# Sistema Matemático NeuroMate

Ronny Isaac Arellano Urgiles

***Abstract*—Este documento presenta el desarrollo e implementación de *NeuroMate*, un sistema matemático de escritorio diseñado para asistir en operaciones matemáticas avanzadas como cálculo de matrices, polinomios, vectores, representación gráfica 2D/3D, resolución de ecuaciones diferenciales ordinarias (EDO) y simulación epidemiológica mediante el modelo SVEIQR extendido con vacunación y cuarentena. La interfaz ha sido desarrollada con PyQt5, integrando múltiples módulos funcionales que permiten al usuario interactuar de manera intuitiva con herramientas matemáticas de propósito general. Este trabajo evidencia la viabilidad de crear una plataforma modular, educativa y de fácil acceso que fomente el aprendizaje práctico de las matemáticas aplicadas.**

***Index Terms*—PyQt5, Matemáticas Computacionales, Interfaz Gráfica, Modelo SVEIQR, Educación STEM.**

## I. Introducción

En el contexto educativo actual, la integración de herramientas tecnológicas que potencien el aprendizaje de las matemáticas se ha vuelto imprescindible. *NeuroMate* surge como una solución integral que permite al estudiante o profesional aplicar conocimientos matemáticos en un entorno interactivo, rápido y eficiente. A diferencia de otras plataformas de cálculo, *NeuroMate* se enfoca en la usabilidad, portabilidad y modularidad de sus componentes.

## II. Metodología

El desarrollo de *NeuroMate* se realizó utilizando Python 3 como lenguaje base, empleando PyQt5 para la construcción de interfaces gráficas [1]. La aplicación se estructura mediante un menú lateral que permite acceder a diferentes módulos:

### A. Matrices

- **Suma:** Para matrices $A = (a_{ij})$ y $B = (b_{ij})$, se define $C = A + B$ con
$$c_{ij} = a_{ij} + b_{ij}.$$
- **Resta:** Para matrices $A$ y $B$,
$$D = A - B, \quad d_{ij} = a_{ij} - b_{ij}.$$
- **Multiplicación:** Si $A$ es $m \times p$ y $B$ es $p \times n$, entonces
$$E = AB, \quad e_{ik} = \sum_{j=1}^{p} a_{ij} b_{jk}.$$
- **Transpuesta:** La transpuesta de $A = (a_{ij})$ de tamaño $m \times n$ es
$$A^T = (a_{ji}), \quad \text{tamaño } n \times m.$$
- **Inversa:** La matriz inversa $A^{-1}$ cumple
$$AA^{-1} = A^{-1}A = I,$$
donde $I$ es la matriz identidad [2].
- **Determinante:** El determinante $\det(A)$ es un escalar que indica propiedades de $A$ y se usa para resolver sistemas lineales (método de Gauss) [3], [4].

### B. Polinomios

- **Suma:** Para polinomios $P(x) = \sum_{i=0}^{n} a_i x^i$ y $Q(x) = \sum_{i=0}^{m} b_i x^i$,
$$R(x) = P(x) + Q(x) = \sum_{i=0}^{\max(n,m)} (a_i + b_i)x^i.$$
- **Multiplicación:**
$$S(x) = P(x)Q(x) = \sum_{k=0}^{n+m} c_k x^k, \quad \text{donde } c_k = \sum_{i+j=k} a_i b_j.$$
- **Derivación simbólica:**
$$\frac{dP}{dx} = \sum_{i=1}^{n} i a_i x^{i-1}.$$
- **Integración simbólica indefinida:**
$$\int ax^n dx = \frac{a}{n+1} x^{n+1} + C, \quad n \neq -1,$$
donde $C$ es la constante de integración.

### C. Vectores

Para vectores en $\mathbb{R}^3$, $\mathbf{a} = (a_x, a_y, a_z)$ y $\mathbf{b} = (b_x, b_y, b_z)$, se implementan:

- **Suma:** $\mathbf{a} + \mathbf{b} = (a_x + b_x, a_y + b_y, a_z + b_z)$.
- **Resta:** $\mathbf{a} - \mathbf{b} = (a_x - b_x, a_y - b_y, a_z - b_z)$.
- **Producto punto:**
$$\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z.$$
- **Producto cruz:**
$$\mathbf{a} \times \mathbf{b} = (a_y b_z - a_z b_y, \quad a_z b_x - a_x b_z, \quad a_x b_y - a_y b_x).$$
- **Magnitud:**
$$\|\mathbf{a}\| = \sqrt{a_x^2 + a_y^2 + a_z^2}.$$

[5]

### D. Ecuaciones Diferenciales Ordinarias (EDO)

Para resolver

$$y' = f(x, y), \quad y(x_0) = y_0,$$

se implementan métodos numéricos:

- **Euler:**
$$y_{i+1} = y_i + hf(x_i, y_i).$$
- **Heun:**
$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f\left(x_{i+1}, y_i + hf(x_i, y_i)\right)\right].$$
- **Runge-Kutta 4º orden (RK4):** Método clásico con cuatro evaluaciones por paso [8].
- **Taylor 2º orden:**
$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f\right).$$

### E. Vectores y Valores Propios

Se calculan valores propios $\lambda$ y vectores propios $\mathbf{v}$ de $A$ tales que

$$A\mathbf{v} = \lambda\mathbf{v},$$

utilizando la función `numpy.linalg.eig` [10].

### F. Generación de Números Aleatorios y Distribuciones

Se implementan algoritmos para números aleatorios uniformes y su transformación a distribuciones como normal (Box-Muller), exponencial, Poisson, binomial, gamma, beta, entre otras, basadas en referencias estándar [11], [12], [13].

### G. Integración Numérica por Monte Carlo

Para una función $f(x)$ en $[a, b]$, se estima la integral mediante

$$\int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i),$$

donde $x_i$ son muestras uniformes en $[a, b]$ [29].

### H. Sistema de Predicción Epidemiológica

*a) Motivación y justificación del módulo de predicción:* La inclusión del módulo de predicción epidemiológica en *NeuroMate* se fundamenta en la necesidad de contar con herramientas que permitan modelar y predecir la evolución de enfermedades infecciosas, considerando intervenciones sanitarias actuales como la vacunación y la cuarentena. Según el estudio realizado por Haq et al. [30], la pandemia de COVID-19 ha demostrado la importancia crítica de modelos matemáticos extendidos que incorporan estrategias de control para entender y anticipar la dinámica de contagio.

Este módulo responde a la problemática real de falta de sistemas accesibles y modulares que permitan no solo resolver ecuaciones diferenciales sino también simular escenarios epidemiológicos complejos, facilitando la toma de decisiones informadas en salud pública y educación. El modelo SVEIQR extendido con vacunación y cuarentena que se implementa aquí es una adaptación directa del planteamiento propuesto en dicho artículo, asegurando rigor científico y aplicabilidad práctica.

*b) Planteamiento formal del problema:* Dado un sistema de ecuaciones diferenciales no lineales que describen la dinámica poblacional bajo un contexto epidémico con vacunación y cuarentena, se desea determinar las funciones $S(t), V(t), E(t), I(t), Q(t), R(t)$ que representan la población susceptible, vacunada, expuesta, infectada, en cuarentena y recuperada respectivamente, para tiempos $t > 0$, con condiciones iniciales y parámetros epidemiológicos conocidos.

El objetivo es obtener soluciones aproximadas numéricas que permitan predecir la evolución temporal de la epidemia y evaluar el impacto de diferentes políticas sanitarias.

*c) Modelo matemático:* El sistema que se implementa corresponde al modelo SVEIQR extendido con vacunación y cuarentena, dado por:

$$\begin{cases} \frac{dS}{dt} = (1-\delta)\tau - (\mu + v)S - \beta S(E+I), \\ \frac{dV}{dt} = \delta\tau + vS - (\lambda + \mu)V, \\ \frac{dE}{dt} = \beta S(E+I) + \lambda V - (\mu + \alpha + \pi)E, \\ \frac{dI}{dt} = \pi E - (v + \mu)I, \\ \frac{dQ}{dt} = \alpha E + vI - (q + \mu)Q, \\ \frac{dR}{dt} = qQ - \mu R, \end{cases} \tag{1}$$

donde los parámetros tienen las siguientes interpretaciones:

- $\tau$: tasa de ingreso de individuos susceptibles
- $\delta$: proporción de vacunación directa
- $\mu$: tasa de mortalidad natural
- $v$: tasa de vacunación de susceptibles
- $\beta$: tasa de transmisión de la enfermedad
- $\lambda$: tasa a la cual la vacuna pierde eficacia y expone
- $\alpha$: tasa de ingreso a cuarentena de los expuestos
- $\pi$: tasa de progresión de expuestos a infectados
- $q$: tasa de recuperación desde cuarentena

*d) Método de solución numérica: Haar Wavelet Collocation (HWCT):* Para resolver el sistema no lineal (1), se utiliza el método Haar Wavelet Collocation [30], que aproxima las funciones desconocidas mediante una combinación lineal de funciones Haar $h_i(t)$ y sus integrales $p_i(t)$:

$$S(t) \approx S_0 + \sum_{i=1}^{M} a_i p_i(t), \quad V(t) \approx V_0 + \sum_{i=1}^{M} b_i p_i(t), \quad \ldots$$

donde los coeficientes $a_i, b_i, \ldots$ se determinan al imponer que las ecuaciones diferenciales se satisfacen en puntos de colación discretos $t_k$, formando un sistema algebraico no lineal que se resuelve con el método de Broyden.

*e) Ejemplo numérico paso a paso:* Para hacer comprensible el método, consideremos los siguientes parámetros y condiciones iniciales ficticias:

Suponiendo que elegimos $M = 4$ funciones Haar para la aproximación, el método genera un sistema algebraico con incógnitas los coeficientes $a_i, b_i, \ldots$.

TABLE I
PARÁMETROS EPIDEMIOLÓGICOS USADOS EN EL EJEMPLO

| Parámetro | Valor |
|---|---|
| $\tau$ | 0.5 |
| $\delta$ | 0.2 |
| $\mu$ | 0.01 |
| $v$ | 0.1 |
| $\beta$ | 0.3 |
| $\lambda$ | 0.05 |
| $\alpha$ | 0.05 |
| $\pi$ | 0.2 |
| $q$ | 0.1 |

TABLE II
CONDICIONES INICIALES

| Variable | Valor inicial |
|---|---|
| $S(0)$ | 1000 |
| $V(0)$ | 200 |
| $E(0)$ | 50 |
| $I(0)$ | 30 |
| $Q(0)$ | 10 |
| $R(0)$ | 5 |

Un ejemplo simplificado de los valores aproximados para los coeficientes resultantes podría ser (valores ficticios para ilustración):

TABLE III
COEFICIENTES APROXIMADOS DEL MÉTODO HWCT

| Coeficiente | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $S$ | -5.0 | 3.2 | -1.1 | 0.5 |
| $V$ | 1.0 | -0.8 | 0.3 | -0.1 |
| $E$ | 0.5 | -0.3 | 0.2 | -0.05 |
| $I$ | -0.2 | 0.1 | -0.05 | 0.02 |
| $Q$ | 0.1 | -0.05 | 0.02 | -0.01 |
| $R$ | 0.05 | 0.02 | -0.01 | 0.005 |

Con estos coeficientes, la función aproximada $S(t)$ se calcula para cualquier $t$ dentro del intervalo de simulación usando:

$$S(t) \approx S_0 + \sum_{i=1}^{4} a_i p_i(t),$$

y análogamente para las demás variables.

Finalmente, se obtienen las gráficas de la evolución temporal que facilitan la interpretación del comportamiento epidémico.

## III. IMPLEMENTACIÓN NUMÉRICA CON PYTHON

El modelo matemático SEIR avanzado con vacunación y mortalidad fue implementado en Python utilizando la librería `scipy.integrate.odeint` para la solución numérica del sistema de ecuaciones diferenciales ordinarias.

En esta implementación:

- La vacunación activa y su cobertura disminuyen el número de susceptibles y aumentan la población recuperada en función de la efectividad.
- La tasa de transmisión, incubación, recuperación y mortalidad se parametrizan según la realidad epidemiológica.
- El sistema se resuelve numéricamente para un intervalo de tiempo definido usando la función `odeint`.

Las variables iniciales y parámetros se obtienen desde una interfaz gráfica desarrollada en PyQt5 que permite ajustar las condiciones y observar resultados en tiempo real mediante gráficos interactivos.

Esto facilita la experimentación con diferentes escenarios epidemiológicos y políticas de salud pública, aportando una herramienta educativa y analítica potente.

## IV. RESULTADOS

*NeuroMate* permite ejecutar operaciones matemáticas complejas y simulaciones en tiempo real. Se verificó la exactitud en cada módulo y se generaron gráficas interactivas para el modelo epidemiológico.

## V. DISCUSIÓN

El sistema consolida diversas herramientas en una interfaz intuitiva. La arquitectura modular facilita futuras expansiones y la integración de nuevas funcionalidades.

## VI. CONCLUSIONES Y TRABAJO FUTURO

Se ha desarrollado una plataforma educativa que combina cálculo y simulación con experiencia de usuario fluida. Se planea incluir exportación de datos, ejercicios interactivos e integración en la nube.

## REFERENCES


[1] M. Summerfield, *Rapid GUI Programming with Python and Qt*. Prentice Hall, 2007.

[2] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley-Cambridge Press, 2016.

[3] R. A. Horn y C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.

[4] D. C. Lay, *Linear Algebra and Its Applications*, 4th ed. Pearson, 2011.

[5] J. VanderPlas, *Python Data Science Handbook*. O'Reilly Media, 2016.

[6] E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed., Wiley, 2011.

[7] R. L. Burden y J. D. Faires, *Numerical Analysis*, 9th ed., Brooks/Cole, 2011.

[8] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, 1971.

[9] J. Stoer y R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed., Springer, 2002.

[10] W. McKinney, *Python for Data Analysis*, O'Reilly, 2017.

[11] M. Matsumoto y T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.

[12] G. Marsaglia, "Xorshift RNGs," *Journal of Statistical Software*, vol. 8, no. 14, 2003.

[13] P. O'Neill, "PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation," Harvard University Tech Report, 2014.

[14] F. Panneton, P. L'Ecuyer, y M. Matsumoto, "Improved Long-Period Generators Based on Linear Recurrences Modulo 2," *ACM Transactions on Mathematical Software*, vol. 32, no. 1, pp. 1–16, 2006.

[15] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed., Addison-Wesley, 1997.

[16] L. Blum, M. Blum, y M. Shub, "A Simple Unpredictable Pseudo-Random Number Generator," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 364–383, 1986.

[17] R. C. Tausworthe, "Random Numbers Generated by Linear Recurrences Modulo 2," *Mathematics of Computation*, vol. 19, no. 90, pp. 201–209, 1965.

[18] J. von Neumann, "Various Techniques Used in Connection with Random Digits," in *Monte Carlo Method*, National Bureau of Standards Applied Math Series, vol. 12, pp. 36–38, 1951.

[19] J. Weyl, "Über die Gleichverteilung von Zahlen mod Eins," *Mathematische Annalen*, vol. 77, no. 3, pp. 313–352, 1916.

[20] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1, 3rd ed., Wiley, 1968.

[21] H. Cramér, *Mathematical Methods of Statistics*, Princeton University Press, 1946.

[22] A. Papoulis y S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed., McGraw-Hill, 2002.

[23] J. A. Rice, *Mathematical Statistics and Data Analysis*, 3rd ed., Duxbury Press, 2006.

[24] A. K. Gupta y D. K. Nagar, *Matrix Variate Distributions*, Chapman & Hall, 2000.

[25] G. E. P. Box, W. G. Hunter, y J. S. Hunter, *Statistics for Experimenters*, Wiley, 1978.

[26] R. J. Larsen y M. L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, 5th ed., Pearson, 2012.

[27] S. Ross, *Introduction to Probability Models*, 11th ed., Academic Press, 2014.

[28] J. M. Hilbe, *Negative Binomial Regression*, 2nd ed., Cambridge University Press, 2011.

[29] J. M. Hammersley y D. C. Handscomb, *Monte Carlo Methods*, Methuen, 1964.

[30] I. U. Haq, N. Ullah, N. Ali, y K. S. Nisar, "A New Mathematical Model of COVID-19 with Quarantine and Vaccination," *Mathematics*, vol. 11, no. 1, p. 142, 2023, doi:10.3390/math11010142.