

### 3 TÉCNICAS NUMÉRICAS

#### Introducción

Como vimos en el capítulo anterior la mayoría de nuestros modelos dinámicos conducen a ecuaciones diferenciales. En buena parte de los casos la resolución analítica es difícil cuando no inexistente y por lo tanto es necesario recurrir a resoluciones numéricas. Sin pretender abarcar los aspectos del cálculo numérico trataremos de hacer una brevísima referencia a algunas técnicas de resolución numérica de ecuaciones, apoyándonos en los paquetes de cálculo como OCTAVE o SCILAB.

OCTAVE es un paquete de cálculo de alto nivel, mayormente compatible con el software comercial Matlab®, con licencia GNU General Public License. Se puede descargar libremente de la siguiente dirección:

<https://www.gnu.org/software/octave/>

SCILAB es un paquete de cálculo de código abierto y licencia libre desarrollado originalmente por el INRIA de Narbonne, Francia pero que ahora se encuentra en la órbita del consorcio Scilab Enterprises, bajo la licencia CeCILL, compatible con la GNU General Public License. Se puede descargar de la siguiente dirección:

[www.scilab.org](http://www.scilab.org)

y tiene asociado un programa gráfico de simulación denominado XCOS.

Estos programas están pensados para el manejo de matrices y vectores aunque se pueden desarrollar todas las ecuaciones de cálculo que interesen. Además de las operaciones elementales, el programa cuenta con funciones predefinidas así como elementos que permiten utilizarlo para la programación. Véase el ejemplo ‘ejem3.1.m’ en Octave; para resolverlo una vez abierta la consola del programa y seleccionado el directorio donde se encuentra el archivo, hacer doble click en el nombre para abrirlo en la consola y apretar el ícono de play; o bien escribir ‘ejem3.1’ en el workspace y apretar enter. En el caso de Scilab, ‘[ejem3.1.sce](#)’ en Scilab; para ejecutarlo, una vez abierta la consola del programa y seleccionado el directorio correspondiente, escribir: `exec(‘ejem3.1.sce’)` y luego presionar la tecla Enter.

#### Ecuaciones algebraicas

Como hemos mencionado, los modelos dinámicos implican en general ecuaciones diferenciales. Para resolver dichas ecuaciones se necesita conocer las condiciones iniciales o de contorno; normalmente se parte del sistema en estado estacionario. Sin embargo, en condiciones estacionarias, al igualar a cero el cambio con la variable tiempo, las ecuaciones diferenciales ordinarias se transforman en algebraicas.

Supongamos un sistema de  $n$  ecuaciones con  $n$  incógnitas

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\
 &\vdots \\
 a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n
 \end{aligned}$$

Expresado en forma matricial

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdot & \cdot & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

O bien en forma más compacta

$$\mathbf{Ax} = \mathbf{b}$$

Resolver este sistema de ecuaciones quiere decir hallar las raíces de

$$\mathbf{f(x)} = \mathbf{Ax} - \mathbf{b}$$

Multiplicando por la matriz inversa de  $\mathbf{A}$

$$\begin{aligned}
 \mathbf{A}^{-1}\mathbf{Ax} &= \mathbf{A}^{-1}\mathbf{b} \\
 \text{como } \mathbf{A}^{-1}\mathbf{A} &= \mathbf{I} \\
 \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b}
 \end{aligned}$$

La resolución de este sistema con el Software se puede hacerse fácilmente (ver 'ejem3.2').

### Funciones de una variable

Aún para resolver funciones de una sola variable muchas veces es necesario recurrir a una técnica numérica. Por ejemplo, si se quiere hallar la solución de  $f(x) = 0$  se realizan  $k$  iteraciones partiendo de determinado valor inicial de  $x$  hasta satisfacer determinado criterio de convergencia

$$|f(x_k)| \leq \varepsilon$$

También suele tomarse la convergencia referida a la variable en lugar de la función:

tolerancia absoluta:  $|x_k - x_{k-1}| \leq \varepsilon_r$

tolerancia relativa:

$$\frac{|x_k - x_{k-1}|}{|x_{k-1}|} \leq \varepsilon_r$$

Un método simple es el de sustitución directa: sea la ecuación  $f(x) = 0$ , la reescribimos de la siguiente forma  $x = g(x)$ . Esto nos permite plantear una interacción de la forma  $x_{k+1} = g(x_k)$ . Se realizan iteraciones hasta encontrar  $x^*$  que satisfaga determinado criterio de convergencia de modo que  $x^* = g(x^*)$ .

Por ejemplo, consideremos un RCAI donde se da una reacción de segundo orden. Del balance de masa

$$\frac{dC_A}{dt} = \frac{v}{V} C_{Ain} - \frac{v}{V} C_A - k C_A^2$$

Si, por ejemplo,  $v/V = 1 \text{ min}^{-1}$ ,  $C_{Ain} = 1 \text{ gmol/L}$ ,  $k = 1 \text{ L}/(\text{gmol}\cdot\text{min})$ , en estado estacionario

$$1 - C_{As} - C_{As}^2 = 0$$

O escribiendo  $x = C_{As}$

$$f(x) = -x^2 - x + 1$$

Escribimos

$$x^2 = -x + 1 \Rightarrow x = \sqrt{-x + 1} = g(x)$$

realizamos las iteraciones

$$x_{k+1} = \sqrt{-x_k + 1} = g(x_k)$$

Por ejemplo, partiendo de  $x_0 = 0.5$

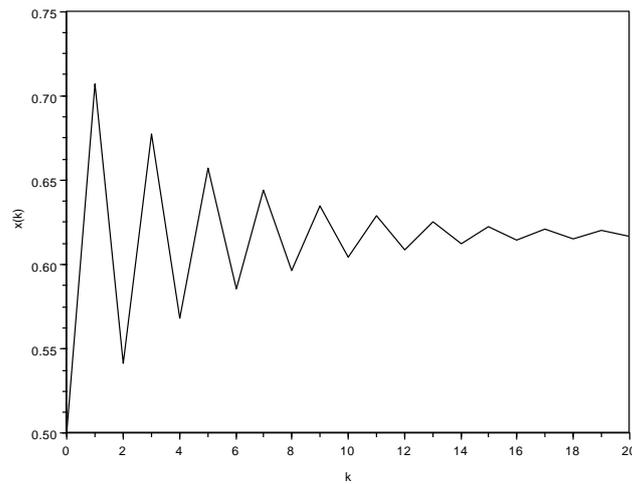
$$x_1 = \sqrt{-0.5 + 1} = 0.7071$$

$$x_2 = \sqrt{-0.7071 + 1} = 0.5412$$

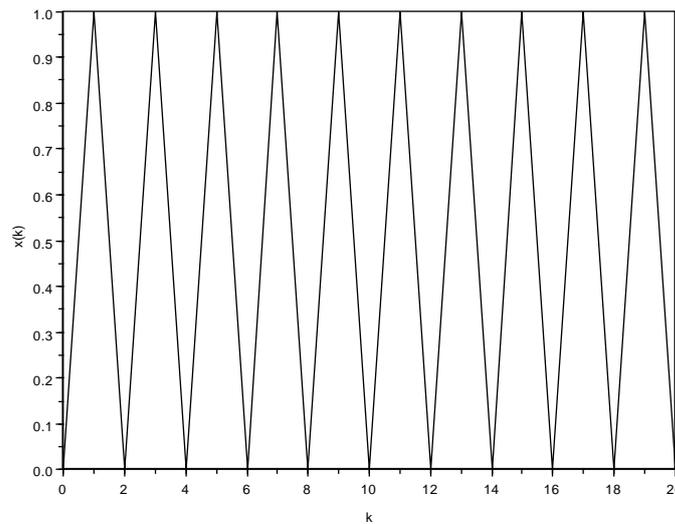
$$x_3 = \sqrt{-0.5412 + 1} = 0.6774$$

$$x_4 = \sqrt{-0.6774 + 1} = 0.5680$$

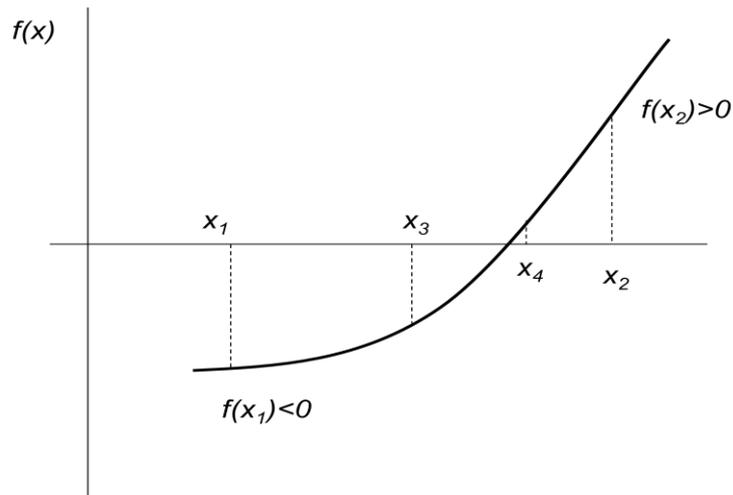
Que converge lentamente a 0.618 (véase ‘[ejem3.3](#)’). Si graficamos la evolución de las distintas soluciones



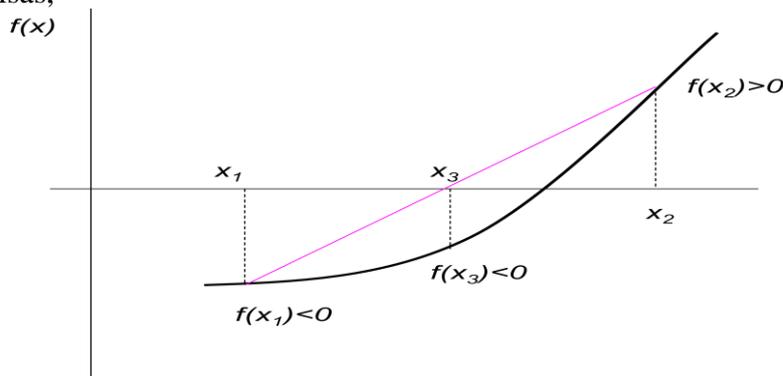
A pesar de su simplicidad se pueden generar problemas si no se elige adecuadamente el punto de partida. Por ejemplo, partiendo de  $x_0 = 0$  las iteraciones no convergen sino que oscilan entre el valor 0 y el valor 1.



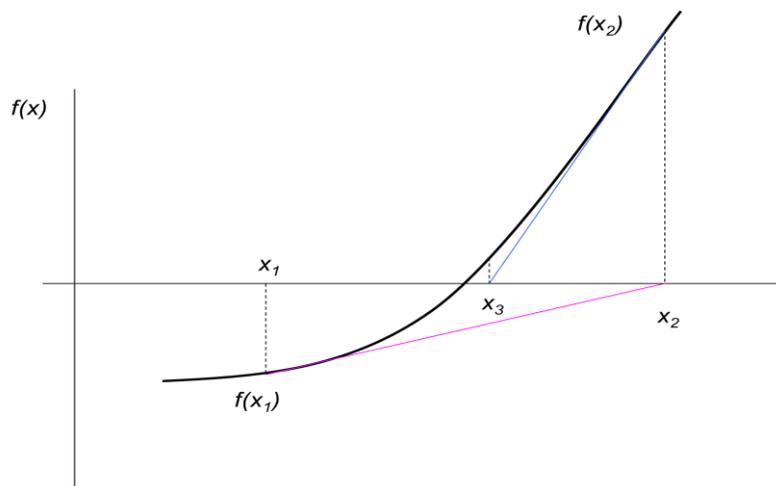
Otro método muy utilizado es el de la bisección: partiendo de dos puntos  $x_1$  y  $x_2$  tales que el valor de la función evaluada en ellos sea de distinto signo, se evalúa la función en el punto intermedio  $x_3$ . Si es del mismo signo que  $x_1$  se sustituye por éste (en idéntica forma si fuera  $x_2$ ) y así sucesivamente hasta satisfacer determinado criterio de convergencia.



El método de *reguli falsi* es similar al anterior, aunque el punto de iteración surge de realizar la intersección de la recta que une los dos puntos de la curva seleccionados con el eje de abscisas,



También se recurre normalmente a los métodos de Newton o variaciones de este. Para ello es necesario conocer la pendiente en el punto en el que se está evaluando la función. De la intersección de dicha pendiente con el eje de abscisas surge el siguiente punto de la iteración. Es un método que tiene una convergencia mucho más rápida que los anteriores, si bien requiere más información (conocer la derivada).



El paquete de cálculo tiene la función *fsolve* que permite halla las raíces de una función (ver ‘[ejem3.4](#)’).

Funciones de varias variables

Para sistemas multivariable la estrategia es similar. Si  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$   
 es el sistema de ecuaciones con varias variables

$$\begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \cdot \\ \cdot \\ f_n(x_1, x_2, \dots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

usando la expansión por Taylor

$$f_i(\mathbf{x} + \Delta\mathbf{x}) = f_i(\mathbf{x}) + \sum_{j=1}^n \frac{\partial f_i}{\partial x_j} \Delta x_j + \dots$$

y despreciando los términos superiores

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{J} \Delta\mathbf{x}$$

donde

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdot & \cdot & \frac{\partial f_1}{\partial x_n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdot & \cdot & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Y las iteraciones se resuelven según

$$\begin{aligned} \mathbf{f}(\mathbf{x}) + \mathbf{J} \Delta\mathbf{x} &= \mathbf{0} \\ \Delta\mathbf{x}_k &= -\mathbf{J}_k^{-1} \mathbf{f}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k - \mathbf{J}_k^{-1} \mathbf{f}(\mathbf{x}_k) \end{aligned}$$

La misma función *fsolve* puede utilizarse para un sistema con varias variables y también admite trabajar con la matriz Jacobiana (ver ‘[ejem3.4](#)’).

Estabilidad de soluciones numéricas

Cuando trabajamos con resoluciones numéricas de ecuaciones es importante evaluar la estabilidad de la solución, esto es, si converge hacia un determinado valor, oscila entre una o más soluciones o diverge, creciendo en forma indefinida. Una forma de analizar la estabilidad es considerar que si se puede escribir la ecuación en la forma  $x = g(x)$  debe cumplirse que  $x^*$  es una solución estable si

$$\left| \frac{\partial g}{\partial x} \right| < 1$$

evaluada en  $x^*$ .

Por ejemplo, en el método de Newton

$$x_{k+1} = g(x_k) = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$g'(x_k) = 1 - \frac{f'(x_k)}{f'(x_k)} + \frac{f(x_k)f''(x_k)}{[f'(x_k)]^2} = \frac{f(x_k)f''(x_k)}{[f'(x_k)]^2}$$

$$g'(x^*) = \frac{f(x^*)f''(x^*)}{[f'(x^*)]^2}$$

Por lo tanto el método converge siempre que  $f'(x^*) \neq 0$

### Integración numérica

Si se tiene una ecuación diferencial ordinaria  $\frac{dx}{dt} = f(x)$

Pueden idearse distintas estrategias para realizar la integración numérica. Por ejemplo, en el método de Euler denominado “explícito” se realiza la siguiente aproximación:

$$\frac{dx}{dt} \approx \frac{x(k+1) - x(k)}{t(k+1) - t(k)}$$

Normalmente se utiliza un intervalo de tiempo constante entre las iteraciones

$$f(x(k)) \approx \frac{x(k+1) - x(k)}{\Delta t}$$

entonces

$$x(k+1) = x(k) + \Delta t f(x(k))$$

o bien en sistemas de varias variables

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \Delta t \mathbf{f}(\mathbf{x}(k))$$

En el denominado método de Euler “implícito” se considera

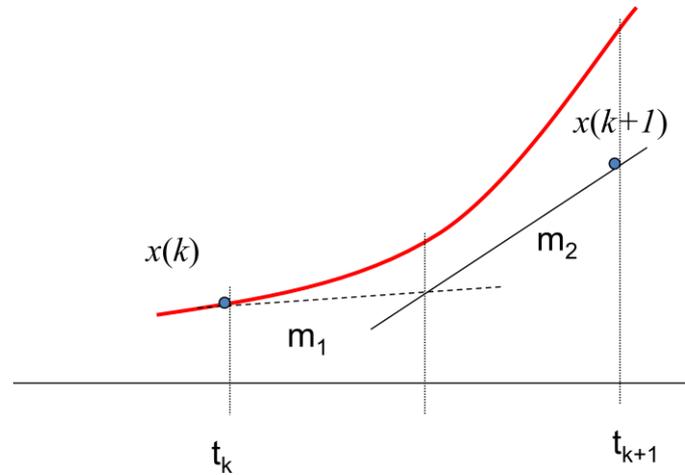
$$f(x(k+1)) \approx \frac{x(k+1) - x(k)}{\Delta t}$$

$$x(k+1) = x(k) + \Delta t f(x(k+1))$$

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \Delta t \mathbf{f}(\mathbf{x}(k+1))$$

Véase el ‘[ejem3.5](#)’.

Otros métodos que proporcionan resultados más ajustados que el de Euler son los de Runge-Kutta. Veamos el caso más sencillo: la idea es usar Euler para evaluar el valor de la función en la mitad del intervalo.



Sea  $m_1$  la pendiente en el punto inicial y  $m_2$  la pendiente en el punto del medio.

$$\begin{aligned}
 m_1 &= f(t(k), x(k)) \\
 m_2 &= f\left(t(k) + \frac{\Delta t}{2}, x(k) + \frac{\Delta t}{2} m_1\right) \\
 x(k+1) &= x(k) + m_2 \Delta t
 \end{aligned}$$

Este concepto se puede extender para utilizar más puntos intermedios. Por ejemplo, para el método de Runge-Kutta de 4º orden

$$\begin{aligned}
 m_1 &= f(t(k), x(k)) \\
 m_2 &= f\left(t(k) + \frac{\Delta t}{2}, x(k) + \frac{\Delta t}{2} m_1\right) \\
 m_3 &= f\left(t(k) + \frac{\Delta t}{2}, x(k) + \frac{\Delta t}{2} m_2\right) \\
 m_4 &= f(t(k) + \Delta t, x(k) + m_3 \Delta t) \\
 x(k+1) &= x(k) + \left[\frac{m_1}{6} + \frac{m_2}{3} + \frac{m_3}{3} + \frac{m_4}{6}\right] \Delta t
 \end{aligned}$$

Octave posee una rutina de integración basada en estos métodos denominada *lsode* (ver [‘ejem3.6’](#) y [‘ejem3.7’](#)).