

# RESOLUCIÓN DE UNA ECUACIÓN DIFERENCIAL ALEATORIA UTILIZANDO EL MÉTODO DE MONTECARLO

---

***Universidad Internacional de la Rioja (UNIR)***

---

***Cuatrimestre II***

---

**Máster en Ingeniería Matemática y Computación**

***Ecuaciones Diferenciales Estocásticas y Aplicaciones***

**Autor:** Javier Blanco Álvarez

## RESUMEN

En este estudio, se aborda un problema de valor inicial aleatorio (PVIA) fundamentado en la ecuación diferencial logística, cuyos valores iniciales se distribuyen uniformemente en el intervalo [0,1]. El propósito es aproximar la solución del PVIA con valores iniciales aleatorios empleando simulaciones de Monte Carlo. Tras 600 simulaciones, se generaron múltiples trayectorias estocásticas para aproximar la solución, visualizando trayectorias y propiedades estadísticas como la media y la varianza. El análisis estadístico arrojó luz sobre la estabilidad y convergencia del proceso estocástico. Se aplicó análisis de histogramas y gráficos de probabilidad acumulada para comprender la distribución y comportamiento de las soluciones en puntos de tiempo discretos. El histograma acumulativo reflejó la acumulación de probabilidades de soluciones en distintos momentos, proporcionando información sobre la convergencia y reducción de la incertidumbre con el tiempo. Además, se investigaron soluciones simbólicas y numéricas, implementadas en MATLAB para ambos métodos. Las soluciones simbólicas se obtuvieron con las herramientas de cálculo simbólico de MATLAB, mientras que las numéricas se lograron mediante el método de Euler. El análisis comparativo destacó mejoras sustanciales en la eficiencia computacional con el enfoque numérico, evidenciando una mejora del 5500% en el tiempo de ejecución.

**Palabras Clave:** Procesos estocásticos, Método Monte Carlo, MATLAB, Problemas de valor inicial aleatorio.

## 1. Planteamiento del problema

Consideremos el siguiente problema de valor inicial aleatorio (PVIA), basado en la ecuación diferencial logística:

$$X'(t) = a(b - X(t)); X(0) = X_0$$

De esta función sabemos que:

$$a, b \in R$$

Y que los valores iniciales se ajustan a una distribución uniforme:

$$X_0 \sim U([0,1])$$

Se pide encontrar una solución aproximada de este PVIA de valores iniciales aleatorios mediante el método de Montecarlo. Además, se pide encontrar la esperanza matemática y la varianza a lo largo de los intervalos  $t$  de la PVIA

## 2. Parámetros de entrada

En la tabla 1 de este informe se encuentran los datos de entrada que servirán como partida para la resolución de la práctica.

Tabla 1. Resumen de los datos y parámetros de entrada usados en esta práctica. Elaboración propia

Parámetro	Valor
a (PVIA)	1.0
b (PVIA)	1.0
Distribución de X0	Uniforme [0,1]
Intervalos de t	[0.0, 5.0]
Numero de Nodos (N)	20
Simulaciones	600
Funciones usadas	dsolve, subs, ode45, Euler

## 3. Solución exacta de la PVIA

$$X'(t) = a(b - X(t)); X(0) = X_0$$

Utilizando la notación diferencial de Leibniz, la expresión anterior queda expresada:

$$\frac{dx}{dt} = a(b - X(t))$$

Aplicando separación de variables:

$$\frac{\frac{dx}{dt}}{(b - X(t))} = a dt$$

Esta EDO puede ser resuelta mediante la técnica de integración directa con respecto a  $t$  (Malumbres, 1996)

$$\int \frac{\frac{dx}{dt}}{(b - X(t))} = \int a dt$$

Aplicando método de cambio de variable para resolver la integral, tenemos:

$$U = b - X(t)$$

$$dU = -\frac{dx}{dt}$$

Por tanto, la integral del lado derecho de la igualdad es resuelta al mismo tiempo que se hace el cambio de variable en el lado izquierdo.

$$\int \frac{-dU}{U} = at + C$$

Resolviendo la integral del lado izquierdo de la igualdad, tenemos:

$$-\ln U + C_1 = at + C_2$$

Retomando el cambio de variable y multiplicando por -1 en ambos lados de la igualdad, obtenemos:

$$(-1) - \ln(b - X(t)) + C_1 = at + C_2 (-1)$$

$$\ln(b - X(t)) + C_1 = -at + C_2$$

Seguidamente, aplicamos exponencial en ambos lados de la igualdad y se obtiene:

$$e^{\ln(b - X(t)) + C_1} = e^{-at + C_2}$$

Lo que se traduce en:

$$b - X(t) + C_1 = e^{-at + C_2}$$

Si despejamos  $X(t)$  obtenemos:

$$X(t) = (b + C_1) * e^{-at}$$

Si consideramos las condiciones iniciales del problema de valor inicial tenemos que:

$$X_o = b + C_1$$

$$C_1 = X_o - b$$

Así, la solución exacta que satisface la EDO de primer orden planteada en este problema es:

$$X(t) = b + [X_o - b] * e^{-at}$$

Aunque para efectos prácticos de este laboratorio no es necesario hacer este paso, conocer la solución exacta nos permite implementar el Método de Montecarlo en otros sistemas informáticos más ligeros y con alta fiabilidad numérica como **Microsoft Excel 2023** o **Google Sheets**.

## 4. Implementación Montecarlo

El procedimiento de **Monte Carlo** implementado en **MATLAB** para esta práctica ([Ver anexos](#)) se utiliza para resolver un problema de valor inicial aleatorio (PVID) mediante la simulación de múltiples trayectorias estocásticas (Gubernatis, 2019). El procedimiento llevado a cabo comienza estableciendo el número de simulaciones, definido por la variable *sims*, que representa el número de trayectorias estocásticas que se generarán. Para generar valores aleatorios de condiciones iniciales para cada simulación se ha utilizado una distribución uniforme en el intervalo [0,1] mediante la función *unifrnd*.

La función *unifrnd* de MATLAB genera números aleatorios uniformes continuos y se basa en la congruencia lineal, por lo que permite obtener un muestro aleatorio uniforme adecuado para la implementación de Monte Carlo (Entacher, 1997).

En cada simulación, La ecuación diferencial que describe el problema de valor inicial aleatorio (PVID) es tratada simbólicamente y luego transformada a sistema numérico (*double*) para que la integración numérica de la función *dsolve* de **MATLAB** obtenga la solución de  $X(t)$  para cada Valor Inicial Aleatorio evaluado. La condición inicial para cada simulación se establece utilizando el valor aleatorio correspondiente generado anteriormente.

El intervalo de tiempo **[a, b]** se divide en **N** pasos de tamaño **h**, donde a y b representan los límites inferior y superior del intervalo de tiempo, respectivamente, y N es el número de pasos. Para cada simulación, se itera a través de estos pasos de tiempo y se calcula el valor de la solución en cada punto de tiempo mediante la evaluación numérica de la solución simbólica en esos puntos de tiempo específicos. Finalmente, se trazan las soluciones de todas las simulaciones en una sola figura para visualizar las trayectorias estocásticas resultantes ([ver figura 1](#)).

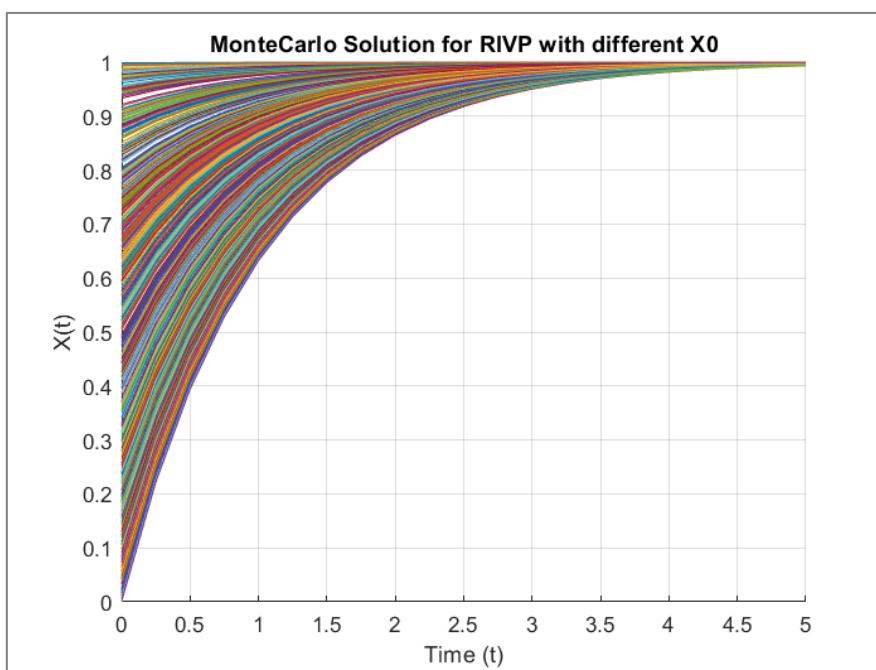


Figura 1. Solución final de Montecarlo para la PVID evaluado el intervalo  $t \in [0, 5]$  con diferentes condiciones iniciales. Elaboración propia

Se llevaron a cabo **600 simulaciones** usando el método de Monte Carlo para aproximar la solución de  $X(t)$  en el intervalo  $t \in [0, 5]$ . En la figura 1 también puede apreciarse la convergencia de los valores finales cercana a 1. En cuanto a las características del método de Monte Carlo, se han de determinar la esperanza matemática, **E** y la varianza de las simulaciones.

### 3.1 Esperanza Matemática y Varianza

En el contexto de las simulaciones Monte Carlo, donde se están generando múltiples trayectorias estocásticas, la esperanza matemática del proceso viene dada por la media aritmética del valor de  $X(t)$  en el instante  $t$  en las simulaciones, mientras que la varianza de la muestra en sí misma proporciona una estimación de la dispersión de los datos alrededor de la media (Sahai, Khurshid, Ojeda, & Velasco, 2009) (ver anexos).

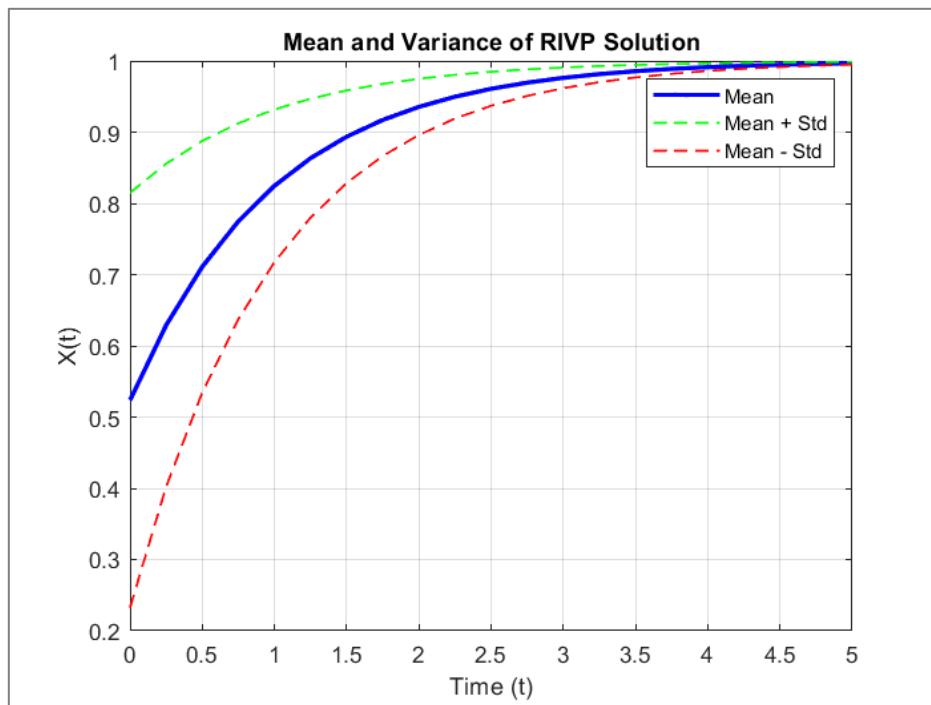


Figura 2. Esperanza Matemática y Varianza de la solución aproximada de PVIAP tras las simulaciones de Monte Carlo. Elaboración propia.

En la figura 2 se muestra el resultado del cálculo de la esperanza matemática sobre las simulaciones del método de Monte Carlo (**línea azul**), junto con los tramas de banda superior (**verde claro**) y banda inferior (**rojo**), que representan los intervalos de confianza alrededor de la media. Se observa que a medida que avanzamos en el intervalo de tiempo  $t$   $[0, 5]$  existe una mayor consistencia de los datos debido al menor grado de dispersión de las soluciones obtenidas con Monte Carlo. Esto sugiere una mayor estabilidad o menos incertidumbre en las soluciones del proceso estocástico a medida que pasa el tiempo (Mathelin, Hussaini, & Zang, 2005); siendo además un indicativo que hemos obtenido una solución con una aproximación significativa sobre el fenómeno que se desea estudiar con la PVIAP logística (Mendoza & Parra, 2012).

## 5. Análisis estadístico

Una vez terminadas las simulaciones del método de Monte Carlo, resulta conveniente para el investigador analizar la distribución de las frecuencias de los valores de  $X(t)$  para determinados valores de  $t$  y dados los 600 valores iniciales probados en el modelo logístico. Para ello, se ha discretizado el intervalo  $t$   $[0,5]$ , en 6 partes ( $t = 0,1,2,3,4,5,6$ ) y se han dibujado histogramas sobre las simulaciones para cada una de las partes discretas del intervalo  $t$  (**ver figura 3**). Para cada uno de los valores discretos se observa que la mayor frecuencia de valores de  $X(t)$  oscilan entre 0.6 y 0.99 en cada una de las 600 condiciones iniciales aleatorias simuladas con Monte Carlo.

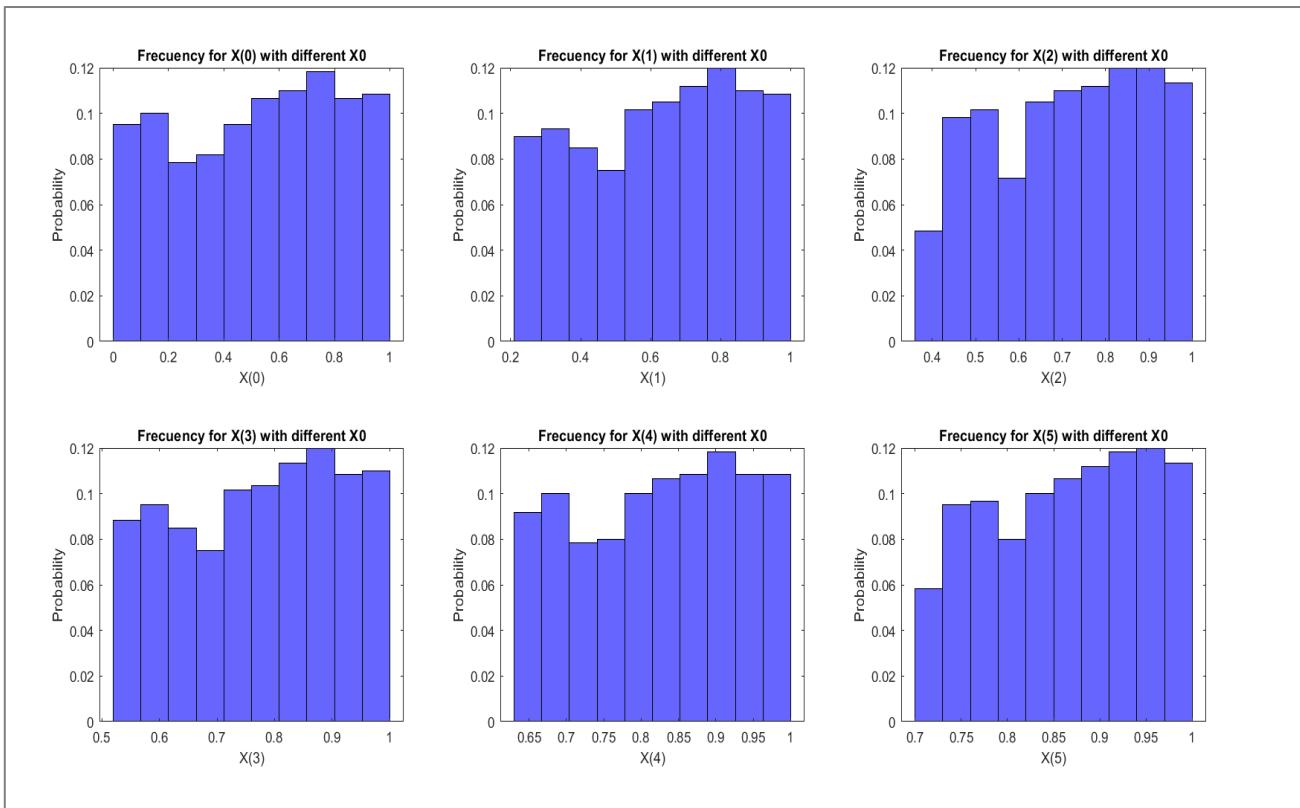


Figura 3. Histogramas de frecuencia para tiempo discreto  $t [0,5]$  para diferentes valores de condición inicial  $X_0$ . Elaboración propia

En cuanto a la distribución acumulada de las soluciones del proceso estocástico en un punto de tiempo específico  $X(t)$  (figura 4), observamos que medida que se avanza a lo largo del eje x, la probabilidad acumulada aumenta, lo que indica la probabilidad acumulada de que las soluciones del proceso estocástico sean menores o iguales a un valor final de  $X(5) = 1$ .

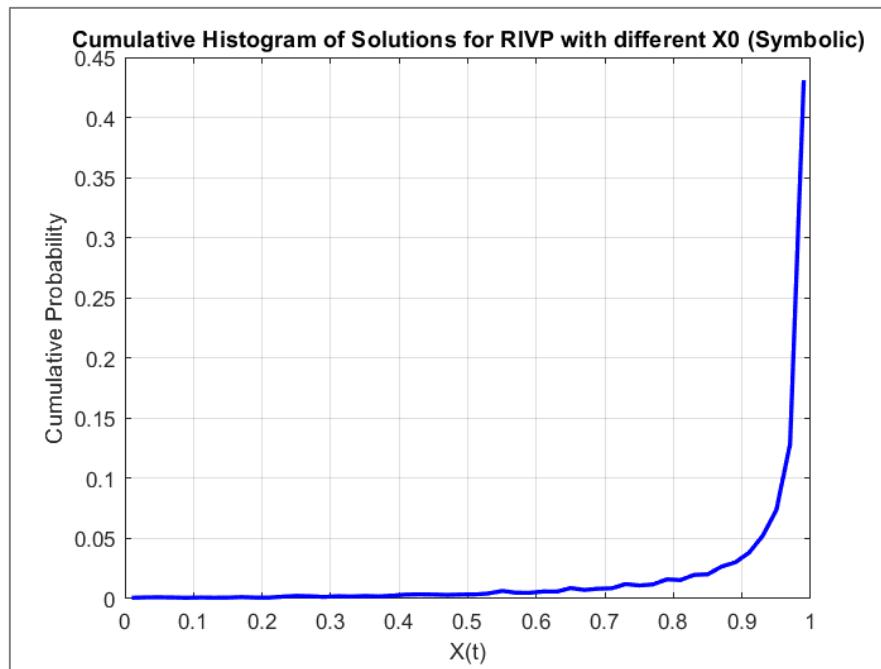


Figura 4. representación gráfica del histograma acumulado de las soluciones de  $X(t)$  a diferentes condiciones iniciales. Elaboración propia.

También se observa una pendiente muy pronunciada de la curva a partir de  $X(t) = 0.8$ , lo que indica una tasa de cambio significativa en la probabilidad acumulada hasta su valor final  $X(t) = 1$ . En este punto se observa una rápida convergencia de la solución del PVI en el intervalo  $t [0, 5]$ .

## 6. Eficiencia Computacional

Tal como se ha descrito en los apartados anteriores, el método de Monte Carlo aplicado en esta práctica incluyó 600 simulaciones del proceso estocástico de una PVI logística. Para cada una de estas simulaciones, se tenía resolver la PVI mediante el uso de funciones simbólicas, que luego son evaluadas en el intervalo  $t \in [0, 5]$  y transformadas a objetos numéricos en MATLAB. Este procedimiento resulta computacionalmente ineficiente, tomando un tiempo de ejecución considerable aun cuando MATLAB está optimizado para este tipo de escenarios.

Para mejorar la eficiencia computacional de las simulaciones de Monte Carlo, se optó por el uso de métodos numéricos aplicados a PVI. Este tipo de métodos logran una aproximación numérica de rápida convergencia evitando el paso de calcular la solución exacta en cada iteración. Para ello, se ha

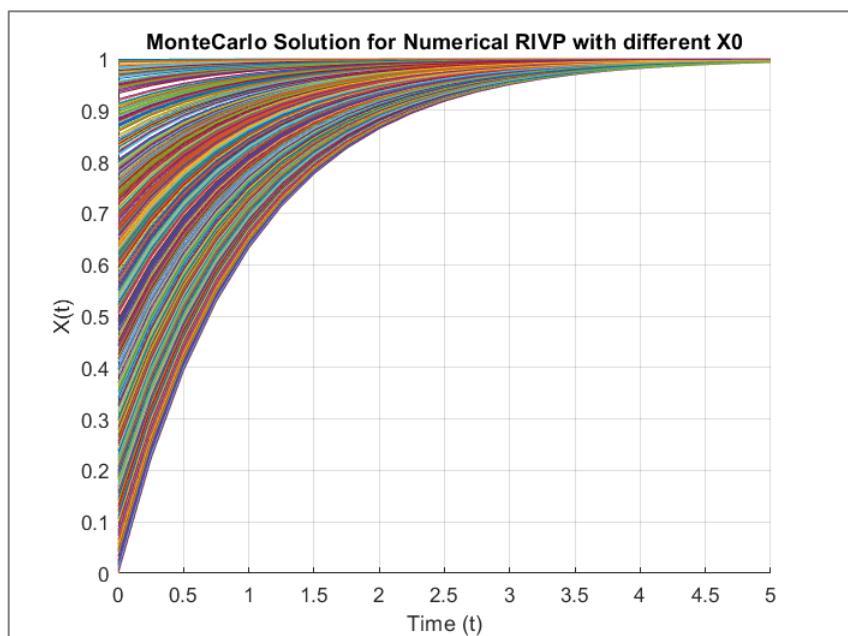


Figura 5. Solución numérica de Montecarlo para la PVI evaluado el intervalo  $t \in [0, 5]$  con diferentes condiciones iniciales. Elaboración propia

utilizado el método de Euler (Abraha, 2020), siendo este uno de los más sencillos de implementar y aplicar en el campo del cálculo numérico.

El resultado de esta simulación numérica con el método de Euler en MATLAB ([ver anexos](#)) se muestra en la figura 5. Al comparar esta solución con la obtenida en el procedimiento simbólico ([ver figura 6](#)) se observa una diferencia significativa en los tiempos de ejecución, pasando de 0.055 segundo en la fase simbólica, al 0.001 segundos para el procedimiento numérico con Euler. Esto representa una mejor del 5500% en cuanto a eficiencia computacional.

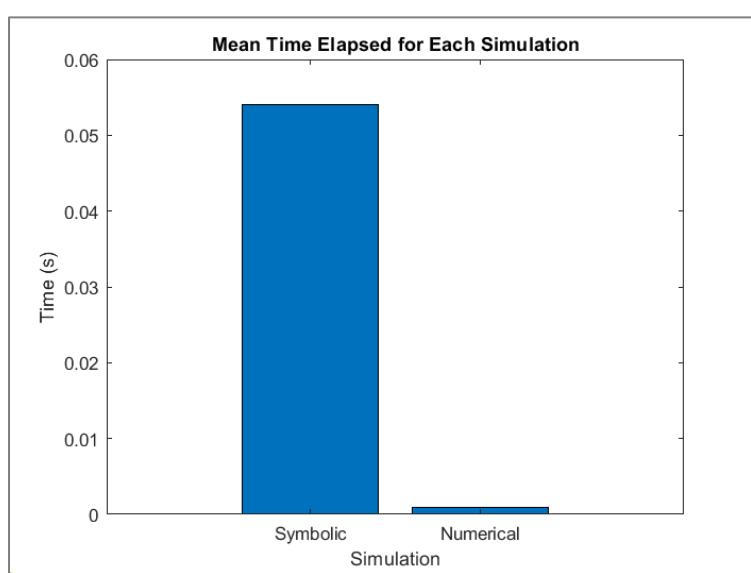


Figura 6. Tiempo medio transcurrido entre las simulaciones de Monte Carlo tanto en la solución simbólica (exacta) como numérica (Euler). Elaboración propia

A tenor de estos resultados mostrados en la figura 6, se recomienda el uso de métodos numéricos para resolución de PVI en el contexto del método de Monte Carlo.

## REFERENCIAS

- Abraha, J. D. (2020). Comparison of numerical methods for system of first order ordinary differential equations. *Pure Appl. Math. J.*, 32.
- Entacher, K. (1997). *A collection of selected pseudorandom number generators with linear structures*. Recuperado el 28 de 4 de 2024, de [https://www.researchgate.net/publication/2683298\\_A\\_Collection\\_of\\_Selected\\_Pseudorandom\\_Number\\_Generators\\_With\\_Linear\\_Structures](https://www.researchgate.net/publication/2683298_A_Collection_of_Selected_Pseudorandom_Number_Generators_With_Linear_Structures)
- Gubernatis, J. E. (2019). The Monte Carlo Method:From Then to Now. *Notices of the American Mathematical Society*, 66(1), 107-107. Recuperado el 28 de 4 de 2024, de <https://dialnet.unirioja.es/servlet/articulo?codigo=6810668>
- Malumbres, J. L. (1996). *Métodos clásicos de resolución de ecuaciones diferenciales ordinarias*. Recuperado el 28 de 4 de 2024, de <http://cartagena99.com/recursos/matemáticas/apuntes/libroecuadiferenc.pdf>
- Mathelin, L., Hussaini, M. Y., & Zang, T. A. (2005). Stochastic Approaches to Uncertainty Quantification in CFD Simulations. *Numerical Algorithms*, 38(1), 209-236. Recuperado el 28 de 4 de 2024, de <https://link.springer.com/article/10.1007/bf02810624>
- Mendoza, J. E., & Parra, G. C. (2012). *Comparación de caos polinomial y Monte Carlo para ecuaciones diferenciales ordinarias aleatorias*. Recuperado el 28 de 4 de 2024, de <http://revistas.saber.ula.ve/index.php/cienciaeingenieria/article/download/4104/3908>
- Sahai, H., Khurshid, A., Ojeda, M. M., & Velasco, F. (2009). *Simultaneous confidence intervals for variance components in two-way balanced crossed classification random effects model with interaction*. Recuperado el 28 de 4 de 2024, de <http://rev-inv-ope.univ-paris1.fr/files/30309/30309-07.pdf>

## ANEXOS

- Fichero MATLAB de ejecución

```
clear, clc
format shortG

%% INITIAL PARAMETERS
% Random Initial Value Problem (RIVP)
% X'(t) = a*(b-X(t)); X(0) = X0
% t € [0, 5]

a=0;
b = 5;
N = 20;

%% MONTECARLO IMPLEMENTATION
% Simulations
sims = 600;
rv = unifrnd(0,1,[sims 1]);
X_sol_sym = zeros(N+1, sims);
h = (b - a) / N;
syms X(t)
a_p=1;
b_p=1;
ode_sym = diff(X,t) == a_p.* (b_p-X);
figure;
hold on;
for i =1:sims
    tic;
    t = a:h:b;
    cond = X(0) == rv(i);
    X_sol_sym(:,i) = double(subs(dsolve(ode_sym, cond),t));
    plot(t, X_sol_sym(:,i))
    time_elapsed_for_sym(i) = toc;
end
grid on;
xlabel('Time (t)');
ylabel('X(t)');
title('MonteCarlo Solution for RIVP with different X0');

% Expected Value and Variance
mean_X = mean(X_sol_sym, 2);
var_X = var(X_sol_sym, 0, 2);

% Visual Results
figure;
t1 = a:h:b;
plot(t1, mean_X, 'b-', 'LineWidth', 2);
hold on;
plot(t1, mean_X + sqrt(var_X), 'g--', 'LineWidth', 1);
plot(t1, mean_X - sqrt(var_X), 'r--', 'LineWidth', 1);
grid on;
xlabel('Time (t)');
ylabel('X(t)');
title('Mean and Variance of RIVP Solution');
legend('Mean', 'Mean + Std', 'Mean - Std');
```

```

%% NUMERICAL SOLUTION (JUST FOR FUN)
X_sol = zeros(N+1,sims);
t_sol = zeros(N+1,sims);

figure;
hold on;
for i=1:sims
    tic;
    [t_sol, X_sol(:,i)] = AdamsBashforth4('logitEqn', a, b, N, rv(i));
    plot(t_sol, X_sol(:,i));
    time_elapsed_for_num(i) = toc;
end
grid on;
xlabel('Time (t)');
ylabel('X(t)');
title('MonteCarlo Solution for Numerical RIVP with different X0');

%% STATISTIC ANALYSIS
% Histograms for each time step
num_steps = 6; % Number of time steps
figure;
for step = 1:num_steps
    subplot(2,3,step); % 2x3 subplots
    histogram(X_sol_sym(step,:), 10, 'Normalization', 'probability', 'FaceColor', 'b');
    xlabel(sprintf('X(%d)', step-1));
    ylabel('Probability');
    title(sprintf('Frequency for X(%d) with different X0', step-1));
end

% Cumulative Histogram of Solutions
figure;
[N, edges] = histcounts(X_sol_sym, 'Normalization', 'probability');
bin_centers = (edges(1:end-1) + edges(2:end)) / 2;
cumulative_probs = cumsum(N, 1);
plot(bin_centers, cumulative_probs, 'b-', 'LineWidth', 2);
xlabel('X(t)');
ylabel('Cumulative Probability');
title('Cumulative Histogram of Solutions for RIVP with different X0 (Symbolic)');
grid on;

%% COMPUTATIONAL EFICIENCY
% Time elapsed execution plots
figure;
bar([mean(time_elapsed_for_sym); mean(time_elapsed_for_num)]);
xlabel('Simulation');
ylabel('Time (s)');
title('Mean Time Elapsed for Each Simulation');
xticklabels({'Symbolic', 'Numerical'});

```

- **Método Numérico de Euler para PVI de primer orden en MATLAB**

```

function [t,y] = Euler(f, a, b, N, ya)
%EULER Solves an ordinary differential equation (ODE) using the Euler method.
% [t, y] = Euler(f, a, b, N, ya) solves the ODE y' = f(t, y) over the interval
t=[a, b]
% with N steps using the Euler method. The initial condition is y(a) = ya.
%
% Input:
%   f: Function defining the ODE y' = f(t, y).
%   a: Lower limit of the interval.
%   b: Upper limit of the interval.
%   N: Number of discretization steps.
%   ya: Initial value y(a).
%
% Output:
%   t: Vector of discrete time points.
%   y: Vector of approximate solutions corresponding to the times in t.
%
% Example:
%   % Solve the ODE y' = -y with y(0) = 1 over the interval [0, 1] with 10
steps.
%   f = @(t, y) -y;
%   [t, y] = Euler(f, 0, 1, 10, 1);
%   plot(t, y);
%   xlabel('Time (t)');
%   ylabel('Solution (y)');
%   title('Solution of the ODE y' = -y using the Euler method');
%
h = (b - a)/ N;
t = a:h:b;
t = t(:);
y = zeros(N+1, 1);
y(1) = ya;
for k = 1:N
    y(k+1) = y(k) + h*feval(f, t(k), y(k));
end
end

```