

Método de Euler en sistemas de ecuaciones diferenciales

Métodos numéricos II

María del Mar Ruiz Martín

Antonio R. Moya Martín-Castaño

Francisco Luque Sánchez

Doble Grado de Ingeniería Informática y Matemáticas

Universidad de Granada - UGR

18001 Granada, Spain

June 2, 2016

1 Introducción

En este trabajo explicaremos el método de Euler para resolución numérica de sistemas de ecuaciones diferenciales de primer orden. Este método es una extensión a sistemas de ecuaciones del método de Euler para ecuaciones diferenciales. Aunque no es muy utilizado en la práctica, ya que por sus características no aporta soluciones lo suficientemente buenas, sí que se suele explicar como introducción a los métodos de resolución numérica de ecuaciones diferenciales, dada su simplicidad. Además, es el más básico de lo que se conocen como *métodos explícitos de resolución numérica de EDs*, y el más básico de los métodos de *Runge-Kutta*. Por tanto, suele ser una buena estrategia introducir este método previo a la explicación de los métodos nombrados, ya que los mismos son una generalización de éste.

El trabajo irá estructurado de la siguiente forma. Primero, daremos una descripción general del concepto de sistema de ecuaciones diferenciales, ya que será el objeto sobre el que trabajaremos. Concretamente, nos centraremos en sistemas de ecuaciones diferenciales de primer orden. Tras esto, veremos cómo podemos transformar una ecuación diferencial de orden superior en un sistema de ecuaciones diferenciales de primer orden, con el fin de tratar una ecuación diferencial de orden superior con el método de Euler. Posteriormente, explicaremos el método de Euler y su aplicación a los sistemas de ecuaciones diferenciales. Después, haremos un estudio teórico del error cometido con el uso de este método para aproximar soluciones, así como demostraremos que el método converge a la solución bajo determinadas condiciones.

Finalmente, cerraremos el trabajo con algunos ejemplos que ilustren de forma práctica todo lo que hemos desarrollado previamente de forma teórica. Nos apoyaremos de un programa escrito en Maxima para calcular las aproximaciones en ciertos puntos con distintos

sistemas de ecuaciones y ecuaciones diferenciales de orden superior, que habremos transformado previamente a sistemas de ecuaciones.

2 Descripción general de un sistema de ecuaciones diferenciales

Para comenzar, veamos lo que es una ecuación diferencial. Una ecuación diferencial es una igualdad en la que interviene una variable independiente (t), una variable dependiente ($x(t)$) y las sucesivas derivadas de la variable dependiente respecto de la independiente. En forma general, podemos escribir dicha ecuación de la siguiente forma:

$$F(t, x, x', \dots, x^n) = 0$$

Se define el orden de dicha ecuación diferencial como el mayor orden de derivación de la variable dependiente que aparece en dicha ecuación.

Una solución de esta ecuación es un conjunto de funciones que satisfacen las condiciones de dicha ecuación.

Asociado a esta ecuación diferencial, se define un problema de valores iniciales (PVI) como dicha ecuación diferencial junto con los n valores siguientes. Fijado t_0 :

$$x(t_0) = y_0, x'(t_0) = y_1, \dots, x^{n-1}(t_0) = y_{n-1}$$

Se dice que un PVI está bien planteado si existe solución, es única y depende de forma continua de los datos del problema.

Pasamos ahora a definir el concepto de sistema de ecuaciones diferenciales. Un sistema de ecuaciones diferenciales es un conjunto de ecuaciones diferenciales en las que se ven implicadas una variable independiente, un conjunto de variables dependientes de la misma, y las sucesivas derivadas de dichas variables dependientes respecto de la independiente. En forma general, podemos escribir un sistema de ecuaciones diferenciales como sigue:

$$\begin{cases} F_1(t, x_1, x'_1, \dots, x_1^{n_1}, \dots, x_m, x'_m, \dots, x_m^{n_m}) = 0 \\ F_2(t, x_1, x'_1, \dots, x_1^{n_1}, \dots, x_m, x'_m, \dots, x_m^{n_m}) = 0 \\ \vdots \\ F_r(t, x_1, x'_1, \dots, x_1^{n_1}, \dots, x_m, x'_m, \dots, x_m^{n_m}) = 0 \end{cases}$$

Al igual que para las ecuaciones diferenciales, la solución de un sistema de ecuaciones diferenciales está compuesto por un conjunto de ecuaciones que satisfacen las condiciones impuestas por las ecuaciones del sistema.

De nuevo, podemos definir un PVI asociado al sistema de ecuaciones anterior. Dicho PVI viene definido por el sistema de ecuaciones diferenciales más el siguiente conjunto de valores:

$$x_1(t_0) = y_{10}, \dots, x_1^{n_1-1}(t_0) = y_{1n_1-1}, \dots, x_m(t_0) = y_{m0}, \dots, x_1^{n_m-1}(t_0) = y_{mn_m-1}$$

Se dice que un sistema de ecuaciones diferenciales está bien planteado si cumple las propiedades definidas previamente:

- \exists solución
- Es única
- Depende de forma continua de las propiedades del problema

Durante el trabajo, dado que el método de Euler se aplica a ecuaciones y sistemas de primer orden, trabajaremos con este tipo de sistemas. Podemos escribir este tipo de sistemas de la siguiente manera:

$$\begin{cases} F_1(t, x_1, \dots, x_m) = \frac{dx_1}{dt} \\ F_2(t, x_1, \dots, x_m) = \frac{dx_2}{dt} \\ \vdots \\ F_m(t, x_1, \dots, x_m) = \frac{dx_m}{dt} \end{cases}$$

Y asociado a este sistema de ecuaciones, el PVI definido por el sistema anterior y el conjunto de valores:

$$x_1(t_0) = y_1, x_2(t_0) = y_2, \dots, x_m(t_0) = y_m$$

Notemos también que podemos escribir el sistema anterior haciendo uso de una notación vectorial. De esta forma, tenemos que podemos escribir dicho sistema como sigue:

$$X'(t) = F(t, X(t)), X(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

Diremos que el PVI está bien planteado si satisface las condiciones anteriores. Tenemos una caracterización de problema bien planteado a partir de la condición de Lipschitz. Dado el sistema anterior, diremos que la función vectorial $F(t, X(t))$ satisface la condición de Lipschitz, con constante de Lipschitz L , si se cumple que $\forall t_1, t_2 \in [a, b]$ se cumple:

$$\|F(t_1, X(t_1)) - F(t_2, X(t_2))\| \leq L\|X(t_1) - X(t_2)\|$$

Tenemos entonces que el problema anterior es un problema bien planteado si $F(t, X(t))$ es continua en su conjunto de definición y Lipschitziana.

3 Ecuaciones diferenciales de orden superior y reescritura como sistemas

En la sección anterior definimos el concepto de ecuación diferencial de orden superior y el de sistema de ecuaciones diferenciales. Veamos ahora cómo reescribir una ecuación diferencial de orden n en un sistema de ecuaciones diferenciales en el que se tienen n ecuaciones diferenciales de primer orden. Este paso es necesario para poder aplicar

posteriormente el método de Euler, ya que este método sólo puede ser aplicado a ecuaciones diferenciales de primer orden. Veamos entonces esta forma de reescribir las ecuaciones diferenciales de orden superior. Sea la ecuación diferencial de orden n : $F(t, x, x', \dots, x^n) = 0$. Podemos escribir el siguiente sistema de ecuaciones diferenciales, cuya solución es equivalente a la de la ecuación anterior. Tomamos los siguientes cambios de variable:

$$y_0 = x, y_1 = x', y_2 = x'', \dots, y_{n-1} = x^{(n-1)}$$

Entonces, la solución de y_0 para el siguiente sistema de ecuaciones siguiente es la solución de la ecuación diferencial inicial:

$$\begin{cases} y'_0 = x' = y_1 \\ y'_1 = x'' = y_2 \\ \vdots \\ y'_{n-2} = x^{(n-1)} = y_{n-1} \\ y'_{n-1} = x^{(n)} = F(t, y_1, y_2, \dots, y_{n-1}) \end{cases}$$

Una vez convertida la ecuación en el sistema anterior, estamos en condiciones de aplicar el método de Euler para resolverlo.

4 Método de Euler para sistemas de ecuaciones diferenciales

Recordemos primero el método de Euler para un problema de valores iniciales (PVI) bien planteado. Si tomamos puntos en el intervalo $[a, b]$ y un N natural, de forma que $h = (b - a)/N$. Para cada $i=0, \dots, N$, los puntos serían

$$t_i = a + ih;$$

y dado el PVI:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$$

Podemos tratar de aproximar la solución mediante el método de Euler. Mediante el desarrollo de Taylor centrado en t_i evaluando en el punto t_{i+1} :

$$y(t_{i+1}) = y(t_i) + y'(t_i)(t_{i+1} - t_i) + y''(\xi_i)(t_{i+1} - t_i)^2/2$$

para algún ξ_i entre (t_i, t_{i+1}) .

Si observamos, lo escrito anteriormente es equivalente a:

$$y(t_{i+1}) = y(t_i) + f(t_i, y(t_i))h + y''(\xi_i)h^2/2,$$

y lo que es lo mismo:

$$y(t_{i+1}) = y(t_i) + f(t_i, y(t_i))h + O(h^2).$$

De este modo el método de Euler construye un $w_i \approx y(t_i)$ para cada $i = 1, \dots, N$

$$\begin{cases} w_0 = y_0 \\ w_{i+1} = w_i + hf(t_i, w_i) \end{cases}$$

Para el caso de sistemas de ecuaciones diferenciales de primer orden:

$$\begin{cases} x'_1 = F_1(t, x_1(t), \dots, x_n(t)) \\ \vdots \\ x'_n = F_n(t, x_1(t), \dots, x_n(t)) \end{cases}$$

tenemos que para cada $j = 1, \dots, n$, podemos hacer una aproximación de $w_{i,j} \approx x_j(t_i)$ de forma que:

$$\begin{cases} w_{0,j} = x_{j,0} \\ w_{i+1,j} = w_{i,j} + hF_j(t_i, w_{i,j}) \end{cases}$$

De este modo, escribiendo el sistema con la siguiente notación matricial:

$$X'(t) = F(t, X(t))$$

donde $X(t)$ es de la forma:

$$X(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

y $F(t, X(t))$:

$$F(t, X(t)) = \begin{bmatrix} F_1(t, x_1(t), \dots, x_n(t)) \\ \vdots \\ F_n(t, x_n(t), \dots, x_n(t)) \end{bmatrix}$$

podemos expresar una aproximación $W_i \approx X(t_i)$ de la siguiente manera:

$$\begin{cases} W_0 = X(t_0) \\ W_{i+1} = W_i + hF(t_i, W_i) \end{cases}$$

donde W_i es de la forma:

$$W_i = \begin{bmatrix} w_1(t_i) \\ \vdots \\ w_n(t_i) \end{bmatrix}$$

5 Estudio del error y análisis de la convergencia del método

Para ver la convergencia del método, vamos a definir dos conceptos previos, que son consistencia y estabilidad. El cumplimiento de estas dos propiedades nos asegurará la convergencia del método. Comenzamos definiendo la consistencia de un método. Se dice

que un método es consistente si el error local que se produce con el método tiende a 0 cuando el tamaño del paso tiende a 0. Es decir, sea $\tau_i(h)$ el error local cometido en el paso i , entonces:

$$\lim_{h \rightarrow 0} \max |\tau_i(h)| \rightarrow 0$$

Veremos más adelante que el método de Euler para sistemas es consistente, cuando hagamos un estudio del error local. Definimos ahora la propiedad de estabilidad. Esta propiedad nos dice cómo se propagan los errores en los datos iniciales al utilizar el algoritmo. Se dice que un método es estable si una perturbación en los datos de entrada no se ve maximizado cuando se utiliza el método para obtener la solución al problema. De forma más rigurosa, se dice que el método es estable si dado un PVI, con $F(t, X(t))$ Lipschitziana, con X_0 como valor inicial exacto y X'_0 como valor de entrada (con una pequeña perturbación), existe una constante K tal que $|\omega' - \omega| \leq K|X'_0 - X_0|$.

Para finalizar, tenemos que por el teorema de Lax-Richtmyer, que no demostraremos por su dificultad, la consistencia y la estabilidad de un método implican su convergencia. Procedemos al estudio del error, que nos dará la consistencia y estabilidad para el método de Euler, por lo que podremos asegurar su convergencia.

Antes de comenzar con el estudio del error, comenzaremos definiendo los tipos de errores que se pueden encontrar cuando se utilizan métodos numéricos para resolución de ecuaciones diferenciales:

Error local Se define como el error que se produce en un sólo paso del algoritmo, si se supone que el valor obtenido en la aproximación anterior es exacto.

Error global Se define como el error total cometido con la aproximación tras las n iteraciones del método. Como veremos más adelante, no es la suma de los errores locales cometidos en todas las iteraciones, sino que se ve incrementado.

Error de truncatura Es el error que se produce por la precisión finita de las máquinas que se utilizan para computar los cálculos.

Estudiemos ahora los errores definidos en el método de Euler aplicado a sistemas de ecuaciones diferenciales.

5.1 Error local

Como ya hemos dicho antes, se define el error local como el que se produce en un paso del algoritmo, suponiendo que el valor del que partimos (t_i, X_i) es un valor real del sistema de ecuaciones. Entonces, tenemos que el error local cometido en la iteración $i + 1$ lo podemos expresar como $E_{i+1} = ||X(t_{i+1}) - [X(t_i) + h * F(t_i, X_i)]||$. Veamos cómo se comporta el error local, a partir del siguiente teorema:

Teorema 1. *Error local del método de Euler.*

El error local para el método de Euler es $O(h^2)$, donde h es el tamaño del paso que hemos elegido para aplicar el método ($h = (b - a)/N$).

Demostración

Veamos la demostración de dicho resultado. Tomamos el PVI asociado al sistema de ecuaciones diferenciales:

$$X' = F(t, X), X(t_0) = X_0$$

Supongamos que tenemos el valor exacto para un determinado t_i , es decir, tenemos (t_i, X_i) exacto. Entonces, el error local para la siguiente iteración es $\|E_{i+1}\| = \|X(t_{i+1}) - [X(t_i) + h * F(t_i, X_i)]\|$. Tomando cada una de las componentes por separado, y expresando $x_j(t_{i+1}), j = 1, \dots, n$ usando su polinomio de Taylor centrado en t_i tenemos:

$$\begin{aligned} x_j(t_{i+1}) &= x_j(t_i) + h * x'_j(t_i) + \frac{h^2}{2} * x''_j(t_i) + \frac{h^3}{3!} * x'''_j(t_i) + \dots = \\ &= x_j(t_i) + h * x'_j(t_i) + \frac{h^2}{2} * x''_j(\xi_j) \end{aligned}$$

Donde $h = t_{i+1} - t_i$. Con el método de Euler, la aproximación que obtenemos para $x_j(t_i) = \omega_{i+1,j}$ es:

$$\omega_{i+1,j} = x_j(t_i) + h * f_j(t_i, x_1, \dots, x_n) = x_j(t_i) + h * x'_j(t_i)$$

Restando ambas expresiones, tenemos que para cada x_j , el error cometido en un paso del método es:

$$e_{i+1,j} = |x_j(t_{i+1}) - \omega_{i+1,j}| = \frac{h^2}{2} * x''_j(\xi_{i+1,j}), \xi_{i+1,j} \in [t_i, t_{i+1}]$$

Tenemos entonces que el vector de los errores lo podemos expresar como sigue:

$$E_{i+1} = \begin{bmatrix} e_{i+1,1} \\ \vdots \\ e_{i+1,n} \end{bmatrix} = \begin{bmatrix} \frac{h^2}{2} * x''_1(\xi_{i+1,1}) \\ \vdots \\ \frac{h^2}{2} * x''_n(\xi_{i+1,n}) \end{bmatrix}$$

Tomando ahora la norma euclídea del vector de los errores, tenemos que el error es $O(h^2)$

□

Merece la pena hacer ahora la apreciación de la consistencia del método de Euler. Como acabamos de demostrar, el error local depende de h^2 , por lo que cuando h tienda a 0, tendremos que el máximo de los errores locales tenderá también a 0, por tanto el método es consistente.

Una vez visto el error local que se comete con el método de Euler, pasamos a estudiar el error global que se comete.

5.2 Error global

Como ya dijimos en la introducción de este apartado, se define el error global como el error total cometido tras las n iteraciones que se realizan del método. Aunque cabría esperar que el error cometido con el método fuese la suma de los errores que se cometen en cada una de las iteraciones, esto no es cierto. Esto se debe a que, en cada paso, el error previo cometido se ve levemente incrementado. Pasamos a estudiar entonces dicho error. Necesitaremos de dos lemas previos, que usaremos luego en la demostración:

Lema 2. Para toda $x \geq -1$ y para cualquier m positiva, tenemos que $0 \leq (1+x)^m \leq e^{mx}$

Demostración

Aplicamos el teorema de Taylor a e^x . Obtenemos:

$$e^x = 1 + x + \frac{1}{2}x^2e^\xi$$

donde $0 < \xi < x$. Por tanto:

$$0 \leq 1 + x \leq 1 + x + \frac{1}{2}x^2e^\xi = e^x$$

y como $1 + x \geq 0$:

$$0 \leq (1+x)^m \leq (e^x)^m = e^{mx}$$

□

Lema 3. Si s y t son números reales positivos, $\{a_i\}_{i=0}^k$ es una sucesión que satisface $a_0 \geq -t/s$ y

$$a_{i+1} \leq (1+s)a_i + t, \forall i = 0, 1, \dots, k$$

entonces se tiene que

$$a_{i+1} \leq e^{(i+1)s} \left(a_0 + \frac{t}{s} \right) - \frac{t}{s}$$

Demostración

Para un entero fijo i , la primera desigualdad del lema implica que:

$$a_{i+1} \leq (1+s) * a_i + t \leq (1+s)[(1+s)a_{i-1} + t] + t \leq \dots$$

$$\dots \leq (1+s)^{i+1}a_0 + [1 + (1+s) + \dots + (1+s)^i]t$$

Por otro lado, tenemos que $1 + (1+s) + (1+s)^2 + \dots + (1+s)^i$ es una serie geométrica de razón $(1+s)$, y que por tanto su suma vale:

$$\frac{1 - (1+s)^{i+1}}{1 - (1+s)} = \frac{1}{s}[(1+s)^{i+1} - 1]$$

Por lo tanto

$$a_{i+1} \leq (1+s)^{i+1}a_0 + \frac{(1+s)^{i+1} - 1}{s}t = (1+s)^{i+1} * \left(a_0 + \frac{t}{s} \right) - \frac{t}{s}$$

y finalmente, usando el lema anterior con $x = 1 + s$, tenemos que

$$a_{i+1} \leq e^{(i+1)s} \left(a_0 + \frac{t}{s} \right) - \frac{t}{s}$$

□

Pasamos ya a ver y demostrar el teorema que nos da el orden de precisión del método de Euler.

Teorema 4. *Error global del método de Euler*

El error global del método de Euler para sistemas de ecuaciones es $O(h)$, donde h es el tamaño de paso que hemos elegido al aplicar el método ($h = \frac{b-a}{n}$)

Demostración

Para probar este resultado, supondremos que nos encontramos ante un problema bien planteado, y que la función vectorial $F(t, x_1, \dots, x_n)$ es Lipschitziana en la variable x_j , con constante de Lipschitz L_j , y que para cada x_j existe una constante M_j tal que $|x_j''(t)| \leq M_j, \forall t \in [a, b]$.

Tomamos entonces una de las funciones f_j (el razonamiento es análogo para las demás). Como ya vimos para el error local, podemos escribir $x_j(t_{i+1})$ como:

$$x_j(t_{i+1}) = x_j(t_i) + h * f_j(t_i, x_i, \dots, x_n) + \frac{h^2}{2} * x_j''(\xi_j)$$

Y podemos escribir también la aproximación para la variable x_j en el paso $i + 1$, $\omega_{i+1,j}$ como:

$$\omega_{i+1,j} = \omega_{i,j} + h * f_j(t_i, \omega_{i,1}, \dots, \omega_{i,n})$$

La diferencia entre ambas cantidades sería el error total acumulado en las primeras $i + 1$ iteraciones del método. Podemos expresar esa diferencia como sigue:

$$x_j(t_{i+1}) - \omega_{i+1,j} = x_j(t_i) - \omega_{i,j} + h * [f_j(t_i, x_i, \dots, x_n) - f_j(t_i, \omega_{i,1}, \dots, \omega_{i,n})] + \frac{h^2}{2} * x_j''(\xi_j)$$

Se cumple entonces que:

$$|x_j(t_{i+1}) - \omega_{i+1,j}| \leq |x_j(t_i) - \omega_{i,j}| + h * |f_j(t_i, x_i, \dots, x_n) - f_j(t_i, \omega_{i,1}, \dots, \omega_{i,n})| + \frac{h^2}{2} * |x_j''(\xi_j)|$$

Además, por ser Lipschitziana en la variable x_j , con constante de Lipschitz L_j , y por ser $|x_j''(t)| \leq M_j$, tenemos:

$$|x_j(t_{i+1}) - \omega_{i+1,j}| \leq (1 + hL_j) * |x_j(t_i) - \omega_{i,j}| + \frac{h^2 * M_j}{2}$$

Usando ahora el lema anterior, tomando $a_{i,j} = |x_j(t_i) - \omega_{i,j}|$ para cada $j = 0, 1, \dots, N$, y $s = hL_j, t = \frac{h^2 M_j}{2}$ tenemos:

$$|x_j(t_{i+1}) - \omega_{i+1,j}| \leq e^{(i+1)hL_j} \left(|x_j(t_0) - \omega_{0,j}| + \frac{h^2 M_j}{2hL_j} \right) - \frac{h^2 M_j}{2hL_j}$$

Ahora, puesto que $|x_j(t_0) - \omega_{0,j}| = 0$ y que $(i + 1)h = t_{i+1} - t_0 = t_{i+1} - a$ se concluye que:

$$|x_j(t_{i+1}) - \omega_{i+1,j}| \leq \frac{hM_j}{2L_j} (e^{(t_{i+1}-a)L_j} - 1)$$

Y dado que tomamos j arbitrario, tenemos este resultado para todas las funciones f_j . Finalmente, tomando la norma euclídea de todos los errores globales, podemos concluir que el método es de orden $O(h)$.

□

Aprovechando que la demostración del error global es similar a la de la estabilidad del método, demostraremos ahora que el método es estable. Para ello, tomamos el PVI:

$$X'(t) = F(t, X(t)), X(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

Para el que tenemos las soluciones $X(t_0) = X_0$, $X'_0 = X_0 + \varepsilon$, donde $\varepsilon = [\varepsilon_1, \dots, \varepsilon_n]$. Esto es, la condición inicial exacta y la condición inicial con una pequeña perturbación para cada variable. Definimos $Z_i = X_i - X'_i$, es decir, la diferencia que hay en la iteración i entre la aproximación calculada con el valor inicial correcto y el valor perturbado. Tenemos que:

$$Z_{i+1} = Z_i + h * (F(t_i, X_i) - F(t_i, X'_i))$$

Tomando la norma euclídea, se tiene que

$$|Z_{i+1}| \leq |Z_i| + h * |F(t_i, X_i) - F(t_i, X'_i)|$$

Por ser $F(t, X(t))$ lipschitziana con constante de Lipschitz L , se tiene que

$$|Z_{i+1}| \leq |Z_i|(1 + hL)$$

Y aplicando la desigualdad de forma recursiva tenemos que

$$|Z_{i+1}| \leq |Z_0|(1 + hL)^{i+1} = |\varepsilon|(1 + hL)^{i+1}$$

Tomando ahora la desigualdad del primer lema que citamos en este apartado, tenemos que de la desigualdad anterior se deriva que

$$|Z_{i+1}| \leq |\varepsilon|e^{hL(i+1)}$$

Y finalmente, tomando $h \rightarrow 0$ podemos concluir que

$$\lim_{h \rightarrow 0} |Z_{i+1}| = \varepsilon$$

Lo que nos determina que el método es estable. Como ya demostramos la consistencia del mismo en el apartado anterior, podemos concluir que el método es convergente. Esto era de suponer, ya que el funcionamiento del método es análogo al que se utiliza para ecuaciones diferenciales ordinarias, para el que teníamos ya demostrada la convergencia. Lo único que hay que hacer es tener cuidado porque ahora se está trabajando con funciones vectoriales, pero tomando la norma de los vectores que aparecen, la demostración es esencialmente la misma.

Vamos a comentar brevemente los errores de truncamiento o truncatura que se producen en los métodos de resolución numérica, y que por tanto afectan también al método de Euler.

5.3 Errores de truncatura

Los errores de truncatura, como ya comentamos al principio, se producen por la precisión finita de las máquinas que utilizamos para efectuar los cálculos. Teóricamente, trabajamos con números reales, en muchos casos irracionales, cuyo valor no puede ser representado completamente en una máquina, ya que las mismas tienen una cantidad de recursos finitos. En este caso, lo que hacen los ordenadores es truncar (o redondear) las cifras menos significativas y trabajar con el resto. Si en nuestro método utilizamos un h demasiado pequeño en relación a los valores que toman nuestras funciones, podría suceder que despreciásemos mucha información en cada iteración, provocando que el error total que se cometiera en la aproximación fuera mayor que para valores de h algo más grandes.

Por tanto, uno de los problemas con los que nos encontramos a la hora de aplicar estos métodos es la elección del tamaño del paso. Si tomamos un tamaño de paso demasiado grande, las aproximaciones que obtendremos no serán fiables, porque el error global del método será grande, aunque los errores de truncatura no nos afecten demasiado. En cambio, si h es demasiado pequeño, el error global teórico del método tendería a 0, pero acumularíamos grandes errores de truncatura, lo que nos daría que la aproximación tampoco es buena. Por tanto, a la hora de aplicar estos métodos es importante seleccionar un h de modo que el error total (la suma del error global y el de truncamiento) sea mínimo. En el apartado de ejercicios resueltos se pondrá un ejemplo que nos permita visualizar el problema que hemos expuesto.

Pasamos ahora a ver ejemplos resueltos con el método de Euler, para ver su aplicación real.

6 Ejemplos y ejercicios resueltos

6.1 Método de Euler para sistemas de ecuaciones (Ejercicio teórico-práctico 1)

Veamos un ejemplo de aplicación del método de Euler en un sistema de ecuaciones de primer orden:

Dado el siguiente sistema:

$$\begin{cases} x' = x^2/y - x \\ y' = x^2 - y \\ x(0) = 1.5 \\ y(0) = 0.7 \end{cases}$$

Queremos calcular la solución en $t = 1$ con un tamaño de paso de 0.2

Encontrar la solución exacta no es tarea fácil en este sistema. Aquí podemos observar la utilidad del método de Euler para obtener una aproximación. Realizaremos el estudio para $h = 0.05$, $N = (1 - 0)/0.05 = 20$.

Observar también que:

$$f(t, x) = \begin{bmatrix} x^2/y - x \\ x^2 - y \end{bmatrix}; x(0) = \begin{bmatrix} 1.5 \\ 0.7 \end{bmatrix}$$

Dicho esto, calculemos, a modo de ejemplo (los demás cálculos se realizarán posteriormente con maxima), x_1 y x_2 :

$$x_1 = x_0 + f(t_0, x_0)h = \begin{bmatrix} 1.5 \\ 0.7 \end{bmatrix} + \begin{bmatrix} 1.5^2/0.7 - 1.5 \\ 1.5^2 - 0.7 \end{bmatrix} 0.05 = \begin{bmatrix} 1.585714 \\ 0.7775 \end{bmatrix}$$

Para el caso de x_2 (recordemos que $t_1 = t_0 + h$):

$$x_2 = x_1 + f(t_1, x_1)h = \begin{bmatrix} 1.585714 \\ 0.7775 \end{bmatrix} + \begin{bmatrix} 1.585714^2/0.7775 - 1.585714 \\ 1.585714^2 - 0.7775 \end{bmatrix} 0.05 = \begin{bmatrix} 1.66813 \\ 0.86435 \end{bmatrix}$$

Si observamos, estos serían los resultados de los 20 pasos:

t_i	$w_1(t_i)$	$w_2(t_i)$
0.05	1.5857142857142856	0.7774999999999997
0.1	1.6681320952818426	0.8643494897959183
0.15	1.7456942144054921	0.96026524967159188
0.2	1.8170869259521916	1.0646244016984527
0.25	1.8813015829199045	1.1764834264368493
0.3	1.9376549272877963	1.2946240374098539
0.35	1.9857759236607302	1.4176181664014949
0.4	2.0255692199779745	1.5439025590309514
0.45	2.0571660703425545	1.6718539643255128
0.5	2.0808718188081592	1.7998578781576686
0.55	2.0971162677973867	1.9263663605652841
0.6	2.1064105125348274	2.0499428745700419
0.65	2.1093116671890022	2.1692939932074013
0.7	2.1063954783162187	2.2832890790140139
0.75	2.0982360235174902	2.3909697206168636
0.8	2.0853913525194629	2.4915509551053447
0.85	2.0683938730731812	2.5844162620082254
0.9	2.0477443832187014	2.6691081096161478
0.95	2.0239088166471921	2.7453155570855272
1.0	1.9973169471152268	2.8128601241363627

Table 1: Valores obtenidos con el método de Euler

Con esto, vemos como los valores se van estabilizando (al final, cuanto menor sea el valor de h que tomemos más cercanos quedarán los valores a la solución).

6.2 Método de Euler ecuaciones de orden superior (Ejercicio teórico-práctico 2)

Vamos mostrar el uso del método de Euler para ecuaciones de orden superior. Para ello consideraremos la siguiente ecuación:

$$x''(t) = \frac{1}{x(t)}$$

$$x(1) = 1, x'(1) = 1$$

Para la que aproximaremos su valor en $t = 1.5$ para una longitud de paso $h = 0.05$. En primer lugar pasaremos la ecuación a un sistema de ecuaciones de orden 1:

$$y_0 = x, y_1 = x'$$

$$\begin{cases} y'_0 = x' = y_1 \\ y'_1 = x'' = \frac{1}{y_0} \end{cases}$$

Puesto que ya tenemos el sistema de ecuaciones, pasamos a verlo en forma matricial:

$$f(t, x) = \begin{bmatrix} y_1 \\ \frac{1}{y_0} \end{bmatrix}; y(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Pasamos entonces a aplicar el método.

$$y_1 = y_0 + f(t_0, x_0)h = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} 0.05 = \begin{bmatrix} 1.05 \\ 1.05 \end{bmatrix}$$

$$y_2 = y_1 + f(t_1, x_1)h = \begin{bmatrix} 1.05 \\ 1.05 \end{bmatrix} + \begin{bmatrix} 1.05 \\ \frac{1}{1.05} \end{bmatrix} 0.05 = \begin{bmatrix} 1.1025 \\ 1.0976190476190477 \end{bmatrix}$$

Repitiendo el proceso sucesivamente obtenemos los siguientes resultados:

t_i	$w_1(t_i)$	$w_2(t_i)$
1.05	1.05	1.05
1.1	1.1025	1.0976190476190477
1.15	1.1573809523809524	1.1429705215419501
1.2	1.2145294784580498	1.1861715089930918
1.25	1.2738380539077043	1.22733971518886
1.3	1.3352050396671473	1.2665911725045031
1.35	1.3985345982923725	1.304038604557684
1.4	1.4637365285202566	1.339790312138641
1.45	1.5307260441271886	1.3739494651185469
1.5	1.599423517383116	1.4066137032373343

Table 2: Valores obtenidos con el método de Euler

6.3 Influencia de los errores de truncatura (Ejercicio teórico-práctico 3)

Vamos a mostrar ahora la influencia que tienen los errores de truncatura cuando se usan métodos numéricos para resolver ecuaciones diferenciales. Para ejemplificar este

hecho, vamos a partir de un problema de valores iniciales, y vamos a calcular la aproximación obtenida con el método de Euler para distintos tamaños de paso.

Dado el PVI:

$$\begin{cases} x' = 2 * x + 3 * y \\ y' = 2 * x + y \\ x(0) = 4 \\ y(0) = 1 \end{cases}$$

Vamos a tratar de aproximar la solución en $t = 1$. La solución exacta para este sistema viene dada por:

$$\begin{aligned} x(t) &= e^{-t} + 3e^{4t}; \\ y(t) &= -e^{-t} + 2e^{4t}; \end{aligned}$$

Y evaluando: $x(1) = 164.1623295406042$ y $y(1) = 108.828420625117$.

Para reducir el tiempo de cómputo, y que estos efectos se vean amplificadas, vamos a establecer la precisión de la máquina en 6 cifras significativas. Una vez hecho esto, calcularemos la solución para el problema anterior con tamaños de paso $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$. Las aproximaciones que obtenemos las exponemos en la tabla siguiente, junto con los errores que se producen:

h	$w_1(1)$	$e_1(1)$	$w_2(1)$	$e_2(1)$
10^{-1}	121.800842	42.361487	80.677551	28.150869
10^{-2}	151.881347	12.280982	100.644134	8.18428662
10^{-3}	162.865905	1.296424	107.964385	0.86403562
10^{-4}	164.146606	0.015723	108.816040	0.01238062
10^{-5}	164.864013	0.701683	109.291381	0.46296038
10^{-6}	171.042297	6.879967	113.523529	4.69510838
10^{-7}	303.327514	139.165184	197.364379	88.53595838

Table 3: Errores obtenidos con el método de Euler para distintos tamaños de paso

Como podemos observar, para tamaños de h menores a 10^{-5} , el error se dispara. Mostramos en la siguiente gráfica el comportamiento de ambos errores, para dar una idea más visual de los mismos:

Concluimos ahora la importancia que tiene la elección correcta del tamaño del paso para obtener unas aproximaciones fiables.

6.4 Aplicación del método de Euler para la ley de Hooke (Ejercicio teórico-práctico 4)

En este ejercicio vamos a aplicar el método de Euler para aproximar la ley de Hooke:

$$F = m \frac{\partial^2 x}{\partial t^2} = -kx$$

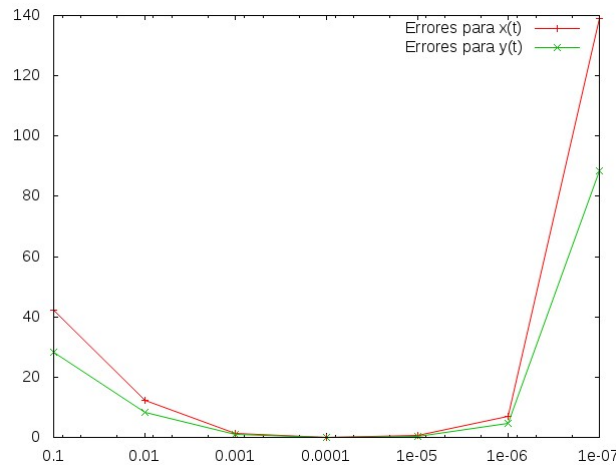


Figure 1: Comportamiento de los errores para distintos tamaños de paso

Donde m es la masa del cuerpo y $x(t)$ la posición del cuerpo en el instante t y k la constante elástica del muelle, que e . Puesto que nos encontramos ante una ecuación de orden superior, en primer lugar obtendremos el sistema sobre el que podremos aplicar el método.

$$\begin{cases} y'_0 = x' = y_1 \\ y'_1 = x'' = \frac{k}{m}y_0 \end{cases}$$

Ahora tan solo tendremos que aplicar el método de Euler al sistema. Lo aplicaremos para determinar la posición en el instante 0.5 de un cuerpo de masa $m = 1$ kg sobre un muelle con constante elástica $k = 3$, para una longitud de paso $h = 0.05$ y con las siguiente condiciones iniciales:

$$x(0) = 0, x'(0) = 0.5$$

Los resultados obtenidos son los siguientes:

t_i	$w(t_i)$
0.05	0.025
0.1	0.05
0.15	0.0751875
0.2	0.10075
0.25	0.12687640625000002
0.3	0.1537584375
0.35	0.18159204179687502
0.4	0.21057883437500002
0.45	0.24092756726660158
0.5	0.27285564141601565

Table 4: Aproximación de la posición según la ley de Hooke

6.5 Método de Euler para sistemas de ecuaciones de primer orden (Ejercicio práctico 1)

Ahora vamos a mostrar el comportamiento real del método, y cómo se comportan las aproximaciones cuando el tamaño del paso disminuye. A diferencia del ejercicio anterior, para obtener resultados reales, trabajaremos con toda la precisión que la máquina es capaz de darnos. Dado el PVI del ejercicio anterior:

$$\begin{cases} x' = 2 * x + 3 * y \\ y' = 2 * x + y \\ x(0) = 4 \\ y(0) = 1 \end{cases}$$

Vamos a tratar de aproximar de nuevo la solución en $t = 1$.

La solución exacta para este sistema viene dada por:

$$\begin{aligned} x(t) &= e^{-t} + 3e^{4t}; \\ y(t) &= -e^{-t} + 2e^{4t}; \end{aligned}$$

Además, en concreto, evaluando:

$$x(1) = 164.1623295406042 \text{ y } y(1) = 108.828420625117$$

Veamos las soluciones para distintos tamaños de paso, observando que, en efecto, cuanto menor es la h , menor es el error:

h	$w_1(1)$	Error en $w_1(1)$	$w_2(1)$	Error en $w_2(1)$
0.1	121.8007656	42.3615638	80.6774927	28.1509278
0.01	151.8808768	12.2814526	100.6438640	8.1845565
0.001	162.8604735	1.3018560	107.9608232	0.8675973
0.0001	164.0967915	0.0655380	108.7848205	0.0436001
0.00001	164.1557725	0.006557	108.8240585	0.0043621

Table 5: Errores obtenidos con el método de Euler

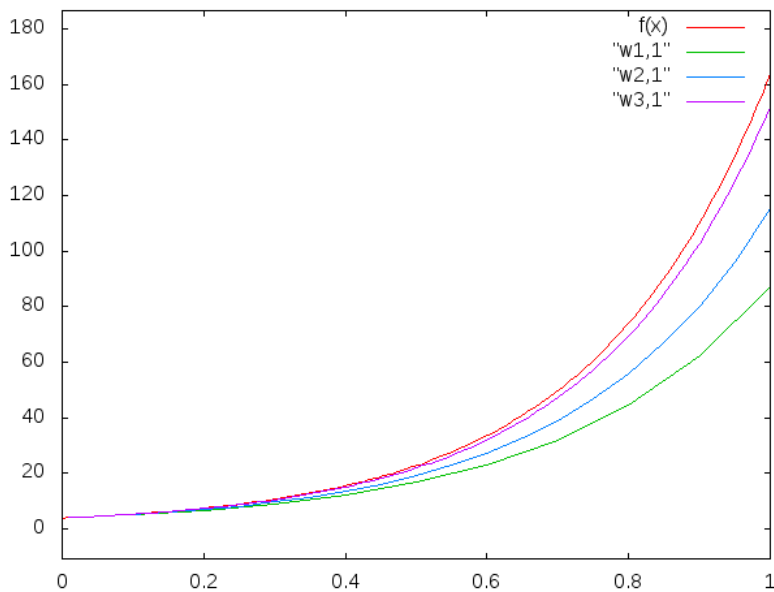


Figure 2: Aproximación para marcas de paso de 0.1, 0.05 y 0.01 respectivamente

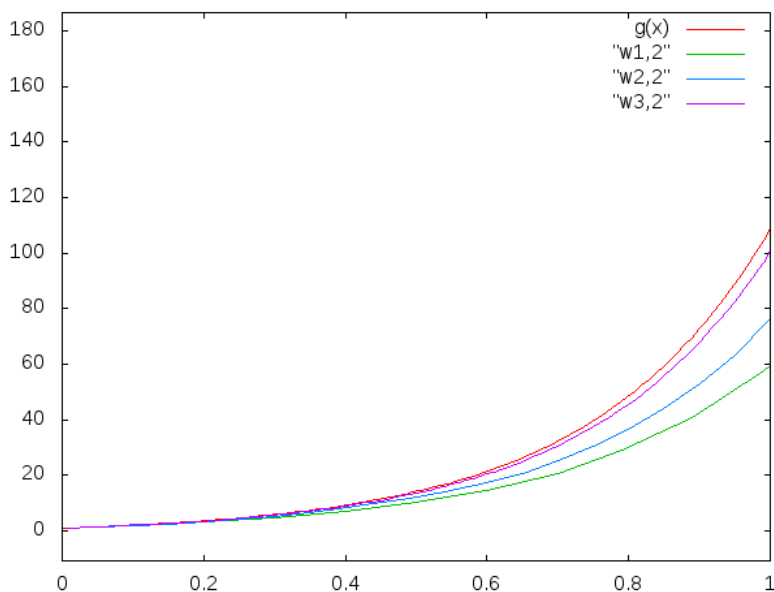


Figure 3: Aproximación para marcas de paso de 0.1, 0.05 y 0.01 respectivamente

6.6 Método de Euler para ecuaciones de orden superior (Ejercicio práctico)

En primer lugar vamos a realizar el paso de una ecuación diferencial de orden superior a un sistema, sobre el cual posteriormente podremos aplicar el método de Euler para sistemas. Para ejemplificar el proceso vamos a usar el siguiente problema de valores iniciales:

$$x''' = 2x'' - 2x'$$

$$x(0) = 1, x'(0) = 1, x''(0) = 0$$

Realizamos el cambio de variable indicado en la sección *Ecuaciones diferenciales de orden superior y reescritura como sistemas*:

$$\begin{cases} y_0' = x' = y_1 \\ y_1' = x'' = y_2 \\ y_2' = x''' = 2y_2 - 2y_1 \end{cases}$$

A continuación pasaremos a resolver el sistema con el método de Euler. Para ello consideraremos el intervalo $[0, \pi/4]$ y 25, 50 y 100 nodos. Los datos obtenidos general la siguiente gráfica:

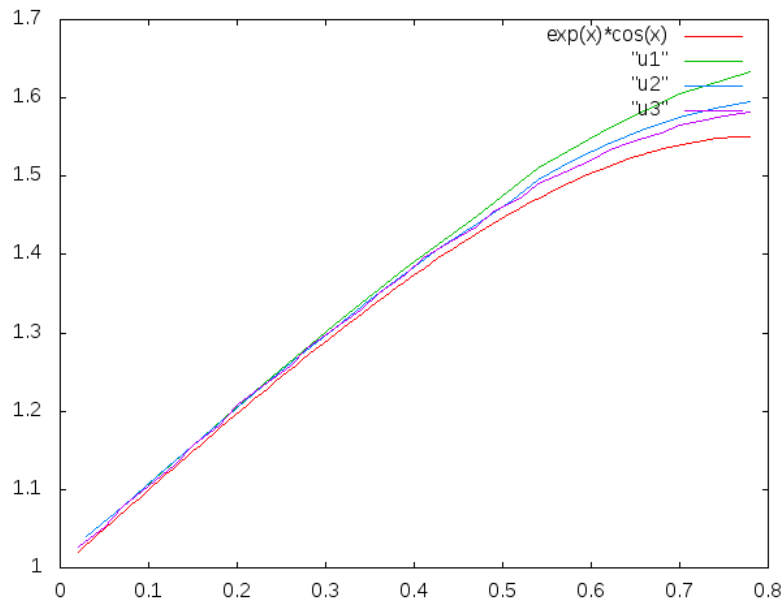


Figure 4: Aproximación para 25 (w1), 50 (w2) y 100 (w3) nodos

A continuación usaremos de nuevo el método de Euler para sistemas, pero esta vez para tamaños de paso menores, de forma que podamos ver la evolución del error según el número de nodos usado. Tras las ejecuciones los datos obtenidos son los siguientes:

h	$u(\pi/4)$	Error
$\pi/120$	1.581351427438043	0.030468230520017
$\pi/240$	1.566883575533628	0.016000378615602
$\pi/360$	1.561353411976403	0.010470215058378
$\pi/480$	1.558765635465628	0.007882438547603
$\pi/600$	1.557203467884946	0.0063202709669205
$\pi/720$	1.556204299202344	0.0053211022843184
$\pi/840$	1.555409383683734	0.0045261867657087

Table 6: Errores obtenidos con el método de Euler

Anexo 1: Código implementado para el trabajo

Se expone aquí el método que se ha implementado para el trabajo. Consiste en una implementación sencilla del método de Euler en Maxima, cuyo funcionamiento es el siguiente. Se acompaña además de un ejemplo de llamada, para facilitar su comprensión:

1. Se definen las funciones que nos dan el valor de la derivada para cada una de las variables dependientes que aparecen
2. Se introducen en un array
3. Se crea un array con el valor inicial que se tiene para cada una de las variables dependientes
4. Se llama al método especificando:
 - (a) Array con las funciones
 - (b) Array con los valores iniciales
 - (c) Número de variables dependientes que intervienen
 - (d) Valor inicial del que se parte (extremo inferior del intervalo)
 - (e) Valor en el que se quiere aproximar la solución (extremo superior del intervalo)
 - (f) Tamaño del paso
5. El método devuelve el valor de $x_i(b)$ para todas las variables dependientes en un array

```

1 euler_method(arr_fs , init_vals , n_vars , a , b , h):=block([t:a , aprx , aprxb],
   array(aprx , n_vars),
3   array(aprb , n_vars),
   for i:1 thru n_vars do(
5     aprx[i]: init_vals[i]
   ),
7   while t < b do(
     t:t+h,
9     for i:1 thru n_vars do(
       aprxb[i]:aprx[i]
11    ),
     for i:1 thru n_vars do(
13      aprx[i]: aprxb[i] + h*arr_fs[i](t , aprxb)
     ),
15   ),
   return(aprx)
17 );

19 Definicion de las funciones y el array que las contiene:
f1(t , xs):= xs[2];
21 f2(t , xs):= -sin(xs[1]);
fs:[f1 , f2];
23
25 Definicion de los valores iniciales:
vals:[%pi/2 , 0];

27 Llamada al metodo (usando las funciones y los valores definidos , 2 variables
   dependientes , extremo inferior 0 , extremo superior 1 , h=0.2):
euler_method(fs , vals , 2 , 0 , 1 , 0.2);

```

7 Artículo

En esta sección realizaremos un resumen del artículo “Numerical solution of differential equations using multiquadric radial basis function networks” (Nam Mai-Duy, Thanh Tran-Cong; 2001). En este artículo se introducirá una técnica novedosa para la resolución de ecuaciones diferenciales ordinarias, a partir de ahora ODEs, y ecuaciones diferenciales parciales, PDEs. Previo al resumen veamos una breve descripción de red neuronal:

En el contexto de la informática una red neuronal es un conjunto de unidades menores llamadas neuronas interconectadas entre sí. Cada neurona, a partir de un conjunto de entradas, es capaz de proporcionar una salida en la que aparentemente se aplica inteligencia. Para generar dicha salida se hace uso de funciones con carácter probabilísticos cuyos pesos habrá que ajustar en la fase de entrenamiento. Esta fase de entrenamiento consiste en proporcionar un conjunto de entradas con sus correspondientes salidas, de forma que la red neuronal se ajusta según estos datos proporcionados, es decir, modifica dichas funciones a través del conocimiento proporcionado y de esta forma “aprende” a dar la respuesta correcta para el problema en cuestión con una probabilidad de acierto muy elevada.

Los métodos clásicos para la obtención de aproximaciones de la solución de ecuaciones diferenciales suponen un proceso costoso al tener que seleccionar un dominio adecuado con un número finito de elementos para proceder con el mismo. En este artículo, como ya se ha comentado, se presenta un método novedoso para la resolución de ecuaciones diferenciales lineales haciendo uso de redes neuronales artificiales en el que no habrá problema para seleccionar el dominio.

En el método presentado en el artículo se hace uso de redes de base radial, que son redes neuronales en las que las funciones que determinan la salida serán funciones de base radial, esto es, funciones reales cuya imagen para cada punto viene determinada por la distancia al origen, es decir, diremos que $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ es radial si

$$\exists \Phi : \mathbb{R} \rightarrow \mathbb{R} \text{ tq } \phi(x) = \Phi(\|x\|) \quad \forall x \in \mathbb{R}^n$$

Existen diversas funciones de base radial, pero los autores harán uso de las funciones multicuadráticas, que son aquellas que se definen como sigue:

$$\phi(r) = \sqrt{\epsilon^2 + r^2}$$

Notaremos a los métodos numéricos que hacen uso de las redes neuronales con funciones de base radial como RBFN, y podremos clasificarlas como directos (DRBFN) e indirectos (IRBFN). Recordemos en primer lugar que los métodos numéricos directos para la aproximación de funciones y sus derivadas son aquellos que aproximan la función y la aproximación de las sucesivas derivadas serán las sucesivas derivadas de dicha aproximación, mientras que los métodos indirectos son aquellos en los que en primer lugar se aproximan las derivadas de mayor orden y posteriormente se consiguen las de orden menor integrando dicha aproximación. Puesto que se hace uso de funciones multicuadráticas, la función aproximación será de la forma:

$$u(x) = \sum_{i=1}^m w^{(i)} g^{(i)}(x) = \sum_{i=1}^m w^{(i)} \sqrt{\epsilon^{(i)2} + x^2}$$

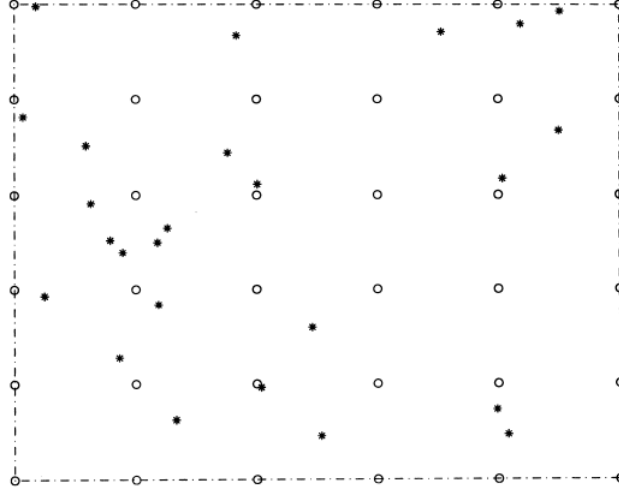


Figure 5: Distribución de los centros (\circ) y nodos (\bullet)

Donde m es el número de neuronas utilizadas en el método y $w^{(i)}$ los pesos asociados a cada neurona obtenidos durante la fase de entrenamiento. Más adelante veremos como determinar $\epsilon^{(i)}$. La norma del error que se cometerá vendrá determinada por:

$$N_e = \sqrt{\frac{\sum_{i=1}^m (u_e(x^{(i)}) - u(x^{(i)}))^2}{\sum_{i=1}^m u_e(x^{(i)})^2}}$$

Para aplicar el método debemos contar con la imagen de una serie de puntos de colocación o nodos $\{u(x^{(i)}) = y^{(i)}\}_{i=1}^n$ que servirán como entrenamiento para la determinación de la función. Se contará además con otro tipo de puntos, que llamaremos centros, y que se repartirán uniformemente en forma de malla a lo largo de todo el dominio de la función a aproximar. En la figura 1 podemos ver la distribución de los centros y los nodos.

A lo largo del artículo los autores darán resultados de la aplicación del método para la resolución de ODEs en dominios regulares y de PDEs en elipses. Entenderemos por dominio regular a aquel que esté formado por el producto cartesiano de intervalos. En dichos dominios nos será relativamente fácil establecer los centros, y aquí es donde reside la única libertad que podemos aplicar en el método. Veamos por qué.

A la hora de crear la malla en nuestro dominio podremos hacerla más densa o menos con total libertad, pero estableceremos la siguiente relación con $\epsilon^{(i)}$:

$$\epsilon^{(i)} = \beta d^{(i)}$$

Donde $d^{(i)}$ es la distancia del i -ésimo centro al centro más cercano. Podremos elegir $\beta > 0$ con libertad, aunque dicho parámetro influirá en la función aproximación obtenida. Aunque no hay forma conocida de determinar el valor óptimo de β en cada ocasión, experimentalmente los autores han obtenido que los valores óptimos suelen centrarse en el intervalo $[7,10]$.

A continuación se pasa al estudio de la aplicación del método para PDEs en dominios elípticos. El problema que surge en este momento es la generación de la malla. La solución es simple: si Ω es nuestro dominio elíptico, tomaremos Ω^* dominio regular tal que $\Omega^* \supset \Omega$, y aplicaremos el método para el dominio Ω^* , aunque los nodos se encontrarán todos dentro de Ω .

Tras una serie de ejemplos los autores muestran como el método IRBFN consigue resultados más adecuados que los conseguidos con el método directo, y además, los resultados son mejores para un mayor rango de β . Los autores dicen no tener una demostración teórica que corrobore este hecho, pero dan la idea intuitiva del porqué de este hecho: al cometer un error determinado con el método directo, al derivar para obtener las sucesivas derivadas intensificamos el error, mientras que en el método indirecto disminuimos el efecto del error cometido.

Durante el artículo también se realizan comparaciones con los métodos clásicos y se concluye que el método directo, en términos de error, es igual de adecuado que los métodos clásicos, mientras que el método indirecto consigue resultados de varios órdenes de magnitud mayor.

Finalmente se habla de una futura adecuación del método para dominios irregulares construyendo un recubrimiento finito de dominios regulares y procediendo de igual forma que con el caso del dominio elíptico.