
LA COLUMNA DE MATEMÁTICA COMPUTACIONAL

Sección a cargo de

Tomás Recio

Software libre para el estudio de sistemas dinámicos

por

Antonio Morante y José Antonio Vallejo

1. INTRODUCCIÓN

Las herramientas informáticas con que se cuenta en la actualidad sin duda suponen una excelente ayuda para los docentes, tanto en centros de enseñanza media como en las universidades. Por ejemplo, permiten realizar esos cálculos directos pero tediosos que nunca se ven en las clases ni en los libros, o ilustrar gráficamente la variación del comportamiento de algún objeto geométrico cuando se recorren los valores de cierto parámetro (algo que siempre acaba en un dibujo ininteligible cuando uno trata de hacerlo en la pizarra).

Existen numerosos paquetes de cálculo simbólico (o CAS, por Computer Algebra System) de tipo profesional, con un desempeño francamente excelente, como Mathematica, Maple o Matlab, pero en este trabajo nos hemos decantado por dos herramientas de software libre: GeoGebra, orientado a la geometría dinámica (en la línea de Cabri o The Geometer Sketchpad) y Maxima, orientado a la manipulación simbólica (como Mathematica, con el cual comparte el ancestro común MACSYMA). Hay varios motivos para esto; dejando aparte los puramente personales como la preferencia por usar software libre, que pueden ser compartidos o no, la razón fundamental es ésta: mientras que en un campus universitario es común que el estudiante tenga acceso a una licencia gratuita de alguno de estos grandes paquetes (normalmente costeada por la universidad a través de algún convenio), cuando acaba su vinculación con la universidad como alumno y comienza a trabajar, digamos impartiendo clases en algún instituto, es difícil que vaya a disponer de los recursos necesarios para pagar una licencia de este tipo. La situación es aún más grave en países en vías de desarrollo, como es el caso de muchos en Iberoamérica, donde ni siquiera es de esperar que se disponga de estas licencias de campus, por su alto coste.

A fin de cuentas, la situación es que se invierte mucho tiempo y esfuerzo en la elaboración de materiales didácticos, prácticas, etc., por parte del personal en las

universidades, tratando de «modernizar» la enseñanza de las matemáticas, pero ese esfuerzo no permea en el sistema educativo básico a través de los egresados de las universidades, porque un instituto de enseñanza media no dispone de los mismos recursos que una universidad. Nosotros estamos convencidos de que una alternativa a esta situación consiste en el uso de programas como GeoGebra y Maxima (sin perjuicio de los grandes paquetes como Mathematica, nada impide usarlos en paralelo).

Casi todas las exposiciones sobre el uso de estos programas que se encuentran en Internet están orientadas a las aulas de secundaria, por lo que pudiera dar la impresión de que tienen limitaciones que impiden su uso en clases de universidad. Sin embargo no es así: en estas notas nuestra intención es «dar una probadita» de GeoGebra y Maxima, sin hacer en modo alguno una exposición completa, sino sólo ver cómo es posible utilizarlos para exponer temas usuales en un curso universitario de sistemas dinámicos, bien sean discretos o continuos. El tratar con sistemas dinámicos tampoco debe interpretarse como una restricción (Maxima puede usarse para hacer desde cálculos con formas exteriores a factorizaciones de polinomios sobre anillos arbitrarios, por poner un ejemplo), se debe a que este tema se presta especialmente al uso de software.

En un artículo previo de LA GACETA DE LA RSME [6] ya se ha tratado la instalación y propiedades básicas de GeoGebra. Además, en Internet se encuentra fácilmente información sobre la instalación y configuración tanto de GeoGebra como de Maxima, en ambientes Windows, Linux y Mac, por lo que no nos detendremos en estos temas. Tampoco (por razones obvias de espacio) explicaremos con detalle la sintaxis de comandos; el lector interesado puede consultar la ayuda de los programas o cualquiera de los tutoriales disponibles en Internet. No obstante, hemos seleccionado los ejemplos de tal forma que en ellos se use el mínimo número de comandos y que éstos sean lo más intuitivos posible (lo cual tiene como efecto que, posiblemente, las construcciones que presentamos no sean ni las más elegantes ni las más directas, algo que el lector debe tener en cuenta).

2. NOCIONES BÁSICAS SOBRE GEOGEBRA

GeoGebra es un sistema de geometría dinámica orientado a la enseñanza de la matemática a nivel preuniversitario [4, 5]. Sus desarrolladores son Markus Hohenwarter, Michael Borchers e Yves Kreis. GeoGebra es un software libre, multilingüe y multiplataforma. El programa y las instrucciones para su instalación se encuentran en <http://www.geogebra.org/cms/es> (en el trabajo presente hemos usado la versión 3.0.0). En GeoGebra, a cada objeto algebraico le corresponde un objeto geométrico y viceversa, con la excepción de algunos objetos algebraicos auxiliares. Los primeros se despliegan en la ventana algebraica y los segundos en la ventana geométrica. Puntos, vectores, rectas, secciones cónicas, etc., se pueden construir pulsando en los respectivos iconos que se encuentran en los cajones de herramientas, o bien tecleando la orden apropiada en la ventana de texto (véase la figura 1). Vale la pena hacer notar que la versión actual de GeoGebra incorpora una hoja de cálculo.

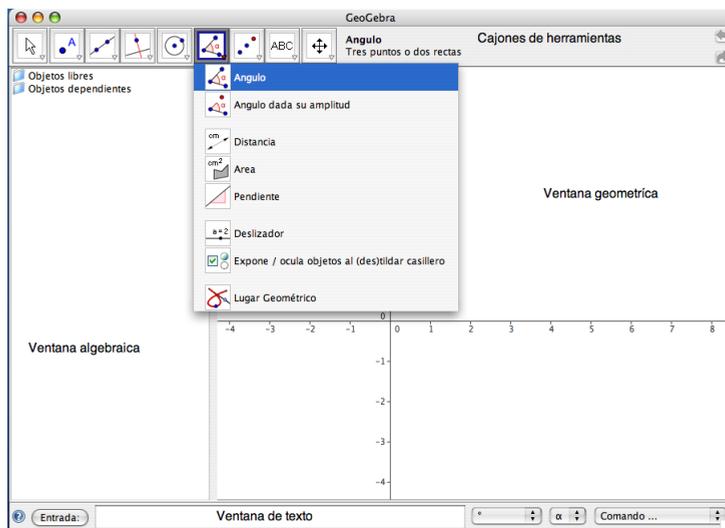


Figura 1: Componentes principales de GeoGebra.

El comportamiento global del programa (unidad angular, posiciones decimales, tamaño de letra, idioma, etc.) se puede modificar seleccionando en la barra superior el menú **Options** (el idioma por defecto es el inglés). En particular, las órdenes que usaremos en este artículo presuponen haber seleccionado en la barra de menús como idioma el español: **Options/Language/Spanish**.

Cada objeto construido tiene atributos (nombre, color, estilo, etc.) que pueden modificarse mediante un menú contextual, que se despliega al pulsar sobre el objeto en cuestión con el botón derecho del ratón (véase la figura 2).

Las construcciones hechas se pueden modificar de forma dinámica seleccionando la herramienta **Desplaza** (primera herramienta en el extremo izquierdo) y arrastrando luego con el ratón los objetos que queremos modificar. Las construcciones hechas pueden exportarse a una plantilla dinámica en formato html, la cual puede verse y modificarse de forma interactiva con la mayor parte de navegadores web. Para tal fin se selecciona en la barra de menús la opción **Archivo/Exporta/Plantilla Dinámica como Página Web (html)**.

3. SISTEMAS DISCRETOS: APLICACIÓN CUADRÁTICA F_k

La capacidad de modificar dinámicamente las construcciones hechas en GeoGebra hace de este software una herramienta ideal para motivar conceptos y resultados de sistemas dinámicos, tales como órbitas especiales, atractores y repulsores, bifurcaciones, etc.

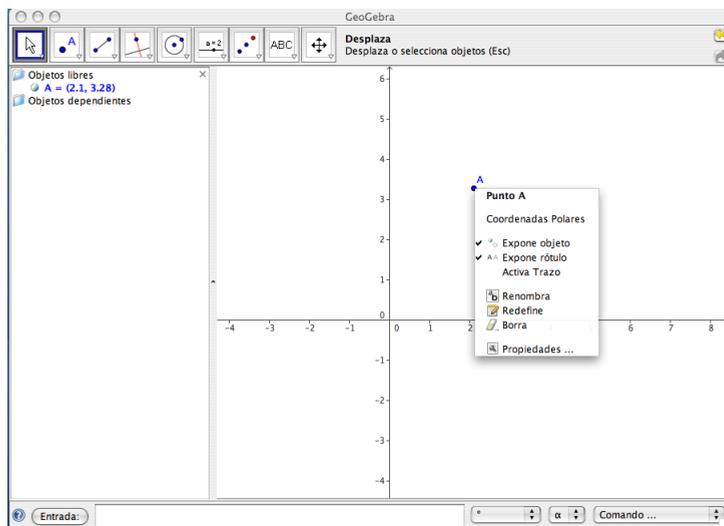


Figura 2: Ejemplo de un menú contextual.

3.1. DEFINICIÓN DE LA DINÁMICA

Consideremos la aplicación cuadrática $F_k : \mathbb{R} \rightarrow \mathbb{R}$ dada por

$$F_k(x) = kx(1 - x),$$

donde $k \geq 1$ es un parámetro (se conoce como aplicación logística de parámetro k). Para modelar en GeoGebra la familia de funciones que se obtiene al variar k , se elige la herramienta **Deslizador**, se hace click en la zona gráfica y se ajustan a nuestras necesidades las propiedades del deslizador (véanse las figuras 1 y 3). Una vez hecho esto, se teclea directamente en la ventana de texto la orden $F_k(x) = k*x*(1-x)$. Al seleccionar la herramienta **Desplaza** y arrastrar con el ratón el botón del deslizador, se dibujan funciones de la familia F_k que cambian dinámicamente al mover el botón.

Para valores apropiados del parámetro k , las funciones F_k se utilizan en el modelado del crecimiento de poblaciones que varían en intervalos discretos de tiempo, por ejemplo anualmente, mediante un proceso iterativo de la llamada ecuación logística discreta

$$x_{n+1} = kx_n(1 - x_n), \quad n \in \mathbb{N}.$$

Aquí x_n representa la fracción de la máxima población soportada por el medio ambiente en tiempo n , mientras que k es la razón de crecimiento para poblaciones pequeñas [2].

Identificamos la condición inicial x_0 con el punto $P_0 = (x_0, x_0)$ en la diagonal $y = x$. Para hacer esta construcción en GeoGebra, damos la orden $y = x$, abrimos un menú contextual para renombrar la recta obtenida (digamos como **Id**), damos la orden **Punto[Id]** y renombramos el punto obtenido (digamos como **P_0**).

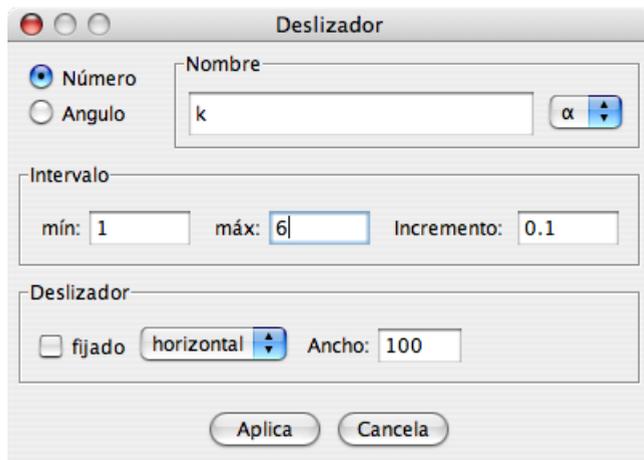


Figura 3: Cuadro de diálogo para definir el deslizador k .

Podemos determinar gráficamente los siguientes n estados de la órbita futura de x_0 bajo F_k (esto es, el conjunto de puntos $x_{i+1} = F_k(x_i)$, $1 \leq i \leq n$) si identificamos a cada número x_i con el punto $P_i = (x_i, x_i)$ en la diagonal $y = x$ y hacemos un diagrama de escalera. En nuestro caso, hemos dibujado sólo los dos primeros puntos de la órbita futura de x_0 , los cuales cambian dinámicamente al mover el punto P_0 sobre la diagonal (véase la figura 4).

Observamos que si $x_0 \notin [0, 1]$ entonces $x_n \rightarrow -\infty$ cuando $n \rightarrow \infty$. Restringimos por tanto nuestra atención a la dinámica de F_k en el intervalo $I = [0, 1]$. Como $F_k(1/2) = k/4$ es el máximo de la función F_k , se tiene que si $1 \leq k \leq 4$ entonces $F_k(I) \subset I$.

La construcción completa en GeoGebra para este experimento se muestra en la figura 5 (para mejorar la apariencia del dibujo se ocultaron los rótulos de los segmentos y vectores trazados, modificando el tipo de línea para dibujarlos, según se explica al final de la sección 2).

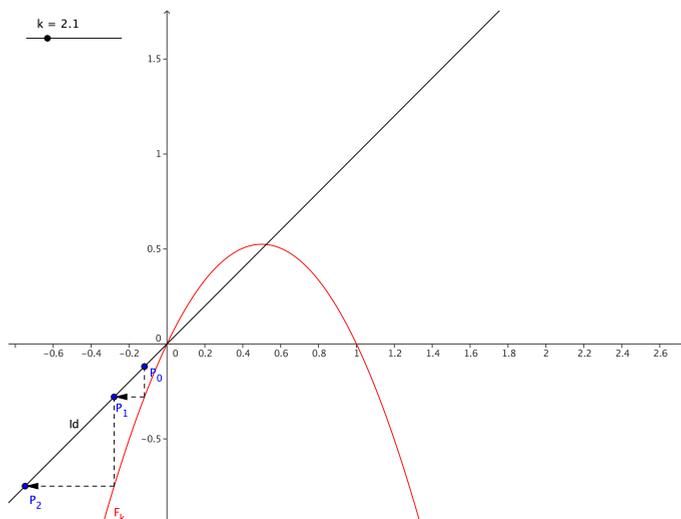
3.2. PUNTOS PERIÓDICOS DE F_k

Decimos que x es un punto n -periódico para la función f si existe un entero positivo n tal que $f^n(x) = x$. El mínimo entero positivo n tal que $f^n(x) = x$ se llama periodo primario de x . En particular, x es un punto fijo si $f(x) = x$.

Por ejemplo, un cálculo simple muestra que el conjunto de puntos fijos de la aplicación F_k es

$$\text{Fix}(F_k) = \left\{ 0, q_k = 1 - \frac{1}{k} \right\}.$$

Para tal fin realizamos una nueva construcción en GeoGebra donde, después de definir la función F_k y la diagonal $y = x$, incluimos los puntos de intersección de estos objetos con la orden `Intersecta[F_k, Id]` (véase la figura 6).



(arrastre con el ratón del punto P_0 sobre la diagonal)

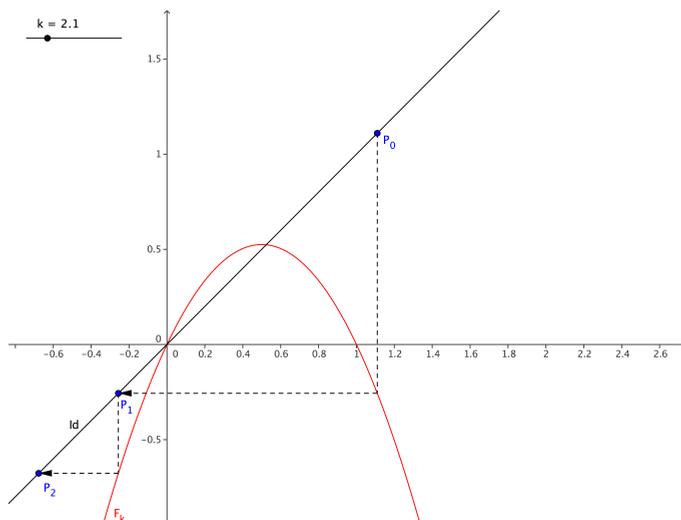


Figura 4: Diagramas de escalera para F_k .

Sea x un punto periódico para una función f , con periodo primario n . El punto x es no hiperbólico si $|(f^n)'(x)| = 1$; en caso contrario el punto es hiperbólico. Cuando $|(f^n)'(x)| < 1$, el punto x es un punto periódico atractor; si $|(f^n)'(x)| > 1$ es un repulsor.

Protocolo de la Construcción		
Archivo Visualiza Ayuda		
No.	Nombre	Comando
1	Número k	
2	Recta l_d	
3	Función F_k	$F_k(x) = k \times (1 - x)$
4	Punto P_0	Punto[l_d]
5	Segmento b	Segmento[$P_0, (x(P_0), F_k(x(P_0)))$]
6	Vector u	Vector[$(x(P_0), F_k(x(P_0))), (F_k(x(P_0)), F_k(x(P_0)))$]
7	Punto P_1	$(F_k(x(P_0)), F_k(x(P_0)))$
8	Segmento c	Segmento[$P_1, (x(P_1), F_k(x(P_1)))$]
9	Vector v	Vector[$(x(P_1), F_k(x(P_1))), (F_k(x(P_1)), F_k(x(P_1)))$]
10	Punto P_2	$(F_k(x(P_1)), F_k(x(P_1)))$

Figura 5: Construcción en GeoGebra del diagrama de escalera para F_k .

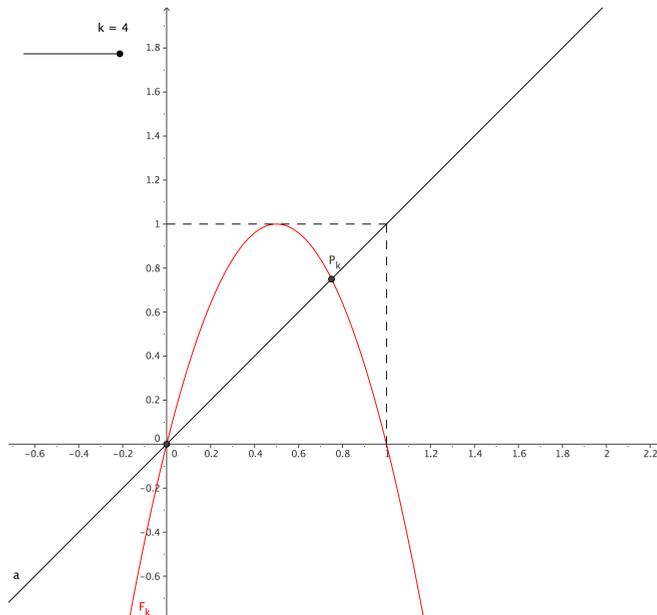


Figura 6: Los puntos fijos de F_k cambian al mover el deslizador k .

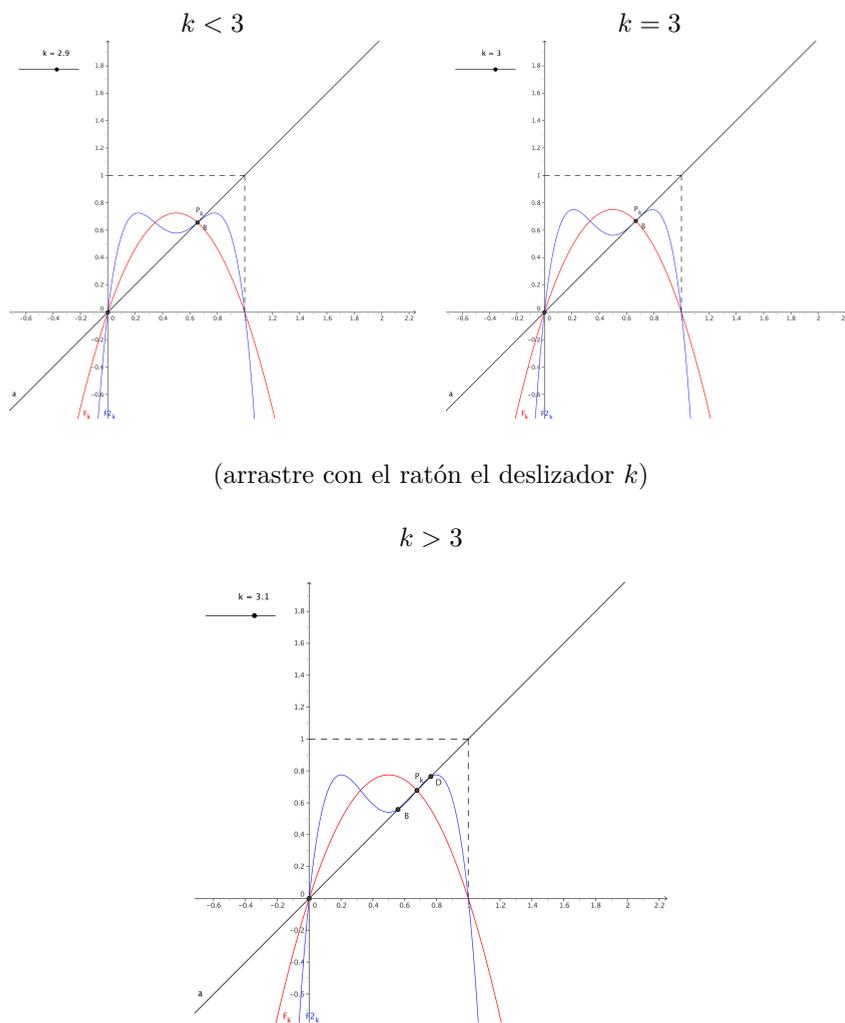


Figura 7: Al mover el deslizador k , el punto fijo q_k da lugar a dos puntos de periodo primario 2.

Notemos que $F'_k(0) = k$ y $F'_k(q_k) = 2 - k$. Luego, para $k > 1$, 0 es un punto fijo repulsor; por otra parte, q_k es un punto fijo atractor si $k \in (1, 3)$ y repulsor si $k \notin (1, 3)$. Para $k = 3$, $|F'_k(q_k)| = 1$, esto es, q_k se convierte en un punto fijo no hiperbólico y esperamos que sufra una bifurcación dando lugar a dos puntos de periodo 2. Para ver esto en GeoGebra, tecleamos la orden $F2_k(x) = F_k(F_k(x))$ seguida de la orden $Intersecta[F2_k, Id]$. Al mover el deslizador k de valores menores que 3 a valores mayores que 3 aparecen de forma dinámica dos nuevos puntos de periodo 2 para F_k , como se muestra en la figura 7. Conforme k crece se

obtienen puntos periódicos de periodo 4, 8, 16, etc., teniéndose para $k = 4$ puntos periódicos de periodo 2^n para todo entero no negativo n .

La construcción completa en GeoGebra para este experimento se muestra en la figura 8.

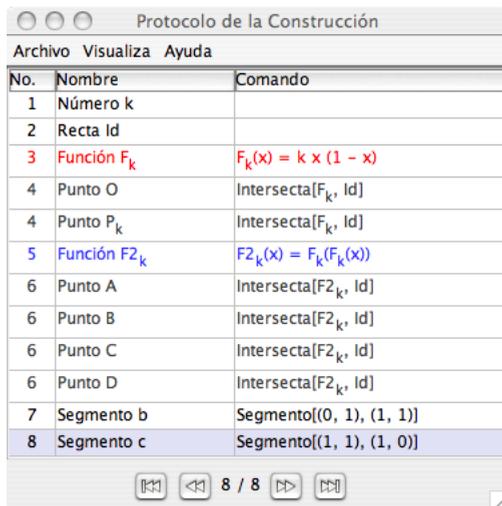


Figura 8: Construcción en GeoGebra de la bifurcación de q_k .

3.3. EL CONJUNTO INVARIANTE Λ_k

Para nuestro último experimento con GeoGebra es conveniente realizar la construcción completa que se muestra en la figura 9.

Si $k > 4$, hay puntos del intervalo $I = [0, 1]$ que lo abandonan después de una iteración. Llamamos A_0 a este conjunto de puntos, esto es,

$$A_0 = \{x \in I : F_k(x) > 1\}.$$

Observemos que A_0 consiste en un intervalo abierto centrado en $1/2$ y que $x \in A_0$ implica que $F_k^n(x) \rightarrow -\infty$ cuando $n \rightarrow \infty$. Llamamos A_1 al conjunto de preimágenes de A_0 bajo F_k , esto es,

$$A_1 = \{x \in I : F_k(x) \in A_0\}.$$

Tenemos que A_1 consiste en dos intervalos abiertos disjuntos $A_{1,0}$ y $A_{1,1}$, como podemos ver en la figura 10.

Nuevamente, $x \in A_1$ implica que $F_k^n(x) \rightarrow -\infty$ cuando $n \rightarrow \infty$. Continuando de forma inductiva obtenemos la sucesión de conjuntos $\{A_m\}_{m \in \mathbb{N}}$ donde

$$A_m = \{x \in I : F_k(x) \in A_{m-1}\}$$

Protocolo de la Construcción

Archivo Visualiza Ayuda

No.	Nombre	Comando
1	Recta Id	
2	Número k	
3	Función F_k	$F_k(x) = k \times (1 - x)$
4	Función $F2_k$	$F2_k(x) = F_k(F_k(x))$
5	Segmento b	Segmento[(0, 1), (1, 1)]
6	Segmento c	Segmento[(1, 1), (1, 0)]
7	Punto A	Intersecta[F_k , b]
7	Punto B	Intersecta[F_k , c]
8	Segmento A_0	Segmento[(x(A), 0), (x(B), 0)]
9	Punto C	Intersecta[$F2_k$, b]
9	Punto D	Intersecta[$F2_k$, c]
9	Punto E	Intersecta[$F2_k$, b]
9	Punto F	Intersecta[$F2_k$, c]
10	Segmento A_{10}	Segmento[(x(C), 0), (x(D), 0)]
11	Segmento A_{11}	Segmento[(x(E), 0), (x(F), 0)]
12	Recta a	Tangente[A, F_k]

12 / 12

Figura 9: Construcción en GeoGebra del conjunto Λ_k .

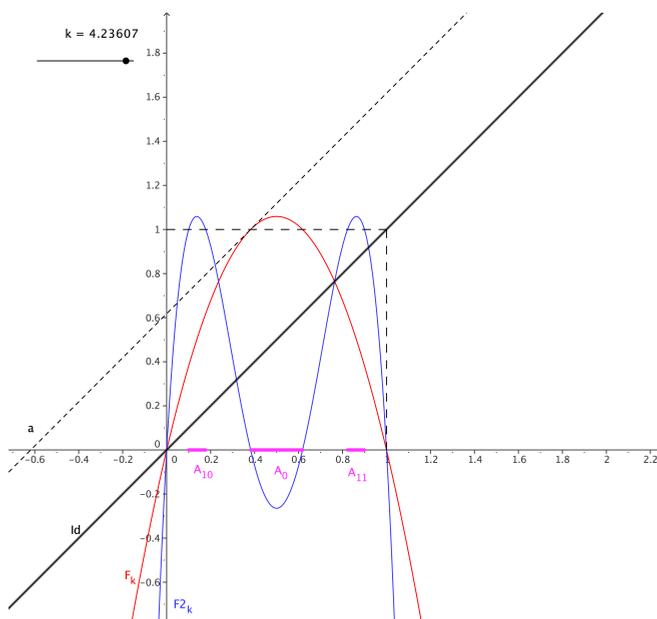


Figura 10: El conjunto invariante Λ_k .

consiste en 2^m intervalos abiertos disjuntos y cuyos puntos tienen órbitas futuras que tienden a $-\infty$. Por otra parte, el conjunto $I \setminus (A_0 \cup A_1 \cup \dots \cup A_m)$ consiste en 2^{m+1} intervalos cerrados disjuntos, con la propiedad de que si $x \in I \setminus (A_0 \cup A_1 \cup \dots \cup A_m)$ entonces $F_k^m(x) \in I$.

El conjunto

$$\Lambda_k = I \setminus \bigcup_{i=0}^{\infty} A_i$$

es el conjunto F_k -invariante maximal, esto es, Λ_k es el subconjunto más grande de $I = [0, 1]$ tal que $F_k(\Lambda_k) \subset \Lambda_k$. En particular, cuando $k > 2 + \sqrt{5}$ es fácil mostrar que Λ_k es un conjunto de Cantor, esto es, un subconjunto cerrado de I que no contiene intervalo alguno y tal que cada uno de sus puntos es un punto de acumulación de él mismo.

4. NOCIONES BÁSICAS SOBRE MAXIMA

Maxima es una versión libre y gratuita del paquete de cálculo simbólico MACSYMA, desarrollado en el MIT a finales de los años 60. Maxima cuenta con un grupo de desarrolladores muy activo, que mantienen una lista de correo extremadamente útil y a la que contribuyen excelentes científicos. El tiempo promedio de respuesta a las cuestiones que se plantean en la lista es de unas pocas horas. Maxima puede descargarse desde <http://maxima.sourceforge.net/es>, donde también se tiene el manual oficial del programa.

Maxima dispone de varias interfaces: Emacs, una basada en Tcl/Tk (la que trae el programa por defecto cuando se compilan sus fuentes, llamada xMaxima), otra en TeXmacs, etc. Para los no iniciados o los que están acostumbrados a Mathematica, probablemente la mejor opción sea wxMaxima, que viene junto con la versión de Maxima para Windows. Desde la versión 0.8.1 (la empleada en este trabajo es la 0.8.4) la interfaz que presenta wxMaxima es prácticamente idéntica a la de Mathematica: el texto y los comandos se introducen en celdas y la ejecución de un comando se lleva a cabo con Shift+Enter (o Intro). Incluso, reflejando el origen común de Maxima y Mathematica que ya hemos mencionado, los comandos son muy parecidos en ambos casos, siendo la principal diferencia que en Maxima se escriben casi siempre en minúscula (una notable excepción son las funciones definidas por el usuario, que se pueden nombrar como uno quiera) y los argumentos van entre paréntesis y no corchetes. Así, en Mathematica integraríamos x^n mediante `Integrate[x^n,x]` y en Maxima haríamos `integrate(x^n,x)`. El operador de derivación, por ejemplo, se escribe `diff(función,variable,orden)`, de manera que `diff(sin(x*y),x,2)` nos devolvería $-y^2 \sin(xy)$, la derivada segunda de $\sin(xy)$ respecto de x . Una función como la que hemos mencionado se define mediante `:=`, como en `f(x,y):=sin(x*y)`, mientras que la asignación de un valor a una variable se hace a través de dos puntos, como en el comando `A:matrix([1,2],[3,4])`, que define la matriz

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

mediante un par de listas que representan las filas y se la asigna a la variable **A**. La mayoría de los comandos son autoexplicativos: `subst(x=2,x+a-2)` realiza la sustitución de x por 2 en la expresión $x + a - 2$. Las constantes como π o el número e se escriben con el símbolo `%`: `%pi`, `%e` respectivamente. El símbolo `%` también se usa para hacer referencia a comandos ya introducidos al sistema: `%o5` hace referencia a la salida del comando que se insertó en la línea `%i5`. Si no se especifica nada, `%` hace referencia al último comando introducido. Un par de manuales muy buenos para introducirse en Maxima se pueden ver en [8, 9]. Uno de los autores tiene un manual específico para asignaturas de cálculo diferencial en la dirección <http://galia.fc.uaslp.mx/~jvallejo/ManualMaxima.pdf>.

4.1. UN EJEMPLO: SISTEMAS LINEALES DE COEFICIENTES CONSTANTES

Supongamos que queremos resolver un sistema de ecuaciones diferenciales del tipo

$$\dot{x} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} x = Ax, \quad (1)$$

donde $x(t) = (x_1(t), x_2(t))$ es una trayectoria en \mathbb{R}^2 y $A = (a_{ij})$ con $1 \leq i, j \leq 2$ es una matriz 2×2 con coeficientes reales. Maxima puede resolver directamente el sistema propuesto con el comando `desolve`. Introducimos las ecuaciones por separado (el apóstrofe ' le indica a Maxima que no evalúe la derivada, de manera que realmente tengamos una ecuación):

```
(%i1) e1: 'diff(x[1](t),t)=5*x[2](t);
```

```
(%o1) 
$$\frac{d}{dt} x_1(t) = 5 x_2(t)$$

```

```
(%i2) e2: 'diff(x[2](t),t)=-x[1](t)+2*x[2](t);
```

```
(%o2) 
$$\frac{d}{dt} x_2(t) = 2 x_2(t) - x_1(t)$$

```

Las soluciones se devuelven en forma de una lista. Para recuperarlas individualmente utilizamos comandos como `first` o `second`, que extraen el primer y segundo elementos de una lista, respectivamente:

```
(%i3) first(desolve([e1,e2],[x[1](t),x[2](t)]));
```

```
(%o3) 
$$x_1(t) = e^t \left( \frac{(2(5x_2(0) - 2x_1(0)) + 2x_1(0)) \sin(2t)}{4} + x_1(0) \cos(2t) \right)$$

```

```
(%i4) second(desolve([e1,e2],[x[1](t),x[2](t)]));
```

```
(%o4) 
$$x_2(t) = e^t \left( \frac{(2x_2(0) - 2x_1(0)) \sin(2t)}{4} + x_2(0) \cos(2t) \right)$$

```

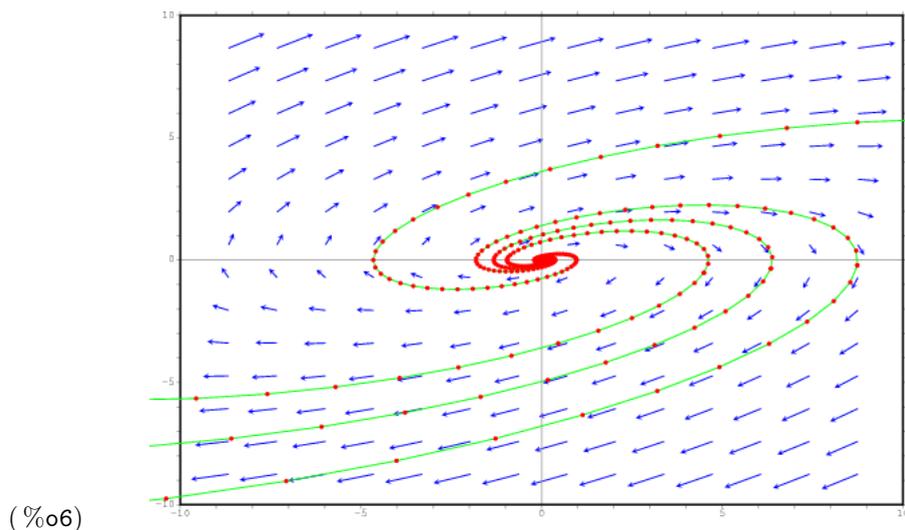
Por último, podemos representar el diagrama de fases del sistema. Para ello, necesitamos cargar primero el paquete `plotdf` mediante la orden `load`:

```
(%i5) load(plotdf);
```

```
(%o5) /usr/share/maxima/5.15.0/share/dynamics/plotdf.lisp
```

Una vez que tenemos el paquete cargado, representamos el sistema introduciendo únicamente el miembro derecho de las ecuaciones en forma de lista. En el gráfico que aparece, al hacer pulsar con el botón izquierdo del ratón en un punto, automáticamente se dibuja la trayectoria (única) que pasa por ese punto como condición inicial.

```
(%i6) plotdf([5*y, -x+2*y]);
```



5. SISTEMAS NO LINEALES

5.1. LA TÉCNICA DE LINEALIZACIÓN

La fuerza elástica que actúa sobre un péndulo viene dada por $F_e = -k \sin \theta$ (con $k > 0$) donde θ es el desplazamiento angular respecto de la posición de equilibrio. Un péndulo amortiguado está sujeto a una fuerza adicional que es proporcional a la velocidad y que tiende a frenar el movimiento, $F_a = -a\dot{\theta}$ (con $a > 0$). De hecho, en esta notación es $k = g/l$, de modo que aproximando $g = 10\text{ms}^{-2}$ y $l = 1\text{m}$, podemos tomar $k = 10\text{s}^{-2}$. Supondremos también que la masa del péndulo vale 1 y que ajustamos la escala de tiempo de manera que $k = 1$. En estas condiciones, la ecuación de Newton para el péndulo es simplemente

$$\ddot{\theta} + a\dot{\theta} + \sin \theta = 0,$$

que puede escribirse, si hacemos $\theta = x$, $\dot{\theta} = y$, como

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -ay - \sin x. \end{cases} \quad (2)$$

Éste es claramente un sistema no lineal, dependiente de un parámetro a que es el amortiguamiento. La técnica estándar para tratar tales sistemas consiste en estudiar el movimiento alrededor de una posición de equilibrio viéndolo como una perturbación de la solución correspondiente. Un punto (x_e, y_e) es una posición de equilibrio para el sistema

$$\begin{cases} \dot{x} = f(x, y), \\ \dot{y} = g(x, y) \end{cases} \quad (3)$$

si se cumple $f(x_e, y_e) = 0 = g(x_e, y_e)$. Se considera entonces un sistema lineal asociado, obtenido desarrollando por Taylor f y g alrededor de (x_e, y_e) :

$$\begin{cases} \dot{x} = D_1f(x_e, y_e)(x - x_e) + D_2f(x_e, y_e)(y - y_e), \\ \dot{y} = D_1g(x_e, y_e)(x - x_e) + D_2g(x_e, y_e)(y - y_e). \end{cases}$$

Nótese que haciendo un cambio de coordenadas

$$x - x_e \mapsto x, \quad y - y_e \mapsto y, \quad (4)$$

este sistema es

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = J_{(x_e, y_e)} F \begin{pmatrix} x \\ y \end{pmatrix}, \quad (5)$$

donde $F(x, y) = (f(x, y), g(x, y))$ y $J_{(x_e, y_e)} F$ es su Jacobiano en el punto de equilibrio (x_e, y_e) . La relación entre el sistema linealizado (5) y el original (3) puede describirse como sigue. El punto (x_e, y_e) se dice que es hiperbólico si todos los valores propios de $J_{(x_e, y_e)} F$ tienen parte real distinta de cero. Entonces, se tiene (véase [1, sección §13H], [7, sección 2.8] o [10, sección 2.2]):

TEOREMA (Großmann-Hartman). *Si (x_e, y_e) es un punto hiperbólico del sistema no lineal (3), el comportamiento de éste en un entorno de (x_e, y_e) es topológicamente equivalente al comportamiento del sistema linealizado (5) en un entorno del origen $(0, 0)$.*

Esto quiere decir que existe un homeomorfismo que lleva las órbitas del sistema (3) alrededor del punto (x_e, y_e) en órbitas del sistema (5) alrededor del $(0, 0)$ y, por tanto, el comportamiento cualitativo de los dos sistemas es el mismo. Así pues, esta técnica reduce el estudio de sistemas no lineales al de sistemas lineales en un entorno del origen. El comportamiento de estos últimos es bien conocido (puede consultarse cualquier libro de ecuaciones diferenciales, por ejemplo [2] o [3]).

5.2. EL PÉNDULO AMORTIGUADO

Vamos a aplicar estas ideas al péndulo amortiguado (2). En este caso, tenemos que $f(x, y) = y$, $g(x, y) = -ay - \sin x$. Las posiciones de equilibrio vienen dadas por las soluciones a

$$\begin{cases} y = 0, \\ -ay - \sin x = 0, \end{cases}$$

y vamos a considerar, por simplicidad, el comportamiento de este sistema no lineal alrededor del equilibrio $(x_e, y_e) = (0, 0)$. El primer paso para poder aplicar el Teorema de Großmann-Hartman es comprobar que tenemos un punto hiperbólico, y para ello Maxima dispone de los comandos `jacobian`, que nos devuelve la matriz Jacobiana de una matriz, y `eigenvalues`, que nos devuelve una lista con los valores propios seguida de otra lista con las multiplicidades respectivas:

```
(%i1) jacobian([y, -a*y-sin(x)], [x,y]);
```

```
0 errors, 0 warnings
```

```
(%o1)      ( 0      1 )
            ( -cos(x) -a )
```

```
(%i2) A:subst([x=0,y=0],%);
```

```
(%o2)      ( 0      1 )
            ( -1     -a )
```

```
(%i3) eigenvalues(A);
```

```
(%o3)      [[-sqrt(a^2-4+a)/2, sqrt(a^2-4-a)/2], [1,1]]
```

Claramente, el comportamiento del sistema depende de si $a < 2$ o $a > 2$. Estudiemos el primer caso con el valor particular $a = 1$ (correspondiente a un amortiguamiento débil):

```
(%i4) A1:matrix([0,1],[-1,-1]);
```

```
(%o4)      ( 0      1 )
            ( -1     -1 )
```

```
(%i5) eigenvalues(A1);
```

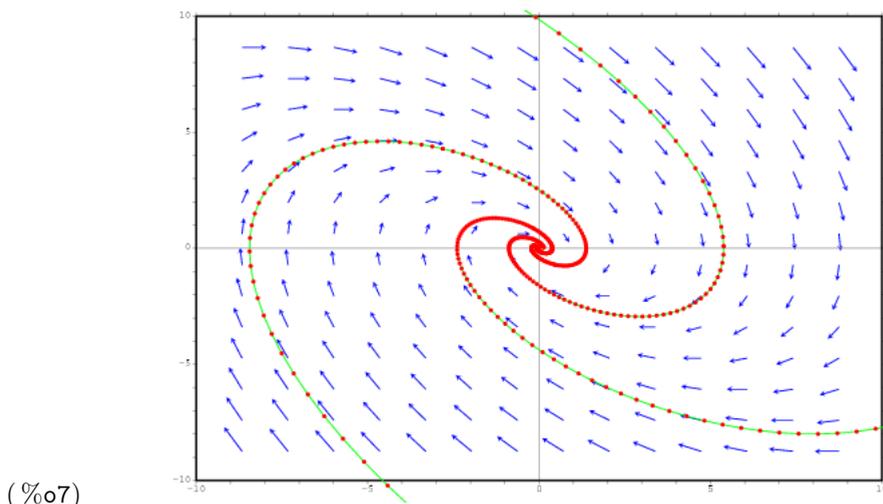
```
(%o5)      [[-sqrt(3)i+1/2, sqrt(3)i-1/2], [1,1]]
```

En este caso, los valores propios tienen parte real negativa, de modo que las soluciones se van desvaneciendo (aproximándose al valor constante 0) a medida que oscilan. El punto de equilibrio es, pues, un atractor. Podemos verlo gráficamente:

```
(%i6) load(plotdf);
```

```
(%o6)      /usr/share/maxima/5.15.0/share/dynamics/plotdf.lisp
```

```
(%i7) plotdf([y, -x-y]);
```



Veamos qué ocurre si $a > 2$ (en este caso se dice que el sistema está sobreamortiguado). Ahora tenemos la matriz

```
(%i8) A3:matrix([0,1],[-1,-3]);
```

```
(%o8)      
$$\begin{pmatrix} 0 & 1 \\ -1 & -3 \end{pmatrix}$$

```

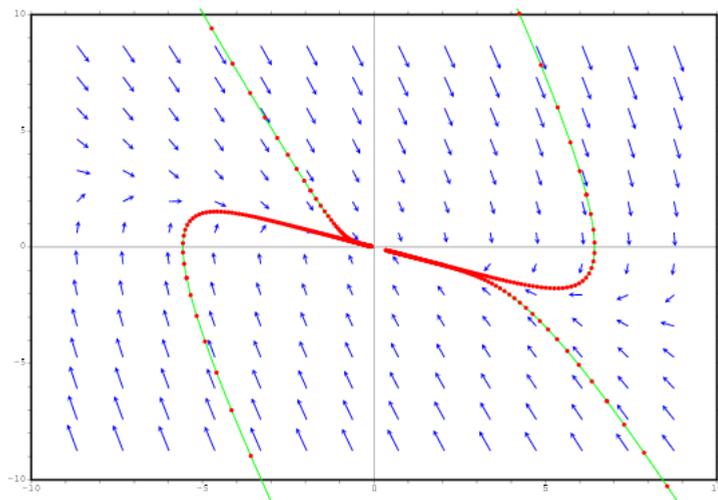
```
(%i9) eigenvalues(A3);
```

```
(%o9)      
$$\left[ \left[ -\frac{\sqrt{5}+3}{2}, \frac{\sqrt{5}-3}{2} \right], [1, 1] \right]$$

```

De nuevo los valores propios tienen parte real negativa, lo cual significa que el punto de equilibrio es un atractor. Pero en este caso no hay oscilaciones, sino que se alcanza el equilibrio de manera casi instantánea; esto se ve en el diagrama de fases porque las trayectorias «no dan vueltas» alrededor del origen, sino que van directamente hacia él.

```
(%i10) plotdf([y, -x-3*y]);
```



(%o10)

Para apreciar la complejidad de este sistema, vamos a analizar lo que ocurre en otro punto de equilibrio del sistema original. Esta vez, consideramos el $(\pi, 0)$.

```
(%i11) jacobian([y,-a*y-sin(x)],[x,y]);
```

0 errors, 0 warnings

```
(%o11) 
$$\begin{pmatrix} 0 & 1 \\ -\cos(x) & -a \end{pmatrix}$$

```

```
(%i12) B:subst([x=%pi,y=0],%);
```

```
(%o12) 
$$\begin{pmatrix} 0 & 1 \\ 1 & -a \end{pmatrix}$$

```

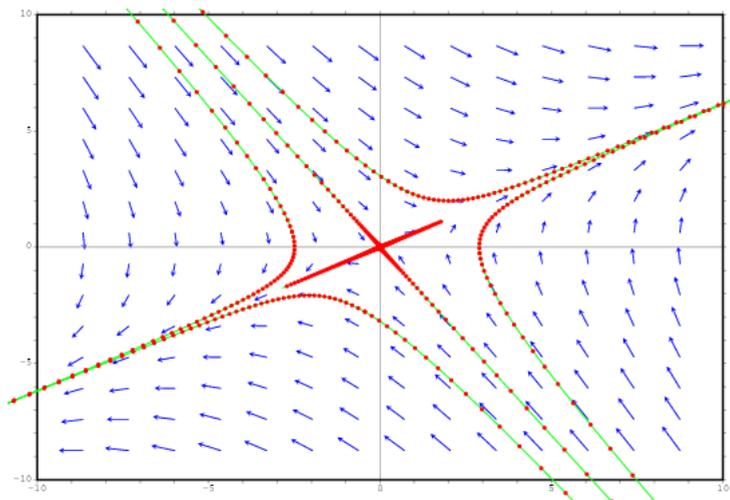
```
(%i13) eigenvalues(B);
```

```
(%o13) 
$$\left[ \left[ -\frac{\sqrt{a^2 + 4} + a}{2}, \frac{\sqrt{a^2 + 4} - a}{2} \right], [1, 1] \right]$$

```

Aquí los valores propios son reales pero de signo contrario. Por tanto, el punto de equilibrio $(\pi, 0)$ es un punto de silla: todas las soluciones excepto dos separatrices se alejan del equilibrio:

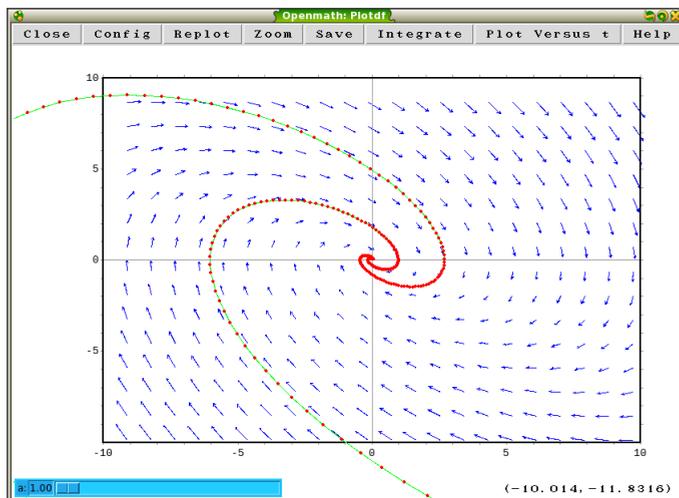
```
(%i14) plotdf([y,x-y]);
```



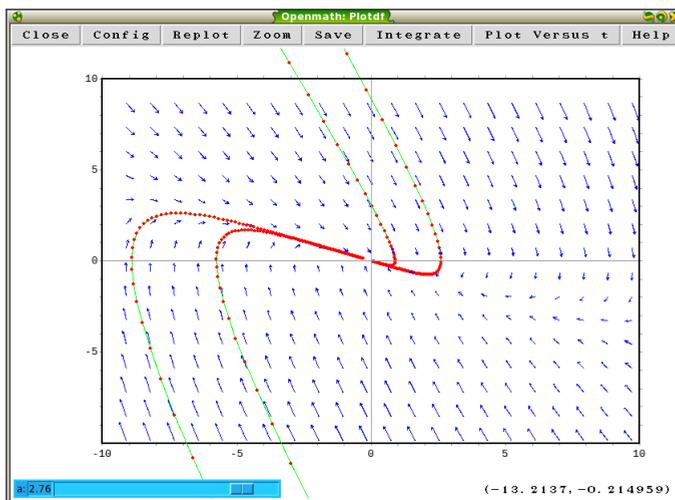
(%o14)

Podemos hacer esto de manera mucho más eficiente aprovechando las capacidades gráficas de Maxima y Gnuplot, introduciendo un deslizador (slider) que modifique los valores del parámetro a ; de esta forma, se visualiza mucho más claramente la dependencia del comportamiento con los parámetros:

```
(%i15) plotdf([y, -x-a*y], [x,y], [parameters,"a"],
[sliders,"a=1:3"]);
```



(%o15)



(%o15)

5.3. SOLUCIONES EXACTAS DE ECUACIONES DE SEGUNDO ORDEN

Utilizando Maxima, podemos resolver de manera exacta el problema del péndulo amortiguado si suponemos que estamos en el régimen de oscilaciones pequeñas, de manera que la ecuación de Newton sea

(%i1) eq1:'diff(%theta,t,2)+a*'diff(%theta,t)+ %theta=0;

(%o1)
$$\left(\frac{d}{dt}\theta\right)a + \frac{d^2}{dt^2}\theta + \theta = 0$$

Para resolver (cuando sea posible) una ecuación diferencial de segundo orden exactamente, Maxima dispone del comando ode2. Fijémonos que, en este caso, la forma concreta de la solución depende de los valores del parámetro. Maxima reconoce esto y por eso pregunta la información que necesita:

(%i2) ode2(eq1,%theta,t);

Is (a-2)*(a+2) positive, negative, or zero? p;

(%o2)
$$\theta = \%k1 e^{\frac{(\sqrt{a^2-4}-a)t}{2}} + \%k2 e^{\frac{(-\sqrt{a^2-4}-a)t}{2}}$$

En este caso, al responder «p» (positivo) le hemos dicho a Maxima que $a > 2$. Los otros casos se calculan análogamente:

(%i3) ode2(eq1,%theta,t);

Is (a-2)*(a+2) positive, negative, or zero? n;

(%o3)
$$\theta = e^{-\frac{a}{2}t} \left(\%k1 \sin\left(\frac{\sqrt{4-a^2}t}{2}\right) + \%k2 \cos\left(\frac{\sqrt{4-a^2}t}{2}\right) \right)$$

```
(%i4) ode2(eq1,%theta,t);
```

Is $(a-2)*(a+2)$ positive, negative, or zero? z;

```
(%o4) 
$$\theta = (\%k2 t + \%k1)e^{-\frac{a t}{2}}$$

```

Podemos incluso resolver el problema con condiciones iniciales. Supongamos, por ejemplo, que $a = 3$:

```
(%i5) eq2:'diff(%theta,t,2)+3*'diff(%theta,t)+ %theta=0;
```

```
(%o5) 
$$\frac{d^2}{dt^2} \theta + 3 \left( \frac{d}{dt} \theta \right) + \theta = 0$$

```

```
(%i6) ode2(eq2,%theta,t);
```

```
(%o6) 
$$\theta = \%k1 e^{\frac{(\sqrt{5}-3)t}{2}} + \%k2 e^{\frac{(-\sqrt{5}-3)t}{2}}$$

```

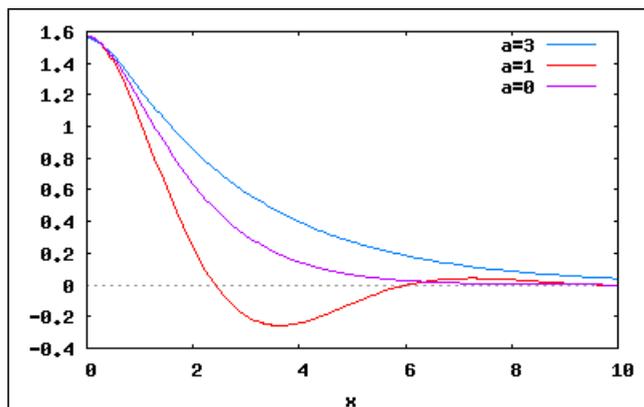
La solución al problema de valor inicial se obtiene con el comando `ic2`. En este ejemplo, imponemos que $\theta(0) = \frac{\pi}{2}$ y $\theta'(0) = 0$:

```
(%i7) ratsimp(ic2(% ,t=0,%theta=%pi/2,'diff(%theta,t)=0));
```

```
(%o7) 
$$\theta = \frac{\left( (3\sqrt{5} + 5) \pi e^{\sqrt{5}t} + (5 - 3\sqrt{5}) \pi \right) e^{-\frac{\sqrt{5}t}{2} - \frac{3t}{2}}}{20}$$

```

Lo mismo puede hacerse con los otros casos ($a < 2$ y $a = 2$). Los resultados pueden verse en una misma gráfica, que confirma los datos cualitativos que obtuvimos en la sección anterior:



6. CONCLUSIONES

En este trabajo, hemos tratado de mostrar posibles usos de un par de herramientas de software libre (GeoGebra y Maxima) en cursos a nivel universitario, lo cual aporta dos ventajas: ahorro en coste de licencias y que los estudiantes puedan disponer en sus casas del mismo software que usan en las clases.

Las características de interactividad de ambos programas los convierte en buenos candidatos a figurar en proyectos educativos basados en el uso de tecnologías informáticas para la enseñanza de las matemáticas, tanto en cursos de bachillerato como en cursos universitarios.

El tema elegido para ilustrar estas características (sistemas dinámicos) no es el único posible, desde luego, pero sí uno especialmente abundante en ejemplos de interés que permiten un tratamiento completo tanto desde el punto de vista del cálculo simbólico como del cálculo numérico.

Agradecemos desde aquí la labor del árbitro anónimo quien, con sus detalladas sugerencias, sin duda ha contribuido a mejorar la calidad de la exposición. José A. Vallejo también agradece el apoyo del Consejo Nacional de Ciencia y Tecnología (México), mediante el proyecto CB-2007-J2-78791.

REFERENCIAS

- [1] V. I. ARNOLD, *Geometrical Methods in the Theory of Ordinary Differential Equations*, Springer Verlag, New York, 1983.
- [2] P. BLANCHARD, R. L. DEVANEY Y G. R. HALL, *Ecuaciones Diferenciales*, Thomson International, México, 2007.
- [3] C. FERNÁNDEZ, J. M. VEGAS Y F. J. VÁZQUEZ, *Ecuaciones Diferenciales y en Diferencias*, Paraninfo, Madrid, 2003.
- [4] M. HOHENWARTER Y K. FUCHS, Combination of dynamic geometry, algebra and calculus in the software system GeoGebra, *Proceedings of Computer Algebra and Dynamic Geometry Systems in Mathematics Teaching 2004* (Pecs, Hungría), CADGME, 2005.
- [5] J. HOHENWARTER Y M. HOHENWARTER, Introduction to GeoGebra. Documento disponible en la dirección electrónica <http://www.lulu.com/content/2650846>.
- [6] R. LOSADA LISTE, GeoGebra: La eficiencia de la intuición, *La Gaceta de la RSME*, vol. 10, no. 1, 223–239, 2008.
- [7] L. PERKO, *Differential Equations and Dynamical Systems*, Springer Verlag, New York, 1991.
- [8] R. RODRÍGUEZ GALVÁN, Maxima con wxMaxima: software libre en el aula de Matemáticas. Documento disponible en la dirección electrónica <https://forja.rediris.es/projects/guia-wxmaxima>.
- [9] M. RODRÍGUEZ RIORTORTO, Primeros pasos en Maxima. Documento disponible en la dirección electrónica <http://www.telefonica.net/web2/biomates/maxima/max.pdf>.

- [10] S. WIGGINS, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Springer Verlag, New York, 1990.

ANTONIO MORANTE, FACULTAD DE CIENCIAS, UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ (MÉXICO)

Correo electrónico: amorante@ciencias.uaslp.mx

JOSÉ A. VALLEJO, FACULTAD DE CIENCIAS, UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ (MÉXICO)

Correo electrónico: jvallejo@fc.uaslp.mx

Página web: <http://galia.fc.uaslp.mx/~jvallejo>