Métodos Numéricos Aplicados I

Sistemas de ecuaciones no lineales

Índice

Esquema 2

Ideas clave 3

10.1 Introducción y objetivos 3

10.2 Conceptos previos 4

10.3 Métodos iterativos para sistemas no lineales 8

10.4 Implementación en Matlab 16

10.5 Comparativa numérica 20

Esquema

SISTEMAS DE ECUACIONES NO LINEALES

$$F(x) = 0, \qquad F: D \subseteq \mathbb{R}^n \to \mathbb{R}^n$$

MÉTODOS ITERATIVOS

$$x^{(k+1)} = G\left(x^{(k+1)}\right), \qquad k = 0,1,2,\ldots$$

- DISEÑO DE MÉTODOS
  - Adaptación de métodos escalares
  - Diseño de métodos iterativos multidimensionales
- EFICIENCIA
  - **Índice de eficiencia**
  - **Índice de eficiencia computacional**
- IMPLEMENTACIÓN NUMÉRICA
  - Tolerancia
  - Iteraciones
  - ACOC

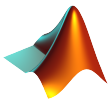
Ideas clave

10.1 Introducción y objetivos

En el tema anterior hemos estudiado algoritmos iterativos para obtener raíces de funciones reales no lineales. Sin embargo, la resolución de ecuaciones no lineales supone una pequeña aproximación a la resolución de problemas no lineales, ya que la resolución de sistemas de ecuaciones no lineales permite resolver problemas de mayores dimensiones. Existen numerosas aplicaciones en ingeniería que se describen a partir de sistemas de ecuaciones diferenciales acopladas, de ecuaciones en derivadas parciales o de problemas de frontera. En cualquier caso, tanto de forma inmediata como a partir de una discretización o transformación del problema, se puede utilizar el cálculo numérico para obtener las soluciones aproximadas como ocurría en el caso de las ecuaciones no lineales.

Retomaremos la mayoría de los términos y conceptos estudiados en el caso escalar, aunque algunos de ellos deberán adaptarse a su versión multidimensional, como son los cocientes, las distancias, u otros que iremos viendo a lo largo de este tema. Por tanto, los objetivos del tema serán:

- Estudiar el diseño de nuevos métodos iterativos para resolver sistemas de ecuaciones no lineales.
- Adaptar la estructura iterativa y la implementación en Matlab de métodos para problemas escalares a problemas multidimensionales.
- Analizar la eficiencia y el coste computacional efectuado a cada iteración del algoritmo en función de las dimensiones del sistema.
- Comparar numéricamente los esquemas iterativos.



Algunas funciones Matlab utilizadas en este tema

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

10.2 Conceptos previos

Partimos de la expresión general de un sistema de n ecuaciones no lineales con n incógnitas

$$\left.\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}\right\}$$

que podemos denotar de la forma

$$F(X) = 0,$$

donde $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ es una función vectorial definida en un conjunto abierto y convexo D y $f_i : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$, $i = 1, 2, \ldots, n$, son las funciones coordenadas. El problema que tenemos que resolver consiste en obtener una solución $\alpha \in \mathbb{R}^n$ del sistema de ecuaciones $F(X) = 0$.

Análogamente a la resolución de ecuaciones no lineales, para aproximar la solución α en el caso multidimensional, en general se utilizan métodos iterativos de punto fijo, descritos por una función $G : \mathbb{R}^n \longrightarrow \mathbb{R}^n$:

$$x^{(k+1)} = G(x^{(k)}), \qquad k = 0, 1, 2, \ldots \tag{1}$$

El proceso consiste en partir de una aproximación inicial $x^{(0)} \in \mathbb{R}^n$ suficientemente próxima a α y generar mediante la función de punto fijo una secuencia de iteraciones sucesivas

$$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\}$$

que se van aproximando a α. Este proceso iterativo es convergente si $\{x^{(k)}\} \to \alpha$ cuando $k \to +\infty$. Decimos que el algoritmo (1) es un método iterativo de punto fijo para resolver sistemas de ecuaciones no lineales.

Relacionado con los métodos iterativos para la resolución de sistemas de ecuaciones no lineales, volvemos a tener una serie de parámetros que caracterizan a los métodos y permiten evaluar su eficiencia en la resolución de un determinado problema.

Definición 1: Orden de convergencia

Dada la secuencia $\{x^{(k)}\}_{k\geq 0}$ generada por un método iterativo que converge a α, decimos que el método tiene orden de convergencia $p \geq 1$ si existen $M > 0$ y $k_0 > 0$ tales que

$$||x^{(k+1)} - \alpha|| \leq M||x^{(k)} - \alpha||^p, \qquad \forall k \geq k_0.$$

Hay diferentes alternativas para demostrar el orden de convergencia de un método iterativo para resolver sistemas de ecuaciones no lineales. Por un lado, podemos utilizar derivadas parciales en la solución, a raíz del Teorema 1.

Teorema 1

Sea $G(X)$ una función de punto fijo con derivadas parciales continuas hasta orden p. El esquema iterativo $x^{(k+1)} = G(x^{(k)})$ tiene orden de convergencia p si

- $G(\alpha) = \alpha$.
- $\dfrac{\partial^k g_i(\alpha)}{\partial x_{j_1} \partial x_{j_2} \cdots \partial x_{j_k}} = 0, \quad \forall 1 \leq k \leq p-1, \quad 1 \leq i, j_1, j_2, \ldots, j_k \leq n.$

- $\dfrac{\partial^p g_i(\alpha)}{\partial x_{j_1}\partial x_{j_2}\cdots\partial x_{j_p}} \neq 0$, para algún $i, j_1, j_2, \ldots, j_p$.

Por otro lado, podemos utilizar desarrollos de Taylor en varias variables en torno a α, donde aparecen los términos del error cometido en cada iteración, definidos como

$$e^{(k)} = x^{(k)} - \alpha, \qquad k = 0, 1, 2, \ldots$$

Decimos que el método tiene orden p si tras realizar los desarrollos de Taylor conforme a la expresión iterativa del método, su ecuación del error es de la forma:

$$e^{(k+1)} = Le^{(k)^p} + \mathcal{O}(e^{(k)^{p+1}}),$$

donde L es una función p-lineal y $e^{(k)^p} = (e^{(k)}, e^{(k)}, \ldots, e^{(k)})$. En los desarrollos de Taylor de varias variables, derivamos de forma sucesiva la función no lineal que describe el sistema $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, de modo que, por tratarse de un problema multidimensional, la derivada de orden $q \geq 1$ de F en $u \in \mathbb{R}^n$ es una función q-lineal

$$F^{(q)}(u) : \mathbb{R}^n \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

que satisface las propiedades:

- $F^{(q)}(u)(v_1, v_2, \ldots, v_{q-1}, \cdot) \in \mathcal{L}(\mathbb{R}^n)$.
- $F^{(q)}(u)(v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(q)}) = F^{(q)}(u)(v_1, v_2, \ldots, v_q)$, para toda permutación σ de los índices.

Sin la necesidad de conocer el valor de α, podemos estimar de forma aproximada el orden de un método para la resolución un problema por medio del orden de convergencia computacional aproximado, $ACOC$, definido como

$$ACOC = \frac{\ln\left(||x^{(k+1)} - x^{(k)}||/||x^{(k)} - x^{(k-1)}||\right)}{\ln\left(||x^{(k)} - x^{(k-1)}||/||x^{(k-1)} - x^{(k-2)}||\right)}, \qquad k = 2, 3, \ldots$$

Con la misma definición que en el caso de ecuaciones no lineales, la eficiencia de los

métodos se puede comparar por medio del índice de eficiencia $I = p^{1/d}$ y el índice de eficiencia computacional $IC = p^{1((d+op))}$, donde p es el orden del método y d y op son, respectivamente, el número de evaluaciones funcionales y el número de productos/cocientes por iteración.

Para comparar la eficiencia de los métodos en la resolución de un sistema $F(X) = 0$ donde $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, deberemos tener en cuenta que, por tratarse de un problema multidimensional, en cada evaluación de la función vectorial F se realizan n evaluaciones de funciones escalares, mientras que por cada matriz Jacobiana F' se realizan n^2 evaluaciones funcionales. Asimismo, a lo largo del tema veremos que los métodos iterativos para resolver este tipo de problemas, por su estructura iterativa, requieren de la resolución de sistemas lineales a cada iteración. Tendremos en cuenta que en la solución directa de un sistema lineal de dimensión $n \times n$ se realizan un total de $\frac{n^3}{3} + n^2 - \frac{n}{3}$ productos/cocientes, mientras que la resolución de q sistemas lineales con la misma matriz de coeficientes requiere de $\frac{n^3}{3} + qn^2 - \frac{n}{3}$ productos/cocientes.

En métodos iterativos para problemas multidimensionales, la conjetura de Kung y Traub ($p \leq 2^{d-1}$) no es válida. En su lugar, la conjetura establecida en un método iterativo para resolver sistemas de ecuaciones lineales es que su orden de convergencia es como máximo

$$p \leq 2^{k_1+k_2-1}, \qquad k_1 \leq k_2,$$

donde $d = k_1 + k_2$ es el número de evaluaciones funcionales requeridas a cada iteración, siendo k_1 el número de evaluaciones funcionales de la matriz Jacobiana y k_2 de la función F. Diremos que el esquema iterativo es óptimo cuando se alcance este máximo valor. No obstante, esta conjetura es muy restrictiva y, de los métodos conocidos, solo el método de Newton es óptimo.

10.3 Métodos iterativos para sistemas no lineales

En este apartado mostraremos diversos esquemas iterativos para resolver sistemas no lineales. Comenzaremos estudiando la posibilidad de adaptar algunos de los métodos para problemas escalares al caso multidimensional. Debido a que el conjunto de métodos que permiten su adaptación directa es reducido, estudiaremos también nuevos esquemas iterativos que pueden utilzarse para la resolución de cualquier problema no lineal.

Adaptación de métodos utilizados en ecuaciones no lineales

En este apartado vamos a comprobar si podemos reutilizar alguno de los métodos del tema anterior. Comenzaremos por el método de Newton, cuya expresión iterativa para ecuaciones no lineales recordemos que es

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

Teniendo en cuenta que matricialmente no podemos evaluar cocientes, deberemos reemplazarlos por sus matrices inversas. También sustituiremos la notación de los iterados x_k para los escalares por la notación vectorial $x^{(k)}$. Con estas consideraciones, la expresión iterativa del **método de Newton** para sistemas de ecuaciones no lineales se escribe como:

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}), \qquad k = 0, 1, 2, \dots,$$

donde $F'(x^{(k)})$ representa la matriz Jacobiana de F. Por tanto, la adaptación del método de Newton a problemas multidimensionales es directa.

Ejemplo 1. Eficiencia del método de Newton

A cada iteración del método de Newton se realiza una única evaluación de la función vectorial en $x^{(k)}$ y una de la matriz Jacobiana también en $x^{(k)}$. Suponiendo que el sistema no lineal que se resuelve con este método es de dimensión n, el número de evaluaciones funcionales total es:

$$d = n^2 + n.$$

Por otro lado, la instrucción $[F'(x^{(k)})]^{-1}F(x^{(k)})$ no se realiza calculando la inversa de la matriz Jacobiana y multiplicando por el vector $F(x^{(k)})$. En su lugar, se resuelve el siguiente sistema lineal:

$$[F'(x^{(k)})]u = F(x^{(k)}),$$

de modo que la estructura iterativa de Newton quedaría expresada como

$$x^{(k+1)} = x^{(k)} - u, \qquad k = 0, 1, 2, \ldots$$

Sabemos que la resolución directa de un sistema lineal requiere de $\frac{n^3}{3} + n^2 - \frac{n}{3}$ productos/cocientes. Como en el método de Newton únicamente se resuelve un sistema lineal a cada iteración, entonces:

$$op = \frac{n^3}{3} + n^2 - \frac{n}{3}.$$

Siendo el orden de convergencia de Newton cuadrático, se tiene:

$$I = 2^{1/(n^2+n)}, \qquad IC = 2^{1/\left(\frac{n^3}{3}+2n^2+\frac{2n}{3}\right)}.$$

La adaptación también es directa con los métodos escalares diseñados mediante fórmulas de cuadratura. Por ejemplo, podemos expresar el método de trapecios para

ecuaciones no lineales

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)},$$
$$x_{k+1} = x_k - \frac{2f(x_k)}{f'(y_k) + f'(x_k)}, \qquad k = 0, 1, 2, \ldots$$

con la notación vectorial onteniendo el **método de trapecios** para la resolución de problemas multidimensionales como

$$y^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}),$$
$$x^{(k+1)} = x^{(k)} - 2[F'(y^{(k)}) + F'(x^{(k)})]^{-1} F(x^{(k)}), \qquad k = 0, 1, 2, \ldots$$

De la misma forma, tendríamos la expresión del método de **punto medio** para resolver sistemas de ecuaciones no lineales como

$$y^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}),$$
$$x^{(k+1)} = x^{(k)} - \left[F'\left(\frac{x^{(k)} + y^{(k)}}{2}\right)\right]^{-1} F(x^{(k)}), \qquad k = 0, 1, 2, \ldots$$

y del **método de Simpson**:

$$y^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}),$$
$$x^{(k+1)} = x^{(k)} - 6\left[F'(x^{(k)}) + 4F'\left(\frac{x^{(k)} + y^{(k)}}{2}\right) + F'(y^{(k)})\right]^{-1} F(x^{(k)}), \quad k = 0, 1, 2, \ldots$$

Diseño de métodos para sistemas de ecuaciones no lineales

Existen diferentes técnicas para la generación de métodos iterativos de resolución de sistemas de ecuaciones no lineales, entre las cuales destacamos las siguientes:

- Fórmulas de cuadratura.
- Composición de métodos iterativos.
- Polinomios de Adomian.

- Pseudocomposición.
- Funciones peso matriciales.
- Diferencias divididas multidimensionales.

Como ya hemos visto, algunos de los métodos para sistemas se pueden diseñar como extensión del caso escalar utilizando estas técnicas. No obstante, nos centraremos en lo que sigue en el diseño de una serie de métodos numéricos a partir de la técnica de composición con el método de Newton.

En primer lugar, partimos de un método iterativo de orden p con expresión iterativa general

$$z^{(k)} = \Phi(x^{(k)}, y^{(k)}), \tag{2}$$

donde $x^{(k)}$ representa la iteración actual e $y^{(k)}$ es un paso intermedio de dicha iteración. Si realizamos la composición del esquema (2) con el método de Newton se obtiene un método iterativo cuyo orden de convergencia es el producto del orden de las dos clases. Por tanto, el esquema tiene orden $2p$ y expresión iterativa:

$$\begin{aligned} z^{(k)} &= \Phi(x^{(k)}, y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - [F'(z^{(k)})]^{-1} F(z^{(k)}). \end{aligned}$$

El mayor problema que presenta esta composición es la evaluación de una matriz Jacobiana nueva, aumentando de forma considerable el coste computacional. Si por el contrario, realizamos esta composición pero mantenemos la misma matriz Jacobiana en ambos pasos, obtendremos el siguiente esquema iterativo:

$$\begin{aligned} z^{(k)} &= \Phi(x^{(k)}, y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - [F'(x^{(k)})]^{-1} F(z^{(k)}) \end{aligned} \tag{3}$$

Teorema 2

Sea $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ una función suficientemente diferenciable en un entorno abierto D de la solución α del sistema no lineal $F(X) = 0$. Supongamos que $F'(X)$ es continua y no singular en α. Si $z^{(k)} = \Phi(x^{(k)}, y^{(k)})$ es un método de orden p, entonces la sucesión de iterados $\{ x^{(k)}\}_{k\geq 0}$ generada por el esquema iterativo (3) converge a α con orden de convergencia $p + 1$.

Ejemplo 2. Método de Traub

Consideremos al expresión iterativa del método de Traub para resolver sistemas de ecuaciones no lineales:

$$z^{(k)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}),$$
$$x^{(k+1)} = z^{(k)} - [F'(x^{(k)})]^{-1}F(z^{(k)}), \qquad k = 0, 1, 2, \ldots$$

El primer paso del esquema es el método de Newton, con orden de convergencia cuadrático, mientras que el segundo paso se obtiene como composición del método de Newton "congelando"la derivada del denominador. El método de Traub sigue la estructura iterativa de (3). Por el Teorema 2, su orden de convergencia es $p + 1 = 3$.

Con la idea de congelar la derivada y la técnica de la composición, se plantea el siguiente método de un número de pasos indeterminado:

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} \left(\sum_{j=1}^{m} b_j F(\eta_j(x^{(k)})) \right), \qquad b_j \in \mathbb{R},$$

siendo

$$\eta_j(x^{(k)}) = x^{(k)} - a_j[F'(x^{(k)})]^{-1}F(x^{(k)}), \qquad a_j \in \mathbb{R}.$$

Al método de dos pasos se le denomina **método Golden Ratio** y tiene la expresión:

$$y^{(k)} = x^{(k)} - a[F'(x^{(k)})]^{-1}F(x^{(k)}),$$
$$x^{(k+1)} = x^{(k)} - b[F'(x^{(k)})]^{-1}F(y^{(k)}), \qquad k = 0, 1, 2, \ldots$$

donde $a = \frac{-1\pm\sqrt{5}}{2}$ y $b = \frac{3\pm\sqrt{5}}{2}$. Se trata de un método de orden 3. Si realizamos la composición de este método con Newton, obtenemos un método de orden 4, denotado **método NA**, cuya expresión iterativa es:

$$\begin{aligned} y^{(k)} &= x^{(k)} - a[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - b[F'(x^{(k)})]^{-1}F(y^{(k)}), \\ x^{(k+1)} &= z^{k} - [F'(x^{(k)})]^{-1}F(z^{(k)}), \qquad k = 0, 1, 2, \ldots, \end{aligned}$$

Notemos que en el método NA solo se realiza una evaluación de una matriz Jacobiana.

Si comparamos en términos de índice de eficiencia los métodos de Newton (N), Golden Ratio (GR) y NA, se obtiene:

$$I_N = 2^{1/(n^2+n)}, \qquad I_{GR} = 3^{1/(n^2+2n)}, \qquad I_{NA} = 4^{1/(n^2+3n)},$$

siendo n la dimensión del sistema no lineal a resolver. Además, se satisface

$$I_{NA} > I_{GR} > I_N, \qquad \forall n > 1.$$

Un método iterativo no solo tiene que tener un buen valor de orden de convergencia, sino que también es necesario que sea computacionalmente eficiente. En este sentido, el número de operaciones requeridas a cada iteración, en especial cuando las dimensiones del sistema aumentan, es un factor determinante de su eficiencia. Para comparar la eficiencia computacional de los métodos de Newton, Golden Ratio y NA, hemos calculado su índice de eficiencia computacional, obteniendo los siguientes valores en función de la dimensión del sistema n:

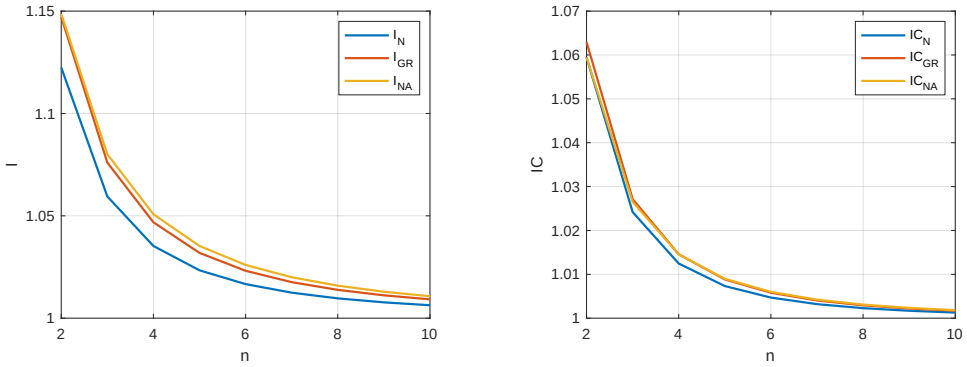
$$IC_N = 2^{1/\left(\frac{n^3}{3}+2n^2+\frac{2n}{3}\right)},$$

$$IC_{GR} = 3^{1/\left(\frac{n^3}{3}+3n^2+\frac{5n}{3}\right)},$$

$$IC_{NA} = 4^{1/\left(\frac{n^3}{3}+4n^2+\frac{8n}{3}\right)},$$

En la Figura 1 podemos comparar ambos índices para cada método para valores concretos de $n \in \{2, 3, \ldots, 10\}$. Se observa cómo se reduce la eficiencia de los métodos

cuando el tamaño del sistema aumenta, tendiendo a 1 en todos los casos. De nuevo, el esquema que tiene un índice de eficiencia computacional más alto es el método NA.



(a) I (b) IC

Figura 1: Índices de eficiencia y eficiencia computacional para $n \in \{2, 3, \ldots, 10\}$

Para una descripción más detallada sobre el número de evaluaciones funcionales y el número de operaciones (productos/cocientes) realizados en cada iteración de los métodos anteriores, puedes consultar el siguiente vídeo.

Accede al vídeo: Cálculo del índice de eficiencia computacional de métodos iterativos para resolver sistemas de ecuaciones no lineales.

Vamos a mostrar un último método, partiendo del **método de Jarratt** para ecuaciones. Recordemos que su expresión iterativa era:

$$\begin{aligned} y_k &= x_k - \frac{2}{3}\frac{f(x_k)}{f'(x_k)} \\ x_{k+1} &= x_k - \frac{1}{2}\left(\frac{3f'(y_k) + f'(x_k)}{3f'(y_k) - f'(x_k)}\right)\frac{f(x_k)}{f'(x_k)}, \qquad k = 0, 1, 2, \ldots, \end{aligned}$$

Al realizar la extensión a la resolución de sistemas no lineales, tenemos la expresión:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \qquad k = 0, 1, 2, \ldots, \\ x^{(k+1)} &= x^{(k)} - \frac{1}{2}[3F'(y^{(k)}) - F'(x^{(k)})]^{-1}\left(3F'(y^{(k)}) + F'(x^{(k)})\right)[F'(x^{(k)})]^{-1}F(x^{(k)}). \end{aligned}$$

El método de Jarratt, aunque tenga orden 4, implica la evaluación de dos matrices Jacobianas y la resolución de dos sistemas lineales con distintas matrices de coeficientes, incrementando de forma considerable el coste computacional. El valor de sus índices de eficiencia y eficiencia computacional es:

$$I_J = 4^{1/(2n^2+n)}, \qquad IC_J = 4^{1/\left(\frac{2n^3}{3}+5n^2+\frac{n}{3}\right)}.$$

Por último, al realizar la composición del método de Jarratt con una variante de Newton, obtenemos el esquema:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \frac{1}{2}[3F'(y^{(k)}) - F'(x^{(k)})]^{-1}\left(3F'(y^{(k)}) + F'(x^{(k)})\right)[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - [aF'(x^{(k)}) + bF'(y^{(k)})]^{-1}F(z^{(k)}), \qquad k = 0, 1, 2, \ldots, \end{aligned}$$

denotado por **método RN** y cuyo orden de convergencia se establece en el Teorema 3.

Teorema 3

Sea $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ una función suficientemente diferenciable en un entorno abierto D de la solución α del sistema no lineal $F(X) = 0$. Suponga-mos que $F'(X)$ es continua y no singular en α. Entonces, la sucesión de iterados $\{ x^{(k)}\}_{k\geq 0}$ generada por el método RN converge a α con orden de convergencia 5 en los esquemas que verifican $a + b = 1$. Además, el método con $a = -\frac{1}{2}$ y $b = \frac{3}{2}$ tiene orden de convergencia 6.

10.4 Implementación en Matlab

Para implementar los métodos iterativos de resolución de sistemas de ecuaciones no lineales en Matlab, el esqueleto de los programas seguirá las mismas pautas que en el tema anterior. No obstante, hay algunas particularidades que debemos tener en cuenta. En primer lugar, modificaremos la forma de introducir el sistema de ecuaciones no lineales F en Matlab.

Para ello, deberemos conocer si el método iterativo que se va a aplicar requiere del valor de la función en un punto y/o de la evaluación de su matriz Jacobiana. Por tanto, generaremos un archivo .m con el nombre de la función, dándole como parámetro de entrada un vector con el punto sobre el que se quiere evaluar, y como parámetros de salida los valores que requiera el método iterativo.

Ejemplo 3.

Escribamos un archivo .m para definir el sistema de ecuaciones no lineales

$$\left.\begin{array}{r} e^x e^y + x \cdot \cos(y) = 0 \\ x + y = 1 \end{array}\right\}$$

suponiendo que el método iterativo que lo resuelve realiza evaluaciones funcionales de la funicón F y de su derivada F'.

En primer lugar, escribimos el sistema con el formato $F(X) = 0$:

$$F(X) = 0 \Leftrightarrow \begin{bmatrix} f_1(X) \\ f_2(X) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} e^x e^y + x \cdot \cos(y) \\ x + y - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

A continuación, calculamos la matriz Jacobiana del sistema:

$$J_F(X) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} e^x e^y + \cos(y) & e^x e^y - x \cdot \sin(y) \\ 1 & 1 \end{bmatrix}$$

Mostramos a continuación la función `Sistema.m` que implementa el sistema no

lineal con Matlab:

```
function [F,dF] = Sistema(X)
    x=X(1); y=X(2);
    F=[exp(x).*exp(y)+x.*cos(y); x+y-1];
    dF=[exp(x).*exp(y)+cos(y), ...
       exp(x).*exp(y)-x.*sin(y); 1, 1];
end
```

También tendremos que redefinir las condiciones de parada en el proceso de aproximación a la solución del problema. De nuevo, pueden ser 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.

Decimos que ϵ es la tolerancia que determina la exactitud en las aproximaciones a la solución del problema.

La siguiente función de Matlab, denominada `Newton_Sist.m`, implementa el método de Newton para resolver sistemas de ecuaciones. El criterio de parada del proceso iterativo es el cumplimiento de una de las siguientes condiciones:

$$||x^{(k+1)} - x^{(k)}|| < tol, \qquad ||F(x^{(k+1)})|| < tol, \qquad iter > maxit,$$

donde tol y $maxit$ son, respectivamente, la toleracia y el número máximo de iteraciones fijados previamente a la ejecución del método.

Newton_Sist.m

```
function [sol,iter,ACOC] = Newton_Sist(F,x0,tol,maxit)
digits(200);
% Inicializacion de las variables
iter=1;
incre1=tol+1;
incre2=tol+1;
x0=x0(:);
[Fx,dFx]=feval(F,x0);

% Criterio de parada
while incre1>tol && incre2>tol && iter<maxit
    % Expresion del metodo de Newton
    x=x0-dFx\Fx;

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

    % Actualizacion de la estimacion inicial
    x0=x;

    [Fx,dFx]=feval(F,x0);
    incre2=norm(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
```

Podemos observar en `Newton_Sist.m` que, por tratarse de un método para funciones vectoriales, los cocientes con derivadas en las iteraciones del método de Newton se han reemplazado por inversas de matrices Jacobianas y se utiliza en su lugar `dFx \ Fx` resolviendo en cada iteración un sistema lineal con la matriz Jacobiana como matriz de coeficientes del sistema. También nos debemos asegurar que el vector de entrada es un vector columna para que no haya errores debido a las dimensiones.

Ejemplo 4.

Utilicemos la función `Newton_Sist.m` para resolver con el método de Newton el sistema de ecuaciones no lineales

$$\left.\begin{aligned} e^x e^y + x \cdot \cos(y) = 0 \\ x + y = 1 \end{aligned}\right\}$$

tomando como estimación inicial $x^{(0)} = [2, -1]^t$, una tolerancia de 10^{-20} y un máximo de 40 iteraciones.

La función que define el problema no lineal se ha implementado en el Ejemplo 3, de modo que la instrucción que ejecutaríamos en la consola de Matlab es la siguiente:

```
>> [sol,iter,ACOC] = ...
   Newton_Sist('Sistema',vpa([2;-1]),1e-20,40);
```

Mostramos los resultados obtenidos en cada iteración en la Tabla 1.

iter	$x^{(k)}$	$\|\|F(x^{(k+1)})\|\|$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	ACOC
1	[5.3247, -4.3247]	0.7051	4.7018	–
2	[5.1697,-4.1697]	0.0486	0.2191	–
3	[5.1573, -4.1573]	0.0003	0.0176	0.8229
4	[5.1572,-4.1572]	1.7091e–8	0.0001	1.9620
5	[5.1572,-4.1572]	4.3406e–17	6.2690e–9	1.9989
6	[5.1572,-4.1572]	2.7997e–34	1.5921e–17	2.0000

Tabla 1: Resultados numéricos del método de Newton

Aunque aparentemente desde la iteración 4 la solución aproximada $x^{(k)}$ no varía, todavía no se ha obtenido con una tolerancia de 10^{-20}, motivo por el cual se siguen realizando iteraciones hasta que en la teración 6 se alcanza este criterio. En la última columna observamos cómo se va estabilizando el orden de convergencia computacional aproximado en torno al valor teórico de 2.

La función `Newton_Sist.m` muestra la estructura general que seguría la implementación de cualquiera de los métodos iterativos que se han estudiado en este tema. No obstante, en los esquemas multipaso, como el método de Jarratt, hay algunas consideraciones a tener en cuenta. En el siguiente vídeo implementamos paso a paso este método.

Accede al vídeo: Implementación del método de Jarratt.

10.5 Comparativa numérica

En la resolución de sistemas de ecuaciones no lineales, comparamos el funcionamiento de distintos métodos iterativos por medio de experimentos numéricos. Para ello, se

estudian las características de cada uno de los métodos estimadas de forma numérica en la resolución de sistemas no lineales de prueba y con las mismas condiciones de parada. Los parámetros que se analizan frecuentemente son los siguientes:

- El valor de la solución: $x^{(k+1)}$.
- El número de iteraciones necesario para converger: `iter`.
- La norma de la función evaluada en el último iterado: $||F(x^{(k+1)})||$
- La distancia entre los dos últimos iterados: $||x^{(k+1)} - x^{(k)}||$
- El orden de convergencia computacional aproximado: `ACOC`

En ocasiones, la precisión que por defecto que aporta Matlab es insuficiente, de modo que se trabaja con aritmética de precisión variable. Para ello, basta con incluir al inicio del código de la función el comando `digits(200)` si, por ejemplo, queremos trabajar con 200 dígitos, e introducir la estimación inicial $x^{(0)}$ con artimética de precisión variable: `vpa(x0)`.

A continuación, aproximaremos la raíz de los siguientes sistemas no lineales:

- $F_1(x, y) = (e^x e^y + x \cdot \cos(x), x + y - 1)$, $\alpha \approx [5.15723, -4.15723]$.
- $F_2(x, y, z) = \left(\cos(y) - \sin(x), z^x - \frac{1}{y}, e^x - z^2\right)$,
  $\alpha \approx [0.909569, 0.661227, 1.57583]$.
- $F_3(x, y, z, t) = (yz + t(y + z), xz + t(x + z), xy + t(x + y), xy + xz + yz - 1)$,
  $\alpha \approx [0.57735, 0.57735, 0.57735, -0.288675]$.

Compararemos el funcionamiento de los métodos de Newton, Trapecios, Golden Ratio, NA, Jarratt y RN con $a = -\frac{1}{2}$ y $b = \frac{3}{2}$. Se han implementado los métodos con Matlab utilizando aritmética de precisión variable con 200 dígitos y criterio de parada

$$||x^{(k+1)} - x^{(k)}|| + ||F(x^{(k+1)})|| < 10^{-12}$$

con un máximo de 40 iteraciones. En las Tablas 2, 3 y 4 se muestran los resultados obtenidos.

En primer lugar, se resumen en la Tabla 2 los resultados obtenidos en la resolución del sistema $F_1(X) = 0$ tomando como estimación inicial en todos los métodos el vector $x^{(0)} = [2, -1]$.

Método	iter	$\|\|F(x^{(k+1)})\|\|$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	ACOC
Newton	5	4.3406e–17	6.2690e–9	1.9989
Trapecios	9	1.9040e–64	6.7281e–22	2.9993
Golden Ratio	7	8.2086e–43	2.4788e–27	2.1867
NA	5	9.5523e–87	5.7738e–36	3.4151
Jarratt	6	2.8540e–195	6.1911e–49	3.9985
RN	4	2.6765e–141	1.1130e–23	6.4561

Tabla 2: Resultados numéricos para $F_1(X)$ con $x^{(0)} = [2, -1]$

Se observa en la última columna de la Tabla 2 que la aproximación computacional del orden de convergencia tiende al valor teórico, aunque en algunos métodos como por ejemplo Golden Ratio, la aproximación no está tan próxima a su orden cúbico calculado de forma teórica. Gracias al uso de la aritmética de precisión variable, podemos obtener resultados precisos con los 200 dígitos que hemos utilizado. Asimismo, podemos comprobar en la columna $||F(x^{(k+1)})||$ que la estimación a la raíz se logra en todos los casos con mucha precisión. En cuanto al número de iteraciones, el método RN es el que requiere de un menor número, coincidiendo con el esquema iterativo de esta comparativa numérica que tiene un orden de convergencia más alto.

Por otro lado, para aproximar las raíces de la función vectorial no lineal $F_2(X)$, la estimación inicial a la raíz ha sido $x^{(0)} = [1, 1, 2]$. Los resultados obtenidos con cada método se observan en la Tabla 3.

Método	iter	$\|\|F(x^{(k+1)})\|\|$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	ACOC
Newton	6	5.7716e–17	7.5973e–9	1.9760
Trapecios	6	6.1613e–57	1.4405e–19	2.9999
Golden Ratio	6	1.0510e–41	1.2430e–25	2.7407
NA	6	5.4870e–92	1.4939e–38	3.2701
Jarratt	4	5.0114e–77	6.9430e–20	3.9638
RN	4	7.7869e–208	1.1636e–41	6.0053

Tabla 3: Resultados numéricos para $F_2(X)$ con $x^{(0)} = [1, 1, 2]$

Podemos observar en la Tabla 3 cómo los métodos de Jarratt y RN son los que menos iteraciones han necesitado para converger a la solución. Además, son los esquemas iterativos que mayor orden de convergencia computacional tienen. En este caso, el método que mejores prestaciones obtiene con diferencia es el método RN, ya que no solo el número de iteraciones es menor, sino la precisión en la aproximación a la raíz es mucho mayor que en las demás clases iterativas.

En cuanto a la resolución de $F_3(X) = 0$, podemos visualizar en la Tabla 4 un comportamiento similiar a los ejemplos anteriores. Tomando como estimación inicial a la raíz el vector $x^{(0)} = [1, 1, 1, 1]$, todas clases requieren de muy pocas iteraciones para converger a ésta, e incluso en el método RN solo se realizan 3 iteraciones, motivo por el cual su ACOC no se ha estabilizado en torno a su orden teórico de 6.

Método	iter	$\|\|F(x^{(k+1)})\|\|$	$\|\|x^{(k+1)} - x^{(k)}\|\|$	ACOC
Newton	5	9.5736e–17	3.3513e–8	2.1557
Trapecios	4	2.2851e–44	2.9035e–14	3.3125
Golden Ratio	5	3.7097e–43	1.5151e–27	2.0071
NA	4	2.2491e–69	1.9625e–26	4.3854
Jarratt	4	5.859e–144	4.7574e–35	4.2916
RN	3	9.0469e–110	1.8928e–17	7.00325

Tabla 4: Resultados numéricos para $F_3(X)$ con $x^{(0)} = [1, 1, 1, 1]$