

RESUMEN

Experimentar con sistemas dinámicos a tiempo discreto, puede representar un recurso didáctico importante en la investigación de las propiedades de los sistemas dinámicos y de sus posibles aplicaciones a disciplinas como la economía. Como una ilustración de esta posibilidad didáctica, en este documento se presenta una breve introducción a la actividad de los sistemas dinámicos a tiempo discreto mediante ejemplos asistidos por el lenguaje simbólico *Mathematica*. Dichos sistemas son esencialmente mapas iterados. En una primera parte, construimos órbitas de puntos bajo iteración de funciones reales y complejas. Si x es un número real o un número complejo, entonces la órbita de x bajo f es la sucesión $\{x, f(x), f(f(x)), \dots\}$. Estas sucesiones pueden ser convergentes o sucesiones que tienden a infinito. En particular, para probar este comportamiento en sucesiones complejas, será necesario el concepto de derivada de una función compleja. En una segunda parte, utilizamos los conceptos revisados en la primera para construir conjuntos Julia, éstos se adquieren de asignar colores a los puntos en una malla rectangular de acuerdo al comportamiento de sus órbitas bajo la función compleja estudiada, los colores se asignan de acuerdo a la clasificación de los puntos. El dibujo obtenido, el conjunto Julia, es un fractal. No obstante, la imagen que se logra será siempre una aproximación.

Palabras clave: iteración, sistema dinámico, Mathematica, punto fijo, órbita, conjunto Julia.

ABSTRACT

Experiment with discrete time dynamical systems, may represent an important educational resource in the investigation of the properties of dynamical systems and their potential applications to disciplines such as economics. As an illustration of this possibility teaching, this paper provides a brief introduction to the dynamics of discrete-time dynamic systems using examples assisted by the symbolic language *Mathematica*. Such systems are essentially iterated maps. In the first part, we construct orbits of points under iteration of real and complex functions. If x is a real number or a complex number, then the orbit of x under f is the sequence $\{x, f(x), f(f(x)), \dots\}$. These sequences may be convergent or sequences that tend to infinity. In particular, to test this behavior in complex sequences will require the concept of derivative of a complex function.

In a second part, we use the concepts reviewed in the first to build Julia sets, these sets are obtained by assigning colors to a rectangular grid points according to the behavior of their orbits under the studied complex function, the colors are assigned according to the classification of the points. The pattern obtained, the Julia set is a fractal. However, the image obtained is always an approximation.

Key words: iteration, dynamic system, Mathematica, fixed point orbits, Julia set.

Una aproximación experimental a los sistemas dinámicos discretos con Mathematica

An experimental approach to discrete dynamical systems with Mathematica

*Michael Rojas Romero*¹

- 1 Nacionalidad: Mexicana. Grado: Maestro en Ciencias. Especialización: Economía matemática y econometría. Adscripción: Universidad Nacional Autónoma de México. Facultad de Ciencias. Facultad de Economía. Correo electrónico: mirojasmx@yahoo.com.mx.

Fecha de recepción: 5 de febrero de 2013

Fecha de aceptación: 1ro. de octubre de 2013

Introducción

Es bien conocido que los modelos no lineales aplicados al estudio de fenómenos económicos se pueden formular como modelos dinámicos a tiempo continuo o a tiempo discreto (ver Gabisch y Lorenz (1989); Guckenheimer y Holmes (1983)). El tiempo en economía, sin embargo, es a menudo discontinuo (discreto) porque las decisiones en economía no pueden ser continuamente revisadas. Es por esta razón que los sistemas dinámicos a tiempo discreto, representados por ecuaciones en diferencias, o más correctamente, por la aplicación iterada de mapas, a menudo son un instrumento más adecuado para modelar procesos económicos dinámicos. Entonces, es útil analizar las peculiaridades de sistemas dinámicos a tiempo discreto dada su importancia particular en el análisis de procesos económicos dinámicos.

De hecho, en la actualidad muchos resultados se han obtenido acerca de sistemas dinámicos discretos (ver por ejemplo Devaney (1987), Guckenheimer y Holmes (1983), Gumowski y Mira (1980), looss (1979), looss y Joseph (1980), Wiggins (1988)). Por ejemplo, las propiedades dinámicas y bifurcaciones de un mapa iterado unidimensional son ahora bien conocidas, así como sus implicaciones sobre los comportamientos periódicos y caóticos de sus trayectorias (ver por ejemplo Sharkovsky, Kolyada, Sivak y Fedorenko (1997); de Melo y van Strien (1991)). El objetivo principal de este trabajo es analizar sistemas dinámicos a tiempo discreto, construyendo órbitas de puntos bajo iteración de funciones reales y complejas, además de conjuntos Julia. Se hace utilizando una aproximación experimental con el lenguaje simbólico *Mathematica*.

Para sus desarrolladores Wolfram Research Inc, *Mathematica* es un sistema para “hacer matemática por computadora”. Está diseñado para tres plataformas básicas: DOS, Windows y UNIX. La estructura básica de *Mathematica* se compone de tres partes: el Kernel, el Front End y Library. El Kernel es idéntico para las tres plataformas y el Front End distinto. El Kernel hace el trabajo computacional. El Front End acepta inputs proporciona outputs y generalmente organiza la información en una sesión de *Mathematica*. El Front End y el Kernel se comunican

mediante un protocolo de comunicación llamado *MathLink*. Hay tres tipos básicos de Front End: Microsoft Windows, Makintosh y UNIX. Estos Front Ends usan notebooks *Mathematica* como la interface entre el usuario y el Kernel. Los notebooks *Mathematica* son similares a hojas de cálculo electrónicas que permiten integrar en *Mathematica* instrucciones de entrada, salidas del kernel y texto. Lo que puede hacer *Mathematica* depende de la versión utilizada del programa. En este trabajo usamos la versión 8 y la plataforma Windows.

Mathematica ha mostrado ser un sistema útil en la investigación experimental de sistemas dinámicos discretos lineales y no lineales. Esta aproximación experimental puede representar un recurso didáctico importante en la investigación de los sistemas dinámicos.

1. Gráfica de la órbita de un punto

Dados una función f y un punto inicial x_0 , el efecto de aplicar sucesivamente la función f al punto x_0 , produce las iteradas de la función f en el punto x_0 , o la *órbita* de f en el punto x_0 , representada por $O(x_0)$, es decir,

$$O(x_0) = \{x_0, f(x_0), f^2(x_0), f^3(x_0), \dots, f^n(x_0), \dots\}$$

donde $f^n(x)$ representa a f compuesta con f^{n-1} , es decir, $f \circ f^{n-1}$. Este procedimiento iterativo origina un *sistema dinámico a tiempo discreto* caracterizado por aplicar una función a un punto una y otra vez.

1.1 Iteración de una función

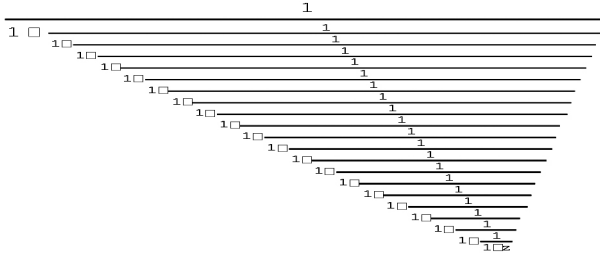
Las iteraciones de funciones se pueden hacer con el comando **Nest** de *Mathematica*.

?Nest

Nest[f, expr, n] gives an expression with f applied n times to expr.

Por ejemplo, la primera línea del siguiente código define la función $f(z) = 1/(1+z)$, la segunda itera esta función veinte veces en el punto z y la tercera línea es el output de la segunda.

```
f[z_]:=1/(1+z)
Nest[f, z, 20]
```



El valor numérico de esta iteración cuando $z = 0.2$ es

```
N[Nest[f, 0.2, 20]]
0.618034
```

1.2 Cálculo de la órbita de un punto

NestList es un comando que proporciona un segmento de la órbita de un punto bajo la acción de alguna función real. Por ejemplo, calculamos 20 iteraciones iniciando con $z=1$ (20 elementos de la órbita del punto 1).

```
NestList[f, 1, 20]
```

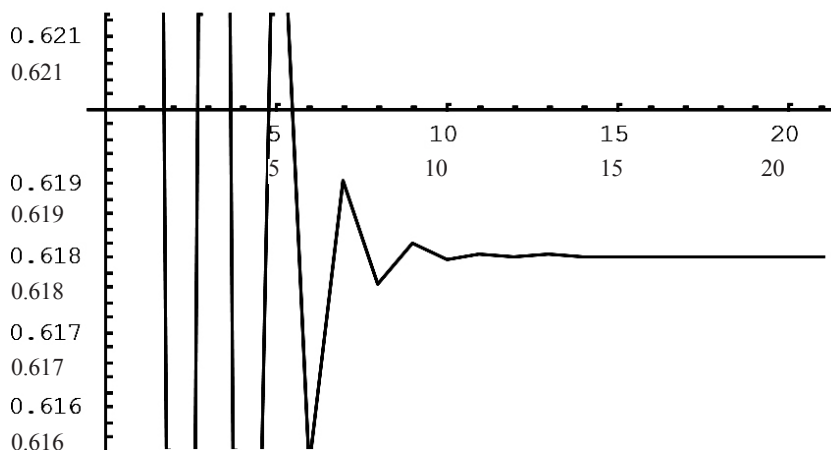
$$\begin{array}{l} \blacksquare, \frac{1}{2}, \frac{2}{3}, \frac{3}{5}, \frac{5}{8}, \frac{8}{13}, \frac{13}{21}, \frac{21}{34}, \\ \frac{34}{55}, \frac{55}{89}, \frac{89}{144}, \frac{144}{233}, \frac{233}{377}, \frac{377}{610}, \frac{610}{987}, \\ \frac{987}{1597}, \frac{1597}{2584}, \frac{2584}{4181}, \frac{4181}{6765}, \frac{6765}{10946}, \frac{10946}{17711} \end{array}$$

Su representación decimal se puede obtener de la siguiente manera:

```
NestList[f, 1.0, 20]
{1.,0.5,0.666667,0.6,0.625,0.615385,0.619048,0.617647,0.61818
2,0.617978,0.618056,0.618026,0.618037,0.618033,0.618034,0.61
8034,0.618034,0.618034,0.618034,0.618034,0.618034}
```

Como se ve, los valores de la órbita de 1 convergen aparentemente al valor 0.618034. Esta convergencia se puede también observar gráficamente usando el comando ListPlot.

```
ListPlot[NestList[f,1.0,20],PlotJoined→ True]
```



El comando FixedPointList[f,expr] da como resultado de la composición repetida de f una lista $\{z, f[z], ff[z], \dots\}$ hasta que sus términos no cambian más. Por ejemplo, la iterada del punto $z = 1.0$ para la función f es

```
FixedPointList[f, 1.0, 1000]
```

```
{1.,0.5,0.666667,0.6,0.625,0.615385,0.619048,0.617647
,0.618182,0.617978,0.618056,0.618026,0.618037,0.618
033,0.618034,0.618034,0.618034,0.618034,0.618034,0.
618034,0.618034,0.618034,0.618034,0.618034,0.61803
4,0.618034,0.618034,0.618034,0.618034,0.618034,0.61
8034,0.618034,0.618034,0.618034,0.618034,0.618034,
0.618034,0.618034,0.618034}
```

El comando `FixedPoint[f,x]` opera como el anterior, pero únicamente proporciona el valor que se hace fijo:

```
FixedPoint[f, 1.0, 1000]
0.618034
```

El mismo comando `NestList` calcula la órbita de un punto bajo la acción de alguna función compleja:

```
NestList[Sin, 1.0 + I, 5]
{1.+TM,1.29846+0.634964TM,1.16392+0.182506TM,0.933697+ 0.07
26277TM,0.805946+0.0432418TM,0.72216+0.0299512TM}
```

Es posible sustituir en el comando **NestList** la f por una función explícita. El siguiente comando proporciona los primeros seis puntos de la órbita del número complejo $1 + I$ bajo la iteración de la función $f: z \rightarrow z^2 - 1$.

```
NestList[(#^2 - 1)&, 1.0 + I, 5]
. . □ 2. □, □ 4. □ 4. □, □ 1. □ 32. □, □ 1024. □ 64. □, 1.04448 □ 106 □ 1.
```

1.3 Gráfica de la órbita de un punto

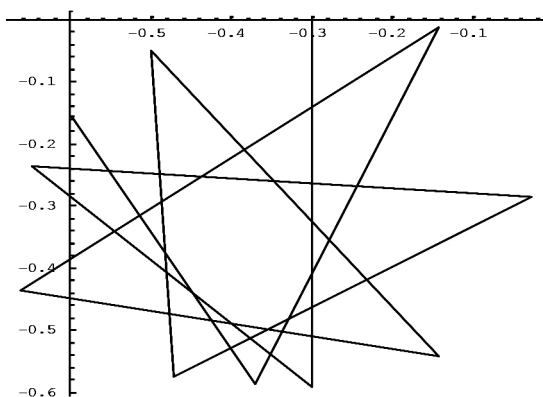
Es posible crear una sucesión de números complejos usando la función pura **(Re[#],Im[#])&**.

```
Re # | Im # | & | NestList [#^2 - 0.39 - 0.59I] &, {-0.3, 10}
```

```
{{-0.3,0},{-0.3,-0.59},{-0.6481,-0.236},{-0.0256624,-0.284097},{-0.470052,-  
0.575419},{-0.500157,-0.049046},{-0.142248,-0.540939},{-0.66238,-0.436105},-  
{-0.14144,-0.0122653},{-0.370145,-0.58653},{-0.597011,-0.155797}}
```

Aplicando el comando **ListPlot** a esta sucesión de números, se obtiene la gráfica de esta sucesión.

```
ListPlot[{Re[#],Im[#]}&/@NestList[(#^2 - 0.39 - 0.59I)&, -0.3, 10],  
PlotJoined→True, AspectRatio → Automatic, PlotRange → All]
```



...Graphics...

Esta es una órbita acotada.

Graficamos la órbita de longitud n del número complejo z bajo la iteración de la función f :

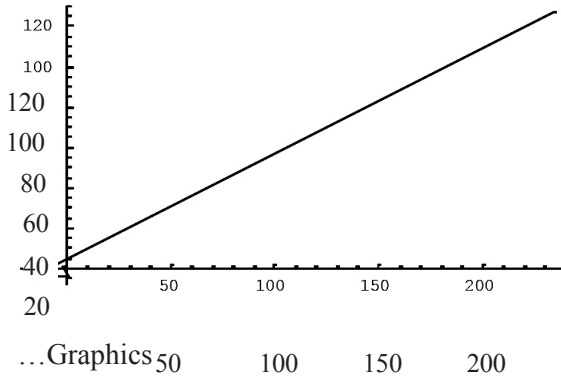

```

orbitacompleja[f_, z_, n_, t_ : Automatic] :=
ListPlot[{Re[#], Im[#]} & /@ NestList[f, z, n],
PlotJoined -> True, AspectRatio -> Automatic, PlotRange -> All, Ticks -> t];

```

La gráfica de longitud 6 del punto $-0.2 + 0.5I$ bajo la iteración de la función $f(z) = z^2 - 1 + I$ (o la función pura $(\#^2 - 1 + I) \&$) es

```
orbitacompleja[(#^2-1+I)&, -0.2 + 0.5I, 6]
```



Esta es una órbita no acotada.

1.4 Tipo de órbitas

La órbita de un punto z bajo iteración de f es acotada si existe un número real $x \in \mathfrak{R}$ tal que $|f^n(z)| \leq M$ para todo $n \in \mathbb{N}$. Si una órbita no está acotada se dice que no es acotada.

Las dos gráficas anteriores son ejemplos de órbita acotada y no acotada.

1.5 Puntos fijos y órbitas periódicas

Sea $f: X \rightarrow X, X \subseteq \mathbb{C}$. Se dice que z es un punto fijo de f si $f(z) = z$. Se dice también que ∞ es un punto fijo de f si $|f(z)| \rightarrow \infty$ cuando $|z| \rightarrow \infty$.

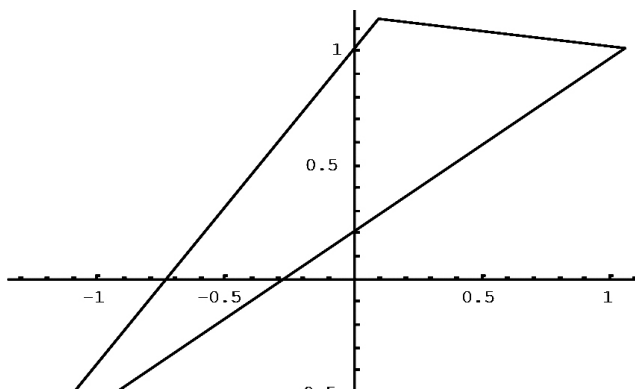
Se dice que z es un punto periódico de periodo n de f si $f^n(z) = z$. En este caso la órbita de z bajo f es $(z, f^1(z), f^2(z), \dots, f^{n-1}(z), z, f^1(z), \dots)$. Si n es el menor número entero tal que $f^n(z) = z$, entonces se dice que z tiene primer periodo n . Si z tiene periodo k , entonces es un punto fijo de f^k y $f^k(z)$ es un punto fijo de f^{k+1} , etcétera. La órbita de un punto de periodo n se llama ciclo de periodo n .

Por ejemplo si $f(z) = z^2 - 1$, los puntos -1 y 0 tienen periodo 2, como se ve en el resultado:

```
NestList[ (#^2 - 1) &, -1, 4]
{-1, 0, -1, 0, -1}
```

La siguiente gráfica muestra una órbita de periodo 3:

```
orbitacompleja[ (#^2 - 1) &, - 1.2904912332417333 - 0.7792817182359892I, 15]
```



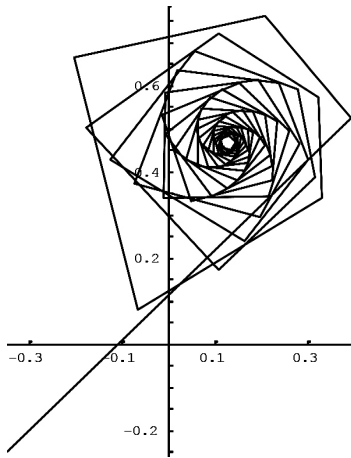
...Graphics...

1.6 Órbitas convergentes

Asumir que la función f es continua y que la órbita del punto z converge a β de modo que $f^n(z) \rightarrow \beta$ cuando $n \rightarrow \infty$.

Ya que f es continua, $f(f^n(z)) = f^{n+1}(z) \rightarrow f(z) \rightarrow f(\beta)$ cuando $n \rightarrow \infty$ de modo que $f(\beta) = \beta$. Esto significa que si una órbita de un punto bajo f es convergente, debe converger a un punto fijo de f . En la siguiente gráfica, parece que la órbita es convergente:

```
h[z_]:=z^2+0.33+0.35 I;
orbitacompleja[h,-0.35-0.25I,100,
{{-0.3,-0.1,0.1,0.3},Automatic}]
```



...Graphics...

Para verificar esta convergencia, primero encontramos los puntos fijos de la función h :

```
NSolve[h[z]==z,z]
{{z -> 0.126485 + 0.468522 I},
 {z -> 0.873515 - 0.468522 I}}
```

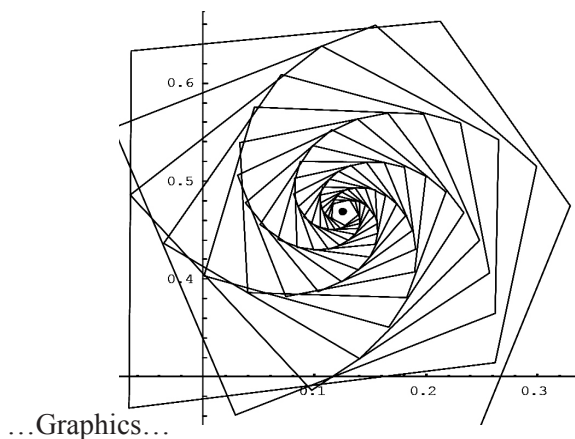
Ahora calculamos parte de la órbita del punto $-0.35 - 0.25 I$ (están sólo los últimos 10 puntos en la órbita de 0 a 350):

```
Drop[NestList[h,-0.35-0.25I,350], 340]
{0.126474 +0.468518™,0.126487+0.46851™,0.126497 +0.468521™,0.12649 +0.468533™,
0.126477+0.468529™,0.126477+0.468516™,0.126489+0.468513™,0.126495+0.468524™,
0.126487+0.468532™,0.126476+0.468526™,0.126479+0.468515™}
```

Parece que la órbita del punto $-0.35 - 0.25 I$ converge al punto fijo $\beta = 0.126485 + 0.468522 I$.

La órbita de un punto distinto bajo h converge al mismo punto fijo como se muestra en la siguiente gráfica en color gris:

```
ListPlot[{Re[#], Im[#]}&/@NestList[(#^2 + 0.33 + 0.35I)&, 0.25 + 0.25I, 100],
PlotJoined -> True, AspectRatio -> Automatic,
PlotRange -> All, Epilog -> {PointSize[0.02], Hue[0, 0, 0.5],
Point[{0.12648530502056088, 0.4685223898047904}]}]
```



Este resultado es explicado por el *teorema del mapeo de contracción para complejos*. Según este teorema, si A es un subconjunto cerrado y acotado de C y $f: A \rightarrow A$ un mapeo de contracción, entonces el teore-

ma dice que f tiene exactamente un punto fijo a en A . Además, si z es cualquier elemento de A , entonces la sucesión $f^n(z)$ converge a a .

Una condición suficiente para que una función sea un mapeo de contracción en un subconjunto de su dominio es el siguiente:

si f tiene un punto fijo en a y es analítica en una vecindad de a y si $|f'(a)| < 1$, entonces existe un disco, D , de centro a de tal manera que f es un mapeo de contracción sobre D .

Por ejemplo encontramos anteriormente que la función h , $h(z) = z^2 + 0.33 + 0.35I$, tiene un punto fijo $a = 0.126485 + 0.468522 I$. Comprobamos la derivada de h en este punto:

```
Abs[2 (0.126485 + 0.468522 I)]
0.97059
```

Entonces, hay una vecindad del punto fijo $a = 0.126485 + 0.468522 I$ tal que la órbita de todo punto en la vecindad converge a a .

1.7 Ciclos atractores y repulsores

Asumir que la función compleja f es analítica en una región A de C , y que tiene un punto fijo en $a \in A$. Entonces se dice que a es:

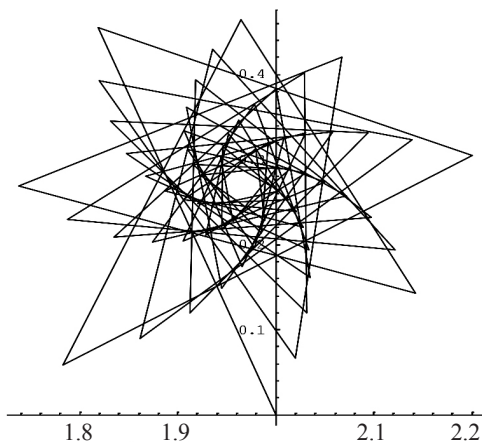
un punto fijo atractor si $|f'(a)| < 1$;
 un punto fijo repulsor si $|f'(a)| > 1$;
 un punto fijo neutral si $|f'(a)| = 1$.

Es posible demostrar que si a es un punto fijo atractor, entonces existe una vecindad D de a tal que si $b \in D$, la órbita de b converge a a . Se mostró una órbita de la función h que converge a un punto fijo atractor.

Si a es un punto periódico repulsor de f , entonces existe una vecindad N de a tal que si $b \in N$, hay puntos en la órbita de b que no están en N . (La órbita de b sale de N , aunque puede volver a N .)

La órbita de un punto bajo una función trigonométrica que converge a un punto fijo se muestra enseguida:

```
orbitacompleja[(2 + 0.5I) Sin[#] &, 2, 65]
```



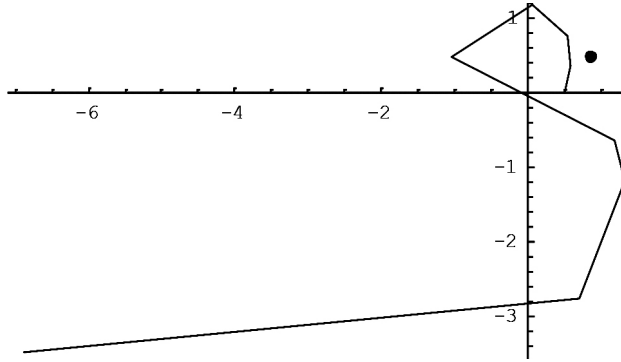
...Graphics...

Lo comprobamos hallando un punto fijo de esta función:

```
FindRoot[(2+0.5I Sin[z],{z,1.96+0.2731}]
{z -> 1.96563 + 0.268928I}
```

La órbita de un punto que está siendo repelido de una vecindad de un punto fijo repulsor de la función $h(z) = z^2 + 0.33 + 0.35I$ se ilustra enseguida:

```
ListPlot[{Re[#], Im[#]} & /@ NestList[ (#^2 + 0.33 + 0.35I) &, 1 - 0.5, 8],
PlotJoined -> True, AspectRatio -> Automatic,
PlotRange -> All, Epilog -> {PointSize[0.02],
Point[{0.873514699456611, 0.46852239083117725}]}]
```



...Graphics...

1.8 Ciclos atractores y repulsores de periodo primo mayor de uno

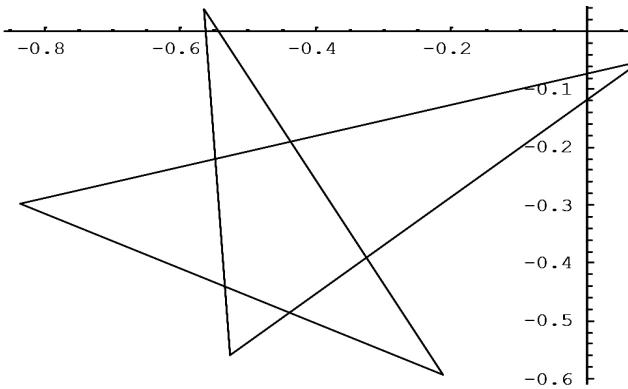
Si f es analítica en un subconjunto A de C , de modo que es f^k , y si f^k tiene un punto fijo $a \in A$, entonces este punto fijo es atractor, repulsor o neutral según si $|(f^k)'(a)|$ es menor, mayor o igual a 1. El correspondiente ciclo de periodo k , $\{a, f(a), \dots, f^{k-1}(a)\}$, se llama atractor, repulsor o neutral, respectivamente. La derivada de f^k es la misma en cada punto del ciclo anterior. La derivada de f^k en su punto fijo se puede calcular de la siguiente manera:

$$(f^k)'(a) = f'(a) f'(f(a)) \dots f'(f^{k-1}(a))$$

De esta manera, la fórmula para $(f^k)'(a)$ no involucra el cálculo de f^k , es el producto de las derivadas de f en cada punto del ciclo.

Un ciclo atractor de periodo 5 se obtiene como sigue:

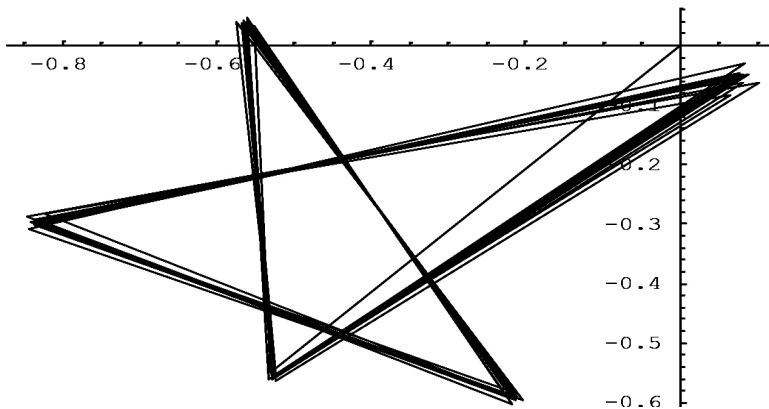
```
a = orbitacompleja[(#^2 - 0.53 - 0.55I) &,
0.07978143067846499 - 0.05170047012496698I, 10]
```



...Graphics...

La órbita de un punto atraído al ciclo anterior se obtiene de la siguiente manera:

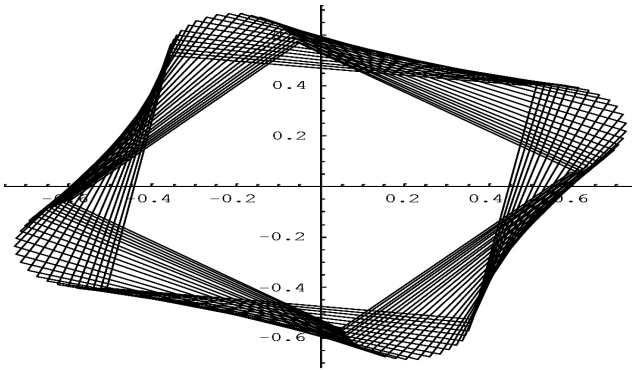
`b = orbitacompleja[(#^2 - 0.53 - 0.55I) &, 0, 100]`



...Graphics...

La órbita de un punto bajo una función trigonométrica atraído a un ciclo de fase 4 es:

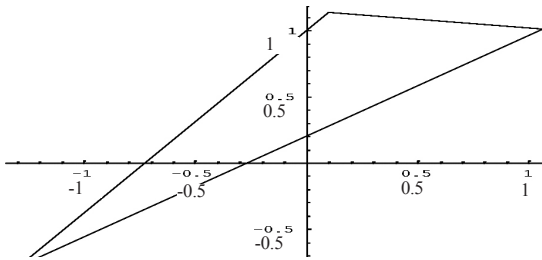
`orbitacompleja[I Sin[#] &, 0.5 + 0.4I, 100]`



...Graphics...

Un ciclo repulsor de fase 3 se tiene como sigue:

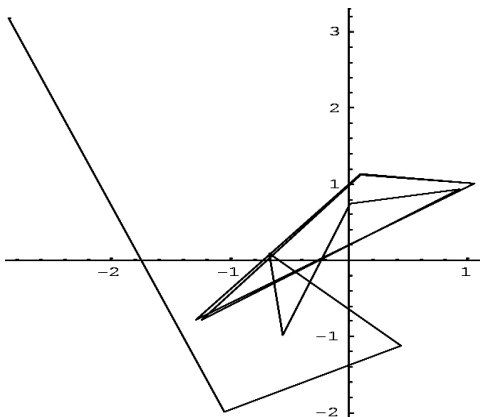
`c = orbitacompleja[(#^2 - 1) &, -1.2904912332417333 - 0.7792817182359892I, 15]`



...Graphics...

La órbita de un punto que es repelido por el ciclo de fase 3 anterior es la siguiente:

`d = orbitacompleja[(#^2 - 1) &, -1.2904 - 0.7792I, 13]`



...Graphics...

1.9 Cuencas de atracción

Sea f una función compleja con un punto fijo atractor a . La cuenca de atracción del punto fijo a bajo f se define como el conjunto $\{z \in \mathbb{C}: f^n(z) \rightarrow a \text{ cuando } n \rightarrow \infty\}$.

Como ilustración retomamos la función $h(z) = z^2 + 0.33 + 0.35I$. Encontramos los puntos fijos de h y comprobamos si son atractores:

```
h[z_]:= z^2+0.33+0.35I
```

```
Replace[z, NSolve[h[z]=z, z]]
```

```
{0.126485 + 0.468522 i, 0.873515 - 0.468522i}
```

```
Abs[D[h[z], z]]/.z%
```

```
{0.970591, 1.98246}
```

El punto $x = 0.126485 + 0.468522 I$ es un punto fijo atractor de h . Para encontrar algunos puntos en la cuenca de atracción de x , elegimos un punto “cerca” de x y calculamos parte de su órbita. Como no se conoce la extensión de la cuenca de atracción, algo de prueba y error es necesario para elegir el punto. Borrarnos los primeros 190 términos de la órbita con el comando Drop:

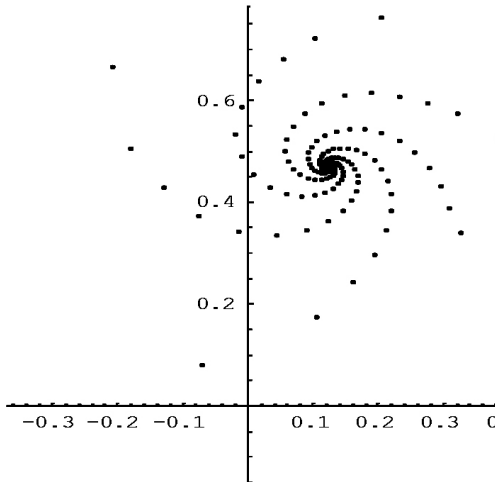
Drop[NestList[h, -0.35-0.25I, 200], 190]

```
{0.125799 + 0.469392 I, 0.125496 + 0.468098 I, 0.126633 + 0.467489 I,
0.12749 + 0.468399 I, 0.126856 + 0.469432 I, 0.125725 + 0.4691 I,
0.125752 + 0.467956 I, 0.126831 + 0.467693 I, 0.12735 + 0.468636 I,
0.126598 + 0.469361 I, 0.125727 + 0.468841 I}
```

Al parecer, el punto que elegimos está en la cuenca de atracción de x . Todos los otros puntos en su órbita también se encuentran en la cuenca de atracción de x . Puntos en la cuenca de atracción de x son:

```
orbitacompleja[f_,s_,n_]:=ListPlot[{Re[#],
Im[#]}&/@NestList[f,s,n],
AspectRatio→Automatic,PlotRange→All,
PlotStyle→PointSize[0.011]]
```

```
bl = orbitacompleja[(#^2 + 0.33 + 0.35I) &, -0.35 - 0.25I, 200]
```



...Graphics...

Los 200 puntos de la órbita anterior pertenecen a la cuenca de atracción de x . No obstante, por supuesto, hay una infinidad (no numerable) de puntos ya que la cuenca de atracción contiene un disco, x en el centro.

1.10 Cuenca de atracción de un ciclo atractor de periodo $p > 1$

Sea f una función con un ciclo atractor, $c = \{z_1, z_2, \dots, z_p\}$, de periodo $p > 1$. Entonces los puntos del ciclo son puntos fijos atractores de f^p y así cada z_k tiene una cuenca de atracción, B_k , que consiste en todos los puntos de C cuyas órbitas bajo la acción de f^p convergen a z_k . La unión de todas estas cuencas de atracción se define como la cuenca de atracción del ciclo atractor c bajo la acción de f .

Como ilustración consideramos una función con un ciclo atractor de periodo 2.

```
j[z_]:=z^2-1-0.2I;
```

Encontramos puntos fijos de j^2 y comprobamos si son atractores:

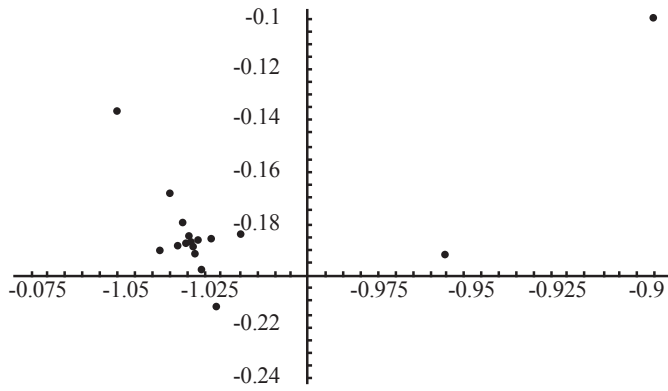
```
Replace[z, NSolve[j[j[z]] = z, z]]
```

```
{-1.03393 - 0.187291 i, -0.621583 - 0.0891597i,  
0.0339271 + 0.187291 i, 1.62158 + 0.0891597 i}
```

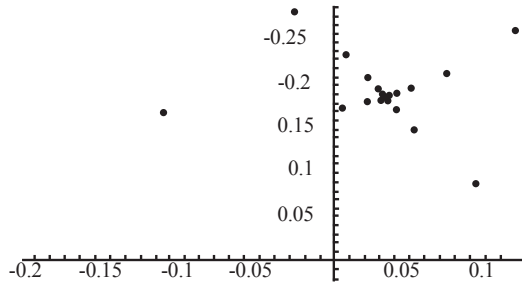
```
Abs[D[j[j[z]], z]]/.z->%
```

Se ve que j^2 tiene dos puntos fijos atractores, de modo que j tiene un ciclo atractor de periodo 2, por ejemplo $\{a_1, a_2\}$. Graficamos algunos puntos en cada una de las cuencas de atracción de estos 2 puntos fijos bajo la acción de j . Primero elegimos el punto $-0.9-0.1I$, que es 'cerca' al punto a_1 y graficamos parte de su órbita; después graficamos parte de la órbita de $j[-0.9-0.1I]$, que está 'cerca' de $a_2 (=j[a_1])$.

orbitacompleja[Composition[j,j], -0.9 - 0.1I, 100]



orbitacompleja[Composition[j,j], j[-0.9 - 0.1I], 100]



Todos los puntos en estas gráficas forman parte de la cuenca de atracción del ciclo atractor $\{a_1, a_2\}$ de j .

1.11 La cuenca de atracción de infinito

Si infinito es un punto fijo atractor de f , entonces la cuenca de atracción de infinito se define como el conjunto $\{z \in \mathbb{C}; f^n(z) \rightarrow \infty \text{ cuando } n \rightarrow \infty\}$.

Primero encontramos un punto cuya órbita tiende a infinito bajo la función $z^2 - 1$:

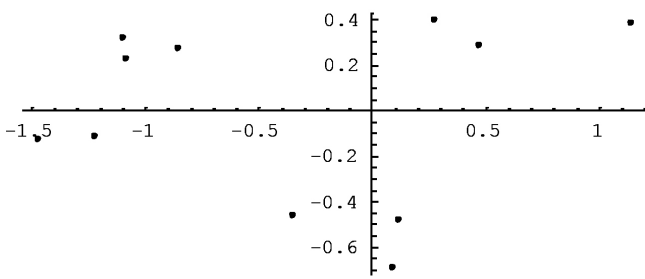
NestList[#^2 - 1 &, 0.28 + 0.4I, 19]

```

0.28 + 0.4 %q - 1.0816 + 0.224 %q 0.119683 - 0.484557 %q - 1.22047 - 0.115986 %q 0.476098 + 0.283115 %q
- 0.853485 + 0.269581 %q - 0.344237 - 0.460167 %q - 1.09325 + 0.316813 %q 0.094835 - 0.692714 %q
- 1.47086 - 0.131387 %q 1.14616 + 0.386503 %q 0.164303 + 0.885991 %q - 1.75798 + 0.291142 %q
2.00574 - 1.02364 %q 1.97516 - 4.10634 %q - 13.9608 - 16.2214 %q - 69.2296 + 452.925 %q
- 200349. - 62711.6 %q 3.6207 ¥ 1010 + 2.51284 ¥ 1010 %q 6.79506 ¥ 1020 + 1.81965 ¥ 1021 %q
    
```

Todos los puntos de esta órbita están en la cuenca de atracción de infinito. La gráfica de algunos de estos puntos es

orbitacompleja[#^2 - 1&, 0.28 + 0.4I, 10]



...Graphics...

1.12 Mapeos simétricos

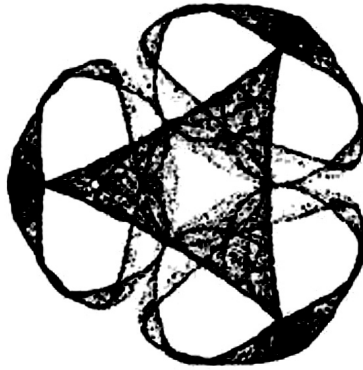
Chossat y Golubitsky (1988) definen mapeos simétricos en el plano complejo. Estos mapeos tienen la propiedad de que las órbitas de algunos puntos bajo la acción muestran un grado de simetría. Ejemplos de dichas asignaciones son dados por la familia definida por:

$$F(z) = (\lambda + \alpha |z|^2 + \beta \operatorname{Re}(z^n)) z + \gamma (z)^{n-1}$$

donde $\lambda, \alpha, \beta, \gamma$ son reales y n un natural.

Enseguida graficamos la órbita de un punto bajo la acción de un miembro de la familia F . El número de puntos a graficar debe ser muy grande y los puntos no se juntan por líneas rectas.

```
g[λ_,α_,β_,γ_,n_,z_]:=ListPlot[{Re[#], Im[#]}&/@NestList[((λ+α(Abs[#])^2+β
Re[#^n])# + γ (Conjugate[#])^n-1)&, z, 100000], AspectRatio→Automatic,
PlotRange→All, PlotStyle→{PointSize[0.0001]}, Axes→False]
```



2. Construcción de conjuntos Julia de funciones cuadráticas como atractores de sistemas de funciones iteradas no afines

Sea C el conjunto de números complejos. Sea $f: C \rightarrow C$, un polinomio en z de grado mayor que dos, donde $z \in C$. La órbita de z bajo f está acotada si existe $K > 0$ tal que $|f^n[z]| < K$ para todo $n \in \mathbb{N}$. El conjunto Julia completo de f es el conjunto:

$$K_f = \{z \in C : \text{la órbita de } z \text{ bajo } f \text{ está acotada}\}$$

El conjunto Julia de f , J_f , es el límite del conjunto Julia completo de f . Ejemplos de conjuntos Julia y Julia completos para la familia de funciones cuadráticas complejas $\{Q_c = z \rightarrow z^2 + c : c \in C\}$

2.1 Familia cuadrática

Consideremos la familia de funciones $\{Q_c = z \rightarrow z^2 + c : c \in C\}$, donde c es un parámetro. Barnsley (1988) muestra que el conjunto Julia de la familia Q_c es el atractor de $\{A: w_1, w_2\}$, donde A es un subconjunto de C y se elige de una manera que depende del parámetro c , y, para $z \in C$, $w_1(z) = \sqrt{z - c}$, $w_2(z) = -\sqrt{z - c}$.

Los mapas w_1 y w_2 no son mapas de contracción sobre la totalidad de C , sino el subconjunto A de C . Por esta razón, el punto de partida elegido para el proceso iterativo que se usa para generar el conjunto de Julia podría presentar el problema de que se tendría que asegurarse que el punto de partida elegido esté en A . Sin embargo, se puede demostrar que el punto de partida puede ser cualquier punto en C , excepto en el caso $c=0$, en cuyo caso el punto 0 debe evitarse (Keen (1994)).

2.2 Elección de c

Los conjuntos Julia se pueden dividir en dos clases. Conectados o desconectados. En general, un conjunto que está conectado está todo en una sola pieza (sin saltos), mientras que un conjunto que está totalmente desconectado es como una nube de partículas de polvo, sus únicos componentes conectados son puntos. Si $|c| < 2$, entonces los dos tipos de conjuntos de Julia surgen y ambos tipos son por lo general visualmente agradables. Si $|c| > 2$, entonces el conjunto Julia de Q_c está desconectado. Este hecho, junto con la amplitud de c , produce conjuntos Julia que son más bien dispersos y extendidos y por lo tanto visualmente no son muy "estéticos". En la sección 2.3 se muestra el conjunto Julia de $Q_{-0.757}$ y en la sección 2.9 se enseña el conjunto Julia de $Q_{0.306-0.214i}$ que está conectado y el conjunto Julia de $Q_{2.1i}$ que está desconectado.

2.3 La construcción de conjuntos de Julia utilizando el algoritmo determinista

Enseguida se construyen conjuntos Julia para la función cuadrática Q_c usando el algoritmo determinista en $\{A: w_1, w_2\}$.

Sea $w = w_1 \cup w_2$ el mapeo contracción en A .

Usando el algoritmo determinista para crear el atractor de w y comenzando con el único conjunto punto $\{z\}$ aplicamos la función w , n veces. La primera aplicación es la siguiente:

- Paso 1

Considere la función w operando sobre $w\{z\}$:

$$w[\{z\}] = \{w_1(z), w_2(z)\} = \{\sqrt{z-c}, -\sqrt{z-c}\}.$$

Teóricamente w es una función del conjunto $\{z\}$, sin embargo, al usar comandos de Mathematica, se omiten las llaves y se dice que w es una función de z , es decir, “ $w[z]$ ”.

Definimos w como una función en Mathematica y sustituimos c con k .

$$w[z_]:= \{\sqrt{z-k}, -\sqrt{z-k}\};$$

Para encontrar $w[1-2I]$, cuando $c = -1$, se utiliza el comando de remplazo /. para remplazar k por -1 .

$$w[1-2I]/.k \rightarrow -1$$

$$LNING = \beta_0 + \beta_1 SEXO + \beta_2 EDAD + \beta_3 EDAD2 + \beta_4 PRIMCOM + \beta_5 SECOM + \beta_6 PCOM + \beta_7 IHOG + \beta_8 IOCUP + \beta_9 AEST + \epsilon_i, \dots \dots \dots (1)$$

Como la construcción de un conjunto Julia implica muchos cálculos, es recomendable acelerar el proceso trabajando con aproximaciones. Esto se puede hacer escribiendo el argumento de la función como una aproximación:

$w[1.0-2I]/.k \rightarrow -1.0$

$\{1.55377_-0.643594^{\text{TM}}, -1.55377+0.643594^{\text{TM}}\}$

- Paso 2

Se realiza el segundo paso iterativo encontrando la imagen del anterior conjunto de puntos bajo w . Es decir, se necesita evaluar $w[w[z]]$. El comando Map de Mathematica se usa para producir el conjunto de 4 puntos $w[w[l - 2I]]$ como sigue:

$\text{Map}[w,w[1.0-2I]]/.k \rightarrow -1$

$\{\{1.6105-0.199812^{\text{TM}}, -1.6105+0.199812^{\text{TM}}\}, \{0.384234_+0.837502^{\text{TM}}, -0.384234-0.837502^{\text{TM}}\}\}$

- Paso 3

Continuando así, ahora tenemos que crear el conjunto, $w[w[w[l - 2I]]]$, que consta de 8 puntos. Sin embargo w no puede ser mapeado al último output generado, ya que los argumentos de w son números complejos, y el último output producido es una lista de pares de números complejos. De las llaves se elimina algo para obtener una lista de coordenadas de puntos. Esto se hace usando el comando Flatten de Mathematica.

$h=\text{Flatten}[\text{Map}[w,w[1.0-2I]]/.k \rightarrow -1,1]$

$\{1.6105-0.199812^{\text{TM}}, -1.6105+0.199812^{\text{TM}}, 0.384234+0.837502^{\text{TM}}, -0.384234-0.837502^{\text{TM}}\}$

Ahora consideremos la siguiente función pura, g , en la que $\#$ se puede sustituir por una lista del tipo anterior:

$g=\text{Flatten}[(\text{Map}[w,\#]/.k \rightarrow -1),1]\&;$

Cuando g es aplicada a una lista de números complejos, aplica ω , con $k = -1$, a cada una, sustituyéndola por un par de números complejos y luego se alisa la lista resultante de los pares, obteniendo así una lista de números complejos individuales. Aplicamos la función anterior a la lista previa:

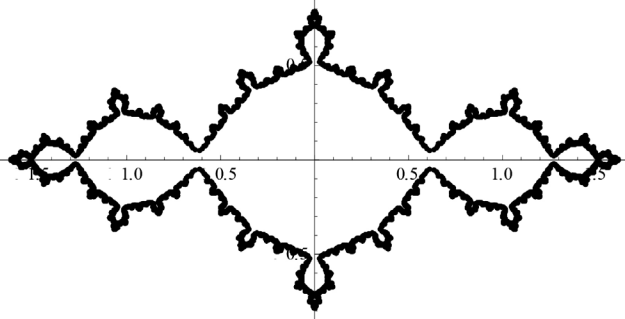
```
Flatten[(Map[w,#]/.k→ -1.0),1]&[h]
{1.61688-0.0617893™,-1.61688+0.0617893™,0.126228+0.791473™,-
0.126228-0.791473™,1.22518+0.341789™,-1.22518-
0.341789ä,0.909746-0.460295™,-0.909746+0.460295™}
```

La función g pura definida anteriormente se puede anidar, o aplicar repetidas veces, a una lista de puntos usando la función Nest de Mathematica:

```
Nest[Flatten[(Map[w,#]/.k→ -1),1]&,{1.0-2I},4]
{1.61779-0.0190968™,-1.61779+0.0190968™,0.0392861+0.786402™,
-0.0392861-0.786402™,1.11865+0.353763™,-1.11865-0.353763™,1.01309
-0.390622™,-1.01309+0.390622™,1.49607+0.114229™,-1.49607-0.114229™,
0.303415-0.563237™,-0.303415+0.563237™,1.39179-0.16536™,
-1.39179+0.16536™,0.528826+0.435204™,-0.528826-0.435204™}
```

Ahora usamos el comando Nest para iterar g un número suficiente de veces para obtener una lista de números complejos que, cuando se grafican, forman una aproximación al conjunto Julia. Primero se aplica la función pura $(\{\text{Re}[\#], \text{Im}[\#]\})\&$ a la lista anterior de números complejos para obtener una lista de coordenadas de puntos y graficamos estos puntos usando el comando ListPlot:

```
ListPlot[{Re[#],Im[#]}&/@Nest[Flatten[(Map[w,#]/.k→-1.0),1]&,{1-21},15],AspectRatio→Automatic,Prolog→PointSize[0.004]]
```



2.4 Conjuntos Julia y conjuntos Julia completos

La iteración de una función analítica compleja f se descompone en el plano complejo en dos conjuntos disjuntos: el conjunto estable o de Fatou en el cual las iteraciones son “bien comportadas” y el conjunto Julia en el que el mapa es caótico.

Consideramos tres clases de funciones analíticas: polinomios, racionales y ciertas funciones trascendentes enteras. (Una función entera es una que es analítica en todo punto del plano complejo. Los polinomios son funciones enteras. Ejemplos de funciones trascendentes enteras son seno, coseno, exponencial y sumas o productos entre éstas y/o polinomios). En los tres casos, el conjunto de Julia se define como la cerradura del conjunto de puntos periódicos repulsores de la función. Sin embargo, el criterio para determinar conjuntos Julia de funciones racionales difiere del de los conjuntos de Julia de funciones trascendentes enteras, debido a esto, distintas técnicas se emplean para producir distintos tipos de conjuntos Julia.

Aunque los polinomios forman un subconjunto del conjunto de funciones racionales, conjuntos Julia y Julia completos de polinomios, se expondrán primero por separado ya que éstos son más fáciles de crear y su generación proporciona una introducción al algoritmo de tiempo de escape.

2.5 Conjuntos Julia y Julia completos de polinomios

Sea P un polinomio de grado $P > 2$, sea J_p el conjunto Julia de P y sea K_p el conjunto Julia completo de P . $K_p = \{z \in C : \text{la órbita de } z \text{ bajo } P \text{ está acotada}\}$ y J_p es la cota de K_p . Para cualquier polinomio P de grado mayor o igual a 2, el infinito es un punto fijo atractor. La cuenca de atracción de infinito, $A(\infty)$, es el conjunto de puntos cuyas órbitas no son acotadas. Para estos polinomios, el conjunto Julia de P es el límite de la cuenca de atracción de infinito. Pero este límite también es el límite del conjunto Julia completo, K_p . Además, el conjunto Julia de P también es el límite de la cuenca de atracción de cualquier ciclo atractor. Esto significa que todos los ciclos atractores tienen el mismo límite.

2.6 Criterios de escape para polinomios

Para construir el conjunto Julia completo de un polinomio, P , de grado mayor o igual a 2, es necesario encontrar el conjunto de puntos en el plano cuyas órbitas tienden a infinito bajo P . Se puede utilizar el siguiente resultado: si P es un polinomio de grado mayor o igual a 2, existe $R > 0$ tal que si $|z| > 1$, entonces $|P[z]| > |z|$ y se deduce que $|P^n[z]| \rightarrow \infty$ cuando $n \rightarrow \infty$ para todo z satisfaciendo $|z| > R$ (Devaney (1989)). La condición $|z| > R$ se llama un criterio de escape.

A veces un criterio de escape es conocido para toda una clase de polinomios. En el caso de polinomios de grado 2, es decir polinomios de la forma $P[z] = Az^2 + bz + d$, ($a, b, d \in C$), se puede mostrar que el conjunto Julia (completo) de P es geoméricamente similar al conjunto Julia (completo) de un polinomio cuadrático de la forma $Q_c[z] = z^2 + c$ para alguna $c \in C$. por esta razón, para polinomios de grado 2, se analizan únicamente los conjuntos Julia (completos) de Q_c . Además, cualquier polinomio de grado 3 se puede escribir en la forma $C_{a,b}[z] = z^3 + az + b$ después de una transformación afín adecuada, por lo que necesitamos sólo examinar polinomios de la forma anterior. Entonces sólo se consideran polinomios de la forma $z^2 + c$, $z^3 + az + b$ y $z^n + c$, donde

$a, b, c \in C$. Para cada una de estas clases de polinomios, se conoce un criterio de escape.

2.7 El papel de los puntos críticos

Los puntos finitos críticos de un polinomio P son los puntos $a \in C$ que satisfacen la ecuación $P'[a] = 0$.

El comportamiento de las órbitas de los puntos críticos da información acerca de la estructura de los conjuntos Julia (completos), en el sentido que establecen que el conjunto Julia (completo) de P está conectado si y sólo si no existe un punto finito crítico de P en $A(\infty)$. Así $(K_p)J_p$ está conectado si y sólo si la órbita de cada punto crítico finito está acotada. Por el contrario, si $P^n[a] \rightarrow \infty$ cuando $n \rightarrow \infty$ para todo punto crítico finito, entonces $(K_p)J_p$ está totalmente desconectado y $K_p = J_p$.

2.8 Polinomios de grado dos

Sea $P[z] = az^2 + bz + d$, ($a, b, d \in C$) un polinomio cuadrático. Como se ha mencionado anteriormente, se puede demostrar que el conjunto Julia (completo) de P es geoméricamente similar al conjunto Julia (completo) de un polinomio cuadrático de la forma $Q_c[z] = z^2 + c$ para algún $c \in C$.

Ya que un polinomio de grado 2 sólo tiene un punto crítico finito, su conjunto Julia (completo) es conectado o desconectado por completo. En particular, el polinomio Q_c tiene el único punto crítico 0, de modo que la prueba de sí o no la órbita de 0 está acotada indica sí o no $(K_p)J_p$ está conectado o desconectado totalmente (La órbita de 0 es muy importante en la generación del conjunto de Mandelbrot para polinomios de la forma Q_c).

2.9 Cálculo del conjunto Julia completo para el polinomio cuadrático $Q_c[z] = z^2 + c$ ($z, c, \in \mathbb{C}$)

Sea $Q_c[z] = z^2 + c$, ($z, c, \in \mathbb{C}$). Se describe un proceso algorítmico para obtener una aproximación a la imagen del conjunto Julia completo K_c de Q_c , donde $K_c = \{z \in \mathbb{C} : \text{h órbita el } z \text{ bajo } Q_c \text{ es acotada}\}$.

Se emplea el siguiente criterio de escape:

Sea $Q_c[z] = z^2 + c$, ($z, c, \in \mathbb{C}$). Asumir que $|z| > 2$, entonces $|Q_c^n[z]| \rightarrow \infty$ cuando $n \rightarrow \infty$. Así, para todo $k \in \mathbb{N}$, si $|Q_c^k[z]| > 2$, entonces $z \notin K_c$ (Devaney (1992)).

Basándose en este hecho, el algoritmo de escape temporal aplica el siguiente razonamiento: si después de m iteraciones, la órbita de z está acotada por el número 2, se debe asumir que la órbita de z está acotada por lo que se encuentra en K_c . La elección de m depende de la precisión requerida y la velocidad y la resolución del equipo que se está utilizando.

Usamos una adaptación de programas de Mathematica para originar conjuntos Julia.

Para obtener iteraciones de z , se aplica la función pura $Q_c = \#^2 + c$ & repetidamente al número complejo z , a lo más 100 veces. Por ejemplo, eligiendo $c = 0.51$, y $z = -1 - 0.4I$, y aplicando Q_c dos veces, se logra:

```
(#^2+0.5 + 1)&[(#^2+0.5+1)&[-1-0.4I]] -0.9444+5.824 I
```

Ahora usamos el comando de Mathematica FixedPointList:

```
FixedPointList[#^2 + c &, x + Iy, 100, SameTest -> (Abs[#] > 2.0 &)];
```

Este comando aplica la función Q_c repetidamente a $x + Iy$ hasta que el resultado, #, tiene un valor absoluto superior a 2 o hasta que se han hecho 100 aplicaciones. SameTest se puede pensar como una

‘prueba de parada’. La función pura $\text{Abs}[\#] > 2$ devuelve True o False para el resultado de toda aplicación de Q_c a z .

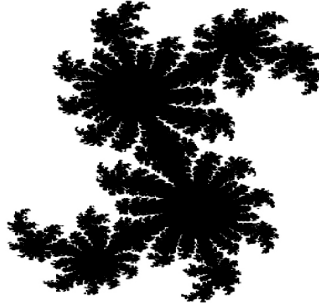
Ahora usamos el comando de Mathematica Length, que cuenta el número de puntos, n , por ejemplo, en la órbita y devuelve el entero n . Por ejemplo:

```
Length[FixedPointList[#^2+I&,1+0.3I,100,SameTest→(Abs[#]>2.0&)]]
```

Si el comando anterior devuelve un entero $n > 100$, entonces $z \in K_c$, ya que z no ha escapado es decir, se supone que la órbita de z es acotada, por lo que el punto que la representa debe ser de color negro. Si $n < 100$, entonces el punto que representa z ha escapado y entonces debe ser de color blanco.

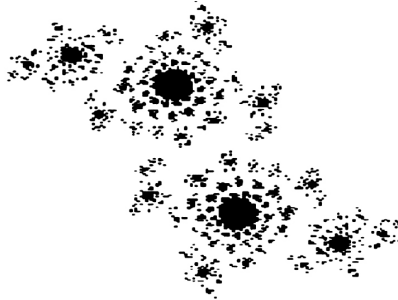
Ahora bien, si el comando `DensityPlot` se aplica al comando anterior `[Length]` entonces cada argumento volviendo de `DensityPlot` será un número natural que está entre 1 y 100 y el output será graficado en `GrayLevel` es decir, en diferentes tonos de gris dentro de los dos extremos de blanco y negro. Si se quiere el output en blanco y negro solamente, se puede usar el comando `ColorFunction` que aparece enseguida.

```
DensityPlot [Length [FixedPointList[#2+0.2481
&,x+Iy,100, SameTest→(Abs [#]> 2.0
&)], {x,-1.6,1.6} , {y, - 1.2,1.2}, Mesh→False,Frame→
False,Frame→False,Axes→False,PlotPoints
→400,AspectRatio→Automatic,ColorFunction→(If [#≥
1, RGBColor [ 0,0,0], RgbColor [1,1,1]&)];
```

Este es un ejemplo de un conjunto Julia completamente conectado.
Un ejemplo de un conjunto Julia totalmente desconectado es

```
DensityPlot [Length [FixedPointList[#2+0.661 - 0.2481
&,x+Iy,100, SameTest→(Abs [#]> 2.0
&)], {x,-1.6,1.6} , {y, - 1.2,1.2}, Mesh→False,Frame→
False,Frame→False,Axes→False,PlotPoints
→400,AspectRatio→Automatic,ColorFunction→(If [#≥
1, RGBColor [ 0,0,0], RGBColor [1,1,1]]&)];
```



2.10 Mathematica en la investigación económica

Mathematica ha sido usado para hacer investigación en prácticamente todos los campos de la economía. Una ilustración de esto, se

hace con uno de los modelos canónicos en la economía: el modelo multiplicador-acelerador de Samuelson (1939). Este modelo es un ejemplo que ilustra el uso de ecuaciones recursivas y la variedad de trayectorias de solución para el ingreso en una economía. El consumo se relaciona con el ingreso rezagado, mientras la inversión en el tiempo t tiene que ver con la diferencia entre el ingreso en el tiempo $t-1$ y $t-2$. El gasto de gobierno se considera constante e igual a G . Entonces el modelo es

$$\begin{aligned}C_t &= a + bY_{t-1} \\ I_t &= v(Y_{t-1} - Y_{t-2}) \\ G_t &= G \text{ para todo } t \\ E_t &= C_t + I_t + G_t \\ Y_t &= E_t\end{aligned}$$

Por sustitución directa se obtiene la ecuación en diferencia de segundo orden no homogénea siguiente:

$$Y_t - (b + v)Y_{t-1} + vY_{t-2} = a + G$$

La solución particular se logra haciendo $y_t = y^*$ para todo t . Por lo tanto

$$Y^* - (b + v)Y^* + vY^* = a + G$$

es decir

$$Y^* = \frac{a + G}{1 - b}$$

Significa que, en equilibrio, el ingreso es igual al resultado del multiplicador simple.

El resultado complementario se obtiene resolviendo el componente homogéneo:

$$y_t - (b + v)Y_{t-1} + vY_{t-2} = 0$$

con ecuación característica

$$x^2 - (b+v)x + v = 0$$

con soluciones

$$r, s = \frac{-(b+v) \pm \sqrt{(b+v)^2 - 4v}}{2}$$

El modelo puede producir una variedad completa de trayectorias para y_t dependiendo de los valores de los parámetros b y v . En efecto, de la ecuación característica se tienen tres resultados posibles:

- (i) Raíces reales distintas: $(b+v)^2 > 4v$
- (ii) Raíces reales iguales: $(b+v)^2 = 4v$
- (iii) Raíces complejas: $(b+v)^2 < 4v$

Al determinar las implicaciones de estos posibles resultados se usan las propiedades de las raíces

$$r + s = b + v$$

$$rs = v$$

También se sigue de usar estos resultados que:

$$\begin{aligned} (1-r)(1-s) &= 1 - (r+s) + rs \\ &= 1 - (b+v) + v \\ &= 1 - b \end{aligned}$$

Y ya que $0 < b < 1$, entonces $0 < (1-r)(1-s) < 1$.

Con raíces reales y distintas la solución general es

$$Y_t = c_1 r^t + c_2 s^t + Y^*$$

donde r es la más grande de las raíces. La trayectoria de es determinada por la raíz más grande, $r > s$. Si $b > 0$ y $v > 0$, entonces $rs = v > 0$ y las raíces deben tener el mismo signo. Además, si $r + s = b + v > 0$, entonces r y s deben ser positivas. La trayectoria del ingreso no puede oscilar. Sin embargo, la trayectoria será amortiguada si la raíz más grande está entre cero y uno. De esta manera, surgirá una trayectoria amortiguada si $0 < s < r < 1$, lo cual sucede si $0 < b < 1$ y $v < 1$; similarmente la trayectoria será explosiva si la mayor raíz es mayor a uno, es decir si, $r > s > 1$ lo que implica que $0 < b < 1$ y $rs = v > 1$.

Con sólo una raíz real, r , se mantienen las mismas condiciones. Por lo tanto, en el caso de raíces reales con $0 < b < 1$ la trayectoria del ingreso es amortiguada para $0 < v < 1$ y explosiva para $v > 1$.

Si la solución es compleja conjugada entonces,

$$r = \alpha + \beta i, \quad s = \alpha - \beta i$$

y la solución general

$$Y_t = c_1 R^t \cos(t\theta) + c_2 R^t \sen(t\theta) + Y^*$$

tendrá oscilaciones cuya naturaleza amortiguada o explosiva dependerá de la amplitud R .

Sabemos que

$$R = \sqrt{\alpha^2 + \beta^2}$$

pero

$$\alpha = \frac{b+v}{2}$$

y

$$\beta = \frac{\sqrt{4v - (b+v)^2}}{2}$$

Por lo tanto

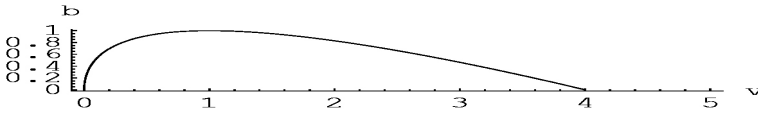
$$R = \sqrt{\left(\frac{b+v}{2}\right)^2 + \frac{4v - (b+v)^2}{4}} = \sqrt{v}$$

para oscilaciones amortiguadas, $R < 1$, es decir, $v < 1$;

para oscilaciones explosivas, $R > 1$, es decir, $v > 1$.

Todos los casos se ilustran en la figura siguiente generada en *Mathematica*. La línea que separa raíces reales de complejas es la curva $(b+v)^2 = 4v$, que se puede dibujar usando en *Mathematica* el comando `ImplicitPlot`.

```
Clear b, v;
<<Graphics'ImplicitPlot'
ImplicitPlot [(b+v)^2 == 4*v, {v, 0, 5}, {b, 0, 1}, AxesLabel { "v", "b" } ]
```



Por otro lado, la división entre trayectorias amortiguadas y explosivas (dado $0 < b < 1$) es determinada por $v < 1$ y $v > 1$, respectivamente.

2.10.1 Solución y análisis del modelo multiplicador-acelerador de Samuelson

El modelo de Samuelson se resume en la ecuación en diferencia de segundo orden no homogénea:

$$Y_t - (b+v)Y_{t-1} + vY_{t-2} = G$$

Utilizando el comando RSolve tenemos

```
Clear [b,v,t,b,g,y];
Rsolve [ y[t+2] - (b+v)* y[t+1]+v* y[t] = =g,y[t],t] //
FullSimplify
{{y[t]→ g1-b+2-t((b+v--4v + (b+v)2)
tC[1]+(b+v+-4v+(b+v)2)tC[2])}}
```

2.10.2 Estabilidad y convergencia del modelo de Samuelson

```
Clear [b,v,t,b,g,y];
Rsolve [ y[t+2] - (b+v)* y[t+1]+v* y[t] = =g,y[t],t] //
FullSimplify
```

```
{{y[t] → (2-1-t(10(-1 + b)((-4 + b + v)(b + v - √(-4v + (b + v)2))t
+ √(-4v + (b + v)2(b + v - √(-4v + (b + v)2))t - (-4 + b
+ v)(b + v + √(-4v + (b + v)2))t
+ √(-4v + (b + v)2(b + v + √(-4v + (b + v)2))t
+ g(-21+t√(-4v + (b + v)2 + (-2 + b + v)(b + v - √(-4v + (b + v)2))t
+ √(-4v + (b + v)2(b + v - √(-4v + (b + v)2))t - (-2 + b
+ v)(b + v + √(-4v + (b + v)2))t
+ √(-4v + (b + v)2(b + v + √(-4v + (b + v)2))t)))/((-1
+ b)√(-4v + (b + v)2))}}
```

y[t]=y[t]/.%[[1]];

Ahora ilustramos varios ejemplos de convergencia o divergencia de la solución:

Primer caso: raíces reales con solución divergente
Definimos los valores

$$\{v, b, g, y_0, y_1\} = \{2, 0.9, 4, 10, 20\}$$

$$\{2, 0.9, 4, 10, 20\}$$

Con estos valores las raíces de la ecuación característica

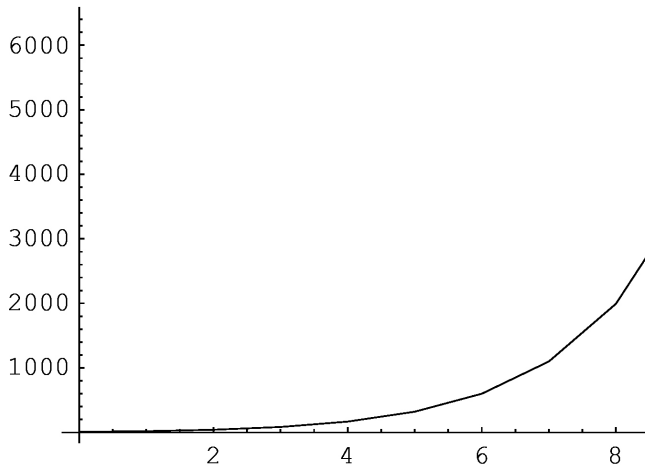
$$\{r_1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r_2$$

$$= \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

$$\{1.1298437881283574, 1.7701562118716425\}$$

Gráficamente:

```
m=table[{t,y[t]},{t,0,10}];
ListPlot [m,PlotJoined→True]
```



... Graphics ...

La solución es divergente.

Segundo caso: raíces reales con solución convergente
($b < 1/v$)

Definimos los valores:

$$\{v,b,g,y0,y1\}=\{0.28,0.78,4,30,29\}$$

$$\{0.28,0.78,4,30,29\}$$

Con estos valores las raíces de la ecuación característica

$$\{r1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r2 = \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

y[t]

$$-75.757575757575272^{-1-t}(-0.2199999999999997(-24.399999999999880.999999999999997^t + 28.000000000000011.120000000000006^t) + 4(-0.8799999999999960.999999999999997^t + 1.0000000000000041.120000000000006^t - 0.060000000000000392^{1+t}))$$

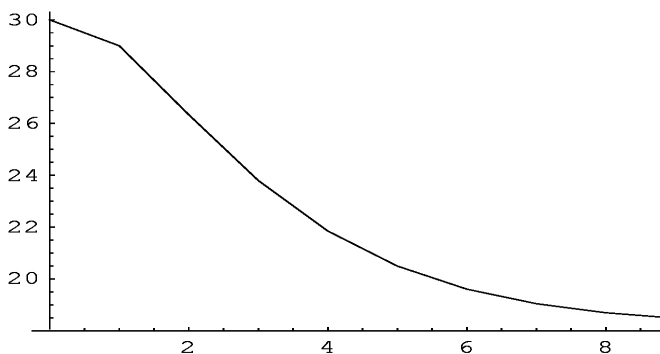
m = Table[{t, y[t]}, {t, 0, 10}];

$$\{r1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r2 = \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

y[t]

m = Table[{t, y[t]}, {t, 0, 10}];

ListPlot[m, PlotJoined -> True]



... Graphics ...

$A=g/(1 - b)= 18.1818$. Este es el valor al que converge la solución particular.

Tercer caso: raíces complejas

Definimos los valores

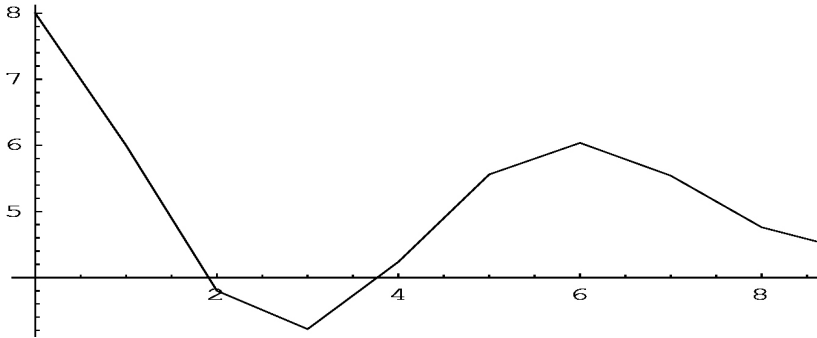
$$\{r1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r2 = \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

$$\{0.44999999999999996 - 0.7053367989832943i, 0.44999999999999996 + 0.7053367989832943i\}$$

y[t]

$$(0. + 0.88610150626042i)2^{-1-t}(-0.8((-4.8000000000000001 + 11.285388783732708i)(0.8999999999999999 - 1.4106735979665885i)^t + (4.8000000000000001 + 11.285388783732708i)(0.8999999999999999 + 1.4106735979665885i)^t) + 4((-1.1 + 1.4106735979665885i)(0.8999999999999999 - 1.4106735979665885i)^t + (1.1 + 1.4106735979665885i)(0.8999999999999999 + 1.4106735979665885i)^t - (0. + 1.4106735979665885i)2^{1+t}))$$

```
m = Table[{t, y[t]}, {t, 0, 10}];
ListPlot[m, PlotJoined -> True]
```



... Graphics ...

$A=g/(1 - b) = 5$. Movimiento oscilatorio amortiguado que converge a la solución particular 5.

Estabilidad

Para determinar la estabilidad de la ecuación en diferencia lineal de segundo orden del modelo de Samuelson

$$Y_t - (b + v)Y_{t-1} + vY_{t-2} = G$$

la escribimos en forma del sistema

$$Y_{t+1} = Z_t$$

$$Z_{t+1} = (b + v)Z_t - vY_t + G$$

Imponemos la condición

$$Y_{t+1} = Y_t$$

y

$$Z_{t+1} = Z_t$$

y el sistema resulta

$$Y_t = Z_t$$

$$Z_t = (b + v)Z_t - vY_t + G$$

El punto de equilibrio lo obtenemos resolviendo este sistema:

Clear[*x,y,z,b,v,g*];

Solve[{*y*==*z*,*z*==(b+v)**z*-v**y*+g},{*y*,*z*}]

$$\left\{ \left\{ y \rightarrow -\frac{g}{-1+b}, z \rightarrow -\frac{g}{-1+b} \right\} \right\}$$

Entonces el punto de equilibrio es $y = g/(1 - b)$.

Autovalores

A::={{0,1},{-v,b+v}};

Eigenvalues[A]

$$\frac{1}{2}(b + v - \sqrt{b^2 - 4v + 2bv + v^2}), \frac{1}{2}(b + v + \sqrt{b^2 - 4v + 2bv + v^2})$$

Primer caso: raíces reales distintas con solución divergente

`Clear[b,v];`

$$\{r1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r2 \\ = \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

{1.1298437881283574,1.7701562118716425}

No hay estabilidad porque ambas raíces mayores a uno.

Segundo caso: raíces reales distintas con solución convergente.

`Clear[b,v];`

`{v,b} = {0.28,0.78};`

$$\{r1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r2 \\ = \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

{0.4999999999999999,0.5600000000000002}

Existe estabilidad porque ambas raíces menores de uno.

Tercer caso: raíces complejas.

`Clear[b,v];`

`{v,b} = {0.7,0.2};`

$$\{r1 = \frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2}), r2 \\ = \frac{1}{2}(b + v + \sqrt{b^2 - 4 * v + 2 * b * v + v^2})\}$$

{0.44999999999999996 - 0.7053367989832943i,0.44999999999999996
+ 0.7053367989832943i}

$$\text{Abs}[\frac{1}{2}(b + v - \sqrt{b^2 - 4 * v + 2 * b * v + v^2})]$$

0.8366600265340756

Ya que las raíces en módulo son menores a uno, las trayectorias tienen un comportamiento estable.

Variantes del modelo multiplicador-acelerador:

1. Modelo de acelerador lineal de Hicks. (Ecuación en diferencia de segundo orden)
2. Modelo de expectativas y ciclos de existencia en el proceso de producción de Metzler.

Algunas consideraciones finales

La experimentación con sistemas dinámicos a tiempo discreto con el uso de un lenguaje simbólico como *Mathematica*, puede ser un recurso didáctico importante en la investigación de las propiedades de dichos sistemas y de sus posibles aplicaciones a disciplinas como la economía donde el tiempo es a menudo discontinuo (discreto) porque las decisiones en economía por lo general no pueden ser continuamente revisadas. Para ilustrar esta posibilidad didáctica, se ha utilizado el lenguaje *Mathematica*, primero para experimentar con el ejemplo de órbitas de puntos bajo iteración de funciones reales y complejas y después con el ejemplo de conjuntos Julia. Estos dos casos pueden ilustrar las posibilidades para la didáctica de los sistemas dinámicos discretos, mediante una aproximación experimental con un lenguaje como *Mathematica*. Esta aproximación puede propiciar el entendimiento intuitivo de los sistemas y complementar el estudio teórico de los sistemas dinámicos a tiempo discreto.

Referencias bibliográficas

- Bamsley, M. (1988). *Fractals everywhere*. Second Edition, Academic Press.
- Beardon, A. F. (1991). *Iteration of rational functions*. First Edition, Springer-Verlag, New York.
- Chossat P. and Golubitsky M. (1998). “Symmetry-increasing bifurcation of chaotic Attractors”. *Physica D* 32. P. 423.
- Devaney, R. L. (1988). *Fractal patterns arising in chaotic dynamical systems*. The science of fractal images, edited by H. O. Peitgen and D. Saupe, Springer-Verlag, New York.
- (1989). *An introduction to chaotic dynamical systems*. Second Edition, Addison-Wesley.
- R. L. (1992). *A first course in chaotic dynamical systems*. First Edition, Addison-Wesley, Boston.
- de Melo, W., and van Strien, S. (1991). *One-dimensional dynamics*. Springer-Verlag, Berlin Heidelberg, New York.
- Gabisch, G. and Lorenz, H.W. (1989). *Business cycle theory*. Second ed. Springer-Verlag, Berlin Heidelberg, New York.
- Guckenheimer, J. and Holmes, P. (1983). *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. Springer-Verlag, New York.
- Gumowski I. and Mira C. (1980). “Recurrences and discrete dynamic system, Lecture notes in Mathematics.” ZAMM. *Journal of applied Mathematics and Mechanics*, Springer-Verlag.
- Holmgren, R. A. (1994). *A first course in discrete dynamical systems*. First edition, Springer-Verlag, New York.
- Iooss, G. (1979). *Bifurcation of maps and applications*. North-Holland Publishing Company, Amsterdam.
- and Joseph, D.D. (1980). *Elementary stability and bifurcation theory*. Springer-Verlag, New York.
- Keen, L. (1989). “Julia Sets.” *Proceedings of Symposia in Applied Mathematics*, vol. 39, p. 57.
- Maeder, Roman E. (1996). *Programming in Mathematica*. Third edition, Addison-Wesley.

- Peitgen, H. O., Jurgens, H. and Saupe, D. (1992). *Chaos and fractals, New frontiers of science*, Springer-verlag.
- Samuelson, P. A. (1939). "Interactions between the multiplier analysis and the principle of acceleration", *The Review of Economics and Statistics*, vol. 21, núm. 2, mayo, 1939, Pp. 75-78.
- Sharkovsky, A.N., Kolyada, S.F., Sivak, A.G. and Fedorenko, V.V (1997). *Dynamics of one-dimensional maps*, Kluwer Academic Publishers, London.
- Wiggins, S. (1988). *Global bifurcations and chaos, analytical methods*, Springer Verlag, New York.