

**V CONFERENCIA NACIONAL DE CIENCIAS DE LA
COMPUTACIÓN
CCBOL'98**

Del 16 al 20 de Noviembre 1998 en Potosí

Tutorial T2

**SISTEMAS NO LINEALES. CONCEPTOS,
ALGORITMOS Y APLICACIONES**

Prof. José Manuel Gutiérrez

Dpto. de Matemática Aplicada y Ciencias de la Computación

Universidad de Cantabria (España)

<http://ccaix3.unican.es/~gutierjm>

Resumen

La aparición de la geometría fractal y el estudio de los sistemas no lineales han permitido analizar y caracterizar fenómenos irregulares que escapaban a las técnicas de análisis clásicas. En los últimos años han sido presentadas numerosas aplicaciones prácticas que muestran el amplio abanico de aplicaciones de esta disciplina. Algunas aplicaciones especialmente destacables con la caracterización de atractores extraños de sistemas dinámicos caóticos, el análisis de algunos tipos irregulares de series temporales (meteorológicas, económicas, etc.), el análisis de la estructura de las secuencias de ADN, etc. En este tutorial se presentan los conceptos básicos de la geometría fractal y los sistemas no lineales (desde un punto de vista teórico y computacional) y se describen algunas aplicaciones prácticas. Con objeto de ilustrar nuevas posibilidades en la programación científica, se mostrará la forma de implementar algunos de los algoritmos que se describen en el tutorial mediante el programa *Mathematica*. Este programa incorpora eficientes herramientas de programación funcional, que permiten realizar los cálculos de forma sencilla, y herramientas de representación gráfica, para observar los resultados directamente.

Introducción

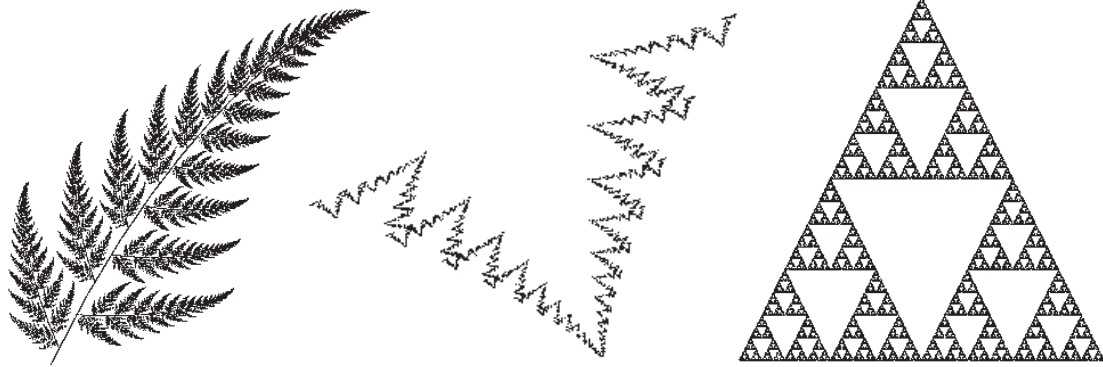
Durante muchos años los sistemas y modelos lineales han sido utilizados sistemáticamente para describir y modelar la dinámica de muchos sistemas físicos, químicos, económicos, etc. Sin embargo, en los últimos años se ha comprobado que los sistemas no lineales pueden presentar dinámicas muy complejas que no pueden aproximarse mediante modelos lineales. Quizás el más claro ejemplo de este fenómeno lo constituyen los conjuntos de Mandelbrot y de Julia. Estos conjuntos son la representación gráfica de los infinitos comportamientos que se obtienen al iterar una ecuación no lineal (una ecuación cuadrática). Por el contrario, la iteración de una ecuación lineal sólo puede dar lugar a dos situaciones distintas: puede converger a un valor constante, o divergir a infinito. Entre la gran variedad de comportamientos posibles de un sistema no lineal el conocido por *caos determinista* destaca por su complejidad. Los sistemas caóticos son sensibles a pequeñas perturbaciones externas y, por tanto se comportan de forma impredecible, a pesar de estar definidos por ecuaciones deterministas.

La geometría fractal y la teoría de los sistemas dinámicos están íntimamente ligadas, ya que la región del espacio hacia la que tiende asintóticamente una órbita caótica tiene estructura fractal (*atractores extraños*). Por tanto, la geometría fractal permite estudiar el soporte sobre el que se definen los sistemas dinámicos caóticos. Los objetos fractales tienen propiedades muy particulares, como la autosemejanza y la apariencia irregular, que permiten caracterizarlos en base a medidas cuantitativas relativas a su grado de irregularidad. La más popular de estas medidas cuantitativas es la dimensión fractal, una extensión de la dimensión euclídea para objetos autosemejantes.

Geometría Fractal

La geometría tradicional (euclídea) se encarga de las propiedades y de las mediciones de objetos tales como puntos, líneas, planos y volúmenes. A diferencia de estos objetos geométricos clásicos, que poseen propiedades de continuidad y diferenciabilidad, existen otros objetos geométricos irregulares que presentan estructura a cualquier escala y tienen un número infinito de singularidades (puntos no diferenciables). Ejemplos de estos objetos son las formas encontradas en la naturaleza, como montañas, franjas costeras, sistemas hidrográficos, nubes, hojas, árboles, vegetales, copos de nieve, y un sinnúmero de otros objetos que no son fácilmente descritos por la geometría tradicional. La geometría fractal provee una descripción matemática de estas formas irregulares que se denominan *fractales*. Una de las principales características de los fractales es la invarianza a los cambios de escala; es decir, un objeto fractal posee estructura a cualquier escala formada por copias de sí mismo a menor escala. La siguiente figura muestra tres objetos fractales típicos (el helecho de Barnsley, el sistema de funciones iteradas Zig-Zag, y triángulo de

Sierpinsky). Podemos ver que, por ejemplo, el triángulo de Sierpinsky está formado por tres copias de sí mismo (a escala 0.5).



Los fractales fueron concebidos aproximadamente en 1890 por el francés Henri Poincaré, que fue pionero en el estudio cualitativo de los fenómenos no lineales. Sus ideas fueron extendidas más tarde fundamentalmente por dos matemáticos Gastón Julia y Pierre Fatou, hacia 1918. El estudio fue renovado en la década de los 70 gracias al fuerte impulso dado por el desarrollo de la computadora digital. Benoit Mandelbrot, con sus experimentos de computadora, es considerado como el padre de la geometría fractal. Él acuñó la palabra fractal derivándola del adjetivo latín fractus. El correspondiente verbo latino: frangere, significa romper, crear fragmentos irregulares.



Gastón Julia



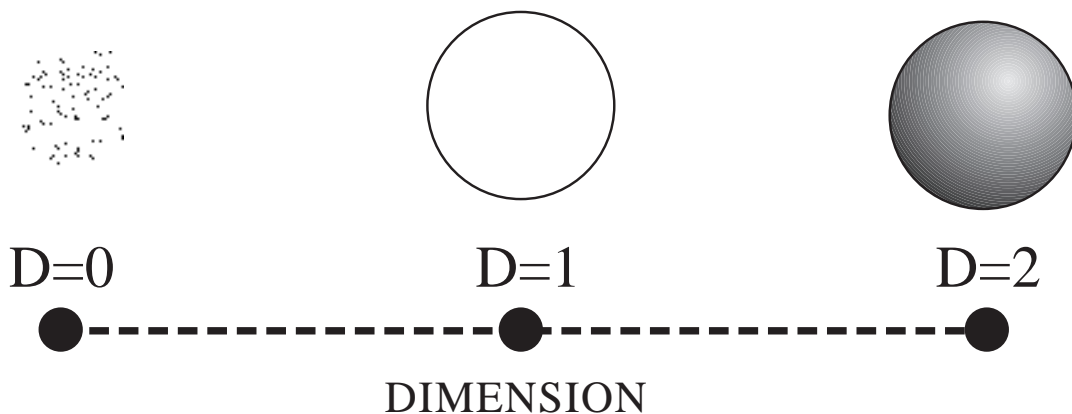
Benoit Mandelbrot

El desarrollo de la geometría fractal ha permitido obtener parámetros cuantitativos para definir el “grado de irregularidad” de un determinado objeto. Uno de los parámetros más representativos es el de *dimensión fractal*, una generalización de la dimensión euclídea para objetos autosemejantes.

Dimensión Fractal

El concepto de dimensión euclídea asigna un número natural a los distintos objetos geométricos que pueden definirse en un espacio dado. Por ejemplo, un conjunto contable

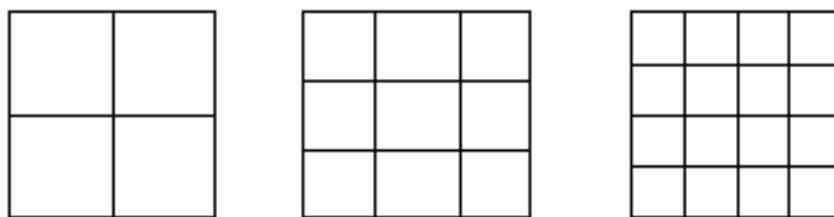
de puntos tiene dimensión cero, una curva dimensión uno, una superficie dimensión 2, etc. Este concepto de dimensión tiene diversas interpretaciones intuitivas como, por ejemplo, el número de grados de libertad que son necesarios para definir el objeto.



Otra forma de entender el concepto de dimensión es la siguiente. Si partimos de un segmento de longitud l , y lo partimos en segmentos de longitud L obtendremos $N(L)$ partes, de manera que $N(L).L^1 = l$, cualquiera que sea L .



Si el objeto inicial es un cuadrado de superficie l , y lo comparamos con unidades cuadradas, cuyo lado tenga de longitud L , el número de unidades que es necesario para recubrirlo $N(L)$, cumple $N(L).L^2 = l$.



De todo esto podemos generalizar que la dimensión de un objeto geométrico es el número D que cumple:

$$N(L).L^D = l \Leftrightarrow D = \log(N(L))/\log(l/L) \tag{1}$$

donde $N(L)$ es el número de objetos elementales, o de unidades, de tamaño L que recubren el objeto.

Asociado al concepto de dimensión está el de *ley de potencias*. La “medida” (longitud y área en los ejemplos anteriores) de un objeto de dimensión D medido a escala $s=l/L$ se

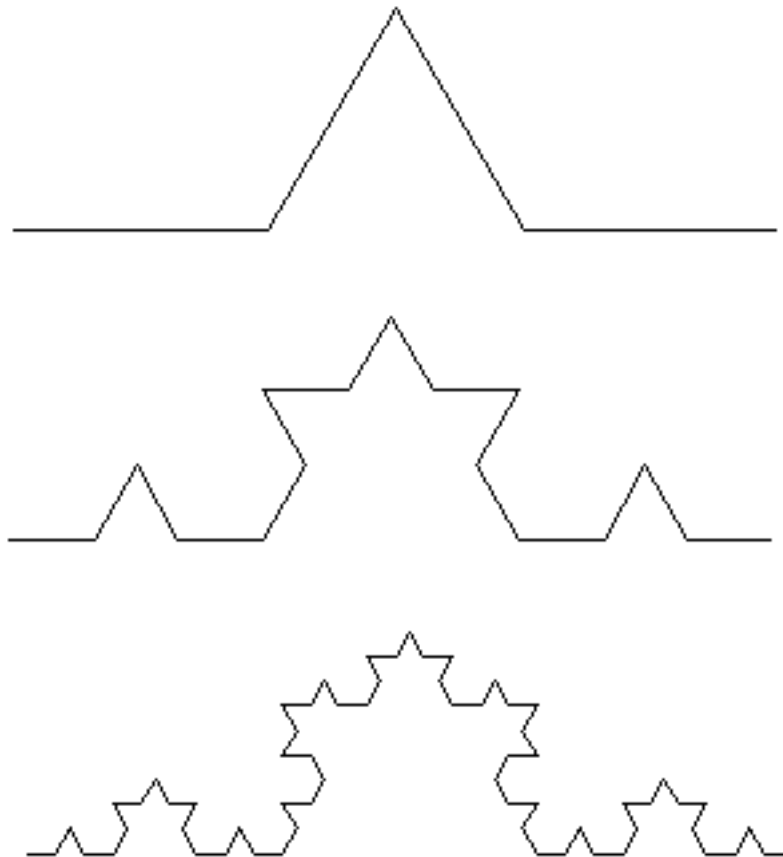
calcula como $m = s^D$. Como se verá más adelante, las leyes de potencias son típicas de objetos con autosemejanza.

En el caso de fractales, que no poseen una escala característica, el concepto de longitud no está claramente definido: Cuando se quiere medir una línea fractal con una unidad determinada, siempre habrá objetos más finos que escapan a la sensibilidad de la regla o el instrumento utilizado; por tanto, a medida que aumenta la sensibilidad del instrumento aumenta la longitud de la línea.

Tomemos, por ejemplo, la curva de Koch.



Cada paso en la génesis de la curva aumenta un tercio su longitud. Así, cuando medimos este objeto a escalas cada vez más finas nos vamos encontrando con:

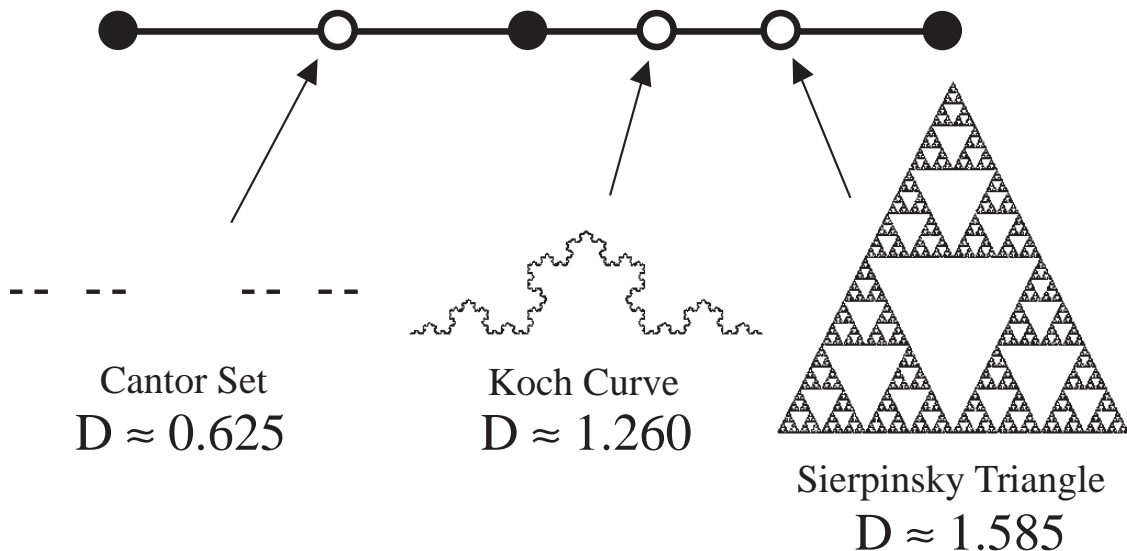


La longitud de cada curva es $\frac{4}{3}$ la de la anterior. Es decir, la curva de Koch (construida en el límite del proceso anterior) tiene longitud infinita.

El concepto de dimensión fractal extiende la idea de dimensión euclídea dada en (1) para abarcar a este tipo de objetos. Ahora, $N(L)$ es el número de conjuntos de longitud L que hacen falta para cubrir el objeto (una definición más rigurosa de la dimensión fractal es la dimensión de Hausdorff, definida por Felix Hausdorff en 1919). Por ejemplo, la dimensión fractal de la curva de Koch es

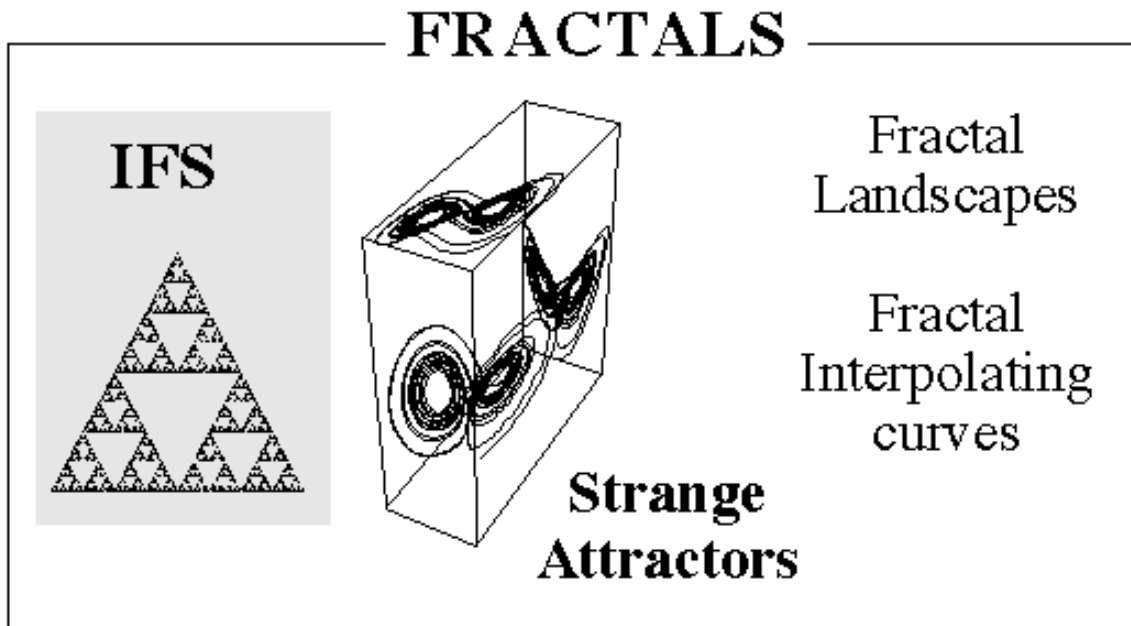
$$D = \log(4)/\log(3) = 1.2618\dots$$

Se suele aceptar, e incluso definir, que un objeto es fractal sólo cuando su dimensión fractal es mayor que su dimensión euclídea. A continuación se muestran algunos objetos fractales y su ubicación comparativa, en relación a sus dimensiones.



Modelos Fractales

En la actualidad existen numerosos modelos matemáticos que permiten definir objetos fractales asociados con problemas particulares. Por ejemplo, se puede construir una curva de interpolación fractal asociada a un conjunto de puntos; también son populares los paisajes fractales creados por ordenador y que simulan paisajes reales de continentes, islas, etc.; otros objetos fractales populares son los atractores extraños asociados a sistemas dinámicos no lineales (como el atractor de Lorenz mostrado en la siguiente figura).



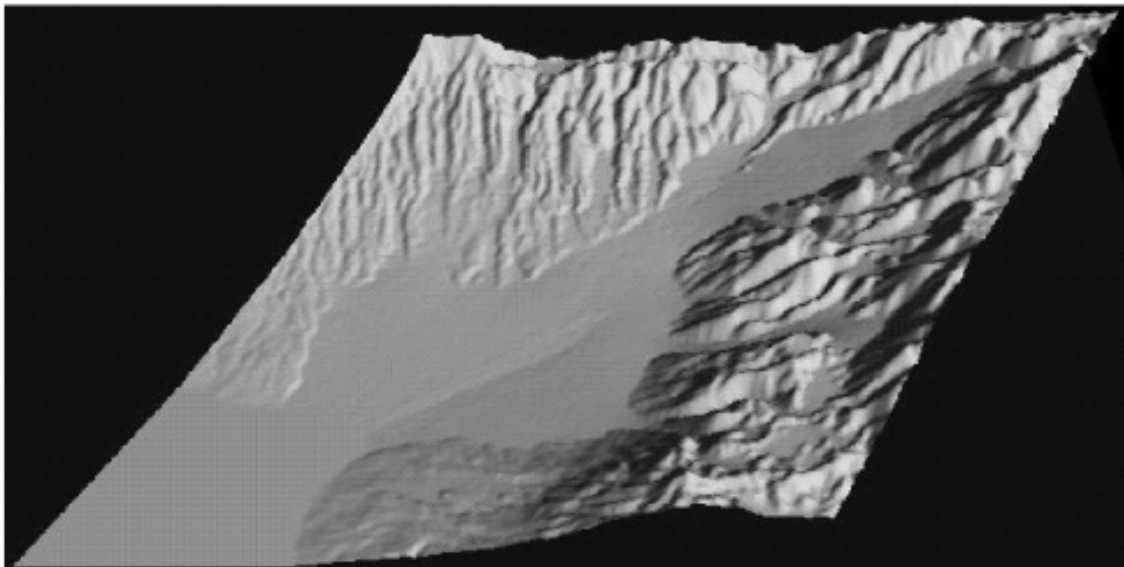
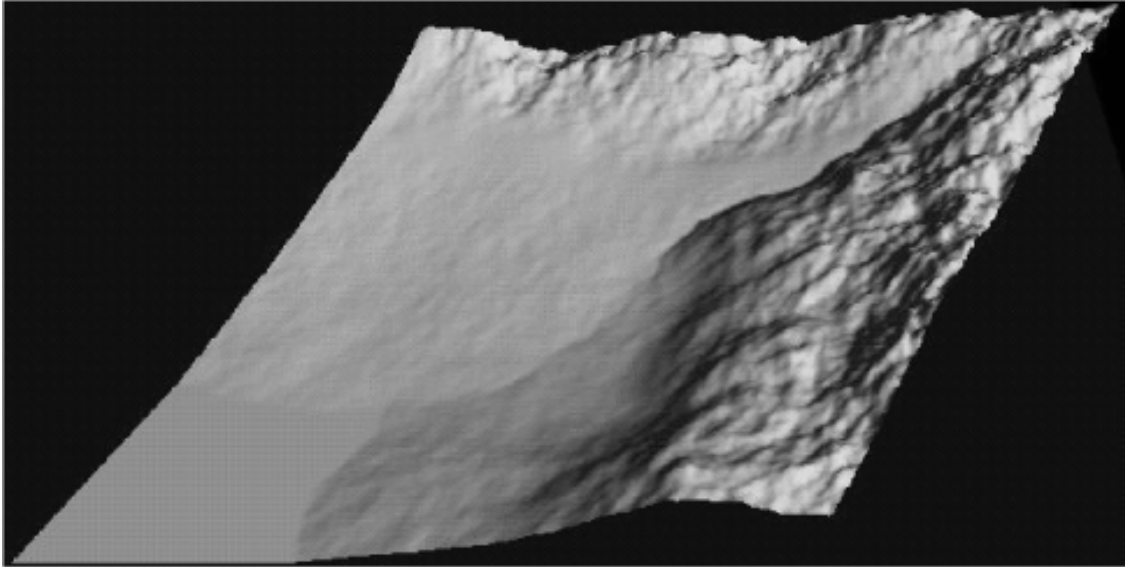
Uno de los modelos matemáticos más populares para crear objetos fractales es el conocido por *sistema de funciones iteradas* (IFS). En este caso, el fractal está definido como el único subconjunto invariante a la unión de un conjunto finito de transformaciones lineales que forman el IFS (por tanto, el fractal se puede descomponer en un conjunto finito de copias afines de sí mismo). Estos modelos se analizan con detalle en Barnsley (1988).

Aplicaciones

Como ejemplos de aplicaciones de la geometría fractal, a continuación se describe una simulación de un proceso erosivo y un método de compresión.

Modelo de Erosión

En la figura siguiente se muestra un paisaje fractal recreado por ordenador (figura superior) y una versión erosionada de este paisaje (figura inferior) creada con un modelo fractal de erosión en el que las irregularidades del terreno se van incrementando en función de parámetros que simulan la erosión fluvial.



Algoritmos de Compresión Fractal

Las técnicas de compresión fractal están basadas en los sistemas de funciones iteradas antes descritos. Aún cuando una imagen no posee aparentemente la propiedad de autosemejanza, es posible que algunas partes de ella sean semejantes a otras. Por tanto, si encontrásemos las transformaciones lineales necesarias para que distintos subconjuntos de la imagen se transformen en otros, podríamos reemplazar la imagen por las transformaciones que la definen (cada transformación lineal en el plano viene dada por 6 parámetros). De esta forma se puede comprimir eficientemente una imagen.

A continuación se muestran los resultados obtenidos tras dos compresiones fractales distintas, de distinta calidad, en las que la imagen se descompone en bloques de distinto tamaño, según la calidad de compresión deseada.

Fractal Image Compression

Lenna 256 colors (gray scale)

BMP
66330 bytes

Fractal
24619 bytes

Fractal
2852 bytes



Iteración (los conjuntos de Julia y Mandelbrot)

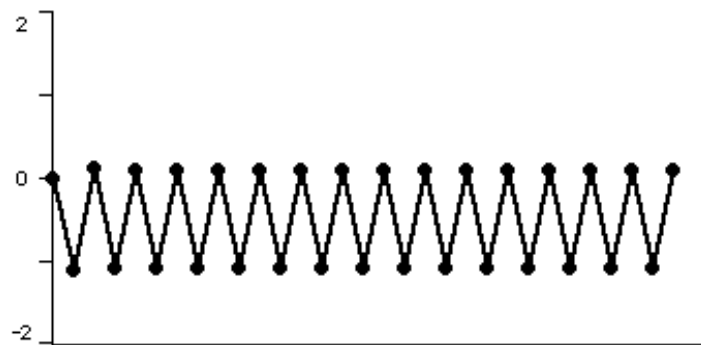
Anteriormente ya se mencionó que la dinámica iterativa de ecuaciones no lineales puede dar lugar a una gran variedad de comportamientos. Por ejemplo, consideremos la función $z^2 + c$, para distintos valores de c . Para iterar esta ecuación, comenzamos con lo que llamaremos una *semilla* para la iteración, z_0 . Aplicando la función $z^2 + c$ a z_0 obtenemos

$z_1 = z_0^2 + c$, y así sucesivamente. La lista de números z_0, z_1, z_2, \dots , generada por esta iteración se denominada órbita de z_0 . Cabe preguntarse:

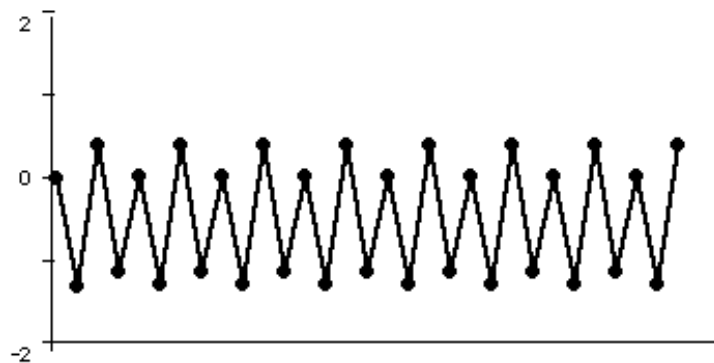
- ¿Cuál es el destino de estas órbitas?
- ¿Convergen o divergen?
- ¿Son cíclicos o se comportan erráticamente?

Comencemos con unos ejemplos. Consideremos el valor $c = 1$. Si elegimos la semilla 0 , la órbita es: $z_0 = 0, z_1 = 1, z_2 = 2, z_3 = 5, z_4 = 26$; es decir, la órbita diverge.

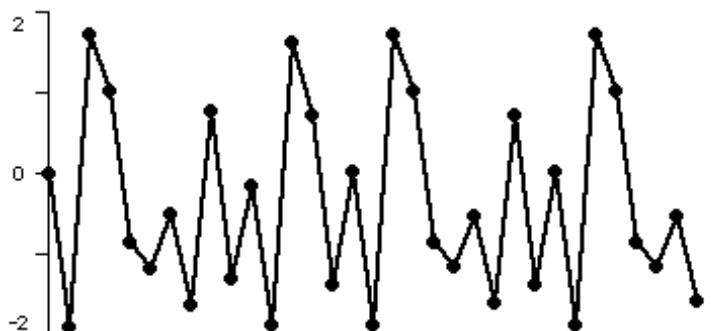
Si ahora suponemos que $c = -1$, para la semilla 0 la órbita es $z_0 = 0, z_1 = -1, z_2 = 0, z_3 = -1, z_4 = 0$. Vemos que la órbita oscila entre los valores 0 y -1 (un ciclo de periodo 2).



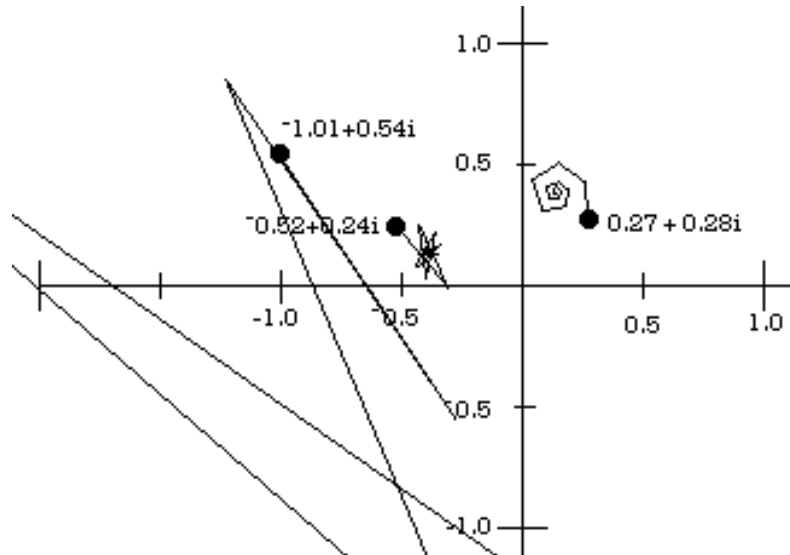
Para valores distintos del parámetro, la órbita puede oscilar entre un número mayor de valores ($c = -1.3$)



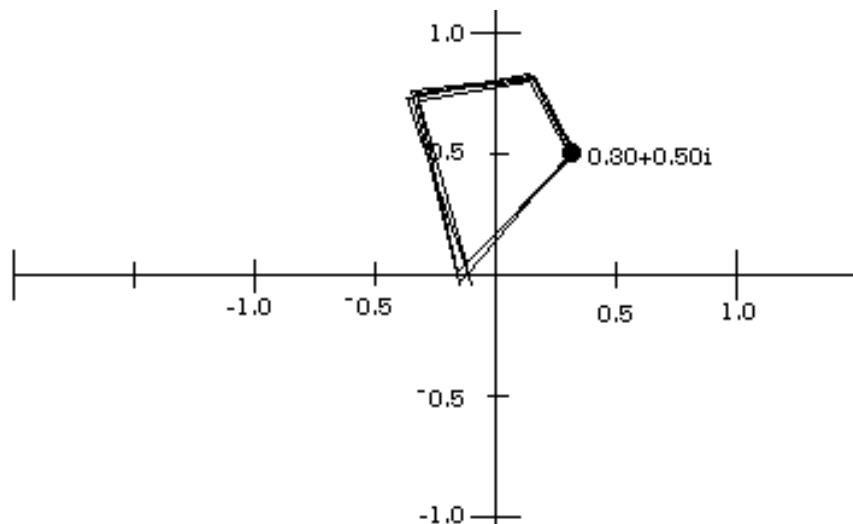
o incluso oscilar de forma aparentemente errática ($c = -1.9$). Este último caso es el conocido por caos determinista.



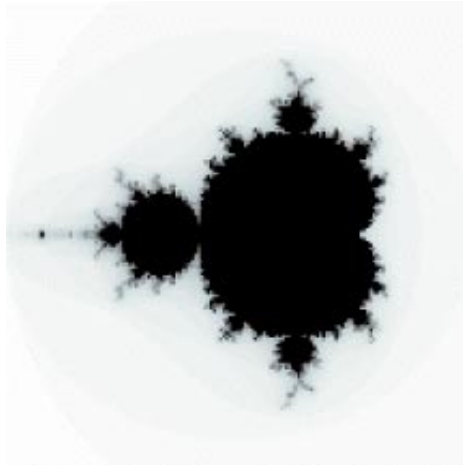
En general, la órbita asociada a un número complejo será una órbita de números complejos, pero la dinámica es la misma. La figura siguiente muestra los procesos iterativos asociados a tres condiciones iniciales distintas (los números complejos $0.27+0.28i$, $-0.52+0.24i$, y $-1.01+0.54i$, respectivamente). Podemos ver que mientras las órbitas de los primeros valores permanecen acotadas, la órbita de la tercera semilla diverge.



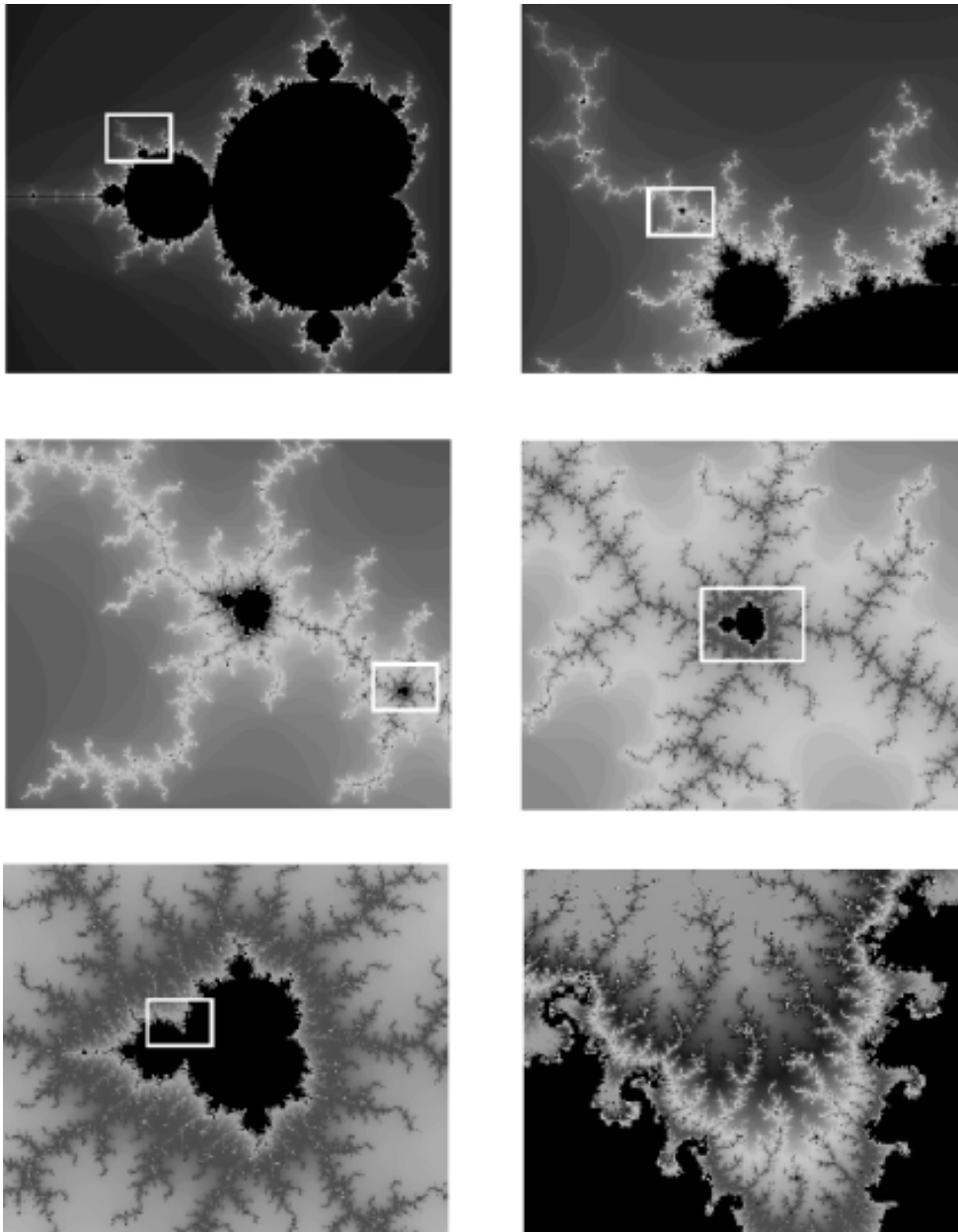
Asimismo, no todas las órbitas que permanecen acotadas presentan el mismo comportamiento: mientras unas convergen a un valor constante (como en la figura anterior), otras oscilan entre un número finito de puntos (como en la figura siguiente) o, incluso, oscilan en un dominio acotado de forma aparentemente errática.



El conjunto de Mandelbrot es una versión gráfica de los posibles comportamientos de este proceso iterativo. El conjunto de Mandelbrot está formado por aquellos valores c del plano complejo tales que la órbita de 0 está acotada. La siguiente figura muestra la complejidad de este conjunto que está formado por copias a menor escala de sí mismo (*auto semejanza*). Esta es una característica propia de los objetos fractales.



La compleja estructura de este conjunto se puede apreciar ampliando distintas regiones consecutivamente como se muestra en la siguiente figura



De la secuencia de ampliaciones puede observarse la estructura autosemejante de este conjunto (contiene copias de sí mismo a escalas menores).

A continuación se muestran dos formas posibles de implementar un comando para representar el conjunto de Mandelbrot: utilizando el lenguaje C y utilizando el programa Mathematica (ver Wolfram, 1996).

Implementación del conjunto de Mandelbrot en C

Para la implementación de un algoritmo para representar el conjunto de Mandelbrot sólo es necesario responder a la siguiente pregunta: ¿Cómo sabemos si la órbita de 0 para un valor de c dado realmente escapa al infinito?. El criterio de escape (fácilmente comprobable) nos dice que una órbita diverge si alguna vez sale del círculo de radio 2 centrado en el origen.

```
#include <graphics.h>
#include <conio.h>

void main()
{
    int sx=320;           //Screenwidth
    int sy=240;           //Screenheight
    double xmin=-2;      //smallest real value (x-axis)
    double xmax=1.25;    //largest real value (x-axis)
    double ymin=-1.25;   //smallest imaginary value (y-axis)
    double ymax=1.25;    //largest imaginary value (y-axis)
    int maxiter=96;      //Max number of iterations

    double old_x;        //temporary variable to store x-value
    double fx,fy;
    int m;                //variable to store number of iterations

    int gd=VGA,gm=2;     //MS-DOS spesific graphics-init.

    double dx=(xmax-xmin)/sx; //how much to add for each x-pixel?
    double dy=(ymax-ymin)/sy; //how much to add for each y-pixel?

    int px;               //Variable storing current x-pixel
    int py=0;             //Variable storing current y-pixel
    double x;             //Variable storing current x-value
    double y=ymin;        //Variable storing current y-value
    initgraph(&gd,&gm,"c:\\tc\\bgi\\"); //code. Update the path!!!

    while (py<sy) {
        px=0;
        x=xmin;
        py++;
        while (px<sx) {
            px++;

            fx=0;
```

```

        fy=0;
        m=0;
        do {
            old_x=fx;
            fx=fx*fx-fy*fy+x;
            fy=2*old_x*fy+y;
            m++;
        } while (((fx*fx+fy*fy)<4) && (m<maxiter));
        putpixel(px,py,m);      //MS-DOS spesific code to give pixel(x,y)
                                //the color (m)
        x+=dx;
    }
    y+=dy;
}

getch();
closegraph();                //MS-DOS spesific code to close
graphicscreen
}

```

Este es el programa que se ha utilizado para obtener los gráficos anteriores, especificando el tamaño del dibujo (variables s_x y s_y), los valores máximo y mínimo de las partes reales (x_{\min} y x_{\max}) y de las partes imaginarias (y_{\min} e y_{\max}), así como el máximo número de iteraciones realizado para comprobar si la órbita converge o diverge ($maxiter$). Nótese que cuanto mayor sea este último valor, mejor será la precisión del gráfico.

Implementación del conjunto de Mandelbrot con Mathematica

La implementación del algoritmo anterior con el programa Mathematica es mucho más sencilla e intuitiva.

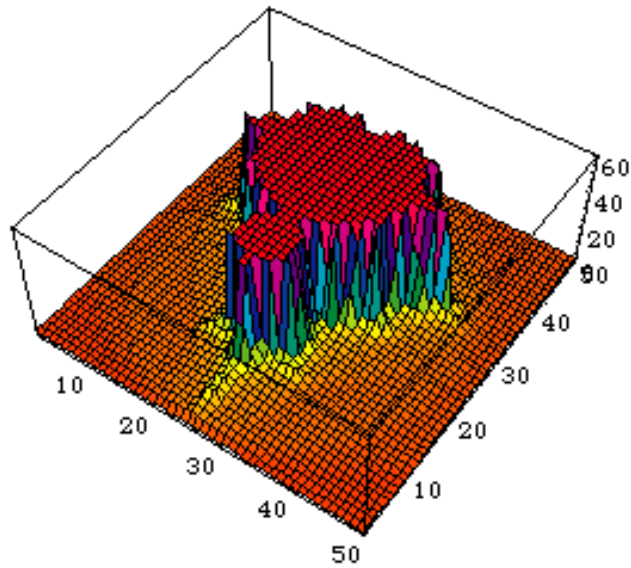
```

mandel = Compile[{{n,_Integer},{c0,_Complex},{r,_Real},
                {lim,_Integer},{i,_Integer},{j,_Integer}},
Module[{dx=r/(n-1),dy=r/(n-1),z=I,c=I,k=1},
  z=0; c=c0+(2*i-1-n)*dx+(2*j-1-n)*I*dy; k=0;
  While[Abs[z]<2 && k<lim, z=z^2+c; k++; k] ];

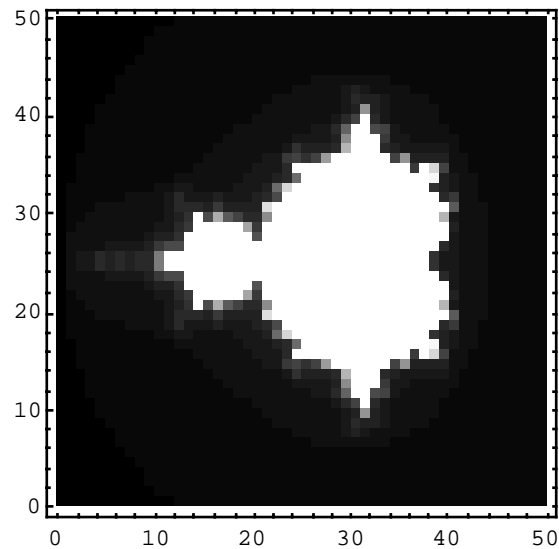
n=50; ans=Table[mandel[n, -0.5+.0*I, 1.5, 60, i, j],{i,1,n}, {j,1,n}];

ListPlot3D[ans, ColorFunction->Hue, ViewPoint->{1.317, -1.859, 2.502}]

```

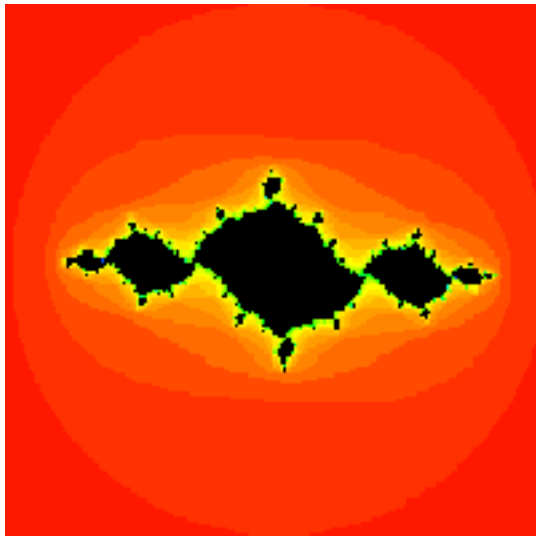


```
ListDensityPlot[ans//Transpose, Mesh->False];
```

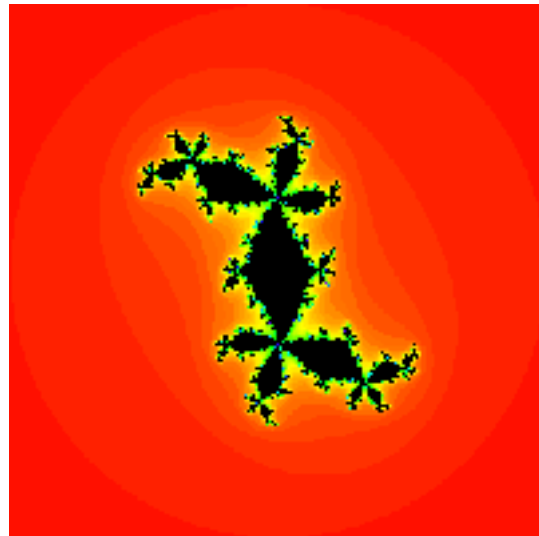


Los Conjunto de Julia

Para calcular el conjunto de Mandelbrot nos interesábamos sólo por la órbita de 0 para cada valor de c . Para los conjuntos Julia, fijamos un valor de c y luego consideramos el destino de todas las posibles semillas para ese valor fijo de c . Aquellas semillas cuyas órbitas no escapan, forman el conjunto Julia. Por tanto obtenemos un conjunto Julia diferente para cada c diferente. En la siguiente figura se muestran dos conjuntos de Julia distintos para dos valores de c :



$$c = -1.037 + 0.17 i$$



$$c = 0.295 + 0.55 i$$

Uno de los teoremas más sorprendentes de la geometría fractal data de 1919 y fue comprobado por Gastón Julia y Pierre Fatou. La Dicotomía Fundamental para la ecuación $z^2 + c$ es: para cada valor de c , el conjunto de Julia es o un conjunto conexo (una pieza) o un conjunto completamente desconexo (infinitas piezas). Sorprendentemente, es la órbita de 0 la cual determina si un conjunto de Julia es o no conexo: si la órbita de 0 tiende a infinito entonces el conjunto de Julia es un conjunto no conexo, y si la órbita de 0 no tiende al infinito, entonces es un conjunto conexo. Por tanto, si tomamos valores de c dentro del conjunto de Mandelbrot, tendremos un conjunto de Julia conexo y, en caso contrario, el conjunto será totalmente desconexo.

Sistemas Dinámicos

La teoría de sistemas dinámicos estudia aquellos modelos que evolucionan en el tiempo y que pueden ser descritos o bien mediante funciones o mapas (sistemas discretos), o bien mediante ecuaciones diferenciales (sistemas continuos). Por ejemplo, las conjunto de Mandelbrot y Julia anteriores resultan del sistema dinámico discreto dado por el mapa $z^2 + c$.

Uno de los sistemas dinámicos más populares y simples es *el mapa logístico*, dado por el mapa cuadrático $x_{n+1} = f(r, x_n) = r x_n (1 - x_n)$, donde n juega el papel del tiempo. Este mapa está definido en el intervalo unidad $(0,1)$, para valores del parámetro $r \in [0,4]$, y en

el intervalo $(-\frac{1}{2}, \frac{3}{2})$ para $r \in [-2,0]$. En Mathematica este mapa puede definirse como una función, que depende del parámetro r

```
Logistic[r_]:=Function[x, r x (1-x)];
```

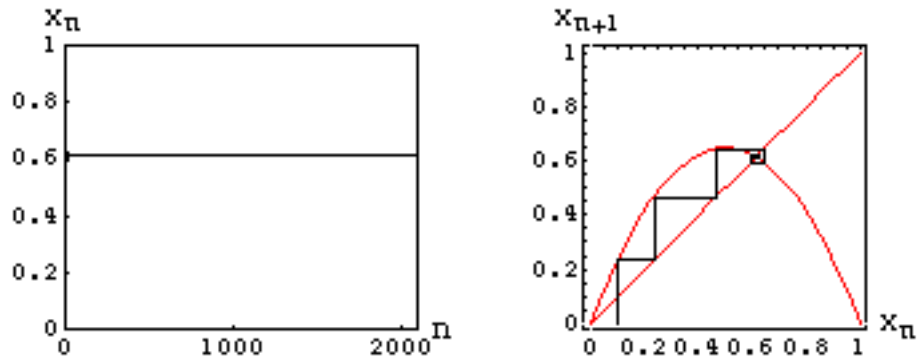
Este sistema fue propuesto originalmente para simular el crecimiento de poblaciones (ver May, 1976). Dada una población inicial x_0 , la población del año siguiente se obtiene mediante $x_1 = f(r, x_0)$ donde el parámetro r es la tasa de crecimiento de la población. Repitiendo este proceso iterativo se obtienen sucesivas poblaciones correspondientes a años consecutivos. Como ya mencionamos anteriormente, la secuencia obtenida se denomina la *órbita* del punto x_0 . Las herramientas de programación funcional de *Mathematica* permiten definir de forma sencilla diversos algoritmos iterativos para el cálculo y tratamiento de órbitas asociadas a un mapa arbitrario. Por ejemplo, el comando `Orbit[map, x0, n]` utiliza la función `NestList` de Mathematica para obtener una órbita de n puntos asociada con la condición inicial x_0 . Por otra parte, el comando `IterativeProcess[map, x0, {min,max}]` ilustra el proceso iterativo obtenido $(x_n, x_{n+1}) \rightarrow (x_{n+1}, x_{n+1}) \rightarrow (x_{n+1}, x_{n+2})$ en el espacio de retrasos correspondiente (x_n, x_{n+1}) .

```
Orbit[map_,x0_,n_]:=NestList[map, x0, n];
```

```
IterativeProcess[map_,x0_,{min_,max_}] :=  
Module[{fr,orb},  
  orb=Orbit[map,x0,50];  
  fr=MapThread[Line[{{#1,#1},{#1,#2},{#2,#2}}]&,  
    {Drop[orb,-1],Drop[orb,1]}];  
  Show[Plot[{map[x],x},{x,min,max}],Graphics[{fr}]]  
]
```

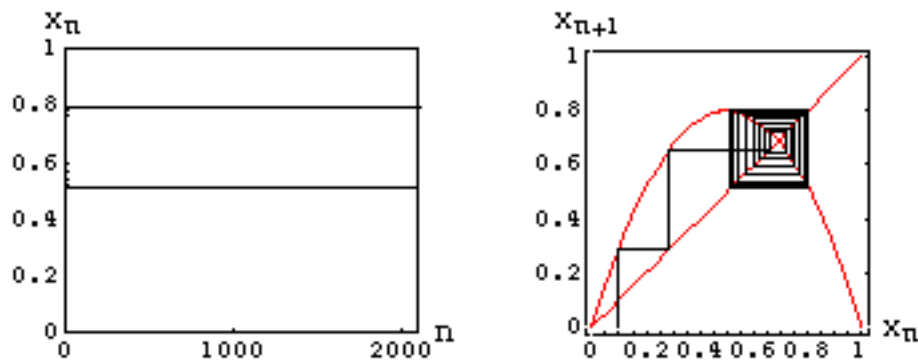
Con la ayuda de este comando podemos analizar los distintos comportamientos de las órbitas, dependiendo del valor de la tasa de crecimiento. Para ciertos valores de este parámetro, la población converge asintóticamente a un cierto valor. Este valor se denomina un punto fijo del sistema:

```
Show[GraphicsArray[  
  {ListPlot[Orbit[Logistic[2.6],0.1,2100]],  
  IterativeProcess[Logistic[2.6],0.1,{0,1}]}]]
```



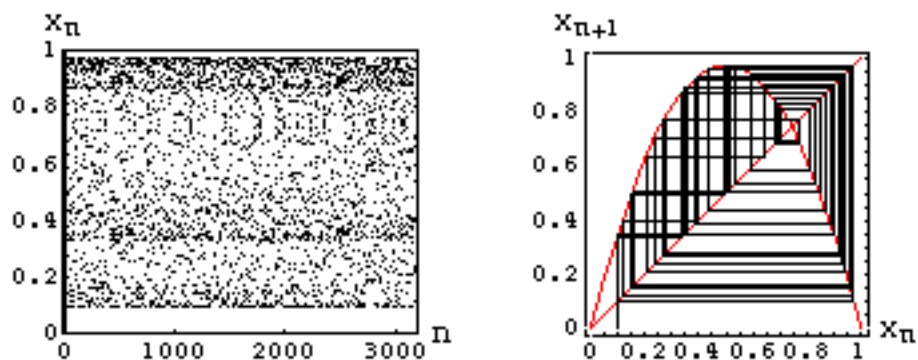
Cuando se incrementa el valor del parámetro, el sistema oscila entre dos puntos distinto. Esta órbita se denomina *periódica de período 2*:

```
Show[GraphicsArray[
  {ListPlot[Orbit[Logistic[3.2],0.1,2100]],
  IterativeProcess[Logistic[3.2],0.1,{0,1}]}]]
```



Si se sigue incrementando el valor del parámetro, se llega a una zona donde las órbitas evolucionan de forma aparentemente aleatoria sobre un conjunto infinito de puntos. Este comportamiento se conoce como *caos determinista*.

```
Show[GraphicsArray[
  {ListPlot[Orbit[Logistic[3.9],0.1,2100]],
  IterativeProcess[Logistic[3.9],0.1,{0,1}]}]]
```

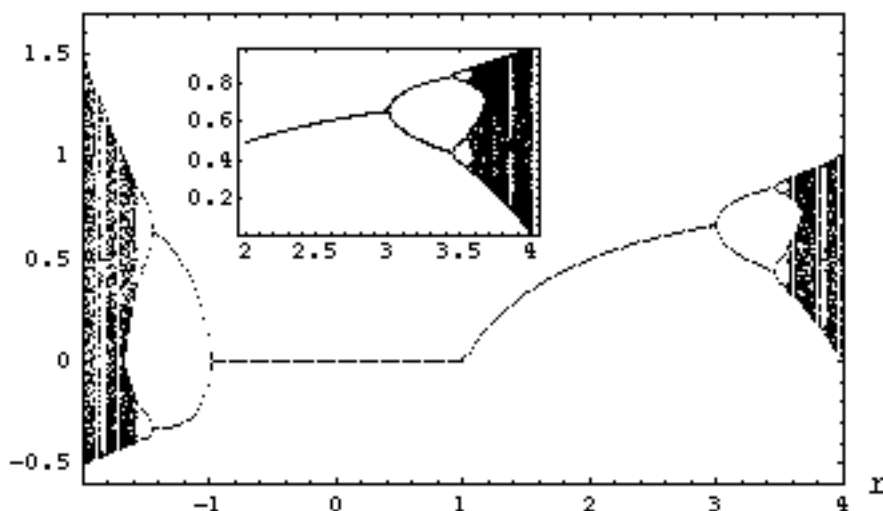


Los distintos comportamientos del sistema para diferentes valores del parámetro se pueden observar cualitativamente en el llamado *diagrama de bifurcación*, que se obtiene representando en un eje (eje y) las órbitas asintóticas del mapa (para una cierta condición inicial), para distintos valores del parámetro (eje x). El comando `Bifurcation[map, {p, pi, pf, np}, n]` dibuja el diagrama de bifurcación del *map* considerando órbitas de *n*-puntos y rechazando el comportamiento transitorio inicial (para obtener el comportamiento asintótico). Este comando considera *np* valores distintos de los parámetros en la región $p = pi$ a pf .

```
Bifurcation[map_, {p_, pi_, pf_, np_}, n_]:=
Module[{pp},ListPlot[ Flatten[
  Table[({pp,#})& /@ Drop[Orbit[map /. p->pp][0.5,n+50],50],
    {pp,pi,pf,(pf-pi)/np}],1], Axes->False,Frame->True,
  PlotStyle->{PointSize[0.003]}]]
```

Por ejemplo, la siguiente figura muestra el diagrama de bifurcación completo del mapa logístico, considerando valores del parámetro de crecimiento en el intervalo $(-2,4)$. El mapa logístico se define normalmente para valores del parámetro en el intervalo $(0,4)$, pues la población está normalizada en ese intervalo y el parámetro toma valores positivos. Una visión detallada de esta zona se muestra en la siguiente figura.

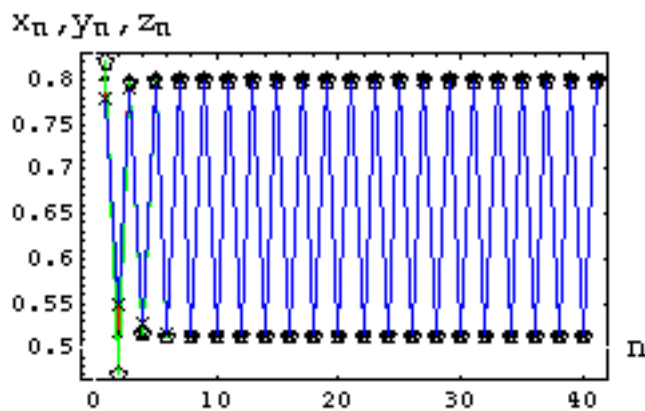
```
Show[Graphics[{Rectangle[{0,0},{1,1},
  Bifurcation[Logistic[r],{r,-2,4,250},0.5,200,1],
  Rectangle[{0.3,0.5},{0.7,1},
  Bifurcation[Logistic[r],{r,0,4,100},0.5,200]}]]]
```



El diagrama de bifurcación ayuda a entender los distintos comportamientos del sistema. A medida que se incrementa el valor del parámetro r el sistema sigue una secuencia de comportamientos periódicos: 1, 2, 4, ..., que se denomina *ruta al caos por doblamiento de períodos*; esta ruta tiene propiedades universales para una amplia gama de mapas (más detalles en Schuster, 1989). Cuando se incrementa el valor del parámetro r por encima de un valor crítico $r_c=3.569\dots$, el sistema se vuelve impredecible y presenta caos determinista. Por tanto, en el mapa logístico el caos aparece como consecuencia de la acumulación de un número infinito de órbitas periódicas inestables que resultan de la ruta de bifurcación de períodos. Es decir, la órbita impredecible descrita por un sistema caótico evoluciona en un espacio lleno de órbitas periódicas inestables (UPOs) que influyen la dinámica del sistema. Esta interesante propiedad de los sistemas caóticos se denomina *complejidad de órbitas* y se utiliza, por ejemplo, para controlar la evolución de un sistema caótico atrayéndolo hacia una de las órbitas inestables (Ott, Grebogi, y Yorke, 1990)

Como ya se ha mencionado, los sistemas de comportamiento regular y los caóticos se diferencian en los efectos que pequeños cambios en las condiciones iniciales producen en las órbitas; es decir, se diferencian en su sensibilidad frente a perturbaciones externas. En la siguiente figura puede verse como una órbita periódica es insensible a perturbaciones (incluso grandes) introducidas en las condiciones iniciales. Sin embargo, un sistema caótico es muy sensible incluso a pequeños cambios.

```
MultipleListPlot[Orbit[Logistic[3.2],#,40]& /@
  {0.8, 0.8+0.02, 0.8-0.02}]
MultipleListPlot[Orbit[Logistic[3.9],#,40]& /@
  {0.8, 0.8+10^(-8), 0.8-10^(-8)}]
```



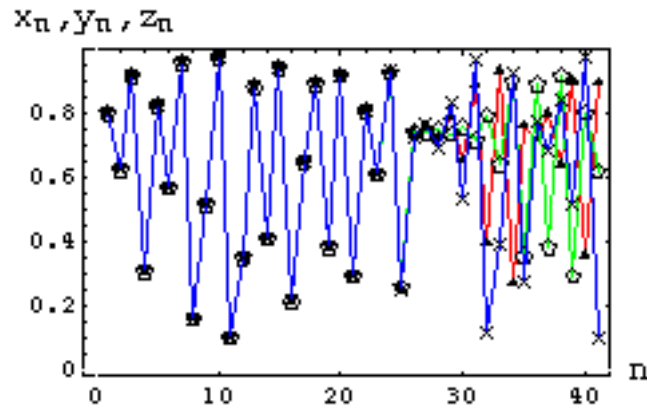


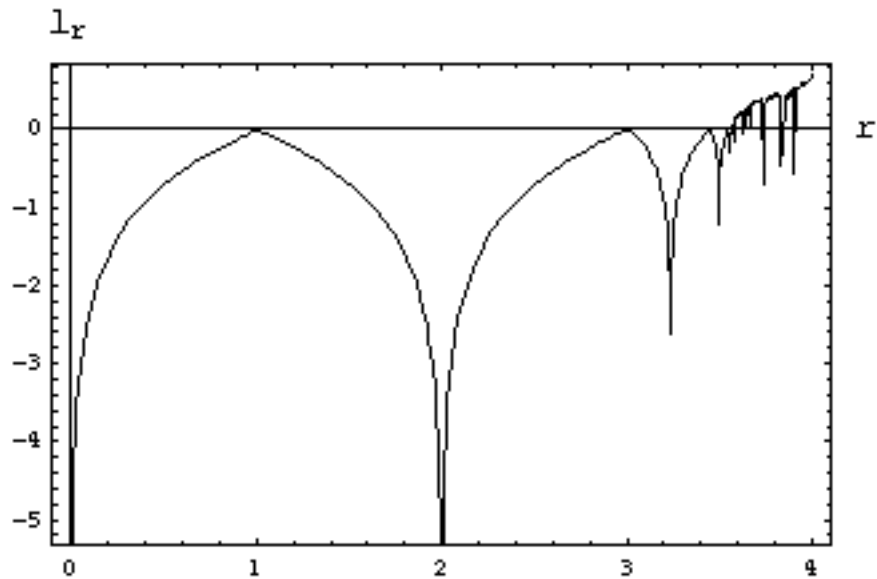
Figura. Tres órbitas distintas x_n , y_n , y z_n correspondientes a un mapa logístico periódico con condiciones iniciales 0.8, 0.8+0.02, y 0.8-0.02 (figura superior) y un mapa caótico con condiciones iniciales 0.8, 0.8+ 10^{-8} , y 0.8- 10^{-8} (figura inferior).

Los *exponentes de Lyapunov* son una medida cualitativa de la sensibilidad de un sistema a los cambios en las condiciones iniciales. Estos exponentes miden la separación exponencial de órbitas cercanas de un sistema. Por tanto, si el exponente es negativo las órbitas se acercan (comportamiento regular), mientras que si es positivo las órbitas se alejan exponencialmente (comportamiento caótico). El comando `LyapunovExp[map, x0, n]` permite calcular el exponente de Lyapunov de un mapa 1D a partir de una órbita de n puntos del mismo. En este caso, el exponente se obtiene de forma simple promediando los logaritmos de las derivadas del mapa a lo largo de la órbita (ver Schuster, 1989). En general, el cálculo de los exponentes de Lyapunov de un mapa n -dimensional se calcula de forma análoga a partir de los valores propios de la matriz jacobiana del mapa a lo largo de la órbita.

```
LyapunovExp[map_,x0_,n_]:=Plus @@ (
  Function[x,Evaluate[Log[Abs[D[map[x],x]]]]] /@
  Drop[Orbit[map, x0, n+500],500])/n
```

Por ejemplo, la siguiente figura muestra el diagrama de exponentes de Lyapunov del mapa logístico para valores del parámetro en el intervalo (0,4). Si comparamos este diagrama con el de bifurcación se puede ver que los regímenes periódicos están asociados a valores negativos del exponente, mientras que regímenes caóticos tienen exponentes positivos.

```
Plot[Lyapunov[Logistic[r],0.5,500],{r,0,4}]
```



La rica estructura del diagrama de bifurcación y, por tanto, del sistema es una consecuencia de los cambios de estabilidad de las distintas órbitas periódicas del sistema en función del parámetro r . Una órbita periódica de período uno (o punto fijo) x_f se dice *estable* si atrae órbitas cercanas, es decir,

$$\left| \left[\partial_x f(r, x) \right]_{x_f} \right| < 1,$$

y se dice *inestable* en caso contrario.

Los comandos `PeriodicPoints[map,x,n]` y `Stability[map,x,r,fp,n]` permiten analizar la estructura analítica del diagrama de bifurcación de forma sencilla:

```
FixedPoints[map_,x_,n_]:= Simplify[Solve[Nest[map,x,n]==x,x]];

PeriodicPoints[map_,x_,n_]:=
  x/. Complement[FixedPoints[map,x,n],FixedPoints[map,x,n-1]];

Stability[map_,x_,r_,fp_,n_]:=
Module[{equ,s1,s2},
  equ=D[Nest[map,x,n],x] /. x->fp;
  {s1,s2}=Solve[equ==#, r]& /@ {-1,1};
  If[s1!={} && s2!={}, Sort /@
    MapThread[List, {Flatten[r/. s1],Flatten[r/. s2]},1,{}]]
]
```

Por ejemplo, los puntos fijos del mapa son

```
p1=PeriodicPoints[Logistic[r],x,1]
```

$$\left\{0, \frac{r-1}{r}\right\}$$

Como se muestra a continuación, estos puntos fijos permanecen estables en las regiones (-1,1) y (1,3), respectivamente.

```
Stability[Logistic[r],r,#,x,1]& /@ p1
```

```
{{{-1,1}},{{1,3}}}
```

Por tanto, si $-1 < r < 1$, el punto fijo $\frac{r-1}{r}$ es inestable, mientras que el cero es estable.

Esto implica que todas las trayectorias del sistema convergerán asintóticamente a cero. Sin embargo, cuando se incrementa el valor del parámetro r , el sistema presenta una *bifurcación tangente* donde el cero pierde la estabilidad y el punto fijo $\frac{r-1}{r}$ se vuelve

estable. Para valores mayores del parámetro, el punto fijo $\frac{r-1}{r}$ también pierde la estabilidad y se vuelven estables dos puntos periódicos de período 2:

```
p2=PeriodicPoints[Logistic[r],x,2]
```

$$\left\{\frac{1+r-\sqrt{-3-2r+r^2}}{2r}, \frac{1+r+\sqrt{-3-2r+r^2}}{2r}\right\}$$

cuyas regiones de estabilidad son

```
Stability[Logistic[r],r,#,x,2]& /@ p2
```

```
{{{-1,1-√6},{3,1+√6}},{{-1,1-√6},{3,1+√6}}}
```

En este caso se tiene dos regiones de estabilidad para cada uno de los puntos. Para valores de $r > 1 + \sqrt{6}$ se tienen cuatro puntos periódicos de período 4, y así sucesivamente. Obsérvese que el cálculo de estos puntos periódicos involucra polinomios de grado seis, que no pueden resolverse de forma simbólica. En este caso, se puede obtener una aproximación numérica de estos puntos con el comando `NFixedPoints[map, x, n]`. Este comando permite ilustrar la propiedad de complejidad de órbitas antes

mencionada. Por ejemplo a continuación se muestran todas las órbitas periódicas inestables hasta período ocho que coexisten con el atractor caótico del mapa logístico:

NFixedPoints[Logistic[3.9],x,8]

```
{0.,0.00570386,0.00717971,0.0636502,0.0656863,0.0691803,0.0750764,
0.0917974,0.0919389,0.0974435,0.100562,0.104305,0.111837,0.114523,
0.121947,0.124823,0.132653,0.156816,0.180986,0.213536,0.237367,
0.289332,0.301209,0.358974,0.358974,0.385122,0.388375,0.413084,
0.43037,0.448718,0.465846,0.467459,0.476801,0.578097,0.619508,
0.697677,0.71665,0.74359,0.74359,0.74359,0.74359,0.75911,0.801914,
0.803777,0.827517,0.848259,0.897436,0.897436,0.9193,0.951213,0.964744}
```

El fenómeno de creación y destrucción de puntos fijos como transición al caos no es exclusivo de este tipo de mapas, sino que también está presente en mapas de dimensión mayor e incluso se ha observado en numerosos sistemas continuos (gobernados por ecuaciones diferenciales). Consideremos ahora el mapa de Hénon (Hénon, 1976):

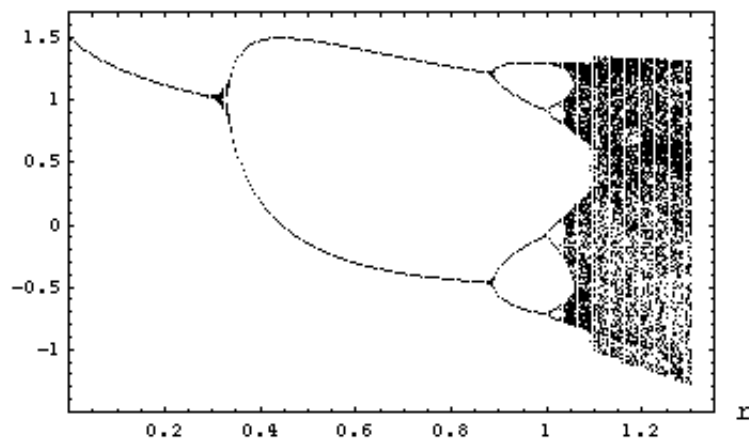
$$(x_{n+1}, y_{n+1}) = f(r, x_n, y_n) = (1 - r x_n^2 + y_n, \frac{1}{3} x_n)$$

Henon[r_] :=

Function[x, {1-r x[[1]]^2+x[[2]], 1/3 x[[1]]}]

Igual que en el caso anterior, dependiendo de los valores del parámetro r , el sistema presenta distintos comportamientos asociados con dinámicas caóticas (transient chaos, interior crisis, etc.). Por ejemplo, para $r = 1.282$ el sistema es caótico. El diagrama de bifurcación de la variable x se muestra en la siguiente figura:

Bifurcation[Henon[r], {r, 0, 1.3, 250}, {0.5, 0.5}, 150]



En este caso también podemos obtener los puntos periódicos que forman la ruta de bifurcación al caos y estudiar su estabilidad. Por ejemplo, los puntos de períodos uno y dos del mapa de Hénon se pueden obtener mediante:

```
p1=PeriodicPoints[Henon[r],{x,y},1]
```

$$\left\{ \left\{ \frac{1+\sqrt{1+9r}}{3r}, -\frac{1+\sqrt{1+9r}}{9r} \right\}, \left\{ \frac{-2+\sqrt{4+36r}}{6r}, \frac{-2+\sqrt{4+36r}}{18r} \right\} \right\}$$

```
p2=PeriodicPoints[Henon[r],{x,y},2]
```

$$\left\{ \left\{ \frac{1+\sqrt{-3+9r}}{3r}, -\frac{-1+\sqrt{-3+9r}}{9r} \right\}, \left\{ \frac{-1+\sqrt{-3+9r}}{3r}, \frac{1+\sqrt{-3+9r}}{9r} \right\} \right\}$$

En este caso, la estabilidad depende de los autovalores de la matriz jacobiana correspondiente:

```
ev=Eigenvalues[D[Henon[r]][{x,y}],#]& /@{x,y}
```

$$\left\{ \frac{1}{6} \left(-6rx - \sqrt{12+36r^2x^2} \right), \frac{1}{6} \left(-6rx + \sqrt{12+36r^2x^2} \right) \right\}$$

El siguiente comando muestra que los puntos periódicos anteriores son inestables pues al menos uno de los valores propios es positivo (es decir, las órbitas se separan exponencialmente en al menos una dirección). Los puntos periódicos con un autovalor positivo y otro negativo tienen una dirección principal que atrae órbitas cercanas y otra que las rechaza. Estos puntos, denominados *saddle nodes*, juegan un papel muy importante en la dinámica del sistema.

```
(ev // . {x->#[[1]],r->1.282})& /@ Union[p1,p2]
```

```
{{-0.2218,1.503},{-2.736,0.1218},{-0.1064,3.134},{-1.872,0.1781}}
```

La siguiente figura muestra el atractor extraño (la órbita caótica asintótica) del mapa de Hénon para $r = 1.282$ y $(x_0, y_0) = (0.5, 0.5)$. Esta figura también muestra los puntos periódicos anteriores.

```
MultipleListPlot[Orbit[Henon[1.282],{0.5,0.2},2000],
  p1/.r->1.282,p2/.r->1.282]
```

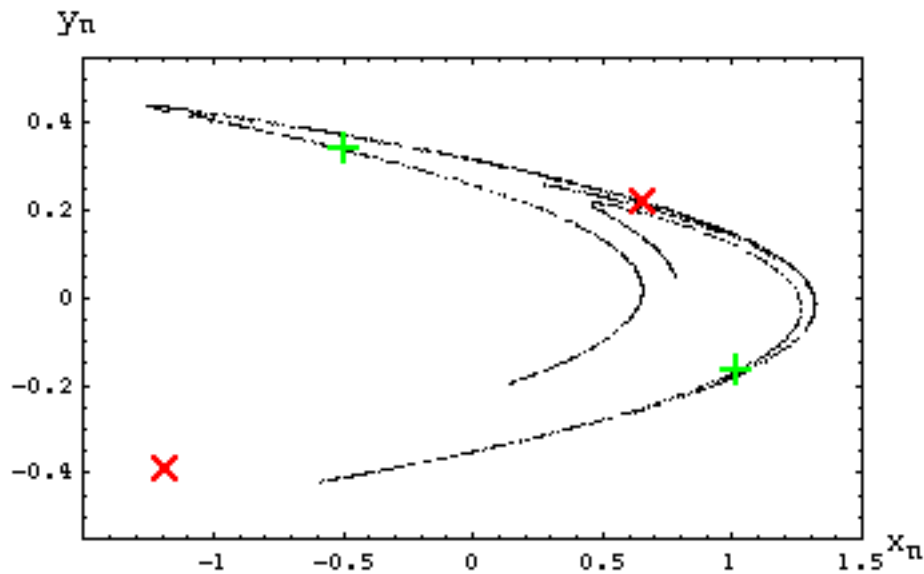


Figura. Atractor caótico del mapa de Hénon con dos puntos periódicos de periodo 1 y dos de período dos etiquetados como x y +, respectivamente.

El caos determinista también está presente en sistemas continuos, descritos por ecuaciones diferenciales. Uno de los sistemas continuos no lineales más famosos es el oscilador de Duffing, que considera términos de rozamiento y una fuerza externa. Este sistema está definido por el siguiente sistema de ecuaciones de segundo orden (ver Moon, 1992):

$$x'' + ax' + x^3 - x = f \cos(\omega t)$$

o, de forma equivalente, por el siguiente sistema de tres ecuaciones de primer orden

$$\begin{cases} x' = v \\ v' = -av - x^3 + x + f \cos(z) \\ z' = \omega \end{cases}$$

donde $v = x'$, $z = \omega t$, a es la constante de rozamiento, y f y ω son la amplitud y la frecuencia de la fuerza externa. En lo siguiente se mantienen constantes la constante de rozamiento y la frecuencia externa ($a = 0.5$ y $\omega = 1$) y se considera la amplitud de la fuerza externa f como el parámetro del modelo.

```
Duffing[f_,w_] := {v, -1/2*v - x^3 + x + f*Cos[z], w};
```

Los puntos fijos del sistema son aquellos en los que se anula el vector jacobiano (x', v', z') y pueden calcularse mediante `FixedPointsFlow[flow, vars]`. En el caso del oscilador de Duffing, en ausencia de fuerza externa ($f = 0$), el sistema tiene tres puntos fijos en $x = -1, 0$, y 1 .

```
fp=FixedPointsFlow[Duffing[0,w], {x,v,z}]
```

```
{{v->0,x->-1},{v->0,x->0},{v->0,x->1}}
```

La estabilidad de estos puntos viene dada por los autovalores de matriz jacobiana:

```
ev=Eigenvalues[D[Duffing[0,w],#]& /@{x,v,z}]
```

$$\left\{ 0, \frac{1}{4} \left(-1 - \sqrt{17 - 48x^2} \right), \frac{1}{4} \left(-1 + \sqrt{17 - 48x^2} \right) \right\}$$

Los tres autovalores correspondientes a los puntos anteriores son:

```
Re[ev /. fp]
```

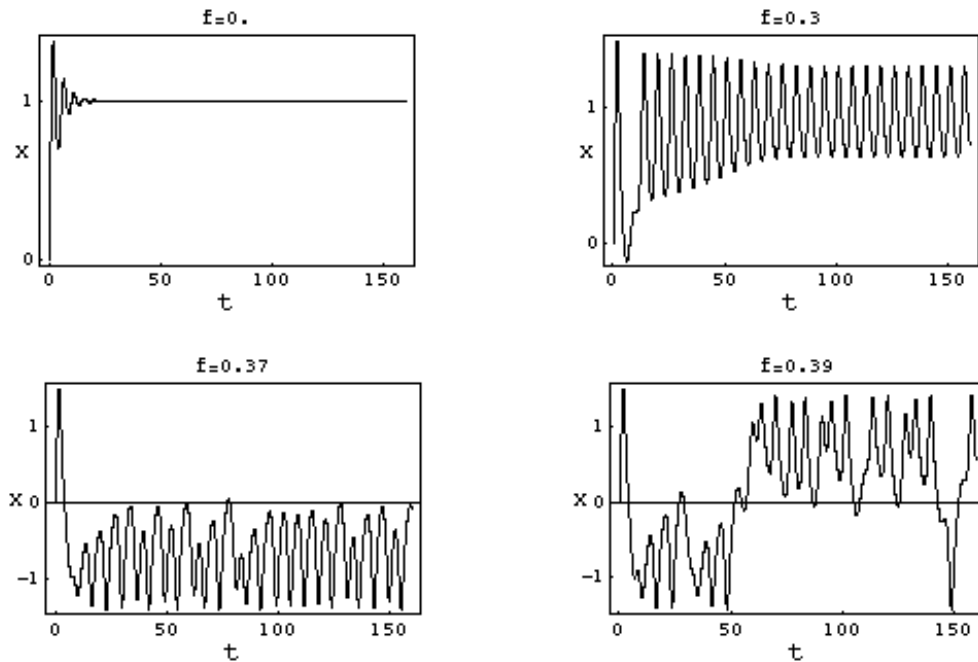
$$\left\{ \left\{ 0, -\frac{1}{4}, -\frac{1}{4} \right\}, \left\{ 0, \frac{1}{4}(-1 - \sqrt{17}), \frac{1}{4}(-1 + \sqrt{17}) \right\}, \left\{ 0, -\frac{1}{4}, -\frac{1}{4} \right\} \right\}$$

Por tanto, $x = -1$ y $x = 1$ son dos puntos fijos estable (la parte real de todos sus autovalores es negativa) y $x = 0$ es un punto fijo inestable. La figura inferior muestra bajo la etiqueta "f=0" una órbita atrapada en el punto fijo $x = 1$.

Cuando se aplica una fuerza externa al sistema, éste oscila alrededor de uno de los puntos fijos estables (figura inferior "f=0.3"). Para valores del parámetro $f > 0.321$ el sistema recorre una ruta al caos por doblamiento de períodos que tiene su punto de acumulación en $f_c = 0.3586$. Por tanto, valores mayores del parámetro dan lugar a dinámicas caóticas. Para $f_c < f < 0.386$, las órbitas caóticas permanecen atrapadas oscilando en torno a un sólo punto. Por ejemplo la figura inferior (f=0.37) muestra una órbita caótica oscilando alrededor del punto fijo $x = -1$. Cuando se incrementa el valor del parámetro la órbita caótica salta de un lado a otro del espacio, oscilando entre los dos puntos fijos ("f=0.39").

El comando `OrbitFlow[flow, x, x0, {t0,t1,dt}]` implementa un algoritmo de Runge-Kutta de cuarto orden para resolver un sistema de ecuaciones dado. Por ejemplo, permite obtener las siguientes órbitas del oscilador de Duffing.

```
Show[GraphicsArray[Show[
  OrbitFlow[Duffing[#,1],{x,v,z},{0.,1.,0.},{0,160,0.05},
  Show->Plot]& /@{0.,0.3,0.37,0.39}]
]]
```



Bibliografía

- Barnsley, M.F. (1990) *Fractals Everywhere*. Academic Press, New York.
- Falconer, K. (1993) *Fractal Geometry*. John Wiley and Sons, Chichester.
- Hénon M. (1976) "A two dimensional mapping with a strange attractor", *Communications in Mathematical Physics*, 50, 69-77.
- May, R. (1976) "Simple mathematical models with very complicated dynamics", *Nature*, 261, 459-467.
- Moon, F.C. (1992) *Chaotic and Fractal Dynamics*. John Wiley.
- Ott, E., Grebogi, C., and Yorke, J.A. (1990) "Controlling chaos", *Physical Review Letters*, 64, 1196-1199.
- Schuster, H.G. (1989) *Deterministic Chaos*. 2nd ed. VCH Publishers (in special, Chapter 3).
- Wolfram, S. (1996) *The Mathematica 3.0 Book*. Wolfram Media/Cambridge University Press. (1991) *Mathematica: A System for Doing Mathematics by Computers*, Addison Wesley.