

## Capítulo 3

# Continuación numérica de equilibrios.

En el capítulo anterior, obtuvimos las ecuaciones diferenciales de movimiento, así como una curva representativa de la resonancia 2:1 y las posibles situaciones de equilibrio para nuestro problema, todo ello de forma analítica.

En este capítulo, mediante el programa de continuación numérica AUTO2000, y a partir de las ecuaciones diferenciales de movimiento y de la curva representativa para la resonancia 2:1, vamos a realizar en primer lugar un estudio de las situaciones de equilibrio, y a partir de éste pasaremos a la continuación numérica de órbitas periódicas estudiando para ellas el comportamiento de nuestro sistema.

### 3.1. Introducción.

Para realizar la continuación numérica de una función escalar o vectorial, lo primero que debemos saber es que el número de incógnitas sólo puede ser uno más que el de ecuaciones, es decir, que sólo podemos tener un parámetro de continuación. Esto se debe, a que la continuación numérica, se encarga de evaluar nuestra función para cada valor del parámetro, y así se va obteniendo una serie de puntos que definen nuestros equilibrios u órbitas periódicas, según sea el caso. Una excepción al caso anterior, son los puntos especiales como *puntos límite*, *puntos de bifurcación*, *puntos de Hopf*, etc, ya que cuando se continúa uno de estos puntos, implícitamente se impone una ecuación más (un autovalor cero en caso de puntos límite, por ejemplo), lo cuál nos permite variar dos parámetros en lugar de uno sólo.

En el caso concreto de nuestro sistema, las ecuaciones que rigen el movimiento del péndulo son dos ecuaciones diferenciales de segundo orden con

cuatro variables ( $\varphi$ ,  $\theta$ ,  $\mu$  y  $\nu$ ):

$$0 = \ddot{\theta} + 4 \cdot \nu \cdot \cos(\theta) \cdot \left( \ddot{\theta} \cos(\theta) - \dot{\theta}^2 \cdot \sin(\theta) \right) + 2 \cdot \mu \cdot \nu \cdot \cos(\theta) \cdot \left( \ddot{\varphi} \cdot \cos(\varphi) - \dot{\varphi}^2 \cdot \sin(\varphi) \right) + \sin(\theta) \quad (3.1)$$

$$0 = \mu \cdot \ddot{\varphi} + 2 \cdot \cos(\varphi) \cdot \left( \ddot{\theta} \cdot \cos(\theta) - \dot{\theta}^2 \cdot \sin(\theta) \right) + \sin(\varphi) \quad (3.2)$$

Para poder utilizar el programa de continuación numérica anteriormente señalado, debemos preparar el sistema de ecuaciones que a éste se le introducirá. Debemos tener un sistema de ecuaciones en el que las ecuaciones sean de primer orden.

Para ello las dos ecuaciones (3.1) y (3.2), con derivadas de segundo orden, se pueden transformar en cuatro ecuaciones en derivadas primeras introduciendo dos variables:

$$\begin{aligned} \psi &= \dot{\varphi} \\ \Theta &= \dot{\theta} \end{aligned}$$

Realizando este cambio de variable se obtienen las siguientes ecuaciones:

$$0 = \dot{\Theta} + 4 \cdot \nu \cdot \cos(\theta) \cdot \left( \dot{\Theta} \cdot \cos(\theta) - \Theta^2 \cdot \sin(\theta) \right) + 2 \cdot \mu \cdot \nu \cdot \cos(\theta) \cdot \left( \dot{\psi} \cdot \cos(\varphi) - \psi^2 \cdot \sin(\varphi) \right) + \sin(\theta) \quad (3.3)$$

$$0 = \mu \cdot \dot{\psi} + 2 \cdot \cos(\varphi) \cdot \left( \dot{\Theta} \cdot \cos(\theta) - \Theta^2 \cdot \sin(\theta) \right) + \sin(\varphi) \quad (3.4)$$

Debemos despejar  $\dot{\Theta}$  y  $\dot{\psi}$  de las ecuaciones anteriores. Como en ambas ecuaciones aparecen  $\dot{\Theta}$  y  $\dot{\psi}$ , para obtener las ecuaciones deseadas, lo que hacemos es despejar de ambas una de las variables  $\dot{\Theta}$  o  $\dot{\psi}$  e igualar, seguidamente de la ecuación resultante se despeja la otra variable  $\dot{\Theta}$  o  $\dot{\psi}$ , según el caso, obteniendo así una ecuación en la forma deseada. Análogamente se haría para la otra variable. No se expone el desarrollo matemáticamente intermedio, por

pensar que puede resultar pesado para el lector, y directamente exponemos las dos ecuaciones resultantes:

$$\dot{\Theta} = \frac{\left(4 \cdot \mu \cdot \nu \cdot \cos(\theta) \cdot \sin^2(\varphi)\right) \cdot \Theta^2 + \left(2 \cdot \mu^2 \cdot \nu \cos(\theta) \cdot \sin(\theta)\right) \cdot \psi^2}{\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} - \frac{\mu \cdot \left(2 \cdot \nu \cdot \cos(\theta) \cdot \cos(\varphi) \sin(\varphi) - \sin(\theta)\right)}{\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} \quad (3.5)$$

$$\dot{\psi} = \frac{\left(4 \cdot \mu \cdot \nu \cos^2(\theta) \cdot \sin(\theta) \cdot \cos(\varphi)\right) \cdot \psi^2 - \left(2 \cdot \sin(\theta) \cos(\varphi)\right) \cdot \Theta^2}{-\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} - \frac{\left(2 \cdot \cos(\theta) \cdot \sin(\theta) \cdot \cos(\varphi) - \sin(\varphi) - 4 \cdot \nu \cos^2(\theta) \cdot \sin(\varphi)\right)}{-\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} \quad (3.6)$$

Con lo que el sistema de ecuaciones para la continuación numérica queda como sigue:

$$\dot{\theta} = \Theta \quad (3.7)$$

$$\dot{\varphi} = \psi \quad (3.8)$$

$$\dot{\Theta} = \frac{\left(4 \cdot \mu \cdot \nu \cdot \cos(\theta) \cdot \sin^2(\varphi)\right) \cdot \Theta^2 + \left(2 \cdot \mu^2 \cdot \nu \cdot \cos(\theta) \cdot \sin(\theta)\right) \cdot \psi^2}{\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} \cdot \frac{\mu \cdot \left(2 \cdot \nu \cdot \cos(\theta) \cdot \cos(\varphi) \cdot \sin(\varphi) - \sin(\theta)\right)}{\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} \quad (3.9)$$

$$\dot{\psi} = \frac{\left(4 \cdot \mu \cdot \nu \cdot \cos^2(\theta) \cdot \sin(\theta) \cdot \cos(\varphi)\right) \cdot \psi^2 - \left(2 \cdot \sin(\theta) \cdot \cos(\varphi)\right) \cdot \Theta^2}{-\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} \cdot \frac{\left(2 \cdot \cos(\theta) \cdot \sin(\theta) \cdot \cos(\varphi) - \sin(\varphi) - 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin(\varphi)\right)}{-\mu \cdot \left(1 + 4 \cdot \nu \cdot \cos^2(\theta) \cdot \sin^2(\varphi)\right)} \quad (3.10)$$

Ya tenemos el sistema preparado para realizar la continuación numérica, así en el siguiente apartado explicamos el funcionamiento del programa utilizado, realizando la continuación en apartados posteriores.

## 3.2. Funcionamiento de AUTO2000.

AUTO2000 es el software de continuación numérica que manejaremos en el presente proyecto. En este apartado explicaremos básicamente, los ficheros que hay que introducirle a AUTO para su funcionamiento, las órdenes necesarias para que funcione y los ficheros que genera después de la continuación. Posteriormente, en los apartados sucesivos, iremos particularizando todo ello para nuestro sistema.

El funcionamiento de AUTO se basa en dos tipos de archivo:

**Fichero xxx.c** contiene varias subrutinas en C, en las que se introducen las ecuaciones diferenciales del movimiento, así como las condiciones

iniciales tales como un punto de equilibrio, las condiciones de contorno, etc. . .

**Fichero c.xxx** en el que se refleja la dimensión del sistema, los parámetros a continuar, las tolerancias admisibles, el número de iteraciones, etc. . .

Vamos a mostrar un ejemplo de cada uno de estos ficheros y daremos unas nociones básicas de los parámetros que hay que introducirles.

Para empezar, mostraremos un fichero tipo **xxx.c** a modo de ejemplo y veremos como funciona.

```
#include "auto_f2c.h" /*
-----*/ /*
-----*/

int func (integer ndim, const doublereal *u, const integer *icp,
          const doublereal *par, integer ijac,
          doublereal *f, doublereal *dfdu, doublereal *dfdp) {
    doublereal phi,theta,psi,Theta,mu,nu;

/*...*/
    phi = u[0];
    theta = u[1];
    psi = u[2];
    Theta = u[3];

    mu=par[0];
    nu=par[1];

    f[0]= psi;

    f[1]= Theta;

    f[2]= ((4*mu*nu*cos(theta)*cos(theta)*sin(theta)*cos(phi))
           *psi*psi-(2*sin(theta)*cos(phi))*Theta*Theta
           -(2*cos(theta)*sin(theta)*cos(phi)-sin(phi)
           -4*nu*cos(theta)*cos(theta)*sin(phi)))/(-mu
           *(1+4*nu*cos(theta)*cos(theta)*sin(phi)*sin(phi)));

    f[3]= ((4*mu*nu*cos(theta)*sin(theta)*sin(phi)*sin(phi))
```

```

        *Theta*Theta+(2*mu*mu*nu*cos(theta)*sin(theta))
        *psi*psi+(mu*(2*nu*cos(theta)*cos(phi)*sin(phi)
        -sin(theta)))/(mu*(1+4*nu*cos(theta)*cos(theta)
        *sin(phi)*sin(phi)));

    return 0;
}

/* -----*/
/* -----*/
int stpnt (integer ndim, doublereal t,
           doublereal *u, doublereal *par) {

    /* Parámetros iniciales */
    par[0] = (doublereal)0.5;
    par[1] = (doublereal)2.0;

    /* Solución inicial */

    u[0] = (doublereal)0.0;
    u[1] = (doublereal)0.0;
    u[2] = (doublereal)0.0;
    u[3] = (doublereal)0.0;

    return 0;
}

/* Las siguientes subrutinas no son usadas aquí, */ /* pero
deben aparecer en el fichero. */

/* -----*/
/* -----*/
int bcnd (integer ndim, const doublereal *par, const integer *icp,
          integer nbc, const doublereal *u0, const doublereal
          *Teta, integer ijac, doublereal *fb, doublereal *dbc) {
    return 0;
}

```

```

} /*
-----*/ /*
-----*/
int icnd (integer ndim, const doublereal *par, const integer *icp,
          integer nint, const doublereal *u, const doublereal
          *uold, const doublereal *udot, const doublereal *upold,
          integer ijac, doublereal *fi, doublereal *dint) {
    return 0;
} /*
-----*/ /*
-----*/
int fopt (integer ndim, const doublereal *u, const integer *icp,
          const doublereal *par, integer ijac,
          doublereal *fs, doublereal *dfdu, doublereal *dfdp) {
    return 0;
} /*
-----*/ /*
-----*/
int pvls (integer ndim, const doublereal *u,
          doublereal *par) {
    return 0;
} /*
-----*/ /*
-----*/

```

En este fichero, podemos distinguir dos funciones:

**func** es la encargada de pasar a AUTO las ecuaciones diferenciales y de definir las variables independientes y los parámetros.

**stpnt** es la función que AUTO emplea cuando se le indica que comience la continuación a partir de una solución inicial.

Como hemos dicho antes, además de las funciones **func** y **stpnt**, existen cuatro más (**bcnd**, **icnd**, **fopt** y **pvls**) que se utilizan cuando se definen condiciones de contorno, condiciones de integración, etc... y que, aunque no contengan nada, deben incluirse en este fichero.

Este fichero, sólo es necesario modificarlo si se quiere partir de otra solución inicial, para lo cuál tendríamos que variar únicamente la función **stpnt**. En cualquier otro caso, al permanecer las ecuaciones diferenciales inalterables, no sería necesaria su modificación.

Una vez mostrado un fichero tipo **xxx.c**, explicaremos el otro fichero necesario para trabajar con AUTO, el **c.xxx**.

Este fichero será el que dé a AUTO las condiciones en las que debe realizar la continuación de las ecuaciones definidas en **xxx.c**. Veamos un ejemplo:

```

4 1 0 1          NDIM,IPS,IRS,ILP
2 0 1          NICP,(ICP(I),I=1,NICP)
100 4 3 1 1 0 0 0  NTST,NCOL,IAD,ISP,ISW,IPLT,NBC,NINT
180 0 100 0 100    NMX,RL0,RL1,A0,A1
5 5 2 8 7 5 0     NPR,MXBF,IID,ITMX,ITNW,NWTN,JAC
1e-7 1e-7 1e-5    EPSL,EPSU,EPSS
0.001 0.001 0.01 1 DS,DSMIN,DSMAX,IADS
0                NTHL,(/I,THL(I)),I=1,NTHL)
0                NTHU,(/I,THU(I)),I=1,NTHU)
0                NUZR,(/I,PAR(I)),I=1,NUZR)

```

Vamos a explicar los parámetros más importantes de este fichero:

- **NDIM** indica la dimensión del sistema de ecuaciones. En el caso que nos aborda tenemos cuatro ecuaciones en derivadas primeras.
- **IPS** define el tipo de problema. Nosotros nos moveremos entre dos de sus valores:
  - IPS=1 Soluciones estacionarias de ecuaciones diferenciales ordinarias con detección de bifurcaciones de Hopf.
  - IPS=2 Para continuación de soluciones periódicas.
- **IRS** define la etiqueta de la solución donde la continuación comenzará. Si su valor es cero, como en este caso, el programa tomará como solución inicial la indicada en la función `stpnt`.
- **ILP** tiene dos valores posibles:
  - ILP=0 No detecta los posibles pliegues de la continuación.
  - ILP=1 Detecta los pliegues de la continuación (puntos LP). Es el valor recomendado.
- **NICP** indica el número de parámetros que tiene nuestro sistema. En nuestro caso, inicialmente, son  $\mu$  y  $\nu$ , y por ello le hemos asignado a esta variable el valor 2. En realidad, el número de parámetros está definido en el archivo `xxx.c` con un vector llamado `PAR[ ]`, y en el `NICP` lo único que indicaremos es cuántos de ellos queremos que aparezcan por pantalla al ejecutar AUTO.

- **ICP** depende del valor que hallamos asignado a NICP. Debemos indicar qué parámetros serán los que aparecerán en la continuación y según el orden, cuál será el parámetro de continuación principal y cuál el secundario. En nuestro caso, hemos indicado un valor de NICP=2, luego hemos elegido los dos parámetros de continuación; PAR[0] en primer lugar ( $\mu$ ) como parámetro de continuación principal y PAR[1] ( $\nu$ ) como segundo parámetro de continuación.
- **ISP** es el parámetro que controla la detección de puntos de bifurcación (BP), puntos de bifurcación de doble período (PD), etc. Nosotros emplearemos dos valores de esta variable:
  - ISP=1 Este valor detecta puntos de bifurcación (HB) para soluciones que no sean periódicas, y no detecta puntos de bifurcación de doble período (PD).
  - ISP=2 Este valor detecta todos los puntos especiales y es el que emplearemos cuando trabajemos con órbitas periódicas.
- **ISW** indica el tipo de continuación que realizaremos. Nosotros emplearemos tres valores:
  - ISW=1 Se utiliza para continuar equilibrios con un solo parámetro y para trazar una órbita periódica a partir de un punto de Hopf.
  - ISW=2 Se utiliza para continuar puntos límite (PL) y puntos de Hopf (HB), ya que al añadirse implícitamente una ecuación en estos puntos, se nos permite variar un segundo parámetro.
  - ISW=-1 Se utiliza en órbitas periódicas para continuar puntos de bifurcación (BP) y puntos de bifurcación de doble período (PD).
- **NMX** indica el número máximo de iteraciones que le vamos a permitir a la continuación.
- **RL0, RL1** indican el valor mínimo y máximo, respectivamente, que puede tomar el parámetro de continuación principal, en este caso  $\mu$ .
- **NPR** Si su valor es inferior a NMX, entonces se mostrarán por pantalla los resultados cada NPR iteraciones. Si su valor es igual a NMX, sólo se mostrarán por pantalla los puntos especiales encontrados (dichos puntos también aparecerán en el caso anterior).
- **DS** indica el tamaño de paso normal entre dos puntos de la misma rama. Además, un cambio de signo, provoca que la continuación se realice en sentido contrario.

- **DSMIN**, **DSMAX** indican en valor absoluto los tamaños de paso mínimo y máximo admisibles entre dos puntos de la misma rama.

Una vez introducidos los ficheros, nos situamos en el directorio donde se encuentran y desde consola ejecutamos el programa escribiendo **auto** (escribir en minúsculas). El resultado será algo parecido a la Figura 3.1.

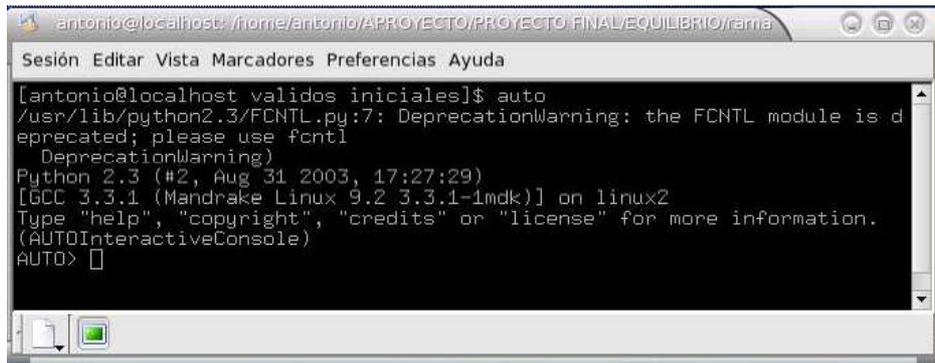


Figura 3.1: Ejecución de AUTO.

Dentro del programa, tenemos que cargar los ficheros **xxx.c** y **c.xxx**, lo cual se puede hacer de dos formas:

- Cargando primero el fichero de ecuaciones y luego el fichero de constantes:

```
AUTO>ld('xxx') AUTO>r(c='xxx')
```

- Cargando los dos ficheros a la vez:

```
AUTO>r(e='xxx',c='xxx')
```

Una vez ejecutadas estas órdenes, el programa nos devuelve tres ficheros fort con las extensiones 7, 8 y 9:

**fort.7** contiene los valores de los parámetros de continuación para cada una de las iteraciones que nos permitirán dibujar el diagrama de bifurcación.

**fort.8** contiene información más extensa de cada una de las órbitas etiquetadas.

**fort.9** contiene mensajes de diagnóstico, convergencia de las soluciones, autovalores, etc...

Será por medio de estos tres ficheros que podamos representar gráficamente la continuación y podamos obtener los autovalores.

### 3.3. Continuación numérica de los equilibrios.

En el capítulo anterior calculamos las distintas situaciones de equilibrio para nuestro sistema. En este apartado, empezaremos recordando dichas situaciones de equilibrio, posteriormente se realizará la continuación numérica, estudiando los autovalores, que nos devuelva el programa AUTO2000, para cada uno de ellos.

#### 3.3.1. Situaciones de equilibrio.

Para facilitar la comprensión de las distintas situaciones de equilibrios, empezamos recordando el modelo de nuestro sistema:

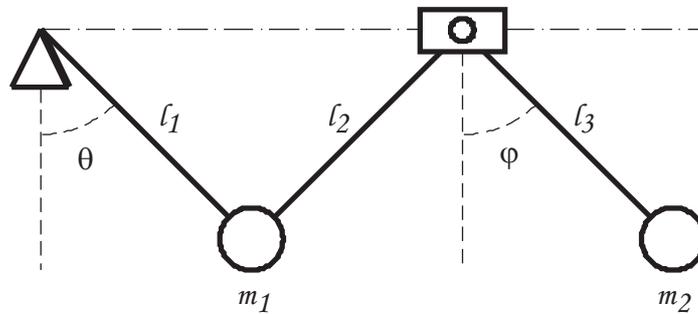


Figura 3.2: Modelo del sistema.

Las situaciones de equilibrio en el sistema anterior, vienen dadas por:

$$\begin{aligned} \sin(\theta) = 0 &\implies \theta = 0, \pi \\ \sin(\varphi) = 0 &\implies \varphi = 0, \pi \end{aligned} \tag{3.11}$$

Por lo que las cuatro posiciones de equilibrio, combinando dos a dos los valores de las variables  $\theta$  y  $\varphi$ , son las siguientes:

1.  $\theta = 0$  y  $\varphi = 0$
2.  $\theta = 0$  y  $\varphi = \pi$
3.  $\theta = \pi$  y  $\varphi = 0$
4.  $\theta = \pi$  y  $\varphi = \pi$

Como puede verse las cuatro posiciones de equilibrio de nuestro sistema están en el eje vertical, por lo que podemos intuir que los resultados obtenidos para uno de ellos serán representativos para el resto.

### 3.3.2. Continuación de los equilibrios.

En este apartado, comenzaremos a utilizar AUTO para continuar los equilibrios hallados anteriormente.

En primer lugar, creamos cuatro ficheros llamados **equi1.c**, **equi2.c**, **equi3.c** y **equi4.c**, los cuáles tienen la misma función *func*, ya que todos tienen las mismas ecuaciones, y distinta función *stpnt*, ya que cada uno de ellos corresponde a uno de los equilibrios y tienen distintas condiciones iniciales, que son las distintas situaciones de equilibrio señaladas en el apartado anterior.

Del mismo modo, creamos ahora cuatro ficheros llamados **c.equi1**, **c.equi2**, **c.equi3** y **c.equi4**, los cuáles contienen los parámetros de continuación de AUTO. Estos cuatro ficheros son iguales, pero como ya se dijo en la introducción de este capítulo deben existir por pareja los ficheros. Mostraremos uno de ellos como ejemplo:

```

4 1 0 1          NDIM, IPS, IRS, ILP
2 0 1          NICP, (ICP(I), I=1 NICP)
100 4 3 1 1 0 0 0 NTST, NCOL, IAD, ISP, ISW, IPLT, NBC, NINT
180 0 100 0 100  NMX, RLO, RLI, A0, A1
180 5 2 8 7 5 0  NPR, M $\times$ BF, IID, ITMX, ITNW, NWTN, JAC
1e-7 1e-7 1e-5  EP $\times$ SL, EP $\times$ SU, EP $\times$ SS
0.001 0.001 0.01 1 DS, DS $\times$ MIN, DS $\times$ MAX, IADS
0              NTHL, (✓, I, THL(I)), I=1, NTHL)
0              NTHU, (✓, I, THU(I)), I=1, NTHU)
0              NUZR, (✓, I, PAR(I)), I=1, NUZR)

```

Los parámetros más destacables de este fichero son:

**IRS=0** Partimos del equilibrio que nos da la función *stpnt*.

**ISW=1** Continuamos un sólo parámetro (en este caso  $\mu$ ).

**NMX=180** Haremos un máximo de 180 iteraciones.

**NPR=180** Se mostrarán sólo las etiquetas especiales.

Una vez creados los ocho archivos mencionados, dos por cada tipo de equilibrio, ejecutaremos AUTO y veremos los resultados obtenidos por pantalla. Para ello, abriremos la consola, e iremos al directorio donde hemos creado los ocho ficheros. Una vez allí teclearemos lo siguiente:

```
auto AUTO>ld('equi1') AUTO>r(c='equi1')
```

La salida por consola se muestra en la figura anterior 3.3. En ella podemos observar como AUTO nos da el valor de los parámetros para las iteraciones 1 y 180 en este caso, ya que no ha existido ningún punto especial

```

antonio@localhost: /home/antonio/APROYECTO/PROYECTO_FINAL/EQUILIBRIO - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
[antonio@localhost EQUILIBRIO]# auto
/usr/lib/python2.3/FCNTL.py:74: DeprecationWarning: the FCNTL module is deprecated; please use fcntl
  DeprecationWarning()
Python 2.3 (#2, Aug 31 2003, 17:27:23)
[GCC 3.3.1 (Mandrake Linux 9.2 3.3.1-1mdk)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
(AUTO)InteractiveConsole
(AUTO) Id('equi1')
Runner configured
(AUTO) r(C='equi1')
gcc -O -DPTHREADS -O -I/usr/local/auto/2000/include -o equi1.o
gcc -O equi1.o -o equi1.exe /usr/local/auto/2000/lib/*.* -lpthread -L/usr/local/auto/2000/lib -lauto_f2c -lm
Starting equi1 ...
  BR  PT  TY  LAB  PAR(0)  L2-NORM  U(1)  U(2)  U(3)  U(4)  PAR(1)
  1    1  EP   1  5.000000E-01  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  2.000000E+00
  1   10  EP   2  2.266000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  2.000000E+00
Total Time 2.401E-01
equi1 done
(AUTO) #

```

Figura 3.3: Ejemplo de fichero.

entre ellos. Si en el archivo **c.equi1** le hubiéramos dado a la constante NPR un valor inferior a 180, podríamos ver más etiquetas en puntos intermedios y además la evolución de los distintos parámetros y variables independientes. Sin embargo, no lo hacemos de esta manera porque AUTO es capaz de representarnos gráficamente la solución de la continuación, donde se ve el resultado más claramente. Además podemos ver como AUTO nos etiqueta los puntos representados con unos determinados números que, como veremos posteriormente, nos servirán para hacer referencia a dichos puntos.

Para guardar los cambios, utilizamos el siguiente comando:

```
AUTO>sv('equi1')
```

Al utilizar el comando *save* o *sv*, se generan los siguientes ficheros:

**b.equi1** que contiene los valores de los parámetros de continuación para cada una de las iteraciones, y nos permitirá dibujar el diagrama de bifurcación.

**s.equi1** que contiene información más extensa de cada una de las iteraciones etiquetadas.

**d.equi1** que contiene mensajes de diagnóstico, convergencia de soluciones, autovalores, etc.

Para representar gráficamente la solución recurrimos al comando *plot* o *pl*. Si lo llamamos sin pasarle ningún argumento, representará lo que encuentre en los ficheros fort7, fort8 y fort9. Sin embargo, si queremos que nos represente la gráfica de unos determinados ficheros b.xxx, s.xxx y d.xxx, entonces le tendremos que pasar como parámetro 'xxx'. Mostraremos como ejemplo lo que tendríamos que escribir por consola para representar la gráfica de 'equi1':

```
AUTO>p1('equi1')
```

Debido a que el entorno gráfico de AUTO no es demasiado preciso, hemos optado por representar las gráficas de este documento con MATLAB importando los datos que genera AUTO en el fichero 'b.xxx'.

Ahora estamos en disposición de estudiar cada una de las situaciones de equilibrio. Al ejecutar los archivos anteriormente citados, para cada uno de los equilibrios, no se produce la continuación de equilibrio buscada con éstos, permaneciendo el sistema en la posición de equilibrio de partida aunque se varíe el valor del parámetro de continuación, para cualquiera de los dos parámetros posibles que podemos usar como parámetros de continuación. En la siguiente gráfica 3.4 se representa la amplitud frente a la variación del parámetro de continuación  $\mu$ :

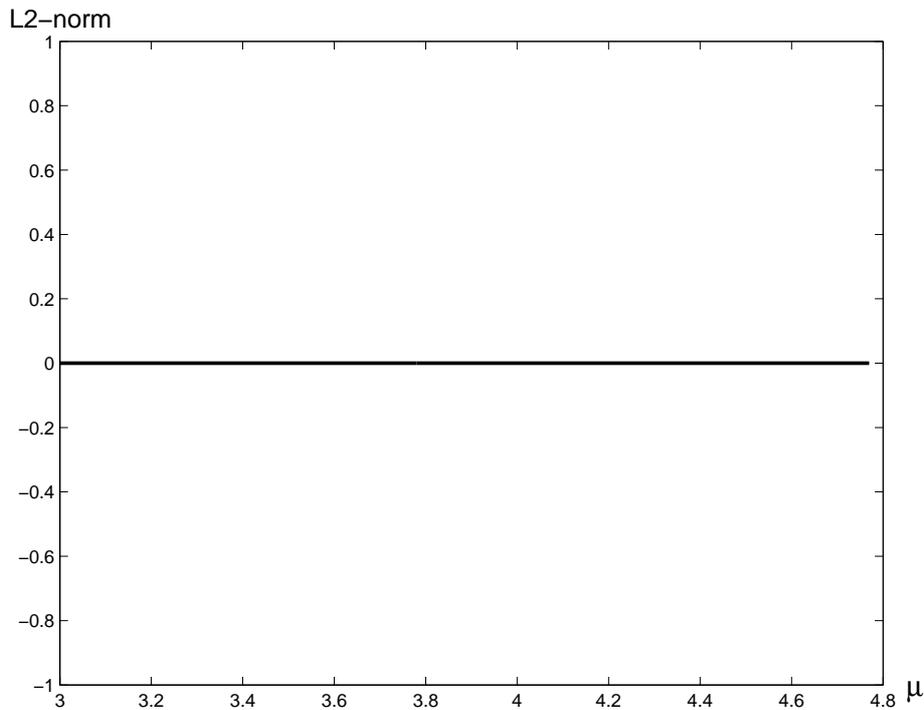


Figura 3.4: Amplitud frente a  $\mu$ .

En ella podemos ver como la amplitud se mantiene en cero, para cualquier valor del parámetro, como tiene que ser.

### 3.3.3. Estabilidad de los equilibrios.

En este apartado exponemos la estabilidad de nuestro sistema, en cada una de las situaciones de equilibrio que hemos obtenido.

La estabilidad de una posición de equilibrio depende de sus autovalores. Si los cuatro autovalores para una posición de equilibrio son imaginarios puros, entonces será una posición de equilibrio estable.

Para que AUTO2000 nos muestre los autovalores, por ejemplo para nuestro fichero 'equi1', lo hacemos mediante la orden *ev*, *eg* o *eigenvalue* de la siguiente forma:

```
AUTO>ev('equi1')
```

Los espectro de los cuatro autovalores en cada situación son:

- Primera posición de equilibrio,  $\theta = 0$  y  $\varphi = 0$ :

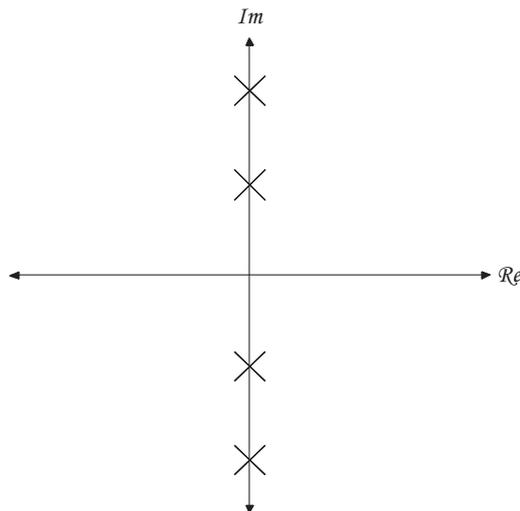


Figura 3.5: Espectro de autovalores(primera situación)

Por lo anterior, esta posición de equilibrio siempre será estable.

- Segunda posición de equilibrio,  $\theta = 0$  y  $\varphi = \pi$ :

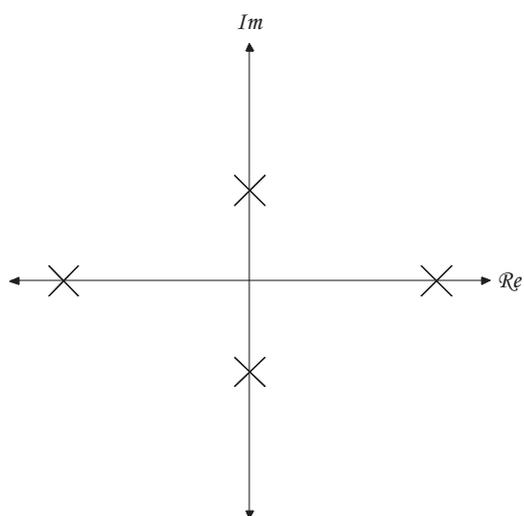


Figura 3.6: Espectro de autovalores(segunda situación)

Por lo anterior, esta posición de equilibrio siempre será inestable.

- Tercera posición de equilibrio,  $\theta = \pi$  y  $\varphi = 0$ :

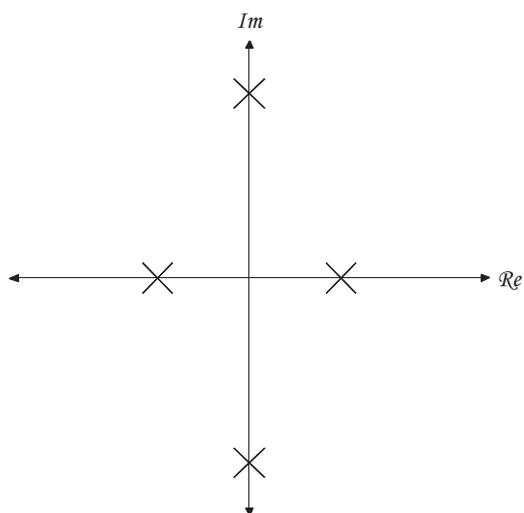


Figura 3.7: Espectro de autovalores(tercera situación)

Por lo anterior, esta posición de equilibrio siempre será inestable.

- Cuarta posición de equilibrio,  $\theta = \pi$  y  $\varphi = \pi$ :

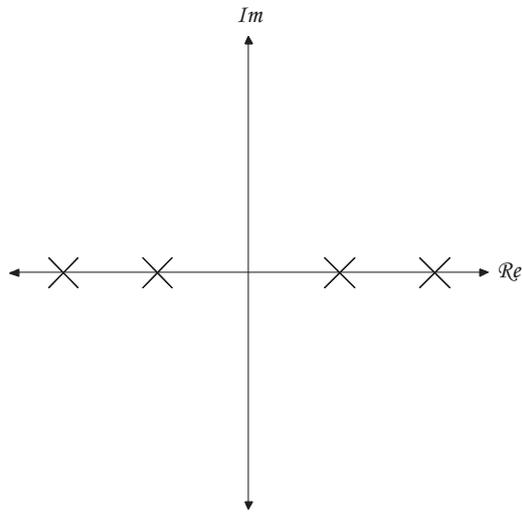


Figura 3.8: Espectro de autovalores(cuarta situación)

Por lo anterior, esta posición de equilibrio siempre será inestable.

En los gráficos anteriores se han representado el espectro de los cuatro autovalores, en un punto cualquiera como representativo de todos los puntos de la evolución, para cada una de las situaciones de equilibrio ya que no habrá cambios en la posición, en toda la evolución, que nos afecte a la estabilidad. Así que sólo la primera situación,  $\theta = 0$  y  $\varphi = 0$ , es estable y será esta posición la que tomemos como inicial en los siguientes apartados.