

# Introducción al Modelamiento Matemático

Notas de Clase

Leon Escobar Diaz  
Karem Rodríguez

Facultad de Ciencias Naturales y Exactas  
Universidad del Valle

## Índice general

<b>1. Elementos de programación científica</b>	<b>1</b>
1.1. Generalidades de los sistemas de cómputo . . . . .	1
1.1.1. Orígenes de los computadores: El transistor . . . . .	1
1.1.2. Elementos básicos de un computador . . . . .	2
1.1.3. El sistema operativo . . . . .	4
1.2. Programas computacionales . . . . .	4
1.2.1. ¿Qué es un programa computacional? . . . . .	4
1.2.2. Lenguajes interpretados y compilados . . . . .	5
1.2.3. Ambientes de desarrollo para escribir programas . . . . .	7
1.3. Desarrollo de algoritmos . . . . .	8
1.3.1. Diagramas de flujo y pseudo-códigos . . . . .	9
1.3.2. Entrada y salida de información: Uso de variables . . . . .	11
1.4. Elementos básicos de programación . . . . .	12
1.4.1. Estructuras de control condicionales . . . . .	13
1.4.2. Estructuras de ciclos . . . . .	15
1.4.3. Listas y arreglos . . . . .	20
1.4.4. Funciones . . . . .	22
1.5. Introducción a la preparación de documentos mediante LaTeX . . . . .	24
1.5.1. Estructura general del código LaTeX de un documento . . . . .	25
1.5.2. El preámbulo del documento . . . . .	26
1.5.3. El cuerpo del documento . . . . .	27
1.5.4. El entorno matemático . . . . .	29
1.5.5. El entorno de figuras . . . . .	33
1.5.6. El entorno de las referencias bibliográficas . . . . .	34
1.6. Ejercicios propuestos . . . . .	36
<b>2. Generalidades del modelamiento matemático</b>	<b>41</b>
2.1. Que es un modelo matemático . . . . .	41
2.2. Clasificación de los modelos matematicos . . . . .	41

---

2.2.1.	Modelos Determinísticos:	42
2.2.2.	Modelos probabilísticos:	42
2.2.3.	Tipos de modelos: continuos o discretos	43
2.3.	Pasos sugeridos para hacer un modelo matematico	44
2.3.1.	Ejemplos de construccion de modelos	44
2.4.	Los datos y los modelos matemáticos	47
<b>3.</b>	<b>Manejo básico de datos</b>	<b>49</b>
3.1.	Proporcionalidad de datos	49
3.1.1.	Proporcionalidad directa	50
3.1.2.	Proporcionalidad inversa	52
3.1.3.	Proporcionalidades ocultas	53
3.2.	Ajuste de datos	56
3.2.1.	Gráficas	56
3.2.2.	Mínimos cuadrados	58
3.3.	Ejercicios	70
<b>4.</b>	<b>Modelos deterministas</b>	<b>78</b>
4.1.	Modelos deterministas en ecuaciones en diferencias	78
4.1.1.	Métodos cualitativos y computacionales	90
4.1.2.	Aplicaciones	95
4.2.	Modelos basados en ecuaciones diferenciales	101
4.2.1.	Estudio de modelos basados en ecuaciones diferenciales	104
4.2.2.	Ejemplos de sistemas que se modelan con ecuaciones diferenciales	107
4.2.3.	Métodos cualitativos y computacionales para EDOs	110
4.2.4.	Algunos Métodos Numéricos para resolver EDOs	115
4.2.5.	Aplicaciones	117
4.3.	Ejercicios	122
<b>5.</b>	<b>Modelos probabilísticos</b>	<b>134</b>
5.1.	Fundamentos de probabilidad	134
5.1.1.	Axiomas de la probabilidad	136
5.1.2.	Combinación de eventos y probabilidades	137
5.1.3.	Probabilidades como porcentajes	137
5.2.	Cálculo de probabilidades	138
5.2.1.	Enfoque clásico	138
5.2.2.	Enfoque frecuentista	140
5.3.	Variables aleatorias	142
5.3.1.	Ejemplo caso discreto	142
5.3.2.	Ejemplo caso continuo	143
5.4.	Función de masa y densidad de probabilidad	143
5.4.1.	Función de masa de probabilidad	143
5.5.	Algunas distribuciones de probabilidad discreta	146
5.5.1.	Distribución de probabilidad uniforme discreta	146
5.5.2.	Distribución binomial	146
5.5.3.	Distribución de Poisson	148

---

5.6.	Función de verosimilitud . . . . .	149
5.6.1.	El método de máxima verosimilitud . . . . .	149
5.6.2.	Estimación de parámetros . . . . .	149

## Elementos de programación científica

Comenzaremos por mencionar algunas generalidades del funcionamiento de los computadores modernos. Luego, introduciremos la noción de programa computacional y mencionaremos algunos de los más usados actualmente haciendo especial énfasis en el lenguaje de programación *Python*, el cual será el lenguaje que más usaremos a lo largo de este curso. Como herramienta fundamental para diseñar programas computacionales, introducimos el concepto de algoritmo y presentamos algunas pautas para representarlos de forma gráfica o para escribirlos a manera de pseudo-código. Después introduciremos las herramientas fundamentales para escribir programas, las cuales comprenden los ciclos, los condicionales, las listas y las funciones. Finalmente, hacemos una breve introducción a la escritura de documentos científicos empleando el lenguaje de programación LaTeX.

### 1.1. Generalidades de los sistemas de cómputo

Antes de introducir un lenguaje de programación y comenzar a programar, es muy recomendable conocer de forma general los componentes de un computador y sus funciones. A continuación mencionaremos de forma breve algunos de los más relevantes.

#### 1.1.1. Orígenes de los computadores: El transistor

Gracias al desarrollo de la mecánica cuántica a principios del siglo XX, los científicos pudieron comprender y desarrollar un artefacto que permitió el desarrollo de toda la electrónica: el transistor. Los transistores son un ensamble de ciertos metales, llamados semiconductores, de tal forma que permiten entregar una determinada señal de salida en respuesta a una cierta señal de entrada cuando son sometidos a la acción de una corriente eléctrica.

De forma más simple, un transistor puede ser interpretado como una “llave de paso”, el cual puede cortar o dejar pasar corriente según su configuración. Adicionalmente, un transistor también puede amplificar corrientes que pasan a través de él. Este particular funcionamiento es algo que únicamente puede ser explicado con claridad en el marco de la mecánica cuántica.

Los circuitos eléctricos se conocían desde el siglo XIX y se explicaban satisfactoriamente gracias a la electrodinámica clásica de Maxwell y Faraday. Sin embargo, la llegada de la mecánica cuántica y del transistor, permitió el desarrollo de la electrónica moderna, mediante el uso compuertas lógicas, capacitores y unidades de almacenamiento. Actualmente, los transistores son agrupados en lo que conocemos como chips. Usualmente un chip contiene miles o inclusive millones de transistores y permite que podamos usar corrientes para inducir un campo magnético positivo o negativo en pequeñas regiones de una placa conductora de una forma controlada. Una cierta configuración de magnetizaciones positivas y negativas en una placa conductora puede ser interpretada como información digital.

### 1.1.2. Elementos básicos de un computador

#### La tarjeta madre o placa

La tarjeta madre es el circuito electrónico principal sobre el cual se ensamblan todos los componentes electrónicos de un computador a través de sus *sockets* o puertos, como por ejemplo, la fuente de voltaje, el procesador, el disco duro, la memoria, etc.

#### El disco duro

Los discos duros son dispositivos de almacenamiento de datos en los que podemos almacenar cualquier tipo de información digital. Como dijimos anteriormente, esta información digital no es más que una cierta configuración de magnetizaciones positivas y negativas en una placa. Esta placa es precisamente el disco duro. La información digital almacenada en el disco se guarda en pequeños espacios de almacenamiento medidos en bits (*binary digit*). Los bits son precisamente las magnetizaciones positivas o negativas en el disco duro que pueden ser interpretados como señales de 1 o 0, lo cual da paso de forma natural al almacenamiento de información en forma binaria (de allí su nombre).

Para entender de forma general como funciona un disco duro, imagine un toca-disco analógico. En él, una gran aguja pasa por encima de un disco en rotación detectando los cambios del relieve del disco. Pues bien, un disco duro funciona de la misma forma, solo que la aguja en este caso detecta las variaciones de los pequeños campos magnéticos en el disco, es decir, detecta los bits almacenados.

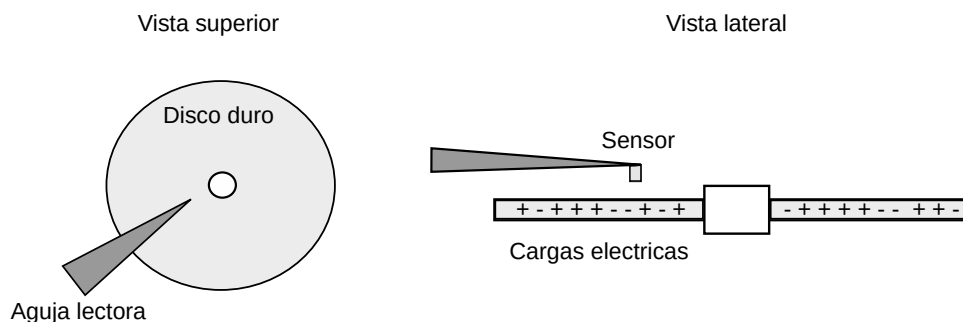


Figura 1.1: Esquema de un disco duro.

A pesar de que el bit es la unidad básica para representar información en el disco duro, usualmente se utilizan otras unidades para medir la cantidad de información digital almacenada, las cuales listamos a continuación en la siguiente tabla:

Unidad de Almacenamiento	Cantidad
1 bytes	8 bits
1 Kilobytes	1024 bytes
1 Megabytes	1024 Kilobytes
1 Gigabytes	1024 Megabytes
1 Terabyte	1024 Gigabytes

### El procesador

El procesador es el cerebro del computador, y es justamente el componente electrónico que procesa y ejecuta toda la información almacenada en el disco duro. Como es de esperar, el procesador tiene una velocidad limitada de procesamiento de la información, la cual usualmente se mide en Giga-Hertz. Esta unidad da una idea básica de cuántas operaciones por segundo puede realizar el procesador. Por ejemplo, si un procesador es de 3 Giga-Hertz, significa que éste puede realizar aproximadamente  $10^9$  operaciones por segundo (la palabra “Giga” es la que da el valor de la potencia de 9). Por lo tanto, entre más grande sea su velocidad de procesamiento, mucho más rápido será la ejecución de los programas. Sin embargo, también es muy relevante tener en cuenta el tamaño de palabra del procesador y si éste es multi-núcleo. Lo primero, da una idea de qué tan grande es la información con la que puede trabajar el procesador por operación; y lo segundo de si este procesador puede ejecutar varios procesos independientes de forma simultánea.

Como ejemplo de lo anterior, digamos que tenemos un procesador de 64 bits octa-core. Es decir, nuestro procesador tiene 8 núcleos<sup>1</sup> que tiene un tamaño de palabra de 64 bits. Esto quiere decir que nuestro procesador puede ejecutar 8 tareas distintas simultáneamente (o en paralelo), además de que cada núcleo tiene un tamaño de palabra de 64 bits.

### La memoria RAM

La memoria RAM (*Random Access Memory*) es donde el computador guarda la información que se está utilizando en el momento, lo cual ayuda que el procesador pueda ejecutar programas de forma más rápida. Sin embargo, la información almacenada en la memoria RAM es temporal porque los datos y programas solo permanecen en ella mientras que el computador permanezca encendido.

Usualmente, en la memoria RAM se almacena la información para poder ejecutar el sistema operativo, así como los programas en ejecución. Es por tanto, que resulta natural pensar que a mayor cantidad de memoria RAM, mayor número de tareas podrá ejecutar un computador

---

<sup>1</sup>Core es una palabra inglesa que significa núcleo.

simultáneamente. Sin embargo, debe tenerse siempre presente que la velocidad de procesamiento del procesador también es esencial para asegurar la buena ejecución de varias tareas de forma simultánea.

### 1.1.3. El sistema operativo

Un sistema operativo es una infraestructura o conjunto de “programas informáticos” que le permiten a un usuario, entre otras cosas, administrar, almacenar y procesar información digital. Generalmente, un sistema operativo viene acompañado de una interfaz gráfica que facilita tanto la administración de la información y periféricos físicos (también llamados *hardware*), como la ejecución de los programas. Sin embargo, esto no es completamente necesario ya que todos los procesos del sistema operativo pueden controlarse a través de una *terminal* mediante la implementación de comandos<sup>2</sup>.

Existe una gran diversidad de sistemas operativos. Entre los más usados, que incluyen interfaces gráficas, se encuentran Windows y los sistemas Unix (IOS y Linux). Algo importante a tener en cuenta es que debido a su arquitectura, en cada sistema operativo solo se pueden ejecutar los programas que estén diseñados específicamente para funcionar dentro de ese sistema operativo en particular. Por esta razón, un programa que funcione en un sistema operativo no necesariamente podrá ser ejecutado en otro sistema operativo, a menos que éste sea “multiplataforma”. Por lo anterior, es recomendable escoger el sistema operativo de acuerdo a los tipos de programas que el usuario desee ejecutar.

## 1.2. Programas computacionales

### 1.2.1. ¿Qué es un programa computacional?

De forma muy general, puede definirse un programa, aplicación o software, como una lista de tareas escritas de forma estructurada que pueden ser ejecutadas por un computador. Generalmente, esta lista de tareas se suele llamar *código fuente del programa*, y el conjunto de comandos utilizado para escribir las instrucciones de la lista se denomina *lenguaje de programación*.

Generalmente, los lenguajes de programación se basan en una serie de “comandos” o “palabras claves” que una persona puede escribir para indicarle a un computador qué tareas ejecutar. Naturalmente, estos comandos están escritos mediante la utilización de caracteres propios de un determinado alfabeto, ya sea el alfabeto latino, cirílico, griego, chino, árabe etc. Sin embargo, el computador por sí mismo no puede comprender estos caracteres debido que su “lenguaje nativo” no está basado en ningún alfabeto, sino en números.

Para hacernos una idea de este lenguaje, el cual es conocido como *lenguaje de máquina*, primero debemos recordar que toda la información que un computador procesa está representada y almacenada en términos de dos números: 1 o 0. Es decir, la información que un computador

---

<sup>2</sup>En Windows, la terminal se llama *código del sistema*



procesa es representada de forma *binaria*. Esto es precisamente el lenguaje que la máquina comprende, es decir, *el lenguaje de máquina*. Para poder traducir la información que el usuario ingresa al computador a forma binaria, o a lenguaje de máquina, es necesario utilizar un cierto sistema de codificación, el cual a cada carácter que un usuario pueda leer, se le asigne una representación binaria. Uno de los sistemas de codificación más usados es el sistema americano ASCII (*American Standard Code for Information Interchange*). A continuación, presentamos una tabla con algunos ejemplos de conversión:

binario	representación
01101000	h
01101111	o
01101100	l
01100000	a

De la tabla, claramente se ve que el byte (el cual contiene 8 bits) correspondiente la letra h es 01101000. Por lo tanto, la palabra *hola* estaría representada por la secuencia binaria 01101000011011110110110001100000.

Después de tener una idea general de qué es el lenguaje de máquina, estamos preparados para introducir de forma esquemática cómo funciona un programa. Primero, escribimos mediante la utilización de un lenguaje de programación el código fuente del programa. Segundo, debemos decidirnos por algún tipo de traductor para ingresarle el código del programa. Tercero, el traductor nos dará el programa computacional en su forma de lenguaje de máquina. Finalmente, tendremos que “enviar” el programa en lenguaje de máquina al procesador del computador para que éste lo ejecute.

Dependiendo de cómo se lleve a cabo el proceso anterior, el cual resumimos de forma esquemática en la Figura 1.2, podremos diferenciar los dos grandes tipos de lenguajes de programación: los lenguajes interpretados y los lenguajes compilados.

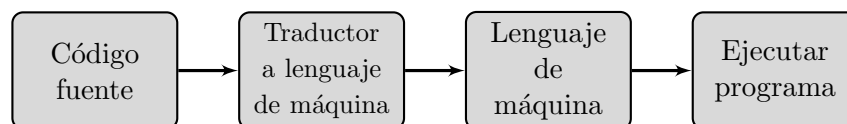


Figura 1.2: Esquema de tránsito de un programa.

### 1.2.2. Lenguajes interpretados y compilados

Un lenguaje interpretado (o precompilado) es aquel que requiere de otro programa llamado intérprete, el cual lee la lista de instrucciones o *código fuente del programa* y lo ejecuta inmediatamente. Es decir, el intérprete traduce la lista de instrucciones (o programa) al lenguaje de máquina por nosotros (usando el código ASCII, por ejemplo) y lo envía al procesador para que éste sea ejecutado por el computador, como se muestra en la Figura 1.3.

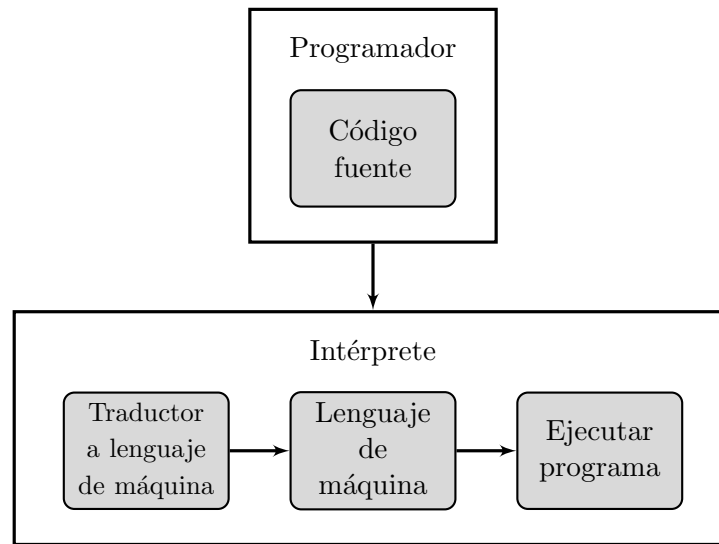


Figura 1.3: Esquema de un lenguaje interpretado.

Por otra parte, un lenguaje compilado es aquel que requiere de un programa que lo “traduzca” al lenguaje de máquina, el cual puede ser ejecutado cuando el usuario lo desee, no necesariamente de forma inmediata. Es decir, se ejecuta el programa en lenguaje de máquina, y no directamente desde el código fuente del programa, como se muestra en la Figura 1.4. Esta diferencia con los programas precompilados hace de los programas compilados mucho más veloces y eficientes, aunque en la mayoría de los casos son difíciles de escribir.

### Algunos lenguajes de programación

Algunos ejemplos de lenguajes de programación compilados son C, C++, Fortran, Arduino. Por otra parte, algunos lenguajes precompilados son Java, PHP y Python. Cada lenguaje de programación tiene sus ventajas y desventajas y por esta razón, generalmente, su implementación en la escritura de ciertos programas está determinada por las funciones que el programa deseado pretenda realizar. En particular, en este curso utilizaremos principalmente el lenguaje Python debido su gran versatilidad y amigable sintaxis con respecto a los comandos. Con fines pedagógicos, ocasionalmente también usaremos ejemplos en otros lenguajes de programación.

Finalmente, cerramos esta sección mencionando que los lenguajes de programación también se suelen dividir entre lenguajes de alto nivel y bajo nivel, refiriéndose a qué “tan cerca” están al lenguaje de máquina, o nivel fundamental. Así, por ejemplo, se dice que Asembler es un lenguaje de bajo nivel ya que sus comandos están cercanos al lenguaje de máquina mientras que Python, por otra parte, es un lenguaje de alto nivel por ser más cercano al lenguaje humano y distante del lenguaje de máquina.

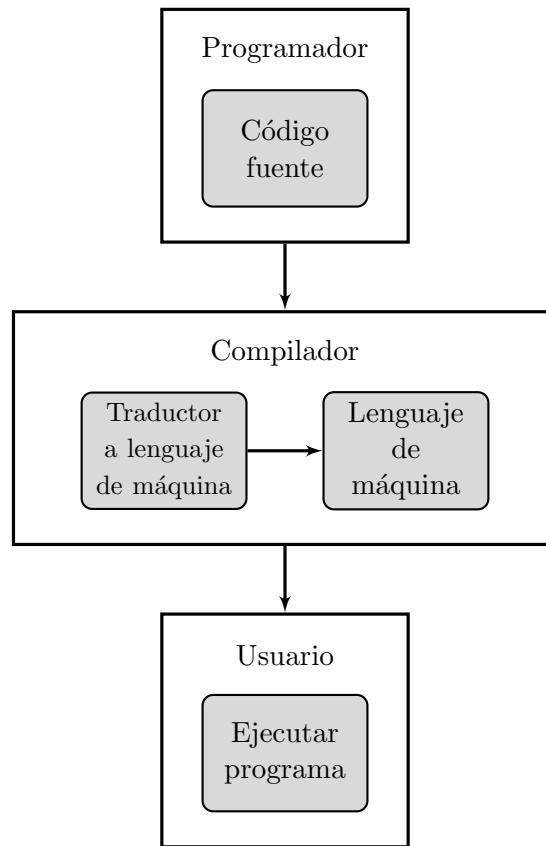


Figura 1.4: Esquema de un lenguaje compilado.

### 1.2.3. Ambientes de desarrollo para escribir programas

Un ambiente de desarrollo integrado, o IDE (Integrated Development Environment), es un software que ofrece una interfaz simple y amigable con el usuario que le permite escribir y ejecutar programas en un determinado lenguaje de programación. Es decir, el IDE puede pensarse como un “contenedor” que almacena, además de un editor de texto para escribir los programas, todas las dependencias necesarias para que el intérprete o el compilador de un determinado lenguaje de programación, puedan transformar el código fuente de un programa en lenguaje de máquina. Adicionalmente, desde el IDE también pueden ejecutarse los programas a través de una ventana de ejecución.

Esquemáticamente, una IDE puede representarse como se muestra en la Figura 1.5. Algo importante que cabe resaltar en cuanto a sus componentes principales es con respecto a las llamadas “herramientas del lenguaje de programación”. Estas suelen llamarse como *librerías* o *módulos*, y permiten ser utilizadas dentro de un programa para realizar tareas específicas. Por ejemplo, podemos usar un módulo (o librería) dentro de un programa que nos permita graficar un conjunto de datos, o también, podríamos usar otro módulo para realizar cálculos con vectores y matrices.

De lo anterior, podemos inferir que los módulos juegan un papel fundamental en la escritura

de un programa y, afortunadamente, en la mayoría de los ambientes de desarrollo tenemos una gran cantidad de ellos a nuestra disposición.

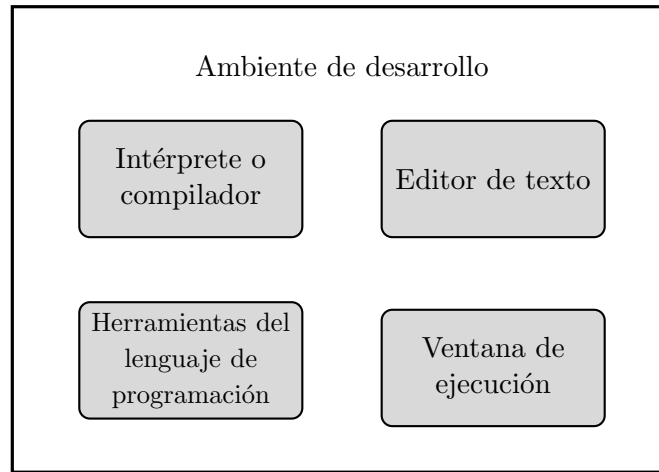


Figura 1.5: Esquema de un ambiente de desarrollo.

Para realizar programas con el lenguaje Python tenemos una gran variedad de IDE a nuestra disposición. Entre ellos, los más usados por la comunidad de programadores en la actualidad se encuentran IDLE, Ecilpse, IPython, Spyder, JupyterLab, JupyterNotebook, entre otros. Todos ellos pueden ser descargados directamente desde sus sitios web oficiales de forma gratuita en conjunto con una gran cantidad de tutoriales de uso. En particular, en este curso utilizaremos JupyterNotebook, el cual puede ser descargado de forma gratuita desde,

<https://jupyter.org>

Después de acceder a la pagina web oficial de Jupyter, la cual es una organización sin ánimo de lucro, creada para desarrollar software de código abierto, puede descargarse el instalador del entorno de desarrollo JupyterNotebook.

### 1.3. Desarrollo de algoritmos

Antes de escribir un programa computacional, no solo debemos tener claridad en la tarea específica que queremos que el programa realice, sino también en los “pasos” que el programa debe seguir para realizar dicha tarea. La definición y ordenamiento de manera secuencial de estos pasos constituyen el *algoritmo* del programa.

De forma general, podemos definir un algoritmo como una secuencia sistemática y finita de instrucciones que permiten realizar una determinada tarea. Por lo tanto, podemos considerar los algoritmos como la “hoja de ruta” de los programas, o dicho de otro modo, los algoritmos son los “diseños” de los programas. Así que, antes de escribir un programa, es decir, antes de escribir una lista de tareas mediante un lenguaje de programación, el programador debe

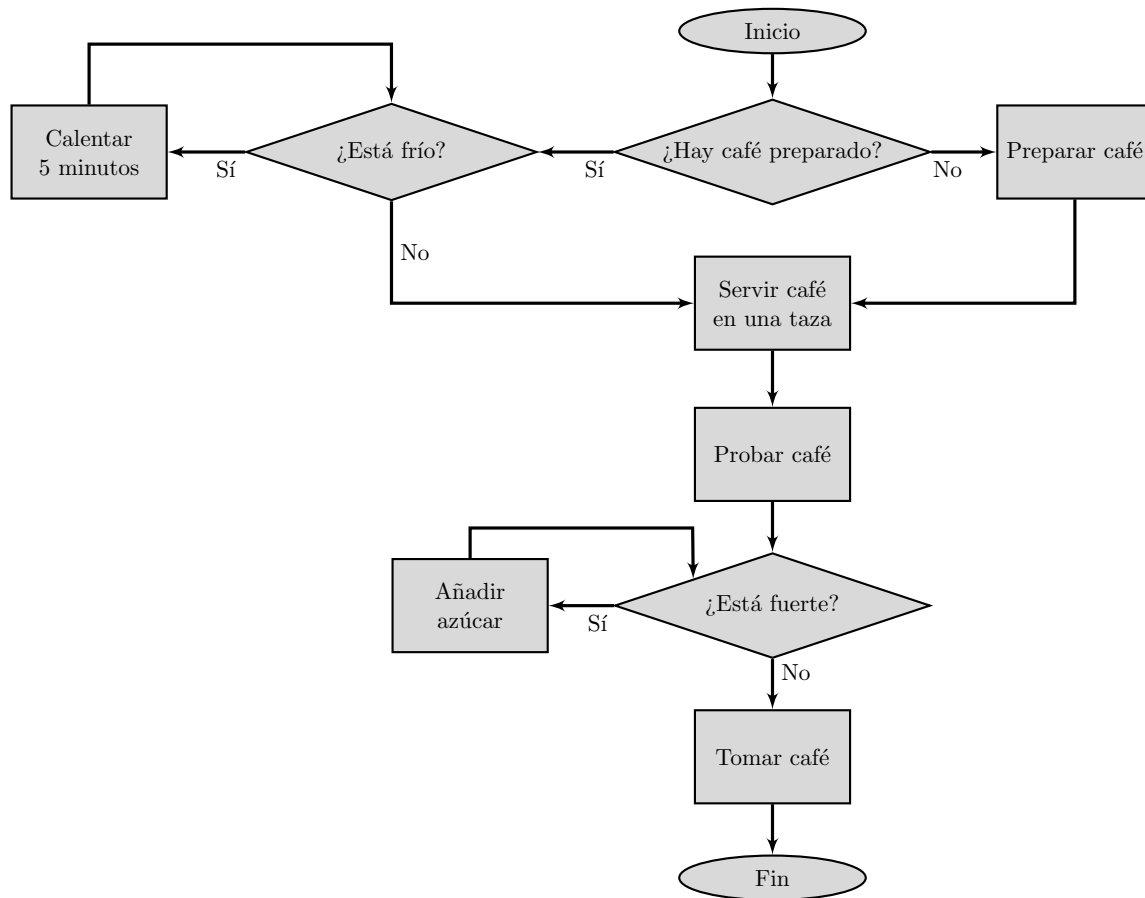


Figura 1.6: Algoritmo para tomar un café en casa.

tener a su disposición el algoritmo del programa. Cabe resaltar que para escribir un algoritmo NO necesitamos de ningún lenguaje de programación, ya que este puede ser escrito de forma tal que pueda ser programado en diversos lenguajes. Es decir, el algoritmo es un “bosquejo” abstracto de unos pasos para realizar una tarea mientras que el programa es el algoritmo codificado mediante un lenguaje de programación.

### 1.3.1. Diagramas de flujo y pseudo-códigos






#### Diagramas de flujo

En general, un algoritmo puede ser escrito de muchas maneras. Entre ellas, los diagramas de flujo son de gran relevancia ya que permiten una representación gráfica de los algoritmos. Como ilustración, consideremos el siguiente algoritmo que describe los pasos que una persona sigue para tomar café, escrito o representado mediante el diagrama de flujo.

En este diagrama observamos que tenemos varias flechas, rectángulos, rombos y círculos. En nuestro caso, hemos utilizado los rectángulos para denotar acciones, los rombos para indicar decisiones y flechas para indicar la acción “seguir” o “camino de ejecución” del algoritmo.

Además, note que en el algoritmo, las preguntas son formuladas de tal forma que la respuesta siempre es Sí o No, y nunca se usa un “tal vez”. Dependiendo del valor de verdad de las respuestas, se sigue uno u otro camino en la ejecución. Es importante resaltar que existen algunos “ciclos” o “caminos cerrados” dentro de un algoritmo, los cuales nos indican repetir una acción hasta que cierta condición se satisfaga. Estas acciones especiales dentro del algoritmo, la toma de decisiones y los caminos cerrados son un ingrediente de suma y serán discutidos detalladamente en la sección 1.4.

Finalmente, en la Figura 1.1 podemos ver la convención estándar para la utilización de algunos símbolos para diagramas de flujos (ver por ejemplo [4]) .

SÍMBOLO	NOMBRE	ACCIÓN
	Terminal	Representa el inicio o el fin del diagrama de flujo.
	Entrada y salida	Representa los datos de entrada y los de salida.
	Decisión	Representa las comparaciones de dos o más valores, tiene dos salidas de información: falso o verdadero.
	Proceso	Indica todas las acciones o cálculos que se ejecutaran con los datos de entrada u otros obtenidos.
	Líneas de flujo de información	Indican el sentido de la información obtenida y su uso posterior en algún proceso subsiguiente.

Cuadro 1.1: Significado de algunos símbolos para diagramas de flujos.

### Pseudo-código

Otra forma eficiente de escribir un algoritmo es mediante la utilización de pseudo-código. Como su nombre lo indica, el pseudo-código no es código de programación en sí mismo, sino una manera de escribir el algoritmo como una lista de pasos de manera ordenada. Es decir, el pseudo-código sirve para representar los pasos del algoritmo de la forma más detallada posible, sin utilizar ningún lenguaje de programación. Por lo tanto, podemos pensar que el pseudo-código de un algoritmo es el “molde general” de un programa, el cual después puede ser traducido fácilmente a cualquier lenguaje de programación. Para ilustrar lo anterior, considere el siguiente algoritmo escrito en pseudo-código:

- 
1. Ingresar primer valor.
  2. Ingresar segundo valor.
  3. Sumar los dos valores anteriores.
  4. Mostrar en pantalla el resultado de la suma.
- 

**Algoritmo 1:** Ejemplo de pseudo-código como una lista ordenada de instrucciones.

Claramente, el Algoritmo 1 describe un programa que recibe dos números y luego calcula su suma. Note que para el buen funcionamiento de este algoritmo necesitamos “ingresar” información al programa y, como resultado, obtenemos una información de “salida”. Esto no es coincidencia, y corresponde a una cuestión fundamental que todo algoritmo debe determinar: ¿Cómo manejar la información de entrada y de salida en un programa? La respuesta a esta pregunta consiste en el uso de variables.

#### 1.3.2. Entrada y salida de información: Uso de variables

Las variables, al igual que en el álgebra, pueden ser utilizadas para “representar” o “almacenar” de forma abstracta un objeto, el cual puede ser un número, una letra, etc. Por lo tanto, mediante su uso en un algoritmo dispondremos de una forma simple de manejar la información de entrada y salida del programa. Por ejemplo, en el Algoritmo 2 definimos las variables **a** y **b** para almacenar los valores que deben ser ingresados al programa y la variable **suma** para “guardar” el valor de su suma.

1. Primer dato de entrada lo guardamos en la variable **a**.
2. Segundo dato de entrada lo guardamos en la variable **b**.
3. Sumamos las variables **a** y **b** y el resultado lo guardamos en la variable **suma**, es decir,

$$\text{suma} = a + b$$

4. Mostramos en pantalla o guardamos el valor almacenado en la variable **suma**.
- 

**Algoritmo 2:** Ejemplo de pseudo-código como una lista ordenada de instrucciones.

En este caso, podríamos decir que las variables **a** y **b** almacenan la **información de entrada** del algoritmo, mientras que por otra parte, la variable **suma** sirve para almacenar la **información de salida**. Por otra parte, el lugar del programa en donde se realizan las operaciones, en nuestro caso la línea donde escribimos **suma = a + b**, se suele llamar *cuerpo del programa*.

En conclusión, un algoritmo siempre puede ser dividido en tres partes principales, como se muestra en la Figura 1.7. Primero, una parte donde se identifica la información de entrada, otra donde se realizan los procesos con esa información y, finalmente, la información de salida donde se obtienen los resultados.

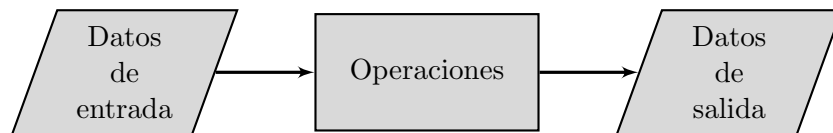


Figura 1.7: Esquema de un programa.

## 1.4. Elementos básicos de programación

Como se observo en el algoritmo de preparación de café representado en el diagrama de flujo Figura 1.6, en un algoritmo puede ser necesario realizar ciertas preguntas que establecen un punto de bifurcación o división de la forma en que un algoritmo es ejecutado. Este tipo de preguntas y toma de decisiones las llamaremos en adelante *estructuras condicionales*. Adicionalmente, también es posible encontrar ciertos caminos cerrados que obligan al algoritmo a ejecutar una tarea de forma repetitiva hasta que una cierta condición se satisfaga. Estas estructuras las llamaremos *estructuras de ciclos*. A continuación explicaremos cada una por separado utilizando pseudo-código.



### 1.4.1. Estructuras de control condicionales

Como se menciono anteriormente, la estructura condicional se basa en establecer un punto de bifurcación de un algoritmo basado en determinar si una condición se satisface o no por medio de una pregunta. Algo importante para resaltar, es que se espera que la respuesta a esta pregunta solo sea VERDADERA o FALSA, es decir, respuestas como “tal vez” no son válidas.

Para introducir la estructura básica de una estructura condicional en forma de pseudo-código, consideremos el siguiente algoritmo de un programa que permita determinar si cierto número es positivo o no.

---

1. Introducimos el número como dato de entrada y lo guardamos en la variable **a**.

2. Preguntamos si **a** es mayor que 0:

Caso Afirmativo: Hacemos **resultado** = VERDADERO.

Caso Negativo: Hacemos **resultado** = FALSO.

3. Imprimimos en pantalla el valor de la variable **resultado**.

4. Mostramos en pantalla o guardamos el valor almacenado en la variable **suma**.

---

**Algoritmo 3:** Ejemplo de pseudo-código como una lista ordenada de instrucciones.

Claramente, el pseudo-código en el Algoritmo 3, escrito como una lista ordenada de instrucciones, nos indica de forma general las pautas para escribir un programa computacional. Sin embargo, debido a esta generalidad, podría resultar que el pseudo-código no resulte muy útil para guiarnos a escribir ciertos “detalles” del programa en cuestión, teniendo que recurrir a diagramas de flujo u otro tipo de ayudas. Por esta razón, en varias ocasiones resulta ser mucho mas útil escribir el pseudo-código de una forma un poco más estructurada y “parecida” a un lenguaje de programación, de tal forma que sea mucho más sencilla su programación en dicho lenguaje. Por ejemplo, el pseudo-código anterior también podemos escribirlo de una forma estructurada y parecida al lenguaje Python como sigue:

**entrada:** a=0

```
if (a mayor que 0) :  
    | resultado = True  
else:  
    | resultado = False
```

**salida:** resultado

---

**Algoritmo 4:** Ejemplo de pseudo-código más específico y cercano a un lenguaje de programación.

En el Algoritmo 4 note que una vez asignamos a la variable **a** el número 0, realizamos la pregunta de si **a** es mayor que cero usando el comando **if**. Si de esta pregunta obtenemos una respuesta verdadera, entonces podemos decir que **a** es verdadero. En caso contrario, ejecutamos la acción que está después de la palabra **else**, que en español traduce *de otro modo*. Es decir, nuestro algoritmo ejecutará una de las dos posibles acciones dependiendo de si la condición se satisface o no.

Algo importante para resaltar es la indentación o la sangría que utilizamos en nuestro pseudo-código para anidar las acciones. Esto es, los espacios en blanco que colocamos antes de escribir el **resultado**, para indicar que esas son las acciones correspondientes al caso verdadero y al caso falso de la pregunta, respectivamente. Esto lo hacemos con el propósito de hacer que nuestro pseudo-código sea lo más cercano a la escritura de un programa en Python. Sin embargo, en otros lenguajes de programación, esta indentación se sustituye por el uso de corchetes o llaves para delimitar la estructura.

El Algoritmo 4 puede ser programado en lenguaje Python fácilmente como sigue:

```
1 #definir variable  
2 a = 2  
3  
4 if ( a > 0 ):  
5     resultado = 'el numero es positivo'  
6  
7 else:  
8     resultado = 'el numero NO es positivo' )  
9  
10 print( resultado )
```

Código 1.1: Programa en Python del algoritmo con condicionales.

Note que en el código 5.6 hemos utilizado un paréntesis para indicar la condición **a>0** seguido de dos puntos **:** y también hemos utilizado la función **print( )** para imprimir en pantalla el resultado del programa.

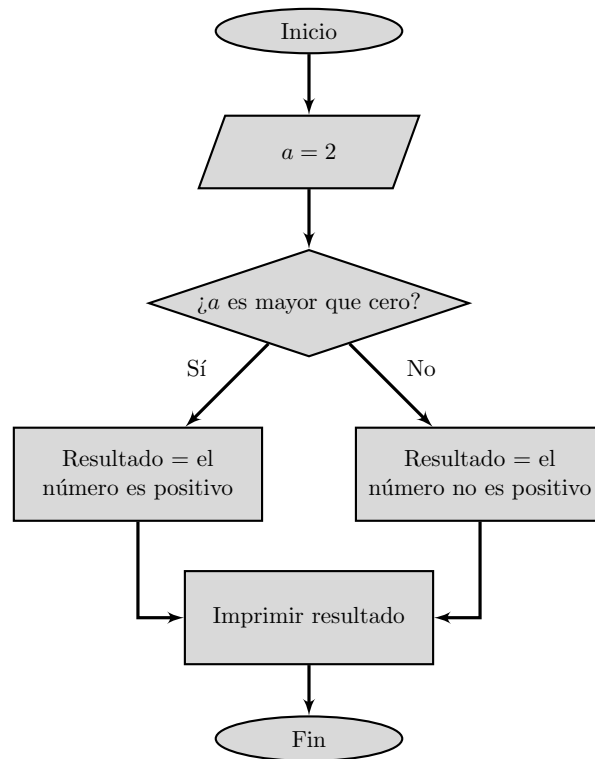


Figura 1.8: Diagrama de flujo del programa con condicionales.

En conclusión de lo anterior, debemos decir que en general no existen reglas predeterminadas para escribir pseudo-códigos, ya que su nivel de especificidad y estructura para describir los detalles de un programa dependen enteramente del programador. Por ello, debemos verlos meramente como una herramienta (y no como un fin) que nos permite construir un programa de la mejor manera posible. Adicionalmente, debemos recordar que, como mencionamos en la sección anterior, también podemos utilizar diagramas de flujo para describir el algoritmo de un programa. Por ejemplo, en este caso en particular, el diagrama de flujo descrito en la Figura 5.3 también describe adecuadamente el algoritmo requerido. Por lo tanto, no existe ni una única forma y ni una mejor manera para escribir el algoritmo de un programa.

### 1.4.2. Estructuras de ciclos

Como se mencionó anteriormente, los ciclos<sup>3</sup> nos permiten *repetir* determinadas acciones dentro de un programa, bien sean por un cierto número de veces o hasta que una condición se satisfaga. En esta sección introduciremos los dos tipos de ciclos más importantes: los ciclos por elementos, comúnmente conocidos como ciclos **for**, y los ciclos por condiciones, también llamados ciclos **while**.

---

<sup>3</sup>En inglés es común llamarlos *loops*.

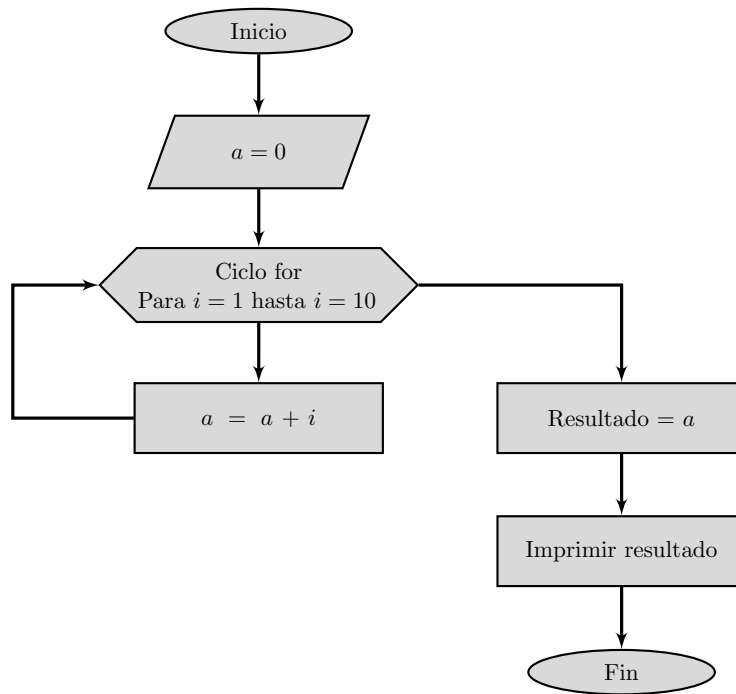


Figura 1.9: Diagrama de flujo del algoritmo con ciclo **for**.

### Ciclos por elementos

Los ciclos por elementos, o ciclos tipo **for**, corresponden a realizar una acción o acciones repetidamente un cierto número de veces hasta que un *contador* alcance un cierto valor. Por lo tanto, de forma general, podemos representar un ciclo **for** mediante el siguiente pseudo-código en forma de lista:

- 
1. Variable contador **i**
  2. Valor inicial del contador **a**
  3. Valor final del contador **b**
  4. (Ciclo for) Para **i** tomando valores desde **a** hasta **b** hacer:  
    ( Acciones )
  5. Fin del ciclo
- 

**Algoritmo 5:** Ejemplo de pseudo-código a manera de lista.

Para una mejor ilustración de lo anterior, consideremos el siguiente diagrama de flujo que contiene el algoritmo de un programa para sumar todos los números comprendidos entre el 1 y el 10.

En este algoritmo inicializamos una variable **a** y le asignamos el valor de 0. Luego, utilizado

un hexágono achatado para representar simbólicamente el ciclo **for**, calculamos la suma  $a = a + i$ , donde la variable  $i$ , que actúa como contador del ciclo, tomará valores desde 1 hasta 10. Por cada valor que tome el contador  $i$ , se calculará el nuevo valor de  $a$ . Por ejemplo, el primer ciclo será para  $i=1$ , entonces tendremos  $a = 0 + 1$  (note que  $a$  pasa a tener un nuevo valor). Luego, en el segundo ciclo tendremos que  $i=1$  y por tanto  $a = 1 + 1$  y así sucesivamente hasta que el contador  $i$  llegue a 10. Esto es, precisamente, lo que indica el camino “cerrado” en el diagrama. Finalmente, cuando el contador alcanza el valor de 10, se sale del ciclo y se asigna el valor de  $a$  a la variable **resultado**, la cual puede ser “impresa” en pantalla o en un archivo.

Finalmente, antes de pasar al siguiente tipo de ciclos, ilustramos como este pseudo-código puede ser programado en lenguaje Python. Note que se usa una indentación, o sangría, para indicar las acciones que se realizan dentro del ciclo **for** (ver [2] para una discusión mas amplia al respecto).

```
1 a = 0
2
3 for i in range(11):
4     a =+ i
5
6 resultado = a
7
8 print(resultado)
9
```

Código 1.2: Programa en Python del algoritmo con ciclo **for**.

*Práctica computacional:* Como complemento de este tema, ver el programa de **verificador de números primos** ubicada en el repositorio del curso, en el cual se combinan el ciclo **for** y los condicionales de la sección anterior.

### Ciclos por condiciones

Los ciclos condicionales o ciclos **while** se diferencian de los ciclos **for** en el hecho de que estos se ejecutan siempre y cuando una cierta condición se satisfaga. Note que **while** es un anglicismo que traduce al español como *mientras*. Así pues, puede decirse que estos ciclos podrían ser ejecutados infinitamente **mientras** la condición nunca deje de satisfacerse. Por ejemplo, consideremos el siguiente pseudo-código bastante “cercano” al lenguaje Python, en el que se plantea sumar todos los números enteros positivos menores o iguales que 45.

---

**entrada:** a=0

**entrada:** i=0

**while** (a menor o igual que 45) :

    a = a + i

resultado = a

**salida:** resultado

---

**Algoritmo 6:** Ejemplo de algoritmo con ciclo **while** mal implementado.

En este algoritmo, realizamos la suma de **a** con la variable **i**. Note que como el valor inicial de **a** es *menor o igual* que 45, podemos “entrar” al ciclo **while**. Sin embargo, como la variable **i** no está cambiando valores e inicialmente es igual a cero, el valor de la variable **a** nunca cambiará y por tanto permanecerá siendo igual a 0. Es decir, el ciclo se ejecutará infinitamente y nuestro programa **nunca** dejará de ejecutarse. Para poder “romper” o “escapar” del ciclo **while**, debemos forzar la variable **a** a cambiar de valores de algún modo. Para ello, consideremos la siguiente modificación del pseudo-código anterior:

---

**entrada:** a=0

**entrada:** i=0

**while** (a menor o igual que 45) :

    i = i + 1  
    a = a + i

resultado = a

**salida:** resultado

---

**Algoritmo 7:** Ejemplo de algoritmo con ciclo **while** bien implementado.

Con la introducción de la línea **i = i + 1** en el Algoritmo 7, logramos que la variable **i** cambie su valor y, por tanto, **a**, a través de la suma **a = a + 1**.

Similarmente a como hicimos para el ciclo **for** y los condicionales, presentamos el código Python de este algoritmo en el código 1.3, así como su diagrama de flujo en la Figura 1.10.

```
1 a = 0
2
3 i = 0
4
5 while ( a <= 45 ):
```

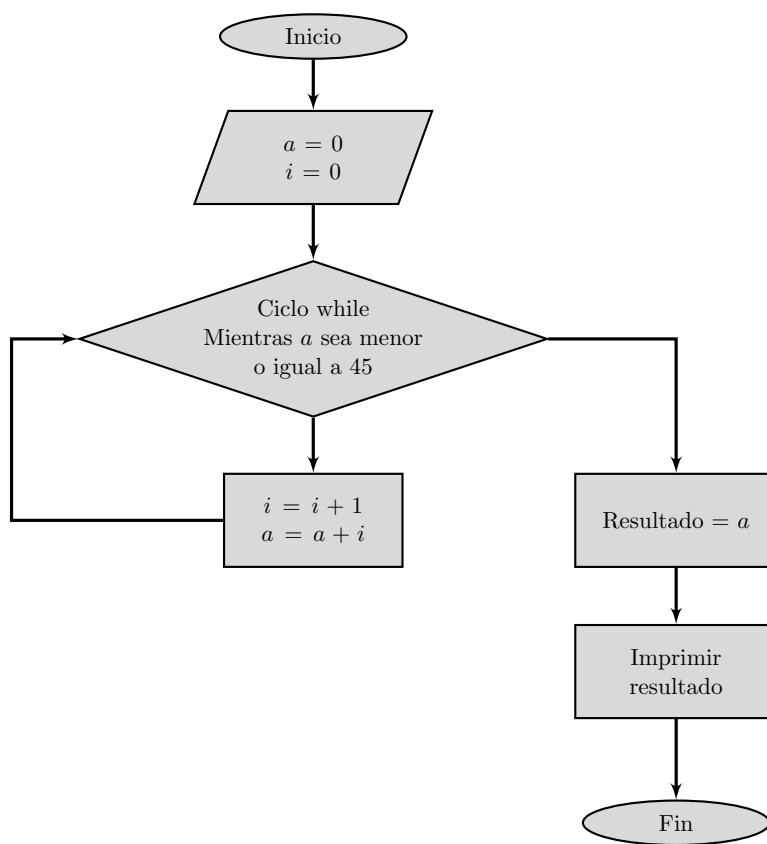


Figura 1.10: Diagrama de flujo del algoritmo con ciclo **while**.

```
6
7     i = i + 1
8
9     a = a + i
10
11 suma_final = a
12
13 print(suma_final)
```

Código 1.3: Programa en Python de un ciclo **while**.

*Práctica computacional:* Como complemento de esta sección, ver el programa del **epsilon de máquina** en el repositorio del curso.

### 1.4.3. Listas y arreglos

Podemos definir una lista como una colección ordenada de elementos. Estos elementos, los cuales no son necesariamente del mismo “tipo”, pueden ser variables, números, cadenas de caracteres, matrices, gráficos, funciones, etc. Para ilustrar este importante concepto imaginemos que queremos utilizar una lista para guardar ciertos datos de una persona. En particular, queremos que esta lista contenga su nombre, su apellido y su edad. Adicionalmente, después de creada esta lista, queremos introducir un condicional que verifique la edad de la persona y que, de acuerdo a esto, determine si es mayor o menor de edad (mayor o igual de 18 años). Para llevar a cabo esta tarea, consideremos el siguiente código en Python:

```
1 Nombre = 'Alan'
2 Apellido= 'Turing'
3 Edad   = '40'
4
5 Datos = [ Nombre , Apellido , Edad ]
6
7 if ( Datos[2] >= 18 ):
8
9     resultado = "La persona es mayor de edad"
10
11 else:
12
13     resultado = "La persona es menor de edad"
14
15 print(resultado )
```

Note que hemos utilizado el nombre **Datos** para llamar a la lista que contiene la información de la persona. Es decir, **Datos** es una variable que contiene la información de otras variables. Adicionalmente, utilizamos las llaves `[ ]` para encerrar o agrupar los elementos de la lista y los hemos separado mediante comas. Observe que los dos primeros elementos de la lista son cadenas de caracteres, mientras que el tercero corresponde a un número. Es decir, los elementos no son del mismo tipo. En el caso particular en que la lista contenga elementos del mismo tipo, por ejemplo una lista de números, decimos que la lista es un *arreglo*.



Algo importante para resaltar es que hemos escrito `Datos[2]` para llamar el valor del tercer elemento de la lista `Datos`. Para más detalles sobre listas y arreglos ver Apéndice.

Para mostrar lo útiles que pueden ser las listas para el cálculo numérico, consideremos el siguiente pseudo-código cercano al lenguaje Python. El propósito de este algoritmo consiste en calcular y guardar en forma de lista los valores de la parábola  $y = x^2$  para determinados valores de la variable  $x$ . Luego, como es costumbre, presentaremos el algoritmo escrito en lenguaje Python.

---

**entrada:** `DatosX` = lista vacía

**entrada:** `DatosY` = lista vacía

**for** (  $x = 1$  hasta 10 ) :

$$y = x^2$$

Agregar el valor de  $x$  a la lista `DatosX`

Agregar el valor de  $y$  a la lista `DatosY`

`DatosTotales` = [ `DatosX` , `DatosY` ]

**salida:** `DatosTotales`

---

**Algoritmo 8:** Ejemplo de pseudo-código usando listas.

Note que en este pseudo-código comenzamos declarando como dato de entrada las listas vacías `DatosX` y `DatosY`. Este paso es fundamental, ya que nos permite crear las dos listas o conjuntos de datos, las cuales procederemos a “llenar” de forma iterativa dentro del ciclo. Luego, utilizamos un **for** para hacer que la variable  $x$  tome valores enteros desde 1 hasta 10 y así obtenemos los valores de la parábola  $x^2$  para cada valor. Naturalmente, en cada evaluación agregamos cada valor de  $x$  y  $y$  a las listas `DatosX` y `DatosY`, respectivamente. Note que después de terminado el ciclo **for**, podemos agrupar las dos listas de datos en una nueva lista llamada `DatosTotales`, es decir, tenemos una lista compuesta por dos listas. Adicionalmente, dado que todos los elementos de las listas `DatosX`, `DatosY` y `DatosTotales` son del mismo tipo, podemos decir que son arreglos. Note que con la intención de hacer el código lo suficientemente claro, hemos incluido comentarios (las líneas que siguen del símbolo `#`), es decir, instrucciones que no son ejecutadas por el compilador y que solo sirven para identificar algunos procesos y variables.

```
1 #Definiendo listas vacias
2 DatosX = []
3 DatosY = []
4
5 for x in range(11):
6
7     #Calculando el cuadrado de x
8     y = x ** 2
```

```
9
10     #Agregando valor de x a lista DatosX
11     DatosX.append(x)
12
13     #Agregando valor de y a lista DatosY
14     DatosY.append(y)
15
16 #Definiendo la lista de listas
17 DatosTotales = [ DatosX , DatosY ]
18
19 #Imprimiendo los datos en la pantalla
20 print(DatosTotales)
```

*Práctica computacional:* Como actividad complementaria para esta parte de listas, ver la práctica de **gráfico de funciones1** ubicada en el repositorio del curso en la cual se grafica la función cuadrática obtenida mediante este programa utilizando el modulo `matplotlib`. Adicionalmente, en la práctica **gráfico de funciones2** se muestra una forma más rápida para hacer gráficos de funciones mediante el uso de arreglos tipo `numpy`. Para mas información de este módulo ver [3] o visitar el sitio web <https://numpy.org/>

#### 1.4.4. Funciones

Llegamos finalmente a la definición y utilización de funciones. Las funciones son esencialmente sub-rutinas que se ejecutan dentro de un programa. Estas, al igual que los programas, requieren unos datos de entrada o argumentos, y nos *retorna* un dato de salida o resultado. Para ilustrar este concepto, consideraremos el siguiente pseudo-código en forma de lista, el cual aborda el mismo problema del Algoritmo 8 de calcular y guardar en forma de lista los valores de la parábola  $y = x^2$  para determinados valores de la variable  $x$ , pero utilizando funciones.

- 
1. Definimos una lista vacía para guardar los datos de la variable independiente  $x$ .
  2. Definimos una lista vacía para guardar los datos de la variable dependiente  $y$ .
  3. Definimos una función que tome un número y retorne su cuadrado.
  4. Para un contador tomando valores desde 1 hasta 10 hacer lo siguiente:
    - a. Evaluamos la función en el valor del contador.
    - b. Agregamos el valor del contador en la lista de la variable independiente  $x$ .
    - c. Agregamos el valor de la función en la lista de la variable dependiente  $y$ .
  5. Agregamos las dos listas a una nueva llamada `DatosTotales`, la cual será el dato de salida del programa.
- 

**Algoritmo 9:** Ejemplo de pseudo-código en forma de lista usando funciones.

En este pseudo-código observamos claramente que para poder usar la función (elevar un número al cuadrado) dentro del ciclo, se debe haber definido con anterioridad de forma abstracta. Adicionalmente, es importante notar que se utilizaron dos listas vacías para almacenar

los valores que obteníamos tanto para la variable dependiente como para la independiente, en cada iteración del ciclo **for**. Para una mejor ilustración del algoritmo que nos facilite su escritura en lenguaje de Python, refinamos un poco más el pseudo-código anterior como sigue:

---

**entrada:** DatosX = lista vacia

**entrada:** DatosY = lista vacia

**FCuadratica** ( argumento ) :

    resultado = el valor de argumento elevado al cuadrado

    retornar valor de la variable resultado

**for** (  $x = 1$  hasta 10 ) :

$y$  = Evaluar la función **FCuadratica** en la variable  $x$

    Agregar el valor de  $x$  a la lista DatosX

    Agregar el valor de  $y$  a la lista DatosY

DatosTotales = [ DatosX , DatosY ]

**salida:** DatosTotales

---

**Algorithm 10:** Ejemplo de pseudo-código usando funciones.

Note que aquí hemos definido una función llamada **FCuadratica**, la cual toma un número como argumento y devuelve su cuadrado. Es decir, **FCuadratica** es la función cuadrática  $y = x^2$ . Luego, dentro del ciclo **for** “evaluamos” la función **FCuadratica** para los valores de  $x$ . La acción “evaluar la función” corresponde a enviar cada valor de esta variable a la sub-rutina **FCuadratica**, la cual calcula su cuadrado y retorna el resultado. Finalmente, el valor que retorna la **FCuadratica** es asignado a la variable  $y$ , la cual es agregada a la lista DatosY.

Por consistencia con las discusiones anteriores, finalizamos esta sección presentando el código *Python* del algoritmo anterior:

```
1 #Definiendo listas vacias
2 DatosX = []
3 DatosY = []
4
5 #Definiendo la funcion cuadratica
6 def Cuadratica( argumento ) :
7
8     return argumento**2
9
10
```

```
11 for x in range(11):
12
13     #Calculando el cuadrado de x
14     y = Cuadratica(x)
15
16     #Agregando valor de x a lista DatosX
17     DatosX.append(x)
18
19     #Agregando valor de y a lista DatosY
20     DatosY.append(y)
21
22 #Definiendo la lista de listas
23 DatosTotales = [ DatosX , DatosY ]
24
25 #Imprimiendo los datos en la pantalla
26 print(DatosTotales)
```

*Práctica computacional:* como actividades complementarias para afianzar este importante tema de funciones, consultar en el repositorio del curso las prácticas de **derivación de funciones, integración numérica y generación de números aleatorios**. Finalmente, para mas ejemplos de funciones en Python recomendamos al lector ver [2].

## 1.5. Introducción a la preparación de documentos mediante LaTeX

LaTeX es un sistema de composición de textos orientado a la creación de documentos escritos que requieren una alta calidad tipográfica. Por su amplia variedad de características es extensamente usado en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas, gráficos y tablas de datos. LaTeX está formado mayoritariamente por órdenes construidas a partir de comandos de TeX, lo cual es una mezcla entre procesador de textos y lenguaje de programación. Por lo tanto, en términos generales, podemos considerar a LaTeX como un “cierto lenguaje de programación” para generar textos de carácter científico con una alta calidad tipográfica.

En esta sección presentaremos una breve introducción a LaTeX concentrándonos, principalmente, en la forma y estructura de los códigos. De forma similar a como hicimos con el lenguaje Python, también usaremos un ambiente de desarrollo (o IDE) para escribir y ejecutar dichos códigos. Actualmente, existe una gran variedad de IDEs gratuitas y de pago. Entre las opciones libres más comunes encontramos Texmaker, TeXnicCenter, LyX, Kile y TeXstudio, los cuales pueden ser descargados fácilmente desde sus correspondientes sitios web. En particular, debido a su gran facilidad de uso, amigable interfaz de usuario y por ser multiplataforma (uso en diversos sistemas operativos), en este curso haremos uso del IDE TeXstudio, la cual puede obtenerse en el sitio web oficial:

<https://www.texstudio.org>

Para finalizar, cabe mencionar que en la actualidad existe una gran cantidad de sitios web gratuitos y de pago, como por ejemplo *Overleaf* (<https://es.overleaf.com>), que proveen IDEs online a los usuarios que les permite escribir y ejecutar su código LaTeX de forma online y de manera inmediata sin la necesidad de instalar paquetes. Además, brinda una amplia variedad

de plantillas.

### 1.5.1. Estructura general del código LaTeX de un documento

Un documento en código LaTeX se divide principalmente en dos partes: el preámbulo y el cuerpo del documento. La primera parte es el lugar en donde seleccionamos las características generales del documento, cargamos los paquetes o módulos (similar a como hacemos en Python) y establecemos ciertos parámetros. Por otra parte, el cuerpo del documento es donde realmente escribimos el contenido del documento como el texto, las ecuaciones, las tablas, los gráficos y demás.

Antes de ilustrar con un ejemplo, debemos tener presente que los comandos o instrucciones en LaTeX se diferencian del texto normal en que éstos siempre comienzan con un backslash (\) seguidos por una palabra clave. Además, las opciones de los diferentes comandos siempre vienen encerradas por unas llaves {} y separadas por comas. Es decir, los comandos en LaTeX tienen la siguiente forma:

`\Comando{opcion1,opcion2,...}`

Con lo anterior en mente, consideremos el siguiente código LaTeX de un documento que tan solo contiene la frase *hola mundo*:

```
1 \documentclass{article}
2
3 \begin{document}
4
5     hola mundo
6
7 \end{document}
```

Claramente este código, el cual debemos guardar con extensión `.tex` para señalar que es un archivo en código LaTeX, está compuesto por tres comandos y una frase. El primer comando, `\documentclass{article}`, establece el tipo del documento. Note que hemos escogido como opción: `article`, lo que indica que nuestro documento tendrá la estructura de un artículo. Algunas clases típicas, dependiendo del tipo de documento que deseemos escribir, son: `book`, `report`, `beamer`, `slides`, `letter`, entre otras.

Luego, tenemos los comandos `\begin{document}` y `\end{document}` los cuales delimitan el inicio y final del cuerpo del documento. Además de esto, se considera que todo lo que está antes de `\begin{document}` es el preámbulo del documento. Finalmente, en medio de estos dos comandos, escribimos el contenido de nuestro documento, lo cual es simplemente la frase *hola mundo*.

Una vez compilado en el IDE escogido, este código producirá un documento en formato pdf que contendrá una hoja con la frase *hola mundo*.

### 1.5.2. El preámbulo del documento

Como mencionamos anteriormente, en el preámbulo podemos, entre otras cosas, definir valores de variables globales del documento y cargar paquetes. A continuación, presentamos algunos ejemplos.

#### Variables generales del documento

Consideremos el siguiente código LaTeX. Note que todo el texto seguido del símbolo % indica comentarios, lo que nos permite delimitar el preámbulo y el cuerpo del documento:

```
1 %%%%%%%%%%%%%%% preambulo%%%%%%%%%%%%%%
2 \documentclass{article}
3
4 \title{Mi primer articulo}
5 \author{Mi nombre}
6 \date{28 de Abril de 2021}
7
8 %%%%%%%%%%%%%%% cuerpo del documento%%%%%%%%%%%%%%
9 \begin{document}
10
11
12 \end{document}
```

En este código hemos adicionado los comandos `\title{Mi primer articulo}` y `\author{Mi nombre}` y `\date{28 de abril de 2021}`, las cuales definen variables globales del documento como el autor, la fecha y el título del documento. Existen muchas otras variables globales, varias de las cuales pueden estar asociadas a paquetes externos. Para más información acerca de otras variables globales de los documentos de LaTeX y demás, consultar [1].

#### Paquetes

De forma similar a como los módulos actúan en Python, podemos considerar los paquetes en LaTeX como pequeños sub-programas que nos permiten hacer ciertas cosas en el documento. Por ejemplo, existen paquetes específicos que nos permiten incluir gráficos, insertar caracteres matemáticos propios del cálculo, incluir hipervínculos (como funciona este documento!) etc. Para cargar un paquete debemos utilizar el comando `\usepackage`, el cual tiene la siguiente estructura:

$$\textcolor{blue}{\code{\usepackage}}[\text{opcion1},\text{opcion2},\dots]\{\text{nombre del paquete}\}$$

Para ilustrar lo anterior, consideremos el siguiente ejemplo de un código LaTeX en el cual invocamos algunos paquetes. Note que los comentarios incluidos después del símbolo %, explican la utilidad de cada paquete dentro del documento.

```
1 %%%%%%%%%%%%%%% preambulo%%%%%%%%%%%%%%
2 \documentclass{report}
3
```

```
4 %%%%%% paquetes%%%%%
5 \usepackage{amsmath} % Permite manipulacion de ecuaciones matematicas
6 \usepackage{graphicx} % Permite incluir graficos
7 \usepackage[spanish]{babel} % Para establecer idioma del doc.
8 \usepackage[utf8x]{inputenc} % Para incluir las tildes.
9
10 %%%%%% variables %%%%
11 \title{Mi primer artículo}
12 \author{Mi nombre}
13 \date{28 de Abril de 2021}
14
15
16 %%%%%%%%%%%%%% cuerpo del documento%%%%%%%%%%%%%
17 \begin{document}
18
19
20
21 \end{document}
```

Para mas información de los diferentes paquetes (y sus funciones) que existen en LaTeX, recomendamos al lector visitar el sitio web oficial <https://www.ctan.org/>. Adicionalmente, el lector puede ver el código `Practica_de_Latex` ubicada en el repositorio del curso para más ejemplos.

### 1.5.3. El cuerpo del documento

#### Título, secciones y sub-secciones

Como mencionamos anteriormente, el cuerpo del documento es la región del código delimitada por los comandos `\begin{document}` y `\end{document}`. En esta parte es donde escribimos el documento como tal. Sin embargo, primero debemos incluir su título y realizar, si es el caso, algunas divisiones del mismo, ya sean en capítulos (solo válido si el documento es tipo `book`), secciones y sub-secciones.

Para ilustrar lo anterior, veamos el siguiente ejemplo, en el que se ha incluido el título del documento seguido de la secciones y sub-secciones.

```
1 %%%%%%%%%%%%%% preambulo%%%%%%%%%%%%%
2 \documentclass{article}
3
4
5 %%%%%% paquetes%%%%%
6 \usepackage{amsmath}%Permite manipulacion de ecuaciones
7 \usepackage{graphicx}%Permite incluir graficos
8 \usepackage[spanish]{babel}%Para establecer idioma del doc
9 \usepackage[utf8x]{inputenc}%Para incluir las tildes
10
11 %%%%%% variables %%%%
12 \title{Mi primer artículo}
13 \author{Mi nombre}
14 \date{28 de Abril de 2021}
```

```
15 |
16 |
17 | %%%%%%%%%% cuerpo del documento%%%%%%%%%
18 | \begin{document}
19 |
20 | \maketitle %comando que construye el título del documento
21 |
22 | \section{Introducción}
23 | Soy la sección de la introducción
24 |
25 | \subsection{Esta es una sub-sección de la introducción}
26 |
27 | Soy una sub-sección!!!!
28 |
29 | \section{Otra sección}
30 |
31 | Aquí escribimos: hola mundo!
32 |
33 | \end{document}
```

Código 1.4: Archivo .tex en código LaTeX de un documento tipo artículo con secciones.

El comando `\maketitle` sirve para crear el título del documento basado en las variables globales `\title`, `\author` y `\date`. Naturalmente, éste debe estar dentro del cuerpo del documento. Seguidamente, se utilizaron los comandos `\section` y `\subsection` para separar el documento en secciones y sub-secciones, como lo podemos observar en la Figura 1.11, que muestra una parte del documento compilado.

Note que tanto las secciones como las sub-secciones están numeradas. Sin embargo, esto lo podemos remover adicionando un `*` a los comandos de secciones y subsecciones, es decir, escribiendo `\section*` y `\subsection*`.

### Entornos o ambientes

En términos generales, un entorno o ambiente es una sección del documento donde se definen ciertas reglas de composición tipográficas, las cuales pueden ser diferentes a las del documento principal. En general, en LaTeX, un entorno está estructurado de la siguiente forma:

```
\begin{nombre del entorno}

    Acciones en el entorno

\end{nombre del entorno}
```

Es decir, los comandos `\begin` y `\end` sirven para delimitar el entorno. Note que de lo anterior, es lógico deducir que el mismo cuerpo del documento también es un entorno, ya que satisface esta definición.



# Mi primer artículo

Mi nombre

28 de Abril de 2021

## 1. Introducción

Soy la sección de la introducción

### 1.1. Esta es una sub-sección de la introducción

Soy una sub-sección!!!!

## 2. Otra sección

Aquí escribimos: hola mundo!

Figura 1.11: Parte del documento pdf obtenido del código 1.4.

Existe una gran variedad de entornos en LaTeX. En este curso introduciremos tres de los más utilizados para escribir documentos científicos: el entorno matemático, el entorno de figuras y el entorno de referencias bibliográficas. Para consultar más información acerca de otros entornos de LaTeX y de su implementación, ver [1].

### 1.5.4. El entorno matemático

Sin lugar a dudas, el entorno matemático es uno de los más importantes dentro del lenguaje LaTeX debido a que nos permite escribir expresiones matemáticas con una alta calidad tipográfica. Este ambiente tiene varias formas de ser iniciado. Una de las formas más utilizadas, corresponde a iniciar el ambiente `equation`, como mostramos en el siguiente ejemplo:

```
1 %%%%%%%%%%%%%%% preambulo %%%%%%%%%%%%%%%
2 \documentclass{article}
3
4
5 %%%%%%% paquetes %%%%
6 \usepackage{amsmath} % Permite manipulacion de ecuaciones matematicas
7 \usepackage[spanish]{babel} % Para establecer idioma del doc.
8 \usepackage[utf8x]{inputenc} % Para incluir las tildes.
9
10 %%%%%%%%% variables %%%%
11 \title{Algunas ecuaciones}
```

```

12 \author{Bruno Diaz}
13 \date{}
14
15 %%%%%%%%%% cuerpo del documento%%%%%%%%%
16 \begin{document}
17
18 \maketitle
19
20 \section{Ecuaciones algebraicas}
21
22 Ecuación lineal:
23 %----- Entorno matematico-----%
24 \begin{equation}
25 \alpha x + 3 = y
26 \end{equation}
27 %-----%
28
29 Ecuación cuadrática:
30 %----- Entorno matematico-----%
31 \begin{equation}
32 x^2 + \beta x + 3 = 8
33 \end{equation}
34 %-----%
35
36 \end{document}

```

Código 1.5: Archivo .tex en código LaTeX de un documento tipo artículo con ecuaciones.

Una vez se compila el código LaTeX, se produce el texto en formato pdf mostrado en la Figura 1.12. Note que hemos declarado el valor de la variable `\date {}` en vacío, lo cual produce un documento sin fecha. Adicionalmente, hemos utilizado en el ambiente matemático los comandos `\alpha` y `\beta` para escribir las letras griegas  $\alpha$  y  $\beta$  respectivamente. Para ver una lista completa de los comandos para escribir caracteres griegos en el ambiente matemático, consultar [3] o simplemente buscar en internet.

Además del ambiente matemático `equation`, también podemos escribir ecuaciones matemáticas dentro de un texto encerrando la expresión por dos caracteres de signo dólar `$`. Adicionalmente, podemos utilizar sub-ambientes dentro del ambiente `equation` para introducir matrices o dividir ecuaciones. Veamos esto en el siguiente ejemplo:

```

1 %%%%%%%%%% preambulo%%%%%%%%%
2 \documentclass{article}
3
4 %%%%%%%%%% paquetes%%%%%%%%%
5 \usepackage{amsmath} % Permite manipulacion de ecuaciones matematicas
6 \usepackage[spanish]{babel} % Para establecer idioma del doc.
7 \usepackage[utf8x]{inputenc} % Para incluir las tildes.
8
9 %%%%%%%%%% variables %%%%%%%%%
10 \title{El entorno matemático}
11 \author{Mi nombre}
12
13 %%%%%%%%%% cuerpo del documento%%%%%%%%%
14 \begin{document}

```

# Algunas ecuaciones

Bruno Diaz

## 1. Ecuaciones algebraicas

Ecuación lineal:

$$\alpha x + 3 = y \quad (1)$$

Ecuación cuadrática:

$$x^2 + \beta x + 3 = 8 \quad (2)$$

Figura 1.12: Parte del documento pdf obtenido del código 1.5.

```
15
16 \section{Ecuaciones con sumatorias, derivadas e integrales}
17 En las siguientes ecuaciones asumimos que  $\alpha$  y  $\beta$  son numeros
    reales:
18 Ecuación con sumatorias:
19 \begin{equation}\label{ecuacion1}
20 f(n) = \sum_{n=1}^{\infty} \alpha n^2
21 \end{equation}
22
23 Ecuación con derivadas:
24 \begin{equation}\label{ecuacion2}
25 \frac{\partial y}{\partial x} + y = x^3
26 \end{equation}
27
28 Ecuación con integrales:
29 \begin{equation}\label{ecuacion3}
30 \int_a^b f(x) dx = F(b)-F(a)
31 \end{equation}
32
33 Problema: resuelva las ecuaciones \ref{ecuacion1}, \ref{ecuacion2} y \ref{
    ecuacion3}.
34
35 \section{Sub-ambientes matemáticos}
36 El sub-ambiente \texttt{split} sirve para dividir ecuaciones muy largas o
    escribir procedimientos:
37 \begin{equation}
38 \begin{split} %sub-entorno para dividir ecuaciones
39 A(x) &= \frac{\pi x^2}{2} \\
40 &= \frac{1}{2} \pi x^2
41 \end{split}
42 \end{equation}
43
```

```
44 El sub-ambiente \texttt{pmatrix} sirve para escribir matrices y vectores
45 \begin{equation}
46 \begin{pmatrix} 1 & 2 & 3 \\ \beta & \alpha & 0 \end{pmatrix}
47 1 & 2 & 3 \\
48 \beta & \alpha & 0
49 \end{pmatrix}
50 \end{equation}
51
52 \end{document}
```

Código 1.6: Archivo .tex en código LaTeX de un documento tipo artículo con split de ecuaciones y matrices.

Después de compilar el anterior código, obtenemos el documento de la Figura 1.13. Note que no hay título ni nombre de autor porque no incluimos el comando `\maketitle`.

## 1. Ecuaciones con sumatorias, derivadas e integrales

En las siguientes ecuaciones asumimos que  $\alpha$  y  $\beta$  son números reales:

Ecuación con sumatorias:

$$f(n) = \sum_{n=1}^{\infty} \alpha n^2 \quad (1)$$

Ecuación con derivadas:

$$\frac{\partial y}{\partial x} + y = x^3 \quad (2)$$

Ecuación con integrales:

$$\int_a^b f(x)dx = F(b) - F(a) \quad (3)$$

Problema: resuelva las ecuaciones 1, 2 y 3.

## 2. Sub-ambientes matemáticos

El sub-ambiente `split` sirve para dividir ecuaciones muy largas o escribir procedimientos:

$$\begin{aligned} A(x) &= \frac{\pi x^2}{2} \\ &= \frac{1}{2} \pi x^2 \end{aligned} \quad (4)$$

El sub-ambiente `pmatrix` sirve para escribir matrices y vectores

$$\begin{pmatrix} 1 & 2 & 3 \\ \beta & \alpha & 0 \end{pmatrix} \quad (5)$$

Figura 1.13: Parte del documento pdf obtenido del código 1.6.

Hemos utilizado los sub-entornos (o sub-ambientes) `split` y `pmatrix`, dentro del entorno `equation`, para descomponer ecuaciones e incluir matrices, respectivamente. Adicionalmente, resaltamos que usamos el comando `\label` para etiquetar las primeras tres expresiones matemáticas con los nombres `{ecuacion1}`, `{ecuacion2}` `{ecuacion3}`, lo que nos permitirá referenciarla en otra parte del documento mediante el comando `\ref`, como se muestra en la línea número 33 del código 1.6.

### 1.5.5. El entorno de figuras

Es bastante común incluir gráficos dentro de documentos científicos. Por esta razón, a continuación discutiremos brevemente cómo incluir gráficos en documentos producidos usando LaTeX.

En primer lugar, debemos cargar el paquete `\usepackage{graphicx}` el cual nos permite incluir gráficos en nuestro documentos. Ahora bien, para usar este paquete utilizaremos el ambiente `figure` y el comando `\includegraphics` el cual tiene la siguiente estructura:

```
\includegraphics[opcion1,opcion2,...]{ubicacion_de_la_imagen/nombre}
```

Este comando dispone de diferentes opciones para ajustar el tamaño y orientación del gráfico. Sin embargo, la más utilizada es la opción `scale`, la cual permite ajustar de forma escalada el tamaño del gráfico.

Antes de presentar un ejemplo de inclusión de una figura, debemos aclarar que si la figura se encuentra en el mismo directorio donde se encuentra nuestro código `.tex`, bastará con escribir el nombre de la figura que queremos incluir. También podemos omitir la extensión del archivo que contiene la figura, es decir, podemos simplemente escribir el nombre del archivo. Para ilustrar lo anterior considere el siguiente ejemplo en donde incluimos una figura con nombre `CirculoEnCuadrado.png` (en este documento no incluimos la fecha ni nombre del autor).

```
1 %%%%%%%%%%%%%%% preambulo%%%%%%%%%%%%%%
2 \documentclass{article}
3
4 %%%%%%%%% paquetes%%%%%%%%
5 \usepackage{graphicx} % Permite incluir graficos
6 \usepackage[spanish]{babel} % Para establecer idioma del doc.
7 \usepackage[utf8x]{inputenc} % Para incluir las tildes.
8
9 %%%%%%%%% variables %%%%
10 \title{Incluyendo un gráfico}
11 \author{Peter Parker}
12
13 %%%%%%%%%%%%%%% cuerpo del documento%%%%%%%%%%%%%%
14 \begin{document}
15
16 \maketitle
17
18 \section{Inclusión de gráfico}
19 A continuación incluimos un gráfico:
```

```
20 |
21 | %-----Incluyendo grafico en ambiente figura-----%
22 | \begin{figure}[h]
23 | \centering
24 | \includegraphics[scale=0.5]{imagen.pdf}
25 | \caption{Gráfico de la función  $z = \cos(x^2 + y^2)$ .}
26 | \label{etiquetadelgrafico}
27 | \end{figure}
28 | %-----%
29 |
30 | \end{document}
```

Código 1.7: Archivo .tex en código LaTeX de un documento tipo artículo con inclusión de figuras.

Una vez se compila el código, se produce el documento en formato pdf mostrado en la Figura 1.14. Note que los comandos `\caption` y `\label` definen el comentario debajo de la figura; y `\ref` define la etiqueta con la cual se puede hacer referencia a la figura dentro del texto, similar a como lo hacemos con las ecuaciones. Adicionalmente, hemos utilizado el comando `\centering` para alinear la figura en el centro y la opción `[h]` del ambiente `figure` para hacer que la figura aparezca (en lo posible) en el lugar que la llamamos.

Para ver más información de este paquete, consultar [1], además de la práctica `ejemplo_documento_en_Latex` ubicada en el repositorio del curso.

### 1.5.6. El entorno de las referencias bibliográficas

Para finalizar esta sección de introducción a LaTeX, abordamos una de las partes más importantes de todo documento científico, esto es, las referencias bibliográficas. En LaTeX, podemos incluir las referencias bibliográficas en un documento mediante dos formas. La primera forma consiste en adicionar la bibliografía de forma manual en formato de lista a través del ambiente `thebibliography`. La segunda forma consiste en la escritura y llamado de un archivo externo de tipo `bibtex`. Dada su estructura simple y por ser un ambiente, en este curso solo explicaremos la primera forma. La segunda forma, la cual es mucho más útil para manejar un gran número de citas bibliográficas, puede ser consultada en [1].

Para incluir referencias bibliográficas manualmente, llamamos el ambiente `thebibliography` al final del cuerpo del documento (realmente se puede ubicar en cualquier parte del documento, pero al final es la convención estándar). Luego, dentro del ambiente debemos utilizar el comando `\bibitem` el cual tiene la siguiente estructura:

```
\bibitem{etiqueta}{Información de la referencia bibliográfica}
```

Mediante este comando podemos generar una lista de referencias bibliográficas dentro del documento. La `etiqueta` sirve para citar la referencia dentro del documento utilizando el comando `\cite{etiqueta}`. Ilustramos lo anterior con el siguiente ejemplo:

## Incluyendo un gráfico

Peter Parker

24 de junio de 2021

### 1. Inclusión de gráfico

A continuación incluimos un gráfico:

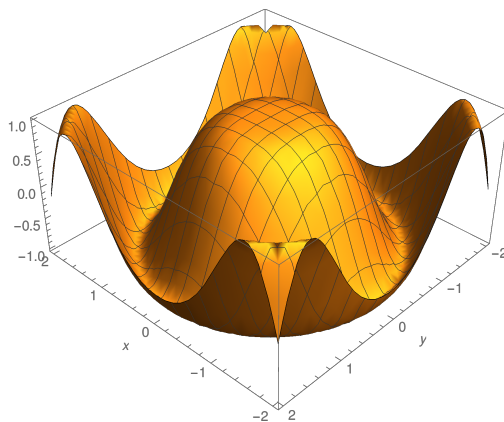


Figura 1: Gráfico de la función  $z = \cos(x^2 + y^2)$ .

Figura 1.14: Parte del documento pdf obtenido del código 1.7.

```
1 %%%%%%%%%%%%%%% preambulo%%%%%%%%%%%%%%
2 \documentclass{report}
3
4 %%%%%%%%% paquetes%%%%%%%%
5 \usepackage{graphicx} % Permite incluir graficos
6 \usepackage[spanish]{babel} % Para establecer idioma del doc.
7 \usepackage[utf8x]{inputenc} % Para incluir las tildes.
8
9 %%%%%%%%% variables %%%%
10 \title{Referencias bibliográficas}
11 \author{Leon Diaz}
12 \date{1 de junio de 2021}
13
14
15 %%%%%%%%%%%%%%% cuerpo del documento%%%%%%%%%%%%%%
16 \begin{document}
17
18 \maketitle
19
```

```

20 \section{Llamando una referencia bibliográfica}
21 Para poder comprender el significado de la ecuación de campo de Einstein
22
23 \begin{equation*}
24 R_{\{\mu \nu\}} - \dfrac{R}{2} g_{\{\mu \nu\}} = \kappa T_{\{\mu \nu\}}
25 \end{equation*}
26
27 sugerimos al lector ver \cite{einstein}. Adicionalmente, para hacer un
    programa numérico que resuelva esta ecuación tensorial utilizando el
    lenguaje de programación Python consultar \cite{python}.
28
29
30 %-----el entorno de la bibliografía-----
31 \begin{thebibliography}{99}
32
33 %----primera referencia-----
34 \bibitem{einstein}{
35 Albert Einstein. \textit{Zur Elektrodynamik bewegter K\"orper}. Annalen der
36 Physik, 322(10): 891-921, 1905.
37 }
38 %-----
39
40
41 %----segunda referencia-----
42 \bibitem{python}{
43 Python official web site \texttt{https://www.python.org/}
44 }
45 %-----
46
47 \end{thebibliography}
48
49
50 \end{document}

```

Código 1.8: Archivo .tex en código LaTeX de un documento tipo artículo con inclusión de referencias bibliográficas.

Después de compilar este código, obtenemos el documento pdf mostrado en Figura 1.15. Note que el ambiente `thebibliography` crea la sección **Referencias** y escribe a manera de lista las referencias creadas mediante el comando `\bibitem`. Adicionalmente, observe que utilizamos el ambiente `equation*` para escribir la ecuación de Einstein y evitar su numeración.

## 1.6. Ejercicios propuestos

1. Escriba el diagrama de flujo para un programa que liste todos los números primos que existen entre 1 y un cierto número natural positivo  $n$  escogido arbitrariamente en el programa. Adicionalmente, programe dicho algoritmo en Python.
2. Considere la expresión general de una ecuación cuadrática

$$ax^2 + bx + c = 0,$$



## Referencias bibliográficas

Leon Diaz

1 de junio de 2021

### 1. Llamando una referencia bibliográfica

Para poder comprender el significado de la ecuación de campo de Einstein

$$R_{\mu\nu} - \frac{R}{2}g_{\mu\nu} = \kappa T_{\mu\nu}$$

sugerimos al lector ver [1]. Adicionalmente, para hacer un programa numérico que resuelva esta ecuación tensorial utilizando el lenguaje de programación Python consultar [2].

### Referencias

- [1] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. Annalen der Physik, 322(10): 891-921, 1905.
- [2] Python official web site <https://www.python.org/>

Figura 1.15: Parte del documento pdf obtenido del código 1.8.

la cual tiene dos soluciones dadas por la fórmula

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Teniendo en cuenta lo anterior:

- (a) Proponga un diagrama de flujo para calcular de manera única las soluciones reales de la ecuación cuadrática para cualquier valor de los coeficientes  $a$ ,  $b$  y  $c$ .
  - (b) Programe en Python el diagrama de flujo descrito (sin ciclos **while**) en el literal anterior. Adicionalmente, el programa deberá hacer un gráfico de la función cuadrática para un cierto intervalo arbitrario  $[a, b]$ .
3. El numero  $\pi$  es una constante de la naturaleza que se puede calcular de diversas maneras. Una de ellas es usando la relación dada por,

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = \frac{\pi}{4}.$$

De aquí se puede aproximar el valor de  $\pi$  truncando la serie en un cierto valor  $n$  lo suficientemente grande. Teniendo en cuenta lo anterior:

- (a) Realice un diagrama de flujo para un programa que calcule esta constante con un error arbitrario escogido por el usuario SIN utilizar ciclos **for**.
- (b) Proponga otro diagrama de flujo para el mismo programa que calcule  $\pi$  con un error arbitrario escogido por el usuario SIN utilizar ciclos **while**.
- (c) Programe en Python el diagrama de flujo descrito (sin ciclos **while**) en el literal anterior tomando como referencia la constante `numpy.pi` del modulo `numpy`, es decir,

$$\text{Error} = |(\text{numpy.pi}) - (\text{valor numerico de pi})|.$$

4. Utilizando la regla de diferencias finitas centradas

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h},$$

donde  $h$  es un valor arbitrariamente pequeño, escriba un programa en Python que encuentre la derivada de cualquier función. Adicionalmente, para el caso de la función  $y(x) = x^2 + 2x + 1$ , realice un gráfico del comportamiento del error de la aproximación con respecto a la variable  $h$  tomando los valores  $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ .

5. Mediante la implementación de la regla de Simpson para integrales dada por,

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

- (a) Proponga el diagrama de flujo de un programa que encuentre numéricamente el área bajo la curva de la función  $f(x) = x^2 + 2$  en el intervalo  $[-1, 1]$ , asegurando que la diferencia entre valor obtenido y el valor exacto es menor que  $10^{-3}$ .
  - (b) Programe en Python el diagrama de flujo descrito en el literal anterior.
6. Imagine una persona que se mueve en el plano Cartesiano de acuerdo a los resultados del lanzamiento de una moneda al aire. Supongamos que la persona está inicialmente en el origen y que los movimientos se hacen de acuerdo al siguiente algoritmo:
- a) Lanza la moneda al aire.
  - b) Si en la moneda sale cara, avanzará una unidad a la derecha.
  - c) Si en la moneda sale sello, no se moverá.

Teniendo en cuenta las reglas de movimiento anteriores, realice un programa en el que la persona se mueva lanzando la moneda 50 veces y realice un gráfico que marque con un punto la posición final donde terminó la persona. Después de esto, repita este mismo proceso 1000 veces (siempre partiendo desde el origen) marcando con un punto todas las posiciones finales. Todas las posiciones finales deberán ir en un mismo gráfico.

Adicionalmente el programa deberá contar cuántos puntos hay en cada posición posible en el intervalo  $[0, 1000]$  y realizar un gráfico de esto (llamado *histograma*).

Finalmente, **superponga** este gráfico con el gráfico de la curva

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2},$$

con  $\mu = 1000 \cdot (1/2)$ ,  $\sigma^2 = \mu \cdot (1/2)$  y responda las siguientes preguntas:

- a) ¿Cuál es la posición en la que más veces termina la persona?
- b) ¿Coinciden las curvas? Explique por qué en un documento tipo artículo hecho en LaTeX de no más de dos páginas.

7. En 1970, el matemático John Conway diseñó el famoso “*Juego de la Vida*”, el cual consiste en un conjunto de reglas que imitan el comportamiento de un conjunto de células. El juego tiene lugar en una cuadrícula bi-dimensional formada por células **vivas** y **muertas**. Estos dos tipos de células pueden ser representadas en la cuadrícula por un círculo, para las células vivas, y una equis para las células muertas.

Si se supone que inicialmente existe en la cuadrícula una cierta configuración aleatoria inicial de células vivas y muertas, las reglas para evolucionar el sistema al siguiente tiempo (o generación) son como sigue:

- Si una célula viva está rodeada por más de tres células vivas, muere.
- Si una célula viva está rodeada por dos o tres células vivas, sobrevive.
- Si una célula viva está rodeada por menos de dos células vivas, muere.
- Si una célula está rodeada exactamente por tres células, se convierte en una célula viva.

Aplicando estas reglas en pasos sucesivos de tiempo, realice un programa en Python que muestre para varios instantes de tiempo (o generaciones) el patrón de células vivas y muertas en la cuadrícula bi-dimensional.

## Bibliografía

- [1] Rodrigo De Castro, *El universo LATEX*, Universidad Nacional de Colombia, 2003. (Citado en paginas 26, 29, and 34.)
- [2] <https://www.python.org/>. *Python 3.8.5 documentation*, Copyright 2001-2020, Python Software Foundation, 2020. (Citado en paginas 17 and 24.)
- [3] Varó, Andrés Marzal and Sevilla, Pedro García y Luengo, Isabel Gracia, *Introducción a la programación con Python 3*, Universitat Jaume I. Servei de Comunicació i Publicacions, 2014. (Citado en paginas 22 and 30.)
- [4] Torres, Anívar Chaves. *Aprenda a diseñar algoritmos. Vol. 1*. Universidad Nacional Abierta ya Distancia, 2017. (Citado en pagina 10.)

## Generalidades del modelamiento matemático

### 2.1. Que es un modelo matemático

Un modelo matemático es una representación simplificada de un fenómeno, sistema o proceso del mundo real utilizando conceptos y herramientas matemáticas. Se utiliza para comprender, analizar y predecir el comportamiento de dicho fenómeno o sistema [1].

**Definición 1** *La modelización es la aplicación de métodos para analizar problemas complejos del mundo real con el fin de hacer predicciones sobre lo que podría ocurrir con diversas acciones.*

De lo anterior, es evidente que los modelos matemáticos son herramientas poderosas para investigar y comprender diversos fenómenos científicos, ingenieriles, económicos y sociales. Permiten simular diferentes escenarios, realizar predicciones y tomar decisiones informadas. Además, los modelos matemáticos pueden ser validados y verificados mediante comparaciones con datos experimentales o empíricos, lo que ayuda a evaluar su capacidad para describir adecuadamente el fenómeno o sistema en cuestión.

Un modelo matemático se compone de variables, ecuaciones y relaciones que describen las interacciones entre esas variables. Estas variables pueden representar cantidades físicas, características, estados o comportamientos del fenómeno o sistema en estudio.

Los modelos matemáticos pueden adoptar diferentes formas, como ecuaciones diferenciales, ecuaciones algebraicas, sistemas de ecuaciones, diagramas de flujo, grafos, entre otros. Estos modelos se construyen a partir de suposiciones simplificadoras y aproximaciones, ya que es necesario simplificar la complejidad del mundo real para poder estudiarlo de manera más manejable y comprensible.

### 2.2. Clasificación de los modelos matematicos

Existen diferentes tipos de modelos matemáticos, cada uno con sus propias características y aplicaciones. En particular, en este curso clasificaremos los modelos matematicos entre

determinísticos y probabilísticos los cuales definimos a continuación.

### 2.2.1. Modelos Determinísticos:

Estos modelos utilizan ecuaciones matemáticas para describir las relaciones deterministas entre las variables del sistema. Proporcionan predicciones precisas y reproducibles del comportamiento del sistema y dependen de las condiciones iniciales del sistema. Algunos ejemplos de modelos determinísticos son descritos en la siguiente tabla:

Modelo	Descripción
Movimiento de un proyectil	Modela el movimiento de un proyectil en el aire utilizando ecuaciones que consideran la velocidad inicial, el ángulo de lanzamiento y la gravedad.
Modelos de crecimiento poblacional	Describen el crecimiento de una población basándose en suposiciones sobre la tasa de natalidad, la tasa de mortalidad y la capacidad de carga del entorno.
Modelo de movimiento planetario	Utiliza las leyes de Kepler y la ley de la gravitación universal de Newton para describir el movimiento de los planetas en el sistema solar.
Modelos de propagación de enfermedades	Predicen la propagación de enfermedades infecciosas en una población, considerando la tasa de infección, la tasa de recuperación y la interacción entre individuos.
Modelo de Equilibrio Químico	Este modelo describe la relación entre las concentraciones de reactivos y productos en una reacción química en equilibrio. Se basa en las leyes de acción de masas y equilibrio químico.

### 2.2.2. Modelos probabilísticos:

A diferencia de los modelos determinísticos, los modelos no deterministas, también llamados estocásticos, tienen en cuenta la incertidumbre y aleatoriedad en el sistema. Utilizan conceptos de probabilidad y estadística para modelar eventos o procesos aleatorios. Algunos ejemplos de modelos probabilísticos son descritos en la siguiente tabla:

Modelo	Descripción
Modelo de Caminata Aleatoria	Este modelo describe el movimiento de una partícula en un espacio en el que cada paso es aleatorio. Se puede utilizar para simular el comportamiento de partículas en física o el movimiento de precios en mercados financieros.
Modelo de Monte Carlo	Este modelo utiliza simulaciones estocásticas para resolver problemas complejos y calcular probabilidades. Se basa en la generación de números aleatorios para simular diferentes escenarios y obtener estimaciones numéricas de integrales.
Modelo de Predicción Meteorológica	Los modelos meteorológicos utilizan ecuaciones y algoritmos estocásticos para predecir el clima. Dado que el clima es un sistema complejo con múltiples variables y factores de incertidumbre, se emplean enfoques no determinísticos para tener en cuenta la variabilidad y las incertidumbres inherentes.
Modelos de estimación de poblaciones	Utilizados para inferir y estimar características o parámetros de una población a partir de una muestra representativa de la misma. Estos modelos son especialmente útiles cuando no es factible o práctico obtener información de todos los individuos de una población.

### 2.2.3. Tipos de modelos: continuos o discretos

Tanto los modelos determinísticos como los probabilísticos pueden estar sub-divididos en dos tipos dependiendo de si sus variables son discretas o continuas.

Un modelo se dice discreto si representa sistemas en los que las variables cambian solo en puntos discretos en el tiempo o en el espacio. Es decir, las variables toman puntos representables por sucesiones contables de números. Por ejemplo si  $x$  es una variable de un modelo discreto, esta puede tomar valores en la sucesión de números  $1, 2, 3, \dots$  etc.

Por otro parte, a diferencia de los modelos discretos, los modelos continuos consideran cambios suaves y continuos en las variables del sistema. Es decir, las variables toman valores en intervalos de la recta real. Por ejemplo si  $x$  es una variable de un modelo discreto, esta puede tomar valores en un intervalo  $[a, b] \in \mathbb{R}$ .

En conclusión: un modelo determinístico puede ser de tipo continuo o discreto dependiendo de los valores que puedan tomar sus variables. Lo anterior también aplica a los modelos probabilísticos

## 2.3. Pasos sugeridos para hacer un modelo matematico

Los pasos para hacer un modelo matemático pueden variar según el problema y el enfoque específico, pero en general, se pueden considerar los siguientes pasos:

1. **Definir el problema:** Comprender claramente el problema que se va a modelar y establecer los objetivos del modelo. Identificar las variables relevantes y los aspectos clave del fenómeno o sistema que se quiere estudiar.
2. **Formular las suposiciones:** Hacer suposiciones simplificadoras y establecer los límites del modelo. Estas suposiciones ayudarán a simplificar y enfocar el problema, aunque es importante tener en cuenta sus limitaciones y considerar su validez en relación con la realidad.
3. **Desarrollar las ecuaciones o relaciones matemáticas:** Identificar las relaciones matemáticas que describen el comportamiento de las variables y cómo interactúan entre sí. Esto puede implicar la formulación de ecuaciones diferenciales, ecuaciones algebraicas u otras expresiones matemáticas.
4. **Establecer condiciones particulares del sistema:** Especificar las condiciones del sistema y los límites dentro de los cuales se aplicará el modelo. Estas condiciones y límites pueden incluir valores iniciales de las variables, restricciones físicas o cualquier otra restricción relevante para el problema.
5. **Resolver las ecuaciones del modelo:** Utilizar métodos matemáticos o herramientas numéricas para resolver las ecuaciones o simular el comportamiento del modelo. Esto puede implicar técnicas numéricas, análisis de sistemas dinámicos o cualquier otro enfoque adecuado para el problema en cuestión.
6. **Validar y ajustar el modelo:** Comparar los resultados del modelo con datos empíricos o evidencia experimental. Evaluar la precisión y eficacia del modelo en la descripción y predicción del fenómeno o sistema. Si es necesario, realizar ajustes o mejoras en el modelo para mejorar su validez y confiabilidad.

Es importante destacar que estos pasos no son necesariamente lineales y pueden requerir iteraciones y refinamientos a lo largo del proceso de modelización. Además, cada paso puede involucrar diferentes técnicas matemáticas y herramientas específicas según el problema y el enfoque utilizado.

### 2.3.1. Ejemplos de construccion de modelos

#### Modelo determinista

Supongamos que deseamos modelar el crecimiento de una población de bacterias en un entorno controlado. Asumiremos que el crecimiento de la población está limitado por los recursos



disponibles y se rige por una tasa de crecimiento proporcional a la población actual.

1. **Definir el problema:** Queremos modelar el crecimiento de la población de bacterias en un entorno controlado.
2. **Formular suposiciones:** Suponemos que el crecimiento de la población está limitado por los recursos disponibles y que la tasa de crecimiento es proporcional a la población actual.
3. **Desarrollar las ecuaciones o relaciones matemáticas:** Llamemos  $P(t)$  a la población de bacterias en función del tiempo  $t$ . Podemos expresar el crecimiento de la población mediante la ecuación diferencial:

$$\frac{d}{dt}P(t) = rP(t)$$

Donde  $r$  es la tasa de crecimiento proporcional.

4. **Establecer condiciones particulares del sistema:** Especificamos la condición inicial de la población de bacterias en un tiempo inicial  $t_0$ , por ejemplo,  $P(t_0) = P_0$ .
5. **Resolver las ecuaciones del modelo:** Podemos resolver esta ecuación diferencial utilizando métodos analíticos o numéricos. En este caso, la solución es una función exponencial (ver []):

$$P(t) = P_0 e^{r(t-t_0)}$$

Donde  $\exp$  es la función exponencial y  $r$  es la tasa de crecimiento.

6. **Validar y ajustar el modelo:** Comparamos los resultados del modelo con datos empíricos o experimentales sobre el crecimiento de bacterias en un entorno controlado. Si los resultados del modelo se ajustan bien a los datos observados, consideramos que el modelo es válido. Si no se ajusta, podemos realizar ajustes en los parámetros o revisar las suposiciones para mejorar la precisión del modelo.

Este es solo un ejemplo básico de construcción de un modelo matemático, pero los pasos y enfoques pueden variar según el problema y el contexto específico. La construcción de modelos matemáticos puede ser más compleja en situaciones del mundo real, pero este ejemplo proporciona una idea general del proceso.

### Modelo probabilístico

Supongamos que deseamos modelar la probabilidad de que un estudiante apruebe un examen en función de su tiempo de estudio. Asumiremos que la probabilidad de aprobación está influenciada por el tiempo de estudio y sigue una distribución binomial.

1. **Definir el problema:** Queremos modelar la probabilidad de aprobación de un examen en función del tiempo de estudio.
2. **Formular suposiciones:** Suponemos que la probabilidad de aprobación está influenciada por el tiempo de estudio y sigue una distribución binomial.
3. **Desarrollar las ecuaciones o relaciones matemáticas:** Identificar las variables relevantes, las cuales son La variable independiente es el tiempo de estudio ( $X$ ), y la variable dependiente es la probabilidad de aprobación ( $P$ ).
4. **Establecer condiciones particulares del sistema:** Establecemos las relaciones probabilísticas de las variables supuniendo que la probabilidad de aprobación sigue una distribución binomial, donde la probabilidad de éxito ( $p$ ) está relacionada con el tiempo de estudio. Podemos expresar esto como:

$$P(X) = \binom{n}{X} \cdot p^X \cdot (1 - p)^{n-X}$$

Donde:

- $C(X)$  es el coeficiente binomial.
  - $p$  es la probabilidad de éxito.
  - $X$  es el número de éxitos (aprobaciones) en  $n$  ensayos (exámenes).
  - $(1 - p)$  es la probabilidad de fracaso.
5. **Resolver las ecuaciones del modelo:** Estimamos los parámetros para construir el modelo probabilístico. Para ello necesitamos estimar el valor de  $p$ , la probabilidad de éxito, a partir de datos o información previa.
  6. **Validar y ajustar el modelo:** Comparamos los resultados del modelo con datos observados sobre la probabilidad de aprobación en función del tiempo de estudio. Si el modelo se ajusta bien a los datos, consideramos que es válido. Si no se ajusta, podemos realizar ajustes en los parámetros o revisar las suposiciones para mejorar la precisión del modelo.

La construcción de modelos probabilísticos puede ser más compleja y puede implicar el uso de distribuciones de probabilidad específicas, técnicas de estimación de parámetros y validación estadística.

## 2.4. Los datos y los modelos matemáticos

Los datos son fundamentales para construir y validar un modelo porque proporcionan información empírica y objetiva sobre el fenómeno o sistema que se desea modelar. A continuación se presentan algunas razones por las cuales los datos son importantes en la construcción de un modelo:

- *Fundamentación empírica:* Los datos nos permiten comprender cómo se comporta el fenómeno en la realidad. Nos brindan información sobre las relaciones, patrones y tendencias presentes en los datos observados, lo que es crucial para construir un modelo matemático que refleje adecuadamente el comportamiento del sistema.
- *Verificación y validación:* Los datos nos permiten verificar y validar el modelo. Al comparar los resultados del modelo con los datos reales, podemos evaluar su precisión y su capacidad para capturar el comportamiento del sistema. Si el modelo no se ajusta bien a los datos, es posible que necesite ser ajustado, mejorado o incluso descartado.
- *Parámetros y estimaciones:* Los datos proporcionan información que nos permite estimar los parámetros del modelo. Por ejemplo, en un modelo de regresión, los datos nos permiten estimar los coeficientes de las variables predictoras. Estas estimaciones son fundamentales para comprender cómo influyen las variables en el comportamiento del sistema y realizar predicciones o proyecciones.
- *Generalización y extrapolación:* Los datos nos permiten generalizar los resultados del modelo a situaciones o contextos más amplios. Al basarnos en datos representativos y diversos, podemos inferir cómo se comportará el sistema en situaciones similares, incluso en aquellas en las que no tenemos datos observados directamente. Esto nos permite realizar predicciones o tomar decisiones informadas en nuevos escenarios.
- *Mejora y refinamiento:* Los datos nos permiten mejorar y refinar continuamente el modelo. A medida que obtenemos nuevos datos y evidencia empírica, podemos actualizar y ajustar el modelo para que sea más preciso y confiable. Los datos nos brindan retroalimentación sobre la efectividad del modelo y nos ayudan a identificar posibles áreas de mejora.

En resumen, los datos son esenciales en la construcción de un modelo porque proporcionan una base empírica, verifican y validan el modelo, permiten estimar parámetros, generalizar los resultados, y mejorar y refinar continuamente el modelo. Sin datos, los modelos pueden carecer de fundamento y precisión, limitando su utilidad para comprender y predecir el comportamiento de los sistemas reales.

## Bibliografía

- [1] Driscoll, H., et. al., (2016), *Introduction to computational science: modeling and simulation for the sciences*, Procedia Engineering 147 884-889. ( No citado.)
- [2] Shiflet, A. B., Shiflet, G. W. (2014). *Introduction to computational science: modeling and simulation for the sciences*. Princeton University Press. ( No citado.)

## Manejo básico de datos

En muchas situaciones reales, aparecen cantidades que cambian proporcionalmente unas con otras, bien sea en forma exacta o aproximadamente. Otra situación frecuente e importante es la búsqueda de una relación funcional entre cantidades de las cuales solo conocemos un número finito de relaciones como el valor diario del dólar y del barril de petróleo, el peso y la talla mensual de un bebé o la posición de un cuerpo en movimiento.

En este capítulo, trataremos de modelar estas dos clases de situaciones comunes presentando así lo que hemos denominado modelos matemáticos básicos. Iniciaremos con modelos donde hayan cantidades que razonablemente se puedan pensar como proporcionales; y terminaremos el capítulo con algunas formas de aproximar la función que podría estar detrás de unos datos observados.

### 3.1. Proporcionalidad de datos

Aunque la proporcionalidad la podemos pensar en cualquier número de cantidades, por simplicidad y sin pérdida de generalidad, tratemos de entenderla entre solo dos cantidades. Empecemos pensando en la relación que existe entre los siguientes pares de cantidades: la talla y el peso de un niño, la posición del extremo de un péndulo en movimiento, el costo de compra y el número de naranjas a comprar, el número de trabajadores y el tiempo que utilizan para hacer una obra, el número de caídas diarias de un bebe durante su segundo año de vida.

1. **Talla y peso de un niño.** Es de esperarse que el crecimiento durante la infancia esté acompañada de un aumento de peso, pero este aumento no siempre es constante por cada centímetro que crezca el niño; es más, generalmente, llega un momento en que hay aumento de peso sin aumento en la talla.
2. **Posición del extremo de un péndulo en movimiento.** Tomando como referencia el punto fijo del péndulo, vemos que el otro extremo se desplaza, horizontalmente, a un lado hasta llegar a un punto donde se devuelve, pasa por el punto fijo nuevamente y se desplaza hacia el otro lado hasta un punto donde también se devuelve, pasa por el punto fijo nuevamente y repite el movimiento. Verticalmente, el extremo móvil del péndulo, sube un poco, luego baja, vuelve a subir, y repite el movimiento.

3. **Costo de compra y número de naranjas compradas.** Es claro que el costo aumenta a medida que compramos más naranjas; además, si las naranjas son similares en tamaño, es de esperarse que el costo aumente lo mismo por cada naranja adicional que compremos.
4. **Número de trabajadores y tiempo que utilizan para hacer una obra.** Dentro de unos límites razonables, es de esperarse que entre más trabajadores se contraten, menos tiempo se demoren haciendo la obra. Es más, suponiendo que los trazadores son similares en cuanto al trabajo que realizan, si un trabajador toma cierto tiempo para hacer una obra, es de esperarse que dos lo hagan en la mitad de ese tiempo y tres, en un tercio.
5. **Número de caídas diarias de un bebe durante su segundo año de vida.** Cuando un bebe empieza a caminar, poco después de cumplir su primer año, su inexperiencia y debilidad muscular solo le permite dar unos pasos antes de terminar en el suelo, pero a medida que va ganando experiencia y fortaleciendo muscular, son más largas sus caminadas y menos las caídas.

¿Cuál o cuáles de estos pares de cantidades podemos modelar como una relación de proporcionalidad? En caso afirmativo ¿cómo lo hacemos?

Antes de contestar, revisemos el concepto de proporcionalidad. Hay dos tipos de proporcionalidad: *Directa e inversa*, aunque generalmente llamamos *proporcionalidad* a la primera. Dado que muchas veces confundimos los conceptos de proporcionalidad con los conceptos de relación creciente y decreciente, recordemos también estos conceptos.

### 3.1.1. Proporcionalidad directa

Como el concepto de proporcionalidad directa (o simplemente proporcionalidad), está relacionado con el de relación creciente entre dos variables, empecemos por este último concepto.

**Relación creciente:** Diremos que entre dos cantidades existe una relación creciente si al aumentar (o disminuir) la una, aumenta (o disminuye) la otra<sup>1</sup> [4].

**Proporcionalidad directa:** Diremos que entre dos cantidades *con relación crecientes* existe proporcionalidad directa si el cociente entre cualquier par de valores correspondientes de ellas es siempre el mismo; este cociente lo llamamos constante de proporcionalidad. Observe que en este caso, el aumento en una cantidad por unidad de la otra es precisamente dicha constante. En consecuencia, si representamos las cantidades mediante las variables  $y$  y  $x$ , el modelo que representa las cantidades con relación proporcional directa ( $y \propto x$ ) sería,

$$\frac{y}{x} = k, \text{ o equivalentemente, } y = kx, \text{ siendo } k > 0 \text{ la constante de proporcionalidad}$$

Geométricamente, lo anterior se representa como una línea recta que pasa por el origen (ver Fig. 3.1 y el programa en Python que aparece a continuación, el cual logra dicha figura).

<sup>1</sup>Formalmente, se dice que la función  $f : [a, b] \rightarrow R$  es creciente, si y solo si, para todo  $x, y \in [a, b]$  tales que  $x \geq y$ , se cumple que  $f(x) \geq f(y)$ .

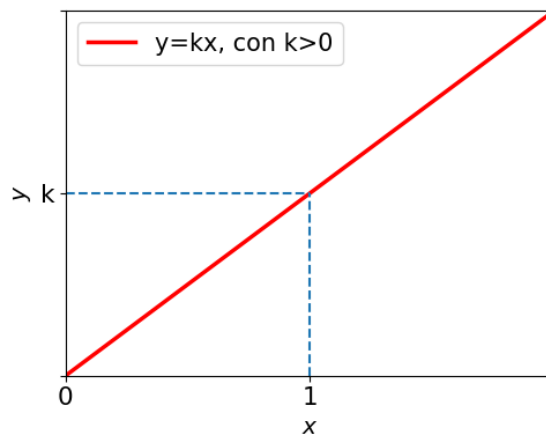


Figura 3.1: Proporcionalidad directa. Observe que, cuando  $x = 1$ ,  $y = k$ , siendo  $k > 0$  la constante de proporcionalidad.

```

1 # Importa librerias necesarias
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 #Genera la recta a graficar
6 x = np.linspace(0, 5, 100)
7 k = 4
8 y = k*x
9
10 #Genera puntos para graficar lineas punteadas
11 a = [1,1]
12 b = [0,4]
13 c = [0,1]
14 d = [4,4]
15
16 # Ajusta tamaño figura
17 plt.style.use('seaborn-poster')
18 fig = plt.figure(figsize = (10, 5))
19
20 #Crea el grafico
21 plt.axvline(x=0, color='black', linewidth=1) # Traza el eje vertical
22 plt.plot(a,b,'-', color='black', linewidth=1) # Traza el eje horizontal
23 plt.plot(x, y, color='black') # Grafica los datos x, y
24 plt.plot(a,b,'--', color='black') # Grafica linea punteada horizontal
25 plt.plot(c,d,'--', color='black') # Grafica linea punteada vertical
26 plt.text(1.8, 5, r'$y = kx$, con $k>0$', fontsize=15) #Escribe una etiqueta
27 #plt.title('Proporcionalidad directa') # Posible titulo de la grafica
28 plt.xlabel('x') #Coloca nombre del eje horizontal
29 plt.ylabel('y') #Coloca nombre del eje vertical
30 plt.xticks([1]) #Coloca una marca en el eje horizontal
31 plt.yticks([0,4],["0","k"]) #Coloca marcas en el eje vertical
32 plt.xlim([0, 3]) # Define limites del eje horizontal
33 plt.ylim([0, 10]) # Define limites del eje vertical
34 plt.xticks([]) # Para que no aparezca escala en el eje horizontal
35 plt.yticks([]) # Para que no aparezca escala en el eje vertical

```

```
36 plt.show() # Muestra el grafico en una ventana
```

A la luz de estos conceptos, revisemos las relaciones entre los pares de cantidades anteriores: Para empezar, los pares 1 y 3 son candidatos a tener proporcionalidad directa (Por qué?).

Como el aumento en el costo de compra es más o menos el mismo por cada naranja comprada, estas dos cantidades pueden ser modeladas usando la proporcionalidad directa; esto es:

*Variables:* Sean  $n$  el número de naranjas compradas y  $C(n)$  el costo de compra de  $n$  naranjas.

*Modelo matemático:* Bajo el supuesto de proporcionalidad directa entre estas dos variables, el modelo sería

$$C(n) = kn$$

¿Cuál sería el modelo matemático para el par de cantidades 1?

### 3.1.2. Proporcionalidad inversa

Al igual que en el caso anterior, como el concepto de proporcionalidad inversa, está relacionado con el de relación decreciente entre dos variables, empecemos por este último concepto.

**Relación decreciente:** Diremos que entre dos cantidades existe una relación decreciente si al aumentar (o disminuir) la una, disminuye (o aumenta) la otra<sup>2</sup> [4].

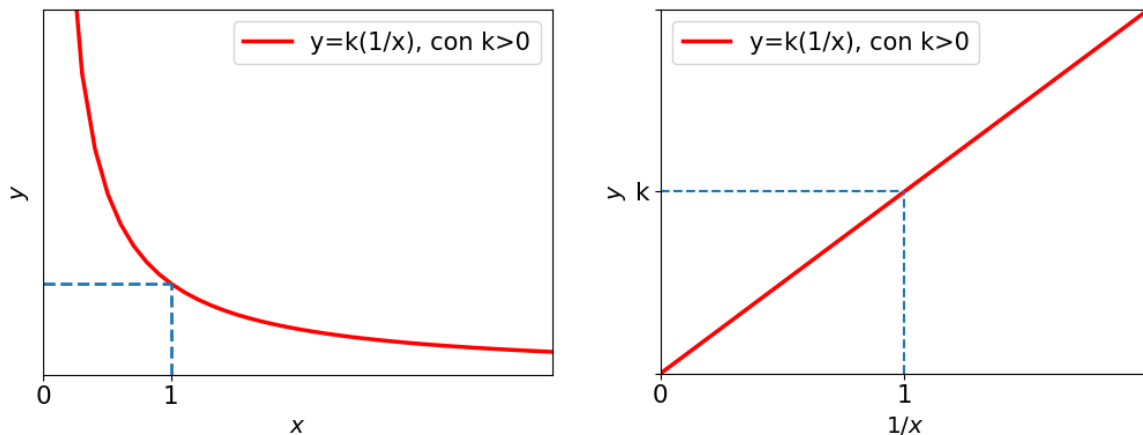


Figura 3.2: Proporcionalidad inversa. Observe que, cuando  $x = 1$ ,  $y = k$ , siendo  $k > 0$  la constante de proporcionalidad. Izq.: Gráfica de  $y$  Vs.  $x$ . Der.: Gráfica de  $y$  Vs.  $1/x$ .

**Proporcionalidad inversa:** Diremos que entre dos cantidades con relación decrecientes existe proporcionalidad inversa si el producto entre cualquier par de valores de ellas es siempre el mismo, este producto lo llamamos constante de proporcionalidad. Observe que en este caso, el aumento en una cantidad por unidad del inverso de la otra es precisamente dicha constante. En consecuencia, si representamos las cantidades mediante las variables  $y$  y  $x$ , el modelo que representa las cantidades con relación proporcional inversa ( $y \propto \frac{1}{x}$ ) sería,  $xy = k$ , o equivalentemente,  $y = k\frac{1}{x}$ , siendo  $k > 0$  la constante de proporcionalidad.

<sup>2</sup>Formalmente, se dice que la función  $f : [a, b] \rightarrow R$  es decreciente, si y solo si, para todo  $x, y \in [a, b]$  tales que  $x \geq y$ , se cumple que  $f(x) \leq f(y)$ .



Geoméricamente, lo anterior se representa por una hipérbola en las variables  $x, y$ ; o por una línea recta (rescalamiento) que “tiende” al origen en las variables  $y, \frac{1}{x}$  (ver Fig. 3.2).

A la luz de estos conceptos, revisemos las relaciones entre los pares de cantidades anteriores: Continuemos con los pares 4 y 5, los cuales son candidatos a tener proporcionalidad inversa (Por qué?) ¿Qué tipo de proporcionalidad puede tener el par 2?

Examinando el par de cantidades 4, observamos que el tiempo es más o menos proporcional al inverso del número de trabajadores, estas dos cantidades pueden ser modeladas mediante la proporcionalidad inversa; esto es:

**Variables:** Sean  $n$  el número de trabajadores y  $T(n)$  el tiempo de elaboración de la obra usando  $n$  trabajadores.

**Modelo matemático:** Bajo el supuesto de proporcionalidad inversa entre estas dos variables, el modelo sería

$$T(n) = k \frac{1}{n}$$

¿Cuál sería el modelo matemático para el par de cantidades 5?

### 3.1.3. Proporcionalidades ocultas

En este proceso de encontrar proporcionalidad, además de buscarla entre las cantidades correspondientes de dos variables dadas, es bastante útil el tratar de establecerla entre una variable y una función conocida de la otra, o entre funciones conocidas de las variables. Aunque no son los únicos casos, a continuación analizamos dos tipos de relación entre dos variables que conducen a este tipo de proporcionalidad funcional entre las variables, lo cual llamamos *proporcionalidades ocultas*.

**Caso 1:** *Función exponencial* ( $y = e^{kx}$ )

Al tomar logaritmo natural en ambos lados de la igualdad que establece la relación exponencial entre dos variables, obtenemos

$$\ln y = kx$$

lo cual nos indica que la variable  $z = \ln y$  es proporcional a la variable  $x$ , lo cual, geoméricamente, lo podemos apreciar en la Fig. 3.3.

Si recordamos que  $y = e^{kx} = (e^k)^x = a^x$ , donde  $a = e^k$  es la base de la relación exponencial entre las variables  $x, y$ , esta aproximación, junto con los métodos de ajuste de la próxima sección, nos permitirán calcular la base de una relación entre dos variables que sospechemos que es exponencial.

Dado el par de tablas de datos 3.1, determinar en cual o cuales de ellas, se puede sugerir una relación exponencial entre las variables.

**Análisis:** Grafiquemos los datos de cada una de las tablas dadas, tomando como variable dependiente el logaritmo natural de la segunda variable<sup>3</sup> (Fig. 3.4, ver como lograr el gráfico

---

<sup>3</sup>Este tipo de gráfica se conoce con el nombre de escala logarítmica en el Eje  $Y$  o gráfica en *escala semilogarítmica*

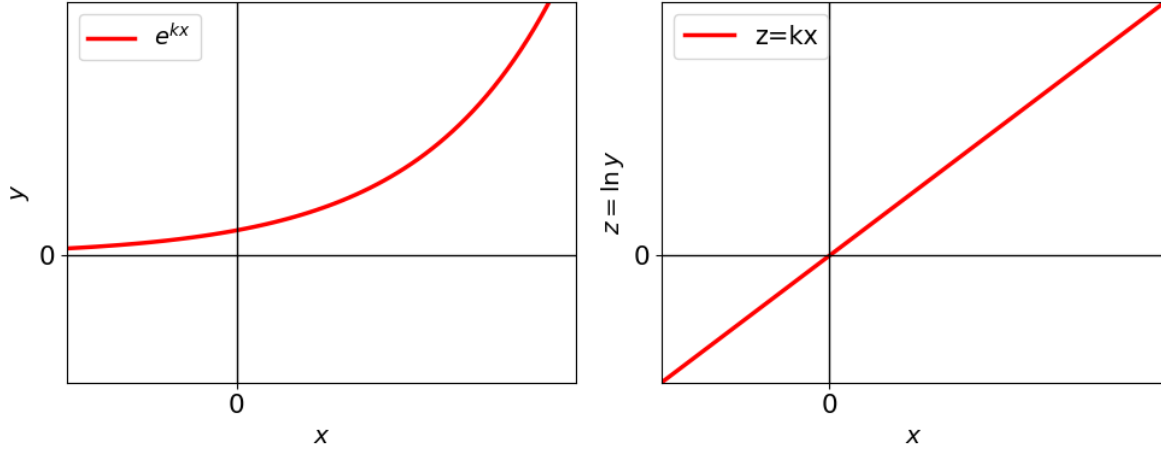


Figura 3.3: Función exponencial  $y = e^{kx}$  (Der) y su proporcionalidad oculta  $z = \ln y = kx$  (Izq)

de una tabla en Ejemplo ??).

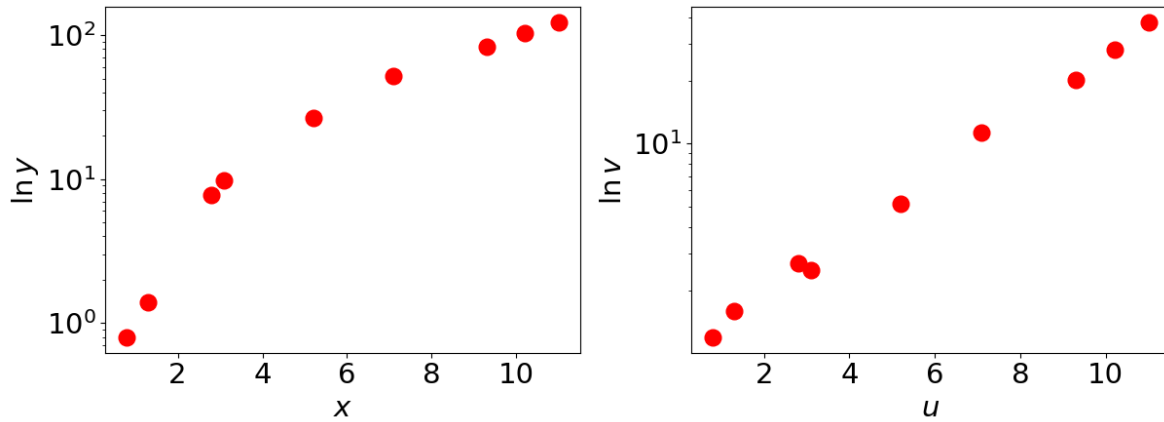


Figura 3.4: Gráfica de los datos de las tablas dadas, tomando como variable dependiente el logaritmo natural de la segunda variable (escala semilogarítmica).

Observemos que la gráfica de la izquierda de la Fig. 3.4 muestra puntos muy poco alineados, mientras que la gráfica de la derecha muestra que los puntos están más o menos alineados.

**Conclusión:** La anterior observación nos permite sugerir que las variables  $u, v$  (tabla de la derecha) tienen una relación exponencial; más precisamente, podemos sugerir que  $v = e^{ku}$ , para alguna constante  $k$ , como modelo de la relación funcional entre estas dos variables (ver Ejercicio 11 para el cálculo de la base de esta relación).

**Caso 2:** *Función potencial* ( $y = x^\alpha$ ) Al tomar logaritmo natural en ambos lados de la igualdad que establece la relación potencial entre dos variables, obtenemos

$$\ln y = \alpha \ln x$$

lo cual nos indica que la variable  $z = \ln y$  es proporcional a la variable  $t = \ln x$ , lo cual, geoméricamente, lo podemos apreciar en la Fig. 3.5.

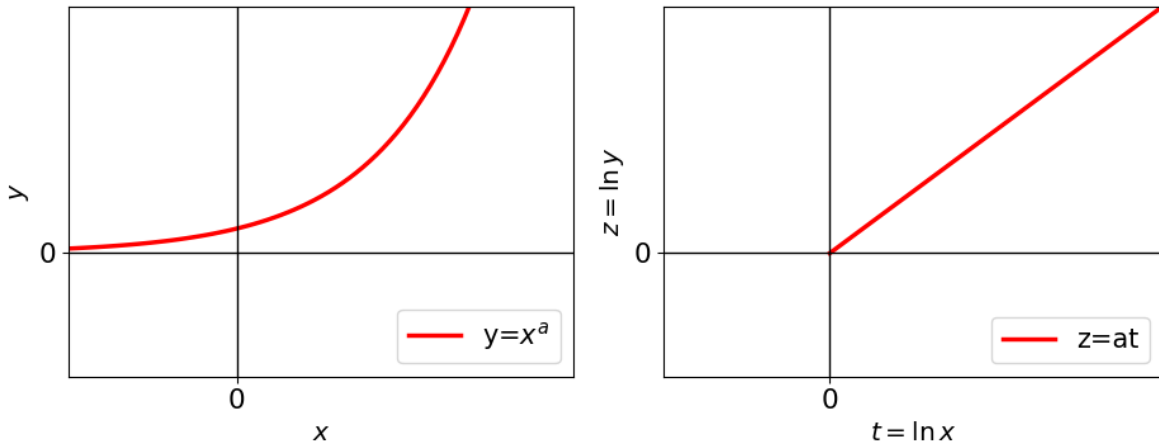


Figura 3.5: Función potencial  $y = x^a$  (Izq) y su proporcionalidad oculta  $z = \ln y = a \ln x = at$  (Der).

Como en el caso anterior, si recordamos que  $a$  es *el exponente* de la relación potencial entre las variables  $x, y$ , esta aproximación, junto con los métodos de ajuste de la próxima sección, nos permitirán calcular el exponente de una relación entre dos variables que sospechemos que es potencial.

**Ejemplo:** Dado el par de tablas de datos 3.2. determinar en cual o cuales de ellas, se puede sugerir una relación potencial entre las variables.

**Análisis:** Grafiquemos los datos de cada una de las tablas dadas, tomando como variables el logaritmo natural de las dos variables<sup>4</sup> (Fig. 3.6, ver como lograr el gráfico de una tabla en Ejemplo ??).

Observemos que ambas gráficas de la Fig. 3.6 muestran puntos más o menos alienados.

**Conclusión:** La anterior observación nos permite sugerir que tanto las variables  $x, y$  como las variables  $u, v$  de las tablas dadas tienen una relación potencial; más precisamente, podemos sugerir que  $y = x^{\alpha_1}$  y  $v = u^{\alpha_2}$ , para algunas constantes  $\alpha_1, \alpha_2$ , como modelos de la relación funcional entre los dos pares de variables dados (ver Ejercicio 12 para el cálculo de la base de esta relación).

<sup>4</sup>Este tipo de gráfica se conoce con el nombre de escala logarítmica en ambos ejes o gráfica en *escala logarítmica*

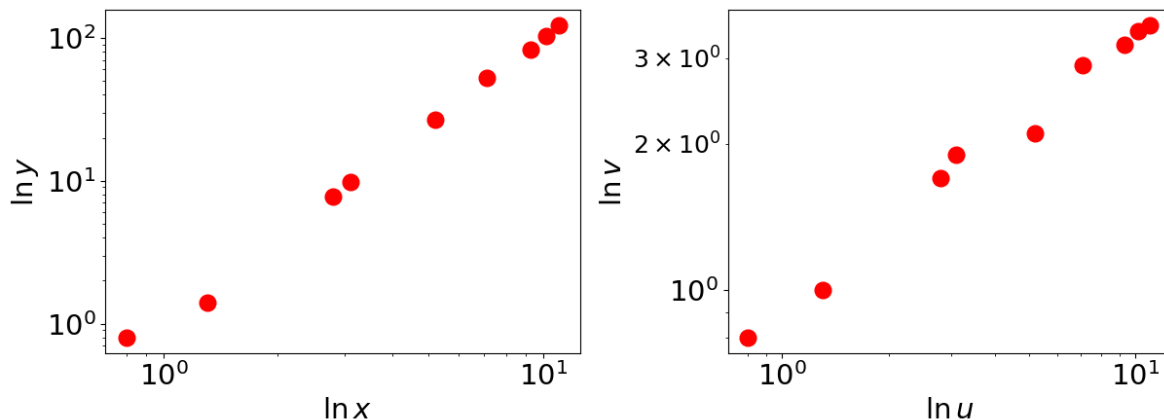


Figura 3.6: Gráfica de los datos de las tablas dadas, tomando como variables el logaritmo natural de las dos variables (escala logarítmica).

## 3.2. Ajuste de datos

En muchas situaciones, conocemos valores de cantidades que están relacionadas así no conozcamos explícitamente dicha relación, por lo que la pregunta que surge es precisamente alrededor de esta relación desconocida. Dependiendo de la información que tengamos sobre la situación o los supuestos que hagamos sobre la relación, podemos plantear diversos tipos de modelos y métodos de solución para encontrar dicha relación desconocida. Como método exploratorio al determinar la relación desconocida contamos con los modelos gráficos; y posteriormente para tratar de encontrar la mejor aproximación lineal de los datos se hacemos uso del método de los mínimos cuadrados. Adicionalmente podemos abordar los conceptos de interpolación, en la búsqueda de nuevos puntos a partir del conocimiento de un conjunto de puntos previos, y la extrapolación, para estimar más allá del intervalo de observación original el valor de una variable en base a su relación con otra, estudiando los casos para funciones lineales y no lineales, para esta última a través de la interpolación polinómica de Lagrange. Aunque hay otras formas de modelar una situación para tratar de encontrar la relación oculta en los datos de cantidades asociadas a una situación, veamos mas detalles de estos casos mencionados.

Sin pérdida de generalidad, supongamos que solo hay dos cantidades involucradas en la situación real: Sean  $(x_i, y_i)$ , para  $i = 1, 2, \dots, n$ , los valores observados de dos cantidades involucradas en una situación dada, en la cual se sabe que existe una función  $f$  tal que  $y = f(x)$ , y el problema consiste en encontrar  $f$ .

### 3.2.1. Gráficas

Tener un modelo gráfico de la situación planteada, en la escala apropiada, es muy útil para visualizar la situación, lo cual sería de gran apoyo a quien modela para determinar al menos el tipo o clase de función a cual pertenece la función desconocida  $f$ , como lo vimos en la sección anterior cuando tratábamos de establecer si había o no proporcionalidad entre cantidades de

las cuales conocíamos algunos datos.

Las gráficas también son útiles para visualizar *el ajuste*, a los datos, de la relación que por algún método encontremos, como veremos mas adelante en las siguientes secciones.

**Ejemplo:** Dada la siguiente tabla de datos,

$x$	1.5	3.7	4.2	6.1	7.8
$y$	0.5	2.9	3.4	7.5	12.0

encuentre qué tipo de función relaciona las dos cantidades  $x, y$ .

**Modelo gráfico:** Ver Fig. 3.7 y el programa en Python que aparece a continuación, el cual logra dicha figura.

```

1 # Importa librerías necesarias
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Ajusta estilo y tamaño figura
6 plt.style.use('seaborn-poster')
7 plt.figure(figsize = (12, 8))
8
9 #Genera puntos a graficar
10 x = np.array([1.5,3.7,4.2,6.1,7.8])
11 y = np.array([0.5,2.9,3.4,7.5,12.0])
12
13 #Crea grafica
14 plt.axvline(x=0, color='black', linewidth=1)           # Traza el eje vertical
15 plt.plot(a,b,'-', color='black', linewidth=1)          # Traza el eje horizontal
16 plt.plot(x, y, 'bo', color='black')                   # Grafica los puntos
17 plt.xlabel('x')                                         # Coloca nombre al eje horizontal
18 plt.ylabel('y')                                         # Coloca nombre al eje vertical
19 plt.xlim([0, 8])                                       # Define límites del eje horizontal
20 plt.ylim([0, 13])                                      # Define límites del eje vertical
21 plt.show()                                              # Muestra el grafico en una ventana

```

**Análisis:** Observando la gráfica en la Fig. 3.7, las cantidades no parecen tener una proporcionalidad directa (Por qué?) y tampoco una proporcionalidad inversa (Por qué). Por la curva inicial que presenta la gráfica y la propiedad creciente de la misma, pareciera que la función desconocida que relaciona las cantidades  $x, y$  es una función potencial (también podría ser una función exponencial). Ensayemos con las potencias 2 y  $\frac{1}{2}$  (Tabla 3.3 y Fig. 3.8).

Observe que hay una aparente linealidad de los puntos en la gráfica de la derecha en la Fig. 3.8, lo cual no ocurre en la gráfica de la izquierda.

**Conclusión:** Por lo anterior, podemos sugerir que la función que relaciona las cantidades  $x, y$  es de tipo cuadrático, o mejor aún, que  $y$  es (directamente) proporcional a  $x^2$ ; es decir, que el modelo matemático para estas dos cantidades es de la forma  $y = Ax^2$ , siendo  $A$  la constante de proporcionalidad (ver Ejercicio 10 para el cálculo de  $A$ ).

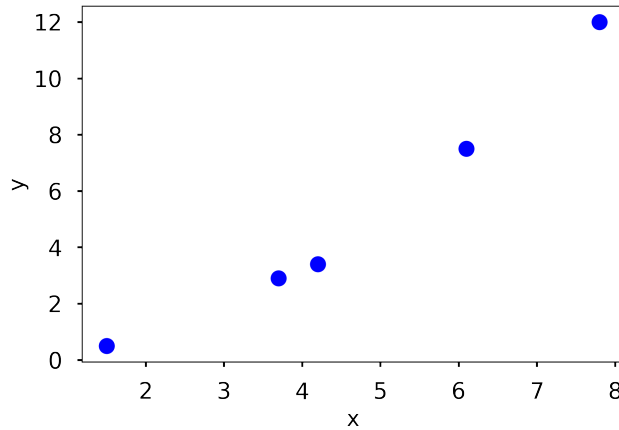


Figura 3.7: Modelo gráfico de los datos dados.

### 3.2.2. Mínimos cuadrados

Como lo vimos previamente re-escalando los datos podemos obtener una relación lineal que podemos visualizar a través de una recta. La cuestión gira entonces en torno a obtener la ecuación de dicha recta de manera que se obtenga la mejor aproximación lineal tal que se “minimice el ‘error’ entre ella y los datos observados. Si además, por el conocimiento que tengamos de la situación o por un análisis de un modelo gráfico de los datos, suponemos o conocemos el tipo de función que relaciona las dos cantidades, y suponemos que el error ocurre solo en la variable dependiente,  $y$ , y medimos “el error” como la suma de los cuadrados del error en cada dato, podemos plantear el siguiente modelo matemático, conocido como mínimos cuadrados.

**Modelo matemático (mínimos cuadrados):** Encontrar una función  $f \in \mathfrak{F}$  tal que  $E = \sum_{i=1}^n (y_i - f(x_i))^2$  sea mínimo, siendo  $\mathfrak{F}$  el conjunto de funciones determinado. Matemáticamente, el problema a resolver es

$$\min_{f \in \mathfrak{F}} \sum_{i=1}^n (y_i - f(x_i))^2$$

el anterior problema se simplifica cuando el conjunto de funciones  $\mathfrak{F}$  se define en término de un número finito de parámetros. En este contexto simplificado, el problema (o modelo) de mínimos cuadrados se clasifica en lineal (polinomio de grado 1) y no lineal (polinomio de grado mayor) según la forma cómo los parámetros se interrelacionan en la definición del conjunto de funciones  $\mathfrak{F}$ .

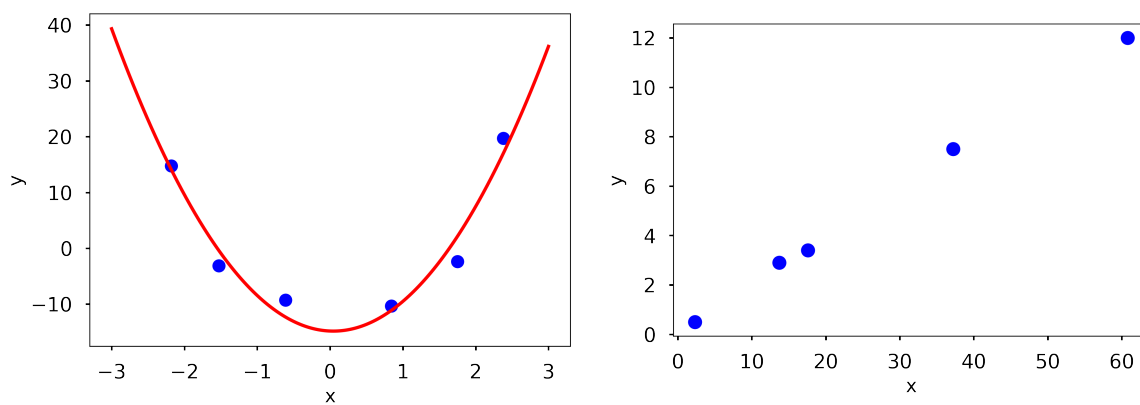
El problema a desarrollar será *lineal* cuando la función  $f$  se pueda escribir como

$$f(x) = a_0 g_0(x) + a_1 g_1(x) + a_2 g_2(x) + \dots + a_k g_k(x), \quad a_0, a_1, \dots, a_k \in R,$$

para algunas funciones  $g_0, g_1, \dots, g_k : R \rightarrow R$ ; en caso contrario, el problema será *no lineal*. El caso lineal es muy frecuente y más fácil de resolver que el caso no lineal. En efecto, la solución del caso lineal se reduce a la solución de un sistema de ecuaciones lineales que se obtienen a partir de las derivadas parciales de  $E$ .

$x$	$y$	$u$	$v$
0.8	0.8	0.8	1.2
1.3	1.4	1.3	1.6
2.8	7.7	2.8	2.7
3.1	9.8	3.1	2.5
5.2	26.6	5.2	5.2
7.1	52.4	7.1	11.3
9.3	83.2	9.3	20.1
10.2	102.7	10.2	28.1
11.0	122.4	11.0	37.8

Cuadro 3.1

Figura 3.8: Modelo gráfico de los datos de  $x$  al cuadrado (Der) y raíz cuadrada (Izq).

$x$	$y$	$u$	$v$
0.8	0.8	0.8	0.8
1.3	1.4	1.3	1.0
2.8	7.7	2.8	1.7
3.1	9.8	3.1	1.9
5.2	26.6	5.2	2.1
7.1	52.4	7.1	2.9
9.3	83.2	9.3	3.2
10.2	102.7	10.2	3.4
11.0	122.4	11.0	3.5

Cuadro 3.2

$x^2$	$y$	$\sqrt{x}$	$y$
2.3	0.5	1.2	0.5
13.7	2.9	1.9	2.9
17.6	3.4	2.0	3.4
37.2	7.5	2.5	7.5
60.8	12.0	2.8	12.0

Cuadro 3.3: Tabla de  $x^2$  vs.  $y$  (Izq) y de  $\sqrt{x}$  vs.  $y$  (Der).

Buscamos así encontrar la mejor aproximación lineal tal que  $E = E(m, b)$  sea mínimo. Y por tanto:

$$\frac{\partial E}{\partial b} = 0 \quad \frac{\partial E}{\partial m} = 0$$

siendo derivadas parciales. De modo que

$$\begin{aligned} \frac{\partial E}{\partial b} = 0 &\rightarrow \sum_{i=1}^n (y_i) = m \sum_{i=1}^n (x_i) + \sum_{i=1}^n b \\ \frac{\partial E}{\partial m} = 0 &\rightarrow \sum_{i=1}^n (y_i x_i) = m \sum_{i=1}^n (x_i^2) + b \sum_{i=1}^n (x_i) \end{aligned}$$

Resta resolver el sistema de ecuaciones 2x2 obteniendo así la expresión m para la pendiente y b para el intercepto:

$$\begin{aligned} m &= \frac{n \sum_{i=1}^n (x_i y_i) - \sum_{i=1}^n (x_i) \sum_{i=1}^n (y_i)}{n \sum_{i=1}^n (x_i^2) - (\sum_{i=1}^n x_i)^2} \\ b &= \frac{\sum_{i=1}^n (y_i) \sum_{i=1}^n (x_i^2) - \sum_{i=1}^n (x_i) \sum_{i=1}^n (x_i y_i)}{n \sum_{i=1}^n (x_i^2) - (\sum_{i=1}^n x_i)^2} \end{aligned}$$

El procedimiento se puede generalizar para cualquier polinomio

$$p = a_0 x^0 + a_1 x^1 + a_2 x^2 + \dots + a_n x^n$$



Y aplicando el mismo procedimiento de máximos y mínimos  $(n+1)$ -veces el problema pasa ser un sistema lineal  $(n+1) * (n+1)$ .

En la siguiente tabla se muestran algunos datos de una empresa italiana de ropa. Cada fila muestra las ventas anuales de la compañía y la cantidad invertida en el mismo año: Determine una relación entre el dinero invertido anualmente en publicidad (variable predictora) y las ventas realizadas el mismo año (variable de respuesta). ¿Cuál será el estimado de las ventas en un año si se invirtieran 10 millones de euros en publicidad?

Sean:

Variable independiente: Publicidad: [23, 26, 30, 34, 43, 48, 52, 57, 58]

Variable dependiente: Ventas: [651, 762, 856, 1063, 1190, 1298, 1421, 1440, 1518]

Generamos una gráfica de los datos y con las expresiones dadas para la pendiente y el intercepto por el método de los mínimos cuadrados podemos encontrar la línea rectal (ver fig.3.9).

$$S_y = \sum_{i=1}^n (y_i) \quad S_x = \sum_{i=1}^n (x_i) \quad S_{xy} = \sum_{i=1}^n (y_i x_i) \quad S_{xx} = \sum_{i=1}^n (x_i^2)$$

$$m = \frac{nS_{xy} - (S_x S_y)}{nS_{xx} - S_x^2} = 23,4228$$

$$b = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - S_x^2} = 167,6829$$

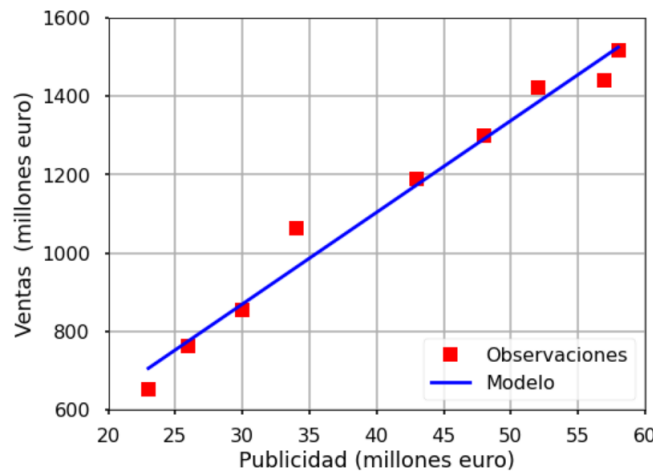


Figura 3.9: Gráfica con los datos experimentales y la respectiva recta que mejor se ajusta a los datos.

A continuación se muestra un programa en Python que logra la figura 3.9.

```
1 #importamos el modulos de numpy (numeric python)
2 import numpy as np
```

```

3
4 #importamos del modulo matplotlib la libreria pyplot con el alias de plt
5 import matplotlib.pyplot as plt
6
7 #datos que queremos ajustar
8 x = [1 ,2 ,3 ,4 ,5 ,6 ,6.6]
9 y = [0.5 ,2.5 ,2 ,4 ,3.5 ,6 ,5.5]
10
11 #-----
12 #Calculo de la pendiente (m) y el intercepto (b) de la linealizacion
13 n = len(x)
14
15 #donde almacenaremos los valores de las sumas
16 suma_xi = 0
17 suma_yi = 0
18 suma_xixi = 0
19 suma_xiyi = 0
20
21 for i in range( n ):
22     suma_xi += x[i]
23     suma_yi += y[i]
24     suma_xixi += x[i]**2
25     suma_xiyi += x[i]*y[i]
26
27 #aplicado las formulas
28
29 m = (n * suma_xiyi - suma_xi * suma_yi) / (n * suma_xixi - suma_xi**2)
30
31 b = (suma_xixi * suma_yi - suma_xiyi * suma_xi) / (n * suma_xixi - suma_xi**2)
32
33
34 #Linea recta aproximada
35 L = m * np.array(x) + b #note que aqui toca transformar la lista de x a un
    arreglo!
36
37 #-----
38
39
40 #-----graficando-----
41 #Graficando las parejas de datos
42 plt.plot( x , y , "or" , label="datos" )
43
44
45 #Graficando la recta segun m y b
46 plt.plot( x , L , label="recta aproximada" )
47
48
49 #mostrando las legendas del grafico
50 plt.grid()
51 plt.legend()
52
53 #mostrando el grafico
54 plt.show()

```

Dada la siguiente tabla de datos,

$x$	-2.18	-1.527	-0.61	0.84	1.75	2.38
$y$	14.765	-3.142	-9.252	-10.342	-2.371	19.705

encuentre la mejor recta y la mejor parábola que aproxime los datos dados.

El siguiente programa usa la función de Python *linregress* para calcular la mejor recta que aproxima los datos en el sentido de mínimos cuadrados (lineales) y gráfica tanto los datos dados como la mencionada recta: Fig. 3.10 (Izq) (observe que la recta no pasa por (o contiene) todos los puntos dados).

```

1 # Importa librerías necesarias
2 from scipy import stats
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Define los datos
7 x = np.array([-2.18, -1.527, -0.61, 0.84, 1.75, 2.38])
8 y = np.array([14.765, -3.142, -9.252, -10.342, -2.371, 19.705])
9
10 # Calcula los parametros de la recta de regresion
11 res = stats.linregress(x,y)
12 a = res.slope # Pendiente de la recta
13 b = res.intercept # Intercepto de la recta con el eje vertical
14
15 # Ajusta tamaño de la figura
16 plt.style.use('seaborn-poster')
17 plt.figure(figsize = (12, 8))
18
19 # Grafica datos y recta de minimos cuadrados
20 x_line = np.arange(min(x), max(x), 1)
21 plt.plot(x, y, 'bo', color='black', label='Datos')
22 plt.text(0.2, 4.5, 'y = %.5f * x + %.5f' % (a, b), fontsize=15)
23 plt.plot( x_line, a*x_line + b, color = 'red' )
24 plt.legend()
25 plt.show()

```

De la misma forma, el siguiente programa usa la función de Python *polyfit* para calcular la mejor parábola que aproxima los datos en el sentido de mínimos cuadrados (lineales) y grafica tanto los datos dados como la mencionada parábola: Fig. 3.10 (Der) (observe que la parábola **no** pasa por (o **no** contiene) todos los puntos dados).

```

1 # Importa librerías necesarias
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Define los datos
6 xdata = np.array([-2.18, -1.527, -0.61, 0.84, 1.75, 2.38])
7 ydata = np.array([14.765, -3.142, -9.252, -10.342, -2.371, 19.705])
8
9 # Calcula la regresion parabolica
10 pol = np.polyfit(xdata,ydata, 2)
11 x_new = np.linspace(-3.0, 3.0, 200)

```

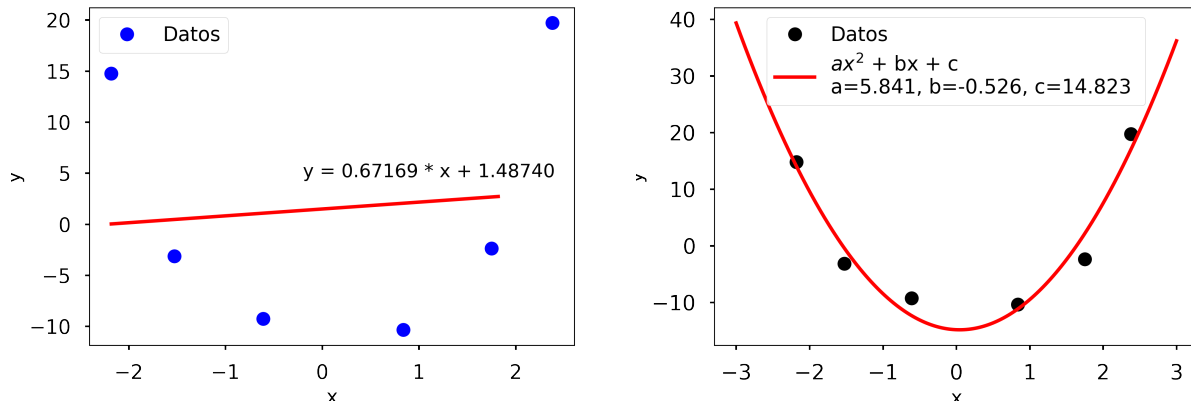


Figura 3.10: Gráfica de los datos dados (Puntos), y de la recta (Izq) y la parábola (Der) que mejor aproxima los datos, respectivamente (Linea continua).

```

12 y_new = np.polyval(pol, x_new)
13
14 # Ajusta tamaño de la figura
15 plt.style.use('seaborn-poster')
16 plt.figure(figsize = (12, 8))
17
18 # Grafica datos y curva de regresion
19 plt.plot(xdata, ydata, 'bo', color='black', label='Datos')
20 plt.plot(x_new, y_new, 'r-', label='y = ax^2 + bx + c: \
21 a=%5.3f, b=%5.3f, c=%5.3f' % (pol[0], pol[1], pol[2]) )
22 plt.xlabel('x')
23 plt.ylabel('y')
24 plt.legend()
25 plt.show()

```

A la acción de inferir información dentro del intervalo donde se encuentran los datos le llamamos interpolar.

Bajo el supuesto que los datos observados de las cantidades son “exactos”, es plausible pensar que la función incluya los puntos observados (dicho de otra manera, que el modelo gráfico de la función “pase” por los puntos dados). Si además, por el conocimiento que tengamos de la situación o por un análisis de un modelo gráfico de los datos, suponemos o conocemos el tipo de función que relaciona las dos cantidades, podemos obtener nuevos puntos partiendo de la relación ya conocida. Se plantea así el modelo matemático:

**Modelo matemático (interpolación):** Encontrar una función  $f \in \mathfrak{F}$  tal que  $y_i = f(x_i)$ , para  $i = 1, 2, \dots, n$ , siendo  $\mathfrak{F}$  el conjunto de funciones determinado.

Gráficamente podemos observar como a partir del conocimiento de un conjunto de puntos es posible la obtención de puntos nuevos. Primero utilizamos el método de mínimos cuadrados, anteriormente estudiado, para encontrar la mejor aproximación lineal Fig. 3.11.

Año	Ventas	Publicidad
	Millones-euros	Millones-euros
1	651	23
2	762	26
3	856	30
4	1063	34
5	1190	43
6	1298	48
7	1421	52
8	1440	57
9	1518	58

Cuadro 3.4

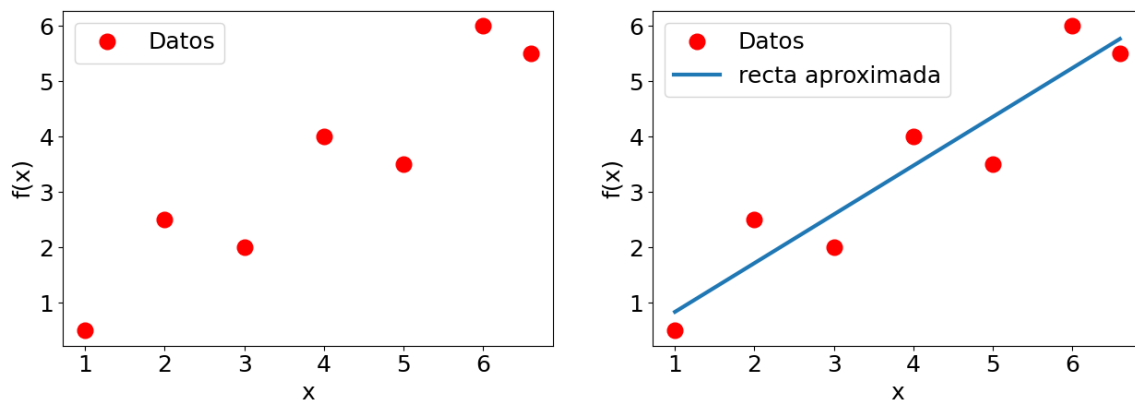


Figura 3.11: Gráfica con los datos experimentales (Izq) y gráfica con la recta que mejor se aproxima a los datos con el método de mínimos cuadrados (Der).

Posteriormente buscamos aproximar el valor de la variable dependiente para un valor de la variable independiente cercano a él. Cuando este valor de la variable independiente está entre los valores de los datos, decimos que estamos **interpolando** Fig.3.12 ( Izq) en caso contrario decimos que estamos **extrapolando** Fig. 3.12 ( Der). Es de anotar que según la teoría del error del proceso de interpolación, el error estimado cuando se interpola es menor que cuando se extrapola.

Para el caso de la interpolación no lineal (Polinomios de grado mayor) de una variable independiente, podemos observar como no resulta tan sencillo encontrar una recta aproximada lineal que pueda relacionar los puntos Fig 3.13.

Así que hacemos uso de la interpolación polinómica de Lagrange.

**Interpolación Polinómica de Lagrange:** Forma de presentar el polinomio que interpola un conjunto dado de  $k+1$  puntos  $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$  Donde todos  $x_j$  se asumen distintos, el polinomio interpolador es la combinación lineal:

$$P(x) = \sum_{j=0}^k y_j L_j(x)$$

Damos por lo tanto primero la definición de combinación lineal.

**Combinación lineal:** es una expresión construida por un grupo de términos del mismo tipo, donde cada uno está multiplicado por una constante, y el resultado se obtiene sumando cada término. Se tienen los siguientes ejemplos de combinaciones lineales. La pregunta ahora es ¿Cómo construimos los polinomios?

Esto se realiza a través de la base polinómica de Lagrange:

$$L_j(x) = \prod_{i=0, i \neq j}^k \frac{x - x_i}{x_j - x_i} = \left( \frac{x - x_0}{x_j - x_0} \right) \dots \left( \frac{x - x_{j-1}}{x_j - x_{j-1}} \right) \left( \frac{x - x_{j+1}}{x_j - x_{j+1}} \right) \dots \left( \frac{x - x_k}{x_j - x_k} \right)$$

Dados los datos de la variable independiente:

x'datos=  $[-1, 0, 1]$   
y'datos=  $[5, 4, 4, 2, 3]$

Encuentre el polinomio interpolador.

Primero realizamos la construcción de los polinomios que forman la base de Lagrange.

$$\begin{aligned} L_0(x) &= \left( \frac{x - x_1}{x_0 - x_1} \right) \left( \frac{x - x_2}{x_0 - x_2} \right) = \frac{1}{2}x(x - 1) \\ L_1(x) &= \left( \frac{x - x_0}{x_1 - x_0} \right) \left( \frac{x - x_2}{x_1 - x_2} \right) = -(x + 1)(x - 1) \\ L_2(x) &= \left( \frac{x - x_0}{x_2 - x_0} \right) \left( \frac{x - x_1}{x_2 - x_1} \right) = \frac{1}{2}x(x + 1) \end{aligned}$$

Gráficamente se pueden observar 3.15.

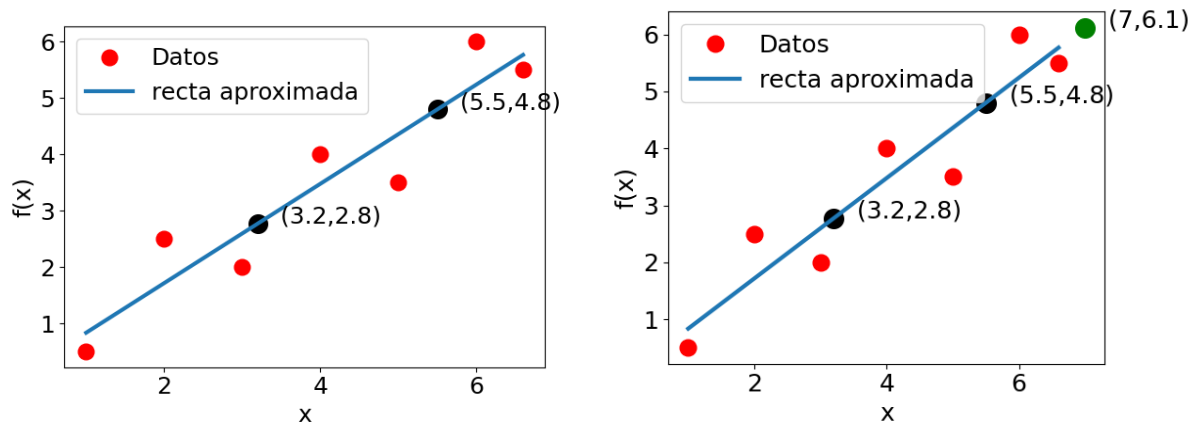


Figura 3.12: Interpolación (Izq) - Extrapolación (Der).

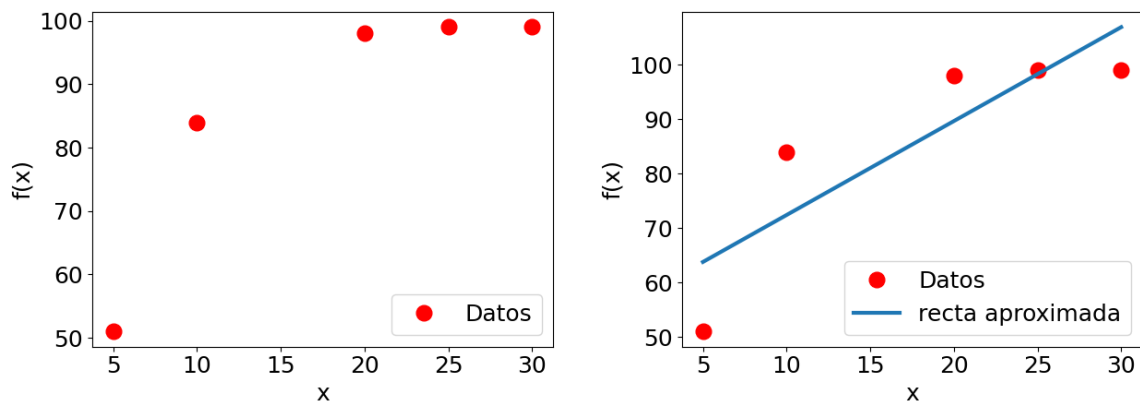


Figura 3.13: Interpolación no lineal (polinomios de grado mayor).

Al reemplazar los polinomios por las variables dependientes se comprueba:

$$L_0(x_0) = L_0(-1) = 1$$

$$L_1(x_1) = L_1(0) = 1$$

$$L_2(x_2) = L_2(1) = 1$$

Ahora construimos el polinomio interpolador

$$P(x) = \sum_{j=0}^k y_j L_j(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x)$$

con  $y_0 = 5.4, y_1 = 4.2, y_2 = 3$

Es importante corroborar que el polinomio sí interpola.

$$P_{x0} = P(-1) = 5.4 = y_0$$

$$P_{x1} = P(0) = 4.2 = y_1$$

$$P_{x2} = P(1) = -3 = y_2$$

Finalmente generamos una gráfica del polinomio interpolador correspondiente a la figura 3.16.

A continuación programa en Python que logra figura 3.16.

```

1 #coding: -utf-8-
2 #importamos el modulo de numpy (numeric python)
3 import numpy as np
4
5 #importamos del modulo matplotlib la libreria pyplot con el alias de plt
6 import matplotlib.pyplot as plt
7
8
9 #-----
10 #Definiendo los polinomios de lagrange Li(x)"""
11 def L( x , x_datos , i):
12
13     #inicializamos a 1
14     Li = 1
15
16     #hago la productoria
17     for k in range( len( x_datos ) ):
18
19         #adiciono la restriccion de la productoria
20         if k != i:
21             Li = Li * (x- x_datos[k]) / ( x_datos[i]- x_datos[k] )
22
23     return Li
24 #-----
25
26
27 #graficando algunos polinomio de lagrange
28
29 #numero de polinomios a construir

```



```

30 #datos
31
32 x_datos = [ -1 , 0 , 1 ]
33
34 #x_datos = [5 ,10 ,20 ,25, 30]
35
36 #dominio en donde los graficaremos
37 x = np.linspace( -1, 1 , 50)
38 #x = np.linspace( 0, 30 , 50)
39
40
41 plt.rcParams['font.size'] = 18
42 #Graficando los polinomios juntos
43 for i in range( len( x_datos ) ):
44     plt.plot( x , L(x,x_datos,i), label = "L"+str(i)+"(x)" )
45
46     #poniendo nombres a de los ejes
47     plt.xlabel('x')
48     plt.ylabel('Li(x)')
49
50 #mostrando las legendas del grafico
51 plt.legend()
52 plt.grid()
53
54 #Mostrando el grafico
55 plt.tight_layout()
56 plt.savefig('Interpolacion_Polinomios_Lagrange_01.png')
57 plt.show()
58
59 y_datos=[5.4, 4.2, -3]
60
61
62 def P(x,y_datos):
63     Pi=0
64     for k in range(len(y_datos)):
65         Pi+=y_datos[k]*L(x,x_datos,k)
66
67     return Pi
68
69 plt.plot(x_datos,y_datos,'or', markersize=10, label='Datos')
70 plt.plot(x,P(x,y_datos),'-b',label='Polinomio aproximado')
71 plt.xlabel('$x$',size=16)
72 plt.ylabel('$y$',size=16)
73 plt.ylim(ymax=6, ymin=-4)
74 plt.grid()
75 plt.tight_layout()
76 plt.legend(loc='lower left')
77 plt.savefig('Interpolacion_Polinomios_Lagrange_Datos.png')
78 plt.show()

```

Con la interpolación de Lagrange el trabajo a realizar era aproximar una función arbitraria en un intervalo cerrado usando un polinomio individual. Sin embargo, los polinomios de orden superior pueden oscilar erráticamente; es decir, una fluctuación menor sobre una pequeña parte del intervalo puede inducir fluctuaciones grandes sobre todo el rango [1].

### 3.3. Ejercicios

1. Demuestre que la relación de proporcionalidad directa es una relación de equivalencia; es decir, es reflexiva ( $x \propto x$ ), es simétrica (Si  $y \propto x$  entonces  $x \propto y$ ) y es transitiva (si  $z \propto y$  y  $y \propto x$  entonces  $z \propto x$ ).
2. ¿La relación de proporcionalidad inversa es reflexiva? ¿Simétrica? ¿Transitiva? ¿Por qué?
3. Enuncie una ley de la física que se exprese usando la proporcionalidad directa.
4. Enuncie una ley de la física que se exprese usando la proporcionalidad inversa.
5. ¿El volumen de una esfera es proporcional a su radio? ¿al cuadrado de su radio? ¿al cubo de su radio?
6. Se sabe que el área de un círculo es proporcional al cuadrado de su radio. ¿Podemos decir que el radio de un círculo es proporcional a su área?
7. Si  $y$  y  $x$  tienen una relación lineal ( $y = mx + b$ ) ¿podemos decir que son proporcionales?
8. ¿Qué tipo de proporcionalidad, si hay alguna, se puede suponer entre las variables (funciones de ellas) de los gráficos en la Fig. 3.17?
9. Dado que visualmente es fácil determinar puntos que están alineados ¿qué tipo de escala sugiere para identificar los siguientes tipo de relaciones?

a)  $y = \frac{A}{B+x}$

b)  $y = Kxe^{-\frac{x}{a}}$

10. Para calcular la constante de proporcionalidad del modelo propuesto en el Ejemplo ?? ¿usaría un proceso de interpolación o de mínimos cuadrados? Calcule la constante de proporcionalidad para dicho modelo.
11. Para calcular la base del modelo propuesto en el Ejemplo 3.1.3 ¿usaría un proceso de mínimos cuadrados lineales o no lineales? Calcule la base para dicho modelo.
12. Para calcular la base del modelo propuesto en el Ejemplo ?? ¿usaría un proceso de mínimos cuadrados lineales o no lineales? Calcule la base para dicho modelo.
13. Un grupo de estudiantes se dio a la tarea de investigar la relación matemática entre la longitud de una circunferencia y la longitud de su diámetro. Para ello, usando diferentes tipos de cordel (piola, nylon, cabuya, etc), obtuvieron los siguientes datos:

Como líder de este grupo de estudiantes ¿cómo manejaría las siguientes preguntas?

- a) ¿Qué hacer inicialmente con los datos obtenidos?
- b) ¿Qué tipo de relación sugieren estos datos?
- c) Con base en los datos ¿podemos precisar el modelo que relaciona estas dos longitudes?

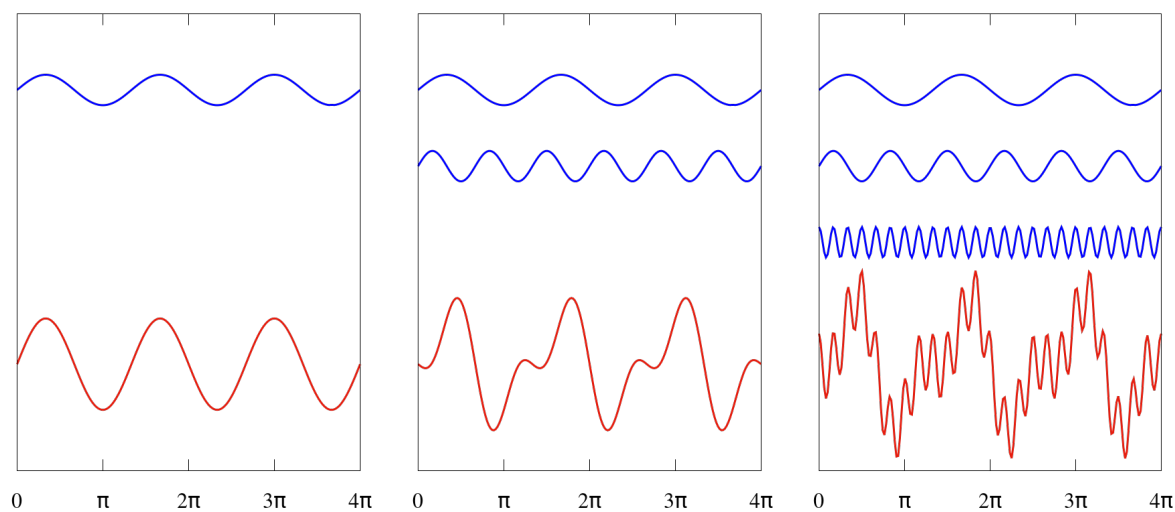


Figura 3.14: Combinaciones lineales

Donde  $f_1(x) = \sin(1,5x)$ ,  $f_2(x) = \sin(3x)$ ,  $f_3(x) = \cos(2x)$  (color azul)  
 y  $h(x) = a_1f_1(x) + a_2f_2(x) + a_3f_3(x)$  (color rojo).

h]

E	Clara	Inés	David	Esteban	Alba	Iván	Nathalie	Carol	Lorena
P (cm)	5	8	17	12	21	10	15	23	18
C(cm)	31.4	50.2	106.8	75.5	131.8	63.0	94.3	144.5	112.9

Cuadro 3.5: Datos correspondientes a Estudiante (E), Perímetro (P) y Longitud de circunferencia (C).

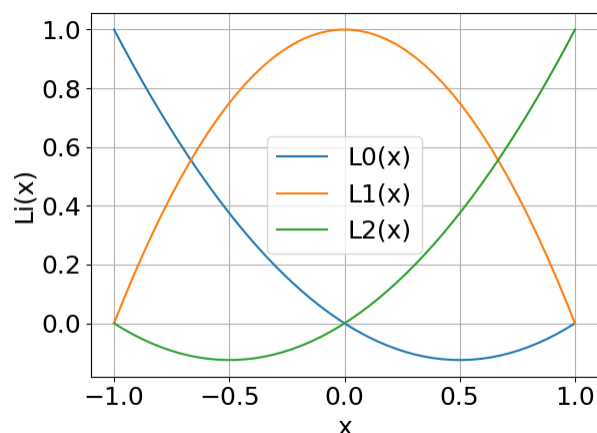


Figura 3.15: Elementos de la base polinómica de Lagrange.

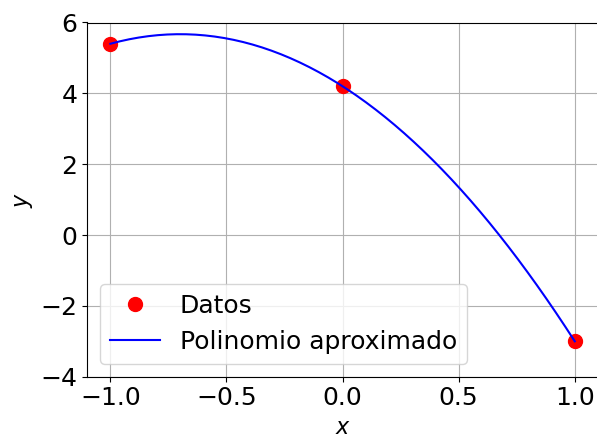


Figura 3.16: Interpolación con Polinomios Lagrange.

- d) Use el modelo obtenido para interpolar y extrapolar valores interesantes de una circunferencia.
  - e) Si otro grupo de estudiantes hiciera el mismo ejercicio de medición ¿obtendría resultados similares? ¿Por qué?
  - f) Revisando la literatura sobre el tema ¿qué cantidad podría estimar con estas mediciones?
14. El mismo grupo de estudiantes se preguntó por la relación matemática entre la longitud de un péndulo como el de la Fig. 2 y el tiempo de una oscilación completa (periodo). Para ello, usando diferentes tipos de cordel (piola, nylon, cabuya, etc), obtuvieron los siguientes datos:

Como líder de este grupo de estudiantes, cómo manejaría las siguientes preguntas:

- a) ¿Qué hacer inicialmente con los datos obtenidos?

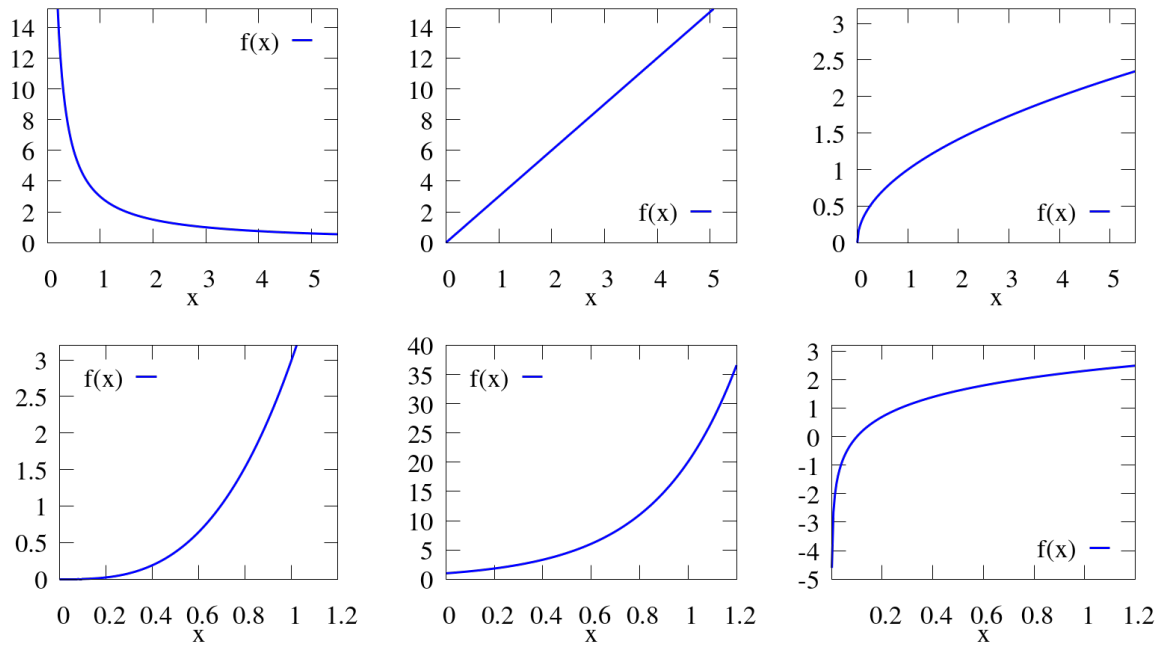


Figura 3.17: Gráficos aproximados de las variables.

E	Clara	Inés	David	Esteban	Alba	Iván	Nathalie	Carol	Lorena
L (cm)	21	33	25	5	7	15	10	12	19
P (seg)	0.92	1.15	1.00	0.40	0.53	0.78	0.64	0.70	0.88

Cuadro 3.6: Datos correspondientes a Estudiante (E), Longitud (L) y Periodo (P).

- b) ¿Qué tipo de relación sugieren estos datos?
- c) Con base en los datos ¿podemos precisar el modelo que relaciona estas dos variables?
- d) Use el modelo obtenido para interpolar y extrapolar valores interesantes de una longitud.
- e) Si otro grupo de estudiantes hiciera el mismo ejercicio de medición ¿obtendría resultados similares? ¿Por qué?
- f) Revisando la literatura sobre el tema ¿qué cantidad podría estimar con estas mediciones?
15. En el estudio de poblaciones biológicas, en lo que se refiere al crecimiento o tamaño de las mismas a través del tiempo, existen dos modelos clásicos que ayudan en su análisis, según su crecimiento esté o no limitado por factores externos a la población misma. Estos modelos se conocen como *crecimiento exponencial* y *crecimiento logístico*, respectivamente, y están dados por la gráficas y expresiones que aparecen en la Fig. 3.18.
- Para tres poblaciones biológicas, se tienen los datos dados en el Cuadro 3.7 sobre el número de individuos en cada una de ellas, en los tiempos indicados.

POBLACIÓN 1		POBLACIÓN 2		POBLACIÓN 3	
t (año)	Individuos (MM)	t (hora)	Individuos	t (días)	Individuos
1970	1.875	7:00	10	0	20
1978	2.043	7:30	15	3	25
1981	2.155	9:00	39	5	28
1985	2.279	10:00	82	10	42
1993	2.340	10:45	136	14	55
1998	2.340	11:30	220	20	83
2000	2.366	12:15	385	22	99
2005	2.422	13:00	625	28	155
2010	2.502	14:00	1.261	30	175
2012	2.514	14:30	1.803	40	362
2018	2.585	16:00	5.105	46	569
2020	2.591	17:15	12.275	50	721
		18:00	20.746	60	1.508
				64	2.075
				70	3.098

Cuadro 3.7: Tablas del número de individuos en tres poblaciones en los tiempos (t) indicados.

- a) Establezca el tipo de crecimiento de cada una de las poblaciones para proponer un modelo de crecimiento de cada población.
- b) Para estimar los parámetros del modelo propuesto ¿utilizaría interpolación o mínimos cuadrados?

- c) Si decide utilizar interpolación ¿qué tipo utilizaría (polinomial, exponencial, etc)?
  - d) Si usa mínimos cuadrados ¿qué tipo utilizaría (lineal, no lineal, etc.)?
  - e) Para cada población, calcule los parámetros del modelo propuesto y grafique el modelo obtenido junto con los datos dados.
  - f) Use el modelo obtenido para interpolar y extrapolar valores interesantes de la población.
16. Considere la reacción química  $X \rightarrow \text{productos}$ . Se sabe que esta reacción puede ser de orden 0, 1 o 2, según satisfaga la relación  $[x]' = c$ ,  $[X]' = a[X] + b$  ó  $[X]' = a[X]^2 + b$ , respectivamente, siendo  $[X](t)$  la concentración de la sustancia química  $X$  en la reacción al tiempo  $t$ , y  $b$  un flujo constante de la sustancia química  $X$ . En tres reacciones químicas, se toman medidas de la concentración de la sustancia química  $X$  en la reacción, a intervalos de uno, dos y tres minutos, respectivamente, y se obtienen los datos dados en el Cuadro 3.8.
- a) ¿Qué orden de reacción hay en cada una de las reacciones? [**AYUDA:** En cada tabla, para cada par de valores, use un modelo para estimar un valor de la función concentración en el intervalo dado (uno, dos o tres minutos) y use otro modelo para estimar el cambio respecto del tiempo de esta concentración (derivada) en el mismo intervalo, luego determine el orden de reacción según las definiciones dadas].
  - b) ¿A qué valores del tiempo se estima fueron tomadas las medidas? [**AYUDA:** Con el orden de la reacción, se obtiene la ecuación diferencial que la función concentración satisface y con ella se puede calcular el tiempo al cual la función concentración toma los valores dados].

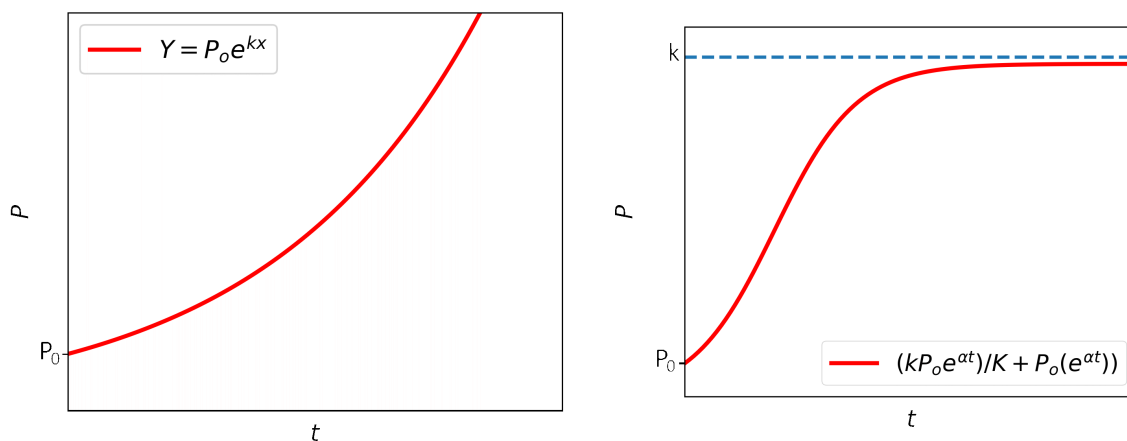


Figura 3.18: Crecimiento exponencial ( $P = P_0 e^{\alpha t}$ ) (Izq) y Crecimiento logístico ( $P = \frac{K P_0 e^{\alpha t}}{K + P_0(e^{\alpha t} - 1)}$ ), siendo  $P_0$  la población inicial,  $\alpha$  la tasa de crecimiento y  $K$  la capacidad de población.

## REACCIÓN 1

Cantidad al tiempo $t$	Cantidad al tiempo $t + 1$
92.54	85.24
72.37	66.72
51.65	49.86
44.67	41.29
36.20	33.78
30.29	28.08
24.19	22.48
17.65	16.37
15.47	13.36
10.79	10.18
8.25	7.85
6.48	6.22
5.56	5.37
4.53	4.48
4.34	4.26
3.87	3.82

## REACCIÓN 2

Cantidad al tiempo $t$	Cantidad al tiempo $t + 2$
65.64	64.61
63.66	62.11
61.07	59.88
58.16	56.83
55.32	53.85
51.80	50.47
49.21	48.02
47.36	46.04
45.21	43.88
43.53	42.02
39.72	38.54
37.37	36.14
31.44	30.30
29.05	27.61
25.54	24.49
22.29	21.54
19.56	18.37

## REACCIÓN 3

Cantidad al tiempo $t$	Cantidad al tiempo $t + 3$
270.98	116.36
97.95	66.54
57.61	44.09
36.04	36.04
29.37	25.82
23.56	20.57
19.57	17.97
16.21	15.12
13.42	13.38
12.61	11.97
10.72	10.27
10.09	9.61
9.15	8.84
8.38	8.13
7.76	7.65
7.30	7.13

Cuadro 3.8: Tablas de la concentración de un reactivo medido a intervalos de 1, 2 y 3 minutos en tres reacciones, respectivamente.



## Bibliografía

- [1] Burden, R.L., Faires, J.D. and Burden, A.M., (2017) *Análisis numérico*, Décima edición, CENGAGE Learning Editores. (Citado en pagina [69](#).)
- [2] Dennis, J.E., Martínez, H.J. and Tapia, R.A. (1989) *A convergence theory for the structured BFGS secant method with an application to nonlinear least squares*. Journal of Optimization Theory and Applications, JOTA, Vol. 61, No. 2, pp. 161-178. ( No citado.)
- [3] Dennis Jr, J.E. and Schnabel, R.B., (1996) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics in Applied Mathematics, SIAM. ( No citado.)
- [4] Edwards Jr., C.H. y Penney, D.E. (1996) *Cálculo con geometría analítica*. 4a Edición, Editorial Pearson. (Citado en paginas [50](#) and [52](#).)
- [5] Engels, J.R. and Martínez, H.J. (1991) *Local and superlinear convergence for partially known quasi-Newton methods*, SIAM Journal on Optimization-SIOPT, Vol.1 No. 1, pp. 42-56. ( No citado.)
- [6] Golub, G.H. and Van Loan, C.F., (2013) *Matrix computations*, Fourth edition, The Johns Hopkins University Press ( No citado.)
- [7] Hurtado, S. (2009) *Laboratorio virtual*, <https://labovirtual.blogspot.com/2009/07/el-pendulo-simple.html> ( No citado.)
- [8] Kincaid, D. and Cheney W. (2002). *Numerical Analysis: Mathematics of Scientific Computing*, AMS book Program. ( No citado.)

## Modelos deterministas

En este capítulo trataremos la construcción de modelos discretos y continuos de tipo dinámico para describir algunos sistemas sencillos. Un **modelo de tipo dinámico** es aquél en el que sus parámetros no son fijos sino que dependen de una variable independiente que suele ser el tiempo. Algunos ejemplos de este tipo de modelos que se analizan en el presente capítulo son: el comportamiento de una población de bacterias en el tiempo, o el número de codornices disponibles para la venta después de un número de días, o la posición de una masa sujeta a un resorte que oscila en torno de un punto de equilibrio, entre otros.

Modelos que puedan describir situaciones como las que se mencionan requieren del uso de algunas herramientas matemáticas, tales como las ecuaciones en diferencias y las ecuaciones diferenciales tanto para su construcción como para su análisis. Por esta razón, haremos una introducción de estos temas donde sea pertinente.

### 4.1. Modelos deterministas en ecuaciones en diferencias

En ocasiones, en el fenómeno físico de interés, existe una variable que toma valores discretos. Así ocurre por ejemplo con el tiempo, ya que el proceso de medición de una propiedad se suele realizar a intervalos de tiempo regulares. Estos datos constituyen un conjunto finito para el caso de las mediciones reales, o infinito numerable, de valores de la variable independiente. En este caso, unas de las herramientas matemáticas fundamentales para el estudio de estas situaciones son las sucesiones y las ecuaciones en diferencias.

Para iniciar veamos el concepto de sucesión de números reales.

**Definición:** Una sucesión es un conjunto finito o infinito de elementos, en un conjunto de números reales  $A$ , el cual sigue un orden determinado. Formalmente, **una sucesión** es una función  $f : \mathbb{N} \rightarrow A$ , la cual asigna a cada  $n \in \mathbb{N}$  un único elemento  $f(n)$  del conjunto  $A$ .

En esta definición  $\mathbb{N}$  denota el conjunto de los números naturales  $\{1, 2, 3, \dots\}$ , el cual, dependiendo de la sucesión, puede incluir el cero:  $\{0, 1, 2, 3, \dots\}$ .

Se acostumbra usar la notación  $y_n = f(n)$  y denotar la sucesión como  $(y_n)_{n \in \mathbb{N}}$ , (o más simplemente  $(y_n)$ ) donde  $y_n$  (llamado también el término  $n$ -ésimo de la sucesión), hace referencia al elemento de la sucesión en la posición  $n$ .

**Definición:** Para una sucesión de números  $(y_n)$  denotamos las primeras diferencias como

$$\begin{aligned}\Delta y_0 &= y_1 - y_0 \\ \Delta y_1 &= y_2 - y_1 \\ \Delta y_2 &= y_3 - y_2.\end{aligned}$$

Análogamente, para cada  $n \in \mathbb{N}$ , la  $n$ -ésima primera diferencia es  $\Delta y_n = y_{n+1} - y_n$ . Además el cociente  $\Delta y_n / y_n$  es la tasa de cambio media de la sucesión en el término  $n$  de la sucesión. Una sucesión de números reales puede graficarse en un plano donde la variable  $n$  se coloca en el eje horizontal, y el valor  $y_n$  en el eje vertical. La misma sucesión también puede graficarse sobre una recta sus valores  $y_n$ . Considere los siguientes ejemplos:

**Ejemplo 1:** La sucesión  $(\frac{1}{n})$  se puede graficar tanto en un plano como la colección de puntos  $(n, y_n), n = 1, 2, 3, \dots$  mostrados en la figura 4.1(a), o sobre la recta mostrada en la Figura 4.1(b), donde se colocan solo los valores  $y_n$  para  $n = 1, 2, 3, \dots$ . Una propiedad de esta sucesión es que cuando  $n$  crece sus valores  $y_n = \frac{1}{n}$  van disminuyendo progresivamente acercándose a cero. Esto se expresa como

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0.$$

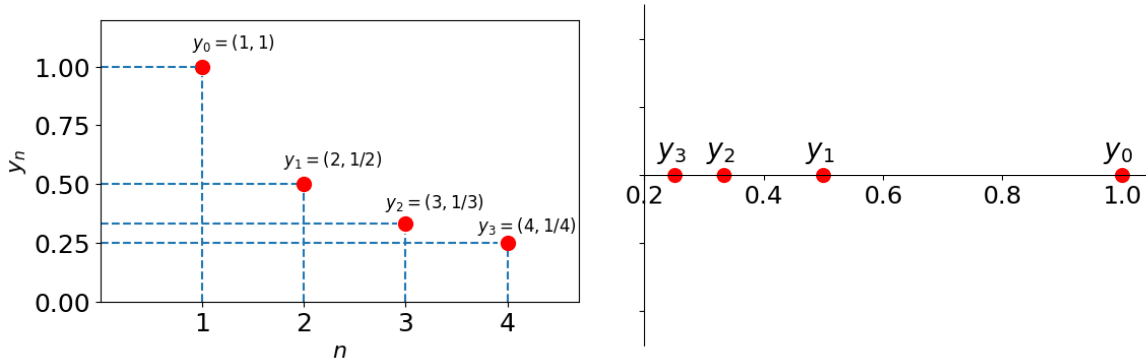


Figura 4.1: Dos representaciones gráficas de la sucesión  $(1/n)$ .

Las sucesiones pueden ser descritas también a través de relaciones de recurrencia (llamadas **ecuaciones en diferencias**) que permiten ir calculando uno a uno cada término de ésta.

**Definición:** Llamamos ecuación en diferencias a una expresión del tipo

$$F(y_{n+k}, y_{n+k-1}, y_{n+k-2}, \dots, y_{n+1}, y_n, n) = 0, \quad (4.1)$$

donde  $F$  es una función dada y  $k$  es un entero positivo que se llama el **orden de la ecuación**. Así, una relación de recurrencia de la forma

$$y_{n+1} = g(y_n), \quad (4.2)$$

es una ecuación en diferencias de primer orden.

**Definición:** Una solución de la ecuación en diferencias 4.1, es toda sucesión  $(y_n)$  tal que se satisfice

$$F(y_{n+k}, y_{n+k-1}, y_{n+k-2}, \dots, y_{n+1}, y_n, n) = 0,$$

para todo  $n$ . Por ejemplo, si  $\lambda$  es una constante, la sucesión  $y_n = \lambda^n$  es una solución de la ecuación en diferencias de orden 1 dada por

$$y_{n+1} = \lambda y_n.$$

En efecto, sustituyendo la expresión de  $y_n$  en la ecuación, obtenemos

$$y_{n+1} = \lambda^{n+1} = \lambda \lambda^n = \lambda y_n,$$

para todo  $n$ . De manera análoga, la sucesión  $y_n = (-3)^n + 1$  es solución de la ecuación en diferencias de orden 2

$$y_{n+2} + 2y_{n+1} - 3y_n = 0,$$

lo cual se puede verificar sustituyendo  $y_n$  directamente en la ecuación en diferencias dada, obteniendo

$$\begin{aligned} y_{n+2} - 2y_{n+1} - 3y_n &= (-3)^{n+2} + 1 + 2((-3)^{n+1} + 1) - 3((-3)^n + 1) \\ &= (-3)^{n+2} + 2(-3)^{n+1} - 3(-3)^n \\ &= (-3)^2(-3)^n + 2(-3)(-3)^n - 3(-3)^n = (9 - 6 - 3)(-3)^n = 0, \end{aligned}$$

para todo  $n$ .

Las ecuaciones en diferencias antes mencionadas son lineales y homogéneas de coeficientes constantes.

**Definición:** Una sucesión constante  $y_n = c$ , para todo  $n$  se llama un equilibrio de la ecuación en diferencias 4.2, si  $g(c) = c$ .

Por ejemplo, si  $k \neq 1$ , tenemos que  $y_n = 0$  es el único equilibrio de la ecuación

$$y_{n+1} = ky_n,$$

y  $y_n = -\frac{b}{k-1}$  es el único equilibrio de

$$y_{n+1} = ky_n + b.$$

Por otro lado, para  $b \neq 0$ , las sucesiones  $y_n = 0$  y  $y_n = \frac{a-1}{b}$  son equilibrios de la ecuación

$$y_{n+1} = ay_n - by_n^2.$$

Algunos ejemplos son los siguientes:

**Ejemplo 2:**

1. Solucionemos la siguiente ecuación en diferencias:

$$y_{n+1} = 3y_n.$$

Observe que se requiere conocer el valor inicial  $y_0$  a fin de poder calcular los términos de la sucesión usando la ecuación en diferencias dada, la cual especifica que para obtener cada término se debe multiplicar por 3 el anterior. Supongamos que  $y_0 = 1$ . Tenemos entonces que

$$\begin{aligned}y_1 &= 3y_0 = 3 \\y_2 &= 3y_1 = 3(3y_0) = 9y_0 = 9 \\y_3 &= 3y_2 = 3(3^2y_0) = 27y_0 = 27 \\y_4 &= 3y_3 = 3(3^3y_0) = 81y_0 = 81,\end{aligned}$$

y así sucesivamente, donde vemos que para el elemento  $n$ -ésimo se cumple que  $y_n = 3^n y_0$ .

Un programa en python que calcula y grafica los primeros términos en la Figura 4.2 de la sucesión ( $y_n$ ) es el siguiente:

```
1
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 y = np.zeros(6)
6 y[0] = 1.0 # Valor inicial de la sucesion
7
8 for n in range(5):
9
10 # En esta linea se calcula el termino n+1 conociendo el termino n
11     y[n+1] = 3 y[n]
12
13 # Esta linea imprime los valores de la sucesion
14     print(y[n+1])
15
16 # Las siguientes lineas grafican la sucesion
17 plt.figure(figsize=(6,4))
18
19 plt.plot (y, '.k', mfc="r", mec="r", markersize=15)
20
21 plt.xlabel('$n$', size = 16)
22 plt.ylabel('$y_n$', size = 16)
23 plt.text (0.2 ,215 ,r'$y_n = 3^n y_0$', size = 16)
24
25 plt.tight_layout()
26 plt.savefig("sucesionnew2.png",dpi=800)
27 plt.show()
```

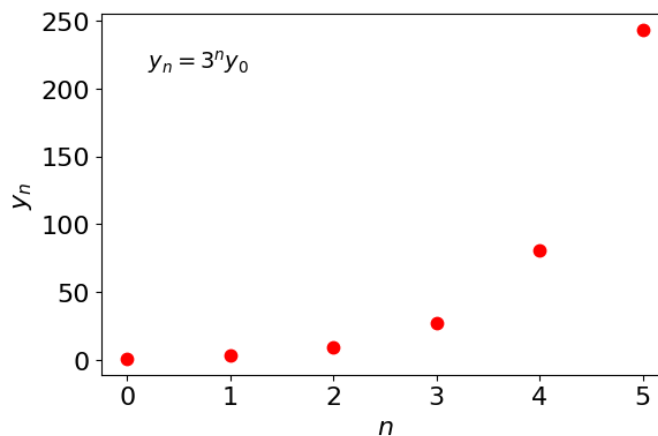


Figura 4.2: Representación gráfica de la sucesión  $y_{n+1} = 3y_n$  con valor inicial  $y_0 = 1$ .

Observe que la sucesión anterior se puede ver también como

$$y_{n+1} = g(y_n),$$

donde  $g(y) = 3y$ . Así podemos también ilustrar la generación de la sucesión  $(y_n)$  usando la construcción en la forma indicada en la Figura 4.3, tomando  $y_0 = 1$ . Compare con la sucesión  $y_{n+1} = (1/3)y_n$  graficada en la Figura 4.4, también tomando  $y_0 = 1$ . Igualmente, en la Figura 4.5 se ilustra la construcción de la sucesión  $y_{n+1} = \cos(y_n)$ , tomando  $y_0 = 0,2$ .

En general, toda relación de recurrencia de la forma  $y_{n+1} = g(y_n)$  puede ser analizada mediante gráficas del tipo mostrado en las Figuras 4.3, 4.4, 4.5.

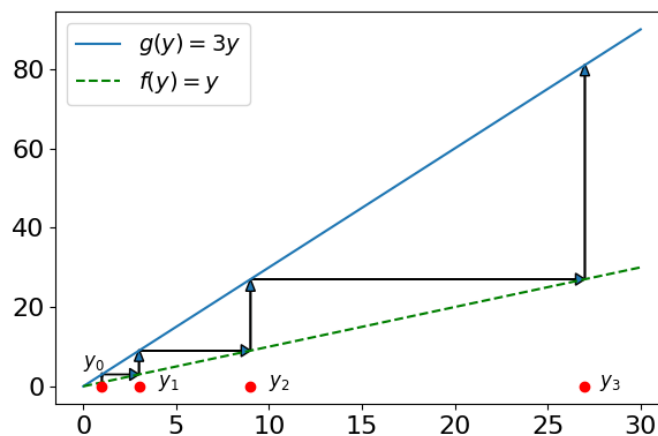


Figura 4.3: Gráfica de la sucesión  $y_{n+1} = 3y_n$ , con  $y_0 = 1$ . En línea discontinua está la diagonal del plano, la cual se usa como apoyo en la construcción de la sucesión cuyos valores quedan sobre el eje horizontal, y la línea sólida corresponde a la gráfica de la función  $g(y) = 3y$ . Observe que la pendiente de gráfica de la función  $g(y) = 3y$  es mayor que la pendiente de la diagonal la cual es 1.

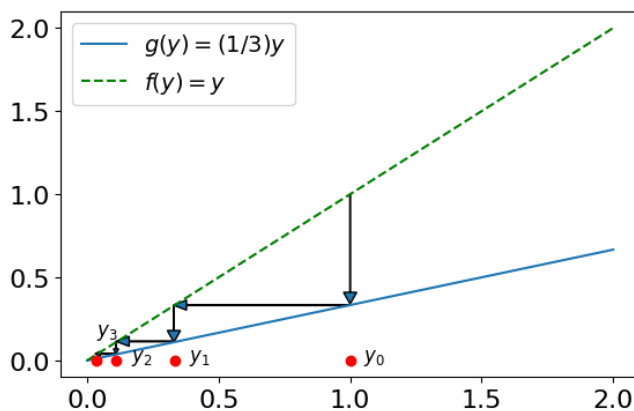


Figura 4.4: Gráfica de la sucesión  $y_{n+1} = \frac{1}{3}y_n$ , con  $y_0 = 1$ . En línea discontinua oblicua está la diagonal del plano, la cual se usa como apoyo en la construcción de la sucesión cuyos valores quedan sobre el eje horizontal, y la línea sólida corresponde a la gráfica de la función  $g(y) = \frac{1}{3}y$ . Observe que la pendiente de la recta tangente a la función  $g(y) = \frac{1}{3}y$  es menor que la pendiente de la diagonal la cual es 1.

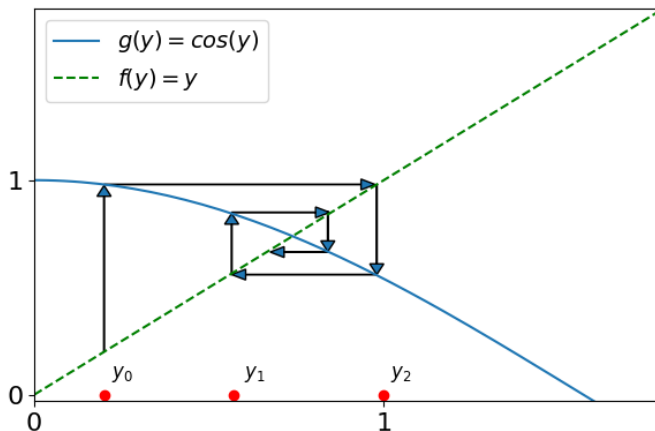


Figura 4.5: Gráfica de la sucesión  $y_{n+1} = \cos(y_n)$ , con  $y_0 = 0,2$ . En línea discontinua está la diagonal del plano, la cual se usa como apoyo en la construcción de la sucesión cuyos valores quedan sobre el eje horizontal, y la línea sólida corresponde a la gráfica de la función  $g(y) = \cos(y)$ .

2. Consideremos la generalización de la ecuación en diferencias anterior, de la forma

$$y_{n+1} = ky_n,$$

donde  $k$  es una constante y siendo  $y_0$  la condición inicial. Vemos que en este caso, para obtener el término siguiente de la sucesión ( $y_n$ ) debemos multiplicar por la constante  $k$

el término anterior. Así obtenemos

$$\begin{aligned}y_1 &= ky_0 \\y_2 &= ky_1 = k(ky_0) = k^2y_0 \\y_3 &= ky_2 = k(k^2y_0) = k^3y_0 \\y_4 &= ky_3 = k(k^3y_0) = k^4y_0.\end{aligned}$$

Continuando con el proceso se llega a

$$y_n = k^n y_0, \quad n \in \mathbb{N}.$$

Observe que mientras la expresión anterior permite calcular cualquier término  $y_n$  de la sucesión, el uso de la ley de recurrencia  $y_{n+1} = ky_n$ , requiere el cálculo de todos los términos anteriores  $y_0, y_1, \dots, y_n$  a fin de poder calcular el término  $y_{n+1}$ .

3.  $y_{n+1} = ky_n + b$ .

Aquí  $k$  y  $b$  son constantes. Se observa que se requiere de nuevo tener el valor inicial  $y_0$  a fin de poder calcular la sucesión. Calculando algunos términos obtenemos

$$\begin{aligned}y_1 &= ky_0 + b \\y_2 &= ky_1 + b = k(ky_0 + b) + b = k^2y_0 + (k + 1)b \\y_3 &= ky_2 + b = k(k^2y_0 + kb) = k^3y_0 + (k^2 + k + 1)b \\y_4 &= ky_3 + b = k(k^3y_0 + (k^2 + k + 1)b) = k^4y_0 + (k^3 + k^2 + k + 1)b.\end{aligned}$$

Continuando con el proceso llegamos a

$$y_n = k^n y_0 + b \left( \sum_{m=1}^n k^{m-1} \right),$$

Convirtiendo la serie en una serie geométrica y resolviendo, se tiene que:

$$y_n = k^n y_0 + b \left( \frac{1 - k^n}{1 - k} \right)$$

### Ecuaciones en diferencias en aplicaciones

En los anteriores ejemplos, se ha querido ilustrar el concepto de sucesión y algunas herramientas matemáticas para su análisis. Además, se ha dado una cierta ecuación en diferencias la cual no está conectada con una aplicación en particular. En los ejemplos que siguen, describiremos diferentes situaciones que surgen en aplicaciones concretas y en las que, al contrario, deberemos buscar un modelo basado en una ecuación en diferencias que las describa apropiadamente.



**Ejemplo 3:** Supongamos que una población que es inicialmente de 100 insectos crece el triple, cada día. Sea  $P_n$  la cantidad de insectos en el día  $n$ . Así la población inicial (es decir, en el día 0) de insectos es  $P_0 = 100$ . De acuerdo a la información dada, para calcular la población de insectos en cada día, es necesario multiplicar por 3 la cantidad de insectos que se tiene en el día anterior. Procediendo de esta manera, obtenemos que

$$\begin{aligned}P_1 &= 3P_0 = 300 \\P_2 &= 3P_1 = 3 \times 300 = 900 \\P_3 &= 3P_2 = 3 \times 900 = 2700.\end{aligned}$$

En general, de acuerdo a lo anterior, en el día  $n$  tendremos

$$P_{n+1} = 3P_n.$$

Observe que un cálculo explícito de la sucesión  $P_n$  permitiría calcular para todo  $n$  la población sin necesidad de conocer los valores en los tiempos anteriores. Para hacer esto, de acuerdo al razonamiento presentado antes llegamos a que

$$P_n = 3^n P_0, \quad n \in \mathbb{N},$$

con la cual podemos encontrar la población de insectos solo conociendo su número inicial  $P_0$ .

**Ejemplo 4:**

Supongamos que una determinada población con 100 individuos iniciales, duplica su número en cada generación, y que además, 10 nuevos individuos se incorporan en cada generación procedente de otro lugar. De acuerdo a la situación planteada, deducimos que si  $P_n$  es el número de habitantes de la población en la generación  $n$ , entonces

$$P_{n+1} = 2P_n + 10, \quad P_0 = 100.$$

Igual que en el ejemplo anterior, para calcular la población  $P_{n+1}$  usando la ley de recurrencia anterior, se requieren calcular previamente todos los anteriores términos. Sin embargo, en este caso simple podemos encontrar la expresión explícita

$$P_n = 2^n P_0 + 2^{n-1} 10 = 100 \times 2^n + 10 \times 2^{n-1}, \quad n \in \mathbb{N}, \quad (4.3)$$

con la cual podemos calcular el tamaño de la población  $P_n$  en cada generación  $n$  sólo conociendo el valor inicial  $P_0$ .

**Ejemplo 6:** Una población de mariposas se cuenta cada cierto periodo de tiempo (cada 3 semanas, por ejemplo), y se observa que en cada periodo crece el triple de lo que creció en el periodo anterior. Deseamos describir el tamaño de la población para cada periodo de tiempo. Se conocen además las cantidades de mariposas  $P_0, P_1$  en los dos primeros periodos.

Para iniciar la construcción del modelo, sea  $P_n$  el tamaño de la población en el periodo de conteo  $n$ . Así, tenemos que el crecimiento de la población en el periodo  $n$  corresponde a

$P_{n+1} - P_n$ , y el crecimiento de la población en el periodo siguiente puede expresarse como  $P_{n+2} - P_{n+1}$ . Por otro lado, de acuerdo al planteamiento del problema, tenemos la relación

$$\text{Crecimiento en un periodo} = 3 \times \text{Crecimiento en el periodo anterior} ,$$

de lo cual llegamos al modelo matemático

$$P_{n+2} - P_{n+1} = 3(P_{n+1} - P_n),$$

o equivalentemente,

$$P_{n+2} - 4P_{n+1} + 3P_n = 0.$$

Dado que esta ecuación en diferencias es lineal y homogénea de coeficientes constantes, proponemos una solución del tipo  $P_n = \lambda^n$ , sustituyendo esto en la ecuación en diferencias, se obtiene que,

$$\lambda^{n+2} - 4\lambda^{n+1} + 3\lambda^n = 0$$

operando, llegamos a

$$\lambda^2 - 4\lambda + 3 = 0$$

Tenemos entonces una ecuación cuadrática cuyas raíces son  $\lambda_1 = 3$  y  $\lambda_2 = 1$ . Entonces la solución general es una combinación lineal de las soluciones para cada raíz:

$$P_n = c_1 + c_2 3^n$$

**Ejemplo 6:** Supongamos que un par de conejos recién nacidos, un macho y una hembra, se colocan en el campo. Los conejos son fértiles a la edad de un mes, así que al final del segundo mes una hembra puede producir otro par de conejos. Supongamos que nuestros conejos nunca mueren, y que las hembras siempre producen un nuevo par (un macho y una hembra) cada mes, desde el segundo de los meses. Una pregunta que Leonardo Pisano (más conocido por su apellido Fibonacci) plantea en la tercera sección de su libro Liber Abbaci (1202 D.C.) fue la siguiente: ¿Cuántos pares de conejos habrán en un año ?

Si observamos atentamente, del enunciado del problema deducimos que:

- Inicialmente tenemos un par.
- Al final del primer mes, hay un solo par.
- Al final del segundo mes, la hembra produce un nuevo par, así que ahora tendremos dos parejas de conejos en el campo.
- Al final del tercer mes, la hembra inicial produce un segundo par, haciendo que en el campo tengamos tres pares de conejos.
- Al final del cuarto mes, la hembra original ha producido otro par y la hembra nacida dos meses antes produce su primer par, con lo que tendremos 5 pares, y así sucesivamente.

Obtenemos entonces que el número de parejas de conejos en el campo forma una sucesión de la forma 1, 1, 2, 3, 5, 8, 13, 21, 34, .... Note que en un mes dado, se tienen los conejos del mes anterior y uno nuevo por cada par de los que había dos meses antes.

Vemos así que el número de parejas de conejos en el mes  $n \geq 2$  se puede obtener sumando el número de parejas en los meses anteriores  $n-1$  y  $n-2$ . Concluimos del razonamiento anterior que si denotamos por  $y_n$  al valor del número de parejas de conejos que hay en el campo en el mes  $n$ , tenemos

$$y_0 = y_1 = 1, \quad y_n = y_{n-1} + y_{n-2}, \quad n \geq 2.$$

Observe que equivalentemente podemos escribir esta ecuación como

$$y_0 = y_1 = 1, \quad y_{n+1} = y_n + y_{n-1}, \quad n \geq 1.$$

La sucesión anterior recibe el nombre de **sucesión de Fibonacci**. Curiosamente Fibonacci nunca llegó a estudiar esta sucesión en profundidad. Por ejemplo, fue posteriormente que Albert Girard en 1625 quien descubrió el patrón que siguen sus términos. Aunque casi desconocido para el público en general, A. Girard tiene el honor de haber sido el primero en conjeturar que toda ecuación polinómica de grado  $n$  admite precisamente  $n$  raíces, que pueden ser tanto reales como imaginarias. Este es el Teorema fundamental del Álgebra, cuya primera demostración fue desarrollada por Carl F. Gauss en su tesis doctoral.

Note que la sucesión  $(y_n)$  crece indefinidamente conforme  $n$  crece, lo cual puede ser constatado en la Figura 4.6 la cual es generada por el siguiente código en Python:

```

1
2 # Calcula y grafica la sucesion de Fibonacci
3
4 import matplotlib.pyplot as plt
5 import numpy as np
6
7 N = 20
8 y = np.zeros(N)
9 y[0] = 1 #valor inicial
10 y[1] = 1 #segundo valor
11
12 for n in range(1, N-1):
13     y[n+1] = y[n] + y[n-1]
14
15 plt.figure(figsize=(6,4))
16
17 plt.plot (y, 'o', mfc="r", mec="r", markersize=10)
18
19 plt.xlabel('$n$', size = 16)
20 plt.ylabel('$y_n$', size = 16)
21
22 plt.xticks(size=16)
23 plt.yticks(size=16)
24 plt.xticks(range(0, 20, 5))
25 plt.yticks(range(1000, 9000, 2000))
26
27 plt.tight_layout()
28 plt.savefig("sucesionfibonacci.png",dpi=800)

```

```
plt.show()
```

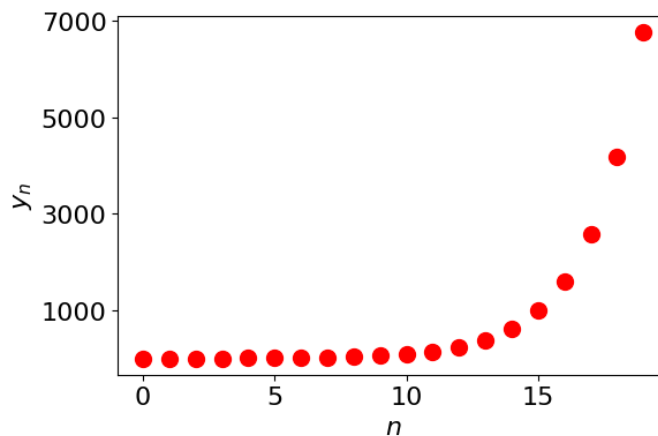


Figura 4.6: Gráfica de la sucesión de Fibonacci ( $y_n$ ).

Las siguientes líneas de código (como continuación del anterior) permiten descubrir una propiedad importante de esta sucesión a través de una gráfica de los cocientes de términos consecutivos en función del número de días  $n$ , la cual se muestra en la Figura 4.7.

```

1 #Estudio de los cocientes consecutivos de la sucesion de Fibonacci
2
3 #Calcula los cocientes en la suc. de Fibonacci y el numero aureo
4
5 a = y[-1]/y[-2]
6
7 for n in range(N-1):
8     plt.plot(n, y[n+1]/y[n], 'or', markersize = 10)
9
10 # Grafica la sucesion de cocientes y la recta del numero aureo
11
12 plt.axhline(y = a, xmin = 0, xmax = N, label = 'a =%5.4f' %a)
13
14 plt.xticks(size=16)
15 plt.yticks(size=16)
16 plt.xticks(range(0, 18, 5))
17
18 plt.ylabel('$y_{n+1}/y_n$', size = 16)
19 plt.xlabel('$n$', size = 16)
20
21 plt.legend(prop = {'size':14})
22
23 plt.tight_layout()
24 plt.savefig("limitefibonacci.png",dpi=800)
25 plt.show()

```

Se observa que aunque la sucesión ( $y_n$ ) crece indefinidamente, cuando  $n \rightarrow \infty$ , el cociente  $y_{n+1}/y_n$  se aproxima a cierto valor positivo el cual descubrimos que está alrededor de 1,6. En

efecto,

$$1/1 = 1, \quad 2/1 = 2, \quad 3/2 = 1,5, \quad 5/3 = 1,666, \quad 8/5 = 1,6, \quad 13/8 = 1,625, \\ 21/13 = 1,61538.$$

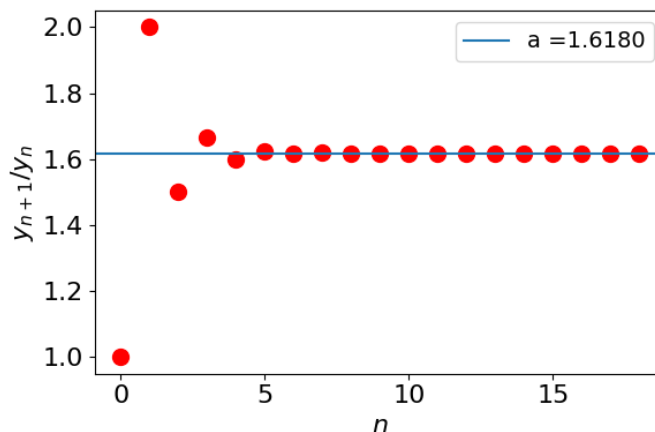


Figura 4.7: Comportamiento del cociente  $(y_{n+1}/y_n)$  en la sucesión de Fibonacci  $(y_n)$ .

Este hecho fue descubierto en 1753 por Robert Simson, profesor de la Universidad de Glasgow. El cálculo exacto de este límite es sencillo, pero demostrar su existencia es más complicado. Si suponemos que el límite  $\lim_{n \rightarrow \infty} y_{n+1}/y_n$  existe y es igual a, digamos  $L$ , éste se puede calcular de manera exacta observando que, dividiendo por  $y_n$  la expresión de la sucesión de Fibonacci, se obtiene

$$\frac{y_{n+1}}{y_n} = 1 + \frac{y_{n-1}}{y_n},$$

de lo cual tomando límite en ambos lados, obtenemos que

$$L = 1 + \frac{1}{L}.$$

Aquí hemos usado el hecho que

$$\lim_{n \rightarrow \infty} \frac{y_{n+1}}{y_n} = \lim_{n \rightarrow \infty} \frac{y_n}{y_{n-1}} = L,$$

y

$$\lim_{n \rightarrow \infty} \frac{y_{n-1}}{y_n} = \lim_{n \rightarrow \infty} \frac{1}{\frac{y_n}{y_{n-1}}} = \frac{1}{L}.$$

De lo anterior, obtenemos que

$$L^2 - L - 1 = 0.$$

Así, el valor positivo posible del límite de la sucesión de cocientes  $y_{n+1}/y_n$  es

$$L = \frac{1 + \sqrt{5}}{2}.$$

Este valor de  $L$  se conoce como **número áureo**. En la Figura 4.7 se corrobora numéricamente que la sucesión de cocientes  $(y_{n+1}/y_n)$  se acerca al valor de  $L$  conforme el número de días  $n$  crece.

Los ejemplos anteriores muestran algunas situaciones donde relaciones de recurrencia permiten definir sucesiones  $(y_n)$ . Podemos visualizar la construcción de la sucesión  $(y_n)$  definida por 4.2, mediante gráficas tales como las Figuras 4.3, 4.4, 4.5.

#### 4.1.1. Métodos cualitativos y computacionales

A continuación utilizaremos un ejemplo sobre la evolución de un cultivo de cierto tipo de levaduras, para construir un modelo matemático (los datos experimentales son tomados de [1]) ilustrando los pasos necesarios en su desarrollo y que fueron descritos en el capítulo 2.

**Descripción del fenómeno real y objetivos del modelo:** El problema que vamos a analizar es conocer el número de bacterias en un cierto cultivo en cualquier tiempo, dado que solo se conoce su cantidad en un número finito de tiempos que han sido medidos experimentalmente de acuerdo a la tabla 4.1. Estos son tomados de [2]: El tipo de preguntas que podemos comenzarnos a hacer son las siguientes: ¿Cómo varía el número de levaduras con el tiempo? ¿Qué tipo de variables influyen en su desarrollo?

**Elección de variables.** En la fase de experimentación se ha podido observar que la célula crece, se divide en dos y cada una de ellas inicia de nuevo el proceso de crecimiento. Se detecta además por ejemplo, que el tiempo necesario para que crezca una célula y se duplique es aproximadamente  $N_0$  horas. Por lo tanto, el tiempo de vida de una célula, podríamos considerarlo como una variable que interviene en el problema. Es evidente que existen muchas otras variables, las cuales pueden ser clasificadas como variables de entrada, que son las que pueden influir en los resultados, y variables de salida, que corresponden a los resultados. En nuestro problema, seleccionamos como variable de salida el tamaño de la población de levaduras  $P_n$  después de  $n$  horas, el cual será la variable independiente. Tenemos entonces una sucesión  $(P_n)_n$  para describir el estado en el tiempo de esta población.

Tiempo en horas	Biomasa de levadura $P_n$	Cambio de biomasa $\Delta P_n = P_{n+1} - P_n$
0	9.6	8.7
1	18.3	10.7
2	29.0	18.2
3	47.2	23.9
4	71.1	48.0
5	119.1	55.5
6	174.6	82.7
7	257.3	

Cuadro 4.1: Crecimiento de un cultivo de levadura frente al tiempo en horas; los datos de la primera y segunda columnas son tomados de [2].

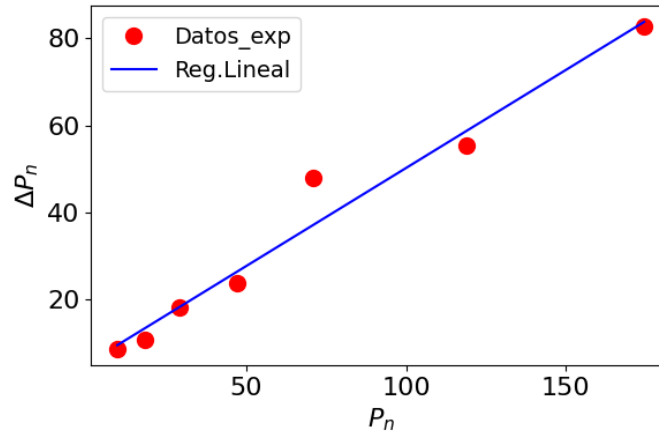


Figura 4.8: Cambio de biomasa versus biomasa de levaduras. Además se ha graficado la recta que mejor ajusta los datos de acuerdo al proceso de mínimos cuadrados visto en el capítulo 3.

#### Relaciones cualitativas entre las variables y construcción del modelo matemático.

Al colocar los datos de la segunda y tercera columna de la tabla 4.1 en la Figura 4.8, donde  $P_n$  se coloca en el eje horizontal y  $\Delta P_n$  en el eje vertical, la disposición en la que quedan los datos experimentales sugiere un patrón de crecimiento donde la tasa media de variación de  $P_n$  es constante, es decir, en la forma

$$\frac{\Delta P_n}{P_n} = k,$$

o equivalentemente,

$$\Delta P_n = P_{n+1} - P_n = kP_n, \quad (4.4)$$

donde  $k$  es una constante.

Para aproximar la constante  $k$  podemos usar una estrategia de ajuste lineal, la cual es una técnica muy importante en el análisis de datos, la cual nos permite determinar el grado de dependencia entre el conjunto de valores  $(P_n, \Delta P_n)$ .

Utilizando la estrategia de mínimos cuadrados vista en el capítulo 3, determinamos la recta que mejor se ajusta a los datos disponibles en el sentido de que la desviación cuadrática media de un conjunto de puntos  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$  a una recta  $y = kx + b$ , sea mínima. Es decir, se buscan los parámetros  $k$  (pendiente de la recta),  $b$  (corte con el eje vertical) tal que los cuadrados de las distancias verticales de la recta a los datos

$$d = \sum_{i=1}^n (y_i - (kx_i + b))^2, \quad (4.5)$$

sea mínima.

En el caso del modelo 4.4 buscaremos un ajuste lineal a través de una recta. Así, el cálculo de estos parámetros puede realizarse utilizando la rutina en python `polyfit()` :

```

1
2 # Calcula recta de regresion lineal que pasa por el origen
3 # necesaria en el ejemplo del crecimiento de bacterias

```

```

4
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 plt.figure(figsize=(6,4))
9
10 y_datos = np.array([8.7, 10.7, 18.2, 23.9, 48.0, 55.5, 82.7])
11 x_datos = np.array([9.6, 18.3, 29.0, 47.2, 71.1, 119.1, 174.6])
12
13 reg = np.polyfit(x_datos, y_datos, 1)
14 x_new = np.linspace(9.6, 174.6, 200)
15 y_new = np.polyval(reg, x_new)
16
17 plt.plot(x_datos, y_datos, 'o', mfc="r", mec="r", label = 'Datos_exp',
18         markersize=10)
19 plt.plot(x_new, y_new, 'b-', label = 'Reg.Lineal')
20
21 plt.xlabel('$P_n$', size = 16)
22 plt.ylabel('$\Delta P_n$', size = 16)
23
24 plt.xticks(size=16)
25 plt.yticks(size=16)
26 plt.xticks(range(50, 180, 50))
27 plt.yticks(range(20, 90, 20))
28
29 plt.legend(prop = {'size':14})
30
31 plt.tight_layout()
32 plt.savefig("poblaciones_gr1.png",dpi=800)
33 plt.show()

```

### Consecuencias del modelo

La recta de regresión se muestra en la Figura 4.8, con un valor aproximado de su pendiente  $k \approx 0,5$  y  $b \approx 5,3$  para el intercepto con el eje y, de lo cual llegamos al modelo lineal

$$\Delta P_n = P_{n+1} - P_n = 0,5P_n + 5,3 \quad (4.6)$$

o equivalentemente

$$P_{n+1} = P_n + 0,5P_n + 5,3 = 1,5P_n + 5,3. \quad (4.7)$$

En la Figura 4.9 se puede ver que hay un ajuste razonable en tiempos pequeños de la predicción del modelo lineal 4.7 con respecto a los datos experimentales conocidos. Se ajusta el crecimiento de la población a la predicción dada por el modelo en el tiempo considerado. Esto significa que el modelo de crecimiento exponencial describe muy bien la población de bacterias en el rango de tiempo considerado.

### Predicción y validación del modelo

Todos los modelos tienen un cierto dominio de validez y tienen su importancia en las circunstancias adecuadas. Por ejemplo, un modelo de un péndulo podría ser válido solamente para posiciones de ángulos pequeños, mientras que otro modelo puede ser apropiado para ángulos



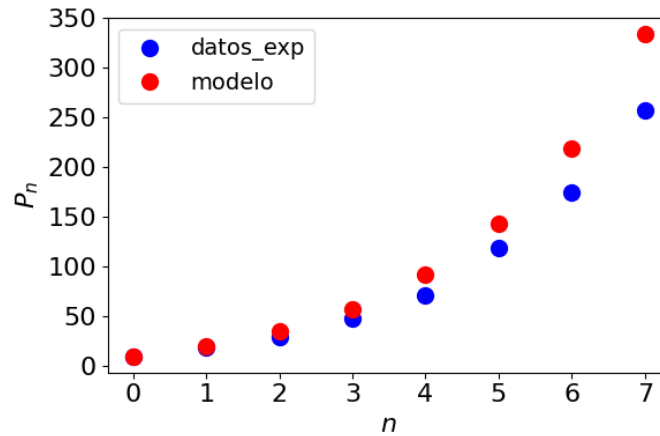


Figura 4.9: Validación de los resultados obtenidos del modelo lineal 4.7 (puntos en rojo) comparando con los datos experimentales conocidos (puntos en azul).

grandes. Las leyes de Newton del movimiento son válidas con buena precisión dentro de un rango de velocidad, mientras que cerca a la velocidad de la luz no son apropiadas para describir el movimiento de las partículas. Así modelos y simulaciones no pueden reemplazar la observación y la experimentación, pero ellos constituyen un complemento útil e importante. En nuestro caso de estudio, verificamos el dominio de validez del modelo construido, utilizando datos experimentales adicionales en tiempos mayores de observación. Resaltamos que el modelo de crecimiento exponencial desarrollado anteriormente es útil y describe apropiadamente la población de bacterias considerada en un tiempo de observación corto y tiene la ventaja de ser un modelo cuya formulación matemática es sencilla.

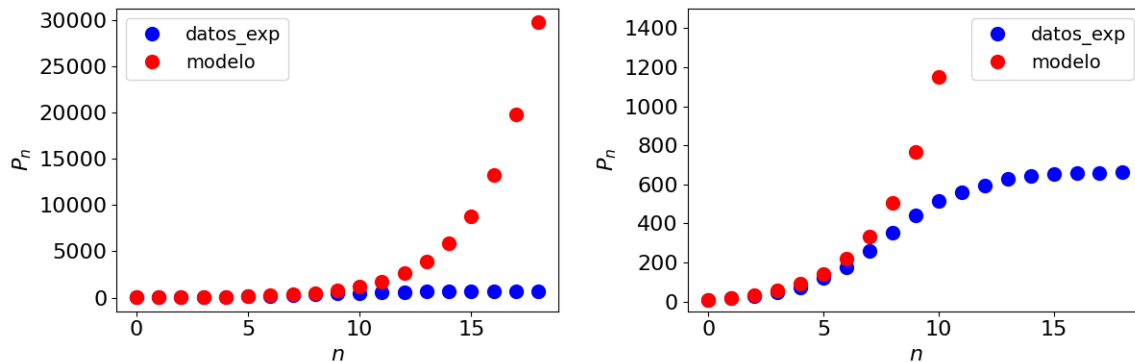


Figura 4.10: Validación de los resultados del modelo lineal 4.7 (en puntos rojos) comparando con los datos experimentales conocidos en un tiempo mayor (en puntos azules). Se agrega una ampliación en la gráfica de la derecha para visualizar mejor los datos para un tiempo mayor (puntos azules).

Ya en la situación concreta que estamos analizando, observamos que si graficamos en la Figura 4.10 datos experimentales de la misma población de levaduras bajo estudio en un tiempo de hasta 18 horas dados en la tabla 4.2, vemos que hay una desviación apreciable entre los datos

Tiempo en horas	Biomasa de levadura $P_n$	Cambio de biomasa $\Delta P_n = P_{n+1} - P_n$
0	9.6	8.7
1	18.3	10.7
2	29.0	18.2
3	47.2	23.9
4	71.1	48.0
5	119.1	55.5
6	174.6	82.7
7	257.3	93.4
8	350.7	90.3
9	441.0	72.3
10	513.3	46.4
11	559.7	35.1
12	594.8	34.6
13	629.4	11.4
14	640.8	10.3
15	651.1	4.8
16	655.9	3.7
17	659.6	2.2
18	661.8	

Cuadro 4.2: Crecimiento de un cultivo de levadura frente al tiempo en horas; datos de R. Pearl, "The Growth of Population", Quart. Rev. Biol. 2 (1927): 532–548.

experimentales y las predicciones dadas por el modelo lineal propuesto 4.7 cuando el tiempo pasa de 10 horas. Se nota en la comparación de los resultados que el modelo predice que la población de levaduras es siempre creciente y de manera exponencial, mientras que los datos experimentales muestran que la tasa de crecimiento de la población cambia disminuyendo al cabo de cierto tiempo al parecer por cierta limitante. Así que es necesario refinar el modelo lineal propuesto inicialmente, a fin de replicar este comportamiento de manera más apropiada.

### Nuevo proceso de modelamiento

Habiendo llegado a la conclusión de que nuestro modelo lineal tiene discrepancias significativas con una muestra mayor de datos experimentales, entonces procedemos a proponer uno nuevo más adecuado para la situación específica tratada.

Para tener en cuenta la limitante en el crecimiento de la población consideraremos una corrección del modelo anterior donde la tasa promedio de incremento de la población está limitado por cierto valor  $\beta$ , es decir,

$$\frac{\Delta P_n}{P_n} = k - \beta P_n, \quad (4.8)$$

o equivalentemente,

$$\Delta P_n = P_{n+1} - P_n = kP_n - \beta P_n^2, \quad (4.9)$$

donde  $\beta$  es una tasa límite del crecimiento de la población. Observe que a diferencia del anterior modelo de crecimiento exponencial, el modelo 4.9 es no lineal debido a la presencia del término cuadrático  $P_n^2$ .

Tiempo en horas	Biomasa de levadura $P_n$	Cambio- biomasa $\frac{\Delta P_n}{P_n}$
0	9.6	0.906
1	18.3	0.585
2	29.0	0.628
3	47.2	0.506
4	71.1	0.675
5	119.1	0.466
6	174.6	0.474
7	257.3	0.363
8	350.7	0.257
9	441.0	0.164
10	513.3	0.090
11	559.7	0.063
12	594.8	0.058
13	629.4	0.018
14	640.8	0.016
15	651.1	0.007
16	655.9	0.006
17	659.6	0.003
18	661.8	

Cuadro 4.3: Crecimiento de un cultivo de levadura frente al tiempo en horas; datos de R. Pearl, "The Growth of Población", Quart. Rev. Biol. 2 (1927): 532–548. Aquí se calcula la razón  $P_{n+1}/P_n$  versus la biomasa de levaduras  $P_n$ .

De la disposición de la lista de datos  $(P_n, \frac{\Delta P_n}{P_n})$  en la Figura 4.11, se observa que de nuevo los parámetros  $k$  y  $\beta$  pueden ser aproximados a través de regresión lineal. De esta manera,  $k$  correspondería al corte con el eje vertical de la recta de regresión y  $-\beta$  sería su pendiente. Los resultados del ajuste lineal son  $k = 0,669351$ ,  $\beta = 0,0011$ . Compruebe estos resultados modificando apropiadamente la rutina en python descrita antes. El modelo dado en 4.9 se conoce como **modelo de crecimiento logístico** de una población y los números  $k, \beta$  se llaman **coeficientes vitales** de la población.

#### Validación del nuevo modelo

En la Figura 4.12 se puede ver la predicción dada por el modelo no lineal 4.9 graficada junto con los datos experimentales. Se observa ahora un mejor ajuste que en el caso del modelo lineal para un dominio de tiempo de hasta 18 horas.

#### 4.1.2. Aplicaciones

En esta sección presentaremos algunas situaciones en las que se requieren modelos dinámicos basados en ecuaciones en diferencias para estudiar los fenómenos considerados.

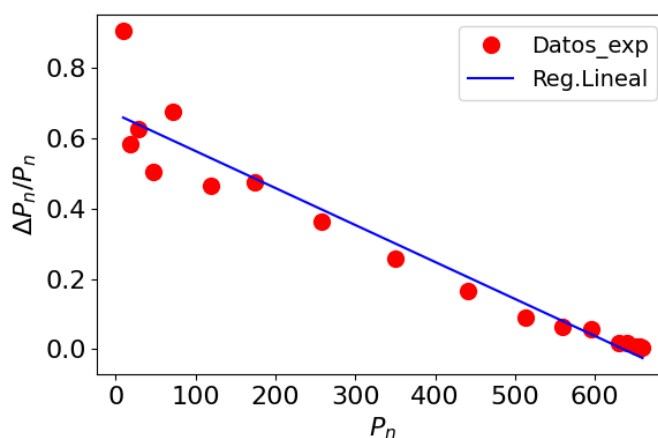


Figura 4.11: Ajuste lineal para aproximar los parámetros  $k$  y  $\beta$ .

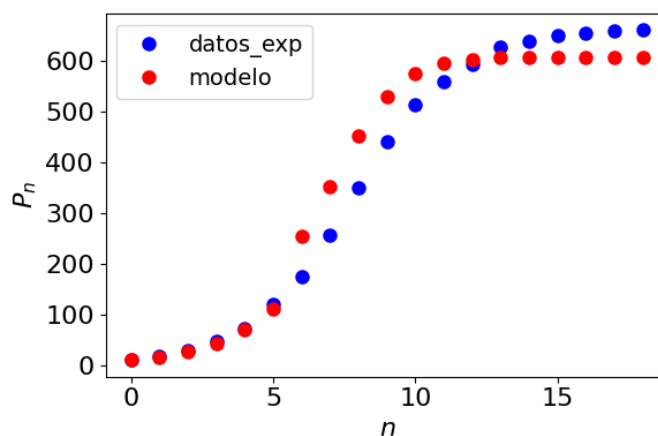


Figura 4.12: Validación de los resultados del modelo No lineal 4.9 (en puntos rojos) comparando con los datos experimentales conocidos (en puntos azules).

### Modelo para la venta de codornices

En esta aplicación ilustraremos el uso del modelamiento para estimar el número de codornices disponibles para vender después de un número de días. Utilizaremos los pasos descritos en el capítulo 1 para la construcción de un modelo matemático.

### Planteamiento del problema

Se quiere montar un negocio de venta de codornices japonesas. Si se compra inicialmente un número de codornices, ¿cuántas habrá para vender después de un número de días?

### Descripción de la situación

En una población de codornices japonesas, de diferentes edades y sexo, solo las hembras ponen

huevos y solo un número de ellos eclosionan en polluelos de codorniz. Además, en promedio las codornices mueren después de alcanzar una edad específica.

### Simplificación de la realidad

Supuestos:

- Al inicio se tiene un número igual de codornices machos y hembras y con edades variadas.
- Del número de codornices que nacen, la mitad son machos y la otra mitad son hembras.
- Todas las codornices mueren a la misma edad.

Variables:

- $t$  : tiempo en días.
- $P_t$  : población de codornices en el día  $t$ . Número de codornices en el día  $t$ .

Parámetros:

- $P_0$  : población inicial de codornices.
- $n$  : tasa de natalidad de las codornices. Número de codornices que nacen por día.
- $m$  : tasa de mortalidad de las codornices. Número de codornices que mueren por día.

### Análisis de datos

Inicialmente se tienen 20 codornices japonesas. Se sabe que estas codornices viven aproximadamente 1000 días. Además, una codorniz hembra pone alrededor de 146 huevos fértiles por año.

La población inicial de codornices es  $P_0 = 20$  codornices.

Si las edades de las codornices es variada, la tasa de mortalidad de las codornices es

$$m = \frac{1}{1000 \text{ días}} = \frac{0,001}{\text{día}}.$$

En promedio, una codorniz hembra produce  $146/365 = 0.4$  codornices cada día. Por tanto, la tasa de natalidad de las codornices es

$$n = \frac{1}{2} \frac{0,4}{\text{día}} = \frac{0,2}{\text{día}},$$

ya que asumimos que la mitad del número de codornices son hembras.

### Construcción del modelo matemático

Nuestro interés es predecir cómo cambia la población de codornices a medida que pasan los días. En términos matemáticos, el cambio en la población de codornices entre un día  $t$  cualquiera y el día siguiente es:

$$\Delta P = P_{t+\Delta t} - P_t, \tag{4.10}$$

donde  $\Delta t$  es un intervalo de tiempo de un día. El nacimiento de codornices hace que la población aumente mientras que la muerte de codornices hace que la población disminuya. La tasa de natalidad  $n$  multiplicada por la población  $P_t$  y el intervalo de tiempo  $\Delta t$  da como resultado el número de codornices que nacen en un intervalo de un día mientras que la tasa de mortalidad  $m$  multiplicada por la población  $P_t$  y el intervalo de tiempo  $\Delta t$  da el número de codornices que mueren en un intervalo de un día:

$$\begin{aligned}\Delta P &= nP_t\Delta t - mP_t\Delta t, \\ \Delta P &= (n - m)P_t\Delta t.\end{aligned}$$

Reemplazando los valores de  $n$  y  $m$  obtenemos:

$$\Delta P = (0,2 - 0,001)P_t\Delta t.$$

Puesto que la tasa de mortalidad de las codornices es mucho más pequeña que la tasa de natalidad, esta ecuación puede aproximarse como:

$$\Delta P \approx 0,2P_t\Delta t. \quad (4.11)$$

Al reemplazar la ecuación 4.10 en la ecuación 4.11, con  $\Delta t = 1$  día, se obtiene la ecuación de diferencias que permite calcular el número de codornices por cada día que pasa:

$$P_{t+1} = P_t + 0,2P_t = 1,2P_t, \quad (4.12)$$

con  $t = 0, 1, 2, \dots$  días.

### Solución del modelo matemático

Puesto que la población inicial de codornices es  $P_0 = 20$ , la ecuación 4.12 arroja los resultados tabulados en la tabla 4.4 para la población de codornices durante los primeros 10 días, donde los resultados han sido redondeados al número entero más cercano. El código en Python que da los resultados de la tabla es el siguiente:

```

1 # Calcula la poblacion de codornices
2
3 y = np.zeros(12) # Vector para guardar la poblacion de codornices
4 y[0] = 20.0 # Valor inicial de la sucesion
5
6 print("t (dia)", "      ", "Pt (Poblacion en el dia t)")
7 print("")
8
9 for n in range(10):
10     y[n+1] = 1.2*y[n]
11     print(n, "      ", round(y[n]) ) # Imprime los valores de la
12         sucesion hasta el dia 9
13 print(10, "      ", round(y[n+1]) ) # Imprime el valor en 10 dias

```

### Validación del modelo matemático

Los resultados del modelo propuesto muestran un incremento acelerado de la población de las codornices. Esto hace que el número de codornices crezca muy rápido en pocos días (ver figura 4.13). Se hace necesario por tanto añadir un elemento adicional al modelo que limite el crecimiento de la población de codornices.

$t$ (día)	$P_t$ (Población en el día $t$ )
0	20
1	24
2	29
3	35
4	41
5	50
6	60
7	72
8	86
9	103
10	124

Cuadro 4.4: Número de codornices disponibles para la venta en el día  $t$ , utilizando la ecuación 4.12. Los resultados han sido redondeados al número entero más cercano.

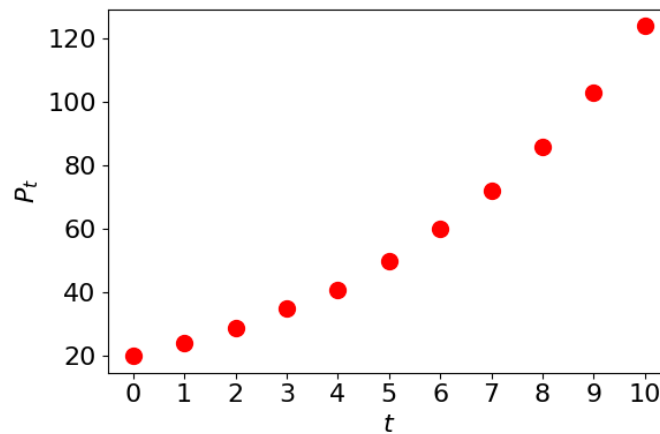


Figura 4.13: Población de codornices como función del número de días calculada mediante la ecuación 4.12 e implementada en el código de python de esta sección.

### Validación del nuevo modelo matemático

Añadamos el supuesto de que cada día se venden 3 codornices. Con esta nueva consideración, tendremos la siguiente ecuación para el cambio de la población de codornices por día si despreciamos la tasa de mortalidad de las codornices:

$$\Delta P = 0,2P_t\Delta t - 3\Delta t, \quad (4.13)$$

con  $\Delta t = 1$  día. Esta ecuación conduce a la siguiente expresión para la población de codornices por cada día que pasa:

$$P_{t+1} = P_t + 0,2P_t - 3 = 1,2P_t - 3, \quad (4.14)$$

con  $t = 0, 1, 2, \dots$  días.

En la figura 4.14 se muestra el crecimiento de la población de codornices incluyendo la venta diaria de tres codornices (puntos azules), comparada con el crecimiento de la población sin venta de codornices (puntos rojos). La figura muestra el descenso de la razón de cambio de la población de codornices cuando se considera la venta diaria de codornices.

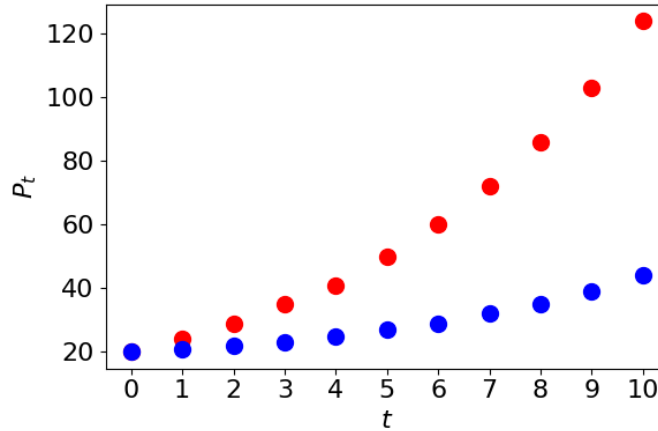


Figura 4.14: Población de codornices como función del número de días calculada mediante la ecuación 4.12 (puntos rojos) y mediante la ecuación 4.14 (puntos azules).

Si el número inicial de codornices cambia, ¿cómo cambia la razón de cambio de la población de codornices? Para resolver este interrogante, dividamos ambos lados de la ecuación 4.13 por  $\Delta t$ :

$$\frac{\Delta P}{\Delta t} = 0,2P_t - 3, \quad (4.15)$$

Si conocemos la población de codornices en un día específico, esta ecuación nos permite obtener la razón de cambio de la población de las codornices por día en dicho momento. Por ejemplo, si  $P_0 = 24$  la razón de cambio de la población de codornices a  $t = 0$  es:

$$\frac{\Delta P}{\Delta t} = 0,2 \times 24 - 3 = 1,8,$$

Debido a la naturaleza exponencial del crecimiento de la población de codornices, un valor pequeño de la razón de cambio indica un crecimiento lento de la población con el tiempo. En cambio, si  $P_0 = 40$ ,



$$\frac{\Delta P}{\Delta t} = 0,2 \times 40 - 3 = 5,0,$$

La población de codornices crecerá rápido. La figura 4.15 muestra el comportamiento de la población de codornices utilizando la ecuación 4.14 para diferentes valores de población inicial  $P_0$ . Esta figura muestra que para una población inicial de  $P_0 = 40$ , el crecimiento de la población de codornices es rápido a medida que los días pasan mientras que si la población inicial es  $P_0 = 24$ , el crecimiento es lento. También se puede observar en la figura que si la población inicial es  $P_0 = 4, 8, 12$  la población de codornices disminuye a medida que los días pasan.

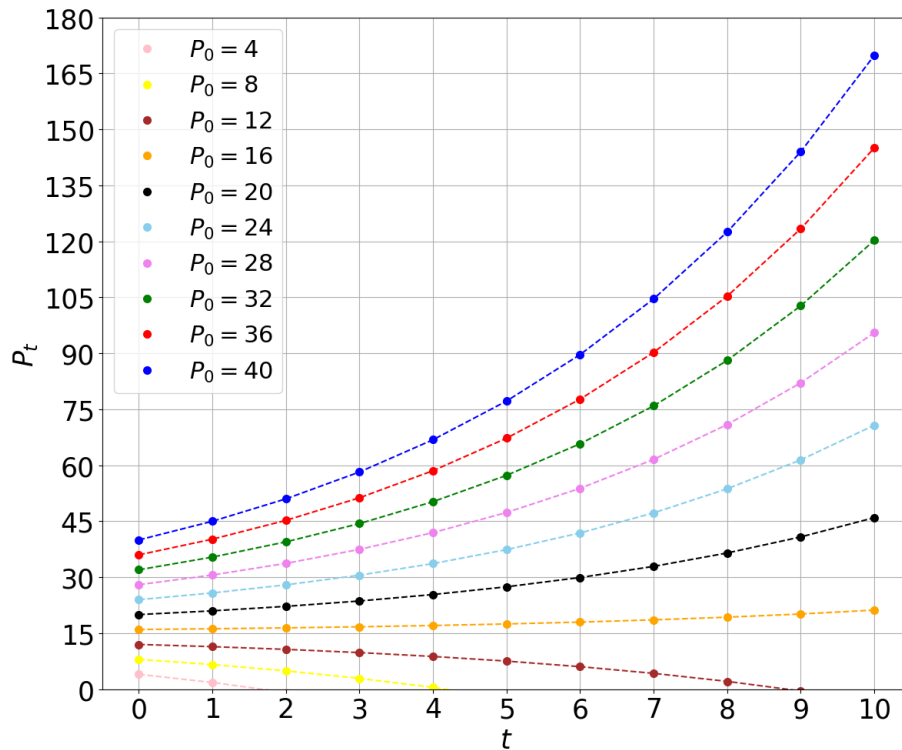


Figura 4.15: Población de codornices como función de la población inicial y del número de días calculada mediante la ecuación 4.14.

## 4.2. Modelos basados en ecuaciones diferenciales

Una ecuación diferencial es una ecuación que relaciona una función desconocida, tanto en una como en varias variables, con sus derivadas hasta cierto orden. Nos limitaremos a considerar un tipo de estas ecuaciones que son llamadas **Ecuaciones diferenciales ordinarias (EDO)**, que son aquellas en las que la función incógnita es una función de una sola variable  $y = y(t)$  y aparecen derivadas ordinarias de la variable independiente  $t$ .

**Notación:** Frecuentemente, utilizaremos la notación

$$y' = \frac{dy}{dt}, \quad y'' = \frac{dy^2}{dt^2}, \quad y''' = \frac{dy^3}{dt^3}, \quad \dots \quad y^{(n)} = \frac{dy^n}{dt^n},$$

para denotar las derivadas de la función  $y(t)$ .

En general, una ecuación diferencial es una expresión en la forma

$$F(t, y, y', y'', \dots, y^{(n)}) = 0,$$

donde  $F$  es una función dada. A la derivada más alta  $n$  que aparece en la ecuación diferencial se le llama el **orden** de la ecuación diferencial.

En diferentes áreas encontramos ejemplos de ecuaciones diferenciales que describen ciertos fenómenos:

**Ejemplos:**

- La ecuación diferencial

$$\frac{dy}{dt} = ky,$$

donde  $k$  es una constante, y  $t$  denota el tiempo, es usada para describir varios fenómenos:

- Dinámica de poblaciones: La población  $y(t)$  en un cultivo de bacterias o la población de un país.
  - Mecanismo del decaimiento de sustancias o materiales radiactivos, donde  $y(t)$  es la cantidad de material y  $k < 0$ . Por ejemplo, usando la cantidad de carbono 14 ( el cual tiene un isótopo radioactivo), se puede encontrar una aproximación del tiempo transcurrido desde la muerte de un organismo.
  - La concentración  $y(t)$  de una sustancia que reacciona en un proceso químico.
- Varios ejemplos simples de ecuaciones diferenciales ordinarias provienen de la segunda ley de Newton  $F = ma$ , donde  $m$  es la masa de un cuerpo,  $F$  es la fuerza aplicada y  $a$  la aceleración inducida sobre la masa. Por ejemplo, si un cuerpo cae bajo la influencia de la fuerza de gravedad (siendo  $g$  la aceleración de la gravedad y suponiendo signo negativo hacia abajo ), entonces

$$ma = -mg,$$

y como  $a = \frac{d^2y}{dt^2}$ , donde  $y(t)$  denota la posición del cuerpo en el tiempo  $t$ , tenemos

$$\frac{d^2y}{dt^2} = -g,$$

la cual es una ecuación diferencial ordinaria. Si además suponemos que sobre el cuerpo actúa una fuerza de fricción con el medio que lo rodea (que se opone al movimiento), cuya magnitud es proporcional a la velocidad instantánea  $dy/dt$ , se obtiene la ecuación diferencial ordinaria

$$m \frac{d^2y}{dt^2} = mg - k \frac{dy}{dt},$$

donde  $k$  es una constante de proporcionalidad (coeficiente de rozamiento).

Otra situación de importancia es el caso del movimiento de un resorte (sin considerar efectos de fricción) que está suspendido verticalmente en el cual la fuerza  $F$  es proporcional a la posición  $y(t)$ , es decir,

$$F = -ky,$$

donde  $k$  es una constante. Así se obtiene la ecuación diferencial ordinaria

$$m \frac{dy^2}{dt^2} + ky = 0.$$

Otros ejemplos de ecuaciones diferenciales son los siguientes:

- $\frac{dy}{dt} + y = t + 2$
- $\frac{dy^2}{dt^2} + y^2 = y$
- $\frac{dy^3}{dt^3} + \frac{dy}{dx} + y = 0$
- $\frac{d^2y}{dt^2} + \sin(y) = F_0 \cos(wt)$ , donde  $F_0, w$  son constantes.
- $\frac{dy}{dt} = ky(M - y)$ , donde  $k, M$  son constantes.

**Solución de una EDO:** Es una función  $y(t)$  que al ser sustituida en la ecuación, la convierte en una identidad.

### Ejemplos:

- La función  $y(t) = Ce^{-kt}$ , siendo  $C$  cualquier constante, es una solución de la ecuación diferencial

$$\frac{dy}{dt} = -ky.$$

- La familia de funciones  $y(t) = c_1 \cos(\sqrt{k}t) + c_2 \sin(\sqrt{k}t)$ , para cualesquier constantes  $c_1, c_2$  son soluciones de la ecuación diferencial

$$\frac{dy^2}{dt^2} + kx = 0.$$

- La función  $y(t) = -\frac{1}{2}gt^2$  es una solución de la ecuación diferencial

$$\frac{d^2y}{dt^2} = -g.$$

**Nota:** Compruebe que las funciones de los ejemplos pasados son efectivamente solución de las respectivas ecuaciones diferenciales.

**Problemas de valor inicial:** A menudo nos interesa resolver una ecuación diferencial sujeta a condiciones adicionales que se imponen a la función  $y(t)$  o a sus derivadas. El problema es resolver

$$\frac{dy^n}{dt^n} = f(t, y, y', y'', \dots, y^{(n-1)}),$$

sujeta a

$$y(t_0) = y_0, y'(t_0) = y_1, \dots, y^{(n-1)}(t_0) = y_{n-1},$$

en donde  $y_0, y_1, \dots, y_{n-1}$  son constantes reales conocidas, se llama un **problema de valor inicial**. Los valores dados de la función desconocida  $y(t)$ , y de sus  $n - 1$  derivadas en un mismo punto  $t_0$ , se llaman condiciones iniciales.

#### 4.2.1. Estudio de modelos basados en ecuaciones diferenciales

Suponga que  $P_t$  es una sucesión que representa el tamaño de una población en cada tiempo  $t$ . Un modelo que vimos antes en la sección 4.1 para este tipo de situación es de la forma

$$\frac{P_{t+\Delta t} - P_t}{\Delta t} = kP_t,$$

donde  $\Delta t = 1$  y  $k$  es una constante real.

Vemos que si hacemos mediciones del tamaño de la población con mayor frecuencia ( lo cual corresponde a tomar  $\Delta t$  cada vez más pequeño), llegamos a

$$\lim_{\Delta t \rightarrow 0} \frac{P_{t+\Delta t} - P_t}{\Delta t} = kP_t,$$

es decir, la ecuación diferencial

$$\frac{dP}{dt} = kP. \quad (4.16)$$

Aquí  $P = P(t)$  es una función continua de la variable tiempo  $t$  que describe el tamaño de la población en cada tiempo  $t$ .

Se observa de lo anterior la estrecha relación existente entre el operador de diferencias finitas

$$\Delta P_t = P_{t+\Delta t} - P_t$$

y la derivada  $\frac{dP}{dt}$ . La ecuación 4.16 es la formulación matemática correspondiente al llamado **modelo exponencial**.

De manera semejante, siendo de nuevo  $P(t) = P_t$  la función que indica el tamaño de una población dada para valores continuos del tiempo  $t$ , se puede llegar a partir de la ecuación en diferencias correspondiente al modelo logístico

$$P_{t+1} = kP_t - \beta P_t^2,$$

con  $k, \beta$  constantes reales, a la formulación continua de este modelo el cual se expresa mediante la ecuación diferencial

$$\frac{dP}{dt} = kP - \beta P^2. \quad (4.17)$$

Podemos encontrar analíticamente las soluciones de la ecuación diferencial 4.16 a través del llamado **proceso de separación de variables**, en el cual en primer lugar se separan las variables de la ecuación en cada lado

$$\frac{dP}{P} = k dt.$$

En segundo lugar, se integran ambos lados de la ecuación anterior obteniendo

$$\int_{P_0}^{P(t)} \frac{dP}{P} = \int_0^t k dt.$$

Así, obtenemos para valores positivos de  $P(t)$  y  $P_0$ ,

$$\ln(P(t)) - \ln(P_0) = kt, \quad (4.18)$$

Así obtenemos aplicando la función exponencial en ambos lados

$$P(t) = P_0 e^{kt},$$

Lo anterior implica que la población  $P(t)$  descrita por la ecuación diferencial 4.16 presenta un crecimiento de tipo exponencial cuando  $k > 0$  y decrecimiento exponencial cuando  $k < 0$ .

Considere los datos de la tabla 4.1, que ya habíamos estudiado en la sección 4.1, acerca la población  $P_t$  de un tipo de levadura. Ahora en lugar de considerar que el tiempo  $t$  toma valores discretos, supondremos que la cantidad de biomasa de levadura  $P(t) = P_t$  es una función continua de la variable temporal  $t$  que toma todos los valores reales positivos. Investigaremos la predicción dada por la formulación matemática continua 4.16 para  $P(t)$  en este caso. Debemos calcular de nuevo un estimativo para el parámetro  $k$  y la constante  $P_0$ , para lo cual como nos sugiere la ecuación 4.18, consideraremos analizar los datos  $\ln(P(t))$  contra el tiempo en horas  $t$ , como se muestra en la Figura 4.16.

De acuerdo a esta gráfica, se observa que el comportamiento del logaritmo de la cantidad de biomasa  $\ln P(t)$  en términos del tiempo es aproximadamente lineal. Realizando el correspondiente proceso de regresión lineal (ver capítulo 3) a los datos experimentales, obtenemos que esta recta tiene pendiente  $\approx 0,463631$  y corte con el eje vertical  $\approx 2,39568$ . Así, de acuerdo con la ecuación 4.18, se obtiene que  $k \approx 0,463631$  y  $P_0 \approx 10,97566$ . Tenemos entonces la siguiente predicción para el crecimiento de  $P(t)$  cuando  $t$  crece:

$$\ln(P(t)) = 0,463631t + 2,39568,$$

o equivalentemente,

$$P(t) = 10,97566 e^{0,463631t}. \quad (4.19)$$

La validación correspondiente del modelo se presenta en la Figura 4.17.

Vemos que el ajuste entre los resultados provenientes de la ecuación 4.19 y los datos experimentales es muy bueno en el corto intervalo de tiempo considerado. Sin embargo, si hacemos como en la sección 4.1, la validación en un tiempo mayor de tiempo usando los datos de la tabla extendida 4.2, vemos una discrepancia significativa entre los resultados experimentales y los previstos por la ecuación 4.19. Ver la Figura 4.18, donde se aprecia que los datos experimentales ponen en evidencia la existencia de cierto factor limitante en el crecimiento de  $P(t)$  que no se está teniendo en cuenta en la ecuación diferencial 4.16.

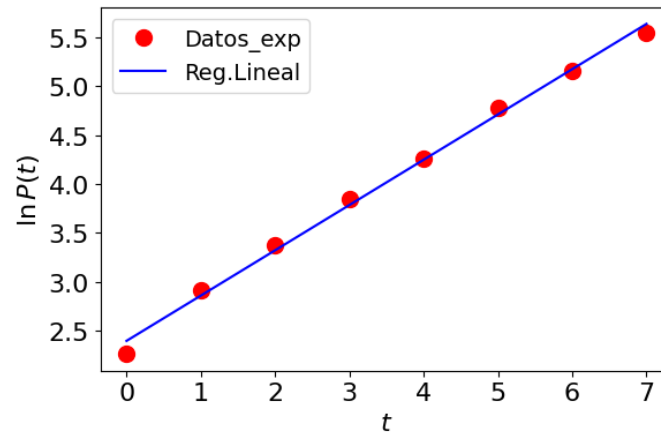


Figura 4.16:  $\ln(P(t))$  en función del tiempo  $t$  en horas y recta de regresión lineal.

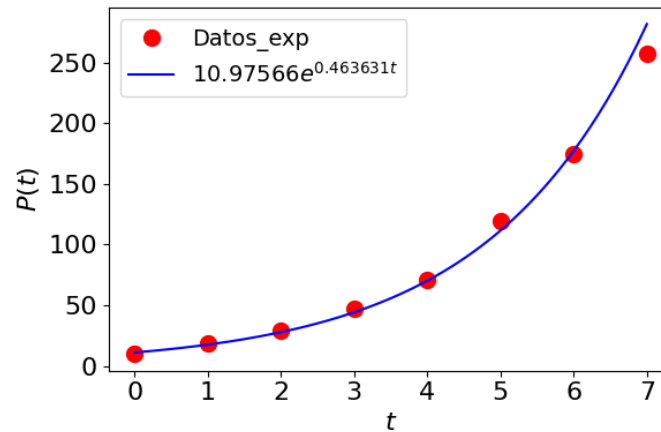


Figura 4.17: Línea continua:  $P(t)$  en función del tiempo  $t$  en horas de acuerdo a la ecuación 4.19. Puntos: datos experimentales tomados de la tabla 4.1. Se observa buen ajuste de la predicción del modelo exponencial con los datos experimentales.

### 4.2.2. Ejemplos de sistemas que se modelan con ecuaciones diferenciales

Las ecuaciones diferenciales permiten modelar sistemas que:

- evolucionan con el tiempo (modelos dinámicos),
- implican una razón de cambio en una o más variables.

En esta sección, nos enfocaremos en la forma en la que surgen las ecuaciones diferenciales dada una situación específica.

#### 1. Crecimiento y decrecimiento.

- a) **Crecimiento demográfico:** Donde la tasa de crecimiento de una población ( $\frac{dP}{dt}$ ) es proporcional a la población total,  $P(t)$ , en cualquier instante de tiempo,

$$\frac{dP}{dt} \propto P.$$

Esto quiere decir que, mientras más personas hayan en el tiempo  $t$ , más personas habrán en el futuro. Dicha proporcionalidad se puede escribir mediante una igualdad,

$$\frac{dP}{dt} = kP,$$

siendo  $k$  la constante de proporcionalidad, la cual depende de problema particular que se esté abordando. El sistema tiene una cierta población conocida en un instante de tiempo  $t = t_0$ , conocida como la condición inicial  $P(t_0) = P_0$ .

- b) **Procesos de Nacimiento-Muerte:** Donde la tasa de crecimiento del número de individuos de una población  $P$  depende del balance entre nacimientos ( $N$ ) y muertes ( $M$ ),

$$\frac{dP}{dt} = NP - MP = rP,$$

donde  $r = N - M$  es la tasa intrínseca de crecimiento, la cual cumple las siguientes condiciones:

- si  $r > 0 \Rightarrow \frac{dP}{dt} > 0$  : la población aumenta,
- si  $r < 0 \Rightarrow \frac{dP}{dt} < 0$  : la población disminuye,
- si  $r = 0 \Rightarrow \frac{dP}{dt} = 0$  : la población permanece constante.

- c) **Decaimiento Radiactivo:** Donde ciertos elementos (denominados radiactivos) son inestables, y en un intervalo de tiempo dado una fracción fija de los átomos se desintegra espontáneamente formando así un nuevo elemento. En este caso la tasa de crecimiento tiene una proporcionalidad negativa con la población total,

$$\frac{dN}{dt} = -\lambda N,$$

siendo  $\lambda$  la constante de decaimiento, la cual define el tiempo de vida medio del elemento ( $\tau$ ). Sea  $N_0 = N(t_0)$  la condición inicial, la solución a la ecuación diferencial es  $N(t) = N_0 \exp(-\lambda(t - t_0))$ . Definiendo a  $\tau$  como el tiempo necesario para

que se desintegre la mitad de los átomos tenemos que  $N(\tau) = \frac{1}{2}N_0$ , lo cual nos lleva a revisar la razón entre la cantidad de átomos a tiempo  $\tau$  y la cantidad inicial, de forma que:

$$\begin{aligned}\frac{N(\tau)}{N_0} &= \frac{1}{2} \\ &= \exp(-\lambda(\tau - t_0)) \\ -\lambda(\tau - t_0) &= \ln\left(\frac{1}{2}\right) \\ \tau &= t_0 + \frac{\ln(2)}{\lambda},\end{aligned}$$

obteniendo así el tiempo de vida medio del elemento como función de la constante de decaimiento  $\lambda$ .

## 2. Modelo logístico.

- a) **Propagación de una enfermedad contagiosa:** Donde el objetivo es modelar la propagación de una enfermedad contagiosa a través de una comunidad de personas donde se encuentra una persona enferma ( $E$ ) que tiene contacto con las personas sanas ( $S$ ). Consideremos una comunidad con una población fija de  $N$  personas. Entonces  $N = E + S$ . La razón con la que se propaga la enfermedad (la razón a la que crece la población de enfermos), es proporcional al número de encuentros o interacciones entre sanos y enfermos. Los encuentros son conjuntamente proporcional a las poblaciones  $E(t)$  y  $S(t)$ ,

$$\frac{dE}{dt} \propto ES \Rightarrow \frac{dE}{dt} = cES,$$

siendo  $c$  la constantes de proporcionalidad. De la conservación del número de personas (población fija), despejamos el número de personas sanas  $S = N - E$ , con lo que la rata de crecimiento de enfermos es ahora de la forma,

$$\begin{aligned}\frac{dE}{dt} &= cE(N - E) \\ &= \tilde{c}E\left(1 - \frac{E}{N}\right),\end{aligned}$$

donde hemos redefinido la constante de proporcionalidad  $\tilde{c} = cN$ .

- b) **Crecimiento limitado:** Donde el tamaño de la población depende de su entorno y los recursos limitados a los que están sujetos los individuos. El análisis de este problema lo basaremos en dos hipótesis:
- Si la población es pequeña, la tasa de crecimiento es proporcional a su tamaño  $\frac{dP}{dt} \propto P \Rightarrow \frac{dP}{dt} = kP$ , donde  $k > 0$  si  $P < \tilde{P}$ , donde  $\tilde{P}$  es la capacidad de soporte del entorno.
  - Si la población es demasiado grande para ser soportada por su entorno y recursos, la población disminuirá y la tasa de crecimiento pasa a ser negativa  $\frac{dP}{dt} < 0$ , cuando  $P(t) > \tilde{P}$ .



Deseamos un modelo que involucre ambas hipótesis. Supongamos

$$\frac{dP}{dt} = k\alpha P,$$

donde  $\alpha$  es una función que satisface las hipótesis:

- Para la primera hipótesis  $\alpha \rightarrow 1$  cuando  $P$  es pequeña.
- Para la segunda hipótesis  $\alpha \rightarrow 0$  cuando  $P \rightarrow \tilde{P}$ .

Entonces, la expresión más simple que satisface estas dos situaciones es  $\alpha = 1 - \frac{P}{\tilde{P}}$ . Tal que el modelo que se ajusta a los requerimientos del problema es:

$$\frac{dP}{dt} = k \left( 1 - \frac{P}{\tilde{P}} \right) P,$$

donde  $k$  es la velocidad de crecimiento y  $\tilde{P}$  la capacidad de soporte.

- c) **Ley de enfriamiento de Newton:** La cual establece que la razón de cambio de la temperatura de un cuerpo con respecto al tiempo  $\frac{dT}{dt}$  es proporcional a la diferencia entre la temperatura del cuerpo  $T(t)$  y la temperatura del medio ambiente  $T_m$ :

$$\frac{dT}{dt} \propto (T - T_m) \Rightarrow \frac{dT}{dt} = -k(T - T_m),$$

con  $k \geq 0$  es la constante de proporcionalidad, y  $T_m$  se considera constante.

- Si  $T < T_m \Rightarrow \frac{dT}{dt} > 0$ : el cuerpo se está enfriando.
- Si  $T > T_m \Rightarrow \frac{dT}{dt} < 0$ : el cuerpo se está calentando.

### 3. Depredador-Presa

Consideremos dos especies: depredadores (zorros,  $Z(t)$ ) y presas (conejos,  $C(t)$ ), ambas poblaciones funciones del tiempo. El modelo debe describir el aumento o disminución de ambas poblaciones de acuerdo a las interacciones entre las especies, si hay o no alimento, entre otros factores. Las hipótesis son:

- Si no hay zorros, los conejos se reproducen a una tasa proporcional a su población y no les afecta la sobrepoblación

$$\frac{dC}{dt} \propto C \Rightarrow \frac{dC}{dt} = pC$$

- Sin conejos que comer, los zorros mueren a una tasa proporcional a su población

$$\frac{dZ}{dt} \propto -Z \Rightarrow \frac{dZ}{dt} = -qZ$$

- Los zorros se comen a los conejos y la razón a la que los conejos son devorados es proporcional a la tasa o interacción zorro-conejo

$$\frac{dC}{dt} \propto -CZ \Rightarrow \frac{dC}{dt} = -rCZ$$

- La tasa de nacimiento de zorros crece en proporción al número de conejos comidos, que es proporcional a la interacción zorro-conejo

$$\frac{dZ}{dt} \propto CZ \Rightarrow \frac{dZ}{dt} = sCZ$$

Con  $p, q, r, s$  constantes positivas, siendo estas:

$p$ : la tasa de nacimiento de conejos.

$q$ : la tasa de muerte de zorros.

$r$ : la tasa de interacción conejo-zorro en la que los conejos son devorados.

$s$ : la tasa de interacción conejo-zorro en la que los zorros son beneficiados proliferandose.

Tal que el modelo está dado por las siguientes ecuaciones diferenciales acopladas:

$$\begin{aligned}\frac{dC}{dt} &= pC - rCZ \\ \frac{dZ}{dt} &= -qZ + sCZ,\end{aligned}$$

las tasas de cambio  $\frac{dC}{dt}$  y  $\frac{dZ}{dt}$  dependen conjuntamente de  $C$  y  $Z$ .

### 4.2.3. Métodos cualitativos y computacionales para EDOs

En esta sección, introducimos las técnicas cualitativas del diagrama de fase, el campo de direcciones y algunas rutinas de solución numérica para analizar las soluciones de ecuaciones diferenciales ordinarias.

Considere una ecuación diferencial de la forma

$$\frac{dy}{dt} = f(y, t) \tag{4.20}$$

En primer lugar, un tipo de soluciones importantes de este tipo de EDOs son soluciones constantes, es decir, las soluciones  $y(t) = c$  para las cuales

$$f(c) = 0.$$

Estas soluciones también se conocen como **puntos críticos o soluciones de equilibrio**.

Además, recuerde el criterio de la primera derivada el cual establece lo siguiente:

**Criterio de la primera derivada.** Si la primera derivada  $y'(t)$  en el punto  $t$  de la función  $y(t)$  es positiva (negativa), entonces la función  $y(t)$  es creciente (decreciente) en  $t$  (ver figura 4.19 ).

De esta manera vemos que sin proceder a realizar una solución analítica de una ecuación diferencial tal como 4.20, podemos proporcionar información de la forma geométrica de sus soluciones.

En cuanto a la curvatura de las soluciones, recuerde que esta noción está conectada con la segunda derivada de la función  $y(t)$ .

### Criterio de la segunda derivada.

$$\frac{d^2y}{dt^2} > 0, \quad (\text{la gráfica se curva hacia arriba})$$

$$\frac{d^2y}{dt^2} < 0, \quad (\text{la gráfica se curva hacia abajo}).$$

### Campo de direcciones

Otra herramienta gráfica que ayuda al entendimiento de las soluciones de una ecuación diferencial de la forma 4.20. Puesto que la función  $f(t, y)$  proporciona la pendiente de la recta tangente a la solución  $y(t)$  en cada punto  $t$ , podemos en cada punto  $(t, y)$  del plano dibujar el vector  $\vec{v} = (1, f(t, y))$ , el cual tiene ángulo de inclinación  $\theta$  dado por  $\tan(\theta) = f(t, y)$  (ver Figura 4.21).

Si este proceso se repite para una colección de puntos del plano  $ty$ , dibujando cada vector  $\vec{v}$  con la misma longitud obtenemos el llamado campo de direcciones de la ecuación diferencial. La longitud que se le ponga a cada vector  $\vec{v}$  puede ajustarse de acuerdo al espacio disponible entre los puntos del plano  $ty$  escogidos. Usualmente realizar manualmente este tipo de gráficos es laborioso y se suele recurrir mejor a un sistema de cómputo, tal como matplotlib para python, a fin de implementarlos.

**Ejemplo 1:** Considere una ecuación diferencial de primer orden en la forma

$$\frac{dy}{dt} = f(y) = 3y. \quad (4.21)$$

En este ejemplo vemos que la única solución constante (solución de equilibrio) de la EDO 4.21 es  $y = 0$ .

De acuerdo a lo que conocemos de los cursos de cálculo, si  $y(t)$  es una solución de la ecuación anterior, entonces la función  $f(y, t)$  está dando la pendiente de la recta tangente a  $y(t)$  en cada punto  $t$ . Debemos para esto analizar los signos de la función  $f(y) = 3y$  en  $-\infty < y < \infty$ . Vemos que tenemos las dos regiones  $y < 0$  y  $y > 0$  en los cuales se tiene

$$y'(t) = f(y) = 3y < 0, \quad \text{para } y < 0,$$

y

$$y'(t) = f(y) = 3y > 0, \quad \text{para } y > 0.$$

Colocando esta información en un plano donde la variable independiente  $t$  se coloca en la horizontal y la variable dependiente  $y$  va en el eje vertical, obtenemos la Figura 4.22 donde se aprecia el comportamiento cualitativo de las soluciones de la ecuación diferencial dada. La conclusión es que en la región  $y < 0$  todas las soluciones son decrecientes y en la región  $y > 0$  las

soluciones de la EDO son crecientes. También en este caso particular, observamos que todas las soluciones de la EDO se apartan de la solución de equilibrio  $y = 0$ . Este tipo de gráficas donde se esbozan algunas soluciones de la ecuación diferencial en el mismo plano se llaman **diagrama de fase** de la EDO.

En cuanto a la curvatura que se le ha colocado a las soluciones esbozadas en este gráfico, revisemos el criterio de la segunda derivada.

Así de la ecuación diferencial

$$\frac{dy}{dt} = 3y,$$

obtenemos derivando de nuevo con respecto a  $t$  que

$$\frac{d^2y}{dt^2} = 3\frac{dy}{dt} = 3(3y) = 9y.$$

Vemos que los signos de la segunda derivada  $\frac{d^2y}{dt^2}$  son iguales a los de la primera derivada  $\frac{dy}{dt}$ :

$$\frac{d^2y}{dt^2} > 0, \quad \text{si } y > 0, \quad (\text{la gráfica se curva hacia arriba})$$

$$\frac{d^2y}{dt^2} < 0, \quad \text{si } y < 0, \quad (\text{la gráfica se curva hacia abajo}).$$

Por esta razón tenemos la curvatura en las gráficas de las soluciones esbozadas en la Figura 4.22. Por otro lado, también se dispone de librerías en Python con las cuales es posible realizar la solución numérica de la ecuación diferencial dada. Una de ellas es `odeint()`, y un código sencillo que permite calcular la solución de la ecuación para una condición inicial  $y(0) = y_0$ , está dado en la siguiente forma:

```

1
2 # Importando librerías
3
4 import numpy as np
5 from scipy.integrate import odeint
6 import matplotlib.pyplot as plt
7 plt.style.use('seaborn-whitegrid')
8
9
10 # Rutina para evaluar la función f(y,t) que define la EDO
11 def model(y,t):
12     dydt = 3*y
13     return dydt
14
15
16 t = np.linspace(0,0.5) # Definición del vector de tiempos
17 y0 = 1 # Condición inicial
18 y = odeint(model, y0 , t ) # Solución numérica de la EDO
19
20
21 # Graficando resultados
22

```

```

23 fig = plt.figure()
24 ax = plt.axes()
25 ax.plot(t, y, 'b')
26 plt.xlabel('t')
27 plt.ylabel('y(t)')
28 plt.ylim(-10,10)
29 plt.show()

```

**Ejemplo 2:** Considere ahora la ecuación diferencial

$$\frac{dy}{dt} = f(y) = y - y^2.$$

En primer lugar, los puntos críticos de este sistema son aquellas soluciones constantes  $y(t) = c$ , tales que  $f(c) = c - c^2 = c(1 - c) = 0$ . Así,  $c = 0$  o  $c = 1$ . Por lo tanto, el sistema dado tiene como puntos (o soluciones) de equilibrio  $y(t) \equiv 0$ ,  $y(t) \equiv 1$ .

Como en el ejemplo anterior, podemos realizar un análisis cualitativo de la ecuación diferencial dada a fin de construir su diagrama de fase. Esto implica analizar las regiones donde la función  $f(y) = y - y^2$  toma signo positivo o negativo. En este caso tenemos que

$$f(y) = y - y^2 = y(1 - y) > 0 \text{ si } y > 0, y < 1 \text{ o si } y < 0, y > 1.$$

Observe que el último caso no se cumple. Esto significa que  $f(y) \geq 0$  en la región  $\{(x, y) : y \in (0, 1)\}$ .

Por otro lado

$$f(y) = y(1 - y) < 0 \text{ si } y < 0, y < 1 \text{ o si } y > 0, y > 1$$

Por lo tanto,  $f(y) < 0$  en la región  $\{(x, y) : y < 0\} \cup \{(x, y) : y > 1\}$ . Recordando el criterio de la primera derivada y el hecho de que la función  $f(y(t))$  nos da la pendiente de la recta tangente a  $y(t)$ , en el punto  $t$ , obtenemos una primera versión del diagrama de fase de las soluciones del sistema mostrada en la Figura 4.23.

Usando la misma información del campo  $f$  podemos incluso estudiar la curvatura de las soluciones  $y(t)$  de la ecuación. En efecto, a partir de  $\frac{dy}{dt} = f(y) = y(1 - y)$ , obtenemos usando la propiedad de la derivada de un producto.

$$\frac{d^2y}{dt^2} = \frac{dy}{dt}(1 - y) + y \left( -\frac{dy}{dt} \right).$$

Dando entonces que  $\frac{dy}{dt} = y(1 - y)$ , llegamos a  $\frac{d^2y}{dt^2} = y(1 - y)^2 + y(1 - y)$ ,

$$\begin{aligned} \frac{d^2y}{dt^2} &= y(1 - y)[1 - y - y] \\ &= y(1 - y)(1 - 2y). \end{aligned}$$

De lo anterior, vemos que para analizar el tipo de curvatura que presentan las soluciones de la ecuación diferencial dada, debemos estudiar el signo de  $\frac{d^2y}{dt^2} = y(1 - y)(1 - 2y)$ . Esto se

facilita a través del siguiente gráfico donde se indican los signos de cada factor que aparece en la segunda derivada, de lo cual se concluye el signo final del producto de los tres factores. Observe que es importante encontrar los puntos donde se anula cada factor, lo cual facilita determinar los signos correspondientes en cada región.

De la figura 3 obtenemos los siguientes resultados:

$$\begin{aligned} \frac{d^2y}{dt^2} > 0 & \text{ si } y \in \left[0, \frac{1}{2}\right) \cup [1, \infty) \\ \frac{d^2y}{dt^2} < 0 & \text{ si } y \in [-\infty, 0) \cup \left[\frac{1}{2}, 1\right). \end{aligned}$$

Descubrimos de los anteriores resultados que las soluciones de la ecuación diferencial estudiada presentan un cambio de curvatura cuando pasan por la línea  $y = 1/2$ . Con la información obtenida, podemos dibujar con mayor precisión en el diagrama de fase presentado en la Figura 4.25, el comportamiento cualitativo de las soluciones de la ecuación  $\frac{dy}{dt} = y - y^2$ , que ya habíamos esbozado en la Figura 4.23.

Para terminar, en las Figuras 4.26 y 4.27 se muestran los campos de direcciones obtenidos para las ecuaciones diferenciales  $\frac{dy}{dt} = 3y$  y  $\frac{dy}{dt} = y - y^2$ .

El campo de direcciones correspondiente a la ecuación  $\frac{dy}{dt} = 3y$ , se puede obtener mediante la siguiente rutina en Python:

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4
5
6  # Define rango del eje temporal
7  a = 0
8  b = 7
9  t = np.linspace(a,b)
10 t = np.arange(a, b + 0.5, .5)
11 y = np.arange(-1, 1.5, .2)
12 T, Y = np.meshgrid(t, y)
13
14
15 def f(t, y):
16     return 3*y
17
18 dYdT = f(T, Y)
19 U = (1/(1 + dYdT**2)**.4)*np.ones(T.shape)
20 V = (1/(1 + dYdT**2)**.4)*dYdT
21
22 # Ajusta dimensiones del grafico
23 fig = plt.figure(figsize = (10,8) )
24
25 # Realiza el grafico
26 plt.quiver(T, Y, U, V) # Grafica campo de direcciones
27 plt.plot( [0, 7], [0, 0] , 'k--') # Grafica solucion de equilibrio y = 0
28 plt.show()

```

#### 4.2.4. Algunos Métodos Numéricos para resolver EDOs

En muchas ocasiones no es posible obtener soluciones exactas o analíticas para una ecuación diferencial. Una herramienta fundamental en su estudio, además de las analizadas hasta ahora, son los métodos numéricos, los cuales consisten básicamente en la transformación ( **discretización** ) de la ecuación diferencial (en tiempo continuo) en una ecuación en diferencias finitas (en tiempo discreto).

En particular estudiaremos los esquemas de discretización de Euler y de Adams, aplicados para aproximar las soluciones de problemas de valor inicial asociados a ecuaciones diferenciales de primer orden en la forma

$$\begin{aligned}\frac{dy}{dt} &= f(t, y(t)), \\ y(t_0) &= y_0,\end{aligned}\tag{4.22}$$

donde  $f$  es una función dada,  $t_0$  es un valor real y  $y_0$  es el valor de la función  $y(t)$  en el punto  $t_0$ .

##### Esquema de discretización de Euler

Supongamos que  $y(t)$  es una solución del problema de valor inicial 4.22, definida para todo valor de  $t$  en nuestro intervalo de trabajo que será  $[t_0, T]$ . Considere que dicho intervalo está dividido en subintervalos  $[t_i, t_{i+1}]$ , de manera que  $t_{i+1} = t_i + h$ , siendo  $h$  la longitud del subintervalo (ver Figura 4.28). El parámetro  $h$  también se llama **tamaño de paso**.

Sabemos que,

$$y'(t_i) = \lim_{h \rightarrow 0} \frac{y(t_i + h) - y(t_i)}{h},$$

tenemos para  $h$  pequeño la aproximación

$$y'(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{h},$$

es decir,

$$y(t_{i+1}) \approx y(t_i) + hy'(t_i).$$

Ahora, utilizando la ecuación diferencial

$$y'(t_i) = \left. \frac{dy(t)}{dt} \right|_{t_i} = f(t_i, y(t_i)),$$

obtenemos

$$y(t_{i+1}) \approx y(t_i) + hf(t_i, y(t_i)).$$

Denotando por  $\eta_i$  la aproximación de la solución del problema 4.22 evaluada en el punto  $t_i$ , tenemos el esquema

$$\eta_{i+1} = \eta_i + hf(t_i, \eta_i).\tag{4.23}$$

La aproximación dada a través de la ecuación en diferencias de primer orden 4.23, se conoce como el **método de Euler** para aproximar soluciones de ecuaciones diferenciales de primer orden. Un diagrama de flujo para el esquema de discretización de Euler se encuentra en la Figura 4.29. Dado que se conoce  $\eta_0 = y_0$ ,  $\eta_1$  se calcula a partir de  $\eta_0$  y se convierte en un valor conocido a partir del cual se puede calcular  $\eta_2$ , y así sucesivamente.

**Ejemplo:**

Considere el problema de valor inicial

$$\begin{aligned}\frac{dy}{dt} &= 1 + (y - t)^2 = f(t, y) \\ y(0) &= 0,5\end{aligned}$$

el cual tiene solución analítica (exacta)  $y(t) = t + 1/(2 - t)$ . Compare el comportamiento de la solución numérica proporcionada por el esquema de discretización de Euler con la solución analítica en el intervalo  $[0, 1,9]$ .

En este ejemplo, el esquema de Euler 4.23 para aproximar las soluciones del problema se puede escribir como la ecuación en diferencias de primer orden

$$\begin{aligned}\eta_0 &= 0,5 \\ \eta_{i+1} &= \eta_i + h(1 + (\eta_i - t_i)^2), \quad i = 0, 1, 2, \dots\end{aligned}$$

La implementación en python para el esquema de Euler se puede escribir como sigue:

```

1 # Implementa el metodo de Euler para aproximar una solucion de una
2 # ecuacion diferencial
3
4 import numpy as np
5 import matplotlib.pyplot as plt
6
7 def F(t, y):
8     return 1 + (y-t)**2
9
10 def exacta (t):
11     return t + 1/(2 - t)
12
13 T= 1.9
14 N = 50
15 t = np.linspace(0, T, N+1)
16 h = t[2] - t[1]
17 print(h) # Imprime el tamaño de paso usado
18 y = np.empty(N+1)
19
20 y[0] = 0.5 # Define la condicion inicial
21
22 for i in range(0, N ):
23     y[i+1] = y[i] + h*F(t[i],y[i])
24
25 fig = plt.figure( figsize =(15 ,8) )
26 plt.plot(t, y, '--k')
27 plt.plot(t, exacta(t), '.k' )
28 plt.xlabel(r'$t$', size = 16)
29 plt.text(0, 12, r'$y(t)$', size = 16)
30 plt.grid()
31 plt.show()

```

En la Figura 4.30 se puede ver el resultado de la simulación numérica usando el método de Euler junto con la solución exacta para el caso en que  $N = 50$ ,  $T = 1,9$ , lo cual quiere decir que  $h \approx 0,038$ . Observe que hay discrepancia entre las dos perfiles apreciable cerca del tiempo



final  $T$ . En las Figuras 4.31 y 4.31, obtenidas usando  $N = 200$  ( $h = 0,0095$ ) y  $N = 500$  ( $h = 0,0038$ ) se aprecia que la solución numérica se ajusta más a la solución exacta conforme el número de puntos  $N$  aumenta (equivalentemente cuando el tamaño de paso  $h$  disminuye).

### Esquema de discretización de Adams

Usamos la misma notación de la sección anterior. En primer lugar, puede mostrarse que si  $h$  es pequeño, tenemos la siguiente aproximación

$$y(t_{i+1}) \approx y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i). \quad (4.24)$$

Ahora utilizando de nuevo la ecuación diferencial

$$y'(t_i) = \left. \frac{dy(t)}{dt} \right|_{t_i} = f(t_i, y(t_i)),$$

y la aproximación para  $h$  pequeño

$$y''(t_i) \approx \frac{f(t_i, y(t_i)) - f(t_{i-1}, y(t_{i-1}))}{h},$$

y sustituyendo en la ecuación 4.24, obtenemos

$$y(t_{i+1}) \approx y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} \left( \frac{f(t_i, y(t_i)) - f(t_{i-1}, y(t_{i-1}))}{h} \right).$$

Simplificando esta última expresión, se obtiene

$$y(t_{i+1}) \approx y(t_i) + \frac{h}{2}(3f(t_i, y(t_i)) - f(t_{i-1}, y(t_{i-1}))).$$

Denotando de nuevo por  $\eta_i$  a la aproximación de  $y(t_i)$ , tenemos el **esquema de discretización de Adams**

$$\eta_{i+1} = \eta_i + \frac{h}{2}(3f(t_i, \eta_i) - f(t_{i-1}, \eta_{i-1})). \quad (4.25)$$

Observe que a diferencia del método de Euler, se requieren conocer dos valores iniciales  $\eta_0$  y  $\eta_1$  a fin poder calcular usando la ecuación en diferencias de orden 2 dada en 4.25. Para calcular  $\eta_1$  se puede usar por ejemplo el método de Euler. El diagrama de flujo para el esquema de Adams se presenta en la Figura 4.33.

#### 4.2.5. Aplicaciones

En esta sección presentaremos algunas situaciones en las que se requieren modelos dinámicos basados en ecuaciones de diferencias y diferenciales para estudiar los fenómenos considerados.

##### El sistema masa – resorte

El sistema masa–resorte permite ilustrar el uso del modelamiento para predecir el comportamiento de un sistema físico descrito por la segunda ley de Newton. Aunque el sistema descrito aquí involucra una ecuación de movimiento con solución analítica, mostraremos una manera general de resolver las ecuaciones de movimiento mediante un método numérico. Esto con

el fin de que el estudiante pueda explorar el comportamiento de sistemas mecánicos cuyas ecuaciones de movimiento no permitan soluciones analíticas.

*La ecuación de Newton de movimiento*

Consideremos la solución numérica de un sistema mecánico general compuesto por una sola partícula de masa  $m$  con movimiento en una dimensión. La ecuación de Newton de movimiento de este sistema es

$$f = m a(t), \quad (4.26)$$

donde  $f$  es la fuerza que se ejerce sobre la partícula y  $a(t)$  es la aceleración de la partícula en un tiempo  $t$ . La ecuación de movimiento 4.26 se puede escribir también como el sistema de ecuaciones diferenciales acopladas

$$\begin{aligned} v(t) &= \frac{dx}{dt} \\ f &= m \frac{dv}{dt}, \end{aligned} \quad (4.27)$$

donde  $x(t)$  y  $v(t)$  son la posición y la velocidad de la partícula en un tiempo  $t$  dado, respectivamente. Como vimos en la sección anterior, un método simple que se puede usar para aproximar las soluciones del sistema 4.27 es el **método de Euler**, el cual se basa en la siguiente aproximación de la derivada

$$\frac{dg(t)}{dt} \approx \frac{g(t+h) - g(t)}{h},$$

en un punto  $t$ , dado que  $h$  es una cantidad pequeña. Aplicando esta idea a cada una de las ecuaciones del sistema 4.27 obtenemos las siguientes aproximaciones para  $h$  pequeño:

$$\begin{aligned} x(t+h) &\approx x(t) + hv(t) \\ v(t+h) &\approx v(t) + h \frac{f}{m}. \end{aligned} \quad (4.28)$$

El esquema de discretización de Euler presenta **inestabilidad numérica** en el sentido de que conlleva el crecimiento de errores de redondeo y de pequeñas fluctuaciones de las condiciones iniciales que pueden ocasionar desviaciones apreciables de la solución exacta. En su lugar, utilizaremos el esquema de **Euler-Cromer** (llamado también método de **Euler semi-implícito**) que se escribe como

$$\begin{aligned} v(t+h) &\approx v(t) + h \frac{f}{m} \\ x(t+h) &\approx x(t) + hv(t+h). \end{aligned} \quad (4.29)$$

Aunque a simple vista ambos esquemas parecen iguales, se debe notar que el esquema Euler-Cromer requiere el cálculo de la velocidad a un tiempo  $t+h$  para poder calcular la posición a dicho tiempo. Para valores específicos de  $h$ , de la posición y de la velocidad a un tiempo inicial  $t_0$ ,  $x(t_0)$  y  $v(t_0)$ , el esquema de Euler-Cromer da como resultado los valores  $x(t_0+h)$  y  $v(t_0+h)$  en el tiempo  $t_0+h$ . Estos últimos valores insertados en el esquema darán los nuevos valores  $x(t_0+2h)$  y  $v(t_0+2h)$  en el tiempo  $t_0+2h$ , los cuales a su vez darán como resultado los valores de  $x$  y  $v$  en el tiempo  $t_0+3h$ , y así sucesivamente.

En resumen, para poder predecir el comportamiento de una partícula de masa  $m$  con movimiento unidimensional gobernado por la segunda ley de Newton, se requiere conocer la posición y la velocidad de la partícula. También se requiere conocer la fuerza que se ejerce sobre la partícula, la cual puede depender explícitamente de la posición, de la velocidad y/o del tiempo.

*La ecuación de Newton de movimiento del sistema masa-resorte*

Sea un resorte de masa despreciable con un cuerpo de masa  $m$  atado a su extremo derecho e inicialmente en su posición de equilibrio (es decir en reposo), definida como  $x = 0$ . Si el cuerpo atado al resorte se desplaza hasta cierta posición  $x$  a la derecha de su posición de equilibrio, sin alcanzar su límite elástico (desplazamiento  $x$  de estiramiento que provoca una deformación permanente del resorte), la fuerza de restitución del resorte, que es opuesta a su desplazamiento, para un pequeño desplazamiento  $x$  es

$$f(x) = -kx, \quad (4.30)$$

donde  $k$  es llamada constante de fuerza, la cual mide la capacidad del resorte para resistirse a ser deformado por una fuerza aplicada. A esta fuerza se le conoce como ley de Hooke.

Si el cuerpo se libera, comenzará a oscilar alrededor de la posición de equilibrio del sistema masa-resorte. Si asumimos que no hay fricción del aire la ecuación de Newton de movimiento para este sistema será

$$f(x) = m \frac{d^2x(t)}{dt^2} = -kx(t). \quad (4.31)$$

Debido al comportamiento oscilatorio de este sistema, la constante de fuerza se suele expresar como

$$k = m\omega^2, \quad (4.32)$$

donde  $\omega$  es la frecuencia angular de oscilación del sistema masa-resorte. La ecuación de movimiento queda como

$$\frac{d^2x(t)}{dt^2} = -\omega^2x(t), \quad (4.33)$$

o

$$\frac{d^2x(t)}{dt^2} + \omega^2x(t) = 0, \quad (4.34)$$

la cual es una ecuación diferencial de segundo orden. Esta ecuación tiene como solución

$$x(t) = C_1 \cos \omega t + C_2 \sin \omega t, \quad (4.35)$$

donde  $C_1$  y  $C_2$  son números reales. La velocidad es entonces

$$v(t) = \frac{dx(t)}{dt} = -\omega C_1 \sin \omega t + \omega C_2 \cos \omega t. \quad (4.36)$$

El comportamiento particular del sistema masa-resorte se determina al establecer valores específicos de la posición y la velocidad a un tiempo inicial  $t_o = 0$ , llamadas *condiciones iniciales*. Por ejemplo,

$$x(0) = x_R, \quad v(0) = 0, \quad (4.37)$$

con  $x_R$  como un número positivo. Para este caso particular, las constantes  $C_1$  y  $C_2$  son

$$C_1 = x(0) = x_R \quad (4.38)$$

$$C_2 = \frac{v(0)}{\omega} = 0. \quad (4.39)$$

Por lo tanto, el comportamiento del sistema a cualquier tiempo  $t$  está dado por las siguientes expresiones matemáticas:

$$x(t) = x_R \cos \omega t \quad (4.40)$$

$$v(t) = -\omega x_R \sin \omega t. \quad (4.41)$$

Estas expresiones muestran que el desplazamiento  $x$  del sistema es un movimiento oscilatorio de amplitud máxima constante, con desplazamiento mínimo y máximo de  $-x_R$  y  $+x_R$ , respectivamente y frecuencia lineal  $\nu = \frac{\omega}{2\pi}$ .

Utilicemos ahora el algoritmo de Euler-Cromer para predecir el comportamiento del sistema masa-resorte

$$v(t+h) = v(t) - k \frac{h}{m} x(t) \quad (4.42)$$

$$x(t+h) = x(t) + hv(t+h). \quad (4.43)$$

Las condiciones iniciales a  $t_0 = 0$  son

$$x(0) = 1,0 \text{ m}, \quad v(0) = 0,0 \text{ m/s}. \quad (4.44)$$

Por simplicidad, utilizaremos el valor de la masa y de la constante de fuerza como iguales a 1.0

$$m = 1,0 \text{ kg}, \quad k = 1,0 \text{ N/m}^2. \quad (4.45)$$

El algoritmo de Euler-Cromer será entonces

$$v(t+h) = v(t) - hx(t) \quad (4.46)$$

$$x(t+h) = x(t) + hv(t+h). \quad (4.47)$$

La implementación del método de Euler-Cromer en python es la siguiente:

```
1
2     import numpy as np
3     import matplotlib.pyplot as plt
4
5     # Parametros del sistema
6     k = 1.0           # Constante del resorte
7     m = 1.0           # Masa del cuerpo
8     w = np.sqrt(k/m) # Frecuencia de oscilacion
```

```

9
10     # Parametros del metodo
11     h = 0.1 # Tamano del paso
12     t = np.arange(0, 10 + h, h) # Malla
13
14     # Condiciones iniciales
15     x0 = 1.0 # Desplazamiento inicial
16     v0 = 0.0 # Velocidad inicial
17
18     # Metodo de Euler-Cromer
19     x = np.zeros(len(t))
20     v = np.zeros(len(t))
21
22     x[0] = x0
23     v[0] = v0
24
25     for i in range(0, len(t) - 1):
26         v[i+1] = v[i] - h*x[i]
27         x[i+1] = x[i] + h*v[i+1]
28
29     #Graficas
30     plt.figure(figsize = (12, 8))
31     plt.plot(t, x, 'ko', label='desplaz. (Euler-Cromer)')
32     plt.plot(t, v, 'k*', label='vel. (Euler-Cromer)')
33     plt.plot(t, x0*np.cos(w*t), 'k--', label='desplaz exacto')
34     plt.plot(t, -x0*w*np.sin(w*t), 'k', label='vel. exacta')
35     #plt.title('Resultado usando el metodo de Euler-Cromer')
36     plt.xlabel('Tiempo/s')
37     #plt.ylabel('Desplazamiento/m')
38     plt.grid()
39     plt.legend(loc='lower right')
40     plt.show()
41

```

En la tabla 4.5 se muestran los resultados numéricos del desplazamiento y la velocidad si se utiliza el paso de tiempo  $h = 0,1$  s y se comparan con los resultados exactos dados por la solución analítica de la ecuación diferencial correspondiente. En la tabla se muestran los resultados con tiempo de cada 0,2s y con 3 cifras decimales.

Los resultados numéricos de la tabla 4.5 muestra también una concordancia en la primera cifra decimal con los resultados exactos. Entre más pequeño sea el valor del paso de tiempo  $h$ , la concordancia será mayor pero el tiempo de cómputo será mayor también. Es posible utilizar algoritmos que permiten obtener valores más precisos con el mismo paso del tiempo, lo cual será explorado en uno de los ejercicios de esta sección.

Los resultados pueden visualizarse en el gráfico 4.34 de tiempo versus desplazamiento/velocidad, en el que se observa el carácter oscilatorio del movimiento del sistema masa-resorte.

$t$ (s)	$v_{num}(t)$ (m/s)	$x_{num}(t)$ (m)	$v_{exacta}(t)$ (m/s)	$x_{exacta}(t)$ (m)
0.0	0.000	1.000	0.000	1.000
0.2	-0.199	0.970	-0.199	0.980
0.4	-0.390	0.901	-0.389	0.921
0.6	-0.566	0.797	-0.565	0.825
0.8	-0.718	0.661	-0.717	0.697
1.0	-0.843	0.498	-0.841	0.540
1.2	-0.933	0.315	-0.932	0.362
1.4	-0.987	0.120	-0.985	0.170
1.6	-1.001	-0.080	-1.000	-0.029
1.8	-0.975	-0.277	-0.974	-0.227
2.0	-0.910	-0.462	-0.909	-0.416
2.2	-0.809	-0.630	-0.808	-0.589
2.4	-0.676	-0.722	-0.675	-0.737
2.6	-0.515	-0.883	-0.516	-0.587
2.8	-0.334	-0.959	-0.335	-0.942
3.0	-0.140	-0.997	-0.141	-0.990

Cuadro 4.5: Resultado numérico del sistema masa-resorte obtenido mediante el algoritmo de Euler-Cromer y su comparación con el resultado exacto. Por simplicidad, los valores de la masa y la constante de fuerza son  $m = 1,0$  kg y  $k = 1,0$  N/m<sup>2</sup>. Las condiciones iniciales a  $t = 0$  son  $x(0) = 1,0$  m y  $v(0) = 0,0$  m/s. El tamaño del paso de tiempo es  $h = 0,1$  s.

### 4.3. Ejercicios

1. Modifique el programa en Python dado en el ejemplo 3 de la sección 4.1 para calcular y graficar las siguientes sucesiones ( $y_n$ ) definidas por las siguientes ecuaciones en diferencias:

- $y_{n+1} = ky_n$ , donde  $y_0 = 2$ . Analice los casos  $k = 0,5$ ,  $k = 1,5$  y usando los resultados compare el comportamiento de las sucesiones obtenidas. Qué influencia tiene el parámetro  $k$  en el comportamiento de la sucesión ?
- $y_{n+1} = ky_n + b$ , donde  $y_0 = 3$ . De nuevo analice la influencia de los parámetros  $k, b$  en el comportamiento de la sucesión. Qué diferencia existe si  $|k| < 1$  o si  $|k| \geq 1$  ? Cómo influye el valor del signo de  $k$  ? El valor de  $b$  afecta el comportamiento de la sucesión cuando  $n$  crece ?
- $y_{n+2} - 5y_{n+1} + 6y_n = 0$ . En este caso, a diferencia de los anteriores ejemplos, vemos que se requieren dos valores iniciales, digamos  $y_0, y_1$  para calcular los términos siguientes de la sucesión de acuerdo a la ley de recurrencia dada. Suponga aquí que  $y_0 = 2, y_1 = 1$ . Analice el comportamiento de la sucesión cuando  $n$  crece.
- Analice las sucesiones siguientes utilizando las herramientas gráficas y computacionales consideradas en los ejemplos anteriores:

$$y_{n+1} = y_n^3$$

$$y_{n+1} = -y_n^2$$

$$y_{n+1} = y_n - y_n^2$$

$$y_{n+1} = \sin(y_n).$$

Considere diferentes valores iniciales  $y_0$  y obtenga algunas conclusiones de los resultados obtenidos.

2. Analice el modelo en el ejemplo 6 de la sección 4.1 usando las herramientas gráficas y computacionales introducidas en los ejemplos anteriores.
3. Suponga que inicialmente se tienen 30 codornices y se venden 4 codornices cada día.
  - a) Modifique el código de python de esta sección para calcular la población de codornices para los primeros 10 días.
  - b) Existe una población inicial de codornices bajo la cual el número de codornices permanece igual cada día. Encuentre dicha población.
  - c) Calcule la población de codornices  $P_t$  para diferentes cantidades iniciales de codornices  $P_0$  y realice el gráfico equivalente a la figura 4.15 para este caso. Con base en el gráfico responda las siguientes preguntas: ¿Qué cantidades iniciales conducen a una reducción (aumento) de la población de codornices con los días? Si cambia el número de codornices que se venden por día, ¿cómo cambiaría este gráfico? ¿Considere este modelo realista o es necesario introducir otras consideraciones? explique brevemente cuáles serían esas consideraciones.
4. Encuentre una situación a la que le pueda aplicar, o tomarlo como punto de partida, el modelo de ecuación de diferencias para la venta de codornices.
  - a) Con esta situación en mente, escriba los pasos para la construcción del modelo de acuerdo a lo visto en el capítulo 1.
  - b) Identifique si es necesario modificar el modelo y realice los cambios pertinentes.
  - c) Modifique el código de python para implementar el modelo y obtener resultados. Analice los resultados y saque conclusiones al respecto.
5. Muestre en cada caso que la función dada es una solución de la ecuación diferencial indicada:
  - $\frac{dy}{dt} = y + t, \quad y(t) = Ce^t - t - 1$
  - $\frac{dy^2}{dt^2} + 9y = 0, \quad y(t) = \cos(3t)$
  - $\frac{dy}{dx} + 2y = \sin(x), \quad y(t) = -\frac{1}{5}e^{-2x}(-1 + e^{2x}\cos(x) - 2e^{2x}\sin(x)).$
6. ¿Es  $y(t) = e^t - 1$  una solución de la ecuación diferencial  $\frac{dy}{dt} = y + t$  ?
7. Compruebe que  $y(t) = 0$  y  $y(t) = t^4/16$  son soluciones del problema de valor inicial

$$y' = ty^{1/2}, \quad y(0) = 0.$$

Igualmente compruebe que  $y(t) = -2\cos(4t) + \frac{1}{4}\sin(4t)$  es solución del problema de valor inicial

$$y'' + 16y = 0, \quad y\left(\frac{\pi}{2}\right) = -2, \quad y'\left(\frac{\pi}{2}\right) = 1.$$

8. Realice un diagrama de fase para el caso de la ecuación diferencial

$$\frac{dy}{dt} = f(y) = ky,$$

analizando por separado los casos  $k \geq 0$ ,  $k < 0$ . Indique las zonas de crecimiento, decrecimiento y tipos de curvatura que tienen las soluciones de esta ecuación diferencial ¿Qué se puede concluir acerca de la influencia del parámetro  $k$  en el comportamiento de las soluciones ?

9. Adapte la rutina presentada en este capítulo para graficar los campos de direcciones correspondientes a las siguientes ecuaciones diferenciales:

$$\begin{aligned}\frac{dy}{dt} &= y - y^2, & \frac{dy}{dt} &= 2y - y^3 \\ \frac{dy}{dt} &= \cos(y), & \frac{dy}{dt} &= -5y - \sin(y), \\ \frac{dy}{dt} &= ty, & \frac{dy}{dt} &= t^2 - y^2.\end{aligned}$$

10. Realice una nueva etapa de modelamiento usando los datos en la tabla 4.2, donde considere un factor limitante en la cantidad de biomasa en el ejemplo anterior, a través de la ecuación diferencial de tipo logístico

$$\frac{dP}{dt} = kP - \beta P^2.$$

Debe calcular los parámetros  $k$  y  $\beta$  correspondientes utilizando un proceso de regresión apropiado.

11. Considere la siguiente tabla de datos:

donde se presenta la expectativa de vida media en años de los norteamericanos entre los años de 1900 a 2010. La función  $P(t)$  denotará la expectativa de vida de los norteamericanos en el tiempo  $t$ , donde comenzamos en  $t = 1$  para el año 1900,  $t = 2$  para el año 1910, y así sucesivamente. Construya un modelo explicando todas las etapas requeridas para describir la cantidad  $P(t)$  en función del tiempo  $t$ . Apóyese en los análisis analíticos y gráficos estudiados hasta el momento.

12. Escriba rutinas en python que implementen los métodos de Euler y de Adams para aproximar las soluciones de los siguientes problemas de valor inicial:

- a)  $y' = (t + 1)y^2$ ,  $y(0) = 1$
- b)  $y' = ty - y^3$ ,  $y(1) = 1$
- c)  $y' = t^2 - y^3$ ,  $y(2) = 2$

13. Con base en la información suministrada para el sistema masa–resorte, escriba los pasos descritos en el capítulo 1 para la construcción del modelo de este sistema.



14. En el desarrollo del modelo del sistema masa-resorte mencionamos que el algoritmo de Euler, ecuaciones 4.28, es inestable para la solución de la ecuación diferencial derivada de la segunda ley de Newton. A partir del código de python del algoritmo de Euler-Cromer, implemente el algoritmo de Euler y genere los resultados numéricos de desplazamiento y velocidad,  $x(t)$  y  $v(t)$  respectivamente, con los mismos parámetros, condiciones iniciales y paso de tiempo que los utilizados en el algoritmo de Euler-Cromer. Realice un gráfico de tiempo versus desplazamiento y velocidad y compárelo con el gráfico de la figura 4.34. Describa cómo se manifiesta la inestabilidad numérica del algoritmo de Euler en los resultados.
15. El modelo aquí descrito del sistema masa-resorte no tiene en cuenta la fricción del aire. A este modelo se le denomina *oscilador armónico simple*. Un modelo para el sistema masa-resorte que considera la fricción debido a un medio incluye una fuerza de resistencia de la forma  $f_r = -b dx/dt$ , la cual da lugar a la siguiente ecuación diferencial:

$$\frac{d^2x(t)}{dt^2} + 2\beta\frac{dx(t)}{dt} + \omega_0^2x(t) = 0,$$

donde  $\beta = b/(2m)$ , con valor siempre positivo, se denomina parámetro de amortiguamiento y  $\omega_0 = \sqrt{k/m}$  es la frecuencia angular característica en ausencia de amortiguamiento. A este modelo se le llama *oscilador amortiguado*.

- a) Utilice el algoritmo de Euler-Cromer para escribir las expresiones de  $v(t+h)$  y  $x(t+h)$  que le permitan solucionar numéricamente esta ecuación diferencial.
- b) Modifique el código de python del algoritmo de Euler-Cromer para calcular el desplazamiento  $x$  y la velocidad  $v$  del sistema masa-resorte con amortiguamiento. Utilice valores de  $b$  para los siguientes casos:  $\beta^2 \neq \omega_0^2$ ,  $\beta^2 = \omega_0^2$  y  $\beta^2 < \omega_0^2$ . Realice un gráfico de tiempo versus desplazamiento y velocidad y compare sus resultados con el caso sin fricción.
16. Considere la siguiente ecuación vista en la aplicación de la venta de codornices:

$$\frac{\Delta P}{\Delta t} = 0,2P_t - a,$$

donde  $P_t$  es la población de codornices en el día  $t$  y  $a$  es el número de codornices que se venden por día. Si realizamos el límite de  $\Delta t \rightarrow 0$  en ambos lados de la ecuación tendremos la siguiente ecuación diferencial:

$$\frac{dP}{dt} = 0,2P - a.$$

Realice el análisis de campo de direcciones de esta ecuación diferencial, con  $a = 4$ , para sacar conclusiones sobre el comportamiento de la población de codornices como función del número inicial de codornices. Señale en el gráfico la solución de equilibrio de esta ecuación diferencial.

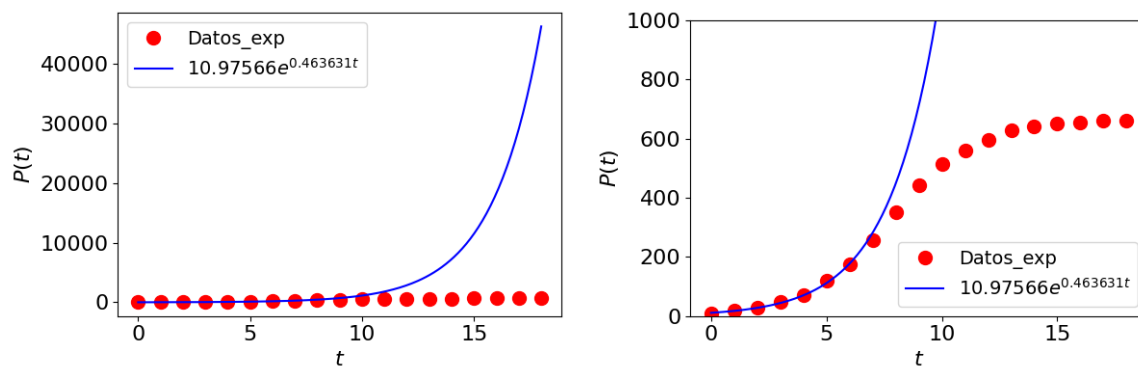


Figura 4.18: Línea continua:  $P(t)$  en función del tiempo  $t$  en horas de acuerdo a la ecuación 4.19. Puntos: datos experimentales tomados de la tabla 4.2. Se observa una discrepancia significativa cuando se considera un tiempo mayor pues los datos experimentales ponen en evidencia la existencia de cierto factor limitante en el crecimiento de  $P(t)$ .

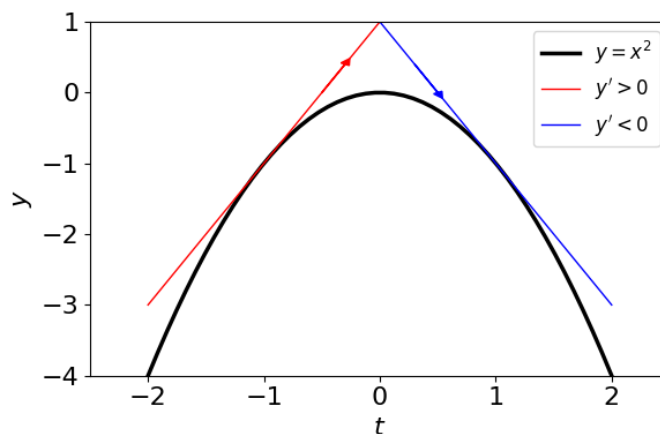


Figura 4.19: Función creciente. La pendiente de la recta tangente es positiva.

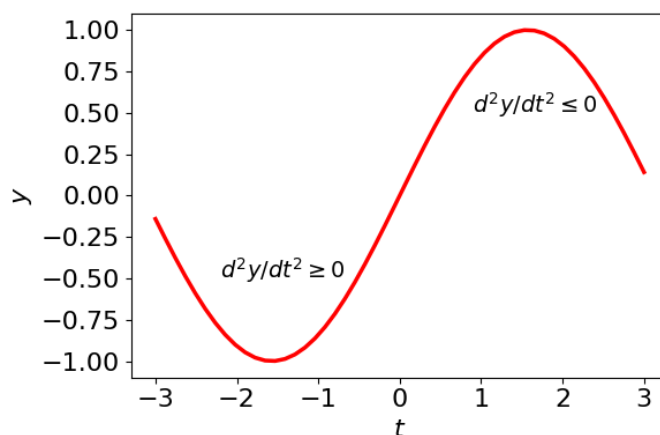


Figura 4.20: Ilustración del criterio de la segunda derivada.

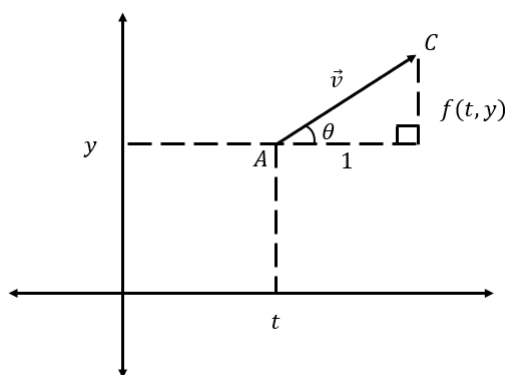


Figura 4.21: Vector asociado al campo  $f(t, y)$  en el punto  $(t, y)$ , correspondiente a la ecuación diferencial  $\frac{dy}{dt} = f(t, y)$ .

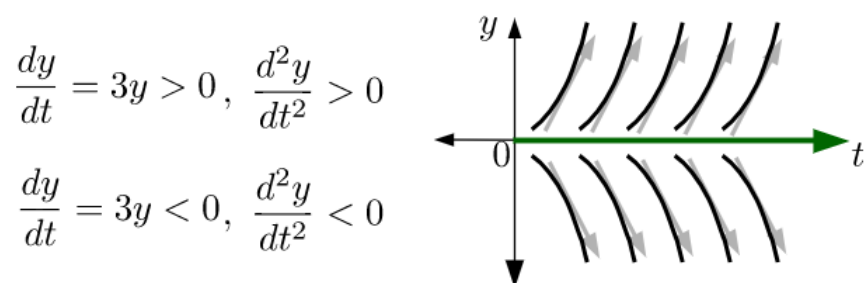


Figura 4.22: Comportamiento cualitativo de las soluciones de la ecuación  $y' = 3y$  en cada una de las regiones. Este tipo de gráfico se conoce como diagrama de fase de la EDO (líneas negras).

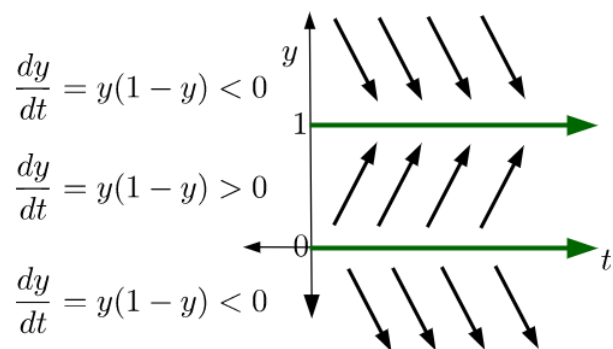


Figura 4.23: Diagrama de fase preliminar de la ecuación diferencial  $y' = f(y) = y - y^2$ .

Año	1900	1910	1920	1930	1940	1950
Expectativa de vida (Años)	47.3	50.0	54.1	59.7	62.9	68.2
Año	1960	1970	1980	1990	2000	2010
Expectativa de vida (Años)	69.7	70.8	73.7	75.4	76.8	78.7

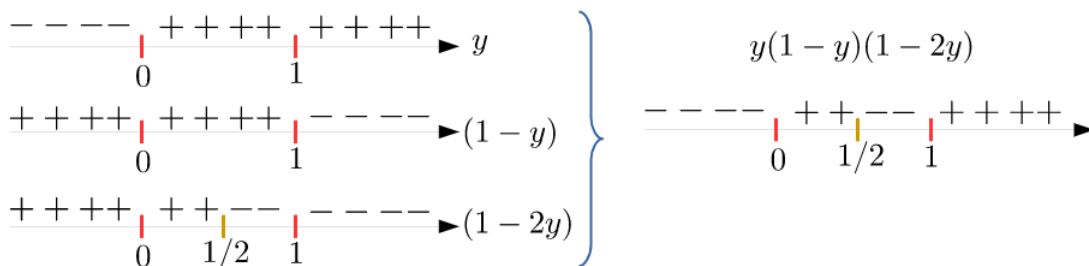
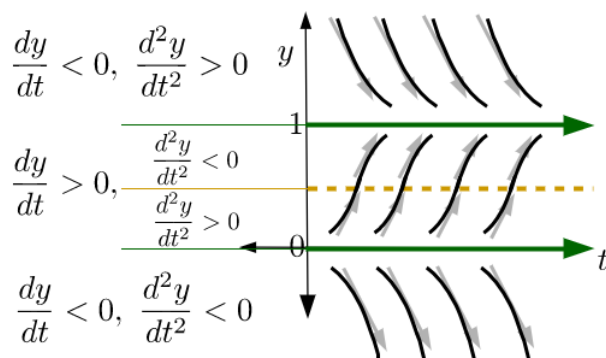
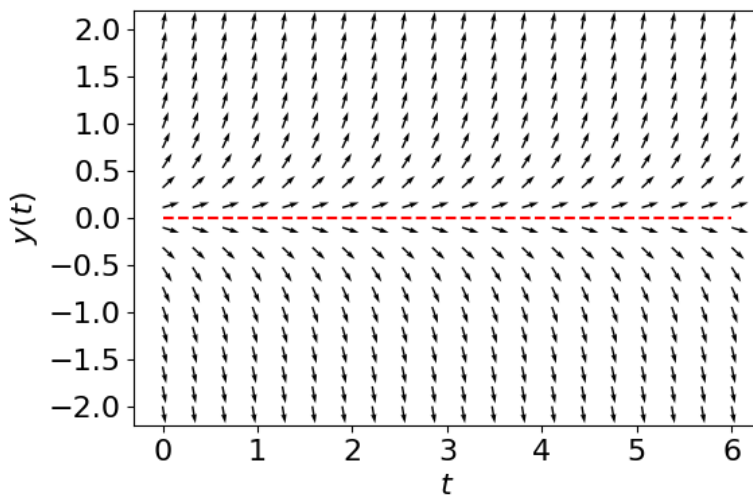


Figura 4.24: Análisis del signo de la segunda derivada.

Figura 4.25: Diagrama de fase de la ecuación  $y' = f(y) = y - y^2$ .Figura 4.26: Campo de direcciones de la ecuación  $\frac{dy}{dt} = 3y$ . En línea punteada se muestra la solución de equilibrio  $y = 0$  de esta ecuación diferencial.

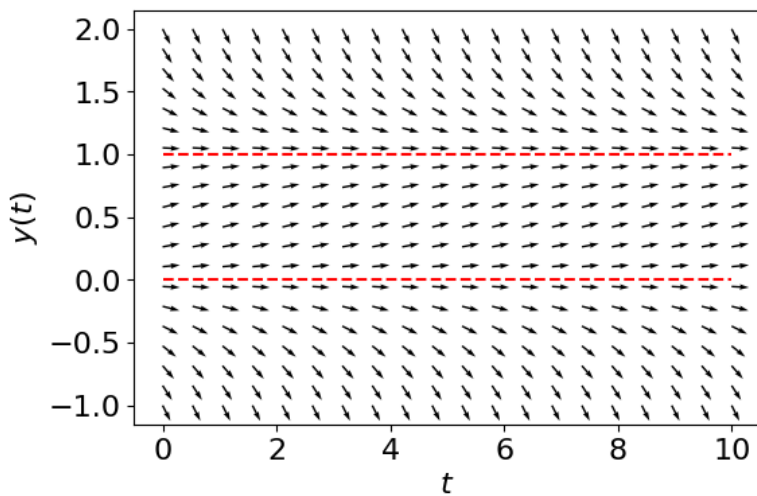


Figura 4.27: Campo de direcciones de la ecuación  $\frac{dy}{dt} = y - y^2$ . En línea punteada se muestran las dos soluciones de equilibrio  $y = 0, y = 1$  de esta ecuación diferencial.

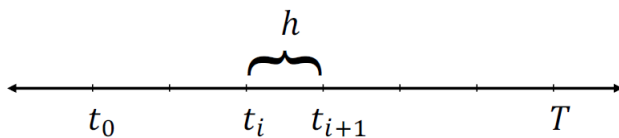


Figura 4.28: Partición del intervalo  $[t_0, T]$ . Cada subintervalo  $[t_i, t_{i+1}]$  tiene longitud  $h$ .

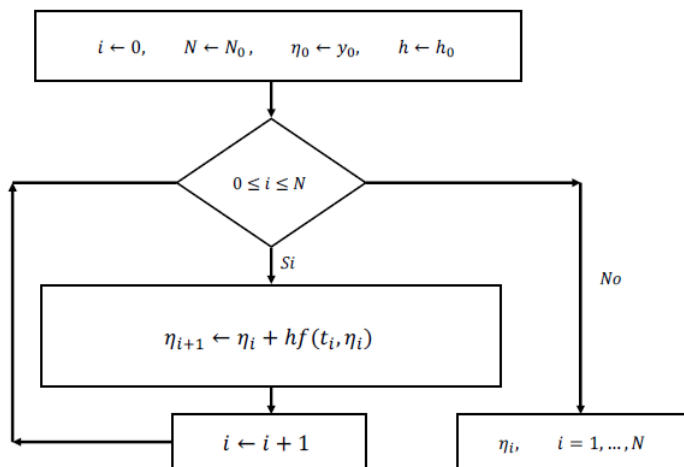


Figura 4.29: Diagrama de flujo para el método de Euler.

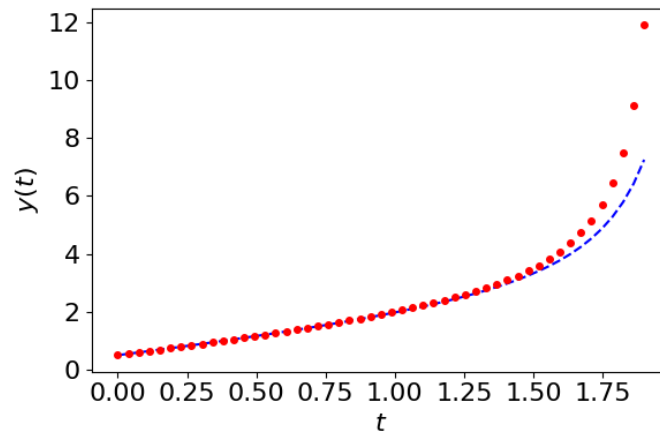


Figura 4.30: Resultado usando tamaño de paso  $h = 0,038$ . Línea discontinua: solución numérica. Puntos: Solución exacta.

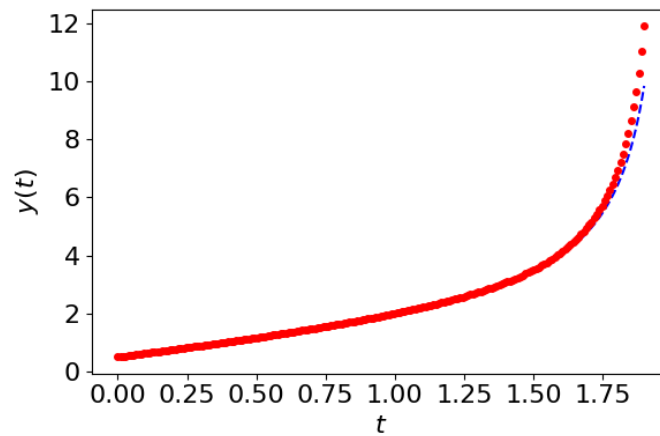


Figura 4.31: Resultado usando tamaño de paso  $h = 0,0095$ . Línea discontinua: solución numérica. Puntos: Solución exacta.

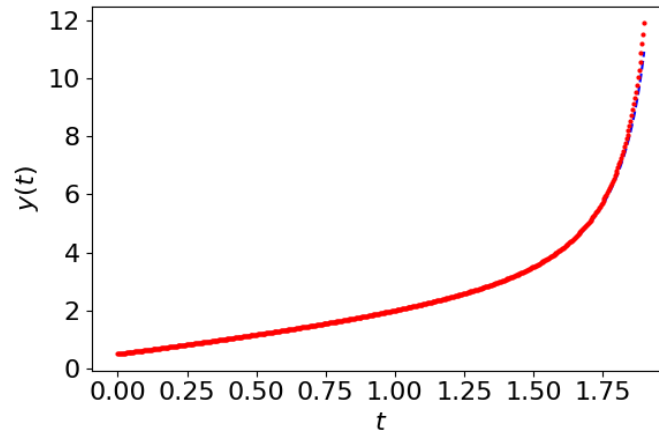


Figura 4.32: Resultado usando tamaño de paso  $h = 0,0038$ . Línea discontinua: solución numérica. Puntos: Solución exacta.

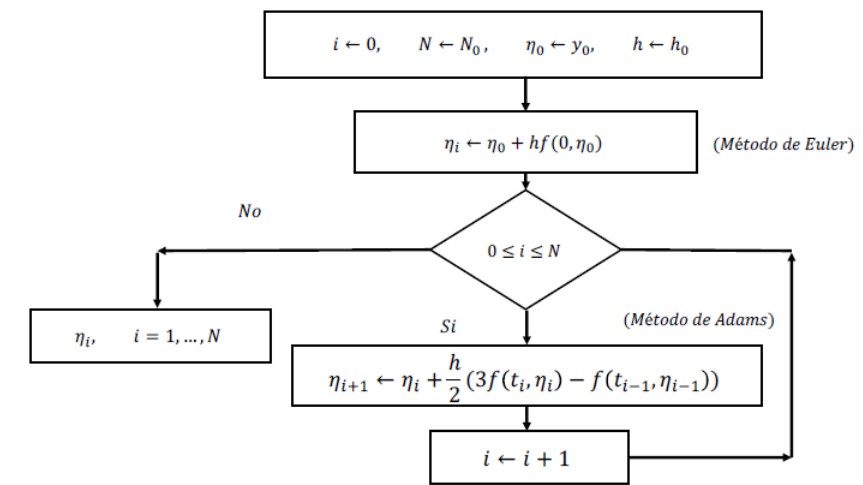


Figura 4.33: Diagrama de flujo para el método de Adams.

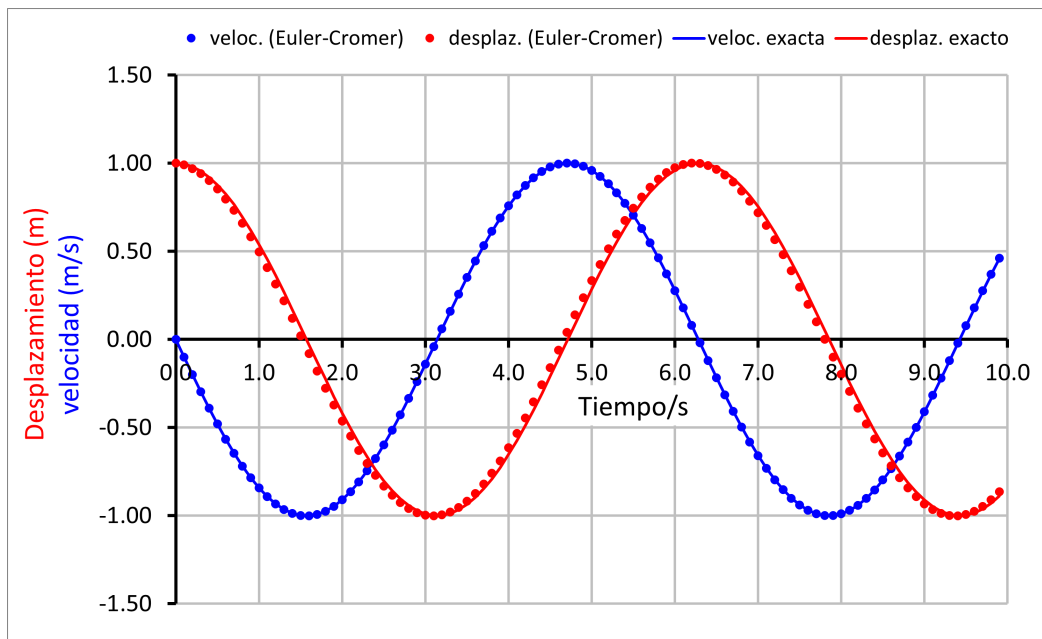


Figura 4.34: Resultados numéricos de la posición y la velocidad del sistema masa-resorte obtenidos mediante el método de Euler-Cromer con tamaño de paso  $h = 0,1$  s y su comparación con los resultados exactos.



## Bibliografía

- [1] F.R. Giordano, W.P. Fox and S. B. Horton. A first course in Mathematical Modeling, 5th ed., Brooks/Cole, 2014. (Citado en pagina [90](#).)
- [2] R. Pearl, The Growth of Population, Quart. Rev. Biol. 2 : 532–548, 1927. (Citado en pagina [90](#).)
- [3] <https://www.physicscurriculum.com/mechanics12>. ( No citado.)
- [4] <https://elabacodemadera.com/2012/08/03/la-sucesion-de-fibonacci-y-los-conejos/comment-page-1> ( No citado.)

## Modelos probabilísticos

Los modelos probabilísticos son representaciones matemáticas utilizadas para describir y analizar fenómenos o sistemas que exhiben incertidumbre o variabilidad. Estos modelos se basan en conceptos y principios de la teoría de la probabilidad y permiten modelar y comprender mejor eventos aleatorios, también conocidos en la literatura como *estocásticos* ([1]).

En términos generales, los modelos probabilísticos capturan la variabilidad inherente en los datos y proporcionan una estructura formal para cuantificar y manejar la incertidumbre. Estos modelos se utilizan en una amplia gama de disciplinas, incluyendo estadísticas, ciencias de la computación, ingeniería, economía, ciencias sociales y muchas otras áreas. Debido al componente aleatorio en los fenómenos modelados, las simulaciones estocásticas no producen resultados determinados, sino conjuntos de posibles resultados.

Dado que los modelos probabilísticos se basan en los principios y conceptos de la teoría de la probabilidad, es importante comprender los fundamentos de esta teoría y para poder aplicarla correctamente estos modelos. Esto implica conocer conceptos como eventos, espacio muestral, variables aleatorias, distribuciones de probabilidad y cálculo de probabilidades. En este capítulo se realizará una breve presentación de estos temas y los usaremos para trabajar algunos modelos probabilísticos de interés en la ciencias naturales .

### 5.1. Fundamentos de probabilidad

Para comenzar esta sección, decimos que la probabilidad es una rama de las matemáticas que se ocupa del estudio y análisis de los fenómenos aleatorios o estocásticos. Proporciona un marco teórico y herramientas matemáticas para comprender y cuantificar la incertidumbre, la variabilidad y el azar en diversos contextos.

La teoría de la probabilidad se construye en base a la teoría de conjuntos, y se establece mediante una serie de axiomas y principios que permiten asignar medidas numéricas a los eventos aleatorios, conocidas como probabilidades, que ocurren dentro de un cierto espacio, llamado espacio muestral. Estas probabilidades se expresan como números reales en un rango de 0 a 1, donde 0 representa la certeza de que un evento no ocurra y 1 representa la certeza de que un evento sí ocurra. Comencemos entonces por introducir algunas definiciones básicas [4].

**Definición 2 (Espacio muestral)** *Un espacio muestral, también conocido como espacio de resultados o conjunto de posibles resultados, es un conjunto que contiene todos los resultados posibles de un experimento aleatorio.*

### Ejemplos

1. *Lanzamiento de una moneda:* El espacio muestral sería Cara, Cruz, ya que esos son los posibles resultados al lanzar una moneda al aire.
2. *Lanzamiento de un dado:* El espacio muestral sería 1, 2, 3, 4, 5, 6, ya que esos son los posibles resultados al lanzar un dado justo.
3. *Extracción de una carta de una baraja:* El espacio muestral sería el conjunto de todas las cartas en una baraja estándar, es decir, las 52 cartas posibles (considerando los valores y palos). Cada elemento del espacio muestral representaría una carta específica.
4. *Lanzamiento de dos dados:* Si lanzamos dos dados al mismo tiempo, el espacio muestral sería el conjunto de todas las combinaciones posibles de los resultados de los dos dados. Por ejemplo, si consideramos el orden de los dados, el espacio muestral podría ser  $(1, 1), (1, 2), (1, 3), \dots, (6, 5), (6, 6)$ .

Por otra parte, los eventos representan conjuntos específicos de resultados que pueden ocurrir dentro del espacio muestral cuando se realiza el experimento. En otras palabras;

**Definición 3 (Evento)** *Un evento es un subconjunto del espacio muestral.*

### Ejemplos

1. *Eventos en lanzamiento de una moneda:*

$A$  = Cara: el evento de obtener cara.

$B$  = Cruz: el evento de obtener cruz.

2. *Eventos lanzamiento de un dado:*

$A$  = 1, 2, 3: el evento de obtener un número menor o igual a 3.

$B$  = Par: el evento de obtener un número par.

$C$  = 5, 6: el evento de obtener un número mayor o igual a 5.

3. *Eventos en extracción de una carta de una baraja:*

$A$  = As de Corazones: el evento de obtener el As de Corazones.

$B$  = Carta Roja: el evento de obtener una carta de corazones o diamantes.

4. *Eventos en lanzamiento de dos dados:*

$A$  =  $(1, 1)$ : el evento de obtener doble uno.

$B$  = Suma igual a 7: el evento de obtener una suma de los resultados igual a 7.

Es importante destacar que los eventos pueden combinar resultados de diferentes maneras. Por ejemplo, se pueden realizar operaciones de unión, intersección y complemento entre eventos para formar eventos más complejos como se hizo en los eventos 2-C, 3-B y 4-B.

### 5.1.1. Axiomas de la probabilidad

Los tres axiomas de la probabilidad son:

- **Axioma de la no-negatividad:** La probabilidad de un evento  $A$  es una función que siempre da un número no negativo. Formalmente decimos que la probabilidad es una función  $P$  que va del espacio de todos los posibles subconjuntos del espacio muestral  $\Omega$  al intervalo  $[0, 1]$ ; es decir

$$\begin{aligned} P : \Omega &\rightarrow [0, 1], \\ A &\mapsto P(A) \end{aligned}$$

donde  $A \subset \Omega$ .

- **Axioma de la normalización:** La probabilidad del espacio muestral completo es igual a 1; es decir  $P(\Omega) = 1$ . Esto significa que la probabilidad de que ocurra al menos uno de los eventos posibles en el espacio muestral es seguro.
- **Axioma de la aditividad:** Para eventos mutuamente excluyentes, es decir, eventos que no pueden ocurrir simultáneamente, la probabilidad de la unión de esos eventos es igual a la suma de las probabilidades individuales. Formalmente, si  $A_1, A_2, A_3, \dots$  son eventos mutuamente excluyentes, entonces la probabilidad de su unión es igual a la suma de sus probabilidades:

$$P(A_1 \cup A_2 \cup A_3 \cup \dots) = P(A_1) + P(A_2) + P(A_3) + \dots$$

Estos axiomas establecen las reglas básicas para asignar probabilidades a eventos y garantizar la coherencia y la consistencia en los cálculos probabilísticos.

### 5.1.2. Combinación de eventos y probabilidades

Dado que los eventos son meramente subconjuntos del espacio muestral, es conveniente introducir las definiciones de probabilidad para sus posibles combinaciones. En particular comenzamos por dar sentido a la probabilidad de una intersección y unión de eventos.

- *Probabilidad de una intersección:*  $P(A \cap B)$  representara la probabilidad de que ocurran simultáneamente los eventos  $A$  y  $B$ .
- *Probabilidad de una unión:*  $P(A \cup B)$  representa la probabilidad de que ocurra al menos uno de los eventos  $A$  o  $B$ .

Basados en lo anterior, podemos definir los siguientes tres tipos de eventos en términos de estas operaciones:

- **Eventos Mutuamente Excluyentes:** Dos eventos  $A$  y  $B$  son mutuamente excluyentes si su intersección es vacía:

$$P(A \cap B) = \emptyset.$$

Es decir, no tienen resultados comunes en el espacio muestral.

- **Eventos Independientes:** Dos eventos  $A$  y  $B$  son independientes si la probabilidad conjunta es igual al producto de las probabilidades individuales:

$$P(A \cap B) = P(A) P(B).$$

Es decir, tienen resultados comunes en el espacio muestral.

- **Evento Complementario:** El evento complementario de  $A$  se denota como  $A' = \Omega - A$ , y denota todos los posibles eventos del espacio muestral que no son parte de  $A$ . La probabilidad del evento complementario es igual a 1 menos la probabilidad del evento original:

$$P(A') = 1 - P(A).$$

### 5.1.3. Probabilidades como porcentajes

La probabilidad de que un evento  $A$  ocurra también se puede expresar mediante porcentajes. Su uso facilita la interpretación de la probabilidad, ya que la mayoría de las personas están familiarizadas con ellos y pueden comprender rápidamente qué tan probable es un evento en relación con el total de posibilidades. En adelante nos referiremos a esta forma de expresar la probabilidad como *probabilidad porcentual*, y esta dada por la siguiente expresión

$$\mathcal{P}(A) = P(A) \times 100\% \tag{5.1}$$

Note que cuando se expresa la probabilidad porcentual de esta forma se representa como valores entre 0 % y 100 %, donde el primero indica una probabilidad nula (evento imposible)

y el segundo indica una probabilidad segura (evento cierto). Claramente estos valores corresponden a las probabilidades de 0 y 1 respectivamente. Por tanto, si tenemos una probabilidad de  $P(A) = 0,8$  de que un evento ocurra, es equivalente a decir que tenemos una probabilidad porcentual de  $\mathcal{P}(A) = 80\%$  de que suceda el evento.

Sin embargo, es importante destacar que los porcentajes son solo una forma de representar la probabilidad y no proporcionan información detallada sobre la base o el razonamiento detrás de esa probabilidad. En contextos académicos, es mas común utilizar fracciones, decimales o notación de proporciones para expresar probabilidades con mayor precisión.

## 5.2. Cálculo de probabilidades

Para calcular probabilidades de eventos en un espacio muestral tenemos dos alternativas: El enfoque clásico y el enfoque frecuentista. Ambos enfoques son útiles en diferentes contextos y se utilizan para calcular probabilidades en situaciones específicas [2].

### 5.2.1. Enfoque clásico

Este enfoque se basa en supuestos de igual probabilidad. Se aplica a situaciones donde todos los resultados posibles son igualmente probables. Para calcular la probabilidad de un evento, se divide el número de resultados favorables entre el número total de resultados posibles. En otras palabras, si queremos calcular la probabilidad de que cierto evento  $A$  ocurra mediante el cociente entre el numero de posibles resultados que hacen que el evento  $A$  se dé, sobre el numero total de todos los resultados posibles como de escribe en la siguiente formula

$$P(A) = \frac{\text{Numero de casos favorables para que ocurra A}}{\text{Numero de todos los casos posibles}} \quad (5.2)$$

En resumen, este enfoque se utiliza en situaciones donde es posible enumerar y contar todos los resultados posibles de manera exhaustiva.

### Ejemplo

Supongamos que queremos lanzar un dado. Entonces el espacio muestral es

$$\Omega = \{1, 2, 3, 4, 5, 6\},$$

donde cada número representa un resultado posible al lanzar el dado. Ahora asignaremos probabilidades a algunos eventos:

- *Evento A*: Obtener un número par.

$$P(A) = P(\{2, 4, 6\}) = \frac{3}{6} = \frac{1}{2}$$

La probabilidad de obtener un número par es de  $\frac{1}{2}$ , ya que hay tres resultados favorables 2, 4 y 6 en un total de seis posibles resultados.

- *Evento B:* Obtener un número mayor a 4.

$$P(B) = P(\{5, 6\}) = \frac{2}{6} = \frac{1}{3}$$

La probabilidad de obtener un número mayor a 4 es de  $\frac{1}{3}$ , ya que hay dos resultados favorables 5y6 en un total de seis posibles resultados.

- *Evento C:* Obtener un número primo.

$$P(C) = P(\{2, 3, 5\}) = \frac{3}{6} = \frac{1}{2}$$

La probabilidad de obtener un número primo es de  $\frac{1}{2}$ , ya que hay tres resultados favorables 2, 3y5 en un total de seis posibles resultados.

### Ejemplo

Supongamos que tenemos un mazo de 52 cartas, que consta de 4 tipos (corazones, diamantes, tréboles y picas) y 13 valores (del 2 al 10, más las cartas de figuras: J, Q, K y A).

A continuación calculemos las probabilidades a algunos eventos:

- *Evento A:* Obtener un corazón.

$$P(A) = P(\{\text{corazones}\}) = \frac{13}{52} = \frac{1}{4}$$

La probabilidad de obtener un corazón es de  $\frac{1}{4}$ , ya que hay 13 cartas de corazones en un total de 52 cartas posibles.

- *Evento B:* Obtener una carta de figura (J, Q, K).

$$P(B) = P(\{J, Q, K\}) = \frac{12}{52} = \frac{3}{13}$$

La probabilidad de obtener una carta de figura es de  $\frac{3}{13}$ , ya que hay 12 cartas de figura en un total de 52 cartas posibles.

- *Evento C:* Obtener un As.

$$P(C) = P(\{A\}) = \frac{4}{52} = \frac{1}{13}$$

La probabilidad de obtener un As es de  $\frac{1}{13}$ , ya que hay 4 Ases en un total de 52 cartas posibles.

### 5.2.2. Enfoque frecuentista

Este enfoque se basa en la repetición de experimentos y el análisis de la frecuencia relativa de un evento en un gran número de repeticiones. En este enfoque, la probabilidad se calcula mediante la observación y el análisis empírico de los resultados de experimentos repetidos. Se utiliza cuando no se pueden aplicar supuestos de igual probabilidad y se requiere una aproximación empírica de la probabilidad.

En este enfoque la probabilidad se define como el límite de la frecuencia relativa de ocurrencia de un evento en un número "suficientemente" grande de repeticiones del experimento.

En resumen, el enfoque frecuentista se basa en la repetición de experimentos y la observación empírica de los resultados.

#### Ejemplo

Supongamos que tenemos un dado de seis caras. usando el enfoque frecuentista calculemos las siguientes probabilidades

- *Probabilidad de obtener un seis.*
- *Probabilidad de obtener un número par*
- *Probabilidad de obtener un número mayor a 4.*

Para realizar este calculo, debemos realizar el siguiente experimento numérico en Python usando el modulo *random*, el cual nos permite generar números aleatorios.

```
1 import random #Modulo para generar numeros aleatorios
2 import matplotlib.pyplot as plt
3
4 # Funci n para simular el lanzamiento de un dado
5 def lanzar_dado():
6     #funcion para generar numeros aleatorios entre 1 y 6
7     return random.randint(1, 6)
8
9 # Realizar una serie de lanzamientos de dado
10 num_lanzamientos = 1000
11 resultados = []
12
13 for _ in range(num_lanzamientos):
14     resultado = lanzar_dado()
15     resultados.append(resultado)
16
17 # Crear un histograma de los resultados
18 plt.hist(resultados,bins=range(1, 8),rwidth=0.65,align= 'mid' )
19 plt.xlabel('Resultado')
20 plt.ylabel('Frecuencia')
21 plt.title('Histograma de lanzamientos de un dado')
22 plt.xticks(range(1, 7))
23 plt.grid(True)
24 plt.show()
25
26 # Calcular la probabilidad de obtener un seis
```



```

27 num_seis= 0
28 for resultado in resultados:
29     if resultado == 6:
30         num_seis += 1
31 probabilidad_seis = num_seis / num_lanzamientos
32
33 # Calcular la probabilidad de obtener un n mero par
34 num_par = 0
35 for resultado in resultados:
36     if resultado % 2 == 0:
37         num_par += 1
38 probabilidad_par = num_par / num_lanzamientos
39
40 # Calcular la probabilidad de obtener un n mero mayor a 4
41 num_mayor_4 = 0
42 for resultado in resultados:
43     if resultado > 4:
44         num_mayor_4 += 1
45 probabilidad_mayor_4 = num_mayor_4 / num_lanzamientos
46
47 # Imprimir los resultados
48 print("Probabilidad de obtener un seis:", probabilidad_seis)
49 print("Probabilidad de obtener un n mero par:", probabilidad_par)
50 print("Probabilidad obtener n mero mayor a 4:", probabilidad_mayor_4)

```

Código 5.1: Programa en Python

En este ejemplo, se simula el lanzamiento de un dado utilizando la función `lanzar_dado()`. Se realiza un número determinado de lanzamientos definidos en la variable `num_lanzamientos`, y se almacenan los resultados en la lista `resultados`. Luego, se crea un histograma de los resultados utilizando la función `hist()` de la biblioteca Matplotlib. El histograma muestra la frecuencia de cada resultado en forma de barras y se muestra el la Figura ??.

Después de representar el histograma, se calculan las probabilidades utilizando el enfoque frecuentista. Se cuentan cuántos resultados cumplen con las condiciones de cada evento y se dividen entre el número total de lanzamientos para obtener las probabilidades.

Finalmente, se imprimen las probabilidades calculadas y se muestran en la consola. En nuestro caso, después de correr el programa una vez obtuvimos los siguientes resultados

- *Probabilidad de obtener un seis:* 0.167
- *Probabilidad de obtener un número par:* 0.497
- *Probabilidad obtener número mayor a 4:* 0.337

Cada vez que ejecutes el código, los resultados del histograma y las probabilidades pueden variar ligeramente debido a la naturaleza aleatoria de los lanzamientos de dados.

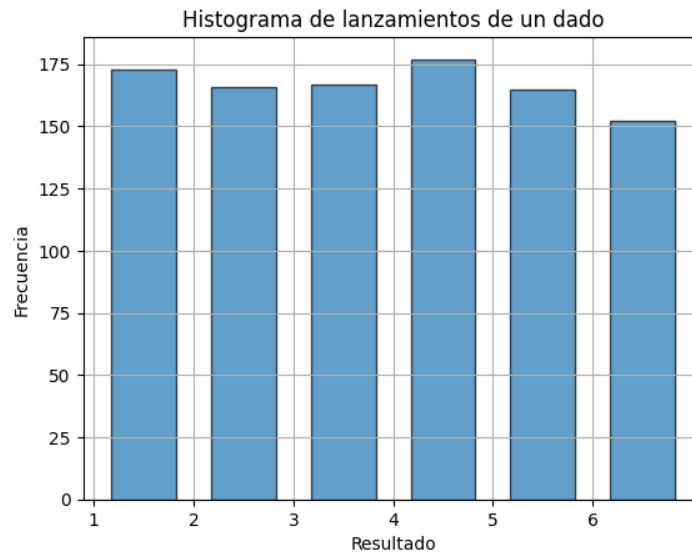


Figura 5.1: Diagrama del histograma de los lanzamientos de un dado

### 5.3. Variables aleatorias

Una variable aleatoria es una función que asigna un número a cada resultado posible de un experimento aleatorio. En otras palabras, es una variable que toma valores numéricos según el resultado de un fenómeno incierto o aleatorio.

Una variable aleatoria se define matemáticamente como una función  $X$  que asigna un valor numérico a cada resultado posible  $\omega$  en el espacio muestral  $\Omega$ . Formalmente, se representa de la siguiente manera:

$$X : \Omega \rightarrow \mathbb{R} \quad (5.3)$$

Donde:

- $X$ : Variable aleatoria
- $\Omega$ : Espacio muestral
- $\mathbb{R}$ : Conjunto de números reales

La función  $X$  asigna a cada resultado posible, o evento  $A$ , un valor numérico  $X(A)$  en el conjunto de números reales. Estos valores numéricos representan las diferentes **observaciones** o resultados que la variable aleatoria puede tomar.

Una variable aleatoria puede ser discreta o continua, dependiendo de si su conjunto de valores posibles es finito o infinito. A continuación damos algunos ejemplos para ilustrar estos casos.

#### 5.3.1. Ejemplo caso discreto

Supongamos que en un experimento lanzamos una moneda justa tres veces y observamos el número de caras que aparecen. Definimos la variable aleatoria  $X$  como:

$$X = \text{El número de caras que resultan en los tres lanzamientos.} \quad (5.4)$$

El espacio muestral  $\Omega$  en este caso consiste en todas las posibles secuencias de tres lanzamientos de una moneda, donde cada lanzamiento puede dar como resultado cara ( $C$ ) o sello ( $S$ ). Por lo tanto:

$$\Omega = \{CCC, CCS, CSC, CSS, SCC, SCS, SSC, SSS\}.$$

La variable aleatoria  $X$  asigna un número a cada resultado posible en  $\Omega$ . En este caso, los posibles valores que la variable aleatoria  $X$  puede tomar son 0, 1, 2 y 3, dependiendo del número de caras obtenidas en los tres lanzamientos. Por ejemplo, si el resultado es  $\{CCC\}$  (las tres caras), entonces  $X(\{CCC\}) = 3$ . De manera similar, si el resultado es  $\{CSS\}$  (cara, sello, sello), entonces  $X(\{CSS\}) = 1$ .

En resume; la variable aleatoria  $X$  representa el número de caras obtenidas al lanzar una moneda justa tres veces, y toma valores discretos de 0 a 3.

### 5.3.2. Ejemplo caso continuo

Supongamos que medimos el tiempo que tarda un estudiante en completar un examen. Definimos la variable aleatoria  $T$  como el tiempo de un estudiante seleccionado al azar para completar un examen que tiene duración máxima de 2 horas.

En este caso el espacio muestral sería el intervalo  $[0, 2]$ , y la variable aleatoria  $T$  tomaría valores reales en dicho intervalo

## 5.4. Función de masa y densidad de probabilidad

En adelante, en este capítulo solo consideraremos variables aleatorias discretas.

### 5.4.1. Función de masa de probabilidad

La función de masa de probabilidad, es una función que asigna una probabilidad a cada valor posible de una variable aleatoria discreta. Esta función permite determinar la probabilidad de que la variable aleatoria tome un valor específico.

Formalmente, para una variable aleatoria discreta  $X$ , la función de masa de probabilidad  $f$  se define como:

$$f(x) = P(X = x) = P(\{\omega \in \Omega : X(\omega) = x\})$$

donde  $P(X = x)$  representa la probabilidad de que la variable aleatoria  $X$  tome el valor  $x$ ,  $\Omega$  es el espacio muestral y  $X(\omega)$  es el valor numérico asignado por la variable aleatoria  $X$  al resultado  $\omega$  en el espacio muestral. La función de masa de probabilidad cumple las siguientes propiedades:

1. Los valores de probabilidad son no negativos:  $P(X = x) \geq 0$  para todo  $x$  en el rango de  $X$ .
2. La suma de las probabilidades es igual a 1:  $\sum_x P(X = x) = 1$ , donde la suma se realiza sobre todos los posibles valores de  $X$ .

La función de masa de probabilidad proporciona información completa sobre como es la "distribución" de la probabilidad de la variable aleatoria discreta.

### Ejemplo

Supongamos que lanzamos dos dados y queremos calcular la función de masa de probabilidad para la suma de los valores obtenidos en ambos dados. Para ello analizamos la información que tenemos:

- *Espacio muestral*  $\Omega$ : El espacio muestral está compuesto por todas las posibles combinaciones de resultados de los dos dados. En este caso, cada dado tiene 6 caras numeradas del 1 al 6, por lo que hay  $6 \times 6 = 36$  resultados posibles en total.
- *Variables aleatorias*  $X$ : Las variables aleatorias  $X$  representan la suma de los valores obtenidos en los dos dados. Los posibles valores de  $X$  van desde 2 (cuando ambos dados muestran un 1) hasta 12 (cuando ambos dados muestran un 6).
- *Función de masa de probabilidad*  $P(X = x)$ : La función de masa de probabilidad asigna la probabilidad de obtener cada valor  $x$  en la variable aleatoria  $X$ .

Si queremos calcular  $P(X = 7)$ , es decir, la probabilidad de obtener una suma de 7, contamos cuántas combinaciones de resultados suman 7 y dividimos por el número total de resultados posibles. En este caso, las combinaciones de resultados que suman 7 son: (1, 6), (2, 5), (3, 4), (4, 3), (5, 2) y (6, 1). Hay 6 combinaciones en total que suman 7. Como el espacio muestral tiene 36 resultados posibles, la probabilidad de obtener una suma de 7 es  $P(X = 7) = 6/36 = 1/6$ . De manera similar, podemos calcular la probabilidad de otros valores de suma. Por ejemplo,  $P(X = 3) = 2/36 = 1/18$  y  $P(X = 12) = 1/36$ .

Por tanto, la función de masa de probabilidad completa se puede representar en forma de tabla o gráfico, donde se muestran todos los posibles valores de  $X$  y sus respectivas probabilidades.

Valor de $X$ ( $x$ )	$f(x) = P(X = x)$
2	1/36
3	2/36
4	3/36
5	4/36
6	5/36
7	6/36
8	5/36
9	4/36
10	3/36
11	2/36
12	1/36

En la tabla se enumeran todos los posibles valores de la variable aleatoria  $X$  (que corresponde a la suma de los valores de los dos dados) y se muestra la probabilidad asociada a cada valor. La suma de las probabilidades para todos los valores debe ser igual a 1. En el siguiente programa se ilustra como se puede generar esta función, y en el gráfico Figura ?? puede observarse el resultado.

```

1 import matplotlib.pyplot as plt
2
3 # Funci n de masa de probabilidad
4 def probabilidad(x):
5     if x < 2 or x > 12:
6         return 0
7     elif x <= 7:
8         return (x - 1) / 36
9     else:
10        return (13 - x) / 36
11
12 # Valores de x
13 valores_x = range(2, 13)
14
15 # Probabilidades correspondientes
16 probabilidades = [probabilidad(x) for x in valores_x]
17
18 # Graficar la funci n de masa de probabilidad
19 plt.plot(valores_x, probabilidades, 'ro' )
20 plt.xlabel('Valor de x')
21 plt.ylabel('f(x)')
22 plt.title('Funci n de Masa de Probabilidad')
23 plt.grid()
24 plt.show()

```

Código 5.2: Programa en Python

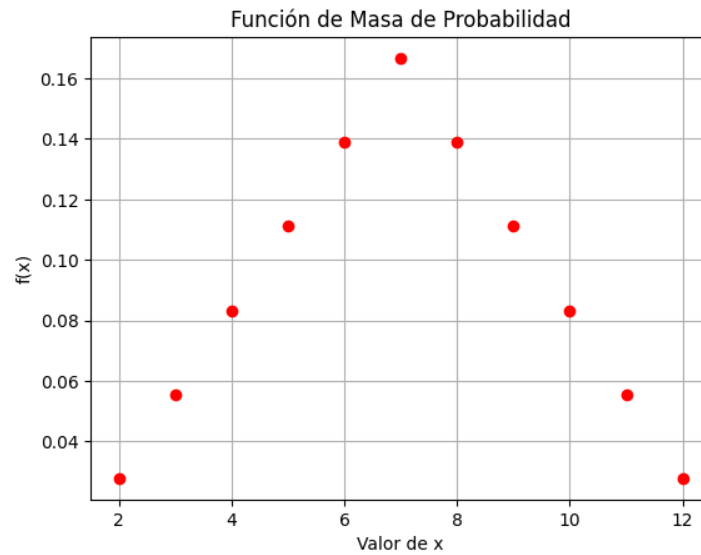


Figura 5.2: Función de masa de probabilidad (discreta)

## 5.5. Algunas distribuciones de probabilidad discreta

La distribución de probabilidad es en ultima instancia, la ecuación que nos proporciona nuestro modelo estadístico.

### 5.5.1. Distribución de probabilidad uniforme discreta

En esta distribución, todos los resultados posibles tienen la misma probabilidad de ocurrir. Por ejemplo, al lanzar un dado justo de 6 caras, cada resultado (1, 2, 3, 4, 5, 6) tiene una probabilidad de  $1/6$ .

Sea  $X$  una variable aleatoria que representa un experimento con resultados discretos. Si la distribución de probabilidad de  $X$  es uniforme discreta, entonces la función de masa de probabilidad de la distribución de probabilidad uniforme discreta se define como:

$$f(x) = P(X = x) = 1/n$$

donde  $P(X = x)$  es la probabilidad de que la variable aleatoria  $X$  tome el valor  $x$ , y  $n$  es el número total de resultados posibles.

Problemas tratados en las secciones anteriores como el de establecer la probabilidad de obtener un número impar al lanzar dos dados, o el de obtener un as de una baraja, son ejemplos de este tipo de distribución.

### 5.5.2. Distribución binomial

La distribución binomial es una distribución de probabilidad discreta que modela  $n$  experimentos aleatorios independientes con dos posibles resultados: éxito (generalmente representado como 1) y fracaso (generalmente representado como 0). Se define mediante dos parámetros: el número de ensayos  $n$  y la probabilidad de éxito en cada ensayo  $p$ .

La función de masa de probabilidad de la distribución binomial se define de la siguiente manera:

$$f(x) = P(X = x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x},$$

donde  $P(X = x)$  es la probabilidad de que la variable aleatoria  $X$  tome el valor  $x$  en  $n$  ensayos.  $p$  es la probabilidad de éxito en cada ensayo y  $(1-p)$  es la probabilidad de fracaso en cada ensayo. La expresión  $\frac{n!}{x!(n-x)!}$  se suele escribir de forma compacta como

$$\frac{n!}{x!(n-x)!} = \binom{n}{x}$$

lo que da el resultado el número de subconjuntos de  $x$  elementos **sin repetición** que se pueden tomar de un conjunto de  $n$  elementos.

### Ejemplo

Supongamos que un fabricante de dispositivos electrónicos sabe por experiencia que la tasa de producción de dispositivos defectuosos es del 5 % y desea estimar la probabilidad de obtener exactamente 3 dispositivos defectuosos en un lote de 10 dispositivos seleccionados al azar. En este caso, podemos utilizar la distribución binomial para modelar la situación. La distribución binomial se define por dos parámetros: el número de ensayos (en este caso, seleccionar 10 dispositivos) y la probabilidad de éxito en cada ensayo (la tasa de dispositivos defectuosos, que es del 5 %).

La fórmula para calcular la probabilidad de obtener exactamente  $x$  éxitos en  $n$  ensayos es:

$$f(x) = P(X = x) = \binom{n}{x} p^x (1-p)^{n-x} \quad (5.5)$$

Donde  $P(X = x)$  es la probabilidad de obtener  $x$  éxitos.  $p$  es la probabilidad de éxito en cada ensayo (la tasa de dispositivos defectuosos, que es del 5 %).  $n$  es el número total de ensayos (seleccionar 10 dispositivos).

En nuestro ejemplo, queremos calcular la probabilidad de obtener exactamente 3 dispositivos defectuosos, por lo tanto,  $x = 3$ .

$$P(X = 3) = \binom{10}{3} (0,05)^3 (1 - 0,05)^{10-3}$$

Calculando los valores, obtenemos:

$$\begin{aligned} P(X = 3) &= \frac{10!}{3!(10-3)!} (0,05)^3 (0,95)^7 \\ P(X = 3) &= 0,1176 \end{aligned}$$

Por lo tanto, la probabilidad de obtener exactamente 3 dispositivos defectuosos en un lote de 10 dispositivos seleccionados al azar es de aproximadamente 0,1176, o de forma porcentual

11,76 %.

**Nota:** Para el caso de  $n = 1$ , es decir, cuando se realiza un solo experimento, la distribución binomial se suele llamar *Distribución de Bernoulli*.

### 5.5.3. Distribución de Poisson

La distribución de Poisson es una distribución de probabilidad discreta que se utiliza para modelar el número de eventos raros que ocurren en un intervalo de tiempo o espacio fijo, cuando la tasa promedio de ocurrencia de esos eventos es conocida. Se define mediante un parámetro lambda ( $\lambda$ ), que representa la tasa promedio de ocurrencia de los eventos.

La función de probabilidad de la distribución de Poisson se define matemáticamente como:

$$f(x) = P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!} \quad (5.6)$$

La fórmula anterior calcula la probabilidad de que la variable aleatoria  $X$  tome un valor específico  $x$ . La tasa promedio de ocurrencia  $\lambda$  determina la forma de la distribución de Poisson, y establece la ventana temporal en donde se calculan las probabilidades.

Algunos ejemplos comunes de aplicaciones de la distribución de Poisson incluyen el modelado de la llegada de clientes a un negocio, el número de llamadas telefónicas recibidas por minuto, o el número de defectos en un producto por unidad de tiempo.

Es importante destacar que la distribución de Poisson asume que los eventos ocurren de manera independiente y que la tasa de ocurrencia promedio es constante en todo el intervalo considerado.

#### Ejemplo

Supongamos que en promedio se reciben 10 llamadas telefónicas por cada 2 hora en una central de atención al cliente. Queremos determinar la probabilidad de que en un intervalo de 1 hora se reciban exactamente 3 llamadas.

En este caso, la tasa promedio de llegada de llamadas es de 10 llamadas por cada dos horas, es decir que se reciben 5 llamadas en una hora. Por tanto  $\lambda = 5$ . Por otro lado la variable aleatoria en este caso sería  $x = 3$ . Así tenemos que

$$P(X = k) = \frac{e^{-\lambda} \lambda^x}{x!}$$

Sustituyendo los valores en la fórmula de Poisson:

$$P(X = 3) = \frac{e^{-5} \cdot 5^3}{3!} = \frac{e^{-5} \cdot 125}{6} \approx 0,14037$$

Por lo tanto, la probabilidad de recibir exactamente 3 llamadas en un intervalo de 1 hora es aproximadamente 0,14037 o 14,037 %.



## 5.6. Función de verosimilitud

El método de máxima verosimilitud (Maximum Likelihood Estimation en inglés) es un método utilizado para estimar los parámetros de un modelo estadístico a partir de un conjunto de datos observados. El objetivo es encontrar los valores de los parámetros que maximizan la verosimilitud de los datos observados, es decir, la probabilidad de obtener los datos dados los valores de los parámetros. En esta sección explicaremos los conceptos básicos en los cual se basa, y realizaremos algunos ejemplos para las distribuciones discretas binomial y de Poisson.

### 5.6.1. El método de máxima verosimilitud

El método de máxima verosimilitud se basa en la idea de que los valores de los parámetros que hacen más probable la ocurrencia de los datos observados son los más probables. Para encontrar los valores óptimos, se busca maximizar la función de verosimilitud, que es la función que describe la probabilidad conjunta de los datos observados como función de los parámetros. En el caso discreto, la función de verosimilitud se define matemáticamente como el producto de las densidades de masa de probabilidad de los datos observados. Aquí asumiremos que esta función solo depende de un parámetro. Entonces, matemáticamente escribimos

$$L(\theta) := \prod_{i=1}^N f(x_i, \theta) \quad (5.7)$$

Donde  $f(x_i)$  es la distribución de probabilidad que se supone describe las probabilidad de ocurrencia de las variables aleatorias, y  $\theta$  es su parámetro.

Para maximizar esta función, debemos encontrar su derivada con respecto al parámetro  $\theta$ . Sin embargo, si aplicáramos las reglas de la derivada de un producto obtendríamos una expresión de  $N \times N$  términos, lo cual nos daría una expresión muy grande.

Una forma de solventar esta dificultad consiste en considerar el logaritmo de esta función, la cual (aplicando la regla del logaritmo de un producto) esta dada por

$$\ln(L(\theta)) = \sum_{i=1}^N \ln(f(x_i, \theta)) \quad (5.8)$$

Esta función al ser derivada solo arroja  $N$  términos, lo cual es mucho mas conveniente de trabajar. Además, puede probarse fácilmente que su valor máximo de esta en el mismo valor del parámetro  $\theta$ , que el de la función original. Es decir;  $\ln(L(\theta))$  y  $L(\theta)$  alcanzan su valor máximo en el mismo valor de  $\theta$ . Usando este hecho se puede estimar el valor del parámetro  $\theta$  para ciertas distribuciones como veremos a continuación.

### 5.6.2. Estimación de parámetros

#### Distribución de Poisson

Supongamos que tenemos  $N$  muestras de un experimento. Es decir, la variable aleatoria  $X$  toma los valores  $x_1, x_2, \dots, x_N$ . Además, supongamos que las variables aleatorias satisfacen una distribución de Poisson (ver (5.6)). Es decir,

$$P(X = x_i) = \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}$$

para cada variable aleatoria. En este caso nuestro parametro  $\theta$  corresponde al valor  $\lambda$  de la distribucion de Poisson. Por tanto la funcion de verosimilitud toma la forma

$$\begin{aligned} L(\theta) &= \prod_{i=1}^N P(x_i = x_i) \\ &= e^{-\lambda} \lambda^{x_1} \frac{1}{x_1!} \dots e^{-\lambda} \lambda^{x_N} \frac{1}{x_N!} \\ &= \frac{1}{x_1!} \dots \frac{1}{x_N!} e^{-\lambda N} \lambda^{(\sum_{i=1}^N x_i)} \end{aligned}$$

Reorganizando los términos como

$$L(\theta) = \frac{1}{x_1! \dots x_N!} e^{-\lambda N} \lambda^{(\sum_{i=1}^N x_i)}.$$

Tomamos logaritmos

$$\ln(L(\theta)) = \ln\left(\frac{1}{x_1! \dots x_N!}\right) - \lambda N \ln(e) + \ln(\lambda) \left(\sum_{i=1}^N x_i\right)$$

Derivando con respecto a  $\lambda$

$$\frac{d \ln(L(\theta))}{d\lambda} = -N + \frac{\sum_{i=1}^N x_i}{\lambda}$$

Igualamos a cero y obtenemos el  $\lambda$

$$\lambda = \frac{\left(\sum_{i=1}^N x_i\right)}{N} \quad (5.9)$$

El lado derecho de la expresión se suele llamar **la media muestral**, y se denota como  $\bar{x}$  para indicar que es un valor promedio de las  $N$  muestras; es decir

$$\bar{x} = \frac{\left(\sum_{i=1}^N x_i\right)}{N}$$

de donde se suele escribir simplemente  $\lambda = \bar{x}$

### Ejemplo

Supongamos que se realizan 10 experimentos en un laboratorio para estudiar la tasa de reacciones químicas en una solución. Se registró el número de reacciones químicas en cada uno de los experimentos durante 1 minuto y se obtuvieron las siguientes reacciones en cada solución:

Experimentos	Éxitos
Experimento 1	20
Experimento 2	40
Experimento 3	30
Experimento 4	60
Experimento 5	50
Experimento 6	20
Experimento 7	10
Experimento 8	30
Experimento 9	40
Experimento 10	20

La tasa media de reacciones por minuto,  $\lambda$ , la encontramos mediante la ecuación (5.9):

$$\lambda = \bar{x} = (20 + 40 + 30 + 60 + 50 + 20 + 10 + 30 + 40 + 20)/10 = 32 \quad (5.10)$$

Ahora, supongamos que queremos hallar la probabilidad porcentual de que ocurran 25 reacciones químicas en un minuto. Para ello simplemente calculamos

$$P(X = 25) = \frac{e^{-\lambda} \lambda^{25}}{25!} \approx 3,4710^{-2} \quad (5.11)$$

Lo cual es una probabilidad porcentual aproximada de 3,47 %. Finalmente, supongamos que queremos encontrar la probabilidad porcentual de que ocurran 1919 reacciones químicas en una hora. Para ello debemos primero expresar el parámetro  $\lambda$  en las unidades correctas. Es decir, debemos expresar  $\lambda$  en reacciones por hora. Esto se hace de forma sencilla mediante transformación de unidades

$$\lambda = 32 \frac{\text{reacciones}}{\text{minuto}} * \frac{60 \text{ minutos}}{\text{hora}} = 1920 \frac{\text{reacciones}}{\text{hora}} \quad (5.12)$$

Luego, reemplazando en la formula de la distribución de Poisson obtenemos

$$100 \% * P(X = 1918) = 100 \% * \frac{e^{-1920} (1920)^{1918}}{1918!} \approx 3,4710^{-2} = 0,909 \%$$

### Distribución de Binomial

De forma similar al caso de Poisson, supongamos que tenemos  $N$  muestras de un experimentos y que las variables aleatorias satisfacen una distribución de Binomial, el número de ensayos por muestra dado por  $n$  fijo. Entonces,

$$P(X = x_i) = \binom{n}{x_i} p^{x_i} (1 - p)^{n - x_i}$$

$$L(p) = \prod_{i=1}^N \binom{n}{x_i} p^{x_i} (1-p)^{n-x_i} \\ p^{(\sum_{i=1}^N x_i)} (1-p)^{Nn - \sum_{i=1}^N x_i} \left( \prod_{i=1}^N \binom{n}{x_i} \right)$$

Donde  $p$  es el parámetro de la distribución. Luego, tomando el logaritmo tenemos

$$\ln L(p) = \ln \left( p^{\sum_{i=1}^N x_i} (1-p)^{Nn - \sum_{i=1}^N x_i} \left( \prod_{i=1}^N \binom{n}{x_i} \right) \right) \\ = \ln \left( \prod_{i=1}^N \binom{n}{x_i} \right) + \left( \sum_{i=1}^N x_i \right) \log p + \left( Nn - \sum_{i=1}^N x_i \right) \log(1-p).$$

Note que el primer termino del lado derecho es una constante. Entonces la derivada solo se ocupa de los dos términos restantes, lo cual da

$$\frac{d \ln L(p)}{dp} = \frac{(\sum_{i=1}^N x_i)}{p} - \frac{(Nn - \sum_{i=1}^N x_i)}{(1-p)}$$

Finalmente, igualando a cero obtenemos

$$\hat{p} = \frac{1}{Nn} \sum_{i=1}^N x_i.$$

En términos de la media muestral escribimos lo anterior como

$$\hat{p} = \frac{\bar{x}}{n} \tag{5.13}$$

### Ejemplo

Los genes son partes del ADN y se componen de secuencias de nucleótidos, A: adenina, G:guanina, C: citosina y T: tiamina. Por ello podemos representarlos, por ejemplo, como

$$\text{Gen} = \text{ATGCACTG}$$

Para replicar estos genes La proteína ARN-polimerasa hace copias de estos genes (para formar el ARN-mesajero) y las envía a los ribosomas para crear proteínas que cumplan la función que ese gen dicta (crear ojos negros, cabello rojo, etc). Sin embargo, cuando ocurre este proceso de transcripción, es posible que ocurran ciertos errores. Es decir, es posible que la ARN-polimerasa se equivoque al copiar un nucleótido y altere la configuración del gen, dando como resultado mutaciones en el gen y en las funciones de las proteínas que son generadas de este. Supongamos que queremos estimar la probabilidad de mutación de un nucleótido en un gen de 8 nucleótidos. Definiremos la variable aleatoria como el número de nucleótidos que han

cambiado o mutado, durante el proceso de transcripción. Tomamos  $N = 10$  copias de ARN mensajero correspondientes a este gen, y en cada muestra contamos el número de mutaciones. Los resultados de las mutaciones en la 10 muestras son los siguientes:

Muestra	Número de mutaciones
Muestra 1	0
Muestra 2	1
Muestra 3	0
Muestra 4	0
Muestra 5	2
Muestra 6	0
Muestra 7	2
Muestra 8	0
Muestra 9	1
Muestra 10	0

### Análisis de los datos

1. Realicemos un programa en python que estime la probabilidad de mutación de un nucleótido en el proceso de transcripción.

```

1  import numpy as np
2  from scipy.stats import binom
3
4  # Datos de muestra
5  muestra exitos = [0, 1, 0, 0, 2, 0, 2, 0, 1, 0]
6  tamano_muestra = 8
7
8  # C lculo de la media muestral
9  media_muestral = np.mean(muestra exitos)
10
11 # Estimaci n del par metro p utilizando la media muestral
12 p_estimado = media_muestral / tamano_muestra
13
14 # Imprimir los resultados
15 print("Media muestral:", media_muestral)
16 print("Par metro estimado p:", p_estimado)

```

Código 5.3: Programa en Python

2. Calculemos la probabilidad de que al seleccionar aleatoriamente una copia de ARN-mensajero de este gen, esta no tiene tiene mutaciones.

```

1  # C lculo de la probabilidad de no mutaciones
2  valor_1 = 0
3
4  #Suando el modulo scipy, la funcion de densidad de masa (density mas funci n)
   binomial

```

```

5 probabilidad_1 = binom.pmf(valor_1, tamano_muestra, p_estimado)
6
7 # Imprimir los resultados
8 print("Probabilidad porcentual de no obtener mutaciones =", probabilidad_1)

```

Código 5.4: Programa en Python

3. Calcule la probabilidad de que al seleccionar aleatoria una copia de ARN-mensajero de este gen, esta tiene mutaciones, es decir; **almenos** uno de sus nucleótidos tiene mutaciones.

```

1
2 # Calcular la probabilidad de almenos un nucleotido mutado. Esto lo podemos
   calcular como
3 # el complemento de la probabilidad de no obtener mutaciones
4
5 probabilidad_complemento = 1 - probabilidad_1
6
7
8 print("Probabilidad porcentual de obtener mutaciones =" ,
   probabilidad_complemento)

```

Código 5.5: Programa en Python

4. Realice un gráfico en donde se muestre cual es el numero de mariposas de alas de color verde mas probable de observar en una muestra aleatoria.

```

1
2 #grafico para ver la probabilidad de ver mas genes cambiados
3
4 import matplotlib.pyplot as plt
5
6 for posibles_valores in range( tamano_muestra ):
7
8     #Usaremos barras porcentuales para graficar
9     plt.bar( posibles_valores , 100* binom.pmf( posibles_valores ,
   tamano_muestra, p_estimado) , color='red' )
10
11 plt.ylabel("Probabilidad (%)")
12 plt.xlabel("Numero de genes cambiados")
13 plt.show()

```

Código 5.6: Programa en Python

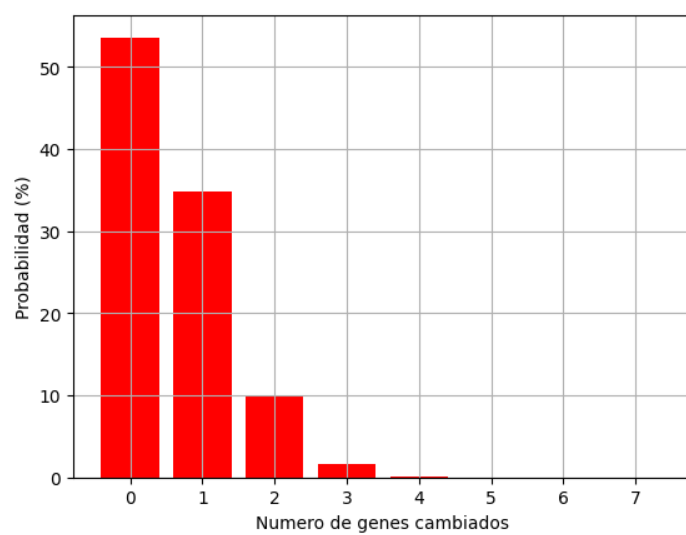


Figura 5.3: Gráfico de la probabilidad porcentual de mutaciones en genes de 8 nucleótidos.

## Bibliografía

- [1] R.E. Walpole, R.H. Myers, S.L. Myers and K. Ye. Probabilidad y Estadística para Ingenierías y Ciencias, novena edición, Pearson Educación, 2012. (Citado en pagina [134](#).)
- [2] D.D. Wackerly, W. Mendenhall III and R.L. Sheaffer. Estadística Matemática con Aplicaciones. Cengage Learning, séptima edición, 2008. (Citado en pagina [138](#).)
- [3] Robert, C. P., Casella, G., Casella, G. (1999). Monte Carlo statistical methods (Vol. 2). New York: Springer. ( No citado.)
- [4] Urias, H. Q., Salvador, B. R. P. (2014). Estadística para ingeniería y ciencias. Grupo Editorial Patria. (Citado en pagina [134](#).)