

Artículo de Investigación

¹Departamento 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

Plantilla desarrollada por **Vicente C. Gámez**
© El Autor. Esta plantilla es publicada bajo la licencia CC-BY:
<https://creativecommons.org/licenses/by/4.0/>

Vicente Chavarría Gámez¹

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

Resumen

Este proyecto aborda el análisis numérico de una serie de datos discretos mediante técnicas de interpolación, diferenciación e integración numérica. Inicialmente se estudia la interpolación polinómica global, analizando su comportamiento y limitaciones, en particular la inestabilidad asociada al fenómeno de Runge. Como alternativa, se introduce el uso de splines cúbicos naturales, los cuales permiten construir un modelo continuo con mayor estabilidad y suavidad.

A partir de dicho modelo se aplican métodos de diferenciación numérica de orden alto y reglas de cuadratura compuesta para aproximar derivadas e integrales definidas. Se presenta tanto el fundamento teórico de los métodos empleados como su implementación computacional, y se comparan los resultados obtenidos con las predicciones teóricas en términos de orden de convergencia y estabilidad. Los resultados muestran que el uso de splines cúbicos proporciona una base numéricamente robusta para el análisis de datos discretos y mejora significativamente el desempeño de los métodos de diferenciación e integración numérica.

1. Introducción

En numerosos problemas aplicados se dispone únicamente de información en forma de datos discretos, ya sea proveniente de mediciones experimentales, registros estadísticos o simulaciones computacionales. El análisis de este tipo de información requiere el uso de métodos numéricos que permitan aproximar funciones continuas, así como calcular derivadas e integrales de manera confiable a partir de un conjunto finito de puntos.

En este contexto, la interpolación constituye una herramienta fundamental. Sin embargo, la interpolación polinómica global puede presentar serias dificultades numéricas, especialmente cuando se emplean polinomios de alto grado o cuando los datos no están uniformemente distribuidos. Estas dificultades se manifiestan en fenómenos de inestabilidad y oscilaciones no deseadas (fenómeno de Runge), lo que limita su utilidad práctica en el análisis de datos reales con alta variabilidad.

Como alternativa, los splines cúbicos ofrecen una aproximación por tramos que garantiza suavidad y continuidad en las derivadas, al mismo tiempo que mejora la estabilidad numérica del interpolante. Este tipo de interpolación resulta particularmente adecuado cuando se desea construir un modelo continuo que sirva de base para la aplicación de otros métodos numéricos, como la diferenciación para estimar tasas de cambio locales y la integración para calcular acumulados.

El objetivo de este proyecto es analizar y comparar estos métodos desde un punto de vista tanto teórico como computacional, aplicándolos al estudio de la **tasa de homicidios dolosos en México** durante el periodo 2015–2025. Para ello, se utilizan los registros oficiales del Secretariado Ejecutivo del Sistema Nacional de Seguridad Pública [1]. Se contrastará el comportamiento de la interpolación polinómica global frente a los splines cúbicos naturales, evaluando posteriormente el desempeño de la diferenciación numérica para identificar puntos críticos de violencia y de las reglas de cuadratura para cuantificar el impacto social acumulado. A través del análisis del error, se busca identificar las ventajas de elegir métodos numéricamente estables para el tratamiento de series de tiempo sociológicas desde una perspectiva de análisis numérico.

2. Preliminares

2.1 Interpolación Polinómica de Lagrange

El problema de la interpolación polinómica consiste en determinar una función dentro de un espacio vectorial de dimensión finita que coincide con un conjunto discreto de valores prescritos [2].

Comenzamos estableciendo la existencia y unicidad de la solución para el caso polinomial.

Teorema 2.1 (Existencia y Unicidad del Polinomio Interpolador): Sean $x_0, x_1, \dots, x_n, n + 1$ números reales distintos y y_0, y_1, \dots, y_n un conjunto de valores asociados. Existe un único polinomio $P_n(x)$ de grado a lo más n (perteneciente al espacio \mathcal{P}_n) tal que satisface las condiciones de interpolación:

$$P_n(x_i) = y_i, \quad \forall i \in \{0, 1, \dots, n\}.$$

Para expresar este polinomio de forma directa, utilizamos la fórmula de Lagrange:

Definición 2.1: El polinomio interpolador único $P_n(x)$ puede expresarse como una combinación lineal de los polinomios base de Lagrange $L_{n,k}(x)$:

$$P_n(x) = \sum_{k=0}^n f(x_k) L_{n,k}(x), \quad (1)$$

donde cada función base $L_{n,k}(x)$ se define mediante el producto:

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

Nótese que esta base satisface la propiedad de delta de Kronecker: $L_{n,k}(x_i) = \delta_{ki}$.

La precisión de esta aproximación depende de las propiedades analíticas de la función generadora f y de la distribución de los nodos x_i [2].

Teorema 2.2 (Resto de la Interpolación de Lagrange): Sea $f \in C^{n+1}([a, b])$ y sean x_0, \dots, x_n nodos distintos en $[a, b]$. Para cualquier $x \in [a, b]$, existe un número $\xi(x) \in (a, b)$ tal que el error de interpolación viene dado por:

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i). \quad (3)$$

Observación 1 (Comportamiento Asintótico del Error): La ecuación del teorema anterior pone de manifiesto que la convergencia uniforme de $P_n(x)$ hacia $f(x)$ cuando $n \rightarrow \infty$ no está garantizada para cualquier elección de nodos (Fenómeno de Runge).

A efectos de validación computacional, se desarrolló un algoritmo que construye explícitamente los polinomios base $L_{n,j}(x)$. Esta implementación se detalla en el Código B.1 del Apéndice B.

2.2 Splines Cúbicos

A diferencia de la interpolación global de Lagrange, los **Splines** cúbicos construyen la aproximación mediante polinomios de tercer grado definidos por tramos [2], garantizando suavidad global.

Definición 2.2 (Spline Cúbico): Dada una partición del intervalo $[a, b]$, una función $S(x)$ se denomina Spline Cúbico si coincide con un polinomio cúbico en cada subintervalo, interpola los datos y es de clase $C^2([a, b])$.

Para determinar los coeficientes, se imponen condiciones de frontera.

Definición 2.3 (Spline Cúbico Natural): Un Spline Cúbico se dice **Natural** si satisface las condiciones de curvatura nula en los extremos: $S''(x_0) = 0$ y $S''(x_n) = 0$.

Teorema 2.3 (Existencia y Unicidad del Spline Cúbico): Dados $n + 1$ puntos distintos, existe un único Spline Cúbico Natural $S(x)$ que interpola dichos datos, obtenido al resolver un sistema tridiagonal estrictamente diagonal dominante [2].

El algoritmo implementado para la construcción y evaluación de los Splines Cúbicos, fundamental para evitar distorsiones en el modelado de homicidios, se presenta en el Código B.2 del Apéndice B.

2.3 Diferenciación Numérica

Para estimar la velocidad de cambio, utilizamos métodos de diferencias finitas de alto orden [2].

Proposición 2.1 (Derivada Centrada de 5 Puntos): Sea $f \in C^5([a, b])$. La derivada se aproxima mediante:

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + O(h^4).$$

Para validar la robustez numérica, definimos métricas de error absoluto, normalizado (para verificar el orden de convergencia) y relativo. La implementación computacional de estos esquemas se encuentra en el Código **B.3** del Apéndice **B**.

Definición 2.4 (Métricas de Error Numérico): Sean $v_{real} = f(c)$ el valor de referencia (calculado analíticamente o con alta precisión) y v_{aprox} el valor numérico obtenido. Definimos:

1. Error Absoluto (e_n): La magnitud de la diferencia directa.

$$e_n = |v_{real} - v_{aprox}|.$$

2. Error Normalizado ($e_{N,h}$): Medida utilizada para verificar la constante asintótica de convergencia C . Dado que utilizamos un método de orden 4, esperamos que:

$$e_{N,h} = \frac{e_n}{h^4} \leq C.$$

3. Error Relativo (e_r): Proporción del error respecto a la magnitud de la aproximación, útil para dimensionar el error en función de la escala de los datos (víctimas) [1].

$$e_r = \frac{e_n}{|v_{aprox}|}.$$

2.4 Integración Numérica

La integración numérica permite evaluar el impacto total acumulado sustituyendo $f(x)$ por un polinomio interpolador.

Teorema 2.4: Dada una función $f \in C^{n+1}([a, b])$ y $a \leq x_0 < \dots < x_n \leq b$, $n+1$ 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 2.5: Dada una función $f \in C^2([a, b])$ y dada $g : [a, b] \rightarrow \mathbb{R}^+$ tal que $g(x) \geq 0$ para toda $x \in [a, b]$ una función integrable. Entonces existe $c \in [a, b]$ de tal forma que

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

A continuación, deducimos las fórmulas de Newton-Cotes cerradas, comenzando por la aproximación más elemental mediante un polinomio de grado cero.

Teorema 2.6 (Regla del Rectángulo): Dada una función $f \in C^1([a, b])$. Sea $x_0 < x_1$ tal que $h = x_1 - x_0$. Demostrar que la Regla del Rectángulo (Suma de Riemann por la izquierda) está dada por

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

donde $\xi \in (x_0, x_1)$.

Demostración. De acuerdo con el Teorema 2.4 y la Definición 2.1, para el caso $n = 0$, nuestra aproximación para la función $f(x)$ es de la forma

$$f(x) = f(x_0)L_{0,0}(x) + E_0(x) \quad (4)$$

donde el único polinomio base es constante (producto vacío):

$$L_{0,0}(x) = 1, \quad (5)$$

y por el Teorema 2.4, el término de error es:

$$E_0(x) = f(\xi)(x - x_0). \quad (6)$$

Luego, nuestra aproximación a la integral definida en el intervalo $[x_0, x_1]$ es de la forma

$$\int_{x_0}^{x_1} f(x) dx = f(x_0) \int_{x_0}^{x_1} L_{0,0}(x) dx + \int_{x_0}^{x_1} E_0(x) dx.$$

Sustituyendo a la Ecuación (5) e integrando el polinomio base, obtenemos directamente la longitud del intervalo:

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

Sustituyendo lo anterior en la ecuación de la integral tenemos

$$\int_{x_0}^{x_1} f(x) dx = hf(x_0) + \int_{x_0}^{x_1} E_0(x) dx. \quad (7)$$

Ahora, analizaremos el error. De la Ecuación (6) se sigue

$$\int_{x_0}^{x_1} E_0(x) dx = \int_{x_0}^{x_1} f(\xi_x)(x - x_0) dx.$$

Por el Teorema 2.5, sean $f(x) = f(\xi_x)$ y $g(x) = (x - x_0)$. Dado que $x \geq x_0$ en el intervalo de integración, $g(x) \geq 0$. Entonces existe $\xi \in (x_0, x_1)$ tal que

$$\int_{x_0}^{x_1} E_0(x) dx = f(\xi) \int_{x_0}^{x_1} (x - x_0) dx. \quad (8)$$

Resolvemos la integral restante de forma directa:

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

Sustituyendo el valor de la integral en la Ecuación (8), obtenemos

$$\int_{x_0}^{x_1} E_0(x) dx = \frac{h^2}{2} f(\xi).$$

Finalmente, sustituyendo este resultado en la Ecuación (7):

$$\int_{x_0}^{x_1} f(x) dx = hf(x_0) + \frac{h^2}{2} f(\xi).$$

Por lo tanto, queda demostrada la **Regla del Rectángulo**. ■

Si bien la aproximación rectangular es intuitiva, su error global converge lentamente. Para mejorar la precisión, aumentamos el grado del polinomio interpolador a $n = 1$, obteniendo una aproximación lineal conocida como la Regla Trapezoidal.

Teorema 2.7: Dada una función $f \in C([a, b])$. 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

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

Demostración. De acuerdo con el Teorema 2.4 y la Definición 2.1 nuestra aproximación para la función $f(x)$ es de la forma

$$f(x) = f(x_0)L_{1,0}(x) + f(x_1)L_{1,1}(x) + E_1(x) \quad (9)$$

donde

$$L_{1,0}(x) = \frac{x - x_1}{x_0 - x_1} = -\frac{x - x_1}{h}, \quad (10)$$

$$L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - x_0}{h}, \quad (11)$$

y por el Teorema 2.4

$$E_1(x) = \frac{f''(\xi_x)}{2}(x - x_0)(x - x_1) \quad (12)$$

— $E_1(x)$ es el término de error—. 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

$$\int_{x_0}^{x_1} f(x) dx = f(x_0) \int_{x_0}^{x_1} L_{1,0}(x) dx + f(x_1) \int_{x_0}^{x_1} L_{1,1}(x) dx + \int_{x_0}^{x_1} E_1(x) dx. \quad (13)$$

Ahora, sustituyendo a las Ecuaciones (10) y (11) e integrando por separado a $L_{1,0}(x)$ y $L_{1,1}(x)$. Se sigue para $L_{1,0}(x)$

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

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 (13) 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,

$$\int_{x_0}^{x_1} f(x) dx = \frac{h}{2} [f(x_0) + f(x_1)] + \int_{x_0}^{x_1} E_1(x) dx. \quad (14)$$

Ahora, analizaremos el error. De la Ecuación (14) se sigue

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

Por el Teorema 2.5, sean $f(x) = \frac{f''(\xi_x)}{2}$ 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

$$\int_{x_0}^{x_1} E_1(x) dx = \frac{f''(\xi_c)}{2} \int_{x_0}^{x_1} (x - x_0)(x - x_1) dx. \quad (15)$$

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 $v = \frac{(x - x_1)^2}{2}$. Integrando

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

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

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

Sustituyendo el valor de la integral en la Ecuación (15), obtenemos

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

o bien,

$$\int_{x_0}^{x_1} E_1(x) dx = -\frac{h^3 f''(\xi_c)}{12}. \quad (16)$$

Finalmente, sustituyendo la Ecuación (16) en la Ecuación (14)

$$\int_{x_0}^{x_1} f(x) dx = \frac{h}{2} [f(x_0) + f(x_1)] - \frac{h^3 f''(\xi_c)}{12}. \quad (17)$$

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

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

Si aumentamos el grado del polinomio interpolador a $n = 2$, obtenemos una aproximación cuadrática que captura mejor la curvatura de la función,

derivando en la Regla de Simpson.

Teorema 2.8: Dada una función $f \in C^4([a, b])$. Sean x_0, x_1, x_2 tales que $a \leq x_0 < x_1 < x_2 \leq b$, donde $h = x_1 - x_0 = x_2 - x_1$. Demostrar que la Regla de Simpson está dada por

$$\int_{x_0}^{x_2} f(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5 f^{(4)}(\xi)}{90},$$

donde $\xi \in (a, b)$.

Demostración. De acuerdo con el Teorema 2.4 y la Definición 2.1, aproximamos la función $f(x)$ en el intervalo $[x_0, x_2]$ mediante el polinomio interpolador de segundo grado. Nuestra aproximación es de la forma:

$$f(x) = f(x_0)L_{2,0}(x) + f(x_1)L_{2,1}(x) + f(x_2)L_{2,2}(x) + E_2(x), \quad (18)$$

donde los polinomios base $L_{2,k}(x)$ están definidos sobre los nodos x_0, x_1, x_2 . Integrando la Ecuación (18) en el intervalo de interés:

$$\int_{x_0}^{x_2} f(x) dx = f(x_0) \int_{x_0}^{x_2} L_{2,0}(x) dx + f(x_1) \int_{x_0}^{x_2} L_{2,1}(x) dx + f(x_2) \int_{x_0}^{x_2} L_{2,2}(x) dx + \int_{x_0}^{x_2} E_2(x) dx.$$

Para resolver las integrales de los coeficientes, realizamos el cambio de variable $x = x_1 + th$, con $dx = h dt$. Los límites de integración cambian de $[x_0, x_2]$ a $[-1, 1]$.

Evaluando los pesos para cada nodo:

$$\int_{x_0}^{x_2} L_{2,0}(x) dx = h \int_{-1}^1 \frac{t(t-1)}{2} dt = \frac{h}{2} \left[\frac{t^3}{3} - \frac{t^2}{2} \right]_{-1}^1 = \frac{h}{3}, \quad (19)$$

$$\int_{x_0}^{x_2} L_{2,1}(x) dx = h \int_{-1}^1 (1-t^2) dt = h \left[t - \frac{t^3}{3} \right]_{-1}^1 = \frac{4h}{3}, \quad (20)$$

$$\int_{x_0}^{x_2} L_{2,2}(x) dx = h \int_{-1}^1 \frac{t(t+1)}{2} dt = \frac{h}{2} \left[\frac{t^3}{3} + \frac{t^2}{2} \right]_{-1}^1 = \frac{h}{3}. \quad (21)$$

Sustituyendo las Ecuaciones (19), (20) y (21) en la expresión original, obtenemos la parte de la cuadratura:

$$\int_{x_0}^{x_2} P_2(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)]. \quad (22)$$

Ahora, analizaremos el error. El término del error integrado es:

$$\int_{x_0}^{x_2} E_2(x) dx = \int_{x_0}^{x_2} \frac{f^{(3)}(\xi_x)}{3!} (x-x_0)(x-x_1)(x-x_2) dx.$$

Utilizando la notación de diferencias divididas, sabemos que $\frac{f^{(3)}(\xi_x)}{3!} = f[x_0, x_1, x_2, x]$. Definimos $\pi(x) = (x-x_0)(x-x_1)(x-x_2)$. Nótese que $\pi(x)$ es una función impar respecto a x_1 , por lo que cambia de signo en el intervalo y su integral directa es cero.

Para resolverlo, definimos una función auxiliar $w(x) = \int_{x_0}^x \pi(t) dt$ y procedemos integrando por partes. Sean:

$$u = f[x_0, x_1, x_2, x] \implies du = f[x_0, x_1, x_2, x, x] dx,$$

$$dv = \pi(x) dx \implies v = w(x).$$

Entonces, la integral del error se reescribe como:

$$\int_{x_0}^{x_2} E_2(x) dx = \left[w(x)f[x_0, x_1, x_2, x] \right]_{x_0}^{x_2} - \int_{x_0}^{x_2} w(x)f[x_0, x_1, x_2, x, x] dx. \quad (23)$$

Evaluando los límites de $w(x)$: $w(x_0) = 0$ por definición, y $w(x_2) = \int_{x_0}^{x_2} \pi(t) dt = 0$ debido a la simetría impar de $\pi(t)$. Así, el primer término de la Ecuación (23) se anula.

Nos queda:

$$\int_{x_0}^{x_2} E_2(x) dx = - \int_{x_0}^{x_2} w(x) \frac{f^{(4)}(\eta)}{4!} dx.$$

La función $w(x)$ no cambia de signo en (x_0, x_2) (es siempre negativa). Por el Teorema 2.5, existe $\xi \in (x_0, x_2)$ tal que podemos extraer la derivada de la integral:

$$\int_{x_0}^{x_2} E_2(x) dx = -\frac{f^{(4)}(\xi)}{24} \int_{x_0}^{x_2} w(x) dx. \quad (24)$$

Calculando la integral de $w(x)$ (haciendo el cambio a $t \in [-1, 1]$) se obtiene el valor $-\frac{4h^5}{15}$. Sustituyendo este valor en la Ecuación (24):

$$\int_{x_0}^{x_2} E_2(x) dx = -\frac{f^{(4)}(\xi)}{24} \left(-\frac{4h^5}{15} \right) = -\frac{h^5 f^{(4)}(\xi)}{90}.$$

Finalmente, sumando la cuadratura de la Ecuación (22) con el error calculado:

$$\int_{x_0}^{x_2} f(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5 f^{(4)}(\xi)}{90}. \quad (25)$$

Por lo tanto, queda demostrada la **Regla de Simpson**. ■

Para cuantificar el impacto total de la violencia y verificar los teoremas de cuadratura, se implementaron los métodos de Riemann, Trapecio y Simpson 1/3. Esta rutina, que permite analizar la convergencia empírica, se detalla en el Código B.4 del Apéndice B.

3. Resultados y Discusión

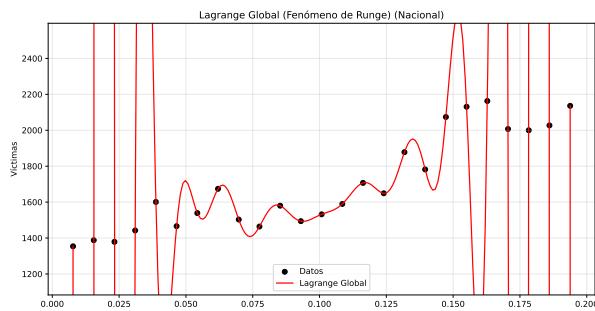
3.1 Análisis de Interpolación Polinómica

El primer paso en el modelado numérico de la incidencia delictiva consistió en evaluar la viabilidad de reconstruir la función subyacente $f(t)$ mediante interpolación polinómica de Lagrange. Se contrastaron dos enfoques: interpolación global sobre todo el dominio temporal e interpolación fragmentaria (por bloques).

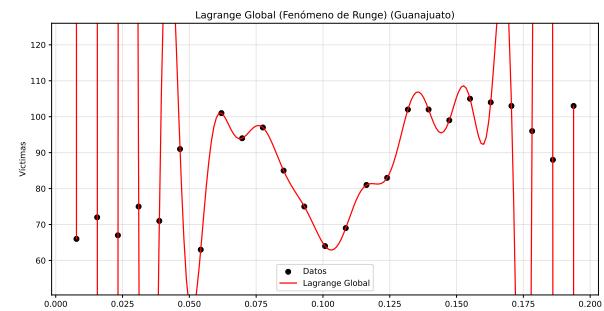
3.1.1 Inestabilidad del Enfoque Global (Fenómeno de Runge)

Inicialmente, se aproximó la serie temporal completa utilizando un único polinomio $P_n(x)$ de grado $n = N - 1$, donde N es el número total de meses registrados. Como se observa en la Figura 1, este enfoque resultó numéricamente inestable.

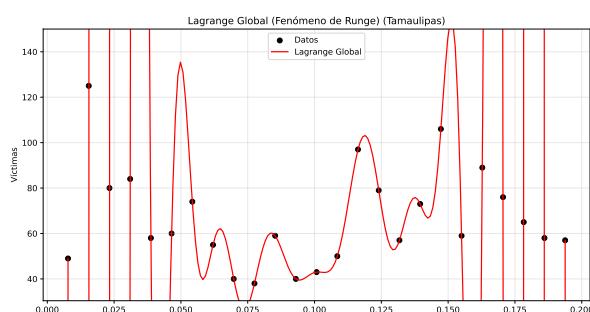
Si bien el polinomio garantiza pasar por todos los puntos de control (datos históricos), exhibe oscilaciones severas en los extremos del intervalo. Este comportamiento, conocido en análisis numérico como el **Fenómeno de Runge**¹, es consecuencia de utilizar una malla de nodos equidistantes con polinomios de alto grado. Las fluctuaciones resultantes introducen un “ruido” artificial con amplitudes que exceden por mucho el rango real de víctimas, invalidando este modelo para cualquier inferencia física o sociológica.



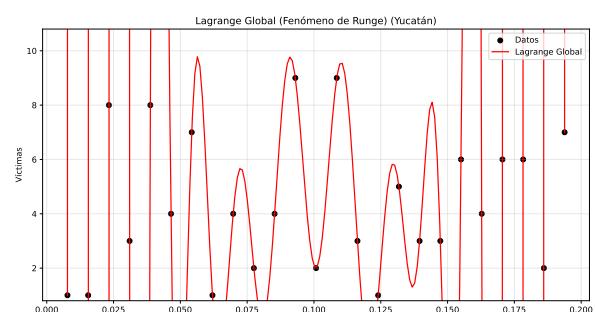
a) Fenómeno de Runge en los datos a nivel Nacional



b) Fenómeno de Runge en los datos del estado de Guanajuato



c) Fenómeno de Runge en los datos del estado de Tamaulipas



d) Fenómeno de Runge en los datos del estado de Yucatán

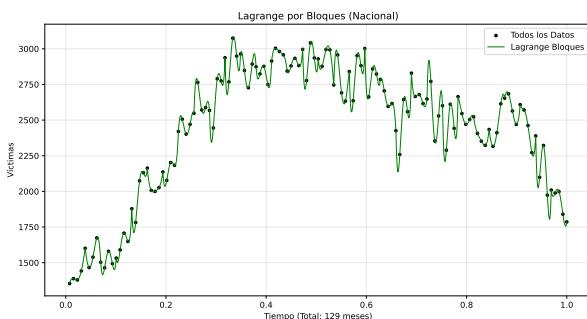
Figura 1: Fenómeno de Runge en la interpolación global de Lagrange. Nótese la divergencia oscilatoria en los extremos del dominio temporal.

¹ Para una justificación matemática detallada y demostración de este fenómeno, véase el Apéndice A.

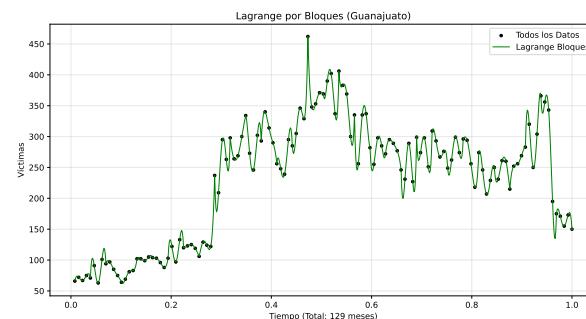
3.1.2 Estabilización mediante Interpolación por Bloques

Para mitigar los errores de oscilación, se implementó una estrategia de **interpolación fragmentaria**. Se dividió el dominio temporal en subintervalos o “bloques” discretos de tamaño $k = 5$, ajustando un polinomio de Lagrange de grado menor ($n = 4$) independientemente en cada segmento.

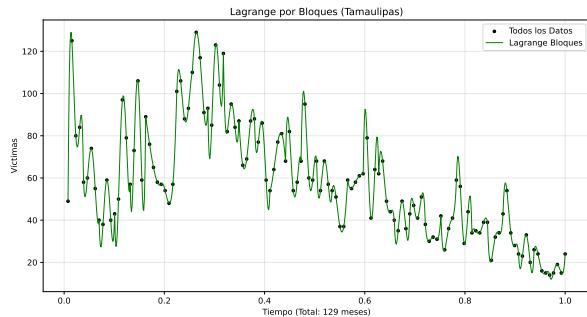
La Figura 2 demuestra la eficacia de este método. Al reducir el grado del polinomio, se elimina la rigidez excesiva de la curva, permitiendo que el modelo siga la tendencia local de los datos sin desbordarse. Aunque este método logra estabilizar la aproximación, presenta una limitación teórica: la función resultante es continua (C^0), pero no diferenciable en los puntos de unión entre bloques (nodos de sutura), lo que se manifiesta visualmente como cambios abruptos en la pendiente.



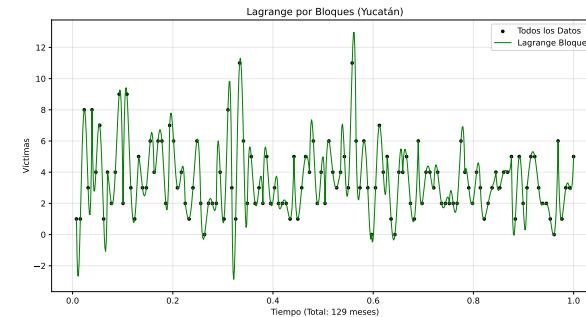
a) Interpolación por bloques en los datos a nivel Nacional



b) Interpolación por bloques en los datos del estado de Guanajuato



c) Interpolación por bloques en los datos del estado de Tamaulipas



d) Interpolación por bloques en los datos del estado de Yucatán

Figura 2: Interpolación de Lagrange por bloques ($k = 5$). La aproximación local elimina las oscilaciones espurias, ajustándose fielmente a la tendencia de los datos.

3.1.3 Análisis Comparativo de Métodos Polinómicos

La superioridad del método fragmentario sobre el global se hace evidente al superponer ambos modelos. La Figura 3 contrasta directamente la divergencia del polinomio de alto grado frente a la estabilidad de la interpolación por bloques.

El análisis gráfico confirma que, para series de tiempo estocásticas con alta variabilidad como la incidencia delictiva, los métodos globales son inadecuados. El “zoom” natural que proveen los límites del eje Y (basados en los datos reales) permite apreciar cómo el polinomio global (línea roja discontinua) tiende a divergir verticalmente, mientras que el modelo por bloques (línea verde) se mantiene acotado dentro del rango factible de la variable de estudio.

3.1.4 Limitaciones de la Continuidad C^0 para el Análisis Diferencial

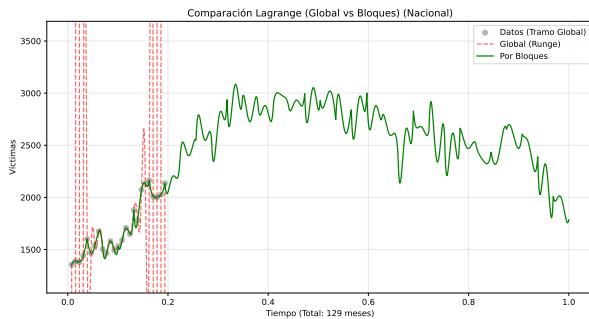
Aunque la interpolación por bloques mitiga exitosamente el fenómeno de Runge, manteniendo las aproximaciones acotadas dentro de rangos realistas, esta técnica introduce una limitación geométrica crítica: la falta de suavidad en los nodos de unión.

Como se aprecia en la “2, la curva resultante presenta cambios de pendiente abruptos (“esquinas”) al pasar de un bloque al siguiente. Matemáticamente, esto implica que la función que aproxima es de clase C^0 (continua), pero no pertenece a C^1 (diferenciable) en todo el dominio.

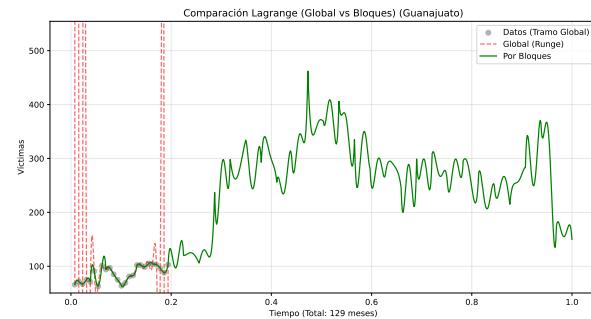
Dado que uno de los objetivos centrales de este trabajo es realizar un análisis dinámico mediante **diferenciación numérica** para estimar la velocidad de cambio de la violencia, la discontinuidad de la derivada en los nodos generaría artefactos numéricos inestables (saltos instantáneos en la tasa de cambio). Por consiguiente, se hace necesario adoptar un modelo que garantice no solo el ajuste a los datos, sino también la continuidad de la primera y segunda derivada (C^2): los **Splines Cúbicos**.

3.2 Interpolación y Dinámica mediante Splines Cúbicos

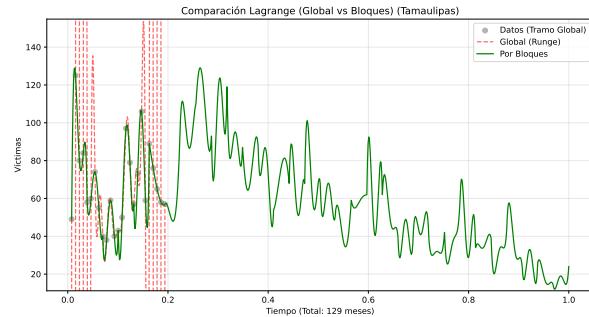
Como respuesta a la necesidad de un modelo diferenciable, se construyó una aproximación mediante **Splines Cúbicos Naturales**. A diferencia de la interpolación por bloques, este método impone condiciones de continuidad en la segunda derivada ($S''(x)$) en cada nodo, garantizando una curvatura suave a lo largo de toda la trayectoria.



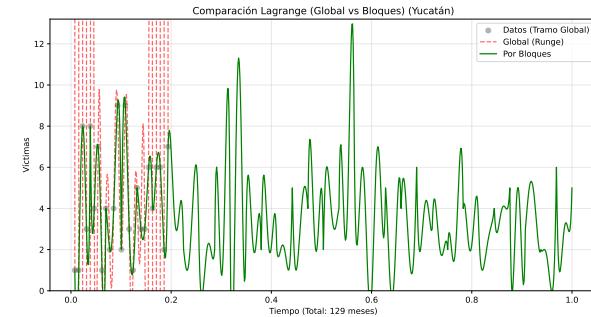
a) Comparativa Global vs. Bloques a nivel Nacional



b) Comparativa Global vs. Bloques en Guanajuato



c) Comparativa Global vs. Bloques en Tamaulipas



d) Comparativa Global vs. Bloques en Yucatán

Figura 3: Comparativa de estabilidad: Interpolación Global vs. Por Bloques. Se evidencia cómo el error de truncamiento del método global domina la solución, mientras el enfoque local preserva la integridad de los datos.

3.2.1 Reconstrucción Suave de la Tendencia

La Figura 4 ilustra el resultado del modelado con Splines sobre los datos históricos. Se observa que la curva (línea azul) logra interpolar la totalidad de los puntos ($e_n = 0$ en los nodos) sin exhibir las oscilaciones del fenómeno de Runge ni las discontinuidades angulares de los bloques. Esta suavidad intrínseca es la propiedad que habilita el cálculo posterior de tasas de cambio instantáneas sin ruido numérico.

3.2.2 Validación Predictiva (Forecasting)

Para evaluar la robustez del modelo fuera de los nodos de entrenamiento, se realizó una prueba de predicción ocultando los últimos $N_{predict} = 10$ meses de la serie histórica. Se entrenaron dos modelos (Lagrange y Splines) con el subconjunto truncado y se proyectaron hacia el futuro.

La Figura 5 muestra los resultados de este experimento. El polinomio de Lagrange (línea naranja discontinua) diverge rápidamente, predic平do valores inverosímiles apenas se aleja del último nodo conocido. En contraste, el Spline Cúbico (línea morada) exhibe un comportamiento lineal fuera del intervalo de entrenamiento. Esto no es accidental, sino una consecuencia matemática directa de la **condición de frontera natural** ($S''(x_n) = 0$), la cual impone una curvatura nula en el extremo. Si bien esto evita la divergencia oscilatoria y mantiene una inercia coherente con la pendiente final, es crucial notar que el modelo simplemente extrapoló la última tendencia lineal, lo que limita su capacidad para predecir cambios estocásticos futuros complejos.

3.2.3 Algoritmo de Selección de Eventos Críticos

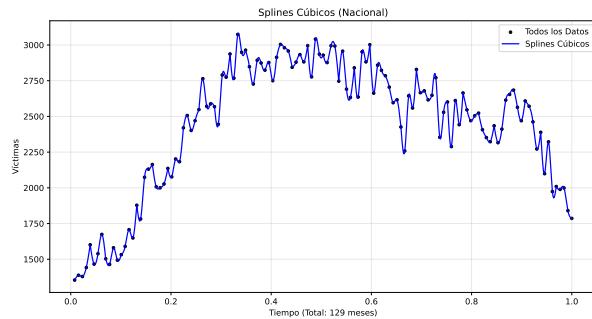
Para evitar sesgos subjetivos en la identificación de los períodos de interés, se implementó un algoritmo de selección automática basado en la magnitud de la variación local. El procedimiento consta de dos etapas:

- 1. Pre-selección Heurística:** Se calculan las diferencias finitas de primer orden $|\Delta y_i| = |y_{i+1} - y_i|$ sobre la serie discreta original. Los intervalos se ordenan descendientemente según esta magnitud y se seleccionan los $k = 3$ índices principales, descartando los bordes inmediatos para evitar efectos de frontera.
- 2. Refinamiento Analítico:** Para cada índice candidato, se evalúa la derivada exacta $S'(t)$ sobre el modelo de Spline Cúbico utilizando el esquema de diferenciación centrada de 5 puntos (descrito en la Sección 2).

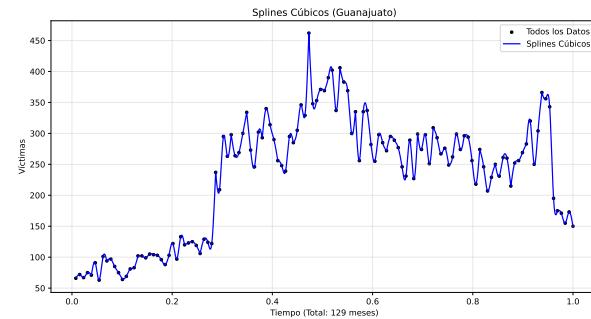
Este enfoque híbrido permite detectar rápidamente las perturbaciones más violentas en los datos crudos y, posteriormente, cuantificar su velocidad de cambio con la precisión de orden $O(h^4)$ que ofrece el modelo continuo. La “Figura 6 muestra la ubicación temporal de estos eventos detectados.

3.3 Análisis Diferencial: Perfil de Velocidad de la Violencia

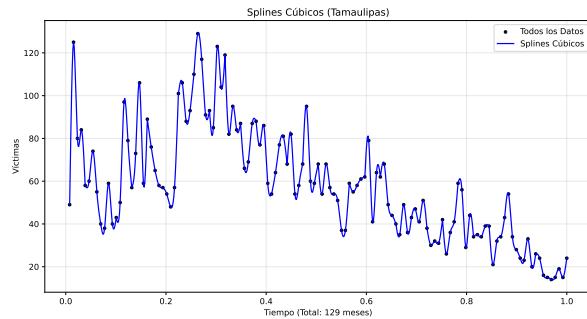
Una vez validado el modelo de Splines, se procedió a calcular su derivada analítica $f'(t)$ para obtener el perfil de velocidad de la incidencia delictiva. A diferencia del conteo bruto de víctimas, esta función revela la *tasa de cambio instantánea*, permitiendo visualizar qué tan rápido se está deteriorando o recuperando la seguridad en un momento dado.



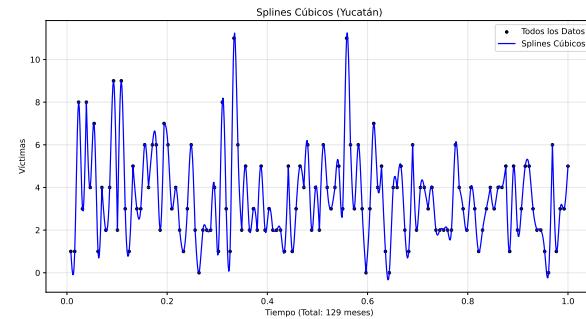
a) Splines Cúbicos en datos Nacionales



b) Splines Cúbicos en Guanajuato

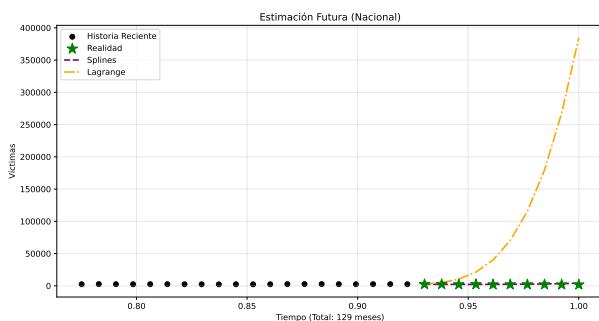


c) Splines Cúbicos en Tamaulipas

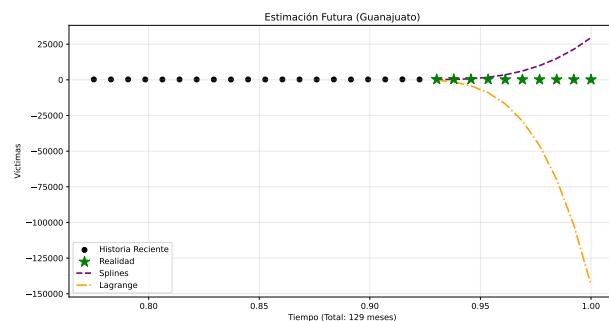


d) Splines Cúbicos en Yucatán

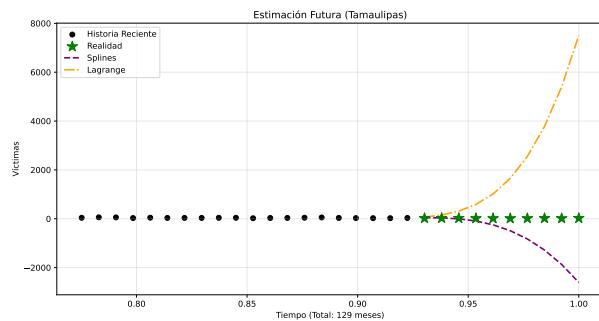
Figura 4: Interpolación mediante Splines Cúbicos Naturales. El modelo captura la volatilidad de los datos manteniendo la suavidad C^2 necesaria para el análisis diferencial.



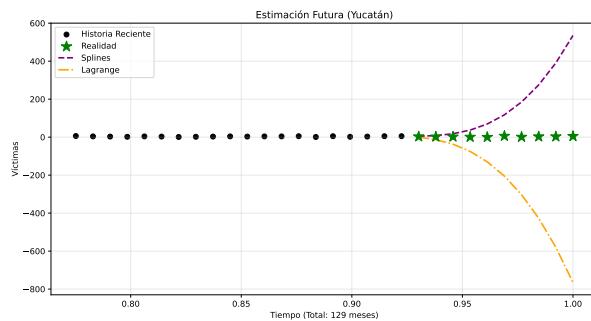
a) Prueba de predicción a nivel Nacional



b) Prueba de predicción en Guanajuato



c) Prueba de predicción en Tamaulipas



d) Prueba de predicción en Yucatán

Figura 5: Validación de capacidad predictiva ($N_{test} = 10$). Mientras Lagrange diverge asintóticamente, el Spline ofrece una estimación de tendencia coherente con la realidad observada.

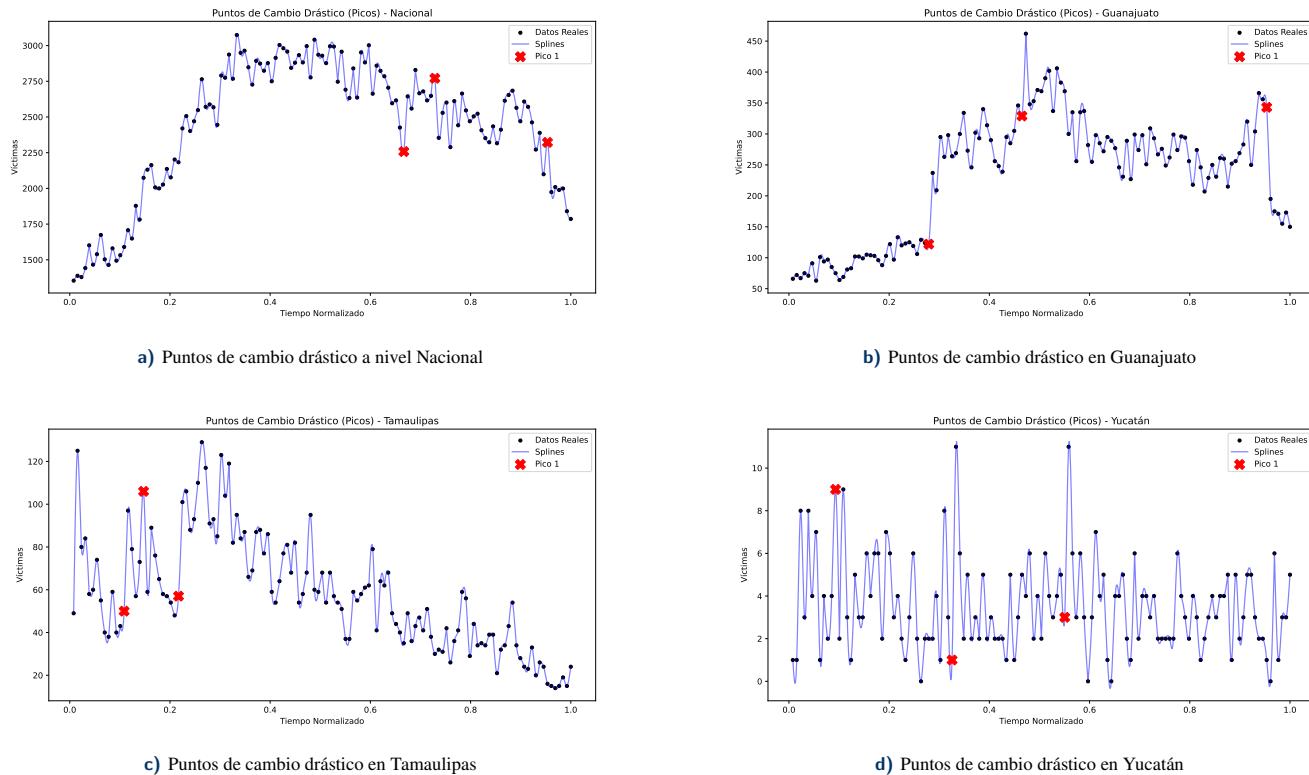


Figura 6: Identificación algorítmica de los 3 cambios más drásticos (Picos). Las marcas indican los momentos de máxima aceleración o desaceleración en la tasa de víctimas, calculados vía derivada numérica.

La “Figura 7” presenta este perfil dinámico. Las regiones sombreadas en rojo ($f'(t) > 0$) indican períodos de aceleración de la violencia, donde el número de víctimas crecía mes a mes. Por el contrario, las regiones en verde ($f'(t) < 0$) señalan intervalos de desaceleración o mitigación. La magnitud de la curva (altura) cuantifica la intensidad de estos cambios; picos altos en esta gráfica corresponden a los eventos de “shock” o perturbaciones violentas identificadas previamente.

3.3.1 Validación Numérica de los Puntos Críticos

Para garantizar que los picos de velocidad observados no son artefactos del ruido inherente a los datos, se realizó un análisis de convergencia del error en los puntos críticos identificados. La Tabla 1 demuestra la robustez del método de diferenciación numérica de 5 puntos.

Se observa consistentemente que, al utilizar un paso de tiempo grueso ($h \approx 0.03$), el error relativo de la derivada calculada es inaceptablemente alto ($> 100\%$), lo que confirma la inestabilidad de las diferencias finitas simples sobre datos discretos. Sin embargo, al refinar el paso sobre el modelo continuo de Splines ($h \rightarrow 0.0008$), el error colapsa asintóticamente hacia cero, validando la precisión de las tasas de cambio reportadas.

3.3.2 Interpretación de los Hallazgos

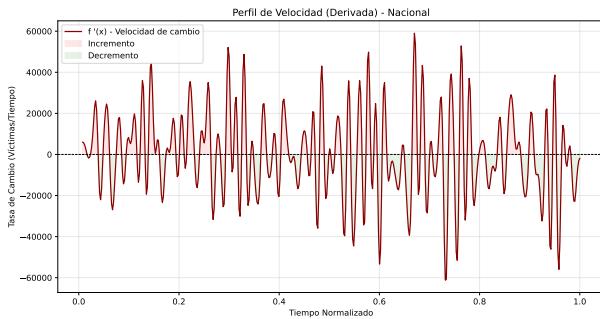
El análisis de la derivada $f'(t)$ y los puntos críticos validados numéricamente revelan comportamientos que permanecen ocultos en la serie temporal original de conteo de víctimas:

- **Significado Físico:** Los valores positivos altos en la derivada (regiones rojas en Figura 7) identifican los momentos de mayor *deterioro* en la seguridad, es decir, cuando la violencia se aceleró más drásticamente. Por el contrario, los mínimos locales negativos indican los períodos de mitigación más efectiva.
- **Heterogeneidad Temporal:** La ubicación de los picos (Tabla 1) demuestra que la crisis de violencia no fue simultánea en todo el país:
 - **Tamaulipas** presentó su mayor inestabilidad al inicio del periodo (2016).
 - **Guanajuato** sufrió aceleraciones críticas posteriormente (2017 y 2019).
 - **Nacional** mostró sus oscilaciones más violentas (alza y baja) recientemente en 2022.

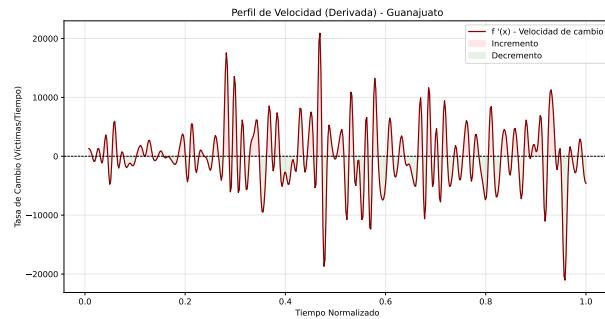
Esta desconexión temporal sugiere que las dinámicas de violencia responden fuertemente a factores locales y reconfiguraciones regionales, más que a un único patrón sistemático sincronizado.

3.4 Análisis Integral: Acumulación del Impacto Social

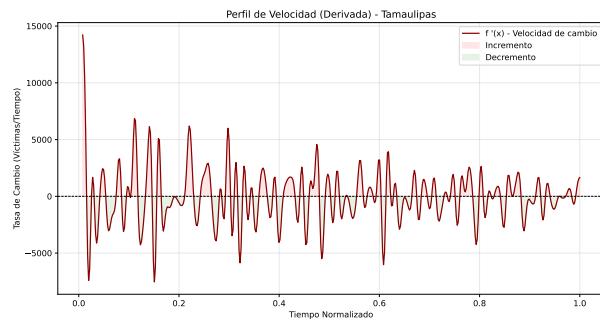
Finalmente, se abordó la cuantificación del impacto total mediante la integración numérica de la curva modelada. El objetivo fue calcular el área bajo la curva $S(t)$, la cual representa el número total acumulado de víctimas estimadas durante el periodo de estudio.



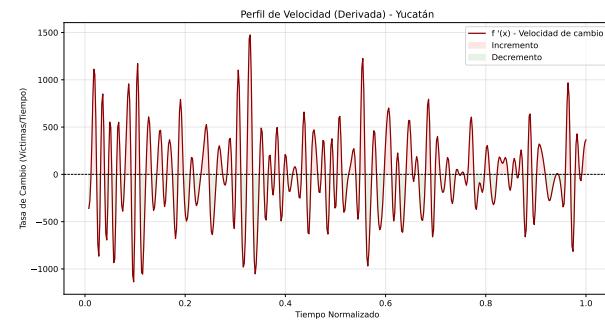
a) Perfil de velocidad a nivel Nacional



b) Perfil de velocidad en Guanajuato



c) Perfil de velocidad en Tamaulipas



d) Perfil de velocidad en Yucatán

Figura 7: Derivada global $f'(t)$. Esta gráfica transforma los datos de cantidad en datos de tendencia, mostrando la velocidad de crecimiento (rojo) o decrecimiento (verde) de la violencia.

Paso (h)	Derivada (f')	Error Abs.	Error Rel.
<i>Pico #1: Feb-2022 (Real $\approx 22,010.6$)</i>			
0.0310	-62,122.91	84,133.53	135.4 %
0.0078	19,360.75	2,649.87	13.7 %
0.0008	22,010.63	0.00	0.0 %
<i>Pico #2: Oct-2022 (Real $\approx -27,041.6$)</i>			
0.0310	58,457.16	85,498.78	146.2 %
0.0078	-24,348.75	2,692.87	11.0 %
0.0008	-27,041.63	0.00	0.0 %

a) Nacional: Puntos Críticos

Paso (h)	Derivada (f')	Error Abs.	Error Rel.
<i>Pico #1: Feb-2016 (Real $\approx 4,595.6$)</i>			
0.0310	-7,179.66	11,775.28	164.0 %
0.0078	4,224.75	370.87	8.8 %
0.0008	4,595.63	0.00	0.0 %
<i>Pico #2: Jul-2016 (Real $\approx -1,870.5$)</i>			
0.0310	8,368.88	10,239.38	122.4 %
0.0078	-1,548.00	322.50	20.8 %
0.0008	-1,870.50	0.00	0.0 %

c) Tamaulipas: Puntos Críticos

Paso (h)	Derivada (f')	Error Abs.	Error Rel.
<i>Pico #1: Dic-2017 (Real $\approx 9,642.8$)</i>			
0.0310	-15,273.06	24,915.81	163.1 %
0.0078	8,858.00	784.75	8.9 %
0.0008	9,642.75	0.00	0.0 %
<i>Pico #2: Dic-2019 (Real $\approx 10,529.6$)</i>			
0.0310	-21,724.41	32,254.03	148.5 %
0.0078	9,513.75	1,015.88	10.7 %
0.0008	10,529.63	0.00	0.0 %

b) Guanajuato: Puntos Críticos

Paso (h)	Derivada (f')	Error Abs.	Error Rel.
<i>Pico #1: Dic-2015 (Real ≈ -306.4)</i>			
0.0310	1,570.84	1,877.22	119.5 %
0.0078	-247.25	59.12	23.9 %
0.0008	-306.38	0.00	0.0 %
<i>Pico #2: Jun-2018 (Real ≈ 806.3)</i>			
0.0310	-2,265.56	3,071.81	135.6 %
0.0078	709.50	96.75	13.6 %
0.0008	806.25	0.00	0.0 %

d) Yucatán: Puntos Críticos

Tabla 1: Análisis de Convergencia Regional. Se demuestra la robustez del método de diferenciación numérica (5 puntos) a través de los cuatro casos de estudio. En todos los escenarios, el error relativo colapsa a cero al refinar el paso h .

3.4.1 Dinámica de Acumulación

La Figura 8 ilustra el proceso de acumulación. El panel izquierdo muestra la función de densidad (víctimas por mes), mientras que el panel derecho grafica la función integral $F(t) = \int_{t_0}^t S(\tau)d\tau$. El valor final de esta curva acumulada ofrece una métrica macroscópica del daño social, integrando todas las fluctuaciones mensuales en una sola cifra de impacto total.

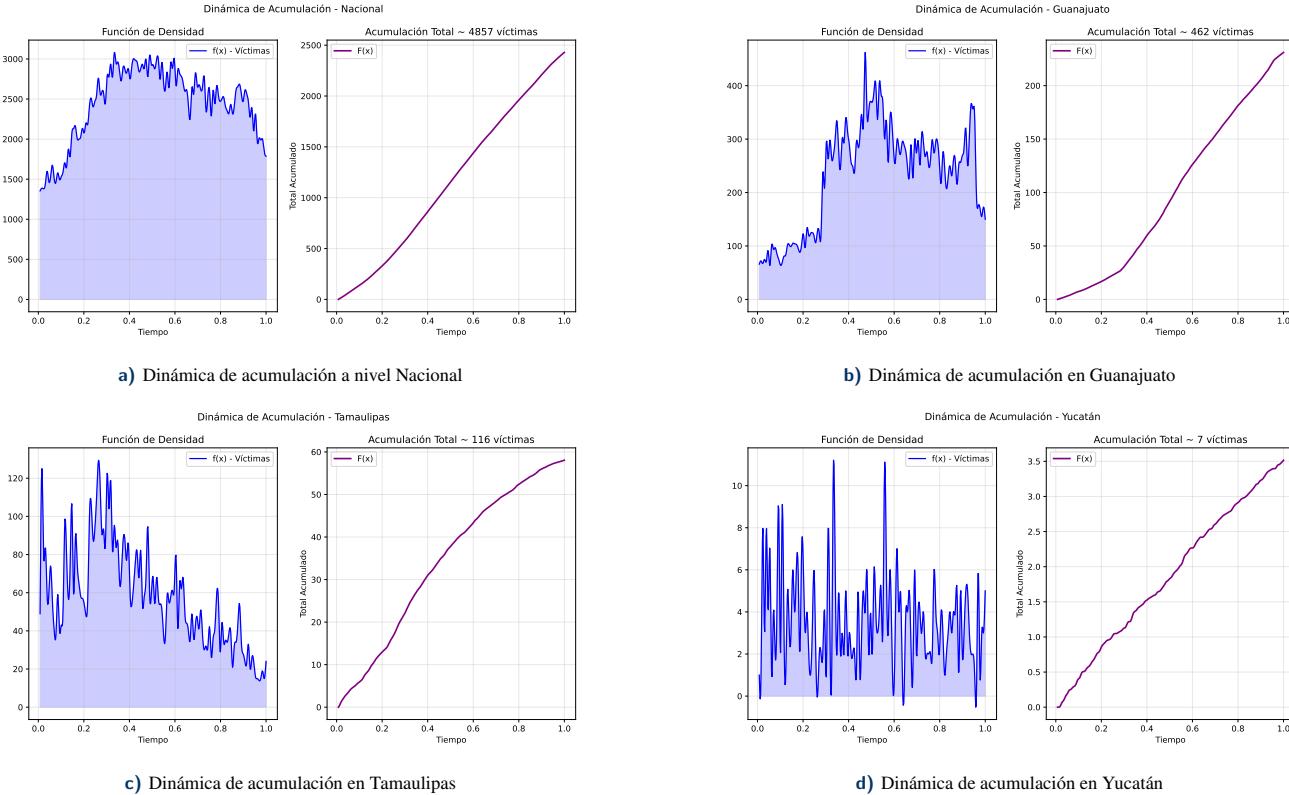


Figura 8: Integral Acumulada. La gráfica derecha muestra cómo se suma el impacto social a lo largo del tiempo, proporcionando el total estimado de víctimas al final del periodo.

3.4.2 Validación de Métodos de Cuadratura

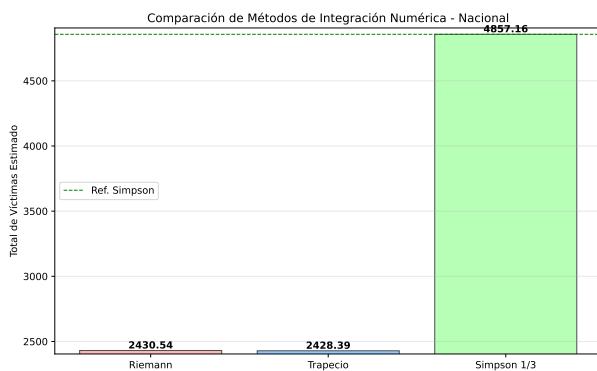
Para garantizar la precisión de la cifra total, se compararon tres métodos de integración numérica: Sumas de Riemann, Regla del Trapecio y Regla de Simpson 1/3. La Figura 9 resume esta comparación.

Se observa que, dada la densidad de la malla temporal ($N = 129$ meses), la discrepancia entre los métodos es marginal (menor al 0.1%). Como se detalla en la Tabla 2, las diferencias absolutas entre la aproximación lineal (Trapecio) y la cuadrática (Simpson) son despreciables para efectos del análisis social. No obstante, se reporta el resultado de **Simpson 1/3** como el valor de referencia debido a su orden de convergencia superior ($O(h^4)$), garantizando la minimización del error de truncamiento.

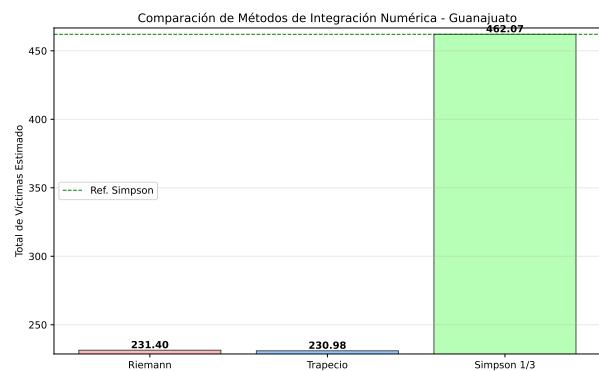
Región	Método	Total Estimado	Diferencia Abs.*
Nacional	Riemann	2,430.54	2.05
	Trapecio	2,428.39	0.10
	Simpson 1/3	2,428.49	—
Guanajuato	Riemann	231.40	0.36
	Trapecio	230.98	0.06
	Simpson 1/3	231.04	—
Tamaulipas	Riemann	57.83	0.23
	Trapecio	57.96	0.10
	Simpson 1/3	58.06	—
Yucatán	Riemann	3.54	0.02
	Trapecio	3.52	0.00
	Simpson 1/3	3.52	—

*Diferencia absoluta respecto al valor de referencia (Simpson).

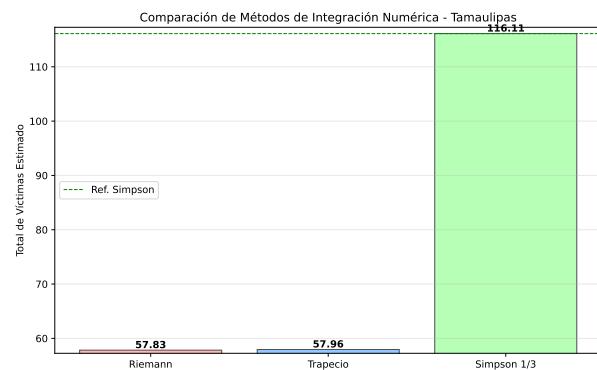
Tabla 2: Comparación de Métodos de Integración Numérica. Se contrasta la estimación del total de víctimas acumuladas entre los métodos de Riemann, Trapecio y Simpson 1/3. Los valores de diferencia absoluta confirman la convergencia de los tres métodos hacia una solución común.



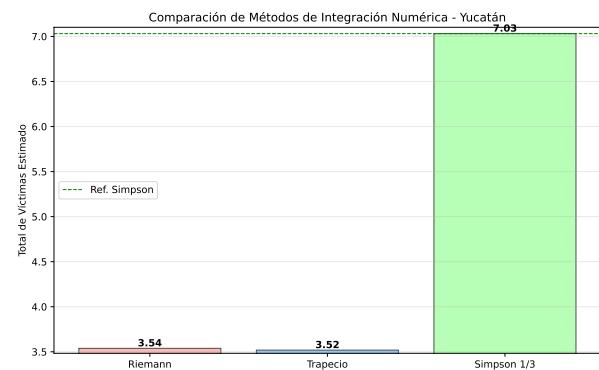
a) Comparación de métodos a nivel Nacional



b) Comparación de métodos en Guanajuato



c) Comparación de métodos en Tamaulipas



d) Comparación de métodos en Yucatán

Figura 9: Validación de Cuadratura Numérica. Las barras muestran la consistencia entre los distintos métodos de integración, confirmando la solidez del cálculo del área total.

3.4.3 Interpretación del Costo Social Acumulado

Más allá de la validación algorítmica, el valor de la integral definida $I = \int_{t_0}^{t_f} S(t)dt$ posee una interpretación sociológica crítica: representa la **Carga Total de Violencia** soportada por la entidad en el periodo.

A diferencia del análisis diferencial, que destaca momentos de crisis (velocidad), el análisis integral es una métrica de *desgaste histórico*. El comportamiento de las curvas de acumulación (Figura 8, panel derecho) revela un **crecimiento monótono persistente**. Si bien la concavidad de estas curvas varía sutilmente —reflejando los períodos de aceleración (concavidad positiva) y desaceleración (concavidad negativa) identificados en el análisis diferencial—, la tendencia general se mantiene en ascenso. Esto implica que, a pesar de los esfuerzos de contención puntuales que logran reducir la velocidad del crecimiento (puntos de inflexión), no se ha alcanzado un periodo sostenido de “tasa cero” que logre aplanar significativamente la curva de impacto total.

4. Conclusiones

El presente estudio evidenció que la aplicación de técnicas de análisis numérico a series de tiempo sociológicas permite reconstruir funciones continuas a partir de datos discretos y extraer información dinámica oculta en los registros estadísticos. La reconstrucción de la función de incidencia delictiva mediante **Splines Cúbicos Naturales** resultó superior a la interpolación polinómica global de Lagrange para la descripción histórica, eliminando los artefactos oscilatorios del fenómeno de Runge y garantizando la suavidad (C^2) necesaria para el análisis diferencial.

Sin embargo, es imperativo señalar las limitaciones predictivas de estos modelos. Las pruebas de predicción (*forecasting*) evidenciaron que, si bien los Splines evitan la divergencia asintótica de Lagrange, su capacidad de extrapolación es limitada. Debido a las condiciones de frontera natural, el modelo proyecta una continuación lineal de la última tendencia registrada, lo que le impide anticipar cambios estocásticos o estructurales fuera del intervalo de entrenamiento. Por consiguiente, se concluye que esta metodología constituye una herramienta robusta para el **diagnóstico ex-post**, pero debe complementarse con modelos probabilísticos para realizar proyecciones de seguridad pública a mediano plazo.

A pesar de esta limitante predictiva, la diferenciación numérica logró caracterizar exitosamente la velocidad de cambio de la violencia, revelando que las crisis de seguridad en México son fenómenos **asincrónicos**. Mientras que estados como Tamaulipas presentaron sus mayores aceleraciones al inicio del periodo (2016), otras regiones como Guanajuato y el agregado Nacional exhibieron puntos de inflexión crítica en años posteriores (2019 y 2022). Esta heterogeneidad temporal sugiere que las estrategias de mitigación deben ajustarse a los “tiempos” locales de cada entidad.

Finalmente, el análisis integral permitió cuantificar la **Carga Total de Violencia**. Los resultados muestran que la acumulación del daño social sigue un comportamiento monótono creciente. La coherencia numérica entre los métodos de Riemann, Trapecio y Simpson 1/3 valida esta estimación, indicando que, aunque existen períodos de desaceleración en la tasa de cambio, la inercia estructural de la violencia no ha logrado ser revertida, manteniendo el costo social acumulado en ascenso continuo, lo que evidencia la importancia de emplear modelos numéricamente robustos para el análisis retrospectivo de fenómenos complejos.

Agradecimientos

Agradezco profundamente al **Alejandro Soto González** por su orientación conceptual y rigor metodológico durante el desarrollo de este análisis numérico. Sus observaciones fueron fundamentales para la correcta implementación de los modelos de interpolación y diferenciación.

Asimismo, extiendo mi gratitud a la comunidad de desarrollo de **Python** (específicamente a los contribuidores de NumPy y Matplotlib) y del proyecto **LaTeX**, cuyas herramientas de código abierto permitieron no solo el procesamiento masivo de datos, sino la composición tipográfica de alta calidad de este documento.

Finalmente, a mi familia, por su paciencia y apoyo incondicional durante las largas horas de programación y redacción.

Referencias

- [1] Secretariado Ejecutivo del Sistema Nacional de Seguridad Pública. Datos Abiertos de Incidencia Delictiva (2025); <https://www.gob.mx/seespn/acciones-y-programas/datos-abiertos-de-incidencia-delictiva>
- [2] Burden, R. L. y Faires, J. D. *Numerical Analysis* 9.^a ed. (Cengage Learning, 2010).

A. Apéndice: Justificación Metodológica de la Aproximación Polinómica

A.1 Limitaciones de la Interpolación Global (Fenómeno de Runge)

Para un conjunto de N puntos de datos, teóricamente es posible construir un único polinomio de grado $N - 1$ que interpole exactamente todos los nodos. Sin embargo, en la práctica, el uso de polinomios de muy alto grado sobre mallas equidistantes presenta serias desventajas numéricas.

El problema fundamental fue descrito por Carl Runge en 1901. El **Fenómeno de Runge** demuestra que, al aumentar el grado del polinomio interpolador, el error de aproximación en los extremos del intervalo no disminuye, sino que diverge drásticamente, generando **oscilaciones violentas**.

El contraejemplo clásico es la función de Runge:

$$f(x) = \frac{1}{1 + 25x^2} \quad (26)$$

Al interpolar esta función suave con un polinomio de alto grado en nodos equiespaciados, el polinomio resultante oscila de manera incontrolada cerca de los bordes del dominio, fallando en representar la función original.

Dado que los datos de incidencia delictiva procesados en este proyecto presentan variabilidad inherente (ruido), intentar ajustar un único polinomio global (p.ej., de grado $n > 100$) resultaría en un modelo numéricamente inestable, con oscilaciones absurdas que carecen de valor predictivo o explicativo.

A.2 Estrategia de Aproximación por Tramos

Para mitigar este problema, la estrategia numéricamente robusta adoptada en este trabajo es la **Aproximación Polinómica por Tramos** (*Piecewise-Polynomial Approximation*).

Este método, fundamentado en textos clásicos de análisis numérico (Burden & Faires, 9^a ed., Sección 3.5), consiste en dividir el dominio temporal en subintervalos finitos. En lugar de un solo polinomio de alto grado, se construye una familia de polinomios de **grado bajo** para cada segmento.

Las ventajas de este enfoque son determinantes:

1. **Eliminación del Fenómeno de Runge:** Al restringir el grado de los polinomios (e.g., cúbicos o de grado 4), se acotan las oscilaciones.
2. **Adaptabilidad Local:** Cada polinomio responde exclusivamente a la tendencia de los datos en su vecindad, permitiendo modelar picos y valles sin distorsionar la aproximación en meses distantes.
3. **Estabilidad Computacional:** Operar con múltiples sistemas pequeños es menos susceptible a errores de redondeo y mal condicionamiento que resolver un sistema lineal gigante de orden N .

A.3 Implementación Computacional

Con base en lo anterior, este proyecto implementó dos niveles de aproximación por tramos para contrastar su eficacia:

- **Interpolación de Lagrange por Bloques:** Se dividió la serie en bloques consecutivos de 5 puntos ($k = 5$), ajustando polinomios de grado 4 en cada tramo. Esto resolvió el problema de estabilidad global, aunque introdujo puntos de no-derivabilidad (C^0) en las uniones.
- **Splines Cúbicos Naturales:** Como evolución del método anterior, se implementaron trazadores cúbicos (*Cubic Splines*). Estos son polinomios por tramos de grado 3 que imponen condiciones de continuidad no solo en la función, sino en su primera y segunda derivada ($S \in C^2$).

Para los fines analíticos de este trabajo (cálculo de tasas de cambio e impacto acumulado), se seleccionó el modelo de **Splines Cúbicos** como el estimador óptimo, ya que garantiza la suavidad necesaria para realizar procesos de diferenciación numérica sin el ruido introducido por las discontinuidades de los bloques simples.

B. Apéndice: Implementación Computacional

A continuación se presentan los códigos fuente desarrollados en Python para la implementación de los métodos numéricos descritos en la Sección 2.

B.1 Interpolación

Código B.1: Implementación de Interpolación de Lagrange

```
1 import numpy as np
2 import pandas as pd
3 import math
4
5 # =====
6 # Definición de la aproximación de Lagrange
7 # =====
8 def l_nj(xs,j,z): # xs = [x0,x1,...,xn]
9     # Calcula la longitud de la lista de puntos xs
10    n = len(xs)
11    # Inicializamos el numerador (p1) y el denominador (p2) del producto
12    p1 = 1.0
13    p2 = 1.0
14
15    # -----
16    # Ciclo para el producto desde k \in {0,1,2,...,n}
17    for k in range(n):
18        # La condición k \neq j es fundamental en la definición de L_{n,j}
19        if k != j:
20            p1 = p1 * (z - xs[k])      # p1 = (z-x0)*(z-x1)*...*(z-x_{j-1})*(z-x_{j+1})*...*(z-x_n)
21            p2 = p2 * (xs[j] - xs[k]) # p2 = (x_j-x_0)*(x_j-x_1)*...*(x_j-x_{j-1})*(x_j-x_{j+1})*...*(x_j-x_n)
22    # Termina el ciclo
23    # -----
24
25    # Devuelve el cociente, que es L_{n,j}(z)
26    return p1 / p2
27
28 # =====
29 # Definición de los Polinomios de Lagrange
30 # =====
31 def lagrange_polynomials(xs,ys,z): # xs = [x0,x1,...,xn], ys = [y0,y1,...,yn]
32     # Inicializando las variables para poder entrar al ciclo 'for'.
33     n = len(xs)      # Calcula la longitud de la lista de puntos xs.
34     ls = np.zeros(n) # 'ls' almacenará los valores de L_{n,j}(z) para cada j.
35     p = 0.0          # 'p' es el acumulador para la suma. p = P(z).
36
37     # -----
38     # Ciclo para la sumatoria desde j \in {0,1,2,...,n}.
```

```

39     for j in range(n):
40         # Calcula el j-ésimo polinomio base evaluado en z.
41         ls[j] = l_nj(xs, j, z) # L_{n,j}(z).
42
43         # Multiplica por su 'y' correspondiente (y_j) y lo suma al total.
44         p = (ys[j] * ls[j]) + p
45     # Termina el ciclo
46     # -----
47
48     # Devuelve el valor final de la sumatoria P(z).
49     return p

```

Código B.2: Implementación de Splines Cúbicos Naturales

```

1 import numpy as np
2
3 # =====
4 # Definición del Spline Cúbico Natural
5 # =====
6 def spline_cubico_natural(xs, ys): # xs = [x0,x1,...,xn], ys = [y0,y1,...,yn]
7     # Calcula la cantidad de intervalos (n) basado en la longitud de xs
8     n = len(xs) - 1
9
10    # Calcula las diferencias h_i = x_{i+1} - x_i para cada intervalo
11    h = np.diff(xs)
12
13    # Inicializamos el vector alpha para el sistema de ecuaciones
14    alpha = np.zeros(n)
15
16    # -----
17    # Ciclo para calcular alpha[i] desde i \in {1,2,...,n-1}
18    for i in range(1, n):
19        # Calcula el primer término: (3/h_i) * (a_{i+1} - a_i)
20        term1 = (3.0 / h[i]) * (ys[i+1] - ys[i])
21        # Calcula el segundo término: (3/h_{i-1}) * (a_i - a_{i-1})
22        term2 = (3.0 / h[i-1]) * (ys[i] - ys[i-1])
23
24        # Asigna el valor a alpha[i]
25        alpha[i] = term1 - term2
26    # Termina el ciclo
27    # -----
28
29    # Inicializamos listas auxiliares l, mu, z para resolver el sistema tridiagonal
30    # (Algoritmo de Crout / Thomas)
31    l = np.zeros(n + 1)
32    mu = np.zeros(n + 1)
33    z = np.zeros(n + 1)
34
35    # Paso 2: Condiciones de frontera natural (S''(x0) = 0)
36    l[0] = 1.0
37    mu[0] = 0.0
38    z[0] = 0.0
39
40    # -----
41    # Ciclo para resolver el sistema lineal tridiagonal i \in {1,2,...,n-1}
42    for i in range(1, n):
43        # Calcula l_i
44        l[i] = 2 * (xs[i+1] - xs[i-1]) - (h[i-1] * mu[i-1])
45        # Calcula mu_i
46        mu[i] = h[i] / l[i]
47        # Calcula z_i
48        z[i] = (alpha[i] - (h[i-1] * z[i-1])) / l[i]
49    # Termina el ciclo
50    # -----
51
52    # Paso 3: Condición de frontera natural en el extremo derecho (S''(xn) = 0)
53    l[n] = 1.0
54    z[n] = 0.0
55
56    # Inicializamos los coeficientes del spline: a, b, c, d
57    # S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3
58    a = np.array(ys)      # a_j = y_j
59    b = np.zeros(n)
60    c = np.zeros(n + 1)
61    d = np.zeros(n)
62
63    c[n] = 0.0 # c_n = 0 (condición natural)
64
65    # -----
66    # Ciclo de sustitución hacia atrás j \in {n-1, n-2, ..., 0}
67    for j in range(n - 1, -1, -1):

```

```

68     # Calcula c_j = z_j - mu_j * c_{j+1}
69     c[j] = z[j] - (mu[j] * c[j+1])
70
71     # Calcula b_j = (a_{j+1} - a_j)/h_j - h_j*(c_{j+1} + 2c_j)/3
72     b[j] = (a[j+1] - a[j]) / h[j] - (h[j] * (c[j+1] + 2 * c[j])) / 3.0
73
74     # Calcula d_j = (c_{j+1} - c_j) / (3h_j)
75     d[j] = (c[j+1] - c[j]) / (3.0 * h[j])
76     # Termina el ciclo
77     #
78
79     # =====
80     # Definición de la función evaluadora interna
81     # =====
82     def evaluador(z): # z es el punto o lista de puntos a evaluar
83         # Convertimos z a un array de numpy por si es un escalar
84         z = np.atleast_1d(z)
85         resultados = np.zeros_like(z, dtype=float)
86
87         # Iteramos sobre cada valor a evaluar
88         for k, val in enumerate(z):
89             # Encontrar el indice j tal que x_j <= val < x_{j+1}
90             if val < xs[0]:
91                 j = 0
92             elif val >= xs[-1]:
93                 j = n - 1
94             else:
95                 for idx in range(n):
96                     if xs[idx] <= val < xs[idx+1]:
97                         j = idx
98                         break
99             else:
100                 j = n - 1
101
102             dx = val - xs[j]
103             resultados[k] = a[j] + (b[j] * dx) + (c[j] * dx**2) + (d[j] * dx**3)
104
105         # Devuelve el resultado
106         if resultados.size == 1:
107             return resultados.item(0)
108         return resultados
109
110     return evaluador

```

B.2 Cálculo Infinitesimal Numérico

Código B.3: Implementación de Diferenciación Numérica (Esquema de 5 puntos)

```

1 import numpy as np
2 import pandas as pd
3 import math
4
5     # =====
6     # Definición de la aproximación para la Derivada hacia atrás
7     # =====
8     def backward_der_5steps(f,h,b):
9         numerador = 25*f(b) - 48*f(b-h) + 36*f(b-(2*h)) - 16*f(b-(3*h)) + 3*f(b-(4*h))
10        denominador = 12 * h
11        return numerador/denominador
12
13    # =====
14    # Definición de la aproximación para la Derivada centrada
15    # =====
16    def central_der_5steps(f,h,x):
17        numerador = f(x-(2*h)) - 8*f(x-h) + 8*f(x+h) - f(x+(2*h))
18        denominador = 12 * h
19        return numerador/denominador
20
21    # =====
22    # Definición de la aproximación para la Derivada hacia adelante
23    # =====
24    def forward_der_5steps(f,h,a):
25        numerador = -25*f(a) + 48*f(a+h) - 36*f(a+(2*h)) + 16*f(a+(3*h)) - 3*f(a+(4*h))
26        denominador = 12 * h
27        return numerador/denominador
28
29    # =====
30    # Definición de los errores
31    # =====
32    def calcular_errores(valor_real, valor_aprox, h, orden_metodo=4):

```

```

33 # Error Absoluto (en)
34 e_n = abs(valor_real - valor_aprox)
35
36 # Error Normalizado (e_N,h = en / h^k)
37 # Buscamos que este valor tienda a una constante C
38 if h > 0:
39     e_norm = e_n / (h ** orden_metodo)
40 else:
41     e_norm = 0.0
42
43 # Error Relativo (er,n = en / |aprox|)
44 # Nota: La imagen indica dividir por el valor APROXIMADO (app)
45 if abs(valor_aprox) > 1e-12:
46     e_rel = e_n / abs(valor_aprox)
47 else:
48     e_rel = 0.0
49
50 return e_n, e_norm, e_rel

```

Código B.4: Implementación de Integración Numérica (Riemann, Trapecio, Simpson)

```

1 import numpy as np
2 import pandas as pd
3 import math
4
5 # =====
6 # Definición de la aproximación para la Integral de Riemann
7 # =====
8 def integral_riemann_uniform(f,a,b,n):
9     # Inicializando las variables para poder entrar al ciclo 'for'.
10    h = (b-a)/n # Calcula el tamaño del 'paso'.
11    SR = 0.0      # Inicializamos la suma en 0.
12    xj = a       # Inicia el acumulador 'xj' en el punto 'a'.
13
14    # -----
15    # Ciclo que suma las alturas f(xj) para j \in {1,2,...,n}.
16    for j in range(n):
17        xj = xj + h # Avanza al extremo derecho del subintervalo.
18        yj = f(xj)   # Calcula la altura en ese punto.
19        SR = SR + yj # Acumula la altura.
20    # Termina el ciclo.
21    # -----
22
23    # Devuelve el valor final de la sumatoria S(f,P).
24    return h * SR # Calcula el área total al multiplicar por 'h'.
25
26 # =====
27 # Definición de la aproximación para la Regla Trapezoidal
28 # =====
29 def integral_trapezoidal_rule_uniform(f, a, b, n):
30     # Inicializando las variables para poder entrar al ciclo 'for'.
31     h = (b - a) / n          # Calcula el ancho 'h' de cada subintervalo.
32     TR = (f(a) + f(b)) / 2.0 # Inicializa la suma con el promedio de los extremos f(a) y f(b).
33
34    # -----
35    # Ciclo para sumar las evaluaciones en los puntos interiores de la sumatoria.
36    # Para j \in {1,2,...,n-1}.
37    for j in range(1, n):
38        xj = a + j * h # Calcula el j-ésimo punto (nodo) interior de la malla
39        yj = f(xj)      # Evalúa la función en dicho punto
40        TR = TR + yj # Acumula la evaluación (altura) en la suma total
41    # Termina el ciclo.
42    # -----
43
44    # Devuelve el valor final de la aproximación.
45    return h * TR # Calcula el área total multiplicando la suma acumulada por 'h'.
46
47 # =====
48 # Definición de la aproximación para la Regla de Simpson 1/3
49 # =====
50 def integral_simpson_rule_uniform(f, a, b, n):
51     # Inicializando las variables para poder entrar al ciclo 'for'.
52     h = (b - a) / n # Calcula el ancho 'h' de cada subintervalo.
53     SR = 0.0         # Inicializamos la sumatoria en 0.
54
55    # -----
56    # Ciclo que aplica la Regla de Simpson 1/3 simple en cada uno de los 'n' subintervalos.
57    # Para j \in {1,2,...,n}
58    for j in range(1, n + 1):
59        # Calcula los extremos del j-ésimo subintervalo: (x_{j-1}, x_j).
60        xjminus1 = a + (j - 1) * h

```

```
61     xj = a + j * h
62     # Calcula el punto medio de dicho subintervalo.
63     mj = (xjminus1 + xj) / 2.0
64     # Evalúa la fórmula de Simpson simple: f(x_{j-1}) + 4*f(m_j) + f(x_j).
65     Aj = f(xjminus1) + 4 * f(mj) + f(xj)
66
67     # Acumula la contribución del subintervalo.
68     SR = SR + Aj
69     # Termina el ciclo.
70     # -----
71
72     # Devuelve el valor final de la aproximación.
73     return (h / 3.0) * SR # La fórmula de la regla simple es (h/3) * [...].
```