PRACTICA 1

MODELOS DISCRETOS

1.1. Objetivo

En esta práctica vamos a usar la potencia de la recursión para experimentar con diferentes modelos discretos lineales y no lineales. Observaremos puntos de bifurcación y caos para el modelo de *Ricker* y el modelo logístico de *May*.

1.2. Introducción

Consideremos la siguiente ecuación $x = \cos x$. Cualquier solución de esta ecuación es la abscisa de la intersección de la recta $y = x$ con la gráfica de la función $y = \cos x$.

EJERCICIO 1 Dibuja las dos gráficas y comprueba que hay un punto de intersección en el intervalo $[0, 1]$**.**

En general, cualquier solución de la ecuación $f(x) = x$ se llama un punto fijo de la función f.

La fórmula de iteración $x_{k+1} = f(x_k)$ se llama **iteración funcional o iteración de punto fijo** y en muchos casos, dependiendo de la función f y del punto inicial x_0, la sucesión $\{x_k\}$ converge al punto fijo.

Para poder construir estas sucesiones podemos utilizar las siguientes órdenes del programa Mathematica®

`Nest[f,`x_0`,k]:` da el término k-ésimo de la sucesión.

`NestList[f,`x_0`,k]` : da una lista con las iteraciones desde x_0 a k.

Las órdenes `FixedPoint[f,`x_0`]` y `FixedPointListNest[f,`x_0`]` son similares a las anteriores salvo que paran cuando encuentran dos iteraciones sucesivas iguales.

EJEMPLO 1.1

- Vamos a utilizar las ordenes anteriores para construir la sucesión de iteraciones que se obtiene para $f(x) = \cos x$ comenzando en $x_0 = 0$. Con `ListPlot[ ]` podemos ahora dibujar los puntos y hacer una interpretación gráfica de lo que ocurre.

  Empezamos definiendo la función

  ```
  f[x_ ]:= Cos[x]
  ```

  A continuación, construimos los veinte primeros términos de su órbita

  ```
  iters=NestList[f,0.,20]
  ```

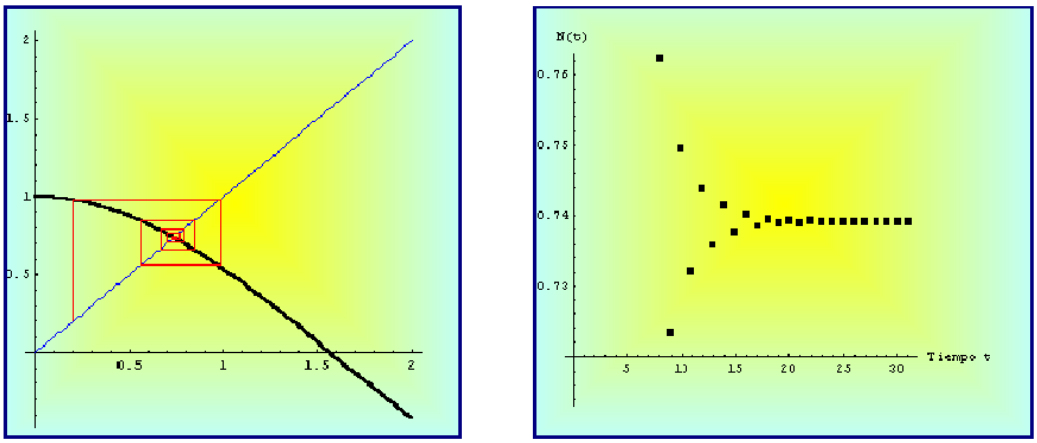

Figura 1.1: Diagrama de Cobweb para $f(x) = \cos x$

  Dibujamos en primer lugar la función $f[x]$ y la bisectriz del primer cuadrante.

  ```
  fg=Plot[{x,f[x]}, {x,0,1.5}, PlotStyle → {RGBColor[1,0,0],
  RGBColor[0,0,1]}, DisplayFunction → Identity]
  ```

  Trazamos la órbita

  ```
  gi=ListPlot[Partition[Flatten[Transpose[{iters, iters}]],2,1],
  PlotJoined → True,DisplayFunction → Identity]
  ```

  Finalmente, superponemos los dos gráficos.

  ```
  Show[fg,gi,AspectRatio → 1, DisplayFunction → $DisplayFunction]
  ```

- La conclusión es que existe un único punto de equilibrio $x_0 = 0.74$ que es estable.

EJERCICIO 2 Supongamos que nos encontramos en una ciudad de un rico país. Estamos interesados en analizar la evolución de la población de este ciudad. Por desgracia, las condiciones económicas no son las más adecuadas, de tal forma que si se encontraran aisladas su población disminuiría según el siguiente modelo exponencial o de *Malthus*

$$N_{t+1} = 0.5N_t\,, \qquad t = 0, 1, 2, \cdots\,, \tag{1.1}$$

donde N_t es la población en el tiempo t.

Cada año 100 personas del país emigran a nuestra ciudad.

(a) **Modifica el modelo (1.1) para tener en cuenta el factor de la emigración**

(b) **Supongamos que inicialmente hay 30 personas. Encontrar los primeros 10 términos de su órbita**

(c) **Describir el comportamiento a "largo plazo"de la población.**

(d) **Encontrar los puntos de equilibrio del modelo y clasificarlos.**

Repetir el ejercicio suponiendo que ahora el modelo exponencial es:

$$N_{t+1} = 1.5N_t\,, \qquad t = 0, 1, 2, \cdots\,.$$

El Teorema del Punto Fijo de *Brouwer* establece que toda función continua de un intervalo cerrado en sí mismo $f : [a, b] \to [a, b]$, tiene al menos un punto fijo en $[a, b]$.

En la próxima sección analizaremos el comportamiento de los términos de la iteración funcional con respecto a los puntos fijos.

Recordemos que si x^* es un punto fijo, entonces $f(x^*) = x^*$. Si la función f es derivable, y

$$|f'(x^*)| < 1\,,$$

entonces el punto x^* se llama **punto fijo atractor**. Cuando una iteración funcional comienza suficientemente cerca de él irremediablemente cae dentro de su ámbito de influencia y la sucesión converge.

Por el contrario, si

$$|f'(x^*)| > 1$$

se trata de un **punto fijo repulsor**, y por muy cerca de él que se comience, la sucesión termina por alejarse.

1.3. Modelo logístico de May

Suele utilizarse con mucha frecuencia para hacer ver como un modelo matemático determinista no lineal que depende de un parámetro, puede presentar múltiples comportamientos. Recordemos que el modelo discreto logístico viene dado por la familia de funciones

$$y_c(x) = cx(1-x)\,, \quad c \in \mathbb{R}^+\,, \quad x \in [0,1]\,,$$

donde x representa la fracción de la máxima población posible de una especie, por lo que sólo consideraremos valores de x comprendidos entre cero y uno.

EJEMPLO 1.2

- Consideremos la familia de parábolas

$$y = cx(1-x)\,, \quad c \in \mathbb{R}^+\,, \quad x \in [0,1]\,.$$

Empezamos definiendo una función (de c y x) para la familia anterior.

$$\texttt{f[c\_, x\_]} := \texttt{c} * \texttt{x} * (1 - \texttt{x})$$

(a) ¿Cuáles son los puntos fijos en función de c?

(b) Consideremos el caso $0 < c < 1$. ¿Cuáles son los puntos fijos? ¿Son los puntos fijos atractores o repulsores? Usar `NestList[ ]` para ver qué ocurre con la iteración funcional para distintas elecciones del punto inicial. Tomar, por ejemplo, $c = 0.5$ y comentar los resultados.

(c) Cuando $1 < c < 3$, ¿de qué tipo son los puntos fijos? Comprobar gráficamente que para $c = 2$ la iteración funcional converge rápidamente al punto fijo 0.5

(d) Tomar $c = 3$ y usar `NestList[ ]`. Comprobar que después de algunas iteraciones, los términos sucesivos oscilan entre dos valores diferentes, y quedan atrapados en un bucle sin fin. Es lo que se llama un **ciclo periódico de orden 2**. ¿Qué ocurre cuando $c = 3.5$? ¿Hay un ciclo periódico? ¿De qué orden?.

(e) Comprobar que cuando $c = 4$ las iteraciones son aleatorias y la situación se vuelve completamente caótica. Hacer un gráfico con

$$\texttt{ListPlot[ ]}$$

para esta situación caótica.

- Gran parte de las preguntas planteadas en el ejemplo han sido contestadas en la teoría. Algunas otras pueden razonarse de la siguiente manera.

$$\texttt{Plot[Evaluate[Table[f[c, x], \{c, 0, 4\}]], \{x, 0, 1\}, PlotStyle} \rightarrow \texttt{RGBColor[1, 0, 0]]}$$

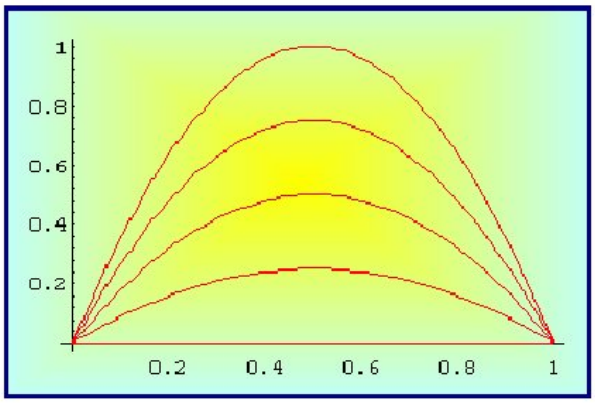

Figura 1.2: $f[c,x] = cx(1-x)$, $c = 0, 1, 2, 3, 4$.

Comenzamos con un punto x entre 0 y 1, y calculamos $f[c,x]$. Como $f[c,x]$ sigue estando entre 0 y 1, tiene sentido escribir $f[c, f[c,x]]$, y así sucesivamente. Este proceso, como sabemos, es conocido con el nombre de iteración y podemos realizarlo con el programa `Mathematica`® utilizando la orden `NestList`. Empezamos fijando para c el valor de 2 y calculamos las veinte primeras iteraciones del punto 0.25

```
g[x_] := f[2, x]
NestList[g, 0.25, 20]
```

{0.25, 0.375, 0.46875, 0.498047, 0.499992, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5}

Resultado 1: Como puede observarse, la órbita converge al punto de equilibrio 0.5.

Si cambiamos el valor de la semilla el resultado sigue siendo el mismo.

```
g[x_] := f[2, x]
NestList[g, 0.7, 20]
```

{ 0.7, 0.42, 0.4872, 0.499672, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 }

Supongamos que ahora el valor del parámetro es $c = 3$.

```
g[x_] := f[3, x]
NestList[g, 0.45, 20]
NestList[g, 0.75, 20]
```

{0.45, 0.7425, 0.573581, 0.733757, 0.586072, 0.727775, 0.594356, 0.723291, 0.600424, 0.719745, 0.605136, 0.716839, 0.608942, 0.714395, 0.612105, 0.712298, 0.614789, 0.71047, 0.617107, 0.708858, 0.619135}

{ 0.75, 0.5625, 0.738281, 0.579666, 0.73096, 0.589973, 0.725715, 0.597158, 0.721681, 0.602573, 0.718436, 0.606857, 0.715745, 0.610362, 0.71346, 0.61330, 0.71148, 0.6158, 0.709757, 0.618006, 0.708224 }

Resultado 2: Ahora las órbitas tienden a los puntos 0.61 y 0.70. Es decir, presenta un comportamiento periódico de orden dos.

Si aumentamos el valor del parámetro $c = 3.5$.

```
g[x_] := f[3.5, x]
NestList[g, 0.15, 20]
NestList[g, 0.83, 20]
```

{ 0.15, 0.44625, 0.864888, 0.408998, 0.846015, 0.455957, 0.868211, 0.400473, 0.84033, 0.469614, 0.871768, 0.391259, 0.83361, 0.48545, 0.87426, 0.384754, 0.82851, 0.49727, 0.874974, 0.382881, 0.826991 }

{ 0.83, 0.49385, 0.874868, 0.38316, 0.827219, 0.500246, 0.875, 0.382813, 0.826935, 0.500897, 0.874997, 0.38282, 0.826941, 0.50088, 0.87499, 0.38282, 0.82694, 0.50088, 0.874997, 0.38282, 0.826941 }

Resultado 3. Ahora los valores se repiten cada cuatro veces. Si aumentamos el valor de c, podríamos ver que se repiten cada ocho, dieciséis, treinta y dos , ... veces. Este proceso donde cada periodo duplica al anterior, culmina hasta llegar a un valor de c que se conoce con el nombre de **constante de Feigenbaum** (aproximadamente $c = 3.5699456718...$).

A continuación repetiremos el proceso anterior, pero variando el valor del parámetro, que se inicia con $c = 0.223$.

```
logistic[n_Integer] := Module[{f, t, x}, f = Compile[{x, t},
Evaluate[(3 + t/n) * x * (1 - x)]];
FoldList[f, 0.223, Range[n]]]
```

La celda próxima representa de forma sonora el efecto del barrido del parámetro c. Puede oírse como el período del sonido va doblándose hasta llegar al caos.

```
ListPlay[logistic[8000]];
```

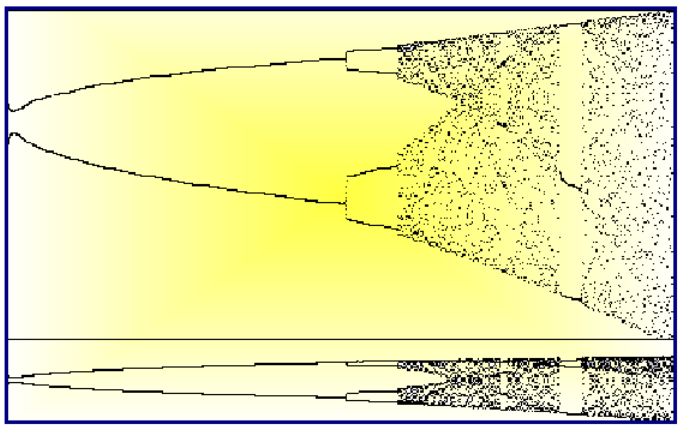

Figura 1.3: Diagrama de bifurcación de $f(x) = cx(1 - x)$.

1.4. Modelo de Ricker

Es un modelo discreto frecuentemente utilizado en dinámica de poblaciones para estudiar su evolución y viene definido por la ecuación:

$$N_{t+1} = f(N_t) = N_t e^{r\left(1-\dfrac{N_t}{k}\right)}, \qquad r\,, k \in \mathbb{R}^+, \quad t = 0,\, 1,\, 2, \cdots .$$

Nos proponemos encontrar y analizar los puntos de equilibrio no triviales.

Para este modelo discreto no lineal, la función que lo define es $f(x) = xe^{r\left(1-\frac{x}{k}\right)}$. Los puntos de equilibrio se obtienen al resolver la ecuación $f(x) = x$, cuyos valores son $x_1^* = 0$ y $x_2^* = k$. Para poder clasificarlos, es necesario encontrar la derivada de la función $f(x)$, es decir

$$f'(x) = e^{r\left(1-\frac{x}{k}\right)}\left(1 - \frac{xr}{k}\right) .$$

Ahora debemos sustituir el punto de equilibrio no trivial en $f'(x)$. Al ser $f'(k) = 1-r$, entonces el $x_2^* = k$ será un punto de equilibrio estable si $|1 - r| < 1$, y para ello $0 < r < 2$.

La siguiente cuestión importante es saber que le sucede al modelo cuando se pierde la estabilidad. Lo primero que podemos pensar es que la población se extinguirá. Para ver su comportamiento podríamos simular la dinámica de la población del modelo para diferentes valores del parámetro r.

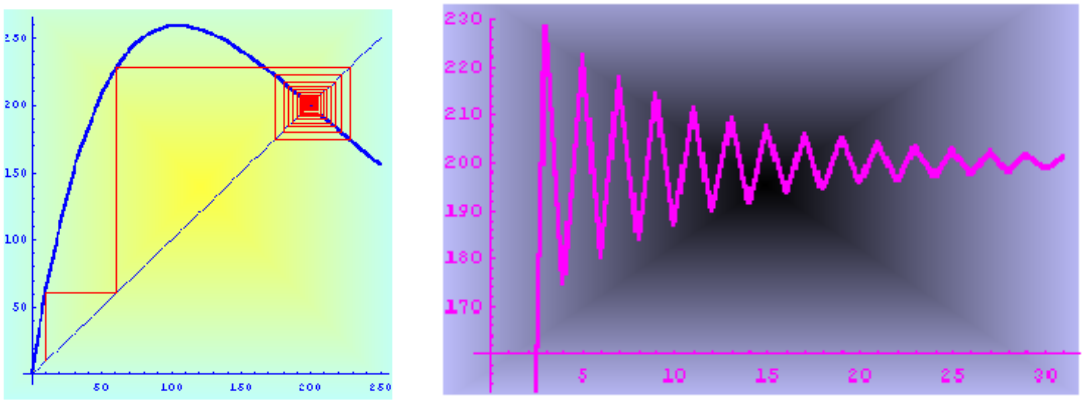

Figura 1.4: Modelo de *Ricker* r=1.9; k=200.

Tomamos como $k = 200$ y cambiamos el valor del parámetro r. La solución N_t tiende de forma monótona al punto de equilibrio si $r = 0.5$, o bien oscilando si $r = 1.9$.

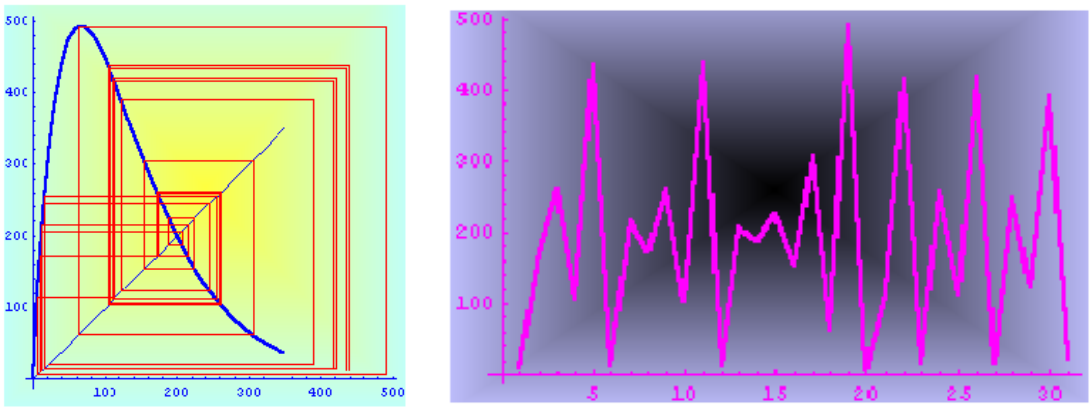

Figura 1.5: Modelo de *Ricker* r=3; k=200

También puede tender a un ciclo límite de periodo dos si $r = 2.3$ o bien tener un comportamiento caótico cuando $r = 3$. Por supuesto, en este caso podemos dibujar su diagrama de bifurcación, que presenta unas características muy parecidas al modelo logístico de *May*.

La dinámica caótica se comporta de manera parecida al ruido estocástico, sin embargo el modelo es absolutamente determinista. Los modelos caóticos, de hecho, se utilizan para generar números aleatorios con ordenadores. Una de las cuestiones que se discuten con frecuencia en la literatura ecológica es si existe realmente el caos en la dinámica de poblaciones. El mayor argumento en favor del caos es que cuando los parámetros del modelo se ajustan a series temporales conocidas de dinámica de poblaciones, entonces la dinámica de ese modelo con estos parámetros es caótica. Otros tipos de argumentos se basan en intentar separar la dinámica caótica del ruido estocástico. Sin embargo, detectar el caos es bastante difícil pues:

- No existen evidencias de que el modelo sea el correcto, ya que es normal que el que diseñemos ignore muchos procesos ecológicos. Se necesita utilizar múltiples modelos para detectar el caos.
- El intervalo de confianza para los valores de los parámetros es usualmente muy grande y esto hace cambiar el comportamiento de la dinámica del modelo (período, caos,...)
- Las series temporales en dinámica de poblaciones son generalmente no lo suficientemente grandes para separar el comportamiento caótico del ruido estocástico.

Por todas estas razones, algunos autores sostienen que probablemente el caos es un fenómeno extraño en dinámica de poblaciones.

Si dibujamos N_{t+1} en función de N_t, observamos que para valores pequeños de la población, entonces ésta aumentará en el próximo año. Por el contrario, para niveles altos de población los mecanismos dependientes de la densidad (competencia) reducen el tamaño de la población en el año próximo.

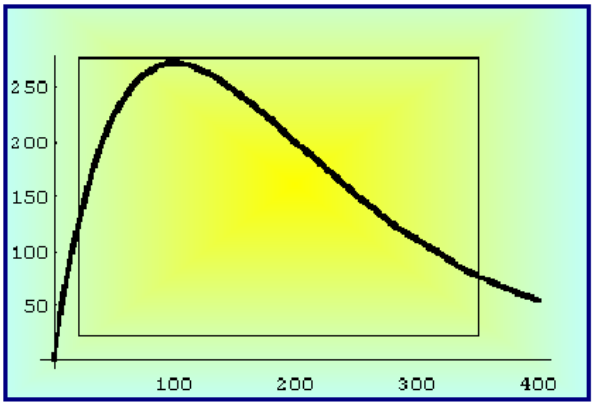

Figura 1.6: Gráfica de $N_{t+1} = f(N_t)$.

1.4.1. Puntos de equilibrio del modelo de Ricker con $r = 0.5$

A continuación incluimos la simulación para diferentes valores del parámetro r utilizando el programa Mathematica®.

$$\texttt{f[x\_]} := \texttt{x} * \texttt{Exp[0.5} * (1 - \texttt{x/200})]$$

- Cálculo de la órbita para el valor inicial $x_0 = 10$.

  $$\texttt{iters} = \texttt{NestList[f, 10, 20]}$$

  {10, 16.0801, 25.467, 39.3981, 58.8635, 83.769, 112.016, 139.575, 162.335, 178.36, 188.277, 193.87, 196.867, 198.415, 199.203, 199.6, 199.8, 199.9, 199.95, 199.97, 199.987, 199.994, 199.997, 199.998, 199.999, 200.}

- Para dibujar el diagrama de Cobweb empezamos construyendo la órbita

  ```
  gi = ListPlot[Partition[Flatten[Transpose[iters, iters]], 2, 1],
  PlotJoined → True, DisplayFunction → Identity, PlotStyle →
  RGBColor[1, 0, 0]]
  ```

  y a continuación representamos la función $f(x)$ que nos define el modelo y la bisectriz del primer cuadrante.

  ```
  fg = Plot[{f[x], x}, {x, 0, 250}, PlotStyle → {{Thickness[0.01],
  Thickness[0.01]}, {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}},
  DisplayFunction → Identity]
  ```

  Por último, superponemos los dos gráficos y construimos la función $N(t)$.

  ```
  grafica1 = Show[fg, gi, AspectRatio → 1, DisplayFunction → $Display
  Function, Background → RGBColor[1, 1, 0]]ListPlot[iters,
  PlotStyle → PointSize[0.02], Background → RGBColor[1, 0.5, 0.2],
  AspectRatio → 1, AxesLabel → {"Tiempo t", "N(t)"}]
  ```

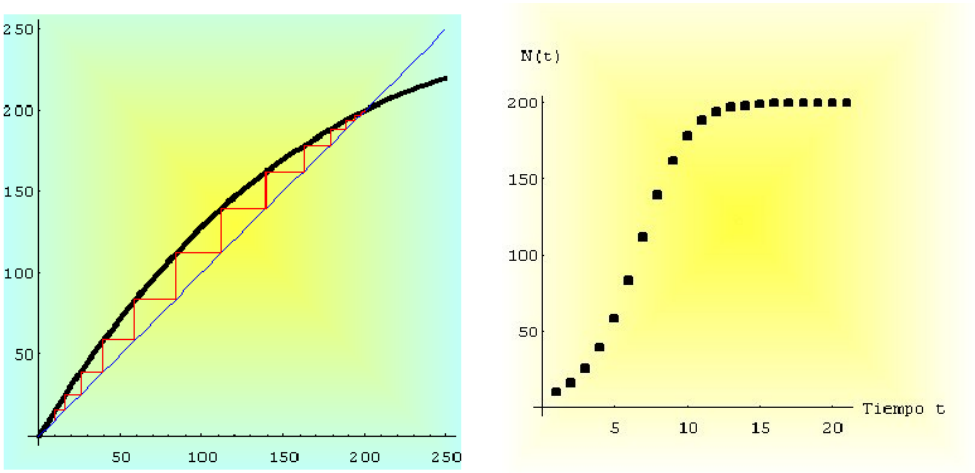

Figura 1.7: Diagrama de Cobweb y Evolución de la población.

- **Comentario:** La población tiende de manera monótona creciente al punto de equilibrio 200 cuando el tiempo tiende hacia infinito.

1.4.2. Puntos de equilibrio del modelo de Ricker con $r = 1.9$

$$\texttt{f[x\_]} := \texttt{x} * \texttt{Exp}[1.9 * (1 - \texttt{x}/200)]$$

- Cálculo de la órbita para el valor inicial $x_0 = 10$.

$$\texttt{iters} = \texttt{NestList}[\texttt{f}, 10, 20]$$

{ 10, 60.7997, 228.146, 174.617, 222.234, 179.919, 217.735, 183.975, 214.228, 187.144, 211.455, 189.652, 209.243, 191.653, 207.469, 193.258, 206.041, 194.549, 204.889, 195.591, 203.958 }

- Para dibujar el diagrama de Cobweb empezamos construyendo la órbita

  $\texttt{gi} = \texttt{ListPlot}[\texttt{Partition}[\texttt{Flatten}[\texttt{Transpose}[\texttt{iters}, \texttt{iters}]], 2, 1], \texttt{PlotJoined} \to \texttt{True}, \texttt{DisplayFunction} \to \texttt{Identity}, \texttt{PlotStyle} \to \texttt{RGBColor}[1, 0, 0]]$

  y a continuación representamos la función $f(x)$ que nos define el modelo y la bisectriz del primer cuadrante.

  $\texttt{fg} = \texttt{Plot}[\{\texttt{f}[\texttt{x}], \texttt{x}\}, \{\texttt{x}, 0, 250\}, \texttt{PlotStyle} \to \{\{\texttt{Thickness}[0.01], \texttt{Thickness}[0.01]\}, \{\texttt{RGBColor}[1, 0, 0], \texttt{RGBColor}[0, 0, 1]\}\}, \texttt{DisplayFunction} \to \texttt{Identity}]$

  Por último, superponemos los dos gráficos y construimos la función $N(t)$.

  $\texttt{grafica1} = \texttt{Show}[\texttt{fg}, \texttt{gi}, \texttt{AspectRatio} \to 1, \texttt{DisplayFunction} \to \$\texttt{Display Function}, \texttt{Background} \to \texttt{RGBColor}[1, 1, 0]]\texttt{ListPlot}[\texttt{iters}, \texttt{PlotStyle} \to \texttt{PointSize}[0.02], \texttt{Background} \to \texttt{RGBColor}[1, 0.5, 0.2], \texttt{AspectRatio} \to 1, \texttt{AxesLabel} \to \{\text{"Tiempo t"}, \text{"N(t)"}\}]$

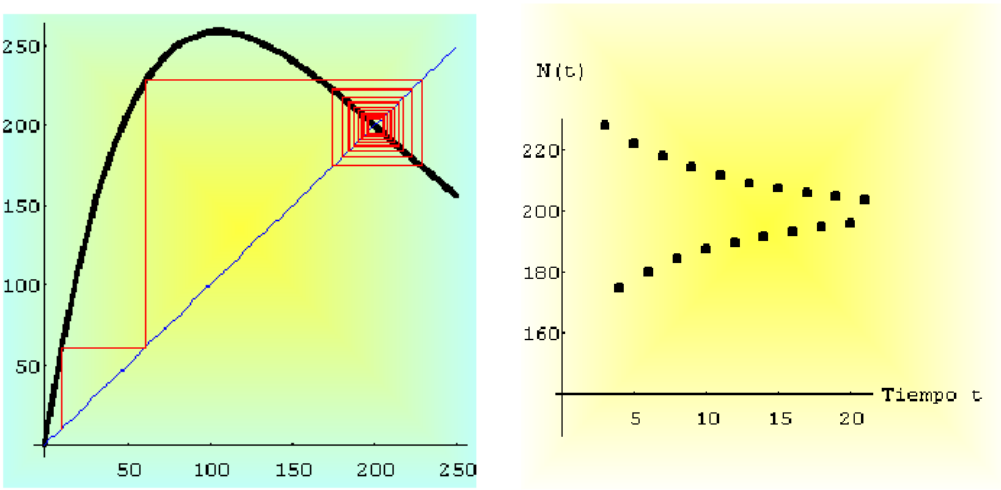

Figura 1.8: Diagrama de Cobweb y Evolución de la población.

- **Comentario:** La población tiende de una forma oscilatoria al punto de equilibrio 200 cuando el tiempo tiende hacia infinito.

1.4.3. Puntos de equilibrio del modelo de Ricker con $r = 2.3$

$$\texttt{f[x\_]} := \texttt{x} * \texttt{Exp}[2.3 * (1 - \texttt{x}/200)]$$

- Cálculo de la órbita para el valor inicial $x_0 = 10$.

$$\texttt{iters} = \texttt{NestList}[\texttt{f}, 10, 20]$$

{ 10, 88.9065, 318.99, 81.1884, 318.335, 81.6342,318.447, 81.5583, 318.428, 81.5708, 318.431, 81.5687, 318.431, 81.5691, 318.431, 81.569, 318.431, 81.569, 318.431, 81.569, 318.431, 81.569, 318.431, 81.569, 318.431}

- Diagrama Cobweb

```
gi = ListPlot[Partition[Flatten[Transpose[iters, iters]], 2, 1],
PlotJoined → True, DisplayFunction → Identity, PlotStyle →
RGBColor[1, 0, 0]]
fg = Plot[{f[x], x}, {x, 0, 350}, PlotStyle → {{Thickness[0.01],
Thickness[0.01]}, {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}},
DisplayFunction → Identity]
grafica1 = Show[fg, gi, AspectRatio → 1, DisplayFunction → $Display
Function, Background → RGBColor[1, 1, 0]]ListPlot[iters,
PlotStyle → PointSize[0.02], Background → RGBColor[1, 0.5, 0.2],
AspectRatio → 1, AxesLabel → {"Tiempo t", "N(t)"}]
```

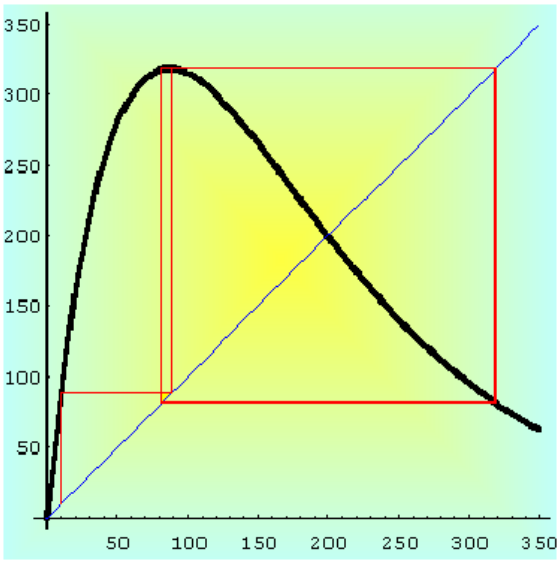

Figura 1.9: Diagrama de Cobweb y Evolución de la población.

- **Comentario:** La población tiene un comportamiento periódico de orden dos cuando el tiempo tiende hacia infinito.

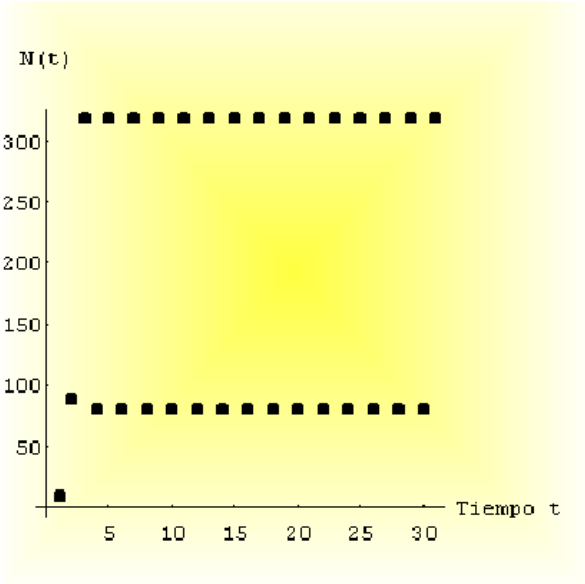

Figura 1.10: Representación gráfica de la órbita.

Podemos observar mejor el comportamiento periódico si unimos los puntos correspondientes a la población en el tiempo t.

$$\texttt{grafica2} = \texttt{ListPlot}[\texttt{iters}, \texttt{PlotJoined} \rightarrow \texttt{True},$$
$$\texttt{Background} \rightarrow \texttt{RGBColor}[0.8, 1, 0], \texttt{PlotStyle} \rightarrow \texttt{Thickness}[0.01]]$$

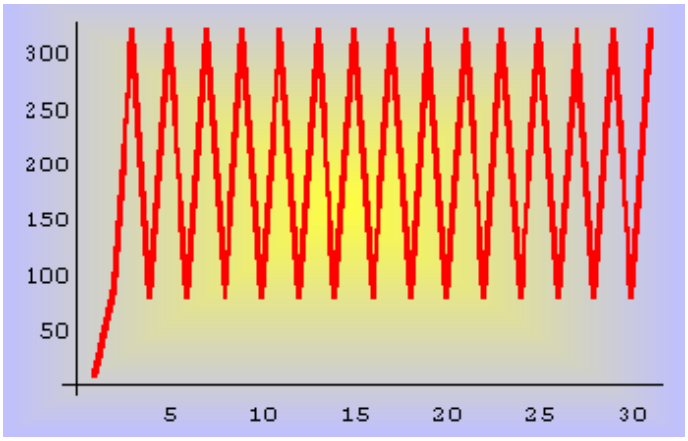

Figura 1.11: Representación gráfica de la órbita.

1.4.4. Puntos de equilibrio del modelo de Ricker con $r = 3$

$$\texttt{f[x\_]} := \texttt{x} * \texttt{Exp}[3 * (1 - \texttt{x}/200)]$$

- Cálculo de la órbita para el valor inicial $x_0 = 10$.

$$\texttt{iters} = \texttt{NestList}[\texttt{f}, 10, 24]$$

{10., 172.878, 259.672, 106.096, 433.939, 12.9853, 214.656, 172.293, 261.072, 104.451, 437.885, 12.3504, 206.114, 188.052, 224.964, 154.699, 305.208, 62.9838, 491.824, 6.17657, 113.082, 416.498, 16.1903, 255.075 }

- Diagrama de Cobweb

```
gi = ListPlot[Partition[Flatten[Transpose[iters, iters]], 2, 1],
PlotJoined → True, DisplayFunction → Identity, PlotStyle →
RGBColor[1, 0, 0]]
fg = Plot[{p[x], x}, {x, 0, 350}, PlotStyle → {{Thickness[0.01],
Thickness[0.01]}, {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}},
DisplayFunction → Identity]
grafica1 = Show[fg, gi, AspectRatio → 1, DisplayFunction → $Display
Function, Background → RGBColor[1, 1, 0]]ListPlot[iters,
PlotStyle → PointSize[0.02], Background → RGBColor[1, 0.5, 0.2],
AspectRatio → 1, AxesLabel → {"Tiempo t", "N(t)"}]
```

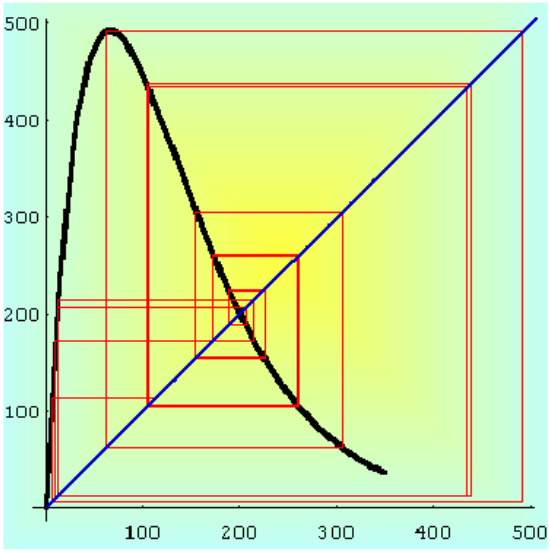

Figura 1.12: Diagrama de Cobweb y Evolución de la población.

- **Comentario:** La población tiene un comportamiento caótico, cuando el tiempo tiende hacia infinito.

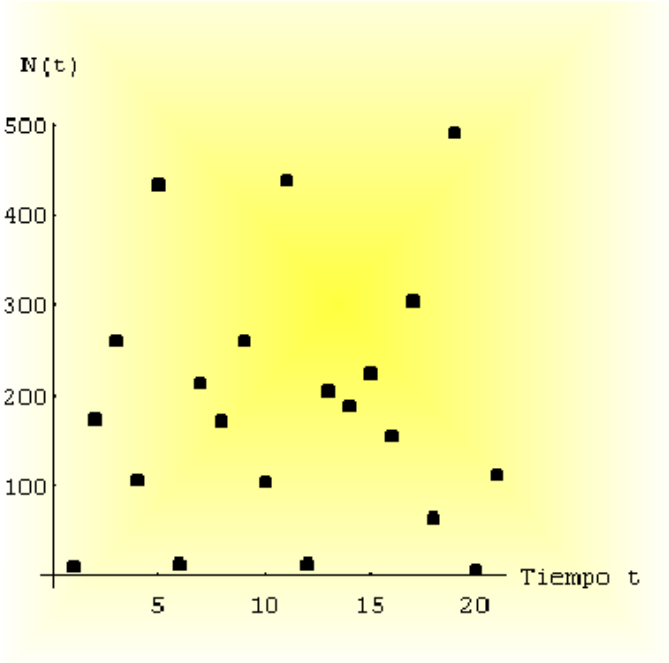

Figura 1.13: Representación gráfica de la órbita.

Podemos observar mejor el comportamiento caótico si unimos los puntos correspondientes a la población en el tiempo t.

$$\texttt{grafica2} = \texttt{ListPlot}[\texttt{iters}, \texttt{PlotJoined} \rightarrow \texttt{True},$$
$$\texttt{Background} \rightarrow \texttt{RGBColor}[0.8, 1, 0], \texttt{PlotStyle} \rightarrow \texttt{Thickness}[0.01]]$$

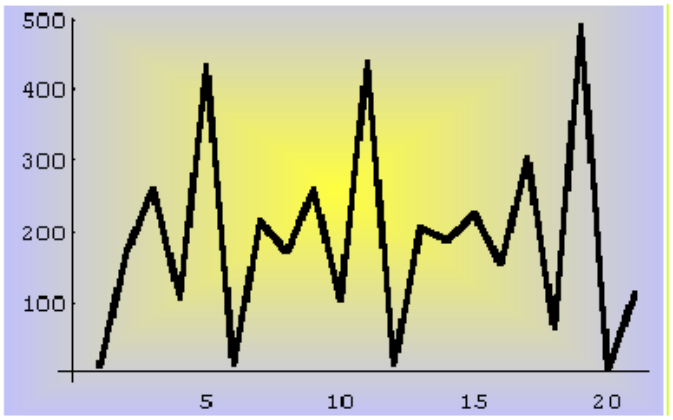

Figura 1.14: Representación gráfica de la órbita

1.4.5. Diagrama de bifurcación del modelo de Ricker

Un método muy útil para entender el comportamiento cualitativo de las soluciones de un sistema dinámico discreto como el que estamos analizando, es construir su diagrama de bifurcación, ya que pueden sufrir cambios en la estructuras de sus órbitas cuando variamos los valores de los parámetros que intervienen en el modelo. Esas modificaciones dan lugar al nacimiento o a la muerte de puntos fijos y ciclos o trasformaciones en el tipo de las órbitas. A estos cambios se le conocen con el nombre de bifurcaciones.

$$\texttt{f[x\_]} := 3 * \texttt{Exp}[3 * (1 - \texttt{x}/200)]$$

Dibujamos la función que nos define nuestro modelo (en este caso, hemos tomado como valor del parámetro $r = 3$).

$$\texttt{Plot}[\texttt{f[x]}, \{\texttt{x}, 0, 250\}, \texttt{PlotStyle} \rightarrow \{\texttt{Thickness}[0.01], \texttt{RGBColor}[1, 0, 0]\}, \texttt{Background} \rightarrow \texttt{RGBColor}[1, 0.6, 0.3]]$$

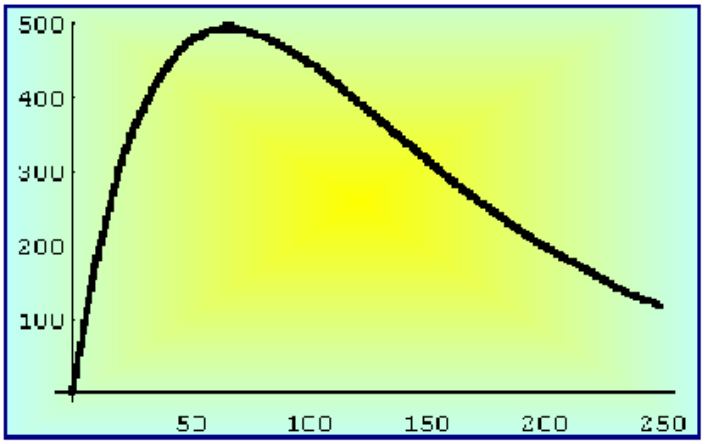

Figura 1.15: Representación de la función $f(x)$ que define al modelo.

El diagrama es una gráfica (en el plano Ory) de las líneas de fase cercanas a un valor de bifurcación, que nos permite ver los cambios experimentados por las líneas de fase, cuando el parámetro pasa por este valor.

```
logistc[n_Integer] := Module[{f, t, x}, f = Compile[{x, t},
Evaluate[x * Exp[(3 + t/n) * (1 - x/200)]]];
FoldList[f, 0.223, Range[n]]]
Null
```

Además podemos escuchar el sonido de este estado caótico con la siguiente instrucción:

```
b = ListPlay[logistc[8000]];
```

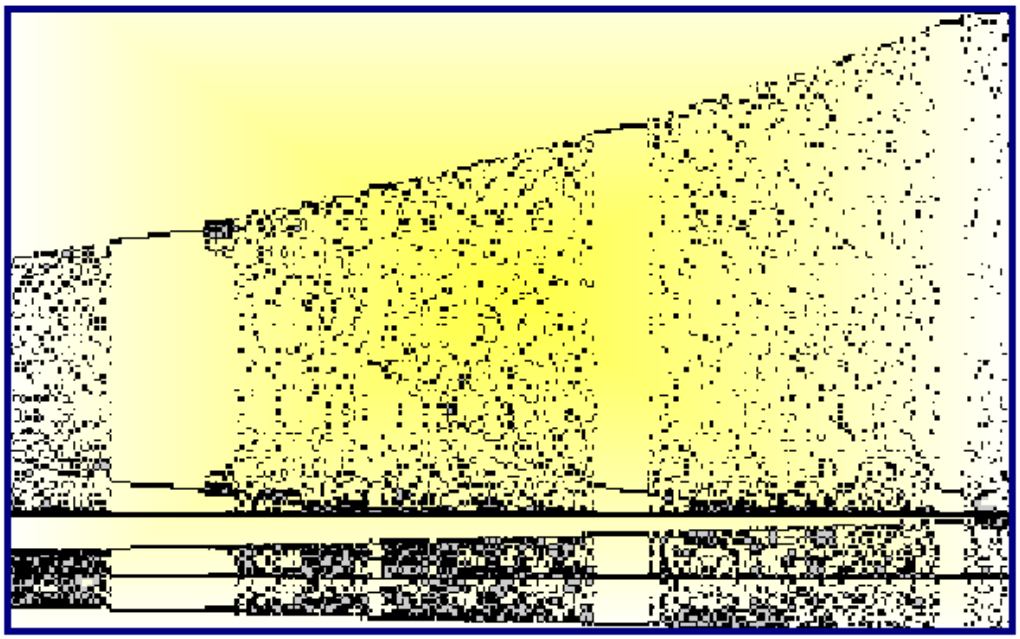

Figura 1.16: Diagrama de bifurcación de $f(x)$.

Nota: Para trazar el diagrama de bifurcación representamos en el eje de abscisas los diferentes valores del parámetro r. Vamos dando a r un número elevado de valores (no necesariamente números enteros) y dibujamos la línea de fase correspondiente para cada uno de los valores del parámetro. De esta manera obtenemos una línea paralela al eje de ordenadas que corta al eje de abscisas en el valor r. Si miramos el dibujo de izquierda a derecha, observamos como evoluciona la línea de fase a través de la bifurcación.