Enviado por: Fecha de envío
Aceptado por: Editor Use
Publicado por: Editor Use

Tipo de artículo

Vicente Chavarría Gámez[1]

Análisis de la tasa de homicidios dolosos en México utilizando Interpolación de Lagrange y Diferenciación e Integración Numérica

[1] Departamento de Matemáticas, Escuela Superior de Física y Matemáticas, Instituto Politécnico Nacional, Ciudad de México, México

Palabras clave

Homicidio doloso, Métodos Numéricos, Series de Tiempo, Interpolación de Lagrange, Predicción (Forecasting)

Correo electrónico

vchavarriag1700@alumno.ipn.mx

Resumen

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1. Introducción

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna

sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.
Solución.

$$e^{i\theta}$$

□

2. Preliminares

2.1. Interpolación de Lagrange

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.1

Definición 1: Sean $x_0,\ x_1,\ \ldots,\ x_n$, $n+1$ números distintos y f es una función cuyos valores están dados por estos números. Entonces el único polinomio $P(x)$ de grado a lo más n existe, tal que

$$f(x_j) = P(x_j), \qquad \text{para toda } j \in \{0,\ 1,\ \ldots,\ n\}.$$

Dicho polinomio está dado por

$$P(x) = f(x_0)L_{n,0}(x) + f(x_1)L_{n,1}(x) + \cdots + f(x_j)L_{n,n}(x) = \sum_{j=0}^{n} f(x_j)L_{n,j}(x) \tag{1}$$

donde, para toda $j \in \{0,\ 1,\ \ldots,\ n\}$, se cumple

$$L_{n,j}(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)},$$

o bien,

$$L_{n,j}(x) = \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{(x-x_k)}{(x_j-x_k)}. \tag{2}$$

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Código 2.1: Implementación de los Polinomios de Lagrange en Python.

```python
import numpy as np
import pandas as pd
import math

# ========================================================
# Definición de la aproximación de Lagrange
# ========================================================
def l_nj(xs,j,z): # xs = [x0,x1,...,xn]
    # Calcula la longitud de la lista de puntos xs
    n = len(xs)
    # Inicializamos el numerador (p1) y el denominador (p2) del producto
    p1 = 1.0
    p2 = 1.0

```

```
    # ----------------------------
    # Ciclo para el producto desde k \in {0,1,2,...,n}
    for k in range(n):
        # La condición k \neq j es fundamental en la definición de L_{n,j}
        if k != j:
            p1 = p1 * (z - xs[k])     # p1 = (z-x0)*(z-x1)*...*(z-xj-1)*(z-xj+1)*...*(z-xn)
            p2 = p2 * (xs[j] - xs[k]) # p2 = (xj-x0)*(xj-x1)*...*(xj-xj-1)*(xj-xj+1)*...*(xj-xn)
    # Termina el ciclo
    # ----------------------------

    # Devuelve el cociente, que es L_{n,j}(z)
    return p1 / p2

# ========================================================
# Definición de los Polinomios de Lagrange
# ========================================================
def lagrange_polynomials(xs,ys,z): # xs = [x0,x1,...,xn], ys = [y0,y1,...,yn]
    # Inicializando las variables para poder entrar al ciclo 'for'.
    n = len(xs)      # Calcula la longitud de la lista de puntos xs.
    ls = np.zeros(n) # 'ls' almacenará los valores de L_{n,j}(z) para cada j.
    p = 0.0          # 'p' es el acumulador para la suma. p = P(z).

    # ----------------------------
    # Ciclo para la sumatoria desde j \in {0,1,2,...,n}.
    for j in range(n):
        # Calcula el j-ésimo polinomio base evaluado en z.
        ls[j] = l_nj(xs, j, z) # L_{n,j}(z).

        # Multiplica por su 'y' correspondiente (y_j) y lo suma al total.
        p = (ys[j] * ls[j]) + p
    # Termina el ciclo
    # ----------------------------

    # Devuelve el valor final de la sumatoria P(z).
    return p
```

2.2. Diferenciación Numérica

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Código 2.2: Implementación de la Aproximación de la Derivada en Python.

```
import numpy as np
import pandas as pd
import math

# ========================================================
# Definición de la aproximación para la Derivada hacia atrás
# ========================================================
```

```python
def backward_der_5steps(f,h,b):
    numerador = 25*f(b) - 48*f(b-h) + 36*f(b-(2*h)) - 16*f(b-(3*h)) + 3*f(b-(4*h))
    denominador = 12 * h
    return numerador/denominador

# =======================================================
# Definición de la aproximación para la Derivada centrada
# =======================================================
def central_der_5steps(f,h,x):
    numerador = f(x-(2*h)) - 8*f(x-h) + 8*f(x+h) - f(x+(2*h))
    denominador = 12 * h
    return numerador/denominador

# =======================================================
# Definición de la aproximación para la Derivada hacia adelante
# =======================================================
def forward_der_5steps(f,h,a):
    numerador = -25*f(a) + 48*f(a+h) - 36*f(a+(2*h)) + 16*f(a+(3*h)) - 3*f(a+(4*h))
    denominador = 12 * h
    return numerador/denominador
```

2.3. Integración Numérica

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Teorema 1: Dada una función $f \in C^{n+1}([a, b])$ y $a \le x_0 < x_1 < x_2 < \cdots < x_n \le b$, $n+1$ números distintos. Entonces para toda $x \in [a, b]$ existe un número $\xi_x \in (a, b)$, de tal forma que

$$f(x) = P_n(x) + \frac{f^{n+1}(\xi_x)}{(n+1!)} \prod_{j=0}^{n} (x - x_j)$$

donde $P_n(x) = \sum_{j=0}^{n} f(x_j) L_{n,j}(x)$ y $L_{n,j}(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{(x - x_k)}{(x_j - x_k)}$.

Teorema 2: Dada una función $f \in C([a, b])$ y dada $g : [a, b] \to \mathbb{R}_0^+$ tal que $g(x) \ge 0$ para toda $x \in [a, b]$ una función integrable. Entonces existe $c \in [a, b]$ de tal forma que

$$\int_a^b f(x) g(x)\, dx = f(c) \int_a^b g(x)\, dx$$

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Teorema 3: Dada una función $f \in C^2([a, b])$. Sea $x_0 < x_1$ tal que $x_0 = a$ y $x_1 = a + h = b$ donde $h = x_1 - x_0$. Demostrar que la Regla Trapezoidal está dada por

$$\int_{x_0}^{x_1} f(x)\, dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3 f^{(2)}(\xi_c)}{12}.$$

Demostración. De acuerdo con el Teorema 1 y la Definición 1 nuestra aproximación para la función $f(x)$ es de la forma

$$f(x) = f(x_0) L_{1,0}(x) + f(x_1) L_{1,1}(x) + E_1(x) \tag{3}$$

donde

$$L_{1,0}(x) = \frac{x - x_1}{x_0 - x_1} = -\frac{x - x_1}{h}, \tag{4}$$

$$L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - x_0}{h}, \tag{5}$$

y por el Teorema 1

$$E_1(x) = \frac{f^{(2)}(\xi_x)}{2}(x - x_0)(x - x_1) \tag{6}$$

—$E_1(x)$ es el término de error[1]—. Luego, nuestra aproximación a la integral definida de a a b es de la forma

$$\int_a^b f(x)\,dx = \int_a^b [f(x_0)L_{1,0}(x) + f(x_1)L_{1,1}(x)]\,dx + \int_a^b E_1(x)\,dx,$$

o bien

$$\int_a^b f(x)\,dx = f(x_0)\int_a^b L_{1,0}(x)\,dx + f(x_1)\int_a^b L_{1,1}(x)\,dx + \int_a^b E_1(x)\,dx$$

Por hipótesis sabemos que $a = x_0$ y $b = x_1$, obteniendo

$$\int_{x_0}^{x_1} f(x)\,dx = f(x_0)\int_{x_0}^{x_1} L_{1,0}(x)\,dx + f(x_1)\int_{x_0}^{x_1} L_{1,1}(x)\,dx + \int_{x_0}^{x_1} E_1(x)\,dx. \tag{7}$$

Ahora, sustituyendo a las Ecuaciones (4) y (5) e integrando por separado a $L_{1,0}(x)$ y $L_{1,1}(x)$. Se sigue para $L_{1,0}(x)$

$$\int_{x_0}^{x_1} L_{1,0}(x)\,dx = \int_{x_0}^{x_1} \frac{x - x_1}{h} = -\frac{1}{2h}(x - x_1)^2\Big|_{x_0}^{x_1} = -\frac{(x_0 - x_1)^2}{2h} = \frac{(x_1 - x_0)^2}{2h} = \frac{h^2}{2h} = \frac{h}{2}.$$

Análogamente, para $L_{1,1}(x)$ hacemos

$$\int_{x_0}^{x_1} L_{1,1}(x)\,dx = \int_{x_0}^{x_1} \frac{x - x_0}{h}\,dx = \frac{1}{2h}(x - x_0)^2\Big|_{x_0}^{x_1} = \frac{(x_1 - x_0)^2}{2h} = \frac{h^2}{2h} = \frac{h}{2}.$$

Sustituyendo lo anterior en la Ecuación (7) tenemos

$$\int_{x_0}^{x_1} f(x)\,dx = f(x_0)\frac{h}{2} + f(x_1)\frac{h}{2} + \int_{x_0}^{x_1} E_1(x)\,dx,$$

o bien

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] + \int_{x_0}^{x_1} E_1(x)\,dx. \tag{8}$$

Ahora, analizaremos el error. De la Ecuación (8) se sigue

$$\int_{x_0}^{x_1} E_1(x)\,dx = \int_{x_0}^{x_1} \frac{f^{(2)}(\xi_x)}{2}(x - x_0)(x - x_1)\,dx$$

Por el Teorema 2, sean $f(x) = \dfrac{f^{(2)}(\xi_x)}{2}$ y $g(x) = (x - x_0)(x - x_1)$. Entonces existe $c \in [a,\, b]$ de tal forma que $\xi_c \in (a,\, b)$. Esto es

$$\int_{x_0}^{x_1} E_1(x)\,dx = \frac{f^{(2)}(\xi_c)}{2}\int_{x_0}^{x_1} (x - x_0)(x - x_1)\,dx. \tag{9}$$

Ahora bien, para resolver la integral, procederemos integrando por partes. Sean $u = x - x_0$ y $dv = (x - x_1)\,dx$, entonces $du = dx$ y $v = \dfrac{(x - x_1)^2}{2}$. Integrando

$$\int_{x_0}^{x_1} (x - x_0)(x - x_1)\,dx = (x - x_0)\frac{(x - x_1)^2}{2}\Bigg|_{x_0}^{x_1} - \frac{1}{2}\int_{x_0}^{x_1} (x - x_1)^2\,dx$$

Al evaluar el primer término en cada límite, el resultado es 0 en ambos casos. Así,

$$\int_{x_0}^{x_1} (x - x_0)(x - x_1)\,dx = -\frac{1}{2 \cdot 3}(x - x_1)^3\Big|_{x_0}^{x_1} = -\frac{1}{6}\left(0 - (x_0 - x_1)^3\right) = -\frac{(x_1 - x_0)^3}{6} = -\frac{h^3}{6}.$$

Sustituyendo el valor de la integral en la Ecuación (9), obtenemos

$$\int_{x_0}^{x_1} E_1(x)\,dx = -\frac{f^{(2)}(\xi_c)}{2} \cdot \frac{h^3}{6},$$

o bien,

$$\int_{x_0}^{x_1} E_1(x)\,dx = -\frac{h^3 f^{(2)}(\xi_c)}{12}. \tag{10}$$

1 Dicho termino de error será tratado más adelante

Finalmente, sustituyendo la Ecuación (10) en la Ecuación (8)

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0)+f(x_1)] - \frac{h^3 f^{(2)}(\xi_c)}{12}. \tag{11}$$

Por lo tanto, queda demostrado que la **Regla Trapezoidal** es de la forma

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0)+f(x_1)] - \frac{h^3 f^{(2)}(\xi_c)}{12}. \qquad \blacksquare$$

Código 2.3: Implementación de la Aproximación de la Integral de Riemann en Python.

```python
import numpy as np
import pandas as pd
import math

# ========================================================
# Definición de la aproximación para la Integral de Riemann
# ========================================================
def integral_riemann_uniform(f,a,b,n):
    # Inicializando las variables para poder entrar al ciclo 'for'.
    h = (b-a)/n # Calcula el tamaño del 'paso'.
    SR = 0.0    # Inicializamos la suma en 0.
    xj = a      # Inicia el acumulador 'xj' en el punto 'a'.

    # ----------------------------
    # Ciclo que suma las alturas f(xj) para j \in {1,2,...,n}.
    for j in range(n):
        xj = xj + h  # Avanza al extremo derecho del subintervalo.
        yj =  f(xj)  # Calcula la altura en ese punto.
        SR = SR + yj # Acumula la altura.
    # Termina el ciclo.
    # ----------------------------

    # Devuelve el valor final de la sumatoria S(f,P).
    return h * SR # Calcula el área total al multiplicar por 'h'.

# ========================================================
# Definición de la aproximación para la Regla Trapezoidal
# ========================================================
def integral_trapezoidal_rule_uniform(f, a, b, n):
    # Inicializando las variables para poder entrar al ciclo 'for'.
    h = (b - a) / n          # Calcula el ancho 'h' de cada subintervalo.
    TR = (f(a) + f(b)) / 2.0 # Inicializa la suma con el promedio de los extremos f(a) y f(b).

    # ----------------------------
    # Ciclo para sumar las evaluaciones en los puntos interiores de la sumatoria.
    # Para j \in {1,2,...,n-1}.
    for j in range(1, n):
        xj = a + j * h # Calcula el j-ésimo punto (nodo) interior de la malla
        yj = f(xj)     # Evalúa la función en dicho punto
        TR = TR + yj   # Acumula la evaluación (altura) en la suma total
    # Termina el ciclo
    # ----------------------------

    # Devuelve el valor final de la aproximación.
    return h * TR # Calcula el área total multiplicando la suma acumulada por 'h'.

# ========================================================
# Definición de la aproximación para la Regla de Simpson 1/3
# ========================================================
```

```python
def integral_simpson_rule_uniform(f, a, b, n):
    # Inicializando las variables para poder entrar al ciclo 'for'.
    h = (b - a) / n # Calcula el ancho 'h' de cada subintervalo.
    SR = 0.0        # Inicializamos la sumatoria en 0.

    # ----------------------------
    # Ciclo que aplica la Regla de Simpson 1/3 simple en cada uno de los 'n' subintervalos.
    # Para j \in {1,2,...,n}
    for j in range(1, n + 1):
        # Calcula los extremos del j-ésimo subintervalo: (x_{j-1}, x_j).
        xjminus1 = a + (j - 1) * h
        xj = a + j * h
        # Calcula el punto medio de dicho subintervalo.
        mj = (xjminus1 + xj) / 2.0
        # Evalúa la fórmula de Simpson simple: f(x_{j-1}) + 4*f(m_j) + f(x_j).
        Aj = f(xjminus1) + 4 * f(mj) + f(xj)

        # Acumula la contribución del subintervalo.
        SR = SR + Aj
    # Termina el ciclo.
    # ----------------------------

    # Devuelve el valor final de la aproximación.
    return (h / 3.0) * SR # La fórmula de la regla simple es (h/3) * [...].
```

3. Resultados

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

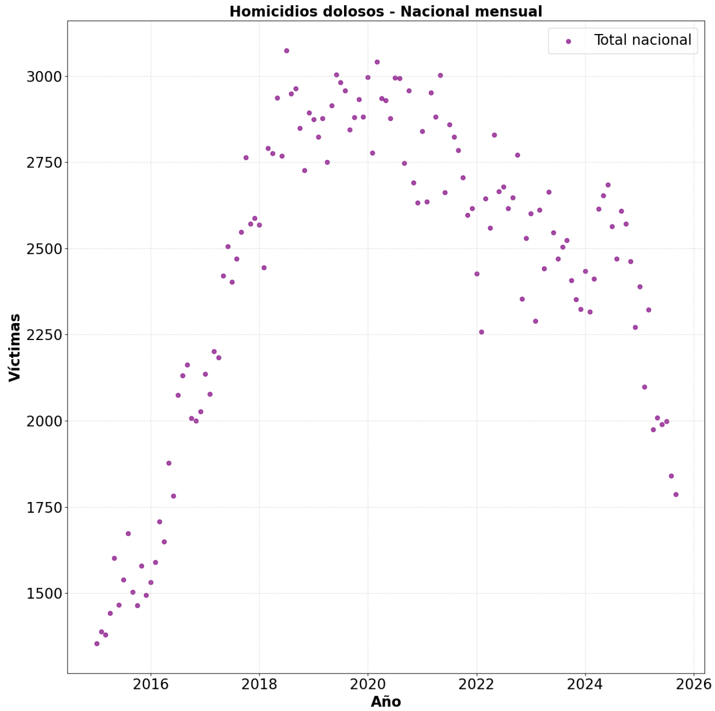
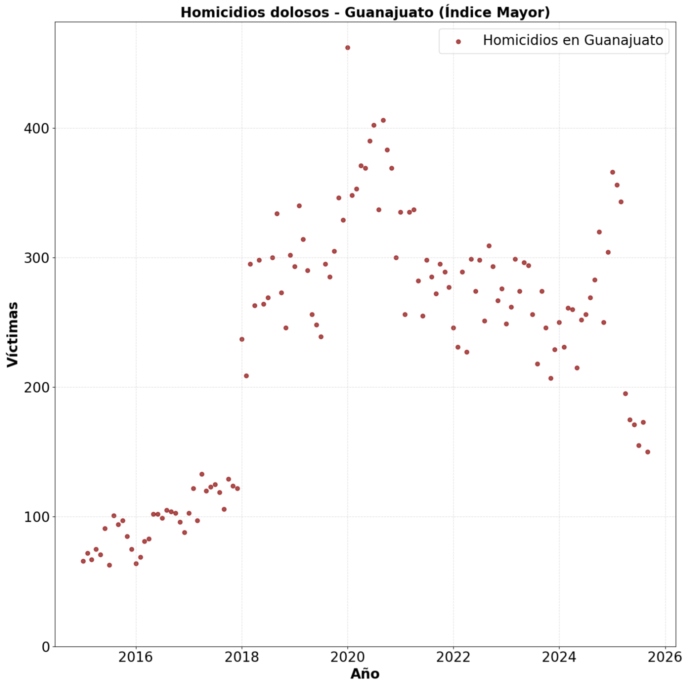
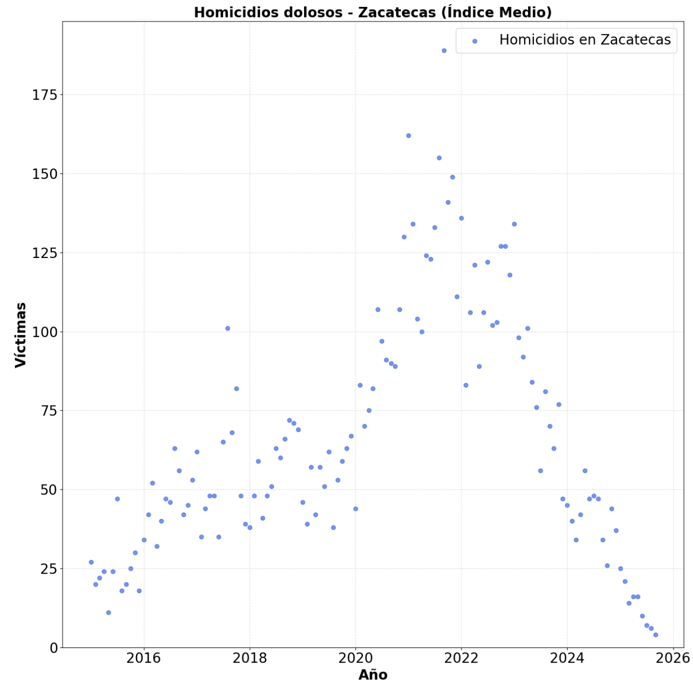


Figura 1: Gráfica de los homicidios dolosos en México desde 2015 hasta 2025.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

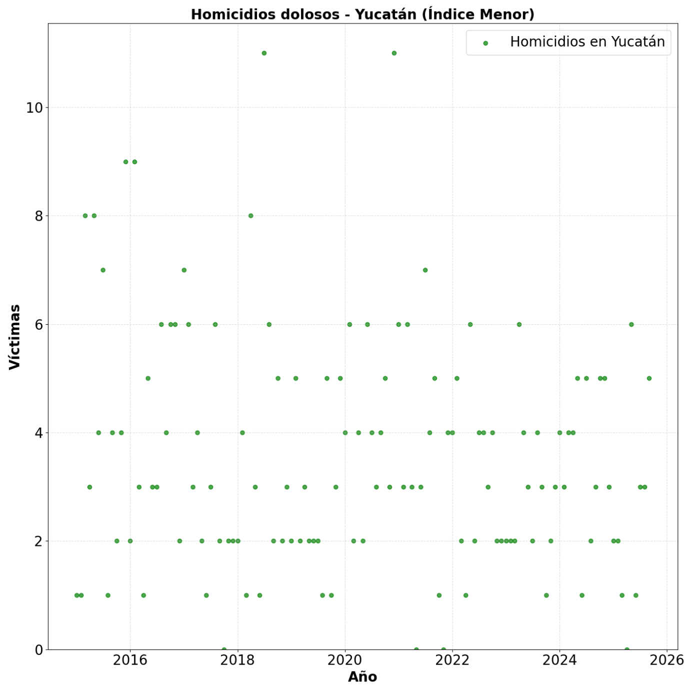
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.





a) Guanajuato tiene el mayor índice de homicidios dolosos.

b)



c)

Figura 2

4. Conclusiones

5. Discusión

6. Agradecimientos

Referencias

1. Burden, R. L. y Faires, J. D. *Numerical Analysis* 9.ª ed. (Cengage Learning, 2010).

A. Justificación Metodológica de la Aproximación Polinómica por Tramos

A.1. Limitaciones de la Interpolación Polinómica de Alto Grado

Para un conjunto de $n + 1$ puntos de datos, como la serie de tiempo `Nacional_Mensual` ($n = 128$), es teóricamente posible construir un único polinomio de grado n que pase exactamente por todos los puntos. Sin embargo, en la práctica, el uso de polinomios de interpolación de grado muy alto presenta serias desventajas numéricas.

El problema fundamental fue ilustrado por Carl Runge en 1901. El **fenómeno de Runge** demuestra que, para ciertos conjuntos de puntos (especialmente aquellos equiespaciados), el polinomio de interpolación de alto grado puede presentar **oscilaciones violentas** cerca de los extremos del intervalo. Estas oscilaciones provocan que el error de aproximación, lejos de disminuir al añadir más puntos, diverja drásticamente.

El ejemplo clásico es la función $f(x) = \frac{1}{x}$. Al interpolar esta función con un polinomio de alto grado, el polinomio resultante oscila de manera incontrolada en los extremos, fallando en representar la función real.

Los datos de este proyecto (víctimas de homicidio) son inherentemente "ruidosos" e irregulares. Intentar ajustar un único polinomio de grado 128 a los 129 puntos de la serie `Nacional_Mensual` resultaría en un modelo numéricamente inestable y un ejemplo claro del fenómeno de Runge, produciendo un polinomio con oscilaciones extremas y nulo valor predictivo.

A.2. Aproximación Polinómica por Tramos (Piecewise-Polynomial Approximation)

Para mitigar los problemas de la interpolación de alto grado, la estrategia numéricamente robusta es la **Aproximación Polinómica por Tramos**.

Este método, descrito en textos de análisis numérico como Burden & Faires (9ª ed., Sección 3.5), consiste en dividir el intervalo completo de datos en una serie de sub-intervalos más pequeños (o "tramos"). En lugar de un solo polinomio de alto grado, se construye un polinomio de **grado bajo** (e.g., grado 3, 5, o 9) para cada tramo.

Las ventajas de este enfoque son significativas:

1. **Evita el Fenómeno de Runge:** Al utilizar exclusivamente polinomios de grado bajo, se eliminan las oscilaciones de alto grado.
2. **Adaptabilidad Local:** Cada polinomio se ajusta a la tendencia *local* de los datos en su tramo. Esto es ideal para series de tiempo, permitiendo que el modelo se adapte a picos, valles y cambios de tendencia sin distorsionar la aproximación global.
3. **Estabilidad Numérica:** El cálculo de varios polinomios de grado bajo es computacionalmente más simple y menos susceptible a errores de redondeo que un solo sistema de alto grado.

A.3. Aplicación en este Proyecto

Este proyecto implementó una **Aproximación Polinómica por Tramos** (Sección 3). La serie de tiempo `Nacional_Mensual` se dividió en tramos consecutivos de 10 puntos, y para cada tramo se construyó un polinomio de interpolación de Lagrange de grado 9.

Este enfoque permite un análisis local preciso (necesario para las derivadas, Sección 4) y una integración robusta (Sección 5), sacrificando la continuidad global de las derivadas (que no es un objetivo de este análisis) en favor de la estabilidad y la precisión local.

Una evolución de este método son los **Trazadores Cúbicos (Cubic Splines)**, los cuales son polinomios por tramos de grado 3 que imponen condiciones de continuidad en la primera y segunda derivada (C^2), garantizando una curva "suave" en los puntos de unión (nodos). Sin embargo, para los fines de este trabajo, la aproximación por tramos de Lagrange es la metodología especificada y aplicada.