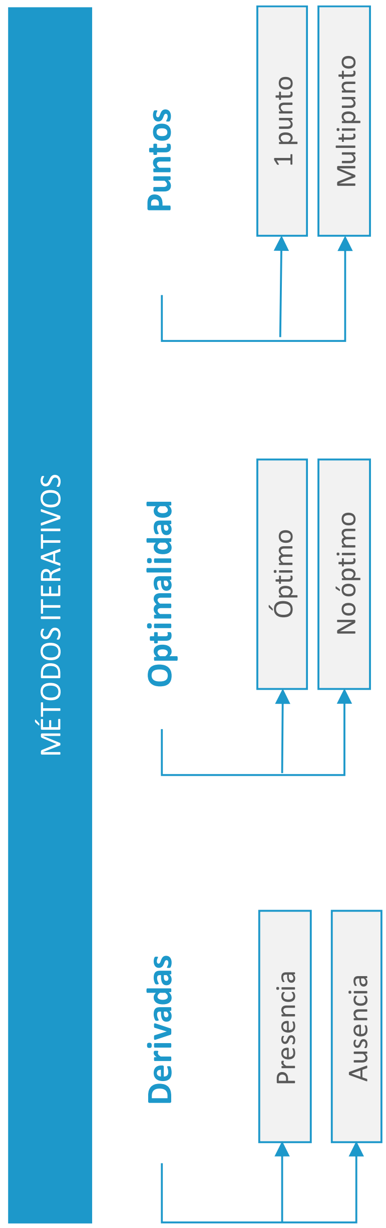
Métodos Numéricos Avanzados en Ingeniería

Ecuaciones no lineales

Índice

Esquema



MÉTODOS ITERATIVOS
Derivadas
Presencia
Ausencia
Optimalidad
Óptimo
No óptimo
Puntos
1 punto
Multipunto

Ideas clave

11.1. ¿Cómo estudiar este tema?

Para estudiar este tema lee las Ideas clave que encontrarás a continuación

Existen multitud de procesos dentro de la ciencia y la ingeniería, entre otros campos de aplicación, en los que es necesaria la resolución de la ecuación $f(x) = 0$ o del sistema de ecuaciones $F(x) = 0$, siendo tanto f como F no lineales. En numerosas ocasiones, no es posible obtener una solución analítica a las ecuaciones o los sistemas de ecuaciones. Entonces, ¿no podremos obtener nunca la solución de estos problemas? La respuesta es no.

A lo largo de este tema, vas a aprender a utilizar **diferentes métodos de resolución de ecuaciones no lineales**. En el tema siguiente, nos centraremos en la resolución de sistemas de ecuaciones no lineales. Partiremos del archiconocido **método de Newton** y, a partir de él, iremos descubriendo **otros métodos**.

Los métodos de resolución se basan en obtener una secuencia de valores, uno en cada iteración. En cada una, nos aproximamos cada vez más a la solución. Pero, si nuestro objetivo es conocer la solución, ¿cómo sabemos que la hemos alcanzado? Hay otras maneras de **saber que estás en la solución sin necesidad de conocerla.**

Los apartados de los que consta este tema son:

- Métodos iterativos:
  - Tipos de métodos iterativos.
  - Implementación en Matlab.

- Ecuaciones no lineales:
    - Conceptos previos.
    - Métodos de un punto.
    - Métodos multipunto.

- Comparativa numérica.
- Implementación en Matlab del método de Newton.

11.2. Introducción a los métodos iterativos

Un método iterativo para la resolución de ecuaciones o sistemas de ecuaciones no lineales es un procedimiento que trata de **aproximarse a la solución partiendo de un valor o una serie de valores iniciales**. De esta manera, un método iterativo obtiene una secuencia de valores:

$$\{x_0, x_1, x_2, \dots, x_n\}$$

Tipos de métodos iterativos

Para la generación de cada uno de los elementos de la secuencia, se utiliza la expresión del **método iterativo**. Si el método solo necesita **un iterado** para obtener el anterior, se denomina método sin memoria, y su expresión general es:

$$x_{k+1} = \phi(x_k), k = 0, \dots, n$$

En caso de que requiera **más de un iterado anterior para obtener el siguiente**, el método se dice que tiene memoria, siendo su expresión general:

$$x_{k+1} = \phi(x_k, x_{k-1}, \dots, x_{k-m}), k = 0, \dots, n$$

donde $m+1$ son los iterados previos que es necesario conocer para obtener el siguiente iterado.

Ejemplo 1. Sean $\phi_1(x_k) = x_k^2/4$ y $\phi_2(x_k, x_{k-1}) = x_{k-1}x_k$. Tomando como estimación inicial $x_0 = 2$ para el método sin memoria y $(x_{-1}, x_0) = (1/3,1)$, obtén una tabla con los sucesivos iterados hasta que la diferencia con el valor del iterado anterior sea menor que 10^{-3}.

Obtengamos la secuencia de iterados del método sin memoria:

$$x_0 = 2 \to x_1 = \phi(x_0) = \phi(4) = \frac{2^2}{4} = 1, |x_1 - x_0| = |2-1| = 1$$
$$x_1 = 1 \to x_2 = \phi(x_1) = \phi(1) = \frac{1^2}{4} = \frac{1}{4}, |x_2 - x_1| = \left|1 - \frac{1}{4}\right| = \frac{3}{4}$$

k	1	2	3	4	5
x_k	1	$\frac{1}{4}$	$\frac{1}{64}$	$\frac{1}{16384}$	$\frac{1}{1073741824}$
$\lvert x_k - x_{k-1}\rvert$	1	$\frac{3}{4}$	$\frac{15}{64}$	$\frac{255}{16384}$	$\frac{65535}{1073741824}$

Tabla 1. Secuencia de iterados. Método sin memoria.

Vemos que la diferencia entre los iterados 4 y 5 es de $\frac{95535}{1073741824} < 10^{-3}$, de forma que no obtenemos más iterados.

Otra clasificación de los métodos iterativos consiste en la **cantidad de puntos que se toman** para obtener el siguiente iterado. Particularizando sobre métodos sin memoria, la expresión general de un método iterativo de un punto es:

$$x_{k+1} = \phi(x_k)$$

Mientras que para métodos multipunto esa expresión es:

$$y_k = \psi(x_k)$$
$$x_{k+1} = \phi(x_k, y_k)$$

Ejemplo 2. Determina si los siguientes métodos son de un punto o multipunto.

Método de Newton:

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

Método de Traub:

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

El método de Newton es un método de un punto, mientras que el método de Traub es un método multipunto.

Implementación en Matlab

Un método iterativo se encarga de realizar operaciones de manera iterativa. Pero es necesario que este proceso finalice en un momento determinado para obtener resultados. Para que un proceso iterativo deje de iterar, hay que establecer unas **condiciones de parada**. Estas condiciones pueden ser alguna de las siguientes o una combinación entre ellas.

- Los dos últimos iterados están muy próximos:

$$|x_k - x_{k-1}| < \epsilon$$

- Se ha alcanzado un valor muy próximo a la solución:

$$|f(x_k)| < \epsilon$$

- Hemos iterado muchas veces y no hemos conseguido ninguna solución:

$$k > M$$

Realizar manualmente las operaciones que conlleva un método iterativo puede resultar muy costoso manualmente. Con la evolución de las computadoras a lo largo del siglo XX, el cálculo numérico ha obtenido muchos más adeptos.

A la hora de implementar el método iterativo en Matlab, vamos a relacionar algunos conceptos previos con el nombre que le daremos en el código. La Tabla 1. Relación entre parámetros y código. establece dicha relación.

Parámetro	k	ϵ	M
Código	iter	tol	maxiter

Tabla 1. Relación entre parámetros y código.

La estructura de un método iterativo en Matlab se presenta en el siguiente pseudocódigo:

```
function [sol,ACOC]=metodoIterativo(x0,f,tol,maxiter)
% Inicialización de las variables
iter=1;
condición_de_parada=1;
while condición_de_parada
      x=Aplicar_método_iterativo(x0);
      actualizar_condición_de_parada;
      actualizar_valores;
      iter=iter+1;
end
% Preparar datos de salida
sol=x;
ACOC=evaluar_ACOC(x);
```

Ejemplo 3. Implementa en Matlab un método iterativo cuyo criterio de parada sea:

$$\begin{cases} |x_k - x_{k-1}| < 10^{-3} \\ k \geq 40 \end{cases}$$

Basándonos en el **¡Error! No se encuentra el origen de la referencia.**anterior, desde la consola de Matlab, ejecutaríamos:

```
>> [sol,ACOC]=metodoIterativo(x0,f,1e-3,40)
```

Y dentro de la implementación del anterior modificaríamos la línea:

```
while abs(x(iter)-x(iter-1))>=tol&&iter<maxiter
```

Otro aspecto importante es **cómo introducimos la ecuación no lineal f en Matlab.** Para ello, deberemos tener en cuenta qué necesita saber el método iterativo acerca de la función: su valor, el valor de su primera derivada, el valor de su segunda derivada, etc. Para ello, generaremos un archivo .m con el nombre de la función, dándole como parámetro de entrada la variable, y como parámetros de salida los valores que requiera el método iterativo. La estructura general se recoge en el siguiente pseudocódigo:

```
function [f,df,d2f]=funcion_f(x)
f=expresión_función_f;
df=expresión_funcion_derivada_primera;
d2f=expresión_función_derivada_segunda;
```

Ejemplo 4. Escribe un archivo .m para la ecuación no lineal $x = e^{-x}$. El método iterativo que lo resuelve requiere del valor de la función y sus dos primeras derivadas.

En primer lugar, debemos obtener las expresiones de f, f' y f'':

$$f(x) = x - e^{-x} = 0, f'(x) = 1 + e^{-x}, f''(x) = -e^{-x}$$

A continuación, escribimos el código basándonos en el pseudocódigo visto:

```
function [f,df,d2f]=ejemplo4_f(x)
f=x-exp(-x);
df=1+exp(-x);
d2f=-exp(-x);
```

11.3. Métodos iterativos para resolver ecuaciones no lineales

La resolución de ecuaciones no lineales tiene tres posibilidades. La primera de ellas es la **solución analítica**; no obstante, no siempre es posible obtener esta solución. La segunda posibilidad es el **método gráfico**. A partir de una ecuación $f(x) = 0$, es suficiente con representar dicha función y comprobar en qué puntos corta con el eje de abscisas.

Ejemplo 5**. Obtén las raíces de la ecuación $x = e^{-x}$ a partir del método gráfico.**

En primer lugar, tenemos que obtener una expresión de la forma $f(x) = 0$, por lo que ponemos todos los términos de la ecuación a un lado de la igualdad. Por tanto, $f(x) = x - e^{-x} = 0$.

A continuación, representamos gráficamente la función f. Para ello, utilizando Matlab, ejecutamos las siguientes líneas de código.

```
>> x=linspace(-1,1);
>> f=x-exp(-x);
>> plot(x,f)
```

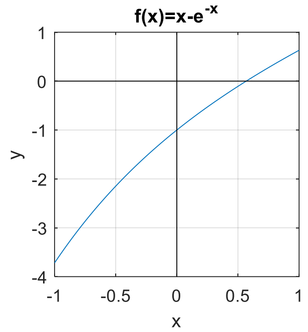


Figura 1. Representación de la función del ejemplo 5.

Vemos que la gráfica corta al eje de abscisas alrededor del punto $x = 0.6$, pero no lo podemos saber con la exactitud que pretendamos.

La tercera posibilidad es el uso de **métodos iterativos**. A partir de una estimación inicial x_0, el método iterativo es capaz de obtener una solución tan aproximada como se requiera.

Conceptos previos

Para evaluar las características de un método iterativo, se pueden utilizar diferentes parámetros. El principal es el **orden de convergencia**. Dada la sucesión $\{x_k\}_{k\geq 0}$ generada por un método iterativo, tiene orden de convergencia local p si encontramos C y p tales que:

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = C$$

donde α es la solución de la ecuación no lineal. La demostración del orden de convergencia de un método iterativo se establece en el teorema 1.

Teorema 1. Sea g una función de punto fijo tal que $g^{(p)}$ es continua en un entorno de la solución α. El método iterativo $x_{k+1} = g(x_k)$ es de orden p si y solo si

$$g(\alpha) = \alpha, g'(\alpha) = 0, \dots, g^{(p-1)}(\alpha) = 0, g^p(\alpha) \neq 0$$

Haciendo el desarrollo en **serie de Taylor** y realizando manipulaciones algebraicas, se obtiene una ecuación del error del tipo:

$$e_{k+1} = Ke_k^p + \mathcal{O}\left(e_k^{p+1}\right)$$

donde $e_k = x_k - \alpha$, $K \in \mathbb{R}$ y $\mathcal{O}\left(e_k^{p+1}\right)$ representa a los términos de orden superior.

A nivel computacional, se utiliza el **orden aproximado de convergencia computacional** ACOC (Cordero & Torregrosa, 2007), definido como:

$$ACOC = \frac{\ln(|x_{k+1} - x_k|/|x_k - x_{k-1}|)}{\ln(|x_k - x_{k-1}|/|x_{k-1} - x_{k-2}|)}$$

Por otro lado, tenemos el **índice de eficiencia** I (Ostrowski, 1960), cuya expresión es:

$$I = p^{1/d}$$

donde d es el número de evaluaciones de la función f en cada iteración. De forma análoga, se define el **índice de eficiencia computacional** IC (Traub, 1982) como:

$$IC = p^{1/(d+op)}$$

donde op representa el número de productos y cocientes por iteración.

La **conjetura de Kung y Traub** (Kung & Traub, 1974) caracteriza a un método como óptimo o no óptimo. Esta conjetura establece que:

$$p \leq 2^{d-1}$$

Cuando se cumple la igualdad, se dice que un método es óptimo.

Métodos de un punto

Vamos a ver a continuación una serie de métodos iterativos de un punto. El método por excelencia es el método de Newton. Su expresión iterativa es:

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

El orden de convergencia del método de Newton se establece en el teorema 2.

Teorema 2. Sea α un cero simple de una función suficientemente diferenciable $f: D \in \mathbb{R} \to \mathbb{R}$ en un conjunto convexo D. Si la estimación inicial x_0 es suficientemente próxima a α, el método de Newton alcanza orden de convergencia cuadrático, siendo su ecuación del error:

$$e_{k+1} = c_2 e_k^2 + \mathcal{O}(e_k^3)$$

donde $c_k = \frac{1}{k!}\frac{f^k(\alpha)}{f'(\alpha)}, k = 2,3, \ldots$ y $e_k = x_k - \alpha$

Demostración. Desarrollando en serie de Taylor en torno a α

$$\begin{aligned} f(x_k) &= f(\alpha) + f'(\alpha)(x_k - \alpha) + \frac{1}{2}f''(\alpha)(x_k - \alpha)^2 + \mathcal{O}((x_k - \alpha)^3) \\ &= f'(\alpha)[e_k + c_2 e_k^2] + \mathcal{O}(e_k^3) \end{aligned}$$

$$f'(x_k) = f'(\alpha) + f''(\alpha)(x_k - \alpha) + \mathcal{O}((x_k - \alpha)^2) = f'(\alpha)[1 + 2c_2 e_k] + \mathcal{O}(e_k^2)$$

De este modo:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \leftrightarrow x_{k+1} - \alpha = x_k - \alpha - \frac{f'(\alpha)[e_k + c_2 e_k^2]}{f'(\alpha)[1 + 2c_2 e_k]} + \mathcal{O}(e_k^3) \leftrightarrow e_{k+1}$$
$$= e_k - e_k + c_2 e_k^2 + \mathcal{O}(e_k^3) = c_{2e_k^2} + \mathcal{O}(e_k^3)$$

A continuación, se presentan una serie de **métodos iterativos que utilizan el grado de convexidad logarítmica definido por:**

$$L_f(x_k) = \frac{f(x_k) f''(x_k)}{\left(f'(x_k)\right)^2}$$

Se trata de métodos de un punto, y sus características se recogen en la Tabla 2**.**

- Método de Halley:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\left[1 + \frac{L_f(x_k)}{2 - L_f(x_k)}\right]$$

- Método de Chebyshev:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\left[1 + \frac{L_f(x_k)}{2}\right]$$

- Método de SuperHalley:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\left[1 + \frac{L_f(x_k) - 2}{2\left(L_f(x_k) - 1\right)}\right]$$

- Familia de métodos de Chebyshev-Halley:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\left[1 + \frac{1}{2}\frac{L_f(x_k)}{1 - \beta L_f(x_k)}\right]$$

Nótese que para $\beta = 0$ estamos ante el método de Chebyshev, para $\beta = 1/2$ tenemos el método de Halley, para $\beta = 1$ estamos ante el método de SuperHalley y para $\beta \to \infty$ nos encontramos con el método de Newton.

Método	p	d	I	**Óptimo**
Newton	2	2	$\sqrt{2}$	✔
Halley	3	3	$\sqrt[3]{3}$	✕
Chebyshev	3	3	$\sqrt[3]{3}$	✕
SuperHalley	3	3	$\sqrt[3]{3}$	✕

Tabla 2. Características de los métodos iterativos de un punto.

Métodos multipunto

Los métodos de un punto tienen una serie de limitaciones como indican los teoremas 3 y 4, que impiden aumentar el orden de convergencia y el índice de eficiencia.

Teorema 3. Sea $x_{k+1} = g(x_k)$ un método iterativo de un punto que utiliza p evaluaciones funcionales por iteración. Entonces, su orden de convergencia es como máximo p.

Teorema 4. Para diseñar un método iterativo de un punto de orden p su expresión iterativa debe contener derivadas de al menos hasta orden $p - 1$.

Para aumentar el orden de convergencia tenemos que recurrir a los métodos multipunto. El teorema 5 nos indica cómo podemos incrementar dicho orden.

Teorema 5. Sean $g_1(x)$ y $g_2(x)$ dos funciones de punto fijo para $f(x) = 0$. Sean los esquemas iterativos $x_{k+1} = g_1(x_k)$ y $x_{k+1} = g_2(x_k)$ de órdenes p_1 y p_2, respectivamente. Entonces, el orden de convergencia del método iterativo asociado a la función de punto fijo $g(x) = g_1(g_2(x))$ es $p_1 p_2$.

Dentro de este tipo de métodos, podemos distinguir entre tres técnicas de diseño: el uso de fórmulas de cuadratura, la composición y las funciones peso.

Fórmulas de cuadratura

Podemos escribir la ecuación no lineal como:

$$f(x)=f(x_0)+\int_{x_0}^{x} f'(t)dt$$

Si aproximamos la integral a partir de la regla de los trapecios:

$$\int_{x_0}^{x} f'(t)dt \approx \frac{1}{2}(x-x_0)\big(f'(x)+f'(x_0)\big)$$

para $x=\alpha$ la ecuación no lineal queda como:

$$0\approx f(x_0)+\frac{1}{2}(\alpha-x_0)\big(f'(\alpha)+f'(x_0)\big)\leftrightarrow \alpha\approx x_0-\frac{2f(x_0)}{f'(\alpha)+f'(x_0)}$$

Utilizando el método de Newton como predictor, obtenemos el **método de los trapecios**, cuya expresión es:

$$\begin{array}{rcl} y_k & = & x_k-\dfrac{f(x_k)}{f'(x_k)} \\ x_{k+1} & = & x_k-\dfrac{2f(x_k)}{f'(y_k)+f'(x_k)} \end{array}$$

Otros métodos generados a partir de las fórmulas de cuadratura son:

- Método de punto medio.

$$\int_{x_0}^{x} f'(t)dt \approx (x-x_0)f'\left(\frac{x+x_0}{2}\right) \rightarrow \begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)} \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f'\left(\frac{x_k+y_k}{2}\right)} \end{cases}$$

- Método de Simpson:

$$\int_{x_0}^{x} f'(t)dt \approx \frac{1}{6}(x-x_0)\left[f'(x) + 4f'\left(\frac{x+x_0}{2}\right) + f'(x_0)\right]$$

$$\rightarrow \begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)} \\ x_{k+1} = x_k - \dfrac{6f(x_k)}{f'(x) + 4f'\left(\frac{x+x_0}{2}\right) + f'(x_0)} \end{cases}$$

- Método de Gauss-Legendre:

$$\int_{x_0}^{x} f'(t)dt \approx \frac{1}{2}(x-x_0)\sum_{i=1}^{n} A_i f'(t_i)$$

$$\rightarrow \begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)} \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f'\left(\frac{(3+\sqrt{3})x_k + (3-\sqrt{3})y_k}{6}\right) + f'\left(\frac{(3-\sqrt{3})x_k + (3+\sqrt{3})y_k}{6}\right)} \end{cases}$$

donde A_i son los pesos y t_i los nodos de la cuadratura de Gauss.

Método	p	d	I	**Óptimo**
Trapecios	3	3	$\sqrt[3]{3}$	✕
Punto Medio	3	3	$\sqrt[3]{3}$	✕
Simpson	3	3	$\sqrt[3]{3}$	✕
Gauss-Legendre	3	4	$\sqrt[4]{3}$	✕

Tabla 3. Características de los métodos iterativos multipunto a partir de fórmulas de cuadratura.

La Tabla 3 recoge las características de los métodos multipunto a partir de fórmulas de cuadratura. Como se puede comprobar, en ningún caso se ha obtenido un método óptimo.

Composición de métodos

Otra alternativa para la generación de métodos iterativos de mayor orden de convergencia es la **composición de métodos.** Por ejemplo, si en el primer paso utilizamos el método de Newton y en el segundo paso lo volvemos a utilizar, obtenemos:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(y_k)}{f'(y_k)} \end{aligned}$$

A este método lo denominaremos **Newton doble** y es un método de orden 4. El problema es que no es óptimo, debido a que realiza 4 evaluaciones funcionales.
Si volvemos a utilizar la composición de Newton, obtendremos el método de **Newton triple**, cuya expresión es:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ z_k &= y_k - \frac{f(y_k)}{f'(y_k)} \\ x_{k+1} &= z_k - \frac{f(z_k)}{f'(z_k)} \end{aligned}$$

En este caso, tenemos un método de orden 8, pero tampoco es óptimo porque tiene 6 evaluaciones funcionales.

Para reducir el número de evaluaciones funcionales, se recurre a la técnica de congelar la derivada de $f'(x_k)$. Para el caso de Newton doble, consistiría en **modificar el denominador del segundo paso y mantener el del primer paso**. Si realizamos esta operación, obtenemos el **método de Traub o Potra-Pták** (Traub, 1982), cuya expresión es:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(y_k)}{f'(x_k)} \end{aligned}$$

El orden del método es 3 y, en este caso, solo tenemos 3 evaluaciones funcionales.

Funciones peso

El procedimiento de las funciones peso consiste en multiplicar al sumando que contiene las evaluaciones funcionales por una función peso del tipo $H(\mu), H \in \mathbb{R}$ y μ es una relación entre la función evaluada en diferentes puntos.

Si al método de Traub ó Potra-Pták le incluimos una función peso en el segundo paso, obtenemos:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - H(\mu)\frac{f(y_k)}{f'(x_k)} \end{aligned}$$

siendo $\mu = f(y)/f(x)$.

Teorema 6. Sea α un cero simple de una función suficientemente diferenciable, y sea x_0 una estimación inicial próxima a α. Si elegimos una función H tal que $H(0) = 1, H'(0) = 2$ y $|H''(0)| < \infty$, entonces el método iterativo anterior tiene orden de convergencia 4 y su ecuación del error es:

$$e_{k+1} = \left[\left(5 - \frac{H''(0)}{2}\right)c_2^3 - c_2c_3\right]e_k^4 + \mathcal{O}(e_k^5)$$

siendo $c_k = \frac{1}{k!}\frac{f^{(k)}(\alpha)}{f'(\alpha)}, k = 2,3, \dots, e_k = x_k - \alpha$.

La **familia de King** (King, 1973) se define a partir del teorema 6, en el que la función peso es:

$$H(\mu) = \frac{1 + \beta\mu}{1 + (\beta - 2)\mu}$$

de forma que los métodos iterativos tienen la expresión:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(x_k) + \beta f(y_k)}{f(x_k) + (\beta - 2)f(y_k)} \frac{f(y_k)}{f'(x_k)} \end{aligned}$$

Se trata de una familia de métodos iterativos óptima de orden 4. Un caso particular de dicha familia es el **método de Ostrowski** (Ostrowski, 1960), en el que $\beta = 0$, siendo su esquema iterativo:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(x_k)}{f(x_k) - 2f(y_k)} \frac{f(y_k)}{f'(x_k)} \end{aligned}$$

Por último, presentaremos el **método de Jarratt** (Jarratt, 1969). Este método se fundamenta en la técnica de funciones peso. No obstante, difiere en algunos aspectos respecto a la Familia de King. Por un lado, el paso de Newton tiene un coeficiente amortiguador $\frac{2}{3}$, mientras que en el segundo paso, la variable de la función peso pasa a ser $\mu = \frac{f'(y)}{f'(x)}$ y la función peso es $H(\mu) = \frac{3\mu+1}{6\mu-2}$. El método iterativo tiene la expresión:

$$
\begin{aligned}
y_k &= x_k - \frac{2}{3}\frac{f(x_k)}{f'(x_k)} \\
x_{k+1} &= y_k - \frac{3f'(y_k)+f'(x_k)}{6f'(y_k)-2f'(x_k)}\frac{f(y_k)}{f'(x_k)}
\end{aligned}
$$

Se trata de un método óptimo de orden de convergencia 4.

11.4. Comparativa numérica

Una de las prácticas más habituales para comparar métodos iterativos es el uso de **comparativas numéricas**. En este tipo de comparativas, se evalúan las características de cada uno de los métodos al resolver una serie de funciones no lineales denominadas funciones test bajo las mismas condiciones de parada. Los parámetros que se examinan son:

- El valor de la solución: x_k.
- El número de iteraciones necesario para converger: k o `iter`.
- El valor absoluto de la función evaluada en el último iterado: $|f(x_k)|$.
- El valor absoluto de la diferencia entre los dos últimos iterados: $|x_k - x_{k-1}|$.
- El orden de convergencia computacional aproximado: $ACOC$.

Matlab trabaja con una precisión por defecto. Sin embargo, existen determinadas aplicaciones como son precisamente las ecuaciones no lineales, en las que necesitamos una mayor precisión en los cálculos. Para ello, Matlab pone a nuestra disposición la **aritmética de precisión variable**. Este tipo de aritmética consiste en que va a realizar los cálculos con la precisión que le indiquemos. Un valor suficiente y razonable es indicarle que trabaje con 200 dígitos. Para ello, introduciríamos en la primera línea del **¡Error! No se encuentra el origen de la referencia.**mostrado el comando `digits(200)`.

Asimismo, es necesario indicarle que la estimación inicial también va a utilizar aritmética de precisión variable, y que todos los cálculos que deriven de dicha estimación inicial los realice con ese tipo de aritmética. Si tenemos una estimación inicial de $x_0 = 0.5$, será suficiente con indicar en la consola de Matlab que la estimación inicial es `vpa(0.5)`.

Ejemplo 6. Aplica el método de Newton para resolver la ecuación no lineal $x = e^{-x}$ tomando como estimación inicial $x_0 = 0.5$. Utiliza como criterio de parada $|x_k - x_{k-1}| < 10^{-9}$. Indica para cada iteración el valor del iterado, el valor de la función, la diferencia entre los dos últimos iterados y el ACOC.

Los resultados de cada iteración se muestran en la siguiente tabla.

$\boldsymbol{x_k}$	$\boldsymbol{\lvert f(x_k)\rvert}$	$\boldsymbol{\lvert x_k - x_{k-1}\rvert}$	$\boldsymbol{ACOC}$
0.566311	0.001305	0.066311	-
0.567143	$1.9648e-7$	$8.321618e-4$	-
0.567143	$4.45743e-15$	$1.25375e-$	2.010119
0.567143	$2.29411e-30$	$2.8443e-15$	2.000063

Tabla 4. Resultados de las interacciones. Ejemplo 6.

Los términos $a \cdot e - b$ equivalen a $a \cdot 10^{-b}$. Es una notación no estandarizada pero muy utilizada en trabajos de investigación dentro de este campo, como por ejemplo (Chicharro, Cordero, Garrido, & Torregrosa, 2018; Behl, Sarría, González, & Magreñán, 2019).

A continuación, vamos a realizar una comparativa al respecto de diferentes métodos numéricos de los que hemos visto a lo largo del apartado 9.3. Estos métodos numéricos van a ser Newton (NE), Halley (HA), Chebyshev (CH), Trapecios (TP), Punto Medio (PM), Newton doble (ND), Traub (TR), Ostrowski (OS) y Jarratt (JA). En todos los casos el criterio de parada es $|x_k - x_{k-1}| < 10^{-6}$.

En la Tabla 55 se recogen los datos correspondientes a la ecuación no lineal $f_1(x) = x - e^{-x}$, tomando $x_0 = 0.5$ como estimación inicial para todos los métodos.

Método	k	$\|f(x_k)\|$	$\|x_k - x_{k-1}\|$	$ACOC$
NE	3	$1.964804e-7$	$1.253749e-7$	2.01011
HA	3	$4.488692e-8$	$2.864251e-8$	2.02441
CH	3	$4.037736e-8$	$2.576494e-8$	2.013384
TP	3	$7.771561e-16$	$4.440892e-16$	3.004292
PM	3	$1.110223e-16$	$1.110223e-16$	2.628669
ND	2	$1.964804e-7$	$1.253749e-7$	-
TR	3	$8.881784e-16$	$6.661338e-16$	2.978505
OS	3	$1.916682e-7$	$1.223042e-7$	2.007250
JA	2	$1.074214e-7$	$6.854602e-8$	-

Tabla 5. Comparativa numérica para $f_1(x) = x - e^{-x}$, con $x_0 = 0.5$.

Las conclusiones que se obtienen de los resultados numéricos de la tabla 5 son solo aplicables sobre una función test, un punto inicial y una serie de métodos en concreto. Si los comportamientos se mantienen para diferentes funciones test y diferentes puntos iniciales, podemos generalizar el comportamiento del método. Pero en caso contrario, solo estaremos en disposición de caracterizar la comparativa para un caso particular.

En este sentido, al aplicar la función test $f_1(x)$ con el punto inicial $x_0 = 0.5$, hemos alcanzado la solución utilizando todos los métodos señalados. Asimismo, podemos observar cómo los métodos Newton Doble y Jarratt son los métodos que menos iteraciones han tardado en converger a la solución. Debido a que tienen menos de 3 iteraciones, no podemos conocer el valor del ACOC. Los métodos de trapecios, punto medio y Traub son los que obtienen una solución más precisa, pues sus valores de $|f(x_k)|$ son los más pequeños, del orden de 10^{-16}. Además, cabe destacar que el método de Trapecios obtiene el mayor orden de convergencia computacional, siendo próximo a 3.

En la tabla 6 recogemos los resultados de la aplicación de los métodos sobre la función $f_2(x) = \sin(x) - x^2 + 1$, tomando como estimación inicial $x_0 = 1$ y usando como criterio de parada $|x_k - x_{k-1}| < 10^{-6}$.

Método	k	$\lvert f(x_k)\rvert$	$\lvert x_k - x_{k-1}\rvert$	$ACOC$
NE	5	$1.394908e-8$	$5.246438e-9$	2.000070
HA	11	$3.021269e-7$	$1.136339e-7$	2.005224
CH	9	$2.203148e-8$	$8.286335e-8$	2.003722
TP	4	$2.775557e-14$	$1.043609e-14$	3.011724
PM	4	$1.110223e-16$	$6.439293e-15$	3.010421
ND	3	$1.394908e-8$	$5.246438e-9$	4.252521
TR	4	$1.541732e-7$	$5.798662e-8$	3.145746
OS	5	$1.933509e-7$	$7.272186e-8$	1.996770
JA	3	$1.037558e-8$	$3.902396e-9$	4.243312

Tabla 6. Comparativa numérica para $f_2(x) = \sin(x) - x^2 + 1$, con $x_0 = 1$.

Podemos observar cómo para el caso particular de la tabla 6 los métodos que menos iteraciones tardan son Newton Doble y Jarratt. Vemos precisamente que el valor de orden aproximado de convergencia computacional es claramente superior al del resto de métodos para estos dos casos. No obstante, los métodos que obtienen soluciones más precisas son punto medio y trapecios.

Analicemos otra función test. En este caso, los resultados de la aplicación de los métodos iterativos sobre $f_3(x) = x^2 - e^x - 3x + 2$ se recogen en la tabla 7, donde se ha tomado como estimación inicial $x_0 = 1$ y como criterio de parada $|x_k - x_{k-1}| < 10^{-6}$.

Método	k	$\lvert f(x_k)\rvert$	$\lvert x_k - x_{k-1}\rvert$	$ACOC$
NE	4	$5.105715e-11$	$1.351191e-11$	1.997943
HA	5	$2.070405e-9$	$5.479190e-10$	2.003359
CH	5	$6.609437e-9$	$1.749143e-9$	2.005352
TP	3	$6.381709e-7$	$1.688877e-7$	3.038062
PM	3	0	$3.925573e-9$	3.314362
ND	3	0	0	3.725609
TR	3	$3.423927e-13$	$9.059418e-14$	2.554927
OS	4	$4.786659e-11$	$1.266758e-11$	1.998132
JA	3	0	0	3.722160

Tabla 7. Comparativa numérica para $f_3(x) = x^2 - e^x - 3x + 2$, con $x_0 = 1$.

Vemos en la tabla 7 cómo hay hasta 5 métodos que obtienen la solución en solo 3 iteraciones. Dentro de estos 5 métodos, hay que destacar tres que se aproximan mucho a la solución: **punto medio, Newton Doble y Jarratt.** El valor 0 no indica que $f(x_k) = 0$, sino que con la precisión de 64 bits que utiliza Matlab, no es capaz de aportar un resultado representable con esos 64 bits. De hecho, para el método de Jarratt, utilizando más de 64 bits, el valor de $f(x_k)$ es $2.680375e-17$. Destaca también que los métodos de Newton Doble y Jarratt son los que mayor valor del ACOC tienen, aproximándose a los valores teóricos del orden de convergencia que, en su caso, son 4.

Vamos a analizar un último caso, que es el de la función no lineal $f_4(x) = \cos(x) - x$, tomando como valor inicial $x_0 = 1$ y como criterio de parada $|x_k - x_{k-1}| < 10^{-6}$. Los resultados numéricos se recogen en la Tabla 8.

Método	k	$\|f(x_k)\|$	$\|x_k - x_{k-1}\|$	$ACOC$
NE	4	$2.847205e-10$	$1.7012335e-10$	1.998848
HA	4	$1.701534e-7$	$1.016683e-7$	1.997838
CH	4	$2.004075e-7$	$1.197455e-7$	1.992555
TP	3	$4.440892e-16$	$2.220446e-16$	2.777595
PM	3	0	$4.089351e-11$	2.947934
ND	3	0	0	3.935961
TR	3	$1.641659e-10$	$9.809075e-11$	2.904098
OS	4	$2.400496e-10$	$1.434320e-10$	1.998513
JA	3	0	0	3.951732

Tabla 8. Comparativa numérica para $f_4(x) = \cos(x) - x$, con $x_0 = 1$.

Una vez obtenidos los resultados de la tabla 8, podemos observar como tenemos varios métodos que convergen en solo 3 iteraciones. De entre estos métodos, destacan de nuevo **Newton Doble y Jarratt**, puesto que se aproximan mucho a la solución final y tienen unos altos valores del orden de convergencia computacional aproximado.

11.5. Implementación en Matlab: el método de Newton

En este apartado vamos a realizar la implementación del método de Newton a partir del primer pseudocódigo, en el que introdujimos el cuerpo de un método iterativo. Recordemos que el método de Newton tiene la expresión:

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

de forma que la función que queremos resolver nos tiene que devolver, en el archivo .m, el valor de la función y su derivada.

El código del método de Newton para ecuaciones se representa en el siguiente código. Señalar que el criterio de parada está formado por el cumplimiento de una de las dos siguientes condiciones: $|x_k - x_{k-1}| < \epsilon$ ó $k > M$.

```
function [sol,iter,i1,i2,ACOC]=NewtonEq(x0,f,tol,maxiter)
% Inicialización de las variables
digits(200)
iter=1;
i1=1;
[fx0,dfx0]=feval(f,x0);
I=[]; ACOC=[];
while and(iter<maxiter,i1>tol)
      x=x0-fx0/dfx0;
     i1=abs(x-x0);
     I=[I,i1];
     [fx,dfx]=feval(f,x);
     i2=abs(fx0);
     iter=iter+1;
```

```
15        x0=x; fx0=fx; dfx0=dfx;
16 end
17 % Preparar datos de salida
18 sol=x;
19 if length(I)>2
20      ACOC=log(I(3:end)./I(2:end-1))./…
21              log(I(2:end-1)./I(1:end-2));
22 end
```

En primer lugar, se inicializan las variables con las que se va a trabajar. En este sentido, la variable `i1` va a representar la diferencia entre iterados, y la variable `i2` será el valor de la función en el iterado actual. Asimismo, la variable `I` va a almacenar los diferentes valores de `i1`, de modo que podamos calcular al final el valor de ACOC. En cuanto al método iterativo de Newton, su expresión tan solo ocupa la línea 9, de modo que este código se puede **reutilizar** para diferentes métodos iterativos.

Entremos en detalle con lo que ocurre en el bucle `while`. Destacar que, previo al acceso al bucle `while`, tenemos disponibles los valores de `fx` y `dfx`, de forma que la primera línea del bucle es el método de Newton y, por tanto, el nuevo iterado. A continuación, se van actualizando valores:

- Línea 10: la diferencia con el iterado anterior.
- Línea 11: el vector que registra todos los valores de diferencias.
- Líneas 12 y 13: la evaluación del nuevo iterado y, por tanto, el valor de `i2`.
- Línea 14: el incremento en el número de iteraciones.
- Línea 15: la actualización de los valores para volver a ejecutar el bucle.

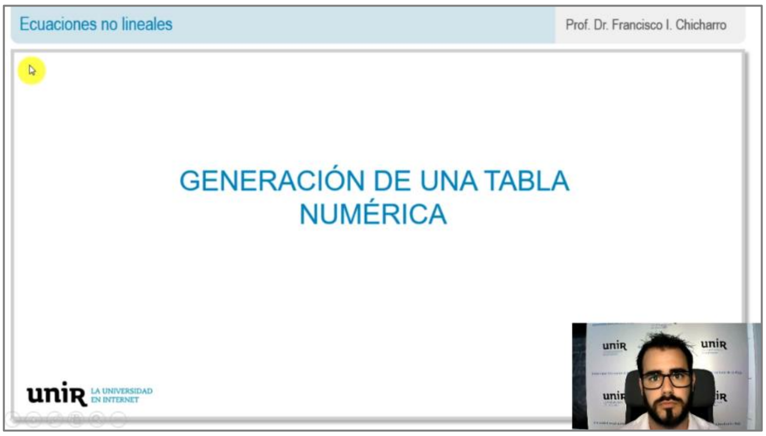
Por último, se preparan los datos de salida que no estaban definidos.

Lo + recomendado

Lecciones magistrales

Generación de una tabla numérica

En esta lección magistral se presenta cómo se puede generar una tabla numérica a partir de diferentes métodos numéricos en Matlab, con las funciones test y los puntos iniciales deseados, y exportar dichos datos para hacer una presentación en Word.



Accede a la lección magistral a través del aula virtual

No dejes de leer

Soluciones numéricas para ecuaciones no lineales

Chapra, S. C. y Canale, R. P. (2007). «Soluciones numéricas para ecuaciones no lineales». En Autores, *Métodos numéricos para ingenieros (5a. ed.)*. McGraw-Hill.

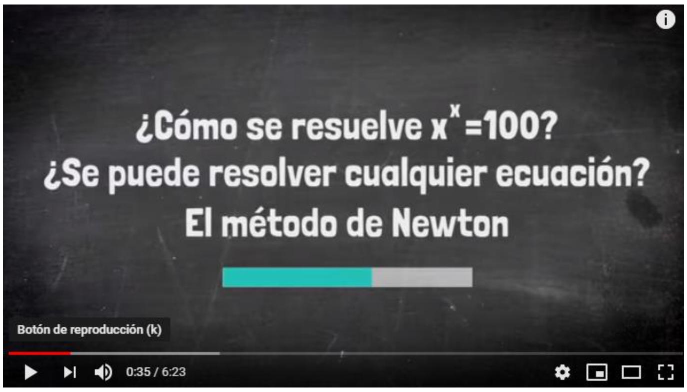
En el capítulo 6 de este libro se resuelven ecuaciones no lineales a partir de los métodos de Newton y Secante. Además, se profundiza en la obtención de estos esquemas iterativos, y se plantean ejemplos numéricos resueltos.

Accede al libro a través de la Biblioteca Virtual de UNIR

No dejes de ver

¿Cómo se resuelve x^x=100?

En este vídeo se muestra de una forma gráfica cómo se aplica de forma geométrica el método de Newton. Como problema de aplicación se plantea la resolución de la ecuación no lineal x^x=100.



Accede al vídeo a través del aula virtual o desde la siguiente dirección web:

https://www.youtube.com/watch?v=o0Pa8UzO62I

+ Información

A fondo

Advances in iterative methods for nonlinear equations

Amat, S. y Busquier, S. (2016) *Advances in iterative methods for nonlinear equations.* SEMA SIMAI Springer Series.

Los avances más recientes en la resolución de ecuaciones no lineales a través de métodos numéricos se encuentran recogidos en este libro. Editado por Amat y Busquier, contiene participaciones de los conocidos profesores Cordero, Magreñán y Torregrosa, entre otros.

Webgrafía

Gaussianos

En este enlace puedes averiguar por qué hay determinada conflictividad acerca de la autoría del método de Newton. Descubre quién era Joseph Raphson

Porque todo tiende a infinito gaussianos

Accede a la página web a través del aula virtual o desde la siguiente dirección web: https://www.gaussianos.com/la-historia-del-metodo-de-newton-raphson-y-otro-caso-mas-de-mala-documentacion-en-el-cine/

Bibliografía

Behl, R., Sarría, Í., González, R. y Magreñán, Á. (2019). Highly efficient family of iterative methods for solving nonlinear models. *Journal of Computational and Applied Mathematics, 346*, 110-132.

Chicharro, F., Cordero, A., Garrido, N., & Torregrosa, J. (2018). Stability and applicability of iterative methods with memory. *Journal of Mathematical Chemistry*, 1--19.

Cordero, A. y Torregrosa, J. R. (2007). Variants of Newton's method using fifth-order quadrature formulas. *Applied Mathematics and Computation, 190*, 686-698.

Jarratt, P. (1969). Some efficient fourth order multipoint methods for solving equations. *BIT, 9*, 119--124.

King, R. (1973). A family of fourth order methods for nonlinear equations. *SIAM Journal on Numerical Analysis, 10*, 876--879.

Kung, H. T. y Traub, J. F. (1974). Optimal order of one-point and multi-point iteration. *Applied Mathematics and Computation*, 643-651.

Ostrowski, A. (1960). *Solutions of equations and systems of equations.* New York: Academic Press.

Traub, J. (1982). *Iterative methods for the solution of equations.* New York: Chelsea Publishing Company.

Test

1. Índica cuál de los siguientes métodos es un método con memoria:

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

B. $x_{k+1} = x_k - \frac{(x_k - x_{k-1})f(x_k)}{f(x_k) - f(x_{k-1})}$

C. $x_{k+1} = x_k - \frac{f^2(x_k)}{f(x_k + f(x_k))}$

2. ¿Qué características cumple el método de Newton?

A. Es un método de un punto.

B. Es un método óptimo.

C. Todas las anteriores son correctas.

3. ¿Cuál de las siguientes no es una condición de parada de un método iterativo?

A. El valor de la estimación inicial es demasiado pequeño.

B. La distancia entre los dos últimos iterados es muy pequeña.

C. El valor de la función en el último iterado es muy pequeño.

4. Si el método iterativo requiere del conocimiento de la función $f(x)$ y $f''(x)$, ¿cuántos parámetros de salida deberá tener como mínimo la función que implementa a f?

A. Uno.

B. Dos.

C. Tres.

5. Cuando utilizamos el método gráfico, la solución está marcada por el corte de la gráfica con:

   A. El eje de abscisas.

   B. El eje de ordenadas.

   C. Ninguna de las anteriores es correcta.

6. El índice de eficiencia de un método numérico de orden 3 y 2 evaluaciones funcionales por iteración es

   A. 9

   B. 1.732051

   C. 8

7. Para que un método de orden p sea óptimo, debe tener como máximo:

   A. p evaluaciones funcionales por iteración.

   B. $1 - \log_2 p$ evaluaciones funcionales por iteración.

   C. $1 + \log_2 p$ evaluaciones funcionales por iteración.

8. ¿Qué significa la nomenclatura $a \cdot e - b$ en las tablas numéricas?

   A. $a + 10 - b$

   B. $a \cdot 10^{-b}$

   C. $a \cdot e^{-b}$

9. El método de Newton doble:

   A. Utiliza la técnica de fórmulas de cuadratura.

   B. Utiliza la técnica de composición.

   C. Utiliza la técnica de funciones peso.

10. El método de Jarratt:

   A. Utiliza la técnica de fórmulas de cuadratura.

   B. Utiliza la técnica de composición.

   C. Utiliza la técnica de funciones peso.