# Método de Euler para sistemas de EDOs

Ander Murua
Curso de *Computación en Ciencia e Ingeniería: Simulación Numérica*
Máster Universitario en Ingeniería Computacional y Sistemas Inteligentes

## 1. Sistemas de ecuaciones de EDOs: Un ejemplo de modelo presa-depredador con caza humana periódica

Consideremos inicialmente el sistema de Lotka-Volterra,

$$\frac{d}{dt}v = (a - bw)v, \quad \frac{d}{dt}w = (cv - d)w,$$

con los valores de los parámetros $a = d = 1$, $b = 1/50$, $c = 3/100$, y bajo las condiciones iniciales

$$v(0) = 51, \quad w(0) = 33.$$

El tiempo $t$ se mide en meses. La variable de estado $v$ representa el número de alces de un parque natural protegido, y $w$ el número de lobos. Supongamos ahora que los guardas del parque cazan de forma periódica (anualmente) cierto porcentaje de alces, con distinta actividad de caza dependiendo de los meses del año. Esto se podría modelar modificando la primera ecuación del sistema de EDOs anterior como sigue:

$$\frac{d}{dt}v = \left(1 - \frac{w}{50} - g(t)\right)\, v, \quad \frac{d}{dt}w = \left(\frac{3\,v}{100} - 1\right)\, w,$$

donde $g(t)$ es una función períodica de $t$ con un período de 12 meses. Como ejemplo ilustrativo, aquí consideraremos la función

$$g(t) = \frac{1}{20}(\sin(\pi\, t/12))^2. \tag{1}$$

En concreto, ello da lugar al siguiente *problema de valor inicial*:

$$\frac{d}{dt}v = \left(1 - \frac{w}{50} - g(t)\right)\, v, \quad \frac{d}{dt}w = \left(\frac{3\,v}{100} - 1\right)\, w, \quad v(0) = 51, \quad w(0) = 33, \tag{2}$$

que tiene solución única $(v(t), w(t))$. Se trata pues de simular la evolución de las poblaciones de presas y depredadores a lo largo del tiempo $t$ calculando $v(t)$ y $w(t)$ para distintos valores de $t$,

$$t_0 = 0, \quad t_1 = h, \quad t_2 = 2h, \quad t_3 = 3h, \ldots \tag{3}$$

donde, por ejemplo, $h = 1/40$.

## 2. El método de Euler para sistemas de EDOs

El valor del vector de estado $(v(t), w(t))$ para $t = t_0 = 0$ es conocido, y queremos primeramente calcular su valor para $t = t_1 = t_0 + h$, es decir $v(t_1)$ y $w(t_1)$. Sabemos por las ecuaciones diferenciales en (2) cómo calcular $\frac{dv}{dt}$ y $\frac{dw}{dt}$ (que denotaremos como es habitual de la forma $v'(t)$ y $w'(t)$ respectivamente) en función de $v(t)$ y $w(t)$. En el caso particular de $t = t_0 = 0$,

$$v'(0) = (1 - w(0)/50 - g(0))\, v(0) = \frac{867}{50}, \quad w'(0) = (3\, v(0)/100 - 1)\, w(0) = \frac{1749}{100}.$$

Por otro lado, puesto que $h = 1/40$ es pequeño, tenemos que

$$v'(t_0) \approx \frac{v(t_1) - v(t_0)}{h}, \quad w'(t_0) \approx \frac{w(t_1) - w(t_0)}{h},$$

(recordemos que $t_1 = t_0 + h$,) de donde obtenemos las igualdades aproximadas

$$\begin{aligned} v(t_1) &\approx v(t_0) + hv'(t_0) = 51 + h\,\frac{867}{50} = 51{,}4335 \\ w(t_1) &\approx w(t_0) + hw'(t_0) = 33 + h\,\frac{1749}{100} = 33{,}4373. \end{aligned}$$

Este mismo procedimiento se puede aplicar para calcular (de forma aproximada) $v(t_2)$ y $w(t_2)$ a partir de $v(t_1)$ y $w(t_1)$, y así sucesívamente. En general, podemos aproximar los valores de $v(t_{j+1})$ y $w(t_{j+1})$ a partir de los valores $v(t_j)$ y $w(t_j)$, basándonos en el siguiente razonamiento: Por un lado, tenemos la igualdades aproximadas

$$v'(t) \approx \frac{v(t+h) - v(t)}{h}, \quad w'(t) \approx \frac{w(t+h) - w(t)}{h},$$

válidas para cualquier $t$ (siempre que $h$ sea suficientemente pequeño), y por tanto,

$$v(t+h) \approx v(t) + h\, v'(t), \quad w(t+h) \approx w(t) + h\, w'(t),$$

y teniendo en cuenta la EDO (2),

$$\begin{aligned} v(t+h) &\approx v(t) + h\, v'(t), \quad \text{donde} \quad v'(t) = \left(1 - \frac{w(t)}{50} - g(t)\right) v(t), \\ w(t+h) &\approx w(t) + h\, w'(t), \quad \text{donde} \quad w'(t) = \left(\frac{3\, v(t)}{100} - 1\right) w(t). \end{aligned}$$

Particularizando esto para $t = t_0, t_1, t_2, t_3, \ldots$, tenemos que para cada $j = 0, 1, 2, 3, \ldots$

$$\begin{aligned} v(t_{j+1}) &\approx v(t_j) + h\, v'(t_j), \quad \text{donde} \quad v'(t_j) = \left(1 - \frac{w(t_j)}{50} - g(t_j)\right) v(t_j), \\ w(t_{j+1}) &\approx w(t_j) + h\, w'(t_j), \quad \text{donde} \quad w'(t_j) = \left(\frac{3\, v(t_j)}{100} - 1\right) w(t_j). \end{aligned}$$

Llegamos de esta forma al **algoritmo de Euler**, que partiendo de $t_0 = 0$ y $(v_0, w_0) = (51, 33)$, y fijada una *longitud de paso* $h$ suficientemente pequeña, calcula las aproximaciones

$$(v_1, w_1) \approx (v(t_1), w(t_1)), \quad (v_2, w_2) \approx (v(t_2), w(t_2)), \quad (v_3, w_3) \approx (v(t_3), w(t_3)), \ldots$$

(así como las aproximaciones $(v_j', w_j') \approx (v'(t_j), w'(t_j))$ para $j = 0, 1, 2, 3, \ldots$) como sigue:

$$\boxed{\begin{array}{lcl} \text{para} \quad j & = & 0,1,2,3,\ldots \\ & & t_{j+1} = t_j + h, \\ & & v_j' = \left(1 - \dfrac{w_j}{50} - g(t_j)\right) v_j, \qquad w_j' = \left(\dfrac{3\, v_j}{100} - 1\right) w_j, \\ & & v_{j+1} = v_j + h\, v_j', \qquad w_{j+1} = w_j + h\, w_j'. \end{array}}$$

Este algoritmo se puede escribir de forma más compacta usando para cada $j$ la notación vectorial $u_j = (v_j, w_j)$, $u_j' = (v_j', w_j')$:

$$\boxed{\begin{array}{lcl} \text{para} \quad j & = & 0,1,2,3,\ldots \\ & & t_{j+1} = t_j + h, \\ & & u_j' = f(t_j, u_j), \\ & & u_{j+1} = u_j + h\, u_j'. \end{array}} \tag{4}$$

donde dado un valor de $t$ y un vector $u = (v, w)$, $f(t, u) = f(t, v, w)$ está definido como

$$f(t, v, w) = \begin{pmatrix} \left(1 - \frac{w}{50} - g(t)\right) v \\ \left(\frac{3\,v}{100} - 1\right) w \end{pmatrix}, \quad \text{donde} \quad g(t) = \frac{1}{20}(\sin(\pi\, t/12))^2. \tag{5}$$

Obsérvese que con está notación, el problema de valor inicial (2) se puede escribir de forma vectorial como

$$\frac{du}{dt} = f(t, u), \quad u(t_0) = u_0. \tag{6}$$

Así, partiendo de $t_0 = 0$ y $(v_0, w_0) = (51, 33)$, si aplicamos el algoritmo de Euler (4) con $h = 1/40$ para el problema de valor inicial (2) (donde por tanto $f(t, u)$ está definida como (5)), obtenemos la siguiente tabla de valores:

| $j$ | $t_j$ | $u_j$ | $u_j'$ |
|---|---|---|---|
| 0 | 0 | $\begin{pmatrix} 51. \\ 33. \end{pmatrix}$ | $\begin{pmatrix} 17{,}34 \\ 17{,}49 \end{pmatrix}$ |
| 1 | 0,025 | $\begin{pmatrix} 51{,}4335 \\ 33{,}4373 \end{pmatrix}$ | $\begin{pmatrix} 17{,}0376 \\ 18{,}1566 \end{pmatrix}$ |
| 2 | 0,05 | $\begin{pmatrix} 51{,}8594 \\ 33{,}8912 \end{pmatrix}$ | $\begin{pmatrix} 16{,}7077 \\ 18{,}8361 \end{pmatrix}$ |
| 3 | 0,075 | $\begin{pmatrix} 52{,}2771 \\ 34{,}3621 \end{pmatrix}$ | $\begin{pmatrix} 16{,}3492 \\ 19{,}5284 \end{pmatrix}$ |
| 4 | 0,1 | $\begin{pmatrix} 52{,}6859 \\ 34{,}8503 \end{pmatrix}$ | $\begin{pmatrix} 15{,}9615 \\ 20{,}2332 \end{pmatrix}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |

De este modo obtenemos las aproximaciones

$$u_j = (v_j, w_j) \approx u(t_j) = (v(t_j), w(t_j)), \quad j = 1, 2, 3, \ldots$$

de las poblaciones de alces y lobos para los valores del tiempo

$$t_0 = 0, \quad t_1 = h, \quad t_2 = 2h, \quad t_3 = 3h, \ldots$$

Obviamente, el razonamiento seguido para el problema particular (2) tiene sentido para cualquier problema de la forma (6), donde $u$ es un vector (de dimensión 2, o 1, o mayor que 2) cuyas componentes son las variables de estado del sistema y $f(t, u)$ es una función vectorial (no necesáriamente definida como (5)) que devuelve para cada $t$ y $u$ concreto un vector con tántas componentes como $u$. Por tanto, el algoritmo (4) se puede aplicar para cualquier problema de la forma (6).