

Métodos Numéricos Aplicados I

---

# Ecuaciones no lineales

# Índice

Esquema. . . . .	2
Ideas clave . . . . .	3
9.1 Introducción y objetivos . . . . .	3
9.2 Soluciones de ecuaciones no lineales . . . . .	4
9.3 Introducción a los métodos iterativos . . . . .	5
9.4 Métodos iterativos para resolver ecuaciones no lineales	11
9.5 Implementación en Matlab . . . . .	19
9.6 Comparativa numérica . . . . .	25



### 9.1 Introducción y objetivos

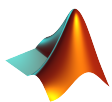
En numerosos problemas de ciencia e ingeniería se requiere resolver una ecuación no lineal  $f(x) = 0$  o un sistema de ecuaciones no lineales  $F(x) = 0$ . Generalmente, no es posible o es difícil obtener la solución de estos problemas de forma analítica, motivo por el cual utilizamos métodos iterativos para aproximar la solución.

En este tema utilizaremos distintos métodos iterativos de resolución de ecuaciones no lineales, mientras que dedicaremos el tema siguiente a la resolución de sistemas de ecuaciones no lineales. En ambos temas partiremos del conocido método de Newton y, a partir de él, iremos descubriendo y analizando otros esquemas iterativos.

A grandes rasgos, los métodos de resolución se basan en obtener una secuencia de valores, uno en cada iteración, de forma que en cada iteración nos aproximamos más a la solución. No obstante, si nuestro objetivo es conocer la solución, ¿cómo sabemos que la hemos alcanzado? En este tema analizaremos, entre otras cosas, cómo de próximos a la solución se encuentran los puntos de la secuencia sin necesidad de conocerla.

Los objetivos de este tema serán los siguientes:

- ▶ Comprender el proceso de aproximación a la solución de una ecuación no lineal.
- ▶ Clasificar los métodos iterativos en función de su estructura iterativa.
- ▶ Estudiar distintas técnicas para diseñar métodos iterativos.
- ▶ Implementar y comparar numéricamente en Matlab los esquemas iterativos más sencillos.



### Algunas funciones Matlab utilizadas en este tema

- ▶ `digits`: establece el número de dígitos decimales significativos
- ▶ `vpa`: aritmética de precisión variable

## 9.2 Soluciones de ecuaciones no lineales

El objetivo fundamental de este tema es encontrar una raíz simple, que denotaremos  $\alpha$ , de una ecuación no lineal

$$f(x) = 0, \quad f : D \subseteq \mathbb{R} \longrightarrow \mathbb{R},$$

donde  $D$  es un intervalo abierto.

La resolución de ecuaciones no lineales tiene tres posibilidades. La primera de ellas es la solución analítica. Sin embargo, no siempre es posible obtener esta solución debido a la complejidad de la ecuación o la escasez de procesos analíticos de resolución. 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 1.

Determinemos por el método gráfico la raíz de la función no lineal

$$f(x) = \cos^2(x) - x.$$

Ejecutamos en Matlab las siguientes instrucciones:

```
>> x=linspace(0,1);  
>> f=cos(x).^2-x;  
>> plot(x,f);
```

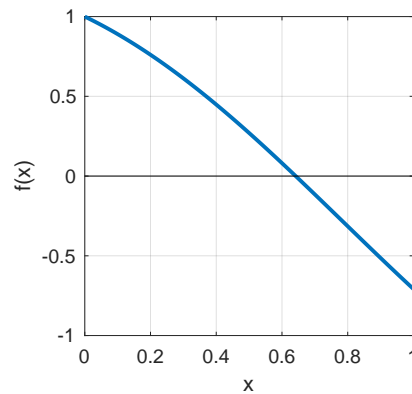


Figura 1:  $f(x) = \cos^2(x) - x$

Vemos en la Figura 1 que la gráfica de la función corta al eje de abscisas alrededor del punto  $x = 0.65$ , pero no lo podemos saber con la exactitud que pretendamos.

La tercera posibilidad es el uso de métodos iterativos. A partir de ciertas estimaciones iniciales a la raíz de la función no lineal, el método iterativo es capaz de obtener una solución tan aproximada como se requiera. Dedicaremos los siguientes apartados a la descripción y características fundamentales de estos métodos.

## 9.3 Introducción a los métodos iterativos

Recordemos que nuestro objetivo es determinar el valor de  $\alpha$ , la solución exacta de una ecuación no lineal  $f(x) = 0$ . Debido a la escasez de métodos analíticos para calcular  $\alpha$ , estaremos la solución utilizando métodos iterativos de punto fijo.

En general, un método iterativo para la resolución de ecuaciones o sistemas de ecuaciones no lineales es un procedimiento que trata de aproximarse a la solución partien-

do de un valor o una serie de valores iniciales. De esta manera, un método iterativo obtiene una secuencia de valores:

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

que, bajo ciertas condiciones de la función  $f$  y las estimaciones iniciales, converge a la solución  $\alpha$ . Para la generación de cada uno de los elementos de la secuencia, se utiliza la expresión del método iterativo. En forma general, podemos escribir la expresión del método como

$$x_{k+1} = \phi(x_k), \quad k = 0, 1, 2, \dots,$$

siendo  $\phi : \mathbb{R} \longrightarrow \mathbb{R}$  la función de punto fijo.

El método iterativo más conocido es el método de Newton, cuya expresión iterativa es

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (1)$$

Se trata de un método de punto fijo  $x_{k+1} = \phi(x_k)$  donde

$$\phi(x) = x - \frac{f(x)}{f'(x)}.$$

## Clasificación de los métodos iterativos

Para la generación de los puntos de la secuencia de aproximaciones a la solución, se utiliza la expresión del método iterativo. Si el método solo necesita un iterado para obtener el siguiente, decimos que es un **método sin memoria**, y su expresión general es

$$x_{k+1} = \phi(x_k), \quad k = 0, 1, 2, \dots$$

En el caso en que se requiera de más de una iteración anterior para obtener la siguiente, decimos que es un **método con memoria** y lo representamos como

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

en cuyo caso se necesitarían  $x_0, x_1, \dots, x_m$  estimaciones iniciales para comenzar el proceso iterativo de cálculo de puntos de la secuencia de aproximaciones.

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  $x_{k+1} = \phi(x_k)$ . En cambio, los **métodos multipaso o multipunto** para calcular la iteración  $k + 1$  utilizan evaluaciones funcionales de la iteración  $k$ -ésima y, a diferencia de los métodos de un punto, también de otros puntos intermedios. Podemos entender los esquemas multipaso como métodos predictor-corrector. Por ejemplo, un esquema de este tipo compuesto por dos pasos tiene la expresión

$$\begin{aligned} y_k &= \psi(x_k), \\ x_{k+1} &= \phi(x_k, y_k), \quad k = 0, 1, 2, \dots \end{aligned}$$

Por último, podemos clasificar los métodos en función de si utilizan o no derivadas en su expresión iterativa. Los métodos libres de derivadas son especialmente útiles cuando en la función no lineal a resolver no conocemos la expresión de sus derivadas o son difíciles de calcular.

### Ejemplo 2.

Consideremos la expresión iterativa del Método de Newton

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots$$

el método de Traub

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= y_k - \frac{f(y_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \end{aligned}$$

y el método de la secante

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_{k-1}]}, \quad k = 1, 2, \dots$$



En la Tabla 1 resumimos las características fundamentales de la estructura iterativa estos métodos dependiendo del número de puntos, las iteraciones previas utilizadas y el uso de derivadas.

	Newton	Traub	Secante
Un paso	✓		✓
Multipaso		✓	
Memoria			✓
Sin memoria	✓	✓	
Derivadas	✓	✓	
Sin derivadas			✓

Tabla 1: Clasificación de los métodos iterativos

## Conceptos básicos

Para evaluar las características de un método iterativo así como su eficiencia en la resolución de un determinado problema, se pueden utilizar diferentes parámetros. A continuación introduciremos los más básicos.

### Definición 1: Orden de convergencia

Consideremos la sucesión  $\{x_k\}_{k \geq 0}$  generada por un método iterativo que converge a  $\alpha$ . Decimos que el método correspondiente tiene orden de convergencia  $p \geq 1$  si existe una constante  $C > 0$  ( $C < 1$  si  $p = 1$ ) tal que

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

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  $\alpha$ . El método iterativo  $x_{k+1} = g(x_k)$  es de orden  $p$  si, y solo si,

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

Sin embargo, existen otros procesos para determinar el orden de convergencia de un método basados en desarrollos de Taylor y manipulaciones algebraicas. En esta línea, si denotamos

$$e_k = x_k - \alpha, \quad k = 0, 1, 2, \dots$$

al error cometido en la  $k$ -ésima iteración del método, decimos que el método iterativo tiene orden  $p$  si, y solo si, se cumple la relación

$$e_{k+1} = C e_k^p + \mathcal{O}(e_k^{p+1}),$$

denominada **ecuación del error** del método iterativo.

Podemos estudiar un ejemplo de cómo se obtiene la ecuación del error de un esquema iterativo para determinar su orden de convergencia en el Teorema 2. En éste se prueba el orden del método de Newton utilizando desarrollos de Taylor.

### Teorema 2

Consideremos  $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$  una función suficientemente diferenciable en un conjunto convexo  $D$ . Sea  $\alpha$  una raíz simple de  $f(x) = 0$ . Entonces, si la estimación inicial  $x_0$  es suficientemente próxima a  $\alpha$ , el método de Newton tiene orden de convergencia cuadrático, siendo su ecuación del error

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

donde  $c_2 = \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}$  y  $e_k = x_k - \alpha$ .

### Demostración

Consideremos el desarrollo en serie de Taylor de  $f$  en torno a  $\alpha$ :

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

siendo  $c_2 = \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}$  y  $e_k = x_k - \alpha$ . Derivando, se obtiene

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

Utilizando los desarrollos anteriores, se tiene

$$\frac{f(x_k)}{f'(x_k)} = e_k - c_2 e_k^2 + \mathcal{O}(e_k^3).$$

Por tanto, la ecuación del error es

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

y el orden de convergencia del método de Newton es  $p = 2$ . □

A nivel computacional, dada una secuencia  $\{x_k\}_{k \geq 0}$  de aproximaciones sucesivas a la solución  $\alpha$  obtenidas en el proceso iterativo, podemos estimar el orden de convergencia teórico de un método utilizando el **orden de convergencia computacional**,  $COC$ , definido como

$$COC = \frac{\ln(|x_{k+1} - \alpha|/|x_k - \alpha|)}{\ln(|x_k - \alpha|/|x_{k-1} - \alpha|)}, \quad k = 1, 2, \dots$$

Como en la práctica no conocemos el valor de  $\alpha$ , es más común utilizar el **orden de convergencia computacional aproximado**,  $ACOC$ , definido por

$$ACOC = \frac{\ln(|x_{k+1} - x_k|/|x_k - x_{k-1}|)}{\ln(|x_k - x_{k-1}|/|x_{k-1} - x_{k-2}|)}, \quad k = 2, 3, \dots$$

Para comparar diferentes métodos bajo el punto de vista del coste computacional, se

define el **índice de eficiencia**,  $I$ , como

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

donde  $p$  es el orden del método y  $d$  es el número de evaluaciones funcionales distintas de la función  $f$  cada iteración. De forma análoga, se define el **índice de eficiencia computacional**,  $IC$ , como

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

siendo  $op$  el número de productos y cocientes efectuados en cada iteración.

El interés en el desarrollo de métodos con memoria reside en la conjetura de Kung y Traub, la cual establece que el orden de convergencia de los métodos iterativos sin memoria está acotado por:

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

Cuando se alcanza esta cota, el método se denomina **óptimo**. No obstante, el orden de convergencia de los métodos con memoria no está limitado por este valor.

## 9.4 Métodos iterativos para resolver ecuaciones no lineales

En esta sección presentaremos las expresiones iterativas de distintos métodos y familias de métodos. Comenzaremos por los esquemas más simples y a continuación utilizaremos distintas técnicas para, a partir de éstos, diseñar métodos más complejos.

## Métodos de un punto

Una de las formas de deducir el método de Newton es aplicando el teorema fundamental del cálculo a la función  $f$  y aproximando la integral por el método de los rectángulos:

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

Si tomamos el punto  $x = \alpha$ , donde  $\alpha$  es la raíz de  $f$ , entonces:

$$0 \approx f(x_0) + f'(x_0)(\alpha - x_0) \quad \Leftrightarrow \quad \alpha \approx x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Una primera aproximación a la raíz es

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

A partir de la estimación inicial  $x_0$ , obtendríamos de forma sucesiva aproximaciones cada vez más próximas a  $\alpha$ :

$$x_0 \Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \Rightarrow x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \Rightarrow x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} \dots$$

Este proceso es el algoritmo iterativo que define el método de Newton.

Los métodos iterativos libres de derivadas adquieren especial relevancia cuando la función no lineal cuya raíz se pretende aproximar no es derivable o tiene una derivada difícil de calcular. En estos casos, la adaptación de un esquema con derivadas a otro libre de derivadas se realiza frecuentemente sustituyendo las derivadas por operadores de diferencias divididas. Siguiendo esta premisa, si en el método de Newton aproximamos la derivada por

$$f'(x_k) \approx \frac{f(x_k + f(x_k)) - f(x_k)}{f(x_k)},$$

se obtiene el **método de Steffensen**, con expresión iterativa

$$x_{k+1} = x_k - \frac{f(x_k)^2}{f(x_k + f(x_k)) - f(x_k)}, \quad k = 0, 1, 2, \dots$$

En cambio, si aproximamos la derivada utilizando puntos anteriores de la secuencia de iterados de la forma

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = f[x_k, x_{k-1}],$$

obtenemos un método con memoria, denominado **método de la secante**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_{k-1}]}, \quad k = 1, 2, \dots$$

Por otro lado, utilizando el grado de convexidad logarítmica

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

también podemos diseñar distintos métodos iterativos con derivadas. Presentamos a continuación algunos de ellos:

► **Método de Halley:**

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \left[ 1 + \frac{L_f(x_k)}{2 - L_f(x_k)} \right], \quad k = 0, 1, 2, \dots$$

► **Método de Chebyshev:**

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \left[ 1 + \frac{L_f(x_k)}{2} \right], \quad k = 0, 1, 2, \dots$$

► **Método Super-Halley:**

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \left[ 1 + \frac{L_f(x_k) - 2}{2(L_f(x_k) - 1)} \right], \quad k = 0, 1, 2, \dots$$

Asimismo, la **Familia de métodos Chebyshev-Halley**, con expresión iterativa

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \left[ 1 + \frac{1}{2} \frac{L_f(x_k)}{1 - \beta L_f(x_k)} \right], \quad k = 0, 1, 2, \dots,$$

incluye los métodos anteriores para determinados valores del parámetro. Para  $\beta = 0$  estamos ante el método de Chebyshev, para  $\beta = \frac{1}{2}$  tenemos el método de Halley, para  $\beta = 1$  estamos ante el método de SuperHalley y cuando  $\beta \rightarrow \infty$  nos encontramos con el método de Newton. Las características fundamentales de estos métodos se recogen en la Tabla 2.

Método	$p$	$d$	$I$	$IC$	Óptimo
Newton	2	2	$2^{1/2}$	$2^{1/(2+1)}$	✓
Halley	3	3	$3^{1/3}$	$3^{1/(3+6)}$	✗
Chebyshev	3	3	$3^{1/3}$	$3^{1/(3+5)}$	✗
Super-Halley	3	3	$3^{1/3}$	$3^{1/(3+6)}$	✗

Tabla 2: Características de distintos métodos de la familia de Chebyshev-Halley

## Métodos multipunto

Los métodos de un punto tienen una serie de limitaciones en su orden de convergencia, tal y como indican los Teoremas 3 y 4.

### 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 al menos hasta orden  $p - 1$ .

Como consecuencia, los métodos de un punto no suelen ser una buena alternativa para incrementar el orden y el índice de eficiencia. Recurrimos a los métodos multipunto para tratar de aumentar el orden de convergencia manteniendo a su vez la optimalidad del método. El Teorema 5 nos indica cómo podemos incrementar dicho orden.

### Teorema 5

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

$$g(x) = g_2(g_1(x))$$

es  $p_1 \cdot p_2$ .

Las técnicas más utilizadas para diseñar métodos multipunto son el uso de fórmulas de cuadratura, la composición de esquemas iterativos y el uso de funciones peso.

### Métodos multipunto mediante fórmulas de cuadratura

Podemos escribir la ecuación no lineal como

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

y utilizar diferentes fórmulas de cuadratura para aproximar la integral.

Si utilizamos el método de Newton como predictor y como aproximación de la integral la regla de los trapecios, se obtendría el **método de los trapecios**:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} &= x_k - \frac{2f(x_k)}{f'(y_k) + f'(x_k)}, \quad k = 0, 1, 2, \dots \end{aligned}$$



Por el contrario, si la integral se aproxima con la regla del punto medio, el esquema resultante es el **método del punto medio**, con expresión:

$$\begin{aligned}y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\x_{k+1} &= x_k - \frac{f(x_k)}{f'\left(\frac{x_k+y_k}{2}\right)}, \quad k = 0, 1, 2, \dots\end{aligned}$$

Y utilizando la regla de Simpson, se obtiene el **método de Simpson**:

$$\begin{aligned}y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\x_{k+1} &= x_k - \frac{6f(x_k)}{f'(x_k) + 4f'\left(\frac{x_k+y_k}{2}\right) + f'(y_k)}, \quad k = 0, 1, 2, \dots\end{aligned}$$

En el siguiente vídeo puedes estudiar una descripción más detallada de los métodos iterativos obtenidos utilizando fórmulas de cuadratura.



Accede al vídeo: Métodos iterativos mediante fórmulas de cuadratura.

### Métodos multipunto por composición

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:

$$\begin{aligned}y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\x_{k+1} &= y_k - \frac{f(y_k)}{f'(y_k)}, \quad k = 0, 1, 2, \dots\end{aligned}$$

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:

$$\begin{aligned}y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\z_k &= x_k - \frac{f(y_k)}{f'(y_k)}, \\x_{k+1} &= z_k - \frac{f(z_k)}{f'(z_k)}, \quad k = 0, 1, 2, \dots\end{aligned}$$

En este caso, tenemos un método de orden 8, pero tampoco es óptimo porque realiza 6 evaluaciones funcionales distintas en cada iteración.

Este procedimiento no es eficiente puesto que, aunque se incrementa el orden de convergencia, requiere de demasiadas evaluaciones funcionales por iteración. Utilizaremos una composición eficiente de métodos iterativos para reducir el número de evaluaciones distintas. Para ello, recurrimos a la técnica de congelar la derivada de  $f'(x_k)$ . En el caso de Newton doble, esta técnica consiste en mantener la misma derivada en los dos denominadores, evitando una evaluación funcional distinta. Con esta operación, obtenemos el método de Traub, también llamado método de Potra-Pták, cuya expresión iterativa es la siguiente:

$$\begin{aligned}y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\x_{k+1} &= y_k - \frac{f(y_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots\end{aligned}$$

El método de Traub tiene orden de convergencia cúbico y solo realiza 3 evaluaciones funcionales en cada iteración. Su índice de eficiencia es mayor que el del esquema Newton doble, pero todavía no es óptimo.

### Métodos multipunto mediante funciones peso

El procedimiento de las funciones peso consiste incorporar en la estructura iterativa, multiplicando de forma conveniente, una función peso real del tipo  $H(\mu)$ , donde  $\mu$  es una relación entre la función evaluada en diferentes puntos.

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

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - H(\mu_k) \frac{f(y_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots, \end{aligned} \quad (2)$$

siendo la variable de la función peso  $\mu = \frac{f(y)}{f(x)}$ .

### Teorema 6

Sea  $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$  una función suficientemente diferenciable en un conjunto convexo  $D$  y  $\alpha \in D$  una raíz simple de  $f(x) = 0$ . Consideremos una estimación inicial  $x_0$  próxima a  $\alpha$  y una función peso  $H$  tal que  $H(0) = 1$ ,  $H'(0) = 2$  y  $|H''(0)| < \infty$ . Entonces, (2) tiene orden de convergencia 4 y siendo su ecuación del error es

$$e_{k+1} = \left[ \left( 5 - \frac{H''(0)}{2} \right) c_2^3 - c_2 c_3 \right] e_k^4 + \mathcal{O}(e_k^5),$$

siendo  $c_j = \frac{1}{j!} \frac{f^{(j)}(\alpha)}{f'(\alpha)}$ ,  $j \geq 2$ , y  $e_k = x_k - \alpha$ .

Con las condiciones del Teorema 6, si consideramos la función peso

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

y la reemplazamos en (2), se obtiene la **familia de King**:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(x_k) + \beta f(y_k)}{f(x_k) - (\beta - 2)f(y_k)} \frac{f(y_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \end{aligned}$$

Se trata de una familia de métodos óptimos de orden cuatro para cualquier valor de  $\beta$ . Concretamente, tomando  $\beta = 0$  se obtiene el **método de Ostrowski**, con expresión iterativa:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= y_k - \frac{f(x_k)}{f(x_k) - 2f(y_k)} \frac{f(y_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots, \end{aligned}$$

Por último, presentamos el **método de Jarratt**. Este método también se obtiene utilizando la técnica de composición y funciones peso. No obstante, difiere en algunos aspectos respecto a la familia de King. Su expresión iterativa es la siguiente:

$$y_k = x_k - \frac{2}{3} \frac{f(x_k)}{f'(x_k)}$$

$$x_{k+1} = x_k - \frac{1}{2} \left( \frac{3f'(y_k) + f'(x_k)}{3f'(y_k) - f'(x_k)} \right) \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots,$$

Puedes consultar el siguiente vídeo para una descripción más detallada del método de Jarratt.



Accede al vídeo: Diseño del método de Jarratt

## 9.5 Implementación en Matlab

Realizar manualmente las operaciones que conlleva un método iterativo puede resultar muy costoso manualmente, de modo que implementaremos los algoritmos iterativos en Matlab.

En primer lugar, para resolver una ecuación no lineal  $f(x) = 0$ , un aspecto que debemos considerar es cómo introducimos la ecuación 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.

### Ejemplo 3.

Escribamos un archivo .m de Matlab para la ecuación no lineal

$$\sin(x) = e^{-x}$$

suponiendo que el método iterativo que la resuelve requiere del valor de la función y sus dos primeras derivadas.

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

$$f(x) = \sin(x) - e^{-x}, \quad f'(x) = \cos(x) + e^{-x}, \quad f''(x) = -\sin(x) - e^{-x}.$$

Por tanto, definimos la siguiente función:

```
function [f,df,d2f] = ejemplo2(x)
    f=sin(x)-exp(-x);
    df=cos(x)+exp(-x);
    d2f=-sin(x)-exp(-x);
end
```

Representando las funciones que hay a cada lado de la igualdad, podemos determinar gráficamente la solución de la ecuación no lineal:

```
>> x=linspace(-1,1);
>> y1=sin(x);
>> y2=exp(-x);
>> plot(x,[y1;y2]);
>> grid on; xlabel('x'); legend('sin(x)', 'e^{-x}')
```

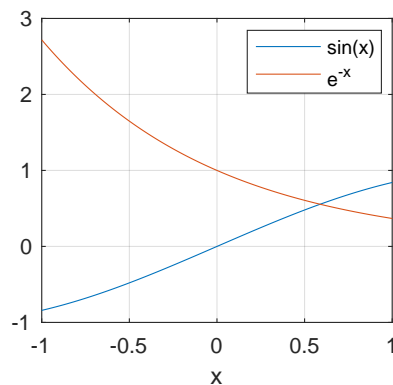


Figura 2:  $\sin(x) = e^{-x}$

En la Figura 2 se observa que la solución de la ecuación no lineal es aproximadamente  $x \approx 0.6$ .

Un método iterativo es un algoritmo que realiza una serie de operaciones de forma sucesiva, por lo que 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. Normalmente, estas condiciones son alguna de las siguientes o una combinación entre ellas:

- Los dos últimos iterados están muy próximos:

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

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

$$|f(x_{k+1})| < \epsilon$$

- Se han realizado muchas iteraciones y no se ha alcanzado la solución.

El valor  $\epsilon$  que determina la proximidad entre puntos consecutivos o la proximidad a la solución se denomina tolerancia. Previamente a la ejecución de un método iterativo

deberemos fijar una tolerancia para determinar la exactitud de la solución obtenida y, en caso contrario, un máximo de iteraciones para garantizar que el proceso sea finito.

Matlab trabaja con una precisión por defecto. Sin embargo, existen determinadas aplicaciones 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 código el comando `digits(200)`.

Asimismo, es necesario indicar 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. Por ejemplo, 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)`.

En general, la implementación en Matlab de un método iterativo sigue los siguientes pasos:

- ▶ Entrada: función no lineal  $f$ , estimación inicial  $x_0$ , tolerancia (`tol`), máximo número de iteraciones (`maxiter`)
- ▶ Proceso:
  - Inicialización del contador de iteraciones: `iter=1`
  - Evaluación de la función y sus derivadas en  $x_0$
  - `CiterioParada=0`
  - Mientras `CiterioParada=0`
    - Cálculo de  $x_1$  aplicando la expresión del método:  $x_1 = \phi(x_0)$
    - Actualizar `CiterioParada`
    - Actualización de la estimación inicial:  $x_0 = x_1$
    - Actualización de valores

- `iter=iter+1`
- Cálculo de *ACOC*

► Salida: solución, iteraciones (*iter*), *ACOC*

Como ejemplo, mostramos a continuación la función que implementa el método de Newton:



```
function [sol,iter,ACOC] = Newton(f,x0,tol,maxiter)
digits(200);
% Inicializacion de las variables
iter=1;
incre1=tol+1;
incre2=tol+1;
[fx,dfx]=feval(f,x0);

% Criterio de parada
while incre1>tol && incre2>tol && iter<maxiter
    % Expresion del metodo de Newton
    x=x0-fx/dfx;

    incre1=abs(x-x0);
    I(iter)=incre1;

    % Actualizacion de la estimacion inicial
    x0=x;
    [fx,dfx]=feval(f,x0);
    incre2=abs(fx);

    % Incremento del contador de iteraciones
    iter=iter+1;
```



```

end
if length(I)>2
    sol=x;
    ACOC=log(I(3:end)./I(2:end-1))...
        ./log(I(2:end-1)./I(1:end-2));
else
    disp('necesito mas iteraciones')
end
end

```

En la función Newton.m, el criterio de parada está formado por el cumplimiento de una de las siguientes tres condiciones:  $|x_{k+1} - x_k| < tol$ ,  $|f(x_{k+1})| < tol$  o bien  $iter < maxiter$ .

#### Ejemplo 4.

Aplica el método de Newton para resolver la ecuación no lineal  $x = \cos^2(x)$  tomando como estimación inicial  $x_0 = 0.3$ . Con un máximo de 20 iteraciones, utiliza como criterio de parada  $|x_{k+1} - x_k| < 10^{-9}$ . Indica para cada iteración el valor del iterado, el valor de la función, la diferencia entre los dos últimos iterados y el ACOC.

En primer lugar, definiríamos la función no lineal teniendo en cuenta que el método de Newton requiere del valor de la función y de su derivada:

```

function [f,df] = ejemplo4(x)
f=(cos(x)).^2-x;
df=2*cos(x).*(-sin(x))-1;
end

```

En el código de la función Newton.m tendríamos que modificar la línea de inicio del bucle iterativo por:

```

incre=incre+tol;
while incre>tol && iter<maxiter
    ...
    incre=abs(x-x0);
    ...
end

```

Para aplicar el método de Newton, ejecutaríamos la instrucción:

```

>> [sol,iter,ACOC] = ...
    Newton('ejemplo4',vpa(0.3),1e-9,20);

```

Los resultados de cada iteración se muestran en la Tabla 3.

iter	$x_k$	$ f(x_{k+1}) $	$ x_{k+1} - x_k $	ACOC
1	0.691570	0.098293	0.39157	–
2	0.641989	0.00053803	0.049581	–
3	0.641714	2.1349e–8	0.00027463	2.5143
4	0.641714	3.3663e–17	1.0898e–8	1.9505
5	0.641714	8.3691e–35	1.7184e–17	1.9999

Tabla 3: Resultados numéricos del método de Newton

## 9.6 Comparativa numérica

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

A continuación, vamos a realizar una comparativa desde el punto de vista práctico de diferentes métodos numéricos de los que hemos visto a lo largo del tema. Con este fin, aproximaremos la solución de diversas funciones de prueba y, utilizando las mismas condiciones en todos los métodos, analizaremos los resultados obtenidos. En todos los casos utilizaremos como criterio de parada un máximo de 60 iteraciones o

$$|x_{k+1} - x_k| < 10^{-100}.$$

Asimismo, se ha utilizado en todos los casos aritmética de precisión variable con 400 dígitos. Tomando distintas estimaciones iniciales, las funciones test que vamos a analizar, así como las soluciones aproximadas obtenidas en cada caso, son las siguientes:

- ▶  $f_1(x) = \sin(x) - e^{-x}$ ,  $\alpha \approx 0.58853274$ .
- ▶  $f_2(x) = \cos^2(x) - x$ ,  $\alpha \approx 0.64171437$ .
- ▶  $f_3(x) = (x - 1)^3 - 1$ ,  $\alpha \approx 2$ .

Se han comparado numéricamente los métodos de Newton, Halley, Ostrowski, Traub, Punto medio, Jarratt y Newton doble. En las Tablas 4, 5 y 6 se muestran los resultados obtenidos para las funciones test  $f_1(x)$ ,  $f_2(x)$  y  $f_3(x)$ , respectivamente. Para cada método se muestra el número de iteraciones necesario para aproximar  $\alpha$  con la tolerancia fijada, el valor de la función en la última iteración, la diferencia entre las dos últimas iteraciones y el último término del ACOC.

En primer lugar, observamos que en todos los resultados numéricos se alcanza la solución de la correspondiente ecuación no lineal. Asimismo, el orden de convergencia computacional aproximado obtenido en todos los métodos es exactamente el que se obtendría de forma teórica, siendo siempre superior en los métodos de Ostrowski, Jarratt y Newton doble.

En la Tabla 4 podemos observar los resultados obtenidos para  $f_1(x)$ , tomando como estimación inicial  $x_0 = 0.1$ . Observamos que el método de Halley es el que más se aproxima a la solución. En cambio, los esquemas con orden de convergencia de cuatro son los que requieren de menos iteraciones logrando también buenas aproximaciones a  $\alpha$ .

Método	iter	$ f(x_{k+1}) $	$ x_{k+1} - x_k $	ACOC
Newton	8	6.5531e-205	1.0865e-102	2.0000
Halley	6	0	5.3661e-187	3.0000
Ostrowski	5	1.2716e-408	6.7766e-199	4.0000
Traub	6	1.2716e-408	9.3924e-166	3.0000
Punto medio	6	1.2716e-408	2.9422e-192	3.0000
Jarratt	5	1.2716e-408	5.1327e-198	4.0000
Newton doble	5	1.2716e-408	4.7250e-205	4.0000

Tabla 4: Resultados numéricos para  $f_1(x)$  con  $x_0 = 0.1$

En la Tabla 5 se muestran las conclusiones numéricas para  $f_2(x)$  con  $x_0 = 0.3$ . Los resultados son similares a los obtenidos en la Tabla 4, siendo el método de Newton el que requiere de más iteraciones. En este ejemplo, las clases que aproximan la solución de forma más precisa son Ostrowski, Traub, punto medio y Jarratt.

Método	iter	$ f(x_{k+1}) $	$ x_{k+1} - x_k $	ACOC
Newton	8	2.8844e-281	1.0088e-140	2.0000
Halley	6	3.8934e-208	5.5816e-162	3.0000
Ostrowski	5	0	5.4889e-197	4.0000
Traub	6	0	1.8990e-207	3.0000
Punto medio	6	0	3.2504e-209	3.0000
Jarratt	5	0	2.8079e-200	4.0000
Newton doble	5	2.2251e-308	1.4724e-281	4.0000

Tabla 5: Resultados numéricos para  $f_2(x)$  con  $x_0 = 0.3$

Por último, en la Tabla 6 se comparan los resultados numéricos correspondientes a la función test  $f_3(x)$  para el punto inicial  $x_0 = 1.5$ . Destacamos que para esta función el esquema de Traub, aunque converge a la solución de la ecuación, no funciona de forma eficiente porque requiere de 58 iteraciones. La diferencia en el número de iteraciones de los métodos de Newton y Traub con respecto a las demás clases es notable para este caso. Cabe destacar el buen funcionamiento de los métodos de Ostrowski y Jarratt que con el menor número de iteraciones logran las mejores aproximaciones a la solución.

Método	iter	$ f(x_{k+1}) $	$ x_{k+1} - x_k $	ACOC
Newton	11	2.8174e-359	3.0646e-180	2.0000
Halley	7	0	1.7850e-214	3.0000
Ostrowski	6	0	7.3471e-239	4.0000
Traub	58	1.2798e-393	5.9750e-132	3.0000
Punto medio	7	2.1994e-399	9.2824e-134	3.0000
Jarratt	6	0	7.3471e-239	4.0000
Newton doble	6	0	3.0646e-180	4.0000

Tabla 6: Resultados numéricos para  $f_3(x)$  con  $x_0 = 1.5$

Es conveniente remarcar que las conclusiones que se obtienen de los resultados numéricos de las Tablas 4, 5 y 6 son solo aplicables sobre cada 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.