer en el nostre camp vectorial. Fem-ho amb l'exemple `lorenz.eq1` que abans teníem escrit així:

```
RR=28.0;
diff(x,t)=10.0*(y-x);
diff(y,t)=RR * x - y - x*z;
diff(z,t)=x*y -(8.0/3.0)*z;
```

Caldrà ara definir les variables RR i coef[i],  $i = 0, 1, 2$  com a externes:

```
extern MY_FLOAT RR, coef[3];
diff(x,t)=coef[0]*(y-x);
diff(y,t)=RR * x - y - x*z;
diff(z,t)=x*y -(coef[1]/coef[2])*z;
```

I al nostre programa principal main1.c haurem d'escriure

```
#include "math.h"
#include "stdio.h"
#include "taylor.h"

#define max(a,b) ((a)<(b) ? (b) : (a))
#define sgn(a) ((a)<0 ? -1 : 1 )
#define abs(a) ((a)<0 ? -(a) : (a))

// Taylor control parameters
#define H_INICIAL 0.001
#define ABS_ERR 1.0e-16
#define REL_ERR 1.0e-16

// Variables externes. Parametres del camp vectorial.
MY_FLOAT RR, coef[3];

int main(int argc, char **argv)
{
// xx[]: variables espaials, t: temps
double xx[3], t;
// Constants al Taylor
double h, abs_err, rel_err, h_return, log10abs_err;
double log10rel_err, endtime;
int nsteps = 20, step_ctrl_method = 2, direction = 1;
int order = 10;
.....
RR=28.0;
```

```
coef[0]=10.0; coef[1]=8.0; coef[2]=3.0;
```

```
.....
```

Trobareu més informació sobre les diverses opcions possibles al manual de Taylor [1].

## 5.2 Càlcul de l'aplicació de Poincaré

Aquesta subsecció i les posteriors van dirigides principalment al càlcul d'òrbites periòdiques, elements importants dins l'*esquelet* del retrat de fases d'un sistema d'equacions diferencials. Seguirem essencialment el capítol 5 de [3], tot i que simplificat i adaptat al nostre cas. Començarem introduint l'anomenada *aplicació de Poincaré* que esdevindrà una eina bàsica d'aquest estudi.

Considerem l'equació diferencial  $\dot{x} = f(x)$  amb  $x \in \mathcal{U} \subset \mathbb{R}^n$ , essent  $\mathcal{U}$  un obert. Sigui  $\phi(t, x)$  el seu flux associat, és a dir, verificant que

$$\frac{d}{dt}\phi(t, x) = f(\phi(t, x)), \quad \phi(0, x) = x.$$

Sabem que si  $f \in \mathcal{C}^r$  llavors  $\phi \in \mathcal{C}^r$ . Sigui  $\Sigma$  una hipersuperfície (secció) transversal al flux definida com  $\Sigma = \{g(x) = 0\}$ , amb  $g$  prou regular. El fet que sigui transversal a un punt  $x$  implica que el producte escalar  $\langle \nabla g(x), f(x) \rangle \neq 0$  a tot punt de  $\Sigma$ . Definim aleshores la següent *aplicació de Poincaré*<sup>1</sup>:  $P : \Sigma \rightarrow \Sigma$  a on  $\bar{x} = P(x) = \phi(t(x), x) \in \Sigma$  i ho fa amb la mateixa orientació respecte de  $\Sigma$  de com ha sortit d' $x$  (vegeu Fig 2). Recordeu que una manera de comprovar que l'orientació

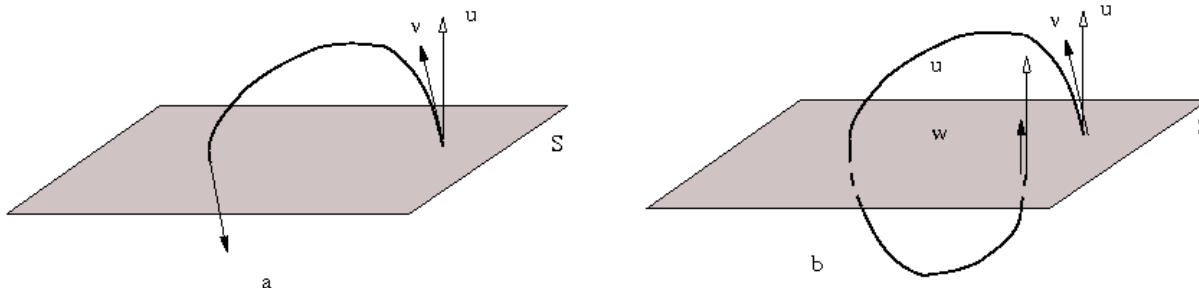


Figure 2: a) Intersecció del flux amb  $\Sigma$  amb orientació invertida; b) Idem però amb la mateixa orientació.

d'arribada és la mateixa que la de sortida és que

$$\langle \nabla g(x), f(x) \rangle \cdot \langle \nabla g(\bar{x}), f(\bar{x}) \rangle > 0.$$

La hipersuperfície  $\Sigma$  s'anomena *secció de Poincaré*. Òbviament, el temps  $t(x)$  necessari fins arribar a  $\Sigma$  (amb la mateixa orientació) depèn del valor inicial  $x$  i cal calcular-ho cada vegada. Noteu que  $t(x)$  és solució de  $\psi(t) = g(\phi(t, x)) = 0$ . Per tant, per trobar-lo podem procedir de la següent manera:

<sup>1</sup>N'hi ha de diferents tipus, segons si ens interessa considerar el primer, segon, tercer, ... tall amb  $\Sigma$  o bé si el que volem és deixar actuar el flux  $\phi(t, x)$  durant un cert temps fixat, etc

- (i) Fent servir Taylor, calculem iterats del flux  $x^{(1)}, x^{(2)}, \dots$  on  $x^{(j)} = \phi(t^{(j)}, x^{(0)})$  fins a arribar al primer iterat  $k$  que satisfà les següents dues condicions:

$$g(x^{(k)}) \cdot g(x^{(k-1)}) < 0, \quad \langle \nabla g(x^{(k)}), f(x^{(k)}) \rangle \cdot \langle \nabla g(x^{(k-1)}), f(x^{(k-1)}) \rangle > 0.$$

La primera equival a dir que acabem de travessar  $\Sigma$ . La segona que ho hem fet amb la mateixa orientació que de sortida. Per tant, sabem que l'arrel  $t(x)$  de  $\psi(t) = g(\phi(t, x))$  que estem buscant es troba dins l'interval  $[t^{(k-1)}, t^{(k)}]$ .

- (ii) Apliquem algun esquema iteratiu que ens permeti refinar aquesta arrel de  $\psi$ . Tenim diferents opcions: bisecció, secant, regula-falsi, Newton, ... Ho farem usant Newton doncs si l'interval inicial és prou petit la convergència serà quadràtica. Prenem com a punt de partida el valor  $x^{(k-1)} = \phi(t^{(k-1)}, x)$  i tenint en compte que el flux compleix que  $\phi(t + s, x) = \phi(s, \phi(t, x))$ , l'esquema iteratiu corresponent és

$$s^{(m+1)} = s^{(m)} - \frac{\psi(s^{(m)})}{\psi'(s^{(m)})} = s^{(m)} - \frac{g(\phi(s^{(m)}, x^{(k-1)}))}{\langle \nabla g(\phi(s^{(m)}, x^{(k-1)})), f(\phi(s^{(m)}, x^{(k-1)}) \rangle}$$

amb condició inicial  $s^{(0)} = t^{(k-1)}$  (vegeu Fig. 3). Un cop obtingut  $s^* = \lim_m s^{(m)}$  tindrem que

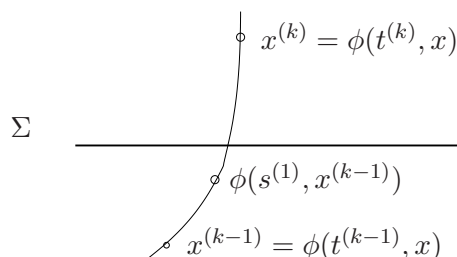


Figure 3: Refinant el punt de tall amb  $\Sigma$

el temps buscat és  $t(x) = t^{(k-1)} + s^*$ . Finalment,

$$P(x) = \phi(t(x), x) = \phi(s^*, \phi(t^{(k-1)}, x)) = \phi(s^*, x^{(k-1)}). \quad (2)$$

**Exercici 5.2** Considereu la següent família de sistemes d'equacions diferencials

$$\begin{cases} \dot{x} &= -f\left(\sqrt{x^2 + y^2}\right) y \\ \dot{y} &= f\left(\sqrt{x^2 + y^2}\right) x \end{cases} \quad (3)$$

on  $f$  és una funció  $C^\infty(\mathbb{R})$  tal que  $f(0) = 0$ .

1. Demostreu que en coordenades polars l'equació (3) s'escriu

$$\begin{cases} \dot{r} &= 0 \\ \dot{\theta} &= f(r) \end{cases}$$

L'origen és un punt d'equilibri (tipus centre) i totes les circumferències amb centre a l'origen són corbes invariants pel flux.

2. (i) Considereu  $f(u) = 1$ ,  $f(u) = u$  i  $f(u) = u^2$ . Useu Taylor i gnuplot per a dibuixar aproximadament el seu retrat de fases entorn a l'origen: preneu condicions inicials sobre l'eix  $x = 0$  i integreu l'edo per a temps positiu  $\tau$  i per a temps negatiu  $-\tau$  ( $\tau = 10$ , per exemple). Comproveu que surten circumferències concèntriques de centre  $(0,0)$ .
- (ii) Considereu la secció de Poincaré  $\Sigma = \{y = 0, x > 0\}$ . Demostreu que el flux és sempre transversal a tot punt de  $\Sigma$ . Definiu  $x_k = 0.1(1 + k)$ ,  $k = 0, 1, \dots, 99$  punts equiespaiats a l'interval  $[0, 10]$ . Per a cada  $x_k$  calculeu la seva imatge  $P(x_k)$  per l'aplicació de Poincaré. Comproveu que  $P(x_k) = x_k$  (això implica, en particular, que totes elles són òrbites periòdiques). Dibuixeu la funció  $t(x_k)$ , on  $t(x_k)$  és tal que  $P(x_k) = \phi(t(x_k), x_k)$ . Què podeu deduir-ne?
- (iii) Feu el mateix prenent  $f(u) = u$  i  $f(u) = u^2$ . Com són ara les gràfiques de  $t(x_k)$ ? Compareu-les amb les de les funcions  $2\pi/u$  i  $2\pi/u^2$ , respectivament.

En aquest exemple, totes les òrbites són periòdiques (o.p.) i, concretament, el valor  $t(x_k)$  coincideix amb el període de l'o.p. que passa per  $(x_k, 0)$ . La funció  $T(x) = t(x)$  s'anomena *funció de període*. És fàcil demostrar que  $T(x) = 2\pi/f(x)$  (penseu-ho en polars). Al cas  $f(u) = 1$  totes les o.p. tenen el mateix període, mentre que a la resta depèn de la distància a l'origen. Al primer cas es diu que l'origen és un *centre isòcron* (del grec, *iso*=igual, *kronos*=temps). A la resta de casos es diu que tenim a l'origen un *centre anisòcron*.

### 5.3 Càlcul d'òrbites periòdiques. Zeros de l'aplicació de Poincaré

Considereu l'anomenat *oscil·lador de Van der Pol*,

$$\ddot{x} + \varepsilon(x^2 - 1)\dot{x} + x = 0 \tag{4}$$

amb  $\varepsilon > 0$ . Aquesta equació va ser introduïda per Van der Pol l'any 1927 per modelitzar el comportament de determinats circuits elèctrics i constitueix un exemple clàssic dins la Teoria de sistemes dinàmics al pla. Per estudiar-ho i com és habitual, considerem el sistema d'ordre 1 associat,

$$\begin{cases} \dot{x} &= y, \\ \dot{y} &= -x + \varepsilon(1 - x^2)y, \end{cases} \tag{5}$$

que té l'origen com a punt d'equilibri. L'anomenat sistema no pertorbat,  $\varepsilon = 0$  correspon a un oscil·lador harmònic  $\ddot{x} + x = 0$ , les òrbites del qual són circumferències de centre  $(0,0)$  recorregudes amb freqüència angular 1. Es tracta doncs (vegeu Problema 5.2) d'un centre isòcron.

Per  $\varepsilon \neq 0$  l'origen és també un punt d'equilibri, tot i que aquesta vegada inestable. És ben coneguda l'existència d'una única òrbita periòdica aïllada a l'entorn del  $(0,0)$ . Aquest tipus d'o.p. s'anomenen *cicles-límit*. En el nostre cas, aquest cicle-límit és atractor, és a dir, totes les trajectòries amb valors inicials a prop d'ell tendeixen a ell quan  $t \rightarrow +\infty$ . A la Figura 4 es mostren algunes trajectòries (parcials, amb c.i. a sobre de la recta  $\{y = 0\}$ ), tendint cap a aquest cicle-límit.

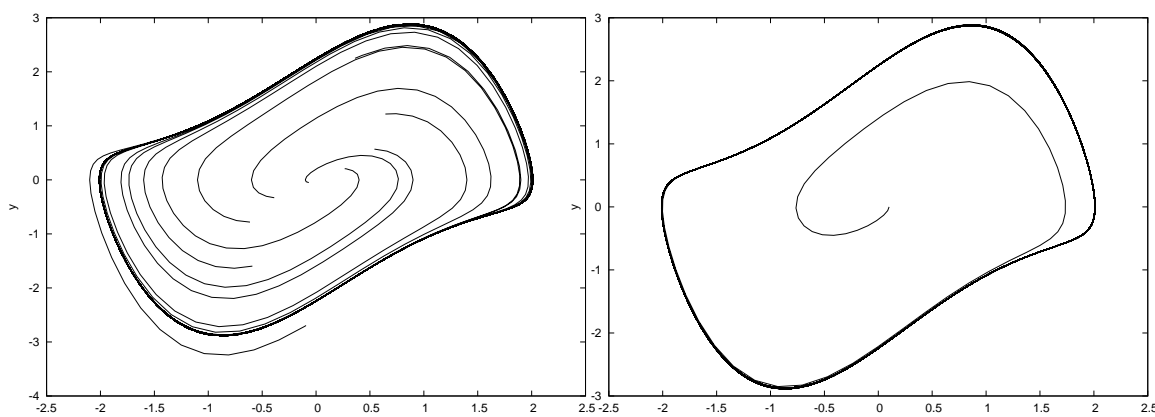


Figure 4: Algunes trajectòries de l'oscil·lador de Van der Pol  $\ddot{x} + \varepsilon(x^2 - 1)\dot{x} + x = 0$  amb  $\varepsilon = 1.2$

**Exercici 5.3** Considereu l'oscil·lador de Van der Pol, escrit en la forma (5).

(i) Preneu c.i. del tipus  $(x_k, 0)$ ,  $x_k = -2.1 + k * 0.5$ ,  $k = 0 \div 8$  i trobeu el flux solució fins a temps  $t_f = 100$ . Feu el mateix integrant cap a darrere fins a  $t_f = -100$ .

(ii) Calculeu l'òrbita de condicions inicials  $(0.1, 0)$  fins a temps  $t_f = 1000$ . Què s'observa?

Indicacions:

1.- Escriviu un fitxer `vdpol.eq1` amb les equacions del nostre sistema.

2.- Demaneu a Taylor que generi l'integrador i el header corresponents:

```
perico@delospalotes: taylor -o vanderpol.c -jet -step -name vanderpol vdpol.eq1
perico@delospalotes: taylor -o taylor.h -header
```

3.- Feu els programes que es demanen als apartats (i),(ii) anteriors, escrivint els resultats en dos fitxers anomenats `retrat-fase-vdpol.dat` i `una-orbita-vdpol.dat`, respectivament. Recordeu que, a

*banda de la compilació i linkat des de línia de comandes, podeu fer-ho també a través d'un makefile. Per exemple, en el nostre cas, aquest fitxer, que anomenarem makefile-retrat-fases-vdpol, podria ser:*

```
CODIGO:= retrat-fases-vdpol.c vanderpol.c
CC:= gcc
CFLAGS:= -g -O3 -s
ENLACE:=$(CFLAGS) -lm
EJECUTABLE:=retrat-vdpol

OBJETOS:=$(CODIGO:.c=.o)
OBJETOS:=$(OBJETOS:.f=.o)

all:$(EJECUTABLE)

$(EJECUTABLE):$(OBJETOS)
$(CC) -o $(EJECUTABLE) $(OBJETOS) $(ENLACE)

clean:
rm -f $(OBJETOS) $(EJECUTABLE)
```

*Per executar-ho feu*

```
perico@delospalotes: make -f makefile-retrat-fases-vdpol
```

*4.- Un cop teniu els fitxers .dat amb els punts de les trajectòries demanades, dibuixeu-les amb el gnuplot. Podeu fer-ho directament a través de línia de comandes (vegeu Secció 5.1.1) o bé usar una shell: escriviu i deseu com a dibuixa-retrat-fases-vanderpol.sh el següent conjunt d'instruccions*

```
/usr/bin/gnuplot << EOF

set term postscript

set xlabel "x"
set ylabel "y"

set output "./retrat-de-fases-vanderpol.ps"
plot "./retrat-fase-vdpol.dat" u 2:3 notitle w l lw 0.1 lt 1

set output "./una-orbita-vdpol.ps"
```

```

plot "./una-orbita-vdpol.dat" u 2:3 notitle w l lw 0.1 lt 1

# Un exemple amb comandes no abreujades seria:
# plot "./retrat-fase-vdpol.dat" using 2:3 title 'Retrat de fases'
# with points point-type 1 point-size 0.4 line-type 1, 0 with lines
# line-width 0.1 line-type 1 notitle

# Recordeu que # indica comentari i la linia corresponent es obviada
# pel sistema.

```

EOF

*Per executar-ho feu*

```
perico@delospalotes: sh dibuixa-retrat-fases-vanderpol.sh.
```

*Generarà dos fitxers postscript que podreu visualitzar amb okular, ghostview, gv, etc.*

El nostre objectiu final en aquesta secció serà trobar numèricament el cicle-límit de l'oscil·lador de Van der Pol (que podeu intuir a la Figura 4), esbrinar el seu període i la seva estabilitat. Per fer-ho ens caldrà conèixer quina és la dependència de les solucions del nostre sistema (5) respecte a condicions inicials, determinada per l'anomenada *equació variacional*. Com ja hem comentat anteriorment seguirem [3].

### 5.3.1 Primera Equació Variacional d'una edo

Considerem una edo autònoma (és a dir, sense dependència explícita en el temps  $t$  al camp de vectors)

$$\dot{x} = f(x), \quad x \in \mathcal{U} = \overset{\circ}{\mathcal{U}} \subset \mathbb{R}^n, \quad f \in \mathcal{C}^r(\mathcal{U}). \quad (6)$$

Si no diem res en contra, durant aquesta secció i les posteriors, suposarem que  $\mathcal{U}$  i  $f$  són d'aquesta forma. Sigui  $\phi(t, x)$  la solució de (6) satisfent que  $\phi(0, x) = x$ . Sabem que si  $f \in \mathcal{C}^r$  aleshores  $\phi \in \mathcal{C}^r$  i, per tant, té sentit fer la seva expansió en sèrie de Taylor entorn a  $x$ :

$$\phi(t, x + h) = \phi(t, x) + D_x \phi(t, x)h + \frac{1}{2} D_x^2 \phi(t, x)h^2 + \dots$$

S'anomenen *equacions variacionals* a aquelles equacions diferencials (lineals) satisfetes per les diferencials  $D_x^j \phi(t, x)$ . En particular, per a  $j = 1$  ens proporciona com varien les solucions de (6) en funció de les seves c.i. Una manera senzilla de deduir aquestes equacions és fer servir que  $\phi(t, x)$  és solució, és a dir,

$$\frac{d}{dt} \phi(t, x) = f(\phi(t, x)).$$

Diferenciant aquesta expressió respecte de  $x$ ,

$$D_x \left( \frac{d}{dt} \phi(t, x) \right) = Df(\phi(t, x)),$$

conmutant ambdues derivacions (respecte de  $t$  i de  $x$ , atès que  $f \in \mathcal{C}^r$ ) i tenint en compte que  $\phi(0, x) = x$ , s'obté la *primera equació variacional*:

$$(EV) : \begin{cases} \frac{d}{dt} D_x \phi(t, x) = Df(\phi(t, x)) D_x \phi(t, x), \\ D_x \phi(0, x) = \text{Id}, \end{cases} \quad (7)$$

on Id denota la matriu identitat  $n$ -dimensional. De manera anàloga s'anirien trobant les equacions variacionals d'ordre superior. Des d'un punt de vista computacional, és important destacar que s'integra numèricament (EV) juntament amb l'edo  $\dot{x} = f(x)$ , és a dir, en realitat s'integra el sistema ampliat

$$\frac{d}{dt} \begin{pmatrix} x \\ D_x \phi(t, x) \end{pmatrix} = \begin{pmatrix} f(x) \\ Df(\phi(t, x)) D_x \phi(t, x) \end{pmatrix}$$

amb condicions inicials  $\phi(0, x) = x$  i  $Z(0) = \text{Id}$ . Per exemple, per  $n = 2$  el corresponent camp vectorial s'escriu

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \\ Df(\phi(t, x))Z \end{pmatrix} \quad (8)$$

amb

$$Z(t) = \begin{pmatrix} z_{11}(t) & z_{12}(t) \\ z_{21}(t) & z_{22}(t) \end{pmatrix}, \quad Df(x_1, x_2) = \begin{pmatrix} D_1 f_1(x_1, x_2) & D_2 f_1(x_1, x_2) \\ D_1 f_2(x_1, x_2) & D_2 f_2(x_1, x_2) \end{pmatrix}.$$

Aleshores, a cada punt  $t_k$  cal integrar primer l'equació (8), trobar-ne el valor aproximat del flux  $\phi(t_k, x) = (\phi_1(t_k, x), \phi_2(t_k, x))$  i amb ell obtenir-ne  $Z(t_k)$  solució de  $\dot{Z} = Df(\phi(t_k, x))Z$ .

**Exercici 5.4** Trobeu el valor aproximat de  $\phi(t_f, x)$  i de la seva primera variacional (EV) a temps  $t_f = 10, 100, 1000$  i condicions inicials  $x = (0.2, 0)$  i  $x = (-1.5, 2)$ .

### 5.3.2 Càlcul de la diferencial de l'aplicació de Poincaré

Donada l'edo  $\dot{x} = f(x)$  i una secció  $\Sigma = \{g(x) = 0\}$  transversal al seu flux, considerem la seva aplicació de Poincaré associada (vegeu Secció 5.2):

$$\begin{aligned} P : \Sigma &\longrightarrow \Sigma \\ x &\longmapsto P(x) := \phi(t(x), x) \end{aligned}$$

de manera que es verifica que  $g(\phi(t(x), x)) = 0$ . La funció  $t(x)$  és el temps necessari per a que l'òrbita amb c.i.  $x$  intersequi de nou i en el sentit establert (vegeu Figura 2) la hipersuperfície  $\Sigma$ . Volem

calcular la diferencial de l'aplicació de Poincaré  $P$  determinada per  $\Sigma$ . Diferenciant la seva definició obtenim

$$DP(x) = \frac{\partial}{\partial t} \phi(t(x), x) \frac{\partial t(x)}{\partial x} + D_x \phi(t(x), x) = f(\phi(t(x), x)) \frac{\partial t(x)}{\partial x} + D_x \phi(t(x), x). \quad (9)$$

El terme  $D_x \phi(t(x), x)$  és la solució de la (EV) per a temps  $t(x)$  (vegeu Secció 5.3.1) i, per tant, computable. Necessitem doncs, calcular  $\partial t(x)/\partial x$ . Per fer-ho usem que  $\phi(t(x), x) \in \Sigma$ , és a dir,  $g(\phi(t(x), x)) = 0$ . Diferenciant-la,

$$D_x (g(\phi(t(x), x))) = 0 \Rightarrow D_x g(\phi(t(x), x)) \cdot \left( f(\phi(t(x), x)) \frac{\partial t(x)}{\partial x} + D_x \phi(t(x), x) \right) = 0.$$

Si denotem  $\bar{x} = \phi(t(x), x)$ , aleshores, de l'anterior expressió s'obté

$$D_x g(\bar{x}) \cdot \left( f(\bar{x}) \frac{\partial t(x)}{\partial x} + D_x \phi(t(x), x) \right) = 0 \Rightarrow$$

$$\langle \nabla g(\bar{x}), f(\bar{x}) \rangle \frac{\partial t(x)}{\partial x} + Dg(\bar{x}) D_x \phi(t(x), x) = 0 \Rightarrow$$

$$\frac{\partial t(x)}{\partial x} = -\frac{1}{\langle \nabla g(\bar{x}), f(\bar{x}) \rangle} Dg(\bar{x}) D_x \phi(t(x), x),$$

que està ben definida ja que  $\langle \nabla g(\bar{x}), f(\bar{x}) \rangle \neq 0$  en ser  $\Sigma$ , per hipòtesi, transversal al flux. Substituint aquesta expressió a l'equació (9) ens queda

$$DP(x) = \left( -\frac{1}{\langle \nabla g(\bar{x}), f(\bar{x}) \rangle} f(\bar{x}) Dg(\bar{x}) + \text{Id} \right) D_x \phi(t(x), x), \quad (10)$$

essent Id la matriu identitat  $n$ -dimensional.

Anem a il·lustra-ho amb un exemple. Suposeu  $x \in \mathbb{R}^n$  i que la nostra secció de Poincaré  $\Sigma$ , transversal al flux al domini considerat, vé definida  $g(x) = x_n$ , la  $n$ -èsima component d' $x$ . És a dir,  $\Sigma = \{x \in \mathbb{R}^n \mid x_n = 0\}$ . Aleshores, a (10) simplificant, tindrem que  $\nabla g(x) = (0, 0, \dots, 1) = e_n$ ,  $\langle \nabla g(\bar{x}), f(\bar{x}) \rangle = f_n(\bar{x})$ , a on  $f = (f_1, f_2, \dots, f_n)$ , i  $Dg(\bar{x}) = (0 \ 0 \ 0 \ \dots \ 1)$ . A més,

$$f(\bar{x}) \cdot Dg(\bar{x}) = \begin{pmatrix} 0 & 0 & \dots & 0 & f_1(\bar{x}) \\ 0 & 0 & \dots & 0 & f_2(\bar{x}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & f_n(\bar{x}) \end{pmatrix}$$

Per tant,

$$DP(x) = \begin{pmatrix} 1 & 0 & \dots & 0 & -f_1(\bar{x})/f_n(\bar{x}) \\ 0 & 1 & \dots & 0 & -f_2(\bar{x})/f_n(\bar{x}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -f_{n-1}(\bar{x})/f_n(\bar{x}) \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} D_x \phi(t(x), x) =: A \cdot Z,$$

a on  $Z = D_x\phi(t(x), x)$  és la solució de l'equació variacional (EV) (vegeu (7)) i  $A = (a_{ij})_{1 \leq i, j \leq n}$  té coeficients

$$\begin{cases} a_{ii} &= 1 \\ a_{in} &= -\frac{f_i(\bar{x})}{f_n(\bar{x})} \end{cases} \quad \text{per } i = 1, 2, \dots, n-1,$$

i  $a_{ij} = 0$  a la resta de casos. Si denotem  $Z = (z_{ij})_{1 \leq i, j \leq n}$  i  $DP(x) = (q_{ij})_{1 \leq i, j \leq n}$  aleshores els coeficients de la diferencial de Poincaré s'obtenen fent  $q_{ij} = \sum_{k=1}^n a_{ik}z_{kj}$ .

### 5.3.3 Càlcul d'òrbites periòdiques

Es vol calcular numèricament òrbites periòdiques d'un cert període  $T$  d'un sistema (autònom)  $\dot{x} = f(x)$ ,  $x \in \mathbb{R}^n$ . La primera idea seria la de cercar condicions inicials  $x_0$  que siguin zero de la funció  $G(x_0) = \phi(T, x_0) - x_0$  i aplicar, per fer-ho, el mètode de Newton (falta referència). Malauradament (o no), si  $x_0$  satisfà que  $\phi(T, x_0) = x_0$  llavors també ho verifica tot punt de la seva òrbita  $\{\phi(t, x_0)\}_{t \in \mathbb{R}}$ . Això provoca, en particular, que el vector tangent a l'òrbita al punt  $x_0$ ,  $f(x_0)$ , sigui vector propi de valor propi 1 de la *matriu de monodromia*  $D\phi(T, x_0)$ . Conseqüentment, la matriu  $DG(x_0) = D\phi(T, x_0) - \text{Id}$  és singular. Per tant no podem aplicar directament el mètode de Newton i caldria buscar alguna de les seves variants. No serà aquesta l'opció que aplicarem en aquesta secció sinó que buscarem treure profit de l'aplicació de Poincaré definida anteriorment.

Sigui de nou  $\Sigma$  una secció transversal al flux de  $\dot{x} = f(x)$ . Observeu que buscar una òrbita periòdica (que intersequi  $\Sigma$ , és clar) és equivalent a cercar un punt  $x_0 \in \Sigma$  de forma que  $P(x_0) = x_0$ , on  $P$  és l'aplicació de Poincaré associada a  $\dot{x} = f(x)$  i a  $\Sigma$ . Noteu, a més, que el període de l'òrbita periòdica que estem buscant no està ara prefixat. Si  $P(x_0) = x_0$  tindrem que  $T = t(x_0)$ , essent  $t(x_0) >$  el temps que fa  $\phi(t(x_0), x_0) \in \Sigma$  amb l'orientació fixada. Per tant, aplicarem el mètode de Newton a la funció  $G(x_0) = P(x_0) - x_0$ , que té per diferencial  $DG(x_0) = DP(x_0) - \text{Id}$ . Recordeu que a la Secció 5.3.2 ja vàrem explicar com calcular-la.

Cal tenir present que, de vegades, la diferencial  $DP(x_0)$  de l'aplicació de Poincaré té 1 com a valor propi i, llavors,  $DG(x_0)$  és singular. Aquest fet és sovint motivat per l'existència d'*integrals primeres*<sup>2</sup> del nostre sistema  $\dot{x} = f(x)$ . Un exemple clàssic d'integral primera és l'energia  $E$  (potencial + cinètica) d'un sistema mecànic no dissipatiu. En aquests casos caldrà restringir l'estudi considerant punts d'un nivell d'energia (o de la integral primera) fixat,  $E = E_0$ .

L'ús de l'aplicació de Poincaré  $P$  en el càlcul d'òrbites periòdiques de  $\dot{x} = f(x)$  esdevé també de gran utilitat per a determinar la seva estabilitat lineal. De fet, l'estabilitat de l'òrbita periòdica associada al punt  $x_0 \in \Sigma$  és equivalent a l'estabilitat lineal de  $x_0$  com a punt fix de l'aplicació  $P$ . Així, si tot *vap* de  $DP(x_0)$  verifica que  $|\lambda| < 1$  o si  $|\lambda| = 1$  que sigui simple, aleshores  $x_0$  és *estable*

<sup>2</sup>Recordeu que una funció escalar  $F$  és una integral primera de  $\dot{x} = f(x)$  si és constant sobre cada una de les seves solucions:  $\frac{d}{dt} F(\phi(t, x)) = 0$ .

i, per tant, també la seva òrbita (periòdica). D'igual manera, si algun *vap*  $\lambda$  té mòdul més gran que 1 llavors ambdós són *inestables*.

**Exercici 5.5** Preneu  $\Sigma = \{y = 0\}$  com a secció de Poincaré per a l'oscil·lador de Van der Pol (4) i trobeu l'única òrbita periòdica (aïllada, és a dir, un cicle-límit) per a diversos valors del paràmetre  $\varepsilon$  (per exemple: 1.2, 0.5, 0.25, 0.01.). Estudieu la seva estabilitat (lineal).

## Bibliografia

- [1] À. Jorba, M. Zou A software package for the numerical integration of ODE by means of high-order Taylor methods, ([pdf](#))
- [2] À. Jorba, M. Zou Taylor User's Manual, ([pdf](#))
- [3] C. Simó. On the analytical and numerical approximation of invariant manifolds. Reprinted from *Les Méthodes Modernes de la Mécanique Céleste* (course given at Goutelas, France, 1989), D. Benest and C. Froeschlé (eds.), 285–329, Editions Frontières, Paris, 1990. ([gzip](#))