

Análisis Climático de Bogotá mediante el Modelo de Lorenz

Proyecto realizado por Alan Acero, Johan López y Nicolás Duque para la clase Modelos Matemáticos I (2019082) del Semestre 2024 — 2 en la Universidad Nacional de Colombia.

Las siguientes librerías fueron utilizadas para el desarrollo del proyecto:

```
1 using Plots, Dates, LinearAlgebra, Optim, DifferentialEquations, NLsolve;
```

1. Introducción

El clima es uno de los factores más relevantes a considerar en diversos ámbitos: desde la construcción de ciudades y la planificación de viajes, hasta la decisión sobre la viabilidad de un cultivo o la capacidad de una ciudad para abastecer de agua a su población. Además, el cambio climático, un fenómeno global de creciente preocupación, nos ha llevado a modificar ciertas rutinas; sin embargo, estos cambios aún no son suficientes para prevenir o revertir sus efectos.

Las ciudades en las que habitamos se han transformado en centros donde los impactos del cambio climático son cada vez más evidentes. Esto puede atribuirse a la concentración de personas que experimentan los mismos fenómenos, lo que facilita su identificación, o a eventos extremos que no podemos ignorar, como las inundaciones por lluvias en nuestra capital, períodos de sequía severa que afectan los ciclos de cultivo, incendios forestales como los registrados a nivel nacional en enero, o huracanes y temporales extremos como las inundaciones de la DANA en Valencia, España.

En este contexto, el análisis del clima se convierte en una herramienta clave para prevenir desastres y optimizar la infraestructura frente a estos eventos. En particular, Colombia, como país tropical, se encuentra entre las naciones más vulnerables al cambio climático. Por ello, es crucial disponer de herramientas que permitan anticipar las dinámicas climáticas. Esto facilitará la implementación de políticas sociales y ambientales orientadas a gestionar información y prevenir fenómenos como inundaciones, deslizamientos, sequías e incendios.

2. Objetivos

El presente trabajo tuvo como fines los siguientes objetivos:

- Estudiar el comportamiento del clima de las ciudades seleccionadas desde la perspectiva de los sistemas dinámicos caóticos.
- Evaluar la precisión, confiabilidad y adaptabilidad del modelo propuesto para predecir el comportamiento climático.
- Explorar y aplicar técnicas de análisis para resolver ecuaciones diferenciales no lineales.

3. Metodología

Inicialmente, el proyecto se centró en analizar los datos históricos del clima de Bogotá y Medellín correspondientes al período **2009 — 2022**, con el objetivo de predecir el comportamiento climático de estas ciudades para el año **2025**. Sin embargo, durante el proceso de modelado, nos encontramos con que los datos presentaban un comportamiento inesperado y difícil de capturar con los modelos tradicionales. Esta complejidad nos llevó a explorar alternativas para refinar nuestro enfoque.

Para superar estos desafíos, decidimos generar datos sintéticos que nos permitieran ajustar y mejorar nuestro modelo antes de aplicarlo nuevamente a los datos reales. Utilizamos modelos matemáticos basados en ecuaciones diferenciales, en particular una derivación de la ecuación de Lorenz, conocida por su capacidad para describir sistemas dinámicos complejos y caóticos. Este enfoque nos permitió adquirir una comprensión tanto cualitativa como cuantitativa del clima en el contexto nacional, proporcionando una idea para futuras predicciones y análisis.

4. Marco Teórico

La modelación del clima ha sido uno de los trabajos más retadores dentro de la física entre los siglos **XX** y **XXI**, sin embargo, se han obtenido algunos resultados muy interesantes. Dentro de la literatura, encontramos el sistema de ecuaciones diferenciales de Lorentz, una aproximación y representación del comportamiento de la atmósfera como un fluido no newtoniano en un sistema de referencia en rotación, donde se producen intercambios de energía en distintos puntos.

Su deducción nació como la simplificación de respuestas a otros problemas:

En primer lugar, se tuvo en cuenta el análisis del comportamiento de la convección térmica provocada por la transferencia vertical del calor absorbido sobre la superficie terrestre a lo largo de la troposfera. Este problema se resolvió con el modelo de Rayleigh-Bénard, donde si la densidad disminuye con la altura, el sistema se mantiene estable, pero si la densidad aumenta en capas superiores, ocurre una situación inestable, donde cualquier perturbación puede hacer que el fluido más denso se mueva hacia abajo.

En segundo lugar, está el fenómeno de convección, los efectos invernadero y el deshielo. En **1962**,

Barry Salztamn dedujo un par de ecuaciones diferenciales que explican las proyecciones sobre concentración de gases, los cambios medios mundiales de la temperatura, el aumento del nivel del mar y la estabilización de los gases invernadero en dos dimensiones:

$$\begin{cases} \dot{\eta} = \theta - \eta \\ \dot{\theta} = b\theta - a\eta \end{cases}$$

donde θ representa la desviación de la temperatura a un cierto nivel de referencia y η representa la desviación de la latitud, y a , b son parámetros (que desconocemos).

Finalmente, el sistema de Lorentz se redujo a tres ecuaciones diferenciales ordinarias acopladas que representan la convección de fluidos atmosféricos en tres dimensiones:

$$\begin{cases} \dot{x} = -\sigma x - \sigma y \\ \dot{y} = -xz + \rho x - y \\ \dot{z} = xy - \beta z \end{cases}$$

con parámetros: σ (número de Prandtl) como la relación entre la viscosidad y la conductividad térmica, ρ (número de Rayleigh) como la intensidad de la convección térmica (si es mayor que un valor crítico, se generan células convectivas) y β la relación con la estructura del flujo y la disipación térmica. Al igual, la función $x(t)$ representa la velocidad y la dirección de circulación de fluido, $y(t)$ la variación de temperatura entre las capas de la atmósfera, y $z(t)$ la variación de energía térmica del sistema.

5. Desarrollo del Modelo

5.1 Análisis Dimensional

Obtener los datos específicos de una ciudad dado las variables anteriores es laborioso, ya que no es posible encontrarlas explícitamente. De esta manera, hicimos un cambio de variables y funciones a partir de la información que podíamos obtener, justificando su comportamiento en el modelo y reasignando un nuevo valor cualitativo y dimensional a cada parámetro.

Tomaremos $x(t)$ como la presión atmosférica, $y(t)$ como la temperatura y $z(t)$ como la precipitación (por simplicidad x , y y z), teniendo el siguiente modelo:

$$\begin{cases} \dot{x} = -\sigma x - \sigma y \\ \dot{y} = \rho x - y - xz \\ \dot{z} = xy - \beta z \end{cases}$$

En nuestra primera ecuación, se relaciona la presión atmosférica (x) con la temperatura (y), donde será la rapidez con que cambia la presión (x) en respuesta a las variaciones en la temperatura (y).

En la segunda ecuación, la temperatura (y) se ve influenciada por la presión (x) y la precipitación,

donde si hay precipitaciones intensas (dadas por la precipitaciones y presión), la temperatura (y) disminuye debido a la absorción de calor por evaporación o enfriamiento. Aquí, determina el nivel crítico de temperatura a partir del cual se generan precipitaciones (z) (a un alto valor de p implica que se requiere una temperatura elevada para generar lluvias).

En nuestra tercer ecuación, la precipitación (z) aumenta cuando la presión (x) y la temperatura (y) interactúan, al igual controla la disipación de las lluvias, es decir, es la velocidad con que desaparece la lluvia tras la formación de nubes.

5.2 Modificaciones al Modelo de Lorenz

Pese a que la ecuación diferencial cuenta con un razonamiento cualitativo, no tiene sentido dimensional, por lo tanto, decidimos agregar más variables y analizar qué unidades deben tener. De esta manera, llegamos a que nuestra EDO quedaría de la siguiente forma:

$$\begin{cases} \dot{x} = \sigma(\lambda x - y) \\ \dot{y} = \rho x - \sigma y - \mu x z \\ \dot{z} = \nu x y - \sigma z \end{cases}$$

donde cada parámetro y función tiene la siguiente dimensión:

$$\begin{aligned} [x] &= ML^{-1}T^{-2} \\ [y] &= \theta \\ [z] &= L \\ [t] &= T \\ [\sigma] &= T^{-1} \\ [\lambda] &= ML^{-1}T^{-2}\theta^{-1} \\ [\rho] &= L\theta TM^{-1} \\ [\mu] &= \theta TM^{-1} \\ [\nu] &= L^2T\theta^{-1}M^{-1} \end{aligned}$$

De esta forma, podemos analizar el valor cualitativo de cada variable:

- El parámetro σ es una frecuencia o tasa de cambio temporal del sistema
- El parámetro λ relaciona presión con temperatura. Por lo tanto, es un coeficiente que relaciona la conductividad térmica respecto a una porción de la atmósfera en función del tiempo.
- El parámetro ρ relaciona precipitaciones, temperatura y tiempo, respecto a la masa. Podría llegar a ser un coeficiente que vincula la conductancia térmica específica (por unidad de masa)
- El parámetro μ es la disipación térmica, ya que relaciona temperatura y tiempo, respecto a la masa
- El parámetro ν es el coeficiente de difusividad térmica, esto es, cómo se transporta el calor en función del tiempo y el área.

5.3 Adimensionalización del Modelo

Ya conociendo los valores dimensionales de cada uno, se puede adimensionalizar nuestro sistema de EDOs.

Para la primera ecuación se tiene que:

$$\begin{aligned} T &= [\sigma]^a [\lambda]^b [x_0]^c \\ [t_c] &= [\sigma]^a [\lambda]^b [x_0]^c \\ &= T^{-a} M^b L^{-b} T^{-2b} \theta^{-b} M^c L^{-c} T^{-2c} \\ &= T^{-a-2b-2c} M^{b+c} L^{-b-c} \theta^{-b} \end{aligned}$$

Por lo tanto, se tiene que:

$$\begin{aligned} -a - 2b - 2c &= 1 \\ b + c &= 0 \\ -b - c &= 0 \\ -b &= 0 \end{aligned}$$

Así, $a = -1$ y $t_c = \sigma^{-1}$.

Para la segunda ecuación se tiene que:

$$\begin{aligned} T &= [\rho]^a [\sigma]^b [\mu]^c [y_0]^d \\ [t_c] &= [\rho]^a [\sigma]^b [\mu]^c [y_0]^d \\ &= L^a \theta^a T^a M^{-a} T^{-b} \theta^c T^c M^{-c} \theta^d \\ &= T^{a-b+c} M^{a-c} L^a \theta^{a+c+d} \end{aligned}$$

Por lo tanto:

$$\begin{aligned} a - b + c &= 1 \\ -a - c &= 0 \\ a &= 0 \\ a + c + d &= 0 \end{aligned}$$

Así, $b = -1$ y $t_c = \sigma^{-1}$.

Para la tercera ecuación se tiene que:

$$\begin{aligned} T &= [\nu]^a [\sigma]^b [z_0]^c \\ [t_c] &= [\nu]^a [\sigma]^b [z_0]^c \\ &= L^{2a} T^a M^{-a} \theta^{-a} T^{-b} L^c \\ &= T^{a-b} M^{-a} L^{2a+c} \theta^{-a} \end{aligned}$$

Por lo tanto:

$$\begin{aligned}a - b &= 1 \\ -a &= 0 \\ 2a + c &= 0 \\ a &= 0\end{aligned}$$

Así, $b = -1$ y $t_c = \sigma^{-1}$.

De esta manera, el tiempo característico de las tres ecuaciones es $t_c = \sigma^{-1}$. Ahora, analicemos las escalas características de x_c , y_c y z_c :

$$\begin{aligned}[x_c] &= [\sigma]^a [\lambda]^b [x_0]^c \\ ML^{-1}T^{-2} &= [\sigma]^a [\lambda]^b [x_0]^c \\ &= T^{-a} M^b L^{-b} T^{-2b} \theta^{-b} M^c L^{-c} T^{-2c} \\ &= T^{-a-2b-2c} M^{b+c} L^{-b-c} \theta^{-b}\end{aligned}$$

Lo que implica que:

$$\begin{aligned}a - 2b - 2c &= -2 \\ b + c &= 1 \\ -b - c &= 1 \\ -b &= 0\end{aligned}$$

Así, $c = 1$ y $x_c = x_0$.

$$\begin{aligned}[y_c] &= [\rho]^a [\sigma]^b [\mu]^c [y_0]^d \\ \theta &= [\rho]^a [\sigma]^b [\mu]^c [y_0]^d \\ &= L^a \theta^a T^a M^{-a} T^{-b} \theta^c T^c M^{-c} \theta^d \\ &= T^{a-b+c} M^{a-c} L^a \theta^{a+c+d}\end{aligned}$$

Lo que implica que:

$$\begin{aligned}a &= 0 \\ b &= 0 \\ c &= 0 \\ d &= 1\end{aligned}$$

Así $y_c = y_0$.

$$\begin{aligned}[z_c] &= [\nu]^a [\sigma]^b [z_0]^c \\ L &= [\nu]^a [\sigma]^b [z_0]^c \\ &= L^{2a} T^a M^{-a} \theta^{-a} T^{-b} L^c \\ &= T^{a-b} M^{-a} L^{2a+c} \theta^{-a}\end{aligned}$$

Por lo tanto, se tiene que:

$$\begin{aligned}a - b &= 0 \\ -a &= 0 \\ 2a + c &= 1 \\ a &= 0\end{aligned}$$

Así, $c = 1$ y $z_c = z_0$.

Finalmente, con la información anterior podemos definir las variables adimensionales de la siguiente forma:

$$\begin{aligned}\bar{x} &= \frac{x}{x_c} = \frac{x}{x_0} \\ \bar{y} &= \frac{y}{y_c} = \frac{y}{y_0} \\ \bar{z} &= \frac{z}{z_c} = \frac{z}{z_0} \\ \bar{t} &= \frac{t}{t_c} = \sigma t\end{aligned}$$

Derivando las anteriores expresiones:

$$\begin{aligned}\frac{d\bar{x}}{d\bar{t}} &= \frac{1}{x_0} \frac{dx}{dt} \frac{dt}{d\bar{t}} = \frac{1}{x_0 \sigma} \frac{dx}{dt} \\ \frac{d\bar{y}}{d\bar{t}} &= \frac{1}{y_0} \frac{dy}{dt} \frac{dt}{d\bar{t}} = \frac{1}{y_0 \sigma} \frac{dy}{dt} \\ \frac{d\bar{z}}{d\bar{t}} &= \frac{1}{z_0} \frac{dz}{dt} \frac{dt}{d\bar{t}} = \frac{1}{z_0 \sigma} \frac{dz}{dt}\end{aligned}$$

Luego, reemplazamos en estos resultados las ecuaciones originales para obtener el modelo adimensionalizado:

$$\begin{aligned}\frac{d\bar{x}}{d\bar{t}} &= \frac{1}{x_0 \sigma} \sigma (\lambda x - y) = \frac{\lambda y_0}{x_0} \bar{y} - \bar{x} \\ \frac{d\bar{y}}{d\bar{t}} &= \frac{1}{y_0 \sigma} (\rho x - \sigma y - \mu x z) = \frac{\rho x_0}{y_0 \sigma} \bar{x} - \bar{y} - \frac{\mu z_0 x_0}{\sigma y_0} \bar{x} \bar{z} \\ \frac{d\bar{z}}{d\bar{t}} &= \frac{1}{z_0 \sigma} (\nu \bar{x} x_0 \bar{y} y_0 - \beta \bar{z} z_0) = \frac{\nu x_0 y_0}{z_0 \sigma} \bar{x} \bar{y} - \frac{\beta}{\sigma} \bar{z}\end{aligned}$$

A partir de lo anterior, se obtiene el modelo adimensionalizado:

$$\begin{cases} \dot{\bar{x}} = \tilde{\lambda} \bar{x} - \bar{y} \\ \dot{\bar{y}} = \tilde{\rho} \bar{x} - \bar{y} - \tilde{\mu} \bar{x} \bar{z} \\ \dot{\bar{z}} = \tilde{\nu} \bar{x} \bar{y} - \bar{z} \end{cases}$$

6. Ajuste de Datos

6.1 Recolección de Datos

Se realizó una sintetización de los datos disponibles en la página de Datos Abiertos del Gobierno de Colombia con el fin de obtener los valores promedio de presión (en *Hpa*), temperatura ($^{\circ}C$) y precipitación (*mm*) mensuales del año **2009** al **2022**.

```
168x4 Matrix{Any}:
"2009-01" 744.4 14.05 31.0
"2009-02" 744.17 14.22 48.0
"2009-03" 745.84 14.32 79.0
"2009-04" 744.13 14.71 96.0
"2009-05" 744.24 14.58 77.0
"2009-06" 744.21 14.35 60.0
"2009-07" 744.82 13.92 34.0
⋮
"2022-07" 751.85 14.7 71.0
"2022-08" 751.88 14.39 58.0
"2022-09" 751.81 14.21 77.0
"2022-10" 751.48 14.61 216.0
"2022-11" 752.15 14.56 164.0
"2022-12" 750.73 14.3 67.0
```

Tras ésto se definió la función que corresponde al modelo y a la minimización por mínimos cuadrados.

```
lorenz (generic function with 1 method)
1 function lorenz(du, u, params, tiempo)
2     du[1] = params[1]*u[2] - u[1];
3     du[2] = params[2]*u[1] - u[2] - params[3]*u[1]*u[3];
4     du[3] = params[4]*u[1]*u[2] - u[3];
5 end
```



```

residuoLorenz (generic function with 1 method)
1 function residuoLorenz(params, vDatos, tiempo)
2     # Time domain for the ODE
3     dominioTiempo = (0.0, 168.0)
4
5     # Initial conditions for the Lorenz system
6     V0 = [738.47, 14.55, 75.80]
7
8     # Define and solve the ODE problem
9     EDO = ODEProblem(lorenz, V0, dominioTiempo, params)
10    Sol = solve(EDO, Tsit5(), saveat=tiempo)
11
12    # Extract model data at specified time points
13    vModelo = hcat(Sol.u...)'
14
15    println("Dimensiones de vDatos: ", size(vDatos))
16    println("Dimensiones de vModelo: ", size(vModelo))
17
18    # Calculate residuals
19    res = vDatos .- vModelo
20    nRes = norm(res)
21
22    return nRes
23 end

```

Además de ésto, se usará una notación particular para esta parte del código, con el fin de facilitar su entendimiento:

Datos:

- **datosClima:** Corresponde a nuestros datos base, donde cada una de sus columnas representa:
 1. Mes y año en cuestion;
 2. Presión promedio en ese mes;
 3. temperatura promedio en ese mes;
 4. y precipitación promedio en ese mes.
- **years:** es un vector con el tamaño de la primera columna de datosClima, y aumenta de a 1. En particular indexa los meses dentro de los cuales trataremos los datos
- **datosC:** Es una matriz cuyas columnas son las correspondientes a la presion, temperatura y precipitacion en datosClima, guardado como Float64.
- **P, T, R:** estos funcionaran como prefijo a funciones o datos correspondientes a presión (**P**), temperatura (**T**) y precipitación (**R** por *Rain*)
- **XPorAños:** corresponde a la columna con el tipo de datos **X**, donde **X** puede ser **P**, **T** o **R**
- **Con, Dis:** sufijos relacionados a si un tipo de datos sera continuo (**Con**) o discreto (**Dis**)
- **meses:** prefijo correspondiente a el eje x de muchas de nuestras funciones, donde se toma el intervalo de **0** a **12** de manera continua o discreta (caso particular **mesesDis2** es la repeticion de tomar datos cada **1** unidad, y repetirlo iteradamente para poder mapearlos repetidamente)
- **DatosPart:** **DatosC** pero particionado por años, así en cada coordenada del vector se guardara una matriz **12 × 3** con los datos de ese año.

- **Param:** se usa para representar en un vector los parametros de ciertas funciones particulares.
- **Prom:** referente al promedio

más adelante se definiran otros.

6.2 Ajuste con Datos

6.2.1 Ajuste con Datos reales:

Este primer ajuste se hizo con los datos reales:

El primer ajuste realizado se hizo con los datos obtenidos, por un error de código (en el cual parecia que el ajuste no convergía de una manera apropiada), entonces se buscó avanzar con datos sintéticos, y con un ajuste intermedio de funciones. Para las funciones aquí vistas se utilizara el sufijo **Real**:

```
rLorenz (generic function with 1 method)
```

```
1 rLorenz(parametros) = residuoLorenz(parametros, datosC , years)
```

```
oLorenzReal = * Status: failure (reached maximum number of iterations)
```

```
* Candidate solution
  Final objective value:      2.479782e+03
```

```
* Found with
  Algorithm:      Simulated Annealing
```

```
* Convergence measures
  |x - x'|          = NaN ≠ 0.0e+00
  |x - x'|/|x'|     = NaN ≠ 0.0e+00
  |f(x) - f(x')|    = NaN ≠ 0.0e+00
  |f(x) - f(x')|/|f(x')| = NaN ≠ 0.0e+00
  |g(x)|           = NaN ≠ 1.0e-08
```

```
* Work counters
  Seconds run:   6 (vs limit Inf)
  Iterations:   10000
  f(x) calls:   10001
```

```
1 oLorenzReal = Optim.optimize(rLorenz, [10.0, 28.0, 8/3, 0.5], SimulatedAnnealing(),
  Optim.Options(iterations = 10000))
```

Debido a la complejidad del modelo y a su carácter caótico, no se logra satisfactoriamente encontrar un único valor para sus parámetros. Ésto resulta en que en ejecuciones continuas, los valores pueden ser distintos (si bien son cercanos).

```
oLorenzTuplaReal = ▶ [14.1655, 26.9461, 0.177124, 0.00653995]
```

```
1 oLorenzTuplaReal = oLorenzReal.minimizer
```

La ecuación diferencial ordinaria es:

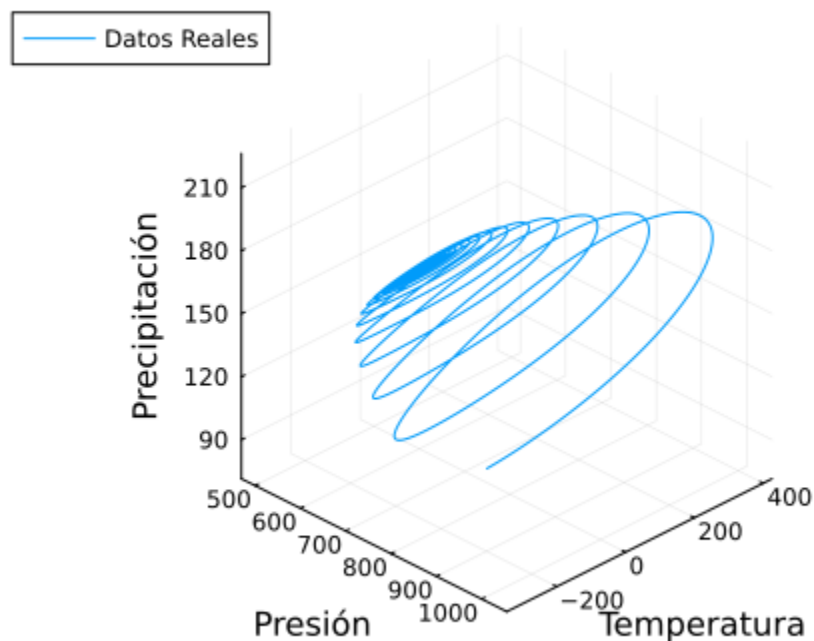
$$x' = 14.165544847052741 y - x$$

$$y' = 26.946145913490376 x - y - 0.1771244440321479 xz$$

$$z' = 0.0065399518120271396 xy - z$$

Se puede ver representado gráficamente en la siguiente figura:

Trayectoria del Sistema



6.2.2 Ajuste con Datos Sinteticos 1:

Debido a ciertos errores de código que llevaron a que el metodo anterior *pareciera* no funcionar, tuvimos otro acercamiento intermedio: con los datos sintéticos.

Cuando hablamos de datos sintéticos para presión, temperatura y precipitación es un acercamiento a través de ciertas funciones, que nuestra hipótesis será que modelan el comportamiento de los parámetros y compararemos con los datos promedio por mes. Estas funciones mantendrán cierto sentido físico a lo largo de un año, para ello le consultamos a algunos físicos y meteorólogos para desarrollar una mejor intuición sobre el tipo adecuado de funciones. Para estas funciones sintéticas

se usará el sufijo **Sint**:

Presión:

La función que modela la presión utiliza una **sinusoidal amortiguada** para representar la variabilidad de la presión en función del tiempo. Este tipo de función es útil cuando se busca modelar fenómenos periódicos cuya amplitud generalmente disminuye con el tiempo. Sin embargo, en este caso particular, debido a los comportamientos de **mayor presión característicos hacia el final del año**, la función ha sido ajustada para que la amplitud crezca en lugar de decaer. La **frecuencia angular** (ω) controla la periodicidad de la oscilación, asegurando que se repita cada seis meses, mientras que el término $e^{-k \cdot (12-x)}$ refleja un comportamiento donde la presión aumenta a medida que se acerca el final del ciclo anual. El parámetro k aún regula la rapidez con la que esta "expansión" de la presión ocurre. De esta manera, la función no solo refleja la periodicidad inherente al comportamiento de la presión, sino también su crecimiento hacia el final del año, lo cual es consistente con los aumentos de presión típicos en ciertas épocas del año, como es el caso en ciudades como Bogotá.

A continuación se presenta la función que modela esta relación:

$$P(x) = P_0 + A \cdot e^{-k \cdot (12-x)} \cdot \sin(\omega \cdot (12-x) + \phi)$$

donde:

$P_0 = 737$	(Presión base, aproximadamente en Bogotá)
$A = 5$	(Amplitud de las oscilaciones)
$k = 0.1$	(Coeficiente de amortiguación)
$\omega = \frac{\pi}{3}$	(Frecuencia angular, asegura que la onda se repita cada 6 meses)
$\phi = 0$	(Fase inicial)

```
PParamSint = ▶ [737.0, 5.0, 0.1, 1.0472, 0.0]
```

```
1 PParamSint = [737., 5., .1 , π / 3 , 0.] # P0, A, k, omega, phi
```

```
PDatos (generic function with 1 method)
```

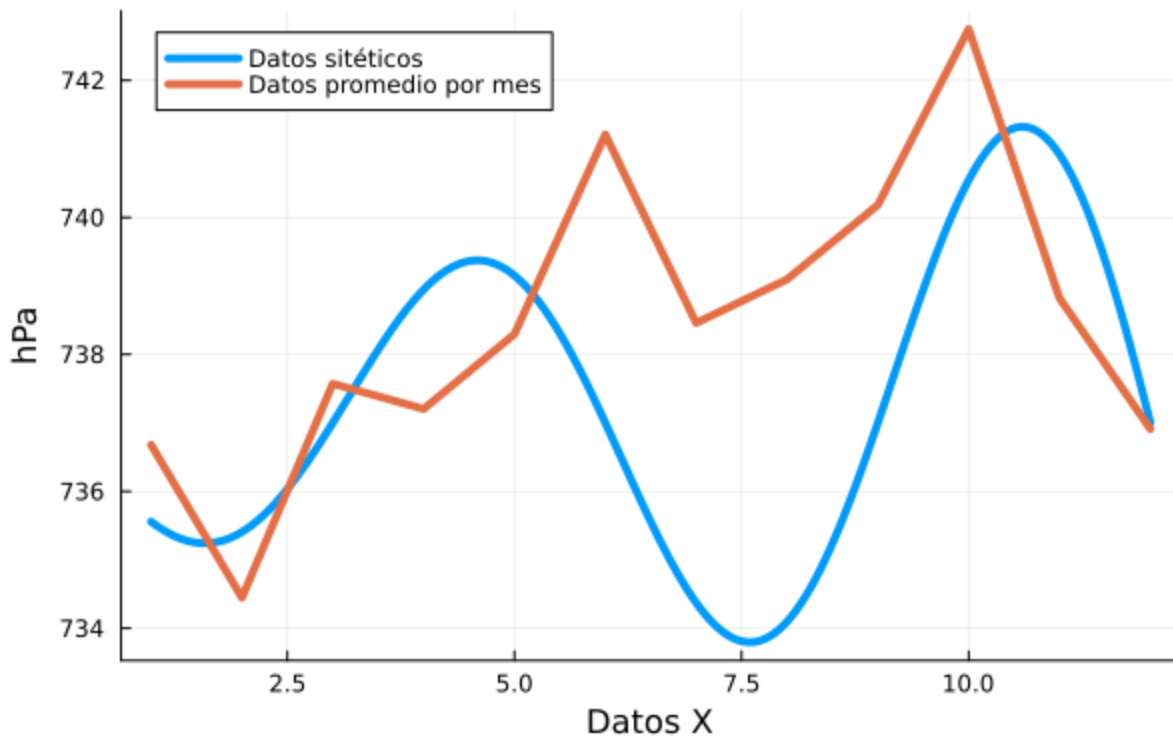
```
1 function PDatos(x, parametros) # función de Presión
2     P0, A, k, omega, phi = parametros
3
4     return (P0 + A * exp(-k * (12-x)) * sin(omega * (12-x) + phi*x))
5 end
```

```
PSint =
```

```
▶ [735.559, 735.558, 735.557, 735.556, 735.555, 735.554, 735.553, 735.552, 735.551, 735.55,
```

```
1 PSint = [PDatos(m, PParamSint) for m in mesesCon]
```

Datos sintéticos presión



Temperatura:

La función que usaremos para modelar la temperatura será una suma entre una distribución normal y un componente lineal adicional. Se decidió la distribución normal puesto que la temperatura suele seguir un patrón estacional, incluso en zonas cercanas del la línea del Ecuador, en particular, las temperaturas suelen ser más altas durante los meses cercanos a la mitad del año.

Nuestra distribución normal se utiliza para modelar estas fluctuaciones suaves en torno al valor central (en este caso, el mes de mayor temperatura), mientras que el componente lineal se agrega para el incremento gradual de temperatura a lo largo de los años por el cambio climático (que se divide entre **12** para asegurar que todos los meses vaya aumentando).

La ecuación que describe esta relación es la siguiente:

$$T(x) = A \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \frac{B}{12} \cdot (x - 1) + C$$

donde:

$A = 1$	(Amplitud de la distribución normal)
$\mu = 6$	(Mes de máxima temperatura, centrado en junio)
$\sigma = 2$	(Desviación estándar de la distribución normal)
$B = 1$	(aumento de temperatura por año)
$C = 14$	(temperatura base en diciembre)

Donde la amplitud refleja las fluctuaciones de temperatura entre valor máximo y mínimo, y la

desviación estándar refleja cuán amplias o estrechas son las fluctuaciones de temperatura respecto al mes central, o en otras palabras, qué tan drástico es el cambio de temperatura de un mes a otro.

```
TParamSint = ▶ [1.0, 6.0, 2.0, 1.0, 14.0]
```

```
1 TParamSint = [1., 6., 2., 1., 14.] # corresponde a A,  $\mu$ ,  $\sigma$ , B, C
```

TDatos (generic function with 1 method)

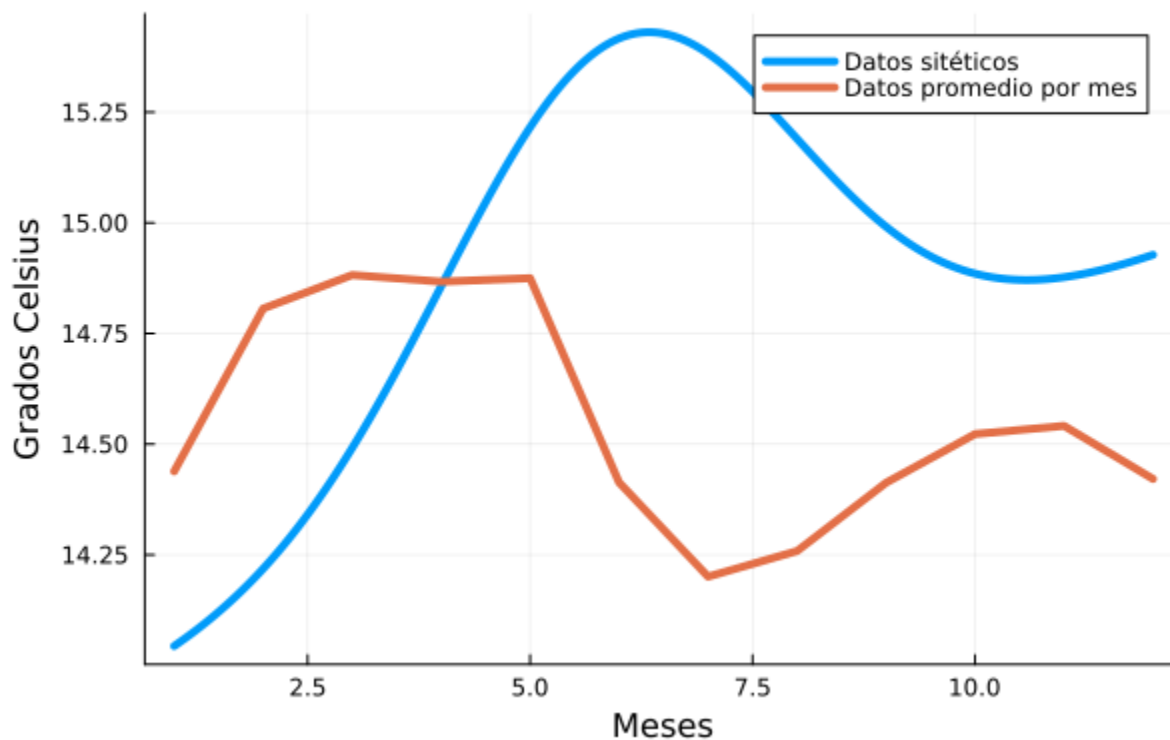
```
1 function TDatos(x, parametros)
2     A,  $\mu$ ,  $\sigma$ , B, C = parametros
3
4     # Componente normal (distribución gaussiana centrada en junio)
5     normal_part = A * exp(-((x -  $\mu$ )^2) / (2 *  $\sigma$ ^2))
6
7     # Componente lineal (tendencia de 1.0 grados por año)
8     linear_part = (B / 12) * (x - 1)
9
10    # Temperatura total
11    return normal_part + linear_part + C
12 end
```

TSint =

```
▶ [14.0439, 14.0441, 14.0442, 14.0444, 14.0445, 14.0446, 14.0448, 14.0449, 14.045, 14.0452,
```

```
1 TSint = [TDatos(m, TParamSint) for m in mesesCon]
```

Datos sintéticos temperatura



Precipitación:

La función que modela la precipitación utiliza una función coseno para representar la variabilidad

de la precipitación a lo largo del año. Las épocas de lluvia se pueden situar alrededor de los meses de marzo/abril y hacia octubre, así que para representar un periodo de **6** meses se tomó abril (**4**) y octubre (**10**).

La ecuación que describe esta relación es la siguiente:

$$P(x) = \frac{A}{2} \cdot \cos\left(\frac{3 \cdot (x - \mu)}{\pi}\right) + B$$

donde:

$A = 160$	(Rango base de la precipitación)
$\mu = 4$	(ubicación del pico de la función coseno)
$B = 80$	(Precipitación mínima)

La **amplitud** A refleja las fluctuaciones entre la cantidad máxima y mínima de precipitación, μ representa el primer mes de mayor lluvia. Mientras tanto, el valor de B representa la precipitación base en los meses secos, garantizando que haya un valor mínimo de precipitación razonable durante el año.

```
RParamSint = ▶ [160.0, 4.0, 80.0]
```

```
1 RParamSint = [160, 4., 80.] # corresponde a A, μ, B
```

```
RDatos (generic function with 1 method)
```

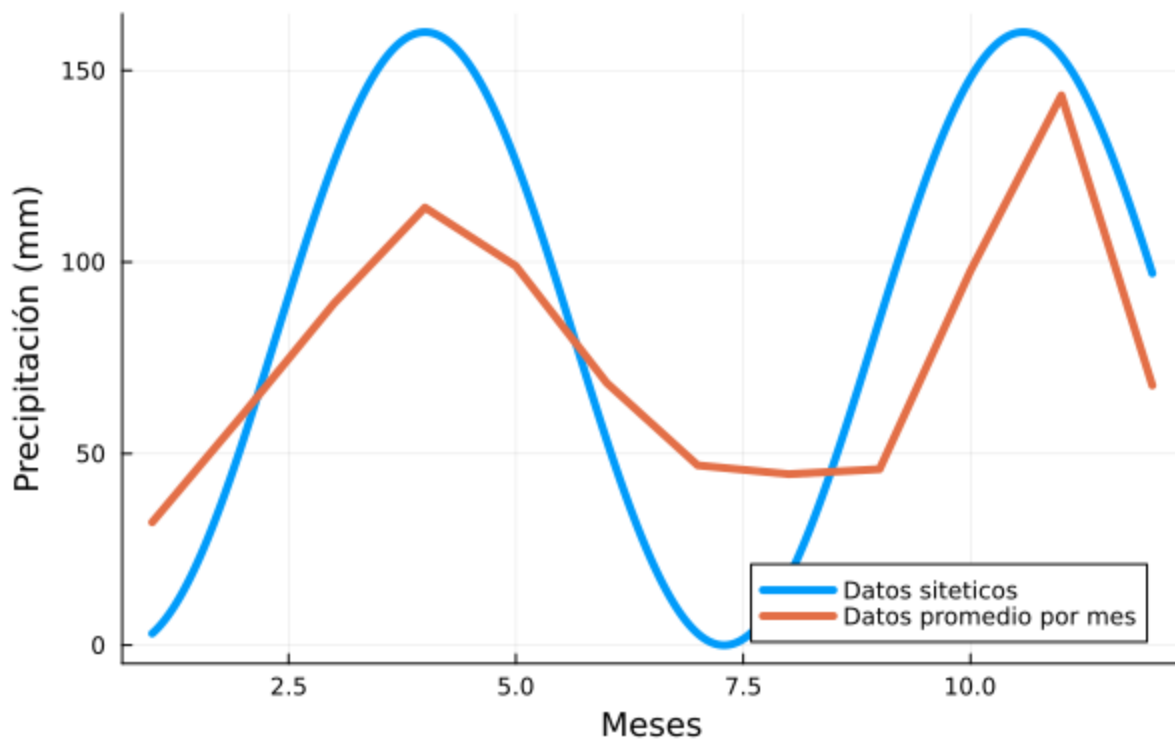
```
1 function RDatos(x, parametros)
2     A, μ, B = parametros
3
4     # Componente coseno para modelar la estacionalidad
5     cos_part = (A / 2) * cos(3 * (x - μ) / π)
6
7     # Precipitación total
8     return cos_part + B
9 end
```

```
RSint =
```

```
▶ [3.04529, 3.0662, 3.08719, 3.10824, 3.12936, 3.15056, 3.17182, 3.19315, 3.21455, 3.23603,
```

```
1 RSint = [RDatos(m, RParamSint) for m in mesesCon]
```

Datos sinteticos precipitación



```
datosSint = 11001x3 Matrix{Float64}:
  735.559  14.0439  3.04529
  735.558  14.0441  3.0662
  735.557  14.0442  3.08719
  735.556  14.0444  3.10824
  735.555  14.0445  3.12936
  735.554  14.0446  3.15056
  735.553  14.0448  3.17182
  ⋮
  737.026  14.9274  97.4052
  737.021  14.9275  97.3306
  737.016  14.9276  97.256
  737.01   14.9276  97.1814
  737.005  14.9277  97.1068
  737.0    14.9278  97.0322
```

```
1 datosSint = hcat(PSint, TSint, RSint) # Datos con nuestras funciones sinteticas a lo largo de mesesCont
```

```
V0Sint = ▶ [735.559, 14.0439, 3.04529]
```

```
1 V0Sint = [PSint[1], TSint[1], RSint[1]] #Condiciones iniciales para el sistema de Lorenz
```

Para este (y el siguiente) ajuste, tenemos que adaptar la función de residuo, puesto que el dominio de tiempo paso de ser de los **14** años que tenemos con los datos reales, a ser tan solo **1**. Esta función queda entonces de la siguiente manera:

residuoLorenz2 (generic function with 1 method)

```
1 function residuoLorenz2(params, vDatos, tiempo, V0)
2     # Time domain for the ODE
3     dominioTiempo = (0.0, 12.0)
4
5     # Define and solve the ODE problem
6     EDO = ODEProblem(lorenz, V0, dominioTiempo, params)
7     Sol = solve(EDO, Tsit5(), saveat=tiempo)
8
9     # Extract model data at specified time points
10    vModelo = hcat(Sol.u...)'
11
12    println("Dimensiones de vDatos: ", size(vDatos))
13    println("Dimensiones de vModelo: ", size(vModelo))
14
15    # Calculate residuals
16    res = vDatos .- vModelo
17    nRes = norm(res)
18
19    return nRes
20 end
```

rLorenz2 (generic function with 1 method)

```
1 rLorenz2(params) = residuoLorenz2(params, datosSint, mesesCon, V0Sint)
```

oLorenzSint = * Status: failure (reached maximum number of iterations)

* Candidate solution

Final objective value: 3.491953e+04

* Found with

Algorithm: Simulated Annealing

* Convergence measures

$ x - x' $	= NaN $\not\approx$ 0.0e+00
$ x - x' / x' $	= NaN $\not\approx$ 0.0e+00
$ f(x) - f(x') $	= NaN $\not\approx$ 0.0e+00
$ f(x) - f(x') / f(x') $	= NaN $\not\approx$ 0.0e+00
$ g(x) $	= NaN $\not\approx$ 1.0e-08

* Work counters

Seconds run: 1 (vs limit Inf)

Iterations: 1000

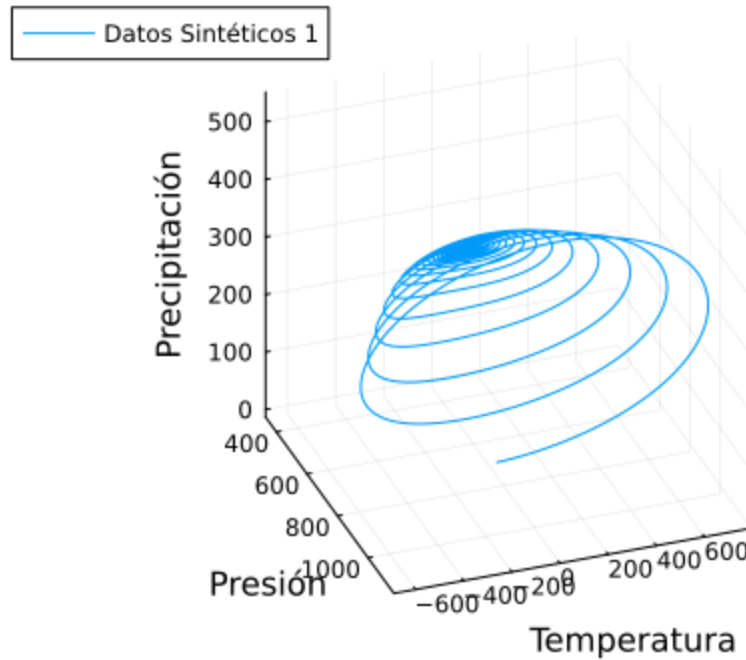
f(x) calls: 1001

```
1 oLorenzSint = Optim.optimize(rLorenz2, [10.0, 28.0, 8/3, 0.5],
    SimulatedAnnealing(), Optim.Options(iterations = 1000))
```

oLorenzTuplaSint = ▶ [12.3328, 30.1128, 0.107763, 0.0148413]

```
1 oLorenzTuplaSint = oLorenzSint.minimizer
```

Trayectoria del Sistema



6.2.3 Ajuste con Datos Sintéticos 2:

Dado que tenemos ya unas funciones que aproximan, con cierto sentido físico a nuestros datos reales, vamos a realizar un ajuste de datos con estas funciones, de tal manera que se acerquen más a nuestros datos originales, lo que realizaremos a continuación y compararemos con nuestros datos reales, antes y después del promedio.

Además, como en nuestras funciones sintéticas cada una de las variables tenía un razonamiento físico, para nuestro ajuste lo interpretaremos acorde a esos mismos criterios.

Para estas funciones de ajuste se usará el sufijo **Ajus**, y su valor inicial para la función de optimización serán los valores que dentro de las hipótesis son óptimos:

Presión:

```
►[734.218, 111.323, 0.315459, 0.0452374, 0.00196557]
```

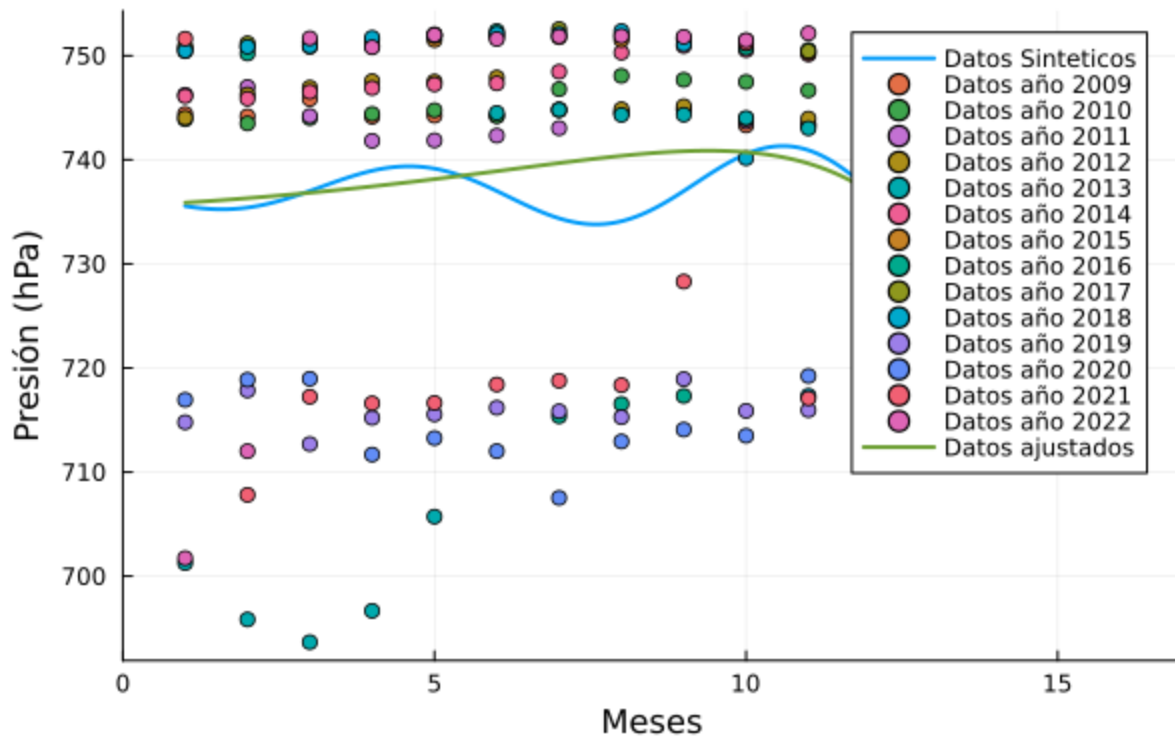
```
1 begin
2     # Función objetivo para ajuste
3     function residuoP(params, mesesDis2, datos) # residuo de la presion
4         errores = 0.0
5
6         for i in 1:length(mesesDis2)
7             # Evaluar la temperatura predicha para cada mes
8             prediccion = PDatos(mesesDis2[i], params)
9             # Calcular el error cuadrado entre la predicción y el dato real
10            errores += (prediccion - datos[i]).^2
11        end
12
13        return errores
14    end
15
16    function rP(parametros)
17        return residuoP(parametros, mesesDis2, PPorAños)
18    end
19
20
21    # Ajustamos la función a los datos
22    POptim = optimize(rP, PParamSint, NelderMead())
23
24    # Extraemos los parámetros ajustados
25    PParamAjus = POptim.minimizer
26    #println("Parámetros ajustados: ", params_ajustados)
27
28    PP0, PA, Pk, Pomega, Pphi = PParamAjus
29 end
```

```
PDatosAjus =
```

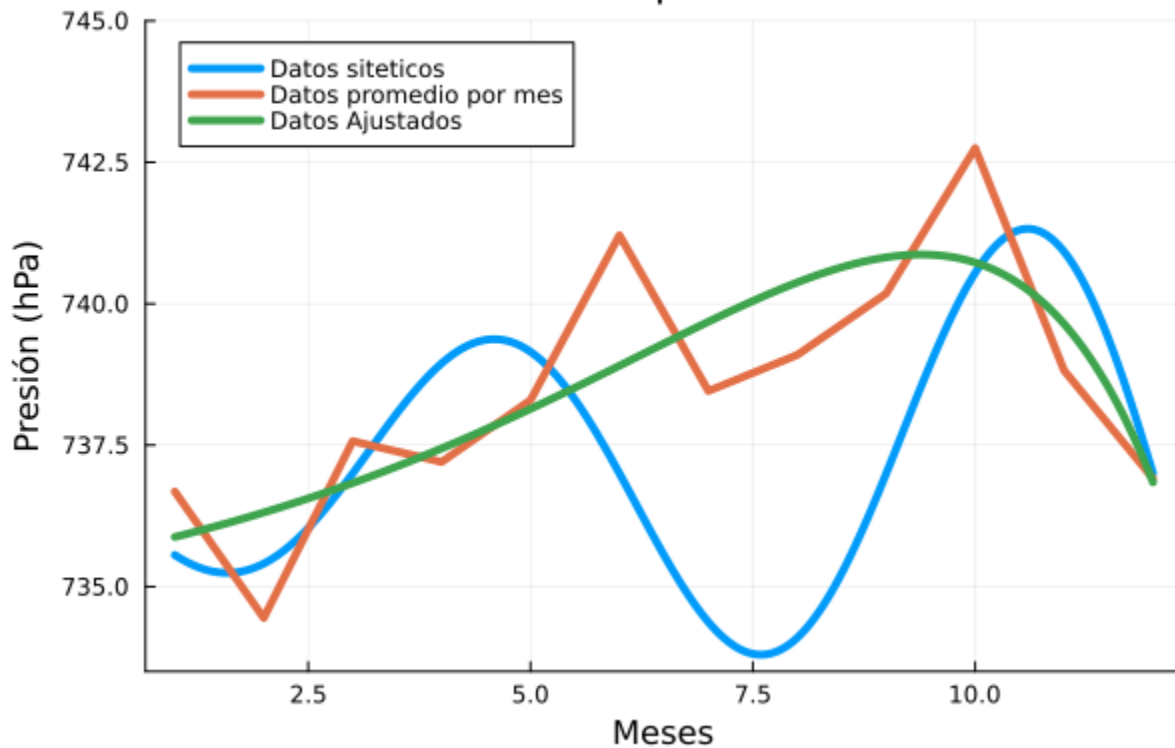
```
►[735.877, 735.877, 735.878, 735.878, 735.879, 735.879, 735.879, 735.88, 735.88, 735.88, 73
```

```
1 PDatosAjus = [PDatos(m, PParamAjus) for m in mesesCon]
```

Modelo de Presión en Bogotá



Datos presión



Dado entonces nuestro ajuste tenemos los siguientes valores:

- Presión base: $P_0 = 734.2175608509293$.
- Amplitud de las oscilaciones: $A = 111.32299495417772$
- coeficiente de amortiguación: $k = 0.31545877036024894$

- frecuencia de repetición: $\omega = 0.045237385390267955$
- Fase inicial: $\phi = 0.0019655729093168302$

Dentro de este ajuste, podemos resaltar principalmente la amplitud de las oscilaciones, y la frecuencia de repetición, que cambiaron drásticamente, ya que la amplitud de las oscilaciones subió, mientras que la frecuencia de repetición, disminuyó bruscamente. Sin embargo, estos son factores que se multiplican, así un mejor acercamiento posible sería evitar la aparición de la función sinusoidal en este caso.

Además viendo los datos individuales, los ajustes que tenemos en esta variable en particular no deben ser tan fiables, puesto que hay una gran brecha entre **720** y **740** mPa donde no hay casi ningún dato original.

Temperatura:

```

1 begin
2     # Función objetivo para ajuste
3     function residuoT(params, mesesDis2, datos)
4         error = 0.0
5
6         for i in 1:length(mesesDis2)
7             # Evaluar la temperatura predicha para cada mes
8             prediccion = TDatos(mesesDis2[i], params)
9             # Calcular el error cuadrado entre la predicción y el dato real
10            error += (prediccion - datos[i]).^2
11        end
12
13        return error
14    end
15
16    function rT(parametros)
17        return residuoT(parametros, mesesDis2, TPorAños)
18    end
19
20    TOptim = optimize(rT, TParamSint, NelderMead())
21
22    # Extraemos los parámetros ajustados
23    TParamAjus = TOptim.minimizer
24    #println("Parámetros ajustados: ", params_ajustados)
25    TA, Tmu, Tsigma, TB, TC = TParamAjus
26 end;

```

TDatosAjus =

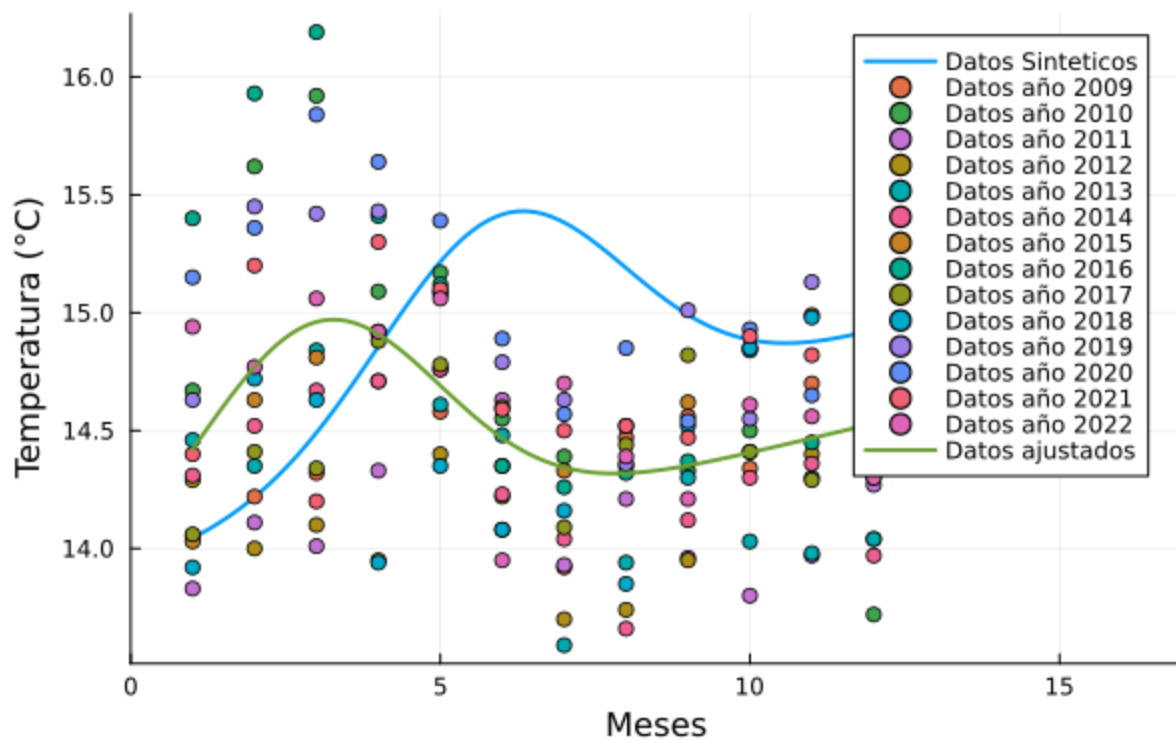
► [14.4215, 14.4218, 14.4222, 14.4226, 14.423, 14.4233, 14.4237, 14.4241, 14.4244, 14.4248,

```

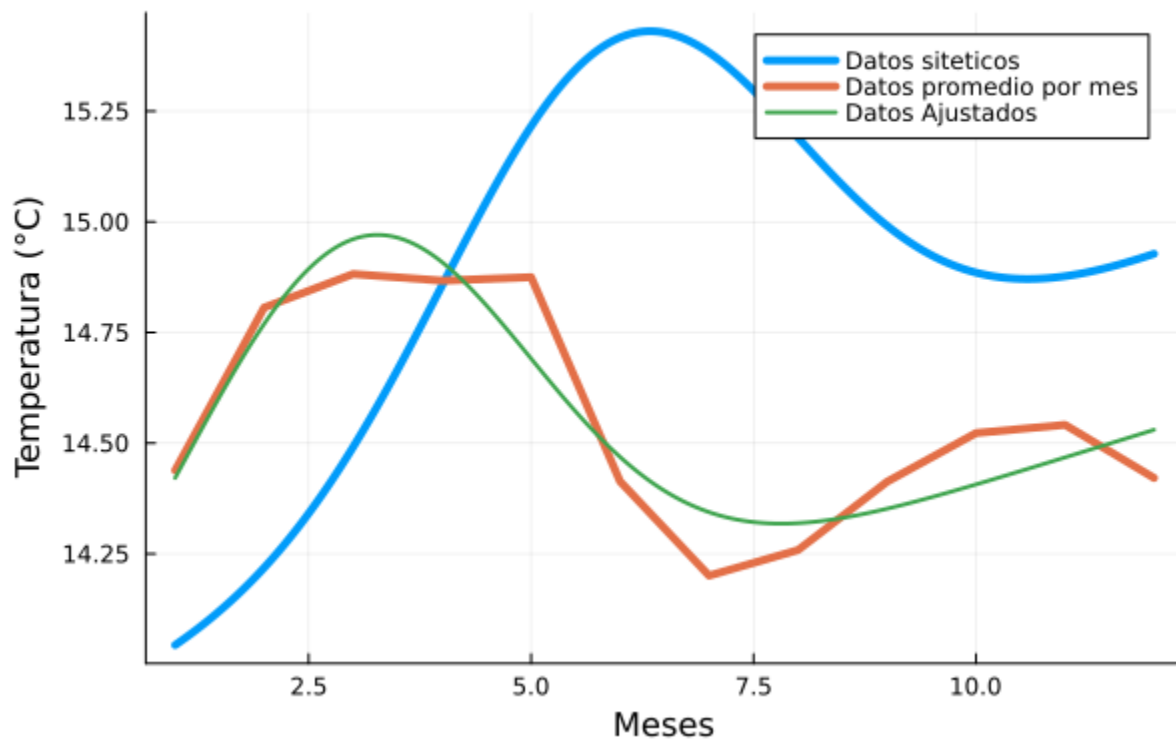
1 TDatosAjus = [TDatos(m, TParamAjus) for m in mesesCon]

```

Modelo de Temperatura en Bogotá



Datos temperatura



Con el ajuste en la temperatura tenemos los siguientes coeficientes:

- Fluctuaciones de temperatura: $A = 0.9937444509838316$
- Mes de máxima temperatura: $\mu = 3.0315365247522035$
- Desviación estandar de la distribución: $\sigma = 1.955401834533046$

- Aumento de temperatura por año: $B = 0.7504593226366318$
- Temperatura base en diciembre: $C = 13.842199156958287$

Para la temperatura, las fluctuaciones, la desviación estándar y la temperatura base se mantuvieron igual. Lo que cambió respecto a nuestras hipótesis fue: qué tanto está aumentando la temperatura cada año (se estimo en **1.0** y terminó siendo alrededor de **0.75**) y aún más importante, cuál es el mes mas cálido del año (El cual resultó siendo marzo)

Al mirar la grafica podemos ver que pese a haber una gran variación de datos la función parece estar encajada en medio de ellos. Pero aún mas importante, que es una buena aproximación a los datos promedio por mes, a diferencia de nuestra función sintética original.

Precipitación:

► [80.9173, 4.22839, 73.6305]

```

1 begin
2     # Función objetivo para ajuste
3     function residuoR(params, mesesDis2, datos)
4         error = 0.0
5
6         for i in 1:length(mesesDis2)
7             # Evaluar la temperatura predicha para cada mes
8             prediccion = RDatos(mesesDis2[i], params)
9             # Calcular el error cuadrado entre la predicción y el dato real
10            error += (prediccion - datos[i]).^2
11        end
12
13        return error
14    end
15
16    function rR(parametros)
17        return residuoR(parametros, mesesDis2, RPorAños)
18    end
19
20    ROptim = optimize(rR, RParamSint, NelderMead())
21
22    # Extraemos los parámetros ajustados
23    RParamAjus = ROptim.minimizer
24    #println("Parámetros ajustados: ", params_ajustados)
25 end

```

RDatosAjus =

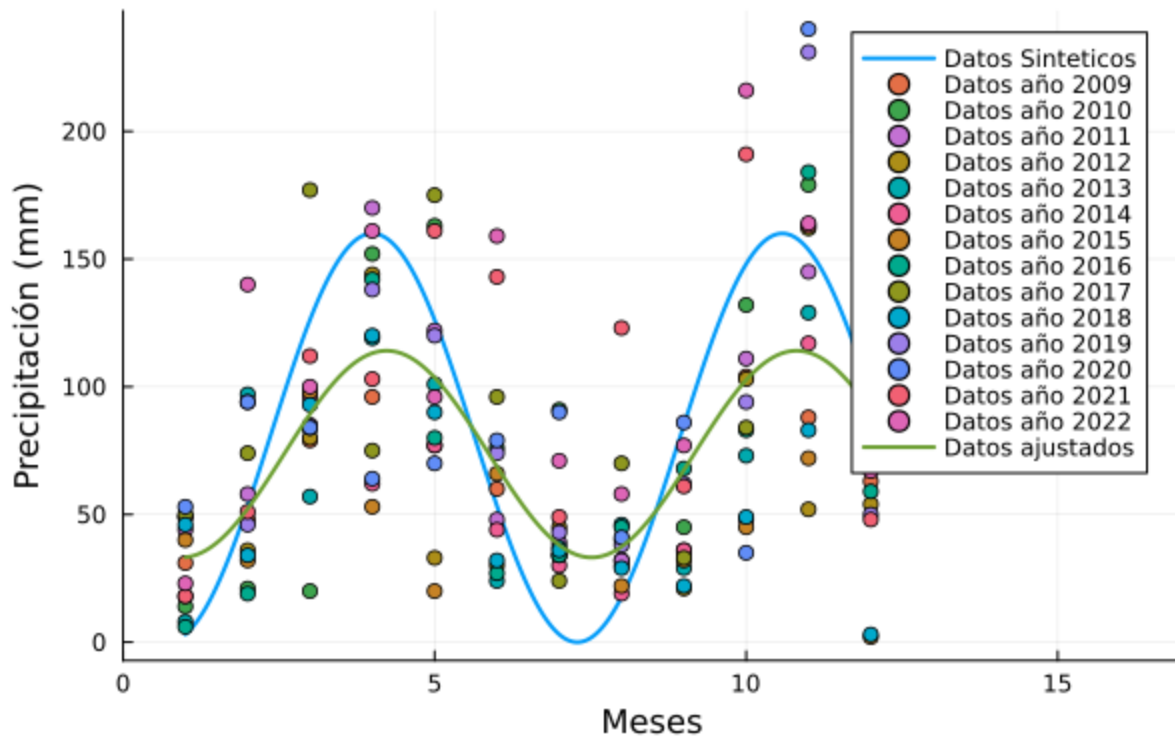
► [33.2415, 33.2438, 33.2461, 33.2485, 33.2509, 33.2533, 33.2558, 33.2583, 33.2608, 33.2634]

```

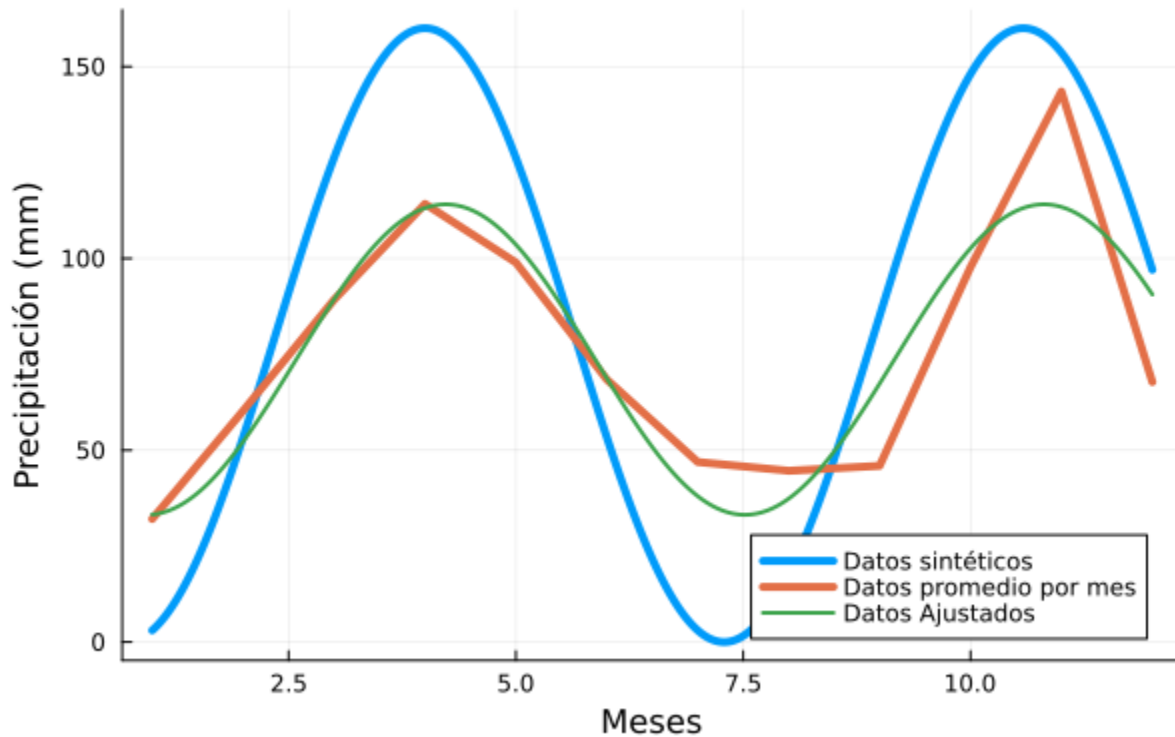
1 RDatosAjus = [RDatos(m, RParamAjus) for m in mesesCon]

```

Modelo de Precipitación en Bogotá



Datos Precipitación en Bogotá



► [80.9173, 4.22839, 73.6305]

1 RA, Rmu, RB = RParamAjus

Dentro del ajuste de las precipitaciones tenemos los siguientes datos:

- Rango base de la precipitación: $A = 80.91729544778048$
- ubicación del primer pico de lluvias: $\mu = 4.228388936815042$
- Precipitación mínima: $B = 73.63046358381226$

Al ver las graficas (y nuestras hipótesis iniciales) podemos llegar a dos conclusiones principales.

La primera es que nuestras hipótesis no estaban tan alejadas de nuestros valores más cercanos, donde teníamos una precipitación menos drástica por meses de lo que supusimos al comienzo, los meses de lluvia tienden hacia mitad de abril y octubre, y la precipitación mínima es algo menor de lo que supusimos.

La segunda conclusión es que la función sintética planteada ajusta los datos de una manera muy cercana, esto se puede apreciar sobre todo viendo como se comporta con el promedio de los datos.

Ajuste de la EDO:

Dadas nuestras funciones de aproximación, vamos a analizar el desarrollo de la ecuación diferencial

```
datosAjus = 11001x3 Matrix{Float64}:
 735.877  14.4215  33.2415
 735.877  14.4218  33.2438
 735.878  14.4222  33.2461
 735.878  14.4226  33.2485
 735.879  14.423   33.2509
 735.879  14.4233  33.2533
 735.879  14.4237  33.2558
      ⋮
 736.863  14.5298  90.7687
 736.859  14.5299  90.7337
 736.855  14.53    90.6987
 736.851  14.53    90.6636
 736.847  14.5301  90.6286
 736.843  14.5301  90.5935
```

```
V0Ajus = ▶ [735.877, 14.4215, 33.2415]
```

```
1 V0Ajus = [PDatosAjus[1], TDatosAjus[1], RDatosAjus[1]] #Condiciones iniciales para
el sistema de Lorenz
```

```
rLorenz3 (generic function with 1 method)
```

```
1 rLorenz3(params) = residuoLorenz2(params, datosAjus, mesesCon, V0Ajus)
```

```

oLorenzAjus = * Status: failure (reached maximum number of iterations)

* Candidate solution
  Final objective value:      5.516834e+04

* Found with
  Algorithm:      Simulated Annealing

* Convergence measures
  |x - x'|          = NaN  $\neq$  0.0e+00
  |x - x'|/|x'|      = NaN  $\neq$  0.0e+00
  |f(x) - f(x')|      = NaN  $\neq$  0.0e+00
  |f(x) - f(x')|/|f(x')| = NaN  $\neq$  0.0e+00
  |g(x)|             = NaN  $\neq$  1.0e-08

* Work counters
  Seconds run:      1 (vs limit Inf)
  Iterations:      1000
  f(x) calls:      1001

```

```

1 oLorenzAjus = Optim.optimize(rLorenz3, [10.0, 28.0, 8/3, 0.5],
  SimulatedAnnealing(), Optim.Options(iterations = 1000))

```

```

oLorenzTuplaAjus = ▶ [12.7507, 26.3575, 2.54987, 0.00200813]

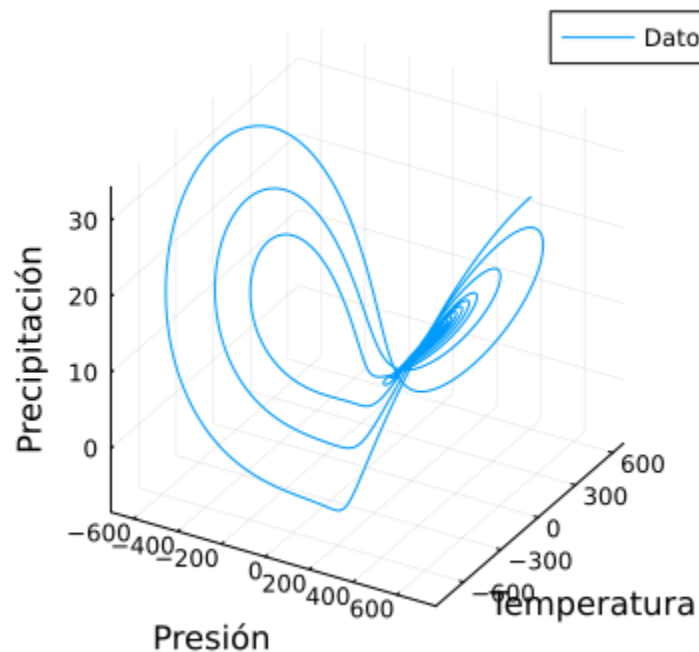
```

```

1 oLorenzTuplaAjus = oLorenzAjus.minimizer

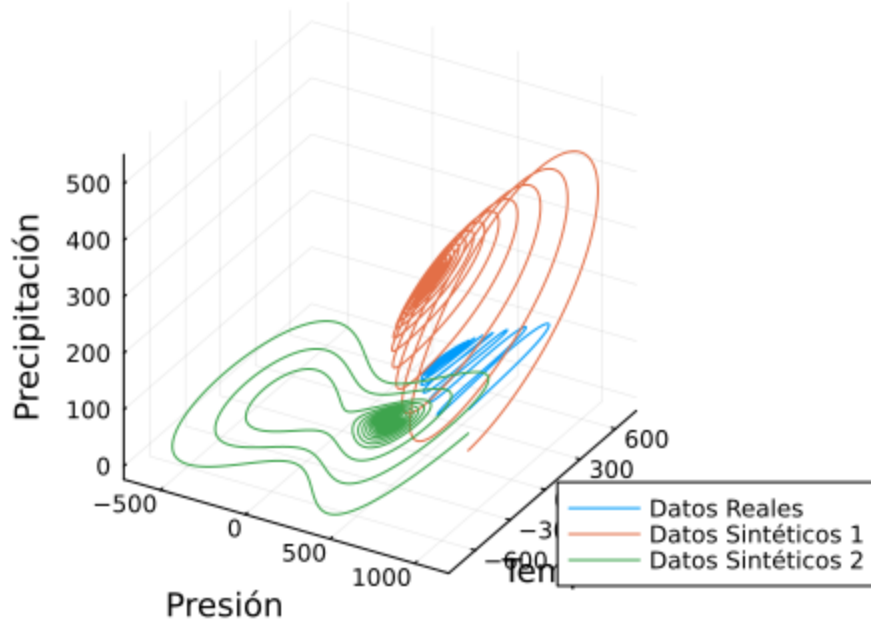
```

Trayectoria del Sistema



Finalmente, podemos observar todas las trayectorias resultantes en la siguiente figura:

Trayectorias del Sistema



7. Análisis del Modelo

Analizaremos a continuación el comportamiento general del modelo a partir de uno de los ajustes obtenidos previamente (es decir no necesariamente el de la presente ejecución). Sea nuestra Ecuación Diferencial Ordinaria

$$\begin{aligned}x' &= 16.4875y - x \\y' &= 28.9506x - y - 1.22764xz \\z' &= 0.00148262xy - z\end{aligned}$$

Para visualizar los retratos de fase de la ecuación, primero hallemos la matriz asociada dada por el Jacobiano:

$$\begin{aligned}D &= \begin{pmatrix} \frac{\partial(16.4875y-x)}{\partial x} & \frac{\partial(16.4875y-x)}{\partial y} & \frac{\partial(16.4875y-x)}{\partial z} \\ \frac{\partial(28.9506x-y-1.22764xz)}{\partial x} & \frac{\partial(28.9506x-y-1.22764xz)}{\partial y} & \frac{\partial(28.9506x-y-1.22764xz)}{\partial z} \\ \frac{\partial(0.00148262xy-z)}{\partial x} & \frac{\partial(0.00148262xy-z)}{\partial y} & \frac{\partial(0.00148262xy-z)}{\partial z} \end{pmatrix} \\ &= \begin{pmatrix} -1 & 16.4875 & 0 \\ 28.9506 - 1.22764z & -1 & -1.22764x \\ 0.00148262y & 0.00148262x & -1 \end{pmatrix}\end{aligned}$$

Ahora encontremos los puntos de equilibrio resolviendo el siguiente sistema de ecuaciones:

$$\begin{aligned}
16.4875y - x &= 0 \\
28.9506x - y - 1.22764xz &= 0 \\
0.00148262xy - z &= 0
\end{aligned}$$

Por lo tanto, se tienen tres puntos fijos:

$$\begin{aligned}
p_1 &= (-511.5644, -31.0274, 23.5329) \\
p_2 &= (0, 0, 0) \\
p_3 &= (511.5644, 31.0274, 23.5329)
\end{aligned}$$

Veamos que comportamiento tiene cada punto: Para p_1 se tiene que:

$$D = \begin{pmatrix} -1 & 16.4875 & 0 \\ 0.0607 & -1 & 628.017 \\ -0.046 & -0.7585 & -1 \end{pmatrix}$$

Ahora calculamos los valores propios de esta matriz:

$$\det \begin{pmatrix} -1 - \lambda & 16.4875 & 0 \\ 0.0607 & -1 - \lambda & 628.017 \\ -0.046 & -0.7585 & -1 - \lambda \end{pmatrix} = 0$$

Así, $\lambda_1 = -2$ y $\lambda_2 \approx -0.5 \pm 21.8191i$. Al igual, los vectores propios son los siguientes respectivamente:

$$\begin{aligned}
\vec{a} &= \begin{pmatrix} -0.9888 + 0.1366i \\ 0.06 - 0.0083i \\ 4.81 \times 10^{-18} - 6.65 \times 10^{-19}i \end{pmatrix} \\
\vec{b} &= \begin{pmatrix} 0.0861 + 0.7499i \\ 0.9950 - 0.0912i \\ -0.0024 - 0.0347i \end{pmatrix} \\
\vec{c} &= \begin{pmatrix} -0.7328 - 0.1884i \\ 0.2271 - 0.9755i \\ 0.0341 + 0.0071i \end{pmatrix}
\end{aligned}$$

Ahora como para todos los valores propios, $Re(\lambda) < 0$, se tendrá en el punto un atractor estable en espiral. Las trayectorias convergen hacia este punto con oscilaciones amortiguadas.

Para p_2 se tiene que:

$$D = \begin{pmatrix} -1 & 16.4875 & 0 \\ 28.9506 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Ahora calculamos los valores propios de esta matriz:

$$\det \begin{pmatrix} -1 - \lambda & 16.4875 & 0 \\ 28.9506 & -1 - \lambda & 0 \\ 0 & 0 & -1 - \lambda \end{pmatrix} = 0$$

Así, $\lambda_1 = -22.8477$, $\lambda_2 = 20.8477$ y $\lambda_3 = -1$. Al igual, los vectores propios son los siguientes respectivamente:

$$\vec{a} = \begin{pmatrix} 0.6024 \\ -0.7982 \\ 0 \end{pmatrix},$$

$$\vec{b} = \begin{pmatrix} 0.6264 \\ 0.83 \\ 0 \end{pmatrix},$$

$$\vec{c} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Ahora como hay un valor propio positivo indica que hay una dirección en la que las soluciones divergen, mientras que los valores negativos indican que en otras direcciones las soluciones tienden a este punto. Esto sugiere que el punto es un silla inestable, con trayectorias que se alejan en algunas direcciones y se acercan en otras.

Para p_3 se tiene que:

$$D = \begin{pmatrix} -1 & 16.4875 & 0 \\ 0.0607 & -1 & -628.017 \\ 0.046 & 0.7585 & -1 \end{pmatrix}$$

Ahora calculamos los valores propios de esta matriz:

$$\det \begin{pmatrix} -1 - \lambda & 16.4875 & 0 \\ 0.0607 & -1 - \lambda & -628.017 \\ 0.046 & 0.7585 & -1 - \lambda \end{pmatrix}$$

Así, $\lambda_1 = -2$ y $\lambda_2 \approx -0.5 \pm 21.8191i$. Al igual, los vectores propios son los siguientes respectivamente:

$$\vec{a} = \begin{pmatrix} -0.9888 + 0.1366i \\ 0.06 - 0.0083i \\ 4.81 \times 10^{-18} - 6.65 \times 10^{-19}i \end{pmatrix}$$

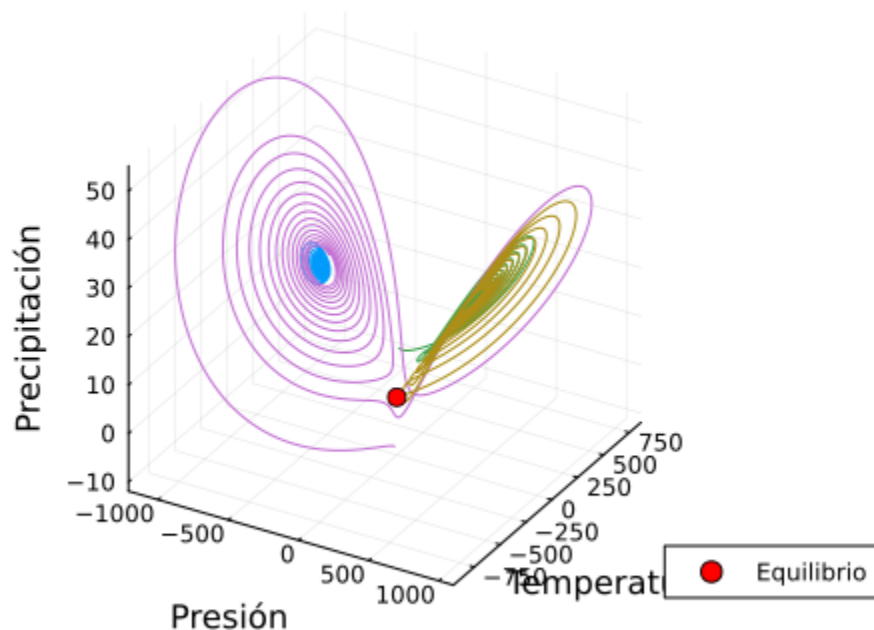
$$\vec{b} = \begin{pmatrix} 0.0861 + 0.7499i \\ 0.9950 - 0.0912i \\ 0.0024 + 0.0347i \end{pmatrix}$$

$$\vec{c} = \begin{pmatrix} 0.7328 + 0.1884i \\ -0.2271 + 0.9755i \\ 0.0341 + 0.0071i \end{pmatrix}$$

Ahora como para todos los valores propios, $Re(\lambda) < 0$, se tendrá en el punto un atractor estable en espiral. Las trayectorias convergen hacia este punto.

En conclusión, se espera que el sistema tenga una dinámica bifurcada entre dos regiones de estabilidad y un punto inestable que separa estas regiones, lo que significa que el sistema dinámico presenta dos regiones donde las soluciones tienden a estabilizarse (atractores estables) y una región donde las soluciones son inestables (un punto silla).

Retrato de Fase del Sistema



7.1 Interpretación cualitativa de los parámetros en el sistema de ecuaciones dado por aproximación de datos

reales

Valores de los parámetros en el sistema de ecuaciones:

$$\lambda = 16.4875:$$

Un valor alto indica que la presión tiene una gran influencia en la temperatura. En Bogotá, la presión atmosférica varía debido a la altitud y los cambios climáticos frecuentes, lo que sugiere que estos cambios afectan fuertemente la temperatura.

$$\rho = 28.9506:$$

Su valor elevado indica que la temperatura responde fuertemente a los cambios en la presión. Esto es coherente con Bogotá, donde la temperatura puede variar bruscamente según la presión atmosférica y la humedad presente.

$$\mu = 1.22764:$$

Su valor mayor a 1 indica que la disipación de calor es significativa y ocurre a una tasa apreciable. Esto se alinea con el clima de Bogotá, donde las lluvias tienden a reducir rápidamente la temperatura.

$$\nu = 0.00148262:$$

Su valor muy pequeño sugiere que el transporte de calor en la atmósfera de Bogotá es lento. Esto es esperable en una ciudad con una atmósfera estable y sin grandes corrientes de convección masivas como en zonas costeras.

8. Conclusiones.

Los datos pueden refinarse y ajustarse de manera más precisa para mejorar la calidad del análisis. Dado que el modelo es caótico, reiniciar el algoritmo genera resultados significativamente diferentes en cada ejecución. Los atractores de Lorenz presentan una estructura fascinante y visualmente impresionante. Aunque la selección de la función de ajuste pueda tener una justificación física inicial, no siempre se adapta correctamente a los datos, por lo que es necesario estar preparados para reconsiderarla.

Partiendo de la premisa de que el modelo nunca será exacto, pero algunos enfoques pueden resultar más útiles que otros, este desarrollo ha logrado una función de ajuste para la precipitación con un desempeño excepcional, una función de ajuste para la temperatura que, aunque buena, aún tiene margen de mejora, y retratos de fase que revelan puntos de estabilidad hacia los cuales convergen las soluciones, lo que sugiere la existencia de estados climáticos estables. Sin embargo, aún faltan coeficientes adecuados para describirlos con precisión.

Pequeños detalles en la implementación pueden generar grandes diferencias en los resultados. Por ejemplo, si los saltos de iteración son demasiado pequeños, el modelo podría no ser capaz de mostrar los característicos atractores de Lorenz. En este caso, el problema no sería nuestro enfoque, sino la acumulación de ceros numéricos que distorsionan el comportamiento del sistema. Como posibles mejoras y ampliaciones del proyecto, se podrían desarrollar funciones de ajuste más precisas, incorporar enfoques probabilísticos o estocásticos para el tratamiento de datos, optimizar la función iterativa para mejorar su estabilidad y precisión, comparar con otros modelos climáticos para identificar patrones comunes y mejorar la visualización de los datos para facilitar la interpretación de los resultados.

Referencias

- [1] P. E. Calderón Saavedra and V. H. Chaux M., 'Descripción del Modelo de Lorenz con aplicaciones', Nov. 2007, Accessed: Mar. 06, 2025. [Online]. Available: <http://hdl.handle.net/10784/146>
- [2] S. Represa, 'Ecuaciones de Lorenz', 2016, doi: 10.13140/RG.2.2.18508.82563.
- [3] R. H. Buitrago Puentes, 'El sistema y el atractor geométrico de Lorenz', 2010, Accessed: Mar. 06, 2025. [Online]. Available: <https://repositorio.unal.edu.co/handle/unal/7557>
- [4] 'Precipitaciones Totales Mensuales | Datos Abiertos Colombia'. Accessed: Mar. 06, 2025. [Online]. Available: https://www.datos.gov.co/Ambiente-y-Desarrollo-Sostenible/Precipitaciones-Totales-Mensuales/mb4n-6mzg/about_data
- [5] 'Presión Atmosférica | Datos Abiertos Colombia'. Accessed: Mar. 06, 2025. [Online]. Available: https://www.datos.gov.co/Ambiente-y-Desarrollo-Sostenible/Presi-n-Atmosf-rica/62tk-nxj5/data_preview
- [6] 'Temperatura en Bogotá D.C. y su relación con ENOS - Temperaturas registradas en Bogotá D.C - Datos Abiertos Bogotá'. Accessed: Mar. 06, 2025. [Online]. Available: <https://datosabiertos.bogota.gov.co/dataset/temperaturas-en-bogota-d-c/resource/c3867459-1aa7-4e16-8ae8-ca41693f7261>